

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΛΟΓΙΣΤΙΚΗΣ

Α.Τ.Ε.Ι. ΠΑΤΡΑΣ

ΘΕΜΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ:

«Ηλεκτρονική διαχείριση του φοιτητολογίου και του προγράμματος σπουδών ενός ελληνικού τμήματος ΑΠΕΙ. Στατιστική επεξεργασία για την εξαγωγή χρήσιμων συμπερασμάτων σχετικά με την ζήτηση του τμήματος και την ακαδημαϊκή πρόοδο και επίδοση των φοιτητών.»

Ο ΕΙΣΗΓΗΤΗΣ:

ΣΙΟΥΤΑΣ ΣΠΥΡΙΔΩΝ

ΟΙ ΣΠΟΥΔΑΣΤΕΣ:

- 1. ΝΙΚΟΛΑΚΟΠΟΥΛΟΣ ΣΠΥΡΙΔΩΝ**
- 2. ΝΙΚΟΛΟΠΟΥΛΟΥ ΕΛΕΥΘΕΡΙΑ**
- 3. ΣΩΤΗΡΟΠΟΥΛΟΥ ΑΘΑΝΑΣΙΑ**

ΠΑΤΡΑ 2004

ΠΕΡΙΕΧΟΜΕΝΑ



➤ ΠΡΟΛΟΓΟΣ.....	2
➤ ΚΕΦΑΛΑΙΟ 1.....	3
Η Εξέλιξη στη ΒΔ.....	4
Η Αρχιτεκτονική του DBMS.....	6
Εισαγωγή στα μοντέλα δεδομένων.....	7
Συνοπτική περιγραφή της ΒΔ.....	8
Εισαγωγή στην SQL.....	9
Η δομή της SQL.....	10
➤ ΚΕΦΑΛΑΙΟ 2.....	11
2.1 Πίνακες.....	12
2.2 Σχέσεις πινάκων.....	18
2.3 Φόρμες.....	22
2.4 Ερωτήματα.....	25
2.5 Εκθέσεις.....	32
➤ ΚΕΦΑΛΑΙΟ 3.....	35
PL/SQL.....	35
➤ ΚΕΦΑΛΑΙΟ 4.....	54
ORACLE.....	54
➤ ΕΠΙΛΟΓΟΣ.....	76
➤ ΒΙΒΛΙΟΓΡΑΦΙΑ.....	77
➤ ΠΑΡΑΡΤΗΜΑ.....	78

ΑΡΙΘΜΟΣ ΕΙΣΑΓΩΓΗΣ	5876
----------------------	------

ΠΡΟΛΟΓΟΣ

Η πτυχιακή αυτή γράφτηκε με σκοπό την παρουσίαση ενός προγράμματος που αποβλέπει στην ηλεκτρονική διαχείριση του φοιτητολογίου και του προγράμματος σπουδών ενός Ελληνικού τμήματος Α.Τ.Ε.Ι.

Αυτό το πρόγραμμα είναι ένα χρήσιμο εργαλείο για τις γραμματείες της κάθε σχολής γιατί μας δίνει την δυνατότητα να περνάμε τα στοιχεία των φοιτητών καθώς και την απόδοσή τους σε κάθε εξάμηνο. Γενικά, μπορούμε ανά πάσα στιγμή να ενημερωνόμαστε για τα πλήρη στοιχεία όλων των φοιτητών που σπουδάζουν στο εν λόγω ΑΤΕΙ.

Μία άλλη δυνατότητα που μας παρέχει το πρόγραμμα αυτό είναι ότι μπορούμε να βγάλουμε χρήσιμα συμπεράσματα σχετικά με την ζήτηση του τμήματος και την ακαδημαϊκή πρόοδο και επίδοση των φοιτητών.

Το πρόγραμμα αυτό θα βασιστεί στην Access, όπου είναι η πιο εύκολη γλώσσα προγραμματισμού για κάθε χρήστη. Η Access προσφέρει πολλές ευκολίες στον χειρισμό της και έτσι η γραμματεία μπορεί πολύ γρήγορα να συλλέγει τις πληροφορίες που θέλει έστω και αν το προσωπικό της δεν έχει ιδιαίτερες γνώσεις στους υπολογιστές. Σκοπός λοιπόν της εργασίας μας είναι να φτιάξουμε ένα απλό και εύχρηστο πρόγραμμα πάνω στην Access όπου και θα καταχωρούνται όλες οι πληροφορίες που έχουν να κάνουν με την πορεία του κάθε φοιτητή στο τμήμα λογιστικής των ΑΤΕΙ Πάτρας.

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΙΚΕΣ ΕΝΝΟΙΕΣ

Χρήσιμο θα ήταν να ξεκινήσουμε την εργασία μας με κάποιες εισαγωγικές έννοιες έτσι ώστε να γίνει πιο κατανοητή η εργασία μας αργότερα.

Ας ξεκινήσουμε με την έννοια του προγραμματισμού γενικότερα. Ο **προγραμματισμός** είναι μέχρι σήμερα η κυριότερη μέθοδος προγραμματισμού, με την οποία ο άνθρωπος καθοδηγεί τον υπολογιστή στην επίλυση προβλημάτων. Πρόκειται για μια από τις συνθετότερες νοητικές διαδικασίες, η οποία αντιστέκεται ακόμα στις διάφορες προσπάθειες συστηματοποίησης της που γίνονται συνεχώς από τους ερευνητές. Βασικά συστατικά κάθε προγράμματος είναι οι αλγόριθμοι και οι δομές δεδομένων. Για τη συγγραφή επιτυχημένων προγραμμάτων είναι απαραίτητες οι τεχνικές δόμησης και αποσύνδεσης ενός προβλήματος σε μικρότερα τμήματα κατάλληλου μεγέθους και πολυπλοκότητας, ώστε να είναι δυνατή η υλοποίησή τους.

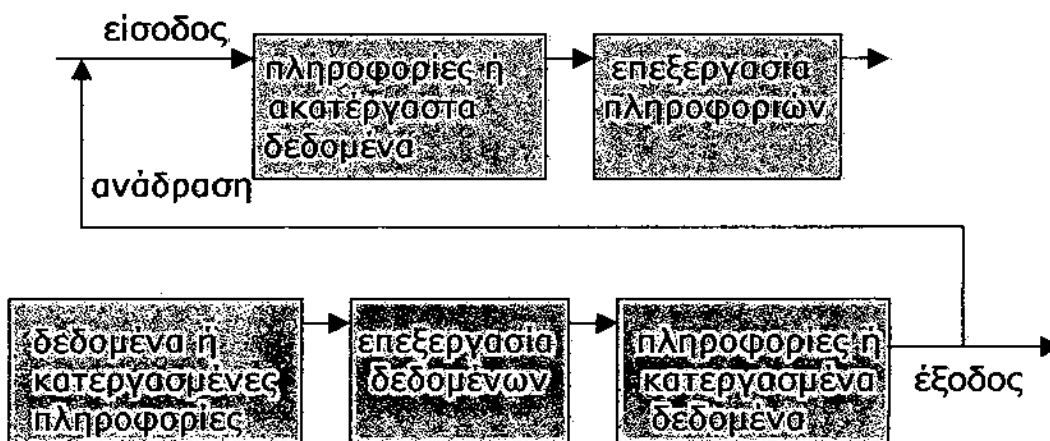
Σε επόμενο στάδιο θα αναλύσουμε την έννοια της βάσης δεδομένων. Μια **βάση δεδομένων** λοιπόν είναι μια συλλογή καλά οργανωμένων εγγραφών μέσα σε κοινός διαθέσιμα μέσα μαζικής αποθήκευσης. Εξυπηρετεί μια ή περισσότερες εφαρμογές με βέλτιστο τρόπο, επιτρέποντας κοινοί και ελεγχόμενη αντιμετώπιση της εισαγωγής, της τροποποίησης και της ανάκτησης συνόλων δεδομένων. Το σύστημα διαχείρισης βάσεων δεδομένων (DBMS) είναι ένα σύνολο προγραμμάτων, τα οποία επιτελούν αυτές τις λειτουργίες με τυποποιημένο και πλήρες ελεγχόμενο τρόπο.

Σε αυτό το σημείο θα αναφερθούμε στους ορισμούς πληροφορία και δεδομένο καθώς έχουν σημαντικές διαφορές, οπωσδήποτε λεπτές και δυσδιάκριτες από πρώτη άποψη. Δίνοντας τον ορισμό του δεδομένου και παραθέτοντας το παρακάτω σχήμα θα

μπορέσουμε να διαχωρίσουμε και να κατανοήσουμε καλύτερα τις έννοιες αυτές.

Δεδομένο συνεπώς καλούμε μία παράσταση γεγονότων, εννοιών ή εντολών σε τυποποιημένη μορφή που είναι κατάλληλη για επικοινωνία, ερμηνεία ή επεξεργασία από τον άνθρωπο ή από αυτόματα μέσα.

Στη συνέχεια δίνουμε το σχήμα της επεξεργασίας πληροφοριών και δεδομένων.



σχ. Επεξεργασία πληροφοριών και δεδομένων

Η ΕΞΕΛΙΞΗ ΣΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Αρχίζοντας από τα τέλη του 18^{ου} αιώνα, η μηχανική, αυτόματη επεξεργασία της πληροφορίας έχει περάσει από διάφορους σταθμούς για να γνωρίσει ραγδαία ανάπτυξη από τη στιγμή που η κωδικοποίηση άρχισε να γίνεται ηλεκτρονικά. Ανάπτυξη σχεδόν επαναστατική, που επηρεάζει όλο και περισσότερο το σύνολο της ανθρώπινης δραστηριότητας. Ο τρόπος με τον οποίο ο σύγχρονος άνθρωπος εργάζεται, εκπαιδεύεται, ψυχαγωγείται, ακόμα και σε ένα μεγάλο βαθμό αυτή καθαυτή η κοινωνική του συμπεριφορά

προσαρμόζονται διαρκώς στις ραγδαίες αλλαγές που επιφέρει η σύγχρονη κοινωνία της πληροφόρησης, όπως ονομάζεται.

Στο κείμενο που ακολουθεί επιχειρείται μια μακροσκοπική θεώρηση και χρονολογική κατάταξη των κυριοτέρων σταδίων από τα οποία περάσαμε για να φτάσουμε στα σημερινά Συστήματα Διαχείρισης Βάσεων Δεδομένων(DBMS)

1880 Εφεύρεση της διάτρητης κάρτας. Γίνεται το κατ' εξοχήν μέσο αποθήκευσης και επεξεργασίας της πληροφορίας.

1940 Ο πρώτος Η.Υ. στα 1946. χρησιμοποιείται για μαθηματικούς υπολογισμούς. Δεν υπάρχει ακόμα μεγάλη ανάγκη μόνιμης αποθήκευσης της πληροφορίας. Εφευρίσκεται η μαγνητική ταινία. Γρήγορη και ευέλικτη, πάντως σειριακή καταχώρηση της πληροφορίας.

1950 Αρχίζουν να εμφανίζονται τα πρώτα πληροφοριακά συστήματα. Συχνά σχεδιάζονται και λειτουργούν σε απομόνωση το ένα από το άλλο. Δεν γίνεται μεταφορά δεδομένων από Η.Υ. σε Η.Υ. ούτε καν από εφαρμογή σε εφαρμογή. Το τυπικό περιβάλλον αποτελείται από πολλά μικρά προγράμματα και πολλά αρχεία. Το καθένα από τα δεύτερα περιέχει τμηματική πληροφορία όσον αφορά το σύνολο της επιχείρησης. Δύσκολη η δημιουργία ολοκληρωμένου περιβάλλοντος με δεδομένα που χρησιμοποιούνται από πολλές εφαρμογές.

1960 Εφευρίσκεται ο μαγνητικός δίσκος. Ταχύτερη πρόσβαση στα στοιχεία. Η προσπέλαση στα δεδομένα εκτός από σειριακή γίνεται και άμεση. Εμφανίζονται συστήματα όπου πολλά αρχεία χρησιμοποιούνται από πολλές εφαρμογές ταυτόχρονα. Προβλήματα συντονισμού και συγχρονισμού πρόσβασης στα κοινά αρχεία. Ακόμη προβλήματα έλλειψης αυτόνομης λογικής συσχέτισης των δεδομένων από αρχείο σε αρχείο.

1970 Γίνεται πλέον συνείδηση ότι χρειάζεται ένα ολοκληρωμένο πληροφοριακό περιβάλλον (βάση δεδομένων) με τον όρο ολοκληρωμένο νοείται η υποστήριξη του συνόλου των πληροφοριακών αναγκών ενός οργανισμού από μία και μόνη βάση δεδομένων. Το σχήμα δημιουργεί πρόσθετες απαιτήσεις όσον αφορά τον σχεδιασμό και τη λειτουργία του λογισμικού. Η συγχρονισμένη συνλειτουργία των διάφορων υποσυστημάτων διαχειρίζεται τα ίδια αξιόπιστα δεδομένα όπου η κάθε στιγμή πληροφορίας καταχωρείται μία και μόνη φορά.

Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ D.B.M.S.

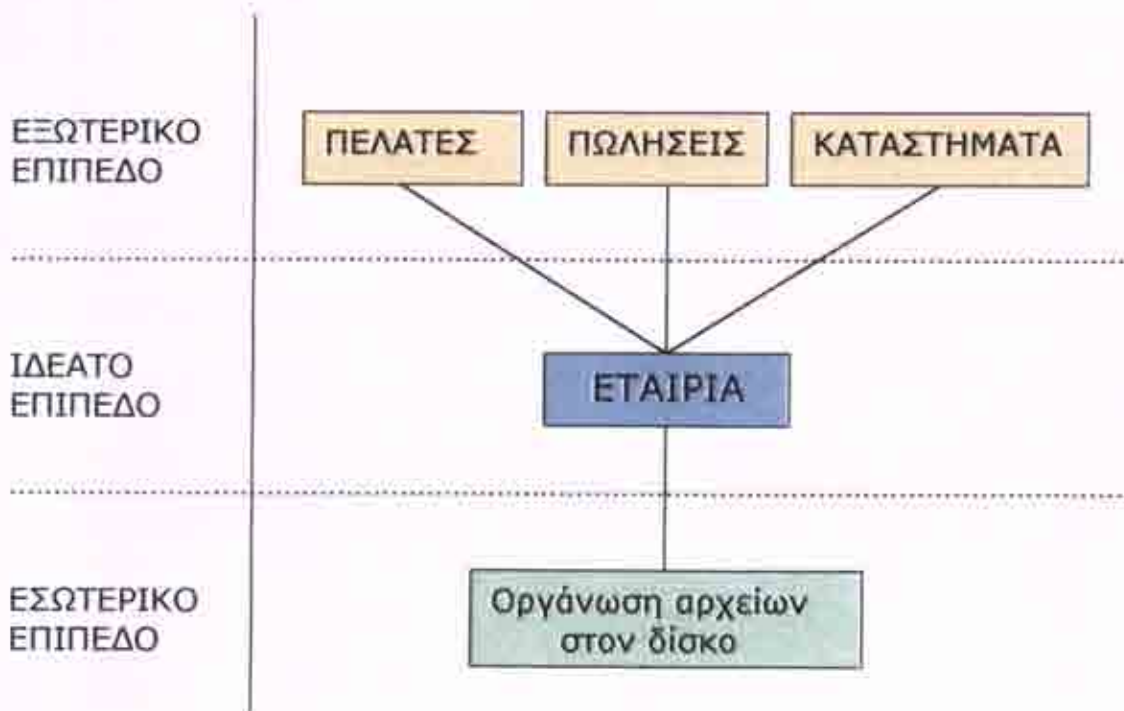
Στη συνέχεια θα αναφερθούμε στην δομή του DBMS που αποτελείται από τρία επίπεδα. Αυτά είναι το εξωτερικό, το ιδεατό και το εσωτερικό.

Το **εξωτερικό** επίπεδο το αποτελούν οι διάφορες ομάδες οπτικές χρηστών και εφαρμογών στα περιεχόμενα της Β.Δ. Στην κάθε περίπτωση, Β.Δ. η (περιορισμένη, εν γένει) αντίστοιχη οπτική.

Σε αντίθεση με το εξωτερικό, η οπτική του **ιδεατού** επιπέδου αγκαλιάζει το σύνολο του πληροφοριακού περιεχομένου της Β.Δ. επάνω σε αυτή χτίζονται οι μερικές όψεις του εξωτερικού επιπέδου.

Τέλος το **εσωτερικό** επίπεδο είναι αυτό που βρίσκεται πλησιέστερα στον τρόπο με τον οποίο η πληροφορία είναι καταχωρημένη στην περιφερειακή μνήμη του Η.Υ.

Στο παράδειγμα που ακολουθεί γίνεται πιο κατανοητή η δομή της Β.Δ. μιας εταιρείας



Σχ. Τα τρία επίπεδα του Σ.Δ.Β.Δ.

ΕΙΣΑΓΩΓΗ ΣΤΑ ΜΟΝΤΕΛΑ ΔΕΔΟΜΕΝΩΝ

Σε αυτή την ενότητα θα δώσουμε κάποιους ορισμούς προκειμένου να αποκαταστήσουμε μια οπτική σε ένα σύστημα πριν το σχεδιάσουμε. Η οπτική αυτή θα περιλαμβάνει όλες τις λεπτομέρειες που μας ενδιαφέρουν και θα αποτελεί μια μερική εικόνα της πραγματικότητας.

Όταν λοιπόν μοντελοποιούμε την πραγματικότητα, την απεικονίζουμε σε ένα σύστημα που έχει να κάνει με οντότητες και συσχετίσεις.

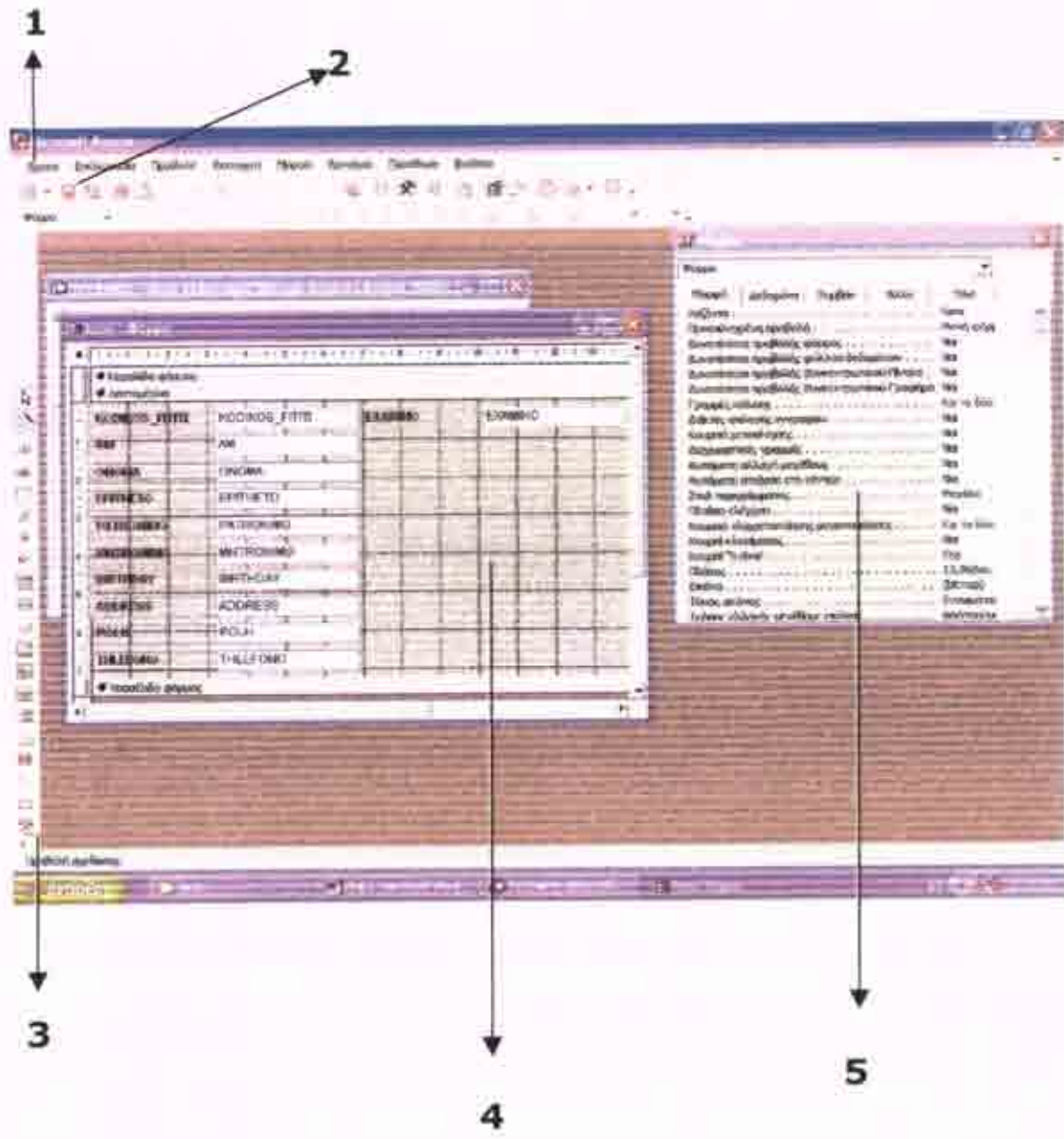
Έτσι **οντότητα** είναι η κάθε περίπτωση μονάδας του πραγματικού συστήματος την οποία ενδιαφερόμαστε να παρακολουθήσουμε πληροφοριακά και η οποία έχει αυτόνομη ύπαρξη μέσα στον κόσμο της υπό ανάπτυξη εφαρμογής. Η κάθε οντότητα έχει μια σειρά από χαρακτηριστικά τις τιμές των οποίων θέλουμε να γνωρίζουμε και να επεξεργαζόμαστε. Οι τιμές των χαρακτηριστικών καθορίζουν την κάθε μια στιγμή οντότητας και την διακρίνουν από τις υπόλοιπες στιγμές της ίδιας οντότητας.

Ένα από αυτά τα χαρακτηριστικά ή κάποιος συνδυασμός τους διακρίνεται από την ικανότητα που έχει να προσδιορίζει μονοσήμαντα, με την τιμή του , την κάθε στιγμή της οντότητας. Αυτό είναι το **κύριο κλειδί** της οντότητας.

Στον χώρο του μοντέλου της εφαρμογής, οι οντότητες δεν είναι απομονωμένες αλλά αλληλεπιδρούν με το περιβάλλον τους. Οι αλληλοεπιδράσεις απεικονίζονται με την μορφή **συσχετίσεων** που ορίζονται να υπάρχουν ανάμεσα σε δύο ή περισσότερες οντότητες.

ΣΥΝΟΠΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΗΣ Β.Δ.

Το περιβάλλον εργασίας της Β.Δ.:



Το περιβάλλον προγραμματισμού της Visual Basic

- 1. Κύριο μενού
- 2. Βασική γραμμή εργαλείων
- 3. Εργαλειοθήκη (βασικά στοιχεία ελέγχου)
- 4. Παράθυρο φόρμας
- 5. Παράθυρο ιδιοτήτων

Έργα, φόρμες και στοιχεία ελέγχου

Στον προγραμματισμό με την Β.Δ. κάθε πρόγραμμα που δημιουργούμε καλείται **έργο**. Θεμελιώδες συστατικό κάθε έργου (και γενικότερα του οπτικού προγραμματισμού) είναι οι **φόρμες**. Μια φόρμα είναι ένας χώρος στον οποίο ο προγραμματιστής μπορεί να τοποθετήσει τα αντικείμενα με τα οποία ο τελικός χρήστης θα έρθει σε επαφή. Τα **αντικείμενα** που τοποθετούνται σε μία φόρμα ονομάζονται **στοιχεία ελέγχου**.

Βασικά στοιχεία ελέγχου

Τα βασικά στοιχεία ελέγχου (συνολικά 21) φαίνονται στην εικόνα (1) και είναι στη διάθεση του προγραμματιστή όποτε αυτός χρησιμοποιεί τη Β.Δ. με τα στοιχεία ελέγχου μπορούμε να κάνουμε διάφορες εργασίες όπως να προσθέτουμε στη φόρμα μας κουμπιά ειδικών λειτουργιών, πλαίσια ελέγχου, ετικέτες, εικόνες.

ΕΙΣΑΓΩΓΗ ΣΤΗΝ SQL

Σε ένα περιβάλλον RDBMS, η επικοινωνία του χρήστη με την βάση και τα δεδομένα που καταχωρεί γίνεται μέσω εντολών υψηλού επιπέδου, η σύνταξη των οποίων φτάνει να πλησιάζει την σύνταξη της γλώσσας με την οποία επικοινωνούν μεταξύ τους οι άνθρωποι. Φυσικά, η όποια σύνταξη σε μια τέτοια υψηλού επιπέδου γλώσσα προγραμματισμού πρέπει να απεικονιστεί σε ισοδύναμη παράσταση τελεστών σχεσιακής άλγεβρας ώστε να γίνει η επεξεργασία της πληροφορίας στο ισοδύναμο αλγεβρικών παραστάσεων σχεσιακής άλγεβρας πριν οι τελευταίες ενεργήσουν επάνω στην σχεσιακή βάση δεδομένων. Η απεικόνιση γίνεται με τρόπο αυτόματο από το λογισμικό του RDBMS χωρίς να το αντιλαμβάνεται ο χρήστης, κάτι ανάλογο με την απεικόνιση, π.χ. εντολών C σε κώδικα που είναι αντιληπτός και εκτελέσιμος από την μηχανή. Η SQL (Structured Query Language), είναι η standard γλώσσα επικοινωνίας χρήστη και προγραμματιστή με το RDBMS.

Η SQL έχει εξελιχθεί σε παγκόσμιο standard προσπέλασης σε περιβάλλον RDBMS. Υπάρχουν και άλλες, αντίστοιχες, γλώσσες που όμως δεν επιβίωσαν απέναντι στις αυξημένες ανάγκες για γρήγορη ευέλικτη επεξεργασία και επικοινωνία της πληροφορίας στην σημερινή εποχή. Οι κατασκευαστές συνεχίζουν την παραγωγή non-standard πρόσβασης στα RDBMS, περιοριζόμενοι όμως στον χώρο των γλωσσών τέταρτης γενιάς (4GLs).

Η SQL αποτελεί την εξέλιξη της SEQUEL που αναδύθηκε από το πρόγραμμα «Σύστημα R» της IBM το 1978. Σήμερα έχει εξελιχθεί σε ISO/ANSI standard με ελάχιστες αποκλίσεις του ενός κατασκευαστή από τον άλλον. Ήδη εξελίσσεται σε SQL/2, SQL/3,.....ενώ πλησιάζει και η TSQL(=Temporal SQL).....

Η ΔΟΜΗ ΤΗΣ SQL

Η SQL περιλαμβάνει πολύ μικρό πλήθος εντολών σε σχέση με τα άλλα περιβάλλοντα διαχείρισης και επεξεργασίας της πληροφορίας σε DBMS. Οι εντολές της ομαδοποιούνται σε δύο βασικές κατηγορίες, την DDL (Data Definition Language) και την DML (Data Manipulation Language). Συγκεκριμένα:

SELECT

Ανακαλεί δεδομένα από το RDBMS

INSERT UPDATE DELETE

Εισάγουν, ενημερώνουν και διαγράφουν εγγραφές στη βάση. Μαζί με την SELECT συναποτελούν το υποσύστημα DML/SQL.

CREATE ALTER DROP

Δημιουργούν, αλλάζουν και διαγράφουν πίνακες και άλλες δομές (π.χ. όψεις, κλειδιά αναζήτησης κλπ) στη βάση. Συναποτελούν το υποσύστημα DDL/SQL.

GRANT REVOKE

Ορίζουν, ακυρώνουν δικαιώματα χρηστών όσον αφορά την πρόσβαση στα δεδομένα της βάσης.

ΚΕΦΑΛΑΙΟ 2

ΚΑΤΑΣΚΕΥΗ ΦΟΙΤΗΤΟΛΟΓΙΟΥ

Σε αυτό το κεφάλαιο θα δούμε βήμα προς βήμα πως σχεδιάσαμε και υλοποιήσαμε το συγκεκριμένο πρόγραμμα. Σε αυτό το σημείο καλό θα ήταν να αναφερθεί ότι τα στοιχεία που θα αναφερθούν παρακάτω είναι από το Α.Τ.Ε.Ι. Πάτρας και συγκεκριμένα από το τμήμα Λογιστικής και ειδικότερα από τη γραμματεία του τμήματος Λογιστικής.

Βασική προϋπόθεση για τη δημιουργία του προγράμματος είναι ο σχεδιασμός του λεγόμενου N – R διαγράμματος. Το πρώτο μας βήμα θα είναι η συγκέντρωση όλων των στοιχείων εκείνων που θα πρέπει να συμπεριληφθούν και να επεξεργαστούν έτσι ώστε να γίνει πιο ολοκληρωμένο, ορθό και αντιπροσωπευτικό το πρόγραμμα .

Έτσι για την συλλογή των στοιχείων μας, χρειάστηκε να επισκεφθούμε την γραμματεία του τμήματος Λογιστικής και να ενημερωθούμε από έναν αρμόδιο για τα στοιχεία που είναι απαραίτητα για τη διεξαγωγή ενός ολοκληρωμένου προγράμματος και για το πώς γενικότερα λειτουργεί το αντίστοιχο πρόγραμμα του τμήματος μας. Φυσικά η ιδιότητά μας ως φοιτητές του τμήματος αυτού μας βοήθησε να κατανοήσουμε καλύτερα τις πληροφορίες που συγκεντρώσαμε. Επίσης προμηθευτήκαμε με τα ανάλογα έντυπα τα οποία συμπληρώνονται σε διάφορες περιπτώσεις. Τα συγκεκριμένα έντυπα που συμπληρώνουν οι φοιτητές κατά την εγγραφή τους καθώς και κατά την διάρκεια των σπουδών τους τα περιλαμβάνουμε στο παράρτημα.

Σε αυτό το σημείο αντιμετωπίστηκαν αρκετές δυσκολίες. Τα στοιχεία ήταν αρκετά και ο συνδυασμός τους πολύπλοκος. Για παράδειγμα κάποια έντυπα χρησιμοποιούνται μόνο για στατιστικούς λόγους και προφανώς δεν έχουν να κάνουν με την παρακολούθηση των φοιτητών. Για αυτό το λόγο αποφασίστηκε η παράληψη κάποιων στοιχείων που θεωρήθηκαν περιττά ή και λίγο δύσκολο να συμπεριληφθούν σε ένα πρόγραμμα βασισμένο μόνο στην Access.

Αφού η συλλογή των στοιχείων μας ολοκληρωθεί το επόμενο βήμα είναι ο συνδυασμός τους και η κατάρτιση ενός πρόχειρου σχεδίου του N – R διαγράμματος.

Έτσι μετά από διαδοχικά διαγράμματα N –R και σε συνεννόηση με τον καθηγητή μας καταλήξαμε στο, κατά τη γνώμη μας, πιο πρακτικό και πραγματοποιήσιμο διάγραμμα. Συνοπτικά το διάγραμμα περιλαμβάνει την βασική οντότητα *fitites* όπου θα περιέχονται όλα τα στοιχεία των φοιτητών όπως ονοματεπώνυμο, διεύθυνση κ.τ.λ. Η οντότητα αυτή θα συνδέεται με σχέση ένα προς πολλά με τις οντότητες εξαμήνου Α', Β', Γ', Δ', Ε', ΣΤ', Ζ' και Πτυχίο Α'. Οι οντότητες των εξαμήνων θα περιέχουν τα αντίστοιχα μαθήματά τους.

Το επόμενο μας βήμα θα είναι η δημιουργία των πινάκων. Έτσι με βάση το παραπάνω διάγραμμα και σε συνδυασμό με κάποιες επιπλέον οδηγίες συνεχίζουμε με τους πίνακες.

2.1 ΠΙΝΑΚΕΣ

Ένα από τα σημαντικότερα κομμάτια της εργασίας αυτής είναι ο σχεδιασμός των πινάκων. Όπως αναφερθήκαμε και στην προηγούμενη ενότητα θα πρέπει να δημιουργηθούν τόσοι πίνακες όσα και τα εξάμηνα μας αλλά και ένας επιπλέον πίνακας που θα αφορά τα στοιχεία του φοιτητή που και με αυτόν θα ξεκινήσουμε.

Η υλοποίηση του προγράμματος ξεκινάει με την δημιουργία της βάσης δεδομένων το Φοιτητολόγιο. Μπαίνοντας στην *access* εμφανίζεται ο πίνακας της με δύο παράθυρα. Το πρώτο αφορά τη δημιουργία νέας βάσης δεδομένων το οποίο και μας αφορά και επιλέγουμε την *κενή βάση δεδομένων της access* και OK.

Στη συνέχεια δίνουμε όνομα στη βάση δεδομένων μας και δημιουργία. Η βάση μας είναι έτοιμη και προχωράμε στην δημιουργία των πινάκων μας. Τον πρώτο μας πίνακα τον ονομάσαμε ΦΟΙΤΗΤΕΣ (*fitites*).

Ο πίνακας αυτός όπως προαναφέραμε θα περιλαμβάνει όλα τα στοιχεία του φοιτητή που εγγράφεται για πρώτη φορά. Βρισκόμαστε ήδη στην βάση μας και κάνουμε διπλό κλικ στην επιλογή *δημιουργία πίνακα σε προβολή σχεδίασης* ή πατάμε το κουμπι *δημιουργία* και επιλέγουμε την *προβολή σχεδίασης* και ΟΚ.

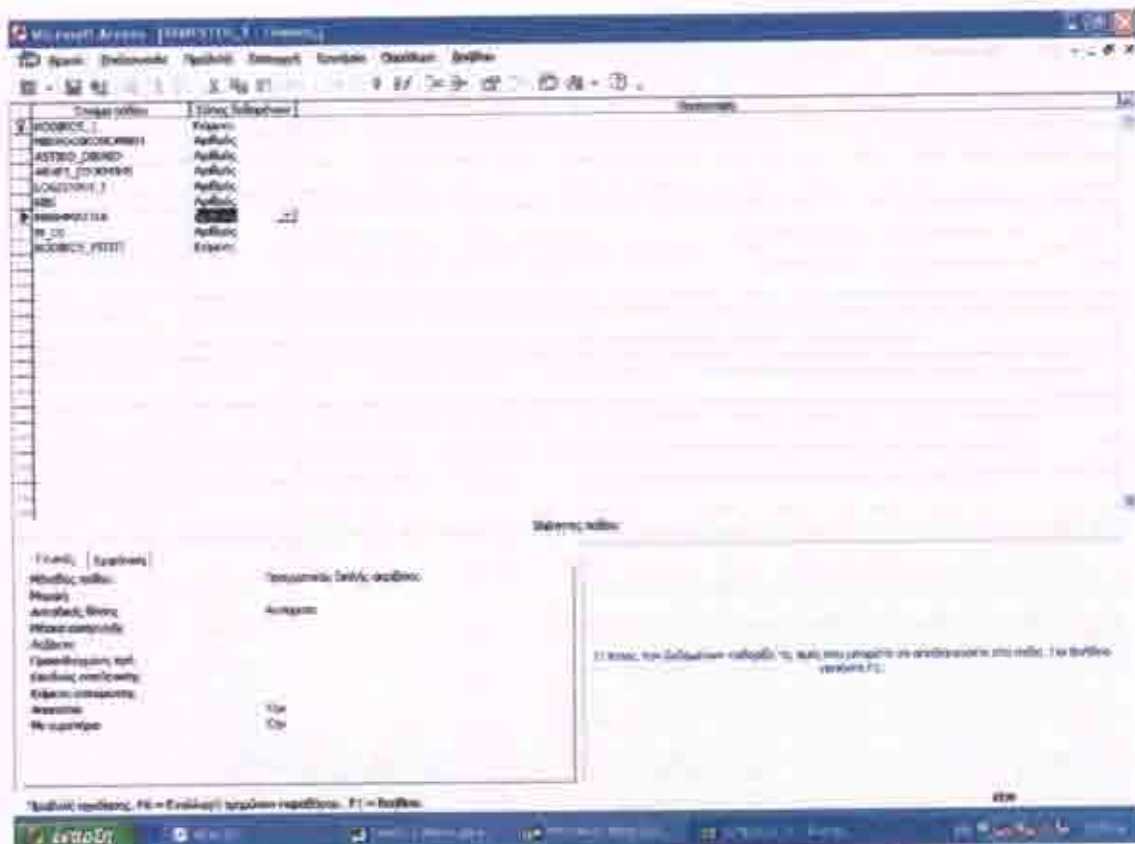
Μας εμφανίζεται αμέσως η προβολή σχεδίασης του πίνακα όπου θα περάσουμε τα στοιχεία που θα θέλουμε να εμφανίζει ο πίνακάς μας. Για την σχεδίαση του πίνακα θα πρέπει να επισημάνουμε κάποιες λεπτομέρειες όπως το ότι κατά την πληκτρολόγηση θα χρησιμοποιούμε πάντα λατινικούς χαρακτήρες γιατί είναι πιθανόν αργότερα να προκύψουν κάποια προβλήματα.

Ξεκινάμε με τη συμπλήρωση της στήλης *όνομα πεδίου* πληκτρολογώντας τα στοιχεία του πίνακα ξεκινώντας με το πεδίο *kodikos foititi* ο οποίος θα μας χρησιμεύσει και ως πρωτεύων κλειδί για να γίνει η σύνδεση των πινάκων. Για να το ορίσουμε ως πρωτεύον κλειδί κάνουμε δεξί κλικ πάνω στο *kodikos foititi* και επιλέγουμε το πρωτεύον κλειδί. Αυτόματα εμφανίζεται το σύμβολο του κλειδιού στην αριστερή στήλη του πεδίου.

Εάν σε περίπτωση ξεχάσουμε να ορίσουμε πρωτεύον κλειδί, τότε όταν πάμε να κλείσουμε την σχεδίαση του πίνακα, ενημερωνόμαστε πως δεν έχουμε ορίσει πρωτεύον κλειδί και ορίζει μόνο του ένα καινούριο πεδίο το οποίο και ονομάζει αναγνωρίσιμο.

Στην στήλη του τύπου επιλέγουμε τι τύπο θέλουμε να έχει το κάθε πεδίο. Για παράδειγμα στο πεδίο *kodikos_fititi* ο τύπος του είναι αριθμός. Στο επόμενο πεδίο που είναι το όνομα ο τύπος είναι κείμενο. Στην στήλη της περιγραφής μπορούμε να σημειώσουμε κάποιες πληροφορίες σχετικά με το πεδίο που θέλουμε να θυμόμαστε. Στο παράδειγμά μας δεν κρίνουμε απαραίτητο να συμπληρώσουμε κάτι.

Σε αυτό το σημείο μπορούμε να αναφέρουμε μία σημαντική ιδιότητα των πινάκων. Στις ιδιότητες πεδίων, στο πεδίο *απαιτείται* αν επιλέξουμε *ναι* τότε το πεδίο αυτό θα πρέπει να συμπληρώνεται υποχρεωτικά.



Με τον ίδιο τρόπο δουλεύουμε και για τα υπόλοιπα εξάμηνα. Παρακάτω παραθέτουμε τα μαθήματα που αφορούν το κάθε εξάμηνο.

Εξάμηνο 2: ΜΑΚΡΟΟΙΚΟΝΟΜΙΚΗ, ΣΤΟΙΧΕΙΑ ΕΜΠΟΡΙΚΟΥ ΔΙΚΑΙΟΥ, ΓΕΝΙΚΗ ΛΟΓΙΣΤΙΚΗ 2, ΟΙΚΟΝΟΜΙΚΑ ΜΑΘΗΜΑΤΙΚΑ, ΕΛΛΗΝΙΚΟ ΛΟΓΙΣΤΙΚΟ ΣΧΕΔΙΟ, ΑΥΤΟΜΑΤΙΣΜΟΣ ΓΡΑΦΕΙΟΥ.

Εξάμηνο 3: ΑΡΧΕΣ ΕΡΓΑΤΙΚΟΥ ΔΙΚΑΙΟΥ, ΣΤΑΤΙΣΤΙΚΗ ΕΠΙΧΕΙΡΗΣΕΩΝ, ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ, ΛΟΓΙΣΤΙΚΗ ΕΤΑΙΡΕΙΩΝ, ΜΗΧΑΝΟΓΡΑΦΗΜΕΝΗ ΛΟΓΙΣΤΙΚΗ Α & Β ΚΑΤΗΓΟΡΙΑΣ ΒΙΒΛΙΩΝ, ΟΙΚΟΝΟΜΙΚΗ ΕΠΙΧΕΙΡΗΣΕΩΝ.

Εξάμηνο 4: ΑΡΧΕΣ ΜΑΡΚΕΤΙΝΓΚ, ΛΟΓΙΣΤΙΚΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ - ΣΥΓΧΡΟΝΕΣ ΥΠΗΡΕΣΙΕΣ, ΝΟΜΙΣΜΑΤΙΚΗ ΘΕΩΡΙΑ & ΠΟΛΙΤΙΚΗ, ΦΟΡΟΛΟΓΙΚΗ ΕΜΜΕΣΩΝ ΦΟΡΩΝ, ΔΙΕΘΝΕΣ ΕΜΠΟΡΙΟ, ΑΝΑΠΤΥΞΗ ΑΝΘΡΩΠΙΝΩΝ ΠΟΡΩΝ, ΟΙΚΟΝΟΜΟΤΕΧΝΙΚΕΣ ΜΕΛΕΤΕΣ.

Εξάμηνο 5: ΧΡΗΜΑΤΟΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ & ΠΟΛΙΤΙΚΗ, ΒΙΟΜΗΧΑΝΙΚΟΣ ΛΟΓΙΣΜΟΣ – ΚΟΣΤΟΛΟΓΗΣΗ, ΜΗΧΑΝΟΓΡΑΦΗΜΕΝΗ ΛΟΓΙΣΤΙΚΗ 1, ΛΟΓΙΣΤΙΚΕΣ ΕΦΑΡΜΟΓΕΣ 1, ΜΕΤΑΤΡΟΠΕΣ – ΣΥΓΧΩΝΕΥΣΕΙΣ – ΕΝΟΠΟΙΗΜΕΝΟΙ ΙΣΟΛΟΓΙΣΜΟΙ, ΦΟΡΟΛΟΓΙΚΗ ΑΜΕΣΩΝ ΦΟΡΩΝ.

Εξάμηνο 6: ΛΟΓΙΣΤΙΚΕΣ ΕΦΑΡΜΟΓΕΣ 2, ΑΝΑΛΥΣΗ ΟΙΚΟΝΟΜΙΚΩΝ ΚΑΤΑΣΤΑΣΕΩΝ, ΜΗΧΑΝΟΓΡΑΦΗΜΕΝΗ ΛΟΓΙΣΤΙΚΗ 2, ΑΝΑΛΥΤΙΚΗ ΛΟΓΙΣΤΙΚΗ, ΚΛΑΔΙΚΗ ΛΟΓΙΣΤΙΚΗ, ΕΦΟΔΙΑΣΤΙΚΗ ΛΟΓΙΣΤΙΚΗ (LOGISTICS), ΕΠΙΧΕΙΡΗΣΙΑΚΗ ΕΡΕΥΝΑ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ.

Εξάμηνο 7: ΣΕΜΙΝΑΡΙΟ ΤΕΛΕΙΟΦΟΙΤΩΝ, ΟΡΟΛΟΓΙΑ ΞΕΝΗΣ ΓΛΩΣΣΑΣ, ΕΣΩΤΕΡΙΚΟΣ ΚΑΙ ΕΞΩΤΕΡΙΚΟΣ ΕΛΕΓΧΟΣ (AUDITING), ΜΗΧΑΝΟΓΡΑΦΗΜΕΝΗ ΛΟΓΙΣΤΙΚΗ ΔΙΑΧΕΙΡΙΣΗ ΑΠΟΘΗΚΗΣ, ΔΙΟΙΚΗΤΙΚΗ ΛΟΓΙΣΤΙΚΗ – ΠΡΟΤΥΠΟ ΚΟΣΤΟΣ.

Εξάμηνο 8(πτυχίο):ο βαθμός της πρακτικής, της πτυχιακής καθώς και ο συνολικός μέσος όρος όλων των εξαμήνων.

ΧΡΗΣΗ ΠΙΝΑΚΩΝ

Κατά την εισαγωγή των δεδομένων στον πίνακα μπορούμε να τροποποιήσουμε το μέγεθος των γραμμών και των στηλών εάν σύρουμε με το ποντίκι τις διαχωριστικές γραμμές των πεδίων μέχρι εκεί που θέλουμε.

Οι πίνακές ολοκληρώθηκαν και έτσι μπορούμε να καταχωρούμε καινούριες εγγραφές στα πεδία του κάθε πίνακα. Χρήσιμο είναι να αναφέρουμε πως θα πρέπει πρώτα να ενημερώνεται ο πίνακας του φοιτητή και στη συνέχεια οι πίνακες εξάμηνα με τις βαθμολογίες των φοιτητών. Εάν χρειαστεί να διαγράψουμε κάποιον φοιτητή θα πρέπει πρώτα να διαγράψουμε τις βαθμολογίες του από τους πίνακες εξάμηνα και μετά τον φοιτητή.

- Σε περίπτωση που χρειαζόμαστε να αλλάξουμε κάποιο στοιχείο σε κάποιον πίνακα θα πρέπει να μεταβούμε στην προβολή σχεδίασης του. Αυτό γίνεται εάν μεταβούμε στον πίνακα και

ενεργοποιήσουμε το κουμπί της προβολής σχεδίασης. Έτσι θα μας εμφανιστεί ο πίνακας σε μορφή προβολής σχεδίασης. Ένας δεύτερος τρόπος είναι να κάνουμε πάνω στον πίνακα δεξί κλικ και να επιλέξουμε την προβολή σχεδίασης. Η εκτύπωση θα εμφανίζεται ως εξής:

Μπορούμε ακόμα να εκτυπώσουμε κάποιον πίνακα από τη βάση δεδομένων κάνουμε δεξί κλικ στον πίνακα που θέλουμε να εκτυπώσουμε και επιλέξουμε εκτύπωση.

Εάν θέλουμε να εισάγουμε μια στήλη ή να την διαγράψουμε την μαρκάρουμε με το ποντίκι, κάνουμε δεξί κλικ και επιλέγουμε εισαγωγή ή διαγραφή στήλης αντίστοιχα. Το ίδιο ενεργούμε και για τις γραμμές.

ΑΚΩΝΟΜΟΣ_ΕΠΗ	ΑΜ	ΟΝΟΜΑ	ΕΠΙΘΕΤΟ	ΠΑΤΡΩΝΥΜΟ	ΜΗΤΡΩΝΥΜΟ	ΣΥΝΤΕΛΕΣ	ADDRESS	POLI	TELEFONO	ΕΙΔΟΣ
1	5712	ΝΙΚΟΣ	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΠΑΥΛΟΣ	ΕΛΕΝΗ	12071900	ΚΟΡΥΝΘΟΥ 80	ΠΑΤΡΑ	261060245	Υ
2	2222	ΚΑΤΕΡΙΝΑ	ΓΕΩΡΓΙΟΥ	ΚΩΝΣΤΑΝΤΙΝΟΣ	ΕΛΕΝΗ	15691979	ΕΡΜΟΥ 6	ΠΑΤΡΑ	261096100	ΠΥΛΟΣ
3	4090	ΝΙΚΟΣ	ΤΡΥΦΙΔ	ΚΩΝΣΤΑΝΤΙΝΟΣ	ΕΥΣΟΦΙΑ	03001070	ΕΡΜΟΥΠΟΛΕΩΣ	ΠΑΤΡΑ	094459970	ΠΥΛΟΣ
4	5700	ΜΑΡΙΑ	ΜΑΡΙΟΥ	ΑΓΓΕΛΟΣ	ΚΩΝΣΤΑΝΤΙΝΟΣ	16001980	ΝΑΥΜΑΧΟΥ 2	ΠΑΤΡΑ	261040007	ΠΥΛΟΣ
5	0700	ΠΑΡΑΣΚΕΥΗ	ΜΑΡΙΑΝΝΗΣ	ΝΙΚΟΣ	ΣΑΒΙΝΗ	09001980	ΣΕΜΕΤΕΧΝΟΕΙ	ΠΑΤΡΑ	007708090	ΠΥΛΟΣ
6	0900	ΑΓΓΕΛΟΣ	ΕΚΚΑΤΑΛΑΧΗ	ΑΝΔΡΕΑΣ	ΔΥΡΑΝΙΑ	03121985	Β. ΗΡΕΣΟΥ 5	ΠΑΤΡΑ	261033200	Α

2.2 ΣΧΕΣΕΙΣ ΠΙΝΑΚΩΝ

Το επόμενο βήμα που θα πρέπει να κάνουμε είναι η δημιουργία σχέσεων μεταξύ των πινάκων. Οι σχέσεις είναι ένα σημαντικό κομμάτι τις εργασίας αυτής για την περαιτέρω επεξεργασία της εργασίας γιατί ένα πιθανό λάθος την στιγμή αυτή θα μας δημιουργήσει πολλά προβλήματα.

Για να δημιουργήσουμε μια σχέση, πηγαίνουμε στο κουμπι σχέσεις και κάνουμε αριστερό κλικ. Στην συνέχεια μας εμφανίζεται ένας πίνακας στον οποίον επιλέγουμε όλους τους πίνακες που φτιάξαμε με τη διαδικασία επιλογή του πίνακα με το ποντίκι, και *προσθήκη*. Η διαδικασία επαναλαμβάνεται για όλους τους πίνακες και τους οποίους τους τοποθετεί σε ένα background οι οποίοι εμφανίζονται όταν κλείσουμε τον πίνακα που ανοίξαμε.

Έπειτα συνδέουμε τον πίνακα FITITES με τους υπόλοιπους πίνακες των εξαμήνων. Αυτό επιτυγχάνεται σύροντας με το ποντίκι τον κωδικό φοιτητή του πίνακα FITITES που έχουμε ορίσει ως κλειδί πάνω στα πεδία των άλλων πινάκων και συγκεκριμένα με τους κωδικούς φοιτητή των άλλων εξαμήνων.

Μόλις αφήνουμε το ποντίκι εμφανίζεται ο πίνακας καθορισμού των σχέσεων. Εκεί ενεργοποιούμε τις ενδείξεις οι οποίες θα μας φανούν χρήσιμες αργότερα.

Η πρώτη ένδειξη *ενεργοποίηση ακεραιότητας αναφορών* θα μας εξασφαλίσει τον σωστό διαχειρισμό των εγγραφών μας. Αυτό σημαίνει πως η εκτέλεση των εγγραφών μας θα γίνεται πάντα σύμφωνα με κάποιες εντολές και παραμέτρους που τους έχουμε δώσει.

Η επόμενη ένδειξη *διαδοχική ενημέρωση των σχετικών εγγραφών* θα μας εξασφαλίσει την ολοκληρωμένη ενημέρωση του προγράμματος. Για παράδειγμα όταν θα περνάμε ένα δεδομένο, με

την ένδειξη αυτή θα ενημερώνονται διαδοχικά όλα τα μέρη του προγράμματος που σχετίζονται με αυτό το καινούριο δεδομένο.

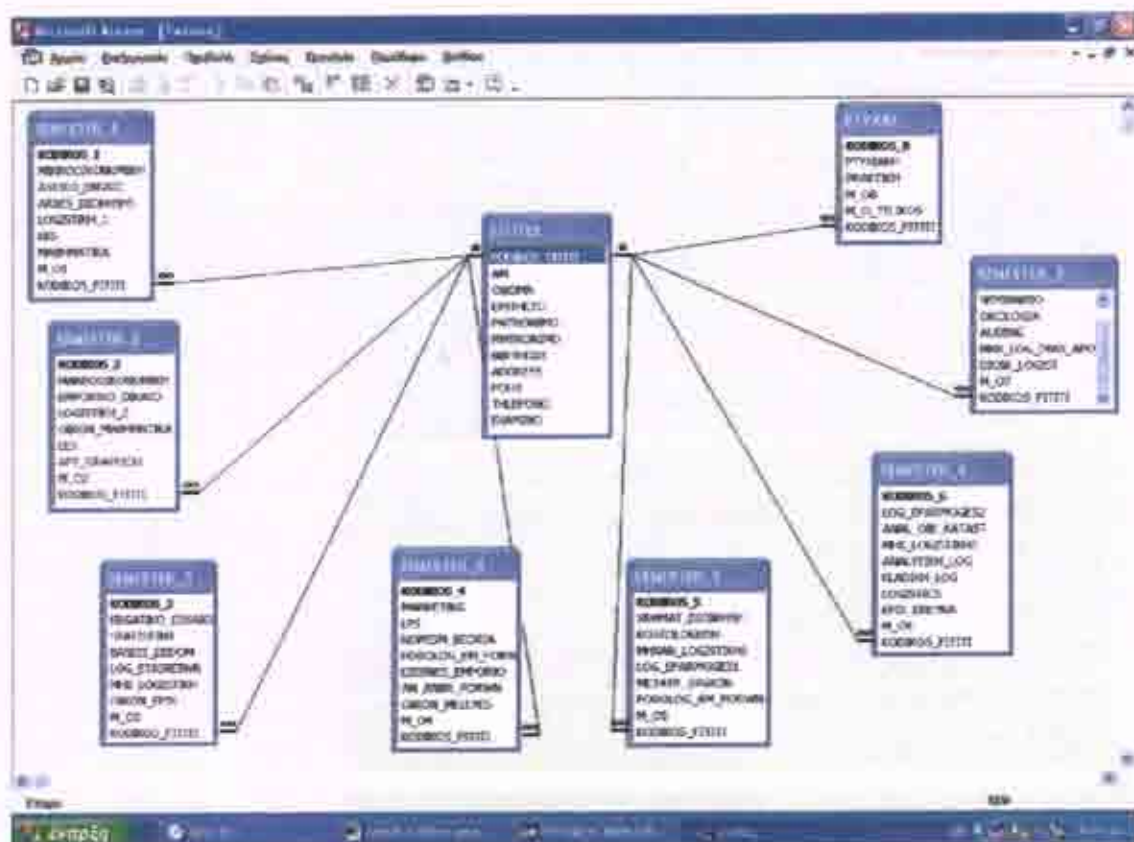
Η τελευταία μας ένδειξη *διαδοχική διαγραφή των σχετικών εγγραφών* μας εξασφαλίζει την σωστή σειρά με την οποία μπορούμε να διαγράψουμε κάποιες πληροφορίες. Για παράδειγμα, δεν μπορούμε να διαγράψουμε τα στοιχεία ενός φοιτητή χωρίς να έχουμε πριν διαγράψει τις βαθμολογίες του στα εξάμηνα.

Τώρα θα πρέπει να ορίσουμε τον τύπο μίας σχέσης. Υπάρχουν τρία είδη σχέσεων. Είναι η σχέση ένα προς ένα, που λέγεται *αμφιμονοσήμαντη* σχέση και η οποία ένα στοιχείο μιας οντότητας αντιστοιχεί μόνο σε ένα στοιχείο της άλλης οντότητας. Για παράδειγμα ένας μαθητής έχει έναν και μοναδικό αριθμό μητρώου.

Η δεύτερη σχέση που είναι ένα προς πολλά και λέγεται *μονοσήμαντη*. Σε αυτήν ένα στοιχείο μίας οντότητας αντιστοιχεί σε πολλά στοιχεία της άλλης οντότητας. Για παράδειγμα ένας καθηγητής έχει πολλούς μαθητές.

Η τελευταία σχέση είναι πολλά προς πολλά η οποία λέγεται *πολυσήμαντη*. Εδώ πολλά στοιχεία της μίας οντότητας αντιστοιχούν σε πολλά της άλλης οντότητας. Στο παράδειγμά μας ο τύπος των σχέσεων είναι ένα προς πολλά.

Για να ορίσουμε τον τύπο της σχέσης πατάμε το κουμπί τύπος συνδέσμου, επιλέγουμε τον κατάλληλο, OK και δημιουργία. Έτσι έχουμε μια σχέση ένα προς πολλά που είναι *μονοσήμαντη* σχέση.



Δημιουργώντας αυτές τις σχέσεις τώρα στον πίνακα FITITES θα μας βγάλει στα αριστερά μας ένα + όπου αν κάνουμε κλικ θα μπορούμε να δούμε απευθείας τους πίνακες εξαμήνα και συνεπώς τις βαθμολογίες του συγκεκριμένου μαθητή.

Microsoft Access [11/11/18 (Πέμπ)]

Αρχή Εξέλιξη Πηχά Επισελί Βασί Επισελί Σημάτι Σημάτι Σημάτι

ΚΟΔΙΚΟΣ ΕΠΙ ΑΜ ΟΙΚΙΑ ΕΠΙΧΕΙΡΤΟ ΠΑΡΟΧΩΜΑ ΜΑΥΡΟΧΩΜΑ ΣΥΝΘΕΣΗ ΑΔΩΚΕΩΣ ΠΟΛΗ Τηλεφώνο

ΚΟΔΙΚΟΣ ΕΠΙ	ΑΜ	ΟΙΚΙΑ	ΕΠΙΧΕΙΡΤΟ	ΠΑΡΟΧΩΜΑ	ΜΑΥΡΟΧΩΜΑ	ΣΥΝΘΕΣΗ	ΑΔΩΚΕΩΣ	ΠΟΛΗ	Τηλεφώνο	ΕΙΔΗ
1	572	ΜΑΚΕ	ΓΡΑΝΔΕΡΟΥ ΠΑΥΛΟΣ	ΕΛΕΦΗ	7257080	ΚΟΡΜΟΥ 80	ΠΑΤΡΑ	261040345	7	
ΚΟΔΙΚΟΣ Τ	ΜΗΡΟΔΟΜΗ	ΑΣΤΙΚΟ	ΟΙΚΙΑΡΧΗΣ	ΔΙΟΧΗ	ΛΟΓΙΣΤΗ Τ	ΚΒΣ	ΜΑΘΗΜΑΤΑ	M_01		
	7	0	4	5,5	7	5,5	0,0000000000			
2	223	ΚΑΤΕΡΙΝΑ	ΓΕΩΡΓΙΟΥ	ΚΩΝΣΤΑΝΤΙΝΟ	ΕΛΕΦΗ	1587079	ΕΡΜΟΥ 8	ΠΑΤΡΑ	261030100	ΠΥΛΟ
3	000	ΝΑΥ	ΠΡΕΚΣ	ΕΛΑΦΗ	ΕΥΔΟΝΑ	0302079	ΚΗΡΜΟΥ 100	ΠΑΤΡΑ	094950070	ΠΥΛΟ
4	522	ΒΟΤΕΡΗ	ΝΗΡΟΥ	ΑΓΓΕΛΟΣ	ΚΩΝΣΤΑΝΤΙΝΑ	15831082	ΝΑΥΜΑΧΑΣ 2	ΠΑΤΡΑ	261042007	ΠΥΛΟ
5	678	ΕΠΙΧΕΙΡΤΟ	ΜΑΥΡΑΝΩ	ΝΑΥ	ΠΑΖΑΡΗ	08070981	ΒΕΜΙΣΤΟΚΑΗ	ΠΑΤΡΑ	067088099	ΠΥΛΟ
6	330	ΧΥΣΑΝΗ	ΣΚΑΛΟΥΔΑΚΗ	ΑΝΔΡΕΑΣ	ΟΥΡΑΝΑ	03121085	Β.ΠΕΤΡΟΥ 5	ΠΑΤΡΑ	261033000	4

Τηλεφώνο 261033000

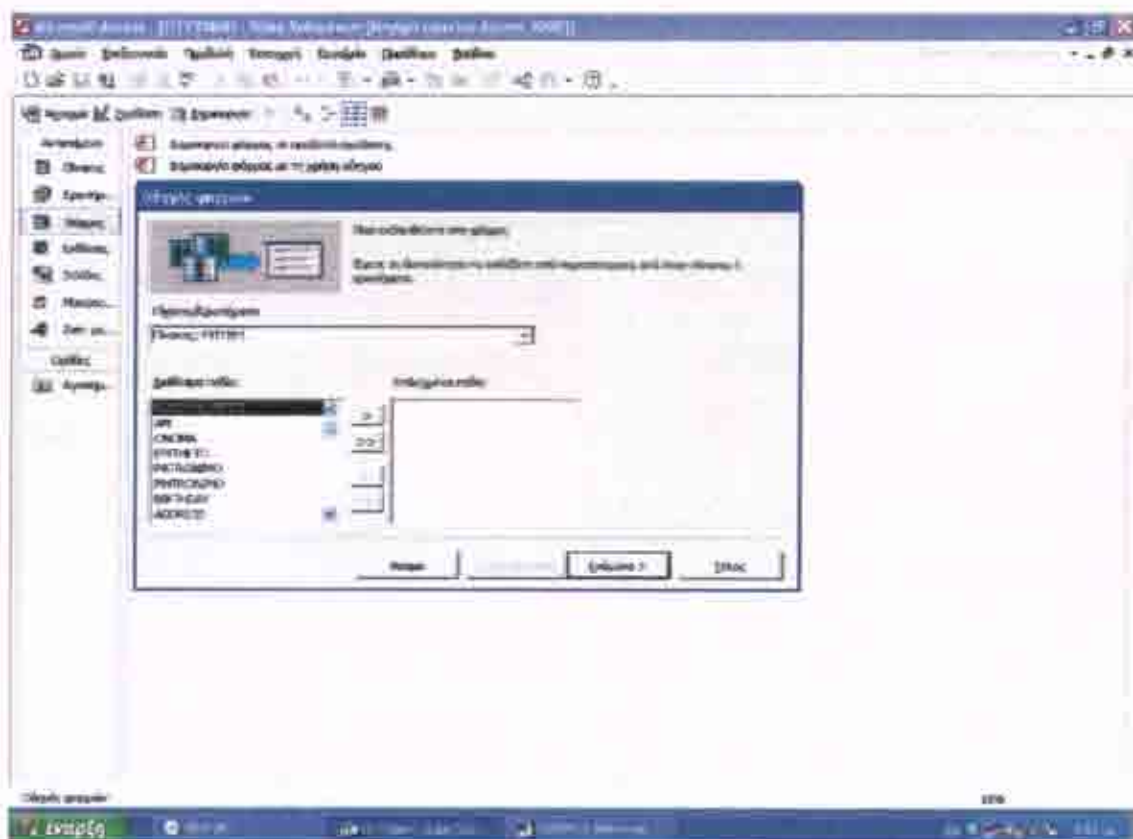
Microsoft Access [11/11/18 (Πέμπ)]

Αφού η δημιουργία των σχέσεων ολοκληρώθηκε είμαστε έτοιμοι να προχωρήσουμε στη δημιουργία των φορμών.

2.3 ΦΟΡΜΕΣ

Όταν πρόκειται να δημιουργήσουμε μια φόρμα, την δημιουργούμε βάση ενός πίνακα. Αυτό σημαίνει πως στη φόρμα θα δώσουμε τα ίδια πεδία με του πίνακα. Η φόρμα είναι απλά ένας πιο εύκολος και εύχρηστος τρόπος να περνάμε τα δεδομένα μας στον πίνακα. Πριν προχωρήσουμε στην δημιουργία φορμών θα πρέπει να σιγουρευτούμε ότι η δημιουργία των πινάκων είναι σωστή και ότι έχουμε περάσει όλα τα δεδομένα που θέλουμε. Στην περίπτωση μας θα δημιουργήσουμε την φόρμα fitites με βάση τον πίνακα fitites.

Για τη δημιουργία φορμών επιλέγουμε το πεδίο φόρμες από τη βάση δεδομένων και στη συνέχεια με διπλό κλικ το πεδίο *δημιουργία φόρμας με τη χρήση οδηγού*.



Εμφανίζεται ο οδηγός φορμών και από το πεδίο *πίνακες / ερωτήματα* επιλέγω τον αντίστοιχο πίνακα για τον οποίο θέλω να δημιουργήσω την φόρμα. Στο παράδειγμα μας επιλέγουμε τον πίνακα *Fitites* . Έτσι στο πλαίσιο διαθέσιμα πεδία εμφανίζονται τα πεδία του πίνακα *Fitites*.

Για τη δημιουργία της φόρμας θα επιλέξω όλα τα πεδία του *Fitites* κάνοντας κλικ στο κουμπί με τα δύο βελάκια (>>).

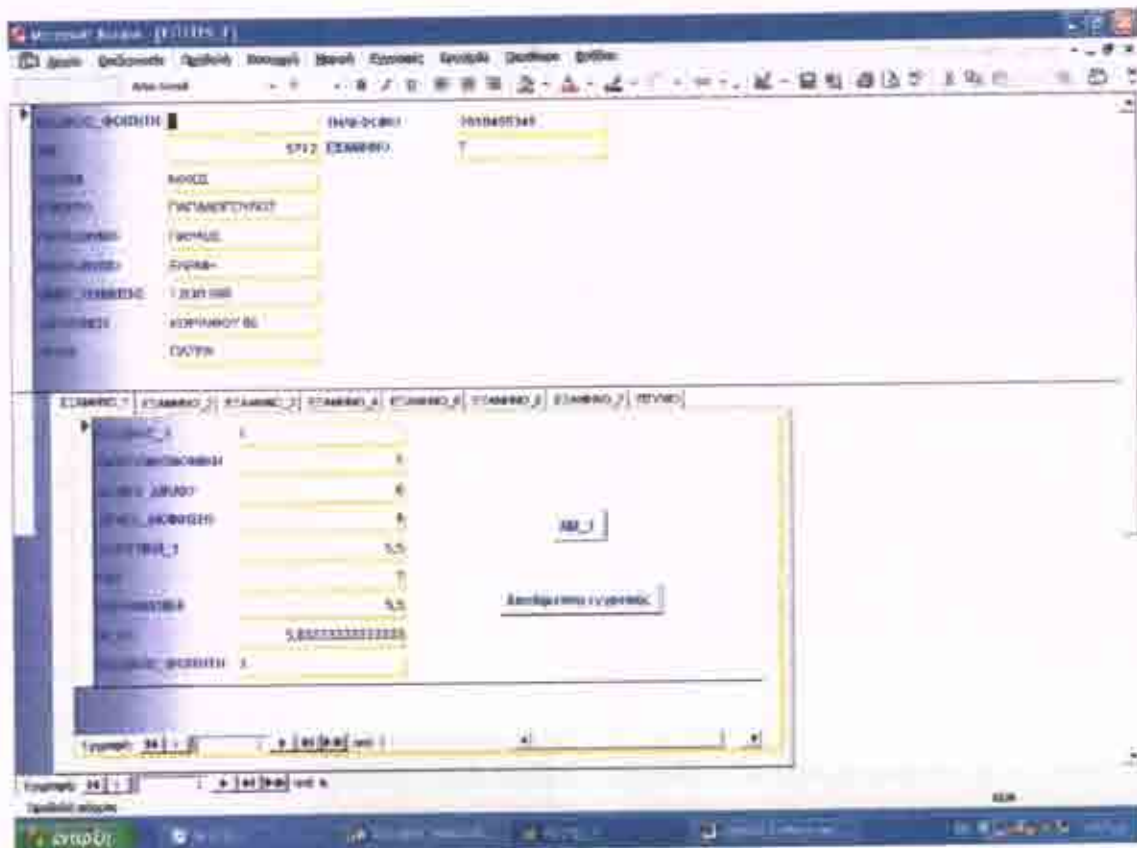
Σε περίπτωση που θέλω να περάσω τα πεδία ένα – ένα επιλέγω το πεδίο με το ποντίκι και στη συνέχεια κάνω κλικ στο κουμπί με το ένα βελάκι (>). Η διαδικασία επαναλαμβάνεται μέχρι να περάσω όλα τα πεδία εκείνα που θέλω. Στο παράδειγμά μας τα πεδία του *fitites* θα τα περάσουμε όλα μαζί με το διπλό βελάκι.

Για να πάω στην επόμενη σελίδα του οδηγού κάνω κλικ στο κουμπί *επόμενο>*. Εδώ επιλέγω την διάταξη της φόρμας που θέλω. Στο παράδειγμά μας επιλέγουμε τη διάταξη της *στήλης* για λόγους ευχρηστίας .

Ομοίως κάνουμε κλικ στο *επόμενο>* και περνάμε στην επόμενη σελίδα στην οποία και επιλέγουμε το επιθυμητό φόντο που θα θέλαμε να έχει η φόρμα μας . Στο παράδειγμά μας επιλέξαμε το φόντο με την επωνυμία *πολυχρωμία* .

Σε περίπτωση που θέλω να προβώ στην προηγούμενη σελίδα για να διορθώσω ή να ελέγξω κάτι κάνω απλά κλικ στο κουμπί *<προηγούμενο* .

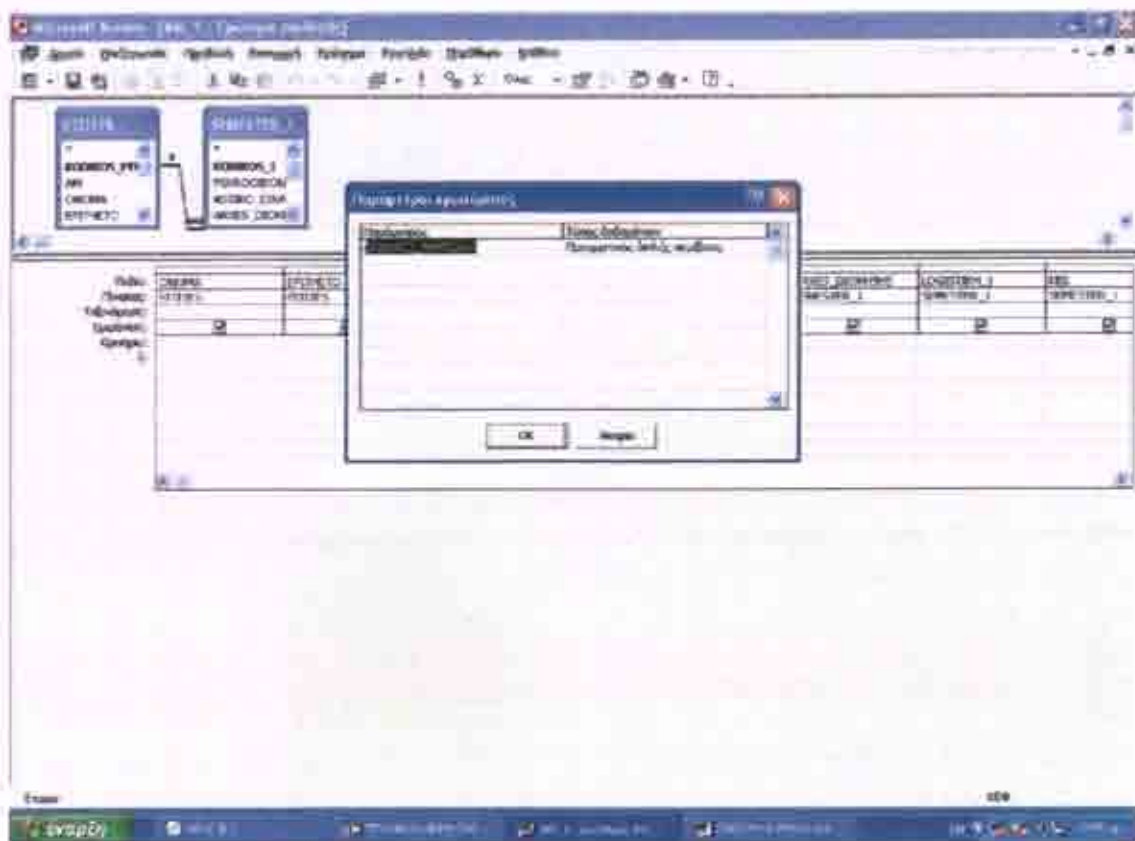
Στην επόμενη και τελευταία σελίδα του οδηγού μας δίνω όνομα στη φόρμα μου και επιλέγω την περίπτωση *άνοιγμα της φόρμας για την προβολή ή την καταχώρηση πληροφοριών* . Για να ολοκληρωθεί η διαδικασία δημιουργίας της φόρμας κάνω κλικ στο τέλος . Έτσι μου εμφανίζεται η φόρμα μου *FITITES*.



Ένας άλλος τρόπος για τη δημιουργία φόρμών είναι από τη βάση δεδομένων και έχοντας μπει στις φόρμες να κάνω κλικ στο κουμπί *δημιουργία* και στη συνέχεια να επιλέξω οδηγό φόρμών . Στην ίδια σελίδα επιλέγω το εξάμηνο βάση στο οποίο θα κάνω την φόρμα μου.

Έτσι στο παράδειγμά μας ολοκληρώνουμε την φόρμα fitites. Ομοίως εργαζόμαστε και για τις υπόλοιπες φόρμες εξάμηνο 1, εξάμηνο 2, εξάμηνο 3, εξάμηνο 4, εξάμηνο 5, εξάμηνο 6, εξάμηνο 7 και πτυχίο.

Ως κριτήριο βάλουμε τον αριθμό μητρώου όπως και επίσης τον ορίσαμε και ως παράμετρο. Για να γίνει αυτό πατάμε δεξί κλικ στο κενό και επιλέγουμε την παράμετρο. Δίνουμε όνομα και ως τύπο δεδομένων επιλέγουμε τον πραγματικό διπλής ακρίβειας.

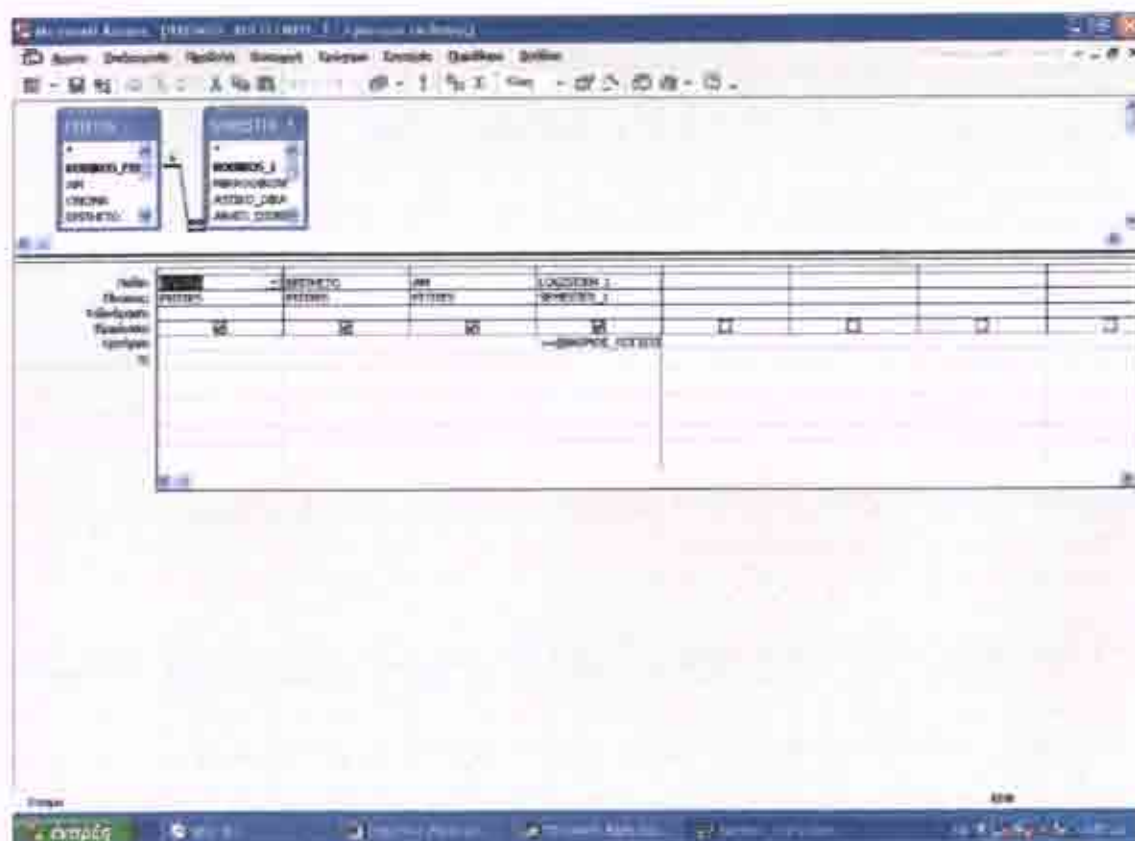


Η σχέση που δημιουργήθηκε και συνέδεσε τα πεδία φαίνεται στην προβολή SQL πατώντας δεξί κλικ.

```
PARAMETERS ΑΡΙΘΜΟΣ_ΜΗΤΡΩΟΥ IEEEDouble;
SELECT FITITES.ONOMA, FITITES.EPITHETO, FITITES.AM,
SEMESTER_1.MIKROOIKONOMIKH, SEMESTER_1.ASTIKO_DIKAIO,
SEMESTER_1.ARXES_DIOKHSHS, SEMESTER_1.LOGISTIKH_1,
SEMESTER_1.KBS, SEMESTER_1.MA8HMATIKA,
SEMESTER_1.KODIKOS_FITITI, SEMESTER_1.M_O1
FROM FITITES INNER JOIN SEMESTER_1 ON
FITITES.KODIKOS_FITITI = SEMESTER_1.KODIKOS_FITITI
WHERE (((FITITES.AM)=[ΑΡΙΘΜΟΣ_ΜΗΤΡΩΟΥ]));
```

Για να φανεί το ερώτημα που δημιουργήσαμε στην φόρμα, επιλέγουμε ένα κουμπί εντολής στη προβολή σχεδίασης και το τοποθετούμε πάνω στην υποφόρμα για την οποία γίνεται λόγος. Επιλέγουμε την εκτέλεση ερωτήματος και αφού δηλώσουμε το συγκεκριμένο ερώτημα που θέλουμε να εκτελείται με αυτό το κουμπί και του δώσουμε μια ονομασία τότε το ερώτημα μπαίνει σε εφαρμογή κάθε φορά που πατάμε κλικ πάνω του. Δηλαδή εμφανίζεται ένα πινακάκι το οποίο ζητάει να γραφτεί ο αριθμός μητρώου που μας ενδιαφέρει και όταν το πληκτρολογούμε εμφανίζεται ο πίνακας του φοιτητή με τα στοιχεία που προαναφέραμε.

Ένα άλλο ερώτημα που κρίναμε πως θα είναι χρήσιμο για μία γραμματεία είναι να ενημερώνεται για τους φοιτητές που έχουν κάποιο βαθμό (συγκεκριμένο ή γενικό) σε κάποιο μάθημα χωρίς να ψάχνει έναν-έναν τους φοιτητές. Για παράδειγμα μπορούμε να βρούμε ποιοι φοιτητές έχουν στο μάθημα λογιστική_1 μεγαλύτερο από βαθμό 6. Η διαδικασία για την δημιουργία του ερωτήματος έχει την ίδια λογική με το παραπάνω παράδειγμα μόνο που εδώ ως κριτήριο μπαίνει το \geq .

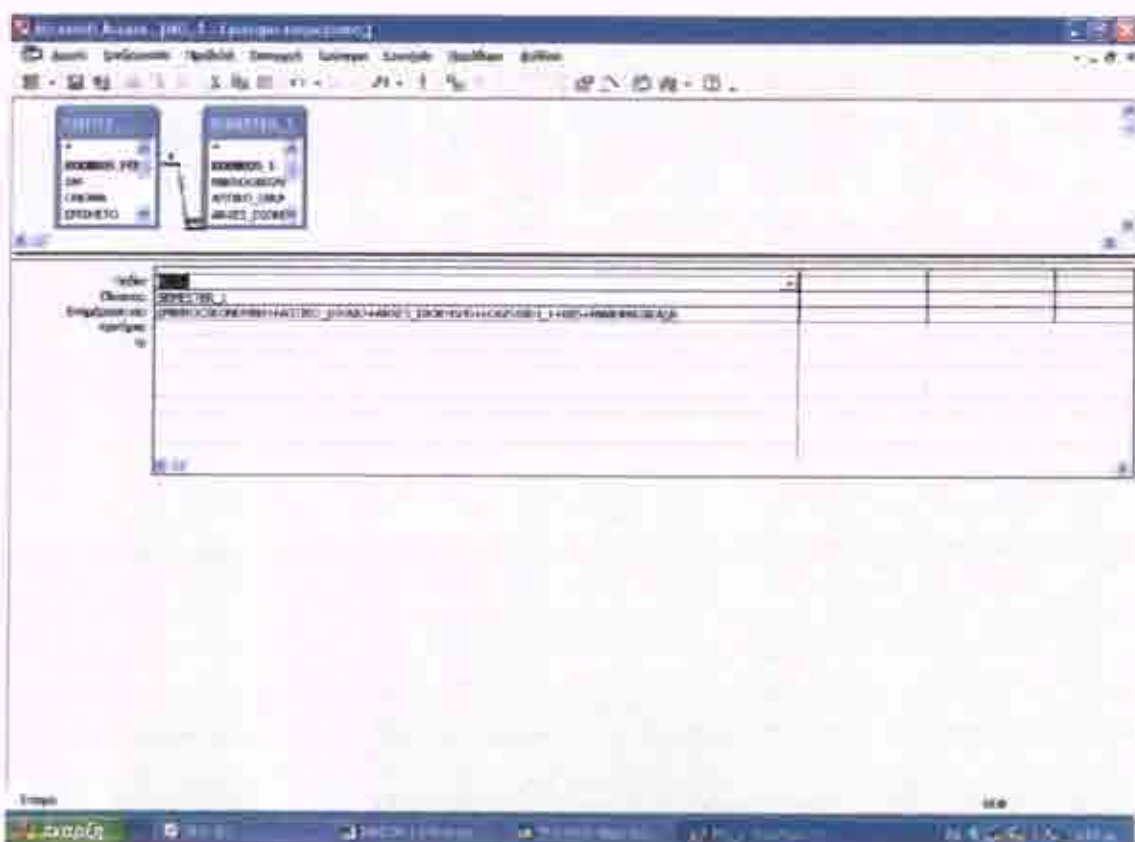


Η σχέση που δημιουργήθηκε και συνέδεσε τα πεδία φαίνεται στην προβολή SQL πατώντας δεξί κλικ.

```
PARAMETERS ΒΑΘΜΟΣ_ΛΟΓΙΣΤΙΚΗΣ_1 IEEEDouble;
SELECT FITITES.ΟΝΟΜΑ, FITITES.ΕΡΠΙΘΗΤΟ, FITITES.ΑΜ,
SEMESTER_1.ΛΟΓΙΣΤΙΚΗ_1
FROM FITITES INNER JOIN SEMESTER_1 ON
FITITES.ΚΟΔΙΚΟΣ_FITITI=SEMESTER_1.ΚΟΔΙΚΟΣ_FITITI
WHERE
(((SEMESTER_1.ΛΟΓΙΣΤΙΚΗ_1)>=[ΒΑΘΜΟΣ_ΛΟΓΙΣΤΙΚΗΣ_1]));
```

Έτσι κάθε φορά που θα εμφανίζεται το πίνακάκι και θα μας ζητείται ο βαθμός που μας ενδιαφέρει εννοείται ότι το πρόγραμμα θα ψάξει να βρει τους φοιτητές που έχουν μεγαλύτερο ή ίσο βαθμό από αυτό που του δώσαμε.

Ένα απαραίτητο στοιχείο για το φοιτητολόγιο είναι να ενημερώνονται οι φόρμες για τους μέσους όρους των εξαμήνων αυτόματα έτσι ώστε να αποφευχθεί η χρήση αριθμομηχανής. Για να γίνει αυτό χρησιμοποιούμε τα ερωτήματα και τις μακροεντολές. Ας δούμε ένα-ένα τα βήματα για την δημιουργία των μέσων όρων ώστε να γίνει κατανοητή η λειτουργία του. Αρχικά δουλεύουμε στα ερωτήματα όπου επιλέγουμε τους πίνακες που μας ενδιαφέρουν. Στο συγκεκριμένο παράδειγμα για το πρώτο εξάμηνο επιλέγουμε τον πίνακα FITITES και τον πίνακα SEMESTER_1. Πατώντας δεξί κλικ επιλέγουμε να φτιάξουμε ερώτημα ενημέρωσης. Σαν πεδίο ορίζουμε τον μέσο όρο του πρώτου εξαμήνου (ΜΟ_1) και η ενημέρωση γίνεται πάνω σε όλα τα μαθήματα του εξαμήνου δια το συνολικό αριθμό των μαθημάτων.



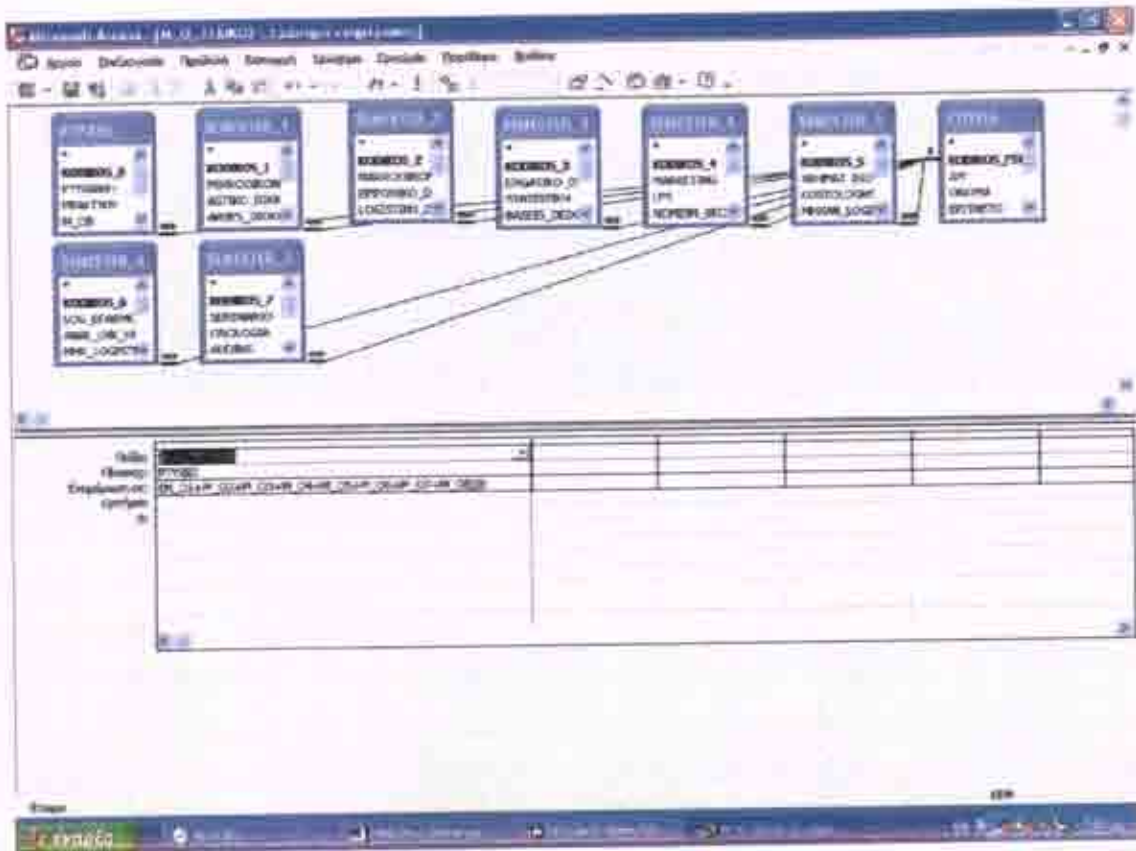
Και σε αυτή την περίπτωση η σχέση που φτιάξαμε φαίνεται στην προβολή SQL.

```

.....
UPDATE FITITES INNER JOIN SEMESTER_1 ON
FITITES.KODIKOS_FITITI=SEMESTER_1.KODIKOS_FITITI SET
SEMESTER_1.M_01 =
(MIKROOIKONΟΜΙΚΗ+ΑΣΤΙΚΟ_ΔΙΚΑΙΟ+ΑΡΧΕΣ_ΔΙΟΚΗΣΗΣ+ΛΟΓΙΣ
ΤΙΚΗ_1+KBS+ΜΑΘΗΜΑΤΙΚΑ)/6;
.....

```

Αφού εκτελέσουμε το ερώτημα προχωράμε στη δημιουργία της μακροεντολής. Η ενέργεια που θα ακολουθήσουμε είναι το άνοιγμα ερωτήματος και επιλέγουμε για ονομασία το ακριβές όνομα που είχαμε δώσει στη δημιουργία ερωτήματος προηγουμένως.



Και σε αυτή την περίπτωση η σχέση που φτιάξαμε φαίνεται στην προβολή SQL.

```

UPDATE (((((((FITITES INNER JOIN PTYXIO ON
FITITES.KODIKOS_FITITI=PTYXIO.KODIKOS_FITITI) INNER JOIN
SEMESTER_1 ON
FITITES.KODIKOS_FITITI=SEMESTER_1.KODIKOS_FITITI) INNER
JOIN SEMESTER_2 ON
FITITES.KODIKOS_FITITI=SEMESTER_2.KODIKOS_FITITI) INNER
JOIN SEMESTER_3 ON
FITITES.KODIKOS_FITITI=SEMESTER_3.KODIKOS_FITITI) INNER
JOIN SEMESTER_4 ON
FITITES.KODIKOS_FITITI=SEMESTER_4.KODIKOS_FITITI) INNER
JOIN SEMESTER_5 ON
FITITES.KODIKOS_FITITI=SEMESTER_5.KODIKOS_FITITI) INNER
JOIN SEMESTER_6 ON
FITITES.KODIKOS_FITITI=SEMESTER_6.KODIKOS_FITITI) INNER
JOIN SEMESTER_7 ON
FITITES.KODIKOS_FITITI=SEMESTER_7.KODIKOS_FITITI SET
PTYXIO.M_O_TELIKOS =
(M_O1+M_O2+M_O3+M_O4+M_O5+M_O6+M_O7+M_O8)/8;

```

Η μακροεντολή καθώς και το κουμπί της αποθήκευσης στην σχεδίαση φόρμας δημιουργούνται όπως περιγράψαμε πιο πάνω.

2.5 ΕΚΘΕΣΕΙΣ

Στο τελευταίο μας κεφάλαιο θα αναφερθούμε στις εκθέσεις. Θεωρήσαμε πως το πρόγραμμα θα ήταν πιο ολοκληρωμένο με την κατασκευή κάποιων εκθέσεων. Όπως είναι γνωστό οι εκθέσεις χρησιμοποιούνται σε περίπτωση που θέλουμε να εκτυπώσουμε κάποια στοιχεία μας. Φυσικά είναι δυνατή και η εκτύπωση μέσω των φορμών και των πινάκων αλλά δεν είναι το ίδιο εμφανίσιμη και πρακτική. Οι εκθέσεις μας δίνουν την δυνατότητα να καθορίσουμε εμείς τα πεδία και τον τρόπο εκείνο που θέλουμε να εκτυπωθούν. Για παράδειγμα μπορούμε να ορίσουμε κάποιο κριτήριο, βάση του οποίου θα εμφανίζονται τα υπόλοιπα στοιχεία στην εκτύπωση.

Στο παράδειγμά μας λοιπόν κρίναμε απαραίτητο να φτιάξουμε εννέα εκθέσεις. Η πρώτη έκθεση θα αφορά τον πίνακα των φοιτητών και συνεπώς θα περιέχει όλα τα στοιχεία των φοιτητών. Όπως είναι λογικό αυτή η έκθεση κρίνεται αρκετά χρήσιμη καθώς ανά πάσα στιγμή μπορεί να χρειαστεί μια λίστα με τα στοιχεία των φοιτητών.

Όσον αφορά τις υπόλοιπες εκθέσεις των εξαμήνων, αυτές δεν θα βασιστούν πάνω στους πίνακες των αντίστοιχων εξαμήνων αφού κρίνονται αρκετά ελλειπτικές. Έτσι αποφασίστηκε οι εκθέσεις αυτές να βασιστούν πάνω στα αντίστοιχα ερωτήματα των εξαμήνων του αριθμού μητρώου. Κατά την εκτύπωση λοιπόν των εκθέσεων αυτών θα ζητείται αριθμός μητρώου και αφού θα επιλέγουμε τον μαθητή για τον οποίο ενδιαφερόμαστε η έκθεση μας θα είναι έτοιμη να εκτυπωθεί έχοντας ως στοιχεία το ονοματεπώνυμό του, τον αριθμό μητρώου του και τους βαθμούς του εξαμήνου.

Στη συνέχεια θα προχωρήσουμε στην δημιουργία των εκθέσεων και θα ξεκινήσουμε με αυτή των φοιτητών. Βρισκόμαστε στην βάση δεδομένων και επιλέγουμε τις εκθέσεις. Επιλέγουμε την

Όσον αφορά τις επόμενες εκθέσεις η διαδικασία είναι ίδια μόνο που δεν επιλέγουμε να την βασίσουμε σε κάποιον πίνακα αλλά στο ερώτημα του αριθμού μητρώου του κάθε εξαμήνου αντίστοιχα. Επίσης αλλάζει το στυλ που είναι απαλό γκρι και στην επιλογή των στοιχείων δεν ορίζουμε κανένα.

Στο παράρτημά μας περιλαμβάνονται η έκθεση των φοιτητών καθώς και οι εκθέσεις για όλα τα εξάμηνα ενός συγκεκριμένου φοιτητή.

ΚΕΦΑΛΑΙΟ 3

Αφού κάναμε την μελέτη μας πάνω στην Access καλό θα ήταν να εξετάσουμε και κάποιες άλλες γλώσσες προγραμματισμού όπως είναι η PL/SQL και η ORACLE. Θα προσπαθήσουμε όσο το δυνατόν να γίνουν κατανοητές οι γλώσσες αυτές προς τον κάθε αναγνώστη της εργασίας αυτής ξεκινώντας από την PL/SQL.

ΒΑΣΙΚΗ ΔΟΜΗ

Η PL/SQL (Procedural Language/SQL) επεκτείνει την SQL προσθέτοντας δομές που βρίσκονται συνήθως σε διαδικαστικές γλώσσες προγραμματισμού (Pascal, PL/I). Το αποτέλεσμα είναι μια δομημένη γλώσσα με περισσότερες δυνατότητες από την SQL. Η βασική μονάδα της PL/SQL είναι το *block*. Όλα τα προγράμματα σε PL/SQL αποτελούνται από blocks τα οποία μπορούν να είναι εμφωλιασμένα. Κάθε block εκτελεί μια λογική πράξη μέσα στο πρόγραμμα. Ένα block έχει την εξής δομή:

```
DECLARE
/* Declarative section: Μεταβλητές, τύποι και τοπικά
υποπρογράμματα.
*/
BEGIN
/* Executable section: διαδικασίες και οι εντολές PL/SQL. */
/* Είναι το μόνο τμήμα που απαιτείται. */
EXCEPTION
/* Exception handling section: εντολές χειρισμού λαθών*/
END;
```

Το μόνο τμήμα που απαιτείται είναι το *executable*. Τα άλλα είναι προαιρετικά. Οι μόνες εντολές της SQL που επιτρέπονται σε ένα

πρόγραμμα PL/SQL είναι οι : *SELECT, INSERT, UPDATE, DELETE* καθώς και κάποιες άλλες εντολές χειρισμού δεδομένων.

Η εντολή *SELECT* έχει μια ειδική μορφή στην οποία μια ν-άδα αποθηκεύεται σε μεταβλητές. Στην PL/SQL δεν επιτρέπονται εντολές ορισμού δεδομένων (data definition statements), όπως οι *CREATE, DROP* και *ALTER*. Το executable section περιέχει επίσης δομές όπως αναθέσεις, διακλαδώσεις (if-statements), βρόγχους(loops), κλήσεις διαδικασιών και μια ειδική δομή η trigger. Η PL/SQL δεν κάνει διαχωρισμό κεφαλαίων-μικρών. Τα σχόλια είναι όπως στη C (*/* */*).

Για να εκτελέσουμε ένα πρόγραμμα PL/SQL πρέπει να γράψουμε μετά τον κώδικα του προγράμματος:

- Μια γραμμή που να περιέχει μόνο μια τελεία (".") και μετά
- Μια γραμμή με την λέξη run;

Μπορούμε να εκτελέσουμε προγράμματα PL/SQL μέσα από την sqlplus είτε γράφοντας τις εντολές στην γραμμή διαταγών (SQL>), είτε γράφοντας τον κώδικα σε ένα αρχείο και καλώντας το από την sqlplus.

ΜΕΤΑΒΛΗΤΕΣ ΚΑΙ ΤΥΠΟΙ

Πληροφορίες μεταξύ της βάσης δεδομένων και ενός προγράμματος PL/SQL μεταδίδονται μέσω μεταβλητών η κάθε μια από τις οποίες έχει και έναν ξεχωριστό τύπο. Αυτός ο τύπος μπορεί να είναι :

- Ένας από τους τύπους που χρησιμοποιεί η SQL για να προσδιορίσει τις στήλες ενός πίνακα.
- Ένας γενικός τύπος που χρησιμοποιείται από την PL/SQL και λέγεται NUMBER.
- Ίδιος με τον τύπο μιας στήλης ενός πίνακα

Ο πιο πολυχρησιμοποιημένος γενικός τύπος είναι ο NUMBER. Μεταβλητές αυτού του τύπου μπορούν να είναι είτε ακέραιοι είτε πραγματικοί αριθμοί. Ο πιο συνηθισμένος τύπος δήλωσης συμβολοσειρών είναι ο VARCHAR(n) όπου n είναι το μέγιστο μήκος της συμβολοσειράς σε bytes. Το μήκος είναι απαραίτητο να δηλωθεί και δεν υπάρχει default τιμή. Για παράδειγμα μπορούμε να δηλώσουμε :

```
DECLARE
  price NUMBER;
  myBeer VARCHAR(20);
```

Οι τύποι της PL/SQL είναι περίεργοι. Κάποιοι συγκεκριμένοι τύποι της SQL δεν μπορούν να παραμετροποιηθούν στην PL/SQL. Για παράδειγμα η δήλωση CHAR(50) δεν λειτουργεί σωστά. Επίσης σημειώστε ότι η PL/SQL επιτρέπει τη χρήση BOOLEAN μεταβλητών, αν και η Oracle δεν υποστηρίζει BOOLEAN ως τύπο για τις στήλες ενός πίνακα.

Σε πολλές περιπτώσεις μια PL/SQL μεταβλητή θα χρησιμοποιείται για τον χειρισμό αποθηκευμένων δεδομένων σε μια υπάρχουσα σχέση. Σε αυτή την περίπτωση είναι ιδανικό οι μεταβλητές να έχουν τον ίδιο τύπο με την αντίστοιχη στήλη της σχέσης. Εάν οι τύποι των μεταβλητών δεν συμφωνούν, οι αναθέσεις και οι συγκρίσεις μεταβλητών δεν θα λειτουργήσουν όπως περιμένετε. Για να είστε σίγουροι, θα πρέπει να χρησιμοποιείται τον τελεστή %TYPE. Για παράδειγμα :

```
DECLARE
  myBeer Beers.name%TYPE;
```

δίνει στην PL/SQL μεταβλητή myBeer τον τύπο που είχε δηλωθεί η name στήλη στη σχέση Beers.

Μια μεταβλητή επίσης μπορεί να έχει τύπο που είναι μια εγγραφή (record) με πολλά πεδία (fields). Ο πιο απλός τρόπος να

δηλώσετε μια τέτοια μεταβλητή είναι να χρησιμοποιήσετε τον τελεστή %ROWTYPE στο σχεσιακό όνομα. Το αποτέλεσμα είναι ένας τύπος εγγραφής όπου τα πεδία έχουν τα ίδια ονόματα και τους ίδιους τύπους όπως οι μεταβλητές της σχέσης. Για παράδειγμα :

```
DECLARE
  beerTuple Beers%ROWTYPE;
```

κάνει τη μεταβλητή beerTuple να είναι μια εγγραφή με πεδία name και manufacture, υποθέτοντας ότι η σχέση έχει την μορφή Beers(name, manufacture).

Η αρχική τιμή κάθε μεταβλητής άσχετα από τον τύπο της είναι NULL . Μπορούμε να αναθέσουμε τιμές σε μεταβλητές, χρησιμοποιώντας τον τελεστή ":=". Η ανάθεση μπορεί να γίνει είτε αμέσως με την δήλωση του τύπου της μεταβλητής, ή οπουδήποτε μέσα στο τμήμα εκτέλεσης του προγράμματος. Ένα παράδειγμα :

```
DECLARE
  a NUMBER := 3;
BEGIN
  a := a + 1;
END;
run;
```

Αυτό το πρόγραμμα δεν έχει κανένα αποτέλεσμα όταν τρέχει, επειδή δεν γίνονται αλλαγές στην βάση.

ΑΠΛΑ ΠΡΟΓΡΑΜΜΑΤΑ ΣΕ PL/SQL

Η πιο απλή μορφή προγράμματος έχει κάποιους ορισμούς οι οποίοι ακολουθούνται από ένα εκτελέσιμο τμήμα, που αποτελείται από μια ή περισσότερες SQL εντολές με τις οποίες έχετε εξοικειωθεί.

Η σύνταξη της εντολής SELECT είναι διαφορετική από αυτή της SQL. Μετά τη δεσμευμένη λέξη SELECT πρέπει οπωσδήποτε να βάλουμε την δεσμευμένη λέξη INTO με μια λίστα από μεταβλητές, μια για κάθε στήλη που θέλουμε να προβάσουμε. Σε αυτές τις μεταβλητές θα αποθηκευτεί το αποτέλεσμα της ερώτησης (ν-άδα). Το αποτέλεσμα της SELECT σε αυτήν την περίπτωση πρέπει να είναι μια ν-άδα και όχι περισσότερες.

Σε περίπτωση που η εντολή επιστρέφει περισσότερες από μια ν-άδες, χρειαζόμαστε μια άλλη δομή για την αποθήκευση τους (cursors). Για παράδειγμα :

```
CREATE TABLE T1(  
  e INTEGER,  
  f INTEGER  
);  
DELETE FROM T1;  
INSERT INTO T1 VALUES(1, 3);  
INSERT INTO T1 VALUES(2, 4);  
/* Το παραπάνω είναι απλή SQL. Αυτό που ακολουθεί είναι  
PL/SQL.*/  
DECLARE  
  a NUMBER;  
  b NUMBER;  
BEGIN  
  SELECT e,f INTO a,b FROM T1 WHERE e>1;  
  INSERT INTO T1 VALUES(b,a);  
END;  
  
run;
```

Στο παραπάνω παράδειγμα υπήρχε μόνο μια δυάδα που επέστρεφε η SELECT η (2, 4). Η εντολή INSERT εισάγει μια νέα γραμμή στον πίνακα T1, την (4, 2).

ΡΟΗ ΕΛΕΓΧΟΥ ΣΤΗΝ PL/SQL

Η PL/SQL επιτρέπει στον χρήστη να ελέγχει τη ροή του προγράμματος, με τρόπο παρόμοιο με τις κλασσικές διαδικαστικές γλώσσες προγραμματισμού.

Μια εντολή IF συντάσσεται ως εξής:

```
IF <condition> THEN <statement_list> ELSE <statement_list> END IF;
```

Το τμήμα ELSE είναι προαιρετικό. Εάν θέλουμε μια πολλαπλή διακλάδωση της ροής χρησιμοποιούμε και το ELSIF.

```
IF <condition_1> THEN ...  
ELSIF <condition_2> THEN ...  
... ..  
ELSIF <condition_n> THEN ...  
ELSE ...  
END IF;
```

Το παράδειγμα που ακολουθεί, είναι το προηγούμενο λίγο διαφοροποιημένο. Αυτό που κάνουμε είναι να εκτελούμε την εισαγωγή της δυάδας στον πίνακα, μόνο εάν το δεύτερο στοιχείο της είναι μεγαλύτερο του 1. Αλλιώς πρώτα προσθέτουμε 10 σε κάθε στοιχείο της δυάδας και μετά την εισάγουμε στον πίνακα:

```

DECLARE
  a NUMBER;
  b NUMBER;
BEGIN
  SELECT e,f INTO a,b FROM T1 WHERE e>1;
  IF b=1 THEN
    INSERT INTO T1 VALUES(b,a);
  ELSE
    INSERT INTO T1 VALUES(b+10,a+10);
  END IF;
END;
run;

```

Οι επαναλήψεις (loops) γίνονται με διάφορους τρόπους:

```

LOOP
  <loop_body> /*λίστα από PL/SQL εντολές. */
END LOOP;

```

Τουλάχιστον μια από τις εντολές μέσα στο βρόγχο επανάληψης, πρέπει να είναι της μορφής:

```
EXIT WHEN <condition>;
```

Η επανάληψη σταματάει εάν η συνθήκη <condition> είναι αληθής. Για παράδειγμα ακολουθεί ένας τρόπος για να εισάγουμε όλες τις γραμμές από (1, 1) μέχρι την (100, 100) στον πίνακα T1.

```

DECLARE
  i NUMBER := 1;
BEGIN
  LOOP
    INSERT INTO T1 VALUES(i,i);
    i := i+1;
    EXIT WHEN i>100;
  END LOOP;
END;

```

run;

Μερικές από τις χρήσιμες εντολές που χρησιμοποιούνται για την δημιουργία δομών επανάληψης στην PL/SQL είναι οι ακόλουθες:

1. EXIT από μόνη της προκαλεί την έξοδο από το loop, χωρίς καμία συνθήκη.
2. Η εντολή WHILE, η οποία συντάσσεται ως εξής:

```
WHILE          <condition>          LOOP
    <loop_body>
END LOOP;
```

3. Η εντολή FOR, η οποία συντάσσεται ως εξής:

```
FOR    <var>    IN    <start>..<finish>    LOOP
    <loop_body>
END LOOP;
```

Η μεταβλητή <var> μπορεί να είναι οποιαδήποτε μεταβλητή. Είναι τοπική στο σώμα του βρόγχου και δεν χρειάζεται να δηλωθεί. Επίσης τα <start> και <finish> είναι σταθερές.

CURSORS

Ο *cursor* είναι μια μεταβλητή με την οποία μπορούμε να διατρέχουμε n-άδες κάποιας σχέσης. Η σχέση αυτή μπορεί να είναι ένας πίνακας, ή το αποτέλεσμα μιας ερώτησης. Με το να αποθηκεύουμε n-άδες στην δομή cursor, είναι δυνατό να φτιάχνουμε προγράμματα που να διαβάζουν και να επεξεργάζονται τις τιμές κάθε στοιχείου αυτών των n-άδων. Εάν η σχέση είναι αποθηκευμένη (πίνακας), μπορούμε να αλλάξουμε τις τιμές αυτές ή να διαγράψουμε τελείως τις n-άδες.

Το παράδειγμα που ακολουθεί παρουσιάζει μια επανάληψη χρησιμοποιώντας cursor. Χρησιμοποιεί τη σχέση T1(e, f), της οποίας οι ν-άδες είναι ζεύγη ακεραίων. Το πρόγραμμα αυτό διαγράφει κάθε δυάδα που το πρώτο της στοιχείο είναι μικρότερο από το δεύτερο και εισάγει στον T1 την ανάποδη δυάδα:

```
1) DECLARE
   /* Μεταβλητές που θα αποθηκεύσουν το αποτέλεσμα της
   ερώτησης:*/
2)   a T1.e%TYPE;
3)   b T1.f%TYPE;
   /* Δήλωση του cursor: */
4)   CURSOR T1Cursor IS
5)     SELECT e, f
6)     FROM T1
7)     WHERE e < f
8)     FOR UPDATE;
9) BEGIN
10)  OPEN T1Cursor;
11)  LOOP
   /* Τοποθέτησε κάθε δυάδα του cursor σε PL/SQL μεταβλητές: */
12)    FETCH T1Cursor INTO a, b;
   /* Εάν δεν υπάρχουν άλλες δυάδες στον cursor σταμάτα το
   loop:*/
13)    EXIT WHEN T1Cursor%NOTFOUND;
   /* Διέγραψε την τρέχουσα δυάδα: */
14)    DELETE FROM T1 WHERE CURRENT OF T1Cursor;
   /* Εισήγαγε την αντίστροφη της στον πίνακα : */
15)    INSERT INTO T1 VALUES(b, a);
16)  END LOOP;
   /* Ελευθέρωσε τον cursor.*/
17)  CLOSE T1Cursor;
18) END;
19) .
20) run;
```

Ας περιγράψουμε λίγο το παραπάνω πρόγραμμα :

- Η γραμμή (1) σηματοδοτεί την περιοχή των δηλώσεων.

- Οι γραμμές (2) και (3) δηλώνουν τις μεταβλητές a και b με τύπους ίδιους με αυτούς των στηλών του πίνακα T1(e, f).

Αν και ξέρουμε ότι ο τύπος των στηλών του πίνακα T1 είναι INTEGER, δηλώνουμε τις μεταβλητές a και b έτσι ώστε να έχουν τον ίδιο τύπο με τις στήλες του T1.

- Στις γραμμές από (4) έως και (8) ορίζουμε τον cursor T1cursor. Ο ορισμός του γίνεται με μια ερώτηση SELECT-FROM-WHERE, η οποία επιλέγει από τις δυάδες του πίνακα, αυτές που το πρώτο τους όρισμα είναι μικρότερο από το δεύτερο. Στη γραμμή (8) δηλώνουμε ότι ο cursor είναι FOR UPDATE, εφόσον τροποποιούμε τον T1 χρησιμοποιώντας δυάδες του. Η δήλωση αυτή δε χρειάζεται αν ο cursor δεν πρόκειται να επηρεάσει n-άδες του πίνακα.

- Η γραμμή (9) ξεκινά το εκτελέσιμο τμήμα του προγράμματος.

- Η γραμμή (10) ανοίγει τον cursor και αποθηκεύει σ' αυτόν τις δυάδες του πίνακα T1 που επιλέχθηκαν από την ερώτηση SELECT-FROM-WHERE.

- Οι γραμμές (11) έως (16) αποτελούν ένα PL/SQL loop, που γίνεται ανάμεσα στις εντολές LOOP και END LOOP. Μέσα σε αυτό το loop γίνονται τα εξής:
 1. Μια εντολή FETCH, που μεταφέρει μια δυάδα του cursor σε δυο PL/SQL μεταβλητές. Γενικά η εντολή FETCH πρέπει να έχει μεταβλητές όσα και τα στοιχεία των n-άδων του cursor και με τους ίδιους τύπους.
 2. Στη γραμμή (13) γίνεται ένας έλεγχος για την έξοδο από το loop. Αυτός ο έλεγχος, το όνομα του cursor ακολουθούμενο από %NOTFOUND, είναι αληθής έχουμε προσπελάσει όλες τις n-άδες του cursor.
 3. Στη γραμμή (14) διαγράφω την τρέχουσα δυάδα του cursor από τον πίνακα T1. Αυτό γίνεται με μια ειδική συνθήκη στην εντολή DELETE, την WHERE CURRENT OF T1Cursor.
 4. Στην γραμμή (15) εισάγουμε στον πίνακα την αντίστροφη n-άδα που μόλις έχουμε διαβάσει από τον cursor.

- Η γραμμή (17) κλείνει τον cursor.

- Η γραμμή (18) τερματίζει το PL/SQL πρόγραμμα.
- Οι γραμμές (19) έως (20) προκαλούν την εκτέλεση του προγράμματος.

ΔΙΑΔΙΚΑΣΙΕΣ (PROSEDURES)

Οι διαδικασίες της PL/SQL συμπεριφέρονται παρόμοια με τις διαδικασίες σε άλλες γλώσσες προγραμματισμού. Για παράδειγμα μια συνάρτηση `addtuple1` που δοθέντος ενός ακέραιου `i` εισάγει την δυάδα (`i`, `'xxx'`) σε έναν πίνακα `T2` :

```
CREATE TABLE T2 (  
  a INTEGER,  
  b CHAR(10)  
);  
  
CREATE PROCEDURE addtuple1(i IN NUMBER) AS  
BEGIN  
  INSERT INTO T2 VALUES(i, 'xxx');  
END addtuple1;  
  
run;
```

Η διαδικασία ορίζεται χρησιμοποιώντας τις λέξεις `CREATE PROCEDURE` ακολουθούμενες από το όνομα της διαδικασίας και τις παραμέτρους τις.

Σε μια διαδικασία υπάρχει ένας αριθμός παραμέτρων που ακολουθείτε από ένα `mode` και τον τύπο της παραμέτρου. Οι δυνατοί `modes` είναι `IN` (μόνο ανάγνωση), `OUT` (μόνο εγγραφή), και `INOUT` (ανάγνωση και εγγραφή). Οι τύποι των μεταβλητών που είναι παράμετροι μιας διαδικασίας δεν πρέπει να έχουν όρια. Για παράδειγμα `CHAR(10)` και `VARCHAR(20)` δεν επιτρέπονται να

περνάνε σαν παράμετροι. Πρέπει να χρησιμοποιήσουμε CHAR και VARCHAR αντίστοιχα.

Τον ορισμό μιας διαδικασίας ακολουθεί η λέξη AS (ή IS) και μετά ακολουθεί το σώμα της διαδικασίας που είναι ένα PL/SQL block. Η επανάληψη του ονόματος της διεργασίας στο END είναι προαιρετική. Η διαφορά από ένα block είναι ότι δεν χρειάζεται η λέξη DECLARE αλλά οι μεταβλητές ορίζονται αμέσως μετά το AS.

```
... AS
<local_var_declarations>
BEGIN
  <procedure_body>
END;

run;
```

Η εντολή run στο τέλος απλώς δημιουργεί τη διαδικασία, δεν την εκτελεί. Η εκτέλεση γίνεται σε ένα άλλο PL/SQL block ως εξής:

```
BEGIN addtuple1(99); END;

run;
```

Η διαδικασία που ακολουθεί εισάγει μια δυάδα στον πίνακα T2.

```
CREATE PROCEDURE addtuple2(
  x T2.a%TYPE,
  y T2.b%TYPE)
AS
BEGIN
  INSERT INTO T2(a, b)
  VALUES(x, y);
END addtuple2;

run;
```

Τώρα με αυτή τη διαδικασία θα εισάγουμε την δυάδα (10, 'abc').

```
BEGIN
```



```

    addtuple2(10, 'abc');
END;
.
run;

```

Το ακόλουθο παράδειγμα δείχνει τη χρήση μια OUT παραμέτρου:

```

CREATE TABLE T3 (
    a INTEGER,
    b INTEGER
);
CREATE PROCEDURE addtuple3(a NUMBER, b OUT NUMBER)
AS
BEGIN
    b := 4;
    INSERT INTO T3 VALUES(a, b);
END;
.
run;
DECLARE
    v NUMBER;
BEGIN
    addtuple3(10, v);
END;
.
run;

```

Όπως φαίνεται στο παραπάνω παράδειγμα η παράμετρος b χρησιμοποιείται μόνο για εγγραφή. Κατά την κλήση της διαδικασίας χρησιμοποιούμε μια μεταβλητή v για να αποθηκεύσουμε την παράμετρο b. Μετά την κλήση της διαδικασίας η μεταβλητή v έχει την τιμή 4.

Στην PL/SQL μπορούμε να γράφουμε συναρτήσεις αντί για διαδικασίες. Στον ορισμό της συνάρτησης πρέπει να ακολουθήσουμε τη λίστα με παραμέτρους της συνάρτησης, με RETURN και με τον τύπο που επιστρέφει η συνάρτηση:

```

CREATE FUNCTION <func_name>(<param_list>) RETURN
<return_type> AS ...

```

Στο σώμα της συνάρτησης η εντολή RETURN <expression>; τερματίζει τη συνάρτηση και επιστρέφει μια τιμή.

Για να βρείτε ποιες διαδικασίες ή συναρτήσεις έχετε φτιάξει μέχρι τώρα πρέπει να γράψετε τα εξής:

```
select object_type, object_name
from user_objects
where object_type = 'PROCEDURE'
or object_type = 'FUNCTION';
```

Για να διαγράψετε μια διαδικασία ή συνάρτηση χρησιμοποιήστε την εντολή DROP.

```
drop procedure <procedure_name>;
drop function <function_name>;
```

TRIGGERS

Το trigger είναι ειδική δομή της PL/SQL παρόμοιο με τις procedures. Όμως, μια procedure καλείται ρητά από κάποιο block του προγράμματος μέσω καλεσμάτων, ενώ ένα trigger εκτελείται αυτόματα όποτε συμβαίνει κάποιο triggering event. Triggering event μπορεί να είναι κάποια από τις εντολές INSERT, DELETE ή UPDATE. Το timing μπορεί να είναι είτε BEFORE είτε AFTER. Το trigger μπορεί να είναι είτε row-level είτε statement-level, όπου το πρώτο εκτελείται μια φορά για κάθε γραμμή που επηρεάζεται από την triggering δήλωση και το δεύτερο μια φορά για όλη τη δήλωση.

Παρακάτω δίνεται η σύνταξη για τη δημιουργία ενός trigger στην Oracle (η σύνταξη έχει απλοποιηθεί και για την πλήρη έκδοση δοκιμάστε HELP CREATE TRIGGER στην sqlplus)

```
CREATE [OR REPLACE] TRIGGER <trigger_name>
  {BEFORE|AFTER} {INSERT|DELETE|UPDATE} ON <table_name>
  [FOR EACH ROW [WHEN (<trigger_condition>)]]
  <trigger_body>
```

Μερικές σημαντικές σημειώσεις

- Μπορείται να δημιουργήσετε μόνο BEFORE ή AFTER triggers για τα tables. (Οι INSTEAD OF triggers είναι διαθέσιμες μόνο για προβολές και τυπικά χρησιμοποιούνται μόνο για να δημιουργήσουν προβολές ενημερώσεων)
- Μπορείται να ορίσετε μέχρι τρία triggering events χρησιμοποιώντας την δεσμευμένη λέξη OR. Επίσης το UPDATE μπορεί προαιρετικά να ακολουθηθεί από τη δεσμευμένη λέξη OF και μια λίστα γνωρισμάτων στο <table_name>. Αν υπάρχει, ο όρος OF ορίζει το event να είναι μόνο ενημέρωση των γνωρισμάτων που υπάρχουν μετά το OF.

Εδώ υπάρχουν μερικά παραδείγματα

```
... INSERT ON R ...
... INSERT OR DELETE OR UPDATE ON R ...
... UPDATE OF A, B OR INSERT ON R ...
```

- Εάν έχει οριστεί η επιλογή FOR EACH ROW το trigger είναι row-level αλλιώς είναι statement-level.
- Για κάθε row-level trigger, περιορισμός στα trigger μπορεί να οριστεί στον όρο WHEN , ανάμεσα σε παρενθέσεις. Ο περιορισμός trigger είναι μια κατάσταση της SQL που πρέπει να ικανοποιηθεί ώστε η Oracle να μπορέσει να ενεργοποιήσει τα triggers. Αυτή η κατάσταση δεν μπορεί να περιέχει υποερωτήσεις. Χωρίς τον όρο WHEN ένα trigger ενεργοποιείται από κάθε triggering event
- <trigger_body> είναι ένα block της PL/SQL αντί για τα statements της SQL. Η Oracle έχει τοποθετήσει συγκεκριμένους περιορισμούς στο τι μπορείς να κάνεις μέσα στο <trigger_body> με σκοπό να αποφύγει περιπτώσεις όπου ένα trigger εκτελεί μια λειτουργία που ενεργοποιεί ένα άλλο trigger το οποίο ενεργοποιεί ένα τρίτο και ούτω καθ' εξής

πράγμα που θα δημιουργήσει ένα άπειρο loop. Οι περιορισμοί στο <trigger_body> περιλαμβάνουν

- Δεν μπορείς να αλλάξεις την ίδια σχέση της οποίας η μετατροπή θα ενεργοποιήσει το trigger.
- Δεν μπορείς να τροποποιήσεις μια σχέση που συνδέεται με την triggering σχέση

Διευκρινίζουμε την σύνταξη της Oracle μέσα από ένα παράδειγμα βασισμένο στους δυο παρακάτω πίνακες

```
CREATE TABLE T4 (a INTEGER, b CHAR(10));  
CREATE TABLE T5 (c CHAR(10), d INTEGER);
```

Δημιουργούμε ένα trigger που μπορεί να εισάγει ένα tuple στο T5 όταν ένα tuple έχει εισαχθεί στο T4. Συγκεκριμένα το trigger εξετάζει αν το καινούριο tuple έχει πρώτο όρο 10 ή λιγότερο και αν έχει τότε εισάγει το αντίστροφο tuple στο T5:

```
CREATE TRIGGER trig1  
  AFTER INSERT ON T4  
  FOR EACH ROW  
  WHEN (NEW.a <= 10)  
  BEGIN  
    INSERT INTO T5 VALUES(:NEW.b, :NEW.a);  
  END trig1;
```

```
run;
```

Οι μεταβλητές NEW και OLD είναι διαθέσιμες να αναφερθούν σε νέα ή παλιά tuples σχετικά. Σημείωση στο σώμα του trigger τα NEW και OLD πρέπει να έχουν πρόθεμα τον χαρακτήρα (:) εκτός και αν βρίσκονται μέσα στον όρο WHEN.

Ξανά σημειώστε ότι τελειώνουμε την δήλωση CREATE TRIGGER με . και run όπως με όλες της δηλώσεις της PL/SQL γενικά. Τρέχοντας την δήλωση CREATE TRIGGER κατασκευάζει μόνο ένα trigger και δεν το εκτελεί. Μόνο ένα trigger event όπως η εισαγωγή στο T4 στο παράδειγμα, μπορεί να ενεργοποιήσει την εκτέλεση.

Για να δείτε πληροφορίες για τα triggers χρησιμοποιείστε τα παρακάτω

```
select trigger_name from user_triggers;  
  
select trigger_type, table_name, triggering_event  
from user_triggers  
where trigger_name = '<trigger_name>';
```

Για να απορρίψετε ένα trigger

```
drop trigger <trigger_name>;
```

Για να απενεργοποιήσετε ή να ενεργοποιήσετε ένα trigger

```
alter trigger <trigger_name> {disable|enable};
```

ΑΝΑΚΑΛΥΠΤΟΝΤΑΣ ΛΑΘΗ

Η PL/SQL δεν σας λέει πάντα για τα λάθη της μεταγλώττισης. Αντιθέτως, σας δίνει ένα απλό μήνυμα όπως "procedure created with compilation errors". Αν δεν προσέξετε αμέσως το λάθος σας προσπαθείστε να δώσετε την εντολή

```
show errors procedure <procedure_name>;
```

Ομοίως μπορείτε να δείτε τα λάθη που σχετίζονται με την δημιουργία του trigger με

```
show errors trigger <trigger_name>;
```

Τέλος το "SHO ERR" είναι μια συντομογραφία του "SHOW ERRORS" και μπορείτε να προσπαθήσετε το "PROCEDURE..." ή το "TRIGGER ..." αν θέλετε μόνο να δείτε το πιο πρόσφατο λάθος μεταγλώττισης.

ΕΚΤΥΠΩΝΟΝΤΑΣ ΜΕΤΑΒΛΗΤΕΣ

Μερικές φορές θέλουμε να εκτυπώσουμε την τιμή μιας τοπικής μεταβλητής της PL/SQL. Ένας γρήγορος τρόπος είναι να την αποθηκεύσουμε σαν sole νιάδα κάποιας σχέσης και μετά την PL/SQL δήλωση να εκτυπώσουμε τη σχέση με μια δήλωση SELECT. Ένας άλλος τρόπος είναι να ορίσουμε μια bind μεταβλητή η οποία είναι ο μόνος τύπος που μπορεί να εκτυπωθεί με την εντολή print. Οι μεταβλητές bind είναι ο τύπος που πρέπει να έχει ως πρόθεμα το (:) στις δηλώσεις PL/SQL όπως :new που συζητήθηκε στην ενότητα **Triggers**.

Τα βήματα είναι τα εξής:

1. Ορίζουμε μια bind μεταβλητή ως ακολούθως

```
VARIABLE <name> <type>
```

όπου το τμήμα <type> μπορεί να είναι ένα από τα τρία: NUMBER, CHAR ή CHAR(n).

2. Μπορούμε να αναθέσουμε στην μεταβλητή μια τιμή σε μια ακόλουθη PL/SQL δήλωση αλλά πρέπει να βάλουμε πρόθεμα το (:) έξω από τη δήλωση.

3. Μπορούμε να εκτελέσουμε μια δήλωση

```
PRINT :<name>;
```

έξω από την PL/SQL δήλωση

4. Εδώ είναι ένα απλό παράδειγμα το οποίο τυπώνει την τιμή 1.

```
VARIABLE x NUMBER
BEGIN
  :x := 1;
END;
.
run;
PRINT :x;
```

ΚΕΦΑΛΑΙΟ 4

Τώρα ας περάσουμε να δούμε πως μπορούμε να δημιουργήσουμε μία Ελληνική βάση ORACLE.

1. ΑΠΑΙΤΗΣΕΙΣ

Μία ORACLE έχει κάποιες απαιτήσεις όπως και κάθε πρόγραμμα τις οποίες θα δούμε παρακάτω.

- **Oracle Server CD για τον SCO Openserver (Version 7.3.3.0.0.)**

Πρέπει να είναι μια νόμιμη κópια. Να θυμάστε πως η Oracle είναι μια κερδοσκοπική επιχείρηση και χρεώνει για τα προϊόντα της. Αν θέλετε μια δωρεάν SQL συμβατή βάση δεδομένων χρησιμοποιήστε την PostgreSQL ή κάποια παρόμοια.

Είναι επίσης δυνατόν να εγκαταστήσετε την oracle, χρησιμοποιώντας μια άδεια δοκιμαστικής χρήσης / αξιολόγησης 60 ημερών, από αρχείο που μπορείτε να κατεβάσετε από το tar web site της Oracle.

- **Ένα Linux Server**
- **Πυρήνα 2.0.30+**
- **iBCS**

- **Μεγάλο χώρο στο δίσκο**

600 Mb+ είναι ένα λογικό ποσό. Είναι δυνατό να εγκαταστήσετε και με λιγότερο αλλά θα χρειαστεί να έχετε αρκετό ελεύθερο χώρο. Παρόλα αυτά, θα προσπαθήσω να υποδείξω τα σημεία εκείνα στα οποία μπορεί να ελευθερωθεί χώρος.

- **32Mb+ Ram**

Είναι υπερβολικός ο χώρος που απαιτείται αλλά η ORACLE είναι ένα σύνθετο κομμάτι λογισμικού.

2.ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ ΤΗΣ ORACLE

2.1 Προετοιμασία του Server

Δημιουργία ενός χρήστη Oracle

Όπως είναι λογικό χρειαζόμαστε έναν χρήστη να κρατάει την βάση δεδομένων της Oracle. Μια που δεν επιθυμούμε να κάνουμε relink τον πυρήνα της Oracle (περισσότερα για αυτό αργότερα) θα πρέπει να αποδεχτούμε τις προκαθορισμένες παραμέτρους της Oracle για το όνομα του χρήστη και το όνομα του group. Αυτές συμπεριλαμβάνουν τον χρήστη ORACLE και το group DBA.

1. Συνδεθείτε ως Root
2. Δημιουργήστε το χρήστη oracle και το group dba.

```
$ groupadd dba
```

```
$ useradd oracle
```

3. Βεβαιωθείτε πως ένα home directory δημιουργείται για τον χρήστη oracle.
-

```
$ mkdir /home/oracle  
$ mkdir /home/oracle/7.3.3.0.0 (Version of Oracle)  
$ chown -R oracle.dba /home/oracle
```

2.2 Εγκατάσταση από CDROM

Δυστυχώς ο Oracle Installer από το δισκάκι του SCO δεν θα δουλέψει. Μπορεί να αντιμετωπίσετε πληθώρα προβλημάτων. Σαν αποτέλεσμα, χρειάζεται να αντιγράψουμε τα αρχεία από το CDROM χειρωνακτικά και να τα αποσυμπιέσουμε:

(Βεβαιωθείτε πως το CDROM είναι προσαρτημένο (mounted) στο σύστημα).

1. Συνδεθείτε σαν Oracle
 2. Αλλάξτε directory σε /home/oracle/7.3.3.0.0.
 3. Αντιγράψτε όλα τα αρχεία εγκατάστασης από το CDROM
-

```
$ cp -a /mnt/cdrom/*
```

4. Αποσυμπιέστε όλα τα αρχεία της Oracle από το CDROM.
-

```
$ find . -name *_ -exec ~/7.3.3.0.0/orainst/oiuncomp {} \;
```

2.3 Εργασίες μετά την Εγκατάσταση

Εργασίες για τον Root

Προσθέστε τις επόμενες γραμμές στο /etc/profile ή στο .profile για κάθε χρήστη που πρόκειται να χρησιμοποιήσει την Oracle.

```
# Oracle Specific
ORACLE_HOME=/home/oracle/7.3.3.0.0
ORACLE_SID=orcl
ORACLE_TERM=vt100
export ORACLE_HOME ORACLE_SID ORACLE_TERM

# Alter path for Oracle
PATH="$PATH:$ORACLE_HOME/bin"
```

Επίσης χρειάζεται να αλλάξουμε τον ιδιοκτήτη και τα προνόμια χρήσης της βοηθητικής εφαρμογής της Oracle ulimit increase.

```
$ chown root.root $ORACLE_HOME/bin/osh
$ chmod u+s $ORACLE_HOME/bin/osh
```

ΕΡΓΑΣΙΕΣ ΓΙΑ ΤΟΝ ORACLE

Αλλάξτε τα προνόμια χρήσης των αρχείων της Oracle για να εξασφαλίσετε την σωστή λειτουργία.

```
$ chmod +x $ORACLE_HOME/bin/*  
$ chmod u+s $ORACLE_HOME/bin/oracle
```

Τα εργαλεία της Oracle απαιτούν τα μηνύματα να κρατούνται στο directory \$ORACLE_HOME/tool_name/mesg. Έτσι, μετακινήστε τα αρχεία msb από τα directories msg_ship στα directories mesg .

```
$ mv $ORACLE_HOME/plsql/mesg/msg_ship/*  
$ORACLE_HOME/plsql/mesg/.  
$ mv $ORACLE_HOME/rdbms/mesg/msg_ship/*  
$ORACLE_HOME/rdbms/mesg/.  
$ mv $ORACLE_HOME/svrmgr/mesg/msg_ship/*  
$ORACLE_HOME/svrmgr/mesg/.
```

Δημιουργήστε τα ακόλουθα directories αν δεν υπάρχουν:

```
$ mkdir $ORACLE_HOME/rdbms/log  
$ mkdir $ORACLE_HOME/rdbms/audit  
$ mkdir $ORACLE_HOME/network/log
```

ΠΡΑΓΜΑΤΑ ΠΟΥ ΜΠΟΡΕΙΤΕ ΝΑ ΑΠΟΜΑΚΡΥ -ΝΕΤΕ :

Τα παρακάτω directories μπορούν να απομακρυνθούν με ασφάλεια:

- \$ORACLE_HOME/guicommon2/
- \$ORACLE_HOME/ctx/
- \$ORACLE_HOME/md/
- \$ORACLE_HOME/mlx/
- \$ORACLE_HOME/precomp/
- \$ORACLE_HOME/slax/

3. ΔΗΜΙΟΥΡΓΙΑ ΜΙΑΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Τώρα, ο Oracle server είναι εγκατεστημένος και χρειάζεται να δημιουργήσουμε μια βάση δεδομένων για να ελέγξουμε την εγκατάσταση.

3.1 Δημιουργία του Αρχείου Αρχικοποίησης

Αντιγράψτε το αρχείο \$ORACLE_HOME/dbs/init.ora στο \$ORACLE_HOME/dbs/initorcl.ora:

```
$ cd $ORACLE_HOME/dbs  
$ cp init.ora initorcl.ora
```

Τροποποιήστε το, προσθέτοντας τις ακόλουθες γραμμές:

```
db_name = orcl
COMPATIBLE=7.3.3.0.0
```

3.2 Δημιουργήστε το Script Εγκατάστασης Βάσης Δεδομένων

Δημιουργήστε ένα αρχείο script που να λέγεται makedb.sql στο \$ORACLE_HOME/dbs directory:

```
connect internal
startup nomount
set echo on
spool makedb.log
create database orcl
  maxinstances 1
  maxlogfiles 8
  datafile '$ORACLE_HOME/dbs/orcl_syst_01.dbf' size 40M
reuse
  logfile
    '$ORACLE_HOME/dbs/orcl_redo_01.dbf' size 1M reuse,
    '$ORACLE_HOME/dbs/orcl_redo_02.dbf' size 1M reuse,
    '$ORACLE_HOME/dbs/orcl_redo_03.dbf' size 1M reuse;
@$ORACLE_HOME/rdbms/admin/catalog.sql
create tablespace rollback
  datafile '$ORACLE_HOME/dbs/orcl_roll_01.dbf' size 8.5M
reuse;
create tablespace temporary
  datafile '$ORACLE_HOME/dbs/orcl_temp_01.dbf' size 5M reuse
temporary;
create tablespace users
  datafile '$ORACLE_HOME/dbs/orcl_user_01.dbf' size 10M
reuse;
create rollback segment r1 tablespace rollback
  storage ( optimal 5M );
alter rollback segment r1 online;
connect system/manager
@$ORACLE_HOME/rdbms/admin/catdbsyn.sql
connect internal
```

```
@$ORACLE_HOME/rdbms/admin/catproc.sql
connect system/manager
@$ORACLE_HOME/sqlplus/admin/pupbld.sql
spool off
exit
```

3.3 Εκτέλεση του Script Εγκατάστασης Βάσης Δεδομένων

Αρχίστε το svrmgrl και τρέξτε το script:

```
$ cd $ORACLE_HOME/dbs
$ svrmgrl
```

Oracle Server Manager Release 2.3.3.0.0 - Production

Copyright (c) Oracle Corporation 1994, 1995. All rights reserved.

Oracle7 Server Release 7.3.3.0.0 - Production Release
PL/SQL Release 2.3.3.0.0 - Production

```
SVRMGR> connect internal
Connected.
SVRMGR> startup nomount
ORACLE instance started.
Total System Global Area      4313312 bytes
Fixed Size                    41876 bytes
Variable Size                 4140364 bytes
Database Buffers              122880 bytes
Redo Buffers                   8192 bytes
SVRMGR> @makedb
<πολλά μηνύματα>
SVRMGR> exit
Server Manager complete.
```

3.4 Εκκίνηση της Βάσης Δεδομένων

Αρχικά, χρειάζεται να σηκώσουμε την βάση δεδομένων με το χέρι (θα το αυτοματοποιήσουμε στη συνέχεια). Για να αρχίσουμε μια βάση δεδομένων Oracle χρειάζεται να δώσουμε την εντολή startup αφού συνδεθούμε μαζί της εσωτερικά:

```
$ svrmgrl
```

```
Oracle Server Manager Release 2.3.3.0.0 - Production
```

```
Copyright (c) Oracle Corporation 1994, 1995. All rights reserved.
```

```
Oracle7 Server Release 7.3.3.0.0 - Production Release  
PL/SQL Release 2.3.3.0.0 - Production
```

```
SVRMGR> connect internal
```

```
Connected.
```

```
SVRMGR> startup
```

```
ORACLE instance started.
```

```
Total System Global Area      4313316 bytes
```

```
Fixed Size                      41876 bytes
```

```
Variable Size                  4140368 bytes
```

```
Database Buffers               122880 bytes
```

```
Redo Buffers                    8192 bytes
```

```
Database mounted.
```

```
Database opened.
```

```
SVRMGR> exit
```

```
Server Manager complete.
```

3.5 Σταματώντας την Βάση Δεδομένων

Αξίζει να αναφερθεί πως η επανεκκίνηση ενός Linux server χωρίς πρώτα το κατέβασμα (shutting down) της βάσης δεδομένων

της Oracle, έχει ένα μεγάλο κίνδυνο να διαβρώσει (corrupt) τη βάση δεδομένων.

Έτσι, πριν δώσουμε την εντολή του Linux shutdown, είναι συνετό να κατεβάσουμε τη βάση δεδομένων:

```
$ svrmgrl
```

```
Oracle Server Manager Release 2.3.3.0.0 - Production
```

```
Copyright (c) Oracle Corporation 1994, 1995. All rights reserved.
```

```
Oracle7 Server Release 7.3.3.0.0 - Production Release  
PL/SQL Release 2.3.3.0.0 - Production
```

```
SVRMGR> connect internal  
Connected.  
SVRMGR> shutdown  
Database closed.  
Database dismounted.  
ORACLE instance shut down.  
SVRMGR> exit  
Server Manager complete.
```

3.6 Δημιουργία ενός Προκαθορισμένου Χρήστη (Default User)

Η βάση δεδομένων, όπως δημιουργήθηκε, έχει δυο ειδικούς χρήστες που δημιουργούνται αυτόματα. Αυτοί είναι:

Username	Password
SYSTEM	MANAGER
SYS	change_on_install

Αυτοί οι χρήστες τυπικά χρησιμοποιούνται για να κρατάν τις πληροφορίες του data dictionary information για τη βάση δεδομένων. Είναι καλή ιδέα να αλλάξουμε τα passwords από τα προκαθορισμένα όσο πιο γρήγορα γίνεται.

Αυτό μπορεί να κατορθωθεί με:

```
sqlplus system/manager
```

```
SQL*Plus: Release 3.3.3.0.0 - Production on Sat Feb 21 12:43:33 1998
```

```
Copyright (c) Oracle Corporation 1979, 1996. All rights reserved.
```

```
Connected to:
```

```
Oracle7 Server Release 7.3.3.0.0 - Production Release
```

```
SQL> alter user system identified by <newpassword>;
```

```
User altered.
```

```
SQL> alter user sys identified by <newpassword>;
```

```
User altered.
```

```
SQL> exit;
```

```
Disconnected from Oracle7 Server Release 7.3.3.0.0 - Production Release
```

```
PL/SQL Release 2.3.3.0.0 - Production
```

Μια που ο χρήστης system/manager είναι παρόμοιος με το να χρησιμοποιείς root σε μια μηχανή UNIX, χρειάζεται να δημιουργήσουμε ένα χρήστη με λιγότερες δυνατότητες να βλάψει το σύστημα.

Συνδεθείτε στο SQL*Plus και δημιουργήστε ένα χρήστη:

```
$ sqlplus system/manager
```

```
SQL*Plus: Release 3.3.3.0.0 - Production on Sat Feb 21 12:43:33  
1998
```

```
Copyright (c) Oracle Corporation 1979, 1996. All rights reserved.
```

```
Connected to:
```

```
Oracle7 Server Release 7.3.3.0.0 - Production Release  
PL/SQL Release 2.3.3.0.0 - Production
```

```
SQL> create user <user> identified by <psw>  
2 default tablespace users  
3 temporary tablespace temp;
```

```
User created.
```

```
SQL> grant connect, resource to <user>
```

```
Grant succeeded.
```

```
SQL> exit
```

```
Disconnected from Oracle7 Server Release 7.3.3.0.0 - Production  
Release  
PL/SQL Release 2.3.3.0.0 - Production
```

Τώρα που έχετε ένα νέο χρήστη στο σύστημα, μπορείτε να παίξετε με το νέο σύστημα. Για να συνδεθείτε με την βάση δεδομένων της Oracle:

```
$ sqlplus <user>/<password>
```

Εάν αυτό ολοκληρώνεται χωρίς μηνύματα λάθους έχετε μια βάση Oracle που δουλεύει. Αν δεν θέλετε να συνδέεστε στη βάση αυτή από πουθενά πλην αυτού του Server, τότε η δουλειά τελείωσε.

Αν, παρόλα αυτά, θέλετε να ρυθμίσετε το λογισμικό δικτύωσης τότε θα κάνετε τα παρακάτω.

4. ΡΥΘΜΙΣΤΕ ΤΟ SQL*NET ΤΟΥ SERVER

Όλα αυτά τα αρχεία ρυθμίζουν το λογισμικό δικτύωσης της Oracle (SQL*Net, τώρα Net8 for Oracle8). Αυτά τα αρχεία θα πρέπει να έχουν ήδη δημιουργηθεί στο server στο \$ORACLE_HOME/network/admin directory.

4.1 tnsnames.ora

Το αρχείο TNSNAMES.ORA αναγνωρίζει τις διαθέσιμες υπηρεσίες της μηχανής. Στο στιγμιότυπο (instance) μας θα περιγράψουμε όλες τις βάσεις δεδομένων που ο server έχει προσκολλημένες. Για κάθε ένα στιγμιότυπο της Βάσης Δεδομένων (database instance) του server σας, προσθέστε ένα τμήμα σαν το παρακάτω:

```
orcl.world =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS =
      (COMMUNITY = tcp.world)
      (PROTOCOL = TCP)
      (Host = <INSERT HOST NAME OF SERVER HERE> )
      (Port = 1521)
    )
  )
)
```

```

    (ADDRESS =
      (COMMUNITY = tcp.world)
      (PROTOCOL = TCP)
      (Host = <INSERT HOST NAME OF SERVER HERE> )
      (Port = 1526)
    )
  )
(CONNECT_DATA = (SID = ORCL)
)
)

```

4.2 listener.ora

Το αρχείο listener.ora περιέχει τις περιγραφές των υπηρεσιών μας που άλλες μηχανές επιτρέπεται να συνδεθούν με αυτές, καθώς και οποιαδήποτε ρύθμιση που απαιτείται από τον listener του server.

Περιέχει τμήμα για το όνομα και τη διεύθυνση του listener, τις βάσεις δεδομένων που εξυπηρετούνται από τον listener, καθώς και παραμέτρους ρύθμισης.

Ας δούμε ένα παράδειγμα:

```

# Name of listener and addresses to listen on
LISTENER =
  ( ADDRESS_LIST =
    (ADDRESS =
      (PROTOCOL=tcp)
      (HOST=<INSERT HOST>)
      (PORT=1521)
      (COMMUNITY=UK_SUP_TCPIP)
    )
    (ADDRESS =
      (PROTOCOL=ipc)
      (KEY=700)
      (COMMUNITY=UK_SUP_TCPIP)
    )
  )

```

```

    )
)

# List of services served by this listener
SID_LIST_LISTENER=
  (SID_LIST=
    (SID_DESC=
      (SID_NAME=orcl)
      (ORACLE_HOME=/home/oracle/7.3.3.0.0)
    )
  )
)

# Start of configuration parameters.
TRACE_LEVEL_LISTENER=OFF
TRACE_FILE_LISTENER = "listener"
LOG_FILE_LISTENER = "listener"
CONNECT_TIMEOUT_LISTENER = 10
STOP_LISTENER = YES
DBA_GROUP = dba

```

4.3 sqlnet.ora

Το αρχείο sqlnet.ora περιέχει ρυθμίσεις για έναν συγκεκριμένο κόμβο του δικτύου. Αυτό είναι ανεξάρτητο του αριθμού των βάσεων ή του αριθμού των listeners. Το πιο σημαντικό πράγμα στο αρχείο αυτό είναι η μεταβλητή ρυθμίσεων Dead Connection Timeout.

Η Dead connection timeout ελέγχει κάθε εισερχόμενη διεργασία (process) προς ένα στιγμιότυπο μιας βάσης δεδομένων και βεβαιώνει πως το μέρος του πελάτη (client) ανταποκρίνεται ακόμα. Αν ο πελάτης (ανεξαρτήτως τύπου) δεν ανταποκρίνεται τότε η Oracle server shadow process σκοτώνεται (kill).

Αυτό είναι πολύ χρήσιμο Εάν έχετε πολλούς πελάτες που κάνουν πρόσβαση στη βάση δεδομένων, ειδικά σε φάση ανάπτυξης όπου αυτοί οι πελάτες είναι πιθανό να αποτυγχάνουν να βγουν ομαλά (exit cleanly) από την Β.Δ. της Oracle.

Παρακάτω είναι ένα αντίγραφο του αρχείου μου sqlnet.ora για να χρησιμοποιήσετε:

```
TRACE_LEVEL_CLIENT = OFF
sqlnet.expire_time = 30      # The number of seconds between
client checks.
names.default_domain = world
name.default_zone = world
```

4.4 Εκκίνηση και Τερματισμός των Listeners

Τώρα που οι ρυθμίσεις των listeners και του SQL*Net ολοκληρώθηκαν, μπορούμε να επιχειρήσουμε να συνδεθούμε στη βάση, χρησιμοποιώντας το λογισμικό δικτύωσης. (Πριν, συνδεόμασταν με απευθείας σύνδεση στη βάση, ενώ εδώ προσομοιώνουμε μια σύνδεση από μια απομακρυσμένη μηχανή πελάτη)

Για να αρχίσουμε τον listener με τις παραπάνω ρυθμίσεις:

```
$ lsnrctl
```

```
LSNRCTL for SCO System V/386: Version 2.3.3.0.0 - Production on
23-FEB-98 20:38:25
```

```
Copyright (c) Oracle Corporation 1994. All rights reserved.
```

```
Welcome to LSNRCTL, type "help" for information.
```

```
LSNRCTL> start
```

```
Starting /home/oracle/7.3.3.0.0/bin/tnslsnr: please wait...
```

```
TNSLSNR for SCO System V/386: Version 2.3.3.0.0 - Production
System          parameter          file          is
/home/oracle/7.3.3.0.0/network/admin/listener.ora
```

```
Log messages written to
/home/oracle/7.3.3.0.0/network/log/listener.log
Listening on:
(ADDRESS=(PROTOCOL=tcp)(DEV=6)(HOST=192.168.1.1)(PORT=
1521))
Listening on: (ADDRESS=(PROTOCOL=ipc)(DEV=10)(KEY=700))
```

```
Connecting to
(ADDRESS=(PROTOCOL=tcp)(HOST=magic.com)(PORT=1521)(CO
MMUNITY=UK_SUP_TCPIP))
STATUS of the LISTENER
```

```
-----
Alias LISTENER
Version TNSLSNR for SCO System V/386: Version
2.3.3.0.0 - Production
Start Date 23-FEB-98 20:38:50
Uptime 0 days 0 hr. 0 min. 0 sec
Trace Level off
Security OFF
SNMP ON
Listener Parameter File
/home/oracle/7.3.3.0.0/network/admin/listener.ora
Listener Log File
/home/oracle/7.3.3.0.0/network/log/listener.log
Services Summary...
 orcl has 1 service handler(s)
The command completed successfully
LSNRCTL> exit
```

Για να τερματίσουμε τους listeners:

```
$ lsnrctl
```

```
LSNRCTL for SCO System V/386: Version 2.3.3.0.0 - Production on
23-FEB-98 20:43:20
```

```
Copyright (c) Oracle Corporation 1994. All rights reserved.
```

```
Welcome to LSNRCTL, type "help" for information.
```

```
LSNRCTL> stop
```



```
Connecting to  
(ADDRESS=(PROTOCOL=tcp)(HOST=magic.com)(PORT=1521)(COMMUNITY=UK_SUP_TCPIP))  
The command completed successfully  
LSNRCTL> exit
```

5. ΡΥΘΜΙΣΕΙΣ ΠΕΛΑΤΗ (CLIENT)

5.1 Πελάτες Windows

Η ρύθμιση του SQL*Net σε PC χρησιμοποιώντας τις νέες εκδόσεις του λογισμικού Oracle Client είναι πολύ εύκολη. Ο καλύτερος και ευκολότερος τρόπος να επιτευχθεί μια πλήρως λειτουργική εγκατάσταση ενός πελάτη, είναι να χρησιμοποιηθεί το εργαλείο SQL*Net Easy Configuration που παρέχεται από την Oracle.

Το εργαλείο αυτό έχει ένα interface αυτοματοποιημένου προγράμματος (wizard) για να σας περάσει από την εγκατάσταση των αρχείων tnsnames.ora και sqlnet.ora.

Διαλέξτε "Add Database Alias" και βάλτε ένα όνομα για το ψευδώνυμο (alias) της βάσης όταν ερωτηθείτε when prompted. Αυτό το ψευδώνυμο είναι το όνομα με το οποίο θα αναφέρεστε στο στιγμιότυπο της βάσης, και σαν τέτοιο θα πρέπει να είναι ίδιο με το όνομα του στιγμιότυπου (orcl στην περίπτωση μας).

Δώστε το TCP/IP ως πρωτόκολλο, και όταν ερωτηθείτε, δώστε το hostname της μηχανής που φιλοξενεί τη Β.Δ. και το όνομα του στιγμιότυπου της βάσης δεδομένων.

Παρόλα αυτά, αν δεν έχετε το εργαλείο SQL*Net Easy Configuration μην ανησυχείτε. Μπορείτε απλά να δημιουργήσετε τα αρχεία `tnsnames.ora` και `sqlnet.ora files` στο `$ORACLE_HOME/network/admin` directory του πελάτη, ακριβώς όπως είναι στο server. Αυτό θα σας παρέχει ένα ψευδώνυμο ίδιο με το όνομα στον server.

5.2 Πελάτες Unix

Οι πελάτες UNIX δεν είναι πολύ διαφορετικοί από τους πελάτες windows. Αν έχετε το Network Manager της Oracle τότε χρησιμοποιήστε το με τον ίδιο τρόπο όπως παραπάνω. Αν όχι, πάλι μπορείτε να χρησιμοποιήσετε απλά τα ίδια αρχεία ρυθμίσεων του server στο `$ORACLE_HOME/network/admin` directory.

6. ΑΥΤΟΜΑΤΟ ΑΝΕΒΑΣΜΑ (STARTUP) ΚΑΙ ΚΑΤΕΒΑΣΜΑ (SHUTDOWN)

6.1 dbstart & dbstop

Το αυτόματο ανέβασμα (Startup) και κατέβασμα (Shutdown) της βάσης δεδομένων Oracle μπορεί να επιτευχθεί (στην 7.3.3.0.0) με τα αρχεία `dbstart` και `dbshut`, που και τα δύο παρέχονται από την Oracle. Αυτά τα αρχεία βασίζονται στην ύπαρξη του αρχείου `/etc/oratab` για να δουλέψουν (αν και αλλάζοντας τα αρχεία `dbshut` και `dbstart` αυτό μπορεί να μεταφερθεί).

To format του αρχείου `/etc/oratab` είναι ως ακολούθως:

SID:ORACLE_HOME:AUTO

Ας δούμε ένα παράδειγμα:

orcl:/home/oracle/7.3.3.0.0:Y
leaveup:/home/oracle/7.3.2.1.0:N

6.2 init.d & rc.d

Για να εκκινήσετε και σταματήσετε τη βάση δεδομένων όταν το μηχάνημα ανεβαίνει και κατεβαίνει μπορείτε να αλλάξετε τις ρουτίνες εκκίνησης της Linux μηχανής σας. Είναι αρκετά εύκολο, αν και θα ήθελα να σημειώσω εδώ, πως μπορεί να διαφέρει ανάλογα με την έκδοση Linux (slackware, debian, redhat, etc). Θα δείξω κάποια παραδείγματα που δουλεύουν για το Redhat Linux 5.0. Για να τα τροποποιήσετε ανάλογα, για τη δικιά σας έκδοση του Linux, διαβάστε την τεκμηρίωση του Linux σας. (Παρόλο που θα πρέπει να στέκει για κάθε UNIX τύπου Sys V).

Πρώτα, χρειάζεται να δημιουργήσουμε το script που θα τρέχει τις dbshut και dbstart στο directory /etc/rc.d/init.d. Δημιουργήστε το ακόλουθο αρχείο ως /etc/rc.d/init.d/oracle:

```
#!/bin/sh
#
# /etc/rc.d/init.d/oracle
# Description: Starts and stops the Oracle database and listeners
# See how we were called.
case "$1" in
  start)
    echo -n "Starting Oracle Databases: "
    echo "-----" >>
/var/log/oracle
    date +"! %T %a %D : Starting Oracle Databases as part of
system up." >> /var/log/oracle
```

```

    echo "-----" >>
/var/log/oracle
su - oracle -c dbstart >> /var/log/oracle
echo "Done."
echo -n "Starting Oracle Listeners: "
su - oracle -c "lsnrctl start" >> /var/log/oracle
echo "Done."
echo ""
echo "-----" >>
/var/log/oracle
date +"! %T %a %D : Finished." >> /var/log/oracle
echo "-----" >>
/var/log/oracle
touch /var/lock/subsys/oracle
;;
stop)
echo -n "Shutting Down Oracle Listeners: "
echo "-----" >>
/var/log/oracle
date +"! %T %a %D : Shutting Down Oracle Databases as
part of system down." >> /var/log/oracle
echo "-----" >>
/var/log/oracle
su - oracle -c "lsnrctl stop" >> /var/log/oracle
echo "Done."
rm -f /var/lock/subsys/oracle
echo -n "Shutting Down Oracle Databases: "
su - oracle -c dbshut >> /var/log/oracle
echo "Done."
echo ""
echo "-----" >>
/var/log/oracle
date +"! %T %a %D : Finished." >> /var/log/oracle
echo "-----" >>
/var/log/oracle
;;
restart)
echo -n "Restarting Oracle Databases: "
echo "-----" >>
/var/log/oracle
date +"! %T %a %D : Restarting Oracle Databases as part of
system up." >> /var/log/oracle
echo "-----" >>
/var/log/oracle
su - oracle -c dbstop >> /var/log/oracle
su - oracle -c dbstart >> /var/log/oracle
echo "Done."

```

```

echo -n "Restarting Oracle Listeners: "
su - oracle -c "lsnrctl stop" >> /var/log/oracle
su - oracle -c "lsnrctl start" >> /var/log/oracle
echo "Done."
echo ""
echo "-----" >>
/var/log/oracle
date +"! %T %a %D : Finished." >> /var/log/oracle
echo "-----" >>
/var/log/oracle
touch /var/lock/subsys/oracle
;;
*)
echo "Usage: oracle {start|stop|restart}"
exit 1
esac

```

Αξίζει να ελεγχθεί πως το αρχείο σταματάει και αρχίζει τη βάση πραγματικά σωστά, στο σύστημά σας. Κοιτάξτε το αρχείο μηνυμάτων (log file), /var/log/oracle για μηνύματα λάθους.

Αφού το script αυτό δουλεύει, χρειάζεται να δημιουργήσουμε συμβολικές ενώσεις (symbolic links) start και kill στα αντίστοιχα runlevel directories /etc/rc.d/rcX.d.

Οι επόμενες εντολές θα εξασφαλίσουν πως οι βάσεις δεδομένων θα ανέβουν στα runlevels 2,3 and 4:

```

$ ln -s ../init.d/oracle /etc/rc.d/rc2.d/S99oracle
$ ln -s ../init.d/oracle /etc/rc.d/rc3.d/S99oracle
$ ln -s ../init.d/oracle /etc/rc.d/rc4.d/S99oracle

```

Για να σταματήσετε τις βάσεις δεδομένων σε επανεκκίνηση θα χρειαστούμε τα ακόλουθα links:

```

$ ln -s ../init.d/oracle /etc/rc.d/rc0.d/K01oracle      # Halting
$ ln -s ../init.d/oracle /etc/rc.d/rc6.d/K01oracle      # Rebooting

```

ΒΙΒΛΙΟΓΡΑΦΙΑ

- ❖ ΜΑΘΗΜΑΤΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ, Δημήτρης Δερβός, εκδόσεις Α.ΤΖΙΟΛΑ Ε.
- ❖ ΣΥΣΤΗΜΑΤΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ, Εμμανουήλ Ι. Παννακουδάκης, εκδόσεις ΕΥΓ. ΜΠΕΝΟΥ, 1999
- ❖ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΟΡΓΑΝΩΣΕΙΣ ΑΡΧΕΙΩΝ, Χρήστος Κοίλιας, εκδόσεις ΝΕΩΝ ΤΕΧΝΟΛΟΓΙΩΝ, 1993
- ❖ ΣΗΜΕΙΩΣΕΙΣ ACCESS '97-ΣΥΝΟΠΤΙΚΟΣ ΟΔΗΓΟΣ ΒΑΣΙΚΩΝ ΛΕΙΤΟΥΡΓΙΩΝ, Δρ. Αλκιβιάδης Παναγόπουλος, ΑΤΕΙ ΠΑΤΡΑΣ-ΣΔΟ-ΤΜΗΜΑ ΤΟΥΡΙΣΜΟΥ
- ❖ ΣΗΜΕΙΩΣΕΙΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΗΥ, Επικ. Καθ. Μεόρχου, ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ-ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ
- ❖ File://A:/PL-SQL.HTM
- ❖ File://A:/Ελληνικό Oracle Database HOWTO.htm

ПАРАРТНМА

ΦΟΙΤΗΤΕΣ

ΕΠΙΘΕΤΟ

ΓΕΩΡΓΙΟΥ

ΟΝΟΜΑ	ΑΜ	ΠΑΤΡΩΝΥΜΟ	ΜΗΤΡΩΝΥΜΟ	ΗΜ_ΓΕΝΝΗΣΗ	ΔΙΕΥΘΥΝΣΗ	ΠΟΛΗ	ΤΗΛΕΦΩΝΟ	ΕΞΑΜΗΝΟ
ΚΑΤΕΡΙΝΑ	2233	ΚΩΝΣΤΑΝΤΙΝ	ΕΛΕΝΗ	15/8/1979	ΕΡΜΟΥ 6	ΠΑΤΡΑ	2610995100	ΠΤΥΧΙΟ Α

ΕΠΙΘΕΤΟ

ΜΗΧΟΥ

ΟΝΟΜΑ	ΑΜ	ΠΑΤΡΩΝΥΜΟ	ΜΗΤΡΩΝΥΜΟ	ΗΜ_ΓΕΝΝΗΣΗ	ΔΙΕΥΘΥΝΣΗ	ΠΟΛΗ	ΤΗΛΕΦΩΝΟ	ΕΞΑΜΗΝΟ
ΦΩΤΕΙΝΗ	5782	ΑΓΓΕΛΟΣ	ΚΩΝΣΤΑΝΤΙΝ	16/03/1982	ΝΑΥΜΑΧΙΑΣ	ΠΑΤΡΑ	2610424007	ΠΤΥΧΙΟ Γ

ΕΠΙΘΕΤΟ

ΜΙΧΑΗΛΙΔΗΣ

ΟΝΟΜΑ	ΑΜ	ΠΑΤΡΩΝΥΜΟ	ΜΗΤΡΩΝΥΜΟ	ΗΜ_ΓΕΝΝΗΣΗ	ΔΙΕΥΘΥΝΣΗ	ΠΟΛΗ	ΤΗΛΕΦΩΝΟ	ΕΞΑΜΗΝΟ
ΤΗΛΕΜΑΧΟΣ	6789	ΗΛΙΑΣ	ΒΑΣΙΛΙΚΗ	09/07/1981	ΘΕΜΙΣΤΟΚΛΕ	ΠΑΤΡΑ	6977098098	ΠΤΥΧΙΟ Β

ΕΠΙΘΕΤΟ

ΠΑΠΑΔΟΠΟΥΛΟΣ

ΟΝΟΜΑ	ΑΜ	ΠΑΤΡΩΝΥΜΟ	ΜΗΤΡΩΝΥΜΟ	ΗΜ_ΓΕΝΝΗΣΗ	ΔΙΕΥΘΥΝΣΗ	ΠΟΛΗ	ΤΗΛΕΦΩΝΟ	ΕΞΑΜΗΝΟ
ΝΙΚΟΣ	5712	ΠΑΥΛΟΣ	ΕΛΕΝΗ	12/3/1980	ΚΟΡΙΝΘΟΥ 80	ΠΑΤΡΑ	2610456345	7

ΕΠΙΘΕΤΟ

ΣΚΑΛΟΥΔΑΚΗ

ΟΝΟΜΑ	ΑΜ	ΠΑΤΡΩΝΥΜΟ	ΜΗΤΡΩΝΥΜΟ	ΗΜ_ΓΕΝΝΗΣΗ	ΔΙΕΥΘΥΝΣΗ	ΠΟΛΗ	ΤΗΛΕΦΩΝΟ	ΕΞΑΜΗΝΟ
ΧΡΥΣΑΝΘΗ	3380	ΑΝΔΡΕΑΣ	ΟΥΡΑΝΙΑ	03/12/1985	Β.ΗΠΕΙΡΟΥ 5	ΠΑΤΡΑ	2610333800	4

ΕΠΙΘΕΤΟ

ΤΡΙΓΚΟΣ

ΟΝΟΜΑ	ΑΜ	ΠΑΤΡΩΝΥΜΟ	ΜΗΤΡΩΝΥΜΟ	ΗΜ_ΓΕΝΝΗΣΗ	ΔΙΕΥΘΥΝΣΗ	ΠΟΛΗ	ΤΗΛΕΦΩΝΟ	ΕΞΑΜΗΝΟ
ΗΛΙΑΣ	5000	ΙΩΑΝΝΗΣ	ΕΥΔΟΚΙΑ	03/02/1978	ΘΕΡΜΟΠΥΛΩ	ΠΑΤΡΑ	6944565670	ΠΤΥΧΙΟ Β

ΒΑΘΜΟΙ-ΕΞΑΜΗΝΟ1

ΟΝΟΜΑ ΚΑΤΕΡΙΝΑ
ΕΠΙΘΕΤΟ ΓΕΩΡΓΙΟΥ
ΑΜ 2233

ΜΙΚΡ/ΝΟΜΙΚΗ	ΑΣΤΙΚΟ_ΔΙΚΑΙΟ	ΑΡΧ_ΔΙΟΙΚΗΣΗΣ	ΛΟΓΙΣΤΙΚΗ_1	ΚΒΣ	ΜΑΘΗΜΑΤΙΚΑ	M_01
5	6	8	7,3	8,1	10	7,4

ΒΑΘΜΟΙ-ΕΞΑΜΗΝΟ2

ΟΝΟΜΑ ΚΑΤΕΡΙΝΑ
ΕΠΙΘΕΤΟ ΓΕΩΡΓΙΟΥ
ΑΜ 2233

ΜΑΚΡ/ΝΟΜΙΚΗ	ΕΜΠ_ΔΙΚΑΙΟ	ΛΟΓΙΣΤΙΚΗ_2	ΨΙΚ_ΜΑΘΗΜΑΤΙΚΑ	ΕΛΣ ΑΥΤ_ΓΡΑΦΕΙΟΥ	Μ_Ο2	
5	6,5	5	8,2	7,5	9	6,866666666666667

ΒΑΘΜΟΙ-ΕΞΑΜΗΝΟ3

ΟΝΟΜΑ ΚΑΤΕΡΙΝΑ
ΕΠΙΘΕΤΟ ΓΕΩΡΓΙΟΥ
ΑΜ 2233

ΕΡΓ_ΔΙΚΑΙΟ	ΣΤΑΤΙΣΤΙΚΗ ΖΗΣΕΙΣ_ΔΕΟΜΕΝ	ΛΟΓ_ΕΤΑΙΡΕΙΩΝ ΜΗΧ_ΛΟΓΙΣΤΙΚΗ	ΟΙΚ_ΕΠΙΧ	M_03		
5	5	5,6	6,2	7	8	6,13333333333333

ΒΑΘΜΟΙ-ΕΞΑΜΗΝΟ4

ΟΝΟΜΑ ΚΑΤΕΡΙΝΑ
ΕΠΙΘΕΤΟ ΓΕΩΡΓΙΟΥ
ΑΜ 2233

MARKETING	ΛΠΣ	ΝΟΜΙΣΜΑΤ_Θ	Φ_ΕΜ_ΦΟΡΩΝ	ΥΙΕΘ_ΕΜΠΟΡΙΟ	ΑΝ_ΑΝΘ_ΠΟΡΩΝΔΙΚ_ΜΕΛΕΤΕΣ	M_04	
6,5	7	6	8	5	5	9	6,64285714285714

ΒΑΘΜΟΙ-ΕΞΑΜΗΝΟ5

ΟΝΟΜΑ ΚΑΤΕΡΙΝΑ
ΕΠΙΘΕΤΟ ΓΕΩΡΓΙΟΥ
ΑΜ 2233

ΧΡ_ΔΙΟΙΚΗΣΗΚΟΣΤΟΛΟΓΗΣΗ ΜΗΧ_ΛΟΓΙΣΤΙΚΗ1ΛΟΓ_ΕΦΑΡΜΟΓΕΣ1 ΜΕΤΑΤΡ_ΣΥΓΧ Φ_ΑΜ_ΦΟΡΩΝ							M_05
6	5	5,5	5	6	7,5	5,83333333333333	

ΒΑΘΜΟΙ-ΕΞΑΜΗΝΟΥ6

ΟΝΟΜΑ ΚΑΤΕΡΙΝΑ
ΕΠΙΘΕΤΟ ΓΕΩΡΓΙΟΥ
ΑΜ 2233

ΛΟΓ_ΕΦΑΡΜΟΓΕΣ2	ΙΝΑΛ_ΟΙΚ_ΚΑΤΜΗΧ_ΛΟΓΙΣΤΙΚΗ2	ΙΝΑΛΥΤΙΚΗ_ΛΟΓ_ΚΛΑΔΙΚΗ_ΛΟΓ	LOGISTICS	ΕΠΙΧ_ΕΡΕΥΝΑ	M_06		
8	5	7	5,5	5,5	7	6,5	6,35714285714286

ΒΑΘΜΟΙ-ΕΞΑΜΗΝΟ7

ΟΝΟΜΑ ΚΑΤΕΡΙΝΑ
ΕΠΙΘΕΤΟ ΓΕΩΡΓΙΟΥ
ΑΜ 2233

ΣΕΜΝΑΡΙΟ	ΟΡΟΛΟΓΙΑ	ΑΥΔΙΝΓ	ΜΗΧ_ΛΟΓ_ΔΙΑΧ_ΑΠΘΘ	ΔΙΟΙΚ_ΛΟΓΙΣ	M_07
	8,5	7,5	6,5	5,2	5
					6,54

ΒΑΘΜΟΙ-ΠΤΥΧΙΟ

ΟΝΟΜΑ ΚΑΤΕΡΙΝΑ
ΕΠΙΘΕΤΟ ΓΕΩΡΓΙΟΥ
ΑΜ 2233

ΠΤΥΧΙΑΚΗ	ΠΡΑΚΤΙΚΗ	Μ_Ο8	Μ_Ο_ΤΕΛΙΚΟΣ
8,5	9,5	9	6,846666666666667

ΤΕΙ ΠΑΤΡΑΣ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΛΟΓΙΣΤΙΚΗΣ

Πατρ.φορτίς : κ. Γ. Κοτσανοπούλου, τηλ:369218-20-21,369190

Αρ. Πρωτοκόλλου :

Ημερομηνία:

ΔΗΛΩΣΗ ΜΑΘΗΜΑΤΩΝ ΓΙΑ ΤΟ ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2003 - 2004

Επώνυμο: _____ Όνομα: _____ Τρέχον Εξάμ.: _____

Πατρώνυμο: _____ ΑΕΜ: _____ Ημερομηνία: _____

Διεύθυνση: _____ Τηλέφωνο: _____

Κωδ.	Τίτλος/Υπότιτλος	Τύπ.	Χαρ.	Κατ.	ΔΜ	Ωρες	Προαπ.	Θ	Ε
ΕΞΑΜΗΝΟ : Α									
11	ΜΙΚΡΟΟΙΚΟΝΟΜΙΚΗ	ΔΟ	Θ	ΚΟΡ	4	Θ2,Α1,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
16	ΑΡΧΕΣ ΑΣΤΙΚΟΥ ΔΙΚΑΙΟΥ	ΔΟ	Θ	ΚΟΡ	4	Θ2,Α1,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
12	ΑΡΧΕΣ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ	ΓΥ	Θ	ΚΟΡ	6	Θ2,Α4,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
132	ΓΕΝΙΚΗ ΛΟΓΙΣΤΙΚΗ Ι	ΓΥ	Θ	ΚΟΡ	7	Θ2,Α6,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
14	ΚΒΣ-ΟΡΓΑΝΩΣΗ ΛΟΓΙΣΤΗΡΙΟΥ	ΓΥ	Θ	ΚΟΡ	5	Θ2,Α2,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
15	ΓΕΝΙΚΑ ΜΑΘΗΜΑΤΙΚΑ	ΓΥ	Θ	ΚΟΡ	4	Θ2,Α2,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
ΕΞΑΜΗΝΟ : Β									
21	ΜΑΚΡΟΟΙΚΟΝΟΜΙΚΗ	ΔΟ	Θ	ΚΟΡ	4	Θ2,Α1,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
26	ΣΤΟΙΧΕΙΑ ΕΜΠΟΡΙΚΟΥ ΔΙΚΑΙΟΥ	ΔΟ	Θ	ΚΟΡ	5	Θ2,Α3,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
222	ΓΕΝΙΚΗ ΛΟΓΙΣΤΙΚΗ ΙΙ	ΓΥ	Θ	ΚΟΡ	6	Θ2,Α4,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
232	ΟΙΚΟΝΟΜΙΚΑ ΜΑΘΗΜΑΤΙΚΑ	ΓΥ	Θ	ΚΟΡ	5	Θ2,Α2,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
24	ΕΞΑΜΗΝΙΚΟ ΛΟΓΙΣΤΙΚΟ ΣΧΕΔΙΟ	ΓΥ	Θ	ΚΟΡ	5	Θ2,Α2,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
25	ΑΥΤΟΜΑΤΙΣΜΟΣ ΓΡΑΦΕΙΟΥ	ΓΥ	Μ	ΚΟΡ	5	Θ2,Α0,Ε3		<input type="checkbox"/>	<input type="checkbox"/>
ΕΞΑΜΗΝΟ : Γ									
32	ΑΡΧΕΣ ΕΡΓΑΤΙΚΟΥ ΔΙΚΑΙΟΥ	ΔΟ	Θ	ΚΟΡ	5	Θ3,Α0,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
342	ΣΤΑΤΙΣΤΙΚΗ ΕΠΙΧΕΙΡΗΣΕΩΝ	ΓΥ	Θ	ΚΟΡ	5	Θ2,Α3,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
35	ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ	ΓΥ	Μ	ΚΟΡ	5	Θ2,Α0,Ε2		<input type="checkbox"/>	<input type="checkbox"/>
31	ΛΟΓΙΣΤΙΚΗ ΕΤΑΙΡΕΙΩΝ	ΕΥ	Θ	ΚΟΡ	6	Θ2,Α4,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
33	ΜΗΧΑΝΟΓΡΑΦΗΜΕΝΗ ΛΟΓΙΣΤΙΚΗ Α & Β ΚΑΤΗΓΟΡΙΑΣ ΒΙΒΛΙΩΝ	ΕΥ	Ε	ΚΟΡ	3	Θ0,Α0,Ε5		<input type="checkbox"/>	<input type="checkbox"/>
36	ΟΙΚΟΝΟΜΙΚΗ ΕΠΙΧΕΙΡΗΣΕΩΝ	ΕΥ	Θ	ΚΟΡ	6	Θ3,Α1,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
ΕΞΑΜΗΝΟ : Δ									
41	ΑΡΧΕΣ ΜΑΡΚΕΤΙΝΓΚ	ΓΥ	Θ	ΚΟΡ	5	Θ3,Α1,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
42	ΛΟΓΙΣΤΙΚΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ - ΣΥΓΧΡΟΝΕΣ ΥΠΗΡΕΣΙΕΣ	ΕΥ	Μ	ΚΟΡ	5	Θ2,Α0,Ε3		<input type="checkbox"/>	<input type="checkbox"/>
43	ΝΟΜΙΣΜΑΤΙΚΗ ΘΕΩΡΙΑ & ΠΟΛΙΤΙΚΗ	ΕΥ	Θ	ΚΟΡ	6	Θ3,Α1,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
44	ΦΟΡΟΛΟΓΙΚΗ ΕΜΜΕΣΩΝ ΦΟΡΩΝ	ΕΥ	Θ	ΚΟΡ	8	Θ3,Α3,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
45	ΔΙΕΘΝΕΣ ΕΜΠΟΡΙΟ	ΜΕ	Θ	ΚΟΡ	5	Θ3,Α0,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
46	ΑΝΑΠΤΥΞΗ ΑΝΘΡΩΠΙΝΩΝ ΠΟΡΩΝ	ΜΕ	Θ	ΚΟΡ	5	Θ3,Α0,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
47	ΟΙΚΟΝΟΜΟΤΕΧΝΙΚΕΣ ΜΕΛΕΤΕΣ	ΜΕ	Θ	ΚΟΡ	5	Θ3,Α0,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
ΕΞΑΜΗΝΟ : Ε									
52	ΧΡΗΜΑΤΟΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ & ΠΟΛΙΤΙΚΗ	ΕΥ	Θ	ΚΟΡ	5	Θ2,Α2,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
51	ΒΙΟΜΗΧΑΝΙΚΟΣ ΛΟΓΙΣΜΟΣ - ΚΟΣΤΟΛΟΓΗΣΗ	Ε	Θ	ΚΟΡ	5	Θ2,Α2,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
53	ΜΗΧΑΝΟΓΡΑΦΗΜΕΝΗ ΛΟΓΙΣΤΙΚΗ Ι	Ε	Ε	ΚΟΡ	4	Θ0,Α0,Ε6		<input type="checkbox"/>	<input type="checkbox"/>
54	ΛΟΓΙΣΤΙΚΕΣ ΕΦΑΡΜΟΓΕΣ Ι	Ε	Ε	ΚΟΡ	2	Θ0,Α0,Ε4		<input type="checkbox"/>	<input type="checkbox"/>
55	ΜΕΤΑΤΡΟΠΕΣ - ΣΥΓΧΩΝΕΥΣΕΙΣ - ΕΝΟΠΟΙΗΜΕΝΟΙ ΙΣΟΛΟΓΙΣΜΟΙ	Ε	Θ	ΚΟΡ	5	Θ2,Α2,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
56	ΦΟΡΟΛΟΓΙΚΗ ΑΜΕΣΩΝ ΦΟΡΩΝ	Ε	Θ	ΚΟΡ	9	Θ4,Α3,Ε0		<input type="checkbox"/>	<input type="checkbox"/>

Κωδ.	Τίτλος/Υπότιτλος	Τύπ.	Χαρ.	Κατ.	ΔΜ	Ώρες	Προσπ.	Θ	Ε
ΕΞΑΜΗΝΟ : ΣΤ									
61	ΛΟΓΙΣΤΙΚΕΣ ΕΦΑΡΜΟΓΕΣ II	Ε	Ε	ΚΟΡ	2	Θ0,Α0,Ε4		<input type="checkbox"/>	<input type="checkbox"/>
62	ΑΝΑΛΥΣΗ ΟΙΚΟΝΟΜΙΚΩΝ ΚΑΤΑΣΤΑΣΕΩΝ	Ε	Θ	ΚΟΡ	7	Θ3,Α2,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
63	ΜΗΧΑΝΟΓΡΑΦΗΜΕΝΗ ΛΟΓΙΣΤΙΚΗ II	Ε	Ε	ΚΟΡ	4	Θ0,Α0,Ε6		<input type="checkbox"/>	<input type="checkbox"/>
64	ΑΝΑΛΥΤΙΚΗ ΛΟΓΙΣΤΙΚΗ	Ε	Θ	ΚΟΡ	7	Θ3,Α2,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
67	ΚΛΑΔΙΚΗ ΛΟΓΙΣΤΙΚΗ	Ε	Θ	ΚΟΡ	6	Θ2,Α4,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
65	ΕΦΟΔΙΑΣΤΙΚΗ ΔΙΑΧΕΙΡΙΣΗ (LOGISTICS)	ΜΕ	Θ	ΚΟΡ	4	Θ2,Α1,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
66	ΕΠΙΧΕΙΡΗΣΙΑΚΗ ΕΡΕΥΝΑ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ	ΜΕ	Θ	ΚΟΡ	4	Θ2,Α1,Ε0		<input type="checkbox"/>	<input type="checkbox"/>

ΕΞΑΜΗΝΟ : Ζ									
72	ΣΕΜΙΝΑΡΙΟ ΤΕΛΕΙΟΦΟΙΤΩΝ	ΕΥ	Θ	ΚΟΡ	5	Θ3,Α0,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
74	ΟΡΟΛΟΓΙΑ ΞΕΝΗΣ ΓΛΩΣΣΑΣ	ΕΥ	Θ	ΚΟΡ	6	Θ2,Α4,Ε0		<input type="checkbox"/>	<input type="checkbox"/>
71	ΕΣΩΤΕΡΙΚΟΣ & ΕΞΩΤΕΡΙΚΟΣ ΕΛΕΓΧΟΣ (AUDITING)	Ε	Μ	ΚΟΡ	8	Θ3,Α0,Ε3		<input type="checkbox"/>	<input type="checkbox"/>
73	ΜΗΧΑΝΟΓΡΑΦΗΜΕΝΗ ΛΟΓΙΣΤΙΚΗ ΔΙΑΧΕΙΡΙΣΗ ΑΠΟΘΗΚΗΣ	Ε	Ε	ΚΟΡ	4	Θ0,Α0,Ε6		<input type="checkbox"/>	<input type="checkbox"/>
75	ΔΙΟΙΚΗΤΙΚΗ ΛΟΓΙΣΤΙΚΗ - ΠΡΟΤΥΠΟ ΚΟΣΤΟΣ	Ε	Θ	ΚΟΡ	7	Θ3,Α3,Ε0		<input type="checkbox"/>	<input type="checkbox"/>

ΣΥΝΟΛΟ ΩΡΩΝ ΠΟΥ ΘΑ ΠΑΡΑΚΟΛΟΥΘΗΣΩ

Ο/Η σπουδαστής/δαστριά

ΠΡΟΑΙΡΕΤΙΚΑ ΜΑΘΗΜΑΤΑ

ΕΙΣΑΓΩΓΙΚΑ ΘΕΜΑΤΑ ΠΛΗΡΟΦΟΡΙΚΗΣ	Α' ΕΞΑΜΗΝΟ	Θ1,Α0,Ε1	<input type="checkbox"/>	<input type="checkbox"/>
ΕΙΣΑΓΩΓΗ ΣΤΑ ΔΙΕΘΝΗ ΛΟΓΙΣΤΙΚΑ ΠΡΟΤΥΠΑ	Γ' ΕΞΑΜΗΝΟ	Θ2,Α0,Ε0	<input type="checkbox"/>	<input type="checkbox"/>
ΟΙΚΟΝΟΜΙΚΗ ΔΗΜΟΓΡΑΦΙΑ	Δ' ΕΞΑΜΗΝΟ	Θ2,Α0,Ε0	<input type="checkbox"/>	<input type="checkbox"/>
ΚΟΙΝΩΝΙΚΗ ΛΟΓΙΣΤΙΚΗ	Ε' ΕΞΑΜΗΝΟ	Θ2,Α0,Ε0	<input type="checkbox"/>	<input type="checkbox"/>
ΕΝΑΛΛΑΚΤΙΚΗ ΟΙΚΟΝΟΜΙΑ	Ε' ΕΞΑΜΗΝΟ	Θ2,Α0,Ε0	<input type="checkbox"/>	<input type="checkbox"/>

Ο βαθμός των παραπάνω προαιρετικών μαθημάτων δεν υπολογίζεται για την εξαγωγή του βαθμού του πτυχίου.

ΠΙΝΑΚΑΣ ΠΡΟΑΠΑΙΤΟΥΜΕΝΩΝ ΚΑΙ ΕΞΑΡΤΩΜΕΝΩΝ ΜΑΘΗΜΑΤΩΝ

ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ

ΕΞΑΡΤΩΜΕΝΑ

Γενική Λογιστική I, Ε.Γ.Λ.Σ.....	Λογιστικές Εφαρμογές I
Λογιστική II.....	Λογιστική Εταιρειών
Λογιστική Εταιρειών.....	Λογιστικές Εφαρμογές II
Βιομηχανικός Λογισμός-Κοστολόγηση.....	Αναλυτική Λογιστική
Γενικά Μαθηματικά.....	Οικονομικά Μαθηματικά
Γενικά Μαθηματικά.....	Στατιστική Επιχ/σεων
Εισαγωγή στις Βάσεις Δεδομένων.....	Λογιστικά Πληροφοριακά
Αυτοματισμός Γραφείου.....	Συστήματα -Σύγχρονες Υπηρεσίες
Μακροοικονομική.....	Νομισματική Θεωρία και Πολιτική

ΟΔΗΓΙΕΣ

★ Πριν δηλώσετε τα μαθήματα πρέπει να λάβετε υπόψη σας τον πίνακα των προαπαιτούμενων. Δεν έχετε δικαίωμα να δηλώσετε εξαρτώμενο μάθημα αν δεν έχετε περάσει το αντίστοιχο προαπαιτούμενο. Όταν το προαπαιτούμενο είναι μικτό μάθημα πρέπει να έχετε παρακολουθήσει επιτυχώς και το θεωρητικό και το εργαστηριακό μέρος του.

★ Οι Σπουδαστές του Α' εξαμήνου υποχρεωτικά δηλώνουν τα μαθήματα του Α' εξαμήνου
 ★ Για τους σπουδαστές των υπολοίπων εξαμήνων το σύνολο των ωρών που θα δηλώνουν θα είναι το σύνολο των ωρών του τυπικού τους εξαμήνου προσαυξημένο κατά 33%.

℞ Κατά προτεραιότητα δηλώνονται:

Οι ώρες των μαθημάτων προηγούμενων εξαμήνων τα οποία δεν δηλώθηκαν είτε δεν ολοκληρώθηκαν με επιτυχία.

Οι ώρες του τυπικού εξαμήνου.

Οι ώρες του αμέσως επομένου εξαμήνου, με την προϋπόθεση ότι δε δηλώνονται μαθήματα, τα προαπαιτούμενα των οποίων δεν έχει παρακολουθήσει με επιτυχία ο σπουδαστής

★ Βαθμός που αποκτήθηκε χωρίς να δηλωθεί το μάθημα θεωρείται άκυρος.

★ Σημειώστε Υ στην αντίστοιχη στήλη με την ένδειξη Θ (Θεωρία) Ε (Εργαστήριο) και

ΠΡΟΣ

ΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΤΗΣ
ΣΧΟΛΗΣ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ

Α Π Τ Η Σ Η Ε Γ Γ Ρ Α Φ Η Σ

Α.

ΕΠΩΝΥΜΟ ΟΝΟΜΑ
ΟΝΟΜΑ ΠΑΤΕΡΑ ΟΝΟΜΑ ΜΗΤΕΡΑΣ
ΔΗΜΟΤΗΣ ΑΡΙΘΜΟΣ ΔΗΜΟΤΟΛΟΓΙΟΥ Ή ΑΡΙΘΜΟΣ Μ
ΤΡΟΥ ΑΡΡΕΝΩΝ ΗΜΕΡΟΜ. ΓΕΝΝΗΣΗΣ
ΥΠΗΚΟΟΤΗΤΑ
ΑΡΙΘΜΟΣ ΔΕΛΤΙΟΥ ΑΣΤΥΝΟΜΙΚΗΣ ΤΑΥΤΟΤΗΤΑΣ
ΗΜΕΡΟΜ. ΕΚΔΟΣΗΣ ΕΚΔΟΥΣΑ ΑΡΧΗ
ΑΡΙΘ. ΔΙΑΒΑΤΗΡΙΟΥ (ΓΙΑ ΑΛΛΟΔΑΠΟΥΣ)

Β.

ΣΧΟΛΕΙΟ ΜΕΣΗΣ ΕΚΠΑΙΔΕΥΣΗΣ
ΧΡΟΝΟΛΟΓΙΑ ΑΠΟΦΟΙΤΗΣΗΣ
ΤΡΟΠΟΣ ΕΙΣΑΓΩΓΗΣ ΣΤΑ Τ.Ε.Ι.
ΜΟΝΙΜΗ ΕΡΓΑΣΙΑ (ναί ή όχι και που)
ΑΣΦΑΛΙΣΜΕΝΟΣ (ναί ή όχι και που)

Γ.

ΔΙΕΥΘΥΝΣΗ ΚΑΤΟΙΚΙΑΣ ΣΠΟΥΔΑΣΤΗ: ΠΟΛΗ
ΟΔΟΣ - ΑΡΙΘΜΟΣ
ΤΑΧΥΔΡΟΜΙΚΟΣ ΚΩΔΙΚΑΣ ΤΗΛΕΦΩΝΟ
ΔΙΕΥΘΥΝΣΗ ΚΑΤΟΙΚΙΑΣ ΓΟΝΕΩΝ : ΠΟΛΗ
ΟΔΟΣ - ΑΡΙΘΜΟΣ
ΤΑΧΥΔΡΟΜΙΚΟΣ ΚΩΔΙΚΑΣ ΤΗΛΕΦΩΝΟ

Σας παρακαλώ να με γράψετε στο Α' χειμερινό/Α' εαρινό εξάμηνο τ
Τμήματος

Πάτρα

Υ Π Ε Υ Θ Υ Η Δ Ε Δ Ο Σ Η

Επώνυμο.....
Όνομα.....
του (όνομα πατέρα)
της (όνομα μητέρας).....
κείμενοςσός ερι
..... δηλώνω με την παρούσα με ατομική μου ευθύνη ότι δεν είμαι γρ
μην..... σε άλλη Σχολή ή τμήμα της Τριτοβάθμιας εκπαίδευσης της Ελλάδας ή σε
Εσωτερικού.

Πάτρα.....

Ο - Η δηλ.....

(Υπογραφή)

ΕΜΠΙΣΤΕΥΤΙΚΟ
Το περιεχόμενο του ερωτηματολογίου είναι εμπιστευτικό και θα χρησιμοποιηθεί αποκλειστικά για στατιστικούς σκοπούς. Η παροχή στοιχείων είναι υποχρεωτική (Ν.Δ. 3627/1956 και Ν. 2392/96).

ΑΤΟΜΙΚΟ ΣΤΑΤΙΣΤΙΚΟ ΔΕΛΤΙΟ ΦΟΙΤΗΤΗ-ΣΠΟΥΔΑΣΤΗ ΚΑΙ ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΦΟΙΤΗΤΗ

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2003/2004

_____|_____|_____|_____|
α/α δελτίου
(σμπληρώνεται από την ΕΣΥΕ)

_____|_____|_____|_____|_____|_____|_____|_____|
ΚΩΔΙΚΟΣ ΤΜΗΜΑΤΟΣ
(σμπληρώνεται από την ΕΣΥΕ)

Αρρεν 1 Θήλυ 2

1 | 9 | ____ | ____ |

_____|_____|

_____|_____| ή _____|_____|
(σμπληρώνεται από την ΕΣΥΕ)

_____|_____| ή _____|_____|
(σμπληρώνεται από την ΕΣΥΕ)

ΝΑΙ 1 ΟΧΙ 2

ΑΕΙ 1

ΑΤΕΙ 2

Επαγγελματική - Εκκλησιαστική Εκπαίδευση 3

Δημόσιο 1 Ιδιωτικό 2

_____|_____|

ΑΕΙ εσωτερικού 1

ΑΕΙ εξωτερικού 2

ΝΑΙ 1 ΟΧΙ 2

πατέρα μητέρα

01 01

02 02

03 03

04 04

05 05

06 06

07 07

08 08

1. Εκπαιδευτικό Ίδρυμα _____

2. Σχολή _____ 3. Τμήμα _____

4. Όνομα και επώνυμο του φοιτητή _____

5. Φύλο: _____
(σημειώστε Χ στο σχετικό τετραγωνίδιο)

6. Έτος γεννήσεως: _____

7. Υπηκοότητα: _____

8. Τόπος γεννήσεως: Νομός _____ ή Χώρα εξωτερικού _____

9. Τόπος στον οποίο ανήκει το Λύκειο από το οποίο αποφοιτήσατε:
Νομός _____ ή Χώρα εξωτερικού _____

10. Εγγράφεστε για πρώτη φορά στην Τριτοβάθμια (ΑΕΙ - ΑΤΕΙ, Επαγγελματική - Εκκλησιαστική Εκπαίδευση, κλπ.):
(σημειώστε Χ στο σχετικό τετραγωνίδιο)

11. Εάν όχι, είστε κάτοχος άλλου πτυχίου Τριτοβάθμιας:
(σημειώστε Χ στο σχετικό τετραγωνίδιο)

12. Αποφοιτήσατε από σχολείο Δευτεροβάθμιας εκπαίδευσεως:
(σημειώστε Χ στο σχετικό τετραγωνίδιο)

13. Έτος αποφοιτήσεως από το σχολείο Δευτεροβάθμιας εκπαίδευσεως: _____

14. Εφόσον είστε μεταπτυχιακοί φοιτητές κατέχετε πτυχίο:
(σημειώστε Χ στο σχετικό τετραγωνίδιο)

15. Εργάζεστε: _____
(σημειώστε Χ στο σχετικό τετραγωνίδιο)

16. Γραμματικές γνώσεις των γονιών σας, είτε είναι στη ζωή είτε όχι:
(για κάθε γονέα σημειώστε Χ, μόνο για τον ανώτερο τίτλο σπουδών, στο σχετικό τετραγωνίδιο)

α. Μεταπτυχιακό _____

β. Πτυχίο ΑΕΙ _____

γ. Πτυχίο ΚΑΤΕΕ-ΤΕΙ ή άλλης ανώτερης σχολής _____

δ. Απολυτήριο λυκείου ή εξατάξιου γυμνασίου ή ισότιμης τεχνικής επαγγελματικής σχολής _____

ε. Απολυτήριο γυμνασίου ή ισότιμης τεχνικής - επαγγελματικής σχολής _____

στ. Απολυτήριο δημοτικού _____

ζ. Δεν τελείωσε το δημοτικό _____

η. Αναλφάβητος _____

17. Ποια είναι η κύρια απασχόληση των γονιών σας (εάν είναι συνταξιούχοι, παρακαλούμε δώστε την κύρια απασχόληση πριν από τη συνταξιοδότηση):

Πατέρα _____

Μητέρα _____

Σημειώστε ένα μόνο από τους κατωτέρω κωδικούς (01, 02, ..., 12) στο διαβήριο χέρι του κωδικού επαγγέλματος του πατέρα και της μητέρας αντίστοιχα.

Πατέρα

Μητέρα

□ □

□ □

ΚΩΔΙΚΟΣ

01. Μέλη των Βουλευόμενων σωμάτων, ανώτερα διοικητικά και διευθυντικά στελέχη του δημόσιου και του ιδιωτικού τομέα. Πρόσωπα τα οποία νομοθετούν και καθορίζουν την κυβερνητική πολιτική σε κεντρικό, περιφερειακό και τοπικό επίπεδο, π.χ. υπουργοί, βουλευτές, γενικοί γραμματείς, δήμαρχοι, κοινοτάρχες κλπ. Διευθυντικά στελέχη μεγάλων επιχειρήσεων και οργανισμών, διευθυντές μικρών επιχειρήσεων.
02. Πρόσωπα που ασκούν επιστημονικά, καλλιτεχνικά και συναφή επαγγέλματα. Όσοι ασκούν επαγγέλματα για τα οποία απαιτείται ανώτατο επίπεδο εκπαίδευσης στους τομείς των επιστημών της διδασκαλίας και της καλλιτεχνικής δημιουργίας, π.χ. φυσικοί, μαθηματικοί, μηχανικοί, ιατροί, εκπαιδευτικοί όλων των βαθμίδων, νομικοί, λογιστές, οικονομολόγοι, κοινωνιολόγοι, αρχαιοθύτες, βιβλιοθηκονόμοι, συγγραφείς, δημοσιογράφοι, συνθέτες, μουσικοί, ηθοποιοί, κληρικοί, διοικητικά στελέχη δημοσίου τομέα μέχρι το βαθμό του Τμηματάρχη.
03. Τεχνολόγοι, τεχνικοί βοηθοί και ασκούντες συναφή επαγγέλματα. Τεχνολόγοι των επιστημών της φυσικής και της μηχανικής, τεχνολόγοι πληροφορικής, χειριστές οπτικού και ηλεκτρονικού εξοπλισμού, αξιωματικοί μηχανικού ναυσιπλοΐας, πιλότοι αεροσκαφών, ελεγκτές εναέριας κυκλοφορίας. Πρόσωπα στους τομείς των χρηματοπιστωτικών υπηρεσιών και των πωλήσεων, εμπορευματομεσίτες και πρώτοι παροχής υπηρεσιών σε επιχειρήσεις, αξιωματικοί σωμάτων ασφαλείας και ιδιωτικοί αστυνομικοί (ντετέκτιβ), οικονομικών υπηρεσιών και οργανισμών, επιθεωρητές ασφαλείας και ποιοτικού ελέγχου, διακοσμητές, εμπορικοί σχεδιαστές (μοντελίστ).
04. Υπάλληλοι γραφείου και ασκούντες συναφή επαγγέλματα. Υπάλληλοι γραφείου, γραμματείς, στενογράφοι και δακτυλογράφοι, υπάλληλοι λογιστηρίου, υπάλληλοι καταγραφής υλικών και υπηρεσιών μεταφορών, ταχυδρομικοί υπάλληλοι, υπάλληλοι βιβλιοθηκών κλπ.
05. Απασχολούμενοι στην παροχή υπηρεσιών και πωλητές σε καταστήματα και υπαίθριες αγορές. Ταξιδιωτικοί συνδοί, φροντιστές και ξεναγοί, εισπράτορες μεταφορικών μέσων, διαχειριστές, οικονομικοί, σερβιτόροι, μάγειροι, κομμωτές, κουρείς, αισθητικοί και ασκούντες συναφή επαγγέλματα, απασχολούμενοι στην παροχή προσωπικής φροντίδας και ασκούντες συναφή επαγγέλματα, λοιποί απασχολούμενοι στην παροχή προσωπικών υπηρεσιών - προσωπικοί υπηρέτες, εργολάβοι κηδεϊών.
06. Ειδιευμένοι γεωργοί, κτηνοτρόφοι, δασοκόμοι και αλιείς. Καλλιεργητές κτηνοτρόφοι, πτηνοτρόφοι, δασοκόμοι και υλοτόμοι, αλιείς ιχθυοτρόφοι, επαγγελματίες κυνηγοί.
07. Ειδιευμένοι τεχνίτες και ασκούντες συναφή επαγγέλματα. Μετελλωρχοί, κροτοί και ασκούντες συναφή επαγγέλματα, λιθοκόπτες, κτίστες, σκυροκονιαστές (μπετατζήδες) και τεχνίτες κατασκευής μωσαϊκών δαπέδων, ξυλουργοί και μαραγκοί, τεχνίτες ανέγερσης και συντήρησης κτιρίων, αμμοκονιαστές (σοβατζήδες), γυψοτεχνίτες, υδραυλικοί, ηλεκτρολόγοι εγκαταστάσεων, ελαιοχρωματιστές, βαφείς, στίλβωτές, καθαριστές και ασκούντες συναφή επαγγέλματα, άλλοι τεχνίτες αποπεράτωσης κτιρίων, σιδηρουργοί, κοσμηματοποιοί, τυπογράφοι, επιπλαποιοί, ράφτες, γοινοποιοί, γαζωτές, υποδηματοποιοί, αγγειοπλάστες, αεροποιοί, ζαχαροπλάστες, σφαγείς.
08. Χειριστές σταθερών βιομηχανικών εγκαταστάσεων, μηχανημάτων και εξοπλισμού και συναρμολόγησης (μοντάδόροι). Χειριστές εγκαταστάσεων ορυχείων και εγκαταστάσεων επεξεργασίας μεταλλευμάτων, χειριστές εγκαταστάσεων μεταλλουργίας, υαλοργίας, κεραμικής, επεξεργασίας ξύλου, χειριστές βιομηχανικών ρομπότ, οδηγό αυτοκινήτων οχημάτων, χειριστές μηχανών παραγωγής ειδών διατροφής, ποτών ή καπνού, κλωστοϋφαντουργικών και δερμάτινων προϊόντων, ναυτικοί.
09. Ανειδίκευτοι εργάτες, χειρωνακτες και μικροεπαγγελματίες. Πλανόδιοι πωλητές, οικιακοί βοηθοί, καθαριστές, επιστάτες - διαχειριστές κτιρίων, θυρωροί, αχθοφόροι, συλλέκτες απορριμμάτων, οδοκαθαριστές, ανειδίκευτοι αγροεργάτες, αλιεργάτες, ανειδίκευτοι εργάτες ορυχείων και κατασκευών, ανειδίκευτοι εργάτες μεταποιητικών βιομηχανιών, λιμενεργάτες, φορτοεκφορτωτές μεταφορικών μέσων.
10. Ένοπλες δυνάμεις.
11. Μη μισθωτοί, που δεν αναζητούν εργασία (π.χ. οικιακή απασχόληση).
12. Άνεργοι που αναζητούν εργασία.

Βεβαιώνεται ότι δόθηκαν πλήρεις και ακριβείς απαντήσεις σε όλα τα ερωτήματα

Ο φοιτητής

ΘΕΩΡΗΘΗΚΕ

για τη σωστή συμπλήρωσή του

Ο αρμόδιος για την εγγραφή

(Υπογραφή) _____

(Υπογραφή) _____

