

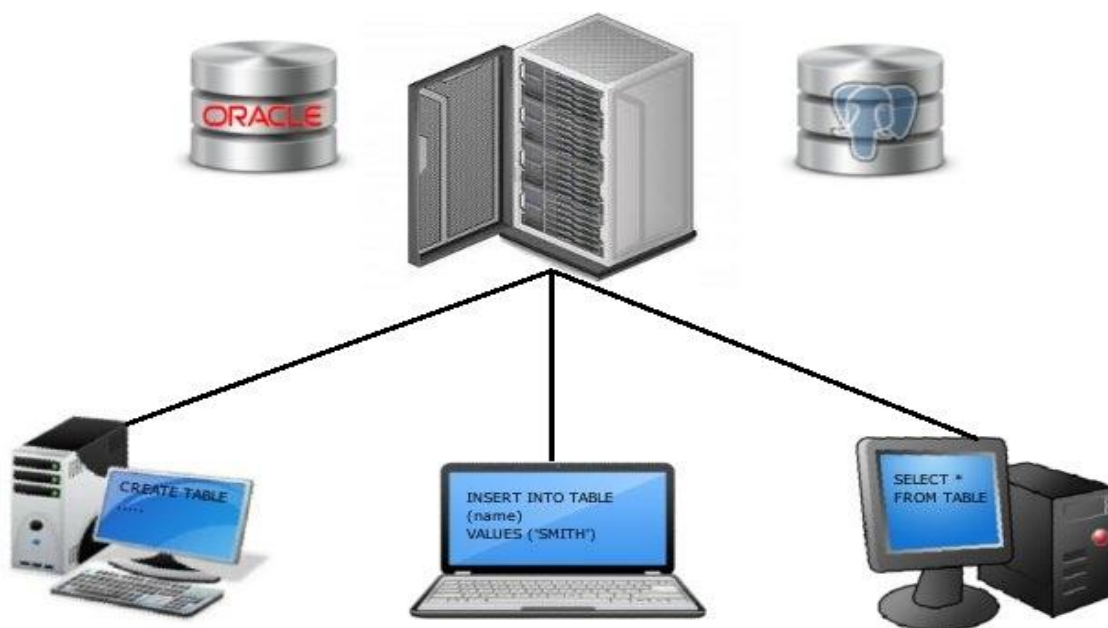


ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι.
ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ
ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.



- ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ -

**«Δημιουργία εκπαιδευτικού περιβάλλοντος με
χρήση ORACLE και δημιουργία Portal για
παρουσίαση και χρήση του υλικού που μας
προσφέρει το Advanced Computer Science
Program»**



Αλαγκιοζίδης Ελευθέριος - Πέτρος
Αρ. Μητρώου: 07/3252

Επιβλέπων Καθηγητής:
Κεραμόπουλος Ευκλείδης

ΘΕΣΣΑΛΟΝΙΚΗ 2013

Αφιερωμένο στην μνήμη του παππού μου Λευτέρη,
στους γονείς μου
και στους φίλους μου που μοναδικά
με στηρίζουν και μου δίνουν την δύναμη
να κάνω το επόμενο βήμα με σιγουριά.

Περιεχόμενα

Πρόλογος.....	7
Περίληψη	9
Abstract.....	10
ΚΕΦΑΛΑΙΟ 1 – ΕΙΣΑΓΩΓΗ	11
1.1 – Επεξήγηση Δομής Και Περιεχομένου	12
1.2 – Το σύστημα Διαχείρισης Βάσεων Δεδομένων Oracle.....	14
1.2.1 – Τι προσφέρει	14
1.2.2 – Τι δυνατότητες μου παρέχει;	16
1.2.3 – Ιδιαίτερα χαρακτηριστικά	17
Ολοκληρώνοντας.....	18
ΚΕΦΑΛΑΙΟ 2 – ΠΡΟΕΤΟΙΜΑΣΙΑ ΤΟΥ SERVER.....	19
2.1 - Προετοιμασία Server για απομακρυσμένες συνδέσεις	19
2.1.1 - Απομακρυσμένη Διαχείριση.....	19
2.2 – Δημιουργία λογαριασμού Oracle και απόκτηση της Oracle 11g	27
2.2.1 Δημιουργία Oracle Account.....	28
2.2.2 – Απόκτηση της Oracle 11g	29
Ολοκληρώνοντας.....	30
ΚΕΦΑΛΑΙΟ 3 – ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ ORACLE 11g.....	31
3.1 – Προετοιμασία του συστήματος.....	32
3.1.1 – Εγκατάσταση Εξαρτήσεων.....	32
3.1.2 – Δημιουργία Symbolic Links	33
3.1.3 – Δημιουργία Χρηστών και Ομάδων	34
3.1.4 – Δημιουργία δομής καταλόγων της Oracle	35
3.1.5 – Μεταβολή παραμέτρων πυρήνα και συστήματος.....	36
3.1.6 – Δημιουργία Μεταβλητών περιβάλλοντος.....	38
3.2 – Εγκατάσταση Oracle 11g.....	39
Ολοκληρώνοντας.....	57
Βιβλιογραφία Κεφαλαίου.....	58
ΚΕΦΑΛΑΙΟ 4 – ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΤΟΥ SERVER.....	59
4.1 - Δημιουργία μεταβλητών περιβάλλοντος.....	59
4.2 - Δημιουργία Listeners και Name Services.....	64
4.2.1 - Listener και Service ID (SID)	73

4.2.2 - Εκκίνηση του Listener	74
4.2.3 – Δοκιμή του Listener	75
4.2.4 - Τερματισμός του Listener	76
4.3 - Δημιουργία Βάσης Δεδομένων	76
4.3.1 - Εκκίνηση της Βάσης Δεδομένων	95
4.3.2 - Δοκιμή της βάσης Δεδομένων	100
4.3.3 - Δημιουργία περισσότερων βάσεων	101
4.3.4 - Τερματισμός Βάσης Δεδομένων	102
Ολοκληρώνοντας.....	103
Βιβλιογραφία Κεφαλαίου.....	104
ΚΕΦΑΛΑΙΟ 5 – ΔΙΑΧΕΙΡΙΣΗ SERVER ΚΑΙ ΠΡΟΣΘΕΤΑ ΕΡΓΑΛΕΙΑ	106
5.1 - Οι χρήστες της βάσης δεδομένων	106
5.1.1 - Δημιουργία Χρηστών στην βάση και απόδοση δικαιωμάτων	108
5.1.2 - Ρόλοι Χρηστών	112
5.1.3 - Προβολή ρόλων και Δικαιωμάτων.....	118
5.1.4 - Tablespace.....	120
5.1.5 – Σύνδεση στην βάση.....	139
5.1.6 – Αλληλεπίδραση με την βάση δεδομένων	140
5.2 – Εργαλεία Διαχείρισης Βάσης Δεδομένων	147
5.2.1 – Η Εφαρμογή SQL*PLUS	148
5.2.2 – Το Περιβάλλον Oracle Enterprise Manager (EM)	157
5.2.3 – Το περιβάλλον Oracle SQL Developer.....	169
5.2.4 – Το περιβάλλον Oracle Application Express (APEX)	175
5.3 – Λοιπές Δυνατότητες Διαχείρισης	197
5.3.1 – Εκτέλεση μεθόδων Java μέσα από την PL/SQL	197
5.3.2 – Δημιουργία δεικτών	201
5.3.3 – Κατάτμηση Πινάκων	203
5.3.4 – Data Dictionary	208
5.3.5 – Ο πίνακας DUAL.....	210
5.3.6 – Προβολή Metadata.....	212
Ολοκληρώνοντας.....	215
Βιβλιογραφία Κεφαλαίου.....	216
ΚΕΦΑΛΑΙΟ 6 – ΣΥΝΤΗΡΗΣΗ ΚΑΙ ΑΣΦΑΛΕΙΑ	218
6.1 – Διαθεσιμότητα Βάσεων Δεδομένων	219

6.1.1 – Δημιουργία Backup και Restore	219
6.1.2 – Virtual Private Database.....	226
6.1.3 – Κλείδωμα Tablespaces.....	232
6.1.4 –Flashback Queries.....	232
6.1.5 – Εξαγωγή Πινάκων Βάσης Σε XML	235
6.1.6 – Εξαγωγή Βάσης σε Dump Αρχείο	241
6.1.7 – Εισαγωγή Βάσης από Dump Αρχείο.....	244
6.2 – Λοιπές Εργασίες	245
6.2.1 – Ανάκτηση Password Διαχειριστών και Χρηστών.....	245
6.2.2 – Απενεργοποίηση χρηστών.....	247
Ολοκληρώνοντας.....	247
Βιβλιογραφία Κεφαλαίου.....	248
ΚΕΦΑΛΑΙΟ 7 – ΣΥΝΔΕΣΗ ΚΑΙ ΕΡΓΑΣΙΑ ΧΡΗΣΤΩΝ	249
7.1 – Client Εφαρμογές και Περιβάλλοντα	249
7.1.1 – Το Oracle SQL Developer.....	250
7.1.2 – Το Oracle Application Express (APEX)	271
7.2 – Προγραμματισμός εφαρμογών αλληλεπίδρασης με την Oracle Database.....	303
7.2.1 – Εφαρμογές σε γλώσσα Java (JDBC).....	303
7.2.3 - Εφαρμογές σε γλώσσα C++ (OCI).....	314
Ολοκληρώνοντας.....	320
Βιβλιογραφία Κεφαλαίου.....	320
ΚΕΦΑΛΑΙΟ 8 – Η ORACLE ACADEMY ΚΑΙ ΤΟ ΠΡΟΓΡΑΜΜΑ ADVANCED COMPUTER SCIENCE.....	322
8.1 – Η Υπηρεσία Oracle Academy	323
8.1.1 – Τι είναι η Oracle Academy;.....	323
8.1.2 – Τι παρέχει η εν λόγω υπηρεσία;	324
8.1.3 – Σύμβαση φορέα και ρόλοι συμμετεχόντων	325
8.1.4 – Εκπαιδευτικά προγράμματα.....	327
8.1.5 – Οφέλη της Oracle Academy.....	329
8.1.6 – Η Oracle Academy στον Ελλαδικό χώρο	330
8.2 – Το πρόγραμμα Advanced Computer Science – A Closer Look	332
8.2.1 – Χαρακτηριστικά.....	332
8.2.2 – Μαθήματα που παρέχονται.....	333
8.2.3 – Εμπορικό Λογισμικό.....	341

8.2.4 – Δωρεάν και ανοιχτού κώδικα λογισμικό.....	342
Ολοκληρώνοντας.....	342
Βιβλιογραφία Κεφαλαίου.....	343
ΚΕΦΑΛΑΙΟ 9 – Ο SERVER ΤΟΥ ΤΜΗΜΑΤΟΣ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΟΥ ΑΤΕΙ ΘΕΣΣΑΛΟΝΙΚΗΣ.....	345
6.1 – Περιγραφή του Server.....	345
6.1.1 – Διεύθυνση και πρόσβαση.....	345
6.1.2 – Σύσταση Hardware.....	346
6.1.3 – Λογισμικό.....	346
6.2 – Υπηρεσίες του Server	347
6.2.1 – Υπηρεσίες Βάσεων Δεδομένων.....	347
6.2.2 – Υπηρεσίες Web.....	350
Ολοκληρώνοντας.....	351
ΕΠΙΛΟΓΟΣ	352
Παράρτημα Α΄ – Εγκατάσταση και Παραμετροποίηση PostgreSQL 9.1 σε Debian 7 Wheezy x64.....	355
Παράρτημα Β΄ – Απόκτηση και εγκατάσταση Oracle SQL Developer	364
Παράρτημα Γ΄ – Εγκατάσταση οδηγού JDBC στον NetBeans IDE	368
Παράρτημα Δ΄ – Κώδικας Ενδεικτικής Εφαρμογής JDBC	372
Παράρτημα Ε΄ – Εγκατάσταση Οδηγού OCCI στο Microsoft Visual Studio 2010	376
Παράρτημα ΣΤ΄ – Κώδικας Ενδεικτικής Εφαρμογής OCCI.....	383
Παράρτημα Ζ΄ – Εγκατάσταση και Παραμετροποίηση XAMPP και MySQL.....	386
Παράρτημα Η΄ – Εγκατάσταση του E107 Website System	391
Παράρτημα Θ΄ – Script δημιουργίας / διαγραφής χρηστών Oracle και αρχείου αναφοράς	397
Παράρτημα Ι΄ – Script Δημιουργίας / Διαγραφής Χρηστών του Oracle APEX.....	401
Παράρτημα ΙΑ΄ - Script Συσχέτισης Schema με ένα APEX Workspace.....	405
Παράρτημα ΙΒ΄ - Script Δημιουργίας χρηστών για την PostgreSQL.....	409

Πρόλογος

Ο τομέας των βάσεων δεδομένων στις μέρες αναντίλεκτα έχει μεγάλη χρήση σε μία διόλου ευκαταφρόνητη μερίδα εφαρμογών. Αν αναλογιστεί βέβαια κανείς την τεράστια εξέλιξη των Συστημάτων Διαχείρισης Βάσεων Δεδομένων, όσον αφορά τις δυνατότητες και τις ευκολίες που παρέχουν, τότε σίγουρα μπορεί να καταλάβει τον λόγο της ραγδαίας εξάπλωσης σε ολόένα και περισσότερα συστήματα, όπως για παράδειγμα στον τομέα της υγείας με το σύστημα της ηλεκτρονικής συνταγογράφησης.

Η ορθή χρήση όμως ενός τέτοιου συστήματος, απαιτεί καλή γνώση από πλευράς του χρήστη. Όπως γίνεται εύκολα αντιληπτό, σε συστήματα με μερικές χιλιάδες εγγραφές δεδομένων, ο παράγοντας λάθους ειδικά αν προέρχεται από ελλιπή εκπαίδευση και κακή γνώση, μπορεί να επιφέρει τεράστιες καταστροφές. Όλα αυτά όμως μπορούν να αποφευχθούν αν η εκπαίδευση αναφορικά με τη λειτουργία ενός τέτοιου συστήματος γίνει με μεθοδικό και πληρέστερο τρόπο, ωστόσο όμως αυτό από μόνο του δεν φτάνει. Το ίδιο το σύστημα διαχείρισης θα πρέπει να είναι εύκολο στην χρήση και πάνω από όλα λειτουργικό και αποδοτικό. Και κάτι ακόμα ποιο σημαντικό. Να είναι αξιόπιστο. Θα πρέπει τα εργαλεία χρήσης του, η φιλοσοφία του, οι δυνατότητές του, η φιλικότητα χρήσης του και πάνω από όλα η αποτελεσματικότητά του να είναι σε υψηλά επίπεδα. Δύσχρηστα συστήματα με περιορισμένες δυνατότητες και ελάχιστη παραγωγικότητα σίγουρα δεν είναι κάτι που θέλει κανένας.

Ένα τέτοιο σύστημα λοιπόν που παρέχει αξιοπιστία, ευκολία χρήσης, πληθώρα εργαλείων και ατέλειωτες δυνατότητες είναι το ORACLE Database. Η εργασία σε αυτό το σύστημα μόνο θετικές εντυπώσεις έχει να δημιουργήσει, κυρίως επειδή είναι πανίσχυρο και καλύπτει και τον πιο απαιτητικό χρήστη. Και το ακόμα ποιο αξιοσημείωτο σε αυτό είναι ότι είναι απόλυτα παραγωγικό και αρκετά εύκολο στην εκμάθηση. Γι αυτό ακριβώς τον λόγο επιλέχτηκε να διδαχτεί στο τμήμα Μηχανικών Πληροφορικής του ΑΤΕΙ Θεσσαλονίκης. Μέσω τις μεθοδικής διδασκαλίας του, μέσω των εργαλείων χρήσης του που αντικαθιστούν παλαιότερα και δύσχρηστα συστήματα και μέσω της καθοδήγησης από τους επαγγελματίες διδάσκοντες, ο προπτυχιακός ή μεταπτυχιακός φοιτητής θα αποκτήσει ισχυρές

γνώσεις οι οποίες θα τον οδηγήσουν με σιγουριά στην επαγγελματική επιτυχία. Εκεί ακριβώς αποσκοπεί το παρόν σύγγραμμα.

Σε αυτήν την πτυχιακή εργασία λοιπόν, μαζί θα δούμε σε βάθος το σύστημα ORACLE Database, θα αναλύσουμε τις δυνατότητες και τα εργαλεία που το συνοδεύουν, θα προβληματιστούμε μαζί μέσω παραδειγμάτων χρήσης και θα βγάλουμε λογικά συμπεράσματα για το αν είναι αξιόλογο σύστημα ή όχι. Προσδοκάται ότι ο αναγνώστης θα αποκομίσει σεβαστό αριθμό γνώσεων σχετικά με το πώς θα πρέπει να εργαστεί σε αυτό το σύστημα, γνώσεις οι οποίες θα του εξασφαλίσουν σιγουριά για να ανταπεξέλθει στις σημερινές απαιτήσεις. Άλλωστε η γνώση είναι ένας θησαυρός που δεν αγοράζεται.

Σας εύχομαι λοιπόν καλή ανάγνωση και εύχομαι το παρόν σύγγραμμα να αποτελέσει το όπλο με το οποίο θα κατακτήσετε τον κόσμο της ORACLE και να ισχυροποιήσετε την σιγουριά σας για το επόμενο βήμα.

Αλαγκιοζίδης Ελευθέριος – Πέτρος

Περίληψη

Η εν λόγω πτυχιακή εργασία επικεντρώνεται κυρίως στο σύστημα διαχείρισης βάσεων δεδομένων Oracle Database, όπου θα αναλυθεί ενδελεχώς και σε βάθος καθώς επίσης και των βασικών εργαλείων που το συνοδεύουν. Η κάλυψη του εν λόγω θέματος θα γίνει βαθμιδωτά ξεκινώντας αρχικά από τον οδηγό εγκατάστασης του συστήματος και στην συνέχεια θα επεκτείνεται σε ειδικότερα θέματα. Θεωρείται σκόπιμο να αναλυθούν οι δυνατότητες και τα εργαλεία χρήσης αυτού του συστήματος τόσο για τους απλούς χρήστες τόσο και για τους διαχειριστές, ώστε να καλυφθούν και οι δύο τομείς χρηστών με όσο το δυνατόν πληρέστερο τρόπο. Το συγκεκριμένο σύστημα έχει εγκατασταθεί σε εσωτερικό εξυπηρετητή (server) του τμήματος Μηχανικών Πληροφορικής του ΑΤΕΙ Θεσσαλονίκης και θα χρησιμοποιηθεί για την διδασχή μαθημάτων σχετικών με βάσεις δεδομένων σε προπτυχιακούς και μεταπτυχιακούς φοιτητές. Ως εκ τούτου με το πέρας της περιγραφής και ανάλυσης του συστήματος θα αφιερώσουμε και λίγο χρόνο στην εφαρμογή των γνώσεων που αποκτηθήκαν σε αυτόν. Επίσης στον ίδιο εξυπηρετητή, έχει εγκατασταθεί διαδικτυακή πύλη (Portal) όπου θα περιγράφονται οι δυνατότητες που παρέχονται από το πρόγραμμα Advanced Computer Science του φορέα Oracle Academy όπου μπορεί κάποιος να αποκτήσει ισχυρή πιστοποίηση από την Oracle, που θα του προσφέρει νέες ευκαιρίες στην επαγγελματική του καριέρα.

Abstract

This thesis presentation mainly focuses on the ORACLE Database Management System which will be analyzed thoroughly along with the tools that it provides. The coverage of this subject is performed gradually starting from the System Installation Guide and it progresses into more specific subjects. It is considered more appropriate to analyze the system's capabilities to both parties which are the plain users and the administrators, in order to accomplish the most effective and complete subject coverage. This particular system has been installed on an internal Database Server at The Department of Computer Engineers of the Alexander Technological Institute of Thessaloniki and will be used for teaching lessons relative to databases. These lessons will be taught to undergraduate and postgraduate students, so after we finish the system's description and analysis, we will devote some time to implement the acquired knowledge to that server. Meanwhile, on the same server, a web Portal will be installed which will contain information about the Advanced Computer Science program offered by Oracle Academy, from which someone can acquire a globally recognized certification from Oracle.

ΚΕΦΑΛΑΙΟ 1 – ΕΙΣΑΓΩΓΗ

Η παρούσα πτυχιακή εργασία αφορά την δημιουργία εκπαιδευτικής πλατφόρμας με χρήση της Oracle Database και δημιουργία Portal για παρουσίαση και χρήση του υλικού που μας προσφέρει το Advanced Computer Science Program. Ως εκ τούτου, επειδή ο σκοπός της είναι εκπαιδευτικός και μαθησιακός, υιοθετήθηκε η συγγραφή της σε στυλ οδηγού χρήσης (user's manual). Μέσα από ένα φιλικό και με σεβασμό στον αναγνώστη ύφος, θα δούμε μαζί βήμα-προς-βήμα πως μπορείτε να εγκαταστήσετε έναν ισχυρό και αποδοτικό Εξυπηρετητή (Server) που θα τρέχει το επίσης ισχυρό Oracle ORDBMS, από την αρχή ως το τέλος. Καλύπτουμε πλήρως τα βήματα προετοιμασίας, εγκατάστασης και παραμετροποίησης του Server σας, ενώ καλύπτουμε διεξοδικά και τις δύο ομάδες χρηστών, τους διαχειριστές (administrators) και τους απλούς χρήστης/πελάτες (clients). Θα δούμε μαζί τις δυνατότητες που μας παρέχει το εν λόγω σύστημα, στην πράξη φυσικά μέσα από αναλυτικές οδηγίες με εικόνες και παραδείγματα και προσδοκείται στο τέλος της ανάγνωσης του παρόντος συγγράμματος να έχετε αποκομίσει χρήσιμες γνώσεις που πολύ πιθανών να σας φανούν χρήσιμες στην αυριανή σας σταδιοδρομία.

Η Oracle Database θα μας απασχολήσει στο μεγαλύτερο μέρος του παρόντος συγγράμματος, καθώς γίνεται μια προσπάθεια να καλυφτεί όσο το δυνατόν μεγαλύτερο μέρος των δυνατοτήτων της και ως εκ τούτου επιλέχτηκαν τα περισσότερο χρήσιμα και βασικότερα εξ αυτών προκειμένου να επιτύχουμε το ζητούμενο. Το υπόλοιπο μέρος της είναι αφιερωμένο στην περιγραφή και λειτουργία του εξυπηρετητή του τμήματος Μηχανικών Πληροφορικής του ΑΤΕΙ Θεσσαλονίκης, όπου έχουν εφαρμοστεί όλα θα δούμε στα προσεχή κεφάλαια. Επίσης θα επεκταθούμε και στο πρόγραμμα Advanced Computer Science της Oracle Academy. Θα δούμε μαζί για ποιο λόγο επιλέχτηκε να διδαχτεί στο Τμήμα Μηχανικών Πληροφορικής και γιατί είναι τόσο αξιόλογο και αξιόπιστο DBMS.

1.1 – Επεξήγηση Δομής Και Περιεχομένου

Προτού ξεκινήσετε την ανάγνωση του παρών συγγράμματος θα θέλαμε να σας δώσουμε ένα γενικό πλάνο από το τι θα συναντήσετε στα παρακάτω κεφάλαια. Εφόσον το ΣΔΒΔ της Oracle επιλέχτηκε ως εκπαιδευτική πλατφόρμα στο Τμήμα Μηχανικών Πληροφορικής, το ύφος του κειμένου, κυμαίνεται αρκετά σε εκπαιδευτικό περιβάλλον χρησιμοποιώντας ονόματα και παραδείγματα που παραπέμπουν σε εκπαιδευτικό ίδρυμα όπως STUDENTS, PREGRADUATE και λοιπά. Γίνεται μια προσπάθεια να διατηρήσουμε μια ισορροπία μεταξύ εκπαιδευτικού συγγράμματος και τεκμηρίωσης λειτουργιών του συστήματος, οπότε θα συναντήσετε και στοχευόμενα παραδείγματα. Να θυμάστε ότι σκοπός μας είναι να σας δώσουμε να καταλάβετε το σύστημα σε βάθος ώστε να μπορείτε μόνοι σας αργότερα να ακολουθήσετε αυτά που μάθατε, και όχι να σας διδάξουμε γενικευμένα Concepts των βάσεων δεδομένων, όπως την γλώσσα SQL. Τμήματα Κώδικα (Snippets) προφανώς και περιέχουμε αλλά για λόγους επίδειξης. Θεωρείται δεδομένο ότι τα βασικά τουλάχιστον Concepts των Βάσεων Δεδομένων τα γνωρίζετε.

Ποιο αναλυτικά, στο παρών σύγγραμμα θα συναντήσετε:

- 1) **Στιγμιότυπα** (screenshots) από καίρια σημεία που θέλουμε να κατανοήσετε ή να ακολουθήσετε προκειμένου να επιτύχετε το ζητούμενο που αφορά.
- 2) **Σχήματα** για καλύτερη κατανόηση κάποιων βασικών εννοιών. Η οπτικοποίηση τους βοηθάει περισσότερο στην κατανόησή τους.
- 3) **Εντολές για το Shell** του λειτουργικού συστήματος που θα εργαστούμε με την τυποποίηση `«<χρήστης>@debserver <εντολή>»` (debserver είναι το όνομα του server μας) οι οποίες έχουν τοποθετηθεί σε εύκολα διακριτό πλαίσιο. Η τυποποίηση χρησιμεύει για να αναγνωρίζετε κάθε φορά ως τι χρήστης πρέπει να εκτελεστούν οι εντολές που ακολουθούν. Επίσης

περιέχονται και στοχαστικά σχόλια για τις χρήσιμες εντολές καθώς και το output ορισμένων εξ αυτών, σε μια προσπάθεια να προσομοιώσουμε όσο το δυνατόν περισσότερο το Terminal και να διευκολύνουμε την ανάγνωση τους από εσάς.

- 4) **Πλαίσια σήμανσης** που περιέχουν **οδηγητικές συστάσεις**. Πλαίσια με λεζάντα **Προσοχή!** περιέχουν συστάσεις που απαιτούν την **άμεση προσοχή σας** για την αποφυγή προβλημάτων που πολύ πιθανόν να σας ταλαιπωρήσουν. Πλαίσια με λεζάντα **Keep In Mind...** περιέχουν συστάσεις που πρέπει **να λάβετε υπόψη**, δηλαδή την **έμμεση προσοχή σας**. Πλαίσια με λεζάντα **Καλή Πρακτική**, περιέχουν συστάσεις που μπορούν να σας βοηθήσουν προτείνοντάς σας λύσεις που αποτελούν **καλή επιλογή** στο να εφαρμοστούν. Τέλος, πλαίσια με λεζάντα **Ορισμός**, περιέχουν βασικούς **βιβλιογραφικούς ορισμούς** που θεωρούνται απαραίτητοι για να κατανοήσετε το περεταίρω περιεχόμενο μιας ή περισσότερων ενοτήτων. Τα πλαίσια με λεζάντα **Troubleshoot** περιέχουν πληροφορίες για το πώς να αντιμετωπίσετε ένα πιθανό πρόβλημα που ίσως προκύψει.
- 5) **Εντολές SQL και PL/SQL** που συμφωνούν με το περιβάλλον το οποίο παρουσιάζεται την προκείμενη στιγμή. Εφαρμόζεται **τονισμός σύνταξης** (Syntax Highlighting) για δική σας διευκόλυνση στην ανάγνωσή τους.
- 6) **Ενότητες και υποενότητες** που ακολουθούν την παραγωγική συγγραφική μέθοδο, δηλαδή από το γενικό στο ειδικό.
- 7) **Πίνακες με ρεπερτόριο εντολών** που μπορεί ή απαιτείται να χρησιμοποιηθούν για την εκτέλεση μίας ή περισσότερων ενεργειών.
- 8) **Σύνταξη εντολών** με γενικευμένο τρόπο ώστε να καταλάβετε πρώτα πως συντάσσονται σωστά και μετά μέσω παραδείγματος. Τα σύμβολα « <> » δηλώνουν **υποχρεωτική προσθήκη** ενώ τα « {} » **προαιρετική**. Το σύμβολο « | » δηλώνει λογικό Ή (OR), δηλαδή ένα από τα δύο.

Σημαντική παρατήρηση είναι πως οι ενότητες που απαρτίζουν το κάθε κεφάλαιο, ακολουθούν μια σειρά, που απαραίτητα πρέπει να ακολουθήσετε για να εξελιχθεί ομαλά η όλη διαδικασία της εκμάθησης/κατανόησης των χαρακτηριστικών που περιγράφουν. Με αυτήν την προσπάθεια, θέλουμε να περιορίσουμε όσο το δυνατόν περισσότερο «άγνωστες λέξεις» δηλαδή μαθησιακά κενά.

Αφού λοιπόν πήρατε μια ιδέα για το τι θα συναντήσετε παρακάτω, μπορούμε να αρχίσουμε να επικεντρωνόμαστε στο θέμα μας, την Oracle Database.

1.2 – Το σύστημα Διαχείρισης Βάσεων Δεδομένων Oracle

Το Σύστημα **ORACLE Database** είναι ένα ισχυρό **Αντικειμενοσχεσιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων** (Object-Relational Database Management System / ORDBMS) της εταιρίας ORACLE Corporation. Πρόκειται για ένα πανίσχυρο και με πολλές δυνατότητες σύστημα, το οποίο αποτελεί άριστη επιλογή για εφαρμογή και χρήση. Και ειδικά στον χώρο των βάσεων δεδομένων, η Oracle θεωρείται κορωνίδα.

1.2.1 – Τι προσφέρει

Η Oracle Database προσφέρει πληθώρα χαρακτηριστικών που δεν τα συναντάμε συχνά σε άλλα DBMS. Μερικά από αυτά είναι:

- 1) **Αξιοπιστία.** Τα δεδομένα που αποθηκεύονται σε αυτήν είναι ασφαλή από αλλοίωση τροποποίηση χωρίς εξουσιοδότηση του ιδιοκτήτη / διαχειριστή, ακόμα και από ασταθή συμπεριφορά του υπολογιστή στον οποίο βρίσκονται.

- 2) **Διαθεσιμότητα.** Η πρόσβαση στα δεδομένα του χρήστη γίνεται από οπουδήποτε και μάλιστα με ευκολότερο τρόπο (παραδειγμα το Oracle APEX που θα δούμε στην πορεία).
- 3) **Φιλικότητα προς το χρήστη.** Ο χρήστης που εργάζεται σε αυτήν μπορεί εύκολα να περατώσει τις ζητούμενες ενέργειες που θέλει χάρη στην ευκολία των εργαλείων που παρέχει αλλά και στις διευκολύνσεις που παρέχει εσωτερικά στον τρόπο λειτουργίας της.
- 4) **Εύκολη Διαχειρισιμότητα.** Η διαχείριση ενός συστήματος σαν της Oracle, είναι εύκολο στην διαχείριση ακόμα και σε ένα απαιτητικό περιβάλλον με πολλούς χρήστες και δεδομένα. Η ιεραρχική δομή που παρέχει βοηθάει τον διαχειριστή να επικεντρώνεται σε συγκεκριμένες λειτουργίες διαχείρισης χωρίς να «χάνεται στη μετάφραση» με πολύπλοκες διαδικασίες και παραμετροποιήσεις.
- 5) **Ασφάλεια.** Οι χρήστες έχουν ρόλους στο σύστημα με συγκεκριμένα δικαιώματα, οπότε τα δεδομένα τους είναι προστατευμένα από μη εξουσιοδοτημένες παρεμβάσεις.
- 6) **Ευκολία εκμάθησης.** Ένας χρήστης που τώρα αρχίζει να εργάζεται σε αυτό, μπορεί γρήγορα να του «πάρει τον αέρα» μέσω τον σχεδόν αυτοματοποιημένων λειτουργιών που παρέχει (όπως πχ ένα Query Builder ή αυτόματη συμπλήρωση εντολών).
- 7) **Ταχύτητα.** Ο χρήστης μπορεί να βρίσκει τα δεδομένα του σε ελάχιστο χρόνο, ακόμα και από «χαώδεις» πίνακες χάρη στην έξυπνη δομή οργάνωσης και λειτουργίας του συστήματος.
- 8) **Ισχυρή τεκμηρίωση.** Η Oracle παρέχει λεπτομερώς τεκμηρίωση για την κάθε μία λειτουργία και χαρακτηριστικό του συστήματος, με παραδείγματα.

Ειλικρινά, αν κάτσει κανείς να αναλύσει πέρα ως πέρα τις δυνατότητες που προσφέρει η Oracle, είναι αμφίβολο αν θα καταφέρει να τις απαριθμήσει όλες. Αξίζει όμως τον κόπο να την εκθειάσουμε γιατί πραγματικά το αξίζει.

1.2.2 – Τι δυνατότητες μου παρέχει;

Με μία λέξη, άπειρες. Υπερβολή; Καθόλου, δείτε και μόνοι σας τι εννοούμε ακριβώς.

Κατ' αρχάς είναι **ταχύτατη**. Ακόμα και σε μια βάση δεδομένων όπου υπάρχει χάος δεδομένων, ο χρήστης θα βρει αυτά που θέλει να βρει χωρίς καθυστερήσεις. Και αν υπάρχουν πολλοί χρήστες που την χρησιμοποιούν ταυτόχρονα; Κανένα πρόβλημα, εξυπηρετούνται όλοι χωρίς να ευνοείται ή να αδικείται κανένας.

Εν συνεχεία, παρέχει **πρωτοποριακά εργαλεία και περιβάλλοντα**. Δεν περιορίζεται απλά σε μια εφαρμογή με ένα SQL Command Editor όπου γράφετε εντολές, όπως συμβαίνει σε πολλά σύγχρονα DBMS. Το κάθε εργαλείο της είναι κατασκευασμένο με γνώμονα την διευκόλυνση του χρήστη. Εφαρμογές όπως το SQL Developer, δεν έχει τίποτα να ζηλέψει από αντίστοιχες εφαρμογές άλλων DBMS διαφορετικών κατασκευαστών. Αντίθετα, τα τελευταία έχουν. Δεν πρέπει να παραληφθεί πως κάποια από αυτά μπορείτε να τα χρησιμοποιήσετε μέσα από τον πλοηγό ιστού σας (Web Browser) χωρίς να χρειαστεί να εγκαταστήσετε το παραμικρό. Τόσο απλά. Τόσο αποδοτικά.

Θα ήταν μεγάλο λάθος να μην αναφερθεί ο παράγοντας **ασφάλεια και αξιοπιστία**. Τα δεδομένα σας είναι θωρακισμένα από οποιασδήποτε μορφής απειλή είτε από μη εξουσιοδοτημένους χρήστες είτε από εξωτερικές παρεμβάσεις. Ακόμα και αν ο υπολογιστής στον οποίο βρίσκονται η βάση, τερματιστεί απότομα (πχ λόγω πτώσης ρεύματος ή κολλήματος) δεν έχετε τίποτα να φοβάστε. (Σας το εγγυώμαι, προσωπικό βίωμα).

Διόλου ασήμαντο θεωρείται το θέμα της **αποδοτικότητας και λειτουργικότητας**. Ο χρήστης είτε είναι διαχειριστής είτε απλός πελάτης, έχει να κάνει με ένα σύστημα που ανταποκρίνεται άψογα στις ανάγκες του, όσο απαιτητικές και να είναι.

Τέλος υπογραμμίζεται πως η **εσωτερική δομή λειτουργίας** της εγγυάται παραγωγική και χωρίς προβλήματα αλληλεπίδραση με τον χρήστη. Ο τελευταίος μπορεί να «υποτάξει» το σύστημα κατά την δική του εργασιακή απαίτηση (στα

πλαίσια φυσικά που του επιτρέπεται). Άλλωστε αυτός είναι και το επίκεντρο. Όλα γύρω από τον χρήστη.

1.2.3 – Ιδιαίτερα χαρακτηριστικά

Η Oracle Database παρέχει όλα τα σύγχρονα χαρακτηριστικά που παρέχει ένα σύγχρονο DBMS προσθέτοντας όμως επιπλέον από αυτά ώστε να γίνει η εργασία του χρήστη/διαχειριστή με την βάση λιγότερο κουραστική και περισσότερο παραγωγική. Ας δούμε μερικά εξ αυτών.

- 1) **Πλήρες ρεπερτόριο εντολών SQL** με ορισμένες διαφοροποιήσεις σχετικά με την ισχύ τους. Η γλώσσα αυτή μπορεί να χρησιμοποιηθεί είτε από τους διαχειριστές, είτε από τους χρήστες και να προκαλέσουν στην βάση την ενέργεια που επιθυμούν ή να ανακτούν τα δεδομένα που απαιτούν.
- 2) Εισαγωγή της διαδικαστικής γλώσσας **PL/SQL**. Η εν λόγω γλώσσα εισάγει την έννοια του αντικειμενοστραφούς προγραμματισμού μέσα στην βάση δεδομένων. Με την PL/SQL η απλή SQL αποκτά περισσότερες δυνατότητες, και ο χρήστης μπορεί να δημιουργεί ισχυρά υποπρογράμματα και αντικείμενα που μπορούν με μία κίνηση να ανακτήσουν ή να αποθηκεύσουν, για παράδειγμα, τεράστιο αριθμό δεδομένων.
- 3) Αποθήκευση δεδομένων του χρήστη σε **πινακοχώρους** (Tablespaces) και μάλιστα σε παραμετροποιήσιμο μέγεθος, ώστε ο κάθε χρήστης να έχει το χώρο που του αναλογεί, ή τον δικό του, προσωπικό χώρο απόθεσης των αντικειμένων του.
- 4) Σημεία αναιρέσεως και επαναφοράς των αλλαγών που έγιναν στην βάση, ώστε να εγγυάται η ακεραιότητα των δεδομένων του χρήστη.
- 5) **Συναλλαγματικές** βάσεις δεδομένων. Οι αλλαγές στην δομή και το περιεχόμενο της βάσης δεν γίνονται αν ο χρήστης δεν το επιβεβαιώσει. Αν

σε περίπτωση το παραγόμενο αποτέλεσμα δεν ικανοποιεί τον τελευταίο, ή προξένησε ανεπιθύμητα αποτελέσματα, μπορεί να τις ακυρώσει, σαν να μην έγιναν ποτέ.

- 6) Ειδικές λειτουργίες δεδομένων όπως κατάτμηση μεγάλων πινάκων (Partitioning) , δημιουργία Δεικτών δεδομένων (Indexes), που κάνουν ακόμα πιο εύκολη την εξόρυξη δεδομένων που χρειάζεται ο χρήστης.

Ολοκληρώνοντας....

Πολλές από τις έννοιες και Concepts που είδαμε παραπάνω, θα μας απασχολήσουν στα προσεχή κεφάλαια με περισσότερη λεπτομέρεια και πρακτική εφαρμογή. Το μόνο πράγμα που θέλω να σας τονίσω, είναι ότι το παρών σύγγραμμα-οδηγός συγγράφηκε με γνώμονα την ευκολία σας, οπότε σε πολλές ενότητες θα δούμε αναλυτικά βήματα αλλά και προτάσεις του τύπου τι είναι καλή πρακτική να εφαρμόσετε, τι θα πρέπει να αποφύγετε κλπ. Ξεκινάμε λοιπόν να ανακαλύψουμε μαζί τον κόσμο της Oracle μέσα από αυτές τις σελίδες. Καλή αρχή!

ΚΕΦΑΛΑΙΟ 2 – ΠΡΟΕΤΟΙΜΑΣΙΑ ΤΟΥ SERVER

Σε αυτό το κεφάλαιο θα ασχοληθούμε με τα προαπαιτούμενα βήματα που απαιτούνται για την διαχείριση του Debian Server από απόσταση, και την απόκτηση των απαραίτητων αρχείων εγκατάστασης της Oracle.

2.1 - Προετοιμασία Server για απομακρυσμένες συνδέσεις

Σε αυτήν την ενότητα θα ασχοληθούμε με τα απαραίτητα εργαλεία που χρειαζόμαστε προκειμένου να είμαστε σε θέση να διαχειριζόμαστε τον Server από μια απομακρυσμένη τοποθεσία.

2.1.1 - Απομακρυσμένη Διαχείριση

Προφανώς η διαχείριση του Server δεν μπορεί να γίνεται πάντα εκ του σύνεγγυς καθώς στην πλειονότητα των περιπτώσεων θα πρέπει να γίνεται απομακρυσμένα. Γι αυτό το λόγο ο Server μας θα πρέπει να «ακούει» για απομακρυσμένες συνδέσεις.

Δύο είναι τα βασικά πρωτόκολλα απομακρυσμένης διαχείρισης στον απομακρυσμένο Server, το **Πρωτόκολλο Απομακρυσμένης Επιφάνειας Εργασίας** (Remote Desktop Protocol – RDP) και το **Πρωτόκολλο Ασφαλούς Κελύφους** (Secure Shell Protocol – SSH).

2.1.1.1 - Το Πρωτόκολλο RDP

Το πρωτόκολλο RDP επιτρέπει την σύνδεση στην επιφάνεια εργασίας ενός απομακρυσμένου υπολογιστή, όπου ο χρήστης εργάζεται σε αυτόν σαν να ήταν μπροστά του. Η επιφάνεια εργασίας του Server βρίσκεται στην διάθεση του χρήστη όπου μπορεί να εκτελέσει οποιαδήποτε εργασία επιθυμεί. Δεν έχει σημασία αν ο διαχειριστής επιφάνειας εργασίας είναι το KDE ή το Gnome,

υποστηρίζονται όλα. Και επειδή ορισμένες εργασίες απαιτούν παραθυρικό περιβάλλον, το RDP είναι πολύ ικανοποιητική λύση.

2.1.1.1.1 - Εγκατάσταση RDP Server

Για να μπορεί ο Server να δεχτεί συνδέσεις RDP πρέπει να διαθέτει και την κατάλληλη υπηρεσία/εφαρμογή για να τις εξυπηρετήσει. Το λογισμικό που διαλέξαμε για την ικανοποίηση της παραπάνω ανάγκης είναι το XRDP το οποίο συνεργάζεται άψογα με το Debian Linux. Και ακόμα καλύτερα για μας, το περιλαμβάνει ο APT package manager έτοιμο προς εγκατάσταση. Οπότε το εγκαθιστούμε με την παρακάτω:

```
root@dbserver# apt-get install xrdp
```

Με το πέρας της εγκατάστασης πρέπει να ενεργοποιήσουμε την υπηρεσία XRDP οπότε το πραγματοποιούμε με την εντολή (ως root)

```
root@dbserver# service xrdp start
```

Τώρα ο Server μας μπορεί να δεχτεί συνδέσεις RDP από κάποια εφαρμογή RDP Client.

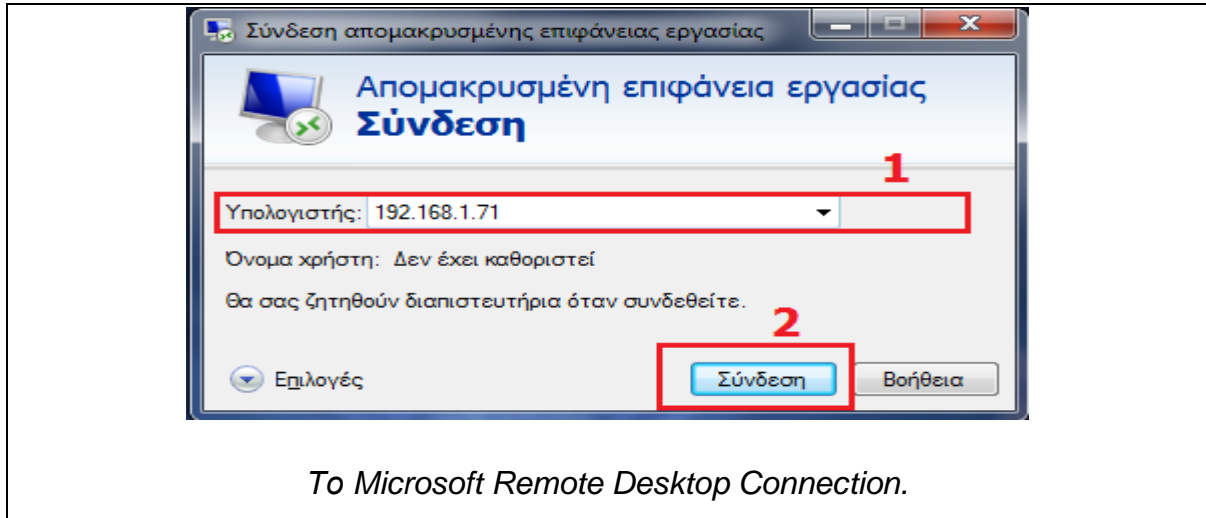
2.1.1.1.2 - Σύνδεση μέσω RDP

Για να συνδεθούμε στον Server μέσω RDP το μόνο που πρέπει να γνωρίζουμε είναι την **IP Διεύθυνσή του** και προφανώς τα **διαπιστευτήριά** μας στο σύστημα. Τα υπόλοιπα θα τα αναλάβει η εφαρμογή RDP Client που θα επιλέξουμε.

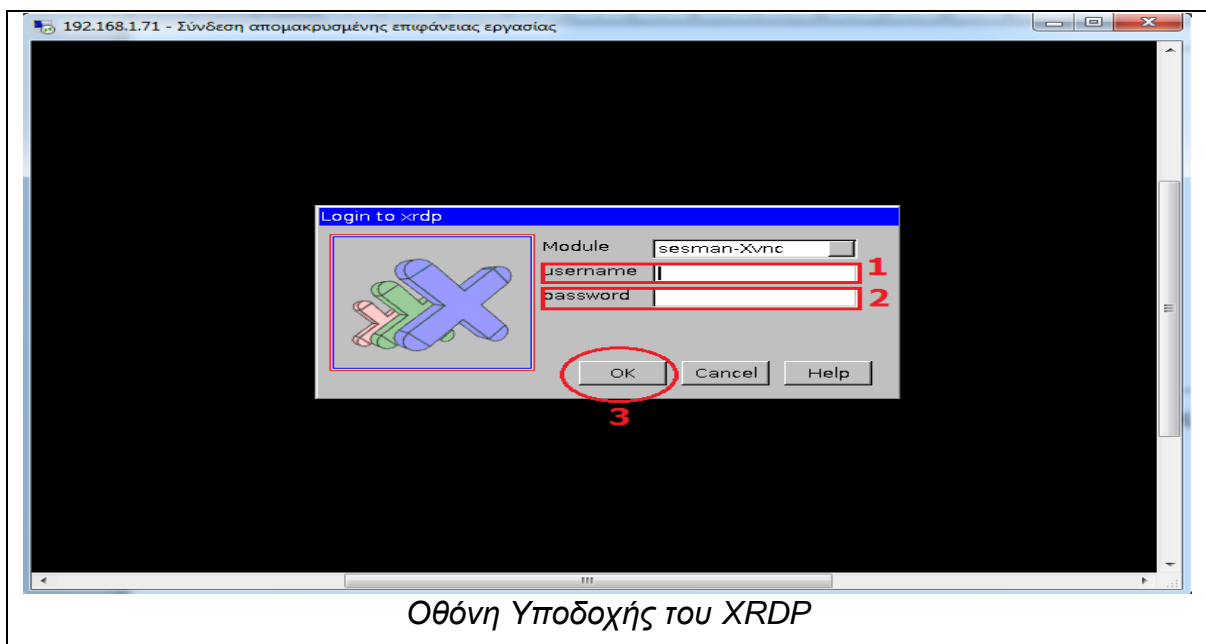
Για χρήστες Windows

Τα Windows (XP, Vista, 7) διαθέτουν ενσωματωμένη εφαρμογή RDP Client η οποία ονομάζεται «Σύνδεση απομακρυσμένης επιφάνειας εργασίας» (Remote Desktop Connection) και βρίσκεται στην διαδρομή **Έναρξη → Όλα τα**

προγράμματα → Βοηθήματα → Σύνδεση απομακρυσμένης επιφάνειας εργασίας. Ανοίγουμε την εφαρμογή και μας εμφανίζεται το κάτωθι παράθυρο:

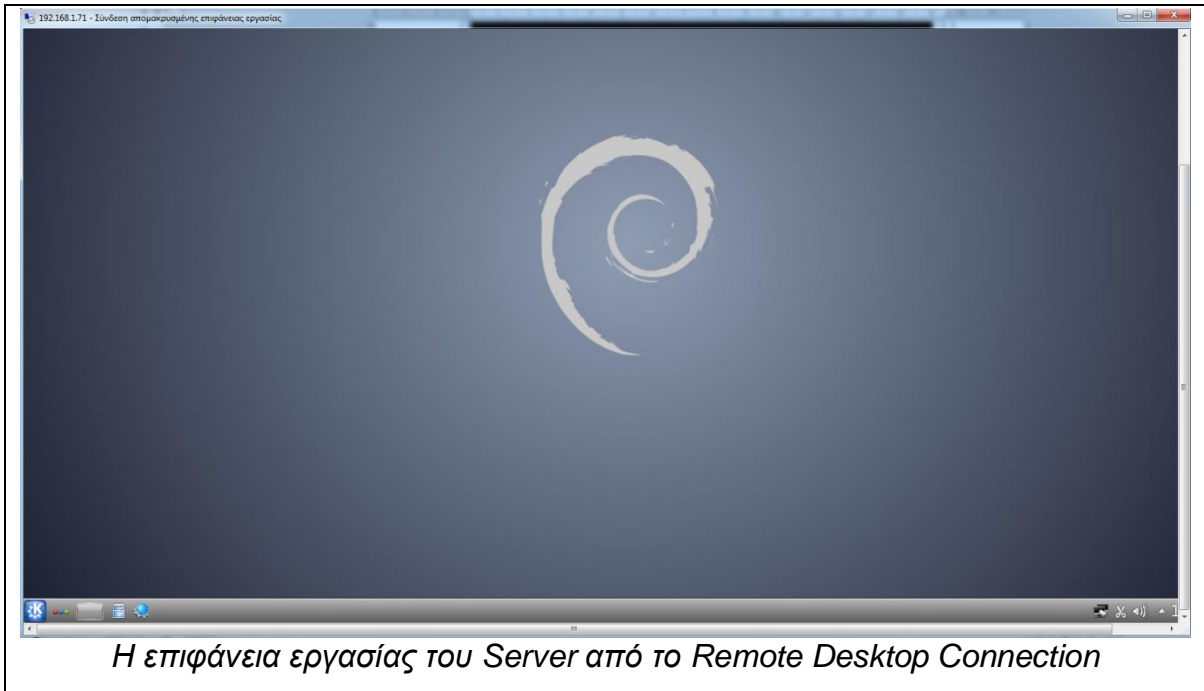


Στο πεδίο «Υπολογιστής» (1) εισάγουμε την IP Διεύθυνση του Server μας, και στη συνέχεια πατάμε «Σύνδεση» (2). Αν εισάγατε την σωστή διεύθυνση και η υπηρεσία XRDP τρέχει στον Server τότε πρέπει να λάβετε ένα παράθυρο όμοιο με το κάτωθι:



Στην παραπάνω εικόνα βλέπουμε την οθόνη υποδοχής του XRDP. Εισάγετε το όνομα χρήστη σας (1) και τον κωδικό σας (2) και στην συνέχεια

πατήστε OK (3). Αν τα στοιχεία που εισάγατε είναι τα σωστά, τότε θα σας εμφανιστεί η επιφάνεια εργασίας του Server.

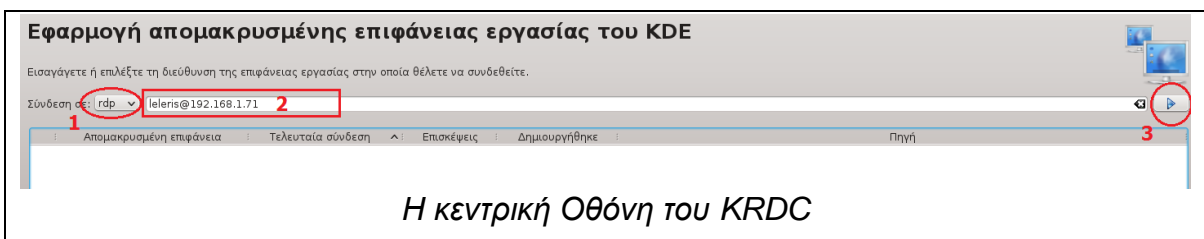


Για χρήστες Linux (KDE)

Οι χρήστες Linux μπορούν να χρησιμοποιήσουν το KRDC για σύνδεση στον Server. Αν το πρόγραμμα δεν το έχετε εγκατεστημένο, εγκαταστήστε το με την παρακάτω εντολή (ως χρήστης root)

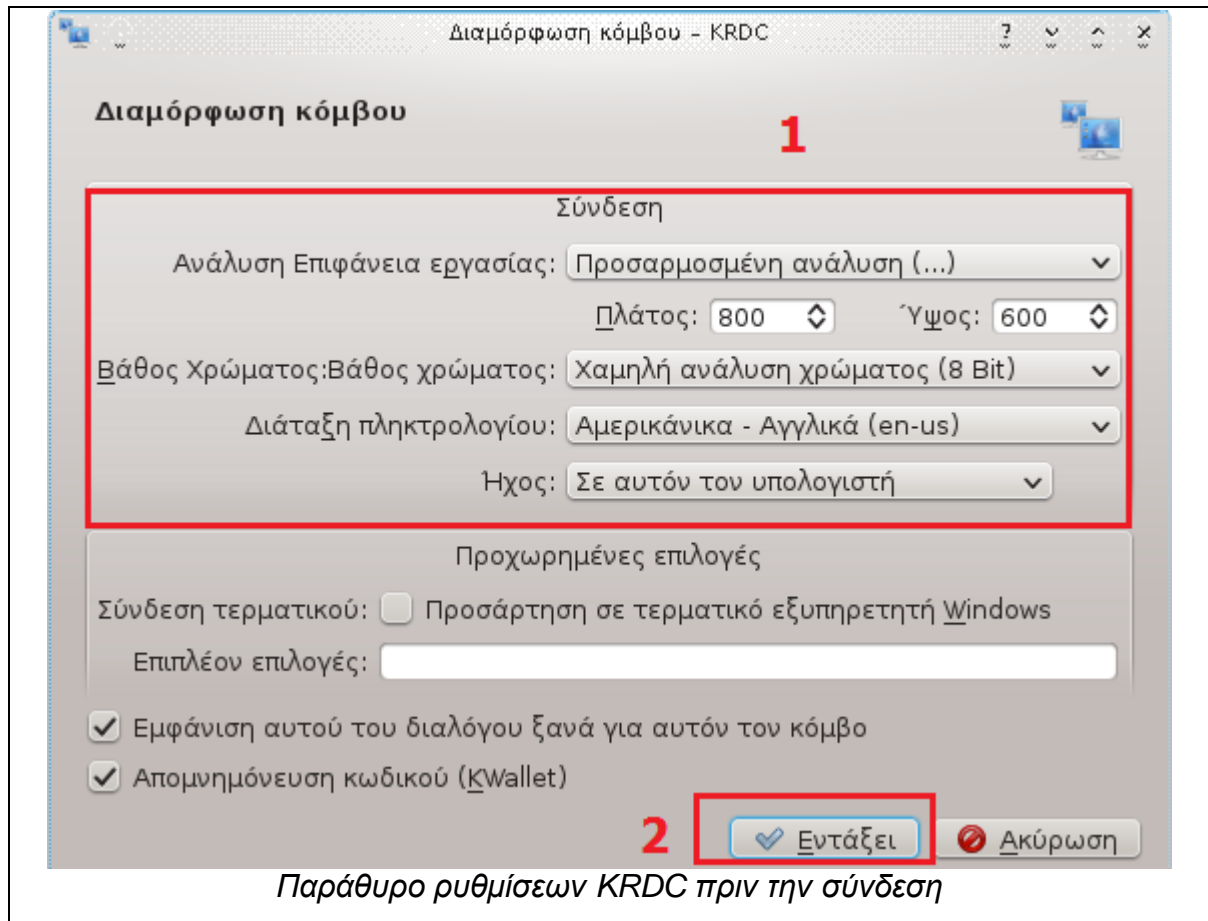
```
root@dbserver# apt-get install krdc
```

Με το πέρας της εγκατάστασης μπορείτε να τρέξετε το KRDC μέσα από τη διαδρομή **K Menu** → **Διαδίκτυο** → **KRDC**. Το παράθυρο της εφαρμογής το βλέπουμε μαζί στην παρακάτω εικόνα:

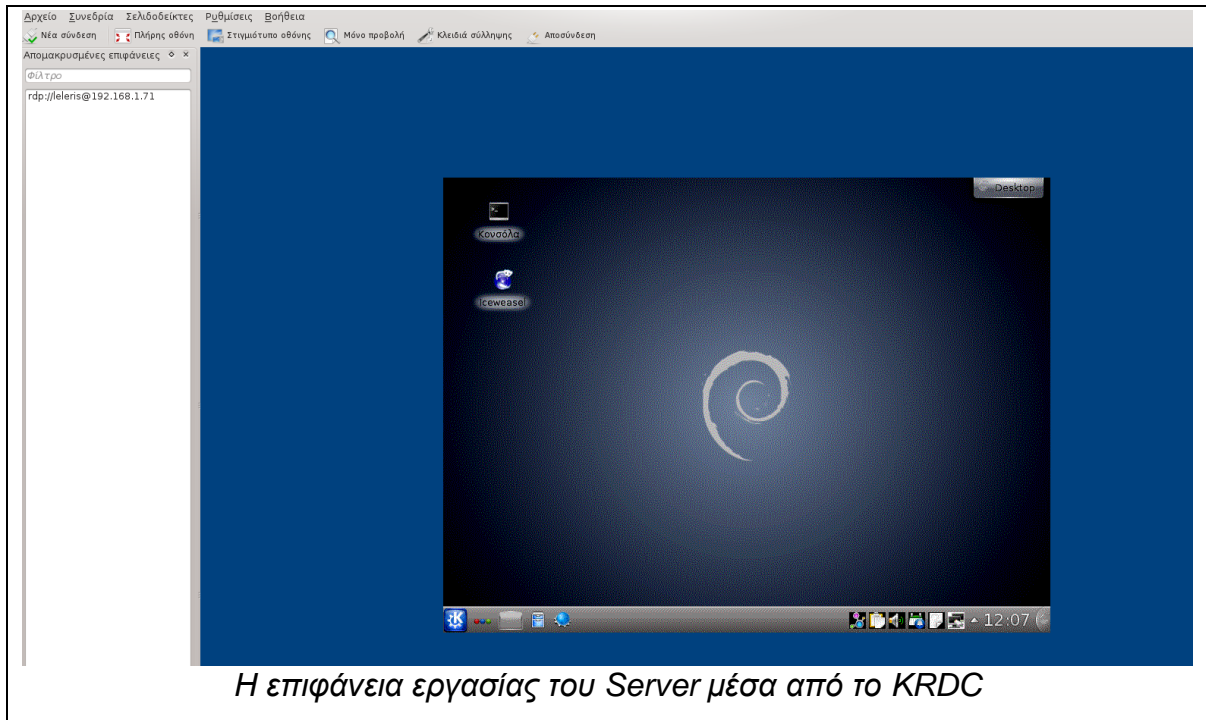


Αρχικά επιλέγουμε πρωτόκολλο RDP (1) από το είδος σύνδεσης. Στην συνέχεια δίνουμε το όνομα χρήστη μας σε συνδυασμό με την IP Διεύθυνση του

Server με την μορφή «*username@IP_ADDRESS*» (2) και στην συνέχεια πατάμε «Σύνδεση» (3). Τότε θα εμφανιστεί ένα παράθυρο ρυθμίσεων όπως παρακάτω:



Οι ρυθμίσεις που μας ενδιαφέρουν άμεσα βρίσκονται μέσα στο πρώτο τετράγωνο (1). Αν η σύνδεσή σας είναι αργή επιλέξτε χαμηλή ανάλυση οθόνης και βάθος χρώματος ή το αντίστροφο, αν η σύνδεσή σας είναι γρήγορη. Τέλος επιλέγουμε «Εντάξει» (2) και η σύνδεση πραγματοποιείται, ζητώντας μόνο ως τελευταίο προαπαιτούμενο, τον κωδικό πρόσβασής μας (password) και η επιφάνεια εργασίας εμφανίζεται:



2.1.1.2 - Το πρωτόκολλο SSH

Αν οι ενέργειές μας μπορούν να γίνουν μόνο μέσα από το Terminal τότε το SSH είναι αυτό που χρειαζόμαστε. Το SSH μας δίνει την δυνατότητα να συνδεθούμε σε ένα απομακρυσμένο υπολογιστή μέσω ενός Terminal Emulator, σε μία σύνδεση κρυπτογραφημένη και απόλυτα ασφαλή. Επίσης επειδή δουλεύουμε μόνο στο κέλυφος (shell) του λειτουργικού συστήματος, και δεν υπάρχει κανενός είδους παραθυρικό στοιχείο, η σύνδεση αυτή έχει και ελάχιστη καθυστέρηση. Θα μας φανεί πολύ χρήσιμο ειδικά για την μελλοντική διαχείριση της βάσης δεδομένων. Επίσης, μέσω SSH μπορούμε να μεταφέρουμε και αρχεία, μέσω της τεχνικής **FTP over SSH** ή **SFTP**.

2.1.2.1 - Εγκατάσταση SSH Server

Το λογισμικό που επιλέξαμε για την υλοποίηση SSH στον Server μας είναι το **OpenSSH** και συγκεκριμένα το server κομμάτι του, το **SSHD** (SSH Daemon).

Το εγκαθιστούμε μέσω του package manager με την παρακάτω εντολή (ως χρήστες root)

```
root@dbserver# apt-get install openssh-server
```

Με το πέρας της εγκατάστασης του πακέτου, πρέπει να ενεργοποιηθεί η υπηρεσία SSH ώστε να μπορεί ο Server να εξυπηρετεί SSH αιτήματα. Η εντολή που πρέπει να δώσουμε είναι η κάτωθι (ως root):

```
root@dbserver# service ssh start
```

Τώρα ο Server μας ακούει για εισερχόμενες συνδέσεις με το πρωτόκολλο SSH. Δεν μένει παρά να το δοκιμάσουμε.

2.1.2.2 - Σύνδεση μέσω SSH

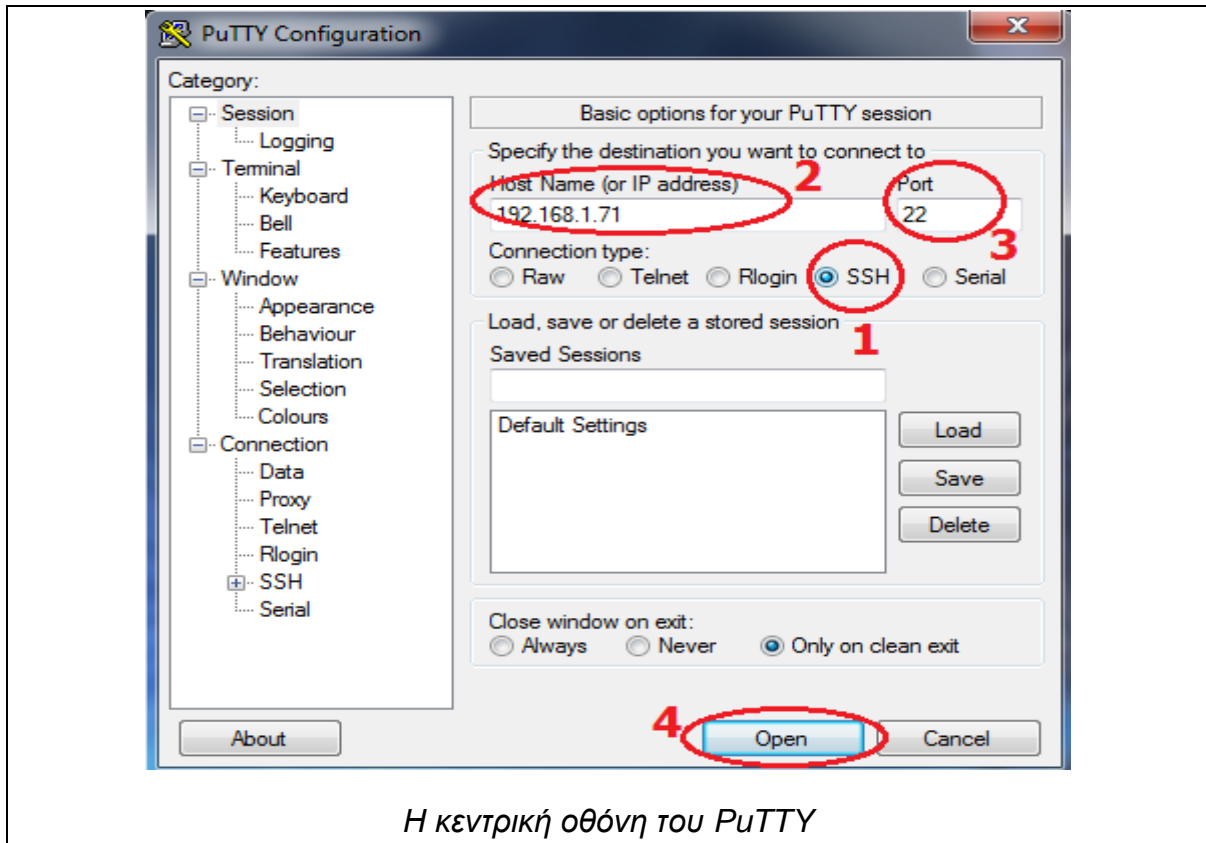
Ο Server ακούει συνδέσεις, οπότε πρέπει να χρησιμοποιήσουμε κάποιον SSH Client για να συνδεθούμε σε αυτόν. Υπάρχει πληθώρα εφαρμογών που μπορούμε να χρησιμοποιήσουμε, ωστόσο εμείς θα επικεντρωθούμε στα περισσότερο γνωστά και αποδοτικά. Θα πρέπει να θυμόμαστε την **IP Διεύθυνση του Server** και τα **διαπιστευτήρια σύνδεσής μας**.

Για χρήστες Windows

Ένα καλό terminal emulator και ssh client είναι το PuTTY. Το βρίσκετε και το κατεβάζετε στην παρακάτω διεύθυνση:

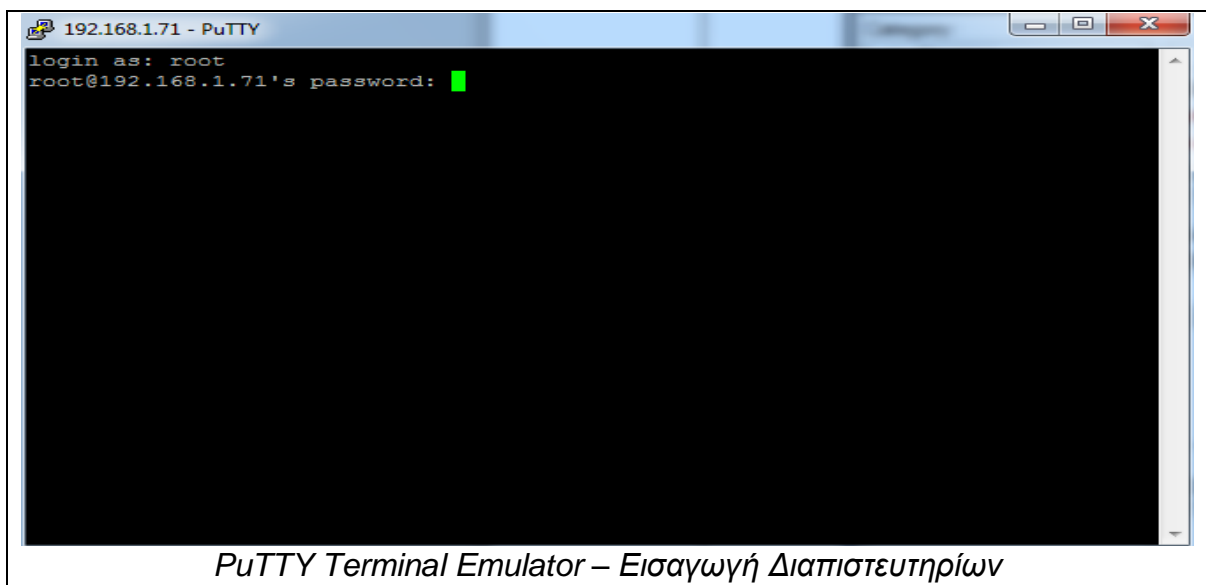
<http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>

Κατεβάζουμε το PuTTY και το τρέχουμε όπως είναι (είναι Standalone Εκτελέσιμο). Η οθόνη του PuTTY εμφανίζεται μπροστά μας, η οποία είναι η παρακάτω:



Η κεντρική οθόνη του PuTTY

Αρχικά επιλέγουμε ως τύπο σύνδεσης το SSH (1). Στην συνέχεια δίνουμε την IP Διεύθυνση του Server μας (2) και την θύρα που ακούει για SSH (3). Τέλος, πατάμε Open για να συνδεθούμε (4). Τότε θα εμφανιστεί το terminal Emulator, το οποίο θα μας ζητήσει τα διαπιστευτήριά μας (username και password λογαριασμού στο λειτουργικό σύστημα), όπως βλέπουμε κάτω:



PuTTY Terminal Emulator – Εισαγωγή Διαπιστευτηρίων

Αφού τα εισάγουμε σωστά, μπορούμε να εργαστούμε σαν να είμασταν μπροστά στον Server και να είχαμε ανοιχτό μπροστά μας ένα Terminal Emulator όπως το Konsole.

Για χρήστες Linux:

Οι χρήστες Linux μπορούν να εργαστούν απευθείας μέσα από το Terminal Emulator τους όπως το Konsole ή το X-Term αρκεί φυσικά να έχουν εγκαταστήσει το πακέτο λογισμικού OpenSSH-Client. Αν και στις περισσότερες διανομές πλέον υπάρχει ήδη προ-εγκατεστημένο, ωστόσο αν δεν υπάρχει στην διανομή σας, εγκαταστήστε το μέσω του package manager (ως χρήστες root):

```
root@localhost# apt-get install openssh-client # ← Debian / Ubuntu / PCLinux OS
```

```
root@localhost# urpmi openssh-client # ← Mandriva / Mageia Linux
```

Με το πέρας της εγκατάστασης, μπορείτε να συνδεθείτε μέσω SSH στον Server επιλέγοντας μία εκ των δύο κάτωθι εντολών:

```
user@localhost$ ssh -l <username> <server_ip_address>
```

εναλλακτικά

```
user@localhost ssh username@<server_ip_address>
```

Όποια από τις δύο επιλέξετε, το αποτέλεσμα θα είναι το ίδιο. Το μόνο που θα σας ζητήσει να εισάγετε είναι το password του χρήστη που επιλέξατε. Στην αλλαγή της προτροπής από «localhost» σε «dbserver» για παράδειγμα, αντιλαμβάνεστε πως πλέον εργάζεστε μέσω SSH στον Server.

2.2 – Δημιουργία λογαριασμού Oracle και απόκτηση της Oracle 11g

Για να προχωρήσουμε στην εγκατάσταση της Oracle πρέπει να έχουμε τα αρχεία εγκατάστασής της. Και για να αποκτήσουμε τα τελευταία χρειαζόμαστε ένα **λογαριασμό Oracle** (Oracle Account).

2.2.1 Δημιουργία Oracle Account

Μεταβείτε στην σελίδα <https://login.oracle.com/mysso/signon.jsp> και πατήστε στο πλήκτρο **Εγγραφείτε** όπως στην παρακάτω εικόνα:



The image shows a screenshot of the Oracle login page. The title is "Σύνδεση". There are two input fields: "Όνομα χρήστη" and "Κωδικός πρόσβασης". Below each field is a link: "Χάσατε το Όνομα χρήστη;" and "Χάσατε τον κωδικό πρόσβασης;". A red button labeled "Σύνδεση" is positioned below the input fields. Below the button, the text reads: "Δεν έχετε λογαριασμό Oracle; **Εγγραφείτε** για δωρεάν Διαδικτυακό λογαριασμό Oracle". The word "Εγγραφείτε" is circled in red, and a large red arrow points to it. At the bottom of the form, there is a link "Χρειάζεστε βοήθεια;" and the text "Powered by Oracle Access Manager 11g". Below the screenshot, the text "Οθόνη εισαγωγής στοιχείων" is written.

Θα φορτώσει η φόρμα εγγραφής όπου θα σας ζητήσει να εισάγετε τα στοιχεία σας. Ακολουθήστε πιστά τις οδηγίες που δίδονται ώστε να δημιουργήσετε τον λογαριασμό σας. Φροντίστε τα διαπιστευτήρια (email και password) να μην τα ξεχάσετε. Ο λόγος για τον οποίο είναι απαραίτητος ο λογαριασμός Oracle είναι για την απόκτηση των εργαλείων που χρειάζονται για να εργαστείτε στην Oracle Database, όπως το SQL Developer και κυρίως τώρα για την απόκτηση της Oracle Database.

Αφού ολοκληρώσετε την εγγραφή σας, είστε έτοιμοι να αποκτήσετε την Oracle 11g και ότι άλλο λογισμικό της Oracle θέλετε.

2.2.2 – Απόκτηση της Oracle 11g

Για να κατεβάσετε τα αρχεία εγκατάστασης της Oracle 11g στον υπολογιστή σας θα πρέπει να μεταβείτε στην παρακάτω σελίδα:

<http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html>

Θα πρέπει να αποδεχτείτε τους όρους χρήσης, και να επιλέξετε για κατέβασμα τα δύο zip αρχεία, για Linux x86_64 , κάτω από την ενότητα «Oracle Database 11g Release 2» όπως βλέπουμε παρακάτω:

You must accept the **OTN License Agreement** to download this software.

Accept License Agreement | Decline License Agreement

**Oracle Database 12c Release 1
Standard Edition, Standard Edition One, and Enterprise Edition**

(12.1.0.1.0)

Microsoft Windows x64 (64-bit)	File 1, File 2 (2.5 GB)	See All
Linux x86-64	File 1, File 2 (2.3 GB)	See All
Oracle Solaris (SPARC systems, 64-bit)	File 1, File 2 (2.6 GB)	See All
Oracle Solaris (x86 systems, 64-bit)	File 1, File 2 (2.3 GB)	See All

- See All page above contains **unzip instructions** plus Database Client (32 and 64-bit), Gateways, Grid Infrastructure, more

**Oracle Database 11g Release 2
Standard Edition, Standard Edition One, and Enterprise Edition**

7/13: Patch Set 11.2.0.4 for Linux and Solaris is now available on support.oracle.com. Note: it is a full installation (you do not need to download 11.2.0.1 first). See the **README** for more info (login to My Oracle Support required).

(11.2.0.2.0)

zLinux64	File 1, File 2 (2GB)	See All
----------	----------------------	---------

(11.2.0.1.0)

Microsoft Windows (32-bit)	File 1, File 2 (2GB)	See All
Microsoft Windows (x64)	File 1, File 2 (2GB)	See All
Linux x86	File 1, File 2 (2GB)	See All
Linux x86-64	File 1, File 2 (2GB)	See All
Solaris (SPARC) (64-bit)	File 1, File 2 (2GB)	See All
Solaris (x86-64)	File 1, File 2 (2GB)	See All
HP-UX Itanium	File 1, File 2 (2GB)	See All
HP-UX PA-RISC (64-bit)	File 1, File 2 (2GB)	See All
AIX (PPC64)	File 1, File 2 (2GB)	See All

Οθόνη λήψης Oracle 11g

Αφού επιλέξετε το πρώτο αρχείο για εγκατάσταση θα σας ζητήσει login με το Oracle Account σας. Με την σωστή εισαγωγή των στοιχείων σας το κατέβασμα θα ξεκινήσει άμεσα.

Keep in mind...

Τα αρχεία εγκατάστασης (zip) της Oracle 11g είναι 2 GB έκαστο, δηλαδή 4 GB στο σύνολο. Όπως καταλαβαίνεται είναι μεγάλα αρχεία ως εκ τούτου θα καθυστερήσουν να κατεβούνε. Προτείνεται ωστόσο η χρήση ενός Download Manager όπως το WGET ή το ARIA2.

Ολοκληρώνοντας...

Αφού ο Server μας έχει προετοιμαστεί κατάλληλα για να μπορούμε να τον διαχειριζόμαστε εξ αποστάσεως και αφού έχουμε αποκτήσει την Oracle 11g τότε είμαστε έτοιμοι να προχωρήσουμε στην εγκατάσταση της. Η δυνατότητα απομακρυσμένης διαχείρισης του Server μας, μας παρέχει την ελευθερία να τον διαχειριζόμαστε με την άνεσή μας και μάλιστα από το χώρο μας, εφόσον φυσικά διαθέτουμε διαδικτυακή πρόσβαση σε αυτόν. Τα δύο ισχυρά αυτά πρωτόκολλα (SSH και RDP) φέρνουν τον Server μπροστά μας.

Στο επόμενο κεφάλαιο θα ασχοληθούμε με την εγκατάσταση της Oracle Database 11g στο Server μας, βήμα προς βήμα για να μην αντιμετωπίσετε καμία δυσκολία και πρόκληση. Μην φοβάστε! Σας εποπτεύω!

ΚΕΦΑΛΑΙΟ 3 – ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ ORACLE 11g

Σε αυτό το κεφάλαιο θα προχωρήσουμε στην εγκατάσταση της Oracle Database 11g. Θα σας οδηγήσουμε βήμα προς βήμα για όλα τα απαραίτητα βήματα που πρέπει να κάνετε ώστε να εγκαταστήσετε σωστά και χωρίς προβλήματα την Oracle στον Server σας από την προετοιμασία του Server σας μέχρι την πλήρη ολοκλήρωση της εγκατάστασης.

Στο παρακάτω πίνακα σας παραθέτω την **προτεινόμενη** σύσταση του υπολογιστή που θα αναλάβει το ρόλο του Oracle Server σας.

Αρχιτεκτονική CPU	x64/AMD64 (κατά προτίμηση διπύρηνος και άνω)
Μνήμη	>=2 GB (τουλάχιστον DDR2)
Χώρος στο δίσκο	>=20 GB
Λειτουργικό Σύστημα	Debian 7 Stable (Wheezy) 64bit
Εικονική Μνήμη (Swap Space)	>= 500 MB

Το δικό μας σύστημα αποτελείται από επεξεργαστή Intel Core 2 Duo, μνήμες 4 GB DDR2, Δίσκο 120 GB, με Swap Space 1GB και λειτουργικό Debian 7 64bit.

Προσοχή!

Επιχειρούμε να εγκαταστήσουμε την Oracle σε λειτουργικό σύστημα Debian 7 Wheezy 64bit το οποίο **δεν προτείνεται** από την Oracle. Ως εκ τούτου πιθανών να συναντήσουμε σφάλματα κατά την διάρκεια της εγκατάστασης. Ακολουθήστε τα βήματα που σας δίνουμε, και θα ανακάμψετε γρήγορα από αυτά.

Η εγκατάσταση καλό είναι να γίνει εκ του σύνεγγυς και όχι εξ αποστάσεως. Έτσι θα είστε σε θέση να επέμβετε άμεσα σε περίπτωση πχ που ο Server κολλήσει ή παγώσει κατά την διάρκεια της εγκατάστασης.

3.1 – Προετοιμασία του συστήματος

Πριν την εκκίνηση της εγκατάστασης πρέπει απαραίτητα να κάνουμε κάποιες προετοιμασίες, ώστε η Oracle να εγκατασταθεί σωστά στον υπολογιστή. Η προετοιμασία περιλαμβάνει δημιουργία καταλόγων, χρηστών, ομάδων και εγκατάσταση εξαρτήσεων από τον Package Manager. Η σωστή προετοιμασία εγγυάται σε μεγάλο βαθμό ότι η εγκατάσταση θα ολοκληρωθεί χωρίς ιδιαίτερα προβλήματα.

3.1.1 – Εγκατάσταση Εξαρτήσεων

Η Oracle για εγκατασταθεί, θα πρέπει να υπάρχουν προεγκατεστημένα πακέτα λογισμικού στον Server προκειμένου να τα βρει κατά την διάρκεια της εγκατάστασης (και λειτουργίας της), καθώς χωρίς αυτά δεν μπορεί να εγκατασταθεί και ως εκ τούτου να λειτουργήσει. Τα πακέτα αυτά είναι οι λεγόμενες **εξαρτήσεις** (dependencies). Ευτυχώς για μας, την δύσκολη αυτή δουλειά θα την αναλάβει ο Package Manager του Debian, ο APT.

Αρχικά θα πρέπει να προσθέσετε στην λίστα των Repositories/Mirrors το αποθετήριο του Μετσόβιου Πολυτεχνείου, που είναι ένα από τα επίσημα ελληνικά αποθετήρια του Debian. Ανοίξτε ένα Terminal και συνδεθείτε ως χρήστες Root:

```
user@debserver$ su
Password: # ← Εισάγετε τον κωδικό του χρήστη Root
root@debserver#
```

Στην συνέχεια πρέπει να τροποποιήσουμε τα περιεχόμενα του αρχείου «sources.lst» που βρίσκεται στην τοποθεσία «/etc/apt/sources.lst» με τον GNU Nano:

```
root@debserver# nano /etc/apt/sources.lst
```

Αφού ο Nano ανοίξει το αρχείο, προσθέστε την παρακάτω γραμμή σε αυτό:

```
deb http://ftp.gr.debian.org/debian stable main contrib non-free
```

Επίσης κάντε σχόλιο την εξής γραμμή προσθέτοντας το σύμβολο # πριν από αυτήν για να μην ζητάει το DVD του Debian σε κάθε εγκατάσταση πακέτου.

```
# deb cdrom:[Debian GNU/Linux 7.0.0 _Wheezy_ - Official amd64 DVD Binary-1  
#20130 504-14:44]/ wheezy contrib main
```

Αφού ολοκληρώσετε, πατήστε Ctrl+X για να κλείσει ο Nano, και στην προτροπή του να σώσει το αρχείο, πατήστε Y (yes) και κλείστε τον.

Τώρα που εισάγαμε νέο Repository στην λίστα του APT πρέπει να τον πούμε να ενημερώσει την βάση δεδομένων του για τα διαθέσιμα πακέτα που υπάρχουν σε αυτό το αποθετήριο. Για να το κάνουμε αυτό γράφουμε την παρακάτω εντολή (ως root).

```
root@debserver# apt-get update
```

Η διαδικασία μπορεί να πάρει λίγο χρόνο, αναλόγως με την ταχύτητα σύνδεσής σας και τις επιδόσεις του υπολογιστή. Με το πέρας της διαδικασίας, μπορούμε να εγκαταστήσουμε πλέον τα πακέτα-εξαρτήσεις που είναι απαραίτητα για την Oracle¹ δίνοντας την παρακάτω εντολή:

```
root@debserver# apt-get install gcc make binutils libmotif4 lesstif2 rpm libaio1  
libstdc++5
```

Τα πακέτα θα απαιτήσουν περίπου 50+ MB στον δίσκο οπότε αφήστε τα να κατεβούνε και να εγκατασταθούν. Η διαδικασία ενδέχεται να καθυστερήσει, όχι πάντως περισσότερο από 3 λεπτά.

3.1.2 – Δημιουργία Symbolic Links

¹ Internet Source – Debian Administration → Making Debian a bit more compatible → “We should install a few packages so that the Oracle installer will be able to do its thing” – Site (<http://www.debian-administration.org/articles/656>)

Ο Installer της Oracle θα χρειαστεί κάποια utilities του συστήματος, τα οποία εκεί όμως που βρίσκονται δεν θα τα βρει ποτέ. Θα πρέπει λοιπόν εμείς να δημιουργήσουμε Symbolic Links² προς αυτά τα αρχεία, στις τοποθεσίες που θα ψάξει ο Installer ώστε –εμμέσως- να τα βρει και να τα χρησιμοποιήσει όταν τα χρειαστεί, δηλαδή στον κατάλογο bin. Το παραπάνω επιτυγχάνεται με τις κάτωθι εντολές:

```
root@debserver# ln -s /usr/bin/awk /bin/awk
root@debserver# ln -s /usr/bin/rpm /bin/rpm
root@debserver# ln -s /usr/bin/basename /bin/basename
```

3.1.3 – Δημιουργία Χρηστών και Ομάδων³

Αρχικά θα πρέπει να δημιουργήσουμε δύο ομάδες οι οποίες θα χρειαστούν κατά την εγκατάσταση, την ομάδα OINSTALL και την ομάδα DBA. Η πρώτη χρειάζεται για την εγκατάσταση της Oracle ενώ η δεύτερη για τους διαχειριστές της βάσης (για τον ρόλο DBA θα μιλήσουμε σε παρακάτω κεφάλαια). Καλό θα είναι οι ομάδες αυτές να έχουν δικό τους ID, ας πούμε 500 και 501 αντίστοιχα. Δημιουργούμε λοιπόν τις ομάδες με τις παρακάτω εντολές (ως χρήστες root):

```
root@debserver# groupadd -g 500 oinstall
root@debserver# groupadd -g 501 dba
```

Επιβεβαιώνουμε την δημιουργία των ομάδων ελέγχοντας τα περιεχόμενα του αρχείου «/etc/group» με την παρακάτω εντολή:

```
root@localhost# cat /etc/group
<output omit>
...
```

² Internet Source – Debian Administration → Making Debian a bit more compatible → “Create [symlinks](http://www.debian-administration.org/articles/656) so the installer can find a few utilities it needs” – Site (<http://www.debian-administration.org/articles/656>)

³ Internet Source – Debian Administration → Oracle account and volumes → “Create an account and group:” – Site (<http://www.debian-administration.org/articles/656>)

```
....  
oinstall:x:500  
dba:x:501
```

Τώρα μας μένει να δημιουργήσουμε τον χρήστη Oracle. Ο χρήστης oracle για το λειτουργικό σύστημα είναι ένας απλός χρήστης, αλλά για την Oracle είναι ο μοναδικός χρήστης που επιτρέπεται να την διαχειρίζεται. Κοινώς για οποιαδήποτε διαδικασία διαχείρισης του server θα πρέπει να γίνεται μέσω αυτού του χρήστη.

Θα δημιουργήσουμε τον χρήστη oracle δημιουργώντας για αυτόν ένα home directory (προεπιλογή /home/oracle) , θα του δώσουμε αναγνωριστικό 1000 και θα τον εντάξουμε στις ομάδες που δημιουργήσαμε προηγουμένως. Επίσης σαν default shell του δίνουμε τον Bash. Για να πραγματοποιήσουμε τα παραπάνω χρειαζόμαστε την κάτωθι:

```
root@debserver# useradd -m -g oinstall -G dba -p passwd -s /bin/bash -u 1000
```

Επιβεβαιώνουμε την ύπαρξη του νέου μας χρήστη ελέγχοντας τα περιεχόμενα του αρχείου «/etc/passwd» ως ακολούθως:

```
root@debserver# cat /etc/passwd  
  
<output omit>  
  
...  
  
oracle:x:1000:500::/home/oracle:/bin/bash
```

Ο χρήστης μας και οι ομάδες έχουν δημιουργηθεί επιτυχώς. Τώρα δεν μένει να πάμε στο επόμενο βήμα.

3.1.4 – Δημιουργία δομής καταλόγων της Oracle

Σε αυτό το βήμα θα πρέπει να δημιουργήσουμε τους καταλόγους στους οποίους θα εγκατασταθεί η Oracle. Πρόκειται δηλαδή για τον χώρο αποθήκευσης του συστήματος και των βάσεων δεδομένων που θα δημιουργήσουμε μελλοντικά.

Καλή πρακτική είναι ο γονικός κατάλογος της Oracle με όλους τους υποκαταλόγους να αποθηκευτεί στην ρίζα (root) του συστήματος αρχείων. Δημιουργήστε λοιπόν τους απαραίτητους καταλόγους με τις παρακάτω εντολές:

```
root@debserver# mkdir /u01

root@debserver# chown oracle.oinstall /u01 # ← Μεταβίβαση σε oracle.oinstall

root@debserver# su oracle # ← αλλαγή σε χρήστη oracle

oracle@debserver$ mkdir /u01/app

oracle@debserver$ mkdir /u01/app/oracle

oracle@debserver$ mkdir /u01/app/orainventory

oracle@debserver$ mkdir /u01/app/oracle/product

oracle@debserver$ mkdir /u01/app/oracle/product/11.2.0

oracle@debserver$ mkdir /u01/app/oracle/product/11.2.0/dbhome
```

Η βασική δομή καταλόγων δημιουργήθηκε και είναι ιδιοκτησίας του χρήστη oracle και της ομάδας OINSTALL. Προχωράμε παρακάτω.

3.1.5 – Μεταβολή παραμέτρων πυρήνα και συστήματος⁴

Εδώ θα πρέπει να τροποποιήσουμε τις παραμέτρους του πυρήνα του λειτουργικού συστήματος. Αυτό θα το κάνουμε εισάγοντας μερικές νέες γραμμές στο αρχείο «/etc/sysctl.conf». Συνδεόμαστε ως χρήστες root και ανοίγουμε το αρχείο με τον Gnu Nano όπως παρακάτω:

```
root@debserver# nano /etc/sysctl.conf
```

Αφού το αρχείο ανοίξει με τον Nano, μετακινήστε τον δρομέα του τέρμα κάτω. Εκεί κάντε Copy-Paste τις παρακάτω γραμμές:

⁴ Internet Source – Debian Administration → Tuning kernel parameters – Site (<http://www.debian-administration.org/articles/656>)

```
kernel.sem = 250 32000 100 128

kernel.shmmax = 2147483648

net.ipv4.ip_local_port_range = 9000 65000

net.core.rmem_default = 262144

net.core.rmem_max = 4194304

net.core.wmem_default = 262144

net.core.wmem_max = 1048576

fs.aio-max-nr = 1048576

fs.file-max = 6815744
```

Εφαρμόστε τις παραμέτρους επί τόπου εκτελώντας την παρακάτω εντολή (ως root)

```
root@debserver# sysctl -p
```

Οι αλλαγές θα λάβουν χώρα αμέσως. Στην συνέχεια πρέπει να τροποποιήσουμε το αρχείο «/etc/security/limits.conf» με τον Gnu Nano.

```
root@debserver# nano /etc/security/limits.conf
```

Προσθέστε του τις παρακάτω γραμμές, πάνω από το σύμβολο «#END OF FILE»:

```
*      soft  nproc  2047

*      hard  nproc  16384

*      soft  nofile 1024

*      hard  nofile 65536
```

Κλείστε το και αποθηκεύστε το. Τώρα θα πρέπει να κάνουμε κάτι λίγο κουραστικό. Να ελέγξουμε τα εξής αρχεία, το «/etc/pam.d/su» , «/etc/pam.d/login»

και «/etc/pam.d/sshd» για να βεβαιωθούμε ότι **δεν έχουν σχόλιο** την πρόταση «session required pam_limits.so». Σε όποιο αρχείο την βρείτε σχόλιο, διαγράψτε τον χαρακτήρα # για να την κάνετε ενεργή.

Τελευταίο αρχείο που πρέπει να τροποποιήσουμε, είναι το αρχείο «/etc/profile» με τον Nano.

```
root@debserver# nano /etc/profile
```

Αφού ανοίξει προσθέτουμε τις παρακάτω γραμμές:

```
if [ $USER = "oracle" ]; then
    if [ $SHELL = "/bin/ksh" ]; then
        ulimit -p 16384
        ulimit -n 65536
    else
        ulimit -u 16384 -n 65536
    fi
fi
```

Με την ολοκλήρωση της εγγραφής των παραπάνω αρχείων, η εγκατάσταση είναι ελάχιστα μακριά.

3.1.6 – Δημιουργία Μεταβλητών περιβάλλοντος

Θα χρειαστούμε μερικές μεταβλητές περιβάλλοντος στο σύστημα καθώς χρειάζονται από την Oracle αλλά παράλληλα θα διευκολύνουν και το δικό μας έργο. Αυτές είναι οι εξής:

ORACLE_BASE, ORACLE_HOME, ORACLE_SID. Οι μεταβλητές αυτές περιγράφονται διεξοδικά στο επόμενο κεφάλαιο, ωστόσο η χρησιμότητά τους είναι η εξής:

Η πρώτη κρατάει το path προς τον κατάλογο /u01/app/oracle , η δεύτερη το path προς τον κατάλογο /u01/app/oracle/product/11.2.0/dbhome και η τρίτη θα κρατάει το Instance ID της βάσης (θα μιλήσουμε σε επόμενο κεφάλαιο για την χρησιμότητα του). Ως χρήστες oracle αποδίδουμε τιμές σε αυτές τις μεταβλητές ως εξής:

```
oracle@debserver$ export ORACLE_BASE=/u01/app/oracle

oracle@debserver$ export
ORACLE_HOME=/u01/app/oracle/product/product/11.2.0/dbhome

oracle@debserver$ export ORACLE_SID=test
```

Με την ολοκλήρωση των παραπάνω βημάτων μπορούμε πλέον να ξεκινήσουμε την εγκατάσταση της Oracle στο Server μας. Φροντίστε να έχετε διαθέσιμα τα αρχεία εγκατάστασης της Oracle σε σημείο εύκολα προσπελάσιμο.

3.2 – Εγκατάσταση Oracle 11g

Αν πραγματοποιείτε την εγκατάσταση **τοπικά**, βεβαιωθείτε αρχικά ότι έχετε διαθέσιμα τα δύο αρχεία zip με τα αρχεία εγκατάστασης της Oracle. Αν τα έχετε κατεβάσει από τον Server απευθείας μεταφέρετε τα στον home directory του χρήστη oracle δημιουργώντας παράλληλα και δύο καταλόγους zip και installer για να βάλετε τα zip αρχεία και τα αποσυμπιεσμένα αντίστοιχα. Πιθανών να χρειαστεί (ως root) να αλλάξετε τα δικαιώματα των αρχείων zip σε 777 (rwx) ώστε να σας αφήσει το σύστημα να τα μετακινήσετε από τη θέση τους ως χρήστης oracle. Αν δημιουργήσετε τους καταλόγους ως χρήστης root μην ξεχάσετε να μεταβιβάσετε την ιδιοκτησία στον χρήστη oracle και στην ομάδα oinstall με την chown. Παρακάτω βλέπουμε ένα ενδεικτικό set εντολών.

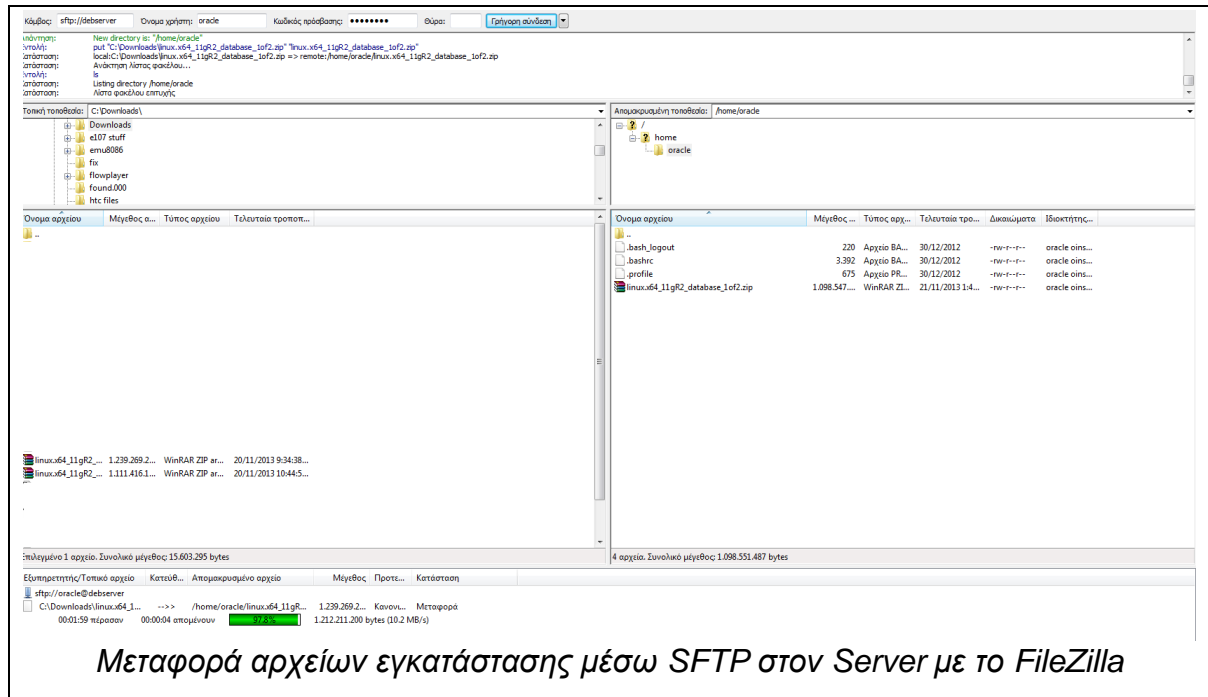
```
root@debserver# cd /home/user

root@debserver# mkdir /home/oracle/oracle11g
```



```
root@debserver# chown oracle.oinstall /home/oracle/oracle11g
root@debserver# chmod 777 /home/user/linux.x64.11gR2_*
su oracle
oracle@debserver$ cd /home/oracle
oracle@debserver$ mkdir oracle11g/installer
oracle@debserver$ mkdir oracle11g/zip
oracle@debserver$ mv /home/user/Downloads/linux.x64.11gR2* ./oracle11g/zip
```

Αν πραγματοποιείτε την εγκατάσταση εξ αποστάσεως (που δεν προτείνεται), θυμηθείτε ότι ο Server σας τρέχει SSH οπότε μπορείτε να μεταφέρετε τα αρχεία στον Server μέσω πρωτοκόλλου SFTP. Καλή εφαρμογή για την μεταφορά των αρχείων είναι το FileZilla. Συνδεθείτε λοιπόν με το FileZilla στον Server ως χρήστες Oracle στην θύρα 22 και μεταφέρετε τα αρχεία στον home directory του χρήστη Oracle. Αν μεταφέρετε τα αρχεία με το FileZilla δεν χρειάζεται να αλλάξετε την ιδιοκτησία των αρχείων, από προεπιλογή ανήκουν στον χρήστη oracle και στην ομάδα oinstall. Μόνο χρειάζεται να δημιουργήσετε τους καταλόγους oracle11g και τους υποκαταλόγους installer και zip. Παρακάτω βλέπουμε ένα στιγμιότυπο από την μεταφορά των αρχείων μέσω FileZilla στον Server.



Με το πέρας της μεταφοράς, μεταφέρετε τα αρχεία στον κατάλογο zip. Από κει και πέρα, όποια μέθοδο εγκατάστασης διαλέξατε, πρέπει να αποσυμπιέσετε τα αρχεία εγκατάστασης της Oracle με τις παρακάτω εντολές (ως χρήστες oracle).

```
oracle@debserver$ cd oracle11g/zip
oracle@debserver$ unzip linux.x64.11gR2_database_1of2.zip
#(περιμένετε να τελειώσει)
oracle@debserver$ unzip linux.x64.11gR2_database_2of2.zip
#(περιμένετε να τελειώσει)
```

Με το πέρας της αποσυμπίεσης, βλέπουμε πως στον ίδιο κατάλογο που βρισκόμαστε, υπάρχει ένας νέος με όνομα «database». Αυτός ο κατάλογος περιέχει τα αρχεία εγκατάστασης. Μεταφέρετέ τον στον κατάλογο installer και μεταβείτε στον κατάλογο database που μεταφέρατε με τις παρακάτω εντολές.

```
oracle@debserver$ mv database ../installer
oracle@debserver$ cd ../installer/database
```

Η εγκατάσταση μπορεί να ξεκινήσει. Μέσα στον κατάλογο database που βρισκόμαστε, υπάρχει ένα αρχείο ονόματι «runInstaller» ο οποίος είναι ο Oracle Universal Installer που θα πραγματοποιήσει την εγκατάσταση.

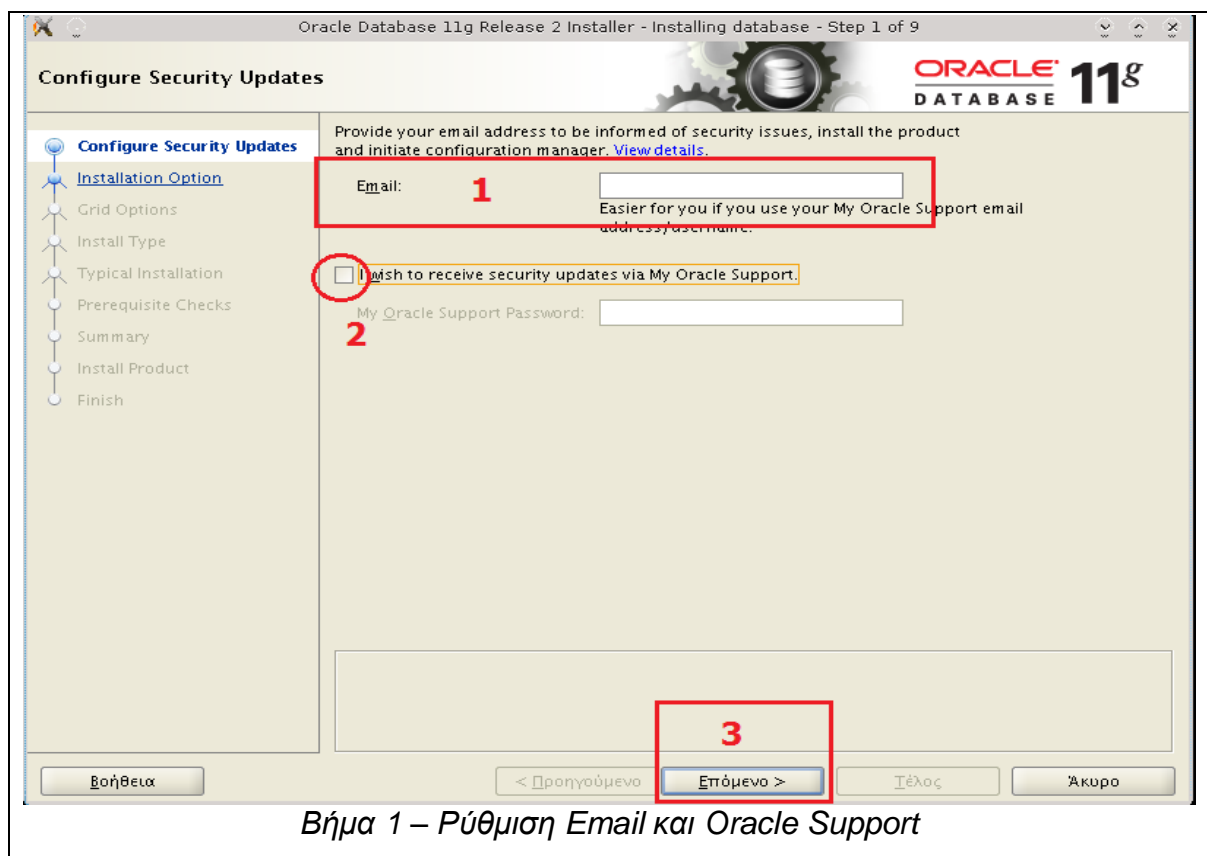
Προσοχή!

Η εγκατάσταση απαιτεί παραθυρικό περιβάλλον, οπότε αν την πραγματοποιείτε εξ αποστάσεως χρησιμοποιήστε πρωτόκολλο RDP και όχι SSH.

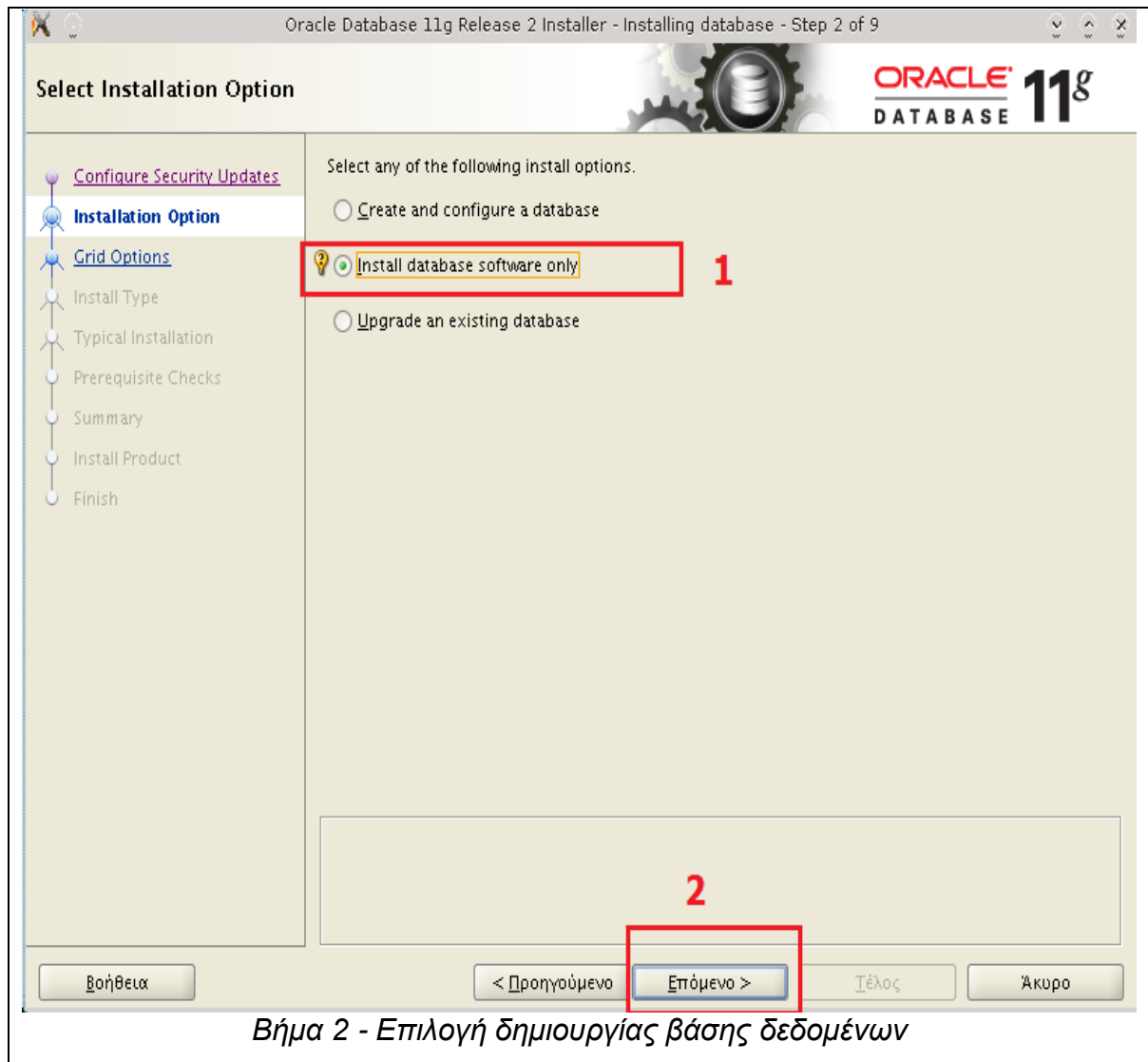
Ξεκινούμε την εγκατάσταση απλά εκτελώντας τον Installer (ως χρήστες Oracle).

```
oracle@debserver$ ./runInstaller
```

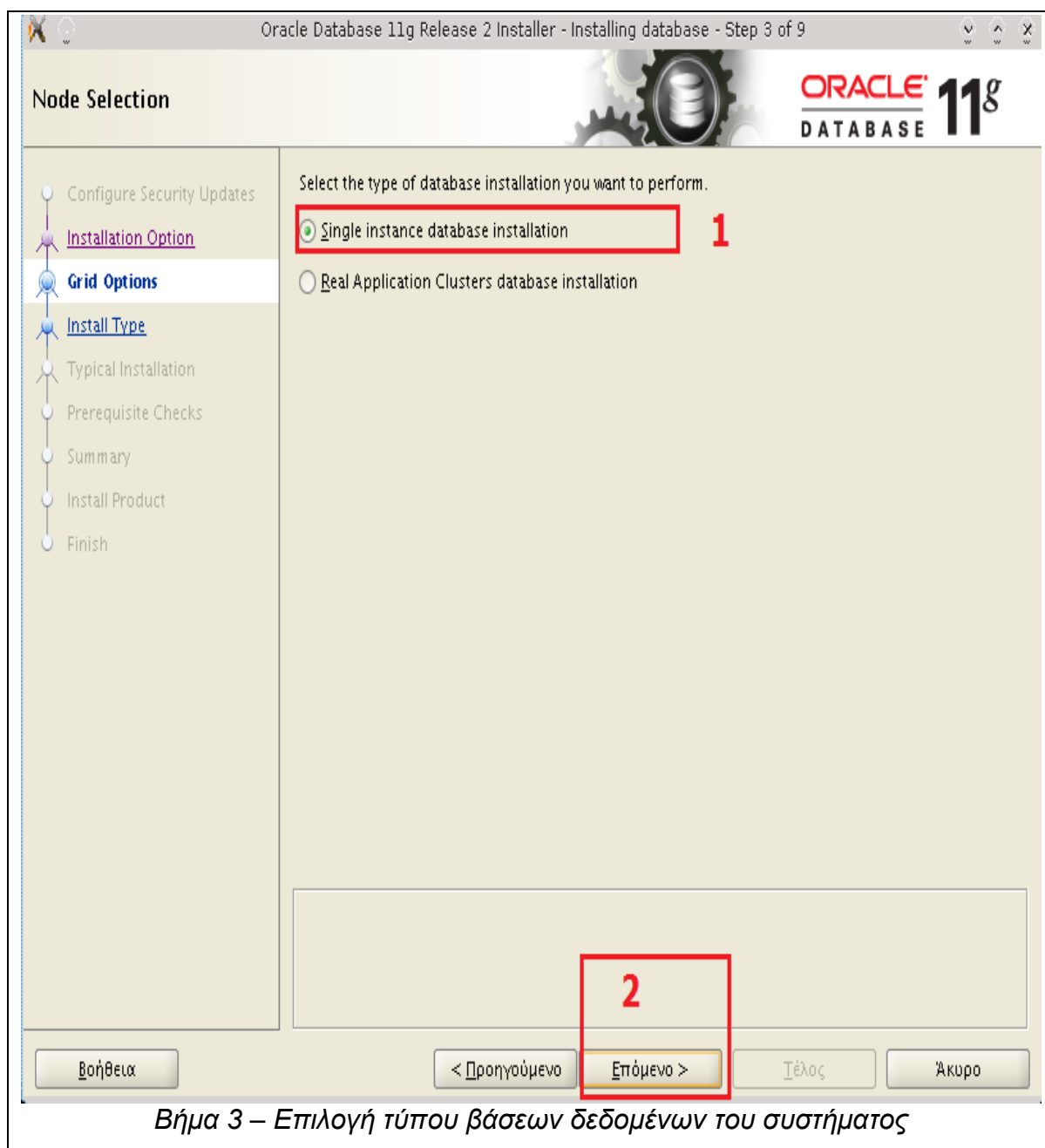
Σε μικρό διάστημα θα δείτε το έμβλημα της Oracle 11g και θα παρουσιαστεί μπροστά σας το πρώτο παράθυρο της εγκατάστασης το οποίο είναι το κάτωθι:



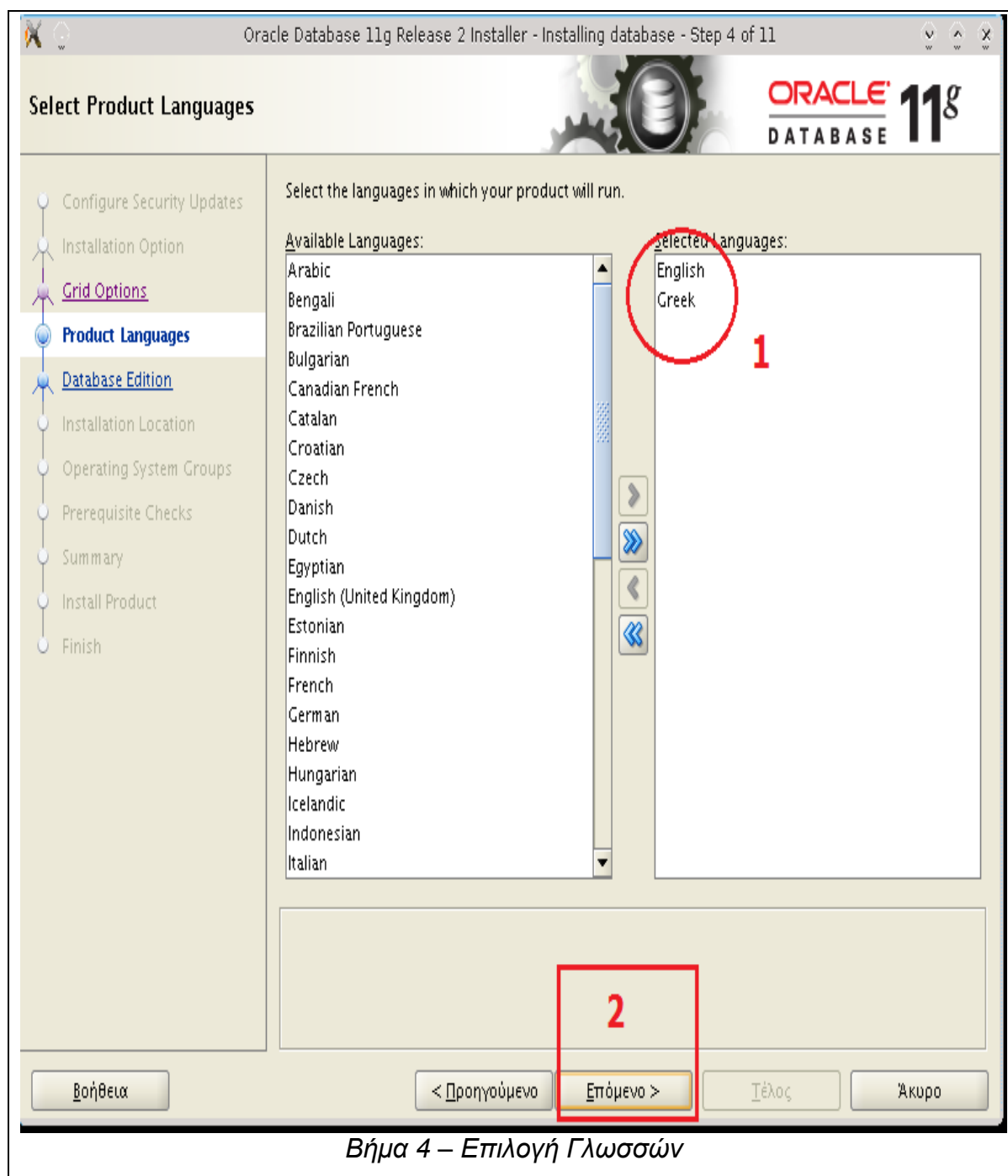
Εισάγετε **προαιρετικά** email σας που έχετε δηλώσει στην Oracle (1), απεπιλέγετε το «I Wish to receive security updates via my Oracle Support» (2) και τέλος πατάτε επόμενο (3). Μας εμφανίζεται το επόμενο παράθυρο:



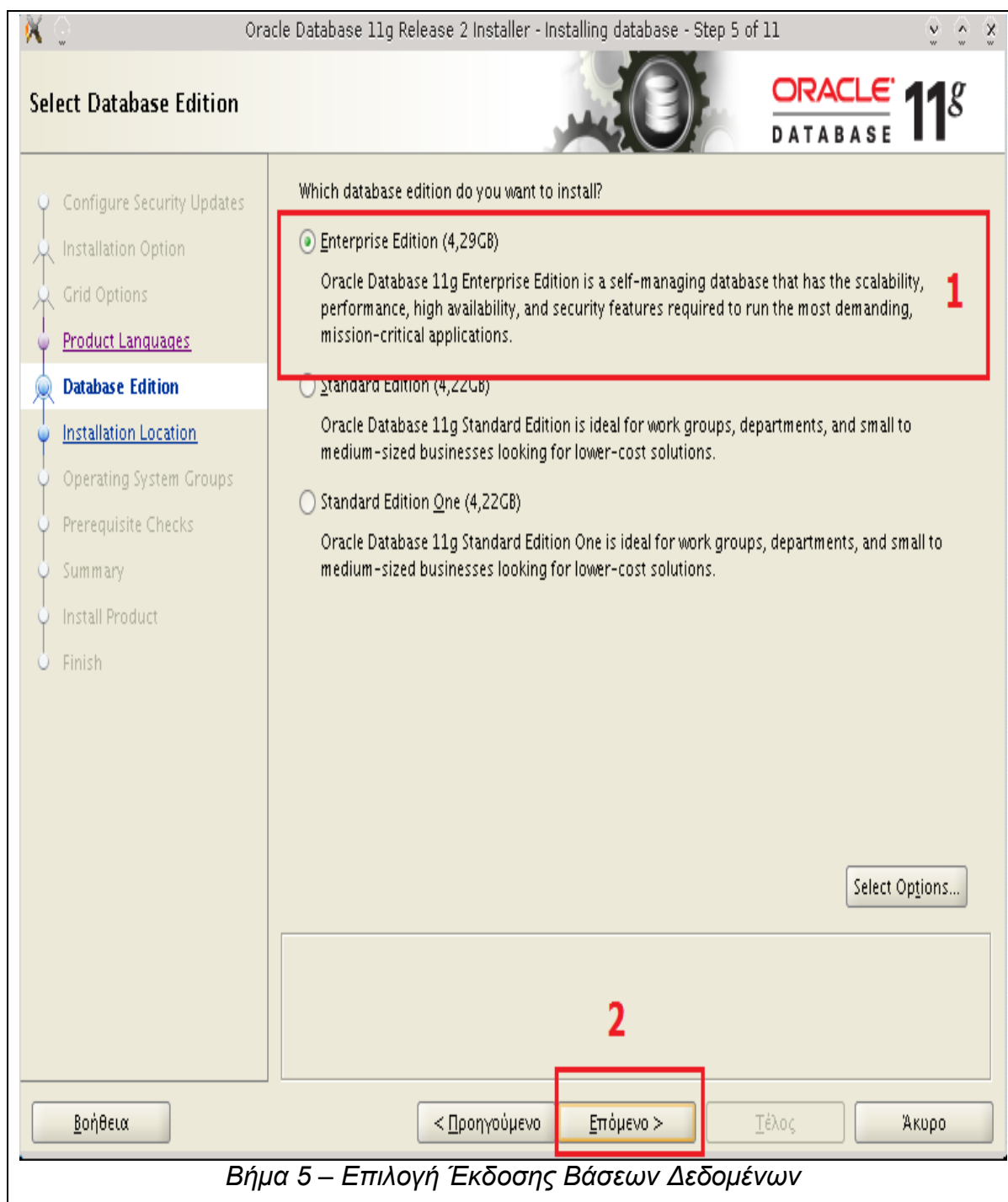
Στο παράθυρο αυτό μας ρωτάει αν θέλουμε να δημιουργήσουμε βάση δεδομένων παράλληλα με την εγκατάσταση. Εσείς επιλέγετε το «Install Database Software Only» (1), για να εγκαταστήσει μόνο το λογισμικό, καθώς βάση δεδομένων θα δημιουργήσουμε μετά το πέρας της εγκατάστασης, για να γίνει και όπως την θέλουμε. Πηγαίνετε στο επόμενο βήμα πατώντας επόμενο (2).



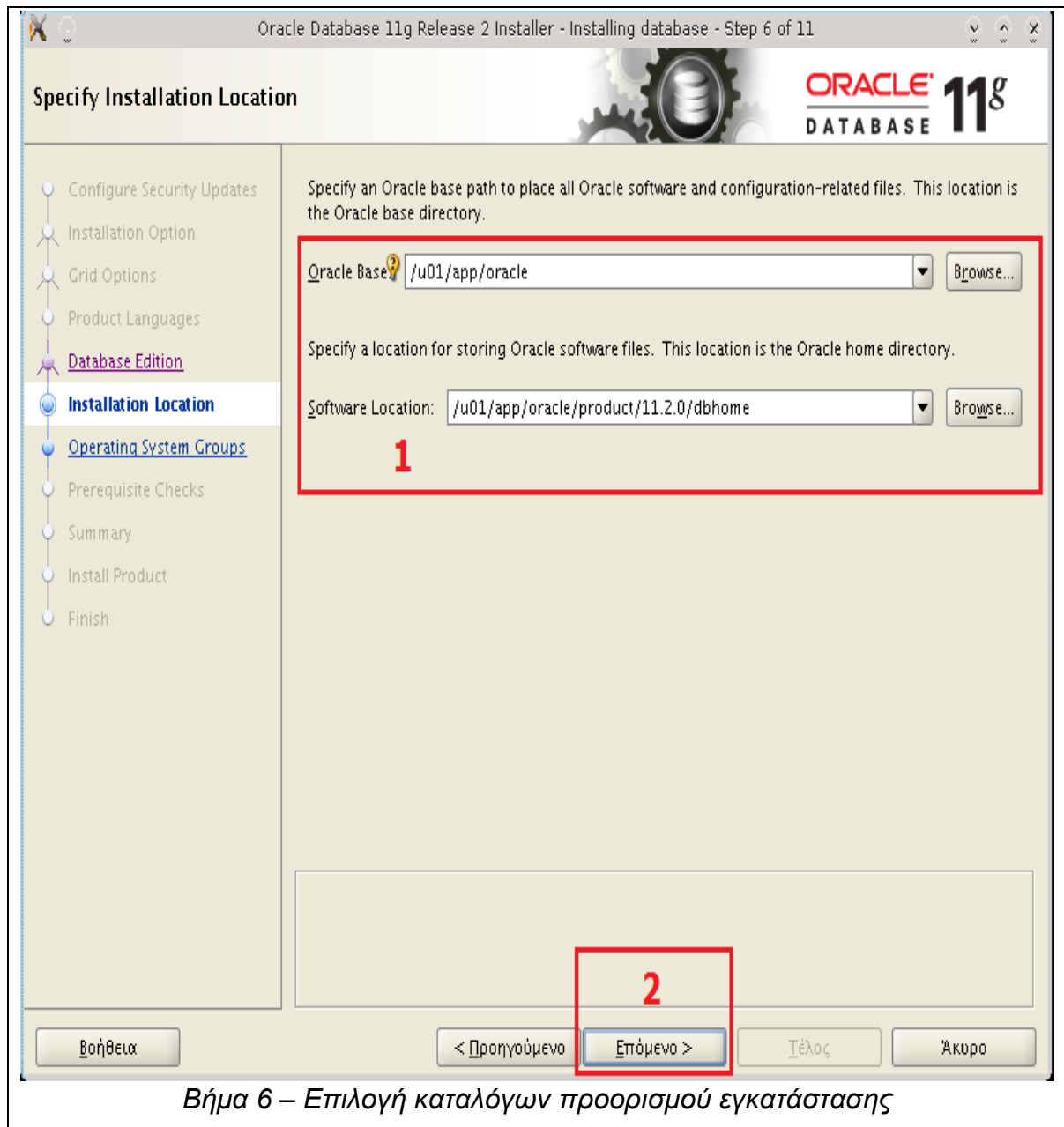
Σε αυτήν την οθόνη μας ζητάει να επιλέξουμε τον τύπο των βάσεων δεδομένων που θα τρέχουν στο σύστημα. Οι επιλογές μας είναι 2, δηλαδή βάσεις ενός Instance και βάσεις με πολλά Instances. Η δεύτερη επιλογή είναι για την τεχνολογία Oracle RAC (Real Application Cluster) όπου χρησιμοποιούνται περισσότεροι Servers. Εμείς επειδή διαθέτουμε έναν, θα επιλέξουμε το πρώτο (1). Τέλος πατάμε επόμενο (2) για να πάμε παρακάτω.



Σε αυτό το παράθυρο καλούμαστε να επιλέξουμε γλώσσες για το προϊόν μας. Εισάγετε ελληνικά και αγγλικά (1) και πατήστε επόμενο (2) για να συνεχίσουμε.



Εδώ καλούμαστε να επιλέξουμε την έκδοση των βάσεων δεδομένων του συστήματός μας. Επιλέγουμε «Enterprise Edition» η οποία θα είναι ικανή να δεχτεί συνδέσεις από πολλούς χρήστες και επίσης παρέχει πρόσθετα χαρακτηριστικά ασφαλείας (1). Επιλέξτε κάποια από τα παρακάτω αν δεν προσδοκάτε μεγάλο αριθμό συνδέσεων ή δεν σας ανησυχούν τα θέματα ασφαλείας. Πατήστε επόμενο για να συνεχίσουμε (2).



Σε αυτήν την οθόνη μας ζητάει να επιλέξουμε τους καταλόγους προορισμού της εγκατάστασης. Εμείς επειδή δημιουργήσαμε αυτούς τους καταλόγους από πρώτο χέρι, ο Installer τους βρήκε και τους συμπλήρωσε αυτόματα στα Textboxes (1). Εμείς απλά πατάμε επόμενο για να πάμε παρακάτω (2).

Προσοχή!

Οι κατάλογοι που δημιουργήσαμε πριν υποτίθεται πως είναι κενοί. Αν για οποιοδήποτε λόγο βάλατε κάποιο αρχείο σε αυτούς τους καταλόγους, διαγράψτε το διαφορετικά ο Installer θα παραπονεθεί πως οι κατάλογοι δεν είναι κενοί.

Βήμα 7 – Επιλογή καταλόγου προορισμού για Oracle Inventory

Σε αυτήν την οθόνη επιλέγουμε τον κατάλογο προορισμού για την τοποθέτηση των αρχείων εγκατάστασης. Ο κατάλογος δημιουργήθηκε από εμάς (orainventory) οπότε ο Installer τον είδε και συμπλήρωσε αυτόματα το πεδίο (1). Στην συνέχεια πρέπει να επιλέξουμε την ομάδα χρηστών του λειτουργικού που θα έχει δικαιώματα στον κατάλογο αυτό. Επιλέγουμε την ομάδα oinstall (2). Τέλος πατάμε επόμενο για να πάμε παρακάτω(3).

Oracle Database 11g Release 2 Installer - Installing database - Step 8 of 12

Privileged Operating System Groups

Configure Security Updates
Installation Option
Grid Options
Product Languages
Database Edition
Installation Location
Create Inventory
Operating System Groups
Prerequisite Checks
Summary
Install Product
Finish

SYSDBA and SYSOPER privileges are required to create a database using operating system (OS) authentication. Membership in OSDBA grants the SYSDBA privilege, and membership in OSOPER grants the SYSOPER privilege, which is a subset of SYSDBA privileges. Select the name of the OSDBA group to grant the SYSDBA privilege. You must be a member of this group.

Database Administrator (OSDBA) Group: dba

Database Operator (OSOPER) Group: oinstall

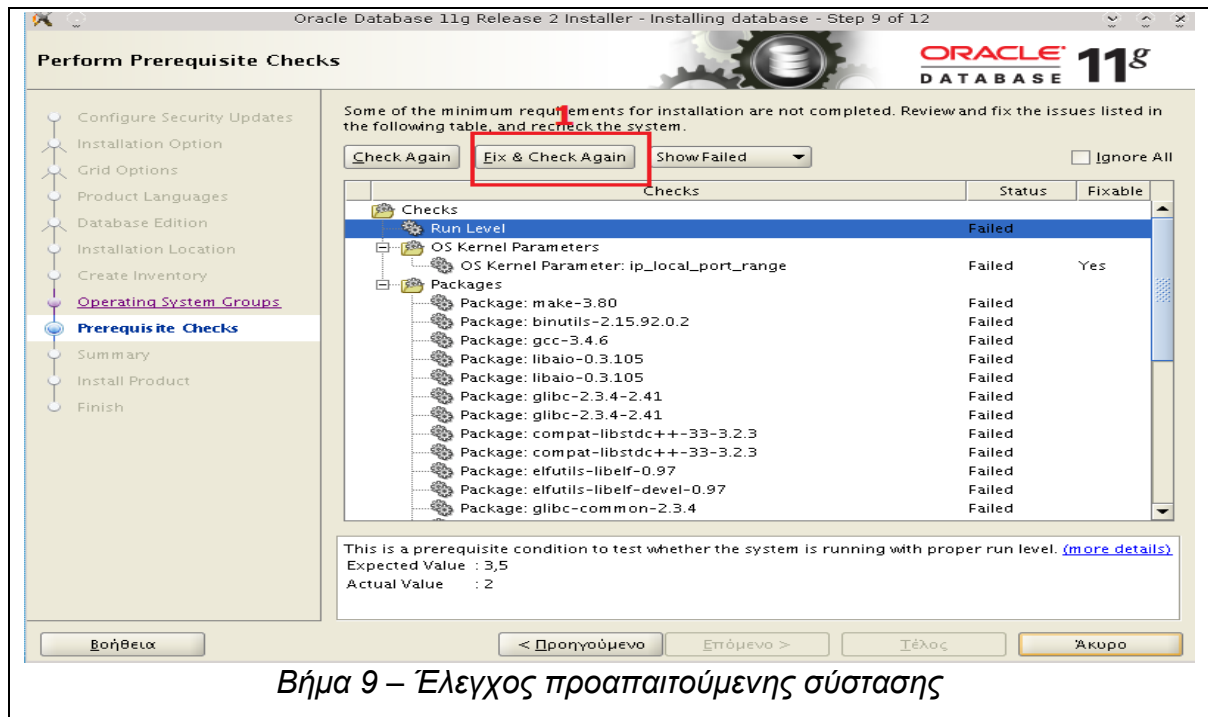
1

2

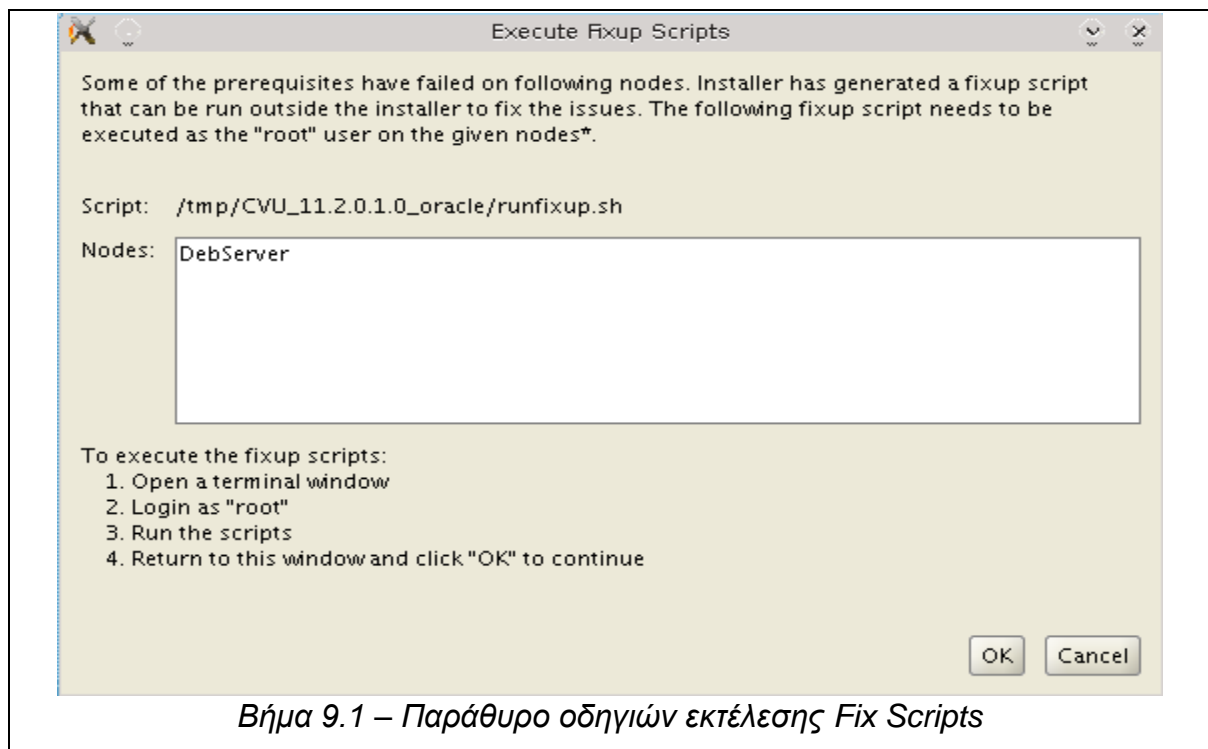
Βοήθεια < Προηγούμενο Επόμενο > Τέλος Άκυρο

Βήμα 8 – Συσχετισμός Διαχειριστών με ομάδες του λειτουργικού συστήματος

Στην σελίδα αυτή επιλέγουμε τις ομάδες των οποίων τα μέλη θα έχουν δικαιώματα διαχειριστή (SYSDA) και χειριστή (SYSOPER). Εμείς στον χρήστη Oracle δώσαμε δύο ομάδες την dba και oinstall οπότε ως ομάδα διαχειριστών (OSDBA) δίνουμε την dba και στην δεύτερη την oinstall (1). Τέλος πατάμε επόμενο (2).



Σε αυτό το παράθυρο αρχίζουν οι δυσκολίες. Η Oracle δεν είναι ευχαριστημένη από την τρέχουσα σύσταση του υπολογιστή (σε θέμα λογισμικού) και παραπονιέται. Εμείς θα πατήσουμε στο κουμπί «Fix & Check Again» για να διορθώσουμε κάποια πράγματα (1), και θα εμφανιστεί τα παρακάτω παράθυρο.



Το παράθυρο αυτό μας δίνει οδηγίες για το πώς να εκτελέσουμε το Script Διόρθωσης (Fix Script) το οποίο «μαγικά» θα διορθώσει κάποιες από τις προαπαιτούμενες συστάσεις που περιμένει ο Installer της Oracle να συμφωνήσουν μαζί του. Εδώ θα χρειαστεί να ανοίξετε δεύτερο terminal και να συνδεθείτε ως χρήστες root για να εκτελέσετε το Script.

```
root@debserver# /tmp/CVU_11.2.0.1.0_oracle/runfixup.sh
```

Troubleshoot

Το Script δεν εκτελείται και ειδοποιεί για συντακτικά σφάλματα του τύπου «[: Unexpected Operator».

Λύση⁵

Αυτό συμβαίνει γιατί το **Shebang** (!) στην αρχή του Script δείχνει ότι το Shell που θα εκτελέσει τις εντολές του Script είναι το /bin/sh. Εμείς όμως το τρέχουμε μέσα από τον /bin/bash. Ανοίξτε λοιπόν το Script με τον Gnu Nano και αλλάξτε το Shebang από «#!/bin/sh» σε «#!/bin/bash».

Επειδή ο κατάλογος αυτός έχει και δεύτερο script, το orarun.sh το οποίο θα εκτελεστεί και αυτό μαζί με το προηγούμενο, πρέπει να κάνετε το ίδιο και σε αυτό. Ανοίξτε το με τον Gnu Nano και αλλάξτε το Shebang από «#!/bin/sh» σε «#!/bin/bash». Στην παρακάτω εικόνα το δείχνουμε:

```
#!/bin/bash
#
# $Header: oui/prov/fixup/linux/orarun.sh /main/2 2009/06/24 04:27:33 sandgoya Exp $
#
# orarun.sh
#
# Copyright (c) 2005, 2009, Oracle and/or its affiliates. All rights reserved.
```

Αφού αλλάξετε τα Shebang και αποθηκεύσετε τα scripts , εκτελέστε το runfixup.sh ξανά με την εντολή:

```
root@debserver# /tmp/CVU_11.2.0.1.0_oracle/runfixup.sh
```

⁵ Oracle OTN Forums → Thread install oracle 11g client in ubuntu 12.04 64bit → “ Either remove #!/bin/sh from the tops of the runfixup.sh and orarun.sh scripts or make a softlink "sudo ln -sf /bin/bash /bin/sh" – Site(<https://forums.oracle.com/thread/2557006>)

Αφού εκτελέσουμε το Script, επιστρέφουμε στο προηγούμενο παράθυρο με τις οδηγίες (βήμα 9.1) και πατάμε OK. Θα επιστρέψουμε στο παράθυρο ελέγχου προαπαιτούμενων συστάσεων (Βήμα 9).

The screenshot shows the Oracle Database 11g Release 2 Installer window at Step 9 of 12, titled "Perform Prerequisite Checks". The window displays a list of checks that are not completed. The "Run Level" check is selected, and its details are shown below the list. The "Ignore All" checkbox is checked, and the "Next" button is highlighted.

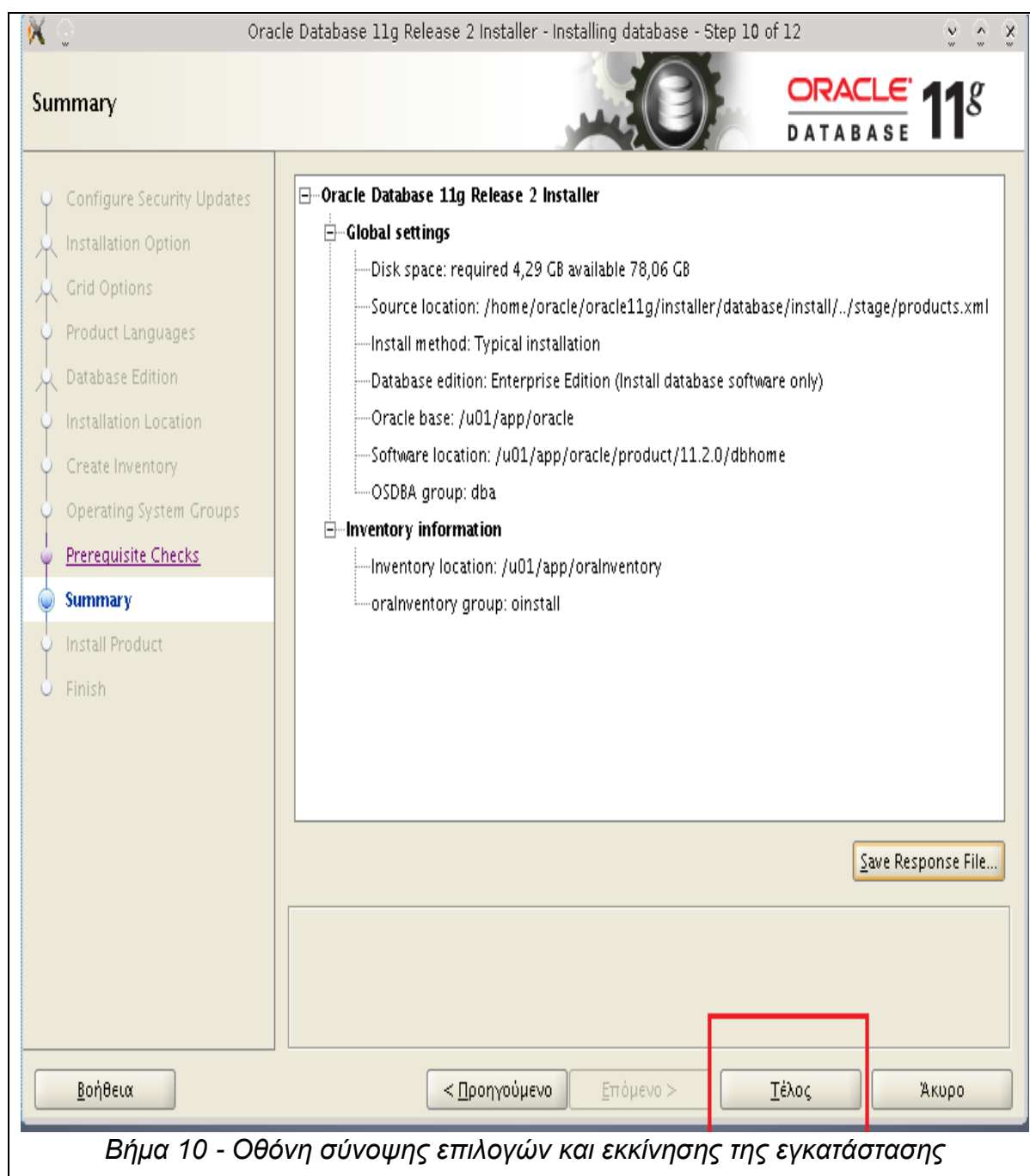
Checks	Status	Fixable
Run Level	Ignored	
Package: make-3.80	Ignored	
Package: binutils-2.15.92.0.2	Ignored	
Package: gcc-3.4.6	Ignored	
Package: libaio-0.3.105	Ignored	
Package: libaio-0.3.105	Ignored	
Package: glibc-2.3.4-2.41	Ignored	
Package: glibc-2.3.4-2.41	Ignored	
Package: compat-libstdc++-33-3.2.3	Ignored	
Package: compat-libstdc++-33-3.2.3	Ignored	
Package: elfutils-libelf-0.97	Ignored	
Package: elfutils-libelf-devel-0.97	Ignored	
Package: glibc-common-2.3.4	Ignored	
Package: glibc-devel-2.3.4	Ignored	
Package: glibc-headers-2.3.4	Ignored	

This is a prerequisite condition to test whether the system is running with proper run level. [\(more details\)](#)
 Expected Value : 3,5
 Actual Value : 2

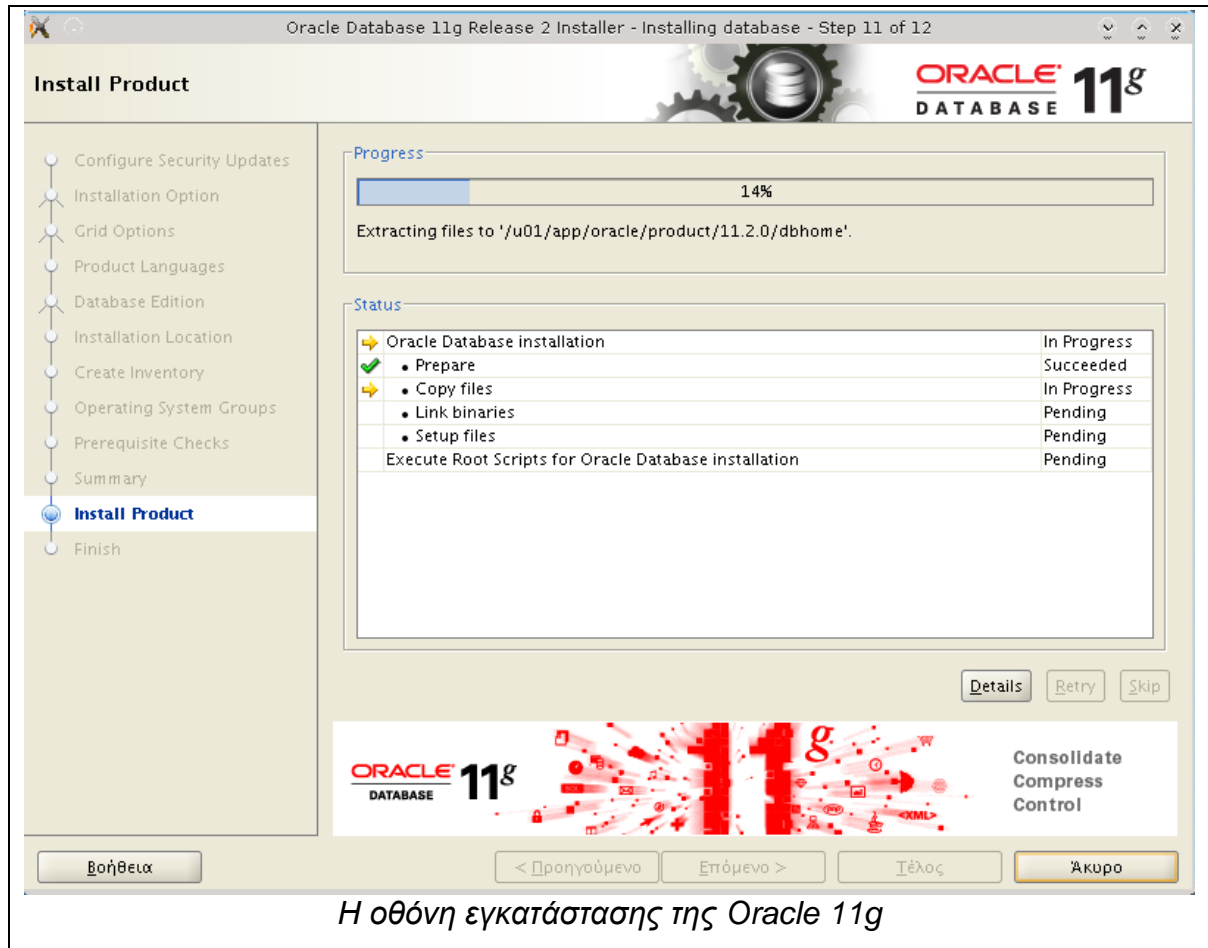
Buttons: Βοήθεια, < Προηγούμενο, Επόμενο >, Τέλος, Άκυρο

Βήμα 9.2 – Επανάλεγχος συστάσεων

Ο Installer συνεχίζει να μην είναι ευχαριστημένος από την σύσταση του συστήματος, αλλά αυτή τη φορά παραπονιέται για λιγότερες μη-συμφωνούμενες προαπαιτήσεις. Οι υπόλοιπες προαπαιτήσεις δεν απασχολούνε πολύ, οπότε πατάμε στο Checkbox «Ignore All» (1) για να τις αγνοήσει και να προχωρήσει η εγκατάσταση, ενεργοποιώντας και το κουμπί «Επόμενο» (2).



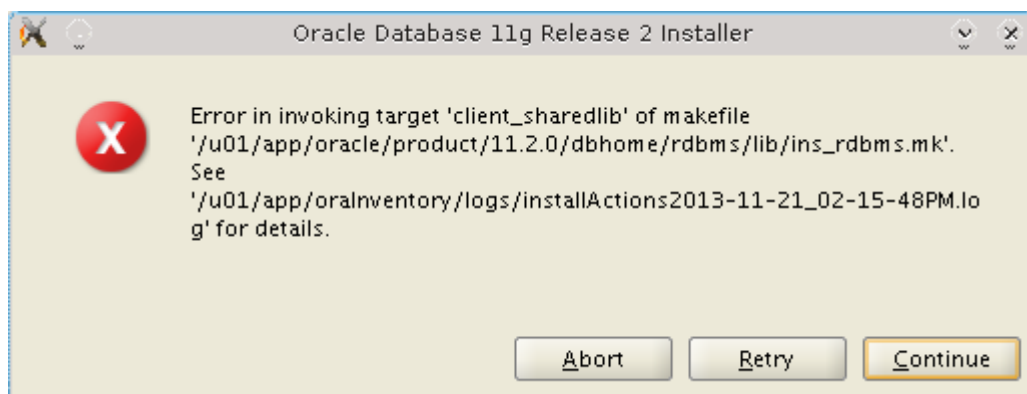
Στην οθόνη αυτή βλέπουμε την σύνοψη των επιλογών που κάναμε ως τώρα. Πατάμε «Τέλος» για να ξεκινήσει η εγκατάσταση της Oracle 11g στον υπολογιστή μας.



Καθίστε αναπαυτικά και περιμένετε να ολοκληρωθεί η εγκατάσταση της Oracle στο σύστημα.

Troubleshoot

Στο 84% μου εμφανίζει το παρακάτω μήνυμα λάθους:



Λύση⁶:

Ανοίξτε ένα δεύτερο Terminal και συνδεθείτε ως χρήστες Root. Δημιουργήστε ένα κατάλογο μέσα στον /usr με όνομα «lib64» ως εξής:

```
root@debserver# mkdir /usr/lib64
```

Στην συνέχεια θα πρέπει να δημιουργήσετε Symbolic Links κάποιων αρχείων στον κατάλογο αυτόν γιατί ο Installer εκεί τα ψάχνει:

```
root@debserver# ln -s /usr/lib/x86_64-linux-gnu/libpthread_nonshared.a  
/usr/lib64/libpthread_nonshared.a
```

```
root@debserver# ln -s /usr/lib/x86_64-linux-gnu/libc_nonshared.a /usr/lib64/
```

```
root@debserver# ln -s /lib/x86_64-linux-gnu/libgcc_s.so.1 /lib64/
```

```
root@debserver# ln -s /usr/lib/x86_64-linux-gnu/libstdc++.so.6 /usr/lib64/
```

```
root@debserver# ln -s /usr/bin/basename /bin/basename
```

Τέλος (για να αποφύγετε και πιθανό σφάλμα «undefined reference to nnfyboot» δώστε τις παρακάτω εντολές ως χρήστης root:

```
root@debserver# export
```

```
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome
```

```
root@debserver# cd $ORACLE_HOME/lib
```

```
root@debserver# ln -s libclient10.a libagtsh.a
```

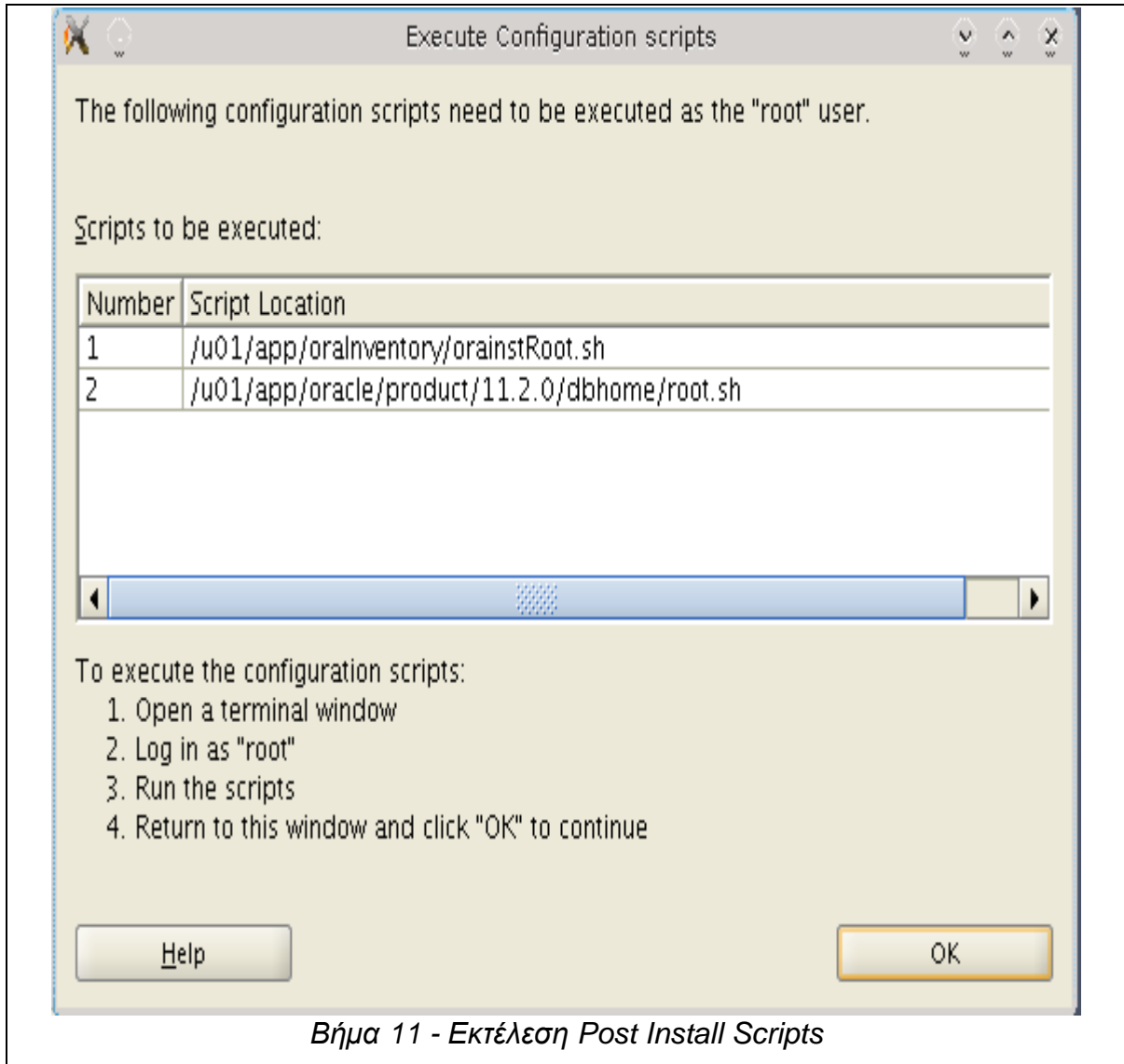
```
root@debserver# cd $ORACLE_HOME
```

```
root@debserver# LD_LIBRARY_PATH=$ORACLE_HOME/lib ./bin/genagtsh  
$ORACLE_HOME/lib/libagtsh.so 1.0
```

⁶ Internet Source → Installing Oracle 10gR2 in Debian Squeeze → Linking problems – Site(http://krustev.net/w/articles/Installing_Oracle_10gR2_in_Debian_Squeeze/)

Μετά από όλα αυτά, πατήστε Retry και ο Installer θα συνεχίσει χωρίς σφάλμα.

Αφού η εγκατάσταση φτάσει στο 94% θα σας εμφανίσει το παρακάτω παράθυρο:

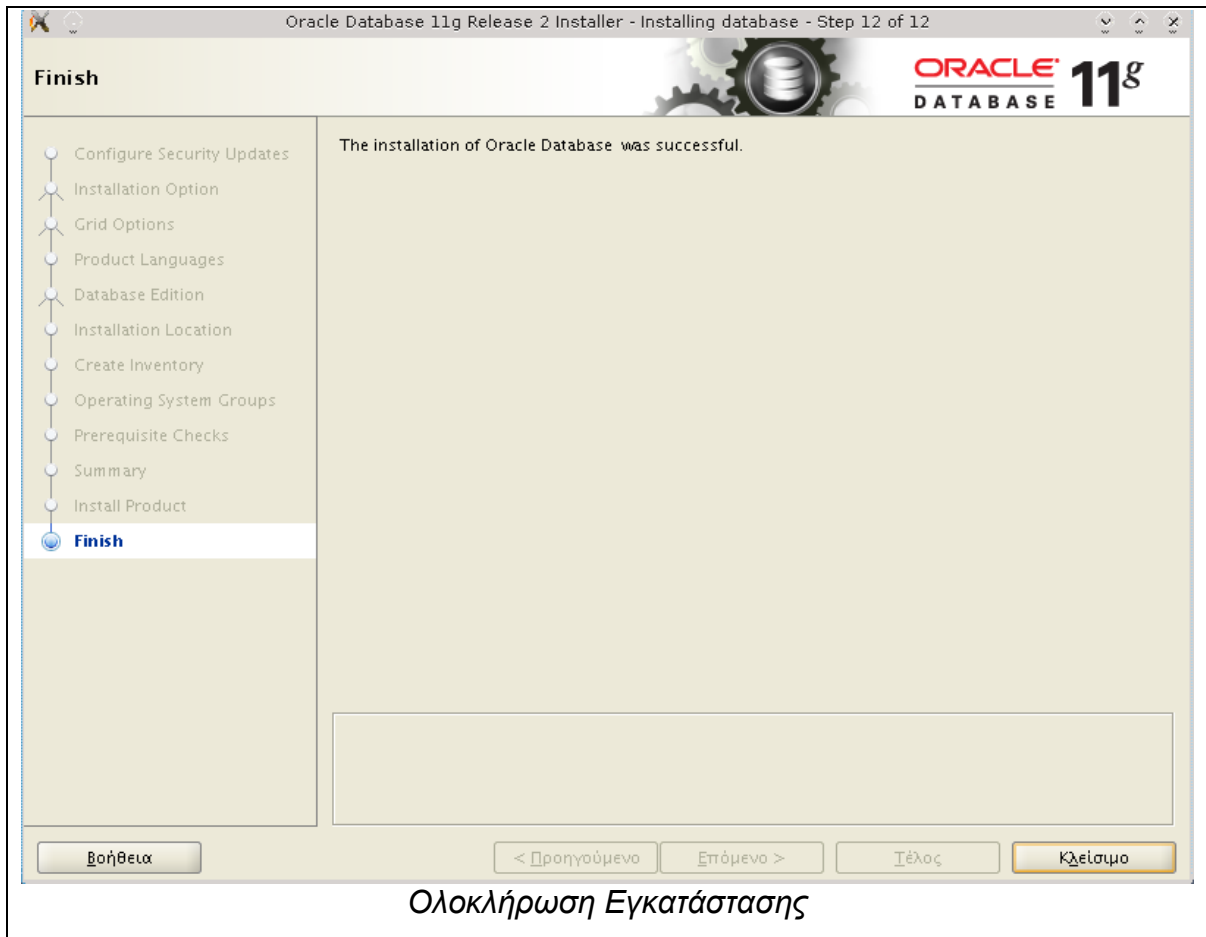


Σε αυτό το σημείο, ως χρήστες root πρέπει να εκτελέσετε τα Scripts που σας δείχνει στη λευκή περιοχή του παραθύρου. Ανοίξτε ένα terminal δεύτερο, και συνδεθείτε ως root για να εκτελέσετε τα scripts:

```
root@debserver# /u01/app/orainventory/orainstRoot.sh
```

```
root@debserver# /u01/app/oracle/product/11.2.0/dbhome/root.sh
```

Η εγκατάσταση ολοκληρώθηκε. Λογικά θα πρέπει να δείτε το παρακάτω παράθυρο:



Η Oracle 11g είναι πλέον εγκατεστημένη στο σύστημά σας. Μπορείτε να την απολαύσετε! Μπορείτε να κλείσετε τα terminal με ασφάλεια. Με για σας!

Ολοκληρώνοντας...

Η Oracle 11g εγκαταστάθηκε με επιτυχία στο σύστημά σας. Εφόσον έχουμε εγκαταστήσει μόνον το λογισμικό της Oracle, δεν υπάρχει ούτε βάση δεδομένων διαθέσιμη, ούτε δικτυακή υπηρεσία για να δέχεται συνδέσεις. Στο επόμενο κεφάλαιο θα ασχοληθούμε με την παραμετροποίηση του Server μας ώστε να του δώσουμε «ζωή» με την παραμετροποίηση της Oracle.

Βιβλιογραφία Κεφαλαίου

A/A	Άρθρο/Πηγή	Ενότητα	Ιστοσελίδα
1	Debian Administration	Installing Oracle11 and Oracle12 on Debian Wheezy, Squeeze	http://www.debian-administration.org/articles/656
2	Installing Oracle 10gR2 in Debian Squeeze	Linking problems	http://krustev.net/w/articles/Installing_Oracle_10gR2_in_Debian_Squeeze/
3	HOWTO install Oracle 11g on Ubuntu Linux 12.04 (Precise Pangolin) 64bits	Preparing the system	http://www.makina-corpus.org/blog/howto-install-oracle-11g-ubuntu-linux-1204-precise-pangolin-64bits
4	Oracle OTN Forums	Thread "install oracle 11g client in ubuntu 12.04 64bit"	https://forums.oracle.com/thead/2557006

ΚΕΦΑΛΑΙΟ 4 – ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΤΟΥ SERVER

Σε αυτό το κεφάλαιο θα ασχοληθούμε αποκλειστικά με την παραμετροποίηση του Oracle Server ώστε να μπορεί να προσφέρει αποδοτικά τις υπηρεσίες του στους χρήστες που συνδέονται σε αυτόν. Περιγράφονται αναλυτικά όλα τα απαραίτητα βήματα που απαιτούνται ώστε ο Server να μπορεί να εξυπηρετεί συνδέσεις, καθώς και για την δημιουργία μιας βάσης δεδομένων για τους χρήστες.

4.1 - Δημιουργία μεταβλητών περιβάλλοντος

Οι **μεταβλητές περιβάλλοντος** (Environmental Variables) είναι μεταβλητές του λειτουργικού συστήματος των οποίων οι τιμές χρησιμοποιούνται κατά την διάρκεια εκτέλεσης διεργασιών, και οι τιμές τους είναι γνωστές σε όλη τη διάρκεια εκτέλεσης του λειτουργικό συστήματος. Μια από τις πιο γνωστές ίσως μεταβλητές περιβάλλοντος σε συστήματα Unix και Windows, είναι η μεταβλητή **PATH** η οποία κρατάει ως τιμή, όλα τα paths προς εκτελέσιμα αρχεία που θέλουμε να είναι γνωστά, οπουδήποτε δουλεύουμε μέσα στο λειτουργικό σύστημα. Επίσης παρέχουν και μια ευκολία ως «τρόπον τινά» συντομεύσεις, καθώς είναι πιο σύντομο π.χ. το «cd \$JAVA_HOME» από το «cd /home/user/programs/jdk-1.7.0_45/».

Στην Oracle οι μεταβλητές περιβάλλοντος, θα λέγαμε ότι είναι περισσότερο υποχρεωτικές παρά προαιρετικές. Ο λόγος που χρειάζονται είναι και για την διευκόλυνση του χρήστη αλλά και για την σωστή εκτέλεση των απαραίτητων, γι αυτήν, διεργασιών. Θα δούμε όμως παρακάτω με περισσότερη λεπτομέρεια τι ακριβώς εννοούμε.

Οι βασικές μεταβλητές περιβάλλοντος που χρειαζόμαστε για την Oracle είναι οι εξής τέσσερις:

- 1) **ORACLE_BASE** = Η μεταβλητή που κρατάει το path προς τον κατάλογο που βρίσκεται εγκατεστημένη η Oracle Database. Το δικό μας είναι «/u01/app/oracle»
- 2) **ORACLE_HOME** = Η μεταβλητή που κρατάει το path προς τον κατάλογο που βρίσκονται τα Binaries της Oracle. Μέσω αυτής θα μπορούμε να έχουμε πρόσβαση στα βασικά εργαλεία μας. Το δικό μας path είναι «/u01/app/oracle/product/11.2.0/dbhome»
- 3) **ORACLE_SID** = Η μεταβλητή που κρατάει το **Instance ID** της διεργασίας της βάσης δεδομένων. Είναι πολύ βασική γιατί προσδιορίζει μοναδικά την βάση δεδομένων που θα συνδέονται οι χρήστες. Είναι γνωστή και ως System ID.
- 4) **NLS_LANG** = Η μεταβλητή που ορίζει την γλώσσα στο SQL Plus, του προγράμματος δηλαδή που θα χρησιμοποιήσουμε για να διαχειριζόμαστε την βάση από τον Server.

Αυτές λοιπόν οι τέσσερις μεταβλητές είναι οι de-facto μεταβλητές περιβάλλοντος που χρειάζεται η Oracle για να λειτουργήσει σωστά. Θα πρέπει λοιπόν να τις δημιουργήσουμε αλλά παράλληλα να φροντίσουμε να παραμένουν γνωστές στο σύστημα ακόμα και μετά από shutdown ή reboot, καθώς οι (περισσότερες) μεταβλητές περιβάλλοντος χάνονται στον τερματισμό του λειτουργικού συστήματος.

Λύση στην παραπάνω πρόκληση είναι η δημιουργία (αν δεν υπάρχουν ήδη) και τροποποίηση των αρχείων **.bashrc** και **.profile** τα οποία διαβάζονται από το κέλυφος (shell) του λειτουργικού (στην συγκεκριμένη περίπτωση του **Bash Shell**), κάθε φορά που ο χρήστης ξεκινά **συνεδρία** (session) με αυτό. Μέσα σε αυτό βάζουμε ότι θέλουμε να ξεκινάει μαζί με την εκκίνηση συνεδρίας του χρήστη με το κέλυφος, μπορούμε για παράδειγμα κάθε φορά που ο χρήστης συνδέεται στο shell να του εμφανίζει το μήνυμα «Welcome». Οπότε η δημιουργία των τριών μεταβλητών περιβάλλοντος μέσω αυτών των αρχείων αποτελεί μια πολύ καλή λύση, γιατί το σκεπτικό είναι οι μεταβλητές αυτές να δημιουργούνται και να τους

αποδίδεται τιμή σε κάθε συνεδρία του χρήστη με το κέλυφος. Και φυσικά αφού αφορούν τον χρήστη που θα διαχειρίζεται την βάση δεδομένων, μιλάμε για τον χρήστη **oracle**.

Τα εν λόγω αρχεία βρίσκονται στο home directory κάθε χρήστη που υπάρχει στο σύστημα και εφόσον ξεκινούν από τελεία (.) πρόκειται για **κρυφά αρχεία**. Εμείς θέλουμε λοιπόν να τροποποιήσουμε αυτά τα αρχεία και να δώσουμε τιμές στις εν λόγω μεταβλητές. Συνδεόμαστε λοιπόν στο σύστημα ως χρήστες oracle και γράφουμε τα εξής:

```
user@debserver$ su oracle # ← Σύνδεση ως Oracle χρήστης  
oracle@debserver$ cd ~ # ← Μετάβαση στο home directory του Oracle  
oracle@debserver$ ls -altr | grep .profile # ← Έλεγχος ύπαρξης .profile (1)  
oracle@debserver$ ls -altr | grep .bashrc # ← έλεγχος ύπαρξης .bashrc (2)
```

Με τις εντολές 1 και 2 που μαρκάρουμε, ελέγχουμε αν τα αρχεία .bashrc και .profile υπάρχουν στο home directory του χρήστη oracle (/home/oracle). Αν υπάρχουν μπορούμε να τα τροποποιήσουμε άμεσα, αν δεν υπάρχουν όμως πρέπει να τα δημιουργήσουμε. Μπορούμε να δημιουργήσουμε και να τροποποιήσουμε τα αρχεία αυτά με μία κίνηση, χρησιμοποιώντας τον GNU Nano, τον terminal based text editor που συνοδεύει τις περισσότερες διανομές Linux (συμπεριλαμβανομένου και του Debian). Θα ξεκινήσουμε πρώτα από το αρχείο .bashrc. Γράφουμε λοιπόν:

```
oracle@debserver$ nano .bashrc
```

Τότε μεταβαίνουμε στην οθόνη επεξεργασίας κειμένου του Nano. Τώρα μπορούμε να εισάγουμε περιεχόμενο στο αρχείο μας. Δίνουμε τις κάτωθι γραμμές με την σειρά που φαίνεται:

```
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome #1  
ORACLE_BASE=/u01/app/oracle #2  
ORACLE_SID=test #3
```

```
NLS_LANG=english; #4

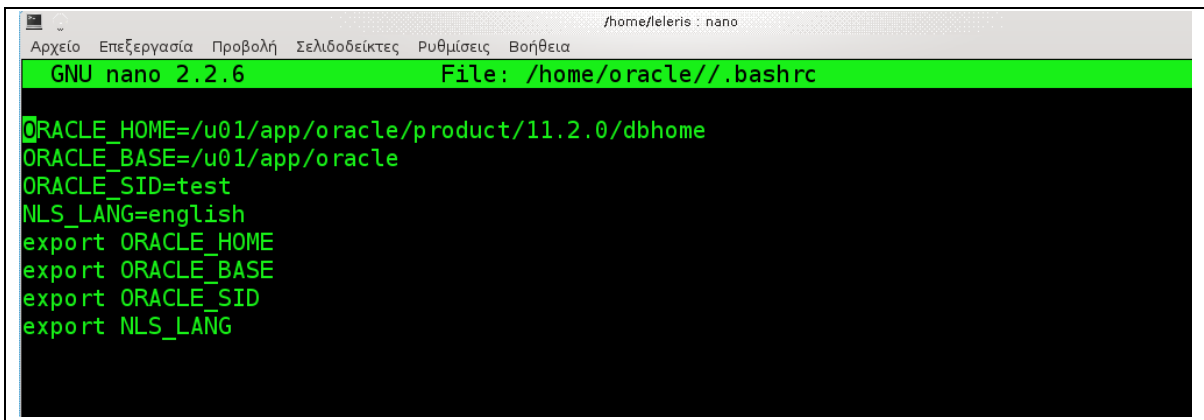
export ORACLE_HOME #5

export ORACLE_BASE #6

export ORACLE_SID #7
```

Οι γραμμές αυτές επεξηγούνται με την σειρά. Η πρώτη (1) είναι η μεταβλητή **ORACLE_HOME** όπου της δίνουμε ως τιμή, το path προς τον κατάλογο με τα Binaries της Oracle. Η δεύτερη (2) είναι η **ORACLE_BASE** όπου τις δίνουμε ως τιμή το path προς τον κατάλογο που εγκαταστάθηκε η Oracle. Η τρίτη (3) είναι η **ORACLE_SID** όπου τις δίνουμε το Instance ID που θέλουμε να έχει η βάση μας. Για λόγους παραδείγματος δώσαμε το SID «test» αλλά μπορείτε να δώσετε ότι άλλο εσείς θέλετε. Εν συνεχεία, δίνουμε την τιμή «english» στην μεταβλητή NLS_LANG (4). Τέλος, οι τρεις τελευταίες γραμμές (5,6,7) εξάγουν τις μεταβλητές περιβάλλοντος με τις τιμές που τους εκχωρήθηκαν και πλέον είναι ενεργές.

Αφού εισάγαμε αυτές τις γραμμές, πρέπει να αποθηκεύσουμε το αρχείο. Πατάμε Ctrl + X για να το πράξουμε και στην προτροπή «Save Modified Buffer ?» πατάμε Y για YES και Enter. Το αρχείο δημιουργήθηκε. Παρακάτω βλέπουμε και την τροποποίηση του αρχείου σε περιβάλλον GNU Nano:



```
GNU nano 2.2.6 File: /home/oracle/.bashrc
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome
ORACLE_BASE=/u01/app/oracle
ORACLE_SID=test
NLS_LANG=english
export ORACLE_HOME
export ORACLE_BASE
export ORACLE_SID
export NLS_LANG
```

Η οθόνη του GNU nano στην δημιουργία και τροποποίηση του bashrc

Επιβεβαιώνουμε την αποθήκευση του αρχείου, τυπώνοντας τα περιεχόμενά του στο STDOUT με την εντολή:

```
oracle@debserver$ cat .bashrc
```

Θα πρέπει να πάρουμε παρόμοια οθόνη με την παρακάτω:

Επιβεβαίωση αποθήκευσης αρχείου με την εντολή cat

Το αρχείο .bashrc δημιουργήθηκε επιτυχώς. Τώρα πρέπει να κάνουμε επακριβώς τις ίδιες ενέργειες για το αρχείο .profile.

```
oracle@debserver$ nano .profile
```

Εισάγουμε ακριβώς τις ίδιες γραμμές που εισάγαμε στο .bashrc και αφού τελειώσουμε, πατούμε Ctrl+X για να αποθηκεύσουμε το αρχείο μας. Κάνουμε και cat για να δούμε τα περιεχόμενα του αρχείου αν θέλουμε να είμαστε σίγουροι ότι είναι ok.

Τώρα που τα αρχεία μας είναι έτοιμα, πρέπει να τους αλλάξουμε τα δικαιώματα ώστε μόνον ο χρήστης oracle να μπορεί να τα τροποποιεί και οι υπόλοιποι χρήστες μόνον να τα βλέπουνε. Εισάγουμε τις παρακάτω εντολές στο terminal:

```
oracle@debserver$ chmod 744 .bashrc
```

```
oracle@debserver$ chmod 744 .profile
```

Τώρα, θα πρέπει να αποσυνδεθούμε από την συνεδρία με το shell και να ξανασυνδεθούμε ώστε να διαπιστώσουμε αν οι μεταβλητές που βάλαμε στα αρχεία εκχωρήθηκαν ή όχι. Γράφουμε λοιπόν:

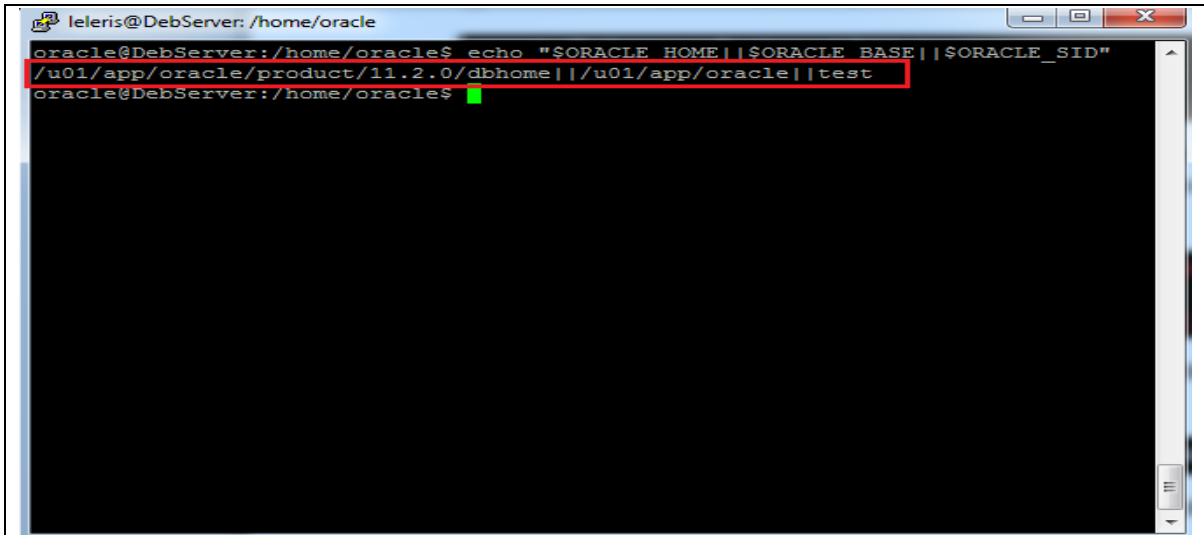
```
oracle@debserver$ exit # ← Αποσύνδεση χρήστη oracle από το shell
```

```
user@debserver$ su oracle # ← Επανασύνδεση ως χρήστης oracle
```


Για να διαπιστώσουμε λοιπόν αν οι μεταβλητές μας είναι εκχωρημένες και γνωστές, γράφουμε τις εξής εντολές:

```
oracle@debserver$ echo "$ORACLE_HOME||$ORACLE_BASE||$ORACLE_SID"
```

Αν όλα πήγαν καλά θα πρέπει να πάρουμε έξοδο όμοια με την παρακάτω:



```
leleris@DebServer: /home/oracle
oracle@DebServer:/home/oracle$ echo "$ORACLE_HOME||$ORACLE_BASE||$ORACLE_SID"
/u01/app/oracle/product/11.2.0/dbhome||/u01/app/oracle||test
oracle@DebServer:/home/oracle$
```

Επιβεβαίωση μεταβλητών περιβάλλοντος με την Echo

Εφόσον έχουμε δημιουργήσει τις μεταβλητές περιβάλλοντος, πλέον μπορούμε να προχωρήσουμε παρακάτω.

4.2 - Δημιουργία Listeners και Name Services

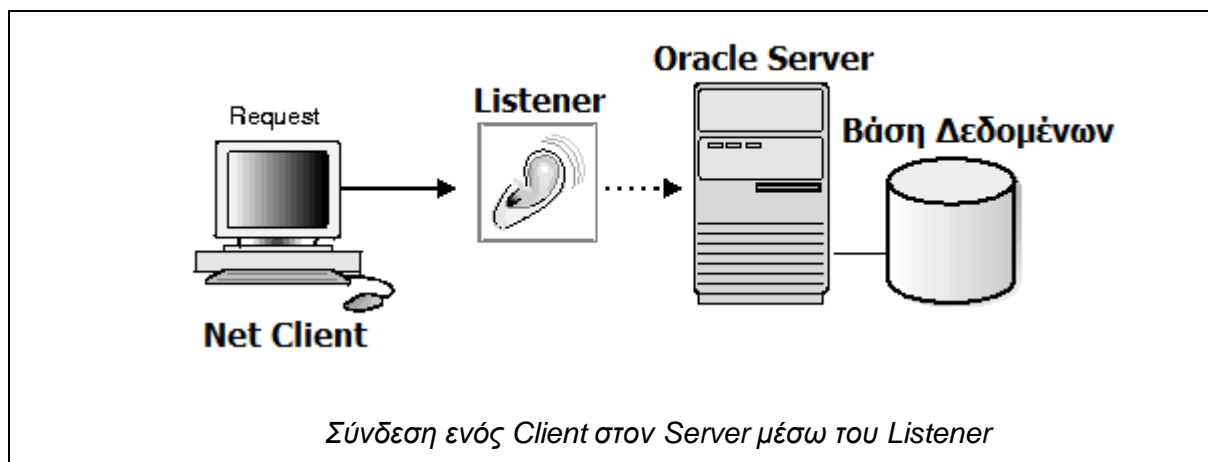
Ορισμός

Ένας **Ακροατής**⁷ (Listener) είναι μια **διεργασία** που επιτρέπει στον Server να «ακούει» σε μια θύρα για εισερχόμενες συνδέσεις. Πρόκειται για πολύ βασικό κομμάτι του εξυπηρετητή καθώς χωρίς αυτόν η σύνδεση ενός χρήστη με τη βάση, δεν μπορεί να επιτευχθεί.

⁷ Oracle Documentation → Configuring and Administering the Listener → “The listener is a separate process that runs on the database server computer. It receives incoming client connection requests...” – Site (http://docs.oracle.com/cd/B10501_01/network.920/a96580/listener.htm)

Ο Listener τρέχει σαν διεργασία στον Server και δέχεται τις συνδέσεις από τους Clients. Τα βασικά χαρακτηριστικά από τα οποία εξαρτάται ο Listener είναι η **θύρα** (port) που ακούει καθώς και το **Instance ID** (SID) της βάσης που τρέχει εκείνη την στιγμή στον Server. Ο Listener όμως δεν χρησιμοποιείται αποκλειστικά και μόνον για τους Clients αλλά και για τον ίδιο τον Server γιατί χάρη σε αυτόν ο διαχειριστής συνδέετε και διαχειρίζεται την βάση δεδομένων, παρόλο που είναι τοπική στον υπολογιστή αυτόν.

Για να συνδεθούμε όμως στην επιθυμητή μας βάση, πρέπει ο Listener να «καταλάβει» σε ποια βάση πρέπει να προωθήσει τη σύνδεση. Αυτό επιτυγχάνεται μέσω της **Ονομασίας Υπηρεσίας** (Name Service), η οποία στην ουσία είναι το **Global όνομα** της βάσης δεδομένων. Στην παρακάτω εικόνα βλέπουμε μια σχηματική αναπαράσταση του ρόλου ενός Listener.



Πρέπει λοιπόν αρχικά να δημιουργήσουμε έναν Listener ώστε να μπορέσουμε πρωτίστως να δημιουργήσουμε μια βάση δεδομένων. Ακολουθούμε τα εξής συνοπτικά βήματα:

- 1) Ορίζουμε ένα μοναδικό **Instance ID** (SID) για την βάση δεδομένων που σκοπεύουμε να δημιουργήσουμε. Η δημιουργία του SID γίνεται μέσα από μεταβλητή περιβάλλοντος **ORACLE_SID**.
- 2) Δημιουργούμε τον Listener που θα ακούει στην **επιθυμητή θύρα** που θέλουμε και με το **επιθυμητό πρωτόκολλο μεταφοράς** (transport layer protocol), μαζί με το κατάλληλο Service Name.

Το βήμα 1 το καλύψαμε στην δημιουργία των αρχείων `.bashrc` και `.profile` ωστόσο όμως για μελλοντική αλλαγή μπορούμε να δώσουμε την παρακάτω εντολή για να αλλάξουμε την τιμή της `ORACLE_SID` (ως χρήστες `oracle`)

```
user@debserver$ su oracle
oracle@debserver$ export ORACLE_SID=test
```

Επαναλαμβάνουμε ότι το παραπάνω βήμα είναι προαιρετικό, αν τα αρχεία `.bashrc` και `.profile` έχουν τροποποιηθεί από εσάς, δεν χρειάζεται να ξανά-ορίσετε την `ORACLE_SID`.

Προσοχή

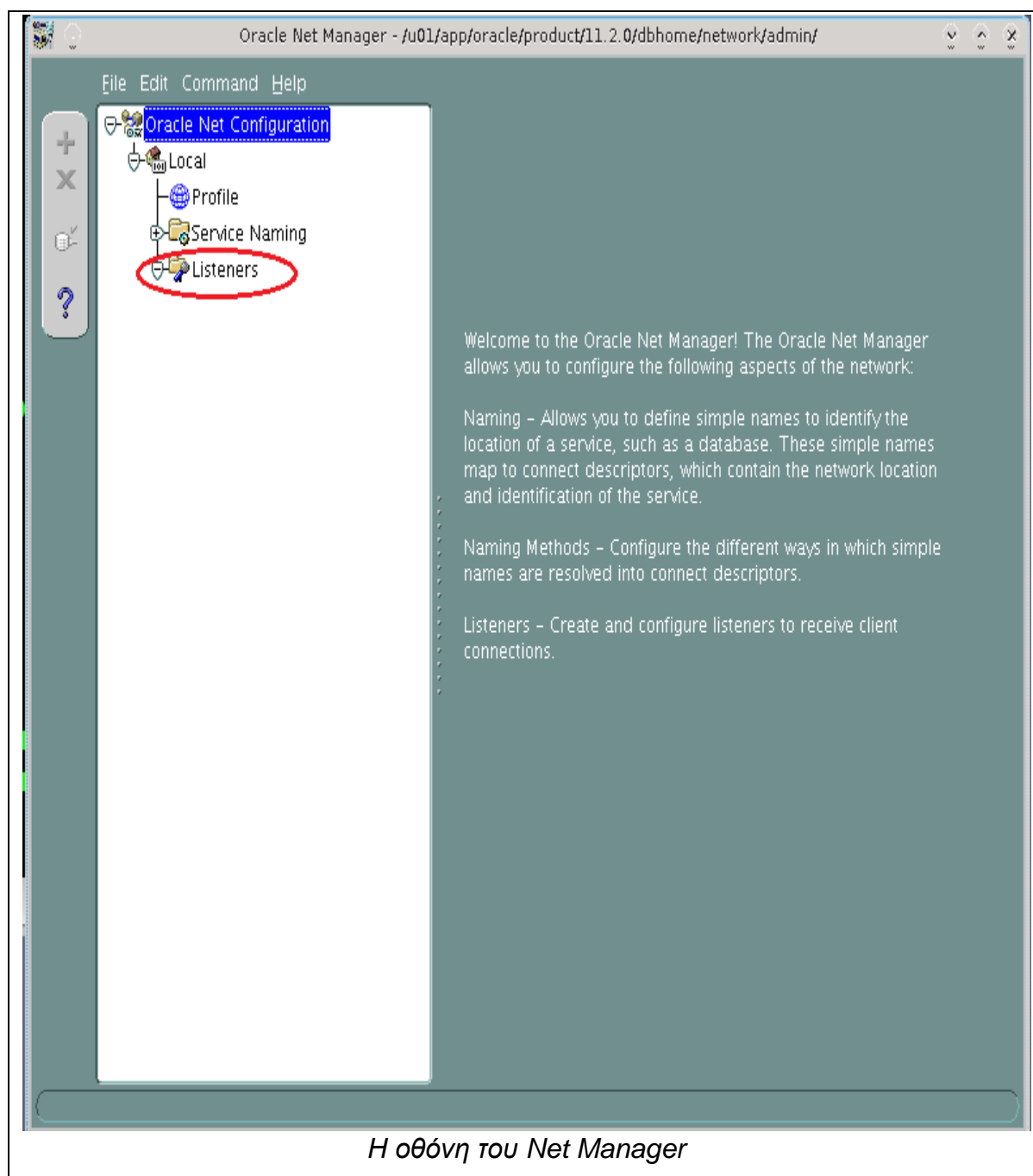
Για την εκτέλεση των παρακάτω απαιτείται είτε η φυσική σας παρουσία μπροστά στον `Server` είτε απομακρυσμένη σύνδεση σε αυτόν μέσω `RDP`. Με `SSH` δεν θα μπορέσετε να ολοκληρώσετε την διαδικασία.

Εφόσον η μεταβλητή `ORACLE_SID` έχει τιμή, μπορούμε να δημιουργήσουμε έναν `Listener`. Η Oracle παρέχει ένα βοηθητικό εργαλείο για την δημιουργία `Listeners`, το **Net Manager** (`netmgr`) το οποίο βρίσκεται στην τοποθεσία `$ORACLE_HOME/bin`.

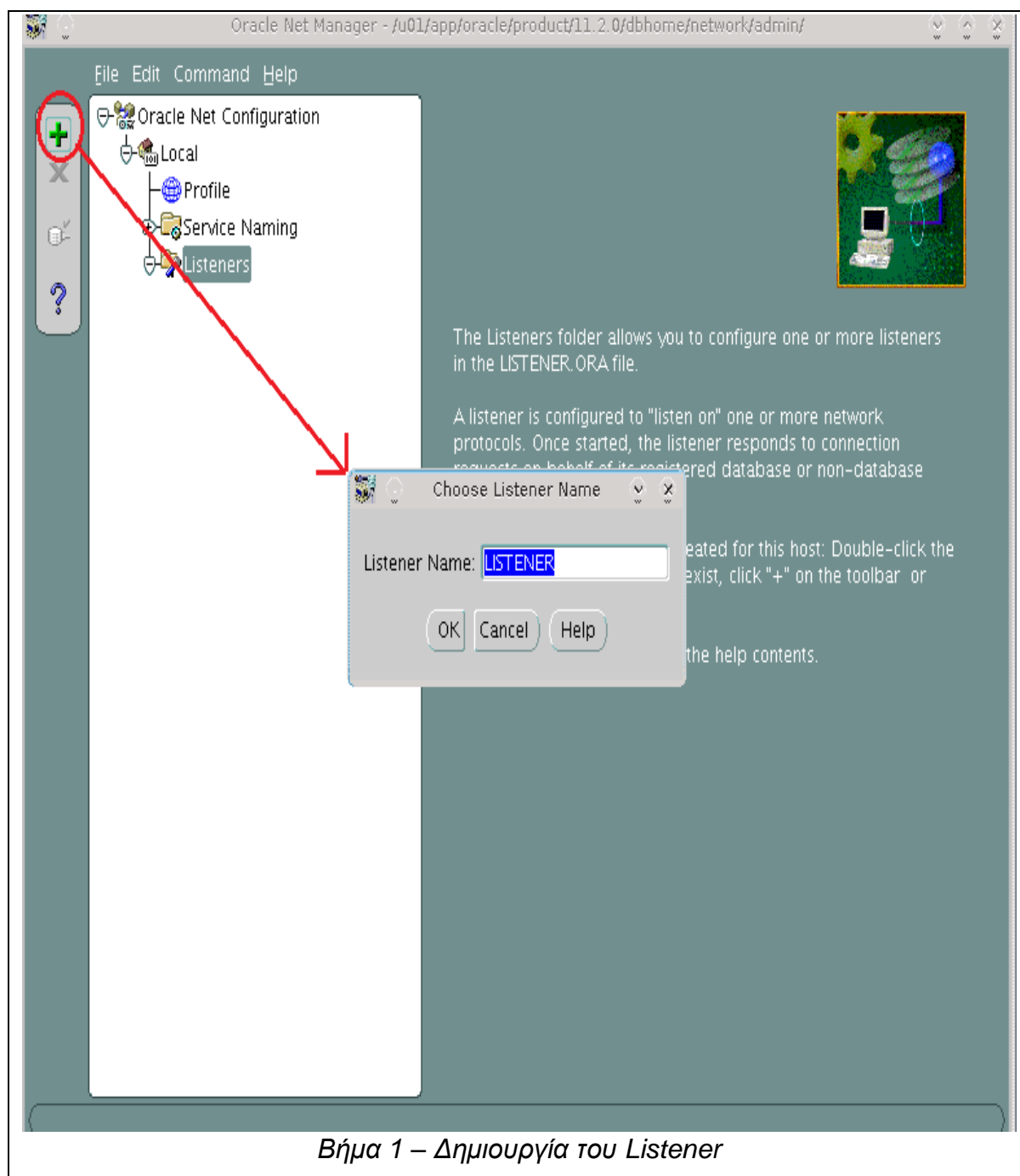
Μεταβαίνουμε λοιπόν σε αυτό με τις εντολές:

```
oracle@debserver$ cd $ORACLE_HOME/bin
oracle@debserver$ ./netmgr
```

Μας εμφανίζεται τότε το παράθυρο του `Net Manager` το οποίο μοιάζει με το παρακάτω:

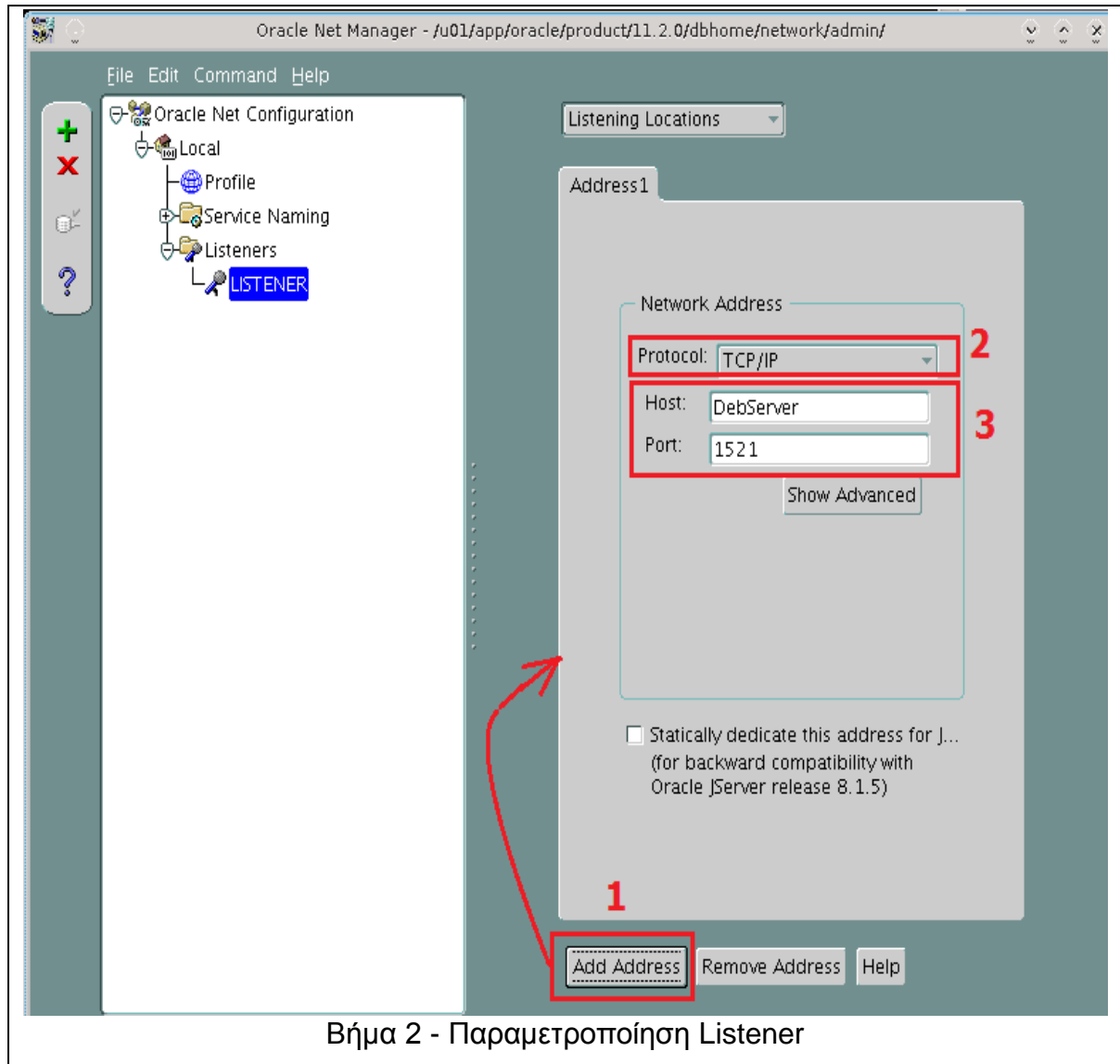


Με το βοηθητικό αυτό εργαλείο μπορούμε να δημιουργήσουμε τον Listener μας. Επιλέγουμε την λίστα «Listeners» που μαρκάρεται στην εικόνα. Πάνω αριστερά, ένας πράσινος σταυρός πλέον είναι ενεργός. Πατάμε κλικ σε αυτόν για να ξεκινήσει η διαδικασία δημιουργίας του Listener:



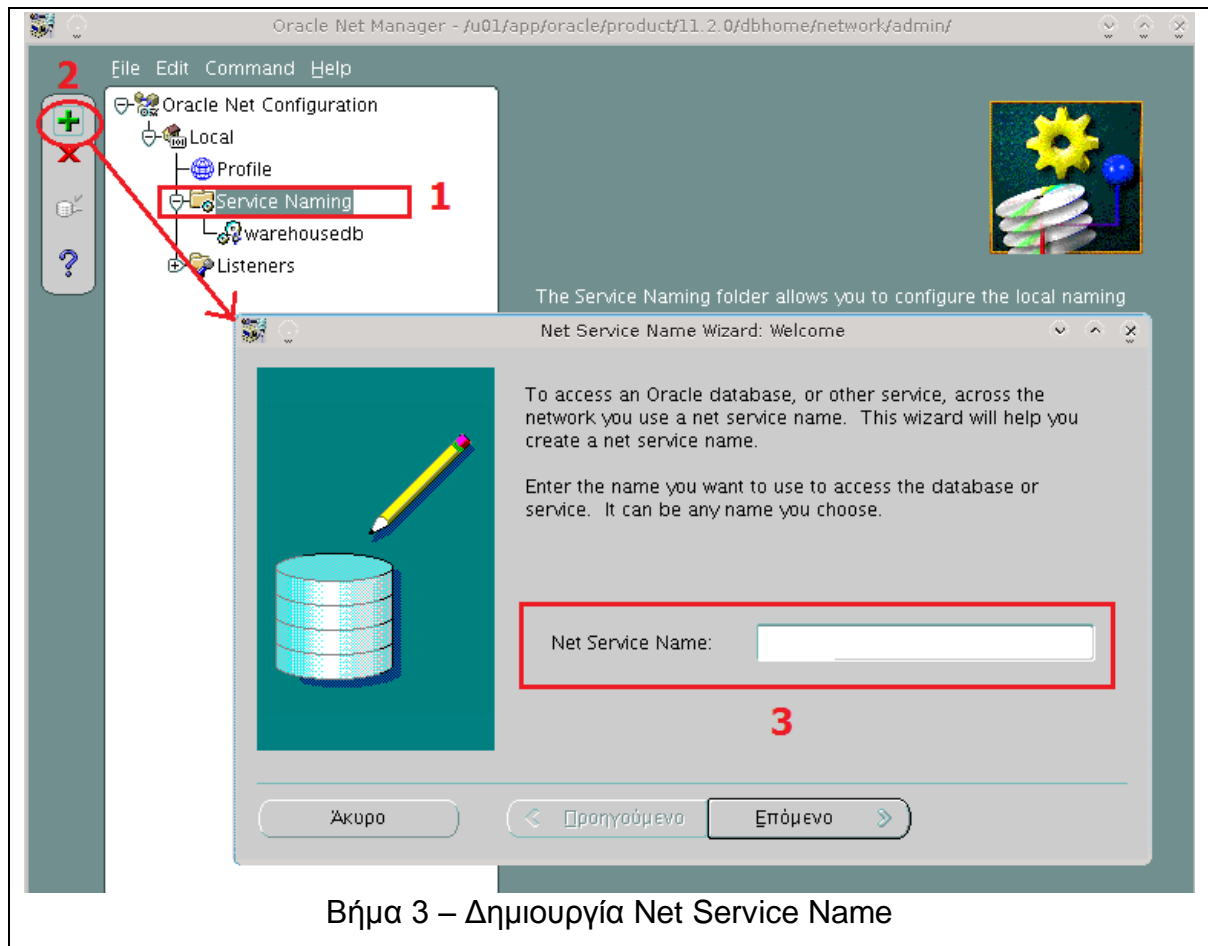
Επιλέγουμε το όνομα που θέλουμε να έχει ο Listener στο πεδίο «Listener Name» αλλά στις περισσότερες περιπτώσεις, το προεπιλεγμένο όνομα LISTENER μας βολεύει. Πατάμε OK για να ολοκληρωθεί η δημιουργία του Listener.

Ο Listener δημιουργήθηκε αλλά πρέπει να του πούμε «που θα ακούει» δηλαδή σε ποια θύρα και σε ποιον host. Παρακάτω βλέπουμε πως γίνεται:

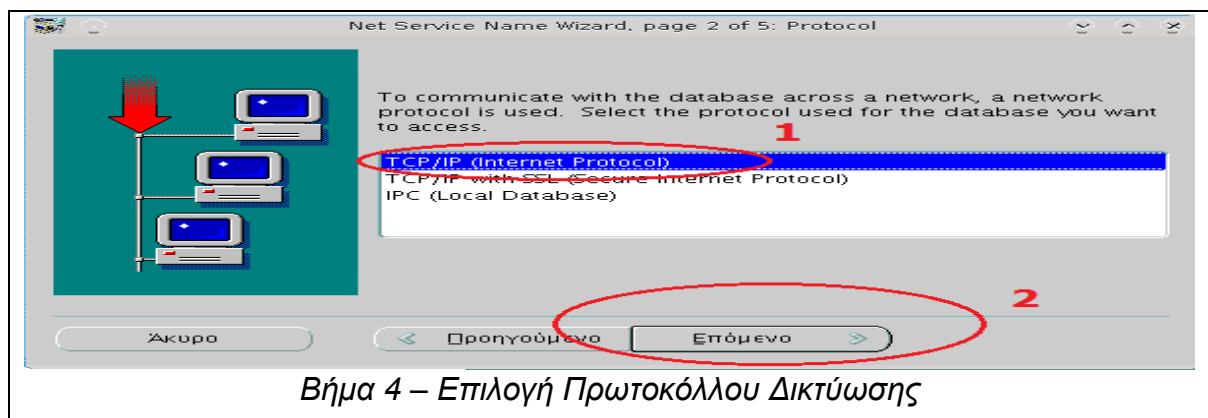


Αρχικά πατάμε στο πλήκτρο «Add Address» (1) και θα εμφανιστεί το παράθυρο «Address 1» που φαίνεται στο βελάκι. Εισάγουμε ως πρωτόκολλο σύνδεσης την σουίτα TCP/IP που είναι και η πιο συνηθισμένη (2). Τελευταία βήματα είναι η εισαγωγή του ονόματος ή της Localhost IP του Server μας, καθώς και της θύρας που θα ακούει ο Listener. Η προεπιλεγμένη θύρα είναι η 1521 οπότε αφήστε αυτήν (αν υπάρχει λόγος να βάλετε κάποια άλλη μπορείτε να το κάνετε).

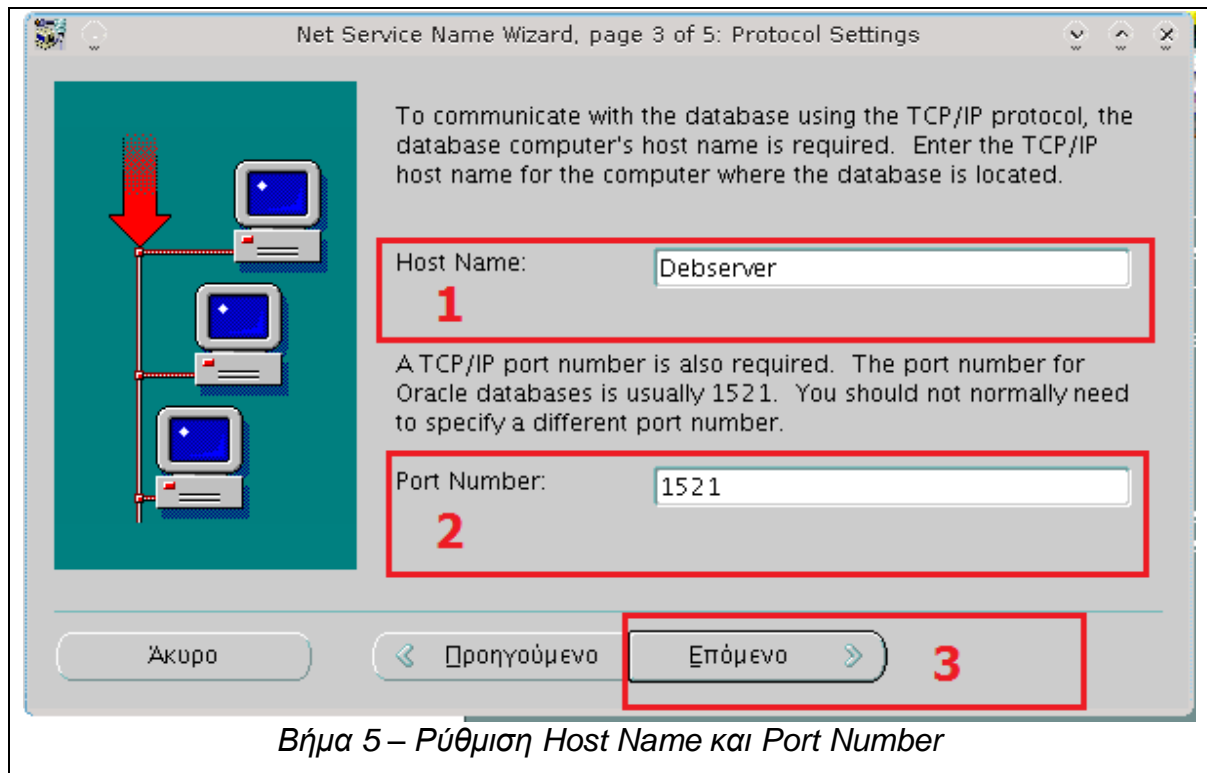
Τελευταία ενέργεια που πρέπει να γίνει, είναι η δημιουργία Service Name όπου θα δώσουμε το όνομα της βάσης στο οποίο θα ανταποκρίνεται ο Listener. Μέσω αυτού του Service Name, ο Listener «καταλαβαίνει» σε ποια βάση δεδομένων πρέπει προωθήσει τις αιτήσεις σύνδεσης. Ακολουθούμε τα βήματα που φαίνονται στην παρακάτω εικόνα:



Επιλέγουμε την κατηγορία «Service Naming» και επιλέγουμε τον πράσινο σταυρό (2) για να δημιουργήσουμε Net Service. Στο παράθυρο που μας εμφανίζει, δίνουμε ένα όνομα για την Net Service στο πεδίο «Net Service Name» (3). **Καλή πρακτική** είναι να δώσουμε το **SID** που σκοπεύουμε να χρησιμοποιήσουμε για την βάση που θα δημιουργήσουμε αργότερα, πχ το «test» για λόγους ευκολίας. Πατάμε «επόμενο» για να πάμε παρακάτω. Στην επόμενη οθόνη ρυθμίζουμε το πρωτόκολλο δικτύου που θα χρησιμοποιείται στην σύνδεση των clients, όπως βλέπουμε παρακάτω:

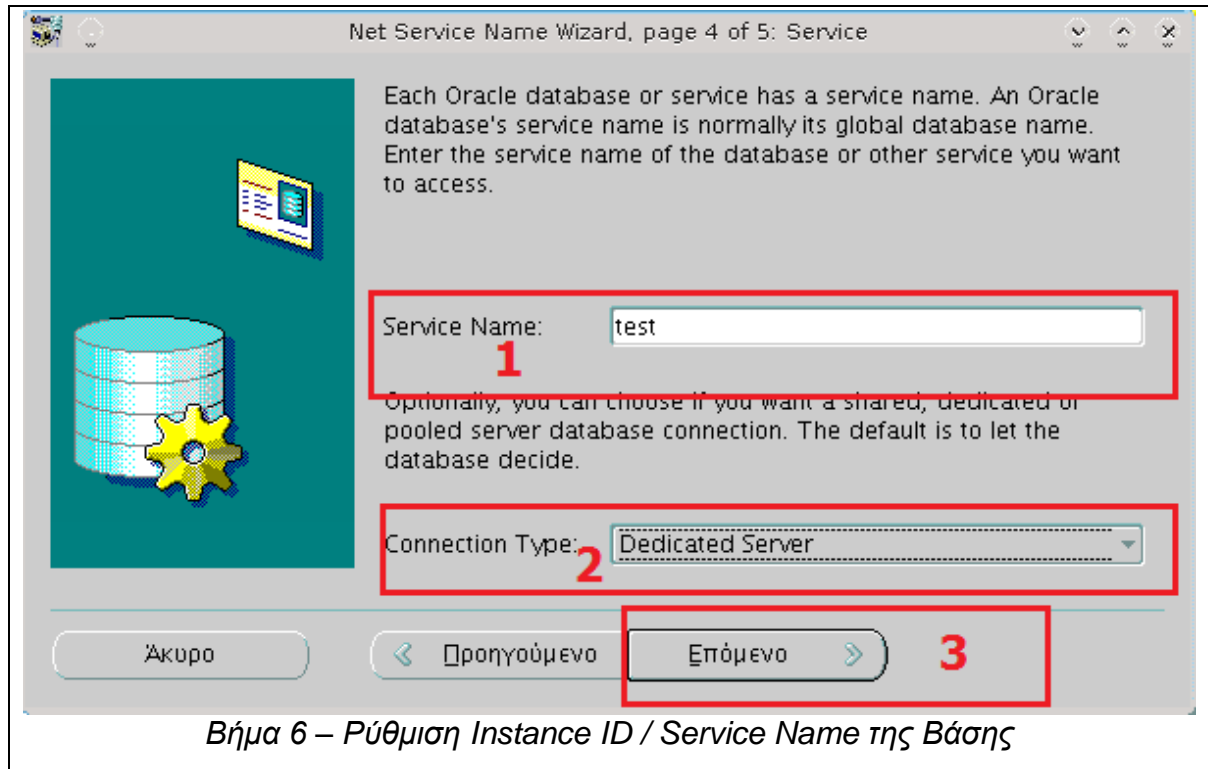


Επιλέγουμε τη σουίτα πρωτοκόλλων TCP/IP (1) και πατάμε «επόμενο» (2). Στην επόμενη οθόνη καλούμαστε να ρυθμίσουμε τον Server Host (τον server μας) και την θύρα που ακούει ο Listener, όπως παρακάτω:



Πρώτα δίνουμε το Host Name ή την IP Διεύθυνση του Server όπου βρίσκεται η βάση δεδομένων. Εφόσον η βάση δεδομένων (θα) βρίσκεται στον υπολογιστή μας, δίνουμε είτε το όνομα του υπολογιστή μας ή την Localhost IP Διεύθυνση (127.0.0.1) (1). Στην συνέχεια, δίνουμε και την Port που ακούει ο Listener για συνδέσεις (2). Στην συνέχεια πατάμε «Επόμενο» (3).

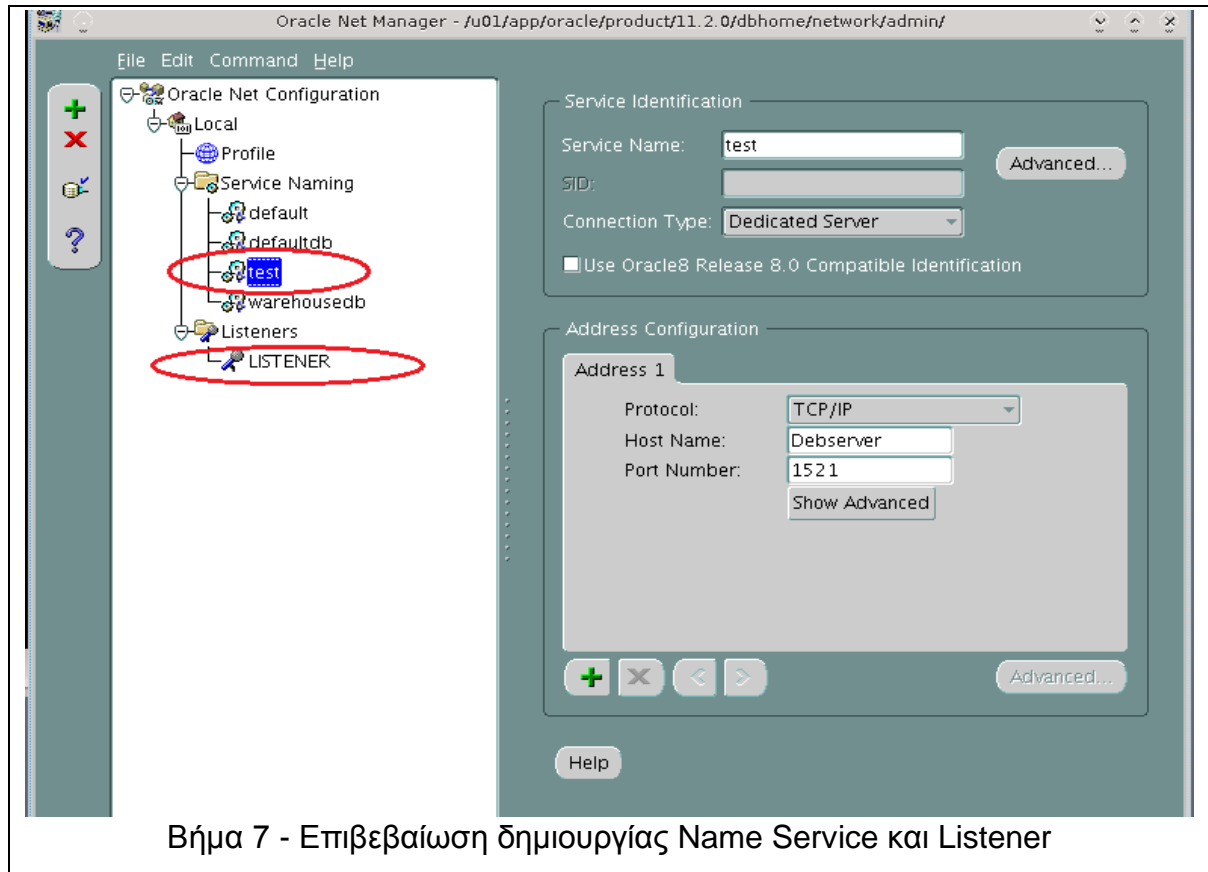
Στην συνέχεια, απαιτείται να ρυθμίσουμε το Service Name της βάσης, όπου θα κάνει Resolve ο Listener σε μια αίτηση σύνδεσης με την βάση. Ακολουθούμε τα βήματα που βλέπουμε παρακάτω:



Βήμα 6 – Ρύθμιση Instance ID / Service Name της Βάσης

Αρχικά ορίζουμε το όνομα υπηρεσίας (το οποίο είπαμε να είναι ίδιο με το SID) στο πεδίο «Service Name» (1) και στην συνέχεια τον τύπο σύνδεσης σε «Dedicated Server» (2). Πατάμε «Επόμενο» (3) για να ολοκληρώσουμε.

Στο επόμενο παράθυρο πατάμε «Τέλος» και ολοκληρώνουμε την δημιουργία του Listener και του Name Service. Τώρα δεν μένει να επαληθεύσουμε την δημιουργία των δύο αυτών βασικών στοιχείων και να αποθηκεύσουμε τις αλλαγές.



Αφού βεβαιωθούμε ότι ο Listener και το Service Name έχουν δημιουργηθεί, επεκτείνοντας τις αντίστοιχες κατηγορίες, πρέπει να αποθηκεύσουμε τις αλλαγές μας. Μεταβαίνουμε **File** → **Save Network Configuration** (πατάμε «Ναι» αν ζητήσει επιβεβαίωση). Τώρα μπορείτε να κλείσετε το Net Manager.

4.2.1 - Listener και Service ID (SID)

Ο Listener ακούει συνδέσεις και τις προωθεί στις ζητούμενες βάσεις δεδομένων. Ως εκ τούτου, το SID ή το Service Name θα είναι ο επιλογέας της βάσης δεδομένων από την πλευρά του Client. Το καίριο αρχείο όπου ο Listener «γνωρίζει» για το ποιες βάσεις δεδομένων υπάρχουν στο σύστημα, είναι το αρχείο «tnsnames.ora»⁸ που βρίσκεται στον κατάλογο

⁸ Oracle Documentation → Configuring and Administering the Listener → Listener Configuration Overview → “... the listener forwards client requests, can be configured in the listener.ora” – Site (http://docs.oracle.com/cd/B10501_01/network.920/a96580/listener.htm)

«ORACLE_HOME/network/admin/». Την μορφή αυτού του αρχείου (για την βάση TEST) την βλέπουμε παρακάτω:

#Δομή του TNSNAMES.ORA

```
TEST = #Βάση TEST
```

```
(DESCRIPTION =
```

```
(ADDRESS_LIST =
```

```
(ADDRESS = (PROTOCOL = TCP)(HOST = Debserver)(PORT = 1521))
```

#Πρωτόκολλο, Server και θύρα Listener

```
)
```

```
(CONNECT_DATA =
```

```
(SERVER = DEDICATED) #Είδος Server = Dedicated
```

```
(SERVICE_NAME = test) # όνομα της βάσης
```

```
)
```

```
)
```

#Άλλες καταχωρήσεις για άλλες βάσεις δεδομένων

Παραπάνω βλέπουμε την δομή του αρχείου tnsnames.ora, το οποίο **δεν χρειάζεται** να το γράψετε με το χέρι, δημιουργείτε αυτόματα. Έτσι κάθε φορά που ξεκινάει ο Listener, από αυτό το αρχείο γνωρίζει σε ποια SID των βάσεων που υπάρχουν στο σύστημα, πρέπει να προωθήσει τους Clients.

4.2.2 - Εκκίνηση του Listener

Για να μπορέσουμε να δημιουργήσουμε την βάση δεδομένων μας, θα πρέπει ο Listener να είναι ενεργός. Αφού βεβαιωθούμε ότι η μεταβλητή

ORACLE_SID έχει τιμή (test) τότε μπορούμε να εκκινήσουμε τον Listener με ένα έτοιμο script που μας παρέχει η Oracle, το **dbstart** που βρίσκεται στον κατάλογο «ORACLE_HOME/bin» ή άμεσα μέσω του εργαλείου **Listener Control** (lsnrctl) Το τρέχουμε λοιπόν (ως χρήστες oracle) με την εντολή:

```
oracle@debserver$ cd $ORACLE_HOME/bin
oracle@debserver$ ./dbstart $ORACLE_HOME
# ή εναλλακτικά απευθείας με το lsnrctl
oracle@debserver$ ./lsnrctl start LISTENER
```

Ο Listener θα ξεκινήσει σαν διεργασία, και θα περιμένει συνδέσεις για να προωθήσει στην βάση δεδομένων TEST. Ωστόσο την βάση ακόμα δεν την δημιουργήσαμε οπότε ακόμη ο Listener δεν μπορεί να εξυπηρετήσει Clients. Θα εξυπηρετήσει εμάς όμως τώρα που θα δημιουργήσουμε την βάση δεδομένων.

4.2.3 – Δοκιμή του Listener

Πως μπορούμε όμως να ξέρουμε ότι ο Listener είναι up and running; Δεν πρέπει με κάποιο τρόπο να τον δοκιμάσουμε; Σε αυτήν την περίπτωση έχουμε το εργαλείο TNSPING το οποίο κάνει Ping τοπικά στον Server, όπου αν ο Listener είναι ενεργός, τα Pings θα πετύχουν. Ας δούμε λοιπόν τι γίνεται:

Πρώτα βλέπουμε την κατάσταση του Listener, γράφοντας:

```
oracle@debserver$ cd $ORACLE_HOME/bin
oracle@debserver$ ./tnsping localhost
<...>
<...>
OK (0 sec). # ← Καλό σημάδι!
```

Βλέπουμε λοιπόν πως ο Listener τρέχει. Θετικό σημάδι για την μετέπειτα εξέλιξη του έργου μας!

4.2.4 - Τερματισμός του Listener

Αν θέλουμε να τερματίσουμε τον Listener ως διεργασία στον Server, μπορούμε να το κάνουμε άμεσα, με ακόμα ένα script που μας παρέχει έτοιμο η Oracle, το **dbshut**. Οι εντολές είναι οι εξής:

```
oracle@debserver$ cd $ORACLE_HOME/bin  
oracle@debserver$ ./dbshut $ORACLE_HOME
```

Ο Listener τερματίζει και πλέον ο Server δεν ανταποκρίνεται σε εισερχόμενες συνδέσεις. Ωστόσο θα πρέπει να είναι πάντα ενεργός όταν δημιουργείτε βάσεις δεδομένων ή εργάζεστε έστω και τοπικά σε κάποια βάση.

Καλή Πρακτική

Φροντίζουμε ο Listener να είναι πάντα ενεργός ώστε όλοι οι χρήστες να μπορούν να συνδεθούν στην βάση. Όποτε συντρέχει λόγος όπου ο Listener πρέπει να είναι ανενεργός, όπως σε εργασίες συντήρησης του Server σας, φροντίστε να ειδοποιείτε τους χρήστες, αν το γεγονός αυτό πρέπει να λάβει χώρα σε χρονική περίοδο που προσδοκείται να συνδεθούν χρήστες.

4.3 - Δημιουργία Βάσης Δεδομένων

Τώρα βρισκόμαστε στο σημαντικότερο βήμα, στην δημιουργία μιας βάσης δεδομένων στον Server. Η βάση αυτή που θα δημιουργήσουμε, θα είναι η βάση στην οποία θα συνδέονται όλοι οι χρήστες και θα δουλεύουν σε αυτήν, οπότε θα πρέπει να είμαστε προσεκτικοί στην δημιουργία και παραμετροποίηση της.

Προσοχή

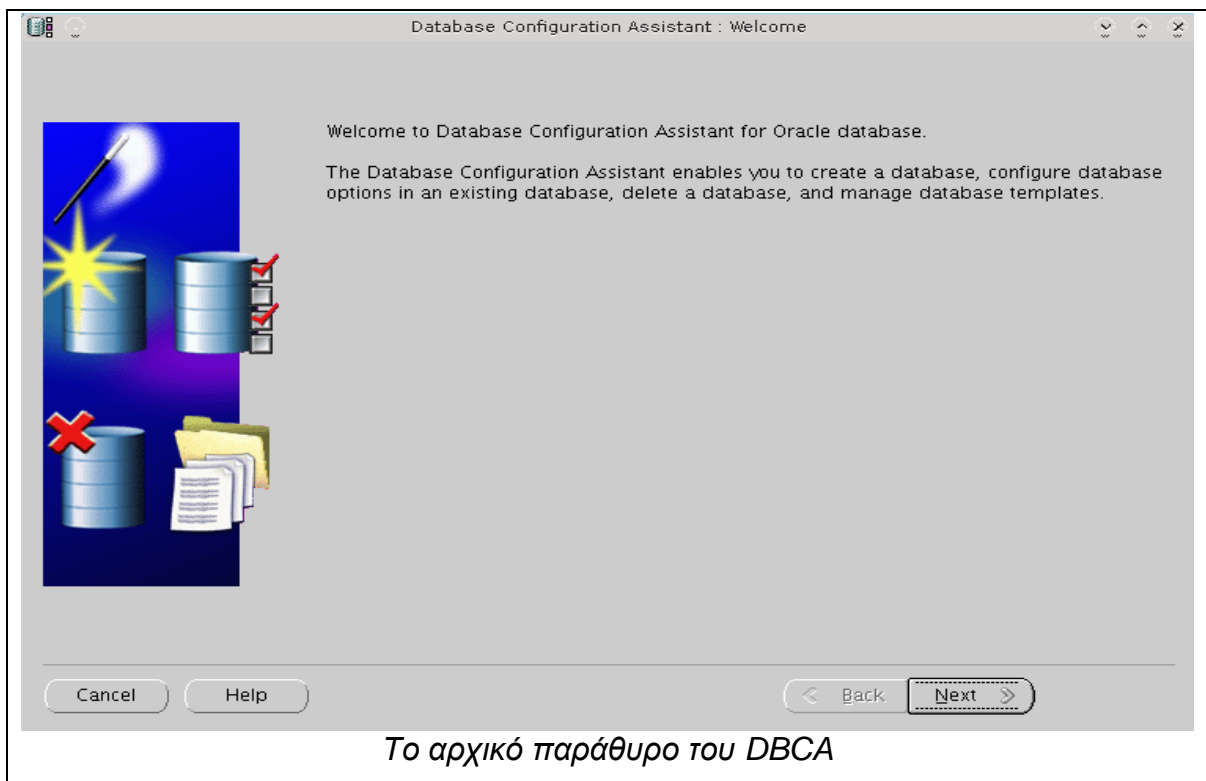
Η δημιουργία βάσης δεδομένων απαιτεί παραθυρικό περιβάλλον για να περατωθεί οπότε πραγματοποιήστε το είτε απευθείας στον Server, είτε με απομακρυσμένη σύνδεση μέσω RDP.

Η Oracle παρέχει έτοιμο βοηθητικό εργαλείο για την δημιουργία και τροποποίησης μιας βάσης δεδομένων, το **Database Configuration Assistant** (DBCA). Βρίσκεται στον κατάλογο ORACLE_HOME/bin οπότε μεταβαίνουμε εκεί και το εκτελούμε με τις παρακάτω εντολές (ως χρήστες oracle):

```
oracle@debserver$ cd $ORACLE_HOME/bin
```

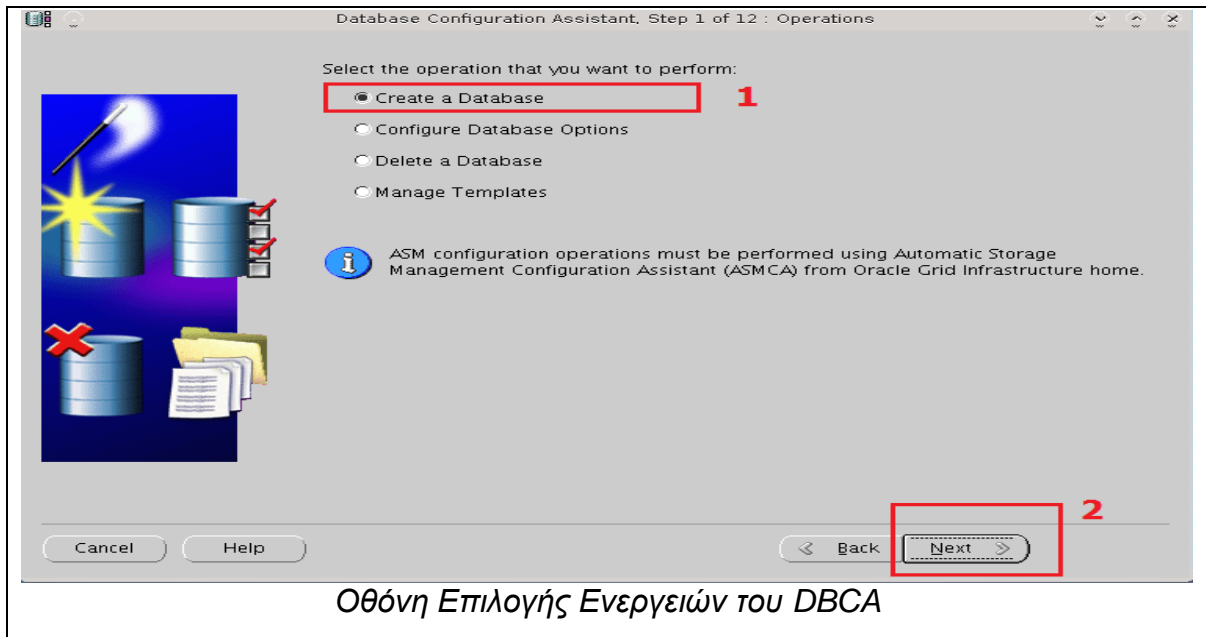
```
oracle@debserver$ ./dbca
```

Το βοηθητικό πρόγραμμα DBCA θα εκτελεστεί και θα εμφανιστεί μπροστά μας το αρχικό του παράθυρο, όπως βλέπουμε παρακάτω:

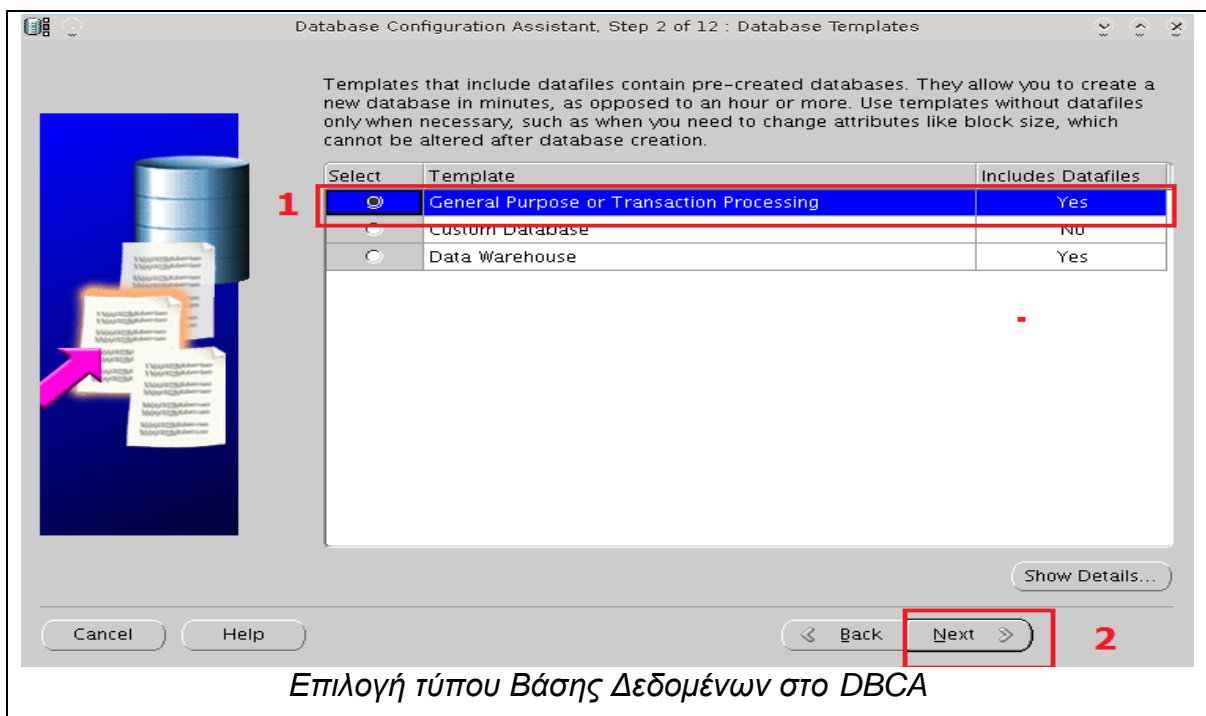


Εμείς πατάμε «επόμενο», όπου μεταβαίνουμε στην οθόνη επιλογής ενεργειών. Σε αυτήν την οθόνη λοιπόν επιλέγουμε αν θέλουμε να

δημιουργήσουμε, να τροποποιήσουμε ή να διαγράψουμε μια βάση δεδομένων. Το βλέπουμε άλλωστε και παρακάτω:



Εμείς, εφόσον θέλουμε να δημιουργήσουμε μια βάση δεδομένων, επιλέγουμε «Create a Database» (1) και στην συνέχεια πατάμε «Next» (2). Στην επόμενη οθόνη, επιλέγουμε τον τύπο της βάσης δεδομένων που θέλουμε να δημιουργήσουμε:



Εδώ καλούμαστε να «πούμε» στο DBCA τι είδους βάση θέλουμε να δημιουργήσουμε. Και εδώ το δίλλημα, τι βάση να δημιουργήσουμε άραγε;

Στην πλειονότητα των περιπτώσεων, όπου θέλουμε μια κανονική βάση δεδομένων για να δουλέψουμε, μπορούμε να επιλέξουμε «General Purpose or Transaction Processing» (1) όπου μας δημιουργεί μια βάση δεδομένων **γενικού τύπου** που στις περισσότερες εφαρμογές που χρειαζόμαστε, πληροί τις προϋποθέσεις. Μπορούμε ωστόσο να δημιουργήσουμε μια βάση δεδομένων τύπου «Data Warehouse» η οποία θα χρειαστεί αν η βάση μας πρόκειται να δεχτεί τεράστιο αριθμό πολύπλοκων δεδομένων. Η επιλογή φυσικά είναι καθαρά δική σας, εμείς ωστόσο για τους λόγους επίδειξης θα δημιουργήσουμε τώρα μια γενικού τύπου. Πατάμε «Next» (2) για να μεταβούμε παρακάτω.

Στην επόμενη οθόνη πρέπει να «βαφτίσουμε» την βάση δεδομένων μας, και να τις εισάγουμε το μοναδικό της **Service / Instance ID (SID)** το οποίο την προσδιορίζει μοναδικά. Η διαδικασία φαίνεται παρακάτω:

Database Configuration Assistant, Step 3 of 12 : Database Identification

An Oracle database is uniquely identified by a Global Database Name, typically of the form "name.domain".

1 Global Database Name:

A database is referenced by at least one Oracle instance which is uniquely identified from any other instance on this computer by an Oracle System Identifier (SID).

2 SID:

Cancel Help < Back Next > **3**

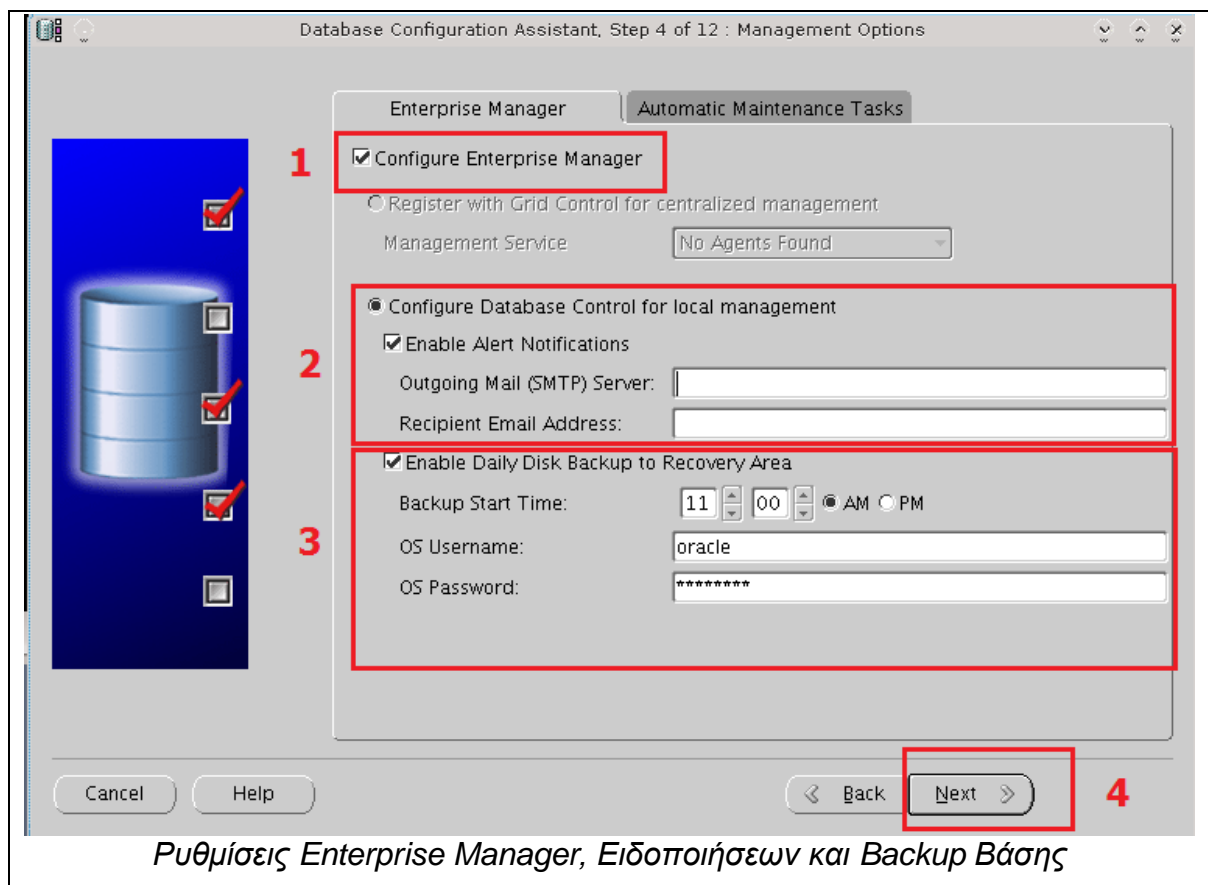
Επιλογή ονόματος βάσης και Service ID (SID)

Η σωστή επιλογή ονόματος της βάσης και του Service ID είναι πολύ **κρίσιμα** οπότε φροντίζουμε να τα θυμόμαστε. Αρχικά δίνουμε ένα Global όνομα στην βάση δεδομένων στο πεδίο «Global Database Name» (1) το οποίο για λόγους επίδειξης θα είναι το «test». Το ίδιο όνομα αναγράφεται ως Instance ID στο «SID» (2) το οποίο φυσικά μπορείτε να το αλλάξετε σύμφωνα με την μεταβλητή ORACLE_SID, ωστόσο εμείς θα χρησιμοποιήσουμε πάλι το «test» για λόγους ευκολίας. Ολοκληρώνουμε το βήμα με «Next» (3).

ΠΡΟΣΟΧΗ

Το SID της βάσης την προσδιορίζει μοναδικά, οπότε φροντίστε κάθε βάση δεδομένων να έχει μοναδικό SID και NAME.

Στην επόμενη οθόνη, ρυθμίζουμε τις διαδικασίες συντήρησης, ειδοποίησης και του Oracle Enterprise Manager, όπως βλέπουμε παρακάτω:



Αρχικά επιλέγουμε το checkbox «Configure Enterprise Manager» (1) το οποίο θα μας επιτρέψει να χρησιμοποιήσουμε Web Based εργαλεία ελέγχου και

εργασίας με την βάση (όπως το Oracle APEX που θα δούμε αργότερα σε άλλο κεφάλαιο και το Oracle Enterprise Manager).

Τα βήματα 2 και 3 είναι **προαιρετικά**, οπότε αν θέλετε μπορείτε να τα παραλείψετε. Αν θέλουμε να λαμβάνουμε ειδοποιήσεις στο Mail μας κάθε φορά που κάτι δεν πάει καλά με την βάση μας ή εξαντλείται ο χώρος της βάσης, τότε ενεργοποιούμε την επιλογή «Enable Alert Notifications» και στα αντίστοιχα δύο πεδία που ενεργοποιούνται εισάγουμε την IP Διεύθυνση ή το Domain Name του SMTP Server στον οποίο έχουμε λογαριασμό, και στην συνέχεια την διεύθυνση email μας (2). Το ίδιο ισχύει για την επιλογή «Enable Daily Backup to Recovery Area», όπου λέμε στο σύστημα να κρατάει καθημερινά backup της βάσης δεδομένων μας στον κατάλογο «{ORACLE_BASE}/flash_recovery_area» (τα **άγκιστρα** για την μεταβλητή περιβάλλοντος ORACLE_BASE είναι απαραίτητα στο πεδίο). Ορίζουμε λοιπόν την ώρα που θέλουμε να λαμβάνουν χώρα τα backup καθώς και τα credentials του χρήστη του λειτουργικού συστήματος (oracle) μέσω του οποίου θα γίνονται τα Backup (3). Ολοκληρώνουμε το βήμα, με «Next» (4).

Η επόμενη οθόνη αφορά τις ρυθμίσεις ασφαλείας, των κωδικών πρόσβασης δηλαδή των διαφόρων διαχειριστών που υπάρχουν στο σύστημα:

Database Configuration Assistant, Step 5 of 12 : Database Credentials

For security reasons, you must specify passwords for the following user accounts in the new database.

Use Different Administrative Passwords

User Name	Password	Confirm Password
SYS		
SYSTEM		
DBSNMP		
SYSMAN		

Use the Same Administrative Password for All Accounts

Password:

Confirm Password:

1

Cancel Help < Back Next > 2

Οθόνη ρύθμισης κωδικών πρόσβασης Διαχειριστών

Πριν περιγράψουμε τα βήματα, πρώτα θεωρούμε σκόπιμο να περιγράψουμε τους διάφορους διαχειριστές⁹ που υπάρχουν στο σύστημα

- 1) **SYS** = Ο κύριος διαχειριστής του συστήματος. Μπορεί να πραγματοποιήσει αλλαγές στην βάση όπως δημιουργία/αλλαγή/διαγραφή χρηστών, αντικειμένων ακόμα και αλλαγές στην δομή της ίδιας της βάσης. Στο Schema του (SYS) αποθηκεύονται οι πίνακες και οι όψεις του Data Dictionary¹⁰ (metadata πληροφορίες).
- 2) **SYSTEM** = Ίδιος με τον SYS σε ότι αφορά τα δικαιώματα, ωστόσο ο SYS είπαμε πως είναι ο ιδιοκτήτης της βάσης
- 3) **DBSNMP** = Χρήστης με δικαιώματα Admin (SYSDBA) για το Oracle Enterprise Manager. Κύριος ρόλος ύπαρξής του, είναι το monitoring της βάσης δεδομένων.
- 4) **SYSMAN** = Super Administrator του Oracle Enterprise Manager. Έχει τα ίδια δικαιώματα με τους χρήστες SYS και SYSTEM.

Όλοι οι διαχειριστές του συστήματος, έχουν τον ρόλο **DBA** (Data-Base Administrator) και μπορούν να πραγματοποιήσουν σχεδόν τα πάντα στην βάση δεδομένων. Ο κύριος όμως διαχειριστής είναι ο χρήστης SYS, ο οποίος είναι και ο μοναδικός που έχει τον ρόλο **SYSDBA**¹¹ (System Data-Base Administrator), ρόλος που του δίνει τον απόλυτο έλεγχο στην βάση (αντίστοιχος του χρήστη Root στα Unix-like συστήματα) οπότε με αυτόν θα ασχοληθούμε κυρίως στο παρών σύγγραμμα.

Μια καλή πρακτική είναι να δημιουργούμε διαφορετικούς κωδικούς πρόσβασης για κάθε έναν διαχειριστή του συστήματος. Επίσης καλή πρακτική

⁹ Oracle Documentation → Overview of Users and Security → Administrative User Accounts Provided by Oracle Database – Site

(http://docs.oracle.com/cd/B16276_01/doc/server.102/b14196/users_secure001.htm)

¹⁰ Oracle documentation → Managing Users and Security → The SYS and SYSTEM Users → “All base tables and views for the database data dictionary are stored in the SYS schema” – Site

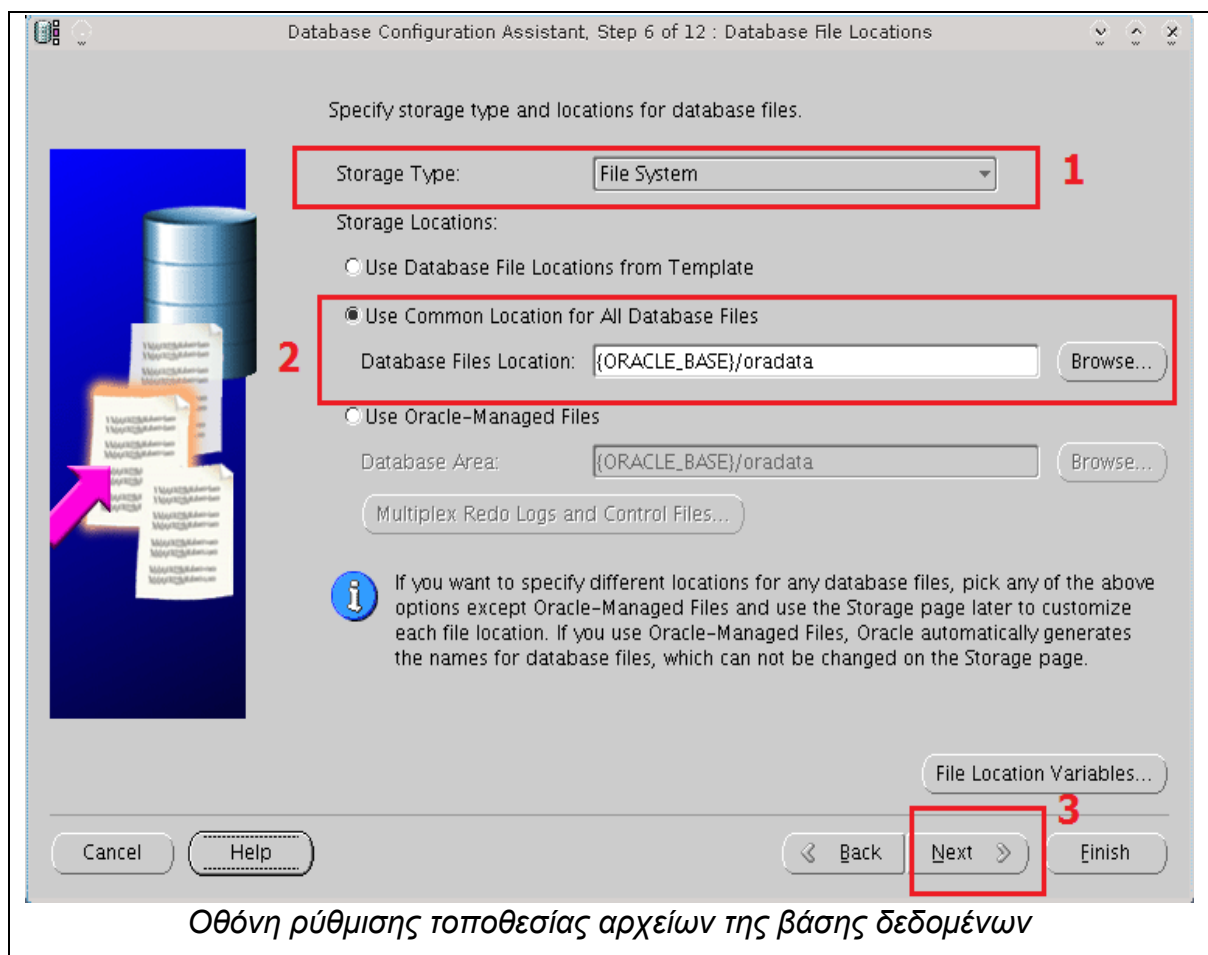
(http://docs.oracle.com/cd/E17781_01/server.112/e18804/users_secure.htm#autold4)

¹¹ Oracle Documentation → Managing Users and Security → The SYSDBA System Privilege → “SYSDBA is a system privilege that is assigned only to user SYS” –

Site(http://docs.oracle.com/cd/E17781_01/server.112/e18804/users_secure.htm#autold4)

είναι οι κωδικοί μας να είναι **μη-προβλέψιμοι** ώστε να ελαχιστοποιούμε τις πιθανότητες παραβίασης ασφαλείας από μη εξουσιοδοτημένους χρήστες. Ωστόσο για λόγους ευκολίας, επιλέγουμε «Use the same administrative password for all accounts» (1), ώστε να χρησιμοποιήσουμε τον ίδιο κωδικό πρόσβασης για όλους τους Administrators. Αφού εισάγουμε το password και στα δύο πεδία, πατάμε «Next» (2) για να προχωρήσουμε παρακάτω.

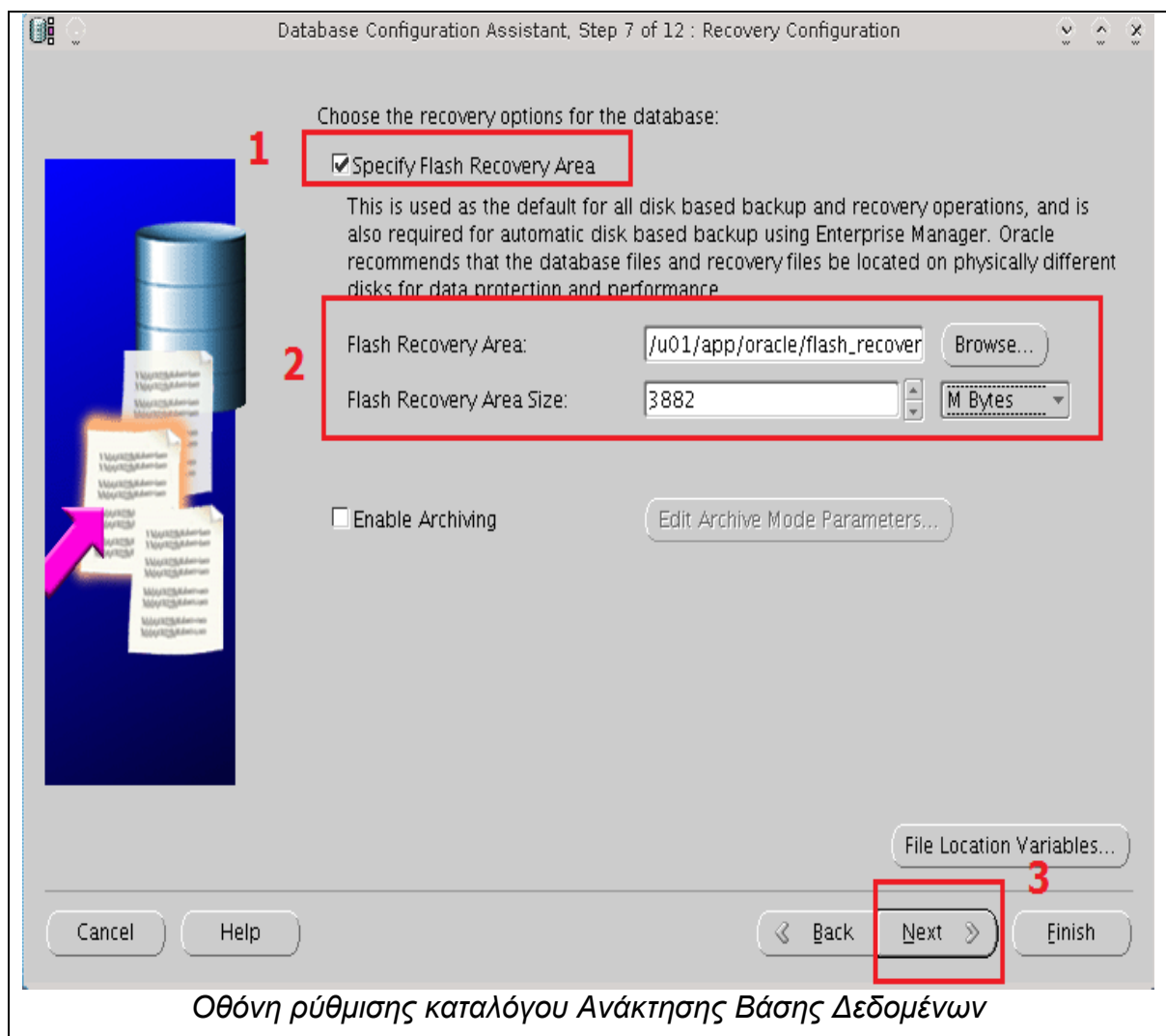
Στην επόμενη οθόνη, καλούμαστε να ρυθμίσουμε την τοποθεσία που θα αποθηκευτούν τα αρχεία της βάσης δεδομένων. Αποτελεί επίσης σημαντικό βήμα, καθώς η ακριβής τοποθεσία εναπόθεσης των αρχείων της βάσης, είναι κρίσιμη για διαδικασίες backup, restore κλπ. Παρακάτω βλέπουμε την οθόνη «Database File Locations»:



Αρχικά επιλέγουμε ως τύπο αποθήκευσης (Storage Type) το File System (1), το οποίο αφορά αποθήκευση της βάσης δεδομένων στον τοπικό δίσκο του

Server μας. Στην συνέχεια επιλέγουμε το radio «Use Common Location for All Database Files» και δίνουμε στο πεδίο «Database File Location» το path προς τον κατάλογο «{ORACLE_BASE}/oradata» (2). Θα πρέπει να σημειωθεί, ότι ο κατάλογος **Oradata** κρατάει τα αρχεία όλων των βάσεων δεδομένων που υπάρχουν στο σύστημα. Τέλος, ολοκληρώνουμε με «Next» (3).

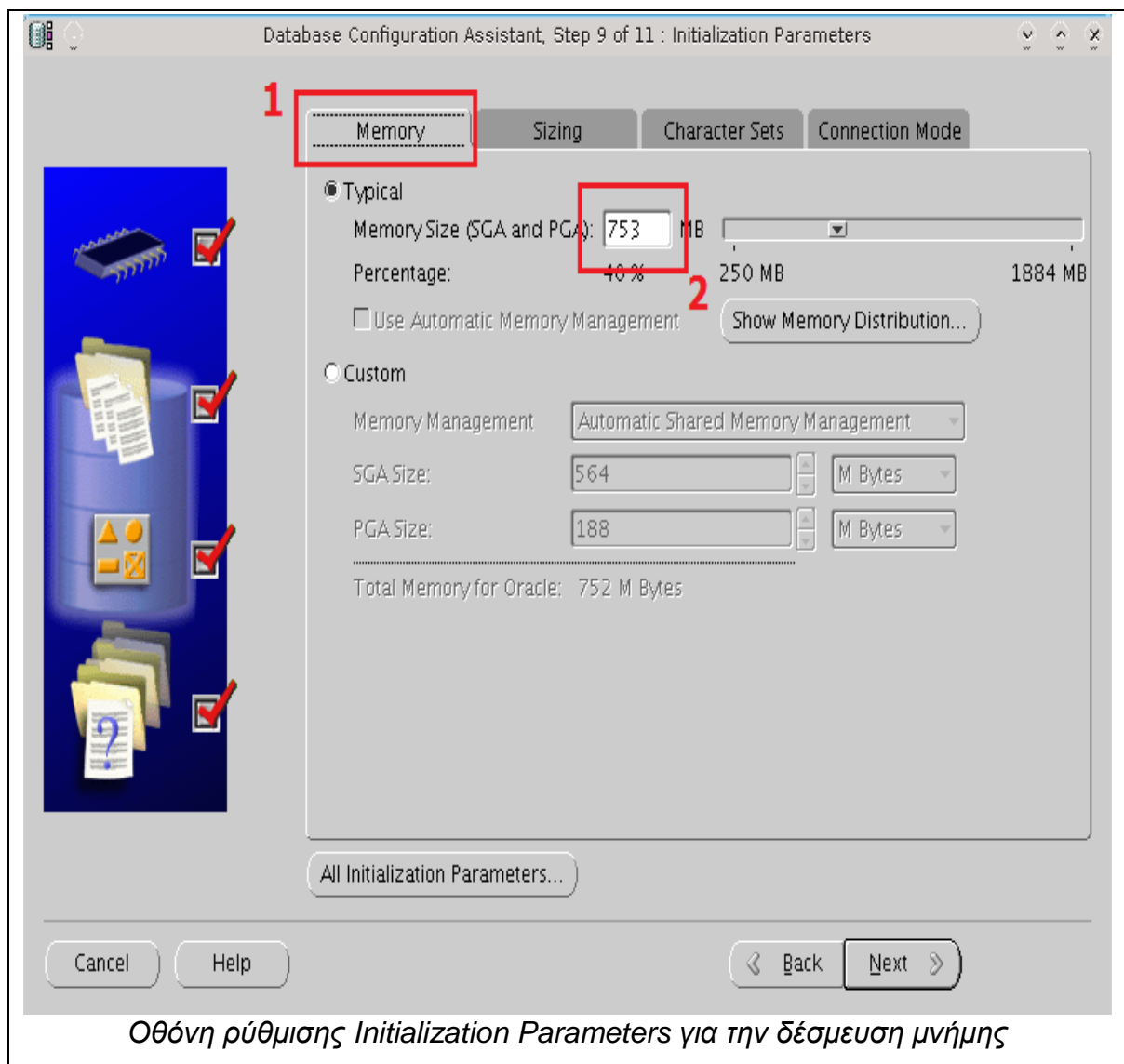
Στην επόμενη οθόνη, καλούμαστε να ρυθμίσουμε τον κατάλογο που θα βρίσκονται τα backups της βάσης μας, σε περίπτωση που χρειαστεί να κάνουμε Restore. Την βλέπουμε παρακάτω:



Το βήμα αυτό είναι **προαιρετικό**. Επιλέγουμε πρώτα «Specify Flash Recovery Area» (1) και στα πεδία που ενεργοποιούνται δίνουμε α) το path προς

τον κατάλογο «flash_recovery_area» που βρίσκεται στο «{ORACLE_BASE}/flash_recovery_area» και β) το μέγεθος σε MB της maximum χωρητικότητας του καταλόγου αυτού (2). Ολοκληρώνουμε με «Next» (3).

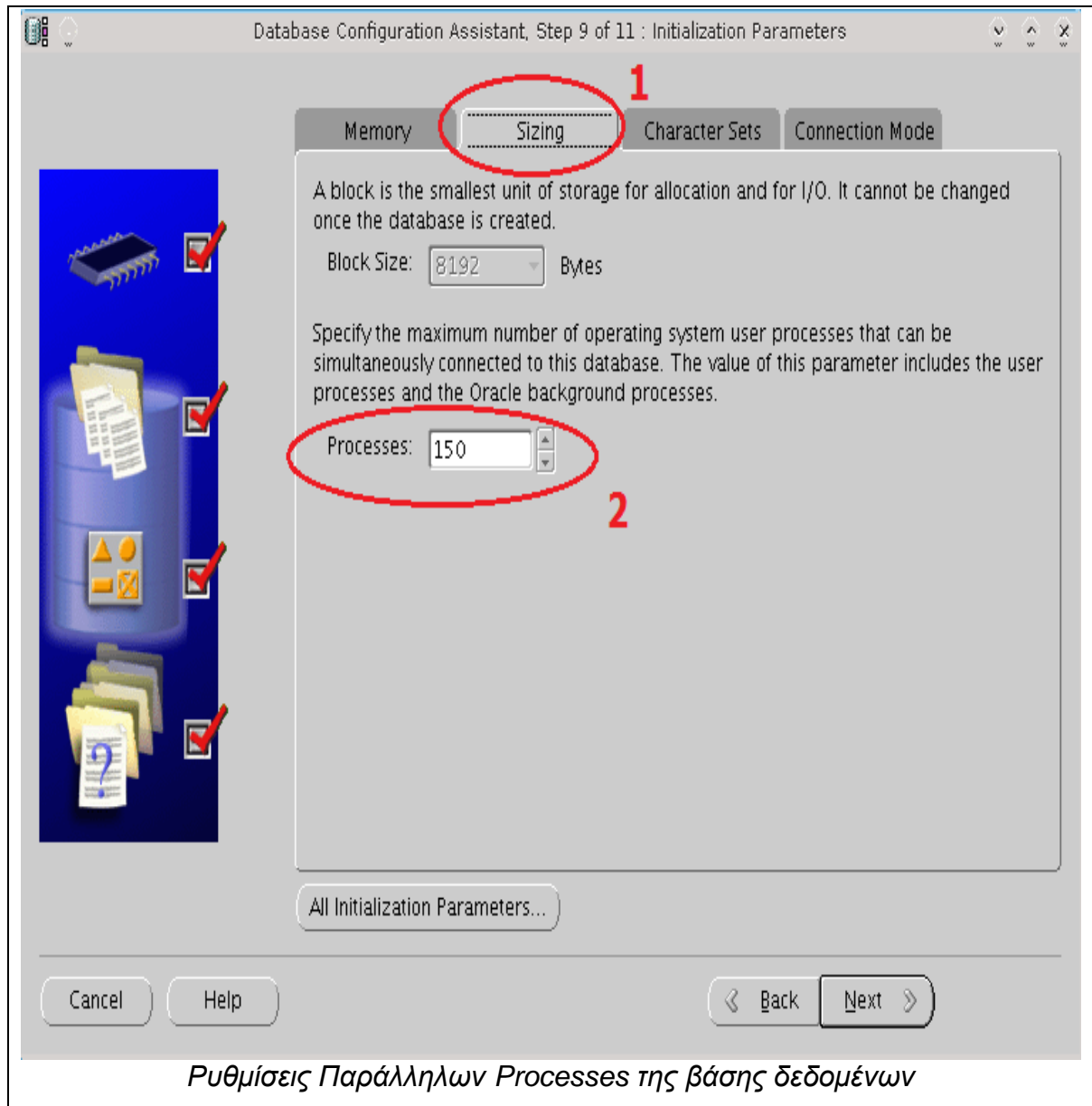
Στην επόμενη οθόνη πατάμε «Next» καθώς δεν περιέχει κάτι χρήσιμο (εκτός αν θέλετε να δημιουργηθούν παραδειγματικά schemas και πίνακες), μέχρι να φτάσουμε στην οθόνη «Initialization Parameters» όπου και υπάρχουν πολύ καίρια σημεία ρυθμίσεων, οπότε δώστε ιδιαίτερη προσοχή εδώ. Την αρχική οθόνη την βλέπουμε κάτωθι:



Η αρχική οθόνη αφορά την δέσμευση μνήμης (1). Από το πεδίο «Memory Size (SGA and PGA)» επιλέγουμε το μέγεθος σε MB που θέλουμε να δεσμεύονται

από τη μνήμη του συστήματος για το Instance της βάσης (SGA) και τα processes των client (PGA). **ΠΡΟΣΟΧΗ** ωστόσο, αν δώσετε μεγάλο μέγεθος δέσμευσης μνήμης μπορεί να έχει αντίκτυπο στην απόδοση του λειτουργικού συστήματος, οπότε καλύτερα να το αφήσετε στην προεπιλεγμένη τιμή που έχει ήδη (2).

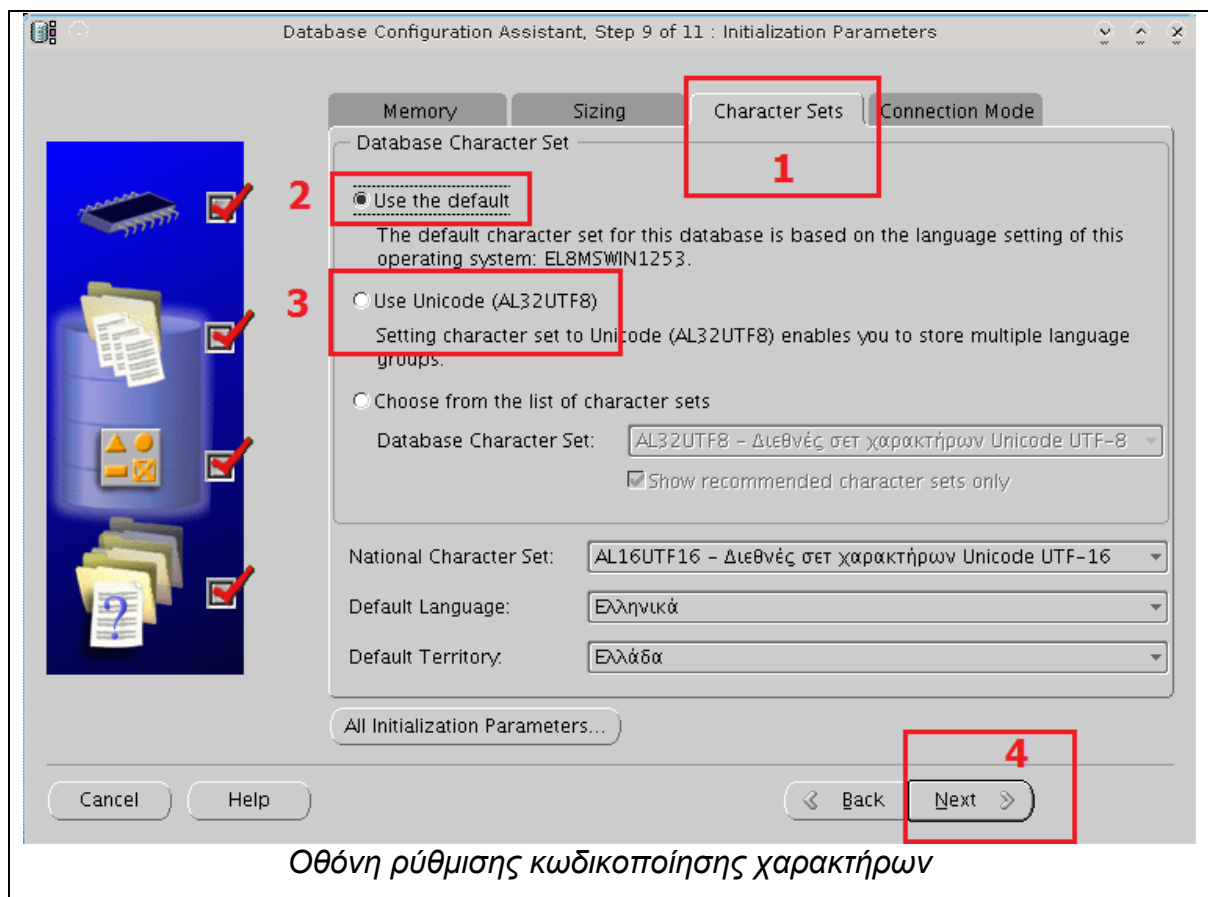
Μεταβαίνουμε στην καρτέλα «Sizing» για να ρυθμίσουμε το πλήθος των παράλληλων processes της βάσης δεδομένων, όπου την βλέπουμε παρακάτω:



Τα processes αυτά δεν είναι τίποτα άλλο, παρά **οι συνδέσεις των clients** στην βάση δεδομένων, καθώς κάθε σύνδεση ενός client, μεταφράζεται σε ένα πρόσθετο process για τον Server. Στην καρτέλα «Sizing» (1) λοιπόν, επιλέγουμε

από το πεδίο «Processes» (2) τον maximum αριθμό ταυτόχρονων συνδέσεων στον Server που προσδοκούμε. Καλή πρακτική είναι να εισάγουμε αριθμό που να ανταποκρίνεται σε όλους τους χρήστες, τους διαχειριστές και λίγο περισσότερο για μελλοντικούς χρήστες.

Μεταβαίνουμε στην καρτέλα «Character Sets» για να ρυθμίσουμε την κωδικοποίηση χαρακτήρων που θα έχει η βάση δεδομένων μας, την οποία βλέπουμε παρακάτω:



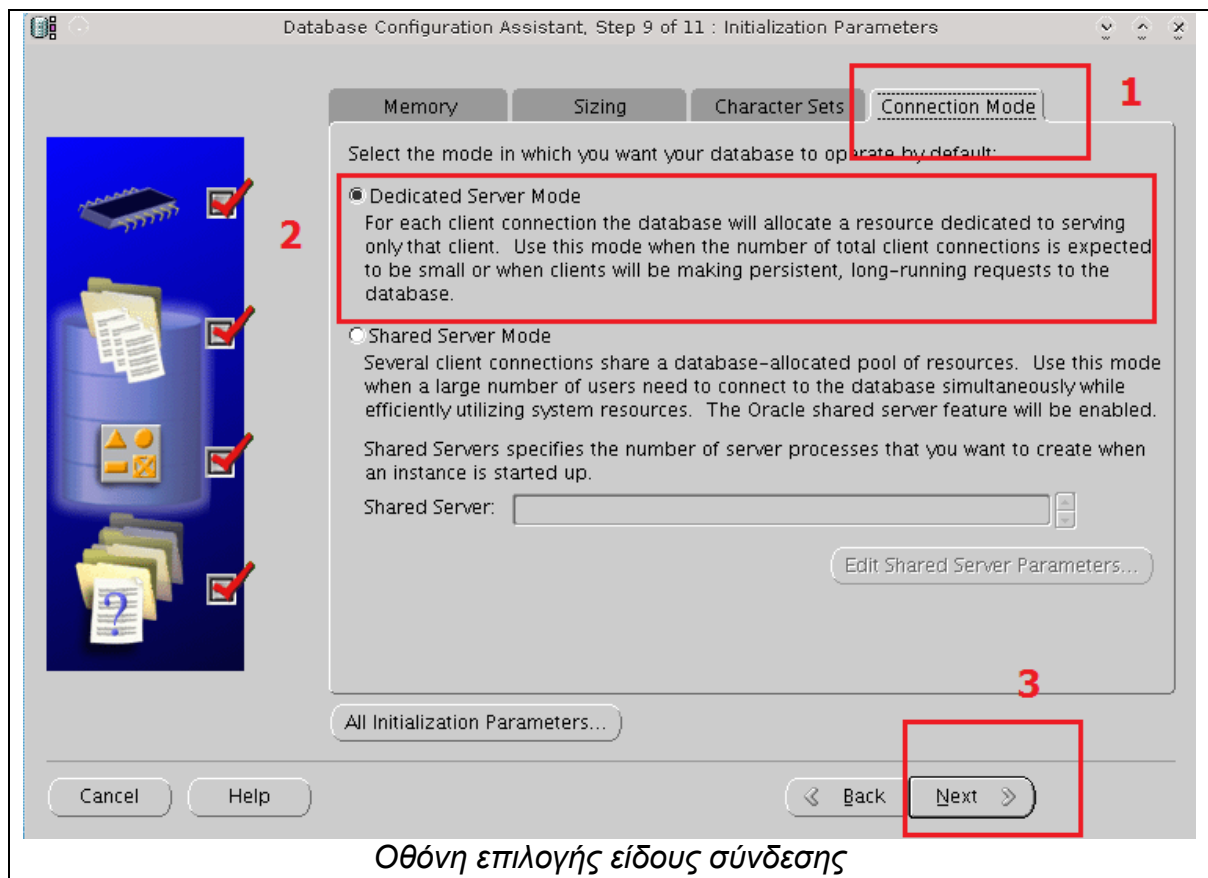
ΠΡΟΣΟΧΗ μεγάλη πρέπει να δώσουμε σε αυτήν την οθόνη (1) καθώς η επιλογή λανθασμένου Character Set θα μας κοστίσει σε μη-κατανοητό (κοινώς αλαμπουρνέζικο) output της βάσης δεδομένων. Έχουμε τις εξής επιλογές βάση των αναγκών μας

- 1) Επιλέγουμε «Use the Default» (2) το οποίο είναι αυτό που χρησιμοποιεί το ίδιο το λειτουργικό σύστημα. Αν εγκαταστήσατε τα Debian με την χρήση Ελληνικής γλώσσας, τότε το Character Set θα είναι το

«EL8MSWIN1253» το οποίο μπορούμε να το χρησιμοποιήσουμε για clients που θα χρησιμοποιούν Ελληνικά Windows (περισσότερο προσδοκώμενο). Η εν λόγω κωδικοποίηση παρέχει καλή υποστήριξη ελληνικών χαρακτήρων, οπότε αποτελεί **καλή επιλογή**.

- 2) Επιλέγουμε «Use Unicode» (3) αν προσδοκούμε clients από πολύγλωσσα λειτουργικά συστήματα. Εφόσον μιλάμε για Unicode, υποστηρίζει πολλαπλούς χαρακτήρες διαφόρων γλωσσών.

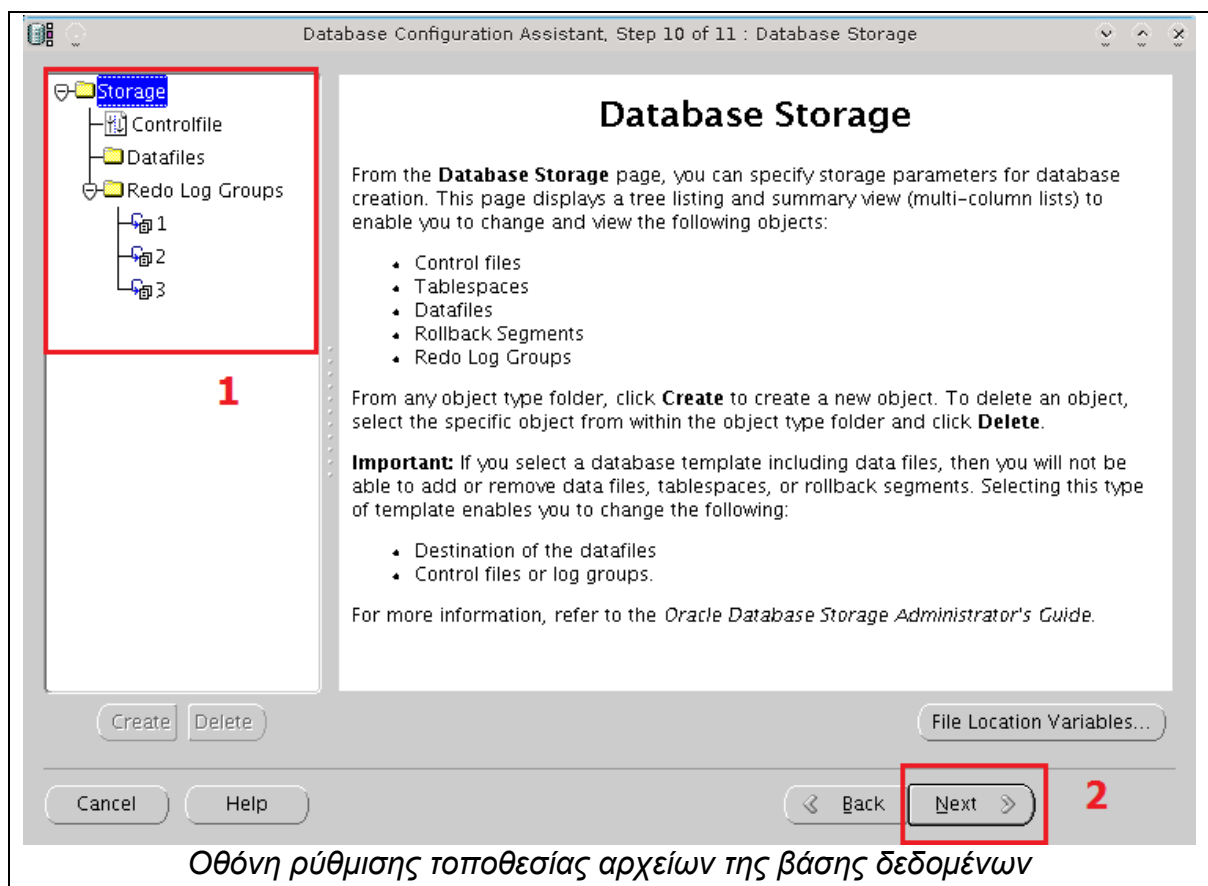
Αφού επιλέξουμε Character Set, μεταβαίνουμε στην τελευταία καρτέλα «Connection Mode» η οποία είναι η παρακάτω:



Στην καρτέλα «Connection Mode» (1) επιλέγουμε το είδος της σύνδεσης που παρέχει ο Server μας. Εφόσον είναι ο μοναδικός, και οι clients θέλουμε να έχουν από ένα instance για αποδοτική λειτουργία της βάσης και του Server,

επιλέγουμε την επιλογή «Dedicated Server Mode» (2). Η εν λόγω επιλογή είναι καλή επίσης αν ο αριθμός των προσδοκώμενων χρηστών δεν θα είναι πολύ μεγάλος. Αν όμως προσδοκούμε μεγάλο αριθμό χρηστών τότε επιλέγουμε την επιλογή «Shared Server Mode», αλλά δεν είναι κάτι τέτοιο το ζητούμενο. Η πρώτη επιλογή μας καλύπτει για τις ανάγκες που χρειαζόμαστε. Τέλος ολοκληρώνουμε το βήμα με «Next» (3).

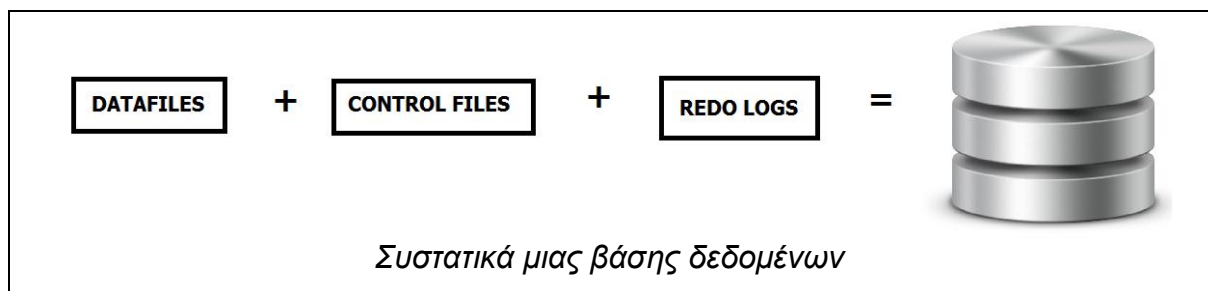
Στην επόμενη οθόνη ρυθμίζουμε επακριβώς την τοποθεσία αποθήκευσης του κάθε αρχείου της βάσης δεδομένων, την οποία και βλέπουμε παρακάτω:



Στην αριστερή περιοχή του παραθύρου, βλέπουμε την ιεραρχική προβολή κατηγοριών του τύπου αρχείων που απαρτίζουν την βάση δεδομένων (1). Τα αρχεία αυτά είναι:

- 1) **Control Files**¹² = Αρχεία που περιγράφουν την δομή της βάσης δεδομένων και περιέχουν πληροφορίες όπως την χρονοσφραγίδα (timestamp) δημιουργίας της, το όνομά της, ποια αρχεία (datafiles, redos) της ανήκουν κ.α.
- 2) **Datafiles** = Πρόκειται στην ουσία για ολόκληρη την βάση δεδομένων, καθώς τα εν λόγω αρχεία περιέχουν όλη τα δεδομένα της βάσης (πίνακες, αντικείμενα κλπ). Σε αυτά φιλοξενούνται τα Tablespace (θα τα δούμε στο επόμενο κεφάλαιο).
- 3) **Redo Logs**¹³ = Τα συγκεκριμένα αρχεία, κρατάνε τα σημεία των αλλαγών στην βάση δεδομένων, τα οποία χρησιμοποιούνται σε περιπτώσεις που είτε η βάση υποστεί βλάβη από κάποια ενέργεια είτε η βάση τερματιστεί απότομα (πχ αν ο Server κολλήσει ή crash-άρει).

Αυτοί λοιπόν οι τρεις τύποι αρχείων συνθέτουν ολόκληρη την βάση δεδομένων. Θα τα δούμε με περισσότερη λεπτομέρεια στο επόμενο κεφάλαιο. Δείτε ωστόσο το παρακάτω σχήμα για να κατανοήσετε καλύτερα την δομή της βάσης:



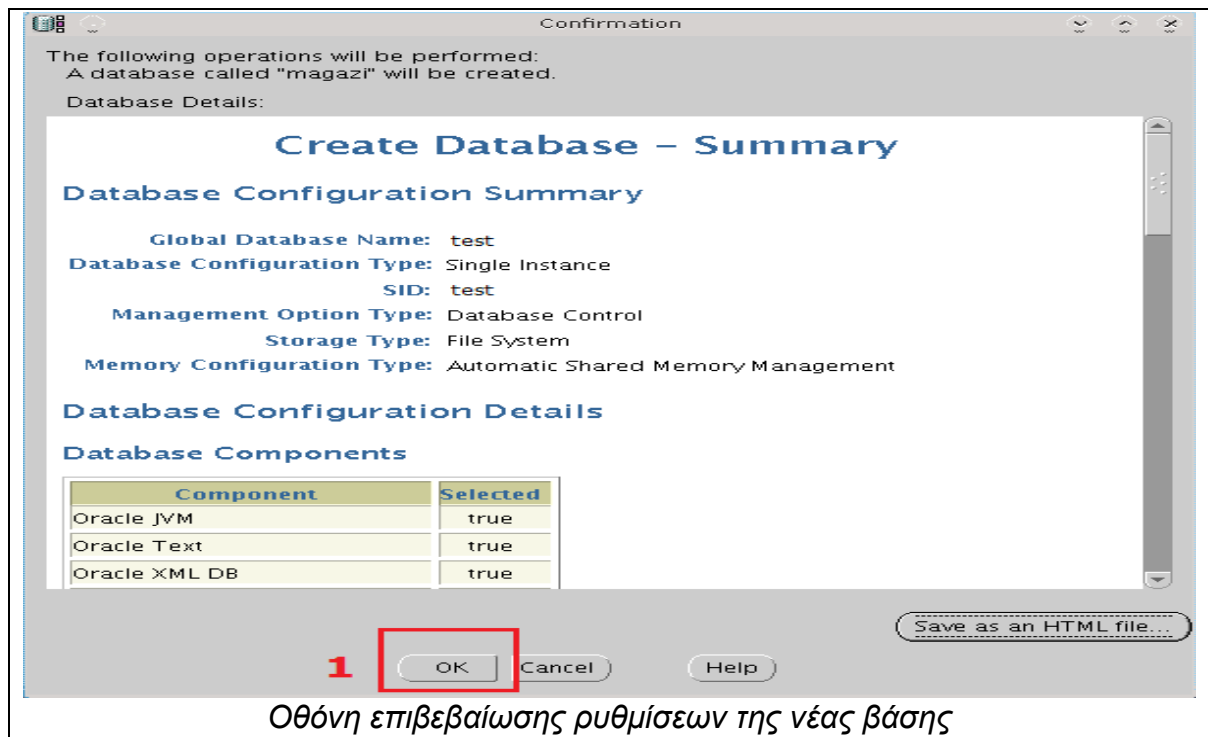
Το path αποθήκευσής τους είναι ο κατάλογος «ORACLE_BASE/oradata/<database_name>» όπου database name είναι το όνομα που ορίσαμε στην δημιουργία της βάσης, δηλαδή το όνομα της βάσης. Δεν χρειάζεται να αλλάξετε τις προεπιλεγμένες ρυθμίσεις του κάθε αρχείου εκτός αν

¹² Oracle Documentation → What is a Control File? – Site
(http://docs.oracle.com/cd/B28359_01/server.111/b28310/control001.htm#ADMIN11281)

¹³ Wikipedia Article → Redo log – Site =(http://en.wikipedia.org/wiki/Redo_log)

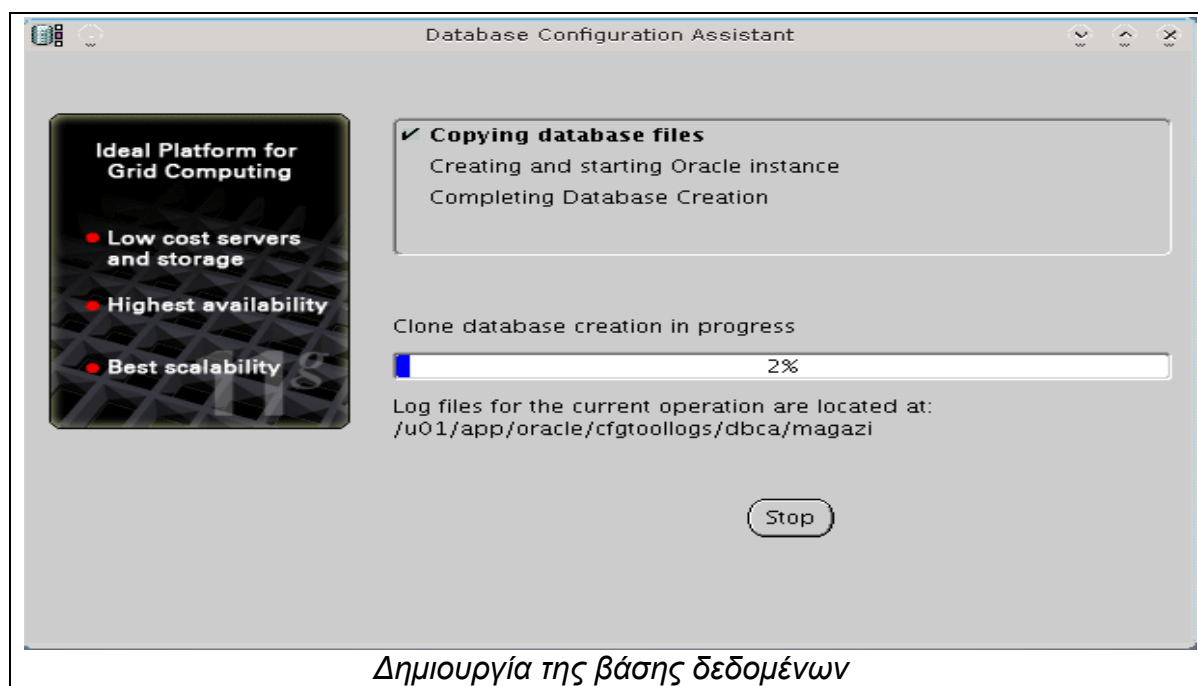
φυσικά θέλετε να τα αποθηκεύσετε σε συγκεκριμένο χώρο στον δίσκο σας. Ολοκληρώστε με «Next» (2).

Στην επόμενη καρτέλα πατάμε «Finish» για να ολοκληρώσουμε όλα τα βήματα. Θα μας εμφανίσει την οθόνη «Confirmation» όπου μπορούμε να κάνουμε μια επαλήθευση των ρυθμίσεων που κάναμε για την νέα μας βάση, την οποία βλέπουμε παρακάτω:

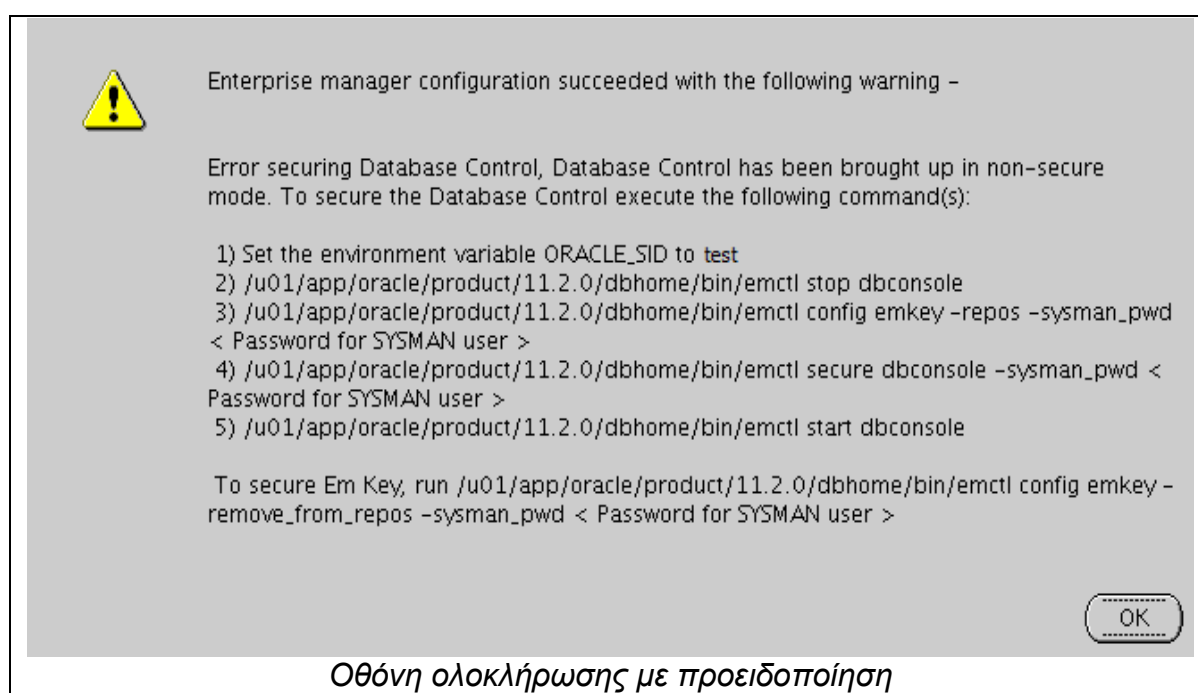


Σε αυτήν λοιπόν την καρτέλα κάνετε μια επισκόπηση των ρυθμίσεων που κάνατε για την νέα βάση που πρόκειται να δημιουργηθεί. Αν θέλετε να αλλάξετε μια ρύθμιση, πατάτε «Cancel» και επιλέγετε από τα προηγούμενα βήματα που περιγράψαμε την ρύθμιση που θέλετε.

Πατήστε «OK» (1) για να ξεκινήσει η δημιουργία της βάσης δεδομένων. Η διαδικασία θα πάρει κάποιον χρόνο, οπότε θα πρέπει να περιμένετε. Η οθόνη δημιουργίας της βάσης δεδομένων φαίνεται παρακάτω:

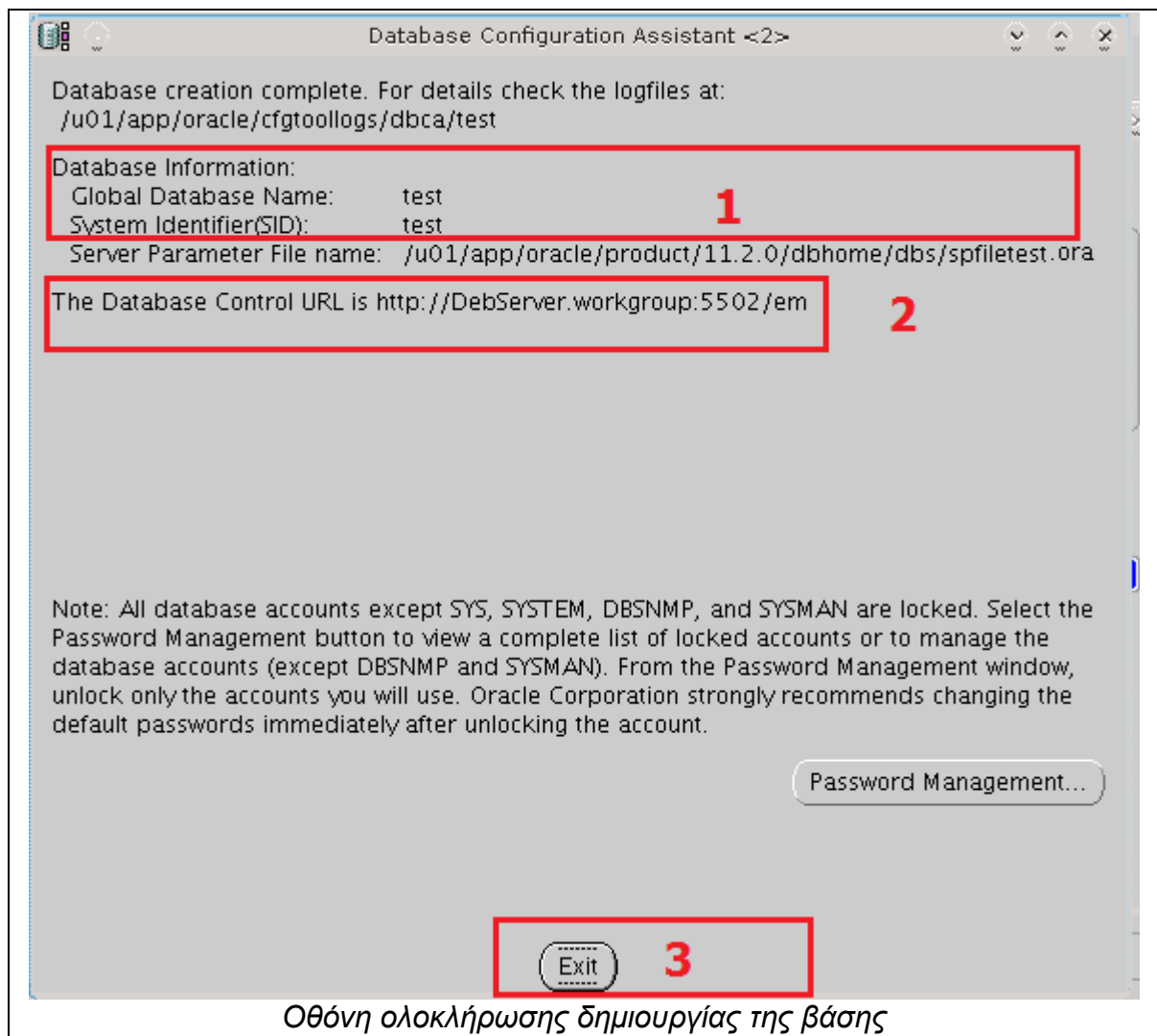


Εμείς τώρα απλά περιμένουμε να ολοκληρωθεί η διαδικασία δημιουργίας της βάσης δεδομένων. Η διαδικασία αυτή είναι πλήρως εξαρτώμενη από τις επιδόσεις του υπολογιστή. Όταν η διαδικασία ολοκληρωθεί θα εμφανίσει το παρακάτω παράθυρο:



Μην σας τρομάζει το θαυμαστικό στην οθόνη, απλά προειδοποίηση είναι για κάποιες ενέργειες που πρέπει να γίνουν. Εμείς θα πρέπει τώρα να πατήσουμε OK. Τα βήματα που περιγράφονται δεν είναι απαραίτητο να γίνουν.

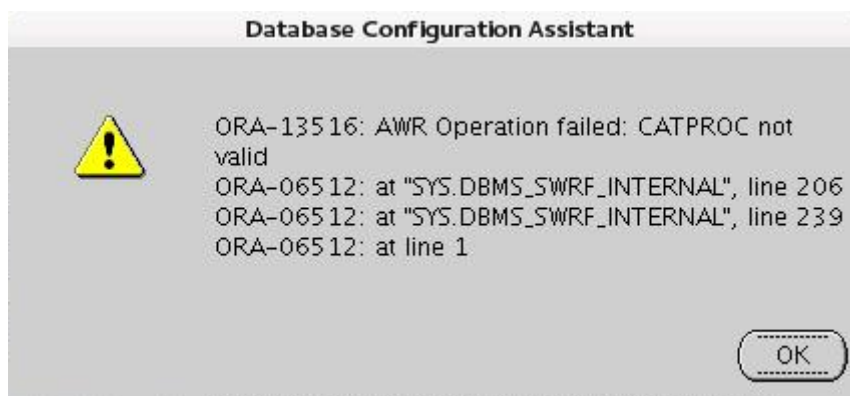
Το επόμενο παράθυρο θα σας επιβεβαιώσει πως η βάση δημιουργήθηκε επιτυχώς και θα σας δώσει ένα url της μορφής **SERVER NAME:PORT NUMBER/em** το οποίο αντιστοιχεί στο Web Management εργαλείο της βάσης, το **Oracle Enterprise Manager**. Γράψτε το κάπου να το θυμόμαστε γιατί πιθανών να το χρειαστείτε (θα το δούμε ωστόσο συνοπτικά σε παρακάτω ενότητα). Στην δική μας περίπτωση το url είναι «Debserver:5502/em».



Πλέον, η βάση δεδομένων μας είναι πραγματικότητα. Στην περιοχή «Database Information» (1) βλέπουμε το όνομα και το SID της βάσης μας. Στην δεύτερη μαρκαρισμένη περιοχή (2) βλέπουμε το URL προς τον διαχειριστή Oracle Enterprise Manager, για την διαχείριση της βάσης μας μέσα από τον περιηγητή ιστού μας (web browser). Πατώντας στο πλήκτρο «Exit» ολοκληρώνουμε την διαδικασία δημιουργίας βάσης δεδομένων. Η βάση μας είναι έτοιμη για δουλειά.

TROUBLESHOOT

Ενώ η βάση δεδομένων δημιουργείται, ξαφνικά βγαίνει ένα παράθυρο σφάλματος, που λέει «AWR Operation Failed: CATPROC NOT VALID» όπως παρακάτω:



Λύση¹⁴:

Κατ' αρχάς μην αγχώνεστε, το εν λόγω πρόβλημα διορθώνεται, το μόνο που πρέπει να ξέρετε είναι πως η βάση έχει δημιουργηθεί, απλά συνέβη ένα σφάλμα κατά την αρχικοποίηση των βιβλιοθηκών του PL/SQL. Πατήστε στο Terminal CTRL+C για να διακοπεί το DBCA, και ανοίξτε το SQL*PLUS από τον κατάλογο ORACLE_HOME/bin με τις εντολές:

```
oracle@debserver$ cd ORACLE_HOME/bin
```

```
oracle@debserver$ ./SQL*PLUS sys as sysdba
```

Το SQL Plus ανοίγει και ζητά πρώτα διαπιστευτήρια:

Password: -- τον κωδικό πρόσβασης που δώσατε στον SYS χρήστη

Connected to an idle instance

¹⁴ Oracle OTN Forums → Thread “ORA-13516:AWR Operation failed:Catproc not valid” → “You need rerun catproc.sql and recompile the invalid objects.” → Site (<https://forums.oracle.com/thread/2554900>)

Ξεκινήστε την βάση δεδομένων με την εντολή:

```
SQL> startup upgrade
```

Αφού ξεκινήσει η βάση και επιστρέψει σε Prompt, εκτελέστε πρώτα το script «catproc.sql» με την εντολή

```
SQL> @?/rdbms/admin/catproc.sql
```

Αφού εκτελεστεί το script, τα οποίο μπορεί να πάρει λίγο χρόνο, εκτελέστε και το «Utilrp.sql» με την εντολή:

```
SQL> @?/rdbms/admin/utlrp.sql
```

Αφήστε το να εκτελεστεί και στο πέρας του, κλείστε την βάση με:

```
SQL> shut immediate
```

Και ξεκινήστε την ξανά με

```
SQL> startup
```

```
SQL> quit # -- κλείσιμο SQL Plus
```

Το πρόβλημα λύθηκε και η βάση είναι εντάξει. Για να δείτε το port για το Oracle Enterprise Manager, γράψτε την παρακάτω εντολή:

```
oracle@debserver$ cat $ORACLE_HOME/install/portlist.ini
```

Αναζητήστε την καταχώρηση με το όνομα της βάσης σας.

4.3.1 - Εκκίνηση της Βάσης Δεδομένων

Για να ξεκινήσουμε την βάση δεδομένων, θα χρησιμοποιήσουμε το εργαλείο **SQL*PLUS**. SQL*PLUS είναι terminal based εφαρμογή η οποία λειτουργεί είτε ως απλός client για μια βάση της Oracle, είτε λειτουργεί ως πίνακας ελέγχου της βάσης αυτής. Ως εκ τούτου, για να ξεκινήσουμε και να τερματίσουμε την βάση θα χρησιμοποιούμε αυτό το εργαλείο. Βρίσκεται στον κατάλογο «ORACLE_HOME/bin».

Για να ξεκινήσουμε την βάση, θα πρέπει να το κάνουμε ως διαχειριστές. Οπότε θα πρέπει να συνδεθούμε στην βάση ως διαχειριστές μέσω του SQL*PLUS για να περατώσουμε το ζητούμενο. Βεβαιωνόμαστε πως το ORACLE_SID έχει ως τιμή το Instance ID της βάσης που θέλουμε να εκκινήσουμε, και εκτελούμε το SQL*PLUS ως εξής (χρήστης oracle):

```
oracle@debserver$ cd $ORACLE_HOME/bin --1
```

```
oracle@debserver$ ./SQL*PLUS --2
```

```
Enter user-name: sys as sysdba --3
```

```
Enter Password: --4
```

```
Connected to an idle instance
```

```
SQL> startup --5
```

```
Total System Global Area 588746752 bytes
```

```
Fixed Size 2215584 bytes
```

```
Variable Size 419430752 bytes
```

```
Database Buffers 159383552 bytes
```

```
Redo Buffers 7716864 bytes
```

```
Database mounted
```

```
Database Opened
```

```
SQL> --6
```

Πρώτα μεταβαίνουμε στον κατάλογο ORACLE_HOME/bin (1) και στην συνέχεια εκτελούμε το SQL*PLUS (2). Μας ζητάει τα Credentials μας, εμείς θα συνδεθούμε ως διαχειριστές, δηλαδή ως χρήστες SYS με ακολουθούμενο το «AS SYSDBA» που δηλώνει ότι ο χρήστης SYS θα πάρει τον ρόλο του Database Administrator (3). Εισάγουμε και το Password του χρήστη SYS (4) και μας πληροφορεί πως συνδεθήκαμε σε ένα **αδρανές Instance** (ανενεργή βάση). Ξεκινούμε τον Instance της βάσης και παράλληλα την ίδια την βάση δεδομένων με την εντολή **startup** (5) και μας εμφανίζονται πληροφορίες σχετικά με το μέγεθος των αρχείων της βάσης και τελικώς ότι η βάση είναι **mounted** (δηλαδή τα Control Files έχουν ανοίξει) και open (δηλαδή ανοιχτή για εργασία και τα datafiles με τα Tablespace της διαθέσιμα). Τελικώς επιστρέφουμε στο SQL*PLUS prompt (6).

Όπως είδατε στο SQL*PLUS δεν είπαμε σε ποια βάση θέλουμε να συνδεθούμε και να εκκινήσουμε. Πως το SQL*PLUS όμως ήξερε ότι θέλουμε την βάση TEST; Μα φυσικά, χάρη στην μεταβλητή ORACLE_SID που του το «μαρτύρησε». Τι σημαίνει αυτό πρακτικά; Σημαίνει ότι **για κάθε βάση δεδομένων που θέλουμε να ξεκινήσουμε, θα πρέπει πρώτα να αλλάζουμε το ORACLE_SID πριν εκτελέσουμε το SQL*PLUS.**

Τι είναι όμως τελικά αυτό το SID που το λέμε και το ξαναλέμε χωρίς να επεκτεινόμαστε σε λεπτομέρειες; Για να απαντήσουμε σε αυτό το ερώτημα, πρέπει πρώτα να εξηγήσουμε τι είναι ένα στιγμιότυπο.

Ορισμός

Ένα **στιγμιότυπο** (Instance) είναι ένα σύνολο από **δομές μνήμης** (Memory Structures) σε συνδυασμό με τα **Processes** του λειτουργικού συστήματος που διαχειρίζονται μια βάση δεδομένων¹⁵.

Κάθε μία βάση δεδομένων, έχει το δικό της Instance. Για να ξεκινήσει μια βάση δεδομένων, δηλαδή να μπορεί κάποιος να συνδεθεί και να εργαστεί σε αυτήν, πρέπει να δημιουργηθεί το Instance της στην μνήμη. Στη διαδικασία αυτή δεσμεύεται κάποιος χώρος στην μνήμη που ονομάζεται **System Global Area (SGA)** και το Instance συσχετίζεται με την βάση αυτή. Ο συσχετισμός του Instance με την βάση δεδομένων, ονομάζεται **προσάρτηση** (mount). Αφού λοιπόν η βάση συσχετιστεί με το Instance της, τότε ανοίγει και μπορεί κάποιος να συνδεθεί σε αυτήν. Το Instance όμως πρέπει να έχει ένα ξεχωριστό **όνομα**, το οποίο το προσδιορίζει μοναδικά. Ανώνυμο Instance **δεν** μπορεί να υπάρχει.

Ορισμός

Το όνομα λοιπόν του Instance, ονομάζεται **Αναγνωριστικό Συστήματος** (System Identifier) ή πιο απλά **SID**¹⁶. Ο συνδυασμός του ονόματος, με την τιμή της

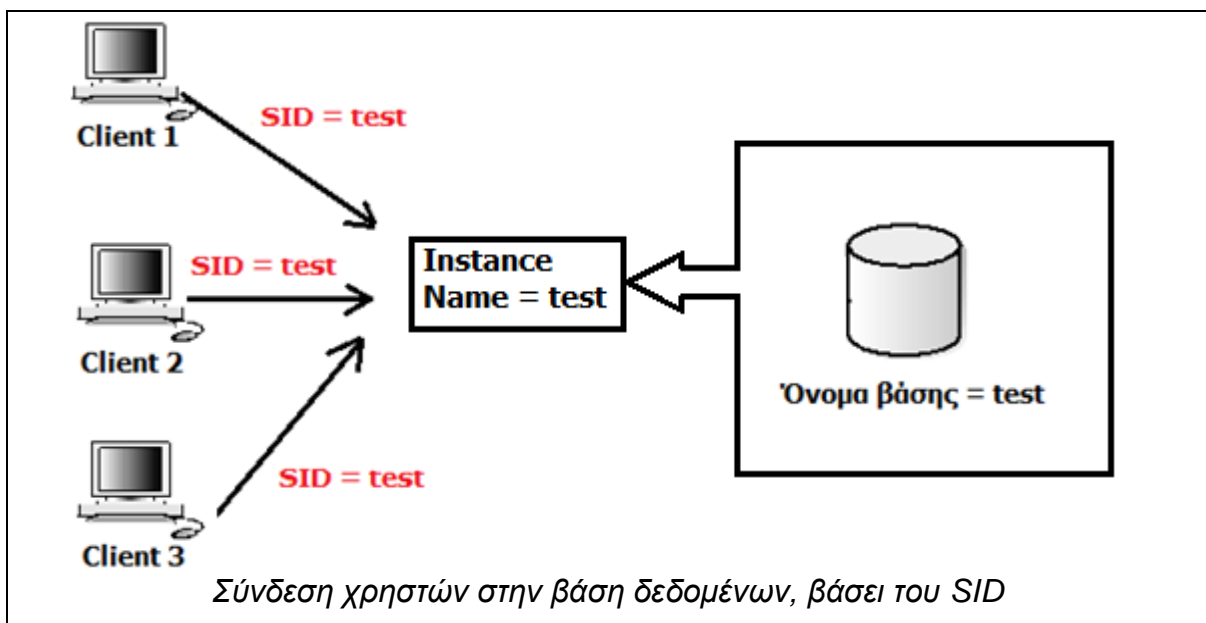
¹⁵ Oracle Documentation → Introduction to the Oracle Database Instance → “A database instance is a set of memory structures that manage database files” – Site (http://docs.oracle.com/cd/E11882_01/server.112/e25789/startup.htm#CNCPT601)

¹⁶ Oracle Documentation → Oracle System Identifier (SID) – Site (http://docs.oracle.com/cd/E11882_01/server.112/e25789/startup.htm#CNCPT89037)

μεταβλητής ORACLE_HOME παράγουν ένα μοναδικό αναγνωριστικό που προσδιορίζει μοναδικά την βάση δεδομένων.

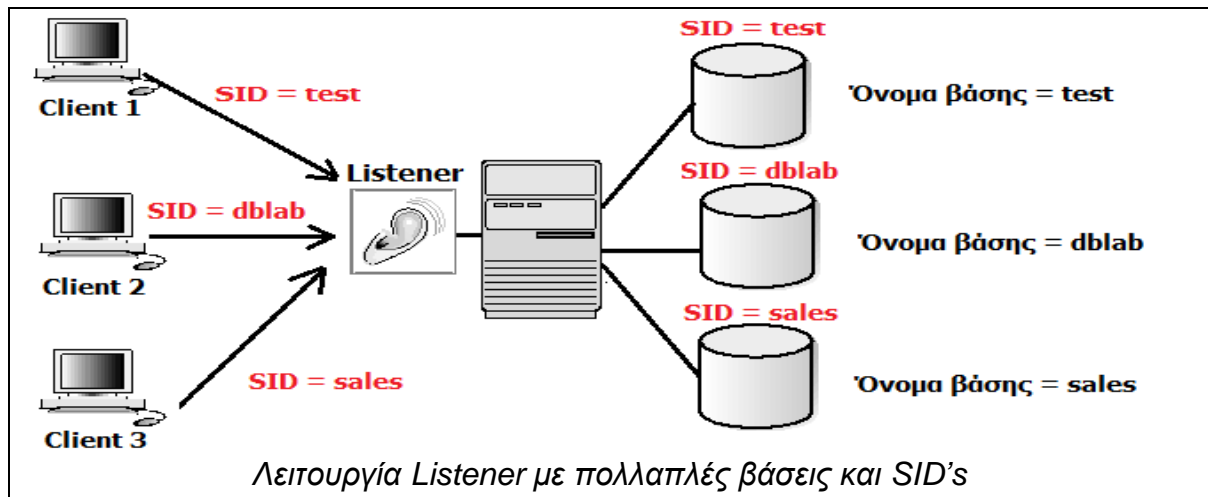
Να σημειωθεί ότι υπάρχει **ένα Instance ανά βάση δεδομένων**. Αν η βάση δεδομένων έχει περισσότερα από ένα Instances, τότε λέμε ότι η τελευταία είναι **βάση δεδομένων πολλαπλών στιγμιότυπων** (multiple instance database), τεχνολογία που η Oracle την ονομάζει **Real Application Cluster (RAC)**¹⁷ η οποία όμως είναι πέραν από τους σκοπούς του παρών συγγράμματος. Το Instance Name / SID της βάσης λοιπόν, αποθηκεύεται στην μεταβλητή περιβάλλοντος ORACLE_SID σε συστήματα Unix. Αναλόγως με την τιμή που έχει εκείνη την προκείμενη στιγμή, με την εντολή startup θα ξεκινήσει και η αντίστοιχη βάση δεδομένων, που έχει το Instance με το όνομα που περιέχει η ORACLE_SID.

Ο Client που πρόκειται να συνδεθεί στην βάση, πρέπει να γνωρίζει το SID της βάσης δεδομένων που θέλει να συνδεθεί. Κοινώς, με το SID από την Client Εφαρμογή του επιλέγει την βάση που θέλει. Το ίδιο όμως μπορεί να κάνει και με το Global Name (Service Name) της βάσης δεδομένων. Ένα παράδειγμα βλέπουμε στο παρακάτω σχήμα που παραθέτουμε:



¹⁷ Oracle Documentation → Database Instance Configurations → “Oracle Real Application Clusters (Oracle RAC) configuration ... A one-to-many relationship exists between the database and instances” – Site (http://docs.oracle.com/cd/E11882_01/server.112/e25789/startup.htm#CNCPT89033)

Μπορούν όμως να τρέχουν περισσότερες βάσεις δεδομένων στον Server, με διαφορετικό SID και Global Name η κάθε μία. Στο παρακάτω σχήμα βλέπουμε ένα κατατοπιστικό παράδειγμα:



Οι βασικοί 6 τρόποι (υπάρχουν περισσότεροι αλλά δεν θα μας απασχολήσουν) εκκίνησης της βάσης δεδομένων¹⁸, αναλόγως με το επιθυμητό αποτέλεσμα από το διαχειριστή:

Είδος Startup	Αποτέλεσμα
STARTUP {OPEN}	Εκκινεί το Instance της βάσης, την προσαρτεί και στην συνέχεια την ανοίγει.
STARTUP RESTRICT	Εκκινεί την βάση σε Restricted Mode, δηλαδή μόνο οι διαχειριστές έχουν δικαίωμα πρόσβασης σε αυτήν. Χρησιμοποιείται σε εργασίες συντήρησης για αλλαγή της υπόστασης της βάσης.
STARTUP NOMOUNT	Εκκινεί μόνο το Instance της βάσης χωρίς να την προσαρτήσει και ανοίξει.
STARTUP MOUNT	Εκκινεί το Instance της βάσης και την

¹⁸ Oracle Documentation → STARTUP – Site
(http://docs.oracle.com/cd/B19306_01/server.102/b14357/ch12045.htm)

	προσαρτίζει, χωρίς να την ανοίγει.
STARTUP FORCE	Τερματίζει το Instance της βάσης, κλείνει την βάση και στην συνέχεια την εκκινεί ξανά μαζί με το instance της.
STARTUP UPGRADE	Ανοίγει την βάση σε ειδική λειτουργία αναβάθμισης αν πρόκειται π.χ. να εγκατασταθεί νέα έκδοση του DBMS

4.3.2 - Δοκιμή της βάσης Δεδομένων

Ωραία, η βάση μας δημιουργήθηκε και ενεργοποιήθηκε. Δεν πρέπει όμως να την δοκιμάσουμε να δούμε κατά πόσο λειτουργική ή όχι, είναι; Ξεκινούμε πάλι το SQL*PLUS ως χρήστες SYS as SYSDBA με την διαδικασία που περιγράψαμε πιο πάνω και έφτασε η ώρα να γράψουμε λίγο **SQL κώδικα**. Θα δημιουργήσουμε ένα πίνακα, θα του εισάγουμε μια γραμμή και μετά θα στείλουμε επερώτημα (Query) στην βάση για να δούμε αν ο πίνακας περιέχει τιμές. Ο κώδικας μας δίδεται παρακάτω:

```
SQL> CREATE TABLE myFirstTable (id integer not null primary key,
name varchar(30)
);
TABLE CREATED. # -- 1ο Θετικό σημάδι, ο πίνακας δημιουργήθηκε!!!
SQL> INSERT INTO myFirstTable VALUES (1,'Hello_from_Oracle_!!!');
1 ROW CREATED. # -- 2ο Θετικό σημάδι, η νέα γραμμή μπήκε!!!
SQL> SELECT * FROM myFirstTable;

   ID NAME
-----
1 Hello from Oracle!!! --3ο Θετικό σημάδι, το Query επιστρέφει εγγραφές!!!
SQL>
```

Βλέπουμε πως η δημιουργία του πίνακα, η προσθήκη γραμμής σε αυτόν και το επερώτημα SELECT στην βάση, λειτουργούν άψογα. Τώρα μπορούμε να πούμε «με γεια την νέα μας βάση».

4.3.3 - Δημιουργία περισσότερων βάσεων

Αν θέλουμε να δημιουργήσουμε νέα βάση δεδομένων θα πρέπει να εκτελέσουμε με την σειρά τα παρακάτω βήματα:

- 1) Αλλαγή του ORACLE_SID με το SID της νέας μας βάσης
- 2) Εκτέλεση του Net Manager για δημιουργία νέου Name Service (με τον ίδιο Listener, δεν χρειάζεται εκ νέου δημιουργία)
- 3) Δημιουργία της νέας βάσης με το Database Configuration Assistant (DBCA) με τον ίδιο ακριβώς τρόπο που περιγράψαμε παραπάνω
- 4) Εκκίνηση της βάσης μέσα από το SQL*PLUS ως χρήστης SYS as SYSDBA.

Με αυτόν τον τρόπο μπορούμε να δημιουργήσουμε όσες βάσεις θέλουμε, για την ακρίβεια όσες μας επιτρέπει ο αποθηκευτικός μας χώρος. Υπενθυμίζουμε ότι όποτε θέλουμε να δημιουργήσουμε μια νέα βάση, ή να ενεργοποιήσουμε (startup) μια ήδη υπάρχουσα, θα πρέπει να φροντίζουμε να αλλάζουμε την μεταβλητή περιβάλλοντος **ORACLE_SID** στην τιμή που ανταποκρίνεται στο όνομα/instance id της βάσης αυτής.

4.3.4 - Τερματισμός Βάσης Δεδομένων

Αν για οποιονδήποτε λόγο θέλετε να τερματίσετε την βάση δεδομένων, πχ για εργασίες συντήρησης του Server ή αλλαγή της υπόστασής της, το πραγματοποιείτε πάλι μέσα από το SQL*PLUS (ως χρήστες SYS as SYSDBA με ORACLE_SID το SID της βάσης) με τις εντολές:

```
SQL> shut normal; --1
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> disconnect --2
Disconnected
SQL> quit --3
```

Με την εντολή «shut(down) normal» (1) διατάζουμε το SQL*PLUS να κάνει dismount (από-προσάρτηση) και να κλείσει την βάση δεδομένων. Επίσης τερματίζεται και το Instance της βάσης, οπότε περεταιίρω συνδέσεις σε αυτήν από Client πλέον δεν γίνονται. Στην συνέχεια αποσυνδεόμαστε από το SQL*PLUS από χρήστες SYS (2) και τέλος κλείνουμε το SQL*PLUS (3) και επιστρέφουμε το Shell του λειτουργικού συστήματος.

Αναλόγως την περίπτωση, η βάση δεδομένων μπορεί να τερματιστεί με τέσσερις τρόπους¹⁹, γεγονός εξαρτώμενο από τις απαιτήσεις του διαχειριστή:

Είδος Shutdown	Αποτέλεσμα
SHUTDOWN NORMAL	Αποπροσαρτίζει και κλείνει την βάση δεδομένων, τερματίζοντας παράλληλα και το Instance της. Δημιουργεί και σημεία επαναφοράς (Checkpoints). Η βάση κλείνει αφού αποσυνδεθεί και ο τελευταίος client από αυτήν. Η ποιο

¹⁹ Oracle Documentation – SHUTDOWN – Site
(http://docs.oracle.com/cd/B19306_01/server.102/b14357/ch12042.htm)

	σωστή μέθοδος.
SHUTDOWN IMMEDIATE	Αποπροσαρτίζει και κλείνει την βάση, τερματίζει το Instance της και αποσυνδέει αμέσως όλους τους clients. Αν υπάρχουν ενεργά SQL Statements για επεξεργασία, εκτελούνται πριν τον τερματισμό. Γρηγορότερο από το Normal.
SHUTDOWN ABORT	Τερματίζει άμεσα το Instance και κλείνει άμεσα την βάση. Οι χρήστες αποσυνδέονται κατευθείαν ενώ τα ενεργά SQL Statements τερματίζονται. Στην επόμενη εκκίνηση θα χρειαστεί επαναφορά (recovery) καθώς δεν δημιουργήθηκαν Checkpoints. Είναι ο ταχύτερος τρόπος τερματισμού της βάσης (και ο ποιο επικίνδυνος).
SHUTDOWN TRANSACTIONAL	Τερματίζει την βάση αφού ολοκληρωθούν όλες οι ενεργές συναλλαγές (transactions) σε αυτήν, χωρίς να επιτρέπονται νέες. Μόλις γίνει το τελευταίο Commit/Rollback τότε η βάση τερματίζει.

Ολοκληρώνοντας...

Αυτές λοιπόν ήταν οι απαραίτητες ενέργειες που έπρεπε να γίνουν στον Server ώστε να μπορεί πλέον να επιτελέσει τον ρόλο του. Παρόλο που στην παρούσα κατάστασή του μπορεί να λειτουργήσει αποδοτικά, ωστόσο πρέπει να εισάγουμε χρήστες στην βάση, να δώσουμε ρόλους και να πραγματοποιήσουμε πρόσθετες ενέργειες και παραμετροποιήσεις για να λειτουργήσει στο μέγιστο των δυνατοτήτων του. Αφού όμως η βάση μας είναι ενεργή και ο Listener περιμένει συνδέσεις, τότε μπορούμε να δημιουργήσουμε ένα πολύ αξιόλογο σύστημα.

Στο επόμενο κεφάλαιο λοιπόν, θα ασχοληθούμε με τα εργαλεία διαχείρισης του Server, τις ενέργειες εισαγωγής μα και διαγραφής χρηστών, την απόδοση ρόλων και δικαιωμάτων και άλλα ενδιαφέροντα ζητήματα.

Βιβλιογραφία Κεφαλαίου

A/A	Πηγή	Άρθρο/Ενότητα	Ιστοσελίδα
1	Oracle Documentation	Configuring and Administering the Listener	http://docs.oracle.com/cd/B10501_01/network.920/a96580/listener.htm
2	Oracle Documentation	Overview of Users and Security - Administrative User Accounts Provided by Oracle Database	http://docs.oracle.com/cd/B16276_01/doc/server.102/b14196/users_secure001.htm
3	Oracle documentation	Managing Users and Security - The SYS and SYSTEM Users	http://docs.oracle.com/cd/E17781_01/server.112/e18804/users_secure.htm#autold4
4	Oracle Documentation	Managing Users And Security - The SYSDBA System Privilege	http://docs.oracle.com/cd/E17781_01/server.112/e18804/users_secure.htm#autold4
5	Oracle Documentation	What is a control file?	http://docs.oracle.com/cd/B28359_01/server.111/b28310/control001.htm#ADMIN11281
6	Wikipedia	Article "Redo log"	http://en.wikipedia.org/wiki/Redo_log
7	Oracle OTN Forums	Thread "ORA-13516:AWR Operation failed:Catproc not valid"	https://forums.oracle.com/thread/2554900
8	Oracle Documentation	Introduction to the Oracle Database Instance	http://docs.oracle.com/cd/E11882_01/server.112/e25789/startup.htm#CNCPT601
9	Oracle Documentation	Oracle System Identifier (SID)	http://docs.oracle.com/cd/E11882_01/server.112/e25789/startup.htm#CNCPT89037
10	Oracle Documentation	Database Instance Configurations	http://docs.oracle.com/cd/E11882_01/server.112/e25789/startup.htm#CNCPT89033
11	Oracle Documentation	STARTUP	http://docs.oracle.com/cd/B19306_01/server.102/b14357/ch12045.htm
12	Oracle	SHUTDOWN	http://docs.oracle.com/cd/B19306

	Documentation		_01/server.102/b14357/ch12042. htm
--	---------------	--	--

ΚΕΦΑΛΑΙΟ 5 – ΔΙΑΧΕΙΡΗΣΗ SERVER ΚΑΙ ΠΡΟΣΘΕΤΑ ΕΡΓΑΛΕΙΑ

Σε αυτό το κεφάλαιο θα ασχοληθούμε με την διαχείριση του εξυπηρετητή, ώστε να εκτελεί σωστά το έργο του και να εξυπηρετεί αποδοτικά τους χρήστες που εργάζονται σε αυτόν. Δεν σταματάμε όμως εδώ, καθώς θα περιγράψουμε κάποια πολύ σημαντικά εργαλεία που μας παρέχει η Oracle, ώστε να γίνουν οι εργασίες μας στον Server εύκολες και συνάμα λειτουργικές.

5.1 - Οι χρήστες της βάσης δεδομένων

Από την δημιουργία της βάσης και μετά, οι μόνοι χρήστες που υπάρχουν στο σύστημα είναι οι SYS, SYSTEM, DBSNMP και SYSMAN οι οποίοι είναι όλοι διαχειριστές. Υπάρχει βέβαια και ένας απλός χρήστης, για λόγους δοκιμών φυσικά, ο χρήστης SCOTT με κωδικό πρόσβασης TIGER, αλλά εμείς προφανώς θα θέλουμε να δημιουργήσουμε πρόσθετους. (Οι παραπάνω χρήστες είναι προεπιλογή της Oracle και δημιουργούνται με κάθε βάση δεδομένων).

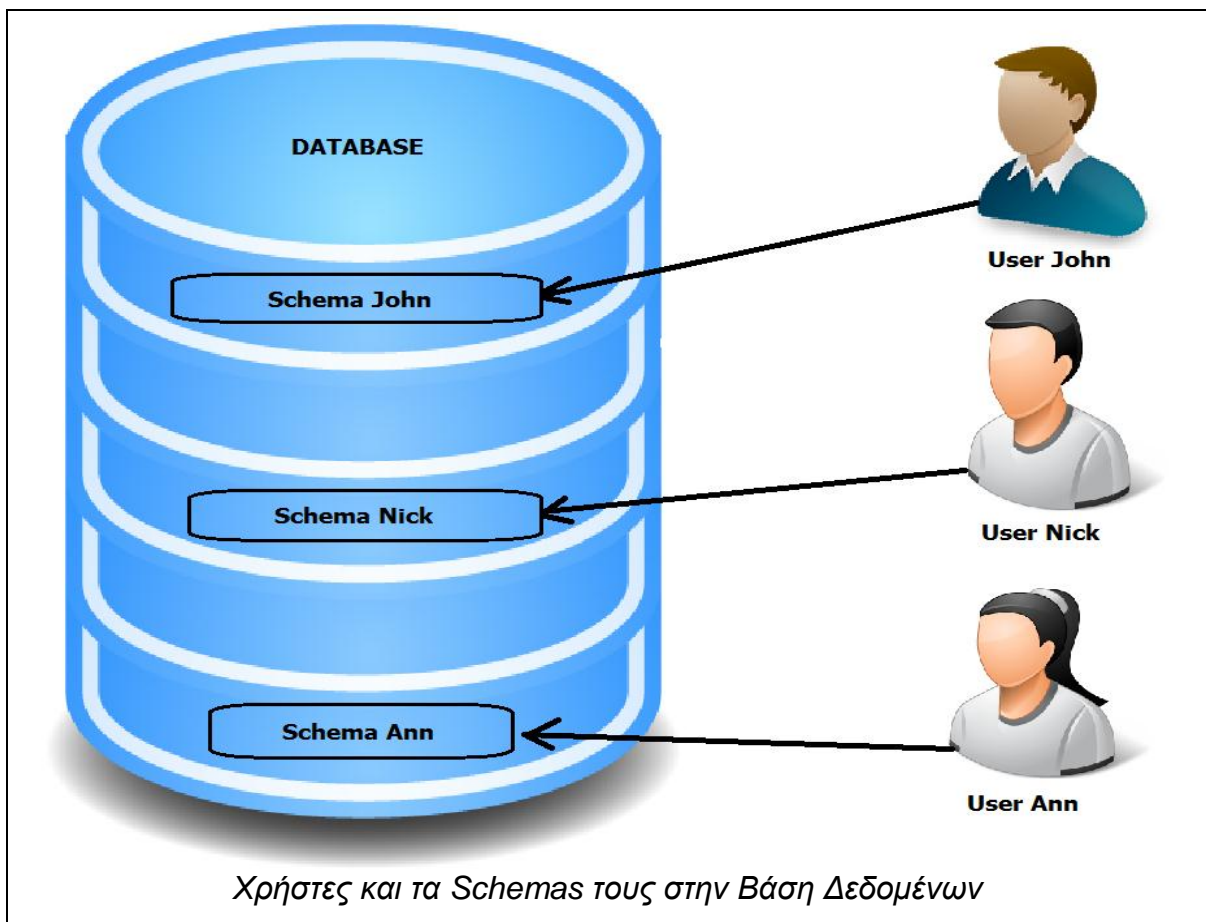
Ορισμός

Ένας **χρήστης** (user) είναι ένα άτομο που εργάζεται στην βάση δεδομένων. Οι ενέργειες και το έργο που μπορεί να επιτελέσει στην βάση καθορίζεται από τα **δικαιώματα** (privileges) που έχει σε αυτήν, δηλαδή τι του επιτρέπεται να κάνει ή όχι πάνω στην βάση δεδομένων.

Ένα επίσης βασικό χαρακτηριστικό που χαρακτηρίζει έναν χρήστη είναι ο **ρόλος** (role) του. Ο ρόλος ενός χρήστη, στην ουσία είναι το σύνολο των δικαιωμάτων του πάνω στην βάση δεδομένων. Μπορεί δηλαδή ένας χρήστης να έχει το ρόλο του διαχειριστή, μπορεί όμως να έχει το ρόλο του απλού χρήστη. Από πού εξαρτάται αυτό όμως; Καθαρά από την **ιδιότητα** του χρήστη στο σύστημα. Ένας φοιτητής για παράδειγμα υποχρεωτικά θα πρέπει να έχει το ρόλο του απλού χρήστη, ώστε τα δικαιώματά του στην βάση να είναι περιορισμένα στον δικό του χώρο πάνω σε αυτήν. Αυτό εκτός από ιεραρχική δομή των ρόλων πάνω στην

βάση, παρέχει και **ασφάλεια**. Φαντάζεστε τι θα γινόταν αν ο οποιοσδήποτε μπορούσε π.χ. να διαγράψει (drop) ή να τροποποιήσει τους πίνακες μιας βάσης δεδομένων μιας εταιρίας; Δεν θα οδηγούσε σε σίγουρη καταστροφή; Σίγουρα ναι. Οπότε το bottom line είναι ένα: **Αποδίδουμε στους χρήστες τους ρόλους που τους αναλογούν.**

Δεν πρέπει επίσης να παραλείψουμε ότι ο χώρος στον οποίο ο χρήστης εργάζεται, ονομάζεται **Σχήμα**²⁰ (schema) το οποίο είναι ιδιωτικό σε κάθε χρήστη, εκτός αν δημιουργηθούν κοινόχρηστα. Στην Oracle, κάθε χρήστης που δημιουργείται, αυτόματα δημιουργείται και ένα schema για αυτόν με το ίδιο όνομα, και φυσικά ιδιωτικό μόνο για αυτόν. Αυτό θα το διαπιστώσουμε και παρακάτω που θα δημιουργήσουμε χρήστες. Στο παρακάτω σχήμα σας δείχνουμε τι εννοούμε:



²⁰ Ask Tom → What is the meaning of schema? → “You should consider a schema to be the user account and collection of all objects therein as a schema for all intents and purposes” – Site (http://asktom.oracle.com/pls/asktom/f?p=100:11:0:::P11_QUESTION_ID:6162110256950)

5.1.1 - Δημιουργία Χρηστών στην βάση και απόδοση δικαιωμάτων

Η δημιουργία νέων χρηστών στην βάση είναι πολύ εύκολη υπόθεση. Δικαιώματα να δημιουργεί και να τροποποιεί χρήστες έχει μόνο ο χρήστης με **δικαιώματα διαχειριστή** (DBA, SYSDBA), και ένας από αυτούς είναι ο γνωστός SYS. Η ενέργεια αυτή πραγματοποιείται μέσα από το γνωστό μας, πλέον, SQL*PLUS. Το εκτελούμε και συνδεόμαστε σε αυτό ως SYS (AS SYSDBA) και γράφουμε τις παρακάτω εντολές, **αφού βεβαιωθούμε ότι η βάση έχει ξεκινήσει με startup:**

```
--Syntax: CREATE USER <όνομα_χρήση> IDENTIFIED BY <password>;  
CREATE USER lefteris IDENTIFIED by secret;
```

Η εντολή **CREATE USER** δημιουργεί έναν νέο χρήστη με όνομα της επιλογής μας (π.χ. lefteris) και με έναν προσωπικό κωδικό πρόσβασης της επιλογής μας (IDENTIFIED BY... Clause).

Η παραπάνω εντολή δημιούργησε έναν χρήστη στην βάση δεδομένων, αλλά ο χρήστης αυτός δεν του έχει αποδοθεί ρόλος, οπότε προς το παρών απλά υπάρχει στην βάση. Για να του επιτρέψουμε να πραγματοποιήσει συγκεκριμένες ενέργειες, θα πρέπει να του δώσουμε δικαιώματα. Αυτό επιτυγχάνεται με την εντολή **GRANT** όπως βλέπουμε παρακάτω:

```
--Syntax: GRANT <δικαιώμα-1>, <δικαιώμα-2>, <δικαιώμα-n> TO <χρήστης>;  
SQL> GRANT CREATE TABLE, CREATE SESSION TO lefteris;
```

Με την Grant δώσαμε δικαιώματα στον χρήστη να πραγματοποιήσει κάποιες ενέργειες, στο συγκεκριμένο παράδειγμα, δώσαμε δικαιώματα δημιουργίας συνεδρίας (session) και δημιουργίας πινάκων στην βάση δεδομένων, στο χρήστη lefteris. Ο χρήστης στον οποίον διαβιβάζονται τα δικαιώματα,

ονομάζεται **δικαιούχος**²¹ (grantee). Αντίθετα ο χρήστης που τα διαβιβάζει (στην προκειμένη, ο SYS) ονομάζεται **εκχωρητής**²² (grantor).

Αν σε περίπτωση που θέλουμε να αφαιρέσουμε δικαιώματα από τον χρήστη τι κάνουμε; Χρησιμοποιούμε την εντολή **REVOKE** όπως βλέπουμε παρακάτω:

```
--Syntax: REVOKE <δικαίωμα-1>, <δικαίωμα-2>, <δικαίωμα-n> FROM
--<χρήστης>;
SQL> REVOKE CREATE TABLE, CREATE SESSION FROM lefteris;
```

Η εντολή REVOKE λοιπόν αφαιρεί δικαιώματα από έναν χρήστη, **που του έχουν αποδοθεί με την εντολή GRANT**. Έτσι λοιπόν μπορούμε να περιορίζουμε τον χρήστη όσον αφορά τις επιτρεπόμενες ενέργειές του πάνω στην βάση.

Με σωστό συνδυασμό της GRANT και της REVOKE μπορούμε αποδοτικά να ρυθμίσουμε τους κατάλληλους χρήστες στο σύστημα. Παρακάτω δίδονται μερικά από τα πιο συνηθισμένα δικαιώματα που μπορούν να δοθούν ή να αφαιρεθούν από την GRANT και την REVOKE αντίστοιχα:

Όνομα Δικαιώματος	Λειτουργία Δικαιώματος
CREATE TABLE	Επιτρέπει την δημιουργία πινάκων στην βάση δεδομένων που ανήκουν στον ίδιο χρήστη (και drop/alter/update/delete αυτού αντίστοιχα).
CREATE SESSION	Επιτρέπει την σύνδεση με την βάση, δηλαδή την εγκαθίδρυση συνεδρίας με αυτήν. Απαραίτητο για Clients.
ALTER SESSION	Επιτρέπει την αλλαγή χαρακτηριστικών της παρούσας συνεδρίας (πχ αλλαγή schema)

²¹ Oracle Documentation → DBA_TAB_PRIVS → Related View → GRANTEE → “Name of the user to whom access was granted” – Site
http://docs.oracle.com/cd/B19306_01/server.102/b14237/statviews_4151.htm

²² Oracle Documentation → DBA_TAB_PRIVS → Related View → GRANTOR → “Name of the user who performed the grant” – Site
http://docs.oracle.com/cd/B19306_01/server.102/b14237/statviews_4151.htm

CREATE TRIGGER	Επιτρέπει την δημιουργία εναύσματος στο χρήστη.
CREATE TYPE	Επιτρέπει την δημιουργία User Defined Types, δηλαδή τύπων οριζόμενων από το χρήστη αυτόν.
CREATE PROCEDURE	Επιτρέπει την δημιουργία Procedures και Functions, ιδιοκτησίας του ίδιου χρήστη (και κλήση αυτής αντίστοιχα).
EXECUTE ON <procedure/function name> (ή ANY PROCEDURE)	Επιτρέπει την εκτέλεση μιας συγκεκριμένης procedure ή function, ή οποιασδήποτε άλλης αν οριστεί το ANY PROCEDURE.
CREATE VIEW	Επιτρέπει την δημιουργία Όψεων, ιδιοκτησίας του ίδιου χρήστη.
SELECT ON <table name> (ή ANY TABLE)	Επιτρέπει επερώτημα SELECT για έναν συγκεκριμένο. Το ANY TABLE επιτρέπει το παραπάνω σε οποιονδήποτε πίνακα στη βάση.
INSERT ON <table name> (ή ANY TABLE)	Επιτρέπει εισαγωγή γραμμών σε ένα συγκεκριμένο πίνακα. Αν χρησιμοποιηθεί ANY TABLE, το παραπάνω πραγματοποιείται σε οποιονδήποτε πίνακα της βάσης.
REFERENCES <table name>	Επιτρέπει την δημιουργία Foreign Key Reference στο συγκεκριμένο πίνακα.
UPDATE ON <table name> (ή ANY TABLE)	Επιτρέπει την τροποποίηση εγγραφών ενός συγκεκριμένου πίνακα, η οποιοδήποτε πίνακα αν οριστεί το ANY TABLE.
DELETE ON <table name> (ή ANY TABLE)	Επιτρέπει την διαγραφή πεδίων από έναν συγκεκριμένο πίνακα, η από οποιονδήποτε πίνακα αν οριστεί το ANY TABLE.

DROP ANY TABLE	Επιτρέπει την διαγραφή οποιοδήποτε πίνακα από τη βάση
DROP ANY PROCEDURE	Επιτρέπει την διαγραφή οποιασδήποτε procedure ή function από τη βάση
DROP ANY VIEW	Επιτρέπει την διαγραφή οποιασδήποτε όψης πάνω στην βάση
ALTER ANY TABLE	Επιτρέπει την τροποποίηση της δομής οποιουδήποτε πίνακα
ALTER ANY PROCEDURE	Επιτρέπει την τροποποίηση οποιασδήποτε procedure ή function στην βάση
ALTER ANY VIEW	Επιτρέπει την τροποποίηση οποιασδήποτε όψης της βάσης.
CREATE USER	Δικαίωμα δημιουργίας νέου χρήστη, μόνο για διαχειριστές

Τα παραπάνω δικαιώματα που παραθέτουμε είναι μόνο μερικά από τα πολλά δικαιώματα που ορίζει η Oracle. Είναι όμως τα πιο συνηθισμένα, οπότε θεωρούμε πως σας καλύπτουν σε μεγάλο βαθμό. Τα παραπάνω είναι γνωστά ως «**System Privileges**»²³

Keep in mind....

Όποτε δημιουργείτε έναν καινούριο χρήστη στο σύστημα, θα πρέπει άμεσα και πριν από οτιδήποτε άλλο, να του δώσετε το δικαίωμα CREATE SESSION, διαφορετικά δεν θα μπορέσει ποτέ να συνδεθεί στην βάση δεδομένων. Από κει και πέρα του δίνετε ότι δικαιώματα εσείς κρίνεται σκόπιμα.

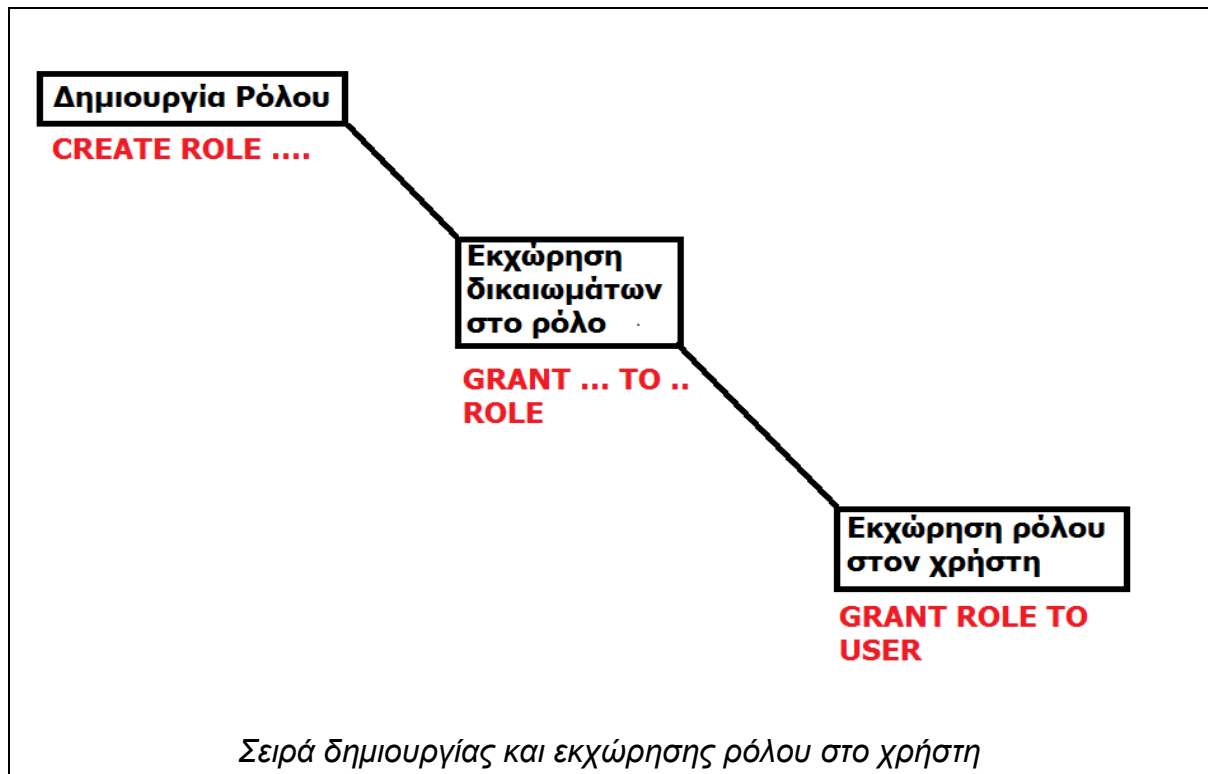
²³ Oracle Documentation → Privileges → System Privileges – Site
(http://docs.oracle.com/cd/E11882_01/timesten.112/e21642/privileges.htm#TTSQL338)

5.1.2 - Ρόλοι Χρηστών

Η απόδοση δικαιωμάτων σε ένα χρήστη είναι κάτι το θεμιτό, ωστόσο όμως αν έχουμε έναν μεγάλο αριθμό χρηστών οι οποίοι προσδοκείται να έχουν ακριβώς τα ίδια δικαιώματα περιπλέκει λίγο τα πράγματα. Και αυτό επειδή, αν οι χρήστες αυτοί χρειάζεται να τους αποδοθούν πολλά από τα παραπάνω δικαιώματα, θα πρέπει να γράψουμε ατελείωτες σειρές κώδικα στο SQL*PLUS για να περατώσουμε το ζητούμενο. Δεν θα ήταν καλύτερα αν μπορούσαμε όλα τα επιθυμητά δικαιώματα να τα ομαδοποιήσουμε σε ένα group και να τα αποδίδουμε όλα μαζί σε ένα χρήστη με μία κίνηση; Φυσικά και θα ήταν.

Η ομαδοποίηση πολλών δικαιωμάτων σε μια ομάδα, ονομάζεται **Ρόλος** (role). Η Oracle περιέχει έτοιμους ρόλους προς απόδοση σε χρήστη, ωστόσο επιτρέπει την δημιουργία custom made ρόλων.

Η δημιουργία ενός ρόλου είναι εύκολη υπόθεση. Κάθε ρόλος πρέπει να έχει ένα χαρακτηριστικό όνομα που προσδιορίζει την ιδιότητα του χρήστη στον οποίο εκχωρείται. Π.χ. αν ένας χρήστης είναι φοιτητής, τότε μπορούμε να δημιουργήσουμε τον ρόλο student και να τον εκχωρήσουμε σε αυτόν. Με την δημιουργία του ρόλου, μπορούμε να αρχίσουμε να εκχωρούμε δικαιώματα στον ρόλο αυτόν. Έτσι, ο ρόλος περιέχει ένα σύνολο δικαιωμάτων τα οποία μπορούμε με τη σειρά μας να τα εκχωρήσουμε στον χρήστη με μία κίνηση. Στο παρακάτω σχήμα βλέπουμε την ιεραρχία δημιουργίας ενός ρόλου:



Η δημιουργία ρόλου στο SQL*PLUS γίνεται με τις παρακάτω εντολές (χρήστης SYS)

```
--Syntax: CREATE ROLE <όνομα ρόλου>;
```

```
SQL> CREATE ROLE employee;
```

Δημιουργήσαμε έναν ρόλο λοιπόν, στην προκειμένη περίπτωση τον ρόλο employee. Ο ρόλος μας όμως δεν εμπεριέχει καθόλου δικαιώματα, οπότε δεν έχει νόημα να εκχωρηθεί σε κάποιον χρήστη. Πρέπει πρώτα να του αποδώσουμε δικαιώματα με την GRANT:

```
--Syntax: GRANT <δικαίωμα-1>, <δικαίωμα-2>, <δικαίωμα-n> to <ρόλος>;
```

```
SQL> GRANT create table, create session, create procedure to employee;
```

Εκχωρήσαμε λοιπόν δικαιώματα στον ρόλο. Να σημειωθεί ότι αυτό το κάνουμε μια φορά. Από δω και πέρα, ο ρόλος μας είναι έτοιμος να εκχωρηθεί σε έναν ή περισσότερους χρήστες.

```
--Syntax: GRANT <ρόλος> TO <χρήστης>;
```

```
SQL> GRANT employee TO lefteris;
```

Εκχωρήσαμε λοιπόν και τον ρόλο στον χρήστη, στην προκειμένη περίπτωση τον ρόλο employee στο χρήστη lefteris. Από δω και πέρα, όλοι οι χρήστες που χρειάζεται να τους αποδοθούν πανομοιότυπα δικαιώματα, μπορούν να αναλάβουν τον ρόλο που δημιουργήσαμε.

Για να αφαιρέσουμε ένα ρόλο από ένα χρήστη, χρησιμοποιούμε την εντολή REVOKE, όπως βλέπουμε παρακάτω:

```
--Syntax: REVOKE <ρόλος> FROM <χρήστης>;
```

```
SQL> REVOKE employee FROM lefteris;
```

Αφαιρέσαμε λοιπόν το ρόλο από τον χρήστη, στην προκειμένη τον ρόλο employee από το χρήστη lefteris.

5.1.2.1 - Απόδοση και αφαίρεση πολλαπλών ρόλων

Φυσικά μπορούμε, αφού έχουμε δημιουργήσει πολλαπλούς ρόλους, να τους εκχωρήσουμε στον χρήστη με μία κίνηση:

```
--Syntax: GRANT <ρόλος-1>, <ρόλος-2>, <ρόλος-n> TO <χρήστης>
```

```
SQL> GRANT employee, operator TO lefteris;
```

Αποδώσαμε λοιπόν πολλαπλούς ρόλους στον χρήστη. Με την ίδια τακτική μπορούμε να τους αφαιρέσουμε:

```
--Syntax: REVOKE <ρόλος-1>, <ρόλος-2>, <ρόλος-n> FROM <χρήστης>;
```

```
SQL> REVOKE employee, operator FROM lefteris;
```

Με μία κίνηση λοιπόν αφαιρέσαμε πολλαπλούς ρόλους από έναν χρήστη. Αυτή η τακτική προσφέρει μεγάλη ευελιξία και μας γλυτώνει από πολλές γραμμές εντολών.

5.1.2.2 - Συνδυασμός Ρόλων και Μεμονωμένων Δικαιωμάτων

Ας υποθέσουμε το εξής σενάριο. Έχουμε τον ρόλο employee ο οποίος εμπεριέχει τα δικαιώματα CREATE TABLE, CREATE VIEW και CREATE SESSION. Μεταξύ όλων των εργαζομένων, έτσι και ο χρήστης lefteris του έχει αποδοθεί ο ρόλος employee. Ο χρήστης lefteris όμως προβιβάζεται σε προϊστάμενο στην εταιρία που εργάζεται. Ως εκ τούτου, μπορεί να έχει περισσότερα δικαιώματα πάνω στην βάση, όπως για παράδειγμα να βλέπει όλους τους πίνακες, όλων των εργαζομένων και να δημιουργεί procedures. Ως εκ τούτου πρέπει να αναβαθμιστούν τα δικαιώματά του, χωρίς όμως να αλλάξει η δομή του ρόλου employee. Μπορεί όμως να γίνει αυτό; Φυσικά και μπορεί.

Μπορούμε να δώσουμε σε ένα συγκεκριμένο χρήστη, περισσότερα δικαιώματα από ότι του προσφέρει ένας ρόλος. Μπορούμε δηλαδή να συνδυάσουμε την απόδοση μεμονωμένων δικαιωμάτων με την απόδοση ρόλων. Παρακάτω βλέπουμε πως:

--Syntax: GRANT <ρόλος>, <n-δικαιώματα> TO <χρήστης>;

```
SQL> GRANT employee, SELECT ANY TABLE, CREATE PROCEDURE TO
lefteris;
```

Με το ίδιο σκεπτικό μπορούμε με την REVOKE να αφαιρέσουμε μεμονωμένα δικαιώματα που δώσαμε με την GRANT. **ΠΡΟΣΟΧΗ** όμως, δεν μπορούμε να αφαιρέσουμε ένα από τα δικαιώματα που παρέχει ο ρόλος, από το χρήστη. Αν πχ ο ρόλος περιέχει το δικαίωμα CREATE PROCEDURE και ο ρόλος έχει εκχωρηθεί στο χρήστη lefteris, τότε πρέπει να το αφαιρέσουμε **από το ρόλο** και όχι από το χρήστη (διαφορετικά μεταφράζεται σε σφάλμα).

Μπορούμε επίσης να αφαιρέσουμε μεμονωμένα δικαιώματα και ρόλο με μία κίνηση όπως θα δούμε παρακάτω:

--Syntax: REVOKE <ρόλος>, <n-δικαιώματα> FROM <χρήστης>

```
SQL> REVOKE employee, CREATE PROCEDURE FROM lefteris;
```

Όπως είδαμε, τα δικαιώματα που εκχωρήσαμε πριν με την GRANT, τα αφαιρούμε μαζί με τον ρόλο με μία κίνηση.

5.1.2.2 - Απόδοση ρόλου σε άλλους ρόλους

Η Oracle μας επιτρέπει να διαβιβάζουμε τα δικαιώματα ενός ρόλου, σε έναν νέο ρόλο. Αντί δηλαδή να δίνουμε τα ίδια δικαιώματα ενός ρόλου σε έναν άλλον μεμονωμένα, εκχωρούμε ολόκληρο τον ρόλο στον καινούριο ρόλο.

Για παράδειγμα, έχουμε τον ρόλο employee με τα δικαιώματα CREATE SESSION, CREATE TABLE, CREATE VIEW και δημιουργούμε τον ρόλο senior_employee όπου θέλουμε να έχει τα ίδια δικαιώματα (και περισσότερα). Αυτό λοιπόν επιτυγχάνεται όπως παρακάτω

--Syntax: GRANT <υπάρχον ρόλος> TO <νέος ρόλος>

```
SQL> CREATE ROLE senior_employee;
ROLE SENIOR_EMPLOYEE CREATED.
SQL> GRANT employee TO senior_employee;
GRANT SUCCEEDED.
SQL> GRANT senior_employee TO lefteris;
```

Από κει και πέρα, μπορούμε να προσθέσουμε πρόσθετα δικαιώματα στον νέο μας ρόλο ή μπορούμε να αφαιρέσουμε τον ρόλο ολόκληρο. Έτσι μπορούμε να πούμε ότι ο παλιός ο ρόλος που εκχωρήθηκε στον νέο, πλέον ονομάζεται **υπό-ρόλος** (sub-role). **ΠΡΟΣΟΧΗ** όμως, για να αφαιρέσουμε ένα υπό-ρόλο από έναν άλλο ρόλο πρέπει να το κάνουμε **από το ρόλο** και όχι από το χρήστη, δηλαδή:

--Syntax: REVOKE <υπό-ρόλος> FROM <ρόλος>

```
SQL> REVOKE employee FROM senior_employee;
```

Έτσι αφαιρέσαμε τον υπό-ρόλο από τον ρόλο. Πλέον τα δικαιώματα του υπό-ρόλου employee έχουν αφαιρεθεί από το ρόλο senior_employee και κατά συνέπεια από το χρήστη που έχει το ρόλο senior_employee. Τα δικαιώματα όμως που εκχωρήθηκαν χωρία στον ρόλο senior_employee **παραμένουν**.

5.1.2.3 - Διαγραφή Ρόλων

Οι ρόλοι που δεν χρειαζόμαστε πλέον, μπορούν να διαγραφούν με την εντολή DROP ROLE όπως βλέπουμε παρακάτω:

--Syntax: DROP ROLE <όνομα ρόλου>

```
SQL> DROP ROLE EMPLOYEE
```

Όταν ένας ρόλος διαγράφεται, τότε αυτόματα αφαιρείται και από όποιον χρήστη του έχει εκχωρηθεί. Χρειάζεται προσοχή η διαγραφή ρόλων γιατί μπορεί να προκληθεί πρόβλημα στους χρήστες.

5.1.2.4 - Έτοιμοι ρόλοι

Η Oracle παρέχει ένα σύνολο από έτοιμους ρόλους²⁴, τους οποίους μπορούμε να αποδώσουμε σε χρήστη, άμεσα. Οι πιο συχνά χρησιμοποιούμενοι εξ αυτών παρατίθενται στον παρακάτω πίνακα:

Όνομα Ρόλου	Δικαιώματα / Λειτουργία Ρόλου
DBA	Δίνει δικαιώματα διαχειριστή σε ένα χρήστη
SYSDBA	Πλήρη δικαιώματα διαχείρισης
SYSOPER	Βασικά δικαιώματα διαχείρισης χωρίς πρόσβαση στα δεδομένα του χρήστη
CONNECT	Επιτρέπει την δημιουργία συνεδρίας ενός χρήστη στην βάση
RESOURCE	Δίνει δικαιώματα χρήστη συστήματος σε έναν απλό χρήστη
APEX_ADMINISTRATOR_ROLE	Επιτρέπει σε ένα Schema να έχει δικαιώματα διαχείρισης στο Oracle APEX.

Υπάρχουν φυσικά περισσότεροι ρόλοι ωστόσο εμείς σας δείχνουμε τους πιο συχνά χρησιμοποιούμενους. Θα χρησιμοποιήσουμε κάποιους από αυτούς σε παρακάτω ενότητες.

²⁴ Oracle Documentation → Managing Users and Security → User Privileges and Roles → Oracle Database Express Edition Predefined Roles – Site
(http://docs.oracle.com/cd/E17781_01/server.112/e18804/users_secure.htm#autold4)

Keep in mind...

Μονάχα οι χρήστες με ρόλο SYSDBA και SYSOPER έχουν δικαιώματα πρόσβασης στις λειτουργίες της βάσης, ακόμη και αν είναι ανενεργή (Idle Instance).

Καλή πρακτική

Καταρχήν θα πρέπει να διαχωρίζετε τους χρήστες ως προς τον ρόλο τους στο σύστημα. Τους απλούς χρήστες που περιμένουμε να δημιουργούν πράγματα στην βάση, θα πρέπει να τους δώσουμε τον κατάλληλο ρόλο γι αυτό το ζητούμενο. Τους χρήστες με περισσότερες ευθύνες και έργο πάνω στην βάση τους δίνετε περισσότερα δικαιώματα. Οπότε καλή πρακτική είναι να δημιουργήσετε ένα ρόλο που να παρέχει τα στοιχειώδη δικαιώματα για τους απλούς χρήστες. Εμείς προτείνουμε τον ρόλο **PLAIN** για τους απλούς χρήστες, με δικαιώματα CREATE SESSION, CREATE TYPE, CREATE TABLE, CREATE PROCEDURE, CREATE VIEW, CREATE TRIGGER. Για τους προνομιούχους χρήστες, επιλέγουμε το ρόλο **PRIVILEGED** όπου συν τα δικαιώματα του PLAIN, να έχουν και δικαιώματα DROP ANY TABLE, DROP ANY PROCEDURE, DROP ANY VIEW, EXECUTE ANY PROCEDURE, SELECT ANY TABLE, INSERT ANY TABLE, DELETE ANY TABLE, UPDATE ANY TABLE.

Προσέχουμε πάντα τα δικαιώματα που δίνουμε να μην απειλούνε την ακεραιότητα και την ορθή λειτουργία της βάσης.

5.1.3 - Προβολή ρόλων και Δικαιωμάτων

Μπορούμε φυσικά αν το θέλουμε, να δούμε ένας χρήστης τι είδους δικαιώματα και ρόλοι του έχουν αποδοθεί, μέσα από τα data dictionary (πίνακες με metadata) της βάσης μας.

Για την προβολή δικαιωμάτων ενός συγκεκριμένου χρήστη γράφουμε το παρακάτω επερώτημα :

```
--Syntax: SELECT * FROM DBA_SYS_PRIVS WHERE GRANTEE LIKE
--'χρήστης%';
```

```
SQL> SELECT * FROM DBA_SYS_PRIVS WHERE GRANTEE LIKE 'LEF%';
```

GRANTEE	PRIVILEGE	ADMIN_OPTION
LEFTERIS	CREATE TRIGGER	NO
LEFTERIS	CREATE SESSION	NO
LEFTERIS	SELECT ANY TABLE	NO

Για την προβολή των ρόλων που έχουν αποδοθεί σε ένα χρήστη, γράφουμε το παρακάτω επερώτημα:

```
--Syntax: SELECT * FROM DBA_ROLE_PRIVS WHERE GRANTEE LIKE
--'χρήστης%';
```

```
SQL> SELECT * FROM DBA_ROLE_PRIVS WHERE GRANTEE LIKE 'LEF%';
```

GRANTEE	GRANTED_ROLE	ADMIN_OPTION	DEFAULT_ROLE
LEFTERIS	S_EMPLOYEE	NO	YES

Τέλος, για να δούμε όλους τους διαθέσιμους ρόλους (εκχωρημένους και μη) που υπάρχουν στην βάση στέλνουμε το εξής επερώτημα:

```
SQL> SELECT * FROM DBA_ROLES;
```

ROLE	PASSWORD	AUTHENTICAT
CONNECT	NO	NONE
RESOURCE	NO	NONE

....

<output omitted>

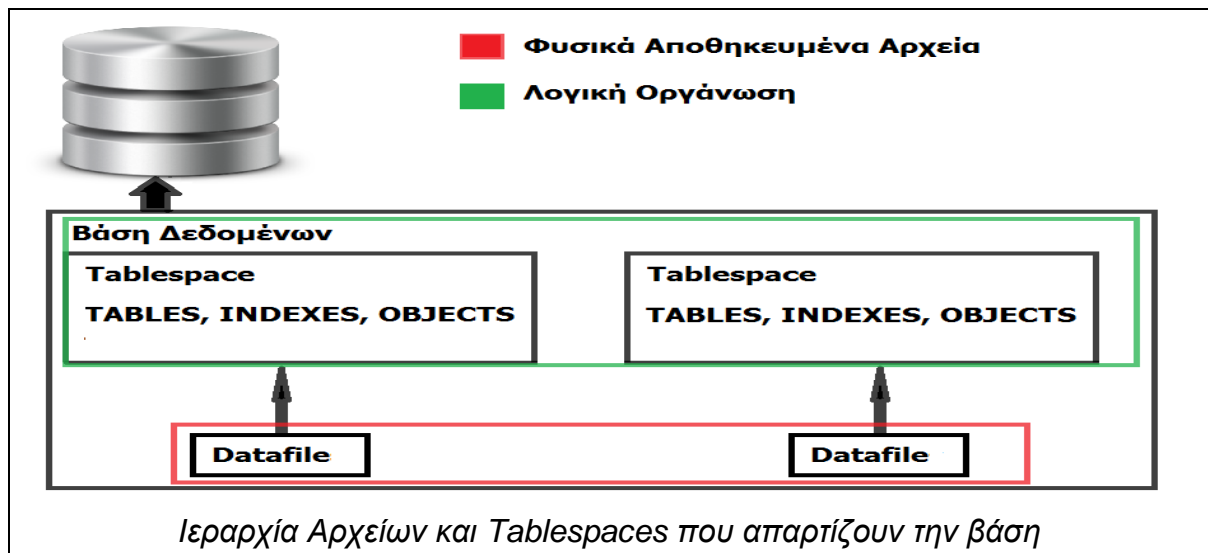
5.1.4 - Tablespaces

Μία βάση δεδομένων, αποτελείται μεταξύ άλλων, από ένα σύνολο αρχείων-αποθήκες που ονομάζονται **Αρχεία Δεδομένων** (datafiles), μέσα στα οποία φιλοξενούνται τα δεδομένα του εκάστοτε χρήστη της βάσης. Σε αυτά τα αρχεία, υπάρχει η έννοια του Tablespace.

Ορισμός

Ένας **πινακοχώρος**²⁵ (Tablespace) είναι μια λογική οργάνωση της βάσης δεδομένων, στην οποία αποθηκεύονται τα δεδομένα της βάσης, στην ουσία δηλαδή είναι η **ίδια η βάση δεδομένων**. Αυτά που αποθηκεύουν μόνιμα τα δεδομένα του χρήστη, αναφέρονται και ως **μόνιμα Tablespaces** (permanent Tablespaces).

Ο πινακοχώρος βρίσκεται μέσα στα αρχεία δεδομένων (datafiles) που απαρτίζουν την βάση δεδομένων. Παρακάτω βλέπουμε ένα κατατοπιστικό σχήμα:



Στο παραπάνω σχήμα βλέπουμε από τι αποτελείται μια βάση δεδομένων. Τα Datafiles περιέχουν τα Tablespaces και όλα μαζί απαρτίζουν την βάση δεδομένων.

Στα Tablespaces λοιπόν, οι χρήστες μπορούν να αποθηκεύουν τα δεδομένα τους. Μπορούμε, με την αφηρημένη έννοια του όρου, να πούμε πως τα

²⁵ Wikipedia Article → Tablespace → “A Tablespace is a storage location where the actual data underlying database objects can be kept” – Site (<http://en.wikipedia.org/wiki/Tablespace>)

Tablespaces είναι τα «ντουλαπάκια» του χρήστη, όπου βάζει τα πράγματά του. Ωραία όλα αυτά, αλλά σε τι χρησιμεύουν;

Ο χρήστης πρέπει υποχρεωτικά να συσχετιστούν με ένα Tablespace καθώς διαφορετικά δεν θα μπορούν να δημιουργήσουν τίποτα στην βάση δεδομένων.

Από προεπιλογή, όποτε δημιουργείται ένας νέος χρήστης, συσχετίζεται στο default Tablespace με το όνομα **USERS**, αλλά σε αυτό δεν μπορεί να δημιουργήσει τίποτα αν δεν του **εκχωρηθεί χώρος** σε αυτό. Φυσικά αν ο χρήστης δεν σκοπεύει να δημιουργήσει τίποτα, δεν υπάρχει θέμα, αλλά αν προσπαθήσει να δημιουργήσει κάτι τότε υπάρχει πρόβλημα.

Η Oracle από προεπιλογή, στην δημιουργία κάθε βάσης δημιουργεί το Tablespace USERS, το οποίο αποτελεί το Default Tablespace για απόθεση αντικειμένων των χρηστών. Έτσι λοιπόν όταν ο χρήστης προσπαθήσει να δημιουργήσει κάτι, το μόνο Tablespace διαθέσιμο για να τον εξυπηρετήσει είναι το Default Tablespace του, το USERS. Ο χρήστης μπορεί να έχει συσχετιστεί με αυτό το Tablespace, αλλά δεν του αποδόθηκε χώρος σε αυτό, οπότε το σύστημα θα τον αποτρέψει. Και φυσικά ο χρήστης δεν θα δημιουργήσει ποτέ δικά του αντικείμενα. Κάτι πρέπει να γίνει γι αυτό όμως.

Προσοχή!

Σαν διαχειριστές (SYS,SYSTEM,SYSTEMMAN κλπ) έχετε ως Default Tablespace το SYSTEM (και εφεδρικό το SYSAUX). Αποφεύγετε ωστόσο να δημιουργείτε αντικείμενα σε αυτό γιατί σε αυτό αποθηκεύονται οι πίνακες του Data Dictionary (metadata) και κινδυνεύετε να αλλοιώσετε την δομή της βάσης²⁶.

5.1.4.1 - Συσχέτιση χρήστη με σε Tablespace και εκχώρηση χώρου

²⁶ Oracle Documentation → 2 Managing Security for Oracle Database Users → Assigning a Default Tablespace for the User → “By separating the user data from the system data, you reduce the likelihood of problems with the SYSTEM tablespace, which can in some circumstances cause the entire database to become nonfunctional” – Site (http://docs.oracle.com/cd/B28359_01/network.111/b28531/users.htm#DBSEG10200)

Αν και είπαμε ότι ο κάθε χρήστης ανήκει από προεπιλογή στο Tablespace USERS, αλλά υποθετικά για λόγους επίδειξης θα πούμε ότι ο χρήστης LEFTERIS δεν ανήκει σε κάποιο. Θα τον συσχετίσουμε λοιπόν με το Tablespace USERS, με την εντολή ALTER USER και DEFAULT TABLESPACE όπως παρακάτω:

```
--Syntax: ALTER USER <όνομα χρήστη> DEFAULT TABLESPACE <tablespace>  
SQL> ALTER USER lefteris DEFAULT TABLESPACE USERS;
```

Τώρα αλλάξαμε το Default Tablespace του χρήστη lefteris το οποίο πλέον είναι το USERS. Τώρα ο χρήστης είναι οκ; Μάλλον όχι...

Το να παραχωρήσουμε ένα Tablespace σε ένα χρήστη δεν φτάνει. Θα πρέπει να επιλέξουμε και την ποσότητα των αποθηκευτικών πόρων του Tablespace που θα δικαιούται ο χρήστης. Από προεπιλογή, κανένας χρήστης δεν δικαιούται αποθηκευτικό χώρο στο Tablespace USERS, οπότε πρέπει να του δώσουμε μια ποσότητα (η και όλη). Η εντολή με την οποία εκχωρούμε αποθηκευτικό χώρο ενός Tablespace σε ένα χρήστη, είναι η εντολή ALTER USER με την ρήτρα (Clause) **QUOTA**.

```
--Syntax: ALTER USER <χρήστης> QUOTA <αριθμός σε mb> M ON  
--<tablespace>;  
SQL> ALTER USER lefteris QUOTA 100 M ON USERS;
```

Δώσαμε λοιπόν 100 MB χώρου από το Tablespace στον χρήστη, για να δουλέψει σε αυτό. Ο χρήστης είναι **έτοιμος** να δημιουργήσει αντικείμενα στην βάση δεδομένων. Και για του λόγου το αληθές, τα αντικείμενά του θα αποθηκεύονται στο USERS Tablespace.

Αν θέλουμε, μπορούμε να εκχωρήσουμε όλο το διαθέσιμο χώρο του Tablespace σε ένα χρήστη (όσος επιτρέπει το μέγεθος του Datafile που φιλοξενεί το Tablespace). Η QUOTA θα το αναλάβει:

```
--Syntax: ALTER USER <χρήστης> QUOTA UNLIMITED ON <tablespace>;  
SQL> ALTER USER lefteris QUOTA UNLIMITED ON <tablespace>;
```

Έτσι ο χρήστης πλέον έχει απεριόριστη πρόσβαση σε όλο τον αποθηκευτικό χώρο του Tablespace. Χρήσιμο αν ο χρήστης σκοπεύει να δημιουργήσει πολλά και απαιτητικά σε αποθηκευτικό χώρο, αντικείμενα.

Τέλος, μπορούμε να αποτρέψουμε τον χρήστη από το να χρησιμοποιεί ένα Tablespace, απλά εκχωρώντας του 0 MB αποθηκευτικό χώρο, με την βοήθεια της QUOTA.

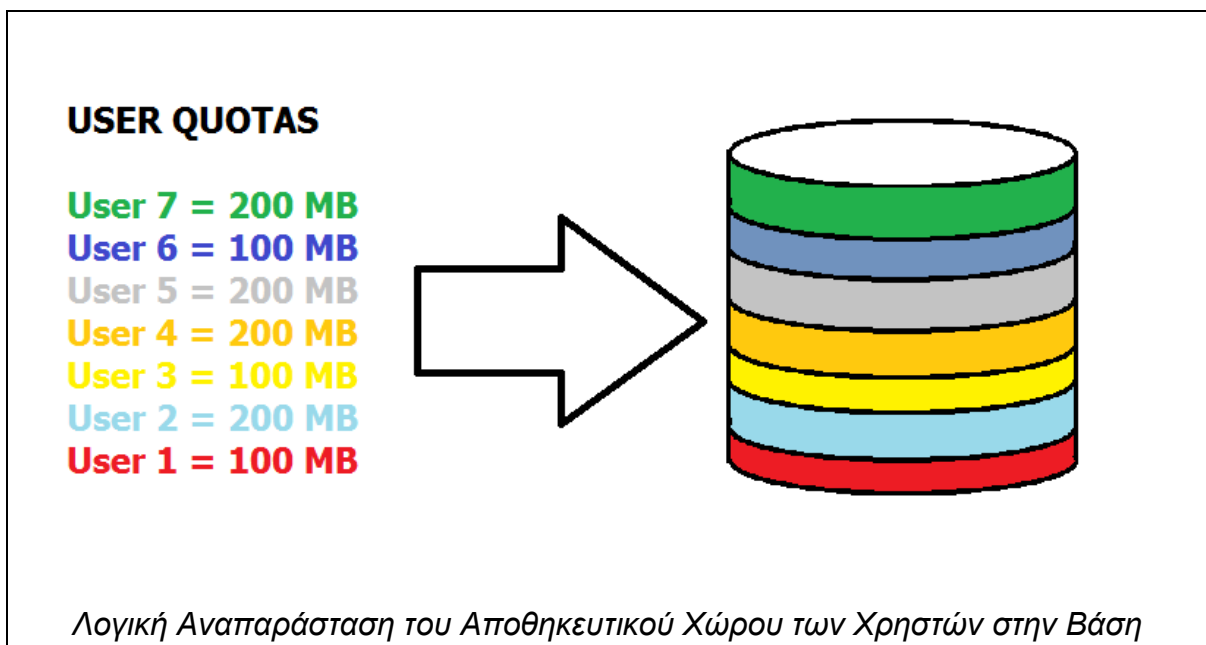
```
--Syntax: ALTER USER <χρήστης> QUOTA 0 M ON <tablespace>;
```

```
SQL> ALTER USER lefteris QUOTA 0 M ON USERS;
```

Έτσι αν ο χρήστης προσπαθήσει πχ, να εισάγει κάτι σε έναν πίνακα θα λάβει το παρακάτω σφάλμα:

```
ORA-01536: space quota exceeded for tablespace 'USERS'
```

Στην παρακάτω εικόνα βλέπουμε ένα λογικό σχήμα με τους χρήστες και τους χώρους που τους παραχωρήθηκαν στην βάση δεδομένων:



Με την βοήθεια του παραπάνω σχήματος, μπορούμε καλύτερα στο μυαλό μας να σχεδιάσουμε την έννοια της παραχώρησης χώρου στην βάση δεδομένων.

Καλή πρακτική

Δίνουμε στους χρήστες τόσο χώρο όσο πραγματικά χρειάζονται. Αν προσδοκάται οι χρήστες να δεσμεύουν αρκετό χώρο στην βάση, φροντίστε να τους δίνετε επαρκή χώρο. Αν όμως ισχύει το αντίθετο, να τους παραχωρείτε λίγο ώστε να μην εξαντλείτε άσκοπα τον διαθέσιμο χώρο της.

5.1.4.2 – Δημιουργία Νέου Tablespace

Κάποιος θα μπορούσε να πει «Γιατί να πρέπει οι χρήστες να χρησιμοποιούν τα Tablespace που παρέχει το σύστημα, και να μην μπορώ να δημιουργήσω κι εγώ δικά μου και να τους τα παραχωρήσω;». Εύλογη ερώτηση. Γίνεται κάτι τέτοιο; Εννοείται!

Η Oracle επιτρέπει την δημιουργία Tablespaces, ότι μεγέθους θέλουμε και όσα θέλουμε (για την ακρίβεια όσο μας επιτρέπει η χωρητικότητα του δίσκου). Τα αντίστοιχα datafiles που θα φιλοξενούν τα Tablespaces, δημιουργούνται αυτομάτως.

Ας πάρουμε για παράδειγμα, ότι θέλουμε οι φοιτητές ενός πανεπιστημίου που παρακολουθούν το μάθημα «Συστήματα Διαχείρισης Βάσεων Δεδομένων», να εργάζονται στο Tablespace **STUDENTS**. Το Tablespace αυτό, θα χρησιμοποιείται αποκλειστικά και μόνον για τους φοιτητές και για κανέναν άλλον. Κάθε φοιτητής για να περατώσει όλες τις ασκήσεις του εργαστηρίου δεν χρειάζεται χώρο πάνω από 50 MB οπότε τόσος θα είναι και ο χώρος που θα δικαιούνται σε αυτό. Ας δημιουργήσουμε το Tablespace, όπως παρακάτω:

```
--Syntax: CREATE TABLESPACE <όνομα> DATAFILE 'path στον
--oradata/βάση/όνομα_αρχείου.dbf' SIZE <μέγεθος σε MB> M EXTENT LOCAL
--AUTOALLOCATE;

SQL>          CREATE          TABLESPACE          STUDENTS          DATAFILE
'$ORACLE_BASE/oradata/test/stud01.dbf' SIZE 500 M
EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
```

Όπως είδαμε, δημιουργήσαμε με την **CREATE TABLESPACE** ένα νέο Tablespace, ονόματι STUDENTS. Το **DATAFILE**, δηλώνει το αρχείο datafile που θα φιλοξενήσει το Tablespace, και θα βρίσκεται στον κατάλογο ORACLE_BASE/oradata/<SID>, στον υποκατάλογο που αντιστοιχεί στην βάση δεδομένων μας. Επιλέξαμε το μέγεθός του να είναι 500 MB, αλλά το πιο ενδιαφέρον είναι το **EXTENT LOCAL AUTOALLOCATE** κομμάτι της εντολής. Το EXTENT LOCAL που είναι η Default επιλογή όταν δημιουργείται ένα νέο Tablespace, δηλώνει πως το αρχείο περιέχει **Μόνιμο** (permanent) Tablespace που θα διαχειρίζεται τοπικά στον Server. Η λέξη-κλειδί AUTOALLOCATE δηλώνει πως το Tablespace θα το διαχειρίζεται αυτόματα το λειτουργικό σύστημα, αναλόγως τους πόρους που απαιτεί κάθε φορά.

Τώρα δημιουργούμε έναν χρήστη, του δίνουμε τον ρόλο STUDENT που υποτίθεται έχουμε δημιουργήσει από πριν και τέλος του εκχωρούμε 50 MB από το νέο μας Tablespace με τις παρακάτω εντολές:

```
SQL> CREATE USER STUDENT_1 IDENTIFIED BY secret_1;
USER STUDENT_1 CREATED.
SQL> GRANT STUDENT TO STUDENT_1;
GRANT SUCCEEDED.
SQL> ALTER USER STUDENT_1 QUOTA 20 M ON USERS;
USER ALTERED.
SQL> ALTER USER STUDENT_1 DEFAULT TABLESPACE STUDENTS
QUOTA 50 M ON STUDENTS;
USER ALTERED.
```

Έτσι λοιπόν, ο φοιτητής, μπορεί να συνδεθεί στην βάση και να δημιουργήσει τα αντικείμενά του, τα οποία θα αποθηκεύονται στο Tablespace STUDENTS. Επίσης μπορούμε να δημιουργήσουμε πίνακες στο συγκεκριμένο Tablespace χωρίς να είναι το Default μας, με την εντολή:

```
Syntax: CREATE TABLE <όνομα πίνακα> (<τύποι στηλών + περιορισμοί>)
TABLESPACE <όνομα tablespace> ;
```

```
SQL> CREATE TABLE SPEEDBOATS (ID INTEGER NOT NULL PRIMARY KEY,  
NAME VARCHAR(30)) TABLESPACE STUDENTS;
```

Keep in mind...

Τα Tablespace είναι αρχεία τα οποία δεσμεύουν χώρο στο δίσκο. Φροντίζουμε να μην δημιουργούμε άσκοπα Tablespace χωρίς να υπάρχει λόγος, διαφορετικά δεσμεύουμε άσκοπα τον διαθέσιμο αποθηκευτικό χώρο, τον οποίο ίσως τον χρειαστούμε αργότερα για ποιο σημαντικές εργασίες.

Προσοχή!

Ο χρήστης θα πρέπει υποχρεωτικά να έχει έστω και ένα μικρό αριθμό εκχωρημένου χώρου στο Tablespace USERS, διαφορετικά το σύστημα δεν θα τον αφήσει να δημιουργήσει/τροποποιήσει αντικείμενα ακόμα και αν συσχετιστεί με άλλο Default Tablespace.

Το γεγονός λοιπόν ότι μπορούμε να δημιουργήσουμε δικά μας Tablespace και να παραχωρούμε στους χρήστες τον αποθηκευτικό χώρο που εμείς θέλουμε παρέχει μια ποιο οργανωμένη δομή στην βάση μας. Ανάλογα με την ιδιότητα και τον ρόλο του κάθε χρήστη μπορούμε να έχουμε το αντίστοιχο Tablespace που ανταποκρίνεται στο ρόλο αυτό.

Καλή πρακτική

Αποτελεί γενικά καλή πρακτική να δίνετε στο αρχείο που θα φιλοξενεί το Tablespace το ίδιο όνομα με αυτό, ώστε να γνωρίζετε ποιο αρχείο το περιέχει. Επίσης καλή πρακτική είναι να χρησιμοποιείτε τόσα Tablespace όσα ανταποκρίνονται στις ιδιότητες και τους ρόλους των χρηστών του συστήματός σας. Μπορείτε για παράδειγμα να έχετε ξέχωρο Tablespace για τους εργαζόμενους, ξέχωρο για τους Επόπτες και ξέχωρο για τους Προϊστάμενους.

5.1.4.3 – Τροποποίηση Tablespaces

Μπορείτε να τροποποιήσετε ένα Tablespace όσον αφορά ορισμένα χαρακτηριστικά του. Οι πιο σημαντικοί λόγοι για τους οποίους χρειάζεται να τροποποιήσετε ένα Tablespace είναι οι ακόλουθοι:

- 1) Μετονομασία του Tablespace.
- 2) Ενεργοποίηση ή απενεργοποίηση του Tablespace.
- 3) Ενίσχυση του με περισσότερα datafiles.
- 4) Αλλαγή Μεγέθους Tablespace

5.1.4.3.1 – Μετονομασία Tablespace

Ας δούμε τώρα ακριβώς τι εννοούμε. Ξεκινάμε από την **μετονομασία** του Tablespace. Ας υποθέσουμε ότι θέλουμε να μετονομάσουμε το Tablespace STUDENTS σε PREGRADUATES, για το λόγο ότι μελλοντικά θα δημιουργήσουμε ένα Tablespace για τους μεταπτυχιακούς φοιτητές. Το ερώτημα τώρα εύλογα, είναι πως γίνεται αυτό.

Πριν όμως προσπαθήσουμε να μετονομάσουμε το Tablespace, ας στείλουμε ένα επερώτημα στην βάση, για να δούμε τα υπάρχοντα Tablespaces.

```
SQL> SELECT * FROM V$TABLESPACE
```

Το Query, θα επιστρέψει τα παρακάτω αποτελέσματα, από το Data Dictionary.


```
SQL> select * from V$TABLESPACE;
```

TS#	NAME	INC	BIG	FLA	ENC
0	SYSTEM	YES	NO	YES	
1	SYSAUX	YES	NO	YES	
2	UNDOTBS1	YES	NO	YES	
4	USERS	YES	NO	YES	
3	TEMP	NO	NO	YES	
6	FLOW_1044911194773778	YES	NO	YES	
8	FLOW_1132425751886116	YES	NO	YES	
9	EMPLOYEES	YES	NO	YES	
10	STUDENTS	YES	NO	YES	

9 rows selected.

Διαθέσιμα Tablespaces της βάσης δεδομένων

Βλέπουμε λοιπόν, τα διαθέσιμα Tablespaces που περιέχει η βάση μας. Επί σκοπού μαρκάρουμε το STUDENTS για να το φέρουμε σε αντιπαράβολή με το πέρας της μετονομασίας.

Και τώρα επιστρέφουμε στο ερώτημα. Πως γίνεται η μετονομασία του Tablespace. Η απάντηση είναι απλή. Με το **RENAME TO** clause της εντολής **ALTER TABLESPACE** όπως βλέπουμε παρακάτω:

--Syntax: ALTER TABLESPACE <παλιό όνομα> RENAME TO <νέο όνομα>'

```
SQL> ALTER TABLESPACE STUDENTS RENAME TO PREGRADUATE;
```

Μετονομάσαμε λοιπόν το Tablespace STUDENTS σε PREGRADUATE. Έγινε όμως η μετονομασία; Ας ξαναστείλουμε ένα επερώτημα στην βάση:

```
SQL> SELECT * FROM V$TABLESPACE
```

Και το επερώτημα θα μας επιστρέψει τα ακόλουθα αποτελέσματα από το Data Dictionary.

```
SQL> select * from v$tablespace;
```

TS#	NAME	INC	BIG	FLA	ENC
0	SYSTEM	YES	NO	YES	
1	SYSAUX	YES	NO	YES	
2	UNDOTBS1	YES	NO	YES	
4	USERS	YES	NO	YES	
3	TEMP	NO	NO	YES	
6	FLOW_1044911194773778	YES	NO	YES	
8	FLOW_1132425751886116	YES	NO	YES	
9	EMPLOYEES	YES	NO	YES	
10	PREGRADUATE	YES	NO	YES	

9 rows selected.

Διαθέσιμα Tablespaces της βάσης μετά την μετονομασία

Όπως βλέπουμε η μετονομασία έγινε. Οι χρήστες όμως που το χρησιμοποιούν; Πρέπει να τους ξανά-συσχετίσουμε με το Tablespace; Η απάντηση είναι όχι, και μπορούμε να το διαπιστώσουμε στέλνοντας το παρακάτω Query στην βάση:

```
SQL> SELECT * FROM dba_ts_quotas WHERE TABLESPACE_NAME like 'PREGRAD%';
```

Το Query εκτελείται και μας επιστρέφει τα παρακάτω αποτελέσματα από τα Data Dictionaries.

TABLESPACE_NAME	MAX_BYTES	BLOCKS	USERNAME	MAX_BLOCKS	DRO	BYTES
PREGRADUATE	52428800	32	VITO	6400	NO	262144
PREGRADUATE	104857600	0	SCOTT	12800	NO	0
PREGRADUATE	52428800	0	LEFTERIS	6400	NO	0
PREGRADUATE	52428800	0	STUDENT_1	6400	NO	0

Χρήστες που χρησιμοποιούν των χώρο του Tablespace PREGRADUATE

Βλέπουμε πως οι χρήστες μας παραμένουν στο Tablespace, παρόλο που αυτό άλλαξε όνομα. (Σημείωση, ο χρήστης Lefteris του είχε ήδη εκχωρηθεί χώρος 50 MB πριν την μετονομασία).

Keep in mind...

Για να μετονομάσουμε ένα Tablespace, η βάση πρέπει να είναι ανοιχτή και mounted. Επίσης το Tablespace θα πρέπει να είναι Online (θα εξηγήσουμε παρακάτω τι σημαίνει).

5.1.4.3.2 - Αλλαγή κατάστασης διαθεσιμότητας

Τώρα θα ασχοληθούμε με την **διαθεσιμότητα** του Tablespace. Ένα Tablespace μπορεί να είναι είτε ενεργό (Online) είτε ανενεργό (Offline). Τι σημαίνει όμως αυτό;

Ενεργό (Online) Tablespace, ονομάζεται το Tablespace που είναι διαθέσιμο για χρήση. Όταν ο χρήστης δημιουργεί ή τροποποιεί αντικείμενα στην βάση δεδομένων, χρειάζεται ένα ενεργό Tablespace για αυτήν την εργασία. Εφόσον λοιπόν ο χρήστης μπορεί και τα κάνει τα παραπάνω, τότε αυτόματα σημαίνει ότι το Tablespace είναι διαθέσιμο, άρα ενεργό.

Ανενεργό (Offline) Tablespace, ονομάζεται το Tablespace που δεν είναι διαθέσιμο για χρήση. Όταν συμβαίνει αυτό, ο χρήστης δεν έχει πρόσβαση στα αντικείμενα που βρίσκονται αποθηκευμένα σε αυτό, άρα δεν μπορεί να δουλέψει σε αυτό.

Γιατί όμως ένα Tablespace να είναι ανενεργό; Μερικοί λόγοι δίνονται παρακάτω .

- 1) **Για λόγους συντήρησης του Server.** Ο διαχωριστής μπορεί να θέλει να μεταφέρει το Datafile του Tablespace σε άλλη τοποθεσία, οπότε πρέπει να το απενεργοποιήσει πρώτα.

- 2) **Για λόγους Backup.** Αν ο διαχωριστής θέλει να κάνει αντίγραφο ασφαλείας το Datafile που κρατάει το Tablespace, καλή πρακτική είναι να το απενεργοποιεί (δεν είναι όμως υποχρεωτικό).
- 3) **Για λόγους μετονομασίας.** Αν ο χρήστης θέλει να μετονομάσει το Datafile που κρατάει το Tablespace, το τελευταίο πρέπει να είναι ανενεργό.

Τι γίνεται όμως όταν ένα Tablespace είναι ανενεργό; Η βάση δεδομένων συνεχίζει να λειτουργεί; Φυσικά, η βάση δεδομένων συνεχίζει να λειτουργεί με τα ενεργά Tablespaces, απλά ένα τμήμα της είναι μη διαθέσιμο. Αν για παράδειγμα, απενεργοποιήσουμε το Tablespace PREGRADUATE, τότε μόνο οι φοιτητές δεν θα μπορούν να εργαστούν στην βάση. Όλοι οι υπόλοιποι (καθηγητές, μεταπτυχιακοί κλπ) που εργάζονται σε άλλα Tablespaces, μπορούν να εργαστούν κανονικά.

Για την εναλλαγή της κατάστασης διαθεσιμότητας ενός Tablespace, χρησιμοποιούμε τις λέξεις-κλειδιά ONLINE και OFFLINE όπως παρακάτω:

```
--Syntax: ALTER TABLESPACE <tablespace> ONLINE/OFFLINE  
--{<παράμετρος>;
```

Ας το δούμε στην πράξη. Θα απενεργοποιήσουμε το Tablespace PREGRADUATE.

```
SQL> ALTER TABLESPACE PREGRADUATE OFFLINE NORMAL;
```

Η παραπάνω εντολή απενεργοποιεί το Tablespace PREGRADUATE. Προσέξτε την παράμετρο NORMAL στο OFFLINE Clause. Οι παράμετροι που παίρνει το OFFLINE Clause είναι οι παρακάτω ²⁷.

²⁷ Oracle Documentation → Altering Tablespace Availability → Taking Tablespaces Offline → “ You can specify any of the following parameters as part of the ALTER TABLESPACE...OFFLINE statement:” – Site (http://docs.oracle.com/cd/B28359_01/server.111/b28310/tspaces005.htm#ADMIN11376)

Παράμετρος	Χρησιμότητα
NORMAL	Απενεργοποιεί το Tablespace κανονικά και κρατάει Checkpoint στα Control Files της βάσης (σημεία αναίρεσεως).
TEMPORARY	Απενεργοποιεί προσωρινά το Tablespace και κρατάει Checkpoint.
IMMEDIATE	Απενεργοποιεί άμεσα το Tablespace χωρίς να κρατάει Checkpoint. Στην ενεργοποίηση θα πρέπει να γίνει Recovery, οπότε θα αργήσει.

Σημειώνεται ότι οι παράμετροι είναι προαιρετικοί, και αν δεν ορίσετε καμία, τότε από προεπιλογή, ορίζεται η NORMAL²⁸.

Τι γίνεται σε περίπτωση που κάποιος χρήστης, ας πούμε ο student_1 προσπαθήσει να ανακτήσει έναν πίνακα της βάσης, ας τον πούμε BOATS, ο οποίος είναι αποθηκευμένος στο ανενεργό Tablespace PREGRADUATE; Ας δούμε:

```
SQL> select * from boats;
select * from boats
*
ERROR at line 1:
ORA-00376: file 8 cannot be read at this time
ORA-01110: data file 8: '/u01/app/oracle/oradata/test/stud.dbf'
```

Επερώτημα επιστροφής πίνακα σε ανενεργό Tablespace

Για να επαναφέρουμε λοιπόν ένα Tablespace σε ενεργή κατάσταση, γράφουμε:

```
SQL> ALTER TABLESPACE PREGRADUATE ONLINE;
```

Από κει και πέρα πλέον, το Tablespace PREGRADUATE είναι και πάλι διαθέσιμο για χρήση.

5.1.4.3.3 – Ενίσχυση Tablespace με περισσότερα Datafiles

²⁸ Oracle Documentation → Altering Tablespace Availability → Taking Tablespaces Offline → “NORMAL is the default” – Site
(http://docs.oracle.com/cd/B28359_01/server.111/b28310/tspaces005.htm#ADMIN11376)

Ας υποθέσουμε το εξής σενάριο. Το Tablespace που εργαζόμαστε έχει αρχίσει να γεμίζει, οπότε σε λίγο δεν θα έχει επιπλέον χώρο για την δημιουργία νέων αντικειμένων. Εμείς όμως αυτό θέλουμε να το αποτρέψουμε. Ευτυχώς όμως η Oracle δίνει την δυνατότητα να έχουμε περισσότερα του ενός datafiles στο ίδιο Tablespace. Έτσι όταν ένα Datafile γεμίσει, χρησιμοποιείται ένα άλλο.

Θα ασχοληθούμε πάλι με το Tablespace PREGRADUATE. Το αρχείο που το φιλοξενεί, είναι το «/u01/app/oracle/oradata/test/stud.dbf». Εμείς θέλουμε να ενισχύσουμε το Tablespace με περισσότερα του ενός αρχεία, στον ίδιο κατάλογο που βρίσκονται τα υπόλοιπα. Δεν έχουμε παρά να το πράξουμε.

Αρχικά θα πρέπει να ενεργοποιήσουμε το Tablespace. Το κάνουμε με την εντολή:

```
SQL> ALTER TABLESPACE PREGRADUATE ONLINE;
```

Στην συνέχεια δημιουργούμε το νέο Datafile στον ίδιο κατάλογο που βρίσκεται το stud.dbf (ORACLE_BASE/oradata/<SID>), μεγέθους 50 MB.

```
SQL> ALTER TABLESPACE PREGRADUATE ADD DATAFILE  
'$ORACLE_BASE/oradata/test/stud2.dbf' SIZE 50 M AUTOEXTEND ON MAX  
SIZE 100 K;
```

Έτσι πλέον το Tablespace PREGRADUATE έχει επιπλέον 50 MB αποθηκευτικού χώρου, ο οποίος θα αρχίσει να χρησιμοποιείται όταν εξαντληθεί ο χώρος του stud.dbf. Επίσης αν χρειαστεί θα κάνει αυτόματα Resize (Auto Extend).

Καλή πρακτική

Φροντίστε να δημιουργείτε πάντα ένα δεύτερο αρχείο Datafile για τα Tablespace σας, χωρητικότητας περίπου της μισής ή μεγαλύτερης από το κύριο αρχείο. Έτσι σε περίπτωση που το πρώτο αρχείο γεμίσει, αυτόματα το Tablespace θα «αποταθεί» στο δεύτερο. Φροντίστε να το κάνετε από πρώτο χέρι για να μην χρειαστεί αργότερα.

Με την ίδια λογική μπορείτε να διαγράψετε ένα Datafile από ένα Tablespace, για όποιον λόγο εσείς κρίνετε απαραίτητο με την εντολή:

```
SQL> ALTER TABLESPACE PREGRADUATE DROP DATAFILE
'$ORACLE_BASE/oradata/test/stud2.dbf'
```

5.1.4.3.4 – Αλλαγή Μεγέθους Tablespace (Resize)

Πολλές φορές θέλουμε να αλλάξουμε το μέγεθος ενός Tablespace (για την ακρίβεια του Datafile) αντί να προσθέτουμε περισσότερα datafiles στην βάση. Η Oracle επιτρέπει την αλλαγή μεγέθους του Datafile (και κατά συνέπεια του Tablespace) φροντίζοντας έτσι για την αποτελεσματική εκχώρηση περισσότερου χώρου στην βάση δεδομένων μας.

Keep in mind...

Το μόνιμο Tablespace δεν μπορεί να υποστεί Resize με άμεσο τρόπο. Αλλάζοντας το μέγεθος του Datafile ωστόσο κάνουμε έμμεσο Resize στο Tablespace.

Η αλλαγή του μεγέθους ενός Datafile είναι εύκολη υπόθεση. Ας ασχοληθούμε με το Tablespace PREGRADUATE, θα μεγαλώσουμε το Datafile του στα 600 MB. Η εντολή που χρειαζόμαστε είναι η κάτωθι:

```
--Syntax ALTER DATABASE DATAFILE <path προς datafile> RESIZE <νέο
μέγεθος> M;
```

```
SQL> ALTER DATABASE DATAFILE '$ORACLE_BASE/oradata/test/stud01.dbf'
RESIZE 600 M;
```

Διαπιστώνουμε τις αλλαγές μας, μέσα από την Κονσόλα Διαχείρισης Oracle Enterprise Manager (θα μιλήσουμε σε παρακάτω ενότητα γι αυτό) όπως παρακάτω:

FLOW_1044614628194104	50.1	2.5		5.0	NO	47.6	✓	1 PERMANENT	LOCAL	AUTO
PREGRADUATE	600.0	1.0		0.2	NO	599.0	✓	1 PERMANENT	LOCAL	AUTO
SYSAUX	530.0	504.6		95.2	YES	25.4	✓	1 PERMANENT	LOCAL	AUTO

Επιβεβαίωση Αλλαγής μεγέθους του Tablespace

Έτσι λοιπόν μπορούμε να αλλάζουμε το μέγεθος των Tablespace μας αποδοτικά, όποτε χρειαζόμαστε περισσότερο χώρο.

5.1.4.4 – Temporary Tablespaces

Η βάση δεδομένων δεν αποτελείται μόνο από τα μόνιμα Tablespaces. Περιέχει και τα λεγόμενα **προσωρινά Tablespaces** (temporary Tablespaces) προκειμένου να λειτουργεί αποδοτικά. Τι είναι όμως ένα προσωρινό Tablespace;

Ορισμός

Προσωρινό Tablespace, ονομάζεται ένα Tablespace που αποθηκεύει δυναμικά αντικείμενα, για όσο διαρκεί η συνεδρία σύνδεσης του χρήστη με την βάση. Με το πέρας της συνεδρίας, το Tablespace αδειάζει.

Το προσωρινό Tablespace λοιπόν δεν αποθηκεύει μόνιμα τα δεδομένα όπως κάνει ένα κανονικό Tablespace αλλά προσωρινά. Παράδειγμα αποτελεί η χρήση του πίνακα DUAL στην βάση δεδομένων, όπου δημιουργείται ένας δυναμικός πίνακας σε κάθε SELECT Query που κάνει ο χρήστης. Αυτός ο πίνακας δεν είναι μόνιμος στην βάση, οπότε αποθηκεύεται προσωρινά σε ένα Temporary Tablespace.

Είναι χρήσιμο γενικά να υπάρχει στην βάση μας προσωρινό Tablespace για λόγους απόδοσης, επειδή απαιτητικές εργασίες όπως ORDER BY Statements αντί να γίνονται στην μνήμη, γίνονται μέσα σε αυτά. Άρα εύκολα μπορεί κάποιος να συμπεράνει πως αποτελούν απαραίτητο κομμάτι της βάσης.

Μπορούμε να δημιουργήσουμε λοιπόν δικά μας Temporary Tablespaces σε αρχεία που θα τα φιλοξενούν, τα οποία ονομάζονται **Temp Files** (σε αντίθεση με τα datafiles). Η διαδικασία δημιουργίας και απόδοσης δεν διαφέρει πολύ από αυτά που ήδη ξέρουμε, αλλά αξίζει να σημειωθούν. Θα δημιουργήσουμε ένα νέο Temporary Tablespace για την βάση TEST μεγέθους 20 MB με αυτόματο Resize. Δημιουργούμε το νέο Tablespace με τις κάτωθι εντολές:


```
--Syntax: CREATE TEMPORARY TABLESPACE <όνομα> TEMPFILE <path με  
αρχείο> SIZE <μέγεθος σε KB ή MB> K|M AUTOEXTEND ON;
```

```
SQL > CREATE TEMPORARY TABLESPACE test_tmp TEMPFILE  
'$ORACLE_BASE/ORADATA/TEST/tmp.dbf' SIZE 20 M AUTOEXTEND ON;
```

Αφού δημιουργήσαμε το Tablespace μας, τώρα πρέπει να το ορίσουμε ως Temporary Tablespace για τους χρήστες μας. Θα πάρουμε τον χρήστη Lefteris ως παράδειγμα:

```
--Syntax: ALTER USER <χρήστης> TEMPORARY TABLESPACE <όνομα  
temporary tablespace>;
```

```
SQL> ALTER USER LEFTERIS TEMPORARY TABLESPACE test_tmp;
```

Έτσι λοιπόν ο χρήστης πλέον έχει ως Temporary Tablespace το TEST_TMP όπου μπορεί να αποθηκεύει τα προσωρινά του αντικείμενα.

Καλή πρακτική

Καλό είναι να έχετε Temporary Tablespaces στην βάση σας για να «αποδεσμεύετε» την μνήμη του Server από πρόσθετο φόρτο. Επίσης δεν είναι ανάγκη τα Temporary Tablespaces να είναι μεγάλα σε μέγεθος, καθώς γεμίζει άσκοπα ο δίσκος.

5.1.4.5 – Undo Tablespaces

Οι βάσεις δεδομένων της Oracle είναι **συναλλαγματικές** (transactional)²⁹, δηλαδή η οποιαδήποτε ενέργεια τροποποίησης του περιεχομένου της, θεωρείται συναλλαγή. Ως εκ τούτου για να εφαρμόσουμε τις όποιες αλλαγές κάνουμε **COMMIT** (εφαρμογή αλλαγών) και για να τις αναιρέσουμε κάνουμε **ROLLBACK** (αναίρεση αλλαγών) (θα μιλήσουμε σε προσεχή ενότητα για τις συναλλαγές).

²⁹ Oracle OTN Forums → Thread “transactional vs non-transactional database” → “Oracle RDBMS is a transactional database.” – Site (<https://forums.oracle.com/thread/2348650>)

Ας πάρουμε τώρα μαζί ένα παράδειγμα. Ας πούμε ότι στον πίνακα BOATS είχαμε πέντε γραμμές και εισάγαμε άλλες δύο, δηλαδή τώρα έχουμε σύνολο επτά γραμμές (δεν κάνουμε COMMIT). Εκτελώντας ένα Query στην βάση, βλέπουμε πως η γραμμές μας έχουν εισαχθεί.

```
SQL> insert into boats values (6,'Gorgoipikoos');
1 row created.
SQL> insert into boats values (7,'Sotiras');
1 row created.
SQL> select * from boats order by id;
```

INSERT

ID	NAME
1	Galena
2	Eliza
3	Marilena
4	Theoskepasti
5	Megalochari
6	Gorgoipikoos
7	Sotiras

7 rows selected.

```
SQL>
```

Query

Uncommitted Insert και Query Επιβεβαίωσης

Οι αλλαγές φαίνεται να γίνανε. Φαίνεται όμως. Δεν έχουν ακόμα εφαρμοστεί στην πραγματικότητα, άσχετο που εμείς τις βλέπουμε. Αυτό συμβαίνει επειδή τα νέα δεδομένα μας διαβάζονται από ένα άλλο Tablespace, το λεγόμενο **UNDO Tablespace** το οποίο αποθηκεύει τις αλλαγές από συναλλαγές που δεν έχουν γίνει COMMIT (uncommitted). Αυτό πρακτικά σημαίνει ότι αν εμείς κάνουμε ROLLBACK οι αλλαγές αυτές θα αναιρεθούν και ο πίνακας θα επιστρέψει στην προηγούμενη κατάσταση που ήτανε πριν.

```
SQL> ROLLBACK;
Rollback complete.
```

ROLLBACK

```
SQL> select * from boats;

   ID NAME
-----
    1 Galena
    2 Eliza
    3 Marilena
    4 Theoskepasti
    5 Megalochari
```

New Query

Εκτέλεση Query μετά από ROLLBACK

Βλέπουμε πως οι γραμμές που εισάγαμε πριν πλέον δεν υπάρχουν. Αυτό έγινε γιατί αναίρεσαμε (Undo) τις αλλαγές που έγιναν από την συναλλαγή, αναίρωντας την συναλλαγή (ROLLBACK). Και πως ήξερε η βάση ποιες γραμμές εισάγαμε χωρίς να κάνουμε COMMIT για τις αφαιρέσει με την αναίρεση; Εδώ ευθύνονται τα UNDO Tablespaces.

Ορισμός

Tablespace Αναίρεσης (Undo Tablespace) ονομάζεται το Tablespace που κρατάει τις αλλαγές που προκλήθηκαν από μια συναλλαγή, αλλά δεν έχουν ακόμη γίνει COMMIT. Έτσι ο χρήστης αν εκτελέσει ROLLBACK, βάση αυτού του Tablespace οι αλλαγές αναίρούνται. Αν γίνει COMMIT τότε οι αλλαγές αυτές δεν υπάρχουν πλέον στο Undo Tablespace, οπότε δεν μπορεί να γίνει αναίρεση. Τα Undo Tablespace επίσης περιέχουν και Metadata με Snapshots των αλλαγών για χρήση της τεχνολογίας Oracle Flashback Queries (θα το δούμε στο κεφάλαιο 6).

Για να δημιουργήσουμε ένα Undo Tablespace, θα ακολουθήσουμε την πρακτική που ξέρουμε ήδη (αν και το σύστημα μας παρέχει ένα από προεπιλογή). Οι εντολή είναι η κάτωθι:

```
--Syntax: CREATE UNDO TABLESPACE <όνομα tablespace> DATAFILE <path
προς τον κατάλογο της βάσης + αρχείο> SIZE <μέγεθος> K|M REUSE
AUTOEXTEND ON;
```

```
SQL> CREATE UNDO TABLESPACE tstund DATAFILE  
'$ORACLE_BASE/oradata/test/undtst.dbf' SIZE 20 M REUSE AUTOEXTEND ON;
```

Το Undo Tablespace μας δημιουργήθηκε. Τώρα μένει να το χρησιμοποιήσουμε. Γράφουμε την παρακάτω εντολή για να το πράξουμε

```
SQL> ALTER SYSTEM SET UNDO_TABLESPACE = tstund;
```

Με την εκτέλεση της εντολής πλέον το Undo Tablespace της βάσης μας θα είναι το tstund.

5.1.5 – Σύνδεση στην βάση.

Αφού λοιπόν η ο listener είναι ενεργός, η βάση μας ανοιχτή, οι χρήστες δημιουργήθηκαν και τους αποδόθηκαν δικαιώματα και χώρος εργασίας σε ένα η περισσότερα Tablespaces, τότε το μόνο που μένει είναι ο χρήστης να συνδεθεί στην βάση.

Σε αυτό το σημείο ο χρήστης/χρήστες απλά έχουν να επιλέξουν την εφαρμογή ή το περιβάλλον Client που επιθυμούν ή επιβάλλουν οι απαιτήσεις. Ο χρήστης μπορεί να συνδεθεί στην βάση είτε από απομακρυσμένο υπολογιστή, είτε απευθείας στον Server, με τα στοιχεία λογαριασμού που του διατέθηκαν από τους διαχειριστές. Στην δεύτερη περίπτωση, ο χρήστης μπορεί να συνδεθεί στην βάση μέσα από το SQL*PLUS. Περισσότερες λεπτομέρειες σχετικά με τις Client Εφαρμογές και Περιβάλλοντα, θα δούμε στο κεφάλαιο 7.

Φυσικά όταν λέμε ότι οι χρήστες μπορούν να συνδεθούν δεν αναφερόμαστε αποκλειστικά στους Clients. Και οι διαχειριστές μπορούν να συνδεθούν απομακρυσμένα στην βάση δεδομένων και να την διαχειρίζονται. Υπάρχουν προφανώς περιβάλλοντα που εξυπηρετούν την ανάγκη αυτήν. Πάντως το κύριο ζητούμενο είναι ότι πλέον η βάση δεδομένων **είναι διαθέσιμη και λειτουργεί**, οπότε οι χρήστες (όποια και να είναι η φύση/ιδιότητά τους) μπορούν να την χρησιμοποιήσουν.

Ορισμός

Η σύνδεση ενός χρήστη με την βάση δεδομένων και η εργασία του σε αυτήν, ονομάζεται **συνεδρία** (session).

5.1.6 – Αλληλεπίδραση με την βάση δεδομένων

Στην βάση δεδομένων μπορούμε είτε να εργαστούμε ως απλοί χρήστες είτε να την διαχειριστούμε. Οι δύο αυτοί διαφορετικοί τρόποι αλληλεπίδρασης με την βάση, στην ουσία διαχωρίζουν τον ρόλο του διαχειριστή από τον χρήστη/πελάτη. Ως αλληλεπίδραση όμως, γενικά εννοούμε την **εργασία** στην βάση.

Στην περίπτωση της εργασίας στην βάση ως απλοί χρήστες, μέλημά μας είναι η αποθήκευση δεδομένων σε αυτή (τροποποίηση των περιεχομένων της) είτε η ανάκτηση δεδομένων από αυτήν. Τίποτα περισσότερο, τίποτα λιγότερο. Έχουμε ένα συγκεκριμένο λογαριασμό στην βάση, και τα δικαιώματά μας περιορίζονται μόνο στην αυστηρή εργασία μας με την **δομή περιεχομένου** της βάσης, δηλαδή με τα δεδομένα που περιέχει αποθηκευμένα μέσα της.

Ορισμός

Η διαδικασία στην οποία ζητάμε από την βάση δεδομένων να μας επιστρέψει δεδομένα είτε γενικά είτε βάση κριτηρίων, ονομάζεται **Επερώτημα** (Query) στην βάση. Τα επερωτήματα προς την βάση γίνονται με την SQL εντολή SELECT.

Τα επερωτήματα δεν επηρεάζουν την δομή περιεχομένου της βάσης δεδομένων, επειδή κάνουμε ανάκτηση των δεδομένων που ήδη περιέχει. Αντιθέτως όμως εντολές όπως η INSERT INTO προκαλούν αλλαγή της δομής περιεχομένου της βάσης.

Ορισμός

Η διαδικασία εισαγωγής, τροποποίησης και διαγραφής δεδομένων από την βάση ή οτιδήποτε άλλο προκαλεί αλλαγή της δομής περιεχομένου της, ονομάζεται **Συναλλαγή** (transaction³⁰). Οι ενέργειες μιας συναλλαγής έχουν ως συνέπεια την τροποποίηση του περιεχομένου της βάσης, και εφαρμόζονται **μόνο αν** η τελευταία πραγματοποιήθηκε χωρίς σφάλματα.

Οι συναλλαγές υπάρχουν για την ταυτόχρονη χρήση της βάσης από πολλούς χρήστες παράλληλα. Οι αλλαγές που προξενεί ένας χρήστης σε ένα κοινόχρηστο αντικείμενο, πχ έναν πίνακα, μπορεί να επηρεάσουν άλλο χρήστη που εργάζεται στο ίδιο αντικείμενο. Γι αυτό το λόγο οι ενέργειες μιας συναλλαγής δεν πραγματοποιούνται άμεσα, αλλά κατόπιν επιβεβαίωσης. Επίσης μπορούν να ακυρωθούν αν όντως το αποτέλεσμα της συναλλαγής προξένησε κάτι το ανεπιθύμητο. Δεν πρέπει όμως να παραληφθεί ότι εκτελούνται με την σειρά, δηλαδή πρέπει να ολοκληρωθεί μια προηγούμενη για να ξεκινήσει μια επόμενη και το ότι μπορούν να δημιουργηθούν **σημεία επαναφοράς** (Save Points) προτού ολοκληρωθεί πλήρως.

Ορισμός

Η επιβεβαίωση ολοκλήρωσης μιας συναλλαγής ονομάζεται **Διάπραξη** (Commit) κατά την οποία η συναλλαγή ολοκληρώνεται και οι ενέργειες τροποποίησης της δομής περιεχομένου της βάσης, λαμβάνουν χώρα. Αν όμως προκληθεί σφάλμα ή το αποτέλεσμά της δεν είναι αυτό που περίμενε ο χρήστης, τότε μπορεί να ακυρώσει την συναλλαγή, πραγματοποιώντας **Επαναφορά** (Rollback), η οποία επαναφέρει την βάση στην κατάσταση που ήταν πριν την εφαρμογή της συναλλαγής είτε ολοκληρωτικά, είτε σε κάποιο σημείο επαναφοράς (Save point). Σημειώνεται ότι μετά από Commit, οι αλλαγές δεν μπορούν να επαναφερθούν.

³⁰ Wikipedia Article → Database Transaction – Site (http://en.wikipedia.org/wiki/Database_transaction)

Οι βάσεις δεδομένων της Oracle είναι **συναλλαγματικές** (transactional) (εκτός από τις βάσεις τύπου Data Warehouse³¹). Η οποιαδήποτε ενέργεια που επηρεάζει το περιεχόμενό της, πρέπει να επιβεβαιωθεί πριν εφαρμοστεί. Γι αυτό το λόγο, πρέπει πάντοτε να εκτελείται η εντολή COMMIT μετά από κάθε τροποποίηση του περιεχομένου της βάσης, εκτός φυσικά αν η εφαρμογή ή το περιβάλλον εργασίας υποστηρίζει λειτουργία **αυτόματης διάπραξης** (auto-commit). Να σημειωθεί εδώ, πως ενώ σε άλλα συστήματα οι συναλλαγές ξεκινούν με την εντολή START TRANSACTION στην Oracle κάθε ενέργεια αλλαγής της δομής περιεχομένου της βάσης θεωρείται συναλλαγή οπότε η εντολή αυτή δεν χρειάζεται.

Ο λόγος για τον οποίο αναφερόμαστε σε αυτούς τους ορισμούς είναι αφενός μεν για καλύτερη κατανόηση της εσωτερικής λειτουργίας ενός Συστήματος Διαχείρισης Βάσεων Δεδομένων / ΣΔΒΔ (Database Management System / DBMS) – κυρίως της Oracle- αφετέρου δε γιατί πολλοί από αυτούς τους ορισμούς θα μας ακολουθήσουν στα επόμενα κεφάλαια.

Η αλληλεπίδραση με την βάση δεδομένων γίνεται απευθείας μέσω εντολών που δέχεται το ΣΔΒΔ και ανάλογα εκτελεί, δηλαδή στην **γλώσσα** (language) που το ΣΔΒΔ καταλαβαίνει. Στις προσεχείς ενότητες θα δούμε τις γλώσσες αυτές που καταλαβαίνει το DBMS της Oracle.

Προσοχή!

Οι συναλλαγές που δεν ολοκληρώθηκαν (με COMMIT ή ROLLBACK), σε περίπτωση τερματισμού της συνεδρίας (αποσύνδεση χρήστη) γίνεται αυτόματα COMMIT. Προσοχή λοιπόν, στις αλλαγές που πραγματοποιείτε πριν αποσυνδεθείτε από την βάση.

³¹ Oracle Documentation → Introduction to Data Warehousing → What Is a Data Warehouse? → “Data warehouses are not optimized for transaction processing” – Site (http://docs.oracle.com/cd/E18283_01/server.112/e10578/tdpdw_intro.htm)

5.1.6.1 – Η γλώσσα SQL

Όπως κάθε «αξιοπρεπές» ΣΔΒΔ, έτσι και η Oracle υποστηρίζει την γλώσσα **Structured Query Language** (SQL). Στην Oracle υποστηρίζεται πλήρως το ρεπερτόριο εντολών της SQL όπως η SELECT, INSERT INTO, UPDATE, DELETE κλπ είτε με την απευθείας συγγραφή τους σε κάποιο client εργαλείο είτε μέσα από αρχείο που ονομάζεται **Script**. Τα περισσότερα εργαλεία διαχείρισης ή/και εργασίας στην βάση δεδομένων της Oracle επιτρέπουν την συγγραφή εντολών SQL, πολλά από τα οποία υποστηρίζουν και **τονισμό σύνταξης** (Syntax Highlighting) των λέξεων-κλειδιών. Η εκδοχή SQL της Oracle έχει εμπλουτισμένο συντακτικό και οι εντολές έχουν περισσότερες δυνατότητες (μέσω περισσότερων **Ρητρών** (Clauses)) από ότι σε άλλα DBMS. Ο οδηγός διαφοροποίησης των εντολών βρίσκεται εδώ

http://docs.oracle.com/cd/B28359_01/server.111/b28286/wnsq1.htm#SQLRF50926

Ο χρήστης μπορεί να εκτελέσει (ανάλογα με τα δικαιώματα που έχει) όποια εντολή SQL επιθυμεί, είτε το ζητούμενο είναι η ανάκτηση δεδομένων είτε η αποθήκευση/τροποποίηση δεδομένων. Στην δεύτερη περίπτωση, για την εφαρμογή των προαναφερθέντων, είναι αναγκαία η **Διάπραξη** (Commit), γιατί όπως είπαμε και πριν οι βάσεις της Oracle είναι συναλλαγματικές. Η δομή εντολών SQL δίνεται παρακάτω:

ΕΝΤΟΛΗ <όρισμα> **PHTPA** (clause) <όρισμα / συνθήκη>

Παράδειγμα:

SELECT NAME **FROM** BOATS **WHERE** ID = 2;

UPDATE BOATS **SET** NAME = 'Mariza' **WHERE** ID = 3;

Δεν θεωρούμε σκόπιμο να επιταθούμε περεταίρω καθώς σκοπός μας δεν είναι η διδασχή της γλώσσας SQL καθώς θεωρείται ότι κατόπιν εργασίας σας σε άλλο DBMS την γνωρίζετε. Απλώς κάνουμε μια σύντομη αναφορά σε αυτήν για να σας δείξουμε πως αυτά που ήδη γνωρίζετε από την SQL μπορείτε να τα εφαρμόσετε και στην Oracle.

5.1.6.2 – Η γλώσσα PL/SQL

Η δυνατότητες που προσφέρει η Oracle καθώς και η πολύπλοκη δομή της βάσης της απαιτεί και μια γλώσσα που να μπορεί αποδοτικά να πραγματοποιεί πολύ συγκεκριμένες ή πολύπλοκες ενέργειες που η απλή SQL δεν καλύπτει. Γι αυτό το λόγο η Oracle δημιούργησε την γλώσσα PL/SQL³².

Η **γλώσσα Procedural Language / Structured Query Language** (PL/SQL) είναι μια επέκταση της SQL, η οποία μετατρέπει την τελευταία από απλή γλώσσα Επερωτημάτων σε μια ενισχυμένη διαδικαστική (procedural) γλώσσα, όπως πχ την C, η οποία εισάγει στοιχεία προγραμματισμού στην βάση δεδομένων. Με την PL/SQL ο χρήστης έχει στην διάθεση του **δομές επανάληψης** (loops), **συνθήκες** (conditions), **μεταβλητές** (variables), σταθερές (constants) κλπ, δηλαδή ότι συναντάμε σε μια κοινή γλώσσα προγραμματισμού.

Η PL/SQL λοιπόν ενισχύει τις δυνατότητες της βάσης δεδομένων και μας επιτρέπει να πραγματοποιήσουμε οποιαδήποτε –επιτρεπτή- ενέργεια επιθυμούμε. Η PL/SQL μας επιτρέπει να δημιουργήσουμε τα παρακάτω:

- 1) **Αυτόνομα Υποπρογράμματα**. Μπορούμε να εσωκλείσουμε μια σειρά εντολών SQL ή PL/SQL οι οποίες εκτελούνται όλες μαζί σαν ένα. Εκτελούνται άμεσα και δεν απαιτούν κλήση, επειδή δεν έχουν όνομα (ανώνυμα υποπρογράμματα).
- 2) **Διαδικασίες και Συναρτήσεις** (Procedures / Functions). Μπορούμε να δημιουργούμε υποπρογράμματα/υπορουτίνες στην PL/SQL οι οποίες μπορούν να δέχονται ορίσματα, να εκτελούν μια σειρά εντολών και να επιστρέφουν τιμή ή αποτέλεσμα (functions). Για να εκτελεστούν, θα πρέπει πρώτα να **κληθούν** (Called).

³² Wikipedia Article → PL/SQL → “PL/SQL (Procedural Language/Structured Query Language) is Oracle Corporation's procedural language extension for SQL and the Oracle relational database.” – Site (<http://en.wikipedia.org/wiki/PL/SQL>)

- 3) **Τύποι οριζόμενοι από το χρήστη** (User Defined Types). Η δυνατότητα που μας δίνει η PL/SQL να δημιουργούμε δικούς μας τύπους δεδομένων και βάση αυτών να δημιουργούμε πίνακες, εισάγει ένα επίπεδο **αντικειμενοστρέφειας** (object orientation) στην βάση δεδομένων καθώς οι τύποι μπορούν να **κληρονομηθούν** (inherited) σε άλλους τύπους που ονομάζονται **υποτύποι** (subtypes). Μέσω των τύπων λοιπόν μπορούμε στη συνέχεια να δημιουργήσουμε πίνακες.
- 4) **Εναύσματα** (Triggers). Πρόκειται για υποπρογράμματα τα οποία εκτελούνται αυτόματα μετά ή πριν την εκτέλεση ενός γεγονότος, πχ μετά την διαγραφή μιας γραμμής από έναν πίνακα να γράφει στην έξοδο «Row <τάδε> deleted from table <τάδε>».

Όλα τα παραπάνω τοποθετούνται σε μια οριοθετημένη περιοχή για να εκτελεστούν, η οποία οριοθετείται με της λέξεις BEGIN ... END; , περιοχή που ονομάζεται **Μπλοκ** (Block). Παρακάτω σας παρουσιάζουμε μια πολύ γενικευμένη εικόνα αυτών των blocks για να πάρετε μια ιδέα, ενώ στα παρακάτω κεφάλαια όπου επικεντρωνόμαστε σε συγκεκριμένα περιβάλλοντα και εφαρμογές θα δείξουμε και παραδείγματα στην πράξη. Προς το παρών, η ενότητα αυτή είναι περισσότερο εισαγωγικού χαρακτήρα.

Ορισμός

Ένα σύνολο εντολών SQL ή PL/SQL που περικλείεται σε BEGIN...END; χωρίς όνομα και εκτελείται άμεσα χωρίς κλήση, ονομάζεται **Ανώνυμο Μπλοκ** (Anonymous Block) ή ανώνυμο υποπρόγραμμα.

Η δομή ενός Ανώνυμου Block δίνεται παρακάτω:

DECLARE

<περιοχή δήλωσης μεταβλητών>

BEGIN

<εντολή ή εντολές SQL ή/και PL/SQL>

END;

Το ανώνυμο block εκτελείται όπως είναι χωρίς να χρειαστεί να κληθεί, και οι εντολές εκτελούνται με την σειρά.

Ορισμός

Ένα Block Εντολών που συγκροτεί μια procedure/function, ένα trigger ή ένα τύπο, ονομάζεται **επώνυμο block** (named block). Τα επώνυμα blocks για να εκτελεστούν πρέπει να **κληθούν** είτε **άμεσα** (στις περιπτώσεις procedure/function) ή **έμμεσα** (σε περιπτώσεις trigger).

Τα επώνυμα Blocks ξεκινούνε συνήθως με την εντολή CREATE (OR REPLACE) προτού δημιουργηθούν, ενώ blocks όπως αυτά των Procedure για να εκτελεστούν, πρέπει πρώτα να κληθούν είτε με την εντολή EXECUTE είτε μέσα από άλλο Ανώνυμο Block. Ενδεικτικό παράδειγμα παρακάτω:

```
CREATE OR REPLACE PROCEDURE test --Δημιουργία Procedure
AS
BEGIN
DBMS_OUTPUT.PUT_LINE('HELLO FROM ORACLE'); --STDOUT
END;
/
BEGIN
test; --Κλήση
END;
```

Keep in mind...

Η ενότητα αυτή έχει σκοπό να σας εισάγει στην PL/SQL ώστε να κατανοήσετε μερικές βασικές έννοιες πολλές από τις οποίες θα δούμε παρακάτω στην πράξη. Στόχος μας δεν είναι να σας διδάξουμε την γλώσσα. Επίσης παρουσιάζουμε τα χαρακτηριστικά της σε πολύ αφηρημένο επίπεδο, λόγω το ότι οι εφαρμογές που θα δούμε παρακάτω διαφέρουν ως προς τον τρόπο με τον οποίο εκτελούν κάποιες εντολές PL/SQL. Οπότε προτιμήσαμε, για δική σας διευκόλυνση, τα εξειδικευμένα χαρακτηριστικά και παραδείγματα με κώδικα να τα δείξουμε στις ενότητες εκείνες.

Έτσι λοιπόν με την χρήση της PL/SQL μπορούμε να κάνουμε πολλά περισσότερα από ότι μπορεί να κάνει η Standard SQL, οπότε έχουμε περισσότερο έλεγχο πάνω στην βάση. Σε παρακάτω ενότητες θα την δούμε και επί το έργω.

5.2 – Εργαλεία Διαχείρισης Βάσης Δεδομένων

Η διαχείριση της βάσης δεδομένων γίνεται μέσα από ειδικές εφαρμογές που επιτρέπουν στον διαχειριστή να πραγματοποιεί τις επιθυμητές ενέργειές του πάνω σε αυτήν, είτε να παρακολουθεί την κατάσταση λειτουργίας της. Οι εφαρμογές αυτές θα πρέπει να είναι σε θέση να εκπληρώσουν αυτό το ζητούμενο, παρέχοντας πλήρη έλεγχο πάνω στην βάση καθώς και λεπτομερείς πληροφορίες σχετικά με την κατάστασή της. Κανένας διαχειριστής δεν θέλει εκπλήξεις.

Η Oracle φροντίζει λοιπόν για την κάλυψη των αναγκών των διαχειριστών, μέσω ισχυρών εφαρμογών που του παρέχουν ισχυρά εργαλεία για την σωστή και αποδοτική διαχείριση της βάσης δεδομένων. Είναι αρκετά εύχρηστα και συνάμα

παραγωγικά. Μαζί θα δούμε τις κυριότερες εφαρμογές και περιβάλλοντα που επιτελούν ακριβώς αυτό το σκοπό.

5.2.1 – Η Εφαρμογή SQL*PLUS

Αν και το χρησιμοποιήσαμε κατά κόρων παραπάνω, ωστόσο αξίζει να μιλήσουμε περισσότερο γι αυτό, κυρίως λόγω των δυνατοτήτων του και της σημαντικότητάς του, στον Server.

Το **SQL*PLUS** είναι μια πλήρης εφαρμογή διαχείρισης βάσεων της Oracle, που παρέχει και τις περισσότερες δυνατότητες από οποιαδήποτε άλλη εφαρμογή αναλόγων καθηκόντων. Είναι **Terminal Based** εφαρμογή, δηλαδή ο διαχειριστής το χρησιμοποιεί μέσα από Terminal Emulator, είτε τοπικά (πχ με το Konsole) είτε απομακρυσμένα (πχ με το PuTTY). Επίσης εγκαθίσταται **μαζί με την Oracle**. Παρακάτω βλέπουμε ένα στιγμιότυπο από την εφαρμογή σε λειτουργία:

```
SQL*Plus: Release 11.2.0.1.0 Production on Wed Nov 20 11:22:09 2013
Copyright (c) 1982, 2009, Oracle. All rights reserved.
Enter password:
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
SQL> SELECT * FROM BOATS;

   ID NAME                LICENSE
-----
    1 Mariza                2352326
    2 Eliza                 125125
    3 Anna                  73462
    4 Skorprios             346346
    5 Kap.Dimitrios        4634634
SQL> █
```

*Η εφαρμογή SQL*PLUS εν λειτουργία*

Αν και θα μας ήταν πιο εύκολο να σας περιγράψουμε τι ΔΕΝ είναι το SQL*PLUS, ωστόσο θα κάνουμε το αντίθετο σε μια προσπάθεια να καλύψουμε όσο το δυνατόν περισσότερο την φύση της εφαρμογής αυτής. Το SQL*PLUS είναι

πραγματικά ένα πολύ-εργαλείο, ένας ελβετικός σουγιάς αν θέλετε το οποίο το καθιστά το πλέον ισχυρότερο εργαλείο διαχείρισης της βάσης δεδομένων. Ας δούμε μαζί τι μπορεί να κάνει και γιατί προτιμάται περισσότερο από άλλες εφαρμογές.

Το SQL*PLUS λοιπόν μπορεί να χρησιμοποιηθεί με τους εξής τρόπους, αναλόγως την περίπτωση:

- 1) **Ως εργαλείο ελέγχου της βάσης.** Η βάση δεδομένων μπορεί να ξεκινάει και να τερματίζεται (STARTUP – SHUTDOWN) μόνο μέσα από το SQL*PLUS. Οι αλλαγές επίσης της φύσης και της υπόστασης της βάσης γίνονται μόνο μέσα από αυτό. Το γεγονός αυτό το καθιστά **γενικό panel ελέγχου** της βάσης.
- 2) **Ως εργαλείο διαχείρισης της βάσης.** Μέσα από το SQL*PLUS ο διαχειριστής μπορεί να δημιουργεί/τροποποιεί/διαγράφει χρήστες, πίνακες, Tablespace, ρόλους, όψεις κλπ. Αποτελεί επίσης πλήρως πληροφοριακού τύπου εργαλείο όπου ο διαχειριστής μπορεί να βλέπει πληροφορίες όπως υπόσταση αντικειμένων (describe) ή τρέχουσες επιλογές εν ισχύ (show).
- 3) **Ως Client εργαλείο.** Ένας Client μπορεί να χρησιμοποιήσει το SQL*PLUS για να εργαστεί στην βάση δεδομένων.

Keep in mind...

Οι μόνες εφαρμογές που μπορούν να εκκινήσουν την βάση δεδομένων είναι το SQL*PLUS, το Enterprise Manager και το RMAN³³. Η καλύτερη και πιο σίγουρη όμως μέθοδος είναι μέσα από το SQL*PLUS.

5.2.1.1 – Εκκίνηση SQL PLUS και δημιουργία συνεδρίας

Το SQL*PLUS βρίσκεται στον κατάλογο ORACLE_HOME/bin. Για να το εκτελέσετε και να συνδεθείτε σε αυτό χρησιμοποιήστε τις εντολές που

³³ Oracle Documentation → Starting Up a Database → Options for Starting Up a Database → “You can start up a database instance with SQL*Plus, Recovery Manager, or Enterprise Manager.” – Site (http://docs.oracle.com/cd/B28359_01/server.111/b28310/start001.htm#ADMIN10063)

παραθέτουμε στον παρακάτω πίνακα, αφού μεταβείτε στον κατάλογο που βρίσκεται.

Εντολή	Παράδειγμα	Αποτέλεσμα
./SQL*PLUS	N/A	Εκκινεί το SQL*PLUS. Τα διαπιστευτήρια σύνδεσης εισάγονται αμέσως μετά.
./SQL*PLUS <user>/<password>	./SQL*PLUS scott/tiger	Εκκινεί το SQL*PLUS δημιουργώντας συνεδρία χρήστη απευθείας στο τρέχων Instance (ORACLE_SID)
./SQL*PLUS <user>/<password>@<instance>	./SQL*PLUS lefteris/secret@test	Εκκινεί το SQL*PLUS, δημιουργεί συνεδρία χρήστη σε συγκεκριμένο Instance (βάση δεδομένων)
./SQL*PLUS <user> AS <ιδιότητα>	./SQL*PLUS sys as sysdba	Εκκινεί το SQL*PLUS δημιουργώντας συνεδρία σε χρήστες με ειδικά δικαιώματα, όπως διαχειριστές, στο τρέχων Instance.
./SQL*PLUS /nolog	N/A	Εκκινεί το SQL*PLUS χωρίς να δημιουργήσει συνεδρία. Η σύνδεση

		πρέπει να γίνει χειροκίνητα με την εντολή CONNECT.
--	--	--

5.2.1.2 – Είδη Εντολών και Ορθή Σύνταξη

Βλέπουμε λοιπόν πως το SQL*PLUS καλύπτει όλα τα «γούστα» ανάλογα με την περίπτωση. Ωστόσο όμως τι ακριβώς μπορεί ο διαχειριστής να εισάγει σε αυτό, δηλαδή ποια είναι η φύση των εντολών που επιτρέπεται να εισαχθούν και να εκτελεστούν; Ας δούμε:

- 1) **Standard SQL εντολές**. Ο διαχειριστής (και ο client) μπορεί να εκτελέσει κανονικές εντολές SQL όπως SELECT, INSERT INTO, UPDATE, DELETE κλπ. Υποστηρίζεται πλήρες ρεπερτόριο εντολών SQL από το SQL*PLUS (το λέει και το όνομά του άλλωστε!).
- 2) **PL/SQL Εντολές**. Μπορούμε να εκτελέσουμε PL/SQL Blocks μέσα από το SQL*PLUS είτε ανώνυμα είτε στοχευόμενα
- 3) **Εσωτερικές Εντολές**. Το SQL*PLUS διαθέτει ένα δικό του ρεπερτόριο εντολών περισσότερο για διευκόλυνση του χρήστη για γενικές εργασίες στην βάση όπως πχ, επανεκτέλεση εντολών μέσα από τον buffer ή μορφοποίηση εξόδου στο STDOUT. Στις εσωτερικές εντολές συγκαταλέγονται και οι εντολές ελέγχου της βάσης δεδομένων (startup – shut, create user ...) και ελέγχου συνεδρίας (connect – disconnect, alter session ...).
- 4) **Εντολές λειτουργικού συστήματος**. Το SQL*PLUS επιτρέπει την εκτέλεση εντολών του λειτουργικού συστήματος χωρίς να χρειαστεί να το κλείσετε.

Για να χρησιμοποιήσουμε αποδοτικά το SQL*PLUS πρέπει να **συντάσσουμε** τις εντολές σύμφωνα με τον τρόπο που αυτό ορίζει. Η σύνταξη των εντολών γίνεται ως εξής:

SQL Εντολές. Τις εισάγουμε αυτούσιες και τις τερματίζουμε με Semicolon «;».

Παράδειγμα:

```
SQL> SELECT * FROM BOATS WHERE ID BETWEEN 2 AND 6;
```

PL/SQL Εντολές: Τα ανώνυμα PL/SQL Blocks τα χρησιμοποιούμε με την τυποποίηση DECLARE...BEGIN...END, ενώ τα επώνυμα σύμφωνα με το είδος τους. Τα τερματίζουμε με το σύμβολο « / »

Παράδειγμα:

```
SQL> DECLARE
2   BOATNAME VARCHAR2(30);
3   BEGIN
4   SELECT NAME INTO BOATNAME FROM BOATS WHERE ID = 1;
5   DBMS_OUTPUT.PUT_LINE(BOATNAME);
6   SET SERVEROUTPUT ON;
7   END;
8   /
```

PL/SQL procedure successfully completed.

Εκτέλεση Scripts. Από το SQL*PLUS μπορούμε να τρέξουμε SQL Scripts από αρχείο, βάζοντας στην αρχή της πρότασης το σύμβολο « @ » και κολλητά το path που βρίσκεται το εν λόγω SQL Script αρχείο.

Παράδειγμα:

```
SQL> @/home/oracle/scripts/addUsers.sql
```

Εκτέλεση Εντολών του λειτουργικού συστήματος. Για να εκτελέσουμε μια εντολή του λειτουργικού συστήματος, όπως την ls ή cd πρέπει να εισάγουμε στην αρχή της πρότασης το σύμβολο « ! ». Να σημειωθεί ότι η εντολή εκτελείται χωρίς να κλείσει το SQL*PLUS.

Παράδειγμα:

```
SQL> !ls -ltr /home/oracle
```

<αποτέλεσμα εντολής>

5.2.1.3 – Επιπρόσθετα Χαρακτηριστικά

Το SQL*PLUS διαθέτει πολλές επιπλέον δυνατότητες. Μια εξ αυτών είναι ένας προσωρινός χώρος αποθήκευσης (ανά συνεδρία χρήστη) που ονομάζεται **SQL Buffer**. Μέσα σε αυτόν μπορείτε να αποθηκεύετε έναν αριθμό εντολών που χρησιμοποιείτε πολύ συχνά, πχ SELECT Statements σε μεγάλους πίνακες, και να τις εκτελείτε με μία κίνηση. Αναντίληκτα, πρόκειται για μεγάλη διευκόλυνση ιδιαίτερα σε μακροσκελείς και πολύπλοκες εντολές. Στον παρακάτω πίνακα παραθέτουμε τις κύριες εντολές εργασίας με τον Buffer.

Εντολή	Σύνταξη	Αποτέλεσμα
INPUT	INPUT <εντολή SQL ή τμήμα της(clauses)>	Αποθηκεύει την εντολή στον Buffer.
LIST	LIST {αριθμός γραμμής}	Εμφανίζει τα περιεχόμενα του Buffer είτε ολόκληρου είτε συγκεκριμένης γραμμής

GET	GET <path προς αρχείο>	Αποθηκεύει στον Buffer τις εντολές από Script Αρχείο.
CLEAR BUFFER	CLEAR BUFFER	Αδειάζει τον Buffer.
RUN	RUN {αριθμός γραμμής}	Εκτελεί τις εντολές ή την γραμμή εντολής που υπάρχει στο Buffer.
SAVE	SAVE <path προς αρχείο>	Αποθηκεύει τα περιεχόμενα του Buffer σε SQL Script αρχείο.

Ένα επίσης αξιοσημείωτο χαρακτηριστικό του SQL*PLUS είναι ότι σας δίνει την δυνατότητα να συντάξετε **Scripts** «στο φτερό», δηλαδή επί τόπου. Και ακόμα καλύτερα, με τον Text Editor που θα του πείτε εσείς να χρησιμοποιήσει. Στον παρακάτω πίνακα βλέπουμε τις βασικές εντολές που χρειάζεστε για να συντάξετε ένα Script.

Εντολή	Σύνταξη	Αποτέλεσμα
DEFINE _EDITOR	DEFINE _EDITOR=<όνομα text editor>	Μεταβλητή που ορίζει ποιον Text Editor θα χρησιμοποιήσει το SQL PLUS για την σύνταξη Scripts. Μπορεί να είναι VI, Nano κ.α.
EDIT	EDIT <path προς αρχείο>	Ενεργοποιεί τον ED και ανοίγει τον Text Editor που επιλέξατε για να συντάξετε το Script

Keep in mind...

Για να μπορέσετε να χρησιμοποιήσετε την λειτουργία σύνταξης Scripts, θα πρέπει να υπάρχει στο σύστημα εγκατεστημένο το πακέτο ED. Αν δεν το έχετε εγκαταστήστε το (ως root) με την εντολή «apt-get install ed». Επίσης μην ξεχάσετε να ορίσετε τον Editor (DEFINE _EDITOR) πριν την εκκίνηση σύνταξης του Script.

Θα θέλαμε να επεκταθούμε κι άλλο στις δυνατότητες του SQL*PLUS αλλά πολύ φοβάμαι ότι θα χρειαστούμε ένα ολόκληρο σύγγραμμα για να το καλύψουμε πλήρως. Ωστόσο σας παρουσιάσαμε τα πιο βασικά χαρακτηριστικά του ώστε να μπορέσετε να το χρησιμοποιήσετε για τις ζητούμενες εργασίες σας, αποδοτικά.

5.2.1.4 – Συνεδρία χρηστών και χρήση ως Client

Ο κάθε χρήστης που συνδέεται στο SQL*PLUS δημιουργεί μια συνεδρία. Ο χρήστης, ανάλογα με τα δικαιώματά του, του επιτρέπονται συγκεκριμένες ενέργειες πάνω στην βάση, είτε αφορά την υπόστασή της είτε τα περιεχόμενά της. Προφανώς στα δεύτερα περιορίζονται μόνο οι απλοί χρήστες. Για να συνδεθεί / αποσυνδεθεί ένας χρήστης στην βάση μέσω του SQL*PLUS χρησιμοποιεί τις κάτωθι εντολές:

Εντολή	Παράδειγμα	Αποτέλεσμα
CONN{ECT} <user>/<password>	CONNECT scott/tiger	Δημιουργεί συνεδρία στον χρήστη, στο τρέχων Instance.
CONN{ECT} <user>/<password> as <ρόλος>	CONNECT SYS AS SYSDBA	Δημιουργεί συνεδρία με εκχώρηση ρόλου στον χρήστη για το τρέχων Instance.

DISC{ONNECT}	N/A	Τερματίζει την τρέχουσα συνεδρία του χρήστη με την βάση
--------------	-----	---

Το SQL*PLUS λοιπόν, εφόσον μας επιτρέπει να συνδεόμαστε και ως απλοί χρήστες, μπορεί κάλλιστα να χρησιμοποιηθεί ως **Client**. Ως εκ τούτου θα έπρεπε να το παρουσιάσουμε στο κεφάλαιο 7 (Σύνδεση και εργασία χρηστών). Ωστόσο επιλέξαμε να το παρουσιάσουμε πιο εκτενώς στο παρών κεφάλαιο, καθώς η χρήση του περιορίζεται **τοπικά** στον Server και δεν μπορεί να εγκατασταθεί σε Client Υπολογιστή, για την ακρίβεια δεν μπορεί να συνδεθεί σε απομακρυσμένη βάση δεδομένων. Ωστόσο πάντα μπορείτε να το χρησιμοποιήσετε μέσω SSH από τον Server. Αυτός είναι και ο λόγος που δεν το αναφέρουμε στο κεφάλαιο 7.

Σαν Client το SQL*PLUS μπορεί να εκτελέσει όλες τις εντολές SQL και PL/SQL που θα μπορούσε ένας απλός χρήστης να χρησιμοποιήσει σε οποιοδήποτε άλλο λογισμικό Client. Φυσικά εξαρτάται από τα δικαιώματα του και από το ρόλο που του έχει εκχωρηθεί το εύρος των εντολών που μπορεί να τρέξει.

5.2.1.5 – Τερματισμός SQL*PLUS

Αφού ο χρήστης/διαχειριστής επιθυμεί να τερματίσει το SQL*PLUS θα πρέπει πρώτα να αποσυνδεθεί και μετά να τερματίσει την εφαρμογή. Δηλαδή θα πρέπει πρώτα να εκτελέσει την εντολή DISCONNECT και μετά να εκτελέσει την εντολή QUIT ή EXIT. Τότε θα επιστρέψει στο Prompt του λειτουργικού συστήματος.

5.2.1.6 – Συμπέρασμα

Το SQL*PLUS είναι ίσως το πιο συχνά χρησιμοποιούμενο εργαλείο από τον διαχειριστή. Του δίνει τον απόλυτο έλεγχο πάνω στην βάση δεδομένων και η

ευχρηστία του σε συνδυασμό με τις τεράστιες δυνατότητές του, το καθιστούν απολύτως απαραίτητο για τον διαχειριστή. Μοναδικά του **μειονεκτήματα** είναι η απουσία τονισμού σύνταξης (Syntax Highlighter), η απουσία δυνατότητας επαναφοράς προηγούμενης εντολής με το πλήκτρο « ↑ » και η αδυναμία του να δεχτεί πολλαπλές γραμμές εντολών με Copy-Paste. Ωστόσο είναι λίγα μπροστά στα πλεονεκτήματα που προσφέρει. Χρησιμοποιήστε το SQL*PLUS κυρίως για την διαχείριση της βάσης σας, και αν θέλετε για την εργασία σας πάνω σε αυτήν. Να θυμάστε ότι ορισμένες ενέργειες πάνω στην βάση (STARTUP, ALTER DATABASE, ALTER SYSTEM, SHUT κ.α.) γίνονται μόνο μέσα από αυτό.

Καλή πρακτική

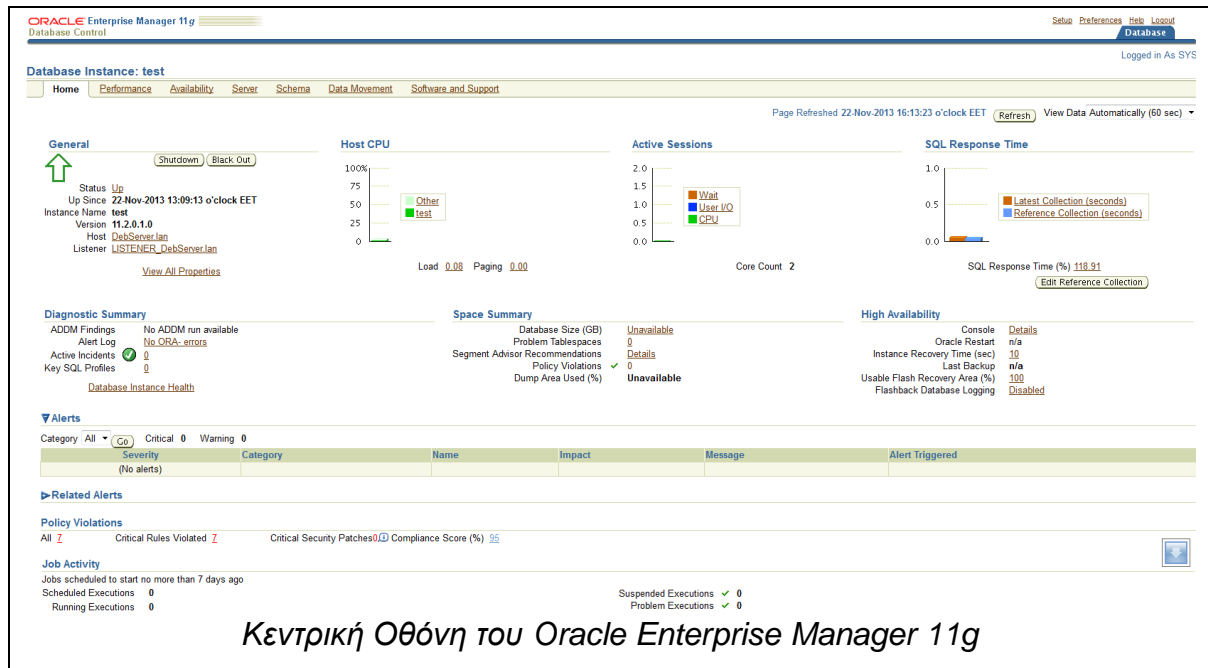
Γενικά καλό είναι τις εντολές του SQL*PLUS οποιασδήποτε φύσεως, να τις γράφετε με κεφαλαία γράμματα. Επίσης μην ξεχνάτε ποτέ να τις τερματίζετε με Semicolon. Αν υποπέσετε σε συνεχή συντακτικά σφάλματα μιας εντολής, μπορείτε να έχετε ανοιχτό δίπλα έναν Text Editor όπως τον Kwrite για να γράφετε εκεί τις εντολές σας πριν τις εκτελέσετε. Εκμεταλλευτείτε επίσης το γεγονός ότι ορισμένες εντολές μπορούν να γραφτούν συντομότερα (όπως η CONN{ECT}).

5.2.2 – Το Περιβάλλον Oracle Enterprise Manager (EM)

Η διαχείριση της βάσης δεδομένων δεν γίνεται μόνο μέσα από το SQL*PLUS. Υπάρχει και ένα πιο εύχρηστο εργαλείο για την ικανοποίηση του προηγούμενου, που επίσης μπορεί να χρησιμοποιηθεί από **οπουδήποτε** και δεν απαιτεί τίποτα περισσότερο από έναν **πλοηγό ιστού** (Web Browser). Μιλάμε για το περιβάλλον Oracle **Enterprise Manager** ή ποιο σύντομα **EM**.

Τι ακριβώς είναι το EM; Το EM είναι μια πλήρης σουίτα ελέγχου και διαχείρισης του Oracle Server μας, στην οποία μπορούμε να πραγματοποιήσουμε πολλές εργασίες διαχείρισης που θα πραγματοποιούσαμε στο SQL*PLUS αλλά με ποιο εύκολο και γρήγορο τρόπο. Και πως γίνεται αυτό; Επειδή πρόκειται για **GUI**

(Graphical User Interface) εφαρμογή και όχι για Console/Terminal Based. Παρακάτω βλέπουμε την κεντρική οθόνη του EM.



Σε αντίθεση με το SQL*PLUS, το EM διαθέτει όλα τα εργαλεία διαχείρισης του Oracle Server διαθέσιμα με μερικά κλικ του ποντικιού, αντί να χρειάζεται ο διαχειριστής να γράφει μία-μία τις εντολές που απαιτούνται. Πρόσθετα όμως, παρουσιάζει στον διαχειριστή και τις επιδόσεις του Server όσον αφορά τις βάσεις δεδομένων, πχ φόρτος απασχόλησης CPU. Και όλα αυτά σε ένα ελκυστικό περιβάλλον, προσβάσιμο από οπουδήποτε, μόνο από τον Web Browser.

Με το EM ο διαχειριστής μπορεί:

- 1) Να διαχειρίζεται την βάση δεδομένων όσον αφορά το περιεχόμενό της (πίνακες, triggers, procedures/functions κλπ).
- 2) Να διαχειρίζεται την βάση δεδομένων όσον αφορά την υπόστασή της (χρήστες, character sets, Tablespaces, control files κλπ).
- 3) Να παρακολουθεί τις επιδόσεις του Server ώστε να γνωρίζει τον τρέχοντα φόρτο του υπολογιστή αυτού και να πράξει ανάλογα σε ανάλογες περιπτώσεις (πχ γεμάτο Tablespace).

- 4) Να εκτελεί εργασίες συντήρησης Server (backup – restore)
- 5) Να εκτελεί εργασίες ασφαλείας (policies, επαναφορά χαμένων passwords, Database Vault κλπ)
- 6) Να εργαστεί στην βάση ως χρήστης (μέσω SQL).

Τα παραπάνω ήταν απλά μερικά από τα πάρα πολλά χαρακτηριστικά του EM. Θεωρείται απολύτως απαραίτητο εργαλείο για κάθε διαχειριστή Oracle Server και για το λόγο της ευχρηστίας του αλλά και για την αποτελεσματικότητά του.

5.2.2.1 – Εκκίνηση του EM

Για να χρησιμοποιήσετε το EM θα πρέπει πρώτα να το ενεργοποιήσετε, ώστε να τρέχει σαν διεργασία στον Server. Από την εγκατάσταση και έπειτα, δημιουργήσαμε μια βάση δεδομένων ωστόσο δεν δημιουργήσαμε κάτι πολύ σημαντικό για το EM. Το αποθετήριο.

Ορισμός

Αποθετήριο (Repository) ονομάζεται ένα σύνολο πινάκων που υπάρχουν στην βάση δεδομένων και ρόλος του είναι να αποθηκεύουν πληροφορίες σχετικά με την εσωτερική κατάσταση της αλλά και για την διαχείριση της. Ο ιδιοκτήτης της είναι ένας χρήστης της βάσης (Συνήθως ο SYSMAN) ο οποίος ονομάζεται **ιδιοκτήτης αποθετηρίου** (Repository Owner).

Το αποθετήριο λοιπόν χρειάζεται οπωσδήποτε να υπάρχει για να υφίσταται το EM. Θα πρέπει λοιπόν να δημιουργήσουμε ένα. Η Oracle μας παρέχει το εργαλείο EMCA (Enterprise Manager Configuration Assistant), το οποίο θα χρησιμοποιήσουμε για να δημιουργήσουμε το αποθετήριό μας.

Προτού όμως δημιουργήσουμε το αποθετήριο μας, θα πρέπει να ενεργοποιήσουμε τον χρήστη DBSNMP ο οποίος θα χρησιμοποιείται από το EM για σκοπούς monitoring, καθώς από προεπιλογή είναι κλειδωμένος. Ανοίγουμε το SQL*PLUS ως SYS χρήστες και γράφουμε:

```
SQL> ALTER USER DBSNMP ACCOUNT UNLOCK;  
  
USER ALTERED.  
  
SQL> ALTER USER DBSNMP IDENTIFIED BY <κωδικός πρόσβασης>  
  
USER ALTERED.
```

Αφού ενεργοποιήσουμε τον χρήστη DBSNMP και αφού βεβαιωθούμε ότι η βάση είναι Mounted και Open, γράφουμε τις παρακάτω εντολές για να δημιουργήσουμε το αποθετήριο:

```
oracle@debserver$ cd $ORACLE_HOME/bin  
  
oracle@debserver$ ./emca -config dbcontrol db -repos create
```

Θα σας ζητήσει συγκεκριμένα στοιχεία όπως το Instance ID της βάσης, τους κωδικούς των διαχωριστών, θύρα Listener κλπ. Συμπληρώστε ότι σας ζητάει και στο τέλος επιβεβαιώστε τις αλλαγές, επιλέγοντας Y (yes) στο Prompt για να συνεχίσει ή όχι. Η διαδικασία θα πάρει κάμποσο χρόνο, οπότε κάντε υπομονή. Σημειώνεται ότι η παραπάνω διαδικασία εκτελείται **μόνο μια φορά** (για την ίδια βάση δεδομένων/Instance).

Με το πέρας της δημιουργίας του αποθετηρίου, είμαστε έτοιμοι να ενεργοποιήσουμε το EM. Το εργαλείο που παρέχει η Oracle για να ενεργοποιήσετε το EM είναι το εργαλείο EMCTL (EM Control) που βρίσκεται στον κατάλογο ORACLE_HOME/bin. Χρησιμοποιείτε την παρακάτω εντολή για να ενεργοποιήσετε το EM.

```
oracle@debserver$ cd $ORACLE_HOME/bin  
oracle@debserver$ ./emctl start dbconsole
```

Η ενεργοποίηση του EM αρχίζει και σύντομα θα ειδοποιηθούμε ότι είναι ενεργή. Θα πρέπει όμως να το διαπιστώσουμε από πρώτο χέρι, με την εντολή

```
oracle@debserver$ ./emctl status dbconsole  
  
<omit output>  
  
https://DebServer.lan:1158/em/console/aboutApplication # ← Socket (HTTPS)  
  
Oracle Enterprise Manager 11g is running  
  
....
```

Καλό σημάδι μέχρι στιγμής. Με την εντολή «emctl start dbconsole» ενεργοποιήσαμε την **Κονσόλα** (Console) του EM, δηλαδή την γραφική διεπιφάνεια (graphical interface) του πίνακα ελέγχου της βάσης δεδομένων. Η κονσόλα όμως από μόνη της δεν μπορεί να κάνει τίποτα γι αυτό ενεργοποιείται σε συνδυασμό με τον **Management Agent**, ο οποίος είναι μια διεργασία που τρέχει στον Server και εκτελεί τις διαδικασίες διαχείρισης και παρακολούθησης. Η επικοινωνία της κονσόλας (στον browser του client) και του Agent (process στο server) γίνεται μέσω HTTP ή HTTPS (στην δική μας περίπτωση το δεύτερο). Για να ελέγξουμε την κατάσταση του Agent γράφουμε την παρακάτω εντολή:

```
oracle@debserver$ ./emctl status agent  
  
<omit output>  
  
....  
  
Agent is Running and Ready  
  
...
```

Πρακτικά αυτό σημαίνει η process του EM τρέχει και περιμένει συνδέσεις. Η κονσόλα όμως ακούει σε συγκεκριμένη θύρα, την οποία πρέπει να ανακαλύψουμε πριν συνδεθούμε. Γράφουμε την παρακάτω εντολή για να δούμε σε ποια θύρα ακούει ο EM.

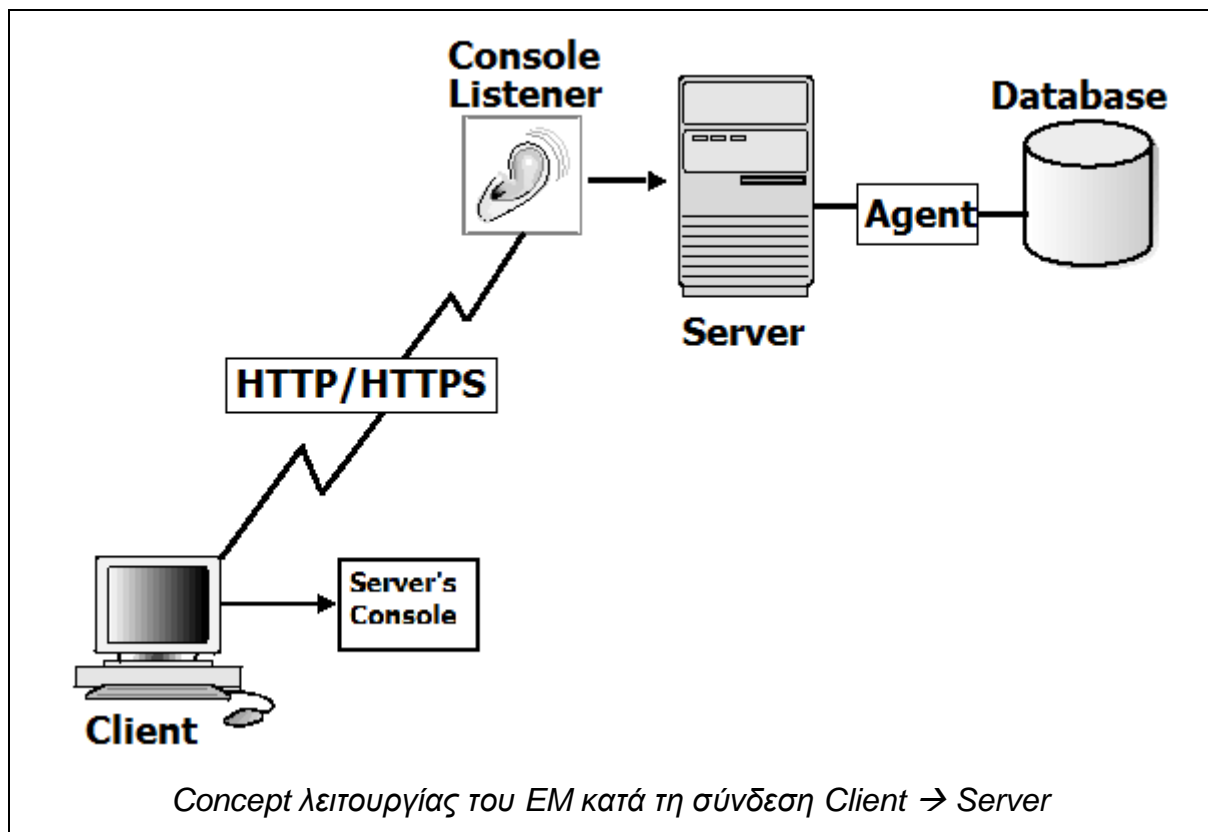
```
oracle@debserver$ cat $ORACLE_HOME/install/portlist.ini
```

```
Enterprise Manager Console HTTP Port (test) = 1158 # ← Κονσόλα
```

```
Enterprise Manager Agent Port (test) = 3938 # ← Agent
```

Η θύρα της κονσόλας μας ενδιαφέρει άμεσα και βλέπουμε πως είναι η 1158. Αυτήν λοιπόν την θυμόμαστε γιατί σε αυτήν θα πραγματοποιήσουμε σύνδεση.

Στο παρακάτω σχήμα δίνουμε μια ιδέα για την λειτουργία του EM κατά την σύνδεση Client στον Server.



5.2.2.2 – Σύνδεση στο EM

Αφού ενεργοποιήσαμε την κονσόλα και τον Agent τώρα δεν μένει παρά να συνδεθούμε στο EM. Μεταφερόμαστε στον Client Υπολογιστή (ή τοπικά αν θέλουμε) και στο address bar του Web Browser δίνουμε το εξής Socket (hint: το είδαμε στην εκτέλεση της emctl status dbconsole).

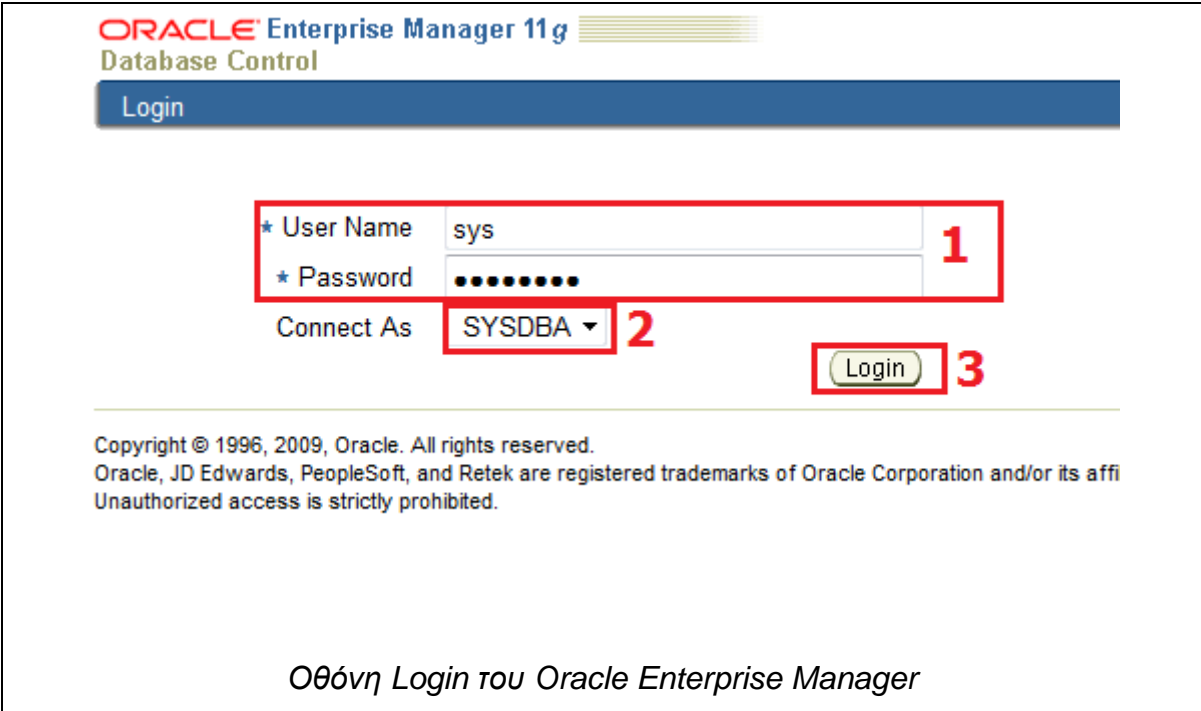
https://<server address>:<console_port>/em

Παράδειγμα:

(τοπικά) <https://127.0.0.1:1158/em>

(απομακρυσμένα) <https://Debserver:1158/em>

Σε λιγάκι θα πρέπει να μας εμφανίσει την σελίδα Login του Oracle Enterprise Manager η οποία θα είναι η παρακάτω:



ORACLE Enterprise Manager 11g
Database Control

Login

* User Name sys **1**

* Password

Connect As SYSDBA **2**

Login **3**

Copyright © 1996, 2009, Oracle. All rights reserved.
Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affi
Unauthorized access is strictly prohibited.

Οθόνη Login του Oracle Enterprise Manager

Σε αυτήν την οθόνη θα πρέπει να εισάγουμε τα στοιχεία μας (1), τον ρόλο μας (2) και να πατήσουμε Login. Οι χρήστες που μπορούν να συνδεθούν στο EM είναι οι παρακάτω:

Χρήστης	Ρόλος	Είδος
SYS	SYSDBA	Γενικός Διαχειριστής της βάσης.
SYSTEM	NORMAL	Super Διαχειριστής του EM.
SYSMAN	NORMAL	Super Διαχειριστής και Ιδιοκτήτης του Αποθετηρίου του EM (ο default).

Διαλέγουμε τον χρήστη που θέλουμε (πχ τον SYS) και συνδεόμαστε στο EM. Από κει και πέρα έχουμε στα χέρια μας ένα ισχυρό εργαλείο παρακολούθησης και διαχείρισης του Server μας.

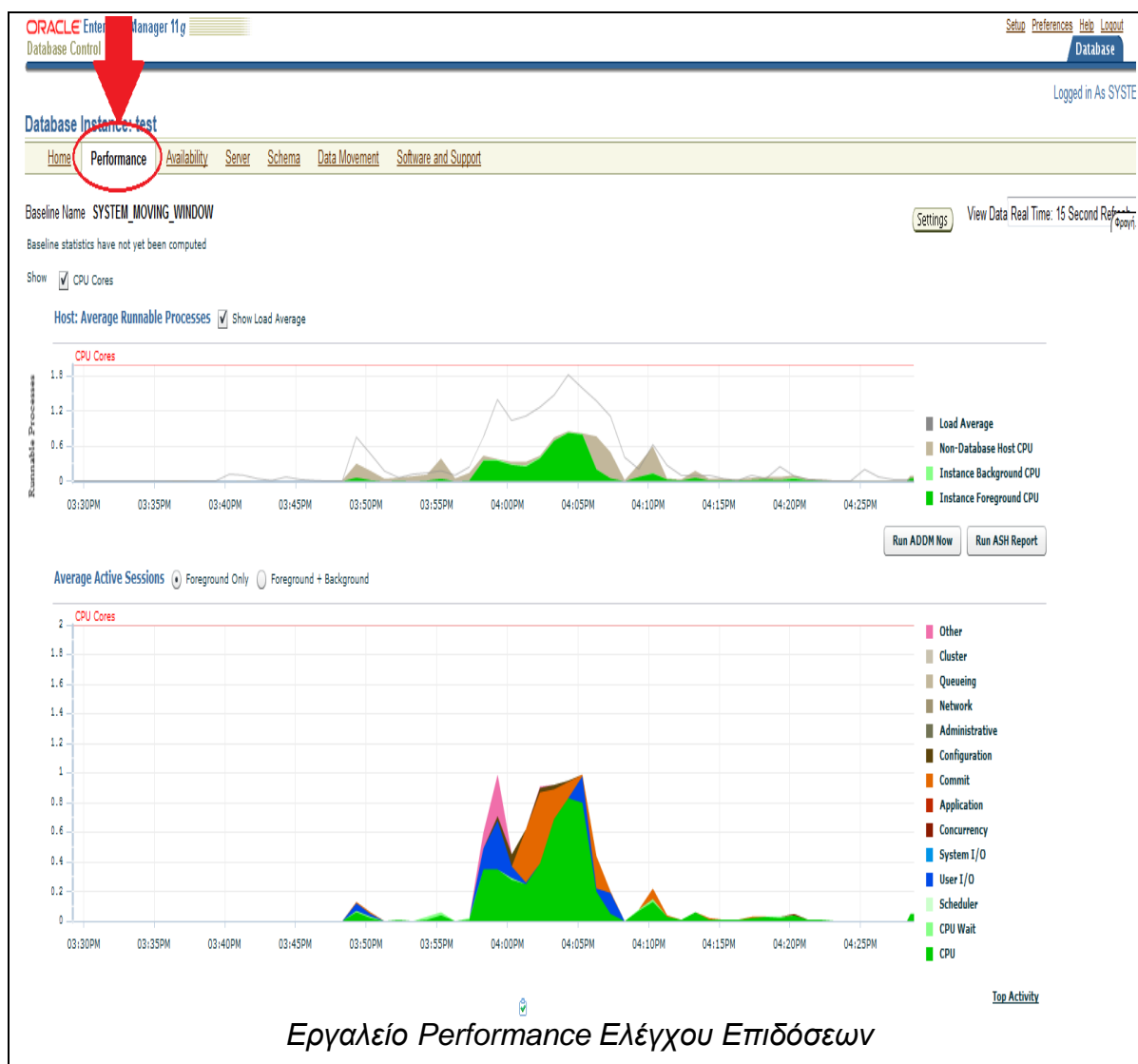
Keep in mind...

Η σύνδεση με το EM γίνεται με πρωτόκολλο HTTPS οπότε πιθανόν να σας ζητήσει να αποδεχτείτε πιστοποιητικό ασφαλείας. Επίσης, η σύνδεσή σας είναι ασφαλής από τυχών παρακολούθηση (eavesdropping).

5.2.2.3 – Χρήσιμα εργαλεία

Εδώ θα δούμε μαζί τα πιο χρήσιμα εργαλεία που μπορείτε να χρησιμοποιήσετε μέσα από το EM.

Για **παρακολούθηση Επιδόσεων** του Server χρησιμοποιείτε το εργαλείο **Performance**.



Το συγκεκριμένο εργαλείο σας δείχνει τον τρέχοντα φόρτο του CPU του Server και μάλιστα ανά χρονική περίοδο, μέσω ενός κομψού γραφήματος. Το γράφημα είναι διαδραστικό, οπότε πατώντας κλικ σε μια περιοχή που απεικονίζει, απομονώνεστε εκεί. Χρήσιμο εργαλείο για περιπτώσεις πχ που ο Server δεν αποκρίνεται όπως θα έπρεπε ή για εκτίμηση του φόρτου σε περιόδους έντονης χρήσης.

Για να **εργαστείτε στην βάση**, χρησιμοποιήστε το SQL Worksheet που βρίσκεται στην ενότητα **Related Links** → **SQL Worksheet**.

Logged in as SYSDBA

SQL Worksheet : test

Enter a SQL statement to execute. If there are multiple statements, the location of the cursor or a highlighted statement determines which will be executed. Statements should be separated with blank lines.

SQL Commands

```
SELECT * FROM SYS.BOATS WHERE ID > 0
```

Use bind variables for execution
 Auto commit
 Allow only SELECT statements

Last Executed SQL

```
SELECT * FROM SYS.BOATS WHERE ID > 0
```

Last Execution Details

[SQL Repair Advisor](#) | [SQL Details](#) | [Schedule SQL Tuning Advisor](#)

Execution Time (seconds) 0.0020

ID	NAME
1	Mariza
2	Eliza
3	Mariena
4	Theoskepasti

[SQL Repair Advisor](#) | [SQL Details](#) | [Schedule SQL Tuning Advisor](#)

To εργαλείο SQL Worksheet

Το SQL Worksheet είναι ένα Web Based SQL Περιβάλλον εργασίας που θυμίζει πολύ το Oracle Application Express (θα το δούμε παρακάτω). Σε αυτό μπορείτε να εκτελείτε Queries και Μεταβολές στην βάση δεδομένων μέσω SQL και PL/SQL Εντολών. Να έχετε υπόψη σας ότι τα σύμβολα « / » για τον διαχωρισμό των Blocks δεν χρειάζονται και πρέπει να είναι κολλητά χωρίς κενή γραμμή να τα χωρίζει. Μπορείτε επίσης να ορίσετε να εκτελούνται μονάδα SQL Queries από το Checkbox που υπάρχει στην επάνω δεξιά περιοχή εκτέλεση των εντολών σας.

Για **πληροφορίες και διαχείριση** αναφορικά με αντικείμενα, χρήστες διαθέσιμου χώρου από τα Tablespace κλπ, χρησιμοποιήστε τα εργαλεία Server και Schema. Παρακάτω βλέπουμε στιγμιότυπα από κάποια εργαλεία:

Πτυχιακή Εργασία Αλαγκιοζίδη Ελευθέριου - Πέτρου

ORACLE Enterprise Manager 11g Database Control

Database Instance: test

Tablesaces

Search

Enter an object name to filter the data that is displayed in your results set.

Object Name

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.

Selection Mode:

Actions: Add Datafile

Select Name	Allocated Size(MB)	Space Used(MB)	Allocated Space Used(%)	Auto Extend	Allocated Free Space(MB)	Status	Datafiles Type	Extent Management	Segment Management
FLOW_1044614628194104	50.1	2.5	5.0	NO	47.6	✓	1 PERMANENT	LOCAL	AUTO
SYSAUX	510.0	483.8	94.9	YES	26.2	✓	1 PERMANENT	LOCAL	AUTO
SYSTEM	680.0	671.3	98.7	YES	8.7	✓	1 PERMANENT	LOCAL	MANUAL
TEMP	20.0	0.0	0.0	YES	20.0	✓	1 TEMPORARY	LOCAL	MANUAL
UNDOTBS1	295.0	11.1	3.8	YES	283.9	✓	1 UNDO	LOCAL	MANUAL
USERS	5.0	1.6	32.5	YES	3.4	✓	1 PERMANENT	LOCAL	AUTO

Total Allocated Size (GB) 1.52
Total Used (GB) 1.14
Total Allocated Free Space (GB) 0.38

✓ Online ✗ Offline Ⓜ Read Only

Database | Setup | Preferences | Help | Logout

Copyright © 1996, 2009. Oracle. All rights reserved.

Το εργαλείο Server → Storage → Tablespaces

Το παραπάνω χρησιμοποιείται για τον έλεγχο του διαθέσιμου και του δεσμευμένου χώρου στα Tablespaces της βάσης.

Search

Enter an object name to filter the data that is displayed in your results set.

Object Name

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.

Selection Mode:

Actions: Create Like

Previous 1:25 of 34 Next 9

Select UserName	Account Status	Expiration Date	Default Tablespace	Temporary Tablespace	Profile	Created	User Type
ANONYMOUS	EXPIRED	15-Aug-2009 00:49:40 EEST	SYSAUX	TEMP	DEFAULT	15-Aug-2009 00:29:48 EEST	LOCAL
APEX_PUBLIC_USER	EXPIRED & LOCKED	15-Aug-2009 00:49:40 EEST	USERS	TEMP	DEFAULT	15-Aug-2009 00:43:14 EEST	LOCAL
APEX_030200	EXPIRED & LOCKED	15-Aug-2009 00:49:40 EEST	SYSAUX	TEMP	DEFAULT	15-Aug-2009 00:43:14 EEST	LOCAL
APPOSSYS	EXPIRED & LOCKED	15-Aug-2009 00:24:20 EEST	SYSAUX	TEMP	DEFAULT	15-Aug-2009 00:24:20 EEST	LOCAL
CTXSYS	EXPIRED & LOCKED	21-Nov-2013 16:24:51 EET	SYSAUX	TEMP	DEFAULT	15-Aug-2009 00:29:26 EEST	LOCAL
DBSNMP	OPEN	20-May-2014 16:26:12 EEST	SYSAUX	TEMP	MONITORING_PROFILE	15-Aug-2009 00:24:19 EEST	LOCAL
DIP	EXPIRED & LOCKED	15-Aug-2009 00:19:11 EEST	USERS	TEMP	DEFAULT	15-Aug-2009 00:19:11 EEST	LOCAL
EXFSYS	EXPIRED & LOCKED	15-Aug-2009 00:29:14 EEST	SYSAUX	TEMP	DEFAULT	15-Aug-2009 00:29:14 EEST	LOCAL
FLWS_FILES	EXPIRED & LOCKED	15-Aug-2009 00:49:40 EEST	SYSAUX	TEMP	DEFAULT	15-Aug-2009 00:43:14 EEST	LOCAL
LELERIS	OPEN	21-May-2014 19:04:24 EEST	USERS	TEMP	DEFAULT	22-Nov-2013 13:15:30 EET	LOCAL
MDDATA	EXPIRED & LOCKED	15-Aug-2009 00:49:40 EEST	USERS	TEMP	DEFAULT	15-Aug-2009 00:36:46 EEST	LOCAL
MDSYS	EXPIRED & LOCKED	15-Aug-2009 00:31:31 EEST	SYSAUX	TEMP	DEFAULT	15-Aug-2009 00:31:31 EEST	LOCAL
MGMT_VIEW	OPEN	21-May-2014 16:05:28 EEST	SYSTEM	TEMP	DEFAULT	22-Nov-2013 16:03:57 EET	LOCAL
OLAPSYS	EXPIRED & LOCKED	15-Aug-2009 00:36:04 EEST	SYSAUX	TEMP	DEFAULT	15-Aug-2009 00:36:04 EEST	LOCAL
ORACLE_OCM	EXPIRED & LOCKED	15-Aug-2009 00:19:53 EEST	USERS	TEMP	DEFAULT	15-Aug-2009 00:19:53 EEST	LOCAL
ORDDATA	EXPIRED & LOCKED	15-Aug-2009 00:31:31 EEST	SYSAUX	TEMP	DEFAULT	15-Aug-2009 00:31:31 EEST	LOCAL
ORDPLUGINS	EXPIRED & LOCKED	15-Aug-2009 00:31:31 EEST	SYSAUX	TEMP	DEFAULT	15-Aug-2009 00:31:31 EEST	LOCAL
ORDSYS	EXPIRED & LOCKED	15-Aug-2009 00:31:31 EEST	SYSAUX	TEMP	DEFAULT	15-Aug-2009 00:31:31 EEST	LOCAL
OUTLN	EXPIRED & LOCKED	15-Aug-2009 00:49:40 EEST	SYSTEM	TEMP	DEFAULT	15-Aug-2009 00:17:14 EEST	LOCAL
OWBSYS	OPEN	21-May-2014 13:14:00 EEST	USERS	TEMP	DEFAULT	15-Aug-2009 00:49:36 EEST	LOCAL
OWBSYS_AUDIT	EXPIRED & LOCKED	15-Aug-2009 00:49:40 EEST	SYSAUX	TEMP	DEFAULT	15-Aug-2009 00:49:37 EEST	LOCAL

Το εργαλείο Server → Security → Users

Το παραπάνω εργαλείο επιτρέπει την διαχείριση των χρηστών της βάσης δεδομένων.

Tables

Object Type Table

Search

Enter a schema name and an object name to filter the data that is displayed in your results set.

Schema SYS

Object Name

Go

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.

Selection Mode Single

Create

Edit View Delete With Options Actions Create Like Go Previous 1-25 of 1805 Next 25

Select	Schema	Table Name	Tablespace	Partitioned	Rows Last Analyzed
<input checked="" type="radio"/>	SYS	ACCESS\$	SYSTEM	NO	103084 22-Nov-2013 16:04:50 EET
<input type="radio"/>	SYS	ALERT_QT	SYSAUX	NO	
<input type="radio"/>	SYS	APPLY\$_CHANGE_HANDLERS	SYSTEM	NO	0 15-Aug-2009 00:24:30 EEST
<input type="radio"/>	SYS	APPLY\$_CONF_HDLR_COLUMNS	SYSTEM	NO	0 15-Aug-2009 00:24:30 EEST
<input type="radio"/>	SYS	APPLY\$_CONSTRAINT_COLUMNS	SYSTEM	NO	0 15-Aug-2009 00:24:30 EEST
<input type="radio"/>	SYS	APPLY\$_DEST_OBJ	SYSTEM	NO	0 15-Aug-2009 00:24:30 EEST
<input type="radio"/>	SYS	APPLY\$_DEST_OBJ_CMAP	SYSTEM	NO	0 15-Aug-2009 00:24:30 EEST
<input type="radio"/>	SYS	APPLY\$_DEST_OBJ_OPS	SYSTEM	NO	0 15-Aug-2009 00:24:30 EEST
<input type="radio"/>	SYS	APPLY\$_ERROR	SYSAUX	NO	0 15-Aug-2009 00:24:30 EEST
<input type="radio"/>	SYS	APPLY\$_ERROR_HANDLER	SYSTEM	NO	0 15-Aug-2009 00:24:30 EEST
<input type="radio"/>	SYS	APPLY\$_ERROR_TN	SYSTEM	NO	0 15-Aug-2009 00:24:30 EEST
<input type="radio"/>	SYS	APPLY\$_SOURCE_OBJ	SYSTEM	NO	0 15-Aug-2009 00:24:30 EEST
<input type="radio"/>	SYS	APPLY\$_SOURCE_SCHEMA	SYSTEM	NO	0 15-Aug-2009 00:24:30 EEST
<input type="radio"/>	SYS	ΔΡΑΣΕΙΣ ΜΕΤΑΒΙΒΑΣΗΣ ΠΡΟΤΥΠΩΝ	SYSTEM	NO	0 15-Aug-2009 00:24:30 EEST

Το εργαλείο Schema → Database Objects → Tables

Το παραπάνω εργαλείο χρησιμεύει για την διαχείριση των πινάκων που υπάρχουν στην βάση.

Αυτά είναι μόνο μερικά από τα πολλά εργαλεία που περιέχει το EM, μπορείτε να τα εξερευνήσετε άφοβα. Σίγουρα θα μείνετε ευχαριστημένοι βλέποντας πόσες ευκολίες και δυνατότητες προσφέρει.

5.2.2.4 – Αποσύνδεση και Τερματισμός του EM

Για να αποσυνδεθείτε από το EM θα πρέπει να πατήσετε Logout από την πάνω-δεξιά γωνία της σελίδας της εφαρμογής, όπως παρακάτω:

Setup Preferences Help **Logout**

Database

Logged in As SYS

Αποσύνδεση από το EM

Αν επιθυμείτε να τερματίσετε την υπηρεσία του EM τότε θα πρέπει να μεταβείτε στον κατάλογο ORACLE_HOME/bin και να τρέξετε την ακόλουθη εντολή για να το πράξετε:

```
oracle@debserver$ ./emctl stop dbconsole
```

Η διεργασία του EM σταματάει, τερματίζοντας τον Agent και την Κονσόλα του EM. Πλέον συνδέσεις στο EM δεν μπορούνε να γίνουνε.

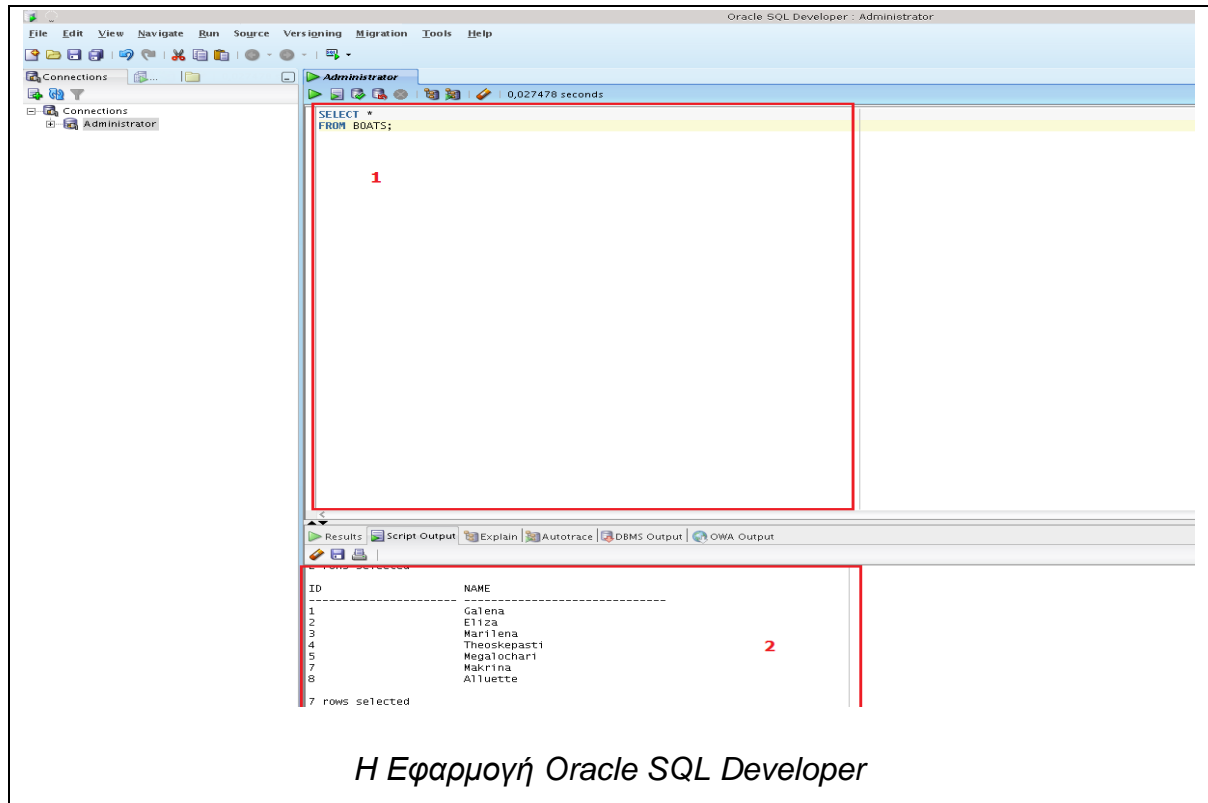
5.2.2.5 – Συμπέρασμα

Το Enterprise Manager είναι πραγματικά ένα καταπληκτικό εργαλείο που δίνει στον διαχειριστή ένα ισχυρό πλεονέκτημα όσον αφορά την σωστή και απρόσκοπτη λειτουργία του Server. Με την δυνατότητες και τις διευκολύνσεις που παρέχει, και πάνω από όλα το γεγονός της εύκολης προσπελασιμότητάς του, το κάνουν και αυτό απαραίτητο συμπλήρωμα στην εργαλειοθήκη του διαχειριστή.

5.2.3 – Το περιβάλλον Oracle SQL Developer

Όπως το κάθε DBMS που σέβεται τον εαυτό του, έτσι και η Oracle δεν θα μπορούσε να μην παρέχει και αυτή ένα ολοκληρωμένο περιβάλλον εργασίας και διαχείρισης για το δικό της DBMS. Και αυτό είναι το Oracle SQL Developer.

Το Oracle SQL Developer είναι μια **Java** Εφαρμογή, με την οποία ο διαχειριστής (και ο απλός χρήστης όπως θα δούμε στο επόμενο κεφάλαιο) μπορεί να εκτελέσει πληθώρα εντολών και Scripts στον Server και να προκαλέσει το ζητούμενο αποτέλεσμα. Παρακάτω βλέπουμε την κεντρική οθόνη του SQL Developer όπου διακρίνουμε τον χώρο συγγραφή εντολών (SQL Worksheet) (1) και την οθόνη έκβασης και αποτελεσμάτων (2).



Το SQL Developer αποτελεί πολύ καλή επιλογή σαν εργαλείο διαχείρισης ειδικότερα για όσους θεωρούνε δύσχρηστο το SQL*PLUS. Με το SQL Developer μπορείτε να εκτελέσετε οποιαδήποτε άλλη εργασία διαχείρισης εκτελούσατε με το SQL*PLUS (πλην τερματισμού / εκκίνησης του Instance της βάσης και τροποποίησης της κατάστασής της). Και το πρόσθετο ατού που έχει είναι ότι είναι **γραφική** (GUI) εφαρμογή. Θέλετε να ακούσετε και το ακόμα καλύτερο; Είναι **προεγκατεστημένο** με την Oracle 11g αν διαχειρίζεστε τον Server σας τοπικά ή απομακρυσμένα μέσω RDP. Πάντα όμως μπορείτε να το εγκαταστήσετε σε άλλο υπολογιστή/πελάτη για απομακρυσμένη διαχείριση (για εγκατάσταση σε Client Υπολογιστή, μεταβείτε στο παράρτημα Β').

5.2.3.1 – Εκκίνηση Oracle SQL Developer

Το εκτελέσιμο αρχείο του SQL Developer βρίσκεται στον κατάλογο ORACLE_HOME/sqldeveloper. Ως χρήστες Oracle, μεταβείτε σε αυτόν τον κατάλογο και εκτελέστε το executable file.

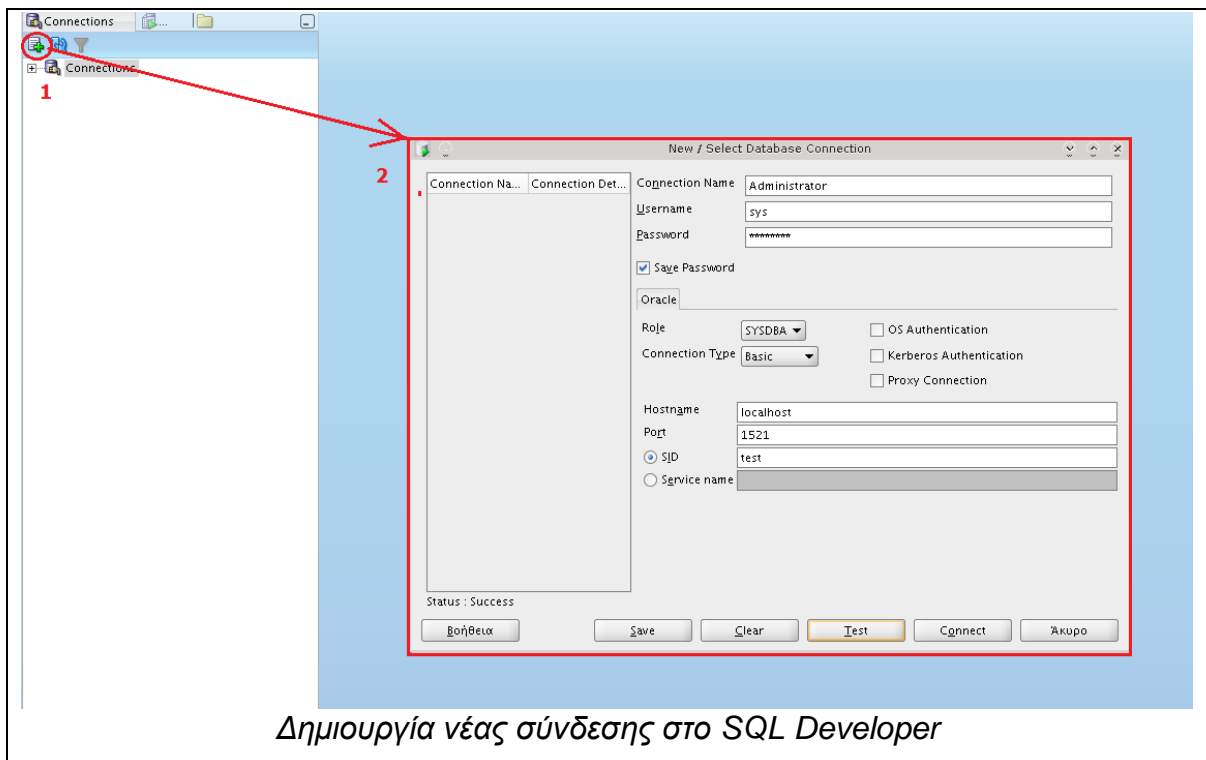
```
oracle@debserver$ cd $ORACLE_HOME/sqldeveloper
```

```
oracle@debserver$ ./sqldeveloper.sh
```

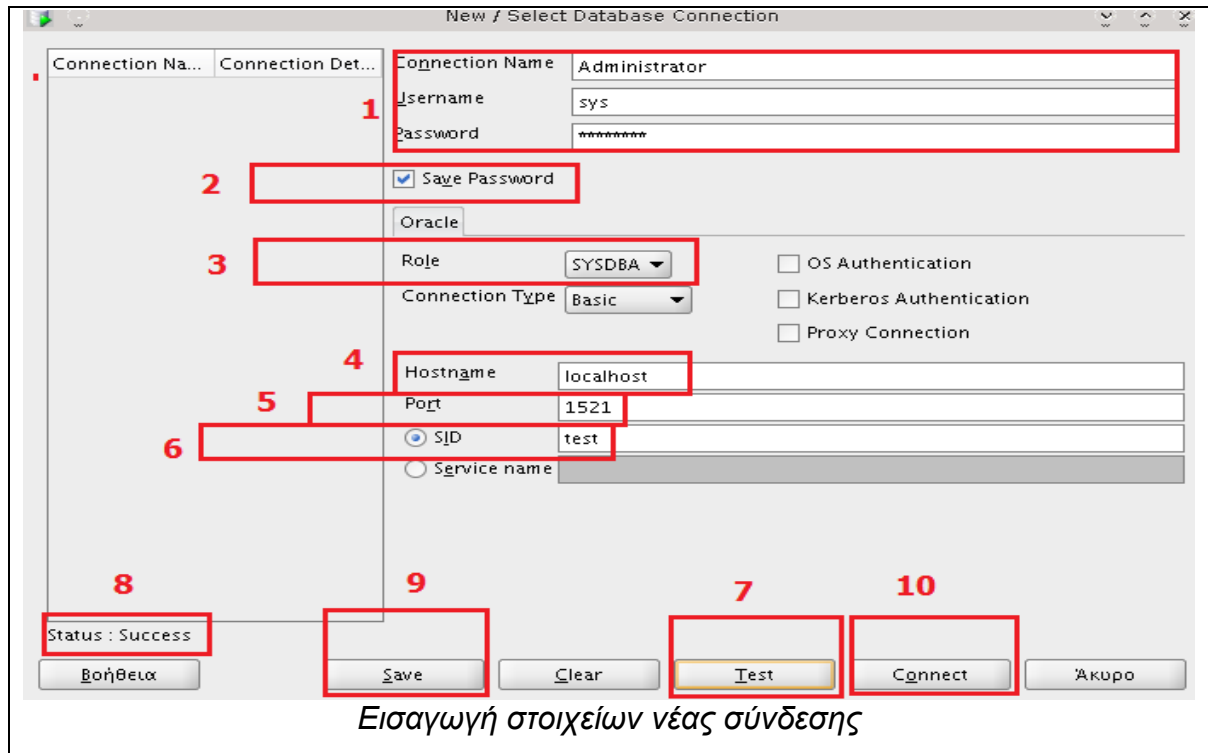
Σχεδόν αμέσως εμφανίζεται το Splash Screen της εφαρμογής και σε ελάχιστο διάστημα φορτώνεται στην οθόνη σας. Από κει και πέρα το μόνο που χρειάζεται να κάνετε για να εργαστείτε/διαχειριστείτε την βάση, είναι να συνδεθείτε σε αυτήν.

5.2.3.2 – Σύνδεση Διαχειριστή

Αφού εκκινηθεί η εφαρμογή θα πρέπει να μεταβείτε στην αριστερή περιοχή «Connections» και να δημιουργήσετε σύνδεση πατώντας στο πλήκτρο «New Connection» (1) για να σας εμφανίσει το παράθυρο παραμετροποίησης της νέας σύνδεσης (2). Παρακάτω βλέπουμε πως:



Θα πρέπει να εισάγετε τα στοιχεία σας ως διαχειριστής για σύνδεση στην κατάλληλη βάση δεδομένων. Παρακάτω βλέπουμε τα βήματα:

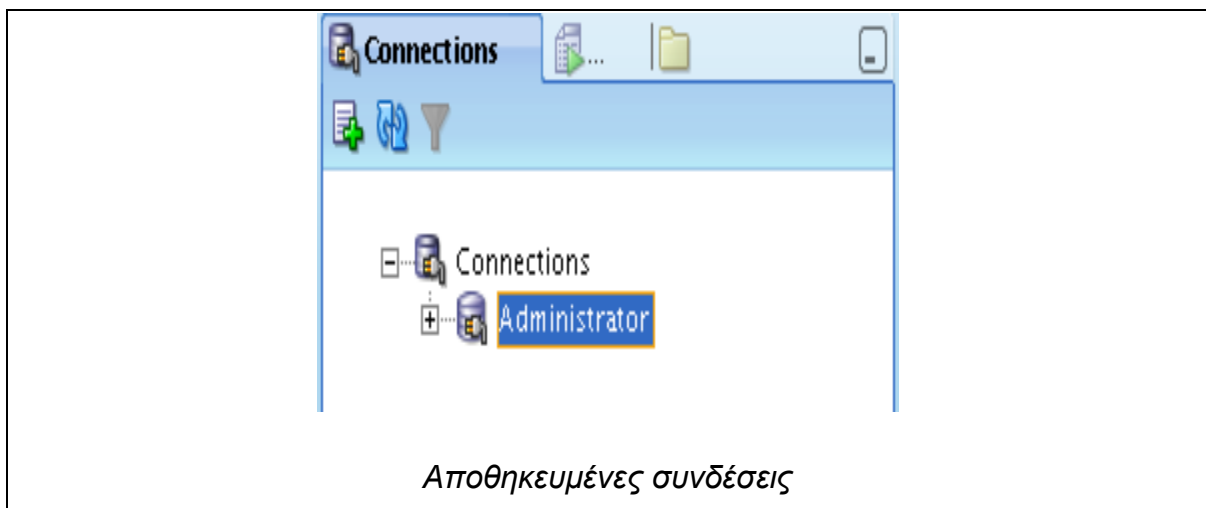


Βήματα σύνδεσης:

1. Πρώτα εισάγετε τα **διαπιστευτήριά** σας ως διαχειριστής (username και password πχ του χρήστη SYS) και δώστε ένα όνομα ως αναγνωριστικό της σύνδεσης (πχ Administrator) (1).
2. Αν ο Server σας / client υπολογιστής σας, είναι προστατευμένος από «αδιάκριτα μάτια» μπορείτε να **αποθηκεύσετε το Password** επιλέγοντας «Save Password» (2).
3. Στην συνέχεια επιλέγετε τον **ρόλο διαχειριστή** που είναι ο **SYSDBA** (SYS as SYSDBA) (3).
4. Σαν «Hostname» δίνετε την διεύθυνση **Localhost** αν τρέχετε την εφαρμογή από τον Server ή την **διεύθυνση** του Server να τρέχετε την εφαρμογή από άλλο υπολογιστή (4).

5. Κατόπιν δίνετε την θύρα που ακούει ο Listener της βάσης. Από προεπιλογή είναι η 1521, εκτός αν επιλέξατε διαφορετική οπότε εισάγετέ την (5).
6. Το κριτήριο επιλογής σε ποια βάση θα συνδεθείτε, είναι το Instance ID (SID) της, οπότε στο πεδίο «SID» εισάγετε το (6).
7. Ολοκληρώνοντας, **δοκιμάστε** αν η σύνδεση δουλεύει, πατώντας στο κουμπί «Test» (7) και θα πρέπει να λάβετε το μήνυμα «Success» στη λεζάντα «Status» (8). Αν δεν λάβετε success, θα λάβετε σφάλμα, το οποίο στις περισσότερες φορές σας λέει από μόνο του τι δεν πάει καλά. **Αποθηκεύετε** την σύνδεσή σας πατώντας «Save» (9) και τελικώς **συνδεθείτε** πατώντας «Connect» (10).

Η σύνδεση επιτυγχάνεται και πλέον είστε μέσα στην βάση. Η διαδικασία αυτή δεν χρειάζεται να την επαναλάβετε γιατί οι παράμετροι της σύνδεσής σας έχουν αποθηκευτεί στην καρτέλα Connections όπως βλέπουμε παρακάτω:



Για σύνδεση/αποσύνδεση απλά πατάτε δεξί-κλικ στο όνομα της σύνδεσης και πατάτε «Connect» ή «Disconnect» αντίστοιχα.

5.2.3.3 – Δυνατότητες Διαχείρισης

Στο SQL Developer μπορείτε να κάνετε σχεδόν οτιδήποτε μπορείτε να κάνετε και στο SQL*PLUS, εκτός από λειτουργίες ελέγχου της βάσης (κατάσταση, λειτουργία). Ποιο αναλυτικά:

- Διαχείριση χρηστών (CREATE/ALTER/DROP USER, GRANT, REVOKE κλπ).
- Διαχείριση Ρόλων και δικαιωμάτων (CREATE/ALTER/DROP ROLE, GRANT/REVOKE κλπ)
- Διαχείριση αντικειμένων (CREATE/ALTER/DROP TABLESPACE/DIRECTORY κλπ).
- Ρύθμιση παραμέτρων Server (πχ SET SERVEROUTPUT ON)
- Διαχείριση Συνεδρίας και Χαρακτηριστικών Βάσης Δεδομένων (**εκτός** από αλλαγή κατάστασης πχ ALTER SESSION, ALTER DATABASE DATAFILE κλπ).

Το SQL Developer παρέχει πλήρη υποστήριξη εντολών SQL και PL/SQL (εκτός από το ρεπερτόριο εντολών του SQL*PLUS όπως edit, run κλπ). Διαθέτει επίσης και εύχρηστο Query Builder το οποίο «λύνει χέρια» σε πολύπλοκα Queries, καθώς επίσης και την δυνατότητα εκτέλεσης και αποθήκευσης Scripts. Θα δούμε το SQL Developer με περισσότερη λεπτομέρεια στην λειτουργία του στο κεφάλαιο 7. Προς το παρόν περιοριζόμαστε σε αυτό μόνον ως εργαλείο διαχείρισης.

5.2.3.4 – Συμπέρασμα

Το SQL Developer είναι ένα καταπληκτικό εργαλείο διαχείρισης της βάσης σας καθώς είναι περισσότερο εύχρηστο και με περισσότερες δυνατότητες από το

SQL*PLUS. Οι ευκολίες που παρέχει καθώς και η ελκυστική του εμφάνιση, είναι σίγουρο ότι θα γίνει το «δεξί χέρι» σας για την διαχείριση της βάσης σας.

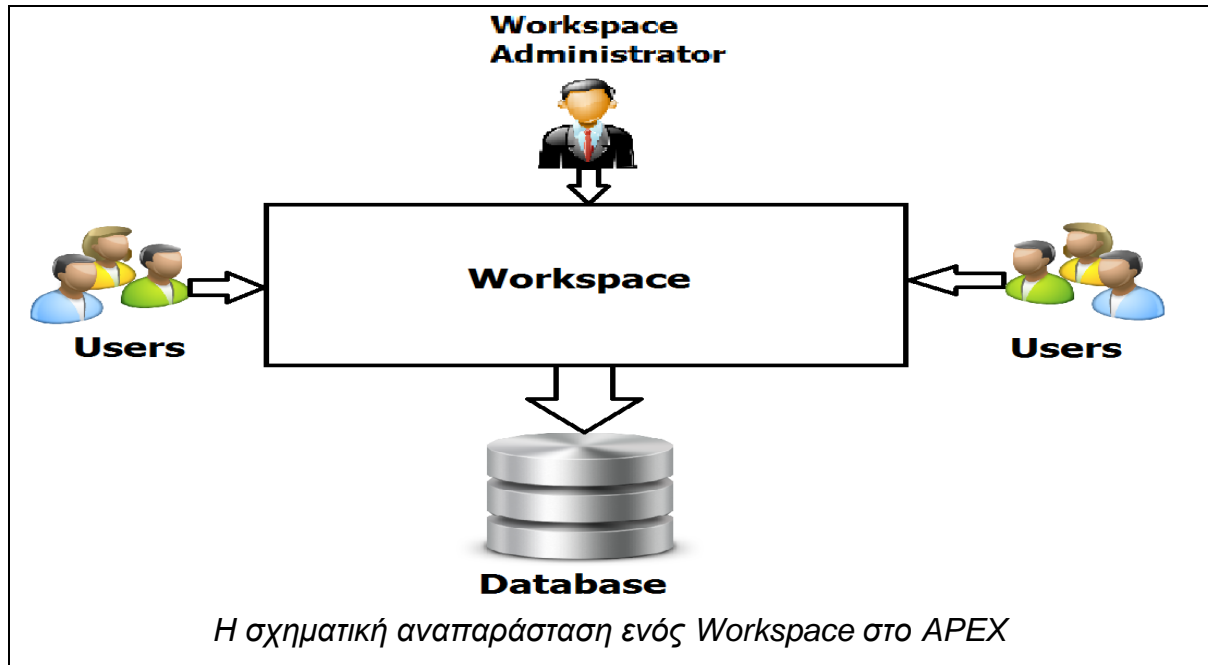
5.2.4 – Το περιβάλλον Oracle Application Express (APEX)

Η Oracle, πρωτοπόρος στον χώρο των βάσεων δεδομένων, δημιούργησε ένα Web Based περιβάλλον εργασίας, το **Oracle Application Express** ή **APEX** για συντομία. Πρόκειται για μια ολόκληρη σουίτα εργασίας στη βάση δεδομένων η οποία το μόνο που χρειάζεται να έχει κάποιος για να την χρησιμοποιήσει, είναι έναν πλοηγό ιστού (Web Browser).

5.2.4.1 – Τι είναι το APEX και πως λειτουργεί

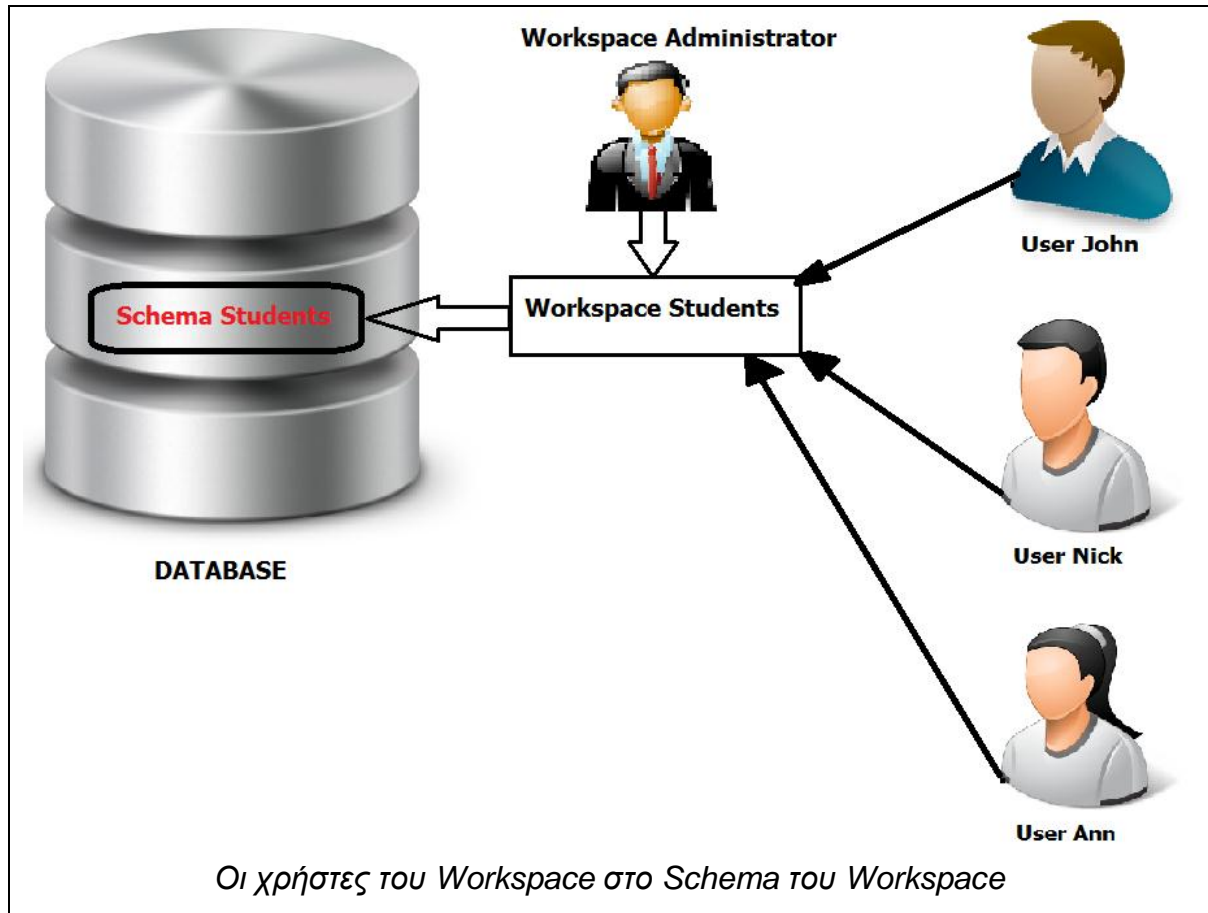
Το Oracle APEX είναι μια πλήρης και διαχειρήσιμη web based πλατφόρμα (σε JSP για την ακρίβεια) εργασίας σε μια βάση δεδομένων της Oracle όπου οι χρήστες και οι διαχειριστές εργάζονται στην βάση μέσα από τον Web Browser τους. Θυμίζει αρκετά το Oracle Enterprise Manager, αλλά το τελευταίο είναι αποκλειστικά περιβάλλον διαχείρισης. Το APEX αποτελεί κυρίως περιβάλλον εργασίας.

Η λειτουργία του APEX είναι λίγο ιδιαίτερη και αξίζει να την συζητήσουμε. Οι χρήστες εργάζονται σε ένα περιβάλλον εργασίας το οποίο ονομάζεται **Workspace**. Το κάθε Workspace διαθέτει τουλάχιστον έναν **διαχειριστή**, ενώ οι **χρήστες** που εργάζονται στην βάση, συνήθως ανήκουν σε ένα κοινόχρηστο Workspace. Στο παρακάτω σχήμα βλέπουμε την αναπαράσταση ενός Workspace.



Το Workspace λοιπόν είναι η κοινόχρηστη ομάδα εργασίας των χρηστών. OK αυτό το καταλάβαμε. Για την βάση δεδομένων όμως τι είναι ένα Workspace?

Η πραγματική φύση ενός Workspace του APEX στην βάση, δεν είναι τίποτα περισσότερο από ένα **Schema**. Όπως κάθε χρήστης της βάσης έχει το δικό του Schema έτσι και το Workspace του APEX έχει το δικό του. Και το εύλογο ερώτημα που θα μπορούσε κάποιος να κάνει είναι, οι χρήστες της βάσης εργάζονται σε αυτό το Schema? Η απάντηση όμως είναι **όχι**. **Οι χρήστες του APEX Workspace δεν έχουν καμία σχέση με τους χρήστες της βάσης**. Αν μπερδεύεστε δείτε το παρακάτω σχήμα για να καταλάβετε:



Οι χρήστες της βάσης λοιπόν έχουν το δικό τους Schema ενώ οι χρήστες του Workspace εργάζονται στο Schema του Workspace. Αν όμως οι χρήστες που εργάζονται στο Workspace είναι τα ίδια πρόσωπα που εργάζονται και στην βάση, δεν μπορούν να έχουν πρόσβαση στο Schema τους, από το APEX; Φυσικά και γίνεται, και αυτό θα κάνουμε παρακάτω. Να σημειωθεί πως και το APEX έχει το δικό του Schema το οποίο είναι το APEX_030200.

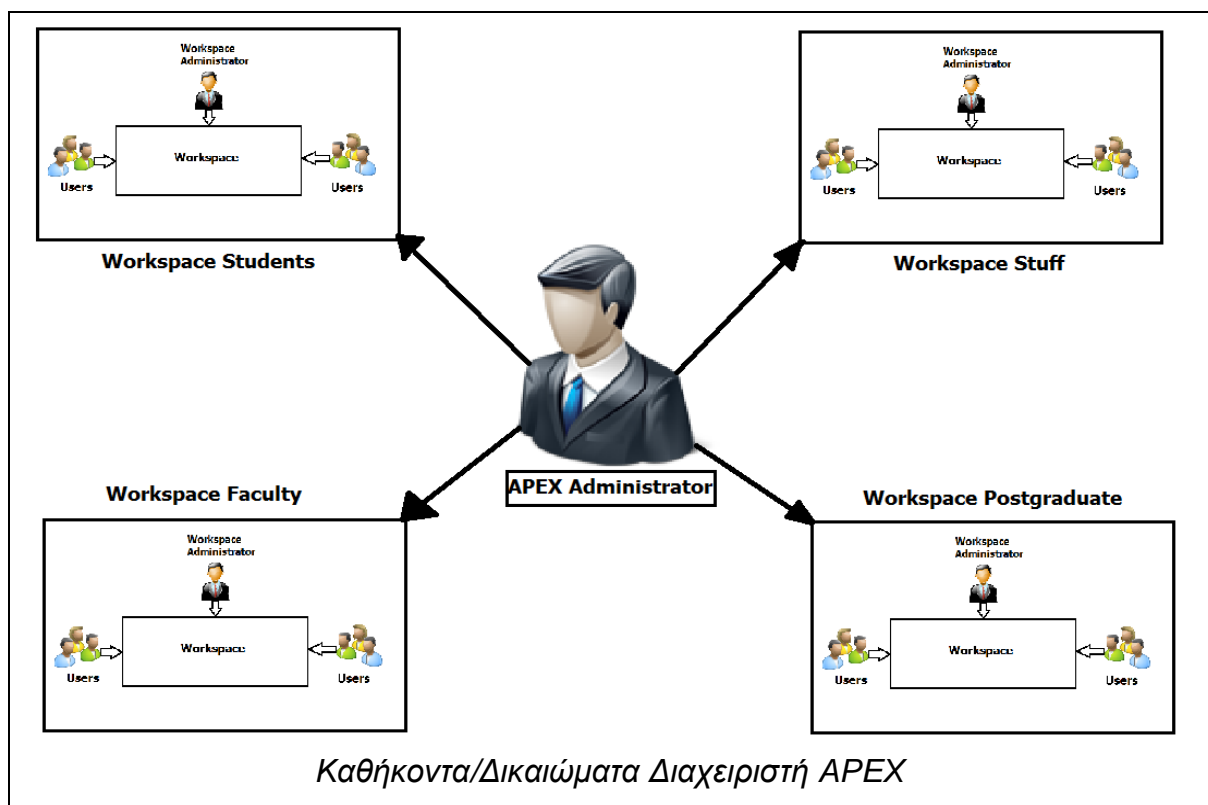
5.2.4.2 – Οι διαχειριστές του APEX

Κατ' αρχάς να ξεκαθαρίσουμε κάτι. Όπως είπαμε για τους χρήστες του Workspace, το ότι δηλαδή δεν έχουν καμία σχέση με τους χρήστες της βάσης, το ίδιο ισχύει και για τους διαχειριστές του APEX. Με άλλα λόγια ο χρήστης SYS as SYSDBA, SYSTEM, SYSMAN κλπ δεν έχουν καμία σχέση με την διαχείριση του APEX (μπορούν όμως να το διαχειρίζονται έμμεσα).

Οι διαχειριστές του APEX διαχωρίζονται σε δύο κατηγορίες οι οποίες είναι οι παρακάτω:

- 1) Γενικοί Διαχειριστές του APEX (APEX Administrators)
- 2) Διαχειριστές Workspace (Workspace Administrators)

Οι **γενικοί διαχειριστές του APEX** είναι αυτοί που διαχειρίζονται τα πάντα, τους χρήστες, όλα τα Workspaces και ενέργειες παρακολούθησης ολόκληρου του περιβάλλοντος του APEX. Στο παρακάτω σχήμα βλέπουμε το ρόλο τους και τους.



Οι διαχειριστές του APEX έχουν τον απόλυτο έλεγχο σε ολόκληρο το περιβάλλον του APEX. Μπορούν να δημιουργούν Workspaces και να τα τροποποιούν (συμπεριλαμβανομένου και διαγραφή τους), να δημιουργούν χρήστες και διαχειριστές Workspaces. Μπορεί να υπάρχει **μόνον ένας**, ανά βάση δεδομένων.

Οι διαχειριστές των Workspaces από την άλλη, περιορίζονται μόνον σε δικαιώματα διαχείρισης ενός Workspace. Οι διαχειριστές αυτοί δημιουργούνται από τον γενικό διαχειριστή του APEX και μπορούν να δημιουργούν και να συσχετίζουν χρήστες και Schemas στο Workspace που διαχειρίζονται. Έξω από αυτό, δεν έχουν καμία απολύτως ισχύ. Οι διαχειριστές Workspace μπορούν να δημιουργούν πρόσθετους Workspace Administrators, για το ίδιο Workspace.

5.2.4.3 – Ενεργοποίηση του APEX.

Με την δημιουργία μιας βάσης δεδομένων, από προεπιλογή το APEX είναι απενεργοποιημένο. Για να το ενεργοποιήσουμε θα πρέπει να επισκεφτούμε ξανά το SQL*PLUS και να συνδεθούμε στην βάση ως χρήστες SYS as SYSDBA. Η ενεργοποίηση του APEX θα γίνει με την εκτέλεση ενός Script, του «apxconf.sql» που βρίσκεται στον κατάλογο ORACLE_HOME/apex. Γράφουμε τις παρακάτω εντολές για να ενεργοποιήσουμε το APEX.

```
oracle@debserver$ cd $ORACLE_HOME/bin
oracle@debserver$ ./sqlplus sys as sysdba
Enter Password:
SQL> @../apex/apxconf.sql
Enter a password for the admin user [] #-- Εισάγετε κωδικό γενικού διαχειριστή
Enter a port for the XDB HTTP Listener [ 8080] #Default HTTP Port
```

Ενεργοποιώντας το APEX θα σας ζητήσει να εισάγετε ένα **προσωρινό** κωδικό πρόσβασης για τον γενικό διαχειριστή και να επιλέξετε σε ποια HTTP Port θα «ακούει» το APEX. Παρατηρούμε πως μοιράζεται την ίδια Port με το αποθετήριο XML (XDB) της βάσης δεδομένων. Το Password είναι προσωρινό και θα σας ζητήσει να το αλλάξετε όταν συνδεθείτε.

Η διαδικασία αυτή ενεργοποιεί το APEX το οποίο όμως ακόμα δεν είναι έτοιμο για χρήση για τον εξής λόγο. Όταν κάνουμε αίτηση να συνδεθούμε στο

APEX από τον Browser μας, η βάση δεδομένων μας βλέπει σαν χρήστη ANONYMOUS. Ο συγκεκριμένος λογαριασμός όμως είναι κλειδωμένος, οπότε κάθε φορά που θα συνδεόμαστε στο APEX θα μας ζητάει Credentials για σύνδεση στο αποθετήριο XML της βάσης, κάτι το οποίο είναι ενοχλητικό και άσχετο με το APEX. Θα πρέπει λοιπόν να ξεκλειδώσουμε τον χρήστη ANONYMOUS με την παρακάτω εντολή:

```
SQL> ALTER USER ANONYMOUS ACCOUNT UNLOCK;
```

Για να ενεργοποιηθεί πλήρως το APEX πρέπει να τερματίσουμε το Instance της βάσης, τον Listener και να τα ενεργοποιήσουμε ξανά. Δώστε τις παρακάτω εντολές:

```
SQL> SHUT IMMEDIATE;

Database Closed.

Database dismounted.

SQL>EXIT

oracle@debserver$ ./dbshut $ORACLE_HOME

oracle@debserver$ ./dbstart $ORACLE_HOME

oracle@debserver$ ./sqlplus sys as sysdba

Enter password:

Connected to an idle instance.

SQL> STARTUP
```

Με την επανεκκίνηση στην ουσία, της βάσης, η υπηρεσία του Oracle APEX είναι έτοιμη για χρήση. Πλέον μπορούμε να συνδεθούμε σε αυτό για να το παραμετροποιήσουμε.

5.2.4.4 – Σύνδεση Γενικού Διαχειριστή

Για να συνδεθούμε στο APEX ως γενικοί διαχειριστές, μπορούμε να το κάνουμε ανοίγοντας τον Web Browser μας και μεταβαίνοντας στην διεύθυνση:

http://<server-address>:8080/apex/apex_admin

Η φόρμα Login για τον APEX Administrator φαίνεται παρακάτω:

ORACLE Application Express

Using this interface the administrator can manage all aspects of the service including creating, removing, and managing workspaces.

Application Express Administration Services

Username

Password

Φόρμα Login για τον γενικό διαχειριστή

Τα Credentials που απαιτείτε να εισάγετε είναι, username = admin και password = <το προσωρινό password που ρυθμίσατε στην εκτέλεση του αρχconf.sql>. Εισάγοντάς τα, μπαίνετε στο σύστημα.

Εκεί μας περιμένει άλλη έκπληξη. Μας ζητάει να αλλάξουμε το Password και να βάλουμε ένα ισχυρότερο. Δεν μπορούμε να το αποφύγουμε οπότε ας του κάνουμε. Παρακάτω βλέπουμε την οθόνη αυτή:

The password for this account must be changed.

Change Password

User Name: ADMIN

* Enter Current Password

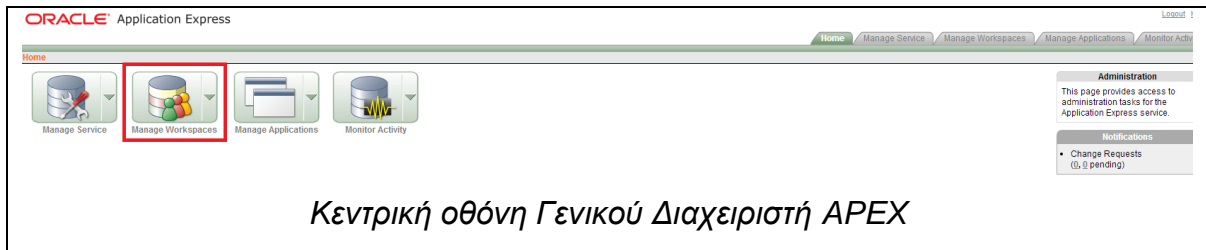
* Enter New Password

* Confirm New Password

Φόρμα αλλαγής password

Θα πρέπει να εισάγουμε ισχυρό password που να περιέχει τουλάχιστον **ένα κεφαλαίο γράμμα**, **ένα σύμβολο** (%,@,& κλπ) και **ένα νούμερο**. Και να είναι μεγαλύτερο από **έξι χαρακτήρες**. Ενδεικτικό παράδειγμα είναι το «Johnny_1987». Αφού το αλλάξετε, θα σας επιστρέψει πίσω στην φόρμα Login. Εισάγετε πάλι τα

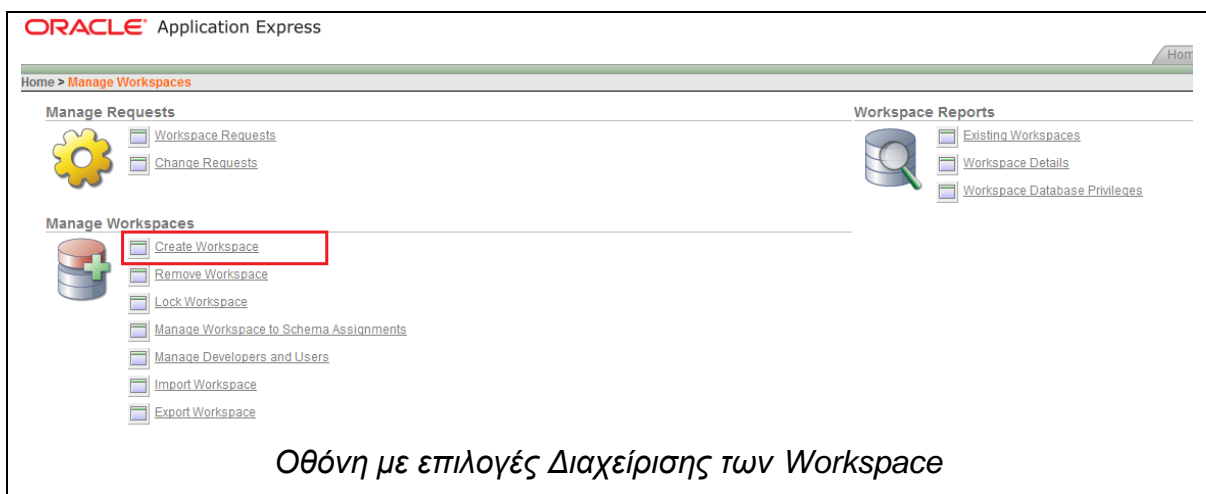
Credentials σας όπως πριν αλλά αυτή τη φορά με το Νέο Password. Επιτέλους Μπαίνουμε στο σύστημα και βλέπουμε την παρακάτω οθόνη:



Από εδώ, το APEX σας ανήκει. Έχετε πολλές επιλογές στην διάθεσή σας, αλλά προς το παρών μας ενδιαφέρει η επιλογή «Manage Workspaces» που την μαρκάρουμε με κόκκινο. Ανοίγοντάς την, μπορούμε να δημιουργήσουμε Workspaces και διαχειριστές για αυτά.

5.2.4.5 – Δημιουργία Workspace και Διαχειριστή

Μέσα από την επιλογή «Manage Workspaces» μπορούμε να δημιουργήσουμε νέα αλλά και να διαχειριστούμε υπάρχοντα Workspaces. Στην παρακάτω εικόνα βλέπουμε τις επιλογές που μας δίνει:



Εμείς δεν έχουμε ακόμα κανένα Workspace (πέραν του **Internal Workspace** στο οποίο βρισκόμαστε τώρα) οπότε πρέπει να δημιουργήσουμε ένα. Επιλέγουμε «Create Workspace» για να ξεκινήσουμε την διαδικασία της

δημιουργίας του. Στην παρακάτω εικόνα βλέπουμε το πρώτο βήμα της δημιουργίας ενός Workspace.

Βήμα 1 – Ρύθμιση αναγνωριστικών Workspace

Στην οθόνη αυτή εισάγουμε όνομα για το Workspace μας (1), μια προαιρετική περιγραφή (2) και πατάμε Next (3) για να μεταβούμε στο επόμενο βήμα, το οποίο βλέπουμε κάτωθι:

Βήμα 2 – Επιλογή Schema και Quota

Στην οθόνη αυτή επιλέγουμε το Schema που θέλουμε να συσχετίσουμε με το Workspace. Επειδή όμως θέλουμε να έχει δικό του Schema, στην επιλογή «Re-use existing schema?» επιλέγουμε «No» (1). Στα επόμενα δύο πεδία εισάγουμε το όνομα του Schema που θέλουμε να δημιουργηθεί καθώς και ένα Password γι αυτό (2). Στην συνέχεια επιλέγουμε πόσα MB χώρου θα μας εκχωρηθεί για χρήση, το γνωστό πλέον Quota (3) και τέλος, ολοκληρώνουμε το βήμα με «Next» (2) για να πάμε στο επόμενο.

Βήμα 3 – Δημιουργία Διαχειριστή Workspace

Πιο πάνω είχαμε πει πως κάθε Workspace πρέπει να έχει από έναν τουλάχιστον **διαχειριστή**. Τον διαχειριστή αυτόν τον δημιουργούμε σε αυτό το βήμα. Εισάγουμε αρχικά το Username του (καλό θα είναι να μην είναι admin) (1), τον κωδικό πρόσβασής του (2), **προαιρετικά** τα στοιχεία του (3) και υποχρεωτικά ένα email (οτιδήποτε) (4). Τέλος, ολοκληρώνουμε με «Next» (5).

Βήμα 4 – Επισκόπηση παραμέτρων και Δημιουργία Workspace

Η οθόνη αυτή είναι πληροφοριακού χαρακτήρα όπου μπορείτε να κάνετε ανασκόπηση των παραμέτρων δημιουργίας του Workspace. Πληροφορικά, θα δημιουργηθεί ένα Tablespace, το όνομα και το Path του οποίου βλέπετε στην

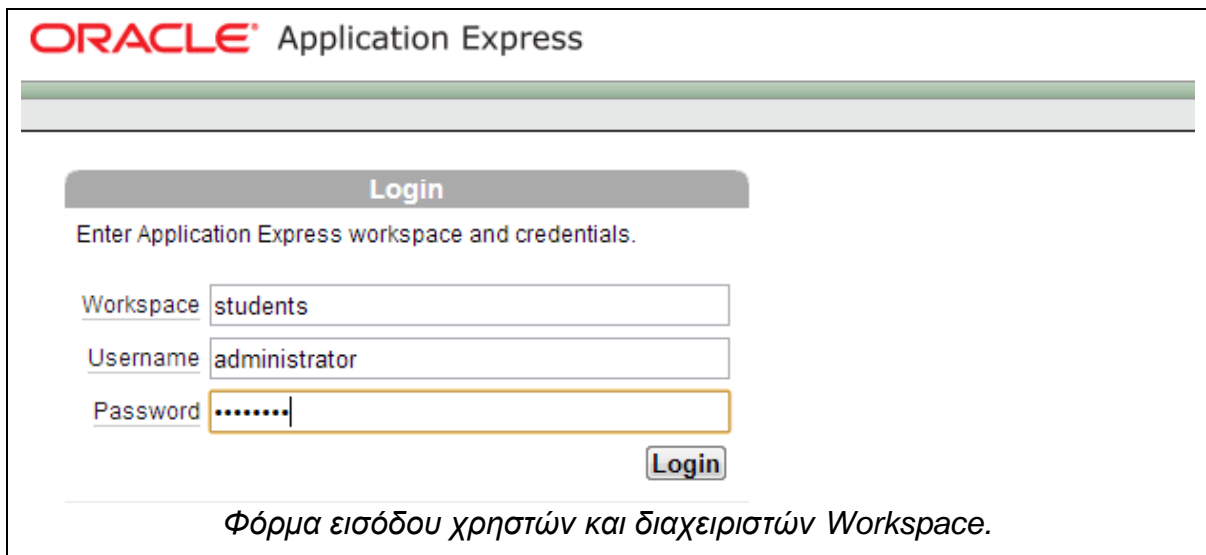
περιοχή «Datafile for Tablespace» (1). Ολοκληρώστε την δημιουργία του Workspace, πατώντας στο «Create» (2). Σε λίγο διάστημα το Workspace δημιουργείται και μπορείτε να το χρησιμοποιήσετε.

5.2.4.6 – Σύνδεση στο Workspace και δημιουργία χρηστών

Την διαχείριση πλέον του Workspace μπορεί να την αναλάβει ο διαχειριστής του. Όχι ότι ο Γενικός Διαχειριστής δεν μπορεί, αλλά καλό είναι να το αναλαμβάνουν οι επιμέρους διαχειριστές του. Αποσυνδεόμαστε από την σελίδα του Γενικού Διαχειριστή, πατώντας Logout από την πάνω δεξιά γωνία της σελίδας. Για να μεταβούμε στην οθόνη Login των χρηστών και διαχειριστών Workspace, μεταβαίνουμε στην παρακάτω διεύθυνση:

`http://<server-address>:8080/apex`

Αμέσως μας φορτώνεται η κεντρική σελίδα εισόδου στο APEX την οποία βλέπουμε και παρακάτω:



ORACLE[®] Application Express

Login

Enter Application Express workspace and credentials.

Workspace

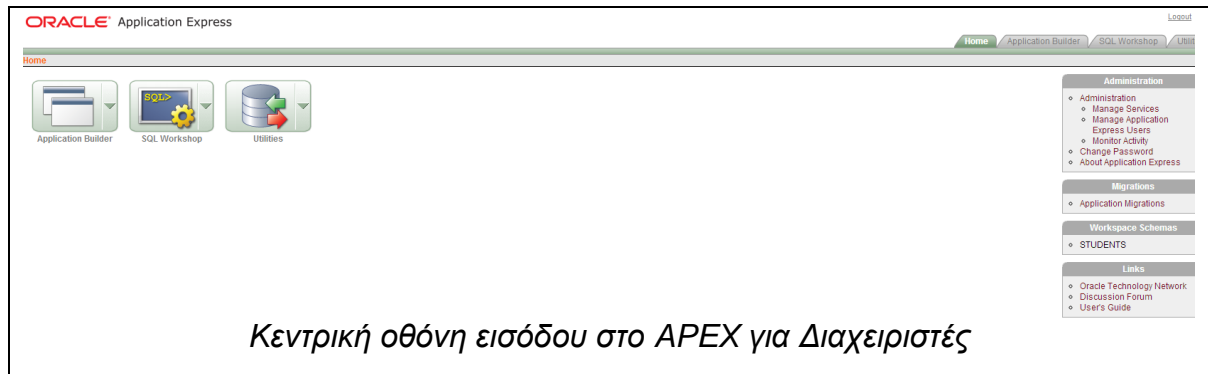
Username

Password

Φόρμα εισόδου χρηστών και διαχειριστών Workspace.

Εδώ απλά πρέπει να επιλέξουμε το Workspace στο οποίο θα συνδεθούμε και να δώσουμε τα Credentials μας ως διαχειριστές του. Αφού τα εισάγουμε τότε

μπαίνουμε μέσα στο Workspace όπου μπορούμε να το διαχειριστούμε. Παρακάτω βλέπουμε την κεντρική οθόνη του:



Η οθόνη αυτή είναι παρόμοια για τους απλούς χρήστες (όπως θα δούμε στο κεφάλαιο 7) αλλά για τους διαχειριστές του Workspace υπάρχουν περισσότερες επιλογές. Δεν θα τις αναλύσουμε όλες, αλλά θα βιαστούμε να δημιουργήσουμε **χρήστες** για το Workspace μας.

Οι χρήστες που θα δημιουργήσουμε μπορούν είτε να εργαζόνται στο Schema του Workspace (προεπιλογή) είτε σε δικό τους. Εμείς θα πάρουμε την δεύτερη περίπτωση, υποθέτοντας πως οι χρήστες του Workspace είναι τα ίδια πρόσωπα με τους χρήστες της βάσης, οπότε θα εργαστούν στο δικό τους Schema. Η δική μας υλοποίηση θα είναι η εξής:

- 1) Κάθε χρήστης θα έχει το δικό του Schema και μόνο αυτό.
- 2) Κανένας δεν θα έχει πρόσβαση στο Schema του Workspace.
- 3) Κανένας δεν θα έχει πρόσβαση στο Schema άλλου χρήστη
- 4) Ο διαχειριστής έχει πρόσβαση σε όλα τα Schemas.

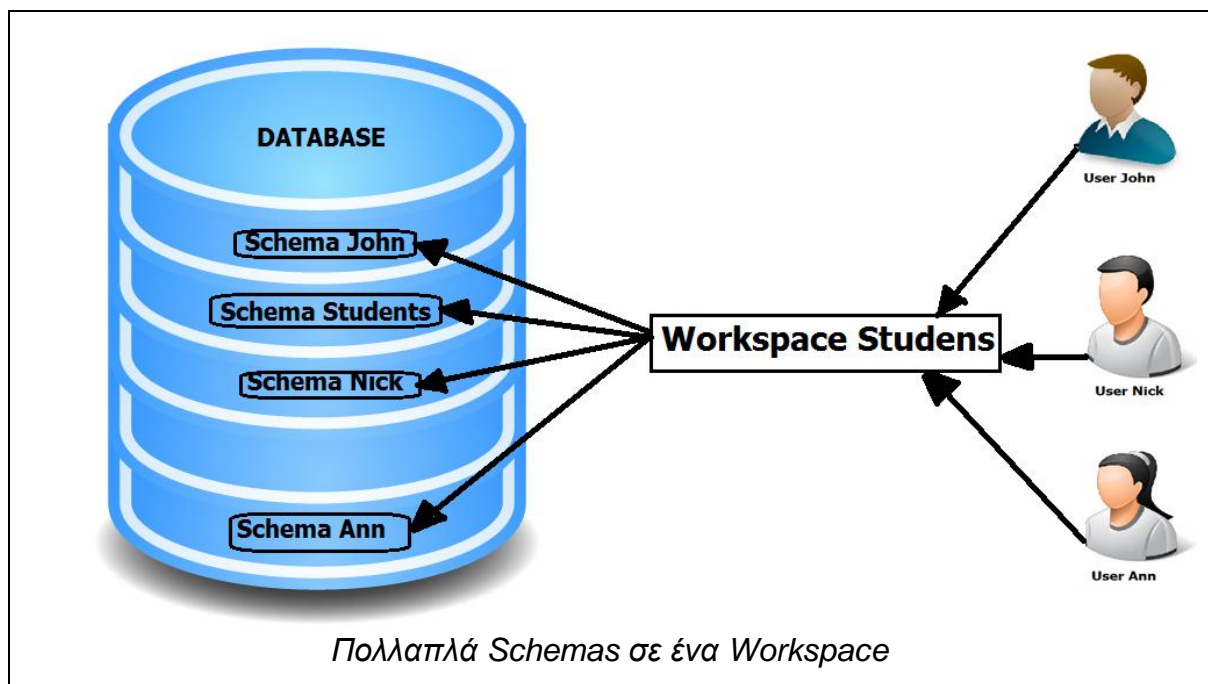
Καλή πρακτική

Η παραπάνω υλοποίηση είναι καλή επιλογή για εκπαιδευτικούς σκοπούς, όπου ο κάθε φοιτητής-χρήστης της βάσης θα έχει το δικό του Schema και δεν θα μπορεί

να έχει πρόσβαση σε άλλα. Ο διαχειριστής του Workspace που θα είναι ο διδάσκοντας, θα έχει πρόσβαση σε όλα τα Schemas.

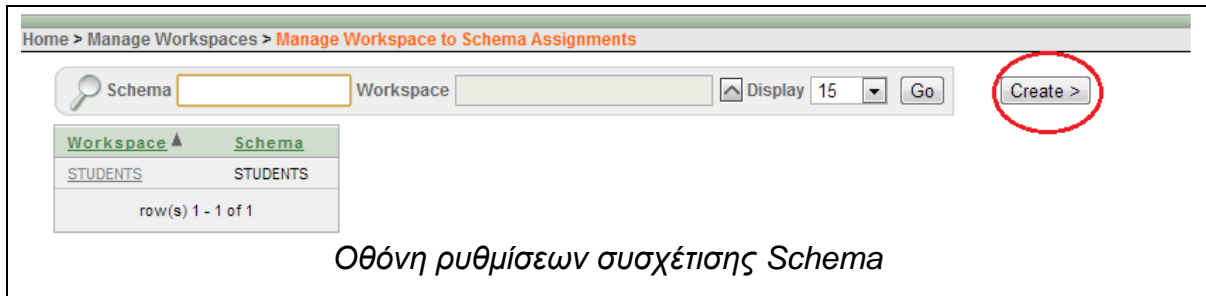
5.2.4.6.1 – Συσχέτιση πολλαπλών Schemas στο ίδιο Workspace.

Στα Workspaces του APEX, παρόλο που έχουν το δικό τους Schema (το λεγόμενο **Default Schema**), μπορούμε να συσχετίσουμε κι άλλα. Στην παρακάτω εικόνα βλέπουμε ένα σχήμα για να καταλάβουμε καλύτερα το Concept.

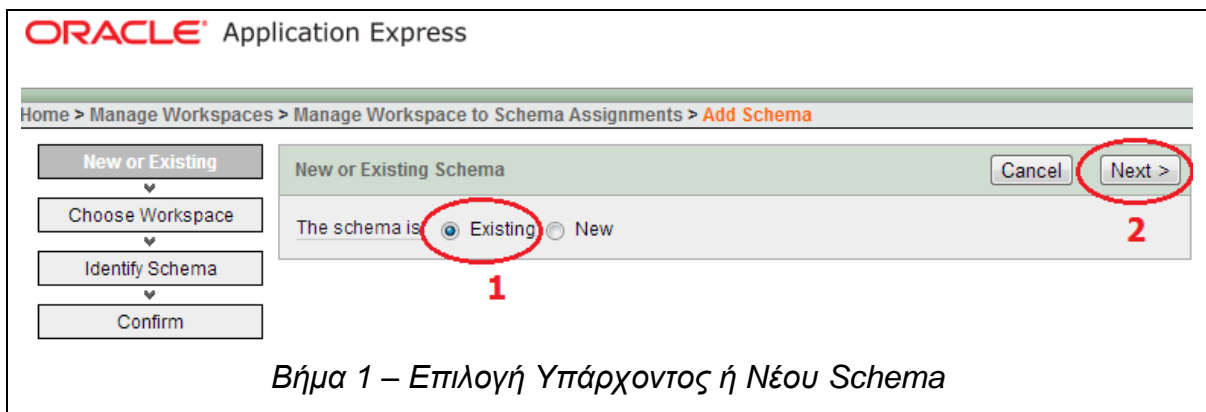


Η διαδικασία συσχέτισης ενός Schema με ένα Workspace μπορεί να γίνει με δύο τρόπους, είτε μέσα από το APEX (γραφικά), είτε μέσα από το SQL*PLUS / SQL Developer (προγραμματιστικά). Θα καλύψουμε και τους δύο τρόπους.

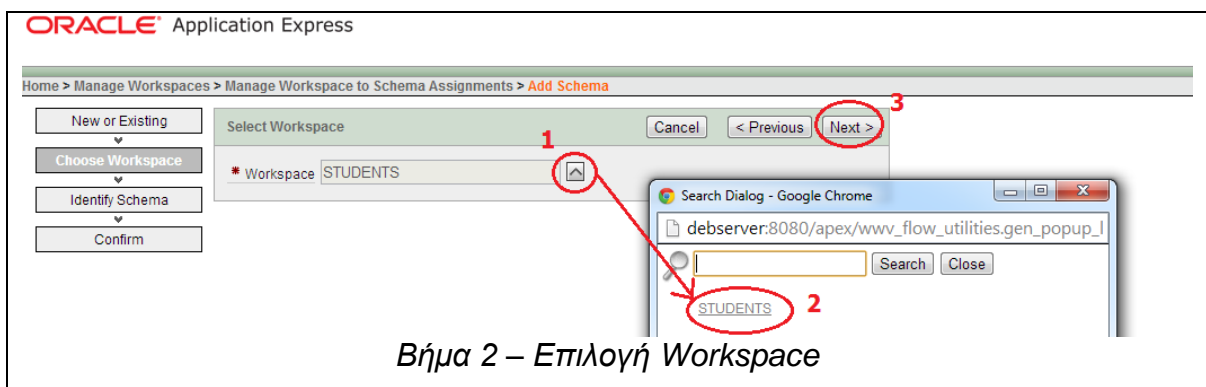
Για να συσχετίσουμε ένα Schema, μέσα από το APEX, θα χρειαστεί να ξανασυνδεθούμε ως Γενικοί Διαχειριστές. Μεταβαίνουμε στην σελίδα του γενικού διαχειριστή (apex_admin σε νέα καρτέλα) εισάγουμε τα διαπιστευτήριά μας για να μπούμε. Επισκεπτόμαστε ξανά το «Manage Workspace» αλλά αυτήν την φορά επιλέγουμε το «Manage Workspace to Schema Assignments» από τις διαθέσιμες επιλογές. Βλέπουμε την οθόνη παραμετροποίησης παρακάτω:



Στην οθόνη αυτή βλέπουμε πόσα Workspaces υπάρχουν και ποια Schemas έχει συσχετιζόμενα. Πατάμε «Create» για να δημιουργήσουμε νέα συσχέτιση. Τότε ξεκινάει η διαδικασία:



Σε αυτό το βήμα επιλέγουμε αν το Schema υπάρχει ή θέλουμε να δημιουργηθεί νέο. Εμείς έχουμε υπάρχοντα Schemas οπότε θα επιλέξουμε «Existing» (1). Ολοκλήρωση βήματος με «Next» (2).



Εδώ καλούμαστε να επιλέξουμε το Workspace που θέλουμε να συσχετίσουμε με υπάρχον Schema. Πατάμε στο βελάκι δίπλα από το πεδίο «Workspace» και στο νέο παράθυρο που ανοίγει, επιλέγουμε το «Students» Workspace (2). Ολοκλήρωση βήματος με «Next» (3).

Βήμα 3 - Επιλογή Schema προς συσχέτιση

Στο βήμα αυτό επιλέγουμε το Schema προς συσχέτισμό. Με το βελάκι δίπλα στο πεδίο «Schema» (1) θα ανοίξει νέο παράθυρο όπου θα επιλέξουμε το Schema LEFTERIS (2) προς συσχέτισμό με το Workspace Students. Ολοκληρώνουμε με «Next» (3).

✓	New or Existing	Existing
	Schema	LEFTERIS
	Workspace	STUDENTS

Βήμα 4 – Ολοκλήρωση

Σε αυτό το βήμα κάνουμε επισκόπηση των ρυθμίσεών μας (1) και ολοκληρώνουμε την διαδικασία πατώντας «Add Schema» (2). Το Schema προστίθεται στο Workspace και μπορούμε επί τόπου να το διαπιστώσουμε από την επόμενη σελίδα που φορτώνεται.

Workspace ▲	Schema
<u>STUDENTS</u>	LEFTERIS
<u>STUDENTS</u>	STUDENTS
row(s) 1 - 2 of 2	

Επιβεβαίωση συσχέτισης Schema με Workspace

Η ίδια διαδικασία όμως μπορεί να πραγματοποιηθεί και μέσα από το SQL*PLUS / SQL Developer, **προγραμματιστικά**.

Ας πούμε τώρα ότι θέλουμε να συσχετίσουμε το Schema ANNA στο Workspace Students. Ανοίγουμε το SQL Developer και συνδεόμαστε ως χρήστες SYS as SYSDBA. Θα εκτελέσουμε την Stored Procedure ADD_SCHEMA του πακέτου APEX_INSTANCE_ADMIN αφού πρώτα αλλάξουμε το τρέχον Schema μας. Γράφουμε τις παρακάτω εντολές:

```
ALTER SESSION SET CURRENT_SCHEMA=APEX_030200; --Αλλαγή Schema
BEGIN
APEX_INSTANCE_ADMIN.ADD_SCHEMA('students','ANNA'); --Συσχέτιση
--(workspace,schema)
END;
COMMIT;
```

Το SQL Developer θα αποκριθεί με «anonymous block completed». Ανοίγουμε πάλι την σελίδα διαχείρισης του APEX (ως γενικός διαχειριστής) και επισκεπτόμαστε πάλι την σελίδα του «Manage Workspace to Schema Assignments». Οι αλλαγές γίνανε, όπως βλέπουμε παρακάτω.

Workspace ▲	Schema
STUDENTS	ANNA
STUDENTS	LEFTERIS
STUDENTS	STUDENTS
row(s) 1 - 3 of 3	

Επιβεβαίωση προσθήκης του Schema, μετά την εκτέλεση της ADD_SCHEMA

Βλέπουμε ότι το ζητούμενο έργο, πραγματοποιήθηκε επιτυχώς. Οπότε έχουμε και αυτήν την εναλλακτική για να προσθέσουμε (συσχετίζουμε) Schemas στα Workspaces, που είναι και **ταχύτερη**.

Καλή πρακτική

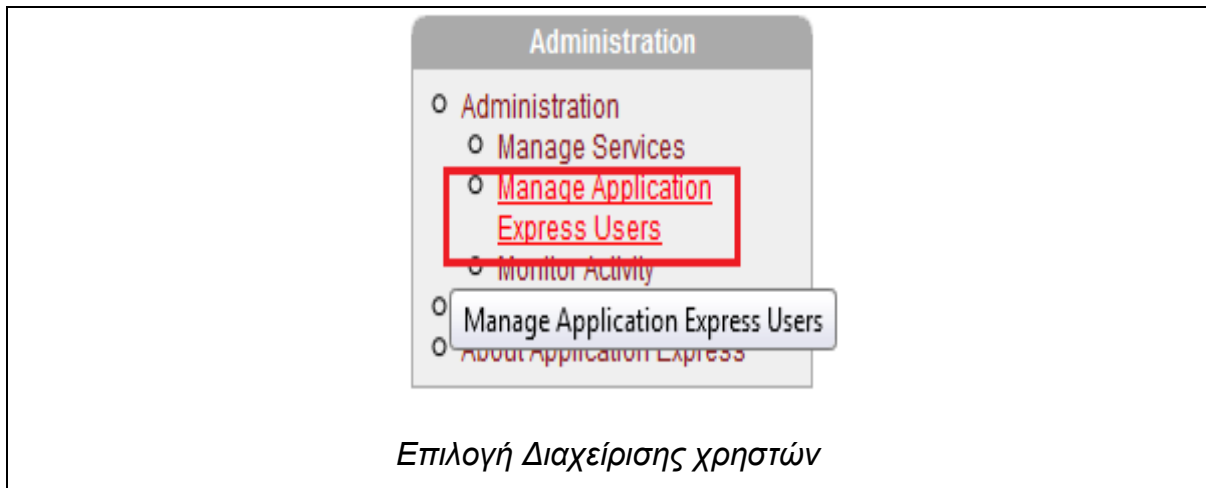
Η προγραμματιστική προσθήκη Schema στο Workspace είναι ταχύτερη. Αν πρόκειται να προσθέσετε πολλά Schemas σε ένα Workspace, η διαδικασία για να το πράξετε μέσα από το APEX είναι χρονοβόρα. Προγραμματιστικά μπορείτε να καλέσετε πολλές φορές την ADD_SCHEMA μέσα από ένα ανώνυμο PL/SQL Block και να προσθέσετε πολλά Schemas με μία κίνηση. Επίσης μπορείτε να το κάνετε και σε Script αρχείο.

5.2.4.6.2 – Δημιουργία χρηστών

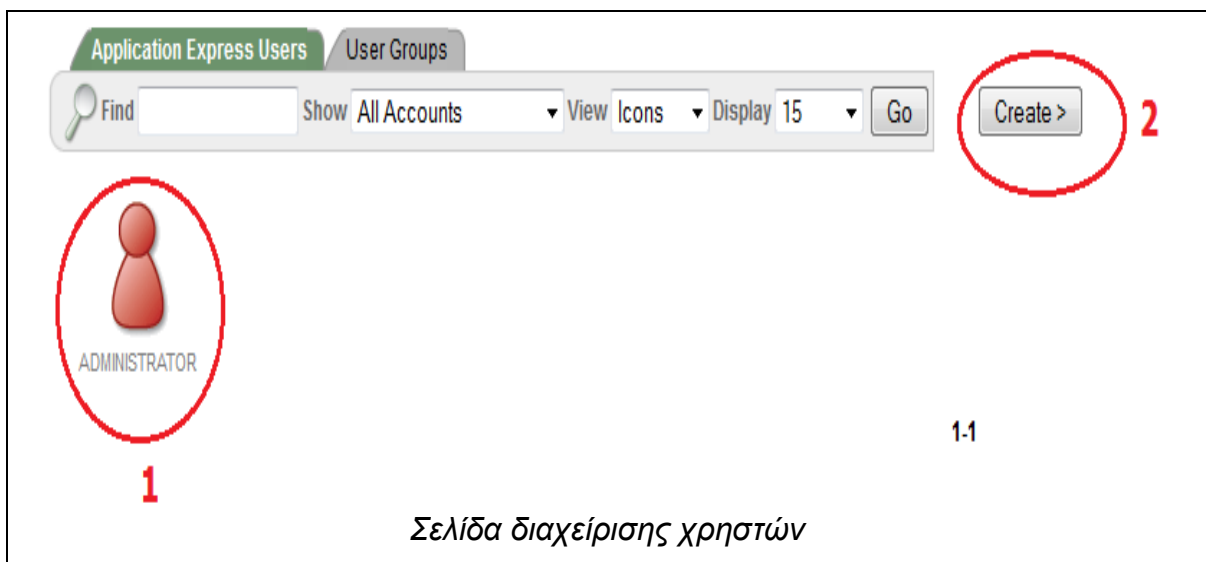
Οι χρήστες σε ένα Workspace του APEX μπορεί να είναι είτε **Developers** είτε **Workspace Administrators**. Οι πρώτοι είναι οι απλοί χρήστες που μπορούν να εργαστούν στο Workspace, που είναι καθήκον του Workspace Administrator να δημιουργήσει. Εδώ πάλι υπάρχουν δύο τρόποι να δημιουργήσουμε χρήστη, είτε **μέσα από το μενού APEX**, είτε **μέσα από το APEX SQL Commands**.

Για να δημιουργήσουμε ένα χρήστη μέσα από το μενού του APEX συνδεόμαστε στο Workspace μας, ως διαχειριστές Workspace, και μεταβαίνουμε

στην επιλογή «Manage Application Express Users» της κατηγορίας «Administration» όπως βλέπουμε παρακάτω:



Μεταβαίνοντας στην επιλογή αυτήν, ανοίγει η σελίδα διαχείρισης χρηστών που είναι η παρακάτω:



Ακόμη δεν υπάρχουν χρήστες στο Workspace μας, πέραν του Workspace Administrator (1), οπότε μεταβαίνουμε στο κουμπί «Create» (2) για να δημιουργήσουμε έναν.

The screenshot shows a user creation form with three main sections: User Identification, Developer Privileges, and Account Control. Red boxes and numbers 1 through 8 highlight specific fields and controls:

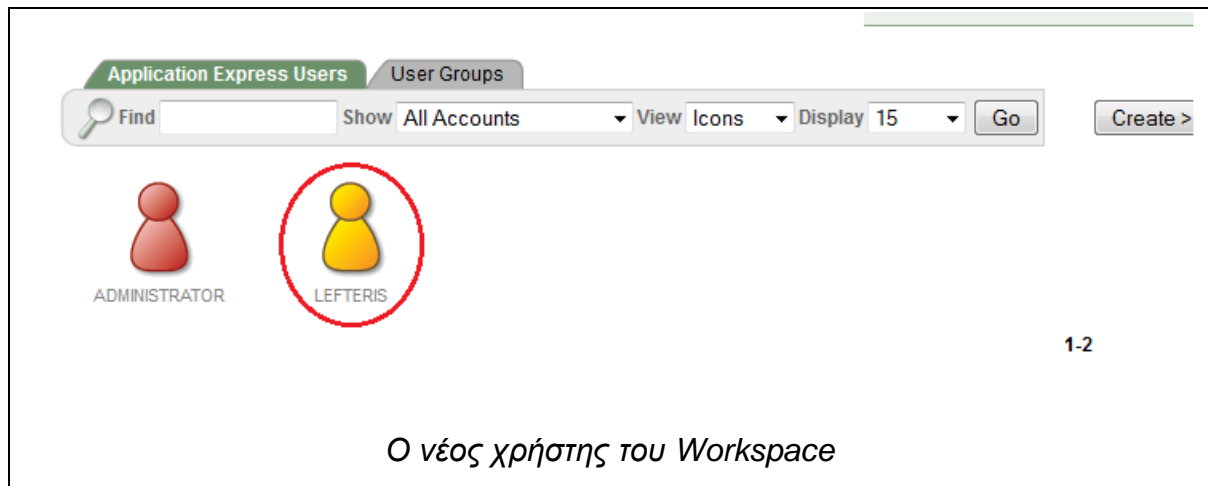
- 1:** The User Identification section, containing fields for Username (lfteris), Password, Confirm Password, and Email Address (lfteris@apex_students.gr).
- 2:** The Accessible Schemas (null for all) field in the Developer Privileges section, set to lfteris.
- 3:** The Default Schema dropdown menu, set to LEFTERIS.
- 4:** The 'User is a developer' radio button, which is selected (Yes).
- 5:** The 'User is a workspace administrator' radio button, which is selected (No).
- 6:** The 'Set Account Availability' dropdown menu, set to Unlocked.
- 7:** The 'Require Change Of Password On First Use' dropdown menu, set to No.
- 8:** The 'Create User' button at the top right of the form.

Σελίδα δημιουργίας χρήστη

Εδώ λοιπόν δημιουργούμε νέο χρήστη στο Workspace μας. Θα χρησιμοποιήσουμε τον χρήστη LEFTERIS. Αρχικά λοιπόν εισάγουμε τα διαπιστευτήρια σύνδεσης του χρήστη, δηλαδή username και password, ενώ ζητάει και ένα email το οποίο μπορεί να είναι οτιδήποτε (1).

Στην συνέχεια επιλέγουμε σε ποια Schema θα μπορεί ο χρήστης να έχει πρόσβαση, γράφοντας το όνομα του Schema **που του ανήκει** (2), στην προκειμένη περίπτωση το LEFTERIS (αν το αφήσετε κενό, θα έχει πρόσβαση σε όλα τα Schemas). Εν συνεχεία επιλέγουμε το Default Schema του χρήστη (3), στην προκειμένη, το Lfteris. Τώρα ρυθμίζουμε την ιδιότητα του χρήστη. Αν ο χρήστης θέλουμε να είναι Developer (που θέλουμε) , επιλέγουμε «YES» (4). Στην επόμενη επιλογή επιλέγουμε αν ο χρήστης μας θα είναι Administrator ή όχι του Workspace. Δεν θέλουμε, οπότε επιλέγουμε «NO» (5).

Ολοκληρώνοντας, επιλέγουμε την διαθεσιμότητα του λογαριασμού, οπότε επιλέγουμε «Unlocked» για να τον ορίσουμε διαθέσιμο (6). Ολοκληρώνουμε επιλέγοντας αν ο κωδικός πρόσβασης που εισήχθηκε παραπάνω θα είναι προσωρινός ή μόνιμος. Εμείς θέλουμε να είναι μόνιμος, οπότε επιλέγουμε «NO» (7). Ολοκληρώνουμε την δημιουργία του χρήστη, πατώντας «Create» (8) (ή Create and Create Another να θέλουμε να δημιουργήσουμε κι άλλον χρήστη συνεχόμενα). Ο νέος μας χρήστης δημιουργήθηκε επιτυχώς όπως βλέπουμε στην παρακάτω εικόνα:



Το κίτρινο χρώμα δηλώνει χρήστη Developer. Ο νέος μας χρήστης μπορεί να συνδεθεί κανονικά στο Workspace και να εργαστεί (εργασία στο APEX καλύπτεται στο κεφάλαιο 7).

Προσοχή!

Μην ξεχνάτε ποτέ, κατά την δημιουργία χρήστη, να αλλάζετε την τιμή του πεδίου «Require Change Of Password On First Use» σε «NO» διαφορετικά, το σύστημα θα απαιτήσει ο χρήστης να αλλάξει τον κωδικό του στην πρώτη του είσοδο στο Workspace. Αν έχετε σοβαρό λόγο που θέλετε να το κάνετε, ή θέλετε ο χρήστης να επιλέξει τον κωδικό του, τότε αφήστε το σε «Yes».

Τώρα πλέον θα εισάγουμε έναν χρήστη προγραμματιστικά, μέσα από το APEX SQL Commands, του εργαλείου συγγραφής εντολών SQL και PL/SQL του APEX (θα το δούμε στο κεφάλαιο 7). Μεταφερόμαστε σε αυτό μέσα από τη σουίτα SQL Workshop και το ανοίγουμε. Σε αυτό μπορούμε να γράψουμε τις εντολές μας και να τις εκτελέσουμε.

Για να εισάγουμε χρήστη μέσα από την PL/SQL πρέπει να καλέσουμε την Stored Procedure ADD_USER του πακέτου APEX_UTIL, μέσα από ανώνυμο PL/SQL Block. Γράφουμε τον παρακάτω κώδικα για να επιτύχουμε το ζητούμενο αυτό.

BEGIN

APEX_UTIL.CREATE_USER(

p_user_name => 'ANNA', --Όνομα χρήστη

P_DESCRIPTION => 'student', --Περιγραφή

P_EMAIL_ADDRESS => 'anna@apex_student.gr', --Email

p_web_password => 'secret', --Password

p_developer_privs => 'CREATE:DATA_LOADER:EDIT:HELP:MONITOR:SQL',

--Δημιουργία Developer χρήστη

--Η παράμετρος «ADMIN:CREATE:DATA_LOADER:EDIT:HELP:MONITOR:SQL»

--δημιουργεί Workspace Administrator χρήστη.

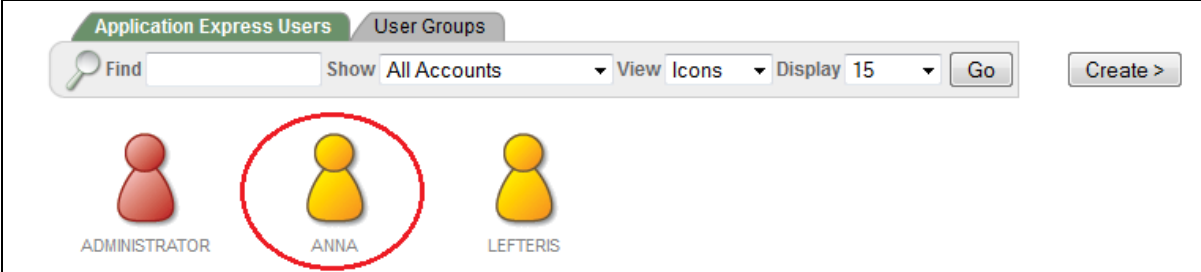
P_DEFAULT_SCHEMA => 'ANNA', --Προεπιλεγμένο Schema

P_allow_access_to_schemas => 'ANNA', --Πρόσβαση μόνο στο Schema ANNA

P_CHANGE_PASSWORD_ON_FIRST_USE => 'N'); --Μόνιμο Password

END;

Εκτελούμε το Block Εντολών, πατώντας «Run». Το SQL Commands θα πρέπει να αποκριθεί με «Statement processed». Μεταβαίνουμε πάλι στο «Manage Application Express Users » και βλέπουμε ότι ο χρήστης ANNA μας περιμένει εκεί.



1-3

Νέος χρήστης από την κλήση της ADD_USER

Ο νέος μας χρήστης δημιουργήθηκε και μάλιστα γρηγορότερα από την πρώτη μέθοδο δημιουργίας.

Καλή πρακτική

Όπως και με τον συσχετισμό πολλαπλών Schemas, έτσι και η προσθήκη χρηστών προγραμματιστικά είναι ταχύτερη. Αν έχετε να προσθέτετε πολλούς χρήστες με μία κίνηση, αυτή η επιλογή είναι η καλύτερη. Μπορείτε επίσης να χρησιμοποιήσετε τον Script Editor του APEX για να εκτελέσετε την διαδικασία μέσα από Script.

Και εδώ κάπου τελειώσαμε με την διαχείριση του Oracle APEX, τον λόγο πλέων έχουν οι χρήστες που θα συνδεθούν και θα εργαστούν σε αυτό.

5.2.4.7 – Συμπέρασμα

Το Oracle APEX είναι ένα υπέροχο περιβάλλον για εργασία το οποίο είναι προσβάσιμο από οπουδήποτε και είναι και πολύ εύκολο στην χρήση και παραμετροποίηση. Όσον αφορά την διαχείρισή του, είδαμε πως είναι εύκολη υπόθεση, καθώς παρέχει πληθώρα επιλογών και εργαλείων που μπορούν να κάνουν την όλη διαδικασία «παιχνίδι».

Keep in mind...

Για να λειτουργήσει το Oracle APEX, θα πρέπει η βάση δεδομένων να είναι Open και Mounted. Στον τερματισμό της βάσης δεδομένων, τερματίζεται και το APEX.

5.3 – Λοιπές Δυνατότητες Διαχείρισης

Στην τελευταία αυτή ενότητα του κεφαλαίου, θεωρούμε σκόπιμο να επεκταθούμε λίγο τις πρόσθετες δυνατότητες που παρέχει η Oracle για να κάνει την διαχείρισή σας ποιο αποδοτική και λειτουργική. Μαζί θα δούμε μερικά από τα «κόλπα» που έχετε στην διάθεσή σας για να διαπιστώσετε κι εσείς με τα μάτια σας την «δύναμη» της Oracle.

5.3.1 – Εκτέλεση μεθόδων Java μέσα από την PL/SQL

Ε, μιας και η Java πλέον είναι ιδιοκτησία της Oracle, σίγουρα δεν θα μπορούσαν τα προϊόντα της να μην συνεργάζονται, σωστά;

Η Oracle σας δίνει την δυνατότητα να δημιουργείτε Methods (functions κατά την C) στην Java και να τις καλείτε μέσα από την PL/SQL. Δηλαδή σχεδόν το αντίστροφο του JDBC, όπου εκτελείτε SQL και PL/SQL μέσα από την Java (θα το δούμε ενδελεχώς στο κεφάλαιο 7).

Ας πάρουμε μαζί ένα παράδειγμα. Ανοίξτε τον αγαπημένο σας Text Editor (πχ Kwrite) και γράψτε τον παρακάτω κώδικα Java:

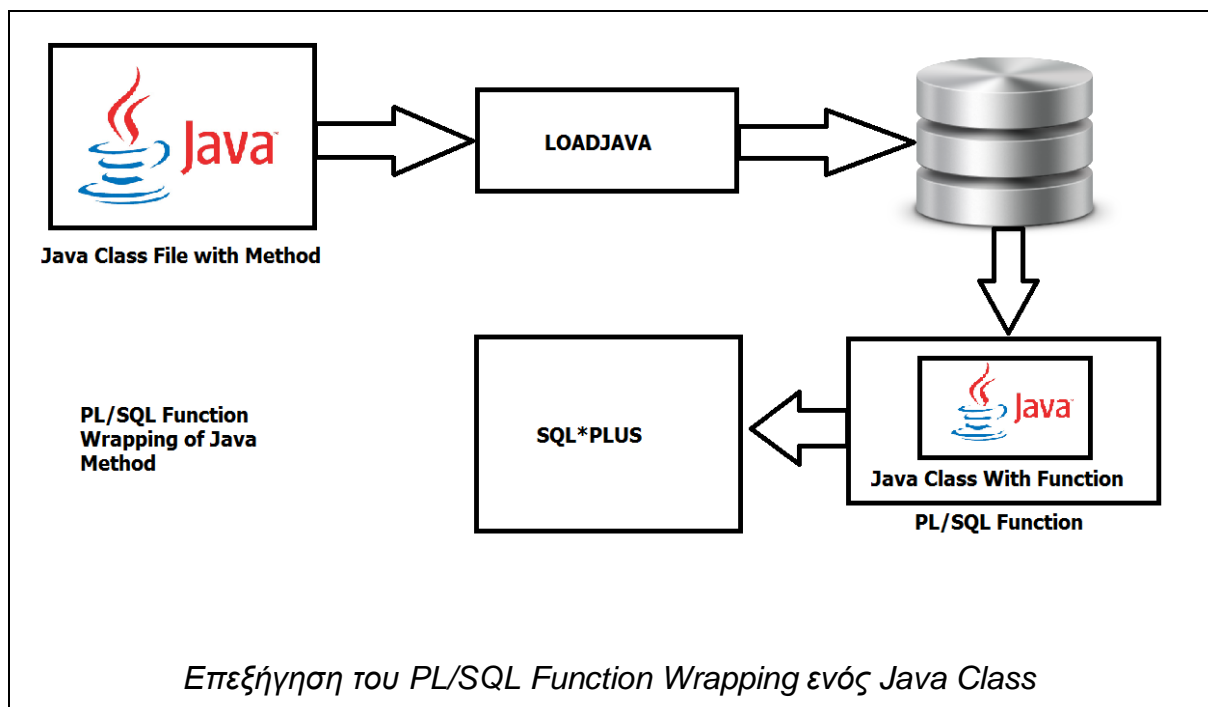
```
public class Hello{  
  
    public static String hi()  
  
    {  
  
        return "Hello from Java and Oracle!!!!";  
  
    } //end of hi  
  
} //end of class
```

Η απλή αυτή μέθοδος δεν κάνει τίποτα άλλο από το να επιστρέφει ένα String με το μήνυμα «Hello from Java and Oracle». Τίποτα το ιδιαίτερο δηλαδή. Αποθηκεύουμε το αρχείο μας ως «Hello.java». Μεταγλωττίζουμε το αρχείο μας με

τον Javac ως εξής (HINT: το JDK βρίσκεται μέσα στον κατάλογο ORACLE_HOME/jdk) :

```
oracle@debserver$ $ORACLE_HOME/jdk/bin/javac Hello.java
```

Η μεταγλώττιση ολοκληρώνεται και δημιουργείται το αρχείο «Hello.class». Να θυμάστε το path του αρχείου. Τώρα θα χρησιμοποιήσουμε ένα εργαλείο της Oracle, που μας επιτρέπει να φορτώνουμε Java Classes με μεθόδους μέσα στην βάση, και να τις καλούμε σαν Stored Procedures, το LOADJAVA το οποίο βρίσκεται στον κατάλογο ORACLE_HOME/bin. Το εργαλείο αυτό φορτώνει το class αρχείο που παρήγαγε ο compiler της Java και επιτρέπει το «κουκούλωμα» (wrap) της μεθόδου που περιέχει, μέσα σε μια PL/SQL Function, την οποία μπορούμε να καλέσουμε μέσα από το SQL*PLUS. Δείτε το παρακάτω σχήμα για να καταλάβετε την διαδικασία:



Τώρα θα πρέπει να φορτώσουμε το αρχείο Java μέσα στην βάση. Χρησιμοποιούμε το LOADJAVA για να το κάνουμε αυτό, ως εξής:

```
oracle@debserver$ cd $ORACLE_HOME/bin  
  
#Syntax: loadjava -user <όνομα χρήστη βάσης> <path προς αρχείο Class>  
  
oracle@debserver$ ./loadjava -user sys /home/oracle/java/Hello.class  
  
Password:
```

Το αρχείο φορτώνεται στην βάση και μπορούμε πλέον να καλέσουμε την μέθοδο.

Troubleshoot:

Δεν ολοκληρώνεται η διαδικασία, αλλά σταματάει με το σφάλμα «ORA-29516:

Aurora assertion failure: Assertion failure at joez.c:3311

Bulk load of method java/lang/Object.<init> failed; insufficient shm-object space». Τι να κάνω;

Λύση³⁴:

Το πρόβλημα το προκαλεί ο Java JIT Debugger (Just In Time), ο οποίος πολλές φορές προκαλεί προβλήματα. Ανοίξτε το SQL*PLUS και συνδεθείτε ως SYS as SYSDBA και απενεργοποιήστε τον με την εντολή:

```
SQL> ALTER SYSTEM SET JAVA_JIT_ENABLED=FALSE;
```

```
System Altered.
```

Επαναλάβετε την εκτέλεση του loadjava, όπου πλέον θα ολοκληρώσει ο έργο του.

³⁴ Internet Source → Lovely Oracle → ORA-29516: Aurora assertion failure: Assertion failure → “But you can get the error Ora-29516 Aurora assertion failure on Linux x64bit platform, and the workaround to overcome that error is turning off the JIT compiler: alter system set java_jit_enabled=false;” – Site (<http://fahdmirza.blogspot.gr/2011/08/ora-29516-aurora-assertion-failure.html>)

Αφού φορτωθεί το αρχείο στην βάση, ανοίξτε το SQL PLUS και συνδεθείτε σαν χρήστες SYS as SYSDBA (ή σαν χρήστες που ορίσατε στο loadjava) και δημιουργήστε μια PL/SQL Function με τις παρακάτω εντολές.

```
SQL> CREATE OR REPLACE FUNCTION helloworld RETURN VARCHAR2 AS
LANGUAGE JAVA NAME 'Hello.hi () return java.lang.String';
/
Function Created.
```

Δημιουργήστε τώρα μια Global μεταβλητή και αποθηκεύστε την τιμή που επιστρέφει η Function (μέσω της μεθόδου Java που περιέχει μέσα της) ως εξής:

```
SQL> VARIABLE TEST;
SQL> CALL helloworld() INTO :TEST;
Call completed.
```

Τυπώστε τώρα τα περιεχόμενα της μεταβλητής TEST στο STDOUT για να δούμε τι περιέχει:

```
SQL> PRINT TEST;
```

Το αποτέλεσμα το βλέπουμε στην παρακάτω εικόνα:

```
SQL> print test;
TEST
-----
Hello from Oracle and Java!
```

Έξοδος τιμής από την μέθοδο της Java που δημιουργήσαμε

Φοβερό ε; Με αυτόν τον τρόπο μπορείτε να δημιουργείτε Μεθόδους στην Java και να τις φορτώνετε στην βάση σας για πάσα χρήση.

Keep in mind...

Οι κλάσεις και οι μέθοδοι που φτιάχνετε σε Java για χρήση μέσα στην PL/SQL φροντίστε να είναι Public. Επίσης οι μέθοδοι φροντίστε να είναι Static και οπωσδήποτε να επιστρέφουν κάτι.

5.3.2 – Δημιουργία δεικτών

Η Oracle μας επιτρέπει να δημιουργήσουμε **Δείκτες** (Indexes) για να επιταχύνουμε τα αποτελέσματα των Queries προς έναν πίνακα. Τι είναι όμως ο δείκτης;

Ο δείκτης είναι αυτό που δηλώνει το όνομά του, δείχνει δηλαδή ακριβώς την θέση των δεδομένων. Σε έναν μεγάλο πίνακα (με πολλές γραμμές), πολλές φορές τα Queries καθυστερούν γιατί ελέγχεται μία-μία η γραμμή κάθε πίνακα για να συμφωνήσει με τα κριτήρια αναζήτησης που όρισε ο χρήστης. Με την χρήση δείκτη όμως, η βάση γνωρίζει επακριβώς που θα ψάξει και έτσι η διαδικασία εύρεσης των δεδομένων επιταχύνεται. Οι δείκτες λοιπόν παίζουν σημαντικό ρόλο στην αύξηση της **απόδοσης** (performance). Επίσης μπορεί να προσθέσει και περιορισμό μοναδικότητας σε κάθε στήλη, ώστε να μην επιτρέπεται να εισαχθεί δεδομένο ίδιο με ένα προηγούμενο, αποτρέποντας έτσι τα **διπλότυπα δεδομένα** (duplicates).

Η δημιουργία ενός δείκτη είναι εύκολη. Θα δημιουργήσουμε έναν με ένα παράδειγμα. Ας πούμε ότι ο πίνακας PROJECTS περιέχει τις εργασίες που έχουν ανατεθεί σε φοιτητές (καλή ώρα). Ο πίνακας απαρτίζεται από τις στήλες PROJECT_ID (Number), TITLE(VARCHAR2), NAME(VARCHAR2) και LASTNAME. Ο πίνακας αυτός έχει πολλές εγγραφές (100+). Εδώ τώρα έχουμε μια ιδιαίτερη περίπτωση. Η στήλη PROJECT_ID είναι Primary Key του πίνακα PROJECTS οπότε έχει ήδη περιορισμό μοναδικότητας (unique constraint) και είναι ήδη δεικτοδοτημένη (indexed). Ναι σαφώς καμία τέτοια εντολή δεν έδωσε ο

δημιουργός της αλλά η Oracle το εφαρμόζει αυθαίρετα³⁵. Και τώρα ερχόμαστε εμείς. Είπαμε ο πίνακας έχει ήδη ένα δείκτη. Πειράζει λοιπόν και εμείς να δημιουργήσουμε έναν ακόμα; Καθόλου, τουναντίον μάλιστα!

Θα δεικτοδοτήσουμε την στήλη TITLE ώστε να βρίσκουμε αμέσως τα δεδομένα που αφορούν αυτήν την στήλη. Επίσης θα εφαρμόσουμε περιορισμό μοναδικότητας στην στήλη ώστε να μην μπορεί κάποιος φοιτητής να πάρει θέμα ίδιο με έναν άλλον (είμαστε λίγο αυστηροί, τω γνωρίζω!). Θα δημιουργήσουμε λοιπόν έναν δείκτη για αυτήν την στήλη με την παρακάτω εντολή:

```
--Syntax: CREATE UNIQUE INDEX <όνομα δείκτη> ON <πίνακας> (<στήλες  
πίνακα>)  
  
SQL> CREATE UNIQUE INDEX TITLE_INDEX ON PROJECTS (TITLE);  
  
Index Created.
```

Από κει και πέρα θα συμβαίνουν δύο πράγματα. Πρώτον τα επερωτήματα στην βάση για αυτόν τον πίνακα θα εκτελούνται γρηγορότερα και δεύτερον δεν θα μπορεί η στήλη TITLE να πάρει πανομοιότυπη τιμή με ήδη υπάρχουσα.

Keep in mind...

Σε μικρούς πίνακες δεν έχει νόημα να δημιουργείτε δείκτες, καθώς η αναζήτηση των γραμμών (το λεγόμενο Fetching) γίνεται γρήγορα ούτως η άλλως.

³⁵ Oracle Documentation → Creating Indexes → Creating an Index Associated with a Constraint → “Oracle Database enforces a UNIQUE key or PRIMARY KEY integrity constraint on a table by creating a unique index on the unique key or primary key” – Site (http://docs.oracle.com/cd/B28359_01/server.111/b28310/indexes003.htm#ADMIN11722)

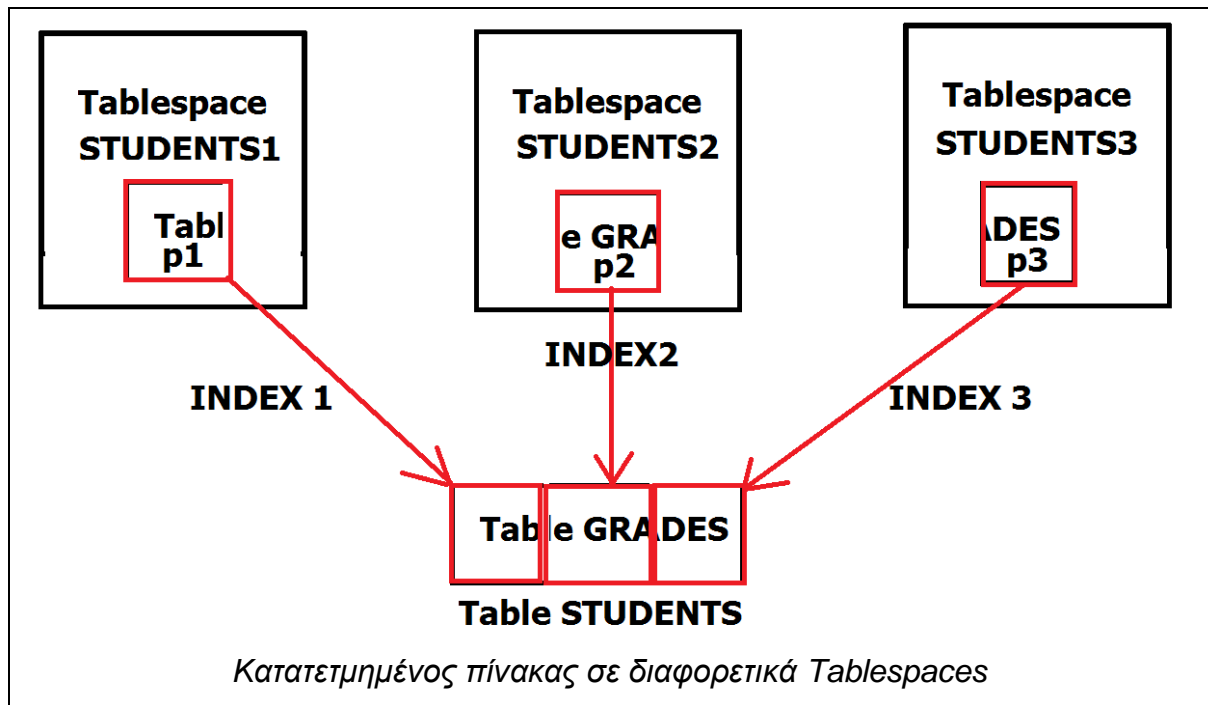
5.3.3 – Κατάτμηση Πινάκων

Υπάρχουν περιπτώσεις όπου η απόδοση παίζει σημαντικό ρόλο σε ένα σύστημα. Και δεν είναι καθόλου απίθανο σε ένα σύστημα όπως η Oracle να υπάρχουν πίνακες με πάρα πολλά δεδομένα μέσα τους. Το θέμα ωστόσο είναι τι γίνεται αν σε περίπτωση που θέλουμε σε τέτοιους πίνακες να αναζητήσουμε συγκεκριμένα δεδομένα, πως θα αποκριθεί το σύστημα. Φανταστείτε πχ σε έναν πίνακα 1000+ γραμμών, πόση ώρα θα κάνει ένα Query για να επιστρέψει αποτέλεσμα. Λογικό δεν είναι να αργήσει, αν χρειαστεί να κάνει Fetch μία-προς-μία της γραμμές του πίνακα; Σαφώς, θα «γονατίσει» μέχρι να βρει αντιστοιχία.

Η λύση που έδωσε η Oracle σε τέτοιες περιπτώσεις είναι την δυνατότητα να μπορούμε να «σπάμε» τους πίνακες σε μικρότερα τμήματα που κρατούν συγκεκριμένα δεδομένα, ώστε να μπορούμε γρηγορότερα να αναζητούμε αυτό που ψάχνουμε χωρίς να μας «γανιάζει» το σύστημα, στην αναμονή. Η όλη διαδικασία ονομάζεται **Κατάτμηση Πίνακα** (Table Partitioning).

ΜΕ την διαδικασία της κατάτμησης μπορούμε να κατακερματίσουμε (Fragment) τους πίνακες σε μικρότερους πίνακες, κάθε ένας από τους οποίους είναι αυτόνομο τμήμα. Και το ακόμα καλύτερο είναι πως η κατάτμηση γίνεται βάση **πεδίου τιμών** (πχ ένα τμήμα για τιμές μικρότερες του 5 και ένα για μεγαλύτερες) ή βάση του hash μιας στήλης. Εμείς θα επιλέξουμε για επίδειξη το πρώτο. Για αυτούς που εκτός από την απόδοση, παίζει και ρόλο και η διαθεσιμότητα, σας έχουμε καλά νέα. Μπορείτε να μετακινείτε τα Partitions σε διαφορετικά Tablespaces.

Στο παρακάτω σχήμα σας δίνουμε ένα ενδεικτικό Concept της κατάτμησης σε διαφορετικά Tablespaces για να καταλάβετε καλύτερα περί τίνος πρόκειται.



Όπως βλέπουμε και στο σχήμα, ρόλο στην διαμέριση παίζουν οι **ΔΕΙΚΤΕΣ** (Indexes) καθώς μέσω αυτών η βάση γνωρίζει που βρίσκεται το κάθε τμήμα του πίνακα.

5.3.3.1 – Δημιουργία Κατάτμησης

Ας πάρουμε το εξής παράδειγμα. Θέλουμε να φτιάξουμε έναν πίνακα βαθμολογιών. Αυτός ο πίνακας προσδοκάται ότι θα φιλοξενήσει πάρα πολλές γραμμές, οπότε στην πάροδο του χρόνου οι αναζητήσεις μας σε αυτόν θα είναι λίγο χρονοβόρες. Ας πούμε τώρα ότι θέλουμε να «σπάσουμε» τον πίνακα σε τρία κομμάτια, ένα για αποθήκευση εγγραφών με βαθμό **κάτω του 3**, ένα για αποθήκευση εγγραφών **κάτω του 7** και ένα για αποθήκευση γραμμών, **κάτω του 10**. Ας μην χρονοτριβούμε λοιπόν, ας το κάνουμε (SQL*PLUS):

```
--Syntax: CREATE TABLE <όνομα πίνακα + (στήλες + {περιορισμούς})>
PARTITION BY RANGE(<στήλη-οδηγός>) (PARTITION <όνομα τμήματος>
VALUES LESS THAN (τιμές στήλης-οδηγού), PARTITION ... VALUES LESS ..)
TABLESPACE <όνομα Tablespace Απόθεσης>
```

```

SQL> CREATE TABLE GRADES
1 (
2     SID INTEGER NOT NULL PRIMARY KEY,
3     NAME VARCHAR2(30),
4     LASTNAME VARCHAR2(30),
5     GRADE NUMBER NOT NULL)
6     PARTITION BY RANGE (GRADE) (
7     PARTITION GRADES_P1 VALUES LESS THAN (3),
8     PARTITION GRADES_P2 VALUES LESS THAN (6),
9     PARTITION GRADES_P3 VALUES LESS THAN (10)) TABLESPACE
USERS;

```

Η κατάτμηση λοιπόν βασίστηκε στην στήλη GRADES που εμείς αυθαίρετα την ονομάζουμε «στήλη-οδηγό ». Δεν χρειάζεται να κάνουμε απολύτως τίποτα καινούριο στην εισαγωγή τιμών, η Oracle αυτόματα βρίσκει τα απαραίτητα τμήματα του πίνακα, και εισάγει τις τιμές σε αυτά. Από εδώ και πέρα, η απόδοση είναι βέβαια, γιατί ταχύτερα βρίσκεις εγγραφές από μικρούς πίνακες παρά από μεγάλους, το είπαμε ξανά αυτό, σωστά;

5.3.3.2 – Μετακίνηση Τμήματος σε Διαφορετικό Tablespace

Αυτή η τεχνική συνεισφέρει την διαθεσιμότητα των δεδομένων του πίνακα. Μπορούμε ένα οποιαδήποτε τμήμα του πίνακα να το μετακινήσουμε σε διαφορετικό Tablespace και αν σε περίπτωση που συμβεί οποιαδήποτε μορφής ζημιά σε ένα, τότε ο πίνακάς μας διατηρεί τις εγγραφές που υπάρχουν σε άλλο Tablespace. Ας δούμε πως γίνεται.

```
--Syntax: ALTER TABLE <όνομα πίνακα> MOVE PARTITION <όνομα τμήματος>
TABLESPACE <όνομα νέου Tablespace>
```

```
SQL> ALTER TABLE GRADES MOVE PARTITION GRADES_P2 TABLESPACE
PREGRADUATE.
```

Η μετακίνηση έγινε, αλλά υπάρχει πρόβλημα. Ο δείκτης της στήλης ήξερε πως το τμήμα GRADES_P2 ήταν στο Tablespace USERS και τώρα δεν μπορεί να το βρει, οπότε δεν μπορούμε INSERT INTO. Τι κάνουμε σε αυτήν την περίπτωση; Ποιος είναι αυτός ο δείκτης που πρέπει να αποσταθούμε; Ευτυχώς που έχουμε το Data Dictionary πρόχειρο. Εκτελούμε επερώτημα στην βάση:

```
SQL> SELECT INDEX_NAME, TABLE_NAME FROM DBA_INDEXES WHERE
TABLE_NAME = 'GRADES';
```

INDEX_NAME	TABLE_NAME
-----	-----
SYS_C0018204	GRADES

Ο δείκτης λέγεται SYS_C0018204. Αυτός λοιπόν «φταίει» για όλα! Ας τον αναδημιουργήσουμε.

```
SQL> ALTER INDEX SYS.SYS_C0018204 REBUILD;
```

Ο δείκτης αναδημιουργήθηκε, και έτσι πλέον ξέρει που βρίσκεται το τμήμα που μετακινήσαμε. Από κει και πέρα, απολαμβάνουμε το νέο μας απόκτημα!

5.3.3.3 – Επανένωση Τμημάτων

Αν για οποιονδήποτε λόγο θέλουμε να κάνουμε επανένωση (Merging) των τμημάτων του πίνακα, πίσω στην αρχική του δομή θα πρέπει γράψουμε μια σειρά εντολών. Θα πρέπει ωστόσο να ακολουθήσουμε κάποιους κανόνες. Επειδή επιλέξαμε κατάτμηση κατά εύρος τιμών, η επανένωση θα πρέπει να γίνει από το partition με το μικρότερο εύρος τιμών, προς το partition με το μεγαλύτερο. Επίσης **μόνο δύο Partitions** την φορά μπορούμε να ενώσουμε. Και προφανώς

θα χρειαστεί να αναδημιουργήσουμε τον δείκτη. Ουφ! Ας το κάνουμε. Ξεκινάμε από την επανένωση.

```
--Syntax ALTER TABLE GRADES MERGE PARTITIONS <partition με μικρότερο  
εύρος τιμών> , <partition με μεγαλύτερο εύρος τιμών> INTO PARTITION <partition  
με το μεγαλύτερο εύρος τιμών>;
```

```
SQL> ALTER TABLE GRADES MERGE PARTITIONS GRADES_P1,  
GRADES_P2 INTO PARTITION GRADES_P2'; --Ένωση p1+p2 στο p2
```

```
SQL>ALTER TABLE GRADES MERGE PARTITIONS GRADES_P2,  
GRADES_P3 INTO PARTITION GRADES_P3; --Ένωση p2+p3 στο p3
```

Μετονομάζουμε το νέο, ενιαίο τμήμα μας, από GRADES_P3 σε GRADES_MAIN με την παρακάτω εντολή:

```
-- Syntax: ALTER TABLE RENAME PARTITION <παλιό όνομα> TO <νέο όνομα>;
```

```
SQL> ALTER TABLE GRADES RENAME PARTITION GRADES_P3 TO  
GRADES_MAIN
```

Μετακινούμε το Ενιαίο Partition από το Tablespace (προαιρετικά) που βρίσκεται, σε ένα νέο:

```
SQL> ALTER TABLE GRADES MOVE PARTITION GRADES_MAIN  
TABLESPACE USERS;
```

Τέλος, αναδημιουργούμε τον δείκτη για να ξέρει πλέον ότι ο πίνακας δεν έχει τμήματα.

```
SQL> ALTER INDEX SYS.SYS_C0018204 REBUILD;
```

Αυτό ήτανε. Ο πίνακας πλέον ενώθηκε σε ένα τμήμα, και μπορεί να χρησιμοποιηθεί ξανά όπως ξέρουμε.

Καλή πρακτική

Είναι γενικά καλή πρακτική να διαμερίζετε πίνακες που προσδοκάτε να δεχτούν μεγάλη ποσότητα δεδομένων για την δική σας ευκολία. Επίσης, η μετακίνηση των τμημάτων σε διαφορετικά Tablespace θεωρείται καλή επιλογή αν θέλετε να σιγουρευτείτε πως οι καίριες εγγραφές του πίνακά σας βρίσκονται σε ξεχωριστό χώρο απόθεσης. Η κατάτμηση επίσης προτείνεται για λειτουργίες Data Warehousing (εκτός σκοπού αυτού του συγγράμματος).

5.3.4 – Data Dictionary

Για να έχει ο διαχειριστής μια ποιο πλήρη εικόνα της βάσης που διαχειρίζεται μπορεί να χρησιμοποιεί το Data Dictionary. Τι είναι όμως το Data Dictionary;

Ορισμός

Το λεξικό δεδομένων (Data Dictionary) είναι ένα σύνολο πινάκων και όψεων **μόνο για ανάγνωση** (δηλαδή μόνο queries μπορούμε να στέλνουμε) που περιέχουν όλες τις πληροφορίες της βάσης, όπως τους υπάρχοντες χρήστες, τους υπάρχοντες πίνακες, τα τρέχοντα Tablespaces και άλλα. Σε αυτά υπάρχουν αποθηκευμένα και **Μεταδεδομένα** (Metadata), γι αυτό το λόγο είναι γνωστά και ως **Αποθετήρια Μεταδεδομένων** (Metadata Repositories)³⁶

Είναι γενικά σωστή η λογική να γνωρίζουμε την υπόσταση της βάσης μας. Δεν νοείται διαχειριστής συστήματος χωρίς να γνωρίζει το σύστημά του εκ τω έσω. Γι αυτό λοιπόν υπάρχει το Data Dictionary, για να βοηθήσει τον διαχειριστή να γνωρίζει με λεπτομέρεια τι συμβαίνει μέσα στην βάση που διαχειρίζεται.

Το Data Dictionary περιέχει τον κατάλληλο πίνακα ή όψη αναλόγως με το τι θέλει ο διαχειριστής να δει στην βάση του. Παρακάτω βλέπουμε ένα παράδειγμα από την εκτέλεση του Query στον πίνακα DBA_USERS που περιέχει τους χρήστες:

³⁶ Wikipedia Article → Data dictionary → “A data dictionary, or metadata repository, as defined ...”, Site (http://en.wikipedia.org/wiki/Data_Dictionary)

```
SQL> select username, user_id, default_tablespace from dba_users;
-----
USERNAME                                USER_ID  DEFAULT_TABLESPACE
-----
MGMT_VIEW                                93       SYSTEM
SYS                                        0        SYSTEM
SYSTEM                                    5        SYSTEM
DBSNMP                                    30       SYSAUX
SYSMAN                                    91       SYSAUX
MARIZA                                    97       USERS
BKEEPER                                   95       USERS
BOAT_KEEPER                              94       USERS
ORACLE                                    99       USERS
OWB                                        96       USERS
VITO                                       90       USERS
```

Query στον πίνακα DBA_USERS για εύρεση χρηστών της βάσης

Οι πίνακες και οι όψεις του Data Dictionary υπάρχουν σε τρεις μορφές³⁷, αναλόγως με τον χρήστη που έχουν να κάνουν. Στον παρακάτω πίνακα τους δείχνουμε.

Τύπος Πίνακα / Όψης	Λειτουργία
DBA	Περιέχουν όλες τις πληροφορίες για την υπόσταση της βάσης. Διαθέσιμη μόνο στους διαχειριστές.
ALL	Περιέχουν πληροφορίες μόνο για τα αντικείμενα που έχει πρόσβαση ο χρήστης.
USER	Περιέχουν πληροφορίες μόνο για τα αντικείμενα που είναι ιδιοκτησία του χρήστη.

Ανάλογα λοιπόν με τον χρήστη που είναι συνδεδεμένος στην βάση, υπάρχουν και τα αντίστοιχα Tables και Views του Data Dictionary που μπορεί να προσπελάσει. Εμείς ως διαχειριστές επικεντρωνόμαστε στους DBA πίνακες/όψεις.

Στον παρακάτω πίνακα σας παραθέτουμε τους σημαντικότερους πίνακες και όψεις που υπάρχουν στην διάθεσή σας.

DBA Πίνακας / Όψη	Περιεχόμενο
DBA_TABLES	Όλοι οι υπάρχοντες πίνακες της βάσης

³⁷ Oracle Documentation → Tables Data Dictionary Views – Site
(http://docs.oracle.com/cd/B28359_01/server.111/b28310/tables014.htm#ADMIN01508)

	δεδομένων
DBA_USERS	Όλοι οι υπάρχοντες χρήστες της βάσης (με τα χαρακτηριστικά τους)
DBA_TABLESPACES	Τα υπάρχοντα Tablespaces της βάσης (ακόμα και τα ανενεργά)
DBA_VIEWS	Όλες οι υπάρχουσες όψεις της βάσης
DBA_CONSTRAINTS	Υπάρχοντες περιορισμοί στους πίνακες της βάσης.
DBA_PROCEDURES	Υπάρχουσες Procedures και Functions στην βάση.

Οι παραπάνω πίνακες και όψεις δέχονται SELECT Queries οπότε χρησιμοποιήστε τους για να αποκομίσετε οποιαδήποτε πληροφορία σας ενδιαφέρει. Μπορείτε επίσης να επιλέγετε συγκεκριμένες στήλες του πίνακα / όψης για να βλέπετε, αφού δείτε τα Metadata του πίνακα με την εντολή DESCRIBE.

5.3.5 – Ο πίνακας DUAL

Ο πίνακας **DUAL** είναι πραγματικά ένας πίνακας «φάντασμα». Τι θέλει να πει ο ποιητής; Θέλει να πει πως, ο πίνακας υπάρχει μεν αλλά δεν περιέχει τίποτα. Και αφού δεν περιέχει τίποτα, τότε τι τον χρειαζόμαστε;

Ε, άμα σας πούμε ότι αυτόν τον πίνακα μπορείτε να τον χρησιμοποιείτε για να εμφανίζετε ότι θέλετε στην έξοδο θα μας πιστεύατε; Πρέπει να μας πιστέψετε όμως γιατί αυτή είναι η χρησιμότητά του.

Ο πίνακας DUAL μπορεί να χρησιμοποιηθεί στο FROM Clause ενός SELECT Statement, και εμφανίζει οτιδήποτε ορίσουμε στο SELECT Statement, σαν να υπήρχαν όντως τα δεδομένα στον πίνακα αυτόν. Με άλλα λόγια, εκτελούμε Queries σε αυτόν τον πίνακα για να λάβουμε ένα **εγγυημένο** (guaranteed) αποτέλεσμα. Και το αποτέλεσμα αυτό μπορεί να είναι οτιδήποτε. Παράδειγμα βλέπουμε κάτω:

```
SQL> SELECT 'CAN YOU SEE ME PAL????' AS MESSAGE FROM DUAL;

MESSAGE
-----
CAN YOU SEE ME PAL????
```

Query στον πίνακα DUAL

Το γεγονός ότι ο DUAL εμφανίζει οτιδήποτε του ζητήσουμε τον καθιστά χρήσιμο εργαλείο δοκιμών αλλά και βοηθητικό εργαλείο για πράξεις και τύπωμα μεταβλητών. Επίσης δέχεται και AS Clause για να δίνετε στην στήλη ότι όνομα θέλετε . Στον παρακάτω πίνακα σας δίνουμε μερικά ενδεικτικά παραδείγματα.

Δυνατότητα	Παράδειγμα SQL
Τύπωμα μηνύματος σε Quotes	<code>SELECT 'HELLO' AS MESSAGE FROM DUAL;</code>
Τύπωμα περιεχομένου Global Μεταβλητής	<code>SELECT :age FROM DUAL;</code>
Τύπωμα Αποτελεσμάτων μαθηματικών πράξεων	<code>SELECT 5+5 AS SUM_RESULT FROM DUAL;</code>
Κλήση και τύπωμα αποτελέσματος από Function (όχι Procedure)	<code>SELECT getLastName('maria') FROM DUAL;</code>
Χρήση με την μορφή Sub-query για αντιπαραβολή πινάκων και δοκιμές περιορισμών	<code>WITH RESULT AS (SELECT 5 AS SID,'Dimitrios' as NAME FROM DUAL) SELECT B.NAME FROM BOATS B, RESULT R WHERE B.ID = R.SID;</code>
Εκχώρηση τιμής σε μεταβλητή σε PL/SQL Block	<code>DECLARE NM VARCHAR2(30); BEGIN SELECT 'A VALUE' INTO NM FROM DUAL; DBMS_OUTPUT.PUT_LINE(NM); END;</code>

Έτσι λοιπόν έχετε έναν πίνακα-βοηθό για όποια δουλειά επιθυμείτε. Σημειώνεται ότι παρά τα ιδιαίτερα χαρακτηριστικά του, ο πίνακας DUAL συμπεριφέρεται όπως ένας κανονικός πίνακας της βάσης, πράγμα που σημαίνει ότι μπορείτε να φωλιάσετε (nest) SELECT Queries προς αυτόν, να κάνετε Concatenate τις στήλες του (με το σύμβολο « || »), να τυπώνετε αποτελέσματα πλάι σε άλλο πίνακα (Καρτεσιανά, Union κλπ) και οτιδήποτε άλλο μπορείτε να κάνετε σε ένα SELECT Statement. INSERT όμως δεν μπορείτε να κάνετε.

Σαν εργαλείο διαχείρισης λοιπόν μπορεί να σας λύσει τα χέρια σε έλεγχο πχ ισχύος περιορισμών ή σαν ένα χρήσιμο **Calculator** καθώς μπορεί να υπολογίσει μαθηματικές πράξεις όπως τις τέσσερις βασικές πράξεις, τετραγωνικές ρίζες (SQRT), υπόλοιπο ακεραίας διαίρεσης (MOD) κλπ.

Keep in mind...

Με το πέρας της εκτέλεσης του Query τα δεδομένα του πίνακα DUAL χάνονται.

5.3.6 – Προβολή Metadata

Πολλές φορές χρειάζεται ο διαχειριστής να βλέπει την υπόσταση κάποιων αντικειμένων (πχ πινάκων) για να γνωρίζει τι τύπο δεδομένων στήλες ο τελευταίος διαθέτει ή άλλες χρήσιμες πληροφορίες. Η πληροφορία αυτές υπάρχουν διαθέσιμες σε κάποια δεδομένα που είναι γνωστά ως **Μεταδεδομένα** (Metadata).

Ορισμός

Μεταδεδομένα (Metadata) ονομάζονται τα δεδομένα που περιγράφουν άλλα δεδομένα, κοινώς δηλαδή δεδομένα για δεδομένα.³⁸ Παρέχουν χρήσιμες πληροφορίες για την υπόσταση αντικειμένων όπως πίνακες

Ας πάρουμε μαζί ένα παράδειγμα. Ο Διαχειριστής θέλει να δει την υπόσταση ενός πίνακα που δημιούργησε ένας χρήστης, ας πούμε τον πίνακα

³⁸ Wikipedia Article → Metadata → “The term metadata refers to “data about data” - Site (<http://en.wikipedia.org/wiki/Metadata>)

LESSONS. Η εντολή που χρειάζεται εδώ για να δούμε τις στήλες που απαρτίζουν τον πίνακα είναι η **DESCRIBE**.

```
--Syntax DESC{RIBE} <όνομα αντικειμένου>
```

```
SQL> DESCRIBE LESSONS
```

```
Name                Null?   Type
-----
LID                  NOT NULL NUMBER(38)
TITLE                VARCHAR2(30)
```

Με την εντολή DESCRIBE διαβάζουμε τα Μεταδεδομένα που αφορούν τον πίνακα, και βλέπουμε την υπόστασή του, δηλαδή πόσες στήλες έχει, το όνομα αυτών, τον τύπο τους και αν έχουν NOT NULL περιορισμό. Την ίδια εντολή μπορούμε να την εφαρμόσουμε σε Procedures, Functions και άλλα αντικείμενα.

Η προβολή των Metadata όμως δεν σταματάει στην εντολή DESCRIBE καθώς η Oracle παρέχει το πακέτο DBMS_METADATA το οποίο περιέχει ιδιότητες (functions) αποκομιδής πληροφοριών για συγκεκριμένα αντικείμενα ή περιορισμούς, τις οποίες μπορείτε να της χρησιμοποιήσετε με τον πίνακα DUAL. Ενδεικτικό παράδειγμα είναι η GET_DDL ιδιότητα, η οποία επιστρέφει όλο το SET εντολών δημιουργίας ενός αντικειμένου, όπως ένας πίνακας. Παράδειγμα, κάτω (SQL Developer):

```
SELECT      DBMS_METADATA.GET_DDL('TABLE','BOATS','SYS')      AS
GET_METADATA FROM DUAL;
```

Και το Query επιστρέφει μια στήλη με όνομα GET_METADATA με τα παρακάτω περιεχόμενα (χωρίς τονισμό σύνταξης).

```
CREATE TABLE "SYS"."BOATS"
(
  "ID" NUMBER(*,0) NOT NULL ENABLE,
  "NAME" VARCHAR2(30),
```

```
PRIMARY KEY ("ID")  
  
USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE  
STATISTICS  
  
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS  
2147483645  
  
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL  
DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)  
  
TABLESPACE "SYSTEM" ENABLE  
  
) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS  
LOGGING  
  
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS  
2147483645  
  
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL  
DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)  
  
TABLESPACE "SYSTEM"
```

Τώρα μάλλον εμείς όταν δημιουργήσαμε τον πίνακα, δεν γράψαμε όλες αυτές τις εντολές, σωστά; Ε, το έκανε η Oracle για μας, το ζήτημα όμως είναι ότι η δομή του πίνακα δημιουργήθηκε από αυτές τις εντολές.

Χρήσιμα Metadata επίσης μπορούν να είναι αυτά που αφορούν το Locale (Character set, μορφή ημερομηνίας και ώρας κλπ) της βάσης μας. Παράδειγμα αποτελεί ο πίνακας NLS_DATABASE_PARAMETERS όπου μπορούμε να δούμε τις πληροφορίες του εκτελώντας ένα απλό Query σε αυτών (SELECT * FROM NLS_DATABASE_PARAMETERS)

```

SQL> SELECT * FROM NLS_DATABASE_PARAMETERS
2 ;

```

PARAMETER	VALUE
NLS_LANGUAGE	AMERICAN
NLS_TERRITORY	AMERICA
NLS_CURRENCY	\$
NLS_ISO_CURRENCY	AMERICA
NLS_NUMERIC_CHARACTERS	.,
NLS_CHARACTERSET	EL8MSWIN1253
NLS_CALENDAR	GREGORIAN
NLS_DATE_FORMAT	DD-MON-RR
NLS_DATE_LANGUAGE	AMERICAN
NLS_SORT	BINARY
NLS_TIME_FORMAT	HH.MI.SSXF AM

```

PARAMETER
```

PARAMETER	VALUE
NLS_TIMESTAMP_FORMAT	DD-MON-RR HH.MI.SSXF AM

Metadata για το Locale της βάσης

Υπάρχουν πολλοί τύποι Metadata που μπορείτε να προβάλετε αλλά είναι πάρα πολλοί για να τους παραθέσουμε εδώ. Χρήσιμο ωστόσο να διαβάζετε τα Metadata στην βάση σας για να γνωρίζετε επακριβώς την δομή της «πιθαμή προς πιθαμή».

Ολοκληρώνοντας...

Αναντίλεκτα, αυτό ήταν το μεγαλύτερο κεφάλαιο του συγγράμματος, καθώς η διαχείριση του Server είναι απαιτητική διαδικασία και ως εκ τούτου θέλουμε να σας προετοιμάσουμε όσο το δυνατόν περισσότερο.

Είδαμε μαζί λοιπόν πολλά χαρακτηριστικά των βάσεων δεδομένων της Oracle, αναλύσαμε τους χρήστες και τους ρόλους τους καθώς και τα κυριότερα εργαλεία διαχείρισης που έχετε στην διάθεσή σας. Προσδοκάται μετά την ανάγνωση του παρόντος κεφαλαίου να έχετε αποκομίσει σημαντικές γνώσεις για το πώς μπορεί να επιτευχθεί μια σωστή διαχείριση ενός Oracle Database Server. Ειδικότερα για την περίπτωση που αφορά το σύγγραμμα, δηλαδή την δημιουργία εκπαιδευτικού περιβάλλοντος με χρήση της Oracle, κάτι τέτοιο καταλαβαίνεται και μόνοι σας ότι χρειάζεται προσοχή, γνώσεις και σιγουριά για να επιτευχθεί σωστή και αποδοτική διαχείριση.

Βιβλιογραφία Κεφαλαίου

A/A	Πηγή	Άρθρο/Ενότητα	Ιστοσελίδα
1	Ask Tom	What is the meaning of schema?	http://asktom.oracle.com/pls/asktom/f?p=100:11:0:::P11_QUESTION_ID:6162110256950
2	Oracle Documentation	DBA_TAB_PRIVS - Related View	http://docs.oracle.com/cd/B19306_01/server.102/b14237/statviews_4151.htm
3	Oracle Documentation	Managing Users and Security - User Privileges and Roles	http://docs.oracle.com/cd/E17781_01/server.112/e18804/users_secure.htm#autold4
4	Oracle Documentation	Assigning a Default Tablespace for the User	http://docs.oracle.com/cd/B28359_01/network.111/b28531/users.htm#DBSEG1020
5	Oracle Documentation	Altering Tablespace Availability	http://docs.oracle.com/cd/B28359_01/server.111/b28310/tspaces005.htm#ADM11376
7	Oracle OTN Forums	Thread “transactional vs. non-transactional database”	https://forums.oracle.com/thread/2348650
8	Wikipedia Article	Database Transaction	http://en.wikipedia.org/wiki/Database_transaction
9	Oracle Documentation	Introduction to Data Warehousing – What is a data Warehouse?	http://docs.oracle.com/cd/E18283_01/server.112/e10578/tdpdw_intro.htm
10	Wikipedia Article	PL/SQL	http://en.wikipedia.org/wiki/PL/SQL
11	Oracle	Starting Up a Database -	http://docs.oracle.com/cd

	Documentation	Options for Starting Up a Database	/B28359_01/server.111/b28310/start001.htm#ADMIN1006
12	Lovely Oracle	ORA-29516: Aurora assertion failure: Assertion failure	http://fahdmirza.blogspot.gr/2011/08/ora-29516-aurora-assertion-failure.html
13	Oracle Documentation	Creating Indexes - Creating an Index Associated with a Constraint	http://docs.oracle.com/cd/B28359_01/server.111/b28310/indexes003.htm#ADMIN11722
14	Wikipedia Article	Data dictionary	http://en.wikipedia.org/wiki/Data_Dictionary
15	Oracle Documentation	Tables Data Dictionary Views	http://docs.oracle.com/cd/B28359_01/server.111/b28310/tables014.htm#ADMIN01508
16	Wikipedia Article	Metadata	http://en.wikipedia.org/wiki/Metadata

ΚΕΦΑΛΑΙΟ 6 – ΣΥΝΤΗΡΗΣΗ ΚΑΙ ΑΣΦΑΛΕΙΑ

Το να διαχειριζόμαστε έναν εξυπηρετητή είναι ένα δύσκολο και υπεύθυνο έργο, ειδικά αν αναλογιστούμε ότι από αυτόν εξυπηρετούνται πολλοί χρήστες. Πρέπει ως εκ τούτου να φροντίζουμε και για την σωστή και απρόσκοπτη λειτουργία του.

Η διαχείριση του Server από μόνη της δεν είναι αρκετή. Καθήκον του διαχειριστή εκτός από το να διαχειρίζεται σωστά τον Server είναι και να το **συντηρεί**. Ο Server μπορεί να είναι ένα μηχάνημα ποιο ισχυρό από τον μέσο υπολογιστή του χρήστη, αλλά δεν παύει και αυτό να είναι ένα μηχάνημα σαν όλα τα άλλα, οπότε είναι και αυτό ευάλωτο πάσης φύσεως δυσλειτουργίες. Και φυσικά χρειάζεται μέριμνα. Σε άλλη περίπτωση, οι συνέπειες μπορεί να είναι καταστροφικές για όλους τους χρήστες, οπότε χρειάζεται προσοχή.

Μερικά από τα προβλήματα που απειλούν την ακεραιότητα και την ορθή λειτουργία του Server μας είναι:

- 1) **Αστοχία Υλικού** (Hardware Failure) όπως χτύπημα δίσκου, προβληματική μνήμη.
- 2) **Δολιοφθορές** (Sabotage) δηλαδή η ηθελημένη πρόκληση ζημιών στον Server όπως DROP TABLE/USER/SCHEMA και οτιδήποτε άλλο απειλεί την υποδομή του εξυπηρετητή.
- 3) **Ανθρώπινο Λάθος**. Είναι στη φύση του ανθρώπου να κάνει λάθη, αυτό δεν μπορεί να αλλάξει. Ωστόσο η μη σωστή εκχώρηση δικαιωμάτων και ρόλων για παράδειγμα σε ένα χρήστη που είναι επιρρεπής σε σφάλματα, μπορεί να αποδειχτεί καταστροφικό είτε γι αυτόν είτε και για άλλους χρήστες.
- 4) **Φυσικά αίτια**, όπως διακοπές ρεύματος ενώ ο Server βρίσκεται σε λειτουργία και υπάρχει συναλλαγή εν εξελίξει.

Δεν χρειάζεται να σας συμβεί κάτι από τα παραπάνω για να αντιληφθείτε την σοβαρότητα της κατάστασης. Πάντα μπορείτε να προνοείτε και να αποτρέπετε τέτοιου είδους προβλήματα. Το κλειδί για την επίτευξη αυτού του στόχου, είναι η σωστή **συντήρηση** (maintenance) του Server. Φροντίστε τον για να σας φροντίσει και αυτός. Σε αυτό το κεφάλαιο θα επικεντρωθούμε στις εργασίες συντήρησης του Server όπως Backup και Restore, ανάκτηση χαμένων κωδικών, ανάκαμψη από σφάλματα και άλλα.

6.1 – Διαθεσιμότητα Βάσεων Δεδομένων

Η βάση δεδομένων του Server πρέπει να είναι πάντοτε διαθέσιμη και λειτουργική, αυτό νομίζω είναι δεδομένο. Τι γίνεται όμως στην περίπτωση που ο δίσκος που την φιλοξενεί, χτυπήσει; Τι γίνεται σε περίπτωση που ενώ ο χρήστης ή ο διαχειριστής εκτελούν συναλλαγή με την βάση, πέσει το ρεύμα; Τι γίνεται αν ο χρήστης ξεχάσει τον κωδικό πρόσβασής του; Και ακόμα χειρότερο σενάριο. Τι γίνεται αν κάποιος εξουσιοδοτημένος ή μη εξουσιοδοτημένος χρήστης (ηθελημένα ή μη) προκαλέσει ανεπανόρθωτες βλάβες στην δομή της; Και εδώ εγώ θα θέσω το εξής ερώτημα. Τι είναι καλύτερα, να αρχίσω να διορθώνω; Ή να φροντίσω να μην χρειαστεί; Η δική σας κρίση τι λέει; Μάλλον κι εσείς τείνετε προς το δεύτερο.

Καλή πρακτική

Όταν πραγματοποιείτε εργασίες συντήρησης στην βάση, καλό θα είναι να την εκκινείτε σε **RESTRICT** Mode για να έχετε πρόσβαση μόνον εσείς σε αυτήν και κανένας άλλος. Δεν θέλετε ενώ πχ πραγματοποιείτε Backup να εργάζεται κάποιος χρήστης σε αυτήν (δεν απαγορεύεται αλλά δεν προτείνεται). Εξυπακούεται ότι οι εργασίες συντήρησης γίνονται μόνο από SYSDBA users.

6.1.1 – Δημιουργία Backup και Restore

Η βάση δεδομένων είχαμε πει πως αποτελείται από αρχεία (Datafiles, Control Files, Redo Logs). Ως εκ τούτου μπορούμε να τα κάνουμε **αντίγραφο**

ασφαλείας (Backup) και, σε περίπτωση, ω μη γένοιτο, τα χρειαστούμε, να τα **ανακτήσουμε** (Restore).

Η Oracle παρέχει το εργαλείο RMAN (Recovery Manager) το οποίο φροντίζει την δουλειά του Backup και του Restore με λίγες σχετικά κινήσεις από πλευράς χρήστη. Σε αυτήν την ενότητα θα δούμε μαζί πως δουλεύει.

6.1.1.1 – Backup με τον RMAN

Για να κάνουμε Backup την βάση μας, χρειάζεται πρώτα να «προετοιμάσουμε το έδαφος». Αρχικά δημιουργούμε έναν κατάλογο στον οποίο θα αποθηκεύονται τα Backups (σε άλλο δίσκο αν υπάρχει αυτή η δυνατότητα). Για λόγους επίδειξης θα δημιουργήσουμε ένα κατάλογο στο home directory του χρήστη Oracle με την κάτωθι εντολή:

```
oracle@debserver$ mkdir /home/oracle/oracle_backup
```

Στην συνέχεια πρέπει να τερματίσουμε την βάση μας, και να την ενεργοποιήσουμε ξανά, αλλά σε λειτουργία ARCHIVELOG³⁹ ώστε να υπάρχει εγγύηση ότι θα γίνει backup της βάσης με όλες τις συναλλαγές που έχουν γίνει COMMIT (με όλες τις αλλαγές που έχουν γίνει). Στο SQL*PLUS συνδεόμαστε σαν SYS as SYSDBA και γράφουμε:

```
SQL> SHUT IMMEDIATE; --Τερματισμός βάσης  
  
Database closed.  
  
Database dismounted.  
  
ORACLE instance shut down.  
  
SQL> STARTUP MOUNT; --Εκκίνηση μόνο με Mount  
  
Database mounted.
```

³⁹ Oracle Documentation → Backing Up Database Files and Archived Logs with RMAN → 4.3.1 Making Consistent and Inconsistent Backups with RMAN → “As long as your database is running in ARCHIVELOG mode...” – Site (http://docs.oracle.com/cd/B19306_01/backup.102/b14192/bkup003.htm)

```
SQL> ALTER DATABASE ARCHIVELOG. -Αλλαγή σε ARCHIVELOG Mode
```

```
Database altered.
```

```
SQL> ALTER DATABASE OPEN -Ανοιγμα βάσης
```

Τώρα που η βάση μας είναι σε λειτουργία ARCHIVELOG μπορούμε να την κάνουμε άφοβα Backup.

Τώρα θα χρησιμοποιήσουμε τον RMAN για να κάνουμε το Backup. Ο RMAN βρίσκεται στον κατάλογο ORACLE_HOME/bin. Εκτελούμε λοιπόν τον RMAN όπως παρακάτω:

```
oracle@debserver$ cd $ORACLE_HOME/bin
```

```
oracle@debserver$ ./rman
```

```
RMAN>
```

Θα πρέπει να συνδεθούμε στην βάση μας. Αφού βεβαιωθούμε ότι έχει τιμή η μεταβλητή ORACLE_SID και ότι η βάση μας είναι ανοιχτή και σε λειτουργία ARCHIVELOG, συνδεόμαστε με την παρακάτω εντολή (ως χρήστης SYS ή SYSTEM).

```
RMAN> CONNECT TARGET SYSTEM/SECRET_PASSWORD
```

Θα πρέπει τώρα να «πούμε» στον RMAN να χρησιμοποιεί τον κατάλογο «/home/oracle/oracle_backup» για να αποθηκεύει τα αντίγραφα ασφαλείας. Αυτό το κάνουμε την παρακάτω εντολή:

```
RMAN> CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT  
'/home/oracle/oracle_backup/ora_bk_timestamp%t_set%s_piece%p';
```

Η μορφοποίηση που δώσαμε (ora_bk_timestamp%t_set%s_piece%p) θα δημιουργεί τα αντίγραφα ασφαλείας με αυτό το πρότυπο. Το %t είναι 4 Byte Timestamp, το %s το Set του αντιγράφου και το %P, το κομμάτι (Piece) που απαρτίζει το αντίγραφο ασφαλείας. Τώρα είμαστε έτοιμοι για Backup.

6.1.1.1.1 – Πλήρες Backup της βάσης

Με τον RMAN μπορούμε να κάνουμε αντίγραφο ολόκληρη την βάση δεδομένων, αυτούσια. Οι εντολή είναι η κάτωθι:

```
RMAN> BACKUP DEVICE TYPE DISK DATABASE PLUS ARCHIVELOG;
```

Η διαδικασία ολοκλήρωσης δημιουργίας αντιγράφου ασφαλείας της βάσης, ενδέχεται να καθυστερήσει λόγο του μεγάλου μεγέθους των Datafiles της. Στην ολοκλήρωση της διαδικασίας θα δείτε παρόμοια οθόνη με την κάτωθι:

```
channel ORA_DISK_1: starting piece 1 at 24-NOV-13
channel ORA_DISK_1: finished piece 1 at 24-NOV-13
piece handle=/home/oracle/oracle_backup/ora_bk_timestamp832309403_set18_piece1.t
ag=TAG20131124T050257 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 24-NOV-13
RMAN>
```

Ολοκλήρωση Backup της βάσης

Για να επιβεβαιώσουμε το αποτέλεσμα, ανοίγουμε ένα δεύτερο Terminal και μεταβαίνουμε στον κατάλογο «/home/oracle/oracle_backup» και ελέγχουμε τα περιεχόμενά του:

```
oracle@debserver$ cd /home/oracle/oracle_backup
oracle@debserver$ ls
```

Θα πρέπει να βρείτε περιεχόμενα μέσα στον φάκελο, με την τυποποίηση που ορίσαμε στο FORMAT του DEVICE TYPE.

```
oracle@DebServer:~/oracle_backup$ ls
ora_bk_timestamp832309378_set17_piece1 ora_bk_timestamp832309403_set18_piece1
oracle@DebServer:~/oracle_backup$
```

Αρχεία αντιγράφων ασφαλείας της βάσης

Έτσι λοιπόν δημιουργήσαμε ένα πλήρες αντίγραφο της βάσης μας και το κρατάμε διαθέσιμο σε περίπτωση που η τελευταία μας «πουλήσει».

6.1.1.1.2 – Μερικό Backup της βάσης

Αν δεν θέλουμε να κάνουμε backup ολόκληρη την βάση, αλλά τμήμα αυτής, μπορούμε να επιλέξουμε συγκεκριμένα συστατικά της. Παρακάτω δίνουμε μερικά παραδείγματα.

Για να κάνουμε Backup ενός **Tablespace** γράφουμε την εντολή:

```
--Syntax: BACKUP DEVICE TYPE DISK TABLESPACE <όνομα tablespace>
```

```
RMAN> BACKUP DEVICE TYPE DISK TABLESPACE USERS;
```

Για να κάνουμε Backup του τρέχοντος **Control File** της βάσης, γράφουμε την εντολή:

```
RMAN> BACKUP DEVICE TYPE DISK CURRENT CONTROLFILE;
```

Για να κάνουμε Backup ενός **Datafile** (βάση του ID του) γράφουμε την εντολή:

```
--Syntax: BACKUP DEVICE TYPE DISK DATAFILE <1,2,3...N>
```

```
RMAN> BACKUP DEVICE TYPE DISK DATAFILE 1; --System Tablespace
```

Έτσι λοιπόν μπορούμε να κάνουμε τμηματικά backup της βάσης μας και να τα χρησιμοποιήσουμε όποτε τα χρειαστούμε (που ελπίζουμε να μην τα χρειαστούμε ποτέ, σωστά;).

Καλή πρακτική

Καλό θα είναι τα Backup σας να γίνονται σε διαφορετικό δίσκο, αν φυσικά υπάρχει εγκατεστημένος στο σύστημα. Αποφεύγετε να αφήνετε αντίγραφα στον ίδιο δίσκο και για λόγο δέσμευσης χώρου και για το λόγο ότι αν ο δίσκος «χτυπήσει» τότε θα χάσετε και τα πρωτότυπα και τα αντίγραφα. Επίσης τα αρχεία μπορείτε να τα μεταφέρετε σε αφαιρούμενες συσκευές αποθήκευσης όπως USB Flash,

Εξωτερικός δίσκος κλπ. Επίσης καλό θα είναι να κάνετε συχνά Backup, πχ ανά βδομάδα ή μήνα.

6.1.1.2 – Ανάκτηση με τον RMAN

Και ενώ ευχόμασταν να μην το πάθουμε, τελικά η «στραβή» συνέβη. Οι λόγοι διάφοροι, όπως ο διαχειριστής κατά λάθος διέγραψε τα αρχεία της βάσης, ή αλλοιώθηκε κάποιο Datafile ή χτύπησε ο δίσκος. Το ζήτημα ωστόσο είναι ένα. Η βάση μας δεν λειτουργεί.

Και τι γίνεται σε αυτήν την περίπτωση; Αν δεν έχουμε αντίγραφα ασφαλείας της βάσης, δεν μπορούμε να κάνουμε τίποτα. Αν έχουμε όμως... πολλά!

Ο RMAN εκτός λοιπόν από Backup αναλαμβάνει και διαδικασίες Restore. Αν έχουμε αντίγραφα ασφαλείας της βάσης μας ή τμήματος αυτής, τότε ο RMAN μπορεί να αναλάβει το λεγόμενο Disaster Recovery και να μας βγάλει από την δύσκολη θέση.

Θα χρησιμοποιήσουμε τα Backups που δημιουργήσαμε νωρίτερα. Θα πρέπει να εκκινήσουμε την βάση μας, αλλά να γίνει μόνο προσάρτηση (mount) ως εξής:

```
SQL> STARTUP MOUNT;
```

--ή μέσα από τον RMAN αν η βάση δεν μπορεί να εκκινηθεί με το SQL*PLUS

--(αφού κάνουμε σύνδεση σε αυτήν με το CONNECT TARGET)

```
RMAN> STARTUP MOUNT FORCE;
```

Τώρα μπορούμε να εκκινήσουμε τον RMAN και να πραγματοποιήσουμε το Restore. Ο RMAN θυμάται τις τοποθεσίες που είχε αποθηκεύσει τα backups οπότε το Restore θα το κάνουμε με δύο κινήσεις. Αρχικά συνδεόμαστε στην βάση:

```
RMAN> CONNECT TARGET SYSTEM/SECRET_PASSWORD
```

Πραγματοποιούμε Restore της βάσης δεδομένων μας πίσω στην προηγούμενη κατάστασή της (μαζί με όλα τα αρχικά της αρχεία) ως εξής:

```
RMAN> RESTORE DATABASE;
```

Η διαδικασία ενδέχεται να αργήσει να ολοκληρωθεί. Αφήστε το όμως, αξίζει τον κόπο. Αν η διαδικασία ολοκληρωθεί σωστά τότε θα πρέπει να λάβετε το ακόλουθο μήνυμα:

```
channel ORA_DISK_1: reading from backup piece /home/oracle/oracle_backup/ora_bk_
timestamp832313160_set28_piece1
channel ORA_DISK_1: piece handle=/home/oracle/oracle_backup/ora_bk_timestamp8323
13160_set28_piece1 tag=TAG20131124T060600
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:01:15
Finished restore at 24-NOV-13
RMAN> █
```

Ολοκλήρωση Restore

Το Restore ολοκληρώθηκε με επιτυχία και η βάση μας επανήλθε. Δεν μπορούμε όμως ακόμα να την χρησιμοποιήσουμε γιατί πρέπει να την κάνουμε και Recover (επαναφορά) για να αποκατασταθούν και τα Datafiles στην θέση τους. Πάλι μέσα από τον RMAN εκτελούμε:

```
RMAN> RECOVER DATABASE;
```

Αν το Recovery ολοκληρώθηκε με επιτυχία, τότε θα πρέπει να λάβετε παρόμοιο μήνυμα με το ακόλουθο:

```
RMAN> recover database;
Starting recover at 24-NOV-13
using channel ORA_DISK_1
starting media recovery
media recovery complete, elapsed time: 00:00:08
Finished recover at 24-NOV-13
RMAN> █
```

Ολοκλήρωση Recover

Τώρα μπορούμε να χρησιμοποιήσουμε την βάση μας. Μέσα από το SQL*PLUS αλλάζουμε την κατάσταση της βάσης από closed σε open.

```
SQL> ALTER DATABASE OPEN;
```

Η βάση μας θα ανοίξει και είναι έτοιμη για εργασία. Το RMAN μας έσωσε. Άρα αυτό μας δίδαξε πόσο σημαντική αξία έχουν τα αντίγραφα ασφαλείας (και όχι μόνο στην βάση δεδομένων, αλλά παντού).

Keep in mind...

Αν στον RMAN δεν του «πείτε» σε ποιον κατάλογο θέλετε να αποθηκεύει και να ανακτά Backups, από προεπιλογή «βλέπει» στον κατάλογο ORACLE_BASE/flash_recover_area. Επίσης αν προσπαθήσετε να κάνετε Restore και Recover ενός Tablespace, φροντίστε να το έχετε θέσει σε κατάσταση **Offline**.

6.1.2 – Virtual Private Database

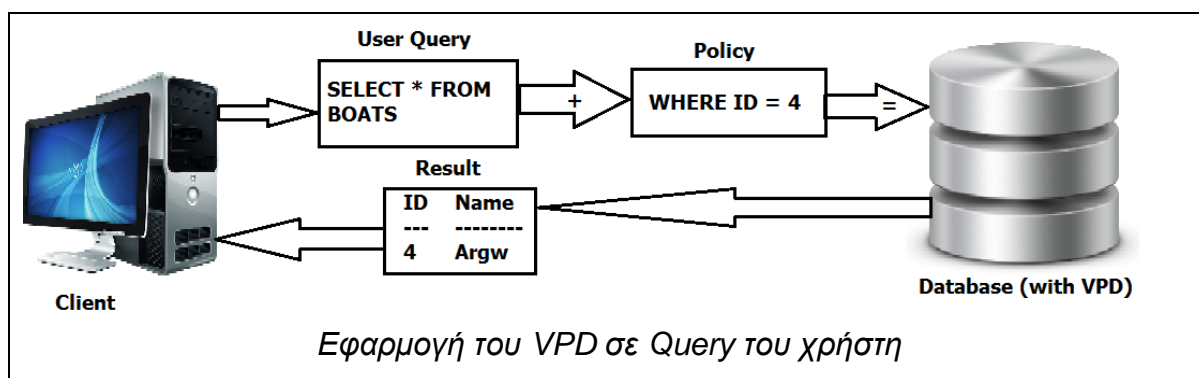
Η Oracle παρέχει έναν ωραίο μηχανισμό για την προστασία των ευαίσθητων δεδομένων από τους πίνακες που περιέχονται στην βάση, οποίος ονομάζεται Virtual Private Database ή VPD για συντομία. Θα εξηγήσουμε τι είναι αυτός ο μηχανισμός με ένα παράδειγμα.

Ας πούμε ότι έχουμε ένα πίνακα ονόματι BOATS με 4 εγγραφές. Ο χρήστης lefteris θέλουμε μεν να έχει πρόσβαση στα περιεχόμενα του πίνακα αλλά δεν θέλουμε να βλέπει τις τρεις πρώτες γραμμές, γιατί αποτελούν ευαίσθητο προσωπικό δεδομένων. Ο χρήστης lefteris είναι ιδιοκτήτης της βάρκας Argw με ID = 4. Οι υπόλοιπες βάρκες δεν είναι δικές του, οπότε δεν θέλει ο διαχειριστής ο χρήστης αυτός να έχει πρόσβαση σε αυτές τις γραμμές. Τι κάνουμε σε αυτήν την περίπτωση; Η απάντηση βρίσκεται στο Virtual Private Database.

Ο Μηχανισμός Virtual Private Database, είναι ένας μηχανισμός ασφαλείας που βασίζεται σε **πολιτικές** (policies) **πρόσβασης**. Κάθε πολιτική πρόσβασης που δημιουργεί ο διαχειριστής, είναι στην ουσία ένας κανόνας πρόσβασης για τον

εκάστοτε χρήστη που αφορά, ο οποίος εφαρμόζεται κάθε φορά που ο χρήστης προσπαθεί να προσπελάσει έναν πίνακα που περιέχει ευαίσθητα προσωπικά δεδομένα. Έτσι στο χρήστη εμφανίζονται μόνο τα δεδομένα που θέλει ο διαχειριστής και όχι όσα δεν τον αφορούν.

Στην πράξη το Virtual Private Database, εφαρμόζει μια πολιτική κάθε φορά που ο χρήστης εκτελεί επερώτηση στην βάση (SELECT). Αν ο χρήστης για παράδειγμα ζητήσει να δει όλες τις γραμμές του πίνακα BOATS (SELECT * FROM BOATS) τότε εφαρμόζεται άμεσα η πολιτική πρόσβασης, που δεν είναι τίποτε άλλο παρά ένα προκαθορισμένο WHERE Clause. Έτσι αν ο διαχειριστής θέλει ο χρήστης για παράδειγμα να βλέπει μόνο την γραμμή με ID = 4 τότε δημιουργεί την πολιτική πρόσβασης και όποτε ο χρήστης εκτελεί Queries στην βάση για τον προστατευμένο πίνακα, τότε αυτόματα προσκολλάται (appends) το WHERE Clause στο SELECT * FROM Statement. Αν ο χρήστης ορίσει το Query του, δικό του WHERE Clause που είναι διαφορετικό από αυτό που εφαρμόζει η πολιτική, τότε δεν θα του επιστραφούν δεδομένα. Στο παρακάτω σχήμα βλέπουμε τον τρόπο λειτουργίας του VPD.



6.1.2.1 - Δημιουργία πολιτικής

Τώρα όμως θα το εφαρμόσουμε στην πράξη. Θυμηθείτε το παραπάνω παράδειγμα με την βάρκα. Ο πίνακας BOATS βρίσκεται στο Schema BKEEPER και περιέχει 5 εγγραφές, εκ των οποίων τον χρήστη lefteris τον αφορά μονάχα η γραμμή με ID = 4 (Argw). Συνεπώς, πρέπει να δημιουργήσουμε μια πολιτική και να την εφαρμόσουμε. Αλλά πριν το κάνουμε, πρέπει πρώτα να δημιουργήσουμε μια PL/SQL Function που θα μας επιστρέφει το WHERE Clause της πολιτικής.

Συνδεόμαστε στην βάση ως χρήστες SYS as SYSDBA είτε από το SQL*PLUS είτε από το SQL Developer. Δημιουργούμε το παρακάτω Function:

```

CREATE OR REPLACE FUNCTION only_1(

  schema_var IN VARCHAR2, --Όρισμα 1 = Schema

  table_var IN VARCHAR2 --Όρισμα 2 = Πίνακας προς προστασία

)

RETURN VARCHAR2

IS

  return_val VARCHAR2 (400); --Μεταβλητή που θα κρατάει το WHERE Clause

BEGIN

  return_val := 'ID = 4 '; --(WHERE) ID = 4

  RETURN return_val; --Επιστροφή του WHERE Clause

END ;

/

COMMIT;
    
```

Αυτό το Function θα εκτελείται σε κάθε εφαρμογή της πολιτικής και θα επιστρέφει το WHERE Clause που θα προσκολλάται στο Query των χρηστών. Τώρα μπορούμε να δημιουργήσουμε την πολιτική μας, με τον παρακάτω κώδικα:

```

BEGIN

  dbms_rls.add_policy( --Κλήση Procedure DBMS_RLS.ADD_POLICY

  object_schema => 'bkeeper', --Schema που περιέχει τον πίνακα

  object_name => 'boats', --Ο πίνακας που θέλουμε να προστατέψουμε

  policy_name => 'only_1', --Όνομα πολιτικής

  function_schema => 'SYS', --Schema που βρίσκεται το Function only_1
    
```

```

policy_function => 'ONLY_1', --η Function only_1

statement_types => 'select,', --Δικαιώματα πίνακα (SELECT)

update_check => FALSE , --Όχι update

enable => TRUE , --Ενεργοποίηση μετά τη δημιουργία

static_policy => FALSE , --Όχι στατικό Policy

policy_type => dbms_rls.SHARED_CONTEXT_SENSITIVE , --Είδος πολιτικής =
--Προστασία δεδομένων πίνακα

long_predicate => FALSE); --Όχι σύνθετο WHERE

END;

/

COMMIT;

```

Η πολιτική δημιουργήθηκε , οπότε πλέον πρέπει να την δοκιμάσουμε. Συνδεόμαστε σαν χρήστης Lefteris στο SQL*PLUS ή SQL Developer και εκτελούμε Query για τον πίνακα BOATS (SELECT * FROM BKEEPER.BOATS). Το αποτέλεσμα φαίνεται παρακάτω:

```

SQL> select * from bkeeper.boats;

```

ID	NAME
4	Argw

```

SQL>

```

Αποτέλεσμα Απλού Query με εφαρμογή πολιτικής

Βλέπουμε πως επιστρέφει μόνο τη γραμμή με ID = 4. Τώρα ας δοκιμάσουμε να τρέξουμε το Query με δικό μας WHERE Clause (SELECT * FROM BOATS WHERE ID BETWEEN 1 AND 3);

```
SQL> select * from bkeeper.boats where ID between 1 and 3 ; Query
no rows selected Αποτέλεσμα
SQL>
```

Αποτέλεσμα Query με WHERE Clause

Βλέπουμε πως το WHERE Clause του χρήστη, δεν συμφωνεί με το WHERE Clause της πολιτικής, οπότε το Query δεν επιστρέφει τίποτα. Η πολιτική μας λειτουργεί!

Η πολιτική επίσης επιτρέπει την προσθήκη δικαιωμάτων ενεργειών στον πίνακα (SELECT,UPDATE,INSERT,DELETE). Είναι στην κρίση σας τι θα επιλέξετε.

Keep in mind...

Αν η Function που εφαρμόζει η πολιτική περιέχει συντακτικά σφάλματα, οι χρήστες δεν θα μπορούν να εκτελούν Queries στους πίνακες (Function has errors).

6.1.2.2 – Διαγραφή Πολιτικής

Για να διαγράψετε μια πολιτική από την βάση, εκτελέστε τον παρακάτω κώδικα ως SYS as SYSDBA.

```
BEGIN
DBMS_RLS.DROP_POLICY('BKEEPER','BOATS','ONLY_1');
--Ορίσματα = 1) Schema, 2) Πίνακας, 3) Όνομα πολιτικής
END;
/
DROP FUNCTION only_1; --Διαγραφή και του Function της πολιτικής
```

COMMIT;

Η πολιτική διαγράφηκε μαζί με το Function που εφαρμόζε το WHERE Clause. Τώρα θα συνδεθούμε ως χρήστες lefteris και θα εκτελέσουμε το ίδιο Query (SELECT * FROM BKEEPER.BOATS).

```
SQL> select * from bkeeper.boats;
```

ID	NAME
1	Mariza
2	Eliza
3	Megalochari
4	Argw
5	Evaggelistria

```
SQL>
```

Query χωρίς την πολιτική πλέων

Βλέπουμε πως πλέον το Query επιστρέφει όλες τις γραμμές του πίνακα. Δεν συνίσταται να διαγράφετε τις πολιτικές σας, εκτός αν υπάρχει σοβαρός λόγος για να το κάνετε.

Καλή πρακτική

Καλό θα είναι να εφαρμόζετε τις πολιτικές του Virtual Private Database στους πίνακες που περιέχουν πολύ ευαίσθητα δεδομένα. Παρέχει καλό βαθμό ασφαλείας για τους «περίεργους». Επίσης την διαδικασία δημιουργίας και διαγραφής πολιτικών μπορείτε να την κάνετε πιο γρήγορα και εύκολα μέσα από το Oracle Enterprise Manager στην περιοχή Server → Security → Virtual Private Database.

6.1.3 – Κλειδωμα Tablespaces

Η Oracle σας δίνει την δυνατότητα να αλλάξετε ένα Permanent Tablespace σε λειτουργία **μόνο για ανάγνωση** (Read Only) όπου επιτρέπεται μόνο οι χρήστες να εκτελούν Queries στην βάση και όχι DDL (INSERT INTO κλπ). Αυτό εγγυάται την διαθεσιμότητα των δεδομένων και αποτρέπει την τροποποίησή τους από τους χρήστες, αν είναι ζωτικής σημασίας δεδομένα που δεν πρέπει να αλλάξουν. Θα κλειδώσουμε το Tablespace PREGRADUATE. Γράφουμε στο SQL*PLUS την εντολή (ως SYS as SYSDBA):

```
SQL> ALTER TABLESPACE PREGRADUATE READ ONLY;
```

Με την εκτέλεση της εντολής αυτής οι χρήστες δεν μπορούν πλέον να τροποποιήσουν τα δεδομένα του Tablespace, καθώς αν προσπαθήσουν το σύστημα θα τους αποτρέψει. Για να αναιρέσετε την διαδικασία, αλλάξτε το ONLY σε WRITE.

6.1.4 –Flashback Queries

Να μια αναποδιά που συνέβη. Στον πίνακα BOATS, η γραμμή με ID=2 είχε το όνομα Eliza και έγινε UPDATE στον πίνακα με το όνομα Danaï. Και ακόμα χειρότερα; Έγινε COMMIT!

Θυμάστε τι είχαμε πει; Όταν γίνονται COMMIT τότε οι αλλαγές μένουνε στην βάση, και διαγράφονται από το Undo Tablespace οι αλλαγές. Κοινώς δηλαδή μετά το COMMIT δεν υπάρχει γυρισμός, οπότε δεν μπορούμε να κάνουμε ROLLBACK. Τι γίνεται όμως τώρα που η στήλη του πίνακα πρέπει να ανακτήσει την προηγούμενη τιμή του; Δεν υπάρχει μια δεύτερη ευκαιρία;

Ευτυχώς για μας, υπάρχουν τα Metadata στο Undo Tablespace που κρατάνε **για κάποιο διάστημα** τις παλιές τιμές των πινάκων, σε ένα χώρο του Tablespace αυτού, που ονομάζεται **Rollback Segment**. Οπότε μπορούμε να κάνουμε ένα SELECT Query και να λάβουμε την παλιά τιμή του πίνακα. Η

τεχνολογία αυτή επιτρέπει την ανάκτηση των παλιών τιμών των πινάκων ακόμη και μετά από COMMIT, και ονομάζεται **Flashback Query**.

Ας δούμε μαζί πως λειτουργεί. Επανερχόμαστε στο παλιό παράδειγμα, με τον πίνακα BOATS. Η αλλαγή του ονόματος έγινε στις 16:00 (μεσημεριάτικα δηλαδή!). Εκτελούμε ένα απλό Query στην βάση για να διαπιστώσουμε ότι όντως η αλλαγή είναι εν ισχύ:

```
SQL> select * from boats;
```

ID	NAME
1	Galena
2	Danai
3	Marilena
4	Theoskepasti
5	Megalochari

Απλό Query για διαπίστωση της αλλαγής της τιμής στη γραμμή 2

Οι αλλαγές έχουν εφαρμοστεί, και εφόσον κάναμε COMMIT τα πράγματα είναι σκούρα. Ή μήπως όχι;

Θα εκτελέσουμε στην βάση ένα Flashback Query για την ίδια μέρα, αλλά για την κατάσταση του πίνακα στις 13:00, δηλαδή τρεις ώρες πίσω. Η εντολή είναι η παρακάτω:

```
SELECT * FROM boats  
  
AS OF TIMESTAMP  
  
TO_TIMESTAMP('23-NOV-13 13:30:00', 'DD-MM-YY HH24:MI:SS')  
  
--Μορφή = Μέρα(2)-Μήνας(2)-Έτος 24ωρη μορφή_Ώρας:Λεπτά:Δευτερόλεπτα  
  
WHERE ID = 2;
```

Εκτελούμε λοιπόν το Query και βλέπουμε τα αποτελέσματα που μας επιστρέφει.

```
SQL> SELECT * FROM boats
AS OF TIMESTAMP
TO_TIMESTAMP('23-NOV-13 13:30:00', 'DD-MM-YY HH24:MI:SS')
WHERE ID = 2;
```

ID	NAME
2	Eliza

```
SQL>
```

Αποτέλεσμα του Flashback Query

Βλέπουμε πως η τιμή υπάρχει ακόμα στα Metadata του Undo Tablespace. Πρέπει οπωσδήποτε να την ανακτήσουμε εκεί που ήταν πριν είναι αργά. Γράφουμε τις κάτωθι εντολές:

```
UPDATE BOATS SET NAME =
(SELECT NAME FROM BOATS AS OF TIMESTAMP
TO_TIMESTAMP('23-NOV-13 13:30:00', 'DD-MM-YY HH24:MI:SS')
WHERE ID = 2 )
WHERE ID = 2
COMMIT; -- ←Οπωσδήποτε!!!!
```

Με την εκτέλεση της εντολής ο πίνακας αποκτάει πάλι την παλιά του τιμή. Αυτό πρέπει να το διαπιστώσουμε όμως με τα μάτια μας. Εκτελούμε πάλι ένα απλό Query στην βάση μας και βλέπουμε τις αλλαγές

```
SQL> select * from boats;
```

ID	NAME
1	Galena
2	Eliza
3	Marilena
4	Theoskepasti
5	Megalochari

```
SQL>
```

Αποτέλεσμα Query για διαπίστωση επαναφοράς τιμής

Ανάσα! Η παλιά τιμή του πίνακα είναι στη θέση της. Τώρα πλέον μπορούμε να ανασάνουμε!

Τα Flashback Queries λοιπόν, μπορούνε να σας σώσουνε αν αντιληφθείτε το ταχύτερο δυνατόν την αλλαγή που δεν έπρεπε να γίνει.

Keep in mind...

Τα Metadata του Undo Tablespace, όσο περνάει ο καιρός **πανωγράφονται**, οπότε τις αλλαγές φροντίστε να τις αντιλαμβάνεστε εγκαίρως πριν να είναι αργά.

Καλή πρακτική

Η Oracle από προεπιλογή την παράμετρο UNDO_MANAGEMENT την έχει ορισμένη σε AUTO. Αν θέλετε να δημιουργήστε περισσότερη «περίοδο χάριτος» είτε αλλάξετε την τιμή UNDO_RETENTION (πχ ALTER SYSTEM SET UNDO_RETENTION = 2400) είτε αλλάξετε την τιμή της σε MANUAL (μέσα από το αρχείο ORACLE_HOME/dbs/init<SID>.ora ενώ είναι **κλειστή**) και δημιουργήστε δικά σας, μεγαλύτερα σε μέγεθος, Rollback Segments με τις παρακάτω εντολές (Ενδεικτικά):

```
SQL> CREATE ROLLBACK SEGMENT roll1 TABLESPACE TSTUNDO  
STORAGE (INITIAL 20 M NEXT 5 M MAXEXTENTS UNLIMITED);
```

(τα κόκκινα τα αλλάζετε εσείς)

```
SQL> ALTER ROLLBACK SEGMENT roll1 ONLINE;
```

Μεγάλα σε μέγεθος Rollback Segments κρατάνε πληροφορίες για περισσότερο χρόνο⁴⁰.

6.1.5 – Εξαγωγή Πινάκων Βάσης Σε XML

Η Oracle μας επιτρέπει να εξαγάμουμε πίνακες με τα δεδομένα τους σε αρχεία XML. Ακόμα καλύτερα, παρέχει το αποθετήριο XML DB το οποίο είναι

⁴⁰ *Burleson Consulting* → *ORA-01555 Snapshot Too Old* → “Action: If in Automatic Undo Management mode, increase the setting of UNDO_RETENTION. Otherwise, use larger rollback segments.” – Site (http://www.dba-oracle.com/t_ora_01555_snapshot_old.htm)

προσβάσιμο από τον Web Browser μέσω HTTP και από εκεί μπορούμε να τα κατεβάσουμε από οπουδήποτε. Και τα XML αρχεία είναι μεταφέρσιμα οπότε μπορούμε πάντα να τα έχουμε διαθέσιμα σε άλλη τοποθεσία, καθώς με αυτήν την μορφή έχουμε ένα είδος Backup.

Θα δουλέψουμε στο SQL Developer, για να έχουμε ευκολία στην εκτέλεση των απαιτούμενων PL/SQL Blocks που θα χρειαστούμε για να δημιουργήσουμε τα αρχεία XML μαζί με τα απαιτούμενα Tags και Attributes. Το Concept είναι το εξής. Θα εξάγουμε τον πίνακα BOATS σε XML αρχείο, με τρία Tags τα οποία θα είναι <rowset> <boat> <name>. Τα δεδομένα της στήλης ID θα τα εισάγουμε σαν Attributes στο Element, boat ενώ τα ονόματα θα βρίσκονται στα name elements. Μην το καθυστερούμε όμως, ας το κάνουμε:

Ας δούμε πρώτα τα δεδομένα του πίνακά μας, εκτελώντας SELECT στην βάση:

ID	NAME
1	Galena
2	Eliza
3	Marilena
4	Theoskepasti
5	Megalochari
6	Makrina
7	Alluette

Δεδομένα πίνακα Boats

Κάθε γραμμή του πίνακα θα είναι το XML Element με όνομα Rowset. Μέσα σε αυτό θα υπάρχει το Element boat με attribute το ID της βάρκας και τελικό Element το name που θα κρατάει το όνομα της βάρκας. Η μορφή δηλαδή θα είναι:

```
<rowset> <boat id="1"> <name> Galena </name></boat></rowset>
```

Τα XML Tags/Element δημιουργούνται από την Stored Procedure XMLELEMENT ενώ τα Attributes από την XLMATTRIBUTES. Θα τις

ενσωματώσουμε όλες σε SELECT όπου θα παίρνει το κάθε στοιχείο και θα δημιουργεί το ανάλογο XML Element. Και όλο μαζί θα το βάλουμε σε ένα ανώνυμο PL/SQL Block. Ο Κώδικας είναι ο παρακάτω:

```

DECLARE

XDB CLOB; --Μεγάλος τύπος χαρακτήρων (μεγαλύτερο από VARCHAR)

res BOOLEAN; --Δημιουργία «πόρου» δηλαδή αρχείο στο Repository

BEGIN

  FOR      x      IN      (SELECT      XMLELEMENT("rowset",XMLAGG(
XMLELEMENT("boat",XMLATTRIBUTES(ID                               AS
"bid"),XMLELEMENT("name",NAME))))).GETCLOBVAL() AS XML FROM boats)

  LOOP

/*

Το For...In loop παίρνει τα αποτελέσματα που επιστρέφει το SELECT από τις
Procedures που δημιουργούν τα XML Elements και τα Attributes αυτών στην
μεταβλητή X. Η XMLELEMENT δημιουργεί τα Elements, Η XMLAGG τα συνενώνει
κάτω από αυτό, και η XMLATTRIBUTES δημιουργεί το Attribute στο Element που
δημιουργεί εκείνη τη στιγμή η XMLELEMENT. Το αποτέλεσμα εξάγεται με όνομα
XML και αποθηκεύεται στην X αφού γίνει Cast σε CLOB με την GETCLOBVAL.

*/

      XDB := XDB||X.XML; --Αποθήκευση X στην XDB μαζί με τα περιεχόμενα που
--ήδη έχει η XDB (αντίστοιχο += στην Java/C

      RES := DBMS_XDB.CREATERESOURCE('/public/xmlboats.xml', XDB);

--Δημιουργία του «πόρου» δηλαδή εγγραφή στο αρχείο που βρίσκεται στο
--Repository (αφού πρώτα δημιουργηθεί).

END LOOP;

```

```
COMMIT; --Απαραίτητα COMMIT για να αποθηκευτούν οι αλλαγές
```

```
END;
```

Εκτελούμε το PL/SQL Block μας και τα αρχεία θα δημιουργηθούν. Τώρα όμως θα πρέπει να το διαπιστώσουμε. Αφού βεβαιωθούμε ότι έχουμε ξεκλειδώσει τον λογαριασμό του χρήστη ANONYMUS ανοίγουμε τον Web Browser και μεταβαίνουμε στην διεύθυνση:

```
http://<server_Address>:8080/public
```

Θα μας ζητήσει διαπιστευτήρια, οπότε εισάγετε τα στοιχεία του SYS. Τότε θα δείτε λογικά το αρχείο μας να μας περιμένει μέσα στο Αποθετήριο Public.



The screenshot shows a web browser window with the address bar displaying 'debserver:8080/public/'. The browser tabs include 'Πιο συχνά αναγνωσμέ...', 'Getting Started', 'Latest Headlines', and 'phpMyAdmin 3.4.5'. The address bar shows 'Index of /public/' with a '+' button. The main content area displays the title 'Index of /public/' and a table with the following data:

Name	Last modified	Size
xmlboats.xml	Sat, 24 Nov 2013 15:18:12 GMT	312

The file 'xmlboats.xml' is circled in red. Below the table, the text 'Το XML αρχείο που εξάγαμε' is displayed.

Πατώντας κλικ επάνω του μπορείτε να το ανοίξετε. Θα πρέπει να δείτε περιεχόμενα παρόμοια με τα παρακάτω:

```

- <rowset>
  - <boat bid="1">
    <name>Galena</name>
  </boat>
  - <boat bid="2">
    <name>Eliza</name>
  </boat>
  - <boat bid="3">
    <name>Marilena</name>
  </boat>
  - <boat bid="4">
    <name>Theoskepasti</name>
  </boat>
  - <boat bid="5">
    <name>Megalochari</name>
  </boat>
  - <boat bid="7">
    <name>Makrina</name>
  </boat>
  - <boat bid="8">
    <name>Alluette</name>
  </boat>
</rowset>

```

Περιεχόμενα αρχείου XML

Είδατε που περιέχει ακριβώς τα στοιχεία που είχε ο πίνακας μας; Μπορείτε αν θέλετε να εκτελέσετε **XQuery** στα αρχεία XML μέσα από την Stored Procedure XMLQUERY. Ενδεικτικό παράδειγμα δίνεται παρακάτω (επιλογή ονόματος βάση του attribute bid = 1) :

```

SELECT XMLQUERY(
'
for $x in doc("/public/xmlboats.xml")/rowset/boat
where $x/@bid = 1
return $x/name
'
RETURNING CONTENT) AS BOATNAME FROM DUAL;

```


Το αποτέλεσμα φαίνεται παρακάτω:

The screenshot shows an XQuery query in a blue editor window:

```
SELECT XMLQUERY(
|
|
| for $x in doc("/public/xmlboats.xml")/rowset/boat
| where $x/@bid = 1
| return $x/name
|
|
| RETURNING CONTENT) AS BOATNAME FROM DUAL;
```

Below the query, the result is displayed in a table with the following content:

BOATNAME
1 <name>Galena</name>

At the bottom of the screenshot, the text "Αποτελέσματα XQuery" is centered.

Λειτουργεί άψογα. Τελευταίο πράγμα που μπορείτε αν κάνετε είναι μέσα από τον κατάλογο Public να κατεβάσετε το αρχείο XML στον υπολογιστή σας, επιλέγοντας το όνομα του αρχείου με δεξί κλικ και πατώντας «Save Link As» όπως βλέπουμε παρακάτω:

The screenshot shows a file index titled "Index of /public/". The index has the following columns: Name, Last modified, and Size.

Name	Last modified	Size
xmlboats.xml	Sat 24 Nov 2013 15:18:12 GMT	312

A context menu is open over the [xmlboats.xml](#) file. The menu items include:

- Άνοιγμα δεσμού σε νέα καρτέλα
- Open Link in IE Tab
- Open Link in Ext.App.
- Άνοιγμα δεσμού σε νέο παράθυρο ιδιωτικής περιήγησης
- Ο δεσμός ως σελιδοδείκτης
- Αποθήκευση δεσμού ως...** (highlighted with a red box)
- Αντιγραφή τοποθεσίας δεσμού
- FlashGot: δεσμός (Ctrl+F1)
- FlashGot: όλα (Ctrl+F3)
- Δημιουργία συλλογής
- Επιλογές FlashGot
- Έλεγχος αντικειμένου

At the bottom of the screenshot, the text "Αποθήκευση αρχείου XML από το Repository" is centered.

Η εξαγωγή πινάκων της βάσης μας σε XML παρέχουν μεγάλη ευελιξία καθώς μπορείτε να τα χρησιμοποιήσετε οπουδήποτε απαιτείται η ύπαρξη XML αρχείων (πχ σε Website), ή σαν μιας μορφής αντιγράφου ασφαλείας των περιεχομένων ενός πίνακα. Είπαμε, τα XML μεταφέρονται, είναι (συνήθως) μικρά σε μέγεθος, και απόλυτα επεκτάσιμα.

6.1.6 – Εξαγωγή Βάσης σε Dump Αρχείο

Η Oracle επιτρέπει την μεταφορά την εξαγωγή αυτούσιας της βάσης δεδομένων ή των πινάκων της σε ένα αρχείο που ονομάζεται Dump File, στο οποίο περιέχονται όλες οι SQL εντολές δημιουργίας της καθώς και των χρηστών της, των αντικειμένων της και οτιδήποτε άλλο περιέχεται σε αυτήν (ακόμα και τους ρόλους και τα Grants).

Το να έχουμε ένα Dump αρχείο με όλες τις εντολές δημιουργίας της βάσης και του περιεχομένου της, μας δίνει την ευελιξία να έχουμε ένα αρχείο-αναφορά για μελλοντική επαναδημιουργία της, επακριβώς όπως ήτανε (μιας μορφής backup-restore).

Η Oracle παρέχει το εργαλείο EXP (Export) το οποίο μπορούμε να χρησιμοποιήσουμε, για να κάνουμε Dump την βάση. Βασική προϋπόθεση είναι να είναι ανοιχτή. Το εργαλείο βρίσκεται στον κατάλογο ORACLE_HOME/bin. Χρειαζόμαστε λοιπόν τις κάτωθι εντολές:

```
oracle@debserver$ cd $ORACLE_HOME/bin
```

```
oracle@debserver$ ./exp
```

Το εργαλείο είναι διαδραστικό, δηλαδή σας ρωτάει τι να κάνει πριν κάνει το οτιδήποτε. Οι επιλογές που σας δίνει είναι οι παρακάτω:

```
Username: sys as sysdba
```

```
Password:
```

```
Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit
Production
```

```
Enter array fetch buffer size: 4096 > #<μέγεθος buffer>
```

```
Export file: expdat.dmp > #path αρχείου dump πχ /home/oracle/oracle_backup
```

```
(1)E(ntire database), (2)U(sers), or (3)T(ables): (2)U > 1 #Επιλογή εξαγωγής
```

```
# 1 = Ολόκληρη βάση , 2= Χρήστες, 3 = Πίνακες
```

```
Export grants (yes/no): yes > #εξαγωγή Grants
```

```
Export table data (yes/no): yes > #Εξαγωγή δεδομένων πινάκων
```

```
Compress extents (yes/no): yes > #Συμπίεση των Extents των Tablespace
```

Η διαδικασία θα ξεκινήσει άμεσα. Αν επιλέξατε να εξαγάγετε ολόκληρη την βάση δεδομένων, τότε η διαδικασία θα πάρει λίγο χρόνο για να ολοκληρωθεί. Στο πέρας της εξαγωγής, αν όλα πήγαν καλά, θα πρέπει να λάβετε το ακόλουθο μήνυμα:

```
. exporting statistics
Export terminated successfully with warnings.
oracle@DebServer:/u01/app/oracle/product/11.2.0/dbhome/bin$
```

Ολοκλήρωση Export

Το αρχείο μας υπάρχει στην τοποθεσία που ορίσαμε κατά την διάδραση με το EXP. Αν το ανοίξετε, θα δείτε όλες τις εντολές SQL που περιέχει για την δημιουργία της βάσης ή των πινάκων της. Τμήμα του αρχείου βλέπουμε παρακάτω:

```
<...>
```

```
METRICSE
```

```
BEGINSYS
```

```
CONNECT SYSTEM
```

```
METRICSEtablespace definitions

CONNECT SYSTEM

CREATE TABLESPACE "SYSAUX" BLOCKSIZE 8192 DATAFILE
'/u01/app/oracle/oradata/te$

CREATE UNDO TABLESPACE "UNDOTBS1" BLOCKSIZE 8192 DATAFILE
'/u01/app/oracle/ora$

CREATE TEMPORARY TABLESPACE "TEMP" BLOCKSIZE 8192 TEMPFILE
'/u01/app/oracle/or$

CREATE TABLESPACE "USERS" BLOCKSIZE 8192 DATAFILE
'/u01/app/oracle/oradata/tes$

CREATE TABLESPACE "FLOW_1044714046206499" BLOCKSIZE 8192
DATAFILE '/u01/app/or$

METRICEE 5

METRICSEprofiles

CREATE PROFILE "MONITORING_PROFILE" LIMIT COMPOSITE_LIMIT
DEFAULT SESSIONS_PER_$

METRICEE 6

<...>
```

Έτσι πάντα έχουμε ένα αρχείο-οδηγό για την δημιουργία της βάσης μας μέσω εντολών, το οποίο φυσικά μπορούμε να μεταφέρουμε.

Troubleshoot:

Η διαδικασία σταματάει με το σφάλμα «ORA-29516: Aurora assertion failure: Assertion failure at joez.c: Bulk load of method java/lang/Object.<init> failed; insufficient shm-object space». Τι κάνω σε αυτήν την περίπτωση;

Λύση:

Το πρόβλημα αυτό προκύπτει λόγω του JAVA JIT Debugger (Just In Time) ο οποίος σε αρκετές περιπτώσεις προκαλεί προβλήματα, οπότε πρέπει να τον απενεργοποιήσουμε. Ανοίξτε το SQL*PLUS και συνδεθείτε ως SYS as SYSDBA. Εκτελέστε την παρακάτω εντολή:

```
SQL> ALTER SYSTEM SET JAVA_JIT_ENABLED=FALSE;
```

```
System Altered.
```

Κλείστε το SQL*PLUS και δοκιμάστε ξανά. Λογικά η διαδικασία θα ολοκληρωθεί.

6.1.7 – Εισαγωγή Βάσης από Dump Αρχείο

Τώρα θα κάνουμε το αντίστροφο, θα χρησιμοποιήσουμε το Dump Αρχείο για να δημιουργήσουμε την βάση από αυτό. Το αντίστοιχο εργαλείο που παρέχει η Oracle γι αυτόν τον σκοπό είναι το IMP (Import) το οποίο βρίσκεται στον κατάλογο ORACLE_HOME/bin. Εκτελούμε τις παρακάτω εντολές:

```
oracle@debserver$ cd $ORACLE_HOME/bin
```

```
oracle@debserver$ ./imp
```

Το πρόγραμμα IMP είναι επίσης διαδραστικό, και παρέχει τις παρακάτω εντολές κατά την προτροπή ενεργειών:

```
Username: sys as sysdba
```

```
Password:
```

```
Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit  
Production
```

```
With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

```
Import data only (yes/no): no > #Επιλογή εισαγωγής μόνο δεδομένων
```

```
Import file: expdat.dmp > #Path προς dump file
```

```
Enter insert buffer size (minimum is 8192) 30720> #Buffer Ανάγνωσης  
Export file created by EXPORT:V11.02.00 via conventional path  
import done in WE8ISO8859P1 character set and AL16UTF16 NCHAR character  
set  
import server uses EL8MSWIN1253 character set (possible charset conversion)  
List contents of import file only (yes/no): no > #Λίστα περιεχομένων  
Ignore create error due to object existence (yes/no): no > #αγνόηση λαθών  
#από υπάρχοντα αντικείμενα και παραβίαση περιορισμών  
Import grants (yes/no): yes > # εισαγωγή grants  
Import table data (yes/no): yes > # εισαγωγή δεδομένων πινάκων  
Import entire export file (yes/no): no > #Εκτέλεση όλων των εντολών μέχρι #τέλος
```

Η διαδικασία θα ξεκινήσει άμεσα. Να έχετε υπόψη σας, ότι η διαδικασία αναδημιουργίας της βάσης και των περιεχομένων της είναι **πολύ χρονοβόρα**.

6.2 – Λοιπές Εργασίες

Σε αυτήν την ενότητα θα δούμε μερικές πρόσθετες διαδικασίες συντήρησης και διαχείρισης για να πάρουμε μια πληρέστερη εικόνα σχετικά με το τι άλλο μπορείτε να κάνετε στην βάση.

6.2.1 – Ανάκτηση Password Διαχειριστών και Χρηστών

Άνθρωποι είμαστε, οπότε λογικό είναι κάποιες φορές να ξεχνάμε. Έτσι είναι και με τους διαχειριστές και χρήστες της βάσης, πάντα υπάρχει η πιθανότητα κάποιος να ξεχάσει τον κωδικό του. Το θέμα είναι, τι γίνεται σε αυτήν την περίπτωση. Η απάντηση είναι απλή. Μην αγχώνεστε!

6.2.1.1 – Ανάκτηση password των χρηστών και διαχειριστών της βάσης

Η Oracle δίνει την δυνατότητα στους διαχειριστές να αλλάζουν το Password των χρηστών ή άλλων διαχειριστών, παρακάμπτοντας το παλιό του. Έτσι ο κωδικός μπορεί να αλλάξει με την παρακάτω εντολή (SQL*PLUS):

```
--Syntax: ALTER USER <χρήστης> IDENTIFIED BY <νέο password>
```

```
SQL> ALTER USER LEFTERIS IDENTIFIED BY NewSecret;
```

Το ίδιο ισχύει για τους διαχειριστές που ξεχάσανε τα Passwords τους. Γι αυτό το λόγο άλλωστε υπάρχουν περισσότεροι του ενός χρήστες διαχειριστή στο σύστημα. Έτσι, αν για παράδειγμα ο χρήστης SYS ξεχάσει το Password του, τότε μπορεί να συνδεθεί ο χρήστης SYSTEM as SYSDBA και να αλλάξει τον κωδικό του SYS με την παραπάνω εντολή.

6.2.1.2 – Ανάκτηση Passwords Διαχειριστών του Apex

Αν ξεχάσετε το Password του Γενικού διαχειριστή του APEX μπορείτε να το κάνετε Reset μέσα από το script «apxchpwd.sql» που βρίσκεται στον κατάλογο ORACLE_HOME/apex. Ανοίξτε το SQL*PLUS, συνδεθείτε σαν SYS as SYSDBA και τρέξτε το Script:

```
SQL> @$ORACLE_HOME/apex/apxchpwd.sql
```

Ακολουθήστε τις οδηγίες που σας δίνει και εισάγετε νέο Password. **Προσοχή** ωστόσο, το password που θα βάλετε με την εκτέλεση του Script είναι **προσωρινό**. Μεταβείτε στην σελίδα login του apex (/apex όχι στην σελίδα διαχείρισης /apex_config) και κάντε login ως admin στο Internal Workspace. Εκεί

θα σας ζητήσει να αλλάξετε το Password όπως την πρώτη φορά που ενεργοποιήσατε το APEX. Την επόμενη φορά ωστόσο να είστε προσεκτικοί!

6.2.2 – Απενεργοποίηση χρηστών

Αν έχετε στην βάση χρήστες που δεν την χρησιμοποιούν συχνά ή τους θεωρείτε ύποπτους για μη-επιτρεπτές ενέργειες μπορείτε απλά να απενεργοποιήσετε το λογαριασμό τους, χωρίς να χρειαστεί να τους διαγράψετε και να χαθούν τα δεδομένα τους. Συνδεθείτε το SQL*PLUS ως χρήστες SYS as SYSDBA και εκτελέστε την παρακάτω εντολή:

```
--Syntax: ALTER USER <χρήστης> ACCOUNT LOCK;
```

```
SQL> ALTER USER LEFTERIS ACCOUNT LOCK;
```

Η παραπάνω εντολή κλειδώνει το λογαριασμό του χρήστη και αποκλείονται οποιοσδήποτε ευκαιρίες σύνδεσης και εργασίας με την βάση. Για να ενεργοποιήσετε ξανά τον λογαριασμό του χρήστη αντικαταστήστε το LOCK με UNLOCK.

Ολοκληρώνοντας...

Είδαμε λοιπόν σε αυτό το κεφάλαιο μερικά από τα πιο σημαντικά Concepts που σχετίζονται με τον παράγοντα Συντήρησης και Ασφάλειας. Θεωρούμε πως πλέον έχετε μια ιδέα για το τι πρέπει να κάνετε για να διατηρείτε τον Server σας σε καλή κατάσταση, γιατί απλά όσο τον προσέχετε, τόσο θα λειτουργεί αποδοτικά. Δεν πρέπει σε καμία περίπτωση να αμελείτε να συντηρείτε τον Server σας, έστω και στις πολύ βασικές εργασίες, ειδικά αν ο Server σας εξυπηρετεί μεγάλο αριθμό ατόμων, όπως φοιτητές ή εργαζόμενους. Take good care of it, and it will take a good care of you!

Κάπου εδώ τελειώνει ο χώρος των διαχειριστών και μεταβαίνουμε στον χώρο των απλών χρηστών, στο παράγοντα που δίνει «ζωή» στο σύστημα. Στο επόμενο κεφάλαιο παρουσιάζουμε πολλά από τα εργαλεία που είδαμε στο κεφάλαιο 5 αλλά από την σκοπιά των χρηστών αυτήν την φορά. Το εκπαιδευτικό σύστημα που δημιουργήσαμε στον εξυπηρετητή του Τμήματος Μηχανικών Πληροφορικής, χρησιμοποιεί τα εργαλεία αυτά, οπότε αυτό το κεφάλαιο είναι ιδανικό για τους σπουδαστές.

Βιβλιογραφία Κεφαλαίου

A/A	Πηγή	Άρθρο/Ενότητα	Ιστοσελίδα
1	Oracle Documentation	Backing Up Database Files and Archived Logs with RMAN	http://docs.oracle.com/cd/B19306_01/backup.102/b14192/bkup003.htm
2	Oracle Documentation	Export	http://docs.oracle.com/cd/B10501_01/server.920/a96652/ch01.htm
3	Oracle Documentation	Import	http://docs.oracle.com/cd/B10501_01/server.920/a96652/ch02.htm
4	Burleson Consulting	ORA-01555 Snapshot Too Old	http://www.dba-oracle.com/t_ora_01555_snapshot_old.htm
5	Oracle Documentation	Using Oracle Flashback Technology	http://docs.oracle.com/cd/E11882_01/appdev.112/e41502/adfns_flashback.htm#ADFNS1008
6	Oracle Documentation	Using Oracle Virtual Private Database to Control Data Access	http://docs.oracle.com/cd/B28359_01/network.111/b28531/vpd.htm#DBSEG007

ΚΕΦΑΛΑΙΟ 7 – ΣΥΝΔΕΣΗ ΚΑΙ ΕΡΓΑΣΙΑ ΧΡΗΣΤΩΝ

Στα προηγούμενα κεφάλαια δώσαμε έμφαση στην λειτουργία και ενασχόληση του συστήματος, από τη σκοπιά του Server. Σε αυτό το κεφάλαιο θα μεταβούμε στην ενασχόληση του συστήματος από την πλευρά του χρήστη/πελάτη. Θα δούμε μαζί τα σημαντικά εργαλεία που παρέχει η Oracle στους χρήστες της και θα ανακαλύψουμε μαζί τις δυνατότητες που παρέχει το εν λόγω σύστημα στους χρήστες αυτούς. Άλλωστε, το σύστημα προορίζεται γι αυτούς σε τελική ανάλυση, συμφωνείτε;

Σε αυτό το κεφάλαιο λοιπόν θα επικεντρωθούμε στις εφαρμογές και στα περιβάλλοντα Client που ο χρήστης θα χρησιμοποιεί προκειμένου να αλληλεπιδρά με την βάση, θα επεκταθούμε στον προγραμματισμό εφαρμογών για εφαρμογές που αλληλεπιδρούν με την βάση και θα βγάλουμε μαζί κάποια λογικά συμπεράσματα σχετικά με την απόδοση και την παραγωγικότητα του συστήματος.

7.1 – Client Εφαρμογές και Περιβάλλοντα

Αφού λοιπόν ο Server είναι «up and running», τότε δεν μένει παρά να αρχίσουν οι χρήστες να συνδέονται σε αυτόν και να αρχίσουν να εργάζονται. Ο διαχειριστής του συστήματος, φρόντισε να δημιουργήσει τους απαραίτητους χρήστες, να τους αποδώσει δικαιώματα ή/και ρόλους, τους γνωστοποίησε τους προσωπικούς τους κωδικούς πρόσβασης και τους παραχώρησε αποθηκευτικό χώρο σε κάποιο Tablespace. Οπότε ο χρήστης είναι παραπάνω από έτοιμος να αρχίσει να ασχολείται με το σύστημα.

Εκτός από τα διαπιστευτήρια του, ο χρήστης τι άλλο χρειάζεται να γνωρίζει προκειμένου να συνδεθεί στην βάση δεδομένων; Η απάντηση είναι, την **IP Διεύθυνση** του Server (ή το Domain Name του), την **θύρα** στην οποία ακούει ο Listener και το **Instance ID (SID)** της βάσης δεδομένων που τρέχει στο Server.

Στην ενότητα αυτή θα δούμε τα κυριότερα ολοκληρωμένα περιβάλλοντα εργασίας, που θα γίνουν ο προσωπικός τους χώρος εργασίας. Ας τα δούμε λοιπόν.

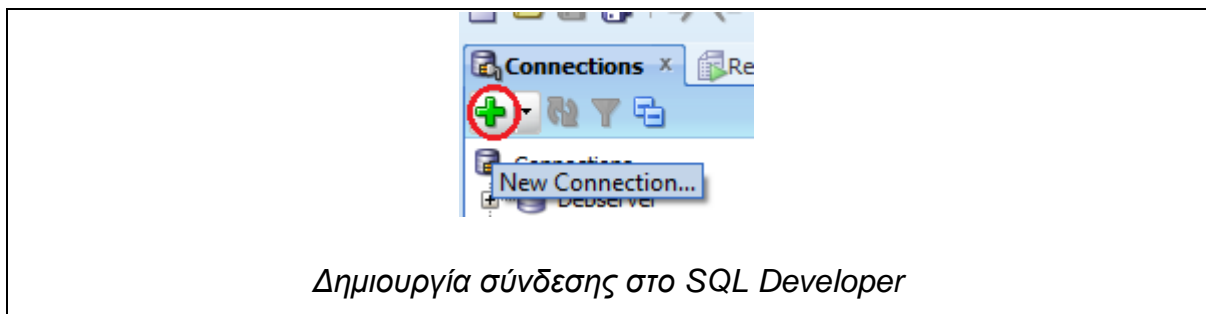
7.1.1 – Το Oracle SQL Developer

Χωρίς αμφιβολία, πρόκειται για το πιο συνηθισμένο αλλά και το πιο πλήρες περιβάλλον εργασίας σε μια βάση δεδομένων της Oracle. Το καλύψαμε σε προηγούμενο κεφάλαιο, αλλά τώρα θα επεκταθούμε στην λειτουργία του από έναν απλό χρήστη / πελάτη, μπαίνοντας σε λίγο περισσότερες λεπτομέρειες.

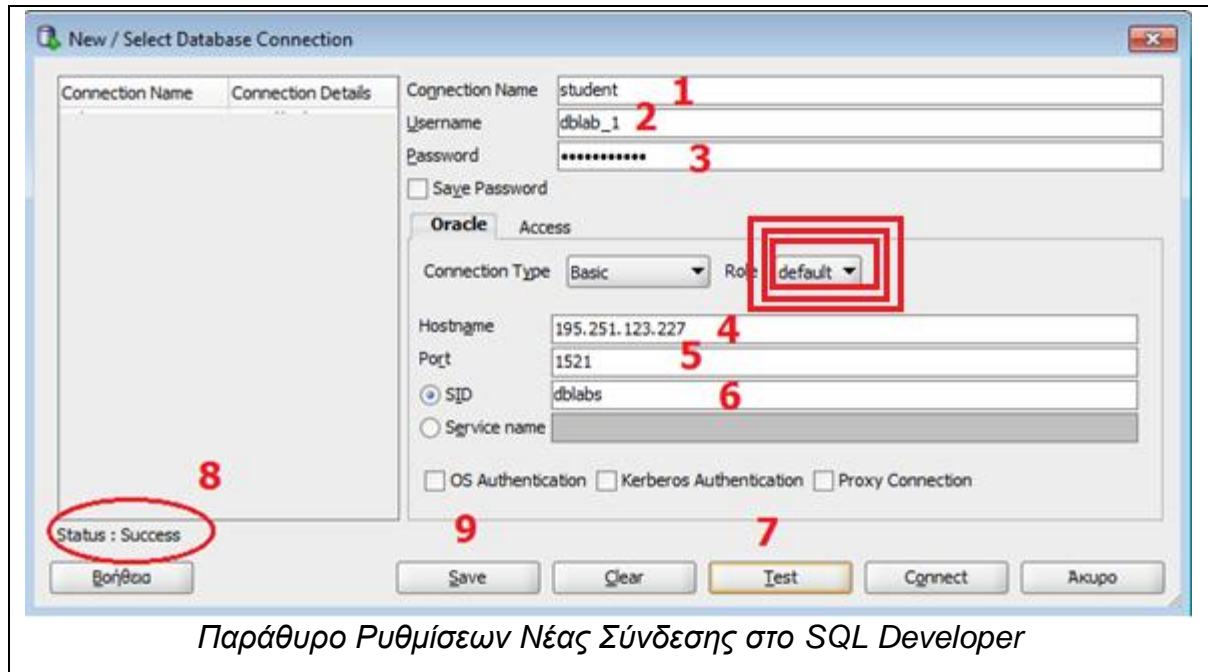
7.1.1.1 - Σύνδεση στην βάση δεδομένων

Η σύνδεση στην βάση δεδομένων από έναν απλό χρήστη δεν διαφέρει και πολύ από την διαδικασία σύνδεσης ενός διαχειριστή. Οι διαφορές είναι απειροελάχιστες. Ο διαχειριστής της βάσης δεδομένων, θα πρέπει να σας έχει γνωστοποιήσει τα διαπιστευτήρια σύνδεσής σας καθώς και επιπλέον πληροφορίες, όπως διεύθυνση του Server και SID της βάσης. Δεν πρέπει όμως ο χρήστης να αγχωθεί, αυτά θα γίνουν μόνο μια φορά.

Αρχικά θα πρέπει να μεταβούμε στην αριστερή περιοχή της εφαρμογής, στην καρτέλα «Connections» και να πατήσουμε κλικ στο κουμπί «New» (πράσινος σταυρός) όπως βλέπουμε στην παρακάτω εικόνα:



Στην συνέχεια θα μας εμφανιστεί το παράθυρο επιλογών της νέας σύνδεσης το οποίο το βλέπουμε μαζί παρακάτω:



Πριν ξεκινήσουμε να περιγράψουμε τα αριθμημένα βήματα, προσέξτε στην εικόνα την έντονη σήμανση στον επιλογέα Role. Ως απλοί χρήστες το αφήνουμε πάντα στην τιμή **DEFAULT**.

Τα στοιχεία που χρειάζεται να εισάγουμε είναι τα εξής:

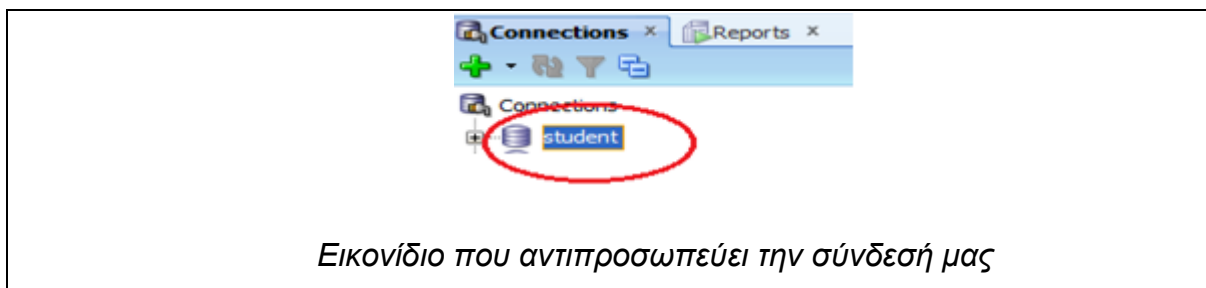
- **Connection Name** (1) = Πρόκειται για το όνομα της σύνδεσης. Έχει **τοπική σημασία** μόνο στο SQL Developer, για να αναγνωρίζει ο χρήστης τον εξυπηρετητή που θέλει να συνδεθεί, οπότε εισάγετε ότι θέλουμε.
- **Username** (2) = Εδώ εισάγουμε το όνομα χρήστη που έχουμε στην βάση δεδομένων.
- **Password** (3) = Εδώ εισάγουμε τον προσωπικό μας κωδικό πρόσβασης που μας έχει δοθεί από τους διαχειριστές. Από κάτω βλέπουμε ένα Checkbox με λεζάντα «Save Password», το οποίο αν το επιλέξουμε, δεν θα μας ζητάει Password σε κάθε απόπειρα σύνδεσής μας. Επιλέξτε το μόνο αν ο υπολογιστής που εργάζεστε είναι ιδιωτικός (π.χ. σπιτιού).
- **Hostname** (4) = Εδώ μπορούμε να εισάγουμε είτε την IP Διεύθυνση του Server της βάσης, ή το Domain Name του τελευταίου. Ο διαχειριστής της

βάσης λογικά θα πρέπει να μας έχει γνωστοποιήσει κάποιο από τα δύο παραπάνω.

- **Port** (5) = Εδώ εισάγουμε την TCP Port όπου ακούει ο Listener για εισερχόμενες συνδέσεις. Συνήθως είναι η 1521 (προεπιλογή της Oracle) ωστόσο αν ο διαχειριστής έχει επιλέξει άλλη τότε θα πρέπει υποχρεωτικά να μας την γνωστοποιήσει.
- **SID** (6) = Εδώ πρέπει να εισάγουμε το Instance ID της βάσης δεδομένων που πρόκειται να συνδεθούμε. Επαναλαμβάνουμε ότι είναι καθήκον του διαχειριστή να το γνωστοποιήσει στους χρήστες.

Κάπου εδώ θα πρέπει να δοκιμάσουμε την σύνδεσή μας. Αφού εισάγουμε τα στοιχεία μας πατάμε στο πλήκτρο «Test» (7) για να γίνει δοκιμή. Αν όλα πάνε καλά, τότε θα λάβουμε το μήνυμα «**Success**» στην λεζάντα «Status» (8). Διαφορετικά θα μας αναφέρει για πιο λόγο δεν μπορεί αν συνδεθεί. Δεν υπάρχει όμως λόγος πανικού. Μπορεί απλά να έχουμε εισάγει λάθος διαπιστευτήρια. Σε άλλη περίπτωση, είναι υποχρέωση του διαχειριστή να διευθετήσει το θέμα. Ολοκληρώνουμε πατώντας στο πλήκτρο «Save» (9) για να αποθηκεύσουμε την σύνδεση μας. Πλέον είμαστε έτοιμοι για σύνδεση!

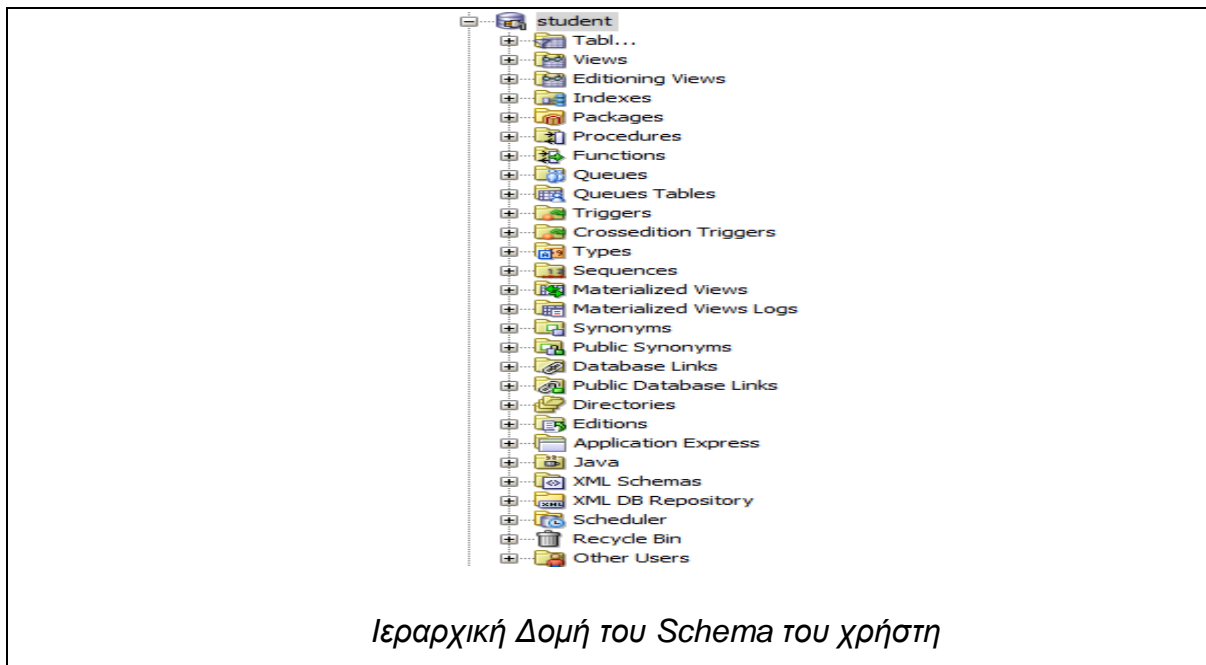
Πατώντας Save, μπορούμε να κλείσουμε το παράθυρο. Τώρα θα πρέπει να μεταβούμε ξανά στην καρτέλα «Connections» Θα δούμε κάτι καινούριο να μας περιμένει εκεί, που δεν είναι άλλο παρά η νέα μας σύνδεση προς τον Server.



Για να εκκινήσουμε την σύνδεση, πατούμε δεξί-κλικ πάνω στο εικονίδιο με το όνομα της σύνδεση μας (το όνομα που δώσαμε στο πεδίο «Connection Name» όταν ρυθμίζαμε την νέα μας σύνδεση) και στο Drop-down μενού που εμφανίζεται, επιλέγουμε «Connect». Αν όλα πάνε καλά, η σύνδεση θα εγκαθιδρυθεί σε σύντομο

χρονικό διάστημα (σημείωση, αν δεν επιλέξατε το checkbox «Save Password» τότε θα σας ζητηθεί να το εισάγετε).

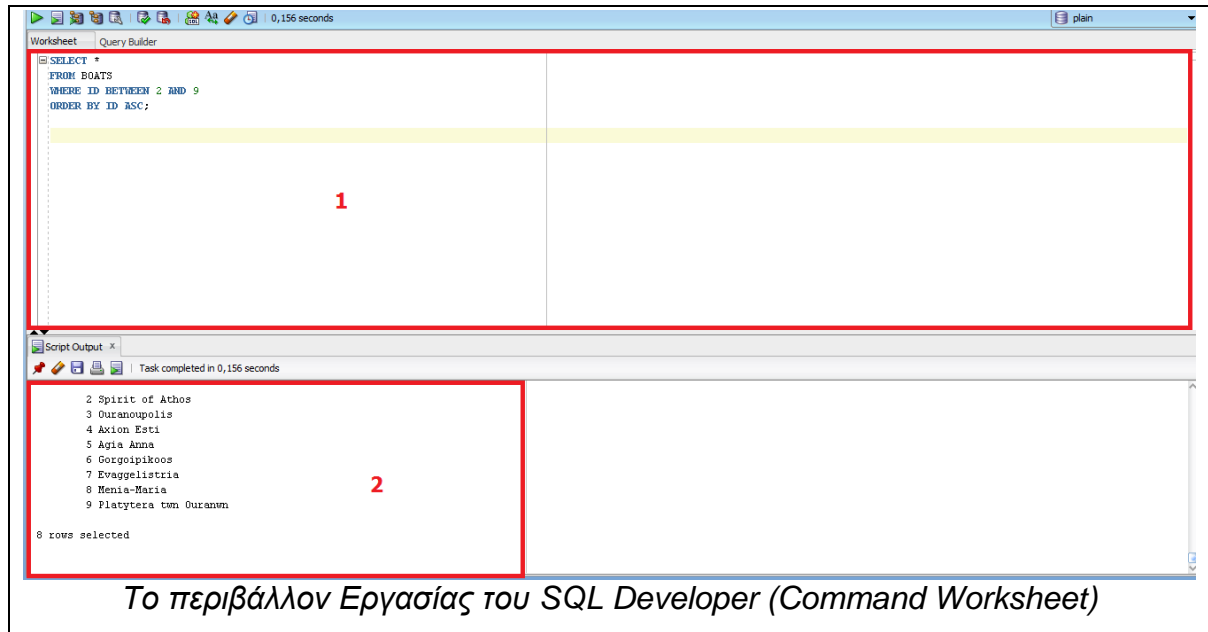
Η σύνδεσή μας με τον Server επετεύχθη, και αυτό το καταλαβαίνουμε από την **Ιεραρχική Δομή** ολόκληρου του Schema του χρήστη, το οποίο το βλέπουμε μαζί στην παρακάτω εικόνα:



Το έργο μας έφτασε εις πέρας και πλέον μπορούμε να αρχίσουμε να εργαζόμαστε στο SQL Developer.

7.1.1.2 – Εργασία στο Oracle SQL Developer

Πλέον συνδεθήκαμε στην βάση και μπορούμε να αλληλεπιδράσουμε με αυτήν. Σε κάθε μας σύνδεση στο SQL Developer, μας διατίθεται το ακόλουθο περιβάλλον εργασίας:



Διακρίνουμε αρχικά το **Command Worksheet** (1) στο οποίο θα γράφουμε τις εντολές SQL και PL/SQL. Παρακάτω βλέπουμε την περιοχή **Script Output** (2), στην οποία θα εμφανίζονται τα κάτωθι:

- 1) Αποτελέσματα από επερώτημα SELECT (πίνακες και όψεις)
- 2) Έκβαση Εκτέλεσης Εντολών (επιτυχία ή αποτυχία και λόγος αυτής)
- 3) Έξοδος εντολής DBMS_OUTPUT.PUT_LINE (τύπωση στο STDOUT)

Βλέπουμε λοιπόν ότι έχουμε ότι χρειαζόμαστε για να «βομβαρδίσουμε» την βάση με τις εντολές μας. Αυτός είναι και ο απώτερος σκοπός άλλωστε.

7.1.1.2.1 – Command Worksheet

Το Command Worksheet είναι ίσως το περισσότερο χρησιμοποιούμενο εργαλείο στο SQL Developer. Μέσα σε αυτό λοιπόν μπορούμε να εισάγουμε τις SQL και PL/SQL εντολές μας, με την σειρά που θέλουμε να εκτελεστούν. Ένα πολύ ωραίο χαρακτηριστικό του Worksheet είναι ότι διαθέτει **τονισμό σύνταξης** (Syntax Highlighter), δηλαδή τις δεσμευμένες λέξεις της SQL και PL/SQL τις τονίζει

έντονα και με διαφορετικό χρώμα. Ακόμη ένα πολύ ωραίο χαρακτηριστικό, είναι ότι διαθέτει **αυτόματη συμπλήρωση** (Auto-completion) ενώ συγγράφουμε εντολές, πράγμα που διευκολύνει αφάνταστα τον χρήστη, ιδιαίτερα σε μακροσκελής εντολές. Αρκετά όμως με το παίνεμα, ας δούμε τι μπορεί να κάνει, στην πράξη.

Στο SQL Developer, μπορούμε να εισάγουμε και να εκτελέσουμε διάφορους τύπους εντολών, δηλαδή **Standard SQL Statements, PL/SQL Blocks** και **Scripts από αρχείο**. Τα πάντα δηλαδή!

7.1.1.2.1.1 – Standard SQL Statements

Η σύνταξη εντολών SQL στο SQL Developer μπορεί να γίνει με πολλούς τρόπους, ανάλογα με την εκάστοτε ενέργεια που απαιτείται να γίνει. Οι τρόποι αυτοί περιγράφονται παρακάτω. Στόχος μας δεν είναι η διδασχή της SQL ή της PL/SQL, απλά η ενημέρωσή σας σχετικά με το τι είδους εντολές μπορείτε να συντάξετε.

Μεμονωμένα SQL Statements. Οι μεμονωμένες εντολές SQL μπορούν να γραφτούν αυτούσιες. Αν και δεν είναι υποχρεωτικό, είναι **καλή πρακτική** να τις γράφετε με **Κεφαλαία** και να τις τερματίζετε με **ελληνικό ερωτηματικό (;)**. Τέτοιες εντολές μπορεί να είναι SELECT...FROM...WHARE, INSERT INTO...VALUES, UPDATE...SET...WHERE κλπ.

Παράδειγμα:

```
SELECT *  
FROM BOATS  
WHERE ID BETWEEN 2 AND 9;
```

Πολλαπλά όμοια SQL Statements. Υπάρχουν περιπτώσεις όπου χρειάζεται να εισάγουμε πολλαπλές εντολές, **με την σειρά που θέλουμε να εκτελεστούν**, όπως πολλαπλά INSERT INTO Statements.

Παράδειγμα:


```
INSERT INTO BOATS VALUES (5,'Mariza');
```

```
INSERT INTO BOATS VALUES (6,'Anna');
```

```
INSERT INTO BOATS VALUES (7,'Sofia');
```

Πολλαπλά διαφορετικά SQL Statements. Σε άλλες περιπτώσεις καλούμαστε να γράψουμε πολλαπλά SQL Statements που διαφέρουν μεταξύ τους ως προς την ενέργεια που πραγματοποιούν στην βάση, **με την σειρά που θέλουμε να εκτελεστούν**, όπως μίξη INSERT INTO με UPDATE...SET....WHERE.

Παράδειγμα:

```
CREATE TABLE SAILORS (ID INTEGER NOT NULL PRIMARY KEY, BOAT  
INTEGER NOT NULL, NAME VARCHAR(30), CONSTRAINT BFK FOREIGN KEY  
(BOAT) REFERENCES BOATS(ID));
```

```
INSERT INTO BOATS VALUES (11, 'Renovatio');
```

```
INSERT INTO BOATS VALUES (12, 'Agios Nikolaos');
```

```
UPDATE BOATS SET NAME = 'Sorokada' WHERE ID = 1;
```

```
DELETE FROM BOATS WHERE ID = 10;
```

```
INSERT INTO BOATS VALUES (10,'Agios Panteleimonas');
```

```
INSERT INTO SAILORS VALUES (1,1,'Gerasimos');
```

```
SELECT * FROM BOATS ORDER BY ID;
```

Φωλιασμένα SELECT Queries και υπερωτήματα (subqueries). Τα **εμφωλευμένα Queries** (nested queries), δηλαδή SELECT μέσα σε ένα άλλο SELECT Clause, μπορούμε να τα εισάγουμε αυτούσια μέσα στον Worksheet Editor. Η Oracle επίσης επιτρέπει τα λεγόμενα **υπερωτήματα** (subqueries) τα οποία επιτρέπουν SELECT επερωτήματα, σε αποτελέσματα προηγούμενου

SELECT. Αυτό επιτυγχάνεται με το **WITH** clause. Παρακάτω βλέπουμε παράδειγμα:

Παράδειγμα:

--Nested Queries

SELECT NAME

FROM BOATS

WHERE ID = (**SELECT** BOAT **FROM** SAILORS **WHERE** NAME **LIKE** 'Geras%')

OR ID = (**SELECT** BOAT **FROM** SAILORS **WHERE** NAME **LIKE** 'Dim%');

--Subqueries

WITH Q **AS** (**SELECT** ID,NAME **AS** NM **FROM** BOATS **WHERE** ID **BETWEEN** 2 **AND** 4)

SELECT NM **FROM** Q **WHERE** ID = 3;

Τελεστές SQL και Συσχετίσεις. Μπορείτε να εισάγετε τελεστές SQL όπως UNION, και συσχετίσεις όπως INNER JOIN, μαζί στο ίδιο Worksheet, είτε μεμονωμένα, είτε μαζί.

Παράδειγμα:

SELECT *

FROM BOATS

LEFT JOIN SAILORS **ON** BOATS.ID = SAILORS.BOAT;

SELECT *

FROM BOATS

UNION ALL

SELECT ID,NAME

```
FROM SAILORS;
```

Queries με συνένωση στηλών και Strings (Concatenation). Στην Oracle, ο τελεστής συνένωσης στηλών είναι ο « || » τον οποίο τον βάζουμε ανάμεσα στις στήλες και στα strings που θέλουμε να συνενώσουμε στην έξοδο.

Παράδειγμα:

```
SELECT ID||'-->'||NAME AS BOATS  
FROM BOATS;
```

Σχόλια. Στην Oracle μπορείτε να εισάγετε σχόλια μίας γραμμής με τους χαρακτήρες « -- » ή πολλαπλών γραμμών, αλλά C/C++/Java στυλ με « /* */ ».

Παράδειγμα:

```
--This Query Shows boat names
```

```
SELECT NAME FROM BOATS;
```

```
/*
```

```
The next query
```

```
Shows Sailor Names
```

```
*/
```

```
SELECT NAME FROM SAILORS;
```

Όπως είδαμε στο Worksheet έχουμε μεγάλη ελευθερία ως προς το είδος και τον αριθμό των Standard SQL Statements που μπορούμε να εισάγουμε. Φυσικά, οι γνώσεις σας και ο πειραματισμός σας σε αυτό, θα σας πει αν είναι αποδοτικό ή όχι.

7.1.1.2.1.2 – PL/SQL Commands

Εκτός από την Standard SQL, στο Worksheet μπορούμε να εισάγουμε και PL/SQL εντολές. Το είδος των εντολών αυτών θα σας τις περιγράψουμε παρακάτω.

Απλά Ανώνυμα PL/SQL Blocks. Τα Blocks χωρίς όνομα που περικλείουν μεταξύ των BEGIN και END εντολές, ονομάζονται **Ανώνυμα** (Anonymous Blocks). Μέσα σε αυτά μπορείτε να εσωκλείσετε μεμονωμένα ή πολλαπλά PL/SQL Statements, κλήση μιας ή περισσότερων Procedure / Function καθώς επίσης και δήλωση και εκχώρηση τιμών σε μεταβλητές.

Παράδειγμα:

DECLARE

number_of_records **NUMBER**;

BEGIN

SET SERVEROUTPUT ON FORMAT WRAPPED

SELECT COUNT(ID) INTO NUMBER_OF_RECORDS FROM BOATS;

DBMS_OUTPUT.PUT_LINE('RECORDS ARE '||NUMBER_OF_RECORDS);

DBMS_OUTPUT.PUT_LINE('All Done');

END;

Συνεχόμενα Ανώνυμα PL/SQL Blocks. Μπορείτε να έχετε περισσότερα του ενός ανώνυμα blocks μέσα στο Worksheet, πρέπει ωστόσο να διαχωρίζονται με τον χαρακτήρα « / » μεταξύ του END του προηγούμενου Block με το BEGIN ή DECLARE του επόμενου.

Παράδειγμα:

```

BEGIN

UPDATE SAILORS SET NAME = 'Lefteris'

WHERE ID = 2;

END;

/ -- Block Separator

BEGIN

DELETE FROM BOATS WHERE ID = 13;

END;

```

Blocks δημιουργίας και κλήσης Procedure / Function. Μπορείτε να δημιουργήσετε Procedures και Functions και παράλληλα να τις καλέσετε σε άλλο Block. Απαραίτητος ο χαρακτήρας « / ».

Παράδειγμα:

```

-- Create Procedure

CREATE OR REPLACE PROCEDURE change_boat_name(BID IN NUMBER,
NEW_NAME IN VARCHAR2 )

IS

BEGIN

UPDATE boats SET NAME = NEW_NAME WHERE ID = BID;

dbms_output.put_line('BOAT WITH ID = ||BID|| CHANGED SUCCESSFULLY');

END;

/ -- Block Separator

-- Call Procedure

BEGIN

change_boat_name (2,'Marilena');

END;

```

Δημιουργία User Defined Types. Μπορείτε επίσης να δημιουργήσετε δικούς σας τύπους όπως και στην DB2 μαζί με υποτύπους.

Παράδειγμα:

```
CREATE OR REPLACE TYPE port_t AS OBJECT
(
  ID NUMBER,
  NAME VARCHAR2(30),
  LOCATION VARCHAR2(30)
) NOT FINAL;
/ -- Block Separator
CREATE TYPE dock UNDER port_t
(CAPACITY NUMBER);
/ -- Block Separator
CREATE TABLE PORT OF DOCK;
```

Δημιουργία Εναυσμάτων (Triggers). Μπορείτε να δημιουργήσετε εναύσματα για ενέργειες που πραγματοποιούνται στους πίνακες.

Παράδειγμα:

```
CREATE OR REPLACE TRIGGER INSERT_NOTIFIER
AFTER INSERT ON BOATS
BEGIN
  DBMS_OUTPUT.PUT_LINE('A new line was inserted');
END;
```

```
/ --Block Separator
```

```
INSERT INTO BOATS VALUES (13,'Kalipsw');
```

Αυτά ήτανε τα πιο βασικά είδη εντολών PL/SQL που μπορείτε να εκτελέσετε στο Worksheet του SQL Developer. Βλέπουμε πως έχουμε μεγάλη ποικιλία εντολών που μπορούμε να εκτελέσουμε, κάτι που μας δίνει απόλυτη ευελιξία στις αλληλεπιδράσεις μας με την βάση δεδομένων.

7.1.1.2.1.3 – Scripts από αρχείο

Άλλο ένα ωραίο χαρακτηριστικό του SQL Developer είναι ότι μπορούμε να εκτελέσουμε SQL ή PL/SQL εντολές από ένα αρχείο. Κάτι τέτοιο είναι χρήσιμο όταν για παράδειγμα ο χρήστης θέλει να εκτελεί μεγάλο αριθμό εντολών σε τακτά χρονικά διαστήματα, αλλά είναι ασύμφορη η διαδικασία συγγραφής τους με το χέρι.

Υπάρχουνε δύο τρόποι με τους οποίους μπορείτε να εκτελέσετε Scripts στο SQL Developer ο οποίοι περιγράφονται παρακάτω:

- 1) Μεταβαίνοντας στο μενού File → Open και επιλέγοντας την τοποθεσία των Scripts από το File Browsing Window.
- 2) Απευθείας από το Command Worksheet, με το σύμβολο @ ακολουθούμενο από το πλήρες path του αρχείου sql που θέλετε να εκτελέσετε.

Παράδειγμα απευθείας εκτέλεσης:

```
@C:\sql_scripts\test.sql --Windows
```

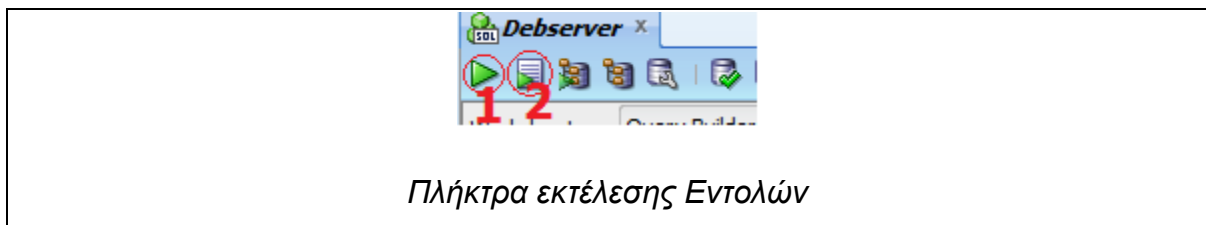
```
@/home/oracle/sql_scripts/test.sql --Linux
```

Τις εντολές που γράφετε στο Command Worksheet μπορείτε επίσης να τις αποθηκεύσετε σε ένα αρχείο Script. Για να το πράξετε, μεταβείτε στο μενού **File** →

Save και δώστε το όνομα αρχείου που επιθυμείτε καθώς και την επιθυμητή τοποθεσία στον δίσκο σας.

7.1.1.2.1.4 – Εκτέλεση Εντολών

Για να εκτελέσετε μια ή περισσότερες εντολές SQL μεταβείτε στο χώρο πάνω από τον Command Worksheet και θα βρείτε δύο κουμπιά όπως παρακάτω:



Το πλήκτρο «Run Statement» (1) εκτελεί την τρέχουσα επιλεγμένη γραμμή. Το πλήκτρο «Run Script» (2) τρέχει όλες τις εντολές που υπάρχουν στο Command Worksheet **με την σειρά**.

7.1.1.2.2 – Query Builder

Η αποστολή **επερωτημάτων** (Queries) στην βάση είναι χωρίς αμφιβολία ίσως από τις συχνότερες λειτουργίες που επιτελούνται σε μια βάση δεδομένων. Ο χρήστης αναζητά συγκεκριμένες πληροφορίες από αυτήν, οπότε στέλνει το αντίστοιχο επερώτημα, δίνοντας ένα κατάλληλα διαμορφωμένο SELECT...FROM...WHARE Statement. Τις περισσότερες φορές αυτό είναι επαρκές, αλλά τι γίνεται όταν ο χρήστης αναζητάει πληροφορίες από πίνακες με πολλές εγγραφές; Τι γίνεται σε περίπτωση που ο χρήστης αναζητάει πληροφορίες που βρίσκονται σε περισσότερους από έναν πίνακες; Και στην χειρότερη περίπτωση, τι γίνεται αν ο χρήστης δεν έχει μεγάλη εμπειρία στην SQL;

Βάση αυτών των αναγκών, δημιουργήθηκαν οι εφαρμογές **Δημιουργίας Επερωτημάτων** (Query Builder) οι οποίες παρέχουν ένα σύνολο από **οπτικά**

(visual) εργαλεία, τα οποία ο χρήστης χρησιμοποιεί για να χτίσει το επερώτημα που θέλει να στείλει στην βάση. Μεγάλο ατού ενός τέτοιου προγράμματος είναι ότι αφ ενός μεν επιτρέπει την δημιουργία σύνθετων και πολύπλοκων SELECT επερωτημάτων, αφ εταίρου δε, δεν απαιτεί πολλές (ή και καθόλου) γνώσεις SQL.

Η φιλοσοφία ενός Query Builder είναι η εξής. Ο χρήστης απλά επιλέγει τους πίνακες που περιέχουν τις πληροφορίες που τον ενδιαφέρουν. Οι στήλες του κάθε πίνακα εμφανίζονται στο Query Builder και εκεί ο χρήστης αρχικά πρέπει να επιλέξει ποια από όλες θα συμμετέχει στο επερώτημα. Στην συνέχεια θα πρέπει να εισάγει **κριτήρια**, δηλαδή συγκεκριμένες οδηγίες φιλτραρίσματος πληροφοριών, στα οποία πρέπει να συμφωνούν οι τελικές πληροφορίες που θα επιστραφούν. Όπως είπαμε και πριν, τα εργαλεία που παρέχει το Query Builder είναι **οπτικά**, οπότε ο χρήστης δεν χρειάζεται να γράψει ούτε γραμμή SQL Κώδικα, αλλά θα την παράγει το Query Builder γι αυτόν.

Ως εκ τούτου, μια σουίτα εργαλείων σαν το SQL Developer, δεν θα μπορούσε να μην έχει Query Builder. Ο εν λόγω Query Builder είναι πολύ εύχρηστος και φιλικός στον χρήστη, χωρίς όμως να υστερεί σε λειτουργικότητα. Θα το διαπιστώσουμε και μαζί αυτό.

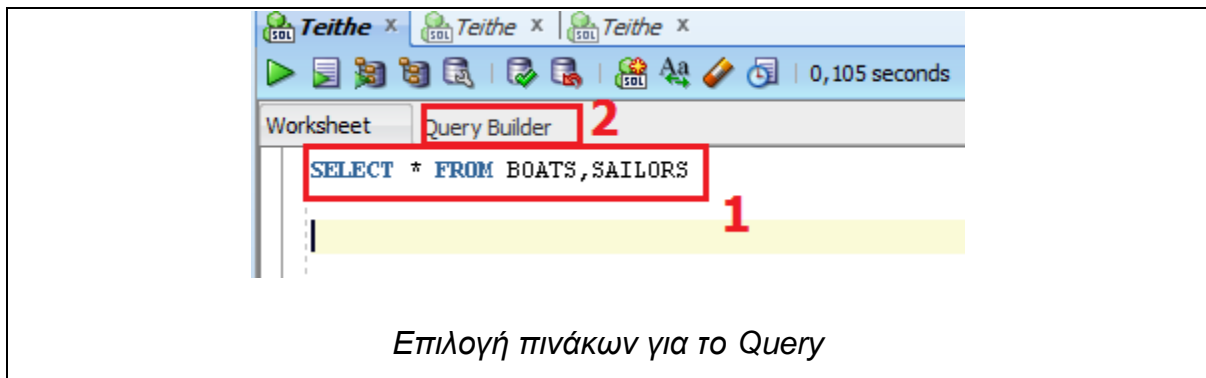
7.1.1.2.2.1 – Δημιουργία Query

Η δημιουργία ενός επερωτήματος, εκτελείται ακολουθώντας τα παρακάτω συνοπτικά βήματα:

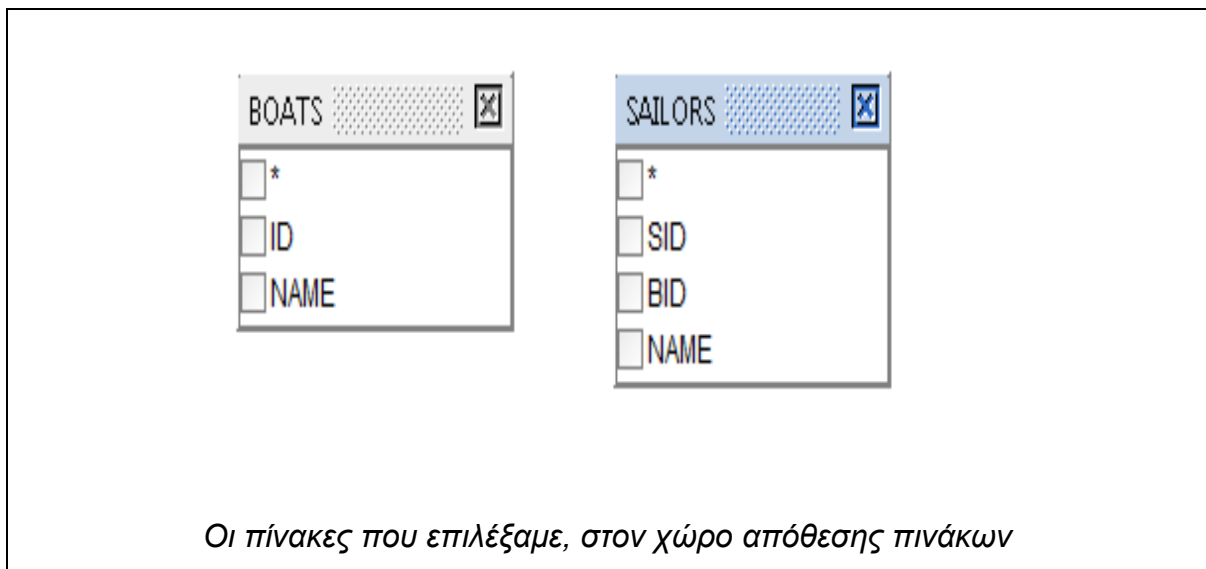
- 1) Επιλογή του πίνακα ή των πινάκων που περιέχουν τις πληροφορίες που αναζητούμε
- 2) Επιλογή των στηλών των πινάκων που περιέχουν ή επηρεάζουν το τελικό αποτέλεσμα

- 3) Εισαγωγή των κατάλληλων κριτηρίων στα οποία πρέπει η αναζήτηση να συμφωνεί για να εμφανιστούν οι ζητούμενες πληροφορίες

Για το **βήμα 1** πρέπει να επιλέξουμε τους πίνακες μας. Αυτό το κάνουμε γράφοντας στον Command Editor «SELECT * FROM table1,table2» (Καρτεσιανό γινόμενο) (1). Τότε μεταβαίνουμε στο Query Builder (2) όπως παρακάτω:

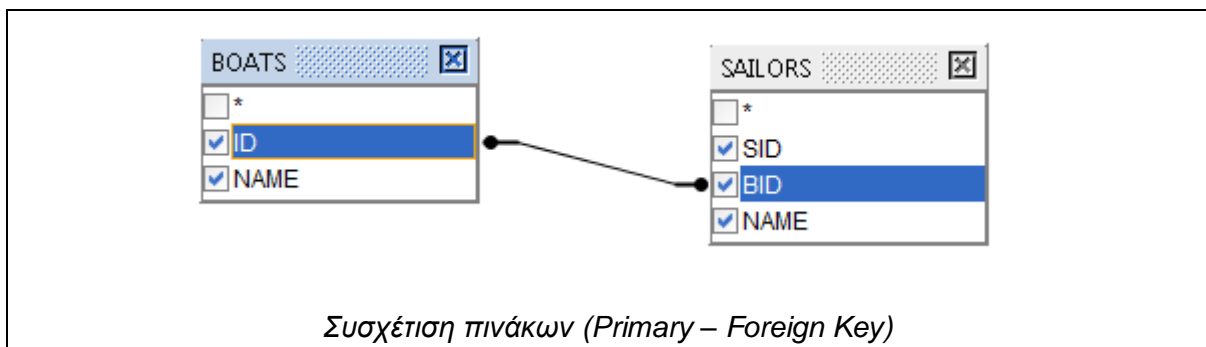


Αφού μεταβούμε στο Query Builder βλέπουμε πως οι πίνακες που επιλέξαμε στο προηγούμενο SELECT εμφανίζονται στον **χώρο απόθεσης πινάκων** όπως βλέπουμε παρακάτω:



Τα τετραγωνάκια παριστάνουν τους πίνακές μας, και τα checkboxes παριστούν τις στήλες τους. Επιλέγοντας ένα checkbox επιλέγουμε μια στήλη αν θα συμμετέχει ή όχι στο Query (**βήμα 2**).

Τους πίνακές μας συνήθως απαιτείται να τους **συσχετίσουμε** μεταξύ τους (το κύριο κλειδί του ενός, με το ξένο κλειδί του άλλου). Για να το κάνετε αυτό πρέπει να επιλέξετε από τον έναν πίνακα και κάνοντας Drag-n-drop αφήνετε τον δείκτη πάνω από την στήλη του πίνακα που θέλετε να συσχετίσετε. Όταν το κάνετε το αποτέλεσμα θα πρέπει να μοιάζει με το κάτωθι:



Τώρα μπορούμε να εισάγουμε τα κριτήριά μας (βήμα 3). Αυτά το εισάγουμε στο GRID εισαγωγής κριτηρίων που βλέπουμε κάτωθι:

Output	Expression	Aggregate	Alias	Sort Type	Sort Order	Grouping	Criteria	Or
<input checked="" type="checkbox"/>	BOATS.ID		B_ID			<input type="checkbox"/>		
<input checked="" type="checkbox"/>	BOATS.NAME		BNAME			<input type="checkbox"/>		
<input checked="" type="checkbox"/>	SAILORS.SID					<input type="checkbox"/>		
<input checked="" type="checkbox"/>	SAILORS.BID					<input type="checkbox"/>		
<input checked="" type="checkbox"/>	SAILORS.NAME		SNAME			<input type="checkbox"/>		

Οι στήλες των πινάκων στο Panel Κριτηρίων

Τώρα μας ενδιαφέρει το Panel κριτηρίων, δηλαδή ο χώρος που παρατίθενται οι στήλες των πινάκων και μπορούμε να εισάγουμε τα κριτήρια αναζήτησης πληροφοριών. Αποτελείται από έναν σεβαστό αριθμό στηλών τον οποίο και εξηγούμε:

Η στήλη «Output» (1) προσδιορίζει αν η επιλεγμένη στήλη θα συμμετέχει στο SELECT κοινώς δηλαδή αν θα εμφανιστεί στην έξοδο. (Αντιστοιχεί στο **SELECT column FROM table**).

Η στήλη «Aggregate» (2) προσδιορίζει την Aggregate Function που θα επιλεγεί σε ένα SELECT Statement όπως Count, AVG, SUM κλπ (αντιστοιχεί στο **SELECT Count(column) FROM table**).

Η στήλη «Alias» (3) επιτρέπει την έξοδο μιας στήλης με άλλο όνομα από αυτό που έχει. (Αντιστοιχεί στο **SELECT column AS NewName FROM table**).

Η στήλη «Sort Type» (4) προσδιορίζει τον τρόπο ταξινόμησης των αποτελεσμάτων που επιστρέφει ένα SELECT και βάση ποιας στήλης θα γίνει η ταξινόμηση αυτή (Αντιστοιχεί στο **SELECT col1,col2 FROM TABLE ORDER BY col1 ASC**).

Η στήλη «Grouping» (5) προσδιορίζει αν η στήλη του πίνακα θα συμμετέχει σε ένα Group By Clause (αντιστοιχεί σε **SELECT col1,col2 FROM table GROUP BY col1 ...**).

Η στήλη «Criteria» (6) είναι η πιο βασική στήλη μας, καθώς σε αυτήν εισάγουμε τα κριτήρια αναζήτησης πληροφορίας. Θα επεξηγηθεί περεταίρω παρακάτω (Αντιστοιχεί σε **SELECT col1,col2 FROM table WHERE <CONDITION>**).

Τέλος, η στήλη «OR» (7) επιτρέπει την εισαγωγή εναλλακτικού κριτηρίου OR για την ίδια στήλη (Αντιστοιχεί σε **SELECT col1,col2 FROM table WHERE <condition 1> OR <condition 2>**).

7.1.1.2.2.2 – Εισαγωγή κριτηρίων

Όπως αναφέρθηκε και παραπάνω, η στήλη Criteria είναι η πιο σημαντική στήλη του Query Builder. Σε αυτήν θα εισάγουμε τα κριτήρια μας, ώστε το Query να μας επιστρέψει τις γραμμές που θέλουμε. Για άλλη μια φορά αναφέρουμε πως η στήλη Criteria αντικατοπτρίζει το **WHERE Clause**.

Στην στήλη Criteria μπορείτε να εισάγετε τα παρακάτω:

- 1) Σύμβολα **Ισότητας, Ανισότητας** η **συνδυασμό** αυτών, δηλαδή =, >, <, >=, <=, <> για αριθμητικές τιμές (πχ ID= 2)
- 2) Τελεστές **LIKE** για αναζήτηση pattern αλφαριθμητικών τιμών συνδυαζόμενοι με τους wildcard χαρακτήρες _ και % (πχ NAME LIKE 'D%')
- 3) Τελεστές **BETWEEN** για ορισμό εύρους τιμών. (πχ ID BETWEEN 2 AND 4)
- 4) Τελεστές **AND, OR, NOT**. Οι τελεστές AND χρειάζεται να γραφτούν αν χρησιμοποιηθεί BETWEEN, ωστόσο αν εισαχθούν κριτήρια και σε άλλες στήλες, δημιουργείται αυτόματα. Ο τελεστής OR μπορεί να παραληφθεί επειδή υπάρχει ήδη στήλη στο Criteria Grid με το ίδιο όνομα όπου μπορείτε να χρησιμοποιήσετε αυτήν. Τέλος ο Τελεστής NOT χρησιμοποιείται για αντιστροφή του προσδοκώμενου αποτελέσματος από κάποιον τελεστή όπως LIKE, BETWEEN

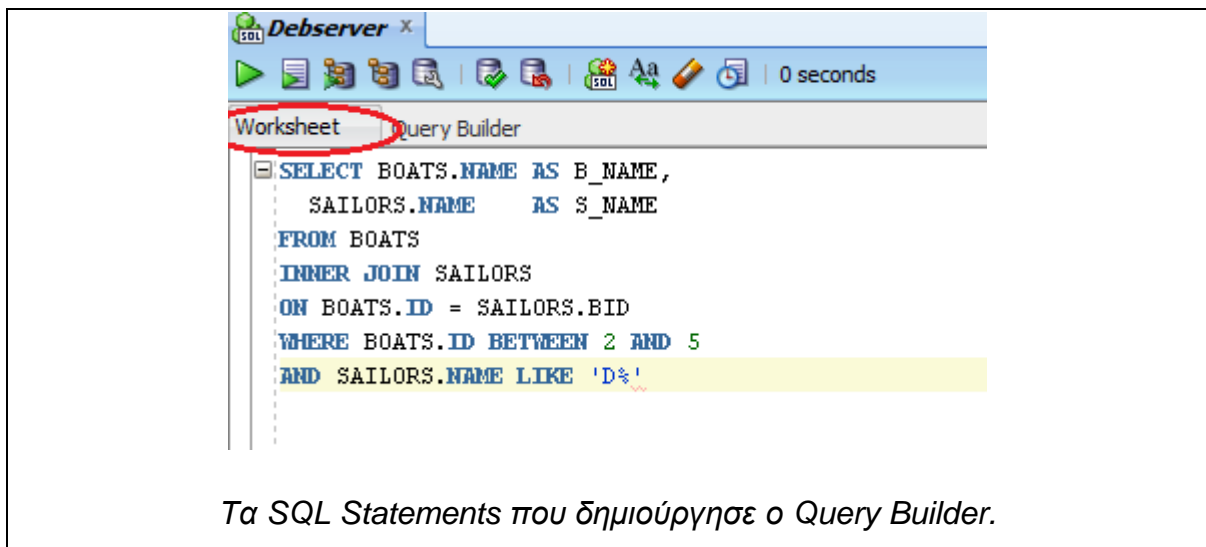
7.1.1.2.2.3 – Προβολή και Εκτέλεση Query

Ας πάρουμε μαζί ένα παράδειγμα. Θέλουμε να εμφανίσουμε από τους πίνακες BOATS και SAILORS **τα ονόματα των ναυτικών και των βαρκών τους**, όπου οι **τιμές των αναγνωριστικών των βαρκών τους** να κυμαίνονται **μεταξύ 2 και 5** και το όνομα του ναυτικού **να ξεκινάει από το γράμμα «D»** Επίσης οι στήλες των ονομάτων, να έχουν λεζάντα B_NAME και S_NAME αντίστοιχα.

Ας το πάμε βήμα-βήμα. Μας λέει ότι θέλουμε να εμφανίσουμε τα ονόματα των ναυτικών και των βαρκών τους, οπότε (αφού συσχετίσουμε τον πίνακα BOATS και τον πίνακα SAILORS) οπότε τσεκάρουμε το Output των στηλών

BOATS.NAME και SAILORS.NAME. Κατόπιν μας ζητάει οι τιμές των αναγνωριστικών των βαρκών τους να κυμαίνονται μεταξύ 2 και 5. Αυτό είναι **κριτήριο**. Οπότε στην στήλη Criteria, στο πεδίο που αντιστοιχεί στην στήλη BOATS.ID εισάγουμε «BETWEEN 2 AND 5». Επόμενο ζητούμενο είναι το όνομα του ναυτικού να ξεκινάει από το γράμμα «D». Κι αυτό είναι ένα κριτήριο. Οπότε στην στήλη Criteria, στο πεδίο που αντιστοιχεί στην στήλη SAILORS.NAME εισάγουμε «LIKE 'D%'». Τέλος, για να μετονομάσουμε τις στήλες σε B_NAME και S_NAME, τα εισάγουμε σαν τιμές στο πεδίο της στήλης Alias, που αντιστοιχεί στην εκάστοτε στήλη Name του πίνακα.

Αφού δημιουργήσουμε το Query δεν μένει παρά να το εκτελέσουμε. Μπορούμε να το εκτελέσουμε απευθείας πατώντας στο πλήκτρο «Run Script» ή με F5, ωστόσο όμως μπορούμε να το δούμε πριν το εκτελέσουμε απλά μεταβαίνοντας στην καρτέλα «Worksheet» όπου θα μας μεταφέρει ξανά στο Command Worksheet όπως βλέπουμε παρακάτω:



Παραπάνω βλέπουμε το SQL Query που δημιούργησε για μας το Query Builder. Το ερωτήμα βλέπουμε πως χτίστηκε όπως το θέλαμε, ωστόσο όμως θα πρέπει να το εκτελέσουμε κιάλας για να δούμε αν πράγματι ανταποκρίνεται στις προσδοκίες μας (πατάμε F5 ή **Run Script**). Η έξοδος είναι η παρακάτω:

B_NAME	S_NAME
Eliza	Dimitris

Έξοδος του ερωτήματος

Είδαμε λοιπόν πως το Query Builder του SQL Developer είναι πολύ ισχυρό εργαλείο και πανεύκολο στην χρήση. Χρησιμοποιήστε το άφοβα αν κύριος παράγοντας για σας είναι η ευκολία χρήσης και ο χρόνος. Θα δείτε και μόνοι σας πως αξίζει τον κόπο να ασχοληθεί κανείς μαζί του.

7.1.1.2.3 – Συμβουλές για το SQL Developer

Σε αυτήν την ενότητα θα σας δοθούν ορισμένες συμβουλές οι οποίες θα σας βοηθήσουν να χρησιμοποιείτε αποδοτικά την εφαρμογή και να κάνει την εργασία σας στην βάση δεδομένων ποιο αποδοτική.

- 1) Τις εντολές SQL και PL/SQL να τις γράφετε με κεφαλαία γράμματα και να τις τερματίζετε με Semicolon (;). Είναι **καλή πρακτική** σύνταξης.
- 2) Τις σύνθετες εντολές PL/SQL και τα Blocks να τα διαχωρίζετε με το σύμβολο « / », διαφορετικά κινδυνεύετε να αντιμετωπίσετε συντακτικό σφάλμα.
- 3) Τις σύνθετες, πολλαπλές ή/και συχνά χρησιμοποιούμενες εντολές, καλή πρακτική είναι να τις αποθηκεύετε σε Scripts αρχεία για δική σας διευκόλυνση.
- 4) Όταν δημιουργείτε συνδέσεις, ειδικά αν δουλεύετε σε περισσότερες από μία βάσεις, φροντίστε να δίνεται ονόματα που να αντικατοπτρίζουν την συγκεκριμένη βάση, για δική σας διευκόλυνση.

- 5) Τα SID είναι πολύ κρίσιμος παράγοντας σύνδεσης στην εκάστοτε βάση δεδομένων, οπότε φροντίστε να είναι σωστά.
- 6) Χρησιμοποιείτε το Query Builder για να γλυτώσετε χρόνο από την χειροκίνητη συγγραφή πολύπλοκων SELECT Statements.

Προσοχή!

Μην ξεχνάτε ποτέ, ύστερα από εντολές αλλαγής της δομής της βάσης (DROP TABLE, INSERT INTO, UPDATE κλπ) να κάνετε **COMMIT** στην βάση, για να αποθηκεύονται οι αλλαγές. Για να το κάνετε αυτό απλά γράφετε την εντολή COMMIT ή πατάτε F11. Σε άλλη περίπτωση, κινδυνεύετε να μην αποθηκευτούν οι αλλαγές σας στην βάση δεδομένων, καθώς οι βάσεις της Oracle είναι Transaction Based.

7.1.2 – Το Oracle Application Express (APEX)

Στην πλειονότητα των περιπτώσεων, η ευκολία και η ευελιξία παίζουν τεράστιο ρόλο σε οτιδήποτε καλούμαστε να πραγματοποιήσουμε. Ο άνθρωπος τείνει να πραγματοποιεί ενέργειες με όσο το δυνατόν πιο εύκολο αλλά συνάμα αποδοτικό τρόπο. Ίσως να μην είναι πάντα εφικτό, αλλά το λεγόμενο «The hard way is the best way» δεν είναι πάντα ο ποιο αποδοτικός τρόπος.

Έτσι είναι και οι χρήστες ενός συστήματος. Ποιος θέλει ένα δύσχρηστο, πολύπλοκο και μικρής παραγωγικότητας σύστημα; Δεν νομίζω (προσωπικά) να υπάρχει κανείς.

Το ίδιο ισχύει και για τους χρήστες σε μια βάση δεδομένων. Ο χρήστης θέλει να αποφύγει την πολυπλοκότητα ενός συστήματος και να κάνει την επιθυμητή του δουλειά, όσο πιο ανώδυνα αλλά και όσο πιο παραγωγικά γίνεται. Και φυσικά, στα πλαίσια του εφικτού πάντα, σε σύντομο χρονικό διάστημα. Το τελικό ζητούμενο είναι ένα, ο χρήστης θέλει να συνδεθεί στην βάση, να αλληλεπιδράσει μαζί της, να πάρει τα αποτελέσματα που θέλει και να γίνει

αποδοτικά η δουλειά του. Ειδικά αν πρόκειται για σπουδαστή, που θέλει να πραγματοποιήσει τις ασκήσεις του πάνω στην βάση.

Όταν λοιπόν η διευκόλυνση του χρήστη είναι το ζητούμενο, τότε η Oracle είναι πρωτοπόρος στην ικανοποίηση αυτού του ζητούμενου. Ο χρήστης μπορεί να συνδεθεί και να δουλέψει στην βάση, χωρίς να έχει απολύτως καμία εφαρμογή Client εγκατεστημένη στον υπολογιστή του. Το κάνει απλά, μέσα από τον περιηγητή ιστού (web browser) της επιλογής του, απλά, γρήγορα και συνάμα αποδοτικά, σε ένα Web Based σύστημα. Σε ένα εύχρηστο και φιλικό στο χρήστη σύστημα. Μιλάμε λοιπόν για το σύστημα **Oracle Application Express (APEX)**.

Το **Oracle Application Express** ή **Oracle APEX** για συντομία, είναι ένα ολοκληρωμένο **web-based** περιβάλλον εργασίας και ανάπτυξης σε μια βάση δεδομένων της Oracle. Μέσω του APEX ο χρήστης μπορεί να εργαστεί στην βάση δεδομένων δημιουργώντας πίνακες, types, procedures/functions και views, με τον ίδιο τρόπο όπως σε μία οποιαδήποτε άλλη εφαρμογή όπως το SQL Developer. Το γεγονός το ότι είναι web-based, εισάγει ευχρηστία στην εργασία ενός χρήστη με την βάση δεδομένων, καθώς το μόνο που χρειάζεται από πλευράς του, είναι ένας web browser, όπως προαναφέρθηκε.

Το APEX δεν έχει να ζηλέψει σχεδόν σε τίποτα από μια «παραδοσιακή» client εφαρμογή εργασίας σε ένα DBMS, όπως το SQL Developer, το PG-Admin ή το MySQL Workbench. Μπορεί με την ίδια ευκολία να δημιουργήσει/τροποποιήσει πίνακες, functions και procedures ή οτιδήποτε άλλο επιθυμεί να πραγματοποιήσει πάνω στην βάση δεδομένων. Προσφέρει μια σειρά από πολύ χρήσιμα εργαλεία, τα οποία «λύνουν τα χέρια» του χρήστη. Ας τα δούμε:

- 1) **SQL Commands:** Το βασικότερο εργαλείο, μέσα στο οποίο μπορεί ο χρήστης να δίνει SQL και PL/SQL Statements και φυσικά να τα εκτελεί. Διαθέτει χώρο εισαγωγής των SQL Statements, πληροφοριακή οθόνη κατάστασης (έκβαση εκτέλεσης, αναγγελία σφάλματος) καθώς επίσης και εύχρηστο μενού άμεσης αλλαγής schema (όπου αυτό επιτρέπεται).
- 2) **Object Browser:** Εφαρμογή προβολής των υπάρχοντων πινάκων, όψεων, procedures, triggers κλπ με αναλυτικές πληροφορίες για αυτά. Επίσης

επιτρέπει την δημιουργία όλων των παραπάνω μέσω οπτικής (visual) καθοδήγησης.

- 3) **SQL Scripts:** Μέσω αυτού του εργαλείου, ο χρήστης μπορεί να ανεβάζει στον server αρχεία με SQL Scripts και να τα εκτελέσει. Διαθέτει κομψό και εύχρηστο μενού όπου ταξινομούνται όλα τα scripts με δυνατότητα ονοματοδοσίας αυτών για αναγνώριση του σκοπού που επιτελούν
- 4) **Query Builder:** Εύχρηστο εργαλείο αυτοματοποιημένης δημιουργίας Queries με οπτική προβολή των πινάκων και των στηλών τους. Ο χρήστης δίνει τα κριτήρια που επιθυμεί, και αυτόματα παράγεται το SQL Select Statement γι αυτόν. Χρήσιμο εργαλείο που γλυτώνει και χρόνο.

Βλέπουμε λοιπόν ότι το APEX είναι πληρέστατο. Δεν μένει λοιπόν να το δούμε στην πράξη.

7.1.2.1 – Σύνδεση στο Oracle APEX

Για να συνδεθείτε στο Oracle APEX, πρέπει αρχικά να ανοίξετε τον Web Browser της επιλογής σας (προτείνονται οι Windows Internet Explorer που έχει άριστη συμβατότητα με το APEX ή ο Mozilla Firefox) και να μεταβείτε στην διεύθυνση του Server όπου τρέχει το APEX και η βάση δεδομένων. Η διεύθυνση είναι της μορφής:

http://server_address:apex_port/apex

Παράδειγμα: http://Debserver:8080/apex

Αφού εισάγετε την διεύθυνση και μεταβείτε σε αυτήν θα σας ζητήσει τα **διαπιστευτήρια** σας, όπως βλέπουμε στην παρακάτω εικόνα:

ORACLE Application Express

Login

Enter Application Express workspace and credentials.

1 Workspace students

2 Username dmlab_1

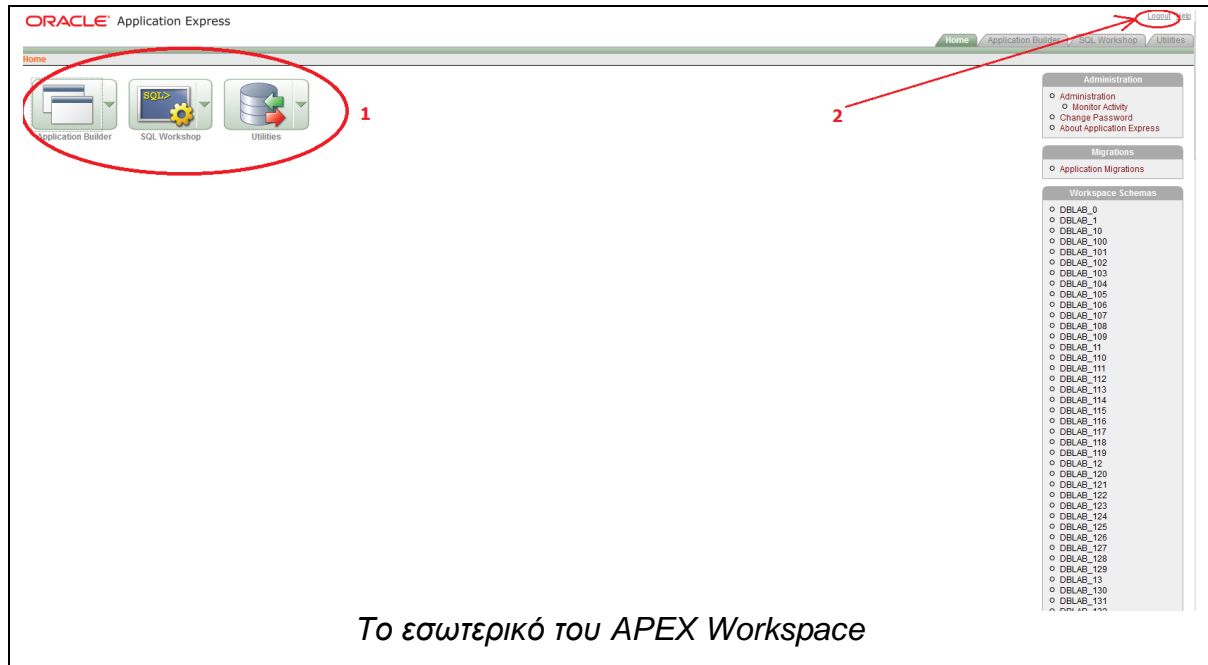
3 Password ●●●●●●●●

4 Login

[Click here to learn how to get started](#)

Φόρμα Login του APEX

Στο πεδίο «Workspace» (1) δίνετε το όνομα του Workspace που σας έχει ανατεθεί από τους διαχειριστές. Το Workspace θα είναι ο ενιαίος χώρος εργασίας σας. Στην συνέχεια δίνετε το Username σας (2) και τον κωδικό πρόσβασής σας (3) και πατάτε «Login» (4). Σε ελάχιστο χρόνο, θα μεταβείτε στο εσωτερικό του Workspace σας, όπου θα μπορείτε να ξεκινήσετε αν εργάζεστε με το APEX. Το εσωτερικό του Workspace το βλέπουμε στην παρακάτω εικόνα:



Στην κεντρική οθόνη του Workspace του APEX διακρίνουμε τις **σουίτες εργαλείων** που προσφέρονται (1), διάφορες άλλες επιλογές και το πλήκτρο **Αποσύνδεσης** (logout) από το APEX (2).

Keep in mind...

Οι χρήστες-πελάτες του Oracle APEX συνήθως έχουν τον ρόλο του Developer, οπότε μπορούν να εργαστούν στο Workspace, χωρίς όμως να έχουν δικαιώματα διαχείρισης αυτού.

Συνήθως ο όνομα χρήστη είναι συσχετιζόμενο με το Schema του στην βάση δεδομένων. Αν ως χρήστες είστε συσχετιζόμενοι σε κοινόχρηστο Schema, τότε έχετε όλοι τα ίδια δικαιώματα πάνω σε αυτό όσον αφορά την δημιουργία / τροποποίηση αντικειμένων, οπότε προσοχή στις ενέργειές σας γιατί μπορεί να έχουν αντίκτυπο στα αντικείμενα που αφορούν και άλλους χρήστες.

7.1.2.2 – Εργασία στο Oracle APEX με το SQL Workshop

Σε αυτήν την ενότητα θα επεκταθούμε στα κυρίως χρησιμοποιούμενα εργαλεία που προσφέρει το Oracle APEX, και συγκεκριμένα η σουίτα εργαλείων **SQL Workshop**, για τον χρήστη. Το SQL Workshop είναι μια σουίτα εργαλείων,

ένας «ελβετικός σουγιάς» δηλαδή, με ισχυρά εργαλεία τα οποία ο χρήστης έχει στην διάθεσή του προκειμένου να εργαστεί στην βάση δεδομένων.

Θα δούμε μαζί την λειτουργικότητά τους και θα είμαστε σε θέση να συμπεράνουμε για τον βαθμό ευκολίας και παραγωγικότητας που παρέχουν. Ας ξεκινήσουμε.

7.1.2.2.1 – Το εργαλείο SQL Commands

Αναντίλεκτα, το πιο ισχυρό εργαλείο του Oracle APEX. Πρόκειται για ένα ολοκληρωμένο περιβάλλον συγγραφής SQL κώδικα, το οποίο δεν έχει να ζηλέψει πολλά από μια αντίστοιχη εφαρμογή SQL Client, όπως το SQL Developer που είδαμε πριν.

Η εφαρμογή SQL Commands επιτρέπει την συγγραφή και εκτέλεση SQL – PL/SQL Statements στην βάση δεδομένων. Μέσα από αυτό ο χρήστης μπορεί να δημιουργεί/τροποποιεί πίνακες, όψεις, procedures/functions **προγραμματιστικά**.

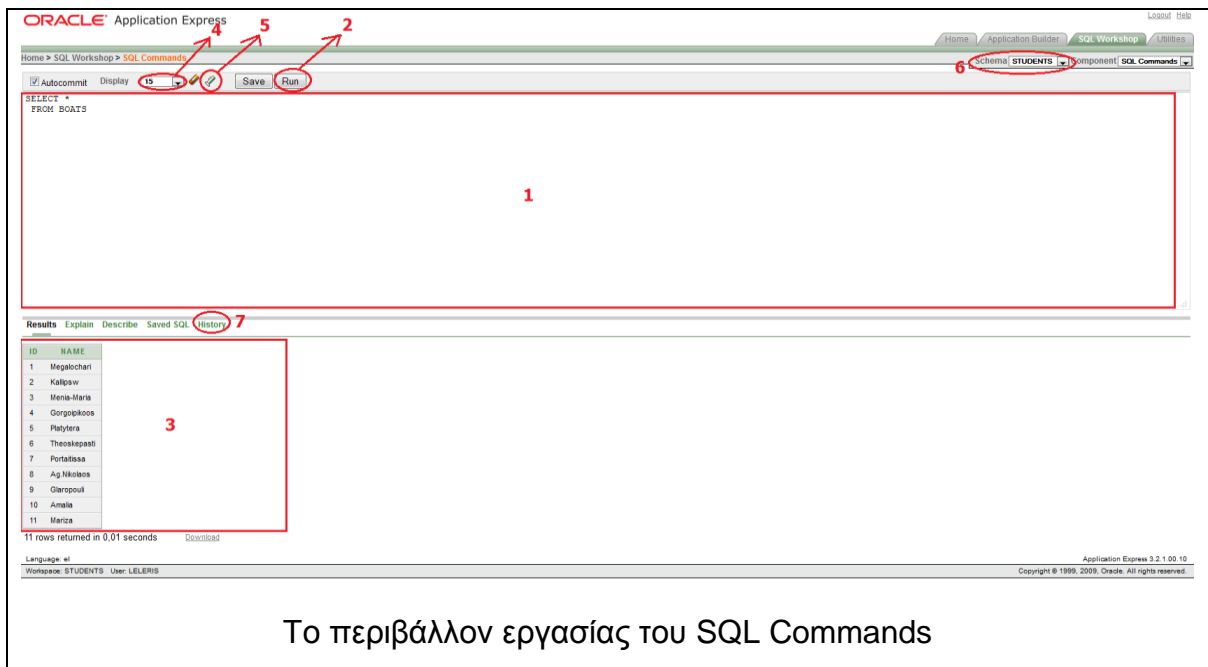
Keep in mind...

Το SQL Commands, δεν διαθέτει **τονισμό σύνταξης** (syntax highlighting), δηλαδή οι δεσμευμένες SQL λέξεις, δεν τονίζονται. Αν οπωσδήποτε χρειάζεστε τονισμό σύνταξης, τότε γράφετε τις εντολές σας σε ένα Text Editor που υποστηρίζει τονισμό σύνταξης και μεταφέρετε τις εντολές με Copy/Cut – Paste στο SQL Commands.

Η πρόσβαση στο SQL Commands γίνεται μέσα από την σουίτα SQL Workshop πατώντας κλικ στο παρακάτω εικονίδιο:



Το περιβάλλον εργασίας του SQL Commands μπορούμε να το διακρίνουμε στην παρακάτω εικόνα:



Το περιβάλλον εργασίας του SQL Commands, αποτελείται από τα παρακάτω βασικά στοιχεία, τα οποία περιγράφουμε (οι αριθμήσεις αντιστοιχούν στα αριθμημένα τμήματα):

- 1) **Περιοχή συγγραφής εντολών SQL – PL/SQL.** Σε αυτήν την περιοχή ο χρήστης γράφει την εντολή ή μπλοκ εντολών SQL-PL/SQL που επιθυμεί.
- 2) **Κουμπί Εκτέλεσης των εντολών SQL.** Το πλήκτρο αυτό εκτελεί τις εντολές SQL που έδωσε ο χρήστης

- 3) **Περιοχή αποτελεσμάτων εκτέλεσης εντολών SQL.** Σε αυτήν την περιοχή εμφανίζεται η έκβαση της εκτέλεσης των εντολών SQL, δηλαδή αν η εκτέλεση πέτυχε ή απέτυχε και για πιο λόγο. Αν πρόκειται για εντολή SELECT, τότε σε αυτήν την περιοχή εμφανίζονται οι ζητούμενες στήλες του ζητούμενου πίνακα.

- 4) **Πεδίο επιλογής αριθμού γραμμών πίνακα προς εμφάνιση (για SELECT Statements).** Επιτρέπει στο χρήστη να επιλέξει πόσες γραμμές του πίνακα θα εμφανιστούν μετά την εκτέλεση ενός SELECT Statement. Αν ο πίνακας έχει πολλές γραμμές, ο χρήστης μπορεί να επιλέξει να εμφανιστούν πχ οι 15 πρώτες.

- 5) **Εργαλείο αναζήτησης πινάκων.** Χρήσιμο εργαλείο στο οποίο ο χρήστης μπορεί να δει τους πίνακες που υπάρχουν στο Schema του, καθώς και τον τύπο δεδομένων των στηλών τους, τους περιορισμούς τους και άλλες πληροφορίες. Χρήσιμο πριν την εκτέλεση SELECT ή INSERT INTO για αποφυγή συντακτικών λαθών.

- 6) **Επιλογέας Schema.** Επιτρέπει τον χρήστη να αλλάξει Schema άμεσα, αν φυσικά έχει τα απαραίτητα δικαιώματα εργασίας σε άλλο Schema.

- 7) **Ιστορικό Εντολών SQL.** Περιέχει εντολές που εκτελέστηκαν σε παρελθοντικό χρόνο. Χρήσιμο αν κάποιες εντολές χρησιμοποιούνται συχνά.

7.1.2.2.1.1 – Ορθή Σύνταξη Εντολών

Η σύνταξη των εντολών στο SQL Commands, έχει μια μικρή διαφοροποίηση από ότι έχουμε δει μέχρι τώρα. Αναλόγως με την ζητούμενη ενέργεια που θέλει να πραγματοποιήσει ο χρήστης, ο τελευταίος μπορεί να εισάγει τα παρακάτω:

Προσοχή!

Στα παρακάτω παραδείγματα, δεν θα χρησιμοποιήσουμε τονισμό σύνταξης, σε μια προσπάθεια να προσομοιώσουμε συνθήκες εργασίας στο SQL Commands.

1) Μία εντολή:

Αν ο χρήστης θέλει να εκτελέσει μία εντολή, όπως SELECT, INSERT INTO, UPDATE κλπ, μπορεί να την γράψει αυτούσια όπως είναι. Το semicolon αν και δεν είναι υποχρεωτικό, καλό θα είναι να τοποθετείται στο τέλος της εντολής. Καλή πρακτική επίσης είναι η χρήση κεφαλαίων στις δεσμευμένες SQL λέξεις.

Παράδειγμα:

```
SELECT * FROM BOATS WHERE ID = 2;
```

2) Πολλαπλές Εντολές

Αν ο χρήστης επιθυμεί να εκτελέσει πολλαπλές εντολές SQL όπως για παράδειγμα πολλαπλά INSERT INTO θα πρέπει να τοποθετεί τις εντολές του μεταξύ των BEGIN...END; λέξεων. Σε διαφορετική περίπτωση τα statements δεν θα εκτελεστούν, παρόλο που η σύνταξή τους είναι σωστή. Έτσι οι εντολές εκτελούνται όλες μαζί, σε μπλοκ που, τα γνωστά δηλαδή ανώνυμα PL/SQL Blocks.

Συνεχόμενα Blocks δεν υποστηρίζονται. Πρακτικά αυτό σημαίνει ότι κάθε block κώδικα, θα το γράφετε μεμονωμένα.

Παράδειγμα:

--Multiple Inserts

BEGIN

INSERT INTO BOATS (ID,NAME) VALUES (4,'Gorgoipikoos');

INSERT INTO BOATS (ID,NAME) VALUES (5,'Platytera');

INSERT INTO BOATS (ID,NAME) VALUES (6,'Theoskepasti');

INSERT INTO BOATS (ID,NAME) VALUES (7,'Portaitissa');

INSERT INTO BOATS (ID,NAME) VALUES (8,'Ag.Nikolaos');

INSERT INTO BOATS (ID,NAME) VALUES (9,'Glaropouli');

INSERT INTO BOATS (ID,NAME) VALUES (10,'Amalia');

INSERT INTO BOATS (ID,NAME) VALUES (11,'Mariza');

END;

3) Procedures/Functions (PL/SQL)

Η σύνταξη εντολών δημιουργίας procedure ή function ακολουθεί την standard σύνταξη PL/SQL, δηλαδή η δημιουργία και η κλήση της γίνεται μέσα στα BEGIN και END; statements. Παρακάτω δίδεται ένα ενδεικτικό παράδειγμα δημιουργίας και κλήσης μιας Procedure.

Παράδειγμα:

```
--Create Procedure  
  
CREATE OR REPLACE PROCEDURE msgOut  
  
IS  
  
BEGIN  
  
dbms_output.put_line('Hello from Oracle');  
  
END;  
  
--Call Procedure (μεμονωμένα, χωρίς τις παραπάνω εντολές)  
  
BEGIN  
  
msgOut;  
  
END;
```

4) User Defined Types (PL/SQL)

Στο APEX μπορείτε να δημιουργήσετε και τύπους/αντικείμενα και βάση αυτών να δημιουργήσετε πίνακες. Ενδεικτικό παράδειγμα δίνεται κάτωθι:

Παράδειγμα:

```
--Create type  
  
CREATE TYPE port_t AS OBJECT  
  
(  
  
pid NUMBER,  
  
name VARCHAR2(30),  
  
location VARCHAR2(30)
```

```
);  
  
--Create table of this type (μεμονωμένα, χωρίς τις παραπάνω εντολές)  
  
CREATE TABLE port OF port_t;
```

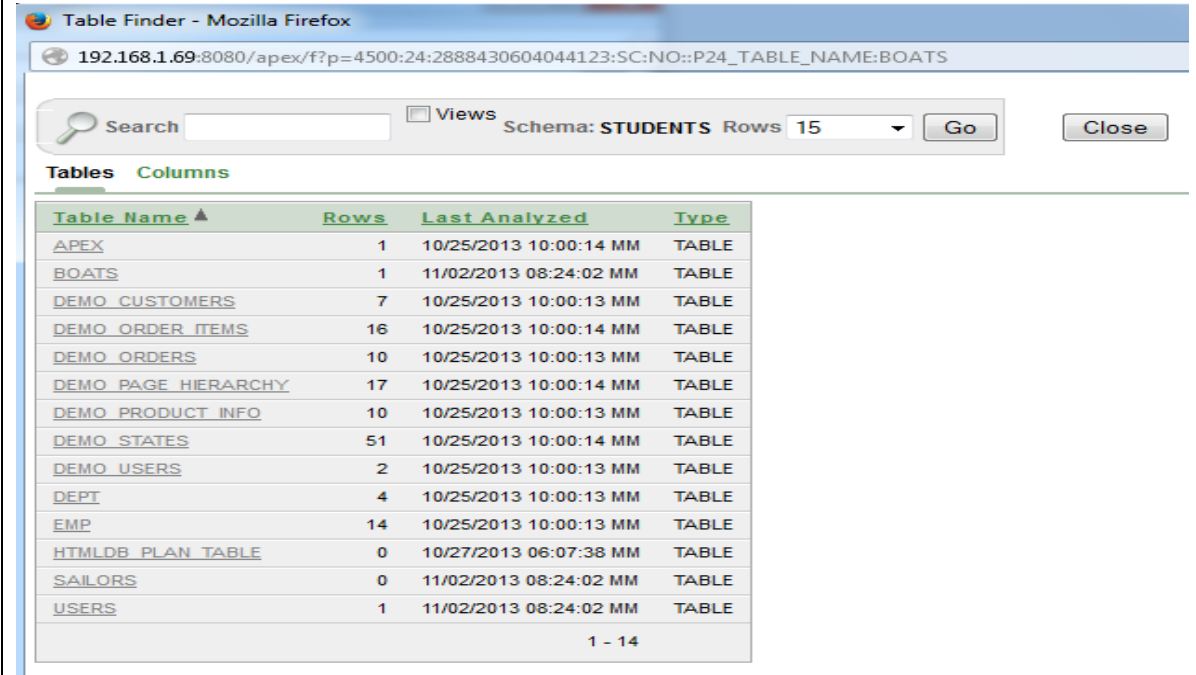
Βλέπουμε λοιπόν πως οι διαφορές δεν είναι τεράστιες με άλλα συστήματα, όπως το SQL Developer, έπρεπε όμως να παρατεθούν για πληρέστερη κατανόηση της λογικής στη λειτουργία του SQL Commands. Βλέπουμε πως έχουμε κάποιους περιορισμούς σχετικά με τον τρόπο σύνταξης των εντολών, αλλά θα τις συζητήσουμε σε παρακάτω ενότητα.

Keep in mind...

Οι ενέργειες αλλαγής της δομής βάσης, γίνονται **αυτόματα COMMIT**. Αν θέλετε να το κάνετε χειροκίνητα τότε απενεργοποιήστε το Checkbox με την λεζάντα «Auto-commit» πάνω από το πεδίο συγγραφής εντολών.

7.1.2.2.1.2 – Table Finder

Μια ωραία ευκολία που παρέχει το SQL Commands, είναι το εργαλείο Table Finder (το νούμερο 5 της προηγούμενης εικόνας), το οποίο επιτρέπει την προβολή των πινάκων του Schema καθώς και τον τύπο τους ή όποια άλλη χρήσιμη πληροφορία αναζητά ο χρήστης. Παρακάτω βλέπουμε ένα παράδειγμα:



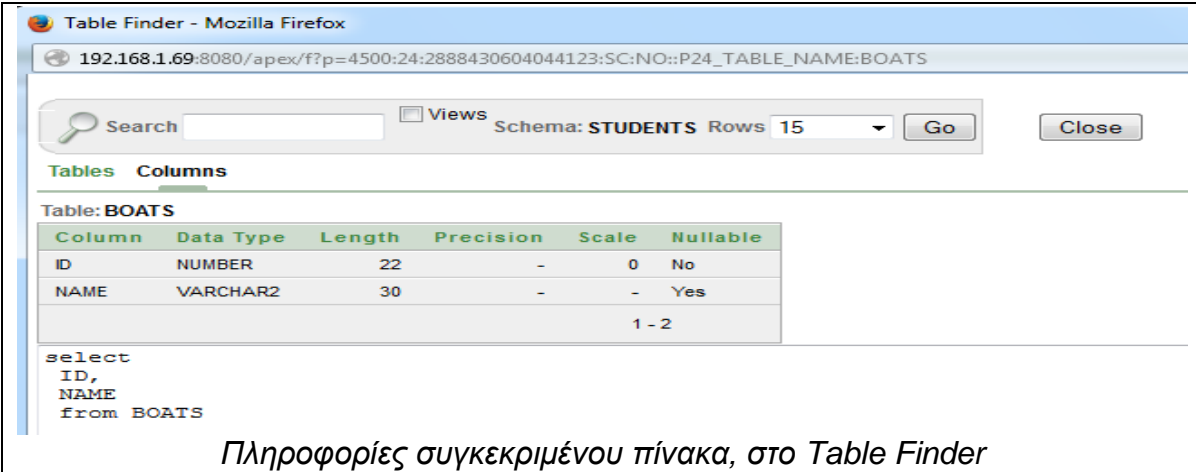
The screenshot shows the Table Finder interface in Mozilla Firefox. The address bar displays the URL: 192.168.1.69:8080/apex/f?p=4500:24:2888430604044123:SC:NO::P24_TABLE_NAME:BOATS. The search bar is empty, and the 'Views' checkbox is unchecked. The 'Schema' is set to 'STUDENTS' and 'Rows' is set to 15. A 'Go' button and a 'Close' button are visible. Below the search bar, there are tabs for 'Tables' and 'Columns'. The 'Tables' tab is active, displaying a table with the following data:

Table Name	Rows	Last Analyzed	Type
APEX	1	10/25/2013 10:00:14 MM	TABLE
BOATS	1	11/02/2013 08:24:02 MM	TABLE
DEMO_CUSTOMERS	7	10/25/2013 10:00:13 MM	TABLE
DEMO_ORDER_ITEMS	16	10/25/2013 10:00:14 MM	TABLE
DEMO_ORDERS	10	10/25/2013 10:00:13 MM	TABLE
DEMO_PAGE_HIERARCHY	17	10/25/2013 10:00:14 MM	TABLE
DEMO_PRODUCT_INFO	10	10/25/2013 10:00:13 MM	TABLE
DEMO_STATES	51	10/25/2013 10:00:14 MM	TABLE
DEMO_USERS	2	10/25/2013 10:00:13 MM	TABLE
DEPT	4	10/25/2013 10:00:13 MM	TABLE
EMP	14	10/25/2013 10:00:13 MM	TABLE
HTMLDB_PLAN_TABLE	0	10/27/2013 06:07:38 MM	TABLE
SAILORS	0	11/02/2013 08:24:02 MM	TABLE
USERS	1	11/02/2013 08:24:02 MM	TABLE

At the bottom of the table, it indicates '1 - 14'.

Οι πίνακες που υπάρχουν στο Schema, μέσω του Table Finder

Στο παραπάνω παράθυρο βλέπουμε τους πίνακες του Schema με τα ονόματά τους, τον αριθμό γραμμών τους και την τελευταία ημερομηνία τροποποίησής τους. Πατώντας πάνω σε ένα όνομα πίνακα, βλέπουμε πληροφορίες σχετικά με τον συγκεκριμένο πίνακα, όπως παρακάτω:



The screenshot shows the Table Finder interface with the 'Columns' tab selected. The 'Table: BOATS' is highlighted. Below the table name, there is a table with the following data:

Column	Data Type	Length	Precision	Scale	Nullable
ID	NUMBER	22	-	0	No
NAME	VARCHAR2	30	-	-	Yes

At the bottom of the table, it indicates '1 - 2'. Below the table, there is a SQL query:

```
select
ID,
NAME
from BOATS
```

Πληροφορίες συγκεκριμένου πίνακα, στο Table Finder

Βλέπουμε για παράδειγμα ότι ο πίνακας BOATS έχει δύο στήλες, την στήλη ID Τύπου Integer και την στήλη name τύπου VARCHAR (30). Επίσης βλέπουμε αν μια στήλη μπορεί να είναι Nullable ή όχι. Χρήσιμο λοιπόν εργαλείο αν πχ ο χρήστης θέλει να εκτελέσει INSERT INTO Statement και θέλει να γνωρίζει ακριβώς τις στήλες του πίνακα.

7.1.2.2.1.3 – Αποθήκευση εντολών με περιγραφή

Ένα ακόμα χρήσιμο εργαλείο που μας παρέχει το SQL Commands είναι το εργαλείο Save SQL. Το εν λόγω εργαλείο μας επιτρέπει να αποθηκεύσουμε ένα κομμάτι κώδικα, πχ δημιουργία μιας procedure, για μελλοντική χρήση. Η αποθήκευση γίνεται με όνομα και περιγραφή, ώστε ο χρήστης να προσδιορίζει που αποσκοπεί ο συγκεκριμένος κώδικας. Στην παρακάτω εικόνα το βλέπουμε επί τω έργω.

The screenshot shows the Oracle Application Express SQL Commands interface. The main window displays a PL/SQL procedure definition for 'change_boat_name'. A red box labeled '1' highlights the code. A red circle labeled '2' highlights the 'Save' button in the toolbar. A dialog box labeled '3' is open, showing the 'Name' field set to 'Change Boat name' and the 'Description' field set to 'Proc that changes the boats name'. A red circle labeled '4' highlights the 'Saved SQL' button in the bottom toolbar. A red circle labeled '5' highlights the resulting table of saved SQL commands, which includes the procedure name, description, and the full SQL code.

Owner	Name	Description	SQL	Updated By	Last Updated
LELERIS	Change Boat name	Proc that changes the boats name	CREATE OR REPLACE PROCEDURE change_boat_name (bid in number, newname in varchar2) IS oldname varchar(30); BEGIN SELECT name INTO oldname FROM boats WHERE id = bid; dbms_output.put_line('Changing name ' oldname ' to ' newname); UPDATE boats SET name = newname WHERE id = bid; END;	LELERIS	0 seconds ago

Αποθήκευση εντολών με περιγραφή

Η αποθήκευση εντολών με περιγραφή πραγματοποιείται με τα εξής βήματα. Αρχικά ο χρήστης συγγράφει τον κώδικα του (1), στην προκειμένη περίπτωση κώδικα δημιουργίας μιας Procedure που αλλάζει το όνομα της βάρκας στον πίνακα Boats. Στην συνέχεια πατώντας στο πλήκτρο **Save** (2) εμφανίζεται ένα παράθυρο στο οποίο εισάγουμε το **όνομα** και την **περιγραφή** του κώδικα που

αποθηκεύουμε (3). Αφού πατήσουμε Save στο προηγούμενο παράθυρο, μεταβαίνουμε στην επιλογή **Saved SQL** (4) και βλέπουμε όλους τους αποθηκευμένους κώδικες μας, με όνομα και περιγραφή (5). Πατώντας κλικ πάνω στο όνομα του αποθηκευμένου κώδικα, **αυτόματα** ο κώδικας εμφανίζεται στο παράθυρο συγγραφής. Χρήσιμο εργαλείο για αποθήκευση σύνθετου κώδικα.

Καλή πρακτική.

Τις εντολές που χρησιμοποιείτε συχνά, η απλά σύνθετες και μεγάλες εντολές, να τις αποθηκεύετε με την κατάλληλη περιγραφή για μελλοντική διευκόλυνσή σας.

7.1.2.2.1.4 – Περιορισμοί του SQL Commands

Παρά την ευχρηστία του και την ευκολία στην χρήση του, το SQL Commands υπόκειται σε ορισμένους περιορισμούς, τους οποίους υποχρεούμαστε να αναφέρουμε:

- 1) Δεν διαθέτει **τονισμό σύνταξης** (Syntax Highlighting) πράγμα που δυσχεραίνει λίγο την συγγραφή εντολών. Η συγγραφή όμως των εντολών σε ένα περιβάλλον με τονισμό σύνταξης όπως το Notepad++ ή το Kwrite και μεταφορά τους στο SQL Command δεν απαγορεύεται.
- 2) Εκτέλεση **ενός μπλοκ εντολών τη φορά**. Αυτό πρακτικά σημαίνει πως δεν μπορείτε να έχετε περισσότερα του ενός μπλοκ εντολών (BEGIN...END;) τη φορά. Αν δηλαδή έχετε ένα μπλοκ δημιουργίας procedure και στην συνέχεια ένα μπλοκ κλήσης της (ακόμα και χωρισμένα με /) οι εντολές δεν θα εκτελεστούν. Μπορείτε ωστόσο, στα όρια του δυνατού φυσικά, να βάζετε πολλές εντολές στο ίδιο μπλοκ, πχ αν καλείτε δύο διαφορετικές procedures, η κλήση τους μπορεί να γίνει από το ίδιο μπλοκ. (Σημείωση, ο Script Editor που θα δούμε παρακάτω υποστηρίζει συνεχόμενα blocks).

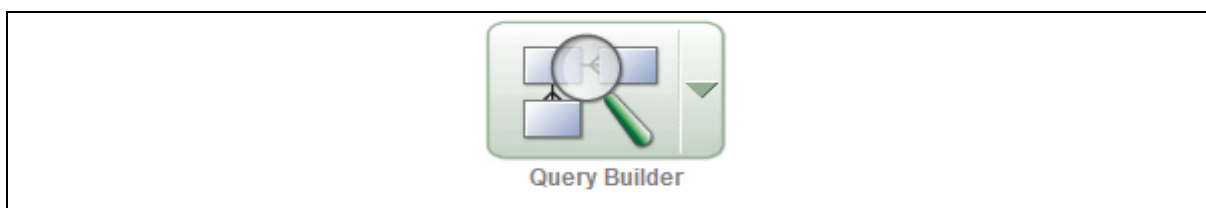
- 3) **Ανακριβής πληροφόρηση** σε περίπτωση συντακτικού σφάλματος. Αυτό πρακτικά σημαίνει ότι σε περίπτωση που ο parser εντοπίσει συντακτικό σφάλμα, οι πληροφορίες για το εν λόγω σφάλμα κάποιες φορές είναι είτε πολύ γενικευμένες είτε πολύ συνοπτικές. Αν και δεν συμβαίνει πάντα, ωστόσο είναι ενοχλητικό.

Συμπερασματικά όμως μπορούμε να πούμε ότι το APEX SQL Commands, είναι ένα αρκετά εύχρηστο εργαλείο με πληθώρα δυνατοτήτων που μπορεί να υποκαταστήσει κάποια εφαρμογή SQL Client όπως το SQL Developer. Αν και σε κάποιες δυνατότητες υστερεί, δεν παύει ωστόσο να είναι ένα ισχυρό εργαλείο όπου ο χρήστης μπορεί εύκολα και γρήγορα να πραγματοποιήσει τις επιθυμητές ενέργειες στην βάση δεδομένων. Και όλα αυτά μόνο από τον Web Browser της επιλογής του.

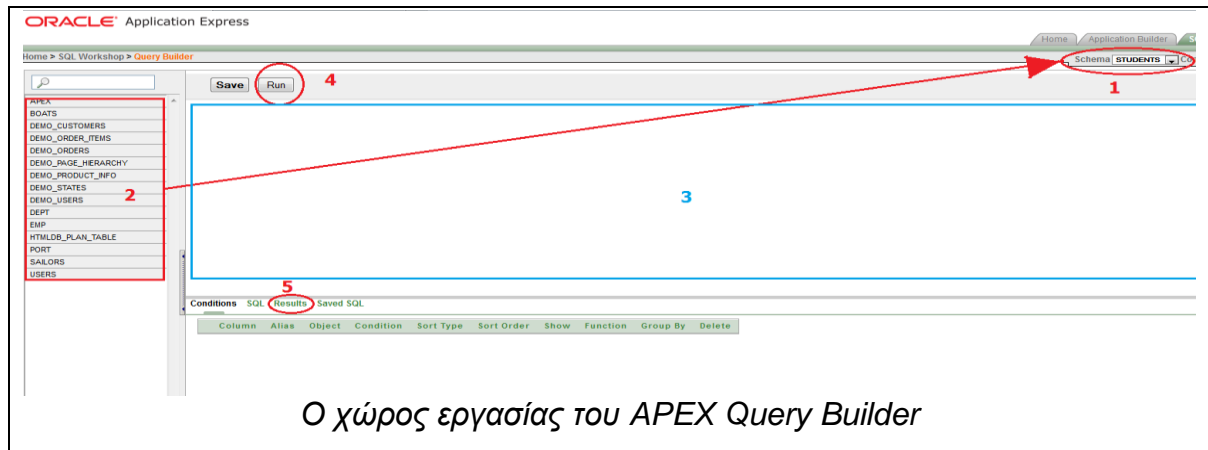
7.1.2.2.2 – Το εργαλείο APEX Query Builder

Στην ενότητα που περιγράψαμε το SQL Developer, είχαμε αναφέρει πως μια εφαρμογή Query Builder αναλαμβάνει την δημιουργία Επερωτημάτων SELECT για ανάκτηση πληροφοριών από την βάση, με την χρήση κριτηρίων φιλτραρίσματος πληροφοριών. Παρέχει τεράστια ευκολία σε αναζήτηση πολύ ιδιαίτερων πληροφοριών, οπότε θεωρείται πολύ χρήσιμη εφαρμογή.

Ως εκ τούτου λοιπόν το APEX δεν θα μπορούσε να στερηθεί Query Builder από την σουίτα εργαλείων του. Έχει όλες τις δυνατότητες ενός σύγχρονου Query Builder με το πρόσθετο ατού ότι είναι web based. Η πρόσβαση σε αυτό γίνεται μέσα από την σουίτα SQL Workshop στο παρακάτω εικονίδιο:



Παρακάτω θα δούμε την αρχική οθόνη του APEX Query Builder παρουσιάζοντας παράλληλα και τα βασικά στοιχεία που περιέχει:



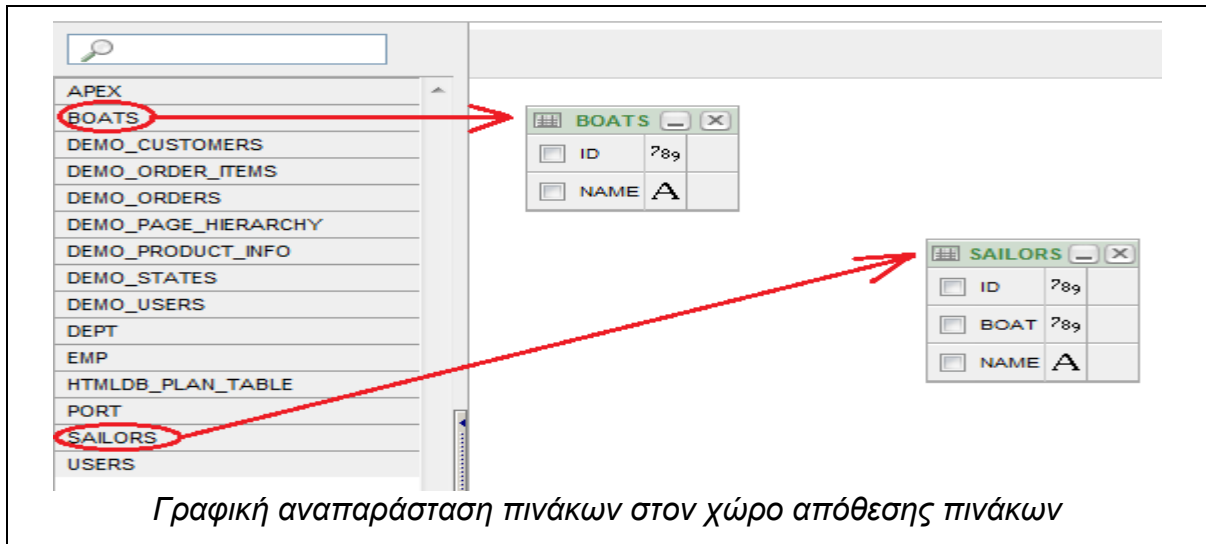
Στην αρχική οθόνη λοιπόν βλέπουμε πως υπάρχει ο επιλογέας Schema (1) και αριστερά υπάρχουν οι πίνακες που υπάρχουν μέσα σε αυτό (2). Στην συνέχεια διακρίνουμε τον χώρο απόθεσης πινάκων (3) όπου θα είναι ο χώρος στον οποίο θα υπάρχουν οι πίνακες που θα συμμετέχουν στο Query. Πάνω διακρίνεται το πλήκτρο εκτέλεσης του Query και τέλος βλέπουμε την επιλογή Results όπου βλέπουμε την έκβαση της εκτέλεσης του Query, κοινώς τις πληροφορίες που ψάχνουμε.

Προσοχή!

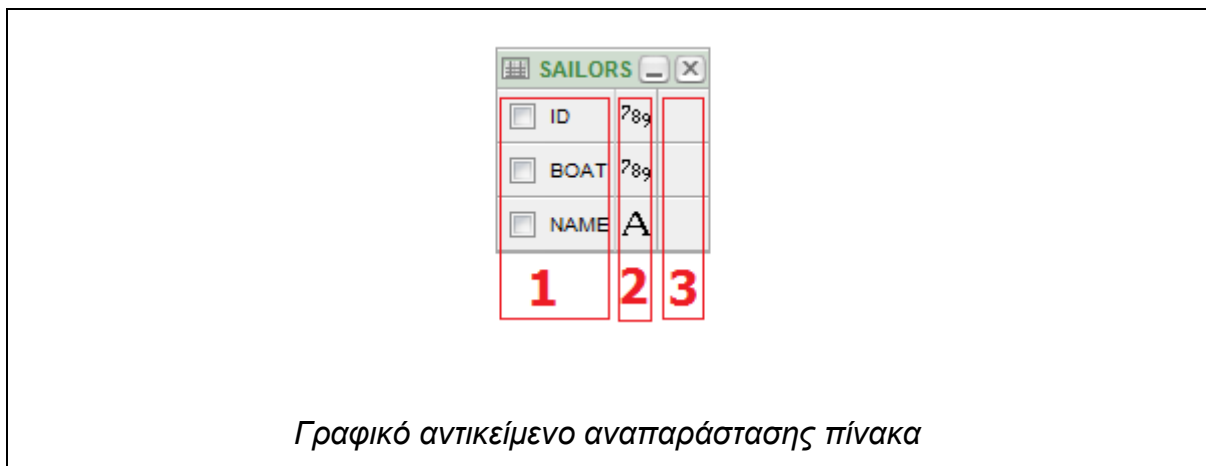
Το APEX Query Builder για να λειτουργήσει απαιτεί τον **Windows Internet Explorer**. Αν θέλετε να το χρησιμοποιήσετε μέσω του Mozilla Firefox, εγκαταστήστε το πρόσθετο **IE TAB** το οποίο θα βρείτε στην ιστοσελίδα <https://addons.mozilla.org/el/firefox/addon/ie-tab/>

7.1.2.2.1 – Εργασία στο APEX Query Builder

Για να δημιουργήσουμε λοιπόν ένα Query θα πρέπει να επιλέξουμε τους πίνακες που θέλουμε να συμμετέχουν σε αυτό. Η επιλογή γίνεται πατώντας απλά κλικ πάνω στο όνομα του πίνακα, στην αριστερή περιοχή της εφαρμογής. Ο πίνακας αυτόματα εμφανίζεται στον οθόνη απόθεσης, όπως βλέπουμε παρακάτω:



Το κουτάκι που αναπαριστά τον πίνακα περιέχει τρεις στήλες όπως βλέπουμε παρακάτω:



Η πρώτη στήλη περιέχει τα ονόματα των στηλών του πίνακα (1). Επιλέγοντας το Checkbox που διαθέτει κάθε γραμμή, επιλέγουμε την συμμετοχή της στήλης στο Query. Οι στήλες που συμμετέχουν στο Query, ορίζουν και το παραγόμενο αποτέλεσμα καθώς τα κριτήρια που θα δούμε παρακάτω, δίδονται με βάση τις τιμές που έχει μια στήλη.

Η δεύτερη στήλη είναι πληροφοριακού χαρακτήρα και μας δείχνει τον τύπο δεδομένων της κάθε στήλης του πίνακα πχ Integer, Varchar κλπ.

Η Τρίτη στήλη είναι ο επιλογέας συσχέτισης (Join). Για να λειτουργήσει φυσικά απαιτεί την ύπαρξη και δεύτερου τετραγώνου (πίνακα) στον χώρο απόθεσης. Για να συσχετίσουμε τις στήλες δύο πινάκων, πατούμε απλά κλικ στις γραμμές που αντικατοπτρίζουν τις στήλες του πίνακα και στα δύο τετράγωνα.

Εμείς λοιπόν επιλέγουμε τις στήλες που θέλουμε να συμμετέχουν στο Query. Απλά λοιπόν επιλέγουμε το Checkbox δίπλα στο όνομα της στήλης του πίνακα. Τότε στην καρτέλα «Conditions» εμφανίζονται οι συμμετέχουσες στήλες όπου ο χρήστης μπορεί να δώσει τα κριτήρια αναζήτησης πληροφορίας. Ας το περιγράψουμε:

Column	Alias	Object	Condition	Sort Type	Sort Order	Show	Function	Group By	Delete
NAME	NAME	BOATS		Asc	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>	x
ID	ID	BOATS		Asc	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>	x
ID	ID	SAILORS		Asc	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>	x
BOAT	BOAT	SAILORS		Asc	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>	x
NAME	NAME	SAILORS		Asc	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>	x

Η περιοχή Conditions, του APEX Query Builder

Η πρώτη στήλη είναι ο **επιλογέας σειράς**, δηλαδή επιλέγουμε με πια σειρά θα εμφανιστούν οι στήλες μέσα στο **SELECT Clause**. (1).

Η δεύτερη στήλη είναι πληροφοριακού χαρακτήρα και έχει το **όνομα της συμμετέχουσας στήλης του πίνακα** (2).

Η Τρίτη στήλη επιτρέπει την εμφάνιση της συμμετέχουσας στήλης στο αποτέλεσμα του Query, **με διαφορετικό όνομα** (3), κοινώς είναι το **κριτήριο AS** (SELECT column AS).

Στην συνέχεια έχουμε πάλι μια πληροφοριακού χαρακτήρα στήλη, που μας εμφανίζει το όνομα του πίνακα που ανήκει η συγκεκριμένη συμμετέχουσα στήλη (4).

Η επόμενη στήλη είναι και η πιο βασική. Σε αυτήν ο χρήστης δίνει τις τιμές που θέλει ως **κριτήρια αναζήτησης** πληροφοριών στον πίνακα. Εδώ ο χρήστης

μπορεί να δώσει τελεστές ισότητας / ανισότητας ή εύρος τιμών, δηλαδή =,>=,<=,<> μαζί με τις ζητούμενες τιμές. (5). Περισσότερες πληροφορίες θα δώσουμε παρακάτω κατά την λειτουργία του. Η στήλη αυτή αντιπροσωπεύει το **WHERE Clause**.

Η επόμενη στήλη επιτρέπει την επιλογή Ταξινόμησης είτε με αύξουσα σειρά (ASCending) είτε με Φθίνουσα (DESCending). (6). Κοινώς πρόκειται για το **ORDER BY Clause**.

Η στήλη Show έχει και αυτή σημαντικότητα καθώς ορίζει αν η συμμετέχουσα στήλη θα εμφανιστεί στο αποτέλεσμα ή όχι (7), δηλαδή αν θα υπάρχει το όνομά της μέσα στο SELECT Clause.

Η επόμενη στήλη δίνει την δυνατότητα επιλογής μιας SQL Function που θα λάβει χώρα στο SELECT δηλαδή **Count, Average, MIN, MAX, SUM**. (8).

Η προτελευταία στήλη δίνει την δυνατότητα **συμμετοχής της στήλης** στο **GROUP BY Clause**. (9).

Τέλος η τελευταία στήλη, **διαγράφει** την στήλη του πίνακα από τις συμμετέχουσες (όχι από τον πίνακα) (10).

7.1.2.2.2 – Δημιουργία Query

Τώρα που πήραμε μια ιδέα από το Query Builder, είμαστε έτοιμοι να δημιουργήσουμε ένα Query.

Αφού έχουμε επιλέξει τους πίνακες και τις στήλες αυτών που θέλουμε να συμμετέχουν στο επερωτήμα, δεν μένει παρά να ορίσουμε τα κριτήρια μας. Η στήλη Condition θα μας φανεί περισσότερο χρήσιμη από όλες τις άλλες. Θυμίζουμε πως η στήλη Conditions αντικατοπτρίζει το **WHERE Clause** του επερωτήματος.

Στην στήλη condition αρχικά πρέπει να δώσουμε τον αριθμητικό τελεστή που θα ορίσει το κριτήριό μας, ακολουθούμενο από την τιμή που θέλουμε να παράγει το επιθυμητό αποτέλεσμα. Μπορούμε να εισάγουμε τα εξής:

- 1) Σύμβολα **Ισότητας**, **Ανισότητας** η **συνδυασμό** αυτών, δηλαδή =,>,<,>=,<=,<> για αριθμητικές τιμές (πχ ID = 2)
- 2) Τελεστές **LIKE** για αναζήτηση pattern αλφαριθμητικών τιμών συνδυαζόμενοι με τους wildcard χαρακτήρες _ και % (πχ NAME LIKE 'Megaloc%')
- 3) Τελεστές **BETWEEN** για ορισμό εύρους τιμών. (πχ boats.ID BETWEEN 2 AND 4).
- 4) Τελεστές **OR**, **AND** και **NOT**. Ο τελεστής OR απαιτεί να ξαναγραφτεί το όνομα της στήλης με την άλλη περίπτωση πχ (boats.ID = 2 OR boats.ID = 6). Ο τελεστής AND δεν χρειάζεται καθώς σε περίπτωση εισαγωγής κριτηρίου σε περισσότερες από μία στήλες του πίνακα, δημιουργείται αυτόματα (πχ boats.ID = 2 AND boats.NAME like 'M%'), χρειάζεται όμως σε τελεστές BETWEEN. Τέλος ο τελεστής NOT χρησιμοποιείται για αντιστροφή ενός LIKE (πχ boats.name NOT LIKE 'A%')

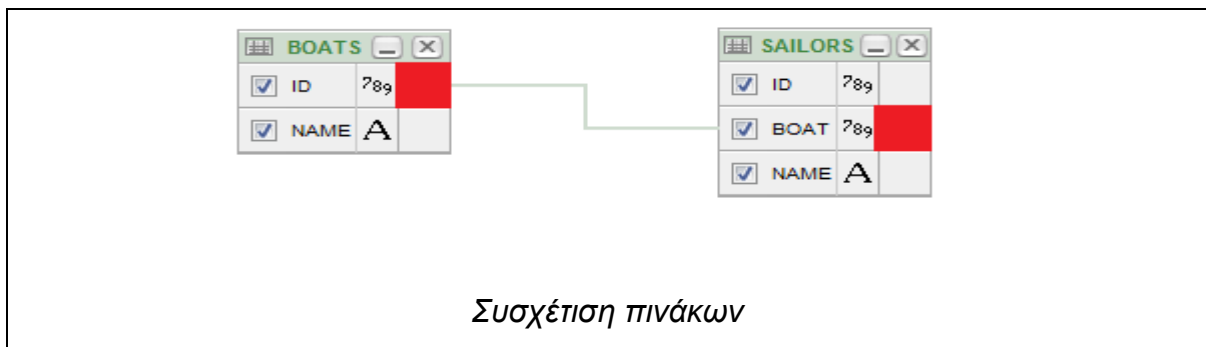
Τώρα λοιπόν που είδαμε τα είδη κριτηρίων μπορούμε να τα εφαρμόσουμε στο παρακάτω **παράδειγμα**:

Έχουμε λοιπόν τους πίνακες BOATS και SAILORS. Πληροφορικά, ο πίνακας BOATS έχει δύο στήλες, την ID που είναι και το Primary Key του πίνακα, και την NAME. Ο πίνακας SAILORS έχει τρεις στήλες, την ID που είναι το Primary Key του πίνακα, την BOAT που είναι Foreign Key στον πίνακα BOATS και την NAME.

Το ζητούμενο είναι το εξής: Θέλουμε να βρούμε τις εγγραφές **και των δύο πινάκων** που περιέχουν τα **ονόματα των βαρκών και των ιδιοκτητών τους**, οι

τιμές των αναγνωριστικών των βαρκών να κυμαίνονται **μεταξύ 2 ΚΑΙ 4** και το όνομα της βάρκας να μην ξεκινάει από το γράμμα **K**.

Ας το αναλύσουμε πρώτα. Αρχικά θέλουμε τις εγγραφές και των δύο πινάκων όποτε πρέπει να συσχετίσουμε τους πίνακες με τα κλειδιά τους (Primary και Foreign Key). Η συσχέτιση γίνεται απλά πατώντας κλικ στην γραμμή της τρίτης στήλης του τετραγώνου που παριστάνει τον πίνακα, με την αντίστοιχη γραμμή της τρίτης στήλης του άλλου τετραγώνου. Αν μπερδεύεστε δείτε παρακάτω πως γίνεται:



Στα κόκκινα τετράγωνα απλά πατάμε κλικ. Όταν επιλεγούν και οι δύο πίνακες, τότε ανάμεσά τους εμφανίζεται μια γραμμή που δηλώνει την συσχέτιση τους. Στην προκειμένη περίπτωση η συσχέτιση είναι *BOAT.ID = SAILORS.BOAT*

Το επόμενο ζητούμενο μας λέει πως θέλουμε να εμφανιστούν μόνο τα ονόματα των βαρκών και των ιδιοκτητών, οπότε από την στήλη «Show» απεπιλέγουμε τα πάντα εκτός από τις στήλες NAME των πινάκων BOATS και SAILORS.

Στη συνέχεια θέλουμε τα αναγνωριστικά των βαρκών (ID) να κυμαίνονται μεταξύ 2 και 4. Άρα στο «Condition» πεδίο της στήλης BOATS.ID γράφουμε *BETWEEN 2 AND 4*.

Τέλος θέλουμε το όνομα της βάρκας να μην ξεκινάει από K. Οπότε στο πεδίο «Condition» της στήλης BOATS.NAME γράφουμε *NOT LIKE 'K%'*.

Παρακάτω βλέπουμε συμπληρωμένα τα κριτήρια:

Conditions SQL Results Saved SQL

Column	Alias	Object	Condition	Sort Type	Sort Order	Show	Function	Group By	Delete
▲ ▼ ID	ID	BOATS	BETWEEN 2 AND 4	Asc		<input type="checkbox"/>		<input type="checkbox"/>	×
▲ ▼ NAME	NAME	BOATS	NOT LIKE 'K%'	Asc		<input checked="" type="checkbox"/>		<input type="checkbox"/>	×
▲ ▼ ID	ID	SAILORS		Asc		<input type="checkbox"/>		<input type="checkbox"/>	×
▲ ▼ BOAT	BOAT	SAILORS		Asc		<input type="checkbox"/>		<input type="checkbox"/>	×
▲ ▼ NAME	NAME	SAILORS		Asc		<input checked="" type="checkbox"/>		<input type="checkbox"/>	×

Τα κριτήρια συμπληρωμένα

Πατώντας στην καρτέλα SQL βλέπουμε το Query που μας δημιούργησε η εφαρμογή:

Conditions SQL Results Saved SQL

```

select  "BOATS"."NAME" as "NAME",
        "SAILORS"."NAME" as "NAME"
from    "SAILORS" "SAILORS",
        "BOATS" "BOATS"
where   "BOATS"."ID"="SAILORS"."BOAT"
and     "BOATS"."ID" BETWEEN 2 AND 4
and     "BOATS"."NAME" NOT LIKE 'K%'
    
```

Το αυτόματα δημιουργημένο SELECT Query

Και τέλος πατώντας στο πλήκτρο «RUN» εκτελείται το Query και βλέπουμε την έκβασή του παρακάτω:

Conditions SQL Results Saved SQL

NAME	NAME
Gorgoipikoos	Savvas
Menia-Maria	Kostas

row(s) 1 - 2 of 2

[Download](#)

Αποτέλεσμα Query

Βλέπουμε λοιπόν πως οι πληροφορίες που ανακτήσαμε συμφωνούν με τα κριτήρια που δώσαμε. Οπότε οδηγούμαστε στον λογικό συμπερασμό, πως το Apex Query Builder είναι μια πολύ αξιόλογη εφαρμογή, εύχρηστη και πολύ

αποδοτική. Ένα πλήρες Query Builder, που ανταγωνίζεται επάξια όμοιά του, και διαθέσιμο στην οθόνη του Web Browser μας.

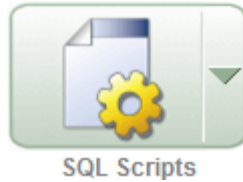
Keep in mind...

Το APEX Query Builder, δεν διαθέτει στήλη OR στην περιοχή Conditions όπως το Query Builder του SQL Developer. Αν θέλετε να εισάγετε OR Condition, τότε πρέπει στο πεδίο που αφορά την στήλη που θέλετε, πρέπει να γράψετε ολογράφως το όνομα της στήλης στο πεδίο Condition. Αν δηλαδή μιλάμε για την στήλη ID του πίνακα BOATS για παράδειγμα, και θέλετε να έχει τιμή 2 ή 5, τότε πρέπει να εισάγετε «=2 OR BOATS.ID = 5»

7.1.2.2.3 – Το εργαλείο APEX SQL Scripts

Υπάρχουν και περιπτώσεις όπου ο χρήστης επιθυμεί να εκτελέσει έναν μεγάλο αριθμό από εντολές. Μπορεί για παράδειγμα να θέλει να δημιουργήσει πίνακες και να τους γεμίσει με δεδομένα. Και για κάποιο A ή B λόγο ο χρήστης αναγκάζεται να κάνει DROP τους πίνακές του και να τους ξαναδημιουργήσει. Σε αυτήν την περίπτωση όμως θα πρέπει να γράψει ξανά όλες τις CREATE TABLE και INSERT INTO εντολές με το χέρι στο SQL Commands, κάτι το οποίο μεταφράζεται σε άσκοπη ταλαιπωρία. Δεν θα ήταν καλύτερα να μπορούσε τις εντολές να τις έχει αποθηκευμένες σε ένα αρχείο και να τις εκτελεί από εκεί όποτε το κρίνει σκόπιμο; Δηλαδή δεν θα ήταν καλύτερα αν μπορούσε να εκτελέσει **Scripts**;

Σίγουρα λοιπόν, το APEX δεν θα μπορούσε να μην παρέχει αυτή την δυνατότητα, γι αυτό περιέχει το εργαλείο SQL Scripts. Μέσω αυτής της web based εφαρμογής, ο χρήστης μπορεί να κάνει upload ένα ή περισσότερα αρχεία με scripts και να τα χρησιμοποιεί κατ' επιλογήν όποτε το κρίνει σκόπιμο. Φυσικά δίνεται η δυνατότητα αποθήκευσης των Script με όνομα ώστε να γνωρίζει ποια είναι η ιδιότητα και η λειτουργία του κάθε script. Και φυσικά ο χρήστης μπορεί να τροποποιήσει ένα script που ανέβασε μέσα από το APEX χωρίς να χρειάζεται να το ανεβάσει ξανά. Η πρόσβαση στο SQL Scripts γίνεται από το παρακάτω εικονίδιο μέσα στην σουίτα του SQL Workshop.

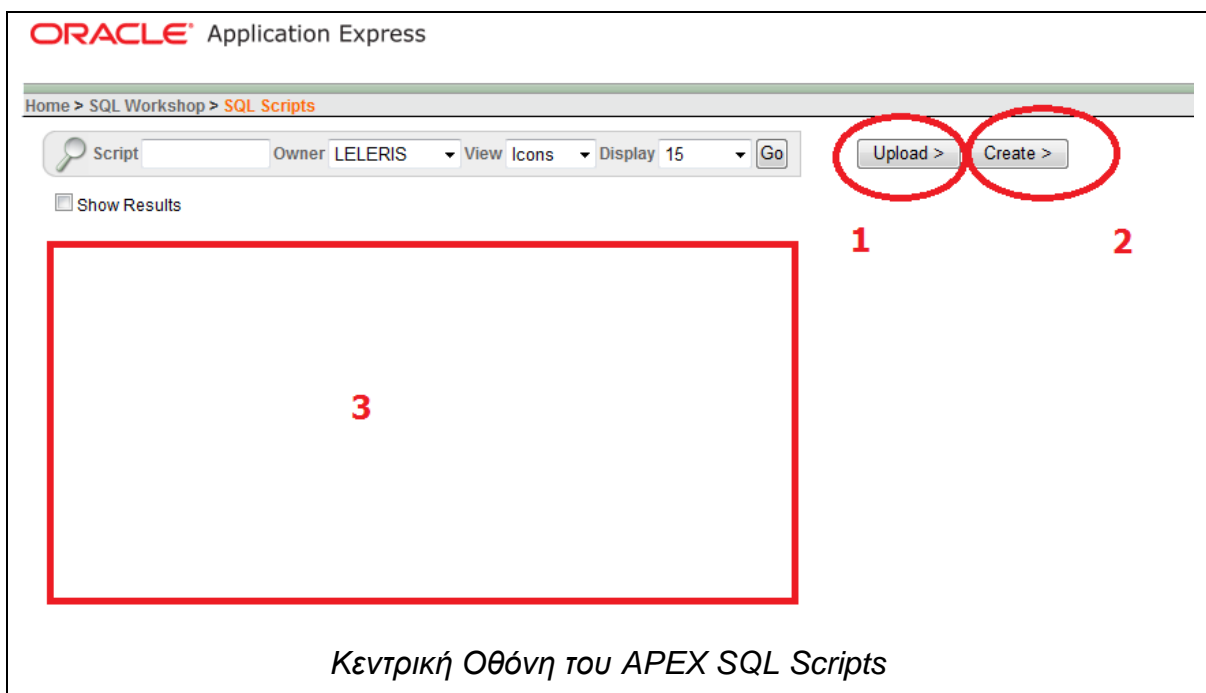


Προσοχή!

Όπως και το APEX Query Builder, έτσι και το APEX SQL Scripts, απαιτεί τον **Windows Internet Explorer** για να λειτουργήσει, διαφορετικά ο Script Editor δεν δουλεύει. Αν επιθυμείτε να εργαστείτε από Mozilla Firefox χρησιμοποιήστε το πρόσθετο **IE Tab**.

7.1.2.2.3.1 – Ανέβασμα αρχείου Script

Στο APEX μπορούμε να ανεβάσουμε (upload) στον Server ένα sql αρχείο με scripts και να το εκτελέσουμε όποτε το επιθυμούμε. Η διαδικασία ανεβάσματος ενός script στο Workspace του APEX είναι εύκολη υπόθεση. Υποθέτουμε πως έχουμε ήδη ένα αρχείο με SQL Scripts προς εκτέλεση οπότε μένει μόνο να το ανεβάσουμε. Παρακάτω βλέπουμε την κεντρική οθόνη του SQL Scripts



ORACLE Application Express

Home > SQL Workshop > SQL Scripts

Script Owner LELERIS View Icons Display 15 Go

Show Results

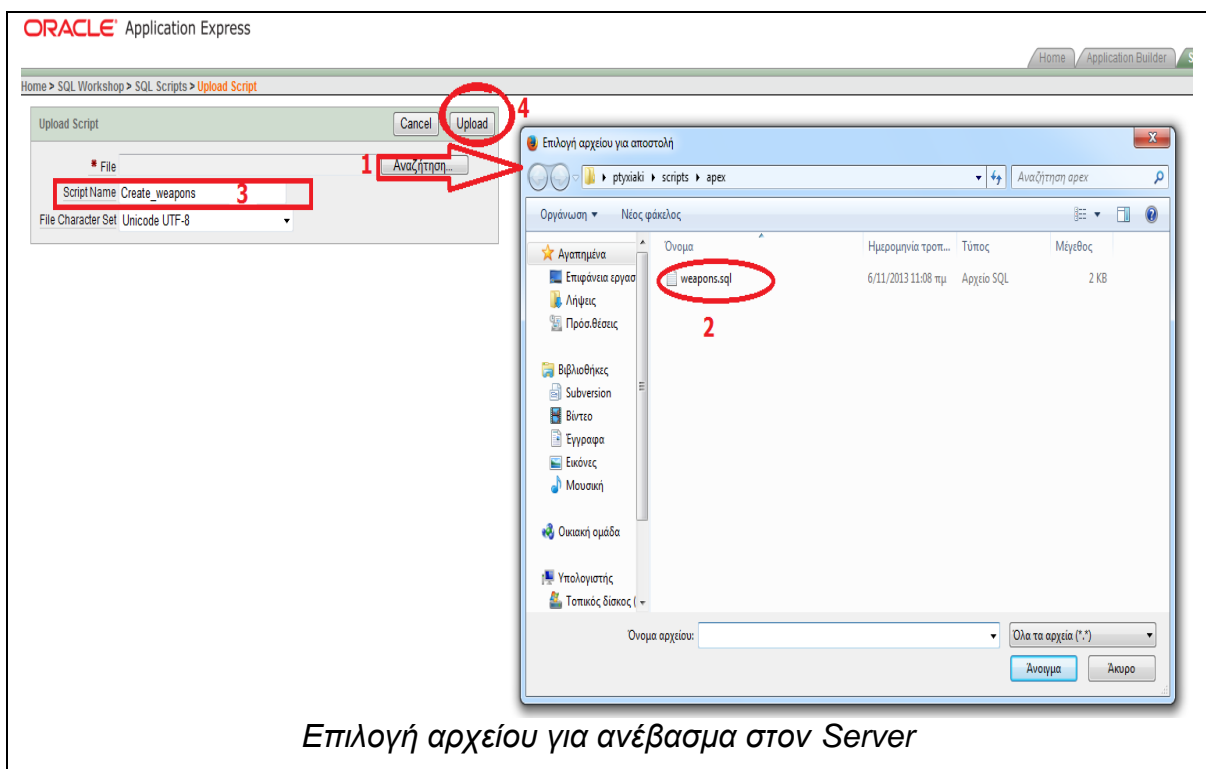
1 Upload > 2 Create >

3

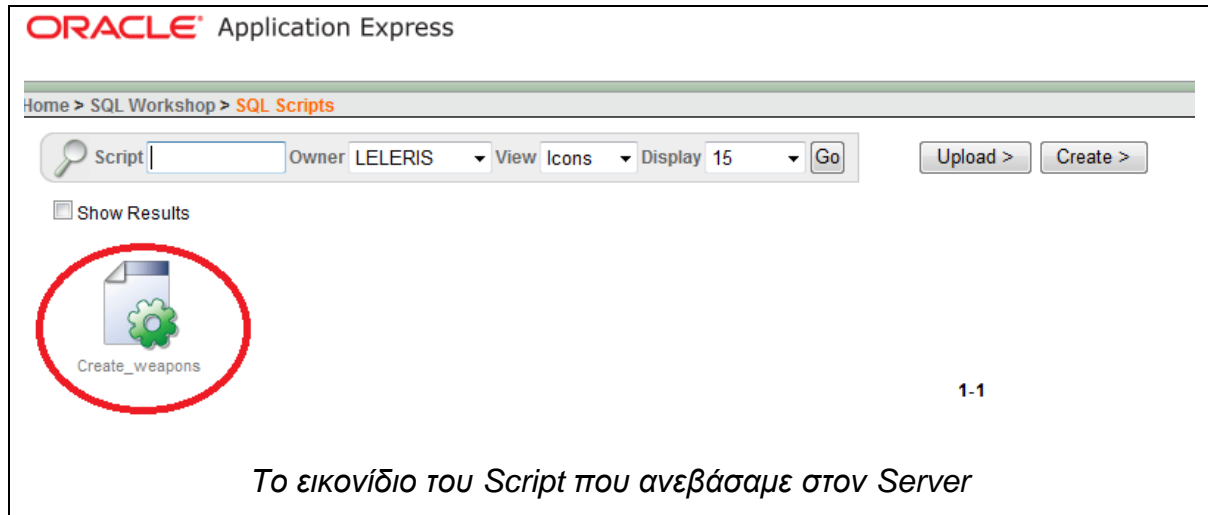
Κεντρική Οθόνη του APEX SQL Scripts

Το βασικό πλήκτρο είναι το «Upload» (1) όπου με αυτό ανεβάζουμε ένα αρχείο με SQL Scripts (κατάληξη .sql). Με το πλήκτρο «Create» (2) δημιουργούμε ένα script «στο φτερό» (on the fly) χωρίς να χρειάζεται να γίνει από αρχείο, αλλά θα το δούμε παρακάτω. Τέλος η περιοχή μαρκαρισμένη με το κόκκινο τετράγωνο (3) είναι ο χώρος απόθεσης των Script που ανεβάσαμε/δημιουργήσαμε.

Πατάμε «Upload» για να ανεβάσουμε ένα αρχείο script και μεταβαίνουμε στην παρακάτω οθόνη:



Αρχικά πατάμε στο κουμπί «Αναζήτηση» (1) όπου θα μας ανοίξει ένα File Browsing Window. Επιλέγουμε το αρχείο sql που θέλουμε να ανεβάσουμε στο APEX (2) και πατάμε «Άνοιγμα». Στην συνέχεια δίνουμε ένα όνομα στο Script (3) ώστε να καταλάβουμε μελλοντικά τον ρόλο του Script που ανεβάζουμε. Τέλος πατάμε «Upload» (4). Τότε μεταβαίνουμε πάλι στην αρχική οθόνη αλλά βλέπουμε κάτι νέο:



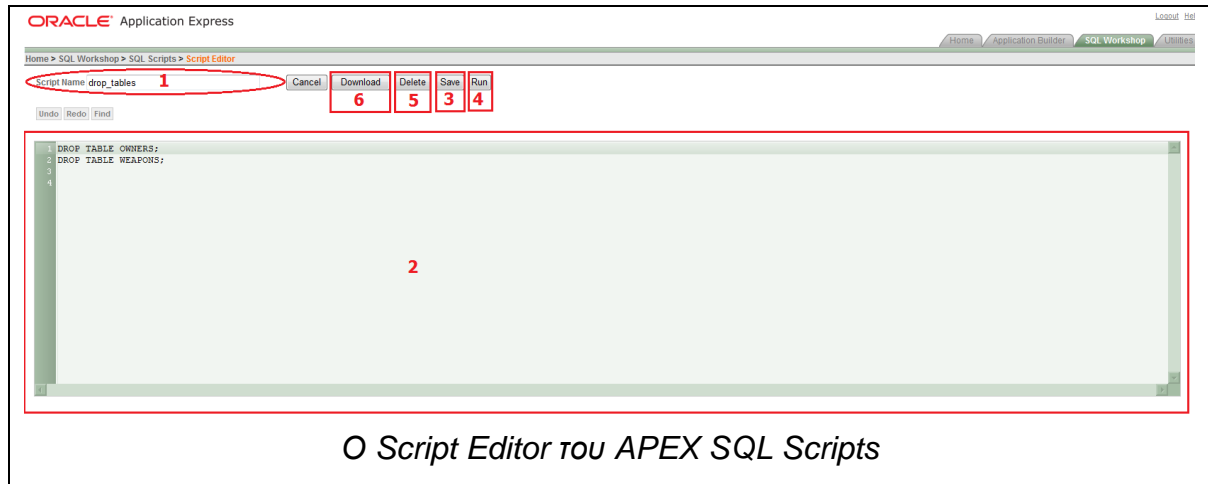
Πρόκειται για το Script που ανεβάσαμε. Πατώντας κλικ πάνω σε αυτό, μεταβαίνουμε στην διαδικασία εκτέλεσής του.

Keep in mind...

Σε αντίθεση με το APEX SQL Commands, στο APEX SQL Scripts τα συνεχόμενα PL/SQL Blocks, **επιτρέπονται**. Οπότε να γνωρίζετε πως μπορείτε να εισάγετε πολλαπλά blocks χωριζόμενα με το σύμβολο « / » στο script αρχείο σας.

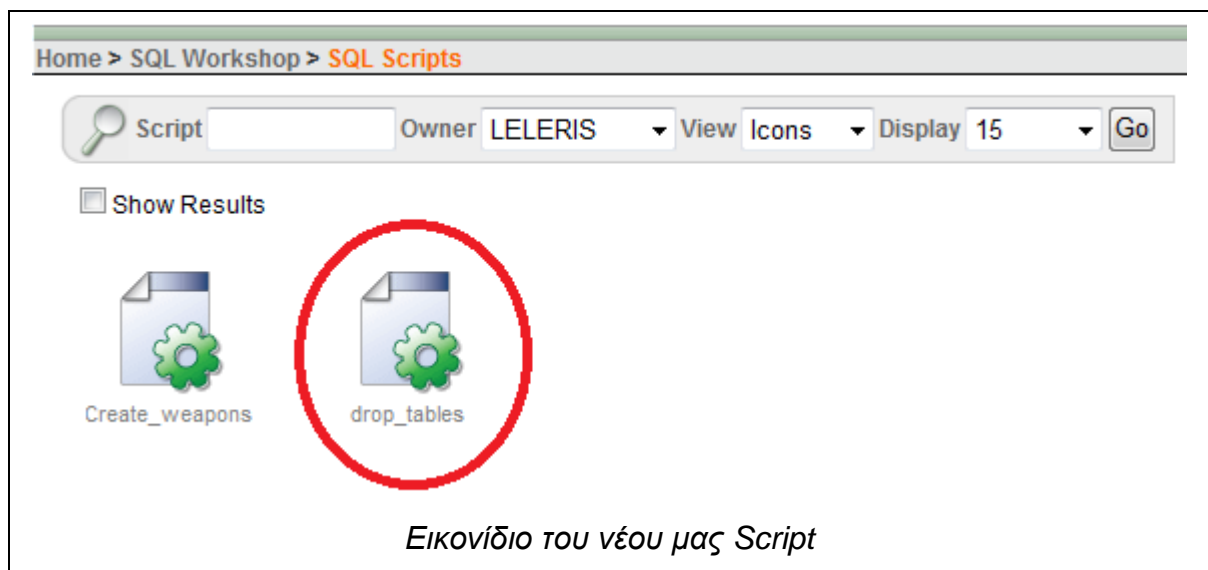
7.1.2.2.3.2 – Δημιουργία Scripts «On the fly»

Το APEX SQL Scripts μας επιτρέπει να δημιουργήσουμε «**στο φτερό**», δηλαδή επί τόπου, ένα Script χωρίς να χρειάζεται να ανεβάσουμε αρχείο στον Server. Απλά στην κεντρική οθόνη του SQL Scripts πατάμε στο πλήκτρο «Create». Τότε μεταβαίνουμε στην οθόνη του Script Editor όπως βλέπουμε παρακάτω:



Βασική πρώτη κίνηση είναι να δώσουμε όνομα στο Script μας, αυτό γίνεται στο πεδίο «Script Name» (1). Στην συνέχεια στην περιοχή του Script Editor (2) γράφουμε τις εντολές SQL-PL/SQL που θέλουμε να εκτελεστούν. Αφού ολοκληρώσουμε, πατούμε στο πλήκτρο «Save» (3) για να αποθηκεύσουμε το Script. Μπορούμε επίσης να πατήσουμε «Run» (4) για να εκτελέσουμε επί τόπου το script, το πλήκτρο «Delete» (5) για να το διαγράψουμε ή το πλήκτρο «Download» (6) για να το αποθηκεύσουμε σε αρχείο.

Πατώντας Save μεταβαίνουμε στην αρχική οθόνη όπου βλέπουμε το νέο μας Script:



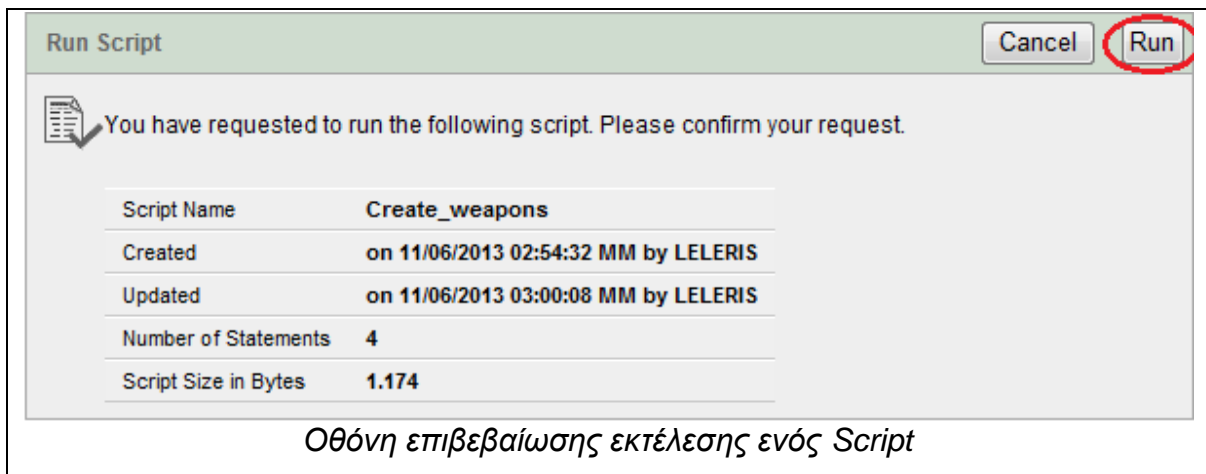
Πατώντας κλικ πάνω του μεταβαίνουμε στη διαδικασία εκτέλεσής του, την οποία θα περιγράψουμε στην προσεχή ενότητα.

Keep in mind....

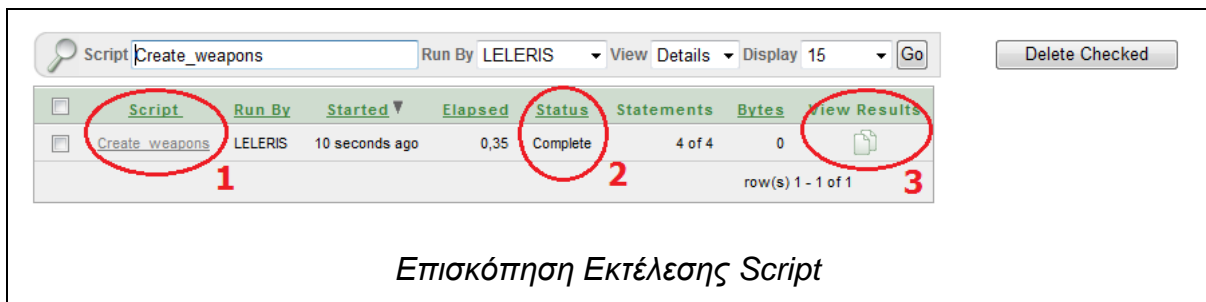
Όπως και στα αρχεία, έτσι και στον Script Editor επιτρέπονται τα συνεχόμενα PL/SQL Blocks.

7.1.2.2.3.3 – Εκτέλεση Scripts

Για να εκτελέσουμε το Script, πατάμε κλικ πάνω στο εικονίδιο με το όνομά του. Τότε μεταφερόμαστε πάλι στον Script Editor. Εκεί μπορούμε να κάνουμε τροποποιήσεις σε αυτό αν κρίνεται απαραίτητο. Διαφορετικά πατάμε στο πλήκτρο «Run» για να το εκτελέσουμε. Τότε μεταβαίνουμε στην οθόνη επιβεβαίωσης:



Απλά πατάμε στο πλήκτρο «Run». Τότε μεταβαίνουμε στην οθόνη εκτέλεσης του Script όπου μπορούμε να δούμε το Status της εκτέλεσης:



Βλέπουμε πως το Script που στείλαμε για εκτέλεση (1) βρίσκεται υπό επεξεργασία από το APEX. Στην στήλη «Status» (2) βλέπουμε την κατάσταση εκτέλεσης οι οποίες μπορεί να είναι:

- 1) **Submitted**, δηλαδή το Script απεστάλλει για εκτέλεση αλλά ακόμα δεν έχει εκτελεστεί
- 2) **Executing**, δηλαδή το Script εκτελείται την προκειμένη χρονική στιγμή
- 3) **Complete**, δηλαδή η εκτέλεση του Script ολοκληρώθηκε.

Αφού λοιπόν η εκτέλεση ολοκληρώθηκε, πατούμε στο εικονίδιο της στήλης View «Results» (3) για να δούμε την έκβαση της εκτέλεσης.

ORACLE® Application Express

Home > SQL Workshop > SQL Scripts > Results

Script: Create_weapons Status: Complete **3**

View: Detail Summary Display 5

Number	Elapsed	Statement	Feedback	Rows
1	0,03	CREATE TABLE WEAPONS(ID INTEGER NOT NULL PRIMARY KEY, NAME V	Table created.	0
2	0,04	CREATE TABLE OWNERS(ID INTEGER NOT NULL PRIMARY KEY, WID INT	Table created.	0
3	0,03	BEGIN INSERT INTO WEAPONS VALUES (1,'M1_Garand','Combat Rifi	1 row(s) inserted.	1
4	0,02	BEGIN INSERT INTO OWNERS VALUES (1,2,'Leanid Urumov'); INSER	1 row(s) inserted.	1
				row(s) 1 - 4 of 4

Download

Download

Statements Processed	4
Successful	4
With Errors	0

1

Οθόνη έκβασης της εκτέλεσης Scripts

Η έκβαση της εκτέλεσης του Script φαίνεται από την κάτω αριστερά περιοχή της οθόνης αποτελεσμάτων (1). Όπου βλέπουμε «Successful» σημαίνει πως ο αριθμός των εντολών που δείχνει ολοκληρώθηκαν επιτυχώς. Στην περιοχή «With Errors» αριθμούνται οι εντολές που δεν ολοκληρώθηκαν με επιτυχία (λόγο συντακτικού ή άλλου σφάλματος).

Στην οθόνη αποτελεσμάτων βλέπουμε τα βήματα εκτέλεσης (2). Οι εντολές που εκτελέστηκαν δίνουν και το σχετικό Feedback στο χρήστη ώστε να γνωρίζει αν εκτελέστηκε ή όχι και αν όχι για πιο λόγο. Να σημειωθεί πως τα block εντολών ο Script Parser τα αντιμετωπίζει σαν μεμονωμένη εντολή. Σε αυτήν την οθόνη ο χρήστης μπορεί να προσδιορίσει τον λόγο αποτυχίας εκτέλεσης μιας ή περισσότερων εντολών μέσω του Feedback.

Τέλος αν ο χρήστης επιθυμεί περισσότερες λεπτομέρειες, μπορεί να επιλέξει το Radio Button με όνομα «Detail» (3) όπου εμφανίζονται περισσότερες πληροφορίες σχετικά με την εκτέλεση του Script.

7.1.2.2.3.4 - Ορθή Σύνταξη

Η σύνταξη των scripts ακολουθεί την προτυποποίηση της Oracle PL/SQL δηλαδή:

- 1) Τις μεμονωμένες εντολές τις γράφουμε όπως είναι.
- 2) Εντολές CREATE TABLE και DROP TABLE τις γράφουμε όπως είναι αυτούσιες, ακόμα και αν είναι περισσότερες από μία.
- 3) Πολλαπλές εντολές όπως INSERT INTO, UPDATE...SET...WHERE κλπ τις βάζουμε σε BEGIN...END; Blocks.
- 4) Για Procedures/Functions και τύπους φροντίζουμε να τα οριοθετούμε με BEGIN...END; Blocks
- 5) Ο διαχωριστής των Blocks, επιτρέπεται, για την ακρίβεια **είναι υποχρεωτικός**.
- 6) SELECT Statements δεν έχει νόημα να βάλετε καθώς τα αποτελέσματά τους δεν θα είναι ορατά στην οθόνη αποτελεσμάτων. (SELECT ...INTO...FROM...WHERE επιτρέπονται για να εκχωρήσετε τιμή σε μεταβλητή).

Καλή πρακτική.

Το APEX SQL Scripts θυμίζει λίγο την αποθήκευση εντολών με περιγραφή που είδαμε στο SQL Commands. Είναι λοιπόν καλή πρακτική τις σύνθετες εντολές σας να τις αποθηκεύετε σε μορφή Script, γιατί εκτός του ότι τα Scripts μπορούν να περιέχουν πιο πολύπλοκες εντολές, έχουν το πρόσθετο ατού, ότι μπορούν να **μεταφερθούν**. Και ακόμα πιο θετικό είναι ότι τα Scripts που αποθηκεύετε (με την διαδικασία δημιουργίας «on the fly», μπορείτε να τα κάνετε και **Download** στον υπολογιστή σας. Μπορείτε αν θέλετε να χρησιμοποιείτε τα Scripts ως μηχανισμό **Backup-Restore** για να διατηρήσετε την ακεραιότητα των αντικειμένων του Workspace σας.

Και το SQL Scripts λοιπόν μας άφησε θετικές εντυπώσεις. Είδαμε πως είναι αρκετά εύχρηστο και πανίσχυρο. Έτσι λοιπόν με μία κίνηση μπορούμε να εκτελούμε μεγάλο αριθμό εντολών SQL και PL/SQL χωρίς κόπο. Από την οθόνη του Web Browser.

7.1.2.2.4 – Συμβουλές για το Oracle APEX

Θεωρούμε χρέος μας, να σας παραθέσουμε μερικές βέλτιστες πρακτικές για το APEX ώστε να γίνει η εργασία σας ευχάριστη και αποδοτική σε αυτό.

- 1) Εντολές που χρησιμοποιείτε συχνά, μπορείτε να τις αποθηκεύετε για μελλοντική χρήση.
- 2) Μεγάλες και σύνθετες εντολές, αποθηκεύετέ τις ως Scripts για να τις έχετε διαθέσιμες με ένα κλικ.
- 3) Χρησιμοποιείτε το Query Builder σε κάθε «περίεργα» και απαιτητικά Queries, για να γλυτώσετε χρόνο και άσκοπη ταλαιπωρία.

- 4) Το SQL Commands δεν περιέχει τονισμό σύνταξης αλλά υπάρχουν Text Editors γι αυτό το σκοπό. Μπορεί να μην τονίζει τις λέξεις κλειδιά, αλλά αντιγραφή και επικόλληση από Text Editor δεν απαγορεύει.
- 5) Γράφετε τις εντολές σας με κεφαλαία και τερματίζετε τις με Semicolon. Θα σας γλυτώσει από συντακτικά σφάλματα σε Case Sensitive λειτουργίες.
- 6) Σε κοινόχρηστα Workspaces με κοινόχρηστο Schema, σεβόμαστε την εργασία των άλλων χρηστών. Δουλεύετε ως ομάδα.

Συμπερασματικά, έχουμε μόνο καλά λόγια να πούμε για το Oracle Application Express. Είναι ισχυρό, εύχρηστο, αποδοτικό, φιλικό και διαθέσιμο από οπουδήποτε. Αποτελεί πολύ καλή επιλογή σαν εκπαιδευτική πλατφόρμα καθώς δίνει μεγάλες ευκαιρίες εύκολης εργασίας των σπουδαστών σε αυτό, και καλύτερη διαχείριση από τους διδάσκοντες. Δεν υπερβάλλουμε, αν εργαστείτε σε αυτό, τα ίδια θα πείτε κι εσείς, εγγυημένα!

7.2 – Προγραμματισμός εφαρμογών αλληλεπίδρασης με την Oracle Database

Σε αυτήν την ενότητα θα δούμε πως μπορείτε να προγραμματίσετε εφαρμογές οι οποίες θα μπορούν να αλληλεπιδρούν με μια βάση δεδομένων της Oracle. Καλύπτουμε τις δύο κυριότερες και πιο χρησιμοποιούμενες γλώσσες προγραμματισμού, την Java και την C++.

7.2.1 – Εφαρμογές σε γλώσσα Java (JDBC)

Η Java, μιας και είναι ιδιοκτησία της Oracle, εννοείται πως παρέχει άριστη υποστήριξη συνδεσιμότητας μιας εφαρμογής με την βάση δεδομένων της. Η

τεχνολογία η οποία παρέχει η Oracle για την σύνδεση μιας εφαρμογής Java με την βάση δεδομένων, ονομάζεται **Συνδεσιμότητα Βάσης Δεδομένων Java** (Java Data-Base Connectivity / JDBC)⁴¹.

Η εν λόγω τεχνολογία, είναι ένα API της Java, με την οποία ο χρήστης μπορεί να συνδεθεί στην βάση δεδομένων και να αλληλεπιδράσει μαζί της, δηλαδή να στέλνει επερωτήματα λαμβάνοντας αποτελέσματα και να εισάγει / τροποποιεί / διαγράφει δεδομένα από αυτήν. Περιέχει λοιπόν πολλές δυνατότητες, πολλές από τις οποίες θα τις ανακαλύψουμε μαζί στην πορεία.

Η τεχνολογία JDBC λειτουργεί σαν μια «γέφυρα» ανάμεσα στην Java και στην Oracle Database (ή σε οποιοδήποτε άλλο συμβατό DBMS όπως η PostgreSQL). Κύριο ρόλο στο JDBC διαδραματίζει ο **JDBC οδηγός** (driver) ο οποίος είναι ένας adapter από πλευράς πελάτη (client-side) και καθήκον του είναι να «μεταφράζει» τις αιτήσεις από μια εφαρμογή Java, σε ένα πρωτόκολλο επικοινωνίας, σε μια πιο κατανοητή γλώσσα αν θέλετε, που μπορεί να κατανοήσει το DBMS . Αναλόγως με το DBMS υπάρχουν τέσσερις τύποι JDBC Οδηγών⁴² οι οποίοι είναι:

- 1) **Τύπος 1.** Ο οδηγός αυτού του τύπου, βρίσκεται στην πλευρά του πελάτη (client side) και επικοινωνεί με τον ODBC (Open Database Connectivity) οδηγό που βρίσκεται στην πλευρά του εξυπηρετητή, και τελικώς στην βάση. Με πιο απλά λόγια μεταφράζει αιτήματα JDBC σε ODBC. Εφόσον λειτουργεί σε συνδυασμό με την τεχνολογία ODBC, το οποίο υποστηρίζεται από μεγάλο αριθμό DBMS, παρέχει προσβασιμότητα σε μεγάλο αριθμό από DBMS. Λόγω της πολυπλοκότητας όμως που εισάγει (μετάφραση του JDBC σε ODBC) έχει αντίκτυπο στην απόδοση.
- 2) **Τύπος 2.** Ο οδηγός αυτού του τύπου, χρησιμοποιεί τις **εγγενείς** (native) **βιβλιοθήκες** του εκάστοτε DBMS που βρίσκονται εγκατεστημένες στον

⁴¹ Wikipedia Article → Java Database Connectivity - Site(http://en.wikipedia.org/wiki/Java_Database_Connectivity)

⁴² Wikipedia Article → JDBC Driver - Site(http://en.wikipedia.org/wiki/JDBC_driver#Type_1_Driver_-_JDBC-ODBC_bridge)

πελάτη, για να μιλήσει απευθείας στην «γλώσσα» που καταλαβαίνει το εκάστοτε DBMS. Είναι γρήγορο αλλά είναι platform dependent και δεν το υποστηρίζουν όλα τα DBMS.

- 3) **Τύπος 3.** Ο οδηγός αυτού του τύπου επικοινωνεί πρώτα με μεσάζοντα Application Server, ο οποίος επιτρέπει την επικοινωνία με DBMS διαφορετικών κατασκευαστών, απλά μεταφράζοντας τα αιτήματα του πελάτη στο πρωτόκολλο δικτύου που καταλαβαίνουν τα DBMS αυτά. Είναι platform-independent ωστόσο προκαλεί κάποιο αντίκτυπο στην απόδοση.
- 4) **Τύπος 4.** Ο οδηγός αυτού του τύπου είναι γνωστός και ως **Pure Java Driver**. Μεταφράζει απευθείας τα αιτήματα του πελάτη στο DBMS, και λόγω αυτού παρέχει καλή απόδοση και επειδή είναι δημιουργημένο σε Java, είναι και platform-independent. Προορίζεται όμως για DBMS συγκεκριμένου κατασκευαστή, οπότε είναι vendor-dependent. Τέτοιου τύπου είναι τα JDBC Drivers της Oracle.

Οι οδηγοί της Oracle, είναι **τύπου 4**⁴³ και επιτρέπουν την απευθείας επικοινωνία (χωρίς μεσάζοντες) της εφαρμογής Java με την βάση της Oracle. Έτσι λοιπόν, έχουν την καλύτερη δυνατή απόδοση και υποστήριξη από την Oracle Database. Ονομάζονται **JDBC Thin Drivers**.

7.2.1.1 – Η τεχνολογία JDBC επί τω έργω

Για να κατανοήσουμε όμως καλύτερα την τεχνολογία JDBC πρέπει να την δούμε εν λειτουργία. Θα χρησιμοποιήσουμε τον JDBC Thin Driver της Oracle, ο οποίος παρέχεται με το πακέτο λογισμικού **Oracle Instant Client**. Θα δημιουργήσουμε λοιπόν μια εφαρμογή σε Java και θα χρησιμοποιήσουμε την

⁴³ Oracle TechNet → JDBC FAQ → JDBC In General → “JDBC Thin client-side driver: This is a JDBC Type 4 driver that” – Site (http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-faq-090281.html#01_01)

JDBC για να συνδεθούμε στην βάση δεδομένων που τρέχει στον Server. Θα εργαστούμε στο περιβάλλον εργασίας **NetBeans IDE**.

Δημιουργούμε λοιπόν ένα νέο Project στο Netbeans και χρησιμοποιούμε τον Oracle JDBC Thin Driver 6 για την σύνδεση (οδηγίες απόκτησης και εγκατάστασης θα βρείτε στο παράρτημα Γ΄) και ξεκινούμε την συγγραφή του κώδικα.

Αρχικά, θα πρέπει να εισάγουμε στην εφαρμογή μας, την κλάση Java.SQL. Αρχίζουμε λοιπόν με τον παρακάτω κώδικα:

```
package oraclejdbc;  
  
import java.sql.*; //1
```

Η γραμμή Import (1) εισάγει στην εφαρμογή μας την κλάση Java.SQL η οποία περιέχει όλες τις απαραίτητες μεθόδους/εργαλεία για να εργαστούμε στην βάση δεδομένων.

Στην συνέχεια μπορούμε να μπούμε στην Main και να αρχίσουμε να αναπτύσσουμε την εφαρμογή μας, ωστόσο πρώτα πρέπει να γράψουμε το κάτωθι:

```
public static void main(String[] args) throws SQLException { //1
```

Στην main μέσα, η εφαρμογή μας μπορεί ανά πάσα στιγμή να πετάξει SQLException (πχ λόγω λάθος credentials σύνδεσης, SQL συντακτικό λάθος κλπ), οπότε πρέπει να ενημερώσουμε ότι η main πετάει SQLExceptions με την λέξη throws(1).

Θα συνδεθούμε στην βάση ως χρήστης JDBC, ο οποίος έχει τον πίνακα boats στο schema του όπως παρακάτω:

```
SQL> select * from boats;

   ID NAME
-----
    1 Ouranoupolis
    2 Panormitis
    3 Seahorse
    4 Mariza
    5 Kap.Dimitrios
```

Τα περιεχόμενα του πίνακα που θα εργαστούμε

Αρχικά θα πρέπει να δημιουργήσουμε ένα αντικείμενο τύπου **Connection** το οποίο θα αναλάβει την σύνδεσή μας με την βάση. Ο παρακάτω κώδικας κάνει ακριβώς αυτό:

```
Connection con=null;

con=DriverManager.getConnection("jdbc:oracle:thin:@192.168.1.69:1521:test","jdb
bc","test123");
```

Το αντικείμενο con έχει την ιδιότητα getConnection το οποίο δέχεται σαν όρισμα το **Connection String** για την σύνδεση με την βάση. Ας το αναλύσουμε:

```
«jdbc:oracle:thin:» = Το JDBC Thin Driver της Oracle

«@192.168.1.69:1521:» = Η IP Διεύθυνση του server και η θύρα που «ακούει» για
συνδέσεις

«test","jdbc","test123"» = Το Instance Name της βάσης (SID), το όνομα χρήστη
(jdbc) και το password (test123).
```

Τώρα θα πρέπει να ελέγξουμε αν η σύνδεση εγκαθιδρύθηκε, ώστε να ξέρουμε από πρώτο χέρι ότι είμαστε μέσα στην βάση. Επίσης δεν θέλουμε απρόοπτα κατά την εκτέλεση του προγράμματός μας, οπότε πρέπει να φροντίσουμε και για τις εξαιρέσεις. Ο παρακάτω κώδικας πραγματοποιεί το ζητούμενο:

```

if (con != null) { //1
    try { //2
        System.out.println("Connection Established\n"); //3
    } //try
    catch (SQLException ex) //4
    {
        System.err.println("Exception : " + ex.getMessage()); //5
    }
} //if
else { //6
    System.err.println("Connection Failed\n"); //7
} //else

```

Στο **IF Statement** (1) ελέγχουμε αν το αντικείμενο con της σύνδεσης είναι null ή όχι. Επίσης για να αποφύγουμε να συμβεί exception και να κολλήσει το πρόγραμμα, βάζουμε τον κώδικα μέσα σε **try** (2). Αν το con δεν είναι null, η σύνδεση πέτυχε, οπότε τυπώνουμε ένα πληροφοριακό μήνυμα στο stdout (3). Αν όμως είναι, στο **else statement** (6) τυπώνουμε ένα μήνυμα σφάλματος στο stderr (7). Αν συμβεί εξαίρεση, θα εκτελεστεί το **Catch Block** (4) το οποίο περιμένει εξαιρέσεις τύπου SQLException, οπότε θα τυπώσει στο stderr ένα μήνυμα με το είδος της εξαίρεσης (5). Εφόσον εμείς θέλουμε να εργαστούμε προφανώς όταν η σύνδεση γίνει, ο κώδικάς μας θα φιλοξενηθεί μέσα στο IF Statement.

Τώρα θέλουμε να δημιουργήσουμε ένα **SELECT Statement** για να δούμε τα περιεχόμενα του πίνακα. Γράφουμε τα κάτωθι (μέσα στο if statement που είδαμε παραπάνω):

```

if (con != null) {
    try{
        System.out.println("Connection Established\n");

```

```
String qry = "SELECT * FROM BOATS"; //1
Statement st=con.createStatement(); //2
ResultSet rs = st.executeQuery(qry); //3
```

Πρώτα φτιάχνουμε ένα string που θα πάρει σαν τιμή, το SQL SELECT Statement μας (1). Στην συνέχεια δημιουργούμε ένα αντικείμενο Statement το οποίο θα αναλάβει να εκτελέσει το Query μας (2). Τέλος δημιουργούμε ένα αντικείμενο **Result Set** το οποίο θα λάβει τα αποτελέσματα του Query, κοινώς τις εγγραφές του πίνακα (3).

Ορισμός.

Το **Result Set**, είναι μια δομή που κρατάει και παρουσιάζει στο χρήστη το περιεχόμενο των γραμμών ενός πίνακα ή περισσότερων πινάκων από τη βάση (συμπεριλαμβανομένου και Meta Data). Πρόκειται δηλαδή για αντίγραφο του πίνακα στην πλευρά του πελάτη ⁴⁴. Για εκτέλεση Επερωτημάτων στην βάση λοιπόν, είναι απαραίτητο.

Ο πίνακας προφανώς δεν θα έχει μόνο μία γραμμή, αλλά πιθανώς περισσότερες. Οπότε για να τις δούμε όλες, γράφουμε τον παρακάτω κώδικα:

```
while(rs.next()) //1
{
    int id = rs.getInt("ID"); //2
    String name = rs.getString("NAME"); //3
    System.out.println(id + "\t" + name); //4
} //while
```

Αρχικά, ορίζουμε ένα βρόγχο while ο οποίος θα τρέχει όσο το Result Set λαμβάνει νέες γραμμές (1). Στην συνέχεια ορίζουμε δύο μεταβλητές, μια τύπου Integer η οποία θα λαμβάνει τις εγγραφές της πρώτης στήλης, που είναι αριθμοί (2) και μια δεύτερη τύπου String η οποία θα λαμβάνει τις εγγραφές της δεύτερης

⁴⁴ Wikipedia Article → Result Set – Site (http://en.wikipedia.org/wiki/Result_set)

στήλης που είναι τύπου VARCHAR (3). Τέλος τυπώνουμε στο stdout τις εγγραφές που «ψαρέψαμε».

Τώρα θα θέλαμε να εισάγουμε στον πίνακα ένα νέο πεδίο, δηλαδή θα εκτελέσουμε **Insert Statement**. Ο παρακάτω κώδικας θα εισάγει στον πίνακα μια νέα γραμμή:

```
PreparedStatement pstmt = con.prepareStatement ("insert into BOATS(id,
name) values (?, ?)"); //1

pstmt.setInt(1,6); //2

pstmt.setString (2, "Megaloxari"); //3

pstmt.executeUpdate();//4

st.executeQuery("commit"); //5

System.out.println("Row Inserted Successfully"); //6
```

Ας δούμε τώρα τι κάναμε. Αρχικά δημιουργούμε ένα **PreparedStatement** το οποίο θα φροντίσει να πραγματοποιήσει το Insert στην βάση μας. Βλέπουμε πως οι τιμές δίδονται ως «μπαλαντέρ» με ερωτηματικά, δηλαδή **παραμετρικά** (1). Στην συνέχεια δίνουμε την πρώτη παράμετρο, δηλαδή την τιμή 6 (στήλη ID) (2). Στην συνέχεια δίνουμε και την δεύτερη παράμετρο, την τιμή «Megaloxari» καθώς η στήλη Name είναι τύπου Varchar (3). Εν συνεχεία εκτελούμε το Update Statement όπου θα εισαχθεί η νέα γραμμή (4). Επειδή θέλουμε να βεβαιωθούμε πως οι αλλαγές στον πίνακα θα μείνουν, εκτελούμε Commit (5). Τέλος, πληροφορούμε στο stdout ότι η γραμμή μπήκε στον πίνακα. (Αν υπάρχει ήδη η εγγραφή όμως, θα συμβεί Exception γιατί παραβιάζεται η μοναδικότητα, η στήλη ID είναι Primary Key).

Ορισμός

Ένα Prepared Statement, είναι ένα Statement το οποίο αποθηκεύεται στην **λανθάνουσα μνήμη** (cache memory) του Server, και επιτρέπει την πολλαπλή και πιο αποδοτική εκτέλεσή του ⁴⁵.

⁴⁵ Wikipedia Article → Prepared Statement -Site (http://en.wikipedia.org/wiki/Prepared_statement)

Τώρα εμείς θέλουμε να αλλάξουμε μια εγγραφή του πίνακα, δηλαδή, να κάνουμε **UPDATE**. Ο παρακάτω κώδικας εκτελεί UPDATE στον πίνακα της βάσης:


```

pstmt = con.prepareStatement("UPDATE BOATS SET NAME = ? WHERE ID =
?"); //1

pstmt.setInt(2, 3); //2

pstmt.setString(1, "Eliza"); //3

pstmt.executeUpdate();//4

st.executeQuery("commit"); //5

System.out.println("Table Updated\n"); //6

```

Χρησιμοποιούμε ξανά το αντικείμενο PreparedStatement που χρησιμοποιήσαμε πριν. Του δίνουμε το UPDATE Statement, ξανά παραμετρικά (1). Δίνουμε ως δεύτερη παράμετρο την τιμή που θέλουμε να συμφωνήσει στο WHERE Clause (2) και ως πρώτη παράμετρο του δίνουμε την νέα τιμή που θέλουμε να εισάγουμε στον πίνακα, δηλαδή η SET (3). Εν συνεχεία εκτελούμε το Update Statement (4), κάνουμε Commit στην βάση (5) και τυπώνουμε ένα μήνυμα stdout για πληροφόρηση του χρήστη.

Τελειώνοντας, θέλουμε να διαγράψουμε από τον πίνακα την γραμμή με την καταχώρηση «Kar.Dimitrios», δηλαδή θέλουμε να εκτελέσουμε **DELETE Statement**. Ο παρακάτω κώδικας πραγματοποιεί αυτό ακριβώς:

```

pstmt = con.prepareStatement("DELETE FROM BOATS WHERE ID = ?"); //1

pstmt.setInt(1, 5); //2

pstmt.executeUpdate();//3

st.executeQuery("commit"); //4

System.out.println("Row Deleted\n"); //5

```

Για άλλη μια φορά θα χρησιμοποιήσουμε το PreparedStatement αντικείμενο, δίνοντας του το DELETE Statement ως κείμενο, με παράμετρο (1). Στην συνέχεια δίνουμε στην πρώτη και μοναδική παράμετρο την τιμή 5 για το WHERE Clause (2), εκτελούμε το Statement (3) και κάνουμε Commit τις αλλαγές (4). Τέλος τυπώνουμε και στο stdout ένα μήνυμα αναγγελίας της διαγραφής (5).

Αφού τελειώσαμε με την εργασία μας στην βάση, πρέπει να τερματίσουμε την σύνδεση με αυτήν. Ο παρακάτω κώδικας το περατώνει:

```

st.close(); //1
con.close(); //2
    } //try
catch(SQLException ex)
..... } //end of main
.... } //end of class
    
```

Πρώτα λοιπόν κλείνουμε το Statement αντικείμενο καθώς δεν θέλουμε να αποστείλουμε άλλα Statements στην βάση (1) και τέλος κλείνουμε την σύνδεση (2). Κάπου εδώ τελειώνει και η εφαρμογή μας. Τώρα δεν μένει παρά να την μεταγλωττίσουμε και να την εκτελέσουμε. Κάνουμε **Build** το Project και στην συνέχεια κάνουμε **Run**. Ενδεικτική έξοδος της εφαρμογής δίδεται κάτωθι:

```

.....
Connection Established ← Σύνδεση OK
1      Ouranoupolis
2      Panormitis
3      Seahorse
4      Mariza
5      Kap.Dimitrios
Row Inserted Successfully ← Insert Statement
Table Updated ← Update Statement
Row Deleted ← Delete Statement
BUILD SUCCESSFUL (total time: 0 seconds)
    
```

Έξοδος του προγράμματος μας στην οθόνη

Βλέπουμε λοιπόν πως το πρόγραμμά μας λειτουργεί άψογα και πραγματοποίησε τις ενέργειες που ζητήσαμε στην βάση. Επίσης δεν παρατηρήθηκε καθυστέρηση στην απόδοση των Statements, πράγμα που μας επιβεβαιώνει πως ο JDBC Thin Driver όντων είναι ταχύτατος. Τον πλήρες κώδικα της εφαρμογής θα τον βρείτε στο παράρτημα Δ΄.

7.2.3 - Εφαρμογές σε γλώσσα C++ (OCCI)

Φυσικά οι «παραδοσιακοί» προγραμματιστές της C++ δεν θα μπορούσαν να μείνουν εκτός. Η Oracle φρόντισε γι αυτούς, προσφέροντας την **Διεπαφή Κλήσεων C++**⁴⁶ (Oracle C++ Call Interface / OCCI). Πρόκειται για μια διεπαφή που δίνει την δυνατότητα σε μια εφαρμογή C++ να αλληλεπιδρά με μια βάση δεδομένων της Oracle και μάλιστα με πολύ καλές επιδόσεις σε ταχύτητα.

Το OCCI παρέχει πλήρη υποστήριξη SQL και PL/SQL, πλήρης υποστήριξη δημιουργίας τύπων και αντικειμένων και πολλές άλλες δυνατότητες που το καθιστούν απαραίτητο εργαλείο για έναν προγραμματιστή C++. Υστερεί ωστόσο σε μεταφερσιμότητα (θα δούμε παρακάτω γιατί).

Keep in mind...

Το OCCI δυστυχώς υποστηρίζεται μόνον από το Microsoft Visual Studio και τον Visual C++ Compiler, οπότε περιοριζόμαστε μόνο σε περιβάλλον Windows. Επίσης υποστηρίζεται μόνον από τις εκδόσεις 2005, 2008 και 2010 του Visual Studio.

7.2.3.1 - Το OCCI επί τω έργω

Θα δημιουργήσουμε λοιπόν μια εφαρμογή και παράλληλα θα εξηγήσουμε τα βήματα, ώστε να δούμε εν λειτουργία το OCCI και να κατανοήσουμε καλύτερα την λειτουργία του. Θα εργαστούμε σε Visual Studio 2010 σε λειτουργικό σύστημα Windows 7 Ultimate. Να σημειωθεί ότι ο πλήρης κώδικας της εφαρμογής, υπάρχει στο παράρτημα ΣΤ΄.

Δημιουργούμε ένα νέο Project και ρυθμίζουμε το Project μας για να εμπεριλάβει τις βιβλιοθήκες του OCCI (πλήρης οδηγίες απόκτησης και ρυθμίσεων θα βρείτε στο παράρτημα Ε΄). Προτού ξεκινήσουμε να γράφουμε τον κώδικα της εφαρμογής, ας δούμε τα περιεχόμενα του πίνακα στον οποίο θα δουλέψουμε:

⁴⁶ Oracle Documentation → Introduction to OCCI → Overview of OCCI – Site (http://docs.oracle.com/cd/B12037_01/appdev.101/b10778/introduction.htm)

ID	NAME
1	Gorgoipikoos
2	Aksion Esti
3	Theoskepasti
4	Gerasimos
5	Lady_Godiva

Περιεχόμενα πίνακα που θα χρησιμοποιήσουμε στην εφαρμογή

Εμείς λοιπόν θέλουμε να συνδεθούμε σε αυτήν την βάση μέσω C++ OCCI. Ξεκινάμε λοιπόν, πρώτα πρέπει να εισάγουμε τα απαραίτητα header files, αυτό θα γίνει με τις παρακάτω γραμμές κώδικα:

```
#include "occi.h" //Βιβλιοθήκη του OCCI
#include <iostream> //I/O Stream
#include <string> //Strings
#include <cstdlib> //C Standard Library ← όχι απαραίτητη
```

Θα χρειαστούμε επίσης τα namespaces std (cout, cerr κλπ) και oracle::occi

```
using namespace std;
using namespace oracle::occi;
```

Τώρα θα μπορούμε στη **main**. Αρχικά θα πρέπει να δημιουργήσουμε τα απαραίτητα αντικείμενα που θα μας επιτρέψουν να συνδεθούμε στην βάση μας. Επειδή ανά πάσα στιγμή μπορεί να προκληθεί SQL Exception, θα πρέπει ο κώδικάς μας να κλειστεί σε **try...catch** blocks.

```
int main(int argc, char* argv[])
{
    try{
        Environment* env = Environment::createEnvironment (Environment::DEFAULT); //1
        Connection* conn = env->createConnection ("cpp",
        "test123", "///192.168.1.69:1521/test"); //2
```

Αρχικά δημιουργήσαμε ένα αντικείμενο τύπου **Environment** το οποίο αντιπροσωπεύει το περιβάλλον στο οποίο θα συνδεθούμε. Δημιουργούμε λοιπόν ένα προεπιλεγμένο (default) περιβάλλον (1). Στις περισσότερες των περιπτώσεων, το προεπιλεγμένο περιβάλλον επαρκεί. Στην συνέχεια, δημιουργούμε ένα

αντικείμενο τύπου **Connection**, το οποίο αναλαμβάνει την σύνδεση με την βάση. Τα ορίσματα που του δίνουμε είναι α) το όνομα του χρήστη της βάσης (cpp), β) το password του χρήστη (test123) και τέλος το connection string το οποίο περιλαμβάνει την IP διεύθυνση του Server, την θύρα που «ακούει» (προεπιλογή 1521) και τέλος το SID (instance name) της βάσης που τρέχει στον server (test) (2). Η μορφοποίηση είναι //Server_IP_ADDRESS:SERVER_PORT/SID

Στην συνέχεια, θέλουμε να εκτελέσουμε ένα **επερώτημα SELECT** στην βάση δεδομένων μας και να ανακτήσουμε τα πεδία του πίνακα BOATS. Ο παρακάτω κώδικας κάνει ακριβώς αυτό:

```
Statement* st = conn->createStatement ("select * from boats "); //1
ResultSet *rset = st->executeQuery (); //2

cout << "SELECT * FROM boats \n\n" << endl;

while(rset->next()) //3
    cout << rset->getInt(1) << "\t" << rset->getString(2) << endl; //4
```

Πρώτα δημιουργούμε ένα αντικείμενο τύπου **Statement** το οποίο θα αποστέλλει το επερώτημα στην βάση. Δημιουργούμε λοιπόν ένα statement και του δίνουμε σαν όρισμα το SQL Select Statement, το οποίο θα επιστρέψει όλα τα πεδία του πίνακα (1). Στην συνέχεια δημιουργούμε ένα αντικείμενο **ResultSet** το οποίο θα «κρατήσει» τα αποτελέσματα της εκτέλεσης του statement. Για να τα λάβει, θα πρέπει να εκτελέσουμε το statement, πράγμα που το κάνουμε με την ιδιότητα executeQuery του statement (2). Ο πίνακας μας έχει περισσότερα του ενός πεδία, οπότε θα πρέπει να τα εμφανίσουμε όλα. Γι αυτό το λόγο ορίζουμε μια while, η οποία θα επαναλαμβάνεται όσο το result set βρίσκει νέες γραμμές του πίνακα (3). Τέλος εμφανίζουμε τα αποτελέσματα μας με μια cout (4). Παρατηρείστε πως τους Integers τους παίρνω με την ιδιότητα getInt και τα varchar με την ιδιότητα getString. Και οι δύο ιδιότητες (συν όλες οι άλλες για άλλους τύπους όπως πχ floats) δέχονται ως όρισμα, την αριθμητική τιμή της στήλης (column index).

Τώρα θα **εισάγουμε** μια **νέα γραμμή** στον πίνακα boats. Η γραμμή αυτή θα έχει ID = 6 και name = "Sofia". Ο κώδικας παρακάτω το πραγματοποιεί:

```
cout << "INSERT INTO BOATS(ID,NAME) VALUES (6, 'Sofia')\n\n" << endl;
string sqlStmt = "INSERT INTO BOATS(ID,NAME) VALUES (:x, :y)"; //1
st=conn->createStatement (sqlStmt); //2
st->setInt (1, 6); //3
st->setString (2, "Sofia"); //4
st->executeUpdate (); //5
cout << "Row Inserted Successfully" << endl;
```

Αρχικά δημιουργούμε ένα String που θα περιέχει το κείμενο του SQL Insert Statement. Οι τιμές :x, :y στα ορίσματα του VALUES είναι «υποδοχές» για τις τιμές που θα δώσουμε παρακάτω (1). Στην συνέχεια δημιουργούμε το statement προς εκτέλεση (2). Εν συνεχεία, ορίζουμε στην στήλη 1 να μπει η τιμή 6 (ID δηλαδή) (3) και στην στήλη 2 να μπει η τιμή Sofia (name δηλαδή) (4). Τέλος εκτελούμε το statement (5).

Τώρα θα πραγματοποιήσουμε **UPDATE** στον πίνακα, ώστε να αλλάξουμε το όνομα Gerasimos σε Ag.Nikolaos στην βάρκα με ID = 4. Ο παρακάτω κώδικας πραγματοποιεί το UPDATE.

```
cout << "\n\n UPDATE BOATS SET NAME = 'Ag.Nikolaos' WHERE ID = 4 \n\n" << endl;
string upd = "UPDATE BOATS SET name = :x WHERE id = :y"; //1
st = conn->createStatement (upd); //2
st->setString (1, "Ag.Nikolaos"); //3
st->setInt (2, 4); //4
st->executeUpdate (); //5
cout << "Update OK " << endl;
```

Πρώτα δημιουργούμε ένα String που θα έχει μέσα το SQL Update Statement. Βλέπουμε πως και αυτή η πρόταση έχει «υποδοχές» για τις τιμές που θέλουμε να δώσουμε (1). Δημιουργούμε το Statement (2) και στην συνέχεια δίνουμε την νέα τιμή που θέλουμε να εισάγουμε (3) και την τιμή στο WHERE CLAUSE του ID της βάρκας που θέλουμε να αλλάξουμε όνομα (4). Τέλος εκτελούμε το Statement (5).

Τέλος θέλουμε να **διαγράψουμε** την βάρκα Sofia από τον πίνακα μας. Ο παρακάτω κώδικας πραγματοποιεί το ζητούμενο:

```
cout << "\n\nDELETE FROM BOATS WHERE ID = 6 \n\n" << endl;
string del = "DELETE FROM boats WHERE id = 6"; //1
st = conn->createStatement (del); //2
st->executeUpdate (); //3
cout << "Row Deleted" << endl;
```

Δημιουργούμε ένα string με το κείμενο του SQL Delete Statement (1) και στην συνέχεια δημιουργούμε το statement (2). Τέλος το εκτελούμε (3).

Εφόσον δεν χρειαζόμαστε πλέον την βάση δεδομένων, μπορούμε να αποσυνδεθούμε από αυτήν.

```
env->terminateConnection(conn); //1
Environment::terminateEnvironment (env); //2
```

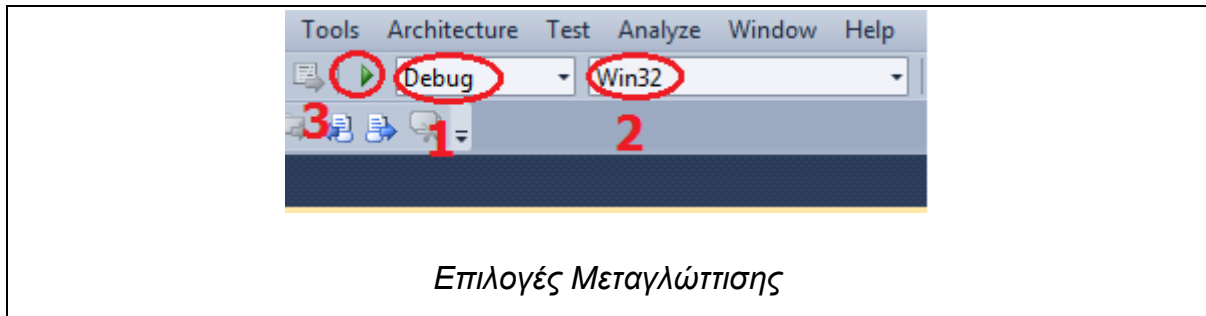
Με τις παραπάνω γραμμές πρώτα τερματίζουμε την σύνδεση με την βάση (1) και στην συνέχεια τερματίζουμε και το περιβάλλον εργασίας μας (2).

Τελειώνοντας, πρέπει να φτιάξουμε τα catch blocks που θα μας γλυτώσουν σε περίπτωση SQL Exception ή οποιασδήποτε μορφής άλλη εξαίρεση. Ο παρακάτω κώδικας πραγματοποιεί το ζητούμενο αυτό:

```
}//end try
catch (SQLException sq) //1
{
    cerr << "exception " << sq.getMessage() << endl; //2
}
catch (exception& ex) //3
{
    cerr << "Exception " << ex.what() << endl; //4
}
system("pause");
return 0; //terminate with exit 0
}//main
```

Η πρώτη catch πιάνει SQL Exceptions (1) όπως λανθασμένα credentials, μη – υπάρχον πίνακας, συντακτικό λάθος κλπ. Αν συμβεί εξαίρεση, τυπώνεται στο STDERR το είδος της. Η δεύτερη catch πιάνει γενικές εξαιρέσεις (3) και της εμφανίζει στο STDERR (4). Αν και δεν είναι απαραίτητη, είναι καλή πρακτική να υπάρχει για παν ενδεχόμενο. Τέλος η main επιστρέφει και το πρόγραμμα τερματίζεται.

Τώρα θα πρέπει να μεταγλωττίσουμε την εφαρμογή. Φροντίζουμε η μεταγλώττιση να γίνει σε **Debug Mode** (1) για έκδοση **Win32** (ή 64 αν όντως έχετε Windows 64bit) (2) και τότε πατούμε Start In Debug Mode (3). Θα μας ζητήσει Build, οπότε πατάμε *ναι*.



Αν όλα πάνε κατ' ευχή, τότε το αποτέλεσμα θα πρέπει να μοιάζει με το παρακάτω:



Είδαμε λοιπόν ότι είναι σχετικά εύκολο να δημιουργήσουμε μια εφαρμογή αλληλεπίδρασης με την Oracle Database. Αν εξαιρέσει κανείς την λίγο πολύπλοκη διαδικασία εγκατάστασής του στο Project, δεν παύει να είναι ένα πολύ χρήσιμο και απαραίτητο εργαλείο για έναν προγραμματιστή σε C++. Θυμίζουμε ότι τον κώδικα της εφαρμογής θα τον βρείτε στο παράρτημα ΣΤ'.

Ολοκληρώνοντας...

Και εδώ τελειώνουμε με την ενασχόληση των απλών χρηστών του συστήματος. Πλέον είμαι σίγουρος πως έχετε πάρει πολύ σημαντικές γνώσεις και εμπειρίες σχετικά από το πώς δουλεύει στο σύστημα για τους χρήστες. Αυτοί έχουν άλλωστε και τον τελευταίο λόγο. Και ειδικά στην περίπτωση μας που αφορά σπουδαστές, το μόνο σίγουρο είναι πως λίγοι (αν όχι κανένας) θα μείνουν αδιάφοροι από αυτό.

Εδώ τελειώνει η ενασχόλησή μας με το Σύστημα Oracle Database 11g. Βάση όλων όσων έχετε δει και (ίσως) δοκιμάσει, θεωρώ πως είστε έτοιμοι να εργαστείτε στο σύστημα από όποια και από τις δύο σκοπιές χρηστών ανήκετε. Ελπίζω να αποκτήσατε γνώσεις, εμπειρία, και σιγουριά, που ίσως αύριο-μεθαύριο σας φανούν χρήσιμα στην επαγγελματική σας πορεία.

Το επόμενο κεφάλαιο αφορά το πρόγραμμα Advanced Computer Science της Oracle Academy, σχετικά με μαθήματα προχωρημένων εφαρμογών πληροφορικής και απόκτησης ισχυρών πιστοποιήσεων.

Βιβλιογραφία Κεφαλαίου

A/A	Πηγή	Άρθρο/Ενότητα	Ιστοσελίδα
1	Wikipedia Article	Java Database Connectivity	http://en.wikipedia.org/wiki/Java_Database_Connectivity
2	Wikipedia Article	JDBC Driver	http://en.wikipedia.org/wiki/JDBC_driver
3	Oracle TechNet	JDBC FAQ - JDBC In General	http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-faq-090281.html#01_01
4	Oracle Document	Introduction to OCCI	http://docs.oracle.com/cd/B12037_01/appdev.101/b10778

	tation		/introduction.htm
5	Wikipedia Article	Oracle C++ Call	http://en.wikipedia.org/wiki/Oracle_C%2B%2B_Call_Interface
6	Wikipedia Article	Result Set	http://en.wikipedia.org/wiki/Result_set
7	Wikipedia Article	Prepared Statement	http://en.wikipedia.org/wiki/Prepared_statement

ΚΕΦΑΛΑΙΟ 8 – Η ORACLE ACADEMY ΚΑΙ ΤΟ ΠΡΟΓΡΑΜΜΑ ADVANCED COMPUTER SCIENCE

Η Oracle, που αδιαμφισβήτητα, είναι μία από τις πιο αναγνωρισμένες εταιρίες στον χώρο της πληροφορικής, γνωστή για τις πρωτοποριακές της υπηρεσίες και τεχνολογίες, κατάφερε αυτό που δύσκολα καταφέρνει μια εταιρία. Και ποιο είναι αυτό; Να κερδίζει συνεχώς όλο και περισσότερους υποστηρικτές, που εμπιστεύονται τα προϊόντα της. Έτσι λοιπόν, κατάφερε να γίνει ένας «κολοσσός», παγκοσμίου φήμης.

Οι ανάγκες των σημερινών επιχειρήσεων είναι αρκετά απαιτητικές και ως εκ τούτου χρειάζονται αξιόπιστες και αξιόλογες τεχνολογίες λογισμικού για να μπορέσουν να ικανοποιηθούν αποτελεσματικά. Και το εύρος τεχνολογιών λογισμικού της Oracle απέδειξε ότι μπορεί να ανταπεξέλθει στις ανάγκες αυτές, για αυτό το λόγο γίνεται ολοένα και πιο περιζήτητο.

Περιζήτητο όμως δεν είναι μόνο το εκάστοτε λογισμικό. Είναι και οι χρήστες που διαθέτουν δεξιότητες και επαρκείς γνώσεις για να εργαστούν σε αυτό. Πάρτε για παράδειγμα έναν καλό προγραμματιστή σε Java ή έναν εξειδικευμένο διαχειριστή βάσεων δεδομένων. Τέτοια άτομα, με τόσο ισχυρό γνωστικό υπόβαθρο, σίγουρα δεν θα έχουν μια εξασφαλισμένη σταδιοδρομία; Προφανώς και θα έχουν. Σε αυτό όμως συμβάλει και η σωστή, δομημένη και ορθά οργανωμένη εκπαίδευση και λίγοι θα διαφωνούσαν με αυτό. Η επιτυχία άλλωστε είναι συνυφασμένη με την ποιοτική εκπαίδευση.

Η Oracle λοιπόν, δεν μπορούσε να μείνει αμέτοχη σε όλο αυτό. Δημιούργησε μια πρωτοποριακή υπηρεσία, που επέτρεψε σε εκπαιδευτικά κέντρα, όπως Τεχνολογικά Εκπαιδευτικά Ιδρύματα, να παρέχουν εκπαιδευτική εξειδίκευση στους σπουδαστές τους, πάνω σε μεγάλο φάσμα του λογισμικού της. Η υπηρεσία αυτή, είναι η **Oracle Academy**, ένα εγχείρημα-θεσμός για τον αυριανό επιτυχημένο επαγγελματία στον χώρο της επιστήμης της πληροφορικής.

8.1 – Η Υπηρεσία Oracle Academy

Σε αυτήν την ενότητα θα γνωρίσουμε μαζί την πρωτοποριακή αυτή υπηρεσία, θα δούμε τα ιδιαίτερα χαρακτηριστικά της, την προσφορά της στο σπουδαστικό κοινό, και τι μπορεί κανείς να αποκομίσει από αυτήν.

8.1.1 – Τι είναι η Oracle Academy;

Η **Oracle Academy** είναι μια υπηρεσία της Oracle, που προσφέρει την δυνατότητα σε εκπαιδευτικούς φορείς-μέλη της, να παρέχουν στους σπουδαστές τους, εκπαίδευση πάνω σε διάφορες τεχνολογίες λογισμικού της, όπως βάσεις δεδομένων, προγραμματισμό εφαρμογών και λειτουργικά συστήματα. Οι σπουδαστές, μέσω μιας οργανωμένης εκπαιδευτικής διαδικασίας, εφαρμόζουν πρακτικά τα όσα διδάσκονται με την χρήση του εκάστοτε διδασκόμενου λογισμικού, αποκτώντας έτσι σημαντική εμπειρία, γνώσεις, αντίληψη και την ικανότητα να ανταπεξέρχονται σε κάθε είδους προκλήσεων που μπορεί να συναντήσουν. Όπως δηλώνει και το όνομά της, ο εκπαιδευτικός φορέας που είναι συμβεβλημένος με αυτήν, γίνεται μια τοπική ακαδημία της Oracle. Παρόμοιο εγχείρημα παρέχει η Cisco, με την υπηρεσία Cisco Networking Academy, διαφορετικής ωστόσο θεματολογίας και προσέγγισης.

Η προσφορά της όμως δεν σταματάει εδώ, καθώς δίνει την δυνατότητα στους σπουδαστές της, να πιστοποιήσουν τις γνώσεις που αποκτήσανε, λαμβάνοντας **διεθνώς αναγνωρισμένη πιστοποίηση** (certification). Η απόκτηση μιας τέτοιας πιστοποίησης, τους δίνει ένα πολύ ισχυρό πλεονέκτημα στην σημερινή αγορά εργασίας και θωρακίζει την σιγουριά τους για την επίτευξη μιας λαμπρής και εξασφαλισμένης καριέρας στον χώρο της επιστήμης της πληροφορικής.

Εκπαιδευτικοί φορείς που μπορούν να γίνουν τοπικές ακαδημίες της Oracle, μπορούν να είναι φορείς της δευτεροβάθμιας αλλά και της τριτοβάθμιας εκπαίδευσης, όπου η εκπαιδευτική ύλη διαμορφώνεται κατάλληλα για να

ανταποκριθεί στην αποδοτική εκμάθηση σε όλα τα καίρια concepts των επί μέρους τεχνολογιών, σε όλα τα επίπεδα.

8.1.2 – Τι παρέχει η εν λόγω υπηρεσία;

Η εκπαίδευση γίνεται με δομημένη **διδασκτέα ύλη** (curriculum) που παρέχεται από την ίδια την Oracle, μέσω του προγράμματος εκπαίδευσης Oracle University, και καλύπτει βήμα προς βήμα όλες τις απαραίτητες ενέργειες που πρέπει να πράξουν οι εκπαιδευόμενοι, προκειμένου να αποκτήσουν μια πλήρη γνώση του αντικείμενου που διδάσκονται την προκείμενη στιγμή. Μέσα από έναν σημαντικό αριθμό από οδηγούς χρήσης, ασκήσεων και παραδειγμάτων που ανταποκρίνονται στο διδασκόμενο αντικείμενο, ο σπουδαστής/σπουδάστρια προβληματίζεται, εκτιμά, καλλιεργεί την σκέψη του, αποκτά αντίληψη και χτίζει όλο και περισσότερο την σιγουριά του για το επόμενο βήμα.

Η εκπαιδευτική διαδικασία όμως δεν περιορίζεται μόνο σε θεωρητικές προσεγγίσεις αλλά και σε πρακτικές εφαρμογές των γνώσεων αυτών που αποκτώνται, στο πραγματικό περιβάλλον ή εφαρμογή που οι σπουδαστές εκπαιδεύονται. Το εύρος των εφαρμογών που διδάσκονται είναι μεγάλο, δίνοντας έτσι στο σπουδαστικό κοινό την δυνατότητα επιλογής **ειδικότητας** όπως ανάπτυξη λογισμικού, διαχείριση Server, middlewares, troubleshooting και άλλα παρεμφερή.

Υπεύθυνοι για την διεξαγωγή της εκπαιδευτικής διαδικασίας, είναι το **εκπαιδευτικό προσωπικό** (faculty), που βάσει της διδασκτέας ύλης που αποκτούν **δωρεάν** από την Oracle, πραγματοποιούν τα **μαθήματα** (courses). Η ύλη είναι συγκεκριμένη και καλύπτει πλήρως το ζητούμενο γνωστικό αντικείμενο χωρίς να αφήνει κενά και χωρίς να χρησιμοποιεί αφηρημένες/γενικευμένες έννοιες.

Τα μαθήματα αφορούν όπως έχουμε ξαναπεί, μεγάλο αριθμό από εφαρμογές και περιβάλλοντα. Οι σπουδαστές αλλά και το εκπαιδευτικό προσωπικό μπορούν να αποκτήσουν εμπορικές εκδόσεις των εφαρμογών της Oracle, **εντελώς δωρεάν**, με ακαδημαϊκή άδεια για εκπαιδευτική χρήση. Παρέχεται όμως και η δυνατότητα απόκτησης και εκπαίδευσης σε λογισμικό

Ανοιχτού Κώδικα (Open Source), όπως MySQL, Oracle Linux, Virtual box κλπ. Βλέπουμε πως οι δυνατότητες που παρέχονται είναι πάρα πολλές.

Φυσικά τα μαθήματα δεν περιορίζονται μόνο για τους σπουδαστές, καθώς μπορεί και το εκπαιδευτικό προσωπικό να επωφεληθεί από αυτά και να αποκτήσει πρόσθετες και ποιο σύγχρονες γνώσεις που μπορεί μελλοντικά να μεταλαμπαδεύσει στο μαθητικό/φοιτητικό κοινό του.

Τέλος, η Oracle Academy, προσφέρει στα μέλη της **έκπτωση** στους οδηγούς προετοιμασίας εξετάσεων για τις πιστοποιήσεις της, καθώς επίσης και στα εξεταστικά παράβολα. Επίσης, έκπτωση δικαιούνται και για την απόκτηση εκπαιδευτικού υλικού από την Oracle University, και για βιβλία της Oracle Press.

8.1.3 – Σύμβαση φορέα και ρόλοι συμμετεχόντων

Ο εκπαιδευτικός φορέας που επιθυμεί να γίνει τοπική ακαδημία της Oracle, θα πρέπει να πληρώσει ένα ποσό ανάλογο με το πρόγραμμα εκπαίδευσης που σκοπεύει να εντάξει στις εκπαιδευτικούς πόρους του. Με την ενεργοποίηση της σύμβασης, ο εκπαιδευτικός φορέας γίνεται επίσημα τοπική ακαδημία της Oracle και μπορεί να διεξάγει μαθήματα, στο επιλεγμένο πρόγραμμα που επέλεξε.

Υπάρχουν δύο διακριτοί ρόλοι στην τοπική ακαδημία Oracle, το **εκπαιδευτικό προσωπικό** (faculty) και οι **σπουδαστές** (students), κάθε ένας από τους οποίους, δημιουργούν ένα λογαριασμό στο Oracle Academy, υπό τον φορέα που είναι συμβεβλημένος με την υπηρεσία, ανάλογα με τον ρόλο τον οποίο έχουν στην τοπική ακαδημία.

8.1.3.1 – Εκπαιδευτικό προσωπικό

Το εκπαιδευτικό προσωπικό δημιουργεί και διατηρεί λογαριασμό τύπου «faculty» στον ιστότοπο της Oracle Academy. Μέσω αυτού του λογαριασμού, το εκπαιδευτικό προσωπικό μπορεί:

- 1) Να κατεβάζει δωρεάν τα curriculums που καλύπτουν την θεματολογία που αναλαμβάνει να διδάξει στους σπουδαστές.
- 2) Να παραγγέλνει εξετάσεις πιστοποίησης, με εκπτωτικό κουπόνι (voucher) είτε για δική του εξέταση, είτε για τους σπουδαστές της ακαδημίας που διδάσκει, μέσω της Pearson VUE.
- 3) Να παραγγέλνει λογισμικό προετοιμασίας εξετάσεων πιστοποίησης, από την KAPLAN SELFTEST, με έκπτωση.
- 4) Να παραγγέλνει βιβλία της Oracle Press, με έκπτωση.
- 5) Να προσπελάσει την σελίδα «Faculty Lounge», όπου μπορεί να έχει πρόσβαση στην δωρεάν βιβλιοθήκη της Oracle «Oracle Learning Library» για χρήσιμα tutorials, να ενημερώνεται για ενδιαφέροντα events, να βρίσκει άλλα συμβεβλημένα ιδρύματα, και να έρθει σε επαφή με άλλους χρήστες (πχ Java Developers)
- 6) Να κατεβάζει εμπορικό λογισμικό από την Oracle Software Cloud υπηρεσία, και να τα εγκαθιστά στο εκπαιδευτικό περιβάλλον που θα διεξαχθεί το μάθημα, δωρεάν φυσικά, στις πρόσφατες εκδόσεις τους.
- 7) Να κατεβάζει δωρεάν λογισμικό ανοιχτού κώδικα (MySQL, Virtual box κλπ).

8.1.3.2 - Σπουδαστές

Οι σπουδαστές της τοπικής ακαδημίας δημιουργούν λογαριασμό τύπου «student» στον ιστότοπο του Oracle Academy. Με αυτόν τον λογαριασμό μπορούν:

- 1) Να συμμετέχουν στα μαθήματα που προσφέρει ο εκπαιδευτικός φορέας στον οποίο φοιτούν.
- 2) Να συμμετέχουν στις εξετάσεις πιστοποίησης που προγραμμάτισε ο διδάσκοντας της τάξης τους, για λογαριασμό τους.
- 3) Να παραγγείλουν λογισμικό προετοιμασίας εξετάσεων πιστοποίησης, από την KAPLAN SELFTEST, με έκπτωση.
- 8) Να κατεβάζουν δωρεάν λογισμικό από την Oracle Software Cloud υπηρεσία, για εγκατάσταση στον προσωπικό τους υπολογιστή, καθώς και δωρεάν λογισμικό ανοιχτού κώδικα.
- 9) Να έχουν πρόσβαση στην σελίδα «Students Center» όπου μπορούν να έρθουν σε επαφή με άλλους χρήστες, να βρίσκουν εξεταστικά κέντρα στην περιοχή τους και σχετικές πληροφορίες για την επαγγελματική τους αποκατάσταση (πρακτικές ασκήσεις κλπ).

8.1.4 – Εκπαιδευτικά προγράμματα

Η Oracle Academy παρέχει εκπαιδευτική ύλη για χρήση των διαφόρων τεχνολογιών της, μέσα από τρία διαφορετικά εκπαιδευτικά προγράμματα. Κάθε πρόγραμμα ανταποκρίνεται σε συγκεκριμένο γνωστικό αντικείμενο και σε συγκεκριμένο σπουδαστικό κοινό. Παρακάτω θα δούμε αυτά τα προγράμματα για να κατανοήσουμε την υπόστασή τους.

8.1.4.1 – Introduction to Computer Science

Το εκπαιδευτικό πρόγραμμα αυτό απευθύνεται για μαθητές της δευτεροβάθμιας εκπαίδευσης (secondary schools) και αποσκοπεί στην παροχή εισαγωγικών γνώσεων σε concepts προγραμματισμού και εξοικείωσης με τις

βάσεις δεδομένων. Είναι το μόνο πρόγραμμα της Oracle Academy που παρέχει **δωρεάν σύμβαση** του εκπαιδευτικού φορέα με αυτήν.

Ποιο συγκεκριμένα, παρέχει εκπαιδευτική ύλη που επικεντρώνεται σε θεμελιώδεις αρχές προγραμματισμού με την γλώσσα Java, σχεδιασμό βάσεων δεδομένων και εξοικείωση με την SQL και τέλος προγραμματισμό με την γλώσσα PL/SQL. Κάθε μία από τις παραπάνω κατηγορίες παρέχει και την αντίστοιχη πιστοποίηση που μπορεί να λάβει ο σπουδαστής.

Η πρόσβαση των σπουδαστών στο λογισμικό της Oracle επικεντρώνεται στις δωρεάν εφαρμογές συγγραφής κώδικα Java που είναι οι εφαρμογές Alice, Greenfoot, BlueJ, Eclipse και NetBeans IDE, ενώ για την κάλυψη των αναγκών εκμάθησης και εργασίας στις βάσεις δεδομένων, υπάρχει εγκατεστημένο στο εκπαιδευτικό περιβάλλον το Oracle Application Express (APEX).

8.1.4.2 – Advanced Computer Science

Το πρόγραμμα αυτό απευθύνεται σε εκπαιδευτικούς φορείς της ανωτέρας εκπαίδευσης όπως μεταδευτεροβάθμια ιδρύματα (πχ ΙΕΚ) και τριτοβάθμια ιδρύματα (πχ ΤΕΙ) και επικεντρώνεται σε ποιο συγκεκριμένα concepts. Το πρόγραμμα αυτό θα μας απασχολήσει ιδιαίτερα στην ενότητα 8.2 όπου θα το δούμε με περισσότερη λεπτομέρεια, ωστόσο θα κάνουμε μια σύντομη περιγραφή.

Οι σπουδαστές του εν λόγω προγράμματος μπορούν να ειδικευθούν σε προχωρημένα concepts της Java SE 7, στην διαχείριση βάσεων δεδομένων της Oracle και στην εξειδίκευση συγγραφής προγραμμάτων με την PL/SQL, Middlewares και άλλα που θα τα δούμε μαζί στην ενότητα 8.2.

8.1.4.3 – Business Applications

Όπως και το πρόγραμμα Advanced Computer Science, έτσι και το πρόγραμμα Business Applications, απευθύνεται στους εκπαιδευτικούς φορείς

ανωτέρας εκπαίδευσης. Η διαφορά του όμως είναι ότι παρόλο που γίνεται εκμάθηση λογισμικού και τεχνολογιών της Oracle, το πρόγραμμα αυτό δεν απευθύνεται σε σπουδαστές της επιστήμης της πληροφορικής. Αντιθέτως επικεντρώνεται σε άλλες ειδικότητες, όπως Οικονομικές επιστήμες, βιολογικές επιστήμες, αεροναυπηγική, επικοινωνίες, δομικών έργων και άλλες. Το φάσμα λογισμικού που καλύπτει η διδακτέα ύλη, ανά ειδικότητα κοστίζει περίπου 369 € ετησίως. Πιστοποιήσεις δεν εκδίδονται.

8.1.5 – Οφέλη της Oracle Academy

Τα οφέλη που προσφέρει αυτό το εγχείρημα της Oracle, πολύ φοβάμαι ότι θα χρειαστούμε πάρα πολλές σελίδες για να αριθμήσουμε. Μπορούμε ωστόσο να κάνουμε μια προσπάθεια να εκθειάσουμε τα περισσότερα σημαντικά και προφανή. Ας ξεκινήσουμε.

- **Ισχυρό γνωστικό υπόβαθρο.** Ο σπουδαστής/σπουδάστρια, μπορεί να αποκτήσει σημαντικές και χρήσιμες γνώσεις πάνω σε σύγχρονα συστήματα που λειτουργούν στο μέσο σημερινό εργασιακό περιβάλλον.
- **Προσιτή εκπαιδευτική διαδικασία.** Η ύλη είναι εξαιρετικά δομημένη και σχεδιάστηκε έτσι ώστε να εγείρει το ενδιαφέρον του σπουδαστικού κοινού.
- **Ολοκληρωμένη εκπαίδευση.** Η ύλη ξεκινάει από εισαγωγικά concepts και προοδευτικά μεταβαίνει σε περισσότερο ειδικά θέματα, καλύπτοντας έτσι κάθε μαθησιακό κενό.
- **Ενίσχυση γνώσεων διδασκόντων.** Οι διδάσκοντες που ανήκουν στο εκπαιδευτικό προσωπικό, μπορούν να ενισχύσουν τις γνώσεις του από την παρεχόμενη διδακτέα ύλη, και να είναι σε θέση να την μεταδώσουν σε μεταγενέστερο σπουδαστικό κοινό.

- **Δωρεάν απολαβές.** Αν εξαιρεθεί φυσικά η ετήσια συνδρομή, οι σπουδαστές και οι διδάσκοντες έχουν δωρεάν πρόσβαση σε λογισμικό, διδακτέα ύλη και tutorials.
- **Εκπτώσεις.** Τα μέλη της ακαδημίας δικαιούνται έκπτωση στην εξέταση πιστοποίησης, και στην απόκτηση βιβλίων ή παρακολούθησης σεμιναρίων από την Oracle University. Ειδικότερα στις μέρες που διανύουμε, η μείωση του κόστους για τα παραπάνω, είναι σημαντικό όφελος.
- **Περισσότερες ευκαιρίες καριέρας.** Η απόκτηση μιας πιστοποίησης από μια εταιρία κύρους όπως η Oracle, αυξάνει τις πιθανότητες ο σπουδαστής/σπουδάστρια, να βρει ανοιχτές πόρτες στην αγορά εργασίας του σήμερα.
- **Απόκτηση εμπειρίας.** Ο εκπαιδευόμενος εργάζεται σε πραγματική εφαρμογή ή περιβάλλον, και εξοικειώνεται σταδιακά με αυτό.
- **Ισχυροποίηση σιγουριάς.** Ο εκπαιδευόμενος, αποκτά αντίληψη και σιγουριά, σημαντικά όπλα για να αντιμετωπίσει οποιαδήποτε πρόκληση κληθεί να αντιμετωπίσει.

Και πάλι τα όσα έχουν γραφτεί, δεν φτάνουν για να περιγράψουν την σπουδαιότητα αυτού του εγχειρήματος. Το βέβαιο όμως είναι πως έχει πολλά να προσφέρει και αξίζει πραγματικά να ασχοληθεί κανείς με αυτό. Σίγουρα μόνο κερδισμένος έχει να βγει.

8.1.6 – Η Oracle Academy στον Ελλαδικό χώρο

Οι τοπικές ακαδημίες της Oracle, ενσωματώνονται ραγδαία σε όλο και περισσότερα ΤΕΙ της χώρας μας. Σύμφωνα με πληροφορίες, οι Έλληνες σπουδαστές που έχουν την τύχη να ενταχθούν και να παρακολουθήσουν μαθήματα μέσω αυτού του εγχειρήματος, ανέρχονται στους πάνω από 11.500, νούμερο διόλου ευκαταφρόνητο. Αυτό μας οδηγεί στο συμπέρασμα ότι η Oracle

Academy θα κατακτήσει τον χώρο της επιστήμης της πληροφορικής στην τριτοβάθμια εκπαίδευση, και θα δώσει την ευκαιρία στα εκπαιδευτικά ιδρύματα της χώρας να βελτιστοποιήσουν τους εκπαιδευτικούς τους πόρους, παρέχοντας περισσότερο ποιοτικό μάθημα σε τεχνολογίες του σήμερα. Τοπικές ακαδημίες επίσης έχουν δημιουργηθεί όμως και σε γυμνάσια και ΕΠΑΛ της χώρας μας, ακόμη και σε δημοτικά σχολεία!

Παρακάτω παραθέτουμε τυχαία, μερικά από τα συμβεβλημένα με την Oracle Academy εκπαιδευτικά ιδρύματα και εκπαιδευτήρια:

Εκπαιδευτικός φορέας	Τμήμα	Πρόγραμμα Oracle Academy
ΑΤΕΙ Θεσσαλονίκης	Μηχανικών Πληροφορικής	Advanced Computer Science
Πανεπιστήμιο Μακεδονία	Εφαρμοσμένης Πληροφορικής	Advanced Computer Science
ΤΕΙ Λάρισας	Πληροφορικής και Επικοινωνιών	Advanced Computer Science
ΤΕΙ Χαλκίδας	Λογιστικής	Advanced Computer Science
ΤΕΙ Πειραιά	Αυτοματισμού	Advanced Computer Science
1ο Δημοτικό Σχολείο Ραφήνας	-	Introduction to Computer Science
1ο Δημοτικό Σχολείο Μυκόνου	-	Introduction to Computer Science
Γυμνάσιο Σταυρούπολης	-	Introduction to Computer Science
ΕΠΑΛ Ευόσμου	-	Introduction to Computer Science
2ο ΓΕΛ Βριλησίων	-	Introduction to Computer Science

Είναι πράγματι μεγάλος ο αριθμός των τοπικών ακαδημιών Oracle στην χώρα μας. Αυτό μεταφράζεται σε εκσυγχρονισμό της χώρας μας όσον αφορά την εκπαίδευση στην επιστήμη των υπολογιστών και στην παραγωγή περισσότερων δεξιοτεχνών στον χώρο με ένα βήμα μπροστά σε ευκαιρίες μιας λαμπρής σταδιοδρομίας.

8.2 – Το πρόγραμμα **Advanced Computer Science – A Closer Look**

Το πρόγραμμα αυτό, όπως είχαμε πει και πριν απευθύνεται σε ανώτερα ινστιτούτα και ιδρύματα, και ως εκ τούτου το γνωστικό αντικείμενο που παρέχει, είναι περισσότερο εστιασμένο σε συγκεκριμένες τεχνολογίες και concepts. Σε αυτήν την ενότητα λοιπόν θα το δούμε με περισσότερη λεπτομέρεια.

8.2.1 – Χαρακτηριστικά

Το πρόγραμμα αυτό κοστίζει περίπου 369 € για ετήσια σύμβαση με την Oracle Academy, τιμή αρκετά προσιτή για το εύρος των τεχνολογιών που καλύπτει (και προσφέρει). Προσφέρει μεγάλο αριθμό διδακτέας ύλης που κάθε μία εστιάζεται σε συγκεκριμένη εφαρμογή ή περιβάλλον.

Η δομή της διδακτέας ύλης, αποτελείται από τις παρακάτω μορφές:

- 1) **Course Labs**. Περιέχει τον εκτελέσιμο κώδικα των scripts που υπάρχουν στο Manual και αρχεία SWF με flash animated επίδειξη χαρακτηριστικών.
- 2) **Presentation Slides** - Περιέχει αρχεία του Power Point, με το κείμενο της διδακτέας ύλης, χωρισμένα ανά μάθημα.
- 3) **Setup Document**. Περιέχει αρχείο οδηγιών για την εγκατάσταση του απαραίτητου εκπαιδευτικού περιβάλλοντος για την διεξαγωγή του μαθήματος (εγκατάσταση, παραμετροποίηση)

- 4) **Student Manual.** Περιέχει συγκεντρωμένο όλο το κείμενο της διδακτέας ύλης είτε σε ένα αρχείο είτε σε περισσότερα (συνήθως σε μορφή PDF).

8.2.2 – Μαθήματα που παρέχονται

Σε αυτήν την ενότητα θα δούμε όλα τα μαθήματα που παρέχονται από το πρόγραμμα Advanced Computer Science. Τα παραθέτουμε με την σειρά στον πίνακα που ακολουθεί, μαζί με τις πιστοποιήσεις που παρέχουν (όσα παρέχουν):

A/A	Λογισμικό	Τίτλος Μαθήματος	Πιστοποίηση
1	Enterprise JavaBeans Java Enterprise Edition	Business Component Development with EJB Technology, Java EE 5	-
2	Java Enterprise Edition	Developing Applications for the Java EE Platform	-
3	Java Standard Edition	Developing Applications With the Java SE 6 Platform	-
4	Java Enterprise Edition	Developing Architectures for Enterprise Java Applications	-
6	JavaServer Pages (JSP) + Ajax	Developing JavaServer Faces Components with Ajax	-
7	Java Standard Edition	Fundamentals of the Java Programming Language, Java SE 6	-
8	Fusion Middleware 11g	Build ADF Faces Clients for EJB and JPA	-
9	Hyperion Interactive	Design Advanced Reports	-

	Reporting 11.1.2		
10	Hyperion Interactive Reporting 11.1.2	Design Reports & Dashboards	-
11	Hyperion Web Analysis 11.1.2	Design Reports	-
12	Oracle Database	Implementing Oracle Audit Vault	-
13	Oracle NoSQL	Introduction to Oracle NoSQL Database	-
14	Java	Java Design Patterns	-
15	Java Standard Edition	Java Programming Language, Java SE 6	-
16	Java Standard Edition 7	Java SE 7 Fundamentals	Oracle Certified Associate, Java SE 7 Programmer
17	Java Standard Edition 7	Java SE 7 New Features	-
18	Java Standard Edition 7	Java SE 7 Programming	Oracle Certified Professional, Java SE 7 Programmer (προαπαιτούμενη η πιστοποίηση του 16)
19	Java Standard Edition 7	Develop Rich Client Applications	-
20	MySQL	MySQL for Beginners	-
21	MySQL	MySQL for Database Administrators	-
22	MySQL	MySQL for Developers English Student Subscription	-
23		Object-Oriented Analysis and Design Using UML	-
24	Oracle Database 10g	Build J2EE Applications	-

		English	
25	Oracle Database 10g	Java Programming (για διδάσκοντες)	-
26	Oracle Database 10g	Oracle 10g: Java Programming	-
27	Oracle Business Intelligence 11g R1	Build Repositories	-
28	Oracle Business Intelligence 11g R1	Create Analyses and Dashboards	-
29	Oracle Business Intelligence Publisher 11g R1	Fundamentals	-
30	Oracle Business Process Management 11g	Oracle BPM 11g Modeling	-
31	Oracle Business Process Management 11g	Implement the Process Model	-
32	Oracle Coherence 3.6	Administer and Troubleshoot Clusters	-
33	Oracle Coherence 3.6	Share and Manage Data In Clusters	-
34	Oracle Data Integrator 11g	Integration and Administration	-
35		Oracle Data Modeling and Relational Database Design	-
36	Oracle Database 10g	2 Day DBA	-
37	Oracle Database 10g	Administration Workshop I English Student Subscription	-
38	Oracle Database 10g	Administration Workshop II	-
39	Oracle Database 10g	Advanced PL/SQL	-

40	Oracle Database 10g	Implement and Administer a Data Warehouse	-
41	Oracle Database 10g	Introduction to SQL	-
42	Oracle Database 10g	Performance Tuning	-
43	Oracle Database 10g	Program with PL/SQL	-
44	Oracle Database 10g	SQL Tuning Workshop	-
45	Oracle Database 11g	Use XML DB English	-
46	Oracle Database 11g	2 Day DBA	-
47	Oracle Database 11g	Administer a Data Warehouse	-
48	Oracle Database 11g	Administration Workshop I Release 2	Oracle Database 11g Administrator Certified Associate (πρέπει να περαστεί το 67)
49	Oracle Database 11g	Administration Workshop II DBA Release 2	Oracle Database 11g Administrator Certified Professional (προαπαιτούμενη η πιστοποίηση από τα 48 + 67)
50	Oracle Database 11g	Advanced PL/SQL	Oracle Advanced PL/SQL Developer Certified Professional
51	Oracle Database 11g	Backup and Recovery Workshop	-
52	Oracle Database 11g	Data Guard Administration Release 2	-
53	Oracle Database 11g	Data Mining Techniques	-
54	Oracle Database 11g	Data Warehousing Fundamentals	-
55	Oracle Database 11g	Implement Database Vault	-
56	Oracle Database 11g	Implement Partitioning	-

		Release 2	
57	Oracle Database 11g	Implement Streams Release 2	-
58	Oracle Database 11g	Introduction to SQL	-
59	Oracle Database 11g + Oracle Linux	Managing Oracle on Linux for DBAs	-
60	Oracle Database 11g	New Features for Administrators	-
61	Oracle Database 11g	OLAP Essentials	-
62	Oracle Database 11g	Performance Tuning DBA Release 2	-
63	Oracle Database 11g	RAC Administration	-
64	Oracle Database 11g	Security Release 2	-
65	Oracle Database 11g	SQL and PL/SQL	-
66	Oracle Database 11g	SQL Tuning Workshop Release 2	-
67	Oracle Database	Introduction to SQL	Oracle Database 11g Administrator Certified Associate (αν περαστεί και το 48)
68	Oracle Database	Program with PL/SQL	Oracle Database PL/SQL Developer Certified Associate (αν έχει περαστεί το 69)
69	Oracle Database	SQL Fundamentals I	Oracle Database PL/SQL Developer Certified Associate (αν περαστεί και το 68)
70	Oracle Database	SQL Fundamentals II	-
71	Oracle Enterprise	Grid Control	-

	Manager 10g		
72	Oracle Enterprise Manager 11g	Grid Control Essentials	-
73	Oracle Essbase 11.1.2	Bootcamp	-
74	Oracle Essbase 11.1.2	Oracle Essbase 11.1.2 for System Administrators	-
75	Oracle Essbase 11.1.2	Calculate Block Storage Databases	-
76	Oracle Essbase 11.1.2	Deploy Aggregate Storage Databases	-
77	Oracle Essbase Studio 11.1.2	Create and Manage Data Structures	-
78	Oracle Fusion Middleware 11g + Application Development Framework	Build Applications with ADF I	-
79	Oracle Fusion Middleware 11g + Application Development Framework	Build Applications with ADF II	-
80	Oracle Fusion Middleware 11g	Virtual Assembly Builder	-
81	Oracle Grid Infrastructure 11g	Manage Clusterware and ASM	-
82	Oracle Hyperion Financial Reporting + Oracle Essbase	Oracle Hyperion Financial Reporting 11.1.2 for Essbase & Planning	-
83	Oracle Hyperion Financial Reporting 11.1.2	Oracle Hyperion Financial Reporting 11.1.2 for Financial Management	-

84	Oracle Hyperion Smart View 11.1.2 + Essbase	Oracle Hyperion Smart View 11.1.2 for Essbase	-
85	Oracle Identity Manager 11g	Develop Identity Provisioning	-
86	Oracle Identity Manager 11g	Essentials	-
87	Oracle Service Oriented Architecture Suite 10g	SOA Essentials	-
88	Oracle Service Oriented Architecture Suite 11g	Administration	-
89	Oracle Service Oriented Architecture Suite 11g	Build Composite Applications	-
90	Oracle Service Oriented Architecture Suite 11g	Essential Concepts	-
91	Oracle Spatial	Advanced	-
92	Oracle Spatial	Essentials	-
93	Oracle Warehouse Builder 10g	Implementation Part 1	-
94	Oracle Warehouse Builder 10g	Implementation Part II	-
95	Oracle WebCenter Content 11g	Content Server Customization	-
96	Oracle WebCenter Content 11g	Site Studio Advanced	-
97	Oracle WebCenter Content 11g	Site Studio Essentials	-
98	Oracle WebCenter Portal 11g	Build Portals With Spaces	-

99	Oracle WebCenter Portal 11g	Spaces Administration	-
100	Oracle WebCenter	Oracle WebCenter Sites for Developers	-
101	Oracle WebCenter	Oracle WebCenter Sites for System Administrators	-
102	Oracle WebCenter	Architecture	-
103	Oracle WebLogic Server 10g R3	Advanced System Administration	-
104	Oracle WebLogic Server 10g R3	System Administration	-
105	Oracle WebLogic Server 11g	Administration Essentials	-
106	Oracle WebLogic Server 11g	Advanced Administration	-
107	Oracle WebLogic Server 11g	Diagnostics and Troubleshooting	-
108	Oracle WebLogic Server 11g	Monitor and Tune Performance	-
109	Oracle9i Forms Developer	Build Internet Applications	-
110	Java + Ajax + JavaScript	The JavaScript Language and Ajax for Java Developers	-
111	Service Oriented Architecture	Architectural Concepts and Design Principles	-
112	Java	Using Java for PL/SQL and Database Developers	-
113	Java + JSP + Java Enterprise Edition	Web Component Development with Servlets & JSPs, Java EE 5	-

8.2.3 – Εμπορικό Λογισμικό

Τα μέλη της τοπικής ακαδημίας δικαιούνται δωρεάν λήψη του εμπορικού λογισμικού (κλειστού κώδικα) στην Oracle, μέσα από το Software Delivery Cloud, με ακαδημαϊκή άδεια για εκπαιδευτική χρήση. Στον παρακάτω πίνακα, βλέπουμε όλη την λίστα με το διαθέσιμο για απόκτηση εμπορικό λογισμικό:

Όνομα λογισμικού	Είδος
Database Enterprise Edition	Βάσεις Δεδομένων
Real Application Clusters	Βάσεις Δεδομένων
Spatial and Graph	Βάσεις Δεδομένων
Advanced Analytics	Βάσεις Δεδομένων
NoSQL	Βάσεις Δεδομένων
Advanced Analytics	Βάσεις Δεδομένων
Diagnostics Pack	Βάσεις δεδομένων
Tuning Pack	Βάσεις δεδομένων
WebLogic Suite	Middleware
Data Integrator Enterprise Edition	Middleware
SOA Suite for Oracle Middleware	Middleware
WebCenter Suite Plus	Middleware
Business Intelligence Suite Enterprise Edition Plus	Middleware
Essbase Plus	Middleware
Identity Manager	Middleware
Unified Business Process Management Suite	Middleware
TopLink	Middleware
Application Development Framework	Middleware

8.2.4 – Δωρεάν και ανοιχτού κώδικα λογισμικό

Τα μέλη επίσης μπορούν να κατεβάσουν ελευθέρως και μια μεγάλη γκάμα από δωρεάν εφαρμογές και από εφαρμογές ανοιχτού λογισμικού. Παρακάτω δίνεται ο πίνακας που περιέχει τις εφαρμογές αυτές.

Όνομα Λογισμικού	Είδος
Oracle Linux	Λειτουργικό σύστημα
MySQL	Βάσεις Δεδομένων
Virtual box	Computer Virtualization
Glassfish	Java Enterprise Edition Server
Alice	Java IDE
Greenfoot	Java IDE
BlueJ	Java IDE
Eclipse	Java IDE
NetBeans IDE	Java IDE

Ολοκληρώνοντας...

Είδαμε λοιπόν το εγχείρημα της Oracle Academy, παρέχει τεράστια οφέλη στα εκπαιδευόμενα μέλη της, καθώς τους δίνει την δυνατότητα να εξειδικευτούν και να αποκτήσουν δεξιότητες που θα τους κάνουν περιζήτητους στην αγορά εργασίας. Είδαμε την Oracle Academy «εκ τω έσω» και έτσι μπορέσαμε να κατανοήσουμε καλύτερα την οργανωτική δομή της και να καταλήξουμε στο συμπέρασμα ότι είναι ένα αξιόπαινο εγχείρημα από την Oracle. Επεκταθήκαμε όμως και με περισσότερη λεπτομέρεια στο πρόγραμμα Advanced Computer Science, όπου είδαμε μαζί την τεράστια γκάμα από μαθήματα σε εξειδικευμένο λογισμικό που παρέχονται και την συλλογή λογισμικού που περιλαμβάνει προς διάθεση. Συμπέρασμα; Απλό. Η Oracle εφόσον μπορεί και δημιουργεί τεχνολογίες που γίνονται γρήγορα ανάρπαστες από το κοινό της, τότε με την Oracle Academy μπορεί να δημιουργήσει περιζήτητους επιστήμονες. Μία από τις πιο μεγάλες

ευκαιρίες ενός σπουδαστή/σπουδάστριας που ονειρεύεται ένα λαμπρό μέλλον στο χώρο της επιστήμης των υπολογιστών. Και πάνω από όλα, χωρίς κανένα κόστος.

Εφόσον ασχοληθήκαμε με την Oracle Academy, στο επόμενο και τελευταίο κεφάλαιο, θα ασχοληθούμε με τον εξυπηρετητή του Τμήματος Μηχανικών Πληροφορικής του ΑΤΕΙ Θεσσαλονίκης, όπου μία από τις υπηρεσίες του, είναι μια διαδικτυακή πύλη για την Oracle Academy, καθώς το τμήμα είναι κι αυτό, τοπική ακαδημία της Oracle.

Βιβλιογραφία Κεφαλαίου

A/A	Πηγή	Άρθρο/Ενότητα	Ιστοσελίδα
1	Oracle Academy	Introduction to computer science	https://academy.oracle.com/oa-web-introcs-description.html
2	Oracle Academy	Advanced Computer Science	https://academy.oracle.com/oa-web-advancedcs-description.html
3	Oracle Academy	Business Applications	https://academy.oracle.com/oa-web-busapps-description.html
4	Oracle Academy	Program Guide	https://oai.oracle.com/en/program_guide.html
5	Oracle Academy	Program Guide → Curriculum	https://oai.oracle.com/en/program_guide.html#curriculum
6	Oracle Academy	Software	https://oai.oracle.com/en/software.html
7	Oracle Academy	Advanced Computer Science – Courseware	https://academy.oracle.com/oa-web-advancedcs-curriculum.html
8	Oracle University	Oracle Certification → Oracle PL/SQL Developer	http://education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=458&get_params=p_track_id:PLSQL
9	Oracle	Oracle Certification →	http://education.oracle.com/pls/w

	University	Java Programmer Certification	eb_prod-plq-dad/db_pages.getpage?page_id=458&get_params=p_track_id:JS_E7Prog
10	ΤΕΙ Λάρισας – Μηχανικών πληροφορικής	Oracle Academy	http://www.cs.teilar.gr/CS/oracle.jsp
11	Tech News	Επέκτασης προγράμματος Oracle Academy	http://greektech-news.blogspot.gr/2012/05/oracle-academy.html
12	Oracle Press Release	Ταχεία Υιοθέτηση του Oracle Academy από τα Ελληνικά Εκπαιδευτικά Ιδρύματα και Προετοιμασία των Φοιτητών για Σταδιοδρομία στο IT	https://emeapressoffice.oracle.com/Content/Detail.aspx?ReleaseID=6305&NewsAreaID=2&ClientID=8

ΚΕΦΑΛΑΙΟ 9 – Ο SERVER ΤΟΥ ΤΜΗΜΑΤΟΣ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΟΥ ΑΤΕΙ ΘΕΣΣΑΛΟΝΙΚΗΣ

Η πλήρης και αποδοτική πλατφόρμα της Oracle που αναλύσαμε διεξοδικά στο παρών σύγγραμμα έχει εγκατασταθεί σε εξυπηρετητή (Server) στο εσωτερικό του Τμήματος Μηχανικών Πληροφορικής του ΑΤΕΙ Θεσσαλονίκης. Ο εν λόγω Server αποσκοπείται στο να χρησιμοποιηθεί ως εκπαιδευτική πλατφόρμα τόσο στους προπτυχιακούς, τόσο στους μεταπτυχιακούς φοιτητές για μαθήματα σχετικά με τις Βάσεις Δεδομένων, με την χρήση της Oracle. Δεν περιορίζεται όμως εκεί, καθώς αποτελεί χώρο φιλοξενίας του Portal για το πρόγραμμα Advanced Computer Science του Oracle Academy.

6.1 – Περιγραφή του Server

Στην ενότητα αυτή θα περιγράψουμε τα χαρακτηριστικά του Server του τμήματος παρέχοντας χρήσιμες πληροφορίες γι αυτόν. Η ενότητα αυτή είναι καθαρά πληροφοριακής χρήσης.

6.1.1 – Διεύθυνση και πρόσβαση

Η IP **Διεύθυνση** του Server είναι η **195.251.123.227**, την οποία χρησιμοποιούμε για να συνδεθούμε σε αυτόν είτε μέσα από Client Εφαρμογές όπως το SQL Developer και το PG-Admin ή για πρόσβαση στις Web Υπηρεσίες του από τον Web Browser μας. Ο Server ακούει για συνδέσεις SSH και RDP στις Default Θύρες τους.

Ο Server είναι μέλος του **ενδοδικτύου** (Intranet) του Τμήματος, οπότε η πρόσβαση σε αυτόν γίνεται **εσωτερικά** από δίκτυο του Τμήματος. Εξωτερική πρόσβαση σε αυτόν (εκτός τμήματος ή εκτός ΑΤΕΙΘ) γίνεται μόνο με **σύνδεση VPN** στο Τμήμα Μηχανικών Πληροφορικής.

6.1.2 – Σύσταση Hardware

Ο Server του τμήματος απαρτίζεται από τα εξής συστατικά υλικού (Hardware), στον παρακάτω πίνακα:

Hardware	Χαρακτηριστικά
Επεξεργαστής	Intel Xeon 3430 στα 2,40 MHz , τεσσάρων πυρήνων, αρχιτεκτονικής x64 (64bit).
Μνήμη	8 GB DDR3
Δίσκος	50 GB

6.1.3 – Λογισμικό

Παραθέτουμε στον παρακάτω πίνακα το λογισμικό που είναι εγκατεστημένο σε αυτόν και οι αντίστοιχες εκδόσεις αυτού.

Λογισμικό	Έκδοση
Λειτουργικό Σύστημα Debian	Debian 7 Wheezy (Stable) 64bit
Oracle Database	11g Express Edition (11.2.0)
PostgreSQL	9.1
Apache HTTP Server (XAMPP)	2.4.4
MySQL (XAMPP)	5.5.32
PHP (XAMPP)	5.4.19
PHPMyAdmin (XAMPP)	4.0.4
XAMPP for Linux (LAMPP)	1.8.2
OpenSSH Server	6.0.p1
XRDP	0.5.0.1
E107 Website System	2.0 alpha 1

6.2 – Υπηρεσίες του Server

Σε αυτήν την ενότητα θα δούμε τις υπηρεσίες που παρέχει ο εν λόγω Server στο Τμήμα Μηχανικών Πληροφορικής. Για διευκόλυνση του αναγνώστη, θα γίνουν ανά κατηγορία και οι ρυθμίσεις / λειτουργίες σε πίνακες.

6.2.1 – Υπηρεσίες Βάσεων Δεδομένων

Περιγράφουμε τα Συστήματα Διαχείρισης Βάσεων Δεδομένων (DBMS) που υπάρχουν εγκατεστημένα στο Server και διατίθενται προς χρήση των σπουδαστών αλλά και των διδασκόντων.

6.2.1.1 – Oracle 11g Express Edition

Στον Server υπάρχει εγκατεστημένη η Oracle 11g Express Edition που παρέχει υπηρεσίες βάσεων δεδομένων για τις εκπαιδευτικές ανάγκες του τμήματος για τους προπτυχιακούς και μεταπτυχιακούς φοιτητές.

6.2.1.1.1 – Χαρακτηριστικά Βάσης Δεδομένων Oracle

Η βάση δεδομένων που τρέχει στον Server έχει **Instance ID** (SID) και Όνομα «**dblabs**» και διαθέτει τέσσερις λογαριασμούς διαχειριστών (SYSDBA) και διακόσιους λογαριασμούς απλών χρηστών για τους σπουδαστές. Είναι προσπελάσιμη μέσα από το SQL Developer και από το διαδικτυακό περιβάλλον Oracle APEX (και προφανώς τοπικά από το SQL*PLUS). Ο Listener της βάσης ακούει στην θύρα **1521** και η υπηρεσία Oracle APEX στην θύρα **8080**.

Παραθέτουμε τα καίρια χαρακτηριστικά της βάσης δεδομένων στον πίνακα που ακολουθεί παρακάτω:

Χαρακτηριστικό	Περιγραφή / Λειτουργία
Instance ID (SID) / Όνομα Βάσης	dblabs
Λογαριασμοί Διαχειριστών	4
Λογαριασμοί Σπουδαστών	200
Προσπελάσιμη από SQL Developer	Ναι (εσωτερικά ή με VPN)
Προσπελάσιμη από SQL*PLUS	Ναι (τοπικά)
Όνομα Listener	LISTENER
Θύρα Listener	1521
Tablespace διαχειριστών	SYSTEM
Tablespace Χρηστών	USERS (182 MB)
Εφεδρικό Tablespace Χρηστών	USERS2 (600 MB)
UNDO_MANAGEMENT	AUTO
Διαδικτυακές Υπηρεσίες της βάσης	APEX, XML DB, Enterprise Manager
Θύρα Ακρόασης APEX	8080
Character Set	ELMSNWIN1253 (Ελληνικά) + ALUTF16 (πολύγλωσση)

6.2.1.1.2 – Σπουδαστές

Οι σπουδαστές του τμήματος που παρακολουθούνε σχετικό μάθημα, διαθέτουν λογαριασμό χρήστη στην βάση δεδομένων. Τα Usernames και τα Passwords είναι καθήκον των διαχειριστών/διδασκόντων τα παρέχουν στους χρήστες, και να τους διαγράφουν με την ολοκλήρωση του εκπαιδευτικού εξαμήνου. Τα διαπιστευτήρια του χρήστη είναι κοινά για τον λογαριασμό που έχουν στην βάση και για τον λογαριασμό που έχουν στο Oracle APEX.

Κάθε σπουδαστής έχει εκχώρηση χώρου (Quota) 100 MB στο Tablespace USERS. Στην βάση μπορούνε να συνδεθούνε είτε μέσα από το SQL Developer είτε μέσα από το APEX. Για την πρώτη περίπτωση, εισάγουμε στο πεδίο «Host» την διεύθυνση του Server (195.251.123.227), ενώ για την δεύτερη, επισκέπτονται την διεύθυνση <http://195.251.123.227:8080/apex>

Στην βάση δεδομένων οι σπουδαστές έχουν δικαιώματα δημιουργίας αντικειμένων (CREATE TABLE, CREATE PROCEDURE, CREATE TYPE,

CREATE VIEW) και δικαίωμα σύνδεσης (CREATE SESSION). **ΔΕΝ** έχουν δικαιώματα εκτέλεσης Procedures άλλων χρηστών, προβολής πινάκων σε άλλα Schemas ή γενικότερα τροποποίησης αντικειμένων πέραν της δικαιοδοσίας του δικού τους Schema. Το κάθε Schema είναι ιδιωτικό και προστατευμένο. Επίσης έχει το ίδιο όνομα με τον σπουδαστή/χρήστη που είναι ο ιδιοκτήτης του. Φυσικά τα Schemas είναι προσπελάσιμα από τους διαχειριστές.

Στο **Oracle APEX** οι σπουδαστές έχουν τον ρόλο του Developer, και εργάζονται μόνο στα δικά τους Schemas. Το Workspace για τις εκπαιδευτικές ανάγκες είναι το **Students** και οι διαχειριστές του Workspace έχουν πρόσβαση σε όλα τα επί μέρους Schemas που είναι συσχετιζόμενα με το Workspace. Αυτός είναι ο τρόπος ελέγχου των επιμέρους εργασιών που ανατίθενται στους σπουδαστές να περατώσουν.

Οι σπουδαστές επίσης μπορούν να συνδεθούν στην βάση μέσω Java χρησιμοποιώντας τον JDBC Driver της Oracle. Το Connection String που θα πρέπει να εισάγουμε είναι το εξής:

"jdbc:oracle:thin:@195.251.123.227:8080:dblabs", "<username>", "<password>"

6.2.1.2 – PostgreSQL 9.1

Στον Server λειτουργεί η PostgreSQL 9.1 για εκπαιδευτική χρήση. Η βάση δεδομένων που δημιουργήθηκε για κάλυψη των εκπαιδευτικών αναγκών, είναι η «db_labs». Τα χαρακτηριστικά της τα παραθέτω στον παρακάτω πίνακα για γρήγορη προβολή:

Χαρακτηριστικό	Περιγραφή/Λειτουργία
SID Βάσης	db_labs
Θύρα Ακρόασης	5432
Λογαριασμοί διαχειριστών	3
Προσβάσιμη από PG-Admin	Ναι (εσωτερικά η με VPN)

Λογαριασμοί απλών χρηστών	200 (με δικά τους schemas)
---------------------------	----------------------------

6.2.1.2.1 – Σπουδαστές

Οι σπουδαστές έχουν τα ίδια διαπιστευτήρια με αυτά της Oracle. Στην βάση έχουν πρόσβαση από το PG-Admin είτε εσωτερικά του τμήματος, είτε εξωτερικά με σύνδεση VPN. Τα Schemas τους, έχουν όνομα «sch_<όνομα_χρήστη>».

Για σύνδεση με το PG-Admin θα πρέπει να εισάγουμε τα εξής χαρακτηριστικά, Host = «195.251.123.227», θύρα = «5432», Maintenance DB = «db_labs» username + password = τα διαπιστευτήρια που τους δόθηκαν.

6.2.1.3 – MySQL 5.5.32

Στον Server παρέχεται η υπηρεσία βάσεων δεδομένων MySQL 5.5.32 κυρίως για την υποστήριξη της ιστοσελίδας που υπάρχει στον Server για το Oracle Academy Advanced Computer Science, καθώς είναι στημένη σε CMS. Συμπεριλαμβάνεται στο πακέτο XAMPP οπότε είναι Προσβάσιμη από το PHPMyAdmin για διαχείριση.

6.2.2 – Υπηρεσίες Web

Παραθέτουμε τις διαδικτυακές υπηρεσίες που παρέχει ο Server στο προσωπικό του τμήματος και στους σπουδαστές. Να υπενθυμίσουμε πως και αυτές είναι διαθέσιμες μόνο στο εσωτερικό του τμήματος ή εξωτερική μέσω VPN.

6.2.2.1 – Apache HTTP Server και PHP

Ο Server παρέχει διαδικτυακές υπηρεσίες στους χρήστες μέσω του Apache HTTP Server 2.4.4, με υποστήριξη PHP (πακέτο XAMPP).

6.2.2.2 – Portal του προγράμματος Oracle Advanced Computer Science

Στον Server υπάρχει η διαδικτυακή πλατφόρμα (Portal) που αφορά της υπηρεσίες του προγράμματος Advanced Computer Science, της Oracle Academy. Είναι στημένο με το CMS E107 Website System, και σε αυτό θα παρέχονται σημαντικές πληροφορίες για το εν λόγω πρόγραμμα καθώς και σημαντικά νέα / ανακοινώσεις.

Ολοκληρώνοντας...

Αυτή ήταν η περιγραφή του Server του Τμήματος Μηχανικών Πληροφορικής του ΑΤΕΙ Θεσσαλονίκης, καθώς και των υπηρεσιών που προσφέρει στο προσωπικό και στους σπουδαστές. Η εκπαιδευτική πλατφόρμα της Oracle, αν και ακόμα βρίσκεται σε στάδιο δοκιμής, δίνει ενθαρρυντικά αποτελέσματα για την μελλοντική χρήση της για την κάλυψη των εκπαιδευτικών αναγκών. Οι σπουδαστές του τμήματος, θα έχουν την ευκαιρία να εξειδικευτούν στην χρήση της Oracle Database, καθώς και να ενταχθούν στο πρόγραμμα Oracle Advanced Computer Science, κάτι που σίγουρα θα τους δώσει ισχυρό πλεονέκτημα στην επαγγελματική τους αποκατάσταση στο μέλλον.

ΕΠΙΛΟΓΟΣ

Και κάπου εδώ φτάνουμε στο τέλος αυτού του συγγράμματος. Εδώ κάπου σταματάει η δική μου δικαιοδοσία και αναλαμβάνετε εσείς τα ηνία της Oracle. Είμαι σίγουρος ότι πήρατε πολλές γνώσεις μέσα από αυτές τις σελίδες.

Εν κατακλείδι λοιπόν, προσεγγίσαμε, αναλύσαμε, και χρησιμοποιήσαμε πολλά από τα επί μέρους χαρακτηριστικά της Oracle Database 11g. Είδαμε τα εργαλεία που παρέχει, της δυνατότητές της, και για το τι είναι ή δεν είναι ικανή να κάνει. Ασχοληθήκαμε μαζί της από την εγκατάστασή της στον Server μέχρι την εκτέλεση του τελευταίου «SELECT» Query. Καλύψαμε τις δύο πιο σημαντικές ομάδες χρηστών, τους διαχειριστές και τους απλούς χρήστες. Η ενασχόλησή μας λοιπόν με όλα τα παραπάνω, μας οδηγεί σε σημαντικά συμπεράσματα. Συμπεραίνουμε λοιπόν πως η Oracle Database είναι «σοβαρό» DBMS, που σχεδιάστηκε με γνώμονα τις ανάγκες του εκάστοτε χρήστη, χρησιμοποιεί καινοτόμες τεχνολογίες και παρέχει αυτό που λίγα DBMS παρέχουν, αξιοπιστία. Ως εκ τούτου, φαίνεται να είναι ιδανική για περιβάλλοντα με πολλούς χρήστες και απαιτητικές εργασίες, όπως πχ μια εκπαιδευτική πλατφόρμα. Με τα εργαλεία και τις ευκολίες που παρέχει, είναι βέβαιο πως θα διεγείρει το ενδιαφέρον του σπουδαστικού κοινού να εργαστεί σε αυτό.

Δεν πρέπει όμως να παραλείψουμε και την υπηρεσία Oracle Academy. Είδαμε πως μπορεί κανείς να αποκτήσει γνώσεις και πιστοποιήσεις κύρους από την ίδια την Oracle, μέσω του εκπαιδευτικού φορέα που σπουδάζουν. Συμπερασματικά λοιπόν, μια κίνηση σαν κι αυτή έχει πολλά να προσφέρει όχι μόνο στους σπουδαστές και τους διδάσκοντες, αλλά και στον ίδιο τον φορέα. Σκεφτείτε λοιπόν, ειδικά στην χώρα μας, ένας σπουδαστής πιστοποιημένος από την Oracle, δεν θα είχε πολύ περισσότερες ευκαιρίες εργασίας; Η δική μου λογική, μου λέει ότι τέτοια άτομα ποτέ δεν χάνονται. Αντιθέτως πάνε όλο και πιο μπροστά.

Τέλος, υπογραμμίζεται ότι ένα μεγάλο μέρος των όλων όσων είδαμε αναφορικά με την Oracle Database και την υπηρεσία Oracle Academy, εφαρμόζονται σε εσωτερικό Server του τμήματος Μηχανικών Πληροφορικής του ΑΤΕΙ Θεσσαλονίκης. Ο δικός μου συμπερασμός είναι ότι οι φοιτητές έχουν να αποκομίσουν πολλά οφέλη από τις υπηρεσίες που παρέχει ο Server, αφενός μεν

γιατί τρέχει την Oracle Database σαν βάση δεδομένων, με τις καινοτομίες της Oracle όπως το APEX, αφετέρου δε θα αποκτήσουν πολύ καλή γνώση και πληροφορίες για το πρόγραμμα Advanced Computer Science που ενδέχεται να ενταχθεί στους εκπαιδευτικούς πόρους του τμήματος. Ειδικά όσον αφορά τα εργαλεία χρήσης όπως το APEX ή/και το SQL Developer, σίγουρα οι σπουδαστές θα σταματήσουν να «βαρυγκωμούν», ύστερα από κακές εμπειρίες με το Data studio για την DB2.

Ελπίζω το παρών σύγγραμμα να σας έδωσε πολύτιμες γνώσεις και εμπειρίες, ώστε να μπορείτε να εφαρμόζετε και μόνοι σας αυτά που περιέχονται σε αυτό. Ελπίζω επίσης να σαν διέγειρα το ενδιαφέρον να ασχοληθείτε ποιο σοβαρά με την Oracle Database ή/και με το πρόγραμμα Advanced Computer Science. Η εκπαιδευτική πλατφόρμα που μπορείτε να δημιουργήσετε, να ξέρετε πως ίσως σας «ζορίσει» στην αρχή μέχρι να την εγκαταστήσετε, αλλά επίσης θα πρέπει να γνωρίζετε πως δεν θα σας απογοητεύσει. Το πιστεύω και για ένα μεγάλο μέρος της καλής λειτουργίας του (αν και εφόσον η εγκατάσταση έγινε σωστά και παραμετροποιήθηκε σωστά) μπορώ να σα εγγυηθώ. Τις αρχικές δοκιμές μου για να σας παραθέσω τα βήματα, τις έκανα σε προβληματικό υπολογιστή που συνέχεια κολλούσε ή έκλεινε. Σας πληροφορώ πως ποτέ δεν έχασα ούτε γραμμή πίνακα. Η βάση πάντοτε άνοιγε και πάντοτε δούλευε.

Ελπίζω λοιπόν να σας άρεσε το παρών σύγγραμμα και να συντέλεσε στην ενίσχυση του γνωστικού σας υπόβαθρου, και ελπίζω να σας βοηθήσει και να σας βγάλει από κάποια δύσκολη θέση για οποιαδήποτε πρόκληση αντιμετωπίσετε στην Oracle 11g.

Σας ευχαριστώ για τον χρόνο που διαθέσατε.

ΑΛΑΓΚΙΟΖΙΔΗΣ ΕΛΕΥΘΕΡΙΟΣ - ΠΕΤΡΟΣ

Παράρτημα Α' – Εγκατάσταση και Παραμετροποίηση PostgreSQL 9.1 σε Debian 7 Wheezy x64

Η PostgreSQL 9.1 υπάρχει διαθέσιμη σε μορφή precompiled πακέτου στον APT Package Manager. Παρακάτω περιγράφονται τα βήματα εγκατάστασης και παραμετροποίησης.

Βήμα 1 – Εγκατάσταση

Αρχικά συνδεόμαστε σαν χρήστης root:

```
user@localhost$ su
```

```
Password:
```

Ενημερώνουμε την βάση δεδομένων των αποθετηρίων του APT:

```
root@localhost# apt-get update
```

Εγκαθιστούμε την PostgreSQL μαζί με τα PostgreSQL Client front ends.

```
root@localhost# apt-get install postgresql-9.1 postgresql-client
```

Η εγκατάσταση θα πρέπει να ολοκληρωθεί χωρίς προβλήματα. **Αν και μόνο αν** υπάρξουν ζητήματα dependencies, εγκαταστήστε και το πακέτο **libc6**.

Βήμα 2 – Αναδημιουργία χρήστη Postgres

Αρχικά θα πρέπει να σταματήσουμε την υπηρεσία postgresql από το σύστημα άμεσα. Σαν χρήστες root γράφουμε:

```
root@localhost# /etc/init.d/postgresql stop
```

```
root@localhost# service postgresql stop
```

Η εγκατάσταση δημιούργησε για μας ένα χρήστη Postgres. Ο συγκεκριμένος χρήστης είναι ο Super Administrator του συστήματος της

PostgreSQL. Ωστόσο όμως ο χρήστη δημιουργήθηκε με default shell το /bin/sh και χωρίς home directory οπότε θα πρέπει να τον διαγράψουμε και να τον ξαναδημιουργήσουμε. Πρώτα μας ενδιαφέρει να βρούμε το ID του χρήστη που δημιουργήθηκε κατά το Installation γιατί θα μας χρειαστεί στην αναδημιουργία του.

```
root@localhost# cat /etc/passwd | grep postgres
```

Θα μας επιστραφεί η γραμμή με τα στοιχεία του postgres με τη μορφή «postgres:x:1002:1005». Στην προκείμενη, το User ID είναι το 1002. Το σημειώνουμε κάπου και ξεκινούμε την διαγραφή του χρήστη

```
root@localhost# userdel postgres -r
```

Ο χρήστης διαγράφηκε. Τώρα τον αναδημιουργούμε με πρόσθετα χαρακτηριστικά, δηλαδή θα δημιουργηθεί για αυτόν **home directory** (/home/postgres) και το default shell του θα είναι το **Bash**.

```
root@localhost# useradd -m -s /bin/bash postgres
```

Ορίζουμε κωδικό πρόσβασης για τον εν λόγω χρήστη με την παρακάτω εντολή:

```
root@localhost# passwd postgres
```

```
Enter new UNIX password: # ← Εισάγετε password
```

```
Retype New Unix password: # ← Ξανά εισαγωγή για επιβεβαίωση
```

```
Password updated successfully.
```

Αλλάζουμε το User ID (αυτό που σημειώσαμε) του χρήστη Postgres με την εντολή:

```
root@localhost# usermod -u <user-id> # ← το user id που σημειώσατε
```

Επιβεβαιώνουμε τις αλλαγές με:

```
root@localhost# cat /etc/passwd | grep postgres
```

```
postgres:x:<user_id>:1005::/home/postgres:/bin/bash:
```

Βήμα 3 – Αρχικοποίηση νέου Cluster

Τώρα θα πρέπει να αρχικοποιήσουμε την PostgreSQL ώστε να χρησιμοποιεί έναν δικό μας κατάλογο για την διαχείριση και την αποθήκευση των βάσεων δεδομένων, δηλαδή ένα νέο Cluster. Συνδεόμαστε ως χρήστες postgres και μεταβαίνουμε στο home directory του τελευταίου με:

```
root@localhost# su postgres
```

```
#η μετάβαση από χρήστη root σε απλό χρήστη δεν ζητάει password
```

```
postgres@localhost$ cd /home/postgres
```

Δημιουργούμε ένα κατάλογο στο home directory, με ένα όνομα της επιλογής μας, π.χ. «postgres_base».

```
postgres@localhost$ mkdir postgres_base
```

Δημιουργούμε το νέο Cluster μέσα στον κατάλογο που δημιουργήσαμε γράφοντας:

```
postgres@localhost$ /usr/lib/postgresql/9.1/bin/initdb -D ./postgres_base
```

Το Cluster μας δημιουργήθηκε με επιτυχία στον νέο κατάλογο που ορίσαμε, οπότε τώρα πρέπει να ασχοληθούμε με το δικτυακό κομμάτι.

Βήμα 4 – Ρυθμίσεις Δικτύου

Για να μπορέσει ο Server να «ακούει» για συνδέσεις και να εξυπηρετεί αυτές που πρέπει, πρέπει να ρυθμιστούνε τα κατάλληλα attributes. Αρχικά θα πρέπει να πούμε στον Server, να ακούει για συνδέσεις από **οποιαδήποτε IP Διεύθυνση** πειράζοντας το αρχείο postgresql.conf με τον GNU NANO όπως βλέπουμε στην παρακάτω:

```
postgres@localhost$ nano ./postgres_base/postgresql.conf
```

Ο Nano ανοίγει το αρχείο μας, και εμείς πρέπει να βρούμε την γραμμή «listen_addresses» η οποία είναι Commented. Για να βρείτε την γραμμή, πατήστε

τον συνδυασμό πλήκτρων Ctrl+W στον Nano, και εισάγετε σαν κριτήριο αναζήτησης το «listen_addresses». Ο δρομέας του Nano θα «παρκάρει» ακριβώς στην αρχή αυτής της γραμμής. Εσείς διαγράψτε τον χαρακτήρα « # » (uncomment) που προηγείται της γραμμής και αλλάξτε την τιμή «'localhost'» σε «'*'» (αστερίσκο). Στην παρακάτω εικόνα βλέπουμε την διαδικασία:

```

GNU nano 2.2.6 File: postgresql.conf
#external_pid_file = '(none)'          # write an extra PID file
                                       # (change requires restart)

#-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -

listen_addresses = '*'                 # what IP address(es) to listen on;
                                       # comma separated list of addresses;
                                       # defaults to 'localhost', '*' = all
                                       # (change requires restart)
#port = 5432                            # (change requires restart)
max_connections = 100                  # (change requires restart)
# Note: Increasing max_connections costs ~400 bytes of shared memory per

```

Η γραμμή listen_addresses που πρέπει να αλλάξουμε

Αφού αλλάξετε τη γραμμή, πατήστε τον συνδυασμό πλήκτρων Ctrl+X και πατήστε Y στο prompt αποθήκευσης για να αποθηκευτεί το αρχείο και να κλείσει ο Nano.

Τώρα θα πρέπει να ρυθμίσουμε τον Server έτσι ώστε να δέχεται ή όχι συγκεκριμένο εύρος IP Διευθύνσεων και από ποιους χρήστες. **Καλή πρακτική** είναι ο Server να είναι προσπελάσιμος από το **ενδοδίκτυο** (intranet) του φορέα σας, ώστε να επιτρέπεται να συνδέονται μόνον όσοι βρίσκονται σε αυτό (συμπεριλαμβανομένου και απομακρυσμένης πρόσβασης με VPN), καθώς παρέχει ασφάλεια από μη εξουσιοδοτημένες εξωτερικές παρεμβάσεις. Για τους λόγους επίδειξης θα χρησιμοποιήσουμε το IP Δίκτυο 192.168.1.0/24.

Επίσης μπορούμε να περιορίσουμε τις επιτρεπτές συνδέσεις, επιλέγοντας ποιοι χρήστες θα μπορούν να συνδέονται και σε ποια βάση δεδομένων. **Καλή πρακτική** είναι να επιτρέπετε στους χρήστες-διαχειριστές να έχουν πρόσβαση σε όλες της βάσεις δεδομένων του Cluster, ενώ τους απλούς χρήστες μόνο στις βάσεις που τους αφορούνε.

Για την πραγματοποίηση των παραπάνω, πρέπει να τροποποιήσουμε το αρχείο `pg_hba.conf` που βρίσκεται αποθηκευμένο στον κατάλογο του `cluster`, με τον GNU Nano:

```
postgres@localhost$ nano ./postgres_base/pg_hba.conf
```

Αφού ο Nano ανοίξει το αρχείο, μεταβείτε τέρμα κάτω. Υπάρχουν πέντε στήλες που μας αφορούν, οι οποίες είναι:

Στήλη	Σημασία
Type	τύπος πελάτη
Database	επιτρεπόμενη βάση δεδομένων
User	Επιτρεπόμενος χρήστης
Address	Επιτρεπόμενη IP Διεύθυνση ή IP Δίκτυο.
Method	Τύπος Ταυτοποίησης χρήστη

Για τους **διαχειριστές της βάσης**, δώστε τις παρακάτω τιμές (με κόκκινο χρώμα αυτά που δίνετε εσείς) αλλάζοντας στήλη με tab. Παράλληλα τις εξηγούμε:

- 1) Type = **host** (απλός host)
- 2) Database = **all** (όλες οι βάσεις)
- 3) User = **<όνομα διαχειριστή>, {όνομα άλλου διαχειριστή (προαιρετικά)}**
- 4) Address = **192.168.1.0/24** (ενδεικτικό IP Δίκτυο)
- 5) Method = **md5** (με MD5 Hashed password)

Για τους **απλούς χρήστες**, σε νέα γραμμή, δώστε τις παρακάτω τιμές:

- 1) Type = **host**
- 2) Database = **<όνομα επιτρεπτής βάσης δεδομένων>** (πχ students)
- 3) User = **all** (όλοι οι χρήστες)
- 4) Address = **192.168.1.0/24** (ενδεικτικό)
- 5) Method = **md5**

Για επιπλέον διευκόλυνση, στην παρακάτω εικόνα δείχνουμε ένα στιγμιότυπο από ένα συμπληρωμένο pg_hba.conf για να πάρετε μια ιδέα επιπλέον:

```
local all all trust
# IPv4 local connections:
host all all 127.0.0.1/32 trust
# IPv6 local connections:
host all all ::1/128 trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local replication postgres trust
#host replication postgres 127.0.0.1/32 trust
#host replication postgres ::1/128 trust
host all leleris 192.168.1.0/24 md5
host db_main all 192.18.1.0/24 md5
```

Παράδειγμα συμπληρωμένου αρχείου pg_hba.conf

Αφού τελειώσετε, κλείνετε τον Nano με Ctrl+X και αποθηκεύετε το αρχείο. Τώρα μπορούμε να ξεκινήσουμε τον Server.

Βήμα 5 – Εκκίνηση Server και Cluster

Για να ξεκινήσουμε τον server, βεβαιωνόμαστε ότι είμαστε συνδεδεμένοι ως χρήστες postgres και δίνουμε την παρακάτω εντολή για να το πράξουμε:

```
postgres@localhost$ /usr/lib/postgresql/9.1/bin/postgres -D ./pg_databases &
```

Με το σύμβολο « & » η διεργασία του Server τρέχει στο παρασκήνιο. Πατήστε Enter για να επιστρέψετε σε prompt. Ο Server είναι ενεργός πλέον και ακούει για συνδέσεις.

Βήμα 6 – Διαχείριση του PostgreSQL Cluster

Αφού επιστρέψετε σε prompt δώστε την εντολή psql (ως χρήστες postgres) για να ξεκινήσει το PostgreSQL shell. Εκεί λοιπόν μπορείτε να δημιουργήσετε βάσεις, χρήστες, πίνακες και ότι άλλο. Σημειώνεται ότι υπάρχει μια προεπιλεγμένη βάση δεδομένων με όνομα «postgres» και ότι ο Super Administrator του Cluster είναι ο χρήστης postgres. Παρακάτω δίνονται μερικές ενδεικτικές εργασίες που μπορείτε να κάνετε. Σημειώνουμε ότι το prompt postgres δηλώνει την Default βάση δεδομένων του Cluster που έχει αυτό το όνομα.

Για να δημιουργήσετε **βάση δεδομένων** γράψετε στο PostgreSQL shell αφού το εκκινήσετε με την psql:

```
postgres=# create database <όνομα βάσης> owner postgres;
```

```
CREATE DATABASE.
```

Για να δημιουργήσετε νέο **χρήστη** γράψετε:

```
postgres=# create user <όνομα χρήστη> with password ' <password χρήστη> ' ;
```

```
CREATE ROLE
```

Για να δημιουργήσετε ένα νέο **Schema** γράψετε:

```
postgres=# create schema <όνομα schema> authorization <όνομα χρήστη-ιδιοκτήτη>;
```

```
CREATE SCHEMA
```

Για να αλλάξετε το **default schema** ενός χρήστη γράφετε:

```
postgres=# alter user <όνομα χρήστη> set search_path = <όνομα schema>;  
  
ALTER ROLE
```

Για να δώσετε **Admin δικαιώματα** σε κάποιον χρήστη σε μια **συγκεκριμένη βάση δεδομένων** γράφετε:

```
postgres=# grant all privileges on database <όνομα βάσης> to <όνομα χρήστη>;  
  
GRANT
```

Για να συνδεθείτε σε **άλλη βάση δεδομένων** ως **άλλος χρήστης**, γράφετε:

```
postgres=# \c <όνομα βάσης> <όνομα χρήστη>; #αν δεν βάλετε χρήστη, θα αντικατασταθεί με postgres  
  
You are now connected to data base "<όνομα βάσης>" as user "<όνομα χρήστη>";  
<όνομα βάσης>=> #to => prompt δηλώνει απλό χρήστη
```

Για να εκτελέσετε **scripts** από αρχείο γράφετε:

```
postgres=# \i <path προς αρχείο> ;
```

Για να δείτε όλες τις **διαθέσιμες ενέργειες** γράφετε:

```
postgres=# \?;
```

Τέλος, για να **κλείσετε** το shell της postgres, γράφετε

```
postgres=# \q;
```

Αφού λοιπόν έχετε δημιουργήσει βάσεις, schemas, χρήστες και οτιδήποτε άλλο, μπορούν οι clients να συνδεθούν στον Server με κάποια Client Εφαρμογή, όπως το **PG-Admin**.

Βήμα 7 – Τερματισμός Server

Για να τερματίσετε τον Server ώστε να σταματήσει η υπηρεσία της PostgreSQL γράφετε:

```
postgres@localhost$ /usr/lib/postgresql/9.1/bin/pg_ctl -D ./pg_databases/ stop -m fast
```

Ο Server πλέον δεν δέχεται συνδέσεις και το Cluster δεν είναι διαθέσιμο. Πραγματοποιείτε αυτήν ενέργεια μόνο αν υπάρχει σοβαρός λόγος ή σε περιπτώσεις συντήρησης.

Παράρτημα Β΄ – Απόκτηση και εγκατάσταση Oracle SQL Developer

Σε αυτό το παράρτημα θα ασχοληθούμε με την απόκτηση και την εγκατάσταση της εφαρμογής Oracle SQL Developer. Να γνωρίζετε πως πρέπει να υπάρχει εγκατεστημένο το Oracle Java Development Kit στον υπολογιστή σας. Καλύπτουμε εγκατάσταση και σε Windows και σε Linux.

Απόκτηση.

Για να αποκτήσετε το Oracle SQL Developer μεταβείτε στην παρακάτω ιστοσελίδα:

<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>

Θα πρέπει να αποδεχτείτε τους όρους χρήσης για να γίνει ενεργός ο σύνδεσμος απόκτησης. Επιλέξτε την έκδοση που θέλετε ανάλογα με το λειτουργικό σας σύστημα και κατόπιν πατάτε Download όπως φαίνεται παρακάτω:

Overview Downloads Documentation Community Learn More

Oracle SQL Developer 3.2.2 (3.2.20.09.87)
November 1, 2012

SQL Developer 4.0 Early Adopter 2 is now available for [download](#)

You must accept the [OTN License Agreement for SQL Developer](#) to download this software.

Accept License Agreement Decline License Agreement

- 3.2.2 Release Notes
- 3.2.2 Bugs Fixed
- New Features
- Documentation

Platform	
Windows 32-bit - zip file <i>includes the JDK1.6.0_35</i>	Download 209 M
Windows 32-bit - Installation Notes	Download 174 M
Windows 64-bit - Installation Notes	Download 174 M
Mac OS X - Installation Notes	Download 173 M
Linux RPM - Installation Notes	Download 173 M
Other Platforms - Installation Notes	Download 174 M

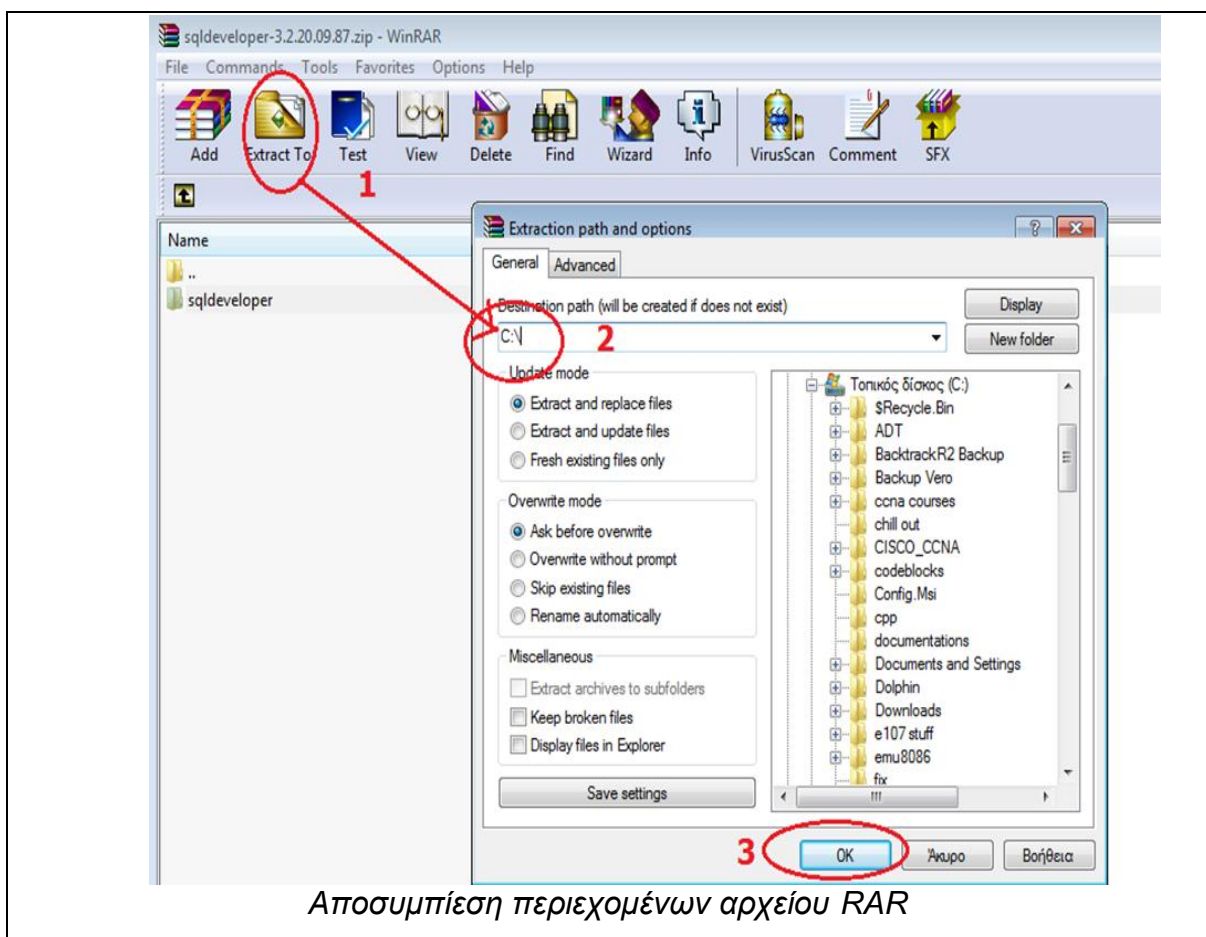
[Download previous releases here](#)

Σελίδα απόκτησης του Oracle SQL Developer

Θα σας ζητήσει **login** με το Oracle Account σας, οπότε εισάγετε τα credentials σας και το Download θα ξεκινήσει. Καθώς το αρχείο είναι μεγάλο, το Download ίσως καθυστερήσει λίγο. Με το πέρας του Download μπορείτε να προβείτε στην εγκατάσταση του SQL Developer.

Εγκατάσταση σε Windows

Αποσυμπίστε τα περιεχόμενα του ZIP στον κατάλογο C:/ ή σε άλλον κατάλογο της επιλογής σας, όπως βλέπουμε κάτωθι (στιγμιότυπο WINRAR):



Με το πέρας του Extract, μπορείτε να τρέξετε το SQL Developer μεταβαίνοντας στον κατάλογο «C:/sqldeveloper» και εκτελώντας το «sqldeveloper.exe». Προτείνεται η δημιουργία μιας συντόμευσης στο Desktop σας.

Εγκατάσταση σε Linux

Για να εγκαταστήσετε το SQL Developer σε περιβάλλον Linux θα πρέπει να βεβαιωθείτε ότι η διανομή σας υποστηρίζει τον RPM Package Manager. Αν είναι Red Hat based λογικά θα υπάρχει ήδη, ωστόσο αν είναι Debian ή Debian Based τότε πρέπει να εγκαταστήσετε τον Alien. Το Alien είναι ένα πρόγραμμα που μετατρέπει τα αρχεία RPM σε αρχεία DEB ώστε να μπορείτε να τα εγκαταστήσετε μέσω του Debian Package Manager (DPKG). Δώστε την παρακάτω εντολή ως χρήστης ROOT.

```
root@localhost# apt-get install alien
```

Αφού εγκατασταθεί το Alien τώρα μεταβείτε στον κατάλογο λήψης του SQL Developer και δώστε τις παρακάτω εντολές για να μετατρέψετε το RPM σε DEB (ως χρήστης root)

Αφού εγκατασταθεί το Alien τώρα μεταβείτε στον κατάλογο λήψης του SQL Developer και δώστε τις παρακάτω εντολές για να μετατρέψετε το RPM σε DEB (ως χρήστης Root)

```
root@localhost# alien -d sqldeveloper-3.2.XXXX.rpm
```

Τώρα μπορείτε να εγκαταστήσετε το SQL Developer με την παρακάτω εντολή (ως root)

```
root@localhost# dpkg -i sqldeveloper_3.2.XXXX.deb
```

Με το πέρας της εγκατάστασης, μπορείτε να εκτελέσετε το SQL Developer μέσα από το Terminal ή μέσα από το K Menu στην κατηγορία Development.

Βοήθεια, Μου ζητάει το JDK Path. Τι να κάνω;

Αν σας ζητήσει το JDK Path σημαίνει πως δεν μπορεί να εντοπίσει το JDK στον υπολογιστή σας. Η λύση είναι απλή. Φτιάξτε μια μεταβλητή περιβάλλοντος πχ JDK_PATH ως εξής

```
export JDK_PATH=/home/user/<path_to_jdk>
```

Τρέξτε ξανά το SQL Developer και στην προτροπή για εισαγωγή του PATH
δώστε:

`$JDK_PATH`

Και το SQL Developer λογικά θα ξεκινήσει

Παράρτημα Γ´ – Εγκατάσταση οδηγού JDBC στον NetBeans IDE

Σε αυτό το παράρτημα θα ασχοληθούμε με τον τρόπο εγκατάστασης του οδηγού JDBC στο NetBeans IDE. Θα πρέπει να υπάρχει εγκατεστημένο το NetBeans στο σύστημα, στην νεότερη έκδοσή του.

Απόκτηση - Εγκατάσταση

Αρχικό βήμα είναι η απόκτηση του Instant Client Basic. Μεταβείτε στην διεύθυνση <http://www.oracle.com/technetwork/topics/winsoft-085727.html> και στην ενότητα Version 11.2.0.3.0 (1) κατεβάστε το Instant Client Basic (2). (Απαιτεί αποδοχή όρων και login με το Oracle Account σας).

*Instant Client Package - Precompiler: Additional files for "proc" binary and related files to precompile a Pro*C application and demo

1 Version 11.2.0.3.0

Instant Client Package - Basic: All files required to run OCI, OCCI, and JDBC-OCI applications

2 [instantclient-basic-nt-11.2.0.3.0.zip \(51,149,941 bytes\)](#)

*Instant Client Package - Basic Lite: Smaller version of the Basic, with only English error messages and Unicode, ASCII, and Western European character set support

[instantclient-basclite-nt-11.2.0.3.0.zip \(20,043,020 bytes\)](#)

*Instant Client Package - JDBC Supplement: Additional support for XA, Internationalization, and RowSet operations under JDBC

[instantclient-jdbc-nt-11.2.0.3.0.zip \(1,565,746 bytes\)](#)

*Instant Client Package - SQL*Plus: Additional libraries and executable for running SQL*Plus with Instant Client

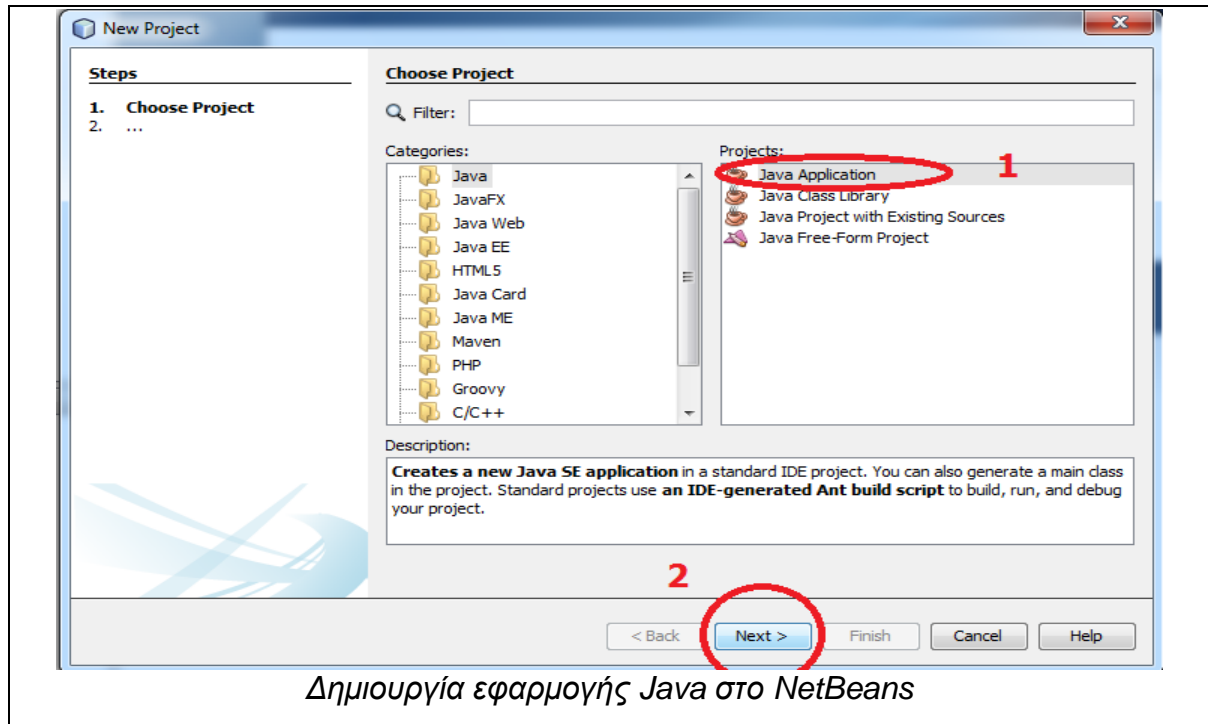
[instantclient-sqlplus-nt-11.2.0.3.0.zip \(759,693 bytes\)](#)

Οθόνη λήψης του Instant Client που περιέχει τον οδηγό JDBC

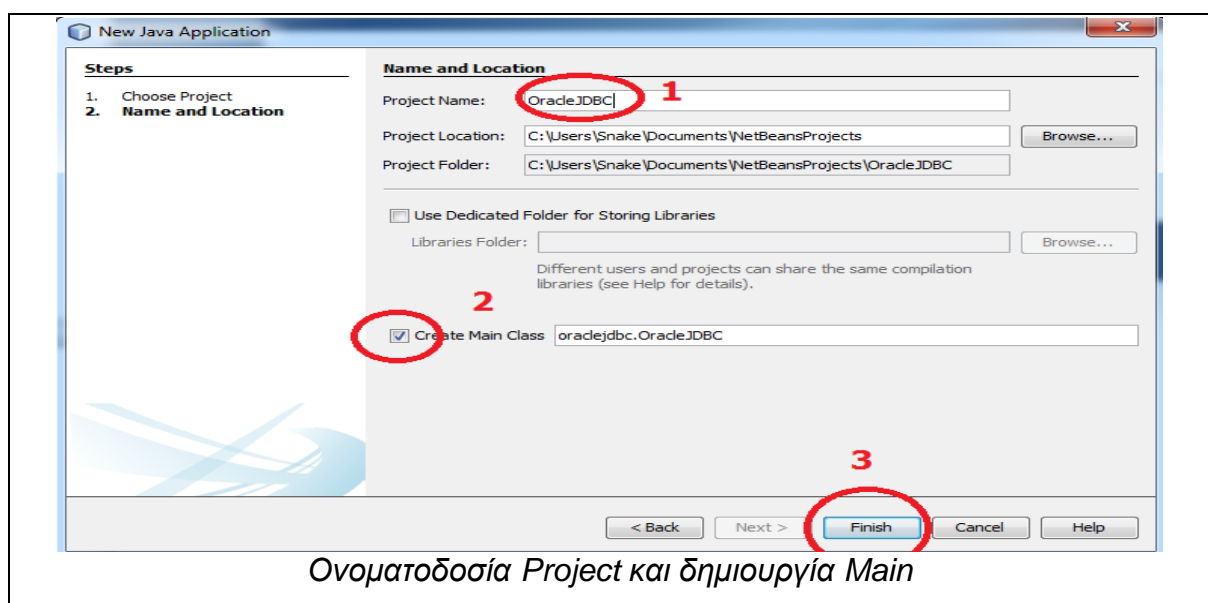
Κατόπιν ανοίξτε το αρχείο zip που κατεβάσατε και αποσυμπιέστε τα περιεχόμενά του στον C:/ (windows) ή στον home directory σας (Linux). Στον κατάλογο της επιλογής σας λοιπόν, λογικά θα υπάρχει ο κατάλογος «instantclient_11_2». Τώρα μπορούμε να αρχίσουμε να το χρησιμοποιούμε.

Χρήση του Oracle JDBC Driver

Ανοίξτε το Netbeans IDE και δημιουργήστε ένα νέο Project για **Java Application** (1) και πατήστε **Next** (2).

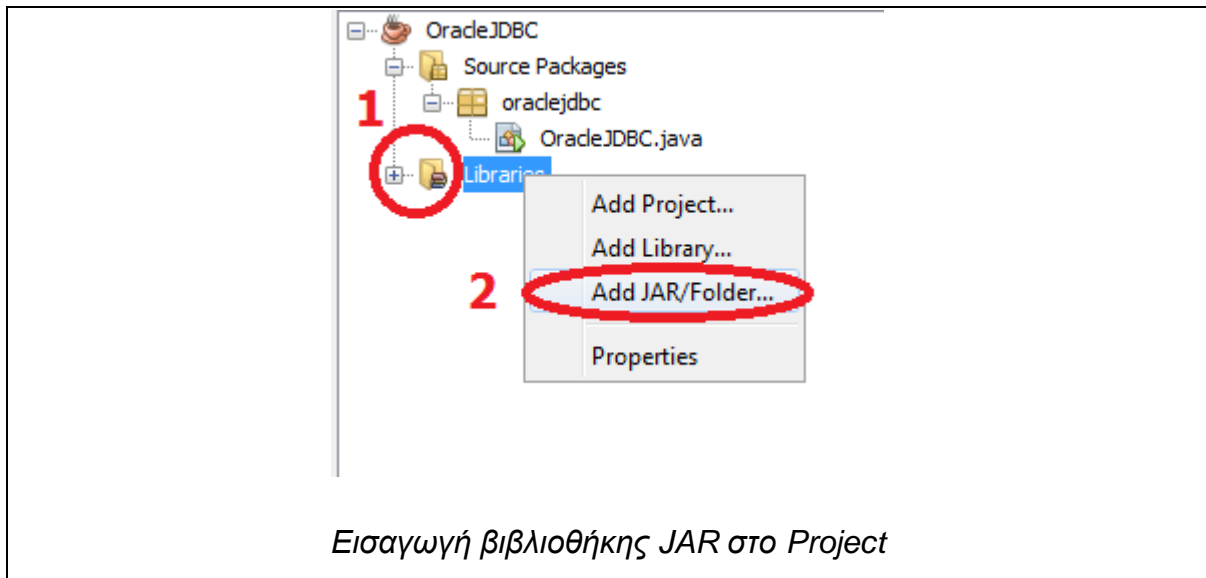


Στην συνέχεια, δώστε **όνομα** στο Project σας (1), επιλέξτε να δημιουργηθεί η **Main Class** (2) και πατήστε **Finish** (3).

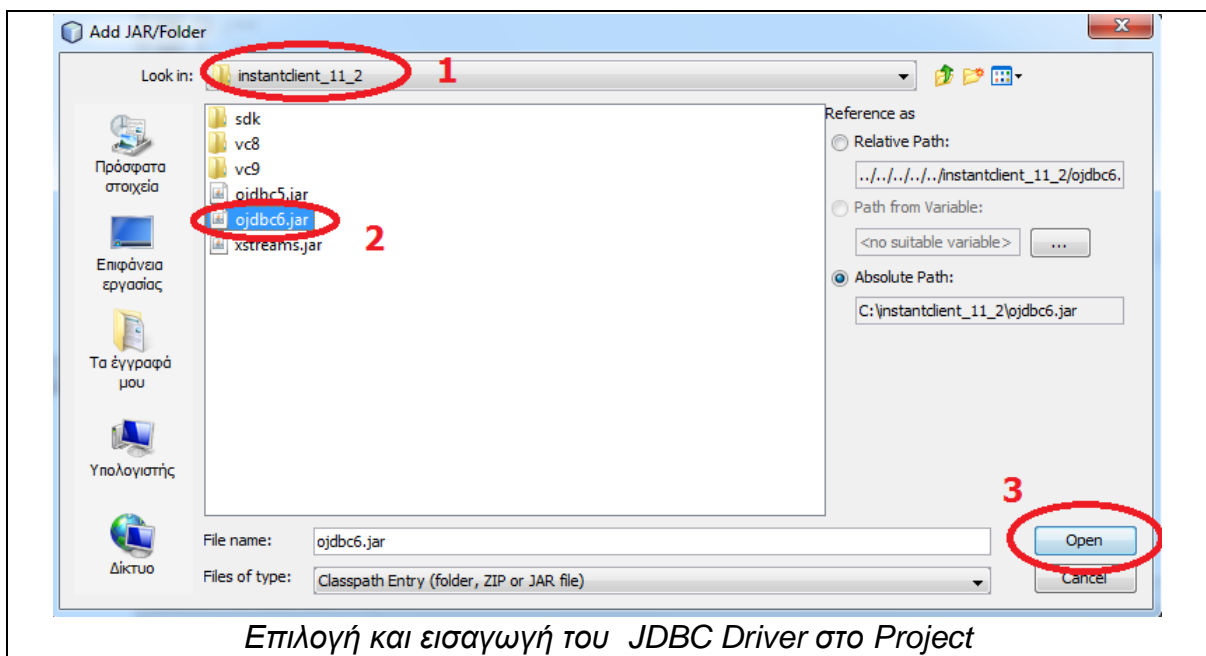


Το Project μας δημιουργήθηκε και είμαστε έτοιμοι για να ξεκινήσουμε. Δεν μένει παρά να εισάγουμε στο Project μας τον JDBC Driver.

Μεταβείτε αριστερά στα Project Properties, πατήστε δεξί-κλικ στο **Libraries** (1) και από το drop-down menu επιλέξτε **Add JAR/Folder** (2).



Στον File Browser που σας ανοίγει, μεταβείτε στον κατάλογο που αποσυμπίσατε τα περιεχόμενα του Oracle Instant Client και βρείτε τον κατάλογο **instant_client_11_2** (1) πχ C:\instantclient_11_2 και στην συνέχεια βρείτε το αρχείο **ojdbc6.jar** (2), επιλέξτε το και πατήστε **Open** (3).



Από αυτό το σημείο και έπειτα μπορείτε να χρησιμοποιήσετε την τεχνολογία JDBC με την εφαρμογής σας.

Παράρτημα Δ΄ – Κώδικας Ενδεικτικής Εφαρμογής JDBC

Παρατίθεται ο κώδικας της ενδεικτικής εφαρμογής της ενότητας «Εφαρμογές σε γλώσσα Java». Τα σχόλια είναι στα αγγλικά για να διευκολυνθείτε στην μεταφορά και δοκιμή του κώδικα αν το IDE σας έχει πρόβλημα με τους ελληνικούς χαρακτήρες.

```

/*
ORACLE JDBC Demo Application
BY ALAGKIOZIDIS ELEFThERIOS
elpalag@it.teithe.gr
aetos.it.teithe.gr/~elpalag
*/

package oraclejdbc;
import java.sql.*; //Java SQL Class
public static void main(String[] args) throws SQLException {
//SQL Exception May Arise During Runtime
Connection con=null;
con=DriverManager.getConnection("jdbc:oracle:thin:@192.168.1.69:1521:test","jd
bc","test123");

if (con != null) { //Connectivity Checking, Not Null means connected
    try { //Try is essential for SQL Exceptions
        System.out.println("Connection Established\n"); //Notify of connection
//Success
    } //try
    catch (SQLException ex) //Catch SQL Exception if occurs
    {

```

```

        System.err.println("Exception : " + ex.getMessage());//Show the type
//of the exception to STDERR
    }
} //if
else { //If Conn is null, no connection exists
    System.err.println("Connection Failed\n"); //Notify User of connection
//failure
} //else
String qry = "SELECT * FROM BOATS"; //SELECT Statement into a string
    Statement st=con.createStatement(); //Create A new Statement
    ResultSet rs = st.executeQuery(qry); //Create a new Result Set and store
//inside the Results from the SELECT query that will be executed

    while(rs.next()) //While the result set has more rows
    {
        int id = rs.getInt("ID"); //Store to id variable the current value of the ID column
        String name = rs.getString("NAME"); //Store to name variable the current
//value of the NAME Column
        System.out.println(id + "\t" + name); //Tab the two Columns
    } //while
    PreparedStatement pstmt = con.prepareStatement ("insert into BOATS(id,
name) values (?, ?)"); //Creation of Prepared Statement for execution of INSERT
//INTO statement. Values are inserted parametrically
    pstmt.setInt(1,6); //Column ID , value 6
    pstmt.setString (2, "Megaloxari"); //Column NAME value Megaloxari
    pstmt.executeUpdate();//Execute the Statement (Insert)
    st.executeQuery("commit"); //Commit Changes to Database
    System.out.println("Row Inserted Successfully"); //Inform of INSERT INTO
//Success

```

```

pstmt = con.prepareStatement("UPDATE BOATS SET NAME = ? WHERE ID =
?"); //Setting the prepared statement to run an UPDATE statement. Values are
//parametric once again.

pstmt.setInt(2, 3); //Column 1 value 3 (parameter 2)
pstmt.setString(1, "Eliza"); //Column 2 value Eliza (parameter 1)
pstmt.executeUpdate();//Execute statement (Update)
st.executeQuery("commit");//Commit Changes to Database
System.out.println("Table Updated\n");//Inform user of UPDATE success
pstmt = con.prepareStatement("DELETE FROM BOATS WHERE ID = ?");
//Set the prepared statement to execute a DELETE statement, parametrically.

pstmt.setInt(1, 5); //Column ID value 5 (parameter 1)
pstmt.executeUpdate();//Execute Statement (DELETE)
st.executeQuery("commit");//Commit Changes to Database
System.out.println("Row Deleted\n");//Inform user of DELETE Success
st.close();//Close Statements, no more needed
con.close();//Terminate Connection
} //try
catch (SQLException ex)
{
    System.err.println("Exception : "+ex.getMessage()); //Show Exception type
} //catch
} //if
else {
    System.err.println("Connection Failed, check your credentials"); //Error
connection
} //else

```

```
} //end of main  
  
} //end of class
```

Παρακάτω παραθέτουμε τον SQL Κώδικα για την δημιουργία του ενδεικτικού πίνακα, όπου θα εκτελέσετε είτε στο SQL Plus είτε στο SQL Developer.

```
--CREATE A TABLE FOR JDBC USER  
  
--By Alagkiozidis Eleftherios - Petros  
  
--elpalag@it.teithe.gr  
  
  
--DROP TABLE BOATS;  
  
  
CREATE TABLE BOATS  
(ID INTEGER NOT NULL PRIMARY KEY,  
NAME VARCHAR(30)  
);  
  
INSERT INTO BOATS(ID,NAME) VALUES (1,'Ouranoupolis');  
INSERT INTO BOATS(ID,NAME) VALUES (2,'Panormitis');  
INSERT INTO BOATS(ID,NAME) VALUES (3,'Seahorse');  
INSERT INTO BOATS(ID,NAME) VALUES (4,'Mariza');  
INSERT INTO BOATS(ID,NAME) VALUES (5,'Kap.Dimitrios');
```


Παράρτημα Ε΄ – Εγκατάσταση Οδηγού OCCI στο Microsoft Visual Studio 2010

Εγκατάσταση

Η διαδικασία εγκατάστασης του OCCI είναι λίγο πολύπλοκη, ωστόσο αν ακολουθήσετε πιστά τα βήματα, δεν θα αντιμετωπίσετε πρόβλημα.

ΠΡΟΣΟΧΗ: Θα πρέπει να έχετε στον υπολογιστή σας εγκατεστημένο το Microsoft Visual Studio 2008 ή το 2010. Τα βήματα που περιγράφουμε παρακάτω αφορούν του Visual Studio 2010. Επίσης απαιτείται και το Oracle Instant Client Basic (καλύφθηκε στο κομμάτι JDBC).

Αρχικά κατεβάστε τα εξής:

Μεταβείτε στο

<http://www.oracle.com/technetwork/database/occi/downloads-083553.html>

Κατεβάστε το Visual C++ 2010 → Release 11.2.0.3.0 → Windows 32 bit → ZIP όπως στην εικόνα (αν έχετε 64bit Windows κατεβάζετε το αντίστοιχο Windows 64 bit). Απαιτεί Login με το Account σας.

You must accept the OTN License Agreement to download this software.
 Accept License Agreement | Decline License Agreement

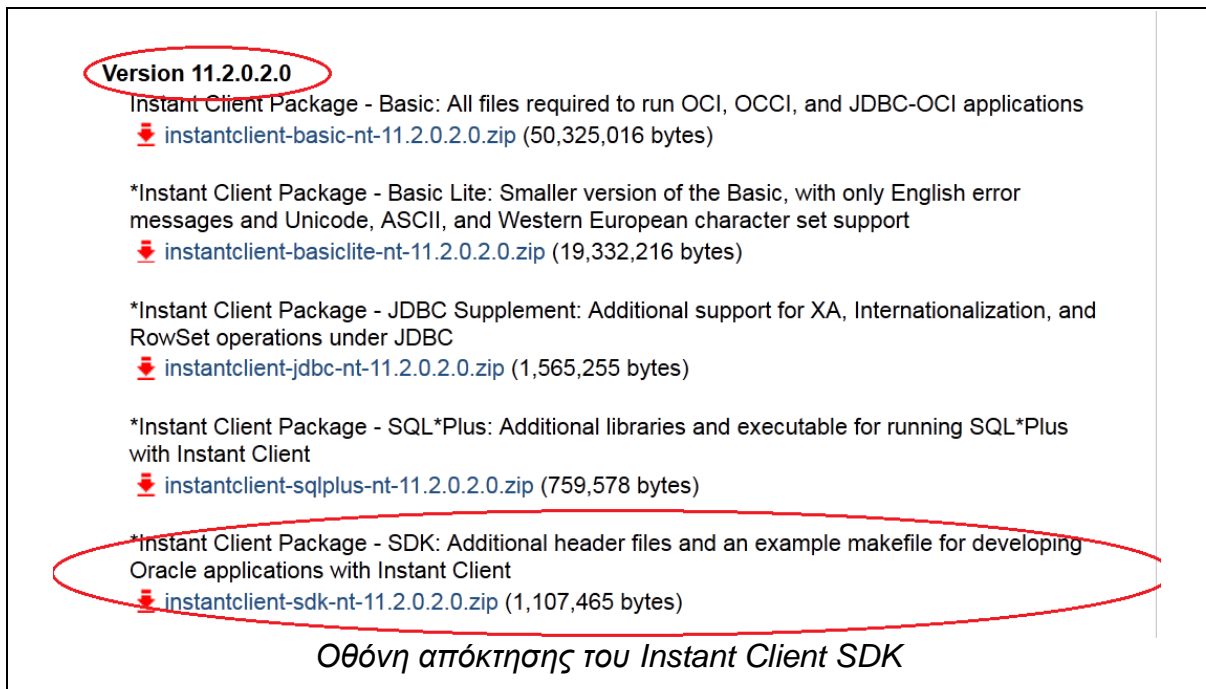
Αποδοχή

Oracle Release =>	11.2.0.3.0	11.2.0.2.0	11.1.0.7.0
Compiler/Platform		64 BIT	
Visual C++10 (VS 2010)[Windows 64-bit]	Link (zip)	Link (zip)	N.A
Visual C++10 (VS 2010)[Windows 32-bit]	Link (zip)	Link (zip)	N.A
Visual C++9 (VS 2008)[Windows x64]	Install	Install	Link (zip)
Visual C++9 (VS 2008)[Windows 32-bit]	Install	Install	N.A
Visual C++8 (VS 2005)[Windows 32-bit]	Install	Install	Install *

• Each downloadable archive contains a README with installation and usage instructions

Οθόνη απόκτησης OCCI

Στη συνέχεια μεταβείτε στην διεύθυνση (<http://www.oracle.com/technetwork/topics/winsoft-085727.html>) και κατεβάστε το Instant Client Package – SDK για την έκδοση 11.2.0.2.0 όπως στην παρακάτω εικόνα. Υποθέτουμε ότι το Instant Client Basic είναι ήδη εγκατεστημένο στο σύστημά σας. Αν δεν το έχετε κατεβάστε το κι αυτό. (Απαιτεί login με το Oracle Account σας)



Version 11.2.0.2.0
Instant Client Package - Basic: All files required to run OCI, OCCI, and JDBC-OCI applications
instantclient-basic-nt-11.2.0.2.0.zip (50,325,016 bytes)

*Instant Client Package - Basic Lite: Smaller version of the Basic, with only English error messages and Unicode, ASCII, and Western European character set support
instantclient-basclite-nt-11.2.0.2.0.zip (19,332,216 bytes)

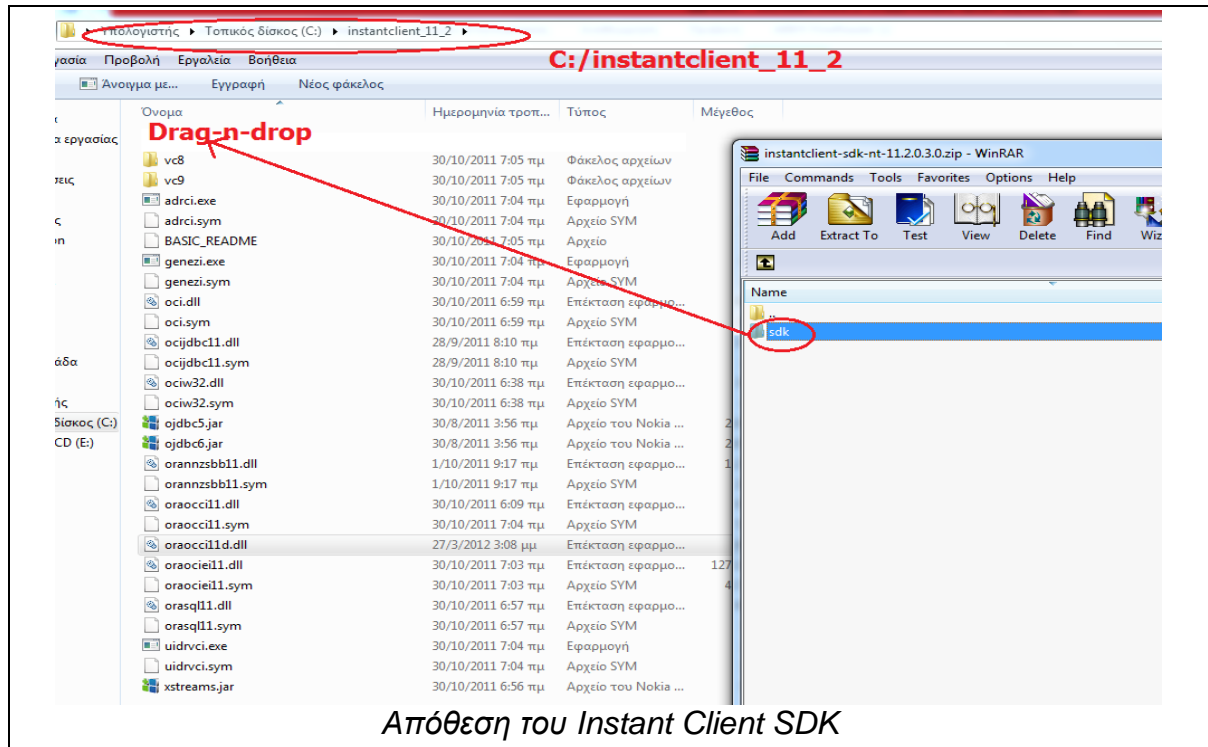
*Instant Client Package - JDBC Supplement: Additional support for XA, Internationalization, and RowSet operations under JDBC
instantclient-jdbc-nt-11.2.0.2.0.zip (1,565,255 bytes)

*Instant Client Package - SQL*Plus: Additional libraries and executable for running SQL*Plus with Instant Client
instantclient-sqlplus-nt-11.2.0.2.0.zip (759,578 bytes)

*Instant Client Package - SDK: Additional header files and an example makefile for developing Oracle applications with Instant Client
instantclient-sdk-nt-11.2.0.2.0.zip (1,107,465 bytes)

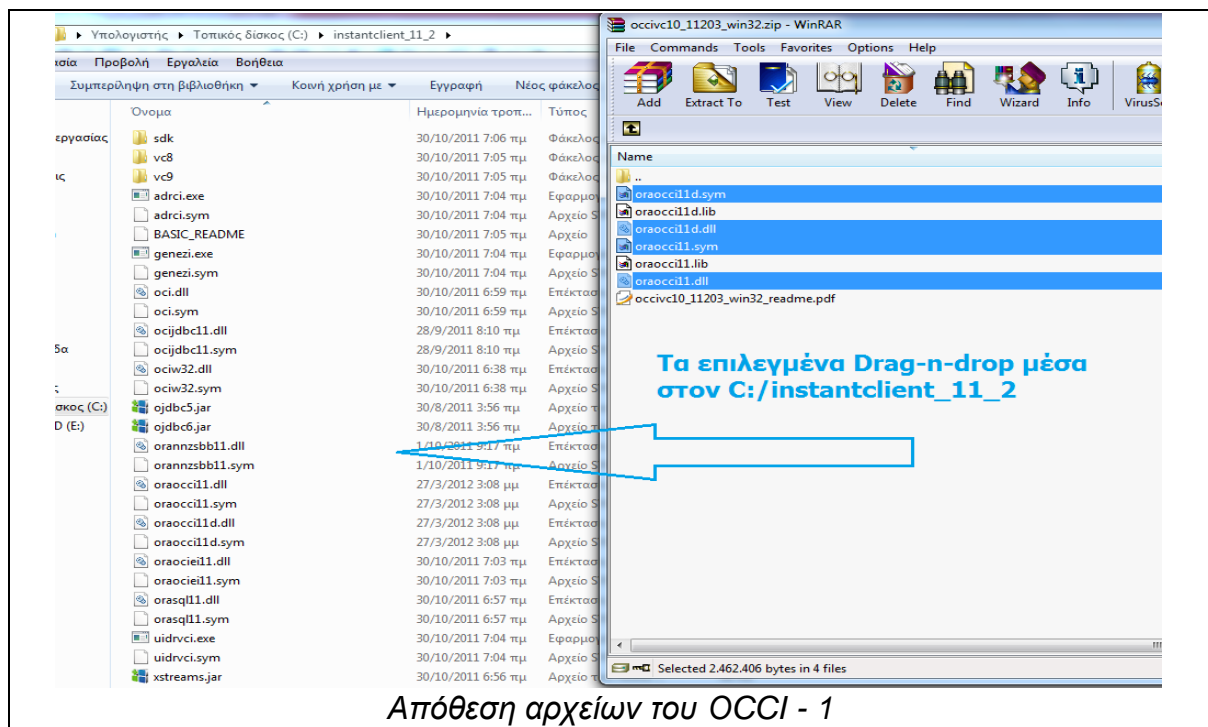
Οθόνη απόκτησης του Instant Client SDK

Στην συνέχεια, ανοίξτε το αρχείο ZIP του Instant Client SDK και μεταβείτε μέσα στον κατάλογο «instantclient_11_2» όπου θα βρείτε τον κατάλογο SDK. Με Drag-and-drop αποθέστε τον μέσα στον κατάλογο C:/instantclient_11_2 όπως διαφαίνεται παρακάτω:

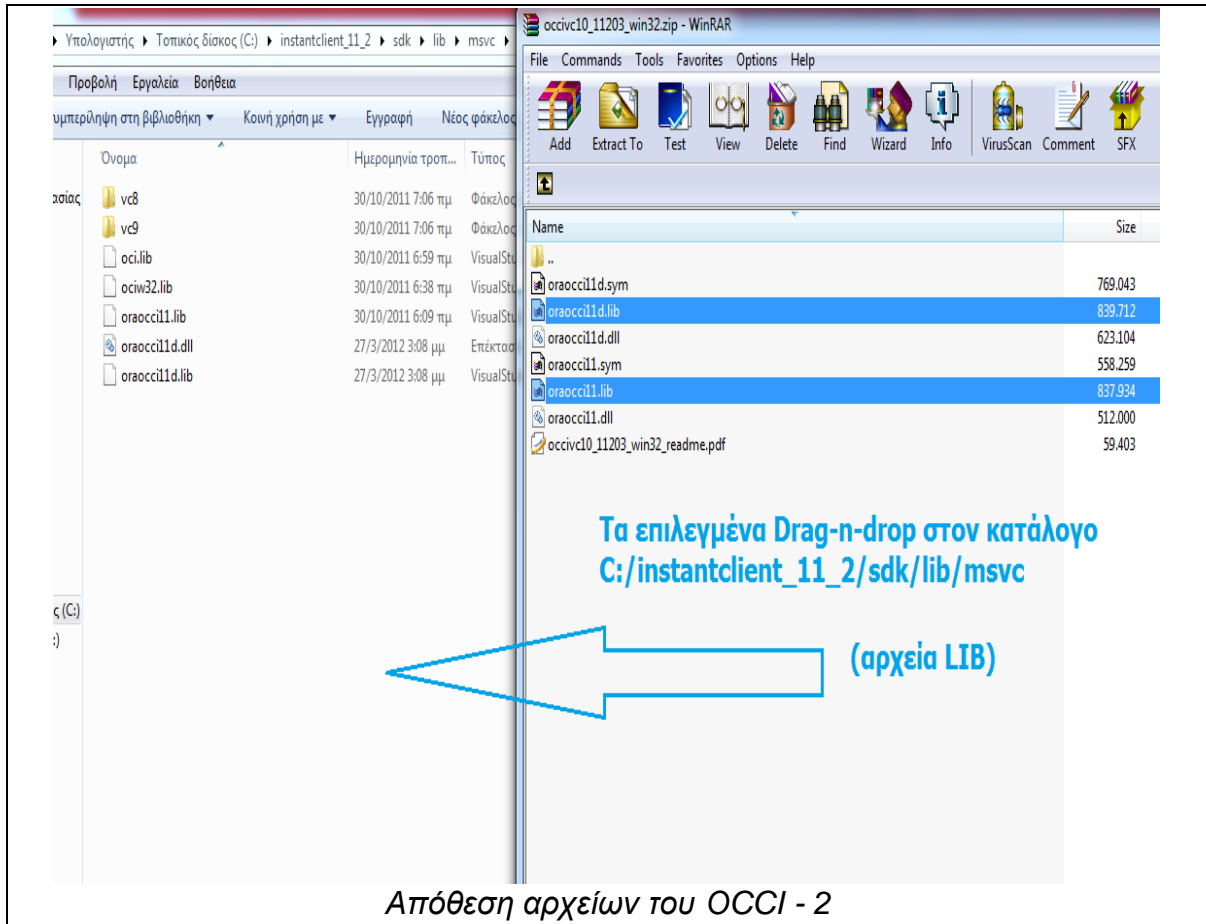


Στην συνέχεια ανοίξτε το ZIP αρχείο του OCCI MSVC 2010 και κάντε τα παρακάτω:

- 1) Τα DLL αρχεία τα κάνετε drag-n-drop στον κατάλογο C:/instantclient_11_2 όπως φαίνεται παρακάτω:



2) Στη συνέχεια μεταβείτε στον υποκατάλογο C:\instantclient_11_2\sdk\lib\msvc όπου θα κάνετε drag-n-drop τα εναπομείναντα lib αρχεία:



Τώρα πλέον το OCCI είναι έτοιμο για χρήση, όποτε δεν μένει παρά να αρχίσουμε να το χρησιμοποιούμε.

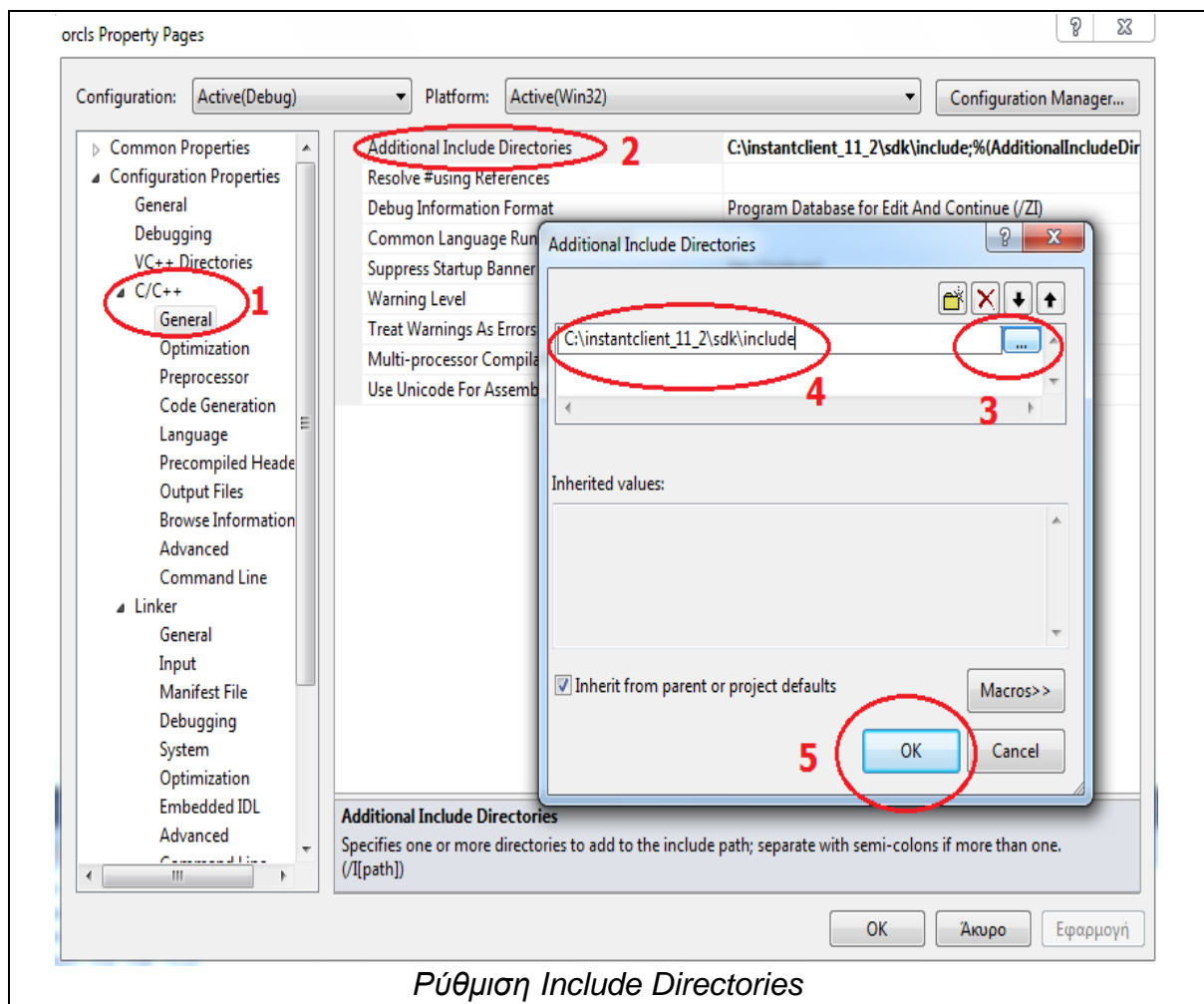
Συσχέτιση του OCCI με το Visual Studio 2010

Δημιουργήστε ένα νέο Project στην Visual C++ με όνομα της επιλογής σας (πχ occi_app). Αφού δημιουργηθεί, μεταβείτε στο μενού Project → <ονομα_project> Properties και στο παράθυρο που μας εμφανίζει πρέπει να

ρυθμίσουμε τα παρακάτω ώστε ο compiler και ο linker να βρουν ότι βιβλιοθήκη χρειάζεται (διαφορετικά το compilation δεν θα γίνει ποτέ).

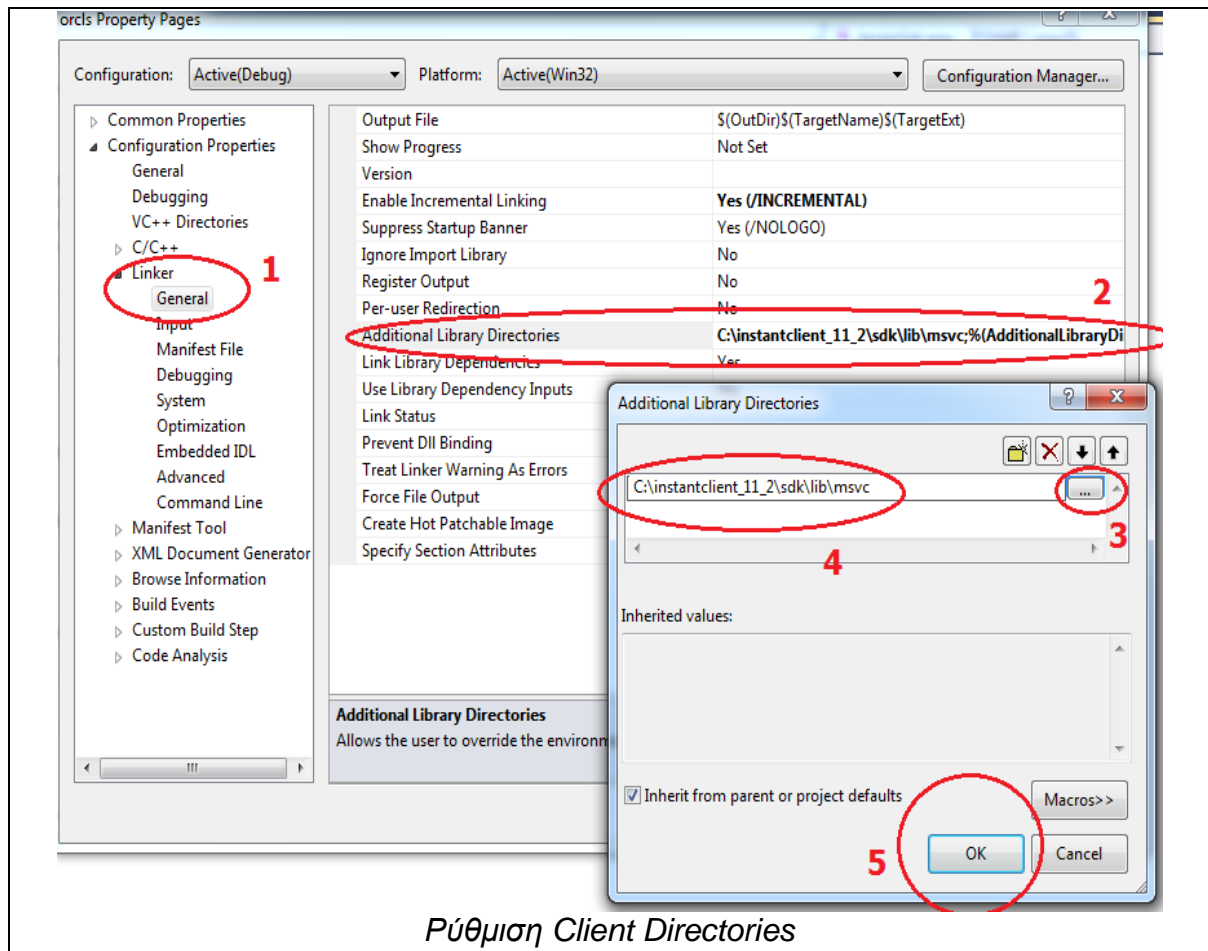
A) Ρύθμιση Include Path (για τα header files)

Μεταβείτε στην κατηγορία C++ → General (1), βρείτε την επιλογή Additional Include Directories (2) και πατήστε Edit για να σας εμφανίσει ένα παράθυρο ρυθμίσεων. Σε αυτό μεταβείτε στο λευκό πλαίσιο όπου θα βρείτε ένα κουμπί με τρεις τελείες (...) (3) και δώστε το path `c:\instantclient_11_2\sdk\include` (4) και πατήστε OK (5).



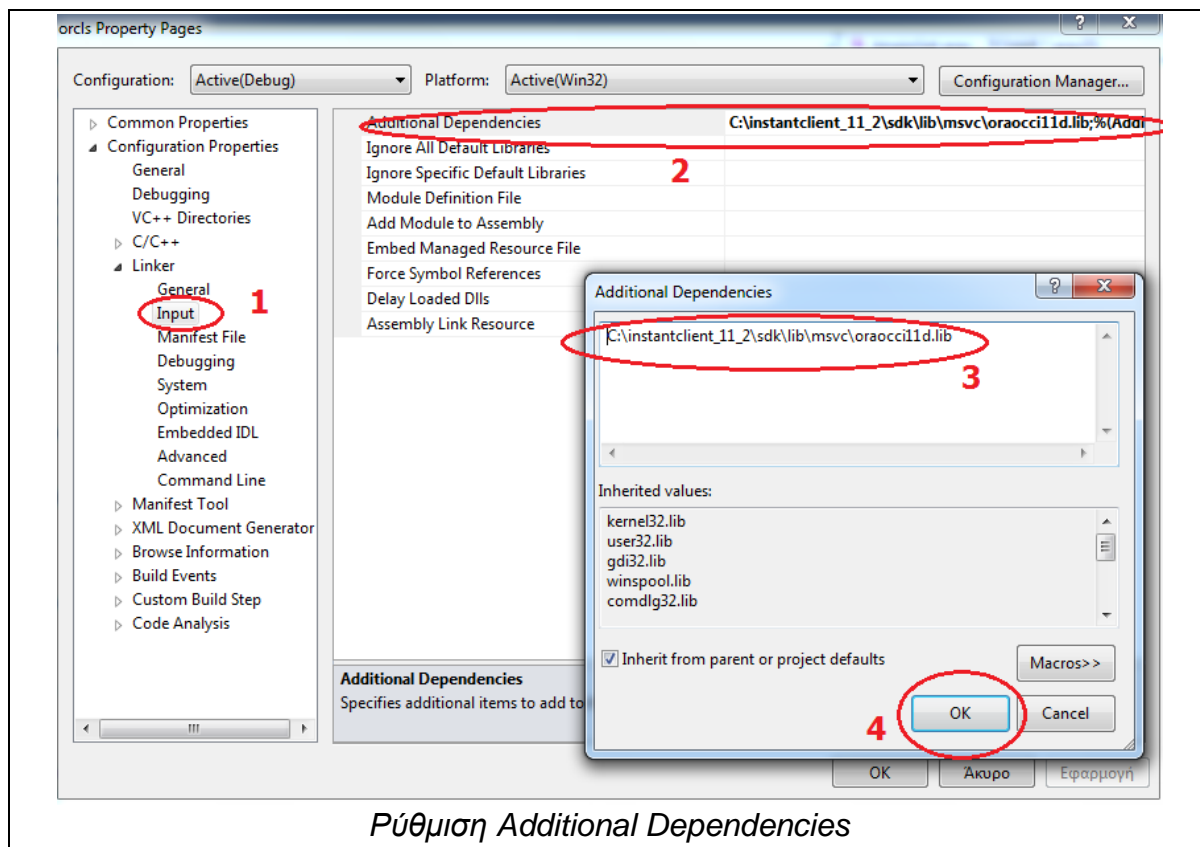
B) Ρύθμιση των Library Directories

Μεταβείτε στην κατηγορία Linker → General (1) και βρείτε την επιλογή Additional Library Directories (2) και πατήστε Edit. Στο παράθυρο που ανοίγει, πατήστε κλικ στο λευκό πλαίσιο ώστε να βγάλει το πλήκτρο με τις τρεις τελείες (3) δώστε το path C:\instantclient_11_2\sdk\lib\msvc (4) και πατήστε OK (5).



Γ) Ρύθμιση Additional Dependencies

Μεταβείτε στην κατηγορία Linker → Input (1) και βρείτε την επιλογή Additional Dependencies (2) και πατήστε EDIT. Στο παράθυρο που βγαίνει, στο λευκό πλαίσιο εισάγετε το path C:\instantclient_11_2\sdk\lib\msvc\oraocci11d.lib (για Debug Compilation) ή C:\instantclient_11_2\sdk\lib\msvc\ oraocci11.lib (για Release Compilation) (3). Προτείνετε για τις δοκιμές σας να βάλετε το Debug αρχείο. Τέλος πατήστε OK (4).



Από αυτό το σημείο και έπειτα μπορείτε να προβείτε στην συγγραφή του κώδικα της εφαρμογής σας.

Παράρτημα ΣΤ' – Κώδικας Ενδεικτικής Εφαρμογής OCCI

Παρακάτω παρατίθεται ο κώδικας της ενδεικτικής εφαρμογής που χρησιμοποιήσαμε στην ενότητα «Εφαρμογές σε Γλώσσα C++». Μπορείτε να τον εισάγετε στο Visual Studio αυτούσιο, αλλάζοντας όπου χρειάζεται τα credentials ή

```

/*
Written By Alagkiozidis Eleftherios
Redistribute Freely
*/
#include "stdafx.h"
#include "occi.h"
#include <iostream>
#include <string>
#include <cstdlib>

using namespace std;
using namespace oracle::occi;

int main(int argc, char* argv[])
{
    try{ //Exception may occur

        Environment* env = Environment::createEnvironment (Environment::DEFAULT);
//Create Environment
        Connection* conn = env->createConnection ("cpp",
"test123","//192.168.1.69:1521/test"); //Create Connection for user cpp with pass
//test123
        Statement* st = conn->createStatement ("select * from boats"); //Select Query
        ResultSet *rset = st->executeQuery (); //Result Set

        //SELECT
        cout << "SELECT * FROM boats \n\n" << endl; //Select notify
        while(rset->next()) //While result set finds new rows
            cout << rset->getInt(1) << "\t" << rset->getString(2) << endl; //print
//them

        //INSERT
        cout << "\n\nINSERT INTO BOATS(ID,NAME) VALUES (6, 'Sofia')" << endl;
//Insert Notify
        string sqlStmt = "INSERT INTO BOATS(ID,NAME) VALUES (:x, :y)"; //insert
//statement
        st=conn->createStatement (sqlStmt); //create statement
        st->setInt (1, 6); //set ID = 6
        st->setString (2, "Sofia"); //set name = Sofia
        st->executeUpdate (); //execute statement
        cout << "Row Inserted Successfully" << endl; //notify success
    }
}

```



```

//UPDATE
cout << "\n\nUPDATE BOATS SET NAME = 'Ag.Nikolaos' WHERE ID = 4 \n\n" <<
endl; //Update Notify
string upd = "UPDATE BOATS SET name = :x WHERE id = :y"; //Update statement
st = conn->createStatement (upd); //create statement

st->setString (1, "Ag.Nikolaos"); //set name Ag.Nikolaos
st->setInt (2, 4); //set ID = 4 (WHERE Clause)
st->executeUpdate (); //execute statement
cout << "Update OK " << endl; //notify success

//DELETE
cout << "\n\nDELETE FROM BOATS WHERE ID = 6 \n\n" << endl; //delete notify
string del = "DELETE FROM boats WHERE id = 6"; //delete statement (boat
//sofia)
st = conn->createStatement (del); //create statement
st->executeUpdate (); //execute statement
cout << "Row Deleted" << endl;

cout << endl;

env->terminateConnection(conn); //terminate connection
Environment::terminateEnvironment (env); //terminate environment

} //try
catch (SQLException sq) //Catch SQL Exceptions
{
    cerr << "exception " << sq.getMessage() << endl;
}
catch (exception& ex) //Catch Generic Exceptions
{
    cerr << "Exception " << ex.what() << endl;
}

system("pause");
return 0; //exit app
} //main

```

Παρακάτω παραθέτουμε τον SQL Κώδικα δημιουργίας του ενδίκτικού πίνακα που χρησιμοποιήσαμε με την εφαρμογή. Εισάγετε τον στο SQL Plus ή στο SQL Developer.

```
--CREATE A TABLE FOR CPP USER
```

```
--By Alagkiozidis Eleftherios - Petros
```

```
--elpalag@it.teithe.gr
```

```
--DROP TABLE BOATS;
```

```
CREATE TABLE BOATS
```

```
(ID INTEGER NOT NULL PRIMARY KEY,
```

```
NAME VARCHAR(30)
```

```
);
```

```
INSERT INTO BOATS(ID,NAME) VALUES (1,'Gorgoipikoos');
```

```
INSERT INTO BOATS(ID,NAME) VALUES (2,'Aksion_Esti');
```

```
INSERT INTO BOATS(ID,NAME) VALUES (3,'Theoskepasti');
```

```
INSERT INTO BOATS(ID,NAME) VALUES (4,'Gerasimos');
```

```
INSERT INTO BOATS(ID,NAME) VALUES (5,'Lady_Godiva');
```

Παράρτημα Ζ' – Εγκατάσταση και Παραμετροποίηση XAMPP και MySQL

Το XAMPP είναι μια **στοίβα πρωτοκόλλων** (Protocol Stack) η οποία προσφέρει Apache HTTP Server, MySQL, PHP και Perl σε ένα πακέτο. Έτσι αποφεύγεται η εγκατάσταση του καθενός ξεχωριστά. Εδώ παραθέτουμε πως θα το αποκτήσετε και πως θα το εγκαταστήσετε σε λειτουργικό σύστημα Debian 7 Wheezy x86_64.

Απόκτηση - Εγκατάσταση

Αρχικά, κατεβάστε το XAMPP από την παρακάτω σελίδα:

<http://www.apachefriends.org/download.php?xampp-linux-x64-1.8.2-2-installer.run>

Αφού ολοκληρωθεί το Download λογικά θα έχει αποθηκευτεί είτε στον κατάλογο /home/user/Downloads είτε στον κατάλογο /home/user. Όπου βρίσκεται λοιπόν, ανοίξτε ένα Terminal και συνδεθείτε ως χρήστες root. Στην συνέχεια εγκαθιστάτε το XAMPP με τις παρακάτω εντολές:

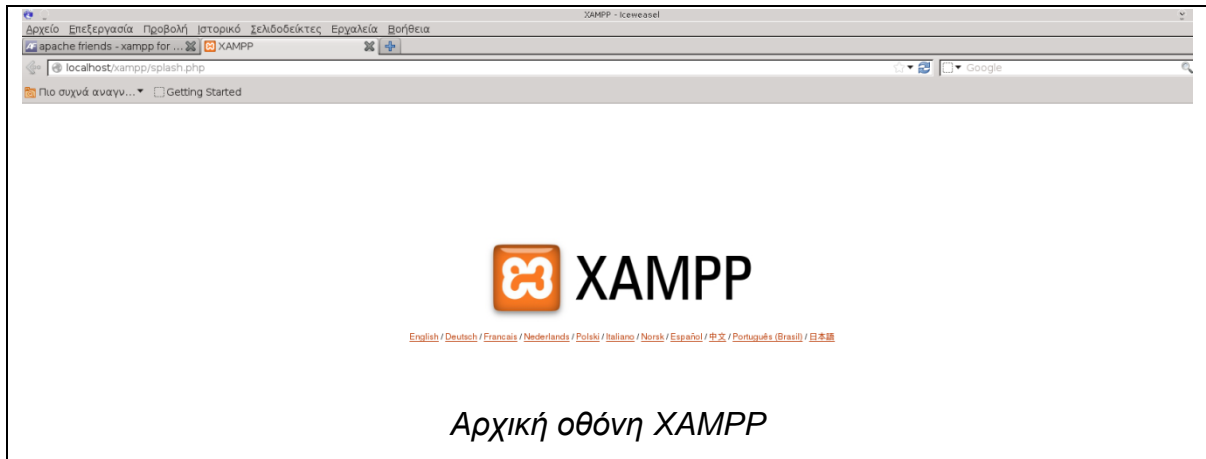
```
user@localhost$ su
root@localhost# cd /home/user/Downloads
root@localhost# ./xampp-linux-x64-1.8.2-2-installer.run
```

Εκκίνηση – Απενεργοποίηση

Αφού ολοκληρωθεί η εγκατάσταση, μπορείτε να **εκκινήσετε** το XAMPP με την παρακάτω εντολή:

```
root@localhost# /opt/lampp/lampp start
```

Για να **δοκιμάσετε** το XAMPP, ανοίξτε ένα Web Browser και πληκτρολογήστε σαν διεύθυνση τη λέξη «localhost» ή δώστε την διεύθυνση 127.0.0.1. Λογικά θα πάρετε την παρακάτω ένδειξη:



Αυτό πρακτικά σημαίνει ότι ο XAMPP λειτουργεί, ο Apache «ακούει» στην θύρα 80 ενώ η MySQL και η PHP ενεργοποιήθηκαν.

Για να απενεργοποιήσετε τον XAMPP, δώστε την παρακάτω εντολή (ως χρήστες root).

```
root@localhost# /opt/lampp/lampp stop
```

Μετά την εκτέλεση αυτής της εντολής, ο XAMPP απενεργοποιείται μαζί με όλα τα πρωτόκολλα που το συνοδεύουν.

Ανέβασμα Σελίδων

Για να μπορέσετε να ανεβάσετε αρχεία ιστοσελίδων, πρέπει αρχικά να κάνετε «προσβάσιμο» τον κατάλογο που θα τις περιέχει, ο οποίος είναι ο /opt/lampp/htdocs. Ως χρήστες root δώστε την παρακάτω εντολή:

```
root@localhost# chmod 755 /opt/lampp/htdocs
```

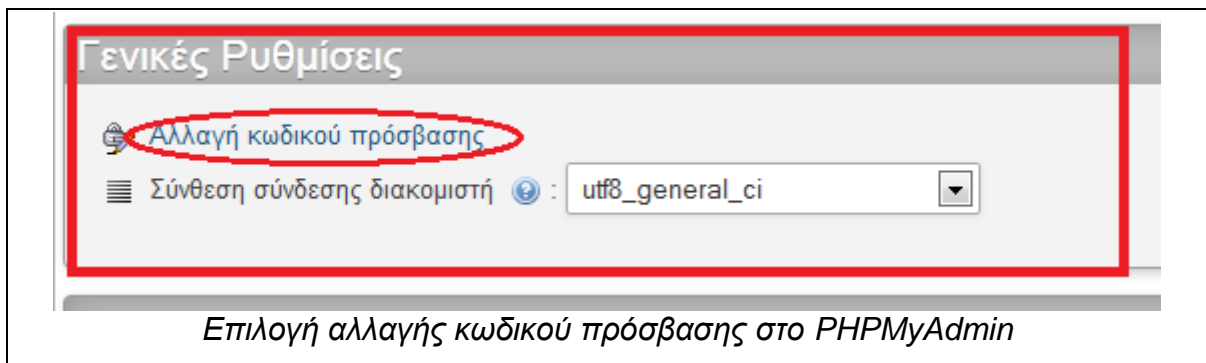
Μετά μπορείτε να ανεβάσετε αρχεία ιστοσελίδων (html, php) είτε τοπικά είτε μέσω κάποιου FTP Client όπως το FileZilla.

Παραμετροποίηση MySQL

Από προεπιλογή, η MySQL επιτρέπει την σύνδεση σε αυτήν χωρίς να απαιτεί Credentials. Για λόγους ασφαλείας όμως αυτό πρέπει να απενεργοποιηθεί. Την MySQL την διαχειρίζεστε με ένα Front End που παρέχει το XAMPP που λέγεται PHPMysqlAdmin. Μεταβαίνετε σε αυτό, από τον παρακάτω σύνδεσμο, **τοπικά από τον υπολογιστή - server**:

<http://localhost/phpmyadmin>

Αυτόματα συνδέεστε στη MySQL ως χρήστες root. Στο άνοιγμα της σελίδας του PHPMysqlAdmin θα βρείτε μια περιοχή «Γενικές Ρυθμίσεις» και ένα σύνδεσμο «Αλλαγή κωδικού πρόσβασης» όπως βλέπουμε στην παρακάτω εικόνα:



Πατήστε σε αυτό, και στην συνέχεια εισάγετε τον κωδικό πρόσβασης της επιλογής σας, επιλέγοντας το Radio Button «Κωδικός Πρόσβασης» είτε αφήστε το PHPMysqlAdmin να σας δημιουργήσει ένα Πατώντας στο κουμπί «Παραγωγή». Αφού τελειώσετε πατάτε «Εκτέλεση». Παρακάτω βλέπουμε την οθόνη αλλαγής κωδικού πρόσβασης:

Αλλαγή Κωδικού Πρόσβασης
✕

Χωρίς Κωδικό Πρόσβασης

Κωδικός πρόσβασης:

Επαναεισαγωγή:

Μείγμα κωδικού πρόσβασης: MySQL 4.1+

Συμβατό με MySQL 4.0

Δημιουργία κωδικού πρόσβασης

Οθόνη ρύθμισης κωδικού πρόσβασης στο PHPMyAdmin

Στην συνέχεια, πιθανών το PHPMyAdmin να σας φορτώσει μια σελίδα που αναφέρει σφάλμα. Αυτό γίνεται επειδή στα αρχεία ρυθμίσεων του PHPMyAdmin ο χρήστης Root δεν έχει ρυθμισμένο κωδικό πρόσβασης, ενώ ο τελευταίος δεν απαιτείται. Εσείς όμως ρυθμίσατε κωδικό πρόσβασης, οπότε τα αρχεία ρυθμίσεων πρέπει να τροποποιηθούν. Μεταβείτε στον κατάλογο /opt/lampp/phpmyadmin και βρείτε το αρχείο «config.inc.php» και ανοίξτε το (προτείνεται με το GNU Nano). Αλλάξτε τα attributes από τις παρακάτω γραμμές (τα αλλαγμένα με κόκκινο χρώμα):

```

/* Authentication type */

$cfg['Servers'][$i]['auth_type'] = 'http';

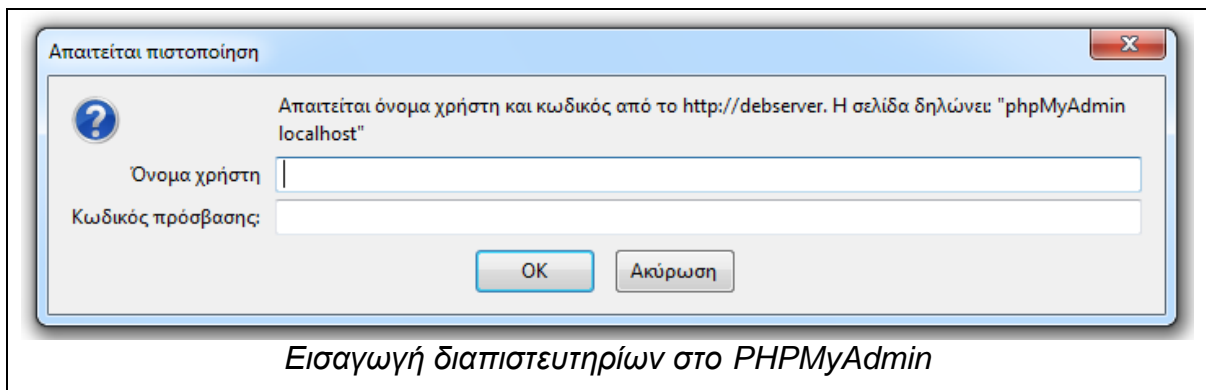
$cfg['Servers'][$i]['user'] = 'root';

$cfg['Servers'][$i]['password'] = '<to password σας>';

$cfg['Servers'][$i]['extension'] = 'mysqli';
    
```

```
$cfg['Servers'][$i]['AllowNoPassword'] = false;  
  
/*  
 * phpMyAdmin configuration storage settings.  
*/  
  
/* User used to manipulate with storage */  
  
// $cfg['Servers'][$i]['controlhost'] = '';  
  
$cfg['Servers'][$i]['controluser'] = 'pma';  
  
$cfg['Servers'][$i]['controlpass'] = '<password χρήστη pma>;'
```

Αφού αλλάξετε τα Attributes επισκεφτείτε ξανά την σελίδα του PHPMYAdmin όπου τώρα θα σας ζητάει Διαπιστευτήρια, όπως βλέπουμε παρακάτω:



Στην παρούσα οθόνη μπορείτε να συνδεθείτε ως χρήστες root με τον κωδικό που εισάγατε, αλλά πρώτα συνδεθείτε ως χρήστες «pma» (χωρίς password) και επαναλαμβάνοντας την διαδικασία αλλαγής κωδικού, αλλάξτε και τον κωδικό του χρήστη pma. Τώρα πλέον η MySQL και το PHPMYAdmin είναι έτοιμα για χρήση.

Παράρτημα Η' – Εγκατάσταση του E107 Website System

Σε αυτό το παράρτημα παραθέτουμε τον οδηγό εγκατάστασης του E107 Website System, στο οποίο βασίζεται η δομή της ιστοσελίδας του Server για το πρόγραμμα Advanced Computer Science.

Απόκτηση – Εγκατάσταση

Για να αποκτήσετε το E107 μεταβείτε στην παρακάτω ιστοσελίδα:

http://sourceforge.net/projects/e107/files/e107/e107%20v2.0%20alpha1/e107_2.0_full_alpha1.zip/download

Το CMS θα κατέβει σε μορφή zip. Ανοίξτε ένα Terminal, συνδεθείτε ως root και μεταβείτε στον κατάλογο που κατέβηκε (πχ /home/user/downloads) και δημιουργήστε ένα κατάλογο πχ mywebsite όπως παρακάτω:

```
root@localhost# cd /home/user/Downloads  
root@localhost# mkdir mywebsite
```

Στην συνέχεια μετακινήστε το αρχείο zip στον κατάλογο που μόλις φτιάξατε, και αποσυμπιέστε μέσα σε αυτόν τα περιεχόμενά του. Το αρχείο zip κρατήστε το για μελλοντική εγκατάσταση, απλά μετακινώντας το στον γονικό κατάλογο.

```
root@localhost# mv e107_2.0_full_alpha1.zip ./mywebsite  
root@localhost# cd ./mywebsite  
root@localhost# unzip e107_2.0_full_alpha1.zip  
root@localhost# mv e107_2.0_full_alpha1.zip ../  
root@localhost# cd ../
```

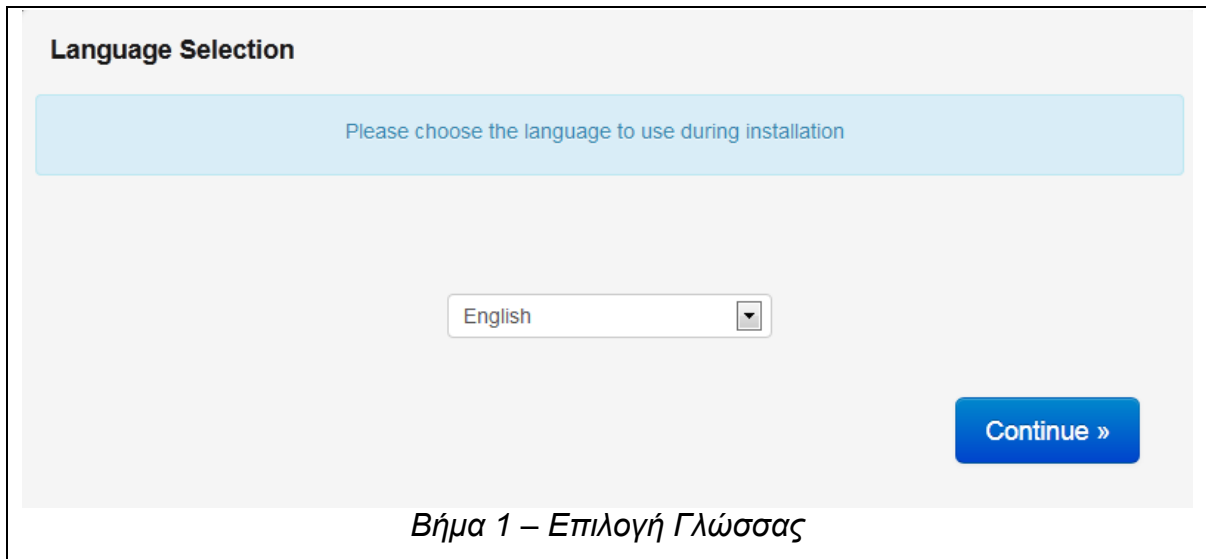

Στην συνέχεια πρέπει να μετακινήσουμε τον κατάλογο με τα περιεχόμενα του E107 στον κατάλογο htdocs του XAMPP ώστε να το «σηκώσουμε» στον Server.

```
root@localhost# cp -r ./mywebsite /opt/lampp/htdocs
```

Αφού λοιπόν μετακινήσουμε τον κατάλογο στο htdocs και βεβαιωθούμε πως ο XAMPP είναι **ενεργός**, τότε ανοίγουμε ένα web browser και μεταβαίνουμε στην διεύθυνση:

<http://localhost/mywebsite>

Η διαδικασία εγκατάστασης ξεκινάει. Το πρώτο βήμα που καλούμαστε να πραγματοποιήσουμε είναι η επιλογή γλώσσας όπως βλέπουμε παρακάτω:



Language Selection

Please choose the language to use during installation

English

Continue »

Βήμα 1 – Επιλογή Γλώσσας

Στην συνέχεια καλούμαστε να ρυθμίσουμε τις παραμέτρους για την MySQL, επειδή το E107 χρησιμοποιεί την MySQL για αποθήκευση του περιεχομένου του. Αρχικά ρυθμίζετε την διεύθυνση του MySQL Server σας, που εφόσον είναι τοπικός, εισάγετε localhost (1). Στην συνέχεια παραθέστε τα Credentials σας ως χρήστες Root (2 και 3) στον MySQL Server. Θα πρέπει να δημιουργήσουμε μια νέα βάση δεδομένων για τον E107, οπότε δώστε ένα όνομα βάσης πχ

e107_database και επιλέξτε το Check-box «Create Database» (4). Τέλος πατήστε «Continue» (5).

The screenshot shows the 'MySQL Server Details' form. It contains the following fields and annotations:

- MySQL Server:** A text input field containing 'localhost', highlighted with a red box and the number '1'.
- MySQL Username:** An empty text input field, highlighted with a red box and the number '2'.
- MySQL Password:** An empty password input field, highlighted with a red box and the number '3'.
- MySQL Database:** An empty text input field followed by a checkbox labeled 'Create Database?'. The checkbox is checked, and the entire field is highlighted with a red box and the number '4'.
- Table prefix:** A text input field containing 'e107_'.
- Buttons:** At the bottom right, there are two buttons: a grey '« Back' button and a blue 'Continue »' button. The 'Continue »' button is highlighted with a red box and the number '5'.

Βήμα 2 – Ρυθμίσεις MySQL

Αν ο SQL Server «ακούει» και είναι up τότε στην επόμενη σελίδα θα δείτε επιβεβαίωση, διαφορετικά θα σας ειδοποιήσει ότι κάτι δεν πάει καλά. Πατήστε Continue για να μεταφερθείτε στην επόμενη σελίδα.

Η επόμενη σελίδα, που αποτελεί το βήμα 3, σας ειδοποιεί πως πρέπει να αλλάξετε τα file permissions ορισμένων αρχείων και καταλόγων του E107 ώστε να προχωρήσει η εγκατάσταση. Παρακάτω σας δείχνουμε τις εντολές με τα αντίστοιχα αρχεία που ζητάνε αλλαγή Permissions (user root)

```
root@localhost# cd /opt/lampp/htdocs/mywebsite
```

```
root@localhost# chmod 777 e107_config.php
```

```
root@localhost# chmod 777 e107_media
```

```
root@localhost# chmod 777 e107_system
root@localhost# chmod 777 e107_plugins
root@localhost# chmod 777 e107_themes
```

Πατήστε στο πλήκτρο «Retest File Permissions» και όλα θα μαρκαριστούν ως «Pass». Τώρα πατήστε «Continue».

Στην επόμενη σελίδα, καλούμαστε να ρυθμίσουμε τις παραμέτρους του διαχειριστή της σελίδας. Αρχικά δίνουμε το username του διαχειριστή (1). Κατόπιν εισάγουμε το όνομα που θα βλέπουν οι χρήστες για τον διαχειριστή (2). Εν συνεχεία εισάγουμε και επιβεβαιώνουμε των κωδικό πρόσβασης του διαχειριστή (3 και 4) και **προορατικά** εισάγουμε και κάποιο email διαφορετικά εισάγετε το admin@mysite.com (5). Τέλος πατάμε «Continue».

Administrator Details	
Admin Username	<input type="text" value="admin"/> 1
Admin Display Name	<input type="text" value="Administrator"/> 2
Admin Password	<input type="password"/> 3
Admin Password Confirmation	<input type="password"/> 4
Admin Email	<input type="text" value="admin@mysite.com"/> 5
<input type="button" value="« Back"/> <input type="button" value="Continue »"/> 6	

Βήμα 4 - Εισαγωγή στοιχείων δημιουργίας διαχειριστή

Στην επόμενη σελίδα ρυθμίζουμε το όνομα της σελίδας μας (1) και επιλέγουμε να χρησιμοποιήσουμε το προεγκατεστημένο θέμα (theme) στη σελίδα μας(2). Τέλος πατάμε «Continue» (3).

Website Preferences

Website Name **1**

Website Theme

Theme Name	Theme Type
<input checked="" type="radio"/> bootstrap 2	generic

Install Plugins

Include Content/Configuration

« Back **3** Continue »

Βήμα 5 - Επιλογή ονόματος σελίδας και θέματος

Στις επόμενες σελίδες ολοκληρώνεται η εγκατάσταση του E107, απλά πατήστε και στις δύο σελίδες «Continue». Για να ολοκληρωθεί όμως πλήρως η εγκατάσταση πρέπει να αλλάξουμε τα file permissions των αρχείων της σελίδας, του αρχείου e107_config.php και να διαγράψουμε για λόγους ασφαλείας το αρχείο install.php. Το κάνουμε με τις κάτωθι εντολές:

```
root@localhost# cd /opt/lampp/htdocs/mywebsite
```

```
root@localhost# chmod 644 e107_config.php
```

```
root@localhost# chmod 755 ./*
```

```
root@localhost# rm install.php
```

Η εγκατάσταση του E107 ολοκληρώθηκε επιτυχώς. Πλέον μπορείτε να επισκεφτείτε την σελίδα στην διεύθυνση:

<http://localhost/mywebsite>

Για την διαχείριση της σελίδας μεταβαίνετε στον σύνδεσμο:

http://localhost/mywebsite/e107_admin

Παράρτημα Θ´ – Script δημιουργίας / διαγραφής χρηστών Oracle και αρχείου αναφοράς

Σε αυτό το παράρτημα παραθέτουμε τον κώδικα του Bash Shell Script για την δημιουργία και διαγραφή των χρηστών της Oracle 11g και του αρχείου αναφοράς για τα υπόλοιπα Scripts (user_generate.sh). Δημιουργήστε τον κατάλογο **main** στον κατάλογο που βρίσκεται το αρχείο script.

```
#!/bin/bash

#Create Users for the Oracle DBMS

#By Alagkiozidis Eleftherios-Petros

#Execute script with the desired number of users as an argument

#it will generate the sql file, which you can execute from SQL*PLUS or sql
developer

#also generates a text file with the usernames and passwords in 2 columns (for
#information purposes, or distribution)

if [ "$#" -lt 1 ] #Έλεγχος για όρισμα

    then # Αν δεν υπάρχει όρισμα

        echo "SYNTAX: cr.sh <number of users>" #δείξε πως συντάσσεται

        echo "No args specified, exiting"; #ενημέρωσε για σφάλμα

        exit 1; #Και τερμάτισε με κωδικό 1

elif [ "$#" -gt 1 ] #Έλεγχος για περισσότερα από 1 ορίσματα

    then #Αν είναι περισσότερα

        echo "Too many args, exiting"; #Ενημέρωσε για σφάλμα

        exit 1; #Τερμάτισε με κωδικό 1
```

```

fi #Τέλος IF

export LIMIT=$1 #Μεταβλητή που κρατάει το όρισμα (πλήθος χρηστών)

export TIMESTAMP=`date` # Μεταβλητή που κρατάει τη χρονοσφραγίδα

touch ./main/users.sql #Δημιουργία αρχείου user.sql στον κατάλογο main

touch ./main/del_users.sql # Δημιουργία αρχείου del_users στον main

touch ./user_pass.txt # Δημιουργία αρχείου user_pass.txt στον τρέχον κατάλογο

echo "--SQL Code generated by cr.sh script at $TIMESTAMP" > ./main/users.sql

# Εγγραφή (ή πανωγραφή αν υπάρχει) χρονοσφραγίδας στο users.sql

echo "--SQL Code generated by cr.sh script at $TIMESTAMP" >
./main/del_users.sql

# Το ίδιο στο αρχείο del_users.sql

echo "Usernames and Passwords, generated at $TIMESTAMP" > user_pass.txt

echo "" >> ./main/users.sql #Κενή γραμμή στο αρχείο

echo "" >> ./main/del_users.sql #ομοίως

echo "" >> user_pass.txt #ομοίως

for((i=0;i<LIMIT;i++ )) #Επανάληψη για όσους χρήστες επιλέξαμε στο arg

do

    RAND=`echo $RANDOM % 1000000 + 10000 | bc`;

    #Παραγωγή τυχαίου πενταψήφιου

    echo "CREATE USER dblab_${i} IDENTIFIED BY dbuser_${i}_${RAND};" >>
./main/users.sql

#Εγγραφή CREATE USER <dblab_??> IDENTIFIED BY <dbuser_??> στο αρχείο
#users.sql (δημιουργία χρήστη με password)

```

```
echo "GRANT CREATE SESSION, CREATE TABLE, CREATE VIEW,  
CREATE TYPE, CREATE PROCEDURE TO dblab_$$;" >> ./main/users.sql  
  
#Εγγραφή εντολής εκχώρησης δικαιωμάτων δημιουργίας συνεδρίας, πινάκων,  
#όψεων, τύπων, precedures στο χρήστη που μόλις φτιάξαμε, στο ίδιο αρχείο  
  
echo "ALTER USER dblab_$$ QUOTA 100M on USERS;" >>  
./main/users.sql  
  
#Εγγραφή εντολής εκχώρησης χώρου 100 στο Tablespace Users, στο ίδιο αρχείο  
  
echo "DROP USER dblab_$$ CASCADE;" >> ./main/del_users.sql  
  
#Εγγραφή εντολής διαγραφής χρήστη μαζί με τα αντικείμενά του, στο αρχείο  
#del_users.sql  
  
echo "dbl_$$ dbuser_$$_$$RAND" >> user_pass.txt  
  
#Δημιουργία δίστηλου πληροφοριακού αρχείου αναφοράς για τα υπόλοιπα  
#scripts και για διανομή  
  
done; #Τέλος Loop  
  
echo "All Done!" #Μήνυμα επιβεβαίωσης τέλους  
  
exit 0; #Εξοδος με κωδικό 0 (όλα OK)
```

Εκτέλεση Shell Script

Για να εκτελεστεί το Script θα πρέπει να γίνει εκτελέσιμο. Αλλάξτε τα File Permissions με την chmod (ως oracle).

```
oracle@localhost$ chmod 755 ./user_generate.sh
```

Εκτελέστε το Script ως εξής (με όρισμα τους χρήστες που θέλετε να παράγει):

```
#Syntax ./user_generate.sh <αριθμός χρηστών>
```

```
oracle@localhost$ ./user_generate.sh 100
```


Στον κατάλογο main δημιουργούνται τα 2 SQL Scripts (users.sql και del_users.sql) ενώ στον τρέχον κατάλογο, δημιουργείται το αρχείο user_pass.txt το οποίο θα το χρησιμοποιήσετε είτε για την εκτέλεση των άλλων Shell Scripts είτε για διανομή κωδικών στους χρήστες (πληροφοριακά).

Εκτέλεση SQL Scripts

Τα SQL Scripts θα τα εκτελέσετε μέσα από το SQL*PLUS ως χρήστες SYS as SYSDBA, με το σύμβολο @ και το path προς το script ως εξής (παράδειγμα τοποθεσίας script /home/oracle/oracle_scripts/main)

```
SQL> @/home/oracle/oracle_scripts/users.sql --ή user_del.sql
```

Παράρτημα Ι΄ – Script Δημιουργίας / Διαγραφής Χρηστών του Oracle APEX

Παραθέτουμε τον κώδικα δημιουργίας και διαγραφής χρηστών για την πλατφόρμα Oracle APEX. Για να τρέξει το Script δημιουργήστε τον κατάλογο apex στον κατάλογο όπου θα το εκτελέσετε, και φροντίστε στον κατάλογο αυτό να υπάρχει διαθέσιμο το αρχείο «user_pass.txt» γιατί το χρησιμοποιεί σαν αναφορά.

```
#!/bin/bash

#Apex User Creation and Removal Script, by Alagkiozidis Eleftherios"

#Make sure file user_pass.txt exist on the directory this script resides

#Make sure the schema of your workspace has been granted with
APEX_ADMINISTRATOR_ROLE (by sysdba admin)

#Run this script with no args

export TIME_STAMP=`date`;

#Αποθήκευση τρέχουσας ημερομηνίας και ώρας στην TIME_STAMP

echo "--The Exported SQL SCRIPT (apex_create.sql) need to be run from apex
workspace admin interface, not SQLPLUS or SQL Developer" >
./apex/apex_create.sql

#Μήνυμα προειδοποίησης και δημιουργία (αν δεν υπάρχει) του αρχείου του SQL
#Script (αν υπάρχει, πανωγράφεται)

echo "--Created on $TIME_STAMP" >> ./apex/apex_create.sql

#Χρονοσφραγίδα στο αρχείο

echo "--The exported SQL SCRIPT (apex_remove.sql) need to be run from apex
workspace admin interface, not SQLPLUS or SQL Developer" >
./apex/apex_remove.sql

#Δημιουργία / πανωγραφή του Δεύτερου SQL Script όμοια με το πρώτο

echo "--Created on $TIME_STAMP" >> ./apex/apex_remove.sql
```

```

#Χρονοσφραγίδα στο δεύτερο αρχείο

export FILE_SIZE=`cat user_pass.txt | wc -l` #GET FILE SIZE

#Αποθήκευση αριθμού γραμμών από το αρχείο user_pass.txt (γραμμή =
#χρήστης)

for((i=3;i<FILE_SIZE+1;i++)); #Για όσες είναι οι γραμμές που διάβασες...

do

export ORA_USR=`cat user_pass.txt | sed -n -e "$i"p | awk '{print $1}' | tr '[a-z]'
'[A-Z]'`;

#Παίρνει την πρώτη στήλη του αρχείου και την μετατρέπει σε κεφαλαία, και την
#αποθηκεύει στην ORA_USR

export ORA_PASS=`cat user_pass.txt | sed -n -e "$i"p | awk '{print $2}'`;

#Παίρνει την δεύτερη στήλη του αρχείου user_pass.txt όπως είναι και την
#αποθηκεύει στο ORA_PASS

#Εντολές δημιουργίας Developer χρήστη (apex_create.sql)

#Κλήση PL/SQL Procedure APEX_UTIL.CREATE_USER σε ανώνυμο PL/SQL
#Block

echo "Generating user $ORA_USR ....."

echo "--USER $ORA_USR" >> ./apex/apex_create.sql

echo "BEGIN" >> ./apex/apex_create.sql

echo "APEX_UTIL.CREATE_USER(" >> ./apex/apex_create.sql

echo "p_user_name => '$ORA_USR'," >> ./apex/apex_create.sql

echo "p_description => 'student'," >> ./apex/apex_create.sql

echo "p_email_address => 'apex@it.teithe.gr'," >> ./apex/apex_create.sql

echo "p_web_password => '$ORA_PASS'," >> ./apex/apex_create.sql

```

```

echo "p_developer_privs                                =>
'CREATE:DATA_LOADER:EDIT:HELP:MONITOR:SQL',"          >>
./apex/apex_create.sql

echo "p_default_schema => '$ORA_USR'," >> ./apex/apex_create.sql

echo "P_allow_access_to_schemas => '$ORA_USR'," >> ./apex/apex_create.sql

echo "p_change_password_on_first_use => 'N');" >> ./apex/apex_create.sql

echo "END;" >> ./apex/apex_create.sql #κλείσιμο PL/SQL Block

echo "-- end of User $ORA_USR" >> ./apex/apex_create.sql

echo "/" >> ./apex/apex_create.sql #Διαχωριστής PL/SQL Block

echo ""

#Εντολές διαγραφής χρήστη (apex_remove.sql)

#Κλήση PL/SQL Procedure APEX_UTIL.REMOVE_USER σε ανώνυμο PL/SQL
#Block

echo "BEGIN" >> ./apex/apex_remove.sql

echo "APEX_UTIL.REMOVE_USER(p_user_name => '$ORA_USR');" >>
./apex/apex_remove.sql

echo "END;" >> ./apex/apex_remove.sql #κλείσιμο PL/SQL Block

echo "/" >> ./apex/apex_remove.sql #Διαχωριστής PL/SQL Block

echo ""

done #Τέλος Loop

echo "-----";

echo "All Done"; #Αναγγελία ολοκλήρωσης

```

```
echo "-----";  
exit 0; #Ολοκλήρωση και έξοδος
```

Εκτέλεση Shell Script

Εκτελέστε το Script αφού έχετε δημιουργήσει τον κατάλογο Main και αφού βεβαιωθείτε ότι υπάρχει στον τρέχοντα κατάλογο το αρχείο «user_pass.txt» που δημιούργησε το Script «user_generate.sh». Εκτελέστε το όπως είναι χωρίς ορίσματα (ως Oracle) αφού του αλλάξετε τα File Permissions σε 755.

```
oracle@localhost# mkdir apex  
  
oracle@localhost$ chmod 755 apex_generate.sh  
  
oracle@localhost$ ./apex_generate
```

Το Script θα παράγει δύο αρχεία, το «apex_create.sql» και το «apex_remove.sql». **Για να τρέξετε τα SQL Scripts**, πρέπει πρώτα να εκχωρήσετε στον χρήστη/schema του Workspace σας (πχ Students) τον ρόλο **APEX ADMINISTRATOR ROLE** με το SQL*PLUS (ως χρήστες SYS as SYSDBA)

```
--Syntax: GRANT APEX_ADMINISTRATOR_ROLE TO <όνομα χρήστη/schema  
του Workspace>
```

```
SQL> GRANT APEX_ADMINISTRATOR_ROLE TO STUDENTS;
```

Εκτέλεση SQL Script

Τα παραγόμενα SQL Scripts τρέχουν μέσα από το περιβάλλον του Oracle APEX. Συνδεθείτε ως διαχειριστής Workspace και κάντε Upload τα αρχεία στο APEX με το εργαλείο Script Editor του SQL Workshop. Με το ανέβασμα θα μείνουν στο APEX για να τα εκτελείτε όποτε θέλετε (χρησιμοποιήστε Internet Explorer).

Παράρτημα ΙΑ´ - Script Συσχέτισης Schema με ένα APEX Workspace

Στο παράρτημα αυτό παραθέτω το Script συσχέτισμού των Schemas των χρηστών της βάσης με το Workspace. Εκτελείται τρίτο στην σειρά.

```
#!/bin/bash

#Assign a schema to a workspace

#The workspace name is given as an an argument

#by Alagkiozidis Eleftherios - elpalag@it.teithe.gr

#Έλεγχος ορισμάτων (ακριβώς ένα – το όνομα του Workspace)

if [ "$#" -lt 1 ] #Αν είναι λιγότερα από 1 , τότε ...

    then

        echo "Syntax: apex_workspace.sh <workspace_name>"

        echo "No args specified, exiting";

        exit 1; #Εμφάνισε σωστή σύνταξη και κλείσε με σφάλμα 1

elif [ "$#" -gt 1 ] #Αν είναι περισσότερα από 1

    then

        echo "Too many args, exiting";

        exit 1; #Εμφάνισε σφάλμα και τερμάτισε με σφάλμα 1

fi #Τέλος IF
```

```

APEX_WSPACE=$1 #Λήψη ορίσματος χρήστη

TIMESTAMP=`date` #Λήψη τρέχουσας χρονοσφραγίδας

FILE_SIZE=`cat user_pass.txt | wc -l` #μέτρηση γραμμών αρχείο user_pass.txt

#Δημιουργία αρχείων Script συσχέτισης και αποσυσχέτισης

touch ./apex/apex_wspace_schema.sql

touch ./apex/apex_remove_schema.sql;

echo "--Assign schemas to workspace $APEX_WSPACE" >
./apex/apex_wspace_schema.sql;

#Εγγραφή/Πανωγραφή αρχείου με πληροφορίες (Script 1)

echo "--Remove schemas from workspace $APEX_WSPACE" >
./apex/apex_remove_schema.sql;

#Εγγραφή/Πανωγραφή αρχείου με πληροφορίες (Script 2)

echo "--GENERATED AT : $TIMESTAMP" >> ./apex/apex_wspace_schema.sql;

echo "--GENERATED AT : $TIMESTAMP" >> ./apex/apex_remove_schema.sql;

#Εγγραφή χρονοσφραγίδας στα δύο αρχεία

echo "" >> ./apex/apex_wspace_schema.sql;

echo "" >> ./apex/apex_remove_schema.sql;

#Κενές γραμμές και στα δύο αρχεία

#Παραγωγή SQL Statements για συσχέτιση/αποσυσχέτιση

for((i=3;i<$FILE_SIZE+1;i++)); #Για όσες είναι οι γραμμές του user_pass.txt
do

echo "BEGIN" >> ./apex/apex_wspace_schema.sql;

echo "BEGIN" >> ./apex/apex_remove_schema.sql;

```

```

#Δημιουργία Ανώνυμων PL/SQL Blocks και στα δύο αρχεία (BEGIN)

export ORA_USR=`cat user_pass.txt | sed -n -e "$i"p | awk '{print $1}' | tr 'a-z'
'[A-Z]`; #Λήψη USERNAMES (στήλη 1) από user_pass.txt

echo
"APEX_INSTANCE_ADMIN.ADD_SCHEMA('$APEX_WSPACE','$ORA_USR');"
>> ./apex/apex_wspace_schema.sql;

#Κλήση PL/SQL Procedure APEX_INSTANCE_ADMIN.ADD_SCHEMA

#Για συσχέτιση Schema χρήστη με το Workspace που δώσαμε σαν όρισμα

echo
"APEX_INSTANCE_ADMIN.REMOVE_SCHEMA('$APEX_WSPACE','$ORA_USR
');" >> ./apex/apex_remove_schema.sql;

#Κλήση PL/SQL Procedure APEX_INSTANCE_ADMIN.REMOVE_SCHEMA

#Για απόσυσχέτιση Schema χρήστη με το Workspace που δώσαμε σαν όρισμα

echo "END;" >> ./apex/apex_wspace_schema.sql;

echo "END;" >> ./apex/apex_remove_schema.sql;

#Κλείσιμο PL/SQL Blocks (END;) και στα δύο αρχεία

echo "/" >> ./apex/apex_wspace_schema.sql; #Διαχωριστής Blocks

echo "/" >> ./apex/apex_remove_schema.sql; #Διαχωριστής Blocks

done #Τέλος Loop

echo "-----";

echo "All Done"; #Αναγγελία Ολοκλήρωσης

echo "-----";

exit 0; #Ολοκλήρωση και έξοδος

```


Εκτέλεση Shell Script

Το Script αυτό το εκτελείτε τρίτο στην σειρά, αφού έχετε παράγει χρήστες με το «user_generate.sh» και το «apex_generate.sh». Επίσης θα χρειαστείτε ως πηγή, το αρχείο «user_pass.txt».

Αλλάξτε του τα File Permissions, και εκτελέστε το δίνοντας ως όρισμα το όνομα του Workspace που θέλετε να συσχετήσετε / αποσυσχετίσετε με τα Schemas των χρηστών.

```
oracle@localhost$ chmod 755 ./apex_workspace.sh  
#Syntax: ./apex_workspace.sh <όνομα workspace>  
oracle@localhost$ ./apex_workspace.sh students
```

Το Script θα παράγει τα SQL Scripts «apex_wspace_schema.sql» και «apex_remove_schema.sql».

Εκτέλεση SQL Scripts

Για να εκτελέσετε τα SQL Scripts, θα χρειαστείτε το SQL*PLUS ή το SQL Developer. Συνδεθείτε ως χρήστης SYS as SYSDBA και εκτελέστε τα

```
SQL> @/home/oracle/oracle_scripts/apex/apex_wspace_schema.sql
```

Παράρτημα IB´ - Script Δημιουργίας χρηστών για την PostgreSQL

Στο τελευταίο αυτό παράρτημα σας παραθέτω των κώδικα του Script για την δημιουργία χρηστών για την PostgreSQL.

```
#!/bin/bash

#PostgreSQL Users Generator Script

#By Alagkiozidis Eleftherios-Petros

#Copyright 2013

export FILE_SIZE=`cat user_pass.txt | wc -l`

#Καταμέτρηση γραμμών αρχείο user_pass.txt

export TIME_STAMP=`date`

#Λήψη χρονοσφραγίδας

touch ./postgres/pg_create.sql

touch ./postgres/pg_delete.sql

#Δημιουργία αρχείων SQL Scripts

echo "--Script Generated at $TIME_STAMP" > ./postgres/pg_create.sql

echo "--Run this script as user Postgres on psql command line" >>
./postgres/pg_create.sql

#Εγγραφή (ή πανωγραφή στο πρώτο) Χρονοσφραγίδας και πληροφοριών στο
#πρώτο SQL Script αρχείο

echo "" >> ./postgres/pg_create.sql

echo "--Script Generated at $TIME_STAMP" > ./postgres/pg_delete.sql
```

```
echo "--Run this script as user Postgres on psql command line" >>
./postgres/pg_delete.sql

#Εγγραφή (ή πανωγραφή στο πρώτο) Χρονοσφραγίδας και πληροφοριών στο
#δεύτερο SQL Script αρχείο

for((i=3;i<$FILE_SIZE+1;i++)); #Για όσες είναι οι γραμμές του αρχείου αναφοράς
do

export PG_USR=`cat user_pass.txt | sed -n -e "$i"p | awk '{print $1}' | tr '[a-z]' '[A-
Z]`;

#Λήψη στήλης usernames (σε κεφαλαία)

export PG_PASS=`cat user_pass.txt | sed -n -e "$i"p | awk '{print $2}`;

#Λήψη στήλης Passwords (ως έχουν)

#Εντολές δημιουργίας χρήστη PostgreSQL (pg_create.sql)

echo "Generating user $PG_USR ...."

echo "--User $PG_USR ...." >> ./postgres/pg_create.sql

#Αναγγελία δημιουργίας τρέχοντος χρήστη...

echo "CREATE USER $PG_USR WITH PASSWORD '$PG_PASS' ;" >>
./postgres/pg_create.sql

#Δημιουργία χρήστη με password

echo "CREATE SCHEMA sch_$PG_USR AUTHORIZATION $PG_USR ;" >>
./postgres/pg_create.sql

#Δημιουργία Schema για τον χρήστη (μορφή sch_όνομα_χρήστη)

echo "ALTER USER $PG_USR SET SEARCH_PATH=sch_$PG_USR ; " >>
./postgres/pg_create.sql

#Αλλαγή του τρέχονος Search path για να συσχετιστεί με το νέο του Schema
```

```
echo "";  
  
#Εντολές διαγραφής χρήστη μαζί με τα Αντικείμενά του (pg_delete.sql)  
  
echo "DROP OWNED BY $PG_USR CASCADE;" >> ./postgres/pg_delete.sql  
  
#Διαγραφή των αντικειμένων του πρώτα  
  
echo "DROP USER IF EXISTS $PG_USR;" >> ./postgres/pg_delete.sql  
  
#και του ίδιου του χρήστη εν συνεχεία (αν υπάρχει)  
  
echo "";  
  
done; #Τέλος Loop  
  
echo "=====";  
  
echo "All Done...."; #Αναγγελία Ολοκλήρωσης  
  
echo "=====";  
  
exit 0; #Ολοκλήρωση και έξοδος
```

Εκτέλεση Shell Script

Για να εκτελέσετε το Script, θα χρειαστεί να δημιουργήσετε ένα κατάλογο με όνομα «postgres», καθώς και την παρουσία του αρχείου «user_pass.txt» στον τρέχοντα κατάλογο. Αλλάξτε τα File Permissions του Script και εκτελέστε το (χωρίς ορίσματα ως χρήστες postgres) .

```
postgres@localhost$ chmod 755 postgres_generate.sh  
  
postgres@localhost$ ./postgres_generate.sh
```

Το Script θα δημιουργήσει στον κατάλογο postgres δύο SQL Scripts, το «pg_create.sql» και το «pg_delete.sql».

Εκτέλεση SQL Scripts

Για να εκτελέσετε τα SQL Scripts θα χρειαστείτε την κονσόλα διαχείρισης της PostgreSQL, δηλαδή το εργαλείο psql. Αφού συνδεθείτε στην βάση που θέλετε να δημιουργήσετε τους χρήστες και τα schemas, εκτελέστε το Script με «\i <path προς; script>». Δείτε παρακάτω πως.

```
postgres@localhost$ psql #έναρξη psql console
postgres# #Prompt psql console
--Syntax: \c <όνομα βάσης δεδομένων προς σύνδεση>
postgres# \c dblabs
You are now connected to database "dblabs" as user "postgres".
--Syntax: \i <path προς αρχείο sql script>
dblabs# \i /home/postgres/pg_create.sql
```