

ΤΕΙ ΗΠΕΙΡΟΥ

Τμήμα : Τηλεπληροφορικής και Διοίκησης

ΚΟΥΤΡΗΣ ΘΕΟΔΩΡΟΣ Π.

“Σχεδίαση – Ανάπτυξη – Τεκμηρίωση
Συστήματος Μηχανοργάνωσης Εκπαιδευτικού Οργανισμού”

Επιβλέπων Καθηγητής
ΚΟΝΕΤΑΣ ΔΗΜΗΤΡΗΣ

Οκτώβριος 2003

**“Σχεδίαση – Ανάπτυξη – Τεκμηρίωση
Συστήματος Μηχανοργάνωσης Εκπαιδευτικού Οργανισμού”**

ΚΟΥΤΡΗΣ ΘΕΟΔΩΡΟΣ Π.

15/09/2003

Πτυχιακή Εργασία μέρος των απαιτήσεων
Του Τμήματος Τηλεπληροφορικής και Διοίκησης.

Αφιερωμένη στον ξάδερφο Κούτρη Θεόδωρο Θ.
και στον Χειμωνέτο Θεόδωρο Κ.

ΠΡΟΛΟΓΟΣ

Η εργασία απλώνεται σε τέσσερα μέρη. Όλα τα μέρη γράφτηκαν έτσι ώστε να μπορούν να διαβαστούν κάθε ένα ξεχωριστά, ανεξάρτητα από τα υπόλοιπα. Ωστόσο για να κατανοηθεί καλύτερα το αντικείμενο της εργασίας αυτής είναι προτιμότερο να διαβαστεί κεφάλαιο προς κεφάλαιο.

Στο πρώτο κεφάλαιο αναφέρονται βασικές έννοιες των βάσεων δεδομένων με έμφαση στο σχεσιακό μοντέλο. Στην ενότητα αυτή καλύπτονται έννοιες όπως Οντότητες, Σύστημα Διαχείρισης Βάσης Δεδομένων, Δομές Δεδομένων και Αρχιτεκτονική Συστήματος Βάσης Δεδομένων.

Στο δεύτερο κεφάλαιο περιγράφονται κάποια βασικά σημεία της εφαρμογής Microsoft Access για σχεσιακές βάσεις δεδομένων. Στη συνέχεια αναλύεται ο Σχεδιασμός μιας βάσης δεδομένων και συγκεκριμένα σχεδιάζεται η βάση δεδομένων για έναν Εκπαιδευτικό Οργανισμό Ξένων Γλωσσών. Το κεφάλαιο ολοκληρώνεται με την δημιουργία της βάσης αυτής, σε Microsoft Access.

Στο τρίτο κεφάλαιο παρουσιάζεται το πρόγραμμα διαχείρισης της βάσης δεδομένων του Εκπαιδευτικού Οργανισμού Ξένων Γλωσσών σε γλώσσα προγραμματισμού Visual Basic. Η ανάπτυξη του θέματος γίνεται με χρήση πολλών παραδειγμάτων. Όλες οι βασικές λειτουργίες του προγράμματος αναφέρονται σ' αυτό το κεφάλαιο όπως επίσης και επεξήγηση βασικών σημείων του κώδικα.

Το τέταρτο κεφάλαιο είναι ένα εγχειρίδιο χρήσης της εφαρμογής διαχείρισης της βάσης δεδομένων που προαναφέραμε. Εδώ αναφέρονται αναλυτικά παραδείγματα με εικόνες που βοηθούν το χρήστη να χρησιμοποιεί την εφαρμογή με τον καλύτερο δυνατό τρόπο, προς αποφυγή λαθών. Σκοπός του κεφαλαίου αυτού είναι να καλύψει κάθε απορία του χρήστη για την εφαρμογή.

Τέλος η εργασία συνοδεύεται από τον κώδικα του προγράμματος διαχείρισης της βάσης δεδομένων του Εκπαιδευτικού Οργανισμού Ξένων Γλωσσών.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Κεφάλαιο 1	ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ	
1.1	Τι είναι Βάση Δεδομένων.....	8
1.2	Οντότητες(Entities) – Ιδιότητες(Attributes).....	9
1.3	Σύστημα Βάσεων Δεδομένων.....	10
1.3.1	Σύστημα Διαχείρισης Βάσεων Δεδομένων.....	11
1.3.1.1	Δεδομένα.....	11
1.3.1.2	Υλικό.....	12
1.3.1.3	Λογισμικό.....	12
1.3.1.4	Χρήστες.....	13
1.4	Δομές Δεδομένων.....	14
1.4.1	Σχεσιακά Συστήματα.....	14
1.4.2	Πλεονεκτήματα Σχεσιακής Βάσης Δεδομένων.....	15
1.4.3	Η Γλώσσα SQL.....	16
1.5	Αρχιτεκτονική Συστήματος Βάσης Δεδομένων.....	17
1.6	Πλεονεκτήματα Χρήσης Βάσεων Δεδομένων.....	19
Κεφάλαιο 2	ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΟΡΓΑΝΙΣΜΟΥ ΞΕΝΩΝ ΓΛΩΣΣΩΝ	
2.1	Η Εφαρμογή Microsoft Access.....	22
2.1.1	Τα Στοιχεία μιας Βάσης Δεδομένων της Microsoft Access.....	22
2.1.2	Συσχετισμός Πινάκων με την Access.....	22
2.1.2.1	Τύποι Σχέσεων.....	23
2.1.2.2	Πεδία – Κλειδιά.....	23
2.2	Σχεδιασμός Βάσης Δεδομένων.....	24
2.2.1	Σχεδίαση Βάσης Δεδομένων Εκπαιδευτικού Οργανισμού.....	25
2.3	Δημιουργία Βάσης Δεδομένων Εκπαιδευτικού Οργανισμού.....	26
2.4	Σχέσεις των Πινάκων της Βάσης Δεδομένων Εκπ. Οργανισμού.....	28
Κεφάλαιο 3	ΠΡΟΓΡΑΜΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΟΡΓΑΝΙΣΜΟΥ ΞΕΝΩΝ ΓΛΩΣΣΩΝ	
3.1	Γλώσσα Προγραμματισμού Visual Basic.....	30
3.2	Δημιουργία Προγράμματος Διαχείρισης Βάσης Δεδομένων Εκπαιδευτικού Οργανισμού.....	30
3.2.1	Φόρμες Διαχείρισης της Βάσης Δεδομένων.....	32
3.2.1.1	Η Φόρμα MDIForm1.....	32
3.2.1.2	Η Φόρμα frmGeneralTblFunctions.....	33
3.2.1.3	Η Φόρμα frmTblInsert.....	35
3.2.1.4	Η Φόρμα frmGeneralTblFunctions1.....	39
3.2.1.5	Η Φόρμα form1.....	40
3.2.2	Πλαίσια Μηνυμάτων (MsgBox).....	41
3.2.3	Εκθέσεις – Αναφορές.....	50
3.2.3.1	Σχεδίαση με τον DataEnvironment.....	
	ActiveX Designer.....	50
3.2.3.2	Δημιουργία εκθέσεων με τον DataReport.....	
	ActiveX Designer.....	51

Κεφάλαιο 4	ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΔΙΑΧΕΙΡΙΣΗΣ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΕΝΟΣ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΟΡΓΑΝΙΣΜΟΥ ΞΕΝΩΝ ΓΛΩΣΣΩΝ (Project1)	
4.1	Εισαγωγή Δεδομένων στην Βάση.....	53
4.2	Αναζήτηση Εγγραφής της Βάσης Δεδομένων.....	56
4.3	Διόρθωση/Αλλαγή Δεδομένων.....	57
4.4	Διαγραφή Εγγραφής από την Βάση Δεδομένων.....	58
4.5	Εμφανίσεις/Εκτυπώσεις.....	58
4.6	Ειδικές Περιπτώσεις.....	59
Συμπεράσμα		61
Παράρτημα		62
Βιβλιογραφία		92

ΔΗΛΩΣΗ ΠΕΡΙ ΛΟΓΟΚΛΟΠΗΣ

Όλες οι προτάσεις οι οποίες παρουσιάζονται σ' αυτό το κείμενο και οι οποίες ανήκουν σε άλλους αναγνωρίζονται από τα εισαγωγικά και υπάρχει η σαφής δήλωση του συγγραφέα. Τα υπόλοιπα γραφόμενα είναι επινόηση του γράφοντος ο οποίος φέρει και καθολική ευθύνη γι' αυτό το κείμενο και δηλώνω υπεύθυνα ότι δεν υπάρχει λογοκλοπή γι' αυτό το κείμενο.

Όνοματεπώνυμο

Υπογραφή

Ημερομηνία.....

ΚΕΦΑΛΑΙΟ 1

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

1.1 ΤΙ ΕΙΝΑΙ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Οι ορισμοί που έχουν δοθεί για το τι είναι μια βάση δεδομένων διαφέρουν από μελετητή σε μελετητή. Παρακάτω παρουσιάζονται μερικοί από αυτούς τους ορισμούς.

ΟΡΙΣΜΟΣ (ENGLES)

Μια Βάση Δεδομένων ενός οργανισμού ή μιας επιχείρησης είναι μια συλλογή αποθηκευμένων λειτουργικών στοιχείων που χρησιμοποιούνται από τις μηχανογραφικές εφαρμογές του.

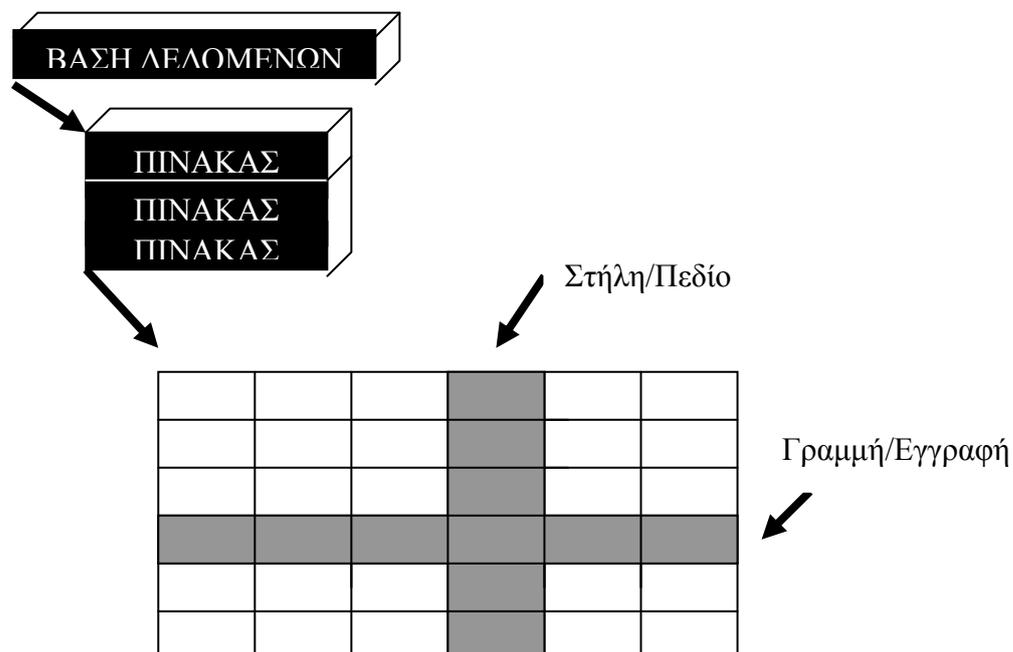
Στο περιγραφικό ορισμό που ακολουθεί γίνεται μια προσπάθεια να ορισθεί η Βάση Δεδομένων σαν αντίποδας της συνηθισμένης οργάνωσης όπου μια συλλογή εγγραφών αρχείων σχεδιάζεται για μια μόνο εφαρμογή.

ΟΡΙΣΜΟΣ (MARTIN)

Μια Βάση Δεδομένων είναι μια συλλογή αλληλοσυσχετιζόμενων δεδομένων που αποθηκεύονται μαζί χωρίς άχρηστους πλεονασμούς για την εξυπηρέτηση πολλών εφαρμογών. Η αποθήκευση των δεδομένων είναι τέτοια ώστε τα δεδομένα είναι ανεξάρτητα των προγραμμάτων που τα διαχειρίζονται. Η εισαγωγή νέων δεδομένων, η τροποποίηση και η ανάκτηση δεδομένων από την βάση δεδομένων ακολουθεί κοινή και ελεγχόμενη προσέγγιση για όλες τις εφαρμογές. Η δομή των δεδομένων παρέχει το θεμέλιο για ανάπτυξη μελλοντικής εφαρμογής. Δηλαδή, η οργάνωση των δεδομένων είναι τέτοια ώστε μπορούμε να υλοποιήσουμε νέες εφαρμογές χωρίς ιδιαίτερο κόπο και χωρίς να αλλάξει οτιδήποτε στις παλιές. Τέλος, λέμε ότι ένα σύστημα περιέχει μια συλλογή από Βάσεις Δεδομένων αν οι βάσεις αυτές είναι τελείως ξεχωριστές σε δομή.

Τι είναι τελικά βάση δεδομένων; Βασικά, μια βάση δεδομένων είναι μια ομαδοποίηση σχετικών πληροφοριών, που οργανώνονται για εύκολη επεξεργασία και ανάκτηση. Η ΕΙΚΟΝΑ 1.1 δείχνει πως δουλεύει. Τα πραγματικά δεδομένα μιας βάσης δεδομένων αποθηκεύονται σε πίνακες, που είναι παρόμοιοι με αρχεία τυχαίας προσπέλασης. Τα δεδομένα σε ένα πίνακα απαρτίζονται από στήλες και γραμμές. Οι γραμμές περιέχουν πανομοιότυπα δομημένα κομμάτια πληροφοριών, που είναι ισοδύναμα με τις εγγραφές των αρχείων τυχαίας προσπέλασης. Μια εγγραφή είναι μια συλλογή τιμών (που καλούνται πεδία).

ΕΙΚΟΝΑ1.1 : Αναπαράσταση Βάσης Δεδομένων και της δομής των πινάκων



1.2 ΟΝΤΟΤΗΤΕΣ (ENTITIES) – ΙΔΙΟΤΗΤΕΣ (ATTRIBUTES)*

Για την ανάπτυξη μιας βάσης δεδομένων πρέπει να προσεγγίσουμε διαδοχικά τρεις χώρους :

1) τον πραγματικό κόσμο, από όπου πρέπει να ανιχνευθούν :

- οι οντότητες ,

δηλαδή, πρόσωπα ,αντικείμενα, γεγονότα, πράξεις ή/και αφηρημένες έννοιες γύρω από τις οποίες θα συγκεντρωθεί και θα αποθηκευθεί πληροφορία,

- οι ιδιότητες τους (ή χαρακτηριστικά οντοτήτων),
- συλλογές ομοίων οντοτήτων,
- σχέσεις (ή συσχετίσεις) μεταξύ οντοτήτων και σχέσεις μεταξύ συλλογών όμοιων οντοτήτων.

Παράδειγμα : Οντότητα είναι ο μαθητής ΦΩΤΗΣ, η συλλογή στην οποία ανήκει είναι το σύνολο των μαθητών και οι ιδιότητες (ή τα χαρακτηριστικά) της οντότητας είναι :

Κωδικός, Όνομα, Επώνυμο, Ημ. Γεννήσεως, Διεύθυνση κ.τ.λ

* Η ανάπτυξη του κεφαλαίου αυτού έχει επηρεαστεί από τα βιβλία Σχεσιακές Βάσεις Δεδομένων, Χρήστου Π. Σκουρλά και Εισαγωγή στα Συστήματα Βάσεων Δεδομένων, C. J. Date.

- 2) τον κόσμο της πληροφορίας, όπως διαμορφώνεται μέσα από τις συγκεκριμένες παραδοχές που γίνονται από τον αναλυτή προγραμματιστή κυρίως. Στο στάδιο αυτό καθορίζεται ποιες ιδιότητες ενδιαφέρουν, τελικά, και γίνεται εκχώρηση συμβολικών ονομάτων στις ιδιότητες αυτές.

Παράδειγμα : Για την οντότητα ΒΙΒΛΙΟ καθορίζεται ότι ενδιαφέρουν οι ιδιότητες Κωδικός , Τίτλος, Συγγραφέας, Εκδόσεις.

Στις ιδιότητες αυτές δίνουμε συμβολικά ονόματα :
ISBN, ΟΝΟΜΑ, ΤΙΤΛΟΣ, ΣΥΓΓΡΑΦΕΑΣ, ΕΚΔΟΣΕΙΣ

- 3) τον κόσμο των δεδομένων (data), όπου ορθοί χαρακτήρων ή δυαδικών ψηφίων πρέπει να χρησιμοποιηθούν για την κωδικοποίηση της πληροφορίας.

Παράδειγμα : Για την οντότητα ΜΑΘΗΤΗΣ καθορίζεται ότι :

- ο κωδικός είναι μοναδικός ακέραιος αριθμός (η αρίθμηση είναι αυτόματη)
- κάθε μαθητής έχει ένα όνομα
- μπορεί να υπάρχουν συνωνυμίες
- το μήκος του ονόματος δεν ξεπερνά τους 50 χαρακτήρες

Ομοίως και για τα υπόλοιπα χαρακτηριστικά υπάρχουν αντίστοιχες ιδιότητες.

1.3 ΣΥΣΤΗΜΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Σύστημα βάσης δεδομένων (database system) ουσιαστικά είναι ένα σύστημα τήρησης εγγραφών σε υπολογιστή. Η ίδια η **βάση δεδομένων** μπορεί να θεωρηθεί ένα είδος ηλεκτρονικής αρχειοθέτησης με άλλα λόγια, ένας χώρος για την αποθήκευση μιας συλλογής ηλεκτρονικών αρχείων δεδομένων. Ο χρήστης του συστήματος έχει στη διάθεση του ορισμένα βοηθήματα, για να εκτελεί σε αυτά τα αρχεία διάφορες εργασίες, στις οποίες συγκαταλέγονται, ανάμεσα σε άλλες, και οι εξής :

- Η προσθήκη νέων κενών αρχείων στη βάση δεδομένων
- Η εισαγωγή νέων δεδομένων σε υπάρχοντα αρχεία
- Η ανάκληση δεδομένων από υπάρχοντα αρχεία
- Η ενημέρωση δεδομένων από υπάρχοντα αρχεία
- Η διαγραφή δεδομένων από υπάρχοντα αρχεία
- Η αφαίρεση υπάρχοντων αρχείων, κενών ή όχι , από την βάση δεδομένων

Οι πληροφορίες που τηρούνται σε ένα τέτοιο σύστημα μπορεί να είναι οτιδήποτε έχει σημασία για το άτομο ή τον οργανισμό που εξυπηρετεί το συγκεκριμένο σύστημα _ με άλλα λόγια οτιδήποτε χρειάζεται για την υποβοήθηση των εργασιών αυτού του ατόμου ή οργανισμού.

Οποιοδήποτε σύστημα Βάσεων Δεδομένων αποτελείται από 4 συνιστώσες :

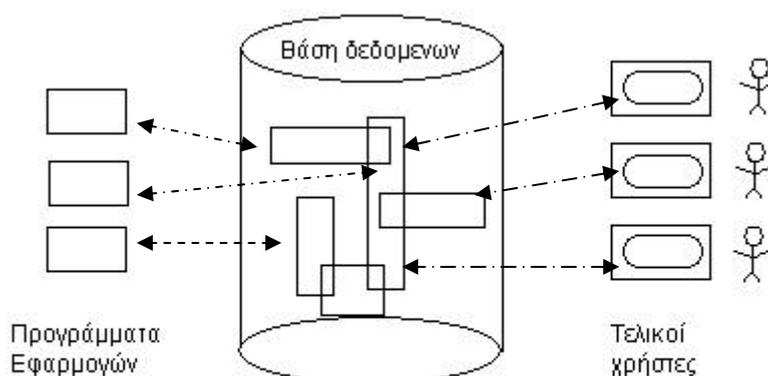
- Δεδομένα (Data)
- Υλικό (Hardware)
- Λογισμικό (Software)
- Χρήστες (Users)

1.3.1 ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων (Data Base Management System – DBMS) είναι ένα λογισμικό που επιτρέπει στο χρήστη (επιχείρηση) την υλοποίηση και τη συντήρηση Βάσεων Δεδομένων. Το Σύστημα Διαχείρισης Βάσεων Δεδομένων αναλαμβάνει τη διαχείριση των δεδομένων όλων των εφαρμογών και συνεργαζόμενο με το λειτουργικό σύστημα τα αποθηκεύει χωρίς άχρηστους πλεονασμούς. Η μεσολάβηση του Συστήματος Διαχείρισης Βάσεων Δεδομένων εξασφαλίζει ότι τα δεδομένα είναι ανεξάρτητα των προγραμμάτων που τα διαχειρίζονται.

Η ΕΙΚΟΝΑ1.2 παρουσιάζει μια άποψη ενός συστήματος βάσης δεδομένων, απλοποιημένη σε πολύ μεγάλο βαθμό. Σκοπός αυτής της εικόνας είναι να δείξει ότι ένα σύστημα βάσης δεδομένων απαρτίζεται από τα τέσσερα βασικά στοιχεία : τα δεδομένα, το υλικό(hardware), το λογισμικό (software) και τους χρήστες.

ΕΙΚΟΝΑ1.2 : Απλοποιημένη εικόνα ενός συστήματος βάσης δεδομένων



1.3.1.1 ΔΕΔΟΜΕΝΑ

Τα δεδομένα της βάσης δεδομένων – τουλάχιστον σε ένα μεγάλο σύστημα – είναι *ενοποιημένα* (integrated) και *μεριζόμενα* (shared – δηλαδή κοινόχρηστα). Αυτές οι δυο πλευρές είναι από τα μεγαλύτερα πλεονεκτήματα των συστημάτων βάσεων

δεδομένων σε ένα “μεγάλο” περιβάλλον. Ας εξηγήσουμε λοιπόν τι εννοούμε με τους όρους “ενοποίηση” και “μερισμός”.

Με τον όρο **ενοποίηση** δεδομένων (data integration) εννοούμε ότι η βάση δεδομένων μπορεί να θεωρείται μια συνένωση πολλών αρχείων δεδομένων, που από κάθε άλλη άποψη είναι ξεχωριστά το ένα από το άλλο, ενώ κάθε πλεονασμός εξαιτίας της επανάληψης δεδομένων μεταξύ αυτών των αρχείων έχει εξαλειφθεί εντελώς ή κατά ένα μέρος.

Με τον όρο **μερισμός** δεδομένων (data sharing) εννοούμε ότι τα μεμονωμένα στοιχεία δεδομένων της βάσης δεδομένων μπορούν να τα μοιράζονται πολλοί διαφορετικοί χρήστες, με την έννοια ότι καθένας από αυτούς τους χρήστες μπορεί να έχει πρόσβαση στο ίδιο στοιχείο δεδομένων (και οι διάφοροι χρήστες μπορούν να το χρησιμοποιούν για διαφορετικό σκοπό). Όπως αναφέραμε και νωρίτερα, οι διάφοροι χρήστες μπορούν να έχουν πρόσβαση στο ίδιο στοιχείο δεδομένων την ίδια στιγμή (“ταυτόχρονη προσπέλαση”). Αυτού του είδους ο μερισμός (ταυτόχρονος ή όχι) είναι σε ένα βαθμό συνέπεια του γεγονότος ότι η βάση δεδομένων είναι ενοποιημένη.

Μια άλλη συνέπεια του γεγονότος ότι η βάση δεδομένων είναι ενοποιημένη είναι ότι κάθε δεδομένος χρήστης ενδιαφέρεται κατά κανόνα μόνο για ένα πολύ μικρό τμήμα της συνολικής βάσης δεδομένων (και επιπλέον, τα τμήματα των διαφορετικών χρηστών θα επικαλύπτονται με πολλούς διάφορους τρόπους). Με άλλα λόγια, η αντίληψη που έχουν οι διαφορετικοί χρήστες για την ίδια βάση δεδομένων διαφέρει από πολλές απόψεις. Στην πραγματικότητα μάλιστα, ακόμη και όταν δύο χρήστες μοιράζονται το ίδιο τμήμα της βάσης δεδομένων, ο τρόπος που βλέπουν αυτό το τμήμα είναι πολύ πιθανό να διαφέρει σημαντικά σε ένα πιο λεπτομερές επίπεδο.

1.3.1.2 ΥΛΙΚΟ

Τα μέρη του συστήματος που έχουν να κάνουν με το υλικό (hardware) είναι τα εξής :

- Τα μέσα δευτερεύουσας αποθήκευσης – κατά κανόνα, μαγνητικοί δίσκοι με κινούμενες κεφαλές – που χρησιμοποιούνται για την αποθήκευση των δεδομένων, καθώς και οι αντίστοιχες συσκευές εισόδου/εξόδου, όπως οι μονάδες δίσκων (disk drives) κ.λ.π., οι ελεγκτές συσκευών (device controllers), τα κανάλια εισόδου/εξόδου, κ.ο.κ.
- Ο επεξεργαστής ή οι επεξεργαστές (processor) και η κύρια μνήμη (main memory), που χρησιμοποιούνται για την εκτέλεση του λογισμικού του συστήματος βάσης δεδομένων.

1.3.1.3 ΛΟΓΙΣΜΙΚΟ

Μεταξύ της ίδιας της φυσικής βάσης δεδομένων (δηλαδή των δεδομένων όπως είναι αποθηκευμένα στην πραγματικότητα) και των χρηστών του συστήματος υπάρχει ένα επίπεδο λογισμικού (software), ο διαχειριστής βάσεων δεδομένων

(database manager) ή όπως είναι ευρύτερα γνωστό, το σύστημα διαχείρισης βάσεων δεδομένων (database management system, DBMS). Το DBMS διαχειρίζεται όλες τις αιτήσεις των χρηστών για προσπέλαση της βάσης δεδομένων. Οι λειτουργίες για την προσθήκη και την αφαίρεση αρχείων (ή πινάκων), την ανάκληση και την ενημέρωση δεδομένων που είναι αποθηκευμένα σε τέτοια αρχεία ή πίνακες κ.λ.π., είναι όλες υπηρεσίες που παρέχονται από το DBMS. Έτσι, μπορεί να πει κανείς ότι μια γενικότερη υπηρεσία που παρέχεται από το DBMS είναι η απομόνωση των χρηστών της βάσης δεδομένων από τις λεπτομέρειες που αφορούν το υλικό (με τον ίδιο τρόπο που τα συστήματα γλωσσών προγραμματισμού απομονώνουν τους προγραμματιστές εφαρμογών από τις λεπτομέρειες που αφορούν το υλικό). Με άλλα λόγια, το DBMS παρέχει στους χρήστες μια άποψη της βάσης δεδομένων “ανυψωμένη” κατά κάποιον τρόπο πάνω από το επίπεδο του υλικού, και υποστηρίζει τις πράξεις των χρηστών, οι οποίες εκφράζονται με βάση αυτή την άποψη ανωτέρου επιπέδου.

Είναι σαφές ότι το DBMS είναι το σημαντικότερο στοιχείο λογισμικού ολόκληρου του συστήματος, αλλά δεν είναι το μόνο. Άλλα τέτοια στοιχεία είναι τα βοηθητικά προγράμματα, τα εργαλεία σύνταξης αναφορών κ.ο.κ.

1.3.1.4 ΧΡΗΣΤΕΣ

Διακρίνουμε τρεις γενικές κατηγορίες χρηστών :

- ♦ Στην πρώτη κατηγορία ανήκουν οι **προγραμματιστές εφαρμογών** (application programmers), οι οποίοι είναι υπεύθυνοι για το γράψιμο προγραμμάτων εφαρμογών που χρησιμοποιούν τη βάση δεδομένων. Αυτά τα προγράμματα μπορεί να είναι συμβατικές εφαρμογές ομαδικής επεξεργασίας (batch application) ή εφαρμογές άμεσης επεξεργασίας (online application), και στόχος τους είναι να υποστηρίζουν τον τελικό χρήστη που προσπελάζει τη βάση δεδομένων από κάποιο σταθμό εργασίας ή τερματικό.
- ♦ Στη δεύτερη κατηγορία χρηστών ανήκουν οι **τελικοί χρήστες**, οι οποίοι αλληλεπιδρούν με το σύστημα μέσω συνδεδεμένων σταθμών εργασίας ή τερματικών. Ένας τελικός χρήστης μπορεί να προσπελάζει τη βάση δεδομένων μέσω κάποιας από τις εφαρμογές άμεσης επεξεργασίας που αναφέραμε στην προηγούμενη παράγραφο, ή να χρησιμοποιεί κάποια διασύνδεση (interface) που είναι οργανικό μέρος λογισμικού του συστήματος βάσης δεδομένων.

Οι περισσότερες εφαρμογές παρέχουν στους χρήστες τους πρόσθετες ενσωματωμένες διασυνδέσεις, όπου οι χρήστες δε δίνουν ρητές διαταγές, αλλά δουλεύουν π.χ. επιλέγοντας στοιχεία από ένα μενού ή συμπληρώνοντας πλαίσια σε μια φόρμα. Αυτού του είδους οι διασυνδέσεις που οδηγούνται από μενού ή από φόρμες (menu-driven or forms-driven interface) τείνουν να είναι πολύ πιο εύχρηστες για τους χρήστες που δεν έχουν τυπική εκπαίδευση στην τεχνολογία πληροφοριών. Αντίθετα οι διασυνδέσεις που οδηγούνται από διαταγές (command-driven interfaces) – όπως οι γλώσσες ερωτημάτων – απαιτούν ένα ορισμένο επίπεδο επαγγελματικής κατάρτισης στην τεχνολογία πληροφοριών, αν και όχι πολύ υψηλό. Από την άλλη όμως, μια διασύνδεση οδηγούμενη από διαταγές είναι μάλλον πιο ευέλικτη από μια διασύνδεση οδηγούμενη από μενού ή φόρμες, επειδή οι γλώσσες ερωτημάτων

διαθέτουν συνήθως και ορισμένες λειτουργίες που δεν υποστηρίζονται από τις άλλες διασυνδέσεις.

- ♦ Στην τρίτη κατηγορία χρηστών ανήκει ο **υπεύθυνος διαχείρισης βάσεων δεδομένων** (database administrator, DBA). Δουλειά του είναι να αποφασίζει εξαρχής ποια δεδομένα θα πρέπει να αποθηκευτούν στη βάση δεδομένων και να ορίσει την πολιτική και την τήρηση και τον χειρισμό αυτών των δεδομένων από την στιγμή που θα αποθηκευτούν.

1.4 ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Η σχεδίαση ενός συστήματος βάσεων δεδομένων συνδέεται άμεσα με την απάντηση στο εξής ερώτημα : “Τι δομές δεδομένων και ποιες λειτουργίες συνδεόμενες με αυτές θα μπορεί να υποστηρίξει το σύστημα;”. Απάντηση δόθηκε σε διάφορες κατευθύνσεις (δομές δεδομένων) :

- ♦ Σχεσιακή προσέγγιση – Relational approach όπου για την περιγραφή του συστήματος και των λειτουργιών του γίνεται χρήση κανονικοποιημένων δομών. Οι δομές αυτές είναι σχέσεις (ή πίνακες).
- ♦ Ιεραρχική προσέγγιση – Hierarchical approach όπου για την περιγραφή του συστήματος και των λειτουργιών του γίνεται χρήση δομών δέντρου.
- ♦ Δικτυωτή προσέγγιση – Network approach όπου για την περιγραφή του συστήματος και των λειτουργιών του γίνεται χρήση δικτυωτών δομών.

Σύμφωνα με τον Codd κανένα από τα προτεινόμενα μοντέλα δεν πρόκειται να αντικαταστήσει το σχεσιακό.

1.4.1 ΣΧΕΣΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

Όλα σχεδόν τα προϊόντα διαχείρισης βάσεων δεδομένων που αναπτύχθηκαν από τα τέλη της δεκαετίας του 1970 και μετά βασίζονται στην λεγόμενη **σχεσιακή προσέγγιση** (relational approach). Είναι γεγονός ότι η σχεσιακή προσέγγιση αποτελεί σήμερα την κυρίαρχη τάση στην αγορά, και ότι το “σχεσιακό μοντέλο” είναι το σημαντικότερο επίτευγμα σε ολόκληρη την ιστορία των βάσεων δεδομένων και έχει στερεές μαθηματικές βάσεις.

Με δύο λόγια, λοιπόν, σχεσιακό είναι ένα σύστημα που :

1. Ο χρήστης αντιλαμβάνεται τα δεδομένα ως πίνακες
2. Οι τελεστές πράξεων που έχει στη διάθεση του ο χρήστης (για παράδειγμα, για την ανάκληση δεδομένων) είναι τελεστές που δημιουργούν νέους πίνακες με βάση κάποιους παλαιότερους.

Ο λόγος για τον οποίο ονομάζονται αυτά τα συστήματα “σχεσιακά” είναι ότι ο όρος “σχέση” είναι ουσιαστικά ένας όρος των μαθηματικών που σημαίνει

“πίνακας”. Πρακτικά μάλιστα, στις περισσότερες οι όροι “σχέση” και “πίνακας” μπορούν να θεωρηθούν συνώνυμοι.

1.4.2 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΣΧΕΣΙΑΚΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Για να κατανοήσετε τα πλεονεκτήματα μιας σχεσιακής βάσης δεδομένων, θα πρέπει κατ’ αρχήν να γνωρίσετε τι είναι μη-σχεσιακές βάσεις δεδομένων οι οποίες αποκαλούνται συνήθως *βάσεις δεδομένων επίπεδου αρχείου*. Όταν δουλεύετε με μια βάση δεδομένων επίπεδου αρχείου, μπορείτε να χρησιμοποιείτε μόνο έναν πίνακα δεδομένων ανά πάσα στιγμή. Για παράδειγμα, εάν θέλατε να εισάγετε τις Εγγραφές Μαθημάτων στον υπολογιστή σας χρησιμοποιώντας μια βάση δεδομένων επίπεδου αρχείου, θα έπρεπε να δημιουργήσετε έναν πίνακα με πεδία για κάθε πιθανό στοιχείο πληροφορίας που μπορεί να περιλαμβάνει μια Εγγραφή. Ουσιαστικά, θα έπρεπε να συμπεριλάβετε τις λεπτομέρειες της Εγγραφής, ο κωδικός και τα στοιχεία του μαθητή, τους κωδικούς των τμημάτων και της ημερομηνίας έναρξης μαθημάτων και εγγραφής. Η δομή ενός τέτοιου πίνακα θα μπορούσε να δείχνει όπως η λίστα των πεδίων του παρακάτω ΠΙΝΑΚΑ 1.3

ΠΙΝΑΚΑΣ 1.3 : Τα πεδία για μια βάση δεδομένων Εγγραφών Μαθητή με δομή επίπεδου αρχείου

Όνομα πεδίου

Κωδικός Εγγραφής
Κωδικός Μαθητή
Όνομα
Επώνυμο
Όνομα Πατρός
Όνομα Μητρός
Ημερομηνία Γεννήσεως
Διεύθυνση Κατοικίας
Πόλη
Ταχ. Κώδικας
Τηλέφωνο(Σταθερό)
Κινητό
Κωδικός Τμήματος
Τμήμα
Γλώσσα
Αίθουσα
Ημερομηνία Εγγραφής
Ημερομηνία Έναρξης

Τα μειονεκτήματα της αποθήκευσης όλων αυτών των δεδομένων σε έναν πίνακα ή αρχείο είναι προφανή. Κατά κύριο λόγο, περιορίζετε στην εισαγωγή ενός σταθερού, μέγιστου αριθμού λεπτομερών στοιχείων εγγραφής για κάθε εγγραφή. Ένα άλλο μειονέκτημα είναι το γεγονός ότι πρέπει να κουβαλάτε όλες αυτές τις

πληροφορίες μαζί, ακόμα και αν θέλετε να δημιουργήσετε επιστολές και εκθέσεις για τους μαθητές, θα έπρεπε είτε να χρησιμοποιήσετε το τεράστιων διαστάσεων αρχείο με τα δεδομένα της εγγραφής και να εξαιρέσετε τα διπλότυπα δεδομένα, στις περιπτώσεις που ένας μαθητής γραφτεί σε περισσότερα από ένα τμήματα.

Εν αντιθέσει, με μια σχεσιακή βάση δεδομένων, μπορούμε να αποθηκεύσουμε τα στοιχεία προσδιορισμού των Εγγραφών (π.χ. αίθουσα, ημερομηνίες εγγραφής και έναρξης μαθημάτων) σε ένα πίνακα και τα υπόλοιπα στοιχεία σε άλλους πίνακες, όπως φαίνετε στον παρακάτω πίνακα

ΠΙΝΑΚΑΣ1.4 : Τα πεδία και οι πίνακες για την διαχείριση των Εγγραφών σε μια σχεσιακή βάση δεδομένων

Εγγραφές	Στοιχεία Μαθητή	Τμήματα
Κωδικός Εγγραφής	Κωδικός Μαθητή	Κωδικός Τμήματος
Κωδικός Μαθητή	Όνομα	Τμήμα
Αίθουσα	Επώνυμο	Γλώσσα
Κωδικός Τμήματος	Όνομα Πατρός	
Ημερομ. Εγγραφής	Όνομα Μητρός	
Ημερομ. Έναρξης	Ημερομ. Γενήσεως	
	Διεύθυνση Κατοικίας	
	Πόλη	
	Ταχ. Κώδικας	
	Τηλέφωνο	
	Κινητό	

Τα πλεονεκτήματα της εργασίας με πολλαπλούς σχετιζόμενους πίνακες δεδομένων όπως αυτοί στον παραπάνω πίνακα είναι:

1. Είμαστε υποχρεωμένοι να εισάγουμε πληροφορίες όπως τα στοιχεία των μαθητών και των τμημάτων μόνο μια φορά, σε μια θέση, και όχι σε κάθε εγγραφή που κάνει ο μαθητής .
2. Κρατώντας όλα αυτά τα δεδομένα(τις λεπτομέρειες των εγγραφών, τα στοιχεία των μαθητών και των τμημάτων) σε ξεχωριστούς πίνακες είναι ευκολότερο να σχεδιάσουμε φόρμες και εκθέσεις για πολλούς διαφορετικούς σκοπούς.

Το μοντέλο σχεσιακής βάσης δεδομένων δημιουργήθηκε το 1970 από το Dr. E. F. Codd της IBM, ο οποίος δημιούργησε επίσης την γλώσσα ερωτημάτων που έγινε η SQL.

1.4.3 Η γλώσσα SQL

Τα περισσότερα σημερινά σχεσιακά προϊόντα υποστηρίζουν κάποια διάλεκτο της καθιερωμένης σχεσιακής γλώσσας, της SQL. Η SQL αναπτύχθηκε αρχικά στο τμήμα έρευνας της IBM, στις αρχές της δεκαετίας του 1970 υλοποιήθηκε για πρώτη φορά σε μεγάλη κλίμακα στο σχεσιακό πρωτότυπο System R της IBM, και στη συνέχεια

υλοποιήθηκε ξανά σε πολυάριθμα προϊόντα του εμπορίου, τόσο από την IBM όσο και από άλλες εταιρίες. Από τις διάφορες διαλέκτους της SQL έχουν προκύψει από τότε ένα αμερικανικό εθνικό πρότυπο (ANSI), ένα διεθνές πρότυπο (ISO), ένα πρότυπο UNIX (X/Open), ένα πρότυπο της IBM (αποτελεί το τμήμα "κοινής διασύνδεσης βάσεων δεδομένων" της αρχιτεκτονικής System Applications Architecture – SAA – της IBM) και ένα αμερικανικό ομοσπονδιακό πρότυπο επεξεργασίας πληροφοριών (Federal Information Processing Standard, FIPS).

Η SQL χρησιμοποιείται για την διατύπωση σχεσιακών πράξεων (δηλαδή, πράξεων που ορίζουν και χειρίζονται δεδομένα σε σχεσιακή μορφή).

Η γλώσσα χειρισμού Βάσεων Δεδομένων SQL – Structured Query Language περιλαμβάνει τρεις υπογλώσσες :

- Την υπογλώσσα *ορισμού* δεδομένων – DDL (Data Definition Language)
- Την υπογλώσσα *χειρισμού* δεδομένων - DML (Data Manipulation Language)
- Την υπογλώσσα *ελέγχου* δεδομένων – DCL (Data Control Language)

Η DDL είναι γλώσσα που δίνει στο χρήστη τη δυνατότητα να ορίσει πίνακες, δείκτες και όψεις και στην συνέχεια να τροποποιήσει ή να διαγράψει αυτούς τους ορισμούς.

Η DML είναι μια γλώσσα που δίνει στο χρήστη τη δυνατότητα εισαγωγής, τροποποίησης, διαγραφής και αναζήτησης στοιχείων της βάσης δεδομένων.

Η DCL είναι γλώσσα που δίνει στο χρήστη τη δυνατότητα να ορίσει πότε θα οριστικοποιηθούν ή θα ακυρωθούν οι μεταβολές των δεδομένων, και να καθορίσει τα δικαιώματα των χρηστών στη δημιουργία ή χρήση πινάκων, δεικτών, όψεων κλπ.

Η γλώσσα χειρισμού δεδομένων της SQL μπορεί να εφαρμόζεται και στο εξωτερικό και στο εννοιολογικό επίπεδο. Αντίστοιχα, η γλώσσα ορισμού δεδομένων μπορεί να χρησιμοποιείται για να ορίζονται αντικείμενα στο εξωτερικό επίπεδο (απόψεις), στο εννοιολογικό επίπεδο (βασικοί πίνακες), και ακόμα – στα περισσότερα συστήματα, αν και όχι σ' αυτά που ακολουθούν το πρότυπο – στο εσωτερικό επίπεδο (π.χ. ευρετήρια).

1.5 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

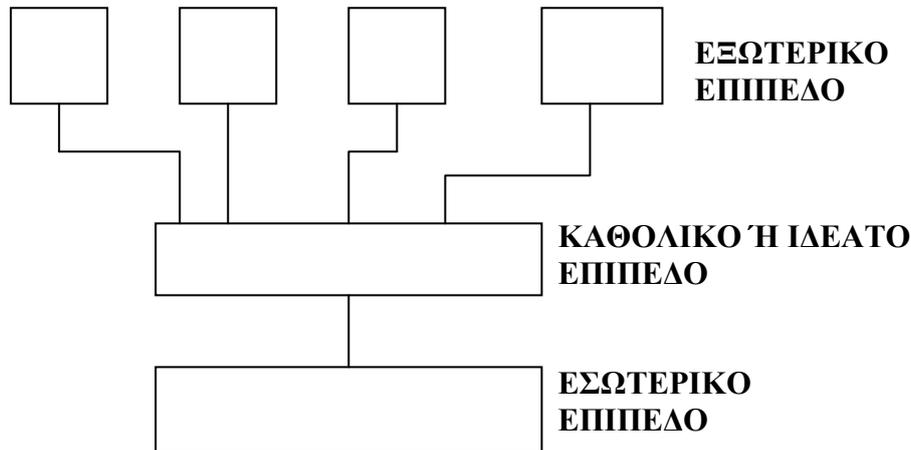
Η αρχιτεκτονική συστήματος βάσης δεδομένων σύμφωνα με την ορολογία του ANSI/SPARC διαιρείται σε τρία επίπεδα (ΕΙΚΟΝΑ 1.5) :

1. Εσωτερικό επίπεδο (internal level)

Το εσωτερικό επίπεδο αντιστοιχεί στην "εσωτερική οπτική γωνία" ή "εσωτερική άποψη" ή "εσωτερική όψη". Είναι η εικόνα της βάσης δεδομένων σύμφωνα με την οπτική γωνία αυτού που ασχολείται με την αποθήκευση

(οργάνωση) των δεδομένων. Αυτό δεν σημαίνει ότι η εσωτερική άποψη προσεγγίζει το επίπεδο των περιορισμών του υλικού.

ΕΙΚΟΝΑ1.5 : Τα τρία επίπεδα της αρχιτεκτονικής ANSI/SPARC



2. Καθολικό επίπεδο

Το καθολικό επίπεδο αντιστοιχεί στην “καθολική οπτική γωνία” ή “ιδεατή άποψη” ή “ιδεατή όψη”. Είναι η εικόνα της βάσης δεδομένων σύμφωνα με την οπτική γωνία αυτού που έχει καθολική εικόνα της βάσης. Το πρόσωπο αυτό είναι ο Διαχειριστής Βάσεων Δεδομένων. Ο Διαχειριστής έχει την ευθύνη της δημιουργίας του πλαισίου όπου οι χρήστες (προγραμματιστές και τελικοί χρήστες) επεξεργάζονται δεδομένα, “χτίζει” τη βάση δεδομένων, ορίζει και τροποποιεί τους τύπους πεδίων, τις ιδιότητες κλπ.

3. Εξωτερικό επίπεδο

Το εξωτερικό επίπεδο αντιστοιχεί στις “εξωτερικές οπτικές γωνίες” ή “εξωτερικές απόψεις” ή “εξωτερικές όψεις” των χρηστών. Είναι η εικόνα της βάσης δεδομένων σύμφωνα με την οπτική γωνία των χρηστών. Οι χρήστες ανήκουν στις παρακάτω κατηγορίες :

- ☐ *Προγραμματιστές που έχουν κάποιες γλώσσες προγραμματισμού στη διάθεση τους και το Σύστημα Διαχείρισης Βάσεων Δεδομένων για να δημιουργήσουν και να υποστηρίξουν βάσεις δεδομένων.*
- ☐ *Τελικοί χρήστες που χρησιμοποιούν ειδικές γλώσσες φτιαγμένες στα μέτρα τους που αποτελούν τμήμα του Συστήματος Διαχείρισης Βάσεων Δεδομένων ή συνεργάζονται με αυτό.*
- ☐ *Τελικοί χρήστες που χρησιμοποιούν προγράμματα εφαρμογής φτιαγμένα από τους προγραμματιστές για εισαγωγή και αναζήτηση στοιχείων από την βάση δεδομένων.*

Όλοι αυτοί οι χρήστες δεν ενδιαφέρονται για ολόκληρη τη βάση δεδομένων αλλά για τμήματα της. Δηλαδή, βλέπουν τη βάση σαν το σύνολο των πληροφοριών που τους ενδιαφέρει.

Συχνά η βάση δεδομένων σε εξωτερικό επίπεδο λέγεται λογική (logical data base) και σε καθολικό επίπεδο φυσική (physical data base).

1.6 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΧΡΗΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Τα πλεονεκτήματα που προκύπτουν από την χρήση βάσεων δεδομένων και τον έλεγχο των δεδομένων είναι τα εξής :

✓ Ο πλεονασμός μπορεί να μειωθεί στο ελάχιστο

Στα συμβατικά συστήματα (εκείνα που δεν είναι συστήματα βάσεων δεδομένων), η κάθε εφαρμογή έχει τα δικά της αρχεία. Αυτό το γεγονός οδηγεί πολύ συχνά σε υψηλό βαθμό πλεονασμού(επανάληψης) για τα αποθηκευμένα δεδομένα, με αποτέλεσμα τη σπατάλη αποθηκευτικού χώρου. Το αντίθετο συμβαίνει στα συστήματα βάσεων δεδομένων. Βέβαια αυτό δεν σημαίνει πως είναι πάντα δυνατό να εξαλειφθούν όλοι οι πλεονασμοί, ούτε πως είναι πάντα επιθυμητό. Είναι όμως απαραίτητο αυτός ο πλεονασμός να είναι καλά ελεγχόμενος – δηλαδή, το DBMS θα πρέπει να είναι ενήμερο γι' αυτή την κατάσταση, αν υπάρχει, και θα πρέπει να έχει την ευθύνη για τις “διαδιδόμενες ενημερώσεις” δεδομένων.

✓ Η ασυνέπεια μπορεί να αποφευχθεί (ως ένα βαθμό)

Στην ουσία, αυτό είναι συνέπεια της προηγούμενης παρατήρησης. Ας υποθέσουμε ότι ένα δεδομένο γεγονός του πραγματικού κόσμου – ας πούμε, το γεγονός ότι ο μαθητής Μ8 παρακολούθη μαθήματα του τμήματος Τ2 – αναπαρίσταται στην αποθηκευμένη βάση δεδομένων από δυο ξεχωριστές εγγραφές. Ας υποθέσουμε επίσης ότι το DBMS δεν είναι ενήμερο γι' αυτή τη “διπλοεγγραφή” (δηλαδή ότι ο πλεονασμός δεν είναι ελεγχόμενος). Τότε, είναι βέβαιο ότι θα υπάρξουν περιπτώσεις που οι δυο καταχωρίσεις δε θα συμφωνούν μεταξύ τους – περιπτώσεις που η μια καταχώριση θα έχει ενημερωθεί ενώ η άλλη όχι. Σε αυτές τις περιπτώσεις, λέμε ότι η βάση δεδομένων είναι **ασυνεπής**. Είναι προφανές ότι μια ασυνεπής βάση δεδομένων μπορεί να δώσει στους χρήστες της λανθασμένες ή αντιφατικές πληροφορίες.

Είναι επίσης ξεκάθαρο ότι, αν το συγκεκριμένο γεγονός αναπαρίσταται από μια μόνο καταχώριση (δηλαδή, αν δεν υπάρχει πλεονασμός), τέτοια ασυνέπεια δεν μπορεί να παρουσιαστεί. Εναλλακτικά, αν υπάρχει πλεονασμός, αλλά είναι ελεγχόμενος (δηλαδή, είναι γνωστός στο DBMS), το DBMS θα μπορεί να εγγυηθεί ότι η βάση δεδομένων δε θα είναι ποτέ ασυνεπής στα μάτια του χρήστη, εξασφαλίζοντας ότι κάθε αλλαγή που θα γίνεται σε οποιαδήποτε από τις δυο καταχωρίσεις θα γίνεται αυτόματα και στην άλλη. Αυτή η διαδικασία είναι γνωστή με το όνομα **διάδοση ενημερώσεων** – όπου ο όρος “ενημέρωση” θεωρείται ότι συμπεριλαμβάνει όλες τις πράξεις εισαγωγής, διαγραφής, και τροποποίησης δεδομένων.

✓ **Τα δεδομένα μπορούν να είναι κοινόχρηστα**

Ο μερισμός (sharing – κοινή χρήση) δε σημαίνει μόνο ότι οι υπάρχουσες εφαρμογές μπορούν να μοιράζονται τα δεδομένα της βάσης δεδομένων αλλά και ότι είναι δυνατή η ανάπτυξη νέων εφαρμογών που θα μπορούν να χρησιμοποιούν τα ίδια αποθηκευμένα δεδομένα. Με άλλα λόγια, είναι δυνατό να ικανοποιούνται οι απαιτήσεις νέων εφαρμογών σε ότι αφορά τα δεδομένα, χωρίς να είναι αναγκαία η δημιουργία νέων αποθηκευμένων δεδομένων.

✓ **Μπορούν να επιβάλλονται πρότυπα**

Με τον κεντρικό έλεγχο της βάσης δεδομένων, ο υπεύθυνος διαχείρισης βάσεων δεδομένων (DBA) – κάτω από την επίβλεψη του υπεύθυνου διαχείρισης δεδομένων – μπορεί να εξασφαλίσει ότι θα τηρούνται όλα τα σχετικά πρότυπα για την αναπαράσταση των δεδομένων. Η τυποποίηση της αναπαράστασης των δεδομένων διευκολύνει ιδιαίτερα την ανταλλαγή δεδομένων, δηλαδή τη μεταφορά δεδομένων μεταξύ διαφορετικών συστημάτων. Τα πρότυπα ονομασίας και τεκμηρίωσης των δεδομένων είναι επίσης πολύ επιθυμητά για να διευκολύνεται ο μερισμός και η καλύτερη κατανόηση των δεδομένων.

✓ **Μπορούν να εφαρμόζονται περιορισμοί ασφαλείας**

Έχοντας πλήρη δικαιοδοσία πάνω στη βάση δεδομένων, ο DBA (α) μπορεί να εξασφαλίσει ότι η πρόσβαση στη βάση δεδομένων θα μπορεί να γίνεται μόνο μέσω των κατάλληλων καναλιών και, κατά συνέπεια, (β) μπορεί να ορίσει κανόνες ασφαλείας με βάση τους οποίους θα γίνεται έλεγχος κάθε φορά που θα υπάρχει απόπειρα προσπέλασης εμπιστευτικών δεδομένων. Είναι δυνατό να καθοριστούν διαφορετικοί κανόνες για το κάθε είδος προσπέλασης (ανάκληση, εισαγωγή, διαγραφή κ.λ.π.) σε κάθε στοιχείο πληροφοριών της βάσης δεδομένων. Σημειώστε, όμως, ότι χωρίς τέτοιους κανόνες η ασφάλεια των δεδομένων μπορεί να διατρέχει μεγαλύτερο κίνδυνο από ότι σε ένα παραδοσιακό σύστημα αρχειοθέτησης (με γεωγραφικά απομακρυσμένα αρχεία) δηλαδή, η κεντρική φύση ενός συστήματος βάσης δεδομένων απαιτεί κατά κάποιον τρόπο την ύπαρξη ενός καλού συστήματος ασφαλείας.

✓ **Μπορεί να διατηρείται η ακεραιότητα**

Το πρόβλημα της ακεραιότητας είναι να εξασφαλίζεται ότι τα δεδομένα της βάσης δεδομένων είναι ακριβή. Η ασυμφωνία μεταξύ δυο καταχωρίσεων που υποτίθεται ότι αντιπροσωπεύουν το ίδιο “γεγονός” είναι ένα παράδειγμα έλλειψης ακεραιότητας φυσικά, αυτό το συγκεκριμένο πρόβλημα μπορεί να παρουσιαστεί μόνο αν υπάρχει πλεονασμός στα αποθηκευμένα δεδομένα. Ακόμα και αν δεν υπάρχει πλεονασμός όμως, πάλι υπάρχει περίπτωση να περιέχει η βάση δεδομένων λανθασμένες πληροφορίες. Για παράδειγμα, ένας μαθητής μπορεί να εμφανίζεται ότι απουσίασε κάποια μέρα 20 ώρες, αντί για 2, ή ότι έχει γραφτεί

στο τμήμα T10 ενώ δεν υπάρχει τέτοιο τμήμα. Ο κεντρικός έλεγχος της βάσης δεδομένων μπορεί να βοηθήσει να αποφευχθούν τέτοιου είδους προβλήματα – στο βαθμό που είναι δυνατό να αποφευχθούν – επιτρέποντας στον υπεύθυνο διαχείρισης δεδομένων να ορίζει κανόνες ακεραιότητας με βάση τους οποίους θα γίνεται έλεγχος κάθε φορά που επιχειρείται κάποια πράξη ενημέρωσης.

Αξίζει να επισημάνουμε ότι η ακεραιότητα των δεδομένων έχει πολύ μεγαλύτερη σημασία σε ένα σύστημα βάσης δεδομένων πολλών χρηστών από ότι σε ένα περιβάλλον “ιδιωτικών αρχείων”, ακριβώς επειδή η βάση δεδομένων είναι μεριζώμενη. Αυτό συμβαίνει γιατί, χωρίς τους κατάλληλους ελέγχους, μπορεί ένας χρήστης να ενημερώσει τη βάση δεδομένων με εσφαλμένο τρόπο, δημιουργώντας με αυτόν τον τρόπο λανθασμένα δεδομένα και “μολύνοντας” τους υπόλοιπους χρήστες με αυτά.

✓ **Οι αντικρουόμενες απαιτήσεις μπορούν να εξισορροπούνται**

Γνωρίζοντας τις συνολικές απαιτήσεις της επιχείρησης – σε αντιδιαστολή με τις απαιτήσεις των μεμονωμένων χρηστών – ο DBA μπορεί να δομήσει το σύστημα με τέτοιο τρόπο ώστε να παρέχει γενικές υπηρεσίες που να είναι “βέλτιστες για την επιχείρηση”. Για παράδειγμα, είναι δυνατό να επιλεγεί μια αναπαράσταση των αποθηκευμένων δεδομένων που θα παρέχει γρήγορη πρόσβαση στις σημαντικότερες εφαρμογές.

Τα περισσότερα από τα πλεονεκτήματα που παραθέσαμε εδώ είναι ίσως αρκετά προφανή. Θα πρέπει όμως να προσθέσουμε στη λίστα ένα ακόμα σημείο, που ίσως δεν είναι και τόσο προφανές: την ανεξαρτησία των δεδομένων. Για να είμαστε ακριβείς, η ανεξαρτησία των δεδομένων είναι στόχος των συστημάτων βάσεων δεδομένων, και όχι απαραίτητα πλεονέκτημα.

ΚΕΦΑΛΑΙΟ 2

ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΟΡΓΑΝΙΣΜΟΥ ΞΕΝΩΝ ΓΛΩΣΣΩΝ

2.1 Η ΕΦΑΡΜΟΓΗ MICROSOFT ACCESS

Η Access είναι μια ιδιαίτερα ισχυρή εφαρμογή με δεκάδες λειτουργίες οι οποίες μας βοηθούν στην αποτελεσματική διαχείριση των δεδομένων/πληροφοριών μας. Επειδή η Microsoft επεκτείνει διαρκώς τις δυνατότητες της Access η εφαρμογή περιλαμβάνει μια τεράστια γκάμα εργαλείων και λειτουργιών.

Η Access είναι μια εφαρμογή διαχείρισης σχεσιακών βάσεων δεδομένων, η οποία μας επιτρέπει να χρησιμοποιούμε διαφορετικούς τύπους δεδομένων μαζί.

2.1.1 ΤΑ ΣΤΟΙΧΕΙΑ ΜΙΑΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ MICROSOFT ACCESS

Η καρδιά μιας βάσης δεδομένων είναι τα δεδομένα που περιέχει. Αλλά υπάρχουν και άλλα σημαντικά στοιχεία τα οποία αναφέρονται συνήθως με τον όρο “αντικείμενα” σε μια βάση δεδομένων της Access. Ακολουθεί μια συνοπτική αναφορά στα είδη των αντικειμένων με τα οποία χρησιμοποιούνται στην Access:

Πίνακες: Περιέχουν τα δεδομένα .

Ερωτήματα: Μας επιτρέπουν να απευθύνουμε ερωτήσεις για την εύρεση των επιθυμητών δεδομένων, ή να κάνουμε αλλαγές στα δεδομένα.

Φόρμες: Είναι κατάλληλες για την παρουσίαση και τροποποίηση των πληροφοριών .

Σελίδες: Είναι HTML αρχεία, τα οποία μας επιτρέπουν να παρουσιάζουμε τα δεδομένα της Access με την εφαρμογή Internet Explorer.

Εκθέσεις: Είναι κατάλληλες για την δημιουργία συγκεντρωτικών αναφορών και την εκτύπωση των δεδομένων.

Μακροεντολές: Εκτελούν μια ή περισσότερες ενέργειες διαχείρισης βάσεων δεδομένων αυτόματα.

2.1.2 ΣΥΣΧΕΤΙΣΜΟΣ ΠΙΝΑΚΩΝ ΜΕ ΤΗΝ ACCESS

Για να παρουσιάσει τα δεδομένα ενός πίνακα μαζί με τις αντίστοιχες εγγραφές ενός άλλου πίνακα, η Access πρέπει να μπορεί να συσχετίσει τους δύο πίνακες. Για να συσχετιστούν οι δύο πίνακες θα πρέπει να έχουν πεδία με κοινές τιμές. Για παράδειγμα, στη βάση δεδομένων ενός Εκπαιδευτικού Οργανισμού Ξένων Γλωσσών(αναλύεται παρακάτω σ’ αυτό το κεφάλαιο), οι πίνακες ΣΤΟΙΧΕΙΑ

ΜΑΘΗΤΗ και ΒΑΘΜΟΙ έχουν κοινό το πεδίο ΚΩΔ_ΜΑΘΗΤΗ. Στον πίνακα ΣΤΟΙΧΕΙΑ ΜΑΘΗΤΗ, το πεδίο ΚΩΔ_ΜΑΘΗΤΗ λειτουργεί σαν μοναδικό προσδιοριστικό για την εγγραφή κάθε ατόμου, ενώ στον πίνακα ΒΑΘΜΟΙ το ίδιο πεδίο υποδεικνύει με ποιο άτομο σχετίζεται κάθε βαθμολογία (εγγραφή). Χρησιμοποιώντας το πεδίο ΚΩΔ_ΜΑΘΗΤΗ σαν σύνδεσμο, η Access μπορεί να μεταβεί στον πίνακα ΣΤΟΙΧΕΙΑ ΜΑΘΗΤΗ και να αναζητήσει το όνομα του μαθητή για κάθε εγγραφή του πίνακα ΒΑΘΜΟΙ.

2.1.2.1 ΤΥΠΟΙ ΣΧΕΣΕΩΝ

Μεταξύ δυο πινάκων μπορεί να υπάρχει μια σχέση η οποία θα εμπίπτει πάντα σε μια από τις ακόλουθες κατηγορίες, ανάλογα με το πόσες φορές μπορούν να εμφανίζονται οι τιμές του πεδίου σύνδεσης σε κάθε πίνακα:

Ένα Προς Ένα: Αυτός ο τύπος σχέσης υφίσταται όταν υπάρχει μόνο μια εγγραφή σε κάθε πλευρά της σχέσης για κάθε τιμή του πεδίου σύνδεσης.

Ένα Προς Πολλά: Όταν για την ίδια τιμή του πεδίου σύνδεσης μπορούν να υπάρχουν περισσότερες από μια εγγραφές στη μια πλευρά της σχέσης, καταλήγεται με μια σχέση ένα προς πολλά. Η σχέση μεταξύ των πινάκων ΣΤΟΙΧΕΙΑ ΜΑΘΗΤΗ και ΒΑΘΜΟΙ είναι ένα παράδειγμα σχέσης ένα προς πολλά. Υπάρχει μόνο μια εγγραφή στον πίνακα ΣΤΟΙΧΕΙΑ ΜΑΘΗΤΗ για κάθε άτομο, αλλά μπορείτε να έχετε πολλές εγγραφές στον πίνακα ΒΑΘΜΟΙ για κάθε άτομο.

Πολλά Προς Πολλά: Αυτός ο τύπος σχέσης περιγράφει την περίπτωση στην οποία οι τιμές του πεδίου σύνδεσης μπορούν να εμφανίζονται σε πολλαπλές εγγραφές και των δύο πινάκων που υπάρχουν στις δυο πλευρές μιας σχέσης.

Στην Access, η φύση μιας σχέσης μεταξύ δυο πινάκων καθορίζεται από το εάν τα πεδία σύνδεσης των δύο πινάκων είναι πεδία-κλειδιά ή όχι. Όταν ένας πίνακας έχει ένα πεδίο-κλειδί, αυτό σημαίνει ότι ένα πεδίο του – ή ένας συνδυασμός πεδίων – μπορεί να προσδιορίσει με μοναδικό τρόπο κάθε εγγραφή του πίνακα. Σε ένα τέτοιο πίνακα, κάθε εγγραφή πρέπει να έχει μοναδική τιμή στο πεδίο κλειδί (ή στον συνδυασμό των πεδίων που χρησιμοποιούνται σαν κλειδί). Λόγω αυτού του περιορισμού, η Access ξέρει εάν μπορεί να υπάρχει μόνο μια πιθανή εγγραφή για μια τιμή πεδίου σε έναν συγκεκριμένο πίνακα. Για παράδειγμα, εάν η Access ψάξει στον πίνακα ΣΤΟΙΧΕΙΑ ΜΑΘΗΤΗ και διαπιστώσει ότι χρησιμοποιεί σαν κλειδί το πεδίο ΚΩΔ_ΜΑΘΗΤΗ, ξέρει ότι μπορεί να υπάρχει μόνο μια εγγραφή για κάθε ΚΩΔ_ΜΑΘΗΤΗ σ' αυτόν τον πίνακα.

2.1.2.2 ΠΕΔΙΑ-ΚΛΕΙΔΙΑ

Ένα κλειδί αποτελείται από ένα ή περισσότερα πεδία, των οποίων οι τιμές προσδιορίζουν με μοναδικό τρόπο κάθε εγγραφή ενός πίνακα. Τα πεδία-κλειδιά σας παρέχουν ορισμένες ειδικές δυνατότητες σε μια βάση δεδομένων :

- Συχνά χρησιμοποιούνται για τη σύνδεση δύο πινάκων. Καθορίζουν επίσης τη φύση μιας σχέσης.

- Η Access δημιουργεί αυτόματα ένα πρωτεύον ευρετήριο χρησιμοποιώντας οποιαδήποτε πεδία-κλειδιά βρίσκει σε έναν πίνακα. Το ευρετήριο αυτό διευκολύνει την Access στον εντοπισμό των τιμών των κλειδιών και επιταχύνει οποιοσδήποτε αναζητήσεις ή άλλες λειτουργίες οι οποίες αναζητούν τιμές στα πεδία-κλειδιά.

2.2 ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Η σχεδίαση είναι πιθανώς η σημαντικότερη διαδικασία για οποιαδήποτε βάση δεδομένων. Πριν αρχίσουμε να δημιουργούμε πίνακες, φόρμες, ή εκθέσεις, το πρώτο πράγμα που πρέπει να γίνει είναι να εξακριβώσουμε τι χρειαζόμαστε από την βάση δεδομένων.

Καλό θα ήταν ,λοιπόν, να αφιερώσουμε λίγο χρόνο για μελετήσουμε και να απαντήσουμε στα ακόλουθα ερωτήματα , πριν αρχίσουμε να δημιουργούμε μια βάση δεδομένων :

- ◆ Από πού θα προέρχονται τα δεδομένα μου; Θα τα πληκτρολογώ μόνος μου, ή μπορώ να τα εισάγω από μια άλλη πηγή; Εάν τα δεδομένα υπάρχουν ήδη σε ηλεκτρονική μορφή, μπορώ να τα διασύνδεω απευθείας σ' έναν πίνακα της Access; Και τέλος, θα χρειάζεται να μοιράζομαι τα δεδομένα με άλλους ανθρώπους;
- ◆ Εάν τα δεδομένα που χρησιμοποιώ είναι αποθηκευμένα σε έναν διακομιστή SQL, πρέπει να χρησιμοποιήσω μια βάση δεδομένων της Access ή ένα αρχείο έργου;
- ◆ Με ποιες προδιαγραφές θέλω να συμμορφώνονται τα δεδομένα;
- ◆ Πως μπορούν να παρουσιάζονται τα δεδομένα στην οθόνη;
- ◆ Θα πρέπει να εκτελούνται κάποιοι υπολογισμοί ή κάποιες άλλες διεργασίες στα δεδομένα;
- ◆ Ποιες εκθέσεις ή γραφήματα χρειάζεται να εκτυπώνω από τα δεδομένα;
- ◆ Μήπως είναι αναγκαία η πρόσβαση στα δεδομένα μέσω του Web;
- ◆ Ποια είδη εργασιών θα χρειάζεται να επαναλαμβάνονται ξανά και ξανά;
- ◆ Υπάρχουν άλλοι άνθρωποι οι οποίοι θα χρησιμοποιούν τη βάση δεδομένων μου; Εάν ναι, τι μπορώ να κάνω για να διευκολύνω την εργασία τους;

Η αναλυτική εξέταση αυτών των ερωτημάτων έχει σαν στόχο να ελαχιστοποιήσει τον κόπο και τον χρόνο που θα αφιερώσουμε για το στήσιμο μια βάσης δεδομένων.

2.2.1 ΣΧΕΔΙΑΣΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΟΡΓΑΝΙΣΜΟΥ

Σύμφωνα με τις απαιτήσεις του Εκπαιδευτικού Οργανισμού – δηλαδή, ποια δεδομένα πρέπει να αποθηκεύονται στην βάση δεδομένων, ποιες πληροφορίες θα παίρνουμε από αυτά και με ποια μορφή – προέκυψαν τα εξής:

- Πρώτο και βασικότερο η βάση δεδομένων πρέπει να περιέχει όλες τις πληροφορίες που αφορούν τους μαθητές :
 1. Πλήρη **στοιχεία του μαθητή** - ονοματεπώνυμο, ονόματα των γονέων, διευθύνσεις, τηλέφωνα.
 2. Στοιχεία για την **εγγραφή** του μαθητή – τμήμα, αίθουσα, ημερομηνία εγγραφής και ημερομηνία έναρξης.
 3. **Πληρωμές των διδάκτρων** του μαθητή τα οποία χωρίζονται στα **Συμφωνηθέντα** και τα **Πληρωθέντα**. Τα Συμφωνηθέντα αφορούν την αρχική συμφωνία για το συνολικό πόσο των διδάκτρων, όλης της περιόδου, και το ποσό κάθε δόσης(ανά μήνα ή δίμηνο κλπ – αυτό αποφασίζεται από το μαθητή ή από τους κηδεμόνες του όταν είναι ανήλικος). Τα Πληρωθέντα αφορούν τις πληρωμές κάθε δόσης για τα δίδακτρα του μαθητή –ποσό δόσης, ημερομηνία πληρωμής, υπόλοιπο χρέος, αριθμός απόδειξης.
 4. Οι **βαθμοί** όλων των τεστ – αριθμός του τεστ, βαθμός, τρίμηνο εξέτασης.
 5. Οι **απουσίες** του μαθητή – αριθμός ωρών, ημερομηνία.
- Στη συνέχεια ο Εκπαιδευτικός Οργανισμός θέλει να κρατάει στη βάση δεδομένων και τα πλήρη στοιχεία του προσωπικού που εργάζεται σ' αυτόν διαχωρισμένο ως εξής :
 1. Το **Διοικητικό Προσωπικό**
 2. Οι **Καθηγητές**
 3. Οι λοιποί **Υπάλληλοι**
- Ένα ακόμα βασικό κομμάτι αυτής της βάσης δεδομένων είναι το **ταμείο** για τα έσοδα και έξοδα του Εκπαιδευτικού Οργανισμού. Υπάρχει η δυνατότητα δημιουργίας Λογαριασμών (Εσόδων και Εξόδων) και Χρέωση ή Πίστωση αυτών.
- Επίσης στην βάση δεδομένων πρέπει να υπάρχουν και τα στοιχεία των **τμημάτων** - ποια γλώσσα και σε ποιο επίπεδο εκμάθησης (π.χ Lower).
- Τελευταίο τμήμα της βάσης δεδομένων είναι για την **βιβλιοθήκη** του Εκπαιδευτικού Οργανισμού. Όλα τα βιβλία πρέπει να καταχωρηθούν στην βάση – τίτλοι, συγγραφείς, εκδόσεις, ποσότητες.

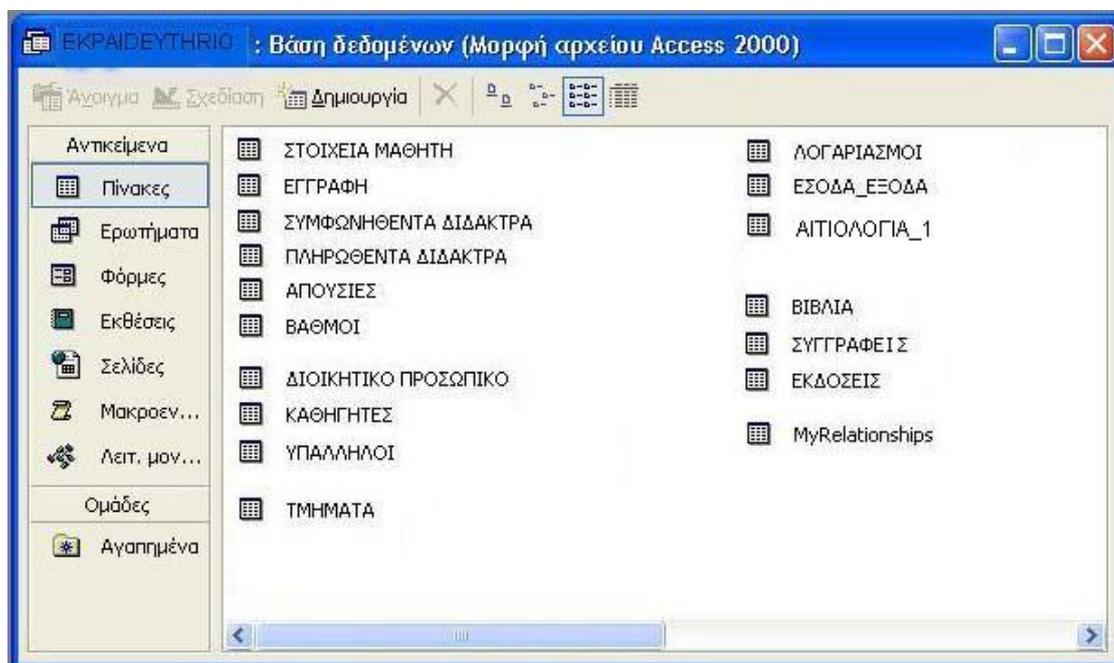
Όλα αυτά που αναλύσαμε παραπάνω για την Βάση Δεδομένων θα υλοποιηθούν με την βοήθεια της Microsoft Access.

2.3 ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΟΡΓΑΝΙΣΜΟΥ ΞΕΝΩΝ ΓΛΩΣΣΩΝ

Με την βοήθεια λοιπόν της Microsoft Access ξεκινάμε να δημιουργήσουμε την Βάση Δεδομένων για τον Εκπαιδευτικό Οργανισμό Ξένων Γλωσσών.

Αρχικά δημιουργούμε μια κενή βάση δεδομένων (**ΕΚΠΑΙΔΕΥΤΗΡΙΟ**) και έχοντας υπόψη όσα προαναφέρθηκαν στην σχεδίαση της βάσης δεδομένων δημιουργούμε τους απαραίτητους πίνακες, οι οποίοι παίρνουν ονόματα τους ανάλογα με το τι είδους στοιχεία αποθηκεύονται στον καθένα. Τα ονόματα αυτά φαίνονται στη ΕΙΚΟΝΑ2.1.

ΕΙΚΟΝΑ2.1 : Οι πίνακες της Βάσης Δεδομένων



Όπως φαίνεται στην παραπάνω εικόνα οι πίνακες που δημιουργήθηκαν είναι 17. Οι πρώτοι 6 (ΣΤΟΙΧΕΙΑ ΜΑΘΗΤΗ , ΕΓΓΡΑΦΗ , ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ , ΠΛΗΡΩΘΕΝΤΑ ΔΙΔΑΚΤΡΑ , ΑΠΟΥΣΙΕΣ , ΒΑΘΜΟΙ) αφορούν αποκλειστικά και μόνο τον μαθητή και όλες τις ενέργειες του σε σχέση με τον Εκπαιδευτικό Οργανισμό.

Οι επόμενοι 3 πίνακες (ΔΙΟΙΚΗΤΙΚΟ ΠΡΟΣΩΠΙΚΟ , ΚΑΘΗΓΗΤΕΣ , ΥΠΑΛΛΗΛΟΙ) αφορούν όλους όσους εργάζονται στον Εκπαιδευτικό Οργανισμό, από τα άτομα της διοίκησης, τους καθηγητές, μέχρι και τον τελευταίο υπάλληλο.

Ο πίνακας ΤΜΗΜΑΤΑ αφορά τα επίπεδα εκμάθησης κάθε γλώσσα, η οποία διδάσκεται στον Εκπαιδευτικό Οργανισμό.

Οι 3 επόμενοι πίνακες (ΛΟΓΑΡΙΑΣΜΟΙ , ΕΣΟΔΑ _ ΕΞΟΔΑ, ΑΙΤΙΟΛΟΓΙΑ_1) αφορούν το ταμείο του Εκπαιδευτικού Οργανισμού, όπου λεπτομερώς αποθηκεύονται όλες οι κινήσεις Εσόδων / Εξόδων.

Οι υπόλοιποι 3 πίνακες (ΒΙΒΛΙΑ , ΣΥΓΓΡΑΦΕΙΣ , ΕΚΔΟΣΕΙΣ) αφορούν την βιβλιοθήκη του Εκπαιδευτικού Οργανισμού με αναλυτικά στοιχεία για όλα τα βιβλία.

Ο τελευταίος πίνακας (MyRelationships) δεν αφορά ούτε τον Εκπαιδευτικό Οργανισμό ούτε τους χρήστες που θα διαχειρίζονται της βάση δεδομένων. Στον πίνακα αυτό αποθηκεύονται πληροφορίες (ΕΙΚΟΝΑ2.2) για τις σχέσεις μεταξύ των πινάκων (για τις οποίες θα μιλήσουμε αργότερα) και αφορά τον προγραμματιστή, γιατί ο πίνακας αυτός θα χρειαστεί για την δημιουργία του προγράμματος διαχείρισης της Βάσης Δεδομένων (Κεφάλαιο 3).

ΕΙΚΟΝΑ2.2 : Ο Πίνακας MyRelationships

Όνομα πεδίου	Τύπος δεδομένων	Περιγραφή
ccolumn	Αριθμός	
grbit	Αριθμός	
icolumn	Αριθμός	
szColumn	Κείμενο	
szObject	Κείμενο	
szReferencedColumn	Κείμενο	
szReferencedObject	Κείμενο	
szRelationship	Κείμενο	

Όλοι οι παραπάνω πίνακες (εκτός από τον MyRelationships) δημιουργήθηκαν σε προβολή σχεδίασης, όπου δηλώθηκαν όλα τα πεδία σε κάθε πίνακα και τα αντίστοιχα χαρακτηριστικά τους: μέγεθος, μορφή, αν επιτρέπονται διπλότυπα ή όχι κλπ.

Επίσης σ' αυτό το στάδιο καθορίζονται και τα πρωτεύοντα κλειδια.

Ένα παράδειγμα βλέπετε παρακάτω στην ΕΙΚΟΝΑ2.3 για τον πίνακα ΣΤΟΙΧΕΙΑ ΜΑΘΗΤΗ

ΕΙΚΟΝΑ2.3 : Ο Πίνακας ΣΤΟΙΧΕΙΑ ΜΑΘΗΤΗ

Όνομα πεδίου	Τύπος δεδομένων	Περιγραφή
ΚΩΔΙΚΟΣ	Αυτόματη Αριθμ.	
ΟΝΟΜΑ	Κείμενο	
ΕΠΩΝΥΜΟ	Κείμενο	
ΟΝΟΜΑ ΠΑΤΡΟΣ	Κείμενο	
ΟΝΟΜΑ ΜΗΤΡΟΣ	Κείμενο	
ΗΜ_ΓΕΝΝΗΣΕΩΣ	Ημερομηνία/Ωρ	
ΔΙΕΥΘΥΝΣΗ ΚΑΤΟΙΚΙΑΣ	Κείμενο	
ΠΟΛΗ	Κείμενο	
ΤΑΧ_ΚΩΔΙΚΑΣ	Αριθμός	
ΤΗΛΕΦΩΝΟ	Αριθμός	
ΚΙΝΗΤΟ	Αριθμός	

2.4 ΣΧΕΣΕΙΣ ΤΩΝ ΠΙΝΑΚΩΝ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΟΡΓΑΝΙΣΜΟΥ

Συνοπτικά, μια βάση δεδομένων είναι μια οργανωμένη συλλογή δεδομένων. Ο τηλεφωνικός κατάλογος είναι ένα απλό και τυπικό παράδειγμα βάσης δεδομένων. Το ιστορικό των παραγγελιών, των τιμολογίων και των πληρωμών μιας εταιρείας είναι επίσης ένα παράδειγμα πολυπλοκότερης βάσης δεδομένων.

Η Access είναι μια ευέλικτη εφαρμογή η οποία μπορεί να χειριστεί τόσο τις απλές, όσο και τις πολύπλοκες βάσεις δεδομένων. Είναι επίσης μια εφαρμογή διαχείρισης *σχεσιακών βάσεων δεδομένων*, πράγμα που σημαίνει ότι σας επιτρέπει να ορίζετε σχέσεις μεταξύ διαφορετικών τύπων πληροφοριών (π.χ. οι μαθητές και οι απουσίες τους), έτσι ώστε να μπορείτε να τις χρησιμοποιείτε μαζί.

Η Βάση Δεδομένων που μόλις δημιουργήσαμε για τον Εκπαιδευτικό Οργανισμό Ξένων Γλωσσών ανήκει στις σχεσιακές βάσεις δεδομένων. Στην ΕΙΚΟΝΑ2.4 φαίνονται όλοι οι πίνακες της βάση μας που σχετίζονται μεταξύ τους.

Όλες οι σχέσεις είναι Ένα – Προς – Πολλά . Αναλυτικά ποιοι πίνακες συσχετίζονται και με ποια πεδία είναι τα εξής :

Ο πίνακας **ΣΤΟΙΧΕΙΑ ΜΑΘΗΤΗ** (*ΚΩΔΙΚΟΣ*) σχετίζεται με τους πίνακες :

- ✓ ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ (*ΚΩΔ_ΜΑΘΗΤΗ*)
- ✓ ΠΛΗΡΩΘΕΝΤΑ ΔΙΔΑΚΤΡΑ (*ΚΩΔ_ΜΑΘΗΤΗ*)
- ✓ ΕΓΓΡΑΦΗ (*ΚΩΔ_ΜΑΘΗΤΗ*)
- ✓ ΒΑΘΜΟΙ (*ΚΩΔ_ΜΑΘΗΤΗ*)
- ✓ ΑΠΟΥΣΙΕΣ (*ΚΩΔ_ΜΑΘΗΤΗ*)

Ο πίνακας **ΤΜΗΜΑΤΑ** (*ΚΩΔΙΚΟΣ ΤΜΗΜΑΤΟΣ*) σχετίζεται με τον πίνακα:

- ✓ ΕΓΓΡΑΦΗ (*ΚΩΔ_ΤΜΗΜΑΤΟΣ*)

Ο πίνακας **ΑΙΤΙΟΛΟΓΙΑ_1** (*ΑΙΤΙΟΛΟΓΙΑ*) σχετίζεται με τον πίνακα:

- ✓ ΛΟΓΑΡΙΑΣΜΟΙ (*ΑΙΤΙΟΛΟΓΙΑ_1*)

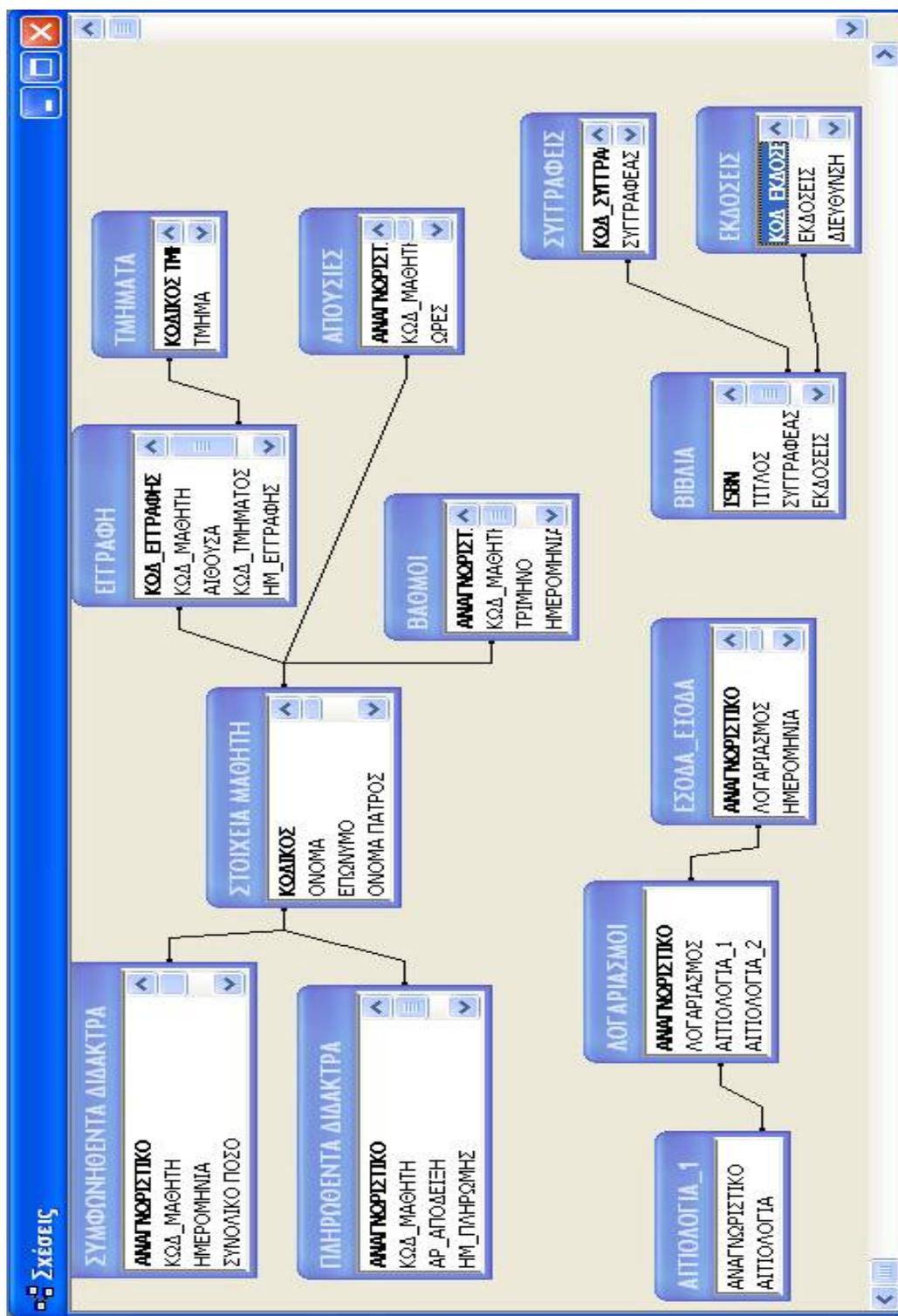
Ο πίνακας **ΛΟΓΑΡΙΑΣΜΟΙ** (*ΛΟΓΑΡΙΑΣΜΟΣ*) σχετίζεται με τον πίνακα:

- ✓ ΕΣΟΔΑ_ΕΞΟΔΑ (*ΛΟΓΑΡΙΑΣΜΟΣ*)

Οι πίνακες **ΣΥΓΓΡΑΦΕΙΣ** (*ΣΥΓΓΡΑΦΕΑΣ*) και **ΕΚΔΟΣΕΙΣ** (*ΕΚΔΟΣΕΙΣ*) σχετίζονται με τον πίνακα:

- ✓ ΒΙΒΛΙΑ (*ΣΥΓΓΡΑΦΕΑΣ* , *ΕΚΔΟΣΕΙΣ*) στο κάθε πεδίο αντίστοιχα.

ΕΙΚΟΝΑ2.4 : Σχέσεις των πινάκων Βάσης Δεδομένων (ΕΚΠΑΙΔΕΥΤΗΡΙΟ)



ΚΕΦΑΛΑΙΟ 3

ΠΡΟΓΡΑΜΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΟΡΓΑΝΙΣΜΟΥ ΞΕΝΩΝ ΓΛΩΣΣΩΝ

3.1 ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ VISUAL BASIC

Πριν από αρκετά χρόνια, όταν η Visual Basic ήταν μια "μικρή" γλώσσα, ένα πλήρες εγχειρίδιο θα κάλυπτε κάθε στοιχείο της. Από την πρώτη της έκδοση όμως, η Visual Basic έχει εξελιχθεί σε ένα μεγάλο περιβάλλον ανάπτυξης, που καλύπτει κάθε στοιχείο προγραμματισμού, από τις εκπαιδευτικές εφαρμογές μέχρι τον προγραμματισμό βάσεων δεδομένων και από τις οικονομικές εφαρμογές μέχρι την ανάπτυξη στοιχείων του Internet. Η συγγραφή ενός βιβλίου που απλώς εισάγει όλα αυτά τα στοιχεία δεν είναι εφικτή. Αυτό ισχύει ακόμη παραπάνω για ένα πλήρες εγχειρίδιο.

Σχεδόν όλες οι εμπορικές εφαρμογές χρειάζονται να αποθηκεύσουν μεγάλες ποσότητες δεδομένων, οργανωμένες σε μια μορφή που απλοποιεί την επαναφορά. Αυτό γίνεται με ένα σύστημα διαχείρισης βάσεων δεδομένων (DBMS), ένα μηχανισμό για χειρισμό δεδομένων σε πίνακες με εντολές υψηλού επιπέδου. Το σύστημα διαχείρισης βάσεων δεδομένων αποκρύβει τις λεπτομέρειες χαμηλού επιπέδου, όπως το πώς αποθηκεύονται τα δεδομένα σε μια βάση δεδομένων, και απελευθερώνει τον προγραμματιστή ώστε να μπορεί να επικεντρώσει την προσοχή του στην διαχείριση πληροφοριών, αντί να ασχολείται με τον χειρισμό των αρχείων ή με την συντήρηση των δεσμών ανάμεσα σε αυτά.

Η Visual Basic παρέχει πολλά εργαλεία για δημιουργία και προσπέλαση βάσεων δεδομένων τόσο σε μεμονωμένα συστήματα, όσο και σε δίκτυα.

3.2 ΔΗΜΙΟΥΡΓΙΑ ΠΡΟΓΡΑΜΜΑΤΟΣ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΟΡΓΑΝΙΣΜΟΥ

Οι Εκπαιδευτικοί Οργανισμοί χρειάζονται σήμερα σύγχρονα Ολοκληρωμένα Συστήματα Πληροφορικής που να αυτοματοποιούν όλες τις λειτουργίες τους και να υποστηρίζουν συνολικά τις καθημερινές δραστηριότητες της επιχείρησης.

Το πρόγραμμα που θα δημιουργήσουμε, Project1, αποτελεί ολοκληρωμένη και ευέλικτη λύση για την αποδοτικότερη οργάνωση της επιχείρησης, αφού προσφέρει :

- ◆ Απόλυτο έλεγχο σε κάθε τομέα της επιχείρησης
- ◆ Ταχύτατη διεκπεραίωση των άλλοτε χρονοβόρων διαδικασιών

- ◆ Άμεση ενημέρωση για όλα τα δεδομένα της επιχείρησης
- ◆ Αξιοπιστία και απόδοση σε κάθε λειτουργία
- ◆ Συνολικό εκσυγχρονισμό του management της επιχείρησης

Το πρόγραμμα αυτό δίνει τη δυνατότητα να έχει την εξελεγμένη μηχανογραφημένη λειτουργία που πρέπει να διαθέτει ένας σημερινός Εκπαιδευτικός Οργανισμός Ξένων Γλωσσών.

Το πρόγραμμα διαθέτει :

Ηλεκτρονική διαχείριση όλων των στοιχείων :

Μαθητές, καθηγητές, διοικητικό προσωπικό, υπάλληλοι τάξεις, τμήματα, βαθμοί, απουσίες, πληρωμές, αποδείξεις, βιβλία βρίσκονται όλα αποθηκευμένα στον υπολογιστή, καλά οργανωμένα και τακτοποιημένα και βρίσκονται στην διάθεση της επιχείρησης οποιαδήποτε στιγμή.

Αξιόπιστος έλεγχος των δεδομένων της επιχείρησης :

Οποιαδήποτε στιγμή η επιχείρηση έχει στην διάθεση της πολύτιμα στοιχεία για την πορεία της επιχείρησης: Οφειλές μαθητών, προϋπολογισμός και κατανομή εξόδων , αναμενόμενα έσοδα κλπ.

Ευελιξία και παραμετρικότητα :

Το πρόγραμμα είναι ανοιχτής αρχιτεκτονικής και προσαρμόζεται ακριβώς στις απαιτήσεις και στις ανάγκες της επιχείρησης και έχει σχεδιαστεί για να καλύπτει κάθε πτυχή της. Είναι παραμετρικό και ευέλικτο, έτσι ώστε να ανταποκρίνεται όχι μόνο στις σημερινές αλλά και στις μελλοντικές ανάγκες της.

Και φυσικά το πρόγραμμα αυτό είναι απλούστατο στην χρήση του, δε χρειάζεται καμία ρύθμιση και μπορεί να εργαστεί μ' αυτό οποιοσδήποτε χωρίς κανένα πρόβλημα ακόμα και αν δεν έχει έρθει ποτέ σε επαφή με τους υπολογιστές.

3.2.1 ΦΟΡΜΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

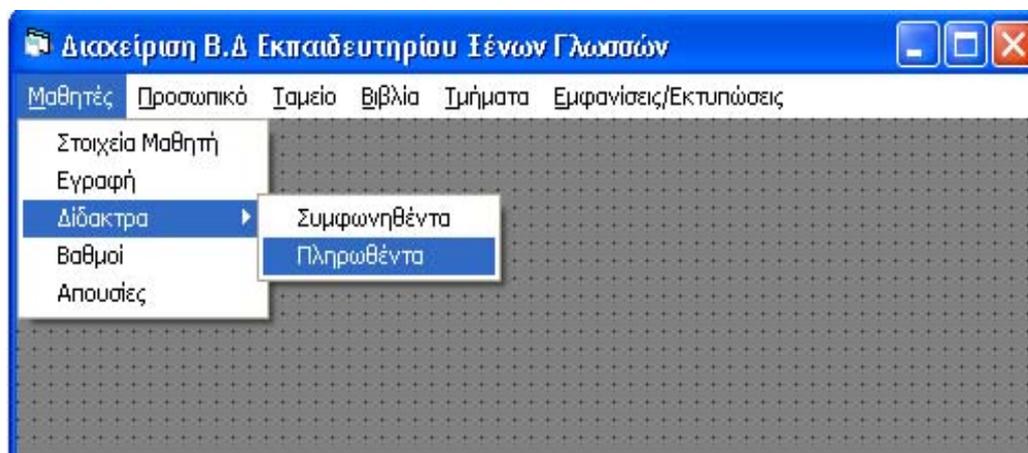
Στην Visual Basic η φόρμα είναι ο υποδοχέας όλων των ελεγκτηρίων που αποτελούν την διασύνδεση χρήστη. Όταν εκτελείτε μια εφαρμογή της Visual Basic, κάθε παράθυρο στην επιφάνεια εργασίας είναι μια φόρμα. Η φόρμα είναι το αντικείμενο ανωτέρου επιπέδου σε μια εφαρμογή της Visual Basic, και κάθε εφαρμογή εκκινεί με μια φόρμα.

Παρακάτω σ' αυτό το κεφάλαιο παρουσιάζονται αναλυτικά όλες οι φόρμες που δημιουργήθηκαν για την διαχείριση της βάσης δεδομένων του Εκπαιδευτικού Οργανισμού Ξένων Γλωσσών.

3.2.1.1 Η φόρμα MDIForm1

Η πρώτη (αρχική) φόρμα που δημιουργήθηκε είναι αυτή που περιέχει το μενού επιλογών, και είναι όσο πιο φιλική γίνεται για διευκόλυνση του χρήστη. Η φόρμα αυτή με όνομα **MDIForm1** φαίνεται στην ΕΙΚΟΝΑ3.1.

ΕΙΚΟΝΑ3.1 : Η φόρμα MDIForm1 που περιέχει το μενού (και τα υπό-μενού) επιλογών για την διαχείριση της Βάσης Δεδομενων



Στην δεύτερη γραμμή της ΕΙΚΟΝΑΣ3.1 μπορούμε να διακρίνουμε τις κύριες οντότητες που αφορούν την βάση δεδομένων του Εκπαιδευτικού Οργανισμού (Μαθητές, Προσωπικό, Ταμείο, Βιβλία, Τμήματα) καθώς επίσης και την επιλογή Εμφανίσεις/Εκτυπώσεις. Κάθε μια από αυτές τις επιλογές, της πάνω γραμμής, περιέχουν κάποιο υπομενού με επιλογές οι οποίες αντιστοιχούν στους πίνακες που έχουμε δημιουργήσει στην Microsoft Access. Στην ΕΙΚΟΝΑ3.1 φαίνεται το υπομενού της οντότητας Μαθητής: *Στοιχεία Μαθητή, Εγγραφή, Δίδακτρα, Βαθμοί, Απουσίες*. Η επιλογή Δίδακτρα περιέχει και αυτή το δικό της υπομενού με δυο επιλογές: *Συμφωνηθέντα* και *Πληρωθέντα*. Όλες οι επιλογές της πάνω γραμμής περιέχουν τα δικά τους υπομενού επιλογών (παρόλο που δεν φαίνονται στην ΕΙΚΟΝΑ3.1).

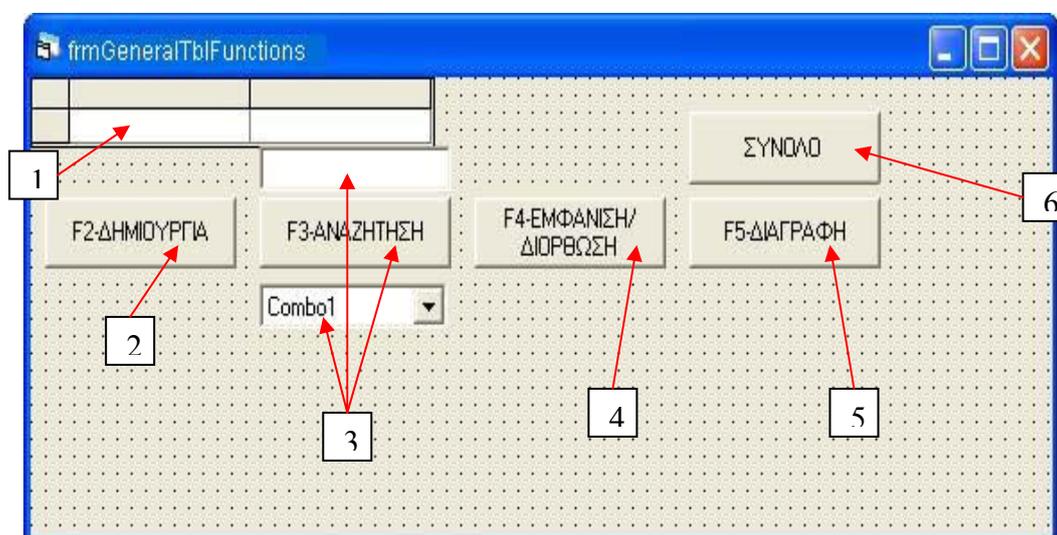
3.2.1.2 Η φόρμα frmGeneralTblFunctions

Όπως αναφέραμε και παραπάνω η κάθε επιλογή των υπό-μενού αντιστοιχεί σε έναν πίνακα της βάσης δεδομένων. Επιλέγοντας λοιπόν μια από αυτές εμφανίζεται στην οθόνη μια φόρμα με όλα τα αποθηκευμένα δεδομένα του αντίστοιχου πίνακα. Η φόρμα αυτή, **frmGeneralTblFunctions** (ΕΙΚΟΝΑ3.2), είναι κοινή για όλους τους πίνακες απλά κάθε φορά που καλείτε παίρνει της αντίστοιχες ρυθμίσεις για τον πίνακα που επιθυμούμε να επεξεργαστούμε. Για παράδειγμα αν από το αρχικό μενού επιλέξουμε την επιλογή *Βαθμοί* θα εκτελεστεί ο παρακάτω κώδικας :

```
Private Sub mnuBathmoi_Click()  
    If Module1.boolGnrIFrmLoaded Then Unload frmGeneralTblFunctions  
    Module1.strTableName = "ΒΑΘΜΟΙ"  
    frmGeneralTblFunctions.Caption = "ΒΑΘΜΟΙ"  
End Sub
```

Ο κώδικας αυτός αποδεσμεύει την φόρμα frmGeneralTblFunctions, αν χρησιμοποιούταν από κάποιον άλλο πίνακα ,και την επαναφέρει με ρυθμίσεις για τον πίνακα ΒΑΘΜΟΙ.

ΕΙΚΟΝΑ3.2 : Η φόρμα frmGeneralTblFunctions για εμφάνιση των εγγραφών της Βάσης Δεδομένων και οι επιλογές επεξεργασίας τους.



[1]. Το εργαλείο **DataGrid** στην ΕΙΚΟΝΑ3.2 προσαρμόζεται κάθε φορά στις ιδιότητες του πίνακα που χρησιμοποιεί την φόρμα. Δημιουργεί τόσες στήλες όσα και τα πεδία του πίνακα και εμφανίζονται σ' αυτό όλες οι αποθηκευμένες εγγραφές.

```
For Each strfield In RS.fields  
    DataGrid1.Columns(intColId).Caption = strfield.Name  
    frmTblInsert.Label1(0).Caption = strfield.Name
```

```
DataGrid1.Columns(intColId).Width = TextWidth(strfield.Name)
Next
```

[2]. Το εργαλείο *Command1* (**F2-ΔΗΜΙΟΥΡΓΙΑ**) ανοίγει μια νέα φόρμα (frmTblInsert), την οποία θα αναλύσουμε παρακάτω, και μ' αυτήν δημιουργούμε νέες εγγραφές στον πίνακα που επεξεργαζόμαστε.

Κώδικας Δημιουργίας/Καταχώρησης:

```
Dim sqlinsert As String
If strValues = "" Then strValues = "("
sqlinsert = "Insert INTO [" & strTblName & "] " & _
strFieldNames & " VALUES " & strValues
Con.Execute (sqlinsert)
```

[3]. Το εργαλείο *Command2* (**F3-ΑΝΑΖΗΤΗΣΗ**) συνδέεται με τα εργαλεία Combo1 και Textbox1. Κάνοντας click στο εργαλείο ΑΝΑΖΗΤΗΣΗ ανοίγει το Combo1 με μια λίστα των πεδίων του τρέχον πίνακα και από κάτω του το Textbox. Επιλέγοντας ένα από πεδία και πληκτρολογώντας στο Textbox το στοιχείο που αναζητούμε (ΕΙΚΟΝΑ3.3), γίνεται η αναζήτηση στην πίνακα της βάσης δεδομένων και τα αποτελέσματα εμφανίζονται στο DataGrid[1]. Η αναζήτηση ξεκινάει από τον πρώτο κίόλας χαρακτήρα που πληκτρολογούμε και επαναλαμβάνεται για κάθε νέο χαρακτήρα. Μαζί με την ανάλυση της ΕΙΚΟΝΑΣ3.3 παρουσιάζεται και ο κώδικας της Αναζήτησης.

[4]. Το εργαλείο *Command3* (**F4-ΕΜΦΑΝΙΣΗ/ΔΙΟΡΘΩΣΗ**) μας εμφανίζει τα αποθηκευμένα δεδομένα της εγγραφής, που έχουμε επιλέξει πριν στο DataGrid, με την φόρμα (frmTblInsert). Η ίδια φόρμα χρησιμοποιείται και στην ΔΗΜΙΟΥΡΓΙΑ νέας εγγραφής. Τα δεδομένα που μας εμφανίζονται μπορούμε να τα διορθώσουμε ή να προσθέσουμε κάποιο νέο που δεν είχε καταχωρηθεί. Το εργαλείο αυτό για τους πίνακες ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ, ΠΛΗΡΩΘΕΝΤΑ ΔΙΔΑΚΤΡΑ και ΛΟΓΑΡΙΑΣΜΟΙ μπορεί να χρησιμοποιηθεί μόνο για Εμφάνιση, επειδή σε περίπτωση αλλαγής/διόρθωσης κάποιου πεδίου να επηρεαστούν αρνητικά εγγραφές άλλων πινάκων της βάσης δεδομένων.

[5]. Το εργαλείο *Command4* (**F5-ΔΙΑΓΡΑΦΗ**) διαγράφει την εγγραφή που έχουμε επιλέξει από το DataGrid.

```
Dim sqlDelete As String
sqlDelete = "DELETE [" & strTblName & "].* FROM [" & strTblName _
& "]" WHERE " & strFieldNames
Con.Execute (sqlDelete)
```

[6]. Το εργαλείο *Command5* (**ΣΥΝΟΛΟ**) εμφανίζεται μόνο στους πίνακες ΑΠΟΥΣΙΕΣ (ΕΙΚΟΝΑ3.11) και ΕΣΟΔΑ_ΕΞΟΔΑ, και ενεργοποιεί ένα πλαίσιο μηνύματος. Το μήνυμα αυτό εμφανίζει το σύνολο απουσιών του μαθητή που επιθυμούμε ή το συνολικό ποσό που έχουμε πιστώσει/χρεώσει έναν λογαριασμό, ανάλογα με ποιον πίνακα επεξεργαζόμαστε. Ο κώδικας για το Command5 αναλύεται στην ενότητα 3.2.2 (TextBox[4])

ΕΙΚΟΝΑ3.3 : Η φόρμα frmGeneralTblFunctions για τον πίνακα ΒΑΘΜΟΙ.

	ΑΝΑΓΝΩΡΙΣΤΙΚΟ	ΚΩΔ. ΜΑΘΗΤΗ	ΤΡΙΜΗΝΟ	ΗΜΕΡΟΜΗΝΙΑ	ΔΡ. ΤΕΣΤ	ΒΑΘΜΟΣ
▶	1	1	A	19/9/2003	A-1	8
	2	1	A	29/9/2003	A-2	7
	3	1	A	15/10/2003	A-3	8
	4	1	A	30/10/2003	A-4	9
	5	1	A	12/11/2003	A-5	10
	6	1	B	3/12/2003	B-1	9
	7	1	B	18/12/2003	B-2	8
	8	1	B	9/1/2004	B-3	6

Στην ΕΙΚΟΝΑ3.3 βλέπουμε ότι ο χρήστης έχει κάνει αναζήτηση για να δει όλες τις εγγραφές του μαθητή με ΚΩΔ_ΜΑΘΗΤΗ (1). Τα αποτελέσματα εμφανίζονται στο DataGrid, όλες οι άλλες εγγραφές του πίνακα ΒΑΘΜΟΙ, εξαφανίζονται προσωρινά από την οθόνη.

Ο κώδικας αναζήτησης και εμφάνισης των αποτελεσμάτων έχει ως εξής:

```
Dim strSQLsearch As String
```

```
strSQLsearch = "SELECT * FROM [" & Module1.strTableName & "] WHERE ([" & Combo1.Text & "] LIKE '" & VBA.Trim(Text1.Text) & "%' OR [" & Combo1.Text & "] LIKE '%" & VBA.Trim(Text1.Text) & "%')"
```

```
Set RS = New ADODB.Recordset
```

```
With RS
```

```
.ActiveConnection = Con  
.CursorLocation = adUseClient  
.CursorType = adOpenForwardOnly  
.LockType = adLockReadOnly  
.Open strSQLsearch  
End With
```

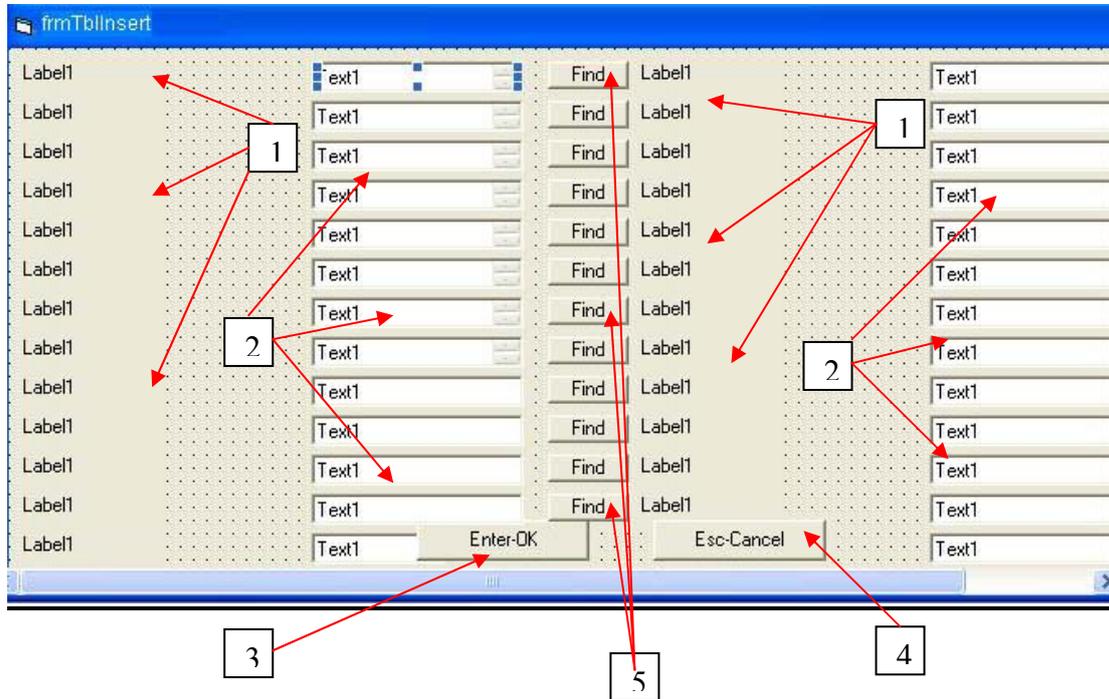
```
Set DataGrid1.DataSource = RS
```

```
DataGrid1.Refresh
```

3.2.1.3 Η φόρμα frmTblInsert

Η τρίτη αυτή φόρμα (**frmTblInsert**) που θα αναλύσουμε εμφανίζεται όπως προαναφέραμε α) για την δημιουργία νέας εγγραφής, β) την εμφάνιση των δεδομένων σε μορφή φόρμας και γ) για την διόρθωση δεδομένων. Η φόρμα αυτή κατά την σχεδίαση της έχει την μορφή που βλέπουμε στην ΕΙΚΟΝΑ3.4

EIKONA3.4 : Η φόρμα frmTblInsert κατά τον σχεδιασμό της



Όπως και η προηγούμενη φόρμα (frmGeneralTblFunctions) έτσι και αυτή εδώ προσαρμόζεται ανάλογα με το μέγεθος του τρέχον πίνακα, δηλαδή πόσα πεδία περιλαμβάνονται σ' αυτόν. Τα χωρίζει ισάριθμα αριστερά και δεξιά και δημιουργεί :

- [1]. Τα **Labels** με τα ονόματα των πεδίων του πίνακα
- [2]. Τα **Textbox** για εισαγωγή, εμφάνιση ή διόρθωση των δεδομένων

Στην φόρμα frmTblInsert περιλαμβάνονται επίσης ακόμα δυο εργαλεία :

- [3]. Το *Command1(0)* (**Enter-OK**) , με το πάτημα του οποίου γίνεται η αποθήκευση της εγγραφής στον πίνακα της βάσης δεδομένων.
- [4]. Το *Command1(1)* (**Esc-Cancel**), με το πάτημα του οποίου ακυρώνονται όλες οι ενέργειες (προσθήκη, διόρθωση) που έχουν προηγηθεί .

[5]. Το εργαλείο *Command2* (**F ind**) εμφανίζεται σε δύο περιπτώσεις :

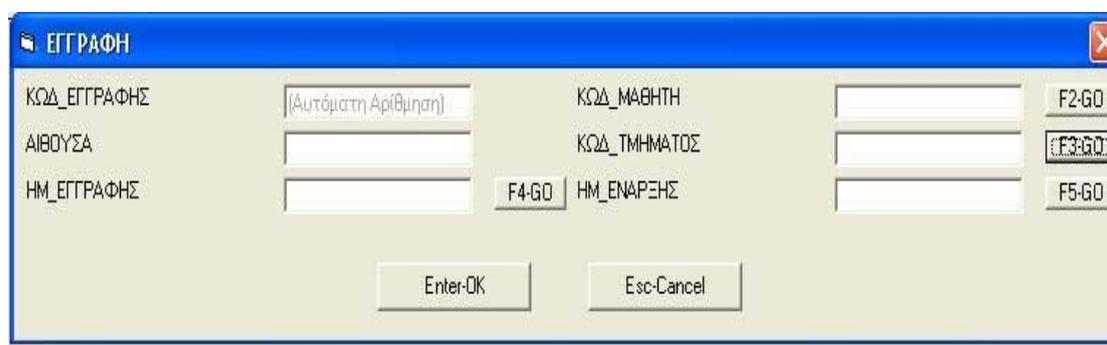
- a) Όταν ο τύπος δεδομένων του πεδίου αριστερά από το *Command2* είναι "Ημερομηνία". Τότε το *Command2* παραπέμπει στην φόρμα **form1**, η οποία περιλαμβάνει ένα ημερολόγιο από όπου μπορεί ο χρήστης να επιλέξει την ημερομηνία, αντί να την εισάγει από το πληκτρολόγιο.
- b) Όταν το πεδίο αριστερά από το *Command2* συνδέεται με κάποιο πεδίο άλλου πίνακα σε (σχέση Ένα-Προς-Πολλά). Τότε το *Command2* παραπέμπει σε μια άλλη φόρμα, την **frmGeneralTblFunctions1** από την οποία πρέπει να γίνει η επιλογή της εγγραφής που μας ενδιαφέρει για να πάρουμε το δεδομένο που θέλουμε.

```
RS2.Open "SELECT szReferencedObject,szReferencedColumn FROM MyRelationships WHERE szColumn='" & field.Name & "' AND szObject ='& Module1.strTableName & '"""
```

Παράδειγμα1: αν θέλουμε να δώσουμε τιμή στον ΚΩΔ_ΜΑΘΗΤΗ για τον πίνακα ΕΓΓΡΑΦΗ, το Command2 θα μας παραπέμψει στον πίνακα ΣΤΟΙΧΕΙΑ ΜΑΘΗΤΗ για να επιλέξουμε την εγγραφή με τον μαθητή που αναζητούμε και να πάρουμε την τιμή από το πεδίο ΚΩΔΙΚΟΣ.

Στην ΕΙΚΟΝΑ3.5 βλέπουμε την φόρμα frmTblInsert κατά την διάρκεια εκτέλεσης της εφαρμογής, θέλοντας να εισάγουμε δεδομένα στον πίνακα ΕΓΓΡΑΦΗ.

ΕΙΚΟΝΑ3.5 : Η φόρμα frmTblInsert για τον πίνακα ΕΓΓΡΑΦΗ



Στην επικεφαλίδα της φόρμας frmTblInsert έχει μπει το όνομα του πίνακα (ΕΓΓΡΑΦΗ) που επεξεργαζόμαστε. Τα έξι πεδία του πίνακα αυτού χωριστήκανε αριστερά και δεξιά ανά τρία.

Στο πεδίο ΚΩΔ_ΕΓΓΡΑΦΗΣ, πρωτεύον κλειδί του πίνακα, εμφανίζεται το μήνυμα (Αυτόματη Αρίθμηση), που είναι και ο τύπος του, και είναι μπλοκαρισμένο ώστε να μην μπορεί ο χρήστης να επέμβει.

Δεξιά στα πεδία ΚΩΔ_ΜΑΘΗΤΗ και ΚΩΔ_ΤΜΗΜΑΤΟΣ υπάρχουν δυο command *(F2-GO και F3-GO) αντίστοιχα. Αυτά όπως εξηγήσαμε και παραπάνω παραπέμπουν στις εγγραφές των πινάκων που σχετίζονται με τον πίνακα ΕΓΓΡΑΦΗ. Το F2-GO παραπέμπει στον πίνακα ΣΤΟΙΧΕΙΑ ΜΑΘΗΤΗ για να πάρουμε την τιμή για το πεδίο ΚΩΔ_ΜΑΘΗΤΗ. Το F3-GO παραπέμπει στον πίνακα ΤΜΗΜΑΤΑ για να πάρουμε τιμή για το πεδίο ΚΩΔ_ΤΜΗΜΑΤΟΣ.

Τα άλλα δύο command(F4-GO και F5-GO)δεξιά από τα πεδία ΗΜ_ΕΓΓΡΑΦΗΣ και ΗΜ_ΕΝΑΡΞΗΣ παραπέμπουν σε ένα ημερολόγιο για την επιλογή της ημερομηνίας.

Το πεδίο ΑΙΘΟΥΣΑ που δεν σχετίζεται με κανένα πίνακα και δεν είναι τύπου ημερομηνίας δεν έχει command δεξιά από το Textbox του. Τα στοιχεία για αυτό το πεδίο πρέπει να δοθούν από το πληκτρολόγιο.

Σε ένα δεύτερο παράδειγμα της φόρμας frmTblInsert για τον πίνακα Λογαριασμοί το πεδίο ΛΟΓΑΡΙΑΣΜΟΣ είναι κλειδωμένο. Η τιμή που θα πάρει αυτό το πεδίο

* Τα πεδία που έχουν δεξιά τους command (F*-GO) δεν παίρνουν τιμή από το πληκτρολόγιο. Τα Textbox που τους αντιστοιχούν είναι κλειδωμένα.

εξαρτάται από την τιμή του πεδίου ΑΙΤΙΟΛΟΓΙΑ1 και το πεδίο ΑΝΑΓΝΩΡΙΣΤΙΚΟ.
Παρακάτω αναλύεται ο κώδικας γι' αυτό το παράδειγμα:

Έλεγχος αν ο πίνακας που επεξεργαζόμαστε είναι ο πίνακας "ΛΟΓΑΡΙΑΣΜΟΙ"

```
If Module1.strTableName = "ΛΟΓΑΡΙΑΣΜΟΙ" Then
```

Μηδενισμός της μεταβλητής el

```
el = 0
```

Έλεγχος αν τα πεδία του πίνακα δεν έχουν πάρει τιμές (εκτός από το ΛΟΓΑΡΙΑΣΜΟΣ)

```
For i = 2 To ind
```

```
    If Text1(i).Text = "" Then el = 1
```

```
Next
```

Αν τα πεδία ΑΙΤΙΟΛΟΓΙΑ_1 και ΑΙΤΙΟΛΟΓΙΑ_2 έχουν πάρει τιμή

```
If ((Text1(2).Text <> "") And (Text1(3).Text <> "")) Then
```

Εισάγει μια εγγραφή στον πίνακα ΛΟΓΑΡΙΑΣΜΟΙ με τυχαίες τιμές

```
    Call Module1.Insert("ΛΟΓΑΡΙΑΣΜΟΙ", "[ΛΟΓΑΡΙΑΣΜΟΣ],[ΑΙΤΙΟΛΟΓΙΑ_1],  
[ΑΙΤΙΟΛΟΓΙΑ_2]", "('ww','ww','ww')", Con)
```

Παίρνει την τιμή από το ΑΝΑΓΝΩΡΙΣΤΙΚΟ της εγγραφής που μόλις δημιουργήθηκε

```
rs3.Open "SELECT [ΑΝΑΓΝΩΡΙΣΤΙΚΟ] FROM [ΛΟΓΑΡΙΑΣΜΟΣ]"
```

Πηγαίνει στο τέλος του αρχείου(πίνακα)

```
rs3.MoveLast
```

Εισάγει στην τρέχουσα (τελευταία) εγγραφή, στο πεδίο ΑΝΑΓΝΩΡΙΣΤΙΚΟ την τιμή που κράτησε από την προηγούμενη εγγραφή

```
Text1(0).Text = rs3(0).Value
```

Αν στο πεδίο ΑΙΤΙΟΛΟΓΙΑ_1 έχει δοθεί τιμή 1, το πεδίο ΛΟΓΑΡΙΑΣΜΟΣ παίρνει την τιμή 98.000.001. + την τιμή από το ΑΝΑΓΝΩΡΙΣΤΙΚΟ

```
If Text1(2).Text = "1" Then
```

```
    Text1(1).Text = "98.000.001." & rs3(0).Value
```

Αλλιώς αν στο πεδίο ΑΙΤΙΟΛΟΓΙΑ_1 έχει δοθεί τιμή 2, το πεδίο ΛΟΓΑΡΙΑΣΜΟΣ παίρνει την τιμή 98.000.002. + την τιμή από το ΑΝΑΓΝΩΡΙΣΤΙΚΟ

```
Else
```

```
    If Text1(2).Text = "2" Then
```

```
        Text1(1).Text = "98.000.002." & rs3(0).Value
```

Τέλος ελέγχων και κλείσιμο του recordset

```
End If
```

```
End If
```

```
rs3.Close
```

```
Else
```

```
    el = 1
```

```
End If
```

```
End If
```

Το πεδίο ΑΙΤΙΟΛΟΓΙΑ_1 παίρνει τιμή από τον πίνακα ΑΙΤΙΟΛΟΓΙΑ_1, ο οποίος περιέχει 2 μόνο εγγραφές. Μια εγγραφή με τιμή ΕΣΟΔΑ και μια με τιμή ΕΞΟΔΑ.

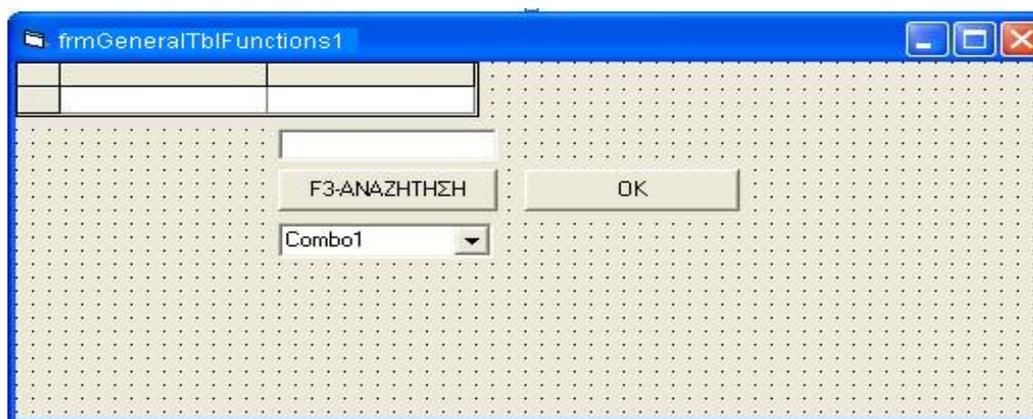
3.2.1.4 Η φόρμα frmGeneralTblFunctions1

Η φόρμα αυτή είναι επίσης κοινή για όλους τους πίνακες όπως και οι προηγούμενες δυο, frmGeneralTblFunctions και frmTblInsert. Κάθε φορά παίρνει τις ρυθμίσεις του πίνακα που την χρησιμοποιεί. Σ' αυτήν την φόρμα βλέπουμε όλες τις εγγραφές του τρέχον πίνακα και μπορούμε να αναζητήσουμε δυναμικά μια από αυτές, όπως και στην φόρμα frmGeneralTblFuncitns.

Η φόρμα αυτή εμφανίζεται αν εκτελέσουμε το Command2(περίπτωση **b**) της φόρμας frmTblInsert.

Επιλέγοντας κάποια από τις εγγραφές που εμφανίζονται στο DataGrid παίρνουμε την τιμή του πεδίου που έχει οριστεί πρωτεύων κλειδί και την επιστρέφουμε στο πεδίο του πίνακα που κάλεσε την φόρμα **frmGeneralTblFunctions1**. Εκτελεί, δηλαδή, την πράξη που εξηγήσαμε στο *Παράδειγμα1* για την φόρμα frmTblInsert για το εργαλείο Command2.

EIKONA3.6: Η φόρμα frmGeneralTblFunctions1



Όπως θα παρατηρήσατε και εσείς στην EIKONA3.6 η φόρμα frmGeneralTblFunctions1 είναι παρόμοια με την frmGeneralTblFunctions με την διαφορά ότι σ' αυτήν εδώ δεν υπάρχει επιλογή για να δημιουργήσουμε, να διορθώσουμε ή να διαγράψουμε μια εγγραφή.

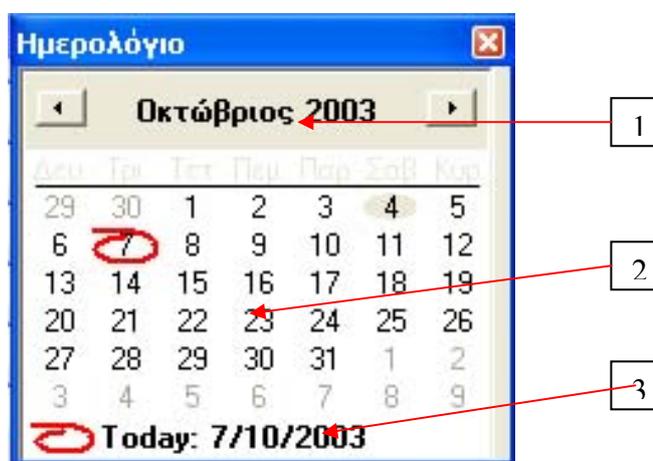
Το μόνο που μπορούμε να κάνουμε είναι αναζήτηση ή επιλογή μιας εγγραφής, και με το εργαλείο *Command* (**OK**) να επιστρέψουμε στην φόρμα frmTblInsert, στον πίνακα που την χρησιμοποιεί, και να γίνουν οι απαραίτητες ρυθμίσεις στην τιμή του πεδίου που κάλεσε την frmGeneralTblFunctions1.

3.2.1.5 Η φόρμα form1(Ημερολόγιο)

Η φόρμα form1 είναι ένα ημερολόγιο για την επιλογή της επιθυμητής ημερομηνίας, από τον χρήστη, ώστε να δοθεί τιμή στα πεδία με τύπο "Ημερομηνία". Όπως φαίνεται στην ΕΙΚΟΝΑ3.7 το ημερολόγιο αυτό περιλαμβάνει:

1. Στο πάνω μέρος του μία γραμμή εμφάνισης και επιλογής του μήνα και της χρονολογίας (π.χ **Οκτώβριος 2003**)
2. Στο κέντρο τους αριθμούς όλων των ημερών του μήνα που έχει επιλεγεί (π.χ **1-30** ή **1-31** κλπ)
3. Στο κάτω μέρος ένα μήνυμα με τη σημερινή (τρέχουσα) ημερομηνία (π.χ **Today: 7/10/2003**)

ΕΙΚΟΝΑ3.7 : Η φόρμα form1 (Ημερολόγιο)



Κάνοντας click σε κάποιον αριθμό της φόρμας form1 (π.χ στο 19), η φόρμα κλείνει, και το πεδίο που την κάλεσε από την φόρμα frmTblInsert παίρνει την τιμή 19/8/2003. Αυτό σαφώς σημαίνει ότι πρώτα επιλέγουμε το μήνα και την χρονολογία και στο τέλος την ημέρα.

3.2.2 ΠΛΑΙΣΙΑ ΜΗΝΥΜΑΤΩΝ (MsgBox)

Το πρόγραμμα αυτό απευθύνεται σε Κέντρα Ξένων Γλωσσών και όχι σε ειδικούς της Πληροφορικής. Η διευκόλυνση του χρήστη αποτέλεσε βασικό μέλημα κατά την κατασκευή του προγράμματος. Γι' αυτό το λόγο έπρεπε να δημιουργηθούν κάποια MsgBox σαν προειδοποιητικά/πληροφοριακά μηνύματα ή μηνύματα λάθους ώστε να κατευθύνουν τον χρήστη. Παρακάτω παρουσιάζονται αναλυτικά όλα τα MsgBox:

1. Το πρώτο MsgBox εμφανίζεται αν προσπαθήσουμε να καταχωρήσουμε δεδομένα (π.χ. ΕΙΚΟΝΑ3.5) σε κάποιο πίνακα και έστω και ένα πεδίο δεν έχει πάρει τιμή. Ο κώδικάς του έχει ως εξής :

Γίνεται έλεγχος αν κάποιο πεδίο έχει τιμή "", δηλαδή κενό(Null)

```
el = 0
For i = 1 To ind
    If Text1(i) = "" Then el = 1
Next
```

Αν το παραπάνω ισχύει τότε εμφανίζεται το MsgBox το οποίο παρουσιάζεται στην ΕΙΚΟΝΑ3.8

```
If el = 1 Then MsgBox " Ελλιπή Στοιχεία" , "ΜΗΝΥΜΑ"
```

ΕΙΚΟΝΑ3.8 : Μήνυμα στον χρήστη ότι κάποιο πεδίο δεν έχει πάρει τιμή



2. Το δεύτερο MsgBox εμφανίζεται αν έχει δοθεί λάθος τιμή στα πεδία *ΣΥΝΟΛΙΚΟ ΠΟΣΟ* ή *ΑΡ_ΔΟΣΕΩΝ* του πίνακα **ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ**(ΕΙΚΟΝΑ3.9) και αποτρέπει την καταχώριση της εγγραφής.

ΕΙΚΟΝΑ3.9 : Καταχώριση δεδομένων στον πίνακα ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ

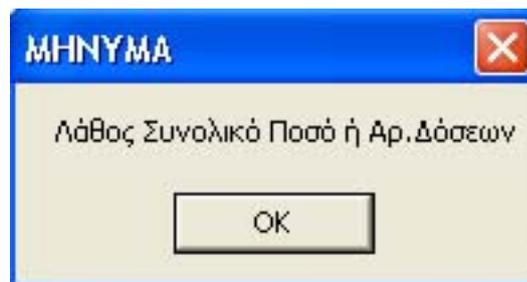
Η τιμή 90 στο πεδίο AP_ΔΟΣΕΩΝ της ΕΙΚΟΝΑΣ3.9 είναι λάθος. Η ανώτερη τιμή για τον AP_ΔΟΣΕΩΝ είναι 18 και για το ΣΥΝΟΛΙΚΟ ΠΟΣΟ 3000€, όπως φαίνεται και στον παρακάτω κώδικα :

```
If (Index = 3 Or Index = 4) Then
  If Module1.strTableName = "ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ" Then
    If (IsNumeric(Text1(3).Text) And IsNumeric(Text1(4).Text)) Then
      If (Text1(3).Text > 0 And Text1(3).Text < 3000) And
        (Text1(4).Text < 19 And Text1(4).Text > 0) Then
        Text1(6).Text = CStr(CInt(Text1(3).Text) /
CInt(Text1(4).Text))
      Else
        Text1(6).Text = ""
      End If
    End If
  End If
End If
```

Το πεδίο ΠΟΣΟ ΔΟΣΗΣ παίρνει τιμή μόνο όταν τα οι τιμές των άλλων δύο είναι σωστές. Η τιμή αυτή ισοδυναμεί με την τιμή που προκύπτει από την πράξη ΣΥΝΟΛΙΚΟ ΠΟΣΟ / AP_ΔΟΣΕΩΝ (Υπογραμμισμένη γραμμή του κώδικα).

Αν προσπαθήσουμε να καταχωρήσουμε τα δεδομένα στον πίνακα ,για το παράδειγμα τις ΕΙΚΟΝΑΣ3.9, πατώντας το command (Enter-OK) θα μας εμφανιστεί το παρακάτω MsgBox (ΕΙΚΟΝΑ3.10).

ΕΙΚΟΝΑ3.10 : Μήνυμα στον χρήστη για λάθος εισαγωγή δεδομένων στα πεδία ΣΥΝΟΛΙΚΟ ΠΟΣΟ και AP_ΔΟΣΕΩΝ

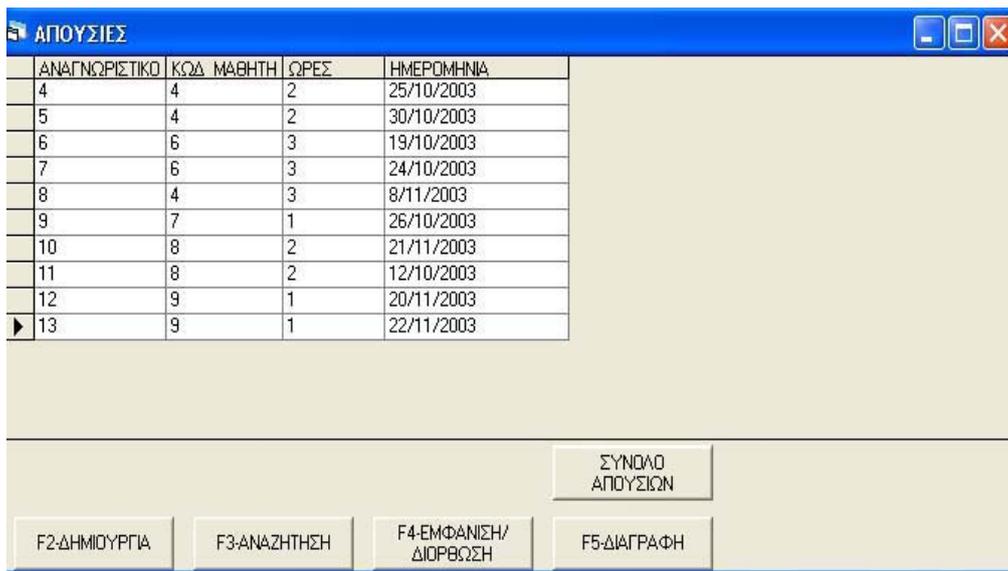


Ο παρακάτω κώδικας ελέγχει αν το πεδίο ΠΟΣΟ ΔΟΣΗΣ του πίνακα ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ δεν έχει πάρει τιμή. Αν ισχύει σημαίνει ότι έχει γίνει κάποιο λάθος και εμφανίζεται το μήνυμα της ΕΙΚΟΝΑΣ3.10.

```
If Module1.strTableName = "ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ" Then
  If Text1(6) = "" Then
    MsgBox "Λάθος Συνολικό Ποσό ή Αρ.Δόσεων", , "ΜΗΝΥΜΑ"
    el = 2
  End If
End If
```

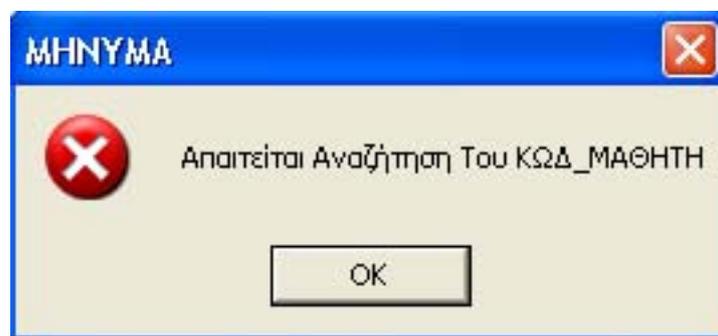
3. Το τρίτο MsgBox είναι ενημερωτικό ως προς τις ενέργειες που πρέπει να κάνει ο χρήστης για να δει: (1) το Σύνολο Απουσιών ενός Μαθητή όταν βρίσκεται στην φόρμα frmGeneralTblFunctions για τον πίνακα ΑΠΟΥΣΙΕΣ (ΕΙΚΟΝΑ3.11) ή (2) το Συνολικό Ποσό ενός Λογαριασμού για τον πίνακα ΕΣΟΔΑ_ΕΞΟΔΑ.

ΕΙΚΟΝΑ3.11 : Η φόρμα frmGeneralTblFunctions για τον πίνακα ΑΠΟΥΣΙΕΣ



Σε περίπτωση που ο χρήστης ενεργοποιήσει την επιλογή ΣΥΝΟΛΟ ΑΠΟΥΣΙΩΝ χωρίς να έχει αναζητήσει προηγουμένως τις εγγραφές του πίνακα ΑΠΟΥΣΙΕΣ για έναν συγκεκριμένο μαθητή θα εμφανιστεί το MsgBox που βλέπετε στην ΕΙΚΟΝΑ3.12.

ΕΙΚΟΝΑ3.12 : Μήνυμα που καθοδηγεί τον χρήστη για το πώς να μάθει το Σύνολο Απουσιών ενός μαθητή



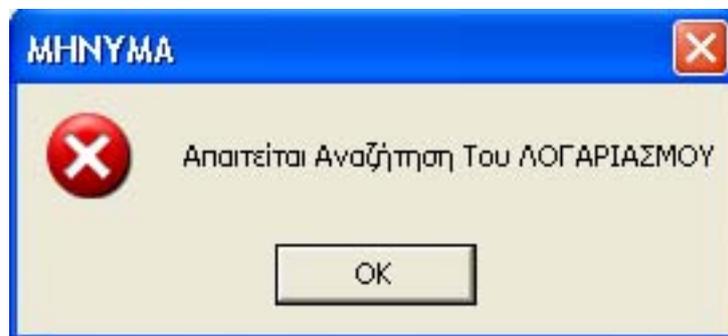
Όπως καταλαβαίνεται πρέπει να γίνει αναζήτηση ενός μαθητή με βάση το πεδίο ΚΩΔ_ΜΑΘΗΤΗ. Μετά την αναζήτηση αυτή, αν υπάρχουν εγγραφές

στον πίνακα για τον συγκεκριμένο μαθητή, εμφανίζονται μόνο αυτές στο DataGridView της φόρμας frmGeneralTblFunctions. Οπότε μπορούμε να δούμε τις ημερομηνίες και τις ώρες που έχει απουσιάσει αυτός ο μαθητής. Αν θέλει ο χρήστης να μάθει και το σύνολο των απουσιών αρκεί να επιλέξει το command (ΣΥΝΟΛΟ ΑΠΟΥΣΙΩΝ), το οποίο θα εμφανίσει ένα νέο MsgBox που θα αναλύσουμε παρακάτω.

Σε περίπτωση που γίνει αναζήτηση με βάση κάποιο άλλο πεδίο του πίνακα ΑΠΟΥΣΙΕΣ, τα αποτελέσματα που θα εμφανιστούν δεν θα είναι σωστά.

Ομοίως και για τον πίνακα ΕΣΟΔΑ_ΕΞΟΔΑ το αντίστοιχο μήνυμα φαίνεται στην ΕΙΚΟΝΑ3.13. Στην περίπτωση του πίνακα αυτού πρέπει να γίνει αναζήτηση στο πεδίο ΛΟΓΑΡΙΑΣΜΟΣ.

ΕΙΚΟΝΑ 3.13 : Μήνυμα που καθοδηγεί τον χρήστη για το πώς να μάθει το Συνολικό Ποσό ενός λογαριασμού



4. Το τέταρτο MsgBox παρέχει μια πληροφορία στον χρήστη, το Σύνολο Απουσιών ενός μαθητή ή το Συνολικό Ποσό ενός λογαριασμού. Είναι το MsgBox που έπεται μετά από τις ενέργειες που εξηγήσαμε παραπάνω στο προηγούμενο MsgBox.

ΕΙΚΟΝΑ3.14 : Μήνυμα που εμφανίζει το Σύνολο Απουσιών ενός μαθητή



Ο παρακάτω κώδικας παίρνει το άθροισμα των απουσιών (πεδίο ΩΡΕΣ) ενός μαθητή και το εμφανίζει με το MsgBox της ΕΙΚΟΝΑΣ3.14. Αν δεν έχει γίνει καμία αναζήτηση, δηλαδή `Combo1.text = ""`, τότε εμφανίζεται το προηγούμενο MsgBox (ΕΙΚΟΝΑ3.12).

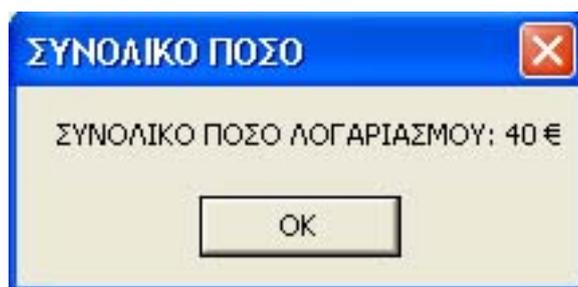
```

If Combo1.Text <> "" Then
    rs3.Open "SELECT SUM([ΩΡΕΣ]) FROM [ΑΠΟΥΣΙΕΣ] WHERE ([ " &
    Combo1.Text & "] LIKE " & VBA.Trim(Text1.Text) & "%' OR [ " &
    Combo1.Text & "] LIKE %" & VBA.Trim(Text1.Text) & "%)"
    MsgBox "ΣΥΝΟΛΟ ΑΠΟΥΣΙΩΝ ΜΑΘΗΤΗ: " & rs3(0).Value, ,
    "ΣΥΝΟΛΟ ΑΠΟΥΣΙΩΝ ΜΑΘΗΤΗ"
    rs3.Close
Else
    MsgBox "Απαιτείται Αναζήτηση Του ΚΩΔ_ΜΑΘΗΤΗ", vbCritical,
    "ΜΗΝΥΜΑ"
End If

```

Το αντίστοιχο MsgBox για το Συνολικό Πόσο ενός λογαριασμού βλέπουμε στην ΕΙΚΟΝΑ3.15. Τα ίδια πράγματα που αναφέρθηκαν για το MsgBox της ΕΙΚΟΝΑΣ3.14 ισχύουν και σ' αυτήν την περίπτωση.

ΕΙΚΟΝΑ 3.15 : Μήνυμα που εμφανίζει το Συνολικό Ποσό που περιέχει ένας λογαριασμός



5. Το πέμπτο MsgBox (ΕΙΚΟΝΑ3.16) εμφανίζεται σε μια τρίτη περίπτωση επιλογής του command (ΣΥΝΟΛΟ ΑΠΟΥΣΙΩΝ) της φόρμας frmGeneralTblFunctions για τον πίνακα ΑΠΟΥΣΙΕΣ ή του command (ΣΥΝΟΛΙΚΟ ΠΟΣΟ) για τον πίνακα ΕΣΟΔΑ_ΕΞΟΔΑ, όταν ο πίνακας αυτός δεν έχει καμία αποθηκευμένη εγγραφή. Σαφώς η επιλογή αυτή μπορεί να συμβεί από λάθος χειρισμό του χρήστη.

```

view_error:
MsgBox "Δεν υπάρχουν εγγραφές", vbCritical, "ΜΗΝΥΜΑ"

```

ΕΙΚΟΝΑ3.16: Μήνυμα στον χρήστη ότι δεν υπάρχουν εγγραφές στον πίνακα ΑΠΟΥΣΙΕΣ



6. Το πεδίο ΥΠΟΛΟΙΠΟ του πίνακα ΠΛΗΡΩΘΕΝΤΑ ΔΙΔΑΚΤΡΑ (ΕΙΚΟΝΑ3.17) είναι κλειδωμένο και η τιμή που παίρνει εξαρτάται από το πεδίο ΠΟΣΟ. Την στιγμή που πληκτρολογούμε την τιμή στο πεδίο ΠΟΣΟ ο κώδικας κάνει αναζήτηση στο πίνακα ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ για να βρει το ΣΥΝΟΛΙΚΟ ΠΟΣΟ του μαθητή με τον τρέχον ΚΩΔ_ΜΑΘΗΤΗ. Όταν τον βρει αφαιρεί από αυτόν την τιμή του πεδίου ΠΟΣΟ και την εισάγει στο πεδίο ΥΠΟΛΟΙΠΟ.

Αυτό βέβαια συμβαίνει την πρώτη φορά που εισάγουμε δεδομένα στον πίνακα ΠΛΗΡΩΘΕΝΤΑ ΔΙΔΑΚΤΡΑ για κάθε μαθητή. Από την δεύτερη φορά και μετά αφαιρείτε η τιμή του πεδίου ΠΟΣΟ από το πεδίο ΥΠΟΛΟΙΠΟ της τελευταίας εγγραφής του μαθητή. Στον κώδικα που ακολουθεί βλέπουμε όλες τις ενέργειες που εξηγήσαμε γι' αυτήν την περίπτωση.

```

If Module1.strTableName = "ΠΛΗΡΩΘΕΝΤΑ ΔΙΔΑΚΤΡΑ" Then
On Error GoTo view_error
    If ((IsNumeric(Text1(4).Text)) And (Text1(1) <> "")) Then
        rs3.Open "SELECT [ΣΥΝΟΛΙΚΟ ΠΟΣΟ] FROM [ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ] WHERE ([ΚΩΔ_ΜΑΘΗΤΗ] = " & Text1(1).Text & ")"
        Text1(5).Text = rs3(0).Value
        rs3.Close
        rs3.Open "SELECT SUM([ΠΟΣΟ]) FROM [ΠΛΗΡΩΘΕΝΤΑ ΔΙΔΑΚΤΡΑ] WHERE ([ΚΩΔ_ΜΑΘΗΤΗ] = " & Text1(1).Text & ")"

        If rs3(0).Value <> 0 Then
            Text1(5).Text = CStr(CInt(Text1(5).Text) - CInt(rs3(0).Value + Text1(4).Text))
        Else
            Text1(5).Text = CStr(CInt(Text1(5).Text) - CInt(Text1(4).Text))
        End If
        rs3.Close
    End If
End If

Exit Sub
view_error:
MsgBox "Δεν Έχουν Καταχωρηθεί τα Συμφωνηθέντα Δίδακτρα", vbCritical, "ΜΗΝΥΜΑ"
Call Form_Load

```

Στις τελευταίες γραμμές του παραπάνω κώδικα καταλαβαίνουμε ότι σε περίπτωση λάθους θα μας εμφανιστεί ένα MsgBox (ΕΙΚΟΝΑ3.18).

ΕΙΚΟΝΑ 3.17: Η φόρμα για την εισαγωγή δεδομένων στον πίνακα ΠΛΗΡΩΘΕΝΤΑ ΔΙΔΑΚΤΡΑ

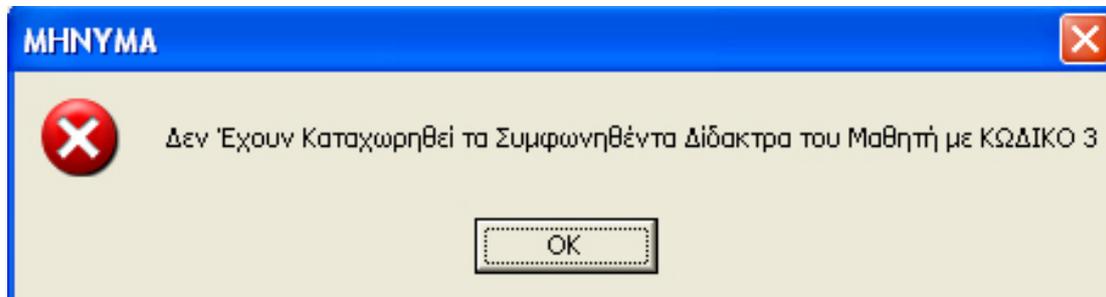
ΠΛΗΡΩΘΕΝΤΑ ΔΙΔΑΚΤΡΑ

ΑΝΑΓΝΩΡΙΣΤΙΚΟ	(Αυτόματη Αρίθμηση)	ΚΩΔ_ΜΑΘΗΤΗ		F2-GO
ΑΡ_ΑΠΟΔΕΙΞΗ		ΗΜ_ΠΛΗΡΩΜΗΣ		F3-GO
ΠΟΣΟ		ΥΠΟΛΟΙΠΟ		

Enter-OK Esc-Cancel

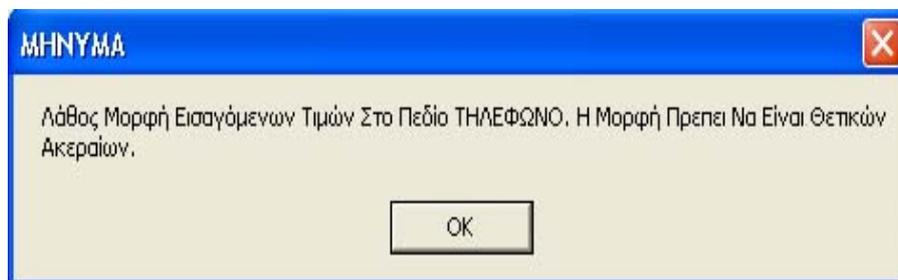
Το έκτο MsgBox εμφανίζεται όταν προσπαθήσουμε να εισάγουμε δεδομένα στον πίνακα ΠΛΗΡΩΘΕΝΤΑ ΔΙΔΑΚΤΡΑ για κάποιο μαθητή, ενώ δεν έχουμε καταχωρήσει στην βάση δεδομένων τα Συμφωνηθέντα Δίδακτρα αυτού.

ΕΙΚΟΝΑ3.18: Μήνυμα στον χρήστη ότι δεν υπάρχουν εγγραφές στον πίνακα ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ, για συγκεκριμένο μαθητή.



7. Το έβδομο MsgBox (ΕΙΚΟΝΑ3.19) εμφανίζεται σε περίπτωση που ο χρήστης δώσει λάθος δεδομένα για εισαγωγή σε πεδία με τύπο Θετικών Ακεραίων (μικρού ή μεγάλου μήκους) ή Νομισματική Μονάδας. Το MsgBox αυτό προειδοποιεί ότι τα δεδομένα, που έχουν δοθεί για εισαγωγή στον πίνακα, έχουν λανθασμένη μορφή και ενημερώνει τον χρήστη ποια πρέπει να είναι η σωστή μορφή.

ΕΙΚΟΝΑ3.19 : Μήνυμα που προειδοποιεί και ενημερώνει τον χρήστη ποια μορφή πρέπει να έχουν τα δεδομένα που θα εισάγει σε συγκεκριμένο πεδίο



Το πεδίο ΤΗΛΕΦΩΝΟ που βλέπουμε στην ΕΙΚΟΝΑ3.19 της είναι ένα τυχαίο πεδίο που επιλέχθηκε για να εμφανίσει του MsgBox. Στον κώδικα που ακολουθεί βλέπουμε ότι η τιμή που εμφανίζεται στο όνομα του πεδίου (πχ ΤΗΛΕΦΩΝΟ) είναι μεταβλητή. Αυτό σημαίνει ότι στο σημείο που τώρα βλέπουμε τη λέξη ΤΗΛΕΦΩΝΟ θα εμφανίζεται το όνομα του πεδίου που θα έχει δεχθεί λάθος δεδομένα, την τρέχον στιγμή, με το ανάλογο μήνυμα της μορφής δεδομένων που πρέπει να πάρει (π.χ. ΕΙΚΟΝΑ3.20).

```
Dim field As ADODB.field
For Each field In fields
On Error Resume Next
If txtBoxes(ind).Text <> "" Then
If field.Name <> Module1.colPrimaryKeys(Module1.strTableName) Then
If Err <> 0 Then Err.Clear
On Error GoTo convert
Select Case fields(ind).Type
Case ADODB.adCurrency
VBA.CCur (txtBoxes(ind).Text)
txtBoxes(ind).Text = VBA.CStr(VBA.CCur(txtBoxes(ind).Text))
Case ADODB.adInteger
VBA.CInt (txtBoxes(ind).Text)
txtBoxes(ind).Text = VBA.CStr(VBA.CInt(txtBoxes(ind).Text))
Case ADODB.adDouble
VBA.CDbl (txtBoxes(ind).Text)
txtBoxes(ind).Text = VBA.CStr(VBA.CDbl(txtBoxes(ind).Text))
Case Else
End Select
End If
End If
ind = ind + 1
Next
validateExpressions = -1
Exit Function
convert:
Dim strPrompt As String
strPrompt = "Λάθος Μορφή Εισαγόμενων Τιμών Στο Πεδίο " & fields(ind).Name

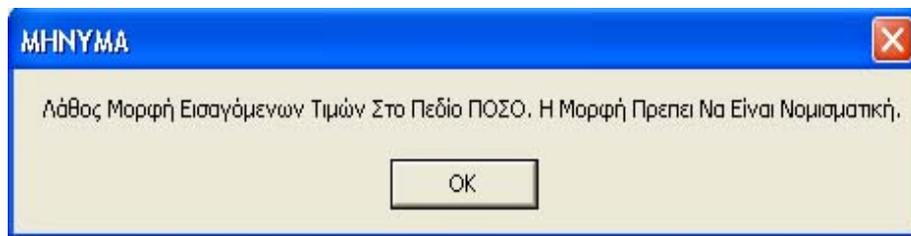
Select Case fields(ind).Type
Case ADODB.adCurrency
```

```

        strPrompt = strPrompt & ". Η Μορφή Πρέπει Να Είναι Νομισματική."
    Case ADODB.adInteger
        strPrompt = strPrompt & ". Η Μορφή Πρέπει Να Είναι Θετικών
Ακεραίων."
    Case ADODB.adDouble
        strPrompt = strPrompt & ". Η Μορφή Πρέπει Να Είναι Θετικών
Ακεραίων."
    Case Else
    End Select
    MsgBox strPrompt, , "ΜΗΝΥΜΑ"
    txtBoxes(ind).SetFocus
    txtBoxes(ind).SelStart = 0
    txtBoxes(ind).SelLength = 1000
    validateExpressions = ind

```

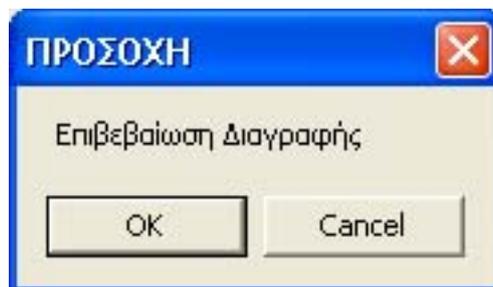
ΕΙΚΟΝΑ3.20 : Δεύτερη περίπτωση μορφής του έβδομου MsgBox



8. Το όγδοο και τελευταίο MsgBox της εφαρμογής εμφανίζεται όταν θέλουμε να διαγράψουμε μια εγγραφή από τον τρέχον πίνακα της φόρμας frmGeneralTblFunctions (π.χ. ΕΙΚΟΝΑ3.3). Επιλέγοντας μια εγγραφή και πατώντας το command (ΔΙΑΓΡΑΦΗ) εμφανίζεται το MsgBox(ΕΙΚΟΝΑ3.21) αυτό το οποίο ζητάει από τον χρήστη επιβεβαίωση διαγραφής. Αυτό χρειάζεται για την πιθανότητα που ο χρήστης έχει κάνει λάθος επιλογή εγγραφής ή command.

MsgBox("Επιβεβαίωση Διαγραφής", vbOKCancel, "ΠΡΟΣΟΧΗ")

ΕΙΚΟΝΑ3.21 : Προειδοποίηση του χρήστη πριν από την οριστική διαγραφή μιας εγγραφής.



Αν ο χρήστης επιλέξει το Command (OK) η επιλεγμένη εγγραφή διαγράφεται. Οποιαδήποτε άλλη ενέργεια ακυρώνει την διαγραφή της εγγραφής.

ΕΚΘΕΣΕΙΣ – ΑΝΑΦΟΡΕΣ

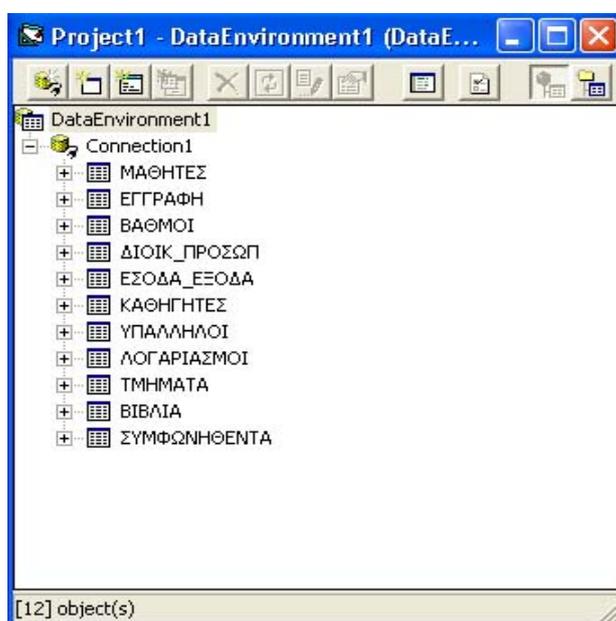
Μια άλλη εξ ίσου σημαντική λειτουργία στην ανάπτυξη εφαρμογών βάσεων δεδομένων είναι η παραγωγή εκθέσεων. Οι εκθέσεις είναι ένας τύπος αντικειμένων τον οποίο μπορούμε να χρησιμοποιούμε για την εξέταση ή εκτύπωση των δεδομένων μας. Αυτές οι εκθέσεις βοηθούν στην ομαλή λειτουργία του Εκπαιδευτικού Οργανισμού και ορισμένες φορές βοηθούν στο να παρθούν αποφάσεις από το διοικητικό προσωπικό.

Παρακάτω παρουσιάζονται οι εκθέσεις που δημιουργήθηκαν σύμφωνα πάντα με τις απαιτήσεις του Εκπαιδευτικού Οργανισμού Ξένων Γλωσσών.

3.2.3.1 Σχεδίαση με τον DataEnvironment ActiveX Designer

Το συστατικό DataEnvironment μας επιτρέπει να σχεδιάζουμε μια σύνδεση προς την βάση δεδομένων και να επαναφέρουμε τις επιθυμητές εγγραφές. Στην ΕΙΚΟΝΑ3.22 βλέπουμε το DataEnvironment που δημιουργήσαμε για την εφαρμογή μας(Project1).

ΕΙΚΟΝΑ3.22 : Το παράθυρο DataEnvironment για σύνδεση με τη βάση δεδομένων και δημιουργία εκθέσεων.



Για να προσπελάσουμε την βάση δεδομένων χρειαζόμαστε δύο τύπους αντικειμένων :

1. Ένα αντικείμενο Connection

Το αντικείμενο Connection1(ΕΙΚΟΝΑ3.22) συνδέει την εφαρμογή μας με την βάση δεδομένων του Εκπαιδευτικού Οργανισμού (ΕΚΡΑΙΔΕΥΤΗΡΙΟ.mdb)

2. Ένα ή περισσότερα αντικείμενα Command

Το αντικείμενο Command επαναφέρει εγγραφές από την βάση δεδομένων. Για τις ανάγκες του προγράμματος μας δημιουργήσαμε 11 Commands με ονόματα (ΜΑΘΗΤΕΣ, ΕΓΓΡΑΦΕΣ, ΒΑΘΜΟΙ, ΔΙΟΙΚ_ΠΡΟΣΩΠ, ΕΣΟΔΑ_ΕΞΟΔΑ, ΚΑΘΗΓΗΤΕΣ, ΥΠΑΛΛΗΛΟΙ, ΛΟΓΑΡΙΑΣΜΟΙ, ΤΜΗΜΑΤΑ, ΒΙΒΛΙΑ). Τα Commands αυτά αντιστοιχούν στους πίνακες της βάσης δεδομένων και με βάση αυτά θα δημιουργήσουμε τις εκθέσεις για το Project1.

3.2.3.2 Δημιουργία Εκθέσεων με το DataReport ActiveX Designer

Το συστατικό DataReport μας επιτρέπει να σχεδιάσουμε εκθέσεις και να τις χρησιμοποιούμε μέσα στο πρόγραμμα μας. Τόσο το DataReport όσο και το DataEnvironment βασίζονται σε οπτικά εργαλεία βάσεων δεδομένων, και δεν απαιτούν προγραμματισμό.

Οι εκθέσεις που δημιουργήσαμε για την εφαρμογή Project1 μας παρουσιάζουν αναφορές για

(α) τα Στοιχεία των Μαθητών, τις Εγγραφές και τους Βαθμούς τους, τα Συμφωνηθέντα και Πληρωθέντα Δίδακτρα.

(β) τα Στοιχεία των Υπαλλήλων, των Καθηγητών και του Διοικητικού Προσωπικού.

(γ) τους Λογαριασμούς και τα Έσοδα/Έξοδα του Οργανισμού

(δ) τα Βιβλία και

(ε) τα Τμήματα

Στην ΕΙΚΟΝΑ3.23 φαίνεται η έκθεση για τα δεδομένα του πίνακα ΛΟΓΑΡΙΑΣΜΟΙ.

ΕΚΟΝΑ3.23 : Σχεδίαση μιας έκθεσης με τον DataReport Designer



Στον χώρο κάτω από την γραμμή Page Header περιλαμβάνονται οι τίτλοι (επικεφαλίδες) για κάθε νέα σελίδα που δημιουργείται. Οτι υπάρχει κάτω από αυτή τη γραμμή εμφανίζεται όπως έχει, κατά την εκτέλεση της εφαρμογής.

Στην πιο κάτω γραμμή Detail βλέπουμε ξανά τα ονόματα των πεδίων του πίνακα ΛΟΓΑΡΙΑΣΜΟΙ. Σ' αυτή την περίπτωση όμως όταν εκτελείται το πρόγραμμα εμφανίζονται οι τιμές των πεδίων αυτών και όχι τα ονόματα τους.

Τέλος, κάτω από την γραμμή Page Footer τοποθετήσαμε μία λεζάντα που εμφανίζει τον αριθμό της τρέχουσας σελίδας και το ανάλογο μήνυμα (ΣΕΛ.) δίπλα από αυτή.

Επιλέγοντας από την εφαρμογή Project1 να δούμε την έκθεση του πίνακα ΛΟΓΑΡΙΑΣΜΟΙ εμφανίζεται στην οθόνη μας το παράθυρο της ΕΙΚΟΝΑΣ3.24.

ΕΙΚΟΝΑ3.24 : Αναφορά όλων των Λογαριασμών της βάσης δεδομένων ΕΚΡΑΙΔΕΥΤΗΡΙΟ.mdb

ΛΟΓΑΡΙΑΣΜΟΣ	ΑΙΤΙΟΛΟΓΙΑ_1	ΑΙΤΙΟΛΟΓΙΑ_2
98.000.001.1	ΕΣΟΔΑ	ΕΙΣΠΡΑΞΕΙΣ
98.000.001.2	ΕΣΟΔΑ	ΕΓΓΡΑΦΕΣ
98.000.002.3	ΕΞΟΔΑ	HARDWARE
98.000.002.4	ΕΞΟΔΑ	ΒΙΒΛΙΑ
98.000.002.5	ΕΞΟΔΑ	ΕΠΙΣΚΕΥΕΣ

Τα δυο κουμπιά στην κορυφή του παραθύρου DataReport9 μας επιτρέπουν να εκτυπώνουμε την αναφορά σε ένα εκτυπωτή ή να την εξάγουμε (export) σε μορφή κειμένου ή HTML. Λίγο πιο δίπλα από τα δυο κουμπιά υπάρχει ένα ComboBox με επιλογές για το ποσοστό του μεγέθους της σελίδας που θέλουμε να προβάλεται στην οθόνη.

Σ' αυτό το σημείο να πούμε ότι το μενού επιλογών για τις εκθέσεις βρίσκεται στην αρχική φόρμα MDIForm1 (ΕΙΚΟΝΑ3.1) και είναι υπό-μενού της επιλογής "Εμφανήσεις/Εκτυπώσεις".

ΚΕΦΑΛΑΙΟ 4

ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΔΙΑΧΕΙΡΙΣΗΣ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΕΝΟΣ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΟΡΓΑΝΙΣΜΟΥ ΞΕΝΩΝ ΓΛΩΣΣΩΝ (Project1)

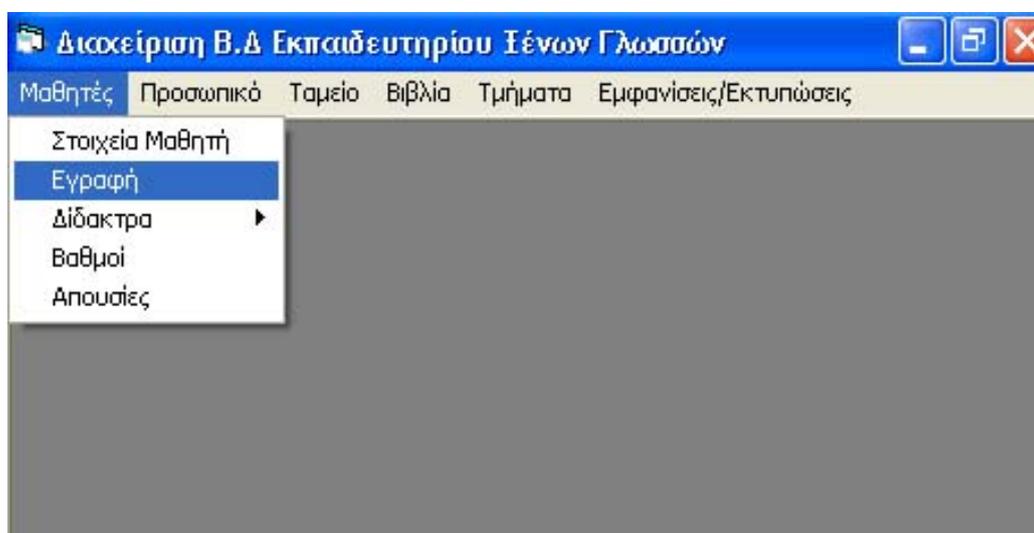
4.1 ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ ΣΤΗΝ ΒΑΣΗ

Παράδειγμα1

Για να εισάγετε δεδομένα σε οποιονδήποτε πίνακα της βάσης αρκεί να επιλέξετε το στοιχείο που σας ενδιαφέρει από την αρχική φόρμα της εφαρμογής. Για παράδειγμα έστω ότι θέλετε να εισάγετε τα στοιχεία μιας Εγγραφής ενός μαθητή (ΕΙΚΟΝΑ4.1) θα ξεκινήσετε επιλέγοντας από το μενού:

Μαθητές —→ Εγγραφή

ΕΙΚΟΝΑ4.1 : Επιλογή εισαγωγής στον πίνακα ΕΓΓΡΑΦΗ για επεξεργασία δεδομένων

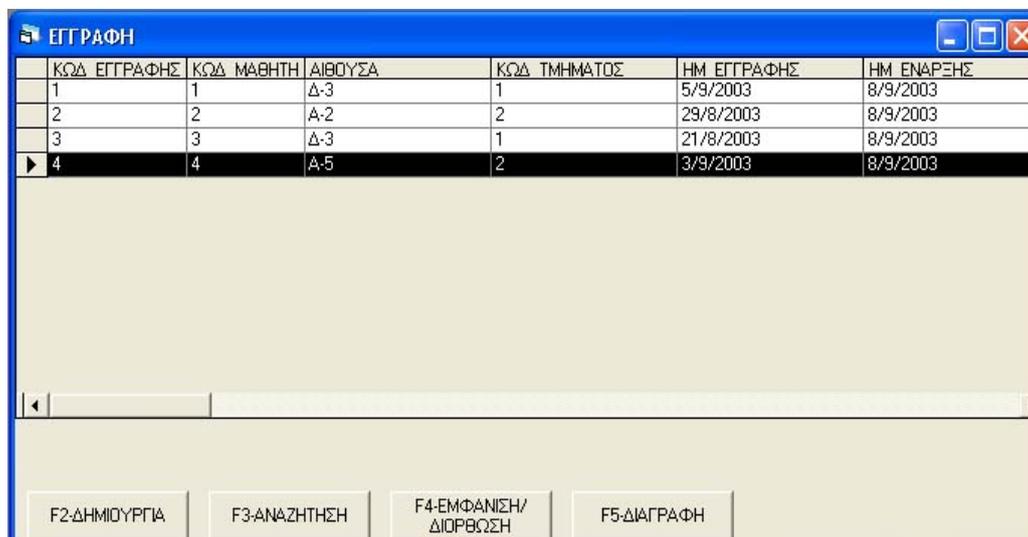


Κάνοντας click στην επιλογή «Εγγραφή» ανοίγει μια φόρμα(ΕΙΚΟΝΑ4.2) που παρουσιάζει όλες τις αποθηκευμένες εγγραφές (αν υπάρχουν) του πίνακα ΕΓΓΡΑΦΗ και μαζί τέσσερις επιλογές επεξεργασίας τους : ΔΗΜΙΟΥΡΓΙΑ, ΑΝΑΖΗΤΗΣΗ, ΕΜΦΑΝΙΣΗ/ΔΙΟΡΘΩΣΗ, ΔΙΑΓΡΑΦΗ.

Για να εισάγετε δεδομένα επιλέγετε το κουμπί ΔΗΜΙΟΥΡΓΙΑ, το οποίο ανοίγει με την σειρά του μια νέα φόρμα (ΕΙΚΟΝΑ4.3). Η φόρμα αυτή εμφανίζει όλα τα ονόματα των πεδίων του πίνακα ΕΓΓΡΑΦΗ και δίπλα τους τα αντίστοιχα Textbox για εισαγωγή δεδομένων από το πληκτρολόγιο. Το πρώτο πεδίο ΚΩΔ_ΕΓΓΡΑΦΗΣ επειδή είναι το πρωτεύον κλειδί του πίνακα και είναι τύπου "Αυτόματης Αρίθμησης"

δεν μπορείτε να του εισάγετε τιμή. Γι' αυτό το λόγο το Textbox αυτού του πεδίου είναι γκριζαρισμένο και μέσα σ' αυτό υπάρχει ανάλογο μήνυμα.

ΕΙΚΟΝΑ4.2 : Τα αποθηκευμένα δεδομένα(εγγραφές) του πίνακα ΕΓΓΡΑΦΗ και οι επιλογές επεξεργασίας τους.



ΚΩΔ ΕΓΓΡΑΦΗΣ	ΚΩΔ ΜΑΘΗΤΗ	ΑΙΘΟΥΣΑ	ΚΩΔ ΤΜΗΜΑΤΟΣ	ΗΜ ΕΓΓΡΑΦΗΣ	ΗΜ ΕΝΑΡΞΗΣ
1	1	Δ-3	1	5/9/2003	8/9/2003
2	2	Α-2	2	29/8/2003	8/9/2003
3	3	Δ-3	1	21/8/2003	8/9/2003
4	4	Α-5	2	3/9/2003	8/9/2003

Buttons: F2-ΔΗΜΙΟΥΡΓΙΑ, F3-ΑΝΑΖΗΤΗΣΗ, F4-ΕΜΦΑΝΙΣΗ/ΔΙΟΡΘΩΣΗ, F5-ΔΙΑΓΡΑΦΗ

Στην ΕΙΚΟΝΑ4.3 δεξιά από τα πεδία ΚΩΔ_ΜΑΘΗΤΗ, ΚΩΔ_ΤΜΗΜΑΤΟΣ, ΗΜ_ΕΓΓΡΑΦΗΣ και ΗΜ_ΕΝΑΡΞΗΣ υπάρχουν τα κουμπιά F2-GO, F3-GO, F4-GO, F5-GO αντίστοιχα. Αυτά τα κουμπιά παραπέμπουν σε άλλες φόρμες για την επιλογή δεδομένων και εισαγωγή αυτών στα πεδία του πίνακα ΕΓΓΡΑΦΗ. Τα Textbox των πεδίων αυτών είναι μπλοκαρισμένα ώστε να μην εισάγουμε δεδομένα από το πληκτρολόγιο.

ΕΙΚΟΝΑ4.3 : Η φόρμα για την εισαγωγή δεδομένων στον πίνακα ΕΓΓΡΑΦΗ



ΚΩΔ_ΕΓΓΡΑΦΗΣ: [Αυτόματη Αρίθμηση] ΚΩΔ_ΜΑΘΗΤΗ: [] F2-GO

ΑΙΘΟΥΣΑ: [] ΚΩΔ_ΤΜΗΜΑΤΟΣ: [] F3-GO

ΗΜ_ΕΓΓΡΑΦΗΣ: [] F4-GO ΗΜ_ΕΝΑΡΞΗΣ: [] F5-GO

Buttons: Enter-OK, Esc-Cancel

Στα δυο τελευταία πεδία που έχουν τύπο δεδομένων "Ημερομηνία", επιλέγοντας τα κουμπιά F4-GO ή F5-GO, εμφανίζεται ένα ημερολόγιο από όπου επιλέγετε τον έτος, τον μήνα και την ημέρα που επιθυμείτε.

Αντίθετα στα άλλα δύο (ΚΩΔ_ΜΑΘΗΤΗΣ και ΚΩΔ_ΤΜΗΜΑΤΟΣ) τα κουμπιά F2-GO και F3-GO σας παραπέμπουν στις εγγραφές των πινάκων ΣΤΟΙΧΕΙΑ ΜΑΘΗΤΗ(ΕΙΚΟΝΑ4.4) και ΤΜΗΜΑΤΑ.

ΕΙΚΟΝΑ4.4 : Οι εγγραφές του πίνακα ΣΤΟΙΧΕΙΑ ΜΑΘΗΤΗ



ΚΩΔΙΚΟΣ	ΟΝΟΜΑ	ΕΠΩΝΥΜΟ	ΟΝΟΜΑ ΠΑΤΡΟΣ	ΟΝΟΜΑ
1	ΘΕΟΔΩΡΟΣ	ΚΟΥΤΡΗΣ	ΠΑΝΑΓΙΩΤΗΣ	ΜΑΡΙΑ
2	ΕΛΕΝΗ	ΛΙΑΝΟΥ	ΑΝΤΩΝΙΟΣ	ΣΠΥΡΙΔ
3	ΔΗΜΗΤΡΗΣ	ΤΖΑΛΛΑΣ	ΒΑΣΙΛΕΙΟΣ	ΑΙΚΑΤΕΡ
4	ΝΙΚΟΛΑΟΣ	ΕΥΘΥΜΙΟΥ	ΓΕΩΡΓΙΟΣ	ΕΛΙΣΑΒΕ

Όταν εμφανιστεί η φόρμα της ΕΙΚΟΝΑΣ4.4 επιλέγετε την εγγραφή που σας ενδιαφέρει (δηλαδή, τον μαθητή για τον οποίο εισάγετε τα στοιχεία της Εγγραφής του στον Εκπαιδευτικό Οργανισμό). Στο παράδειγμα της ΕΙΚΟΝΑΣ4.4 έχει επιλεγεί η εγγραφή με **ΚΩΔΙΚΟ**:1, **ΟΝΟΜΑ**: ΘΕΟΔΩΡΟΣ, **ΕΠΩΝΥΜΟ**: ΚΟΥΤΡΗΣ κ.τ.λ. Αφού λοιπόν επιλέξετε την επιθυμητή εγγραφή πατάτε το κουμπί OK για να επιστρέψετε στην φόρμα εισαγωγής δεδομένων (ΕΙΚΟΝΑ4.3). Στο Textbox του πεδίου ΚΩΔ_ΜΑΘΗΤΗ, τώρα, υπάρχει η τιμή 1 (ο ΚΩΔΙΚΟΣ του μαθητή που επιλέχθηκε). Ομοίως συμπληρώνετε και η τιμή του πεδίου ΚΩΔ_ΤΜΗΜΑΤΟΣ. Όπως καταλαβαίνετε για να εισάγετε δεδομένα στον πίνακα ΕΓΓΡΑΦΗ πρέπει πρώτα να έχετε δημιουργήσει τις απαραίτητες εγγραφές στους πίνακες ΣΤΟΙΧΕΙΑ ΜΑΘΗΤΗ και ΤΜΗΜΑΤΑ.

Όταν λοιπόν συμπληρώσετε * όλα τα Textbox της φόρμας μπορείτε να εισάγετε την εγγραφή στον πίνακα πατώντας το κουμπί Enter-OK. Αν για κάποιο λόγο δεν θέλετε να εισάγετε τα δεδομένα στον πίνακα πατήστε Esc-Cancel.

Τα ίδια βήματα που ακολουθήσατε για να εισάγετε δεδομένα στον πίνακα ΕΓΓΡΑΦΗ ισχύουν και για όλους τους υπόλοιπους πίνακες.

Παράδειγμα2

Έστω ότι θέλετε να εισάγετε δεδομένα στον πίνακα ΒΙΒΛΙΑ. Αρχικά πρέπει να καταχωρήσετε στην βάση δεδομένων τους συγγραφείς και τους εκδοτικούς οίκους στους αντίστοιχους πίνακες. Έπειτα πάτε στην φόρμα εισαγωγής δεδομένων για τον πίνακα ΒΙΒΛΙΑ. Το πεδίο **ISBN** του πίνακα αυτού είναι το μοναδικό πρωτεύων κλειδί το οποίο **δεν** είναι τύπου "Αυτόματης Αρίθμησης". Το Textbox του δεν είναι κλειδωμένο, γι' αυτό πρέπει να του εισάγετε δεδομένα από το πληκτρολόγιο.

Για τα πεδία ΣΥΓΓΡΑΦΕΑΣ και ΕΚΔΟΣΕΙΣ πατάτε τα κουμπιά F2-GO και F3-GO αντίστοιχα για να εμφανιστούν οι εγγραφές των πινάκων ΣΥΓΓΡΑΦΕΙΣ και ΕΚΔΟΣΕΙΣ για να επιλέξετε αυτές που επιθυμείτε.

* Κανένα πεδίο δεν πρέπει να μείνει κενό. Αν για κάποιο δεν γνωρίζετε τι στοιχεία να εισάγετε ή αν δεν υπάρχουν δώστε τιμή μηδέν (0) ή πούλα (-).

Αφού τις επιλέξετε και συμπληρώσετε όλα τα δεδομένα για την νέα εγγραφή του πίνακα ΒΙΒΛΙΑ (ΕΙΚΟΝΑ4.5) πατήστε Enter-OK για να καταχωρηθούν στην βάση.

ΕΙΚΟΝΑ4.5 : Η φόρμα για την εισαγωγή δεδομένων στον πίνακα ΒΙΒΛΙΑ

The screenshot shows a form titled "ΒΙΒΛΙΑ" with the following fields and values:

ISBN	960-512-155-7	ΤΙΤΛΟΣ	ACTIVE STUDY
ΣΥΓΓΡΑΦΕΑΣ	2	ΕΚΔΟΣΕΙΣ	1
ΠΟΣΟΤΗΤΑ	5		

Buttons: Enter-OK, Esc-Cancel

Παρόλο όμως που στην ΕΙΚΟΝΑ4.5 τα πεδία ΣΥΓΓΡΑΦΕΑΣ και ΕΚΔΟΣΕΙΣ έχουν τις τιμές 2 και 1 αντίστοιχα (από τα πρωτεύοντα κλειδιά των εγγραφών των πινάκων ΣΥΓΓΡΑΦΕΑΣ και ΕΚΔΟΣΕΙΣ), στην εγγραφή του πίνακα ΒΙΒΛΙΑ αποθηκεύονται οι τιμές **ΕΜ.Ν ΔΕΛΗΓΙΑΝΝΑΚΗΣ** και **LONGMAN GROUP** όπως βλέπετε στην ΕΙΚΟΝΑ4.6

ΕΙΚΟΝΑ4.6 : Οι εγγραφή που καταχωρήθηκε στον πίνακα ΒΙΒΛΙΑ

ISBN	ΤΙΤΛΟΣ	ΣΥΓΓΡΑΦΕΑΣ	ΕΚΔΟΣΕΙΣ	ΠΟΣΟΤΗΤΑ
960-512-155-7	ACTIVE STUDY	ΕΜ.Ν. ΔΕΛΗΓΙΑΝΝΑΚΗΣ	LONGMAN GROUP	5

Η περίπτωση αυτή, όπου στα πεδία ΣΥΓΓΡΑΦΕΑΣ και ΕΚΔΟΣΕΙΣ δεν αποθηκεύονται οι τιμές των πρωτευόντων κλειδίων των σχετιζόμενων πεδίων, ισχύει και στον πίνακα ΕΣΟΔΑ_ΕΞΟΔΑ για το πεδίο ΛΟΓΑΡΙΑΣΜΟΣ.

4.2 ΑΝΑΖΗΤΗΣΗ ΕΓΓΡΑΦΗΣ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Πολλές φορές για λόγους έλλειψης χρόνου κάποιες ενέργειες πρέπει να γίνουν με τον ταχύτερο τρόπο. Σε περίπτωση, για παράδειγμα, που θέλετε να κάνετε μια αναζήτηση σε έναν πίνακα είναι προτιμότερο να κα κάνετε μια δυναμική αναζήτηση με την βοήθεια που σας προσφέρει το Project1, αντί να ψάξετε μια-μια τις εγγραφές του πίνακα.

Έστω ότι ψάχνετε τα στοιχεία ενός Καθηγητή του Εκπαιδευτικού Οργανισμού, ξεκινήστε με τα ακόλουθα βήματα (από την αρχική φόρμα του Project1):

Προσωπικό → Καθηγητές → ΑΝΑΝΑΖΗΤΗΣΗ

ΕΙΚΟΝΑ4.7 : Αναζήτηση εγγραφής στον πίνακα ΚΑΘΗΓΗΤΕΣ

ΚΩΔ. ΚΑΘΗΓΗΤΗ	ΟΝΟΜΑ	ΕΠΩΝΥΜΟ	ΟΝΟΜΑ ΠΑΤΡΟΣ	ΔΔΤ	ΗΜ. ΓΕΝΝΗΣΕΩΣ
1	ΔΗΜΗΤΡΙΟΣ	ΚΟΝΕΤΑΣ	ΧΡΗΣΤΟΣ	ΝΕΚΤΑΡΙΑ	25/3/1970

ΕΠΩΝΥΜΟ
ΚΟΝ

F2-ΔΗΜΙΟΥΡΓΙΑ F3-ΑΝΑΖΗΤΗΣΗ F4-ΕΜΦΑΝΙΣΗ/ΔΙΟΡΘΩΣΗ F5-ΔΙΑΓΡΑΦΗ

Αφού λοιπόν επιλέξετε να κάνετε ΑΝΑΖΗΤΗΣΗ, εμφανίζεται ένα μενού με όλα τα πεδία του πίνακα. Στο παράδειγμα τις ΕΙΚΟΝΑΣ4.7 από το μενού αυτό έχει επιλεγεί το πεδίο ΕΠΩΝΥΜΟ και σαν στοιχείο αναζήτησης έχει δοθεί η λέξη ΚΟΝ η οποία είναι αρκετή για να βρεθεί η εγγραφή που αναζητείται. Η εγγραφή αυτή εμφανίζεται στο πάνω μέρος της φόρμας και οι υπόλοιπες "εξαφανίζονται" προσωρινά.

Για οποιαδήποτε αναζήτηση θέλετε να κάνετε στην βάση δεδομένων αρκεί να εκτελέσετε τις παραπάνω ενέργειες. Χρησιμοποιώντας βέβαια κάθε φορά τον πίνακα και τα στοιχεία που σας ενδιαφέρουν.

4.3 ΔΙΟΡΘΩΣΗ/ΑΛΛΑΓΗ ΔΕΔΟΜΕΝΩΝ

Κάποιες φορές από κεκτημένη ταχύτητα ή από λάθος συνεννόηση αποθηκεύουμε στην βάση δεδομένων λανθασμένα στοιχεία. Γι' αυτόν το λόγο η εφαρμογή Project1 σας δίνει την δυνατότητα να διορθώνετε αυτά τα λάθη.

Έστω ότι θέλετε να διορθώσετε το πεδίο ΑΙΘΟΥΣΑ μιας εγγραφής του πίνακα ΕΓΓΡΑΦΕΣ. Πηγαίνετε στην φόρμα της ΕΙΚΟΝΑΣ4.2 (η τέταρτη εγγραφή είναι επιλεγμένη*) και πατήστε το κουμπί F4-ΕΜΦΑΝΙΣΗ/ΔΙΟΡΘΩΣΗ. Τότε θα εμφανιστεί η ίδια φόρμα (ΕΙΚΟΝΑ4.3) με την οποία κάνατε εισαγωγή δεδομένων, μόνο που τα Textbox των πεδίων δεν θα είναι κενά αλλά θα περιέχουν τις τιμές των πεδίων της επιλεγμένης εγγραφής (ΕΙΚΟΝΑ4.8).

Θα αναρωτιέστε βέβαια γιατί το κουμπί αυτό λέγεται ΕΜΦΑΝΙΣΗ/ΔΙΟΡΘΩΣΗ και όχι απλά ΔΙΟΡΘΩΣΗ. Αυτό συμβαίνει επειδή στην φόρμα της ΕΙΚΟΝΑΣ4.2 δεν μπορείτε να δείτε μια εγγραφή ολόκληρη(με όλα τα πεδία στην οθόνη ταυτόχρονα), αλλά μετακινώντας την μπάρα κύλισης να βλέπετε μισά και μισά τα πεδία. Η

* Πρέπει οπωσδήποτε να επιλέξετε μια εγγραφή πριν επιλέξετε να κάνετε διόρθωση.

παρακάτω φόρμα σας βοηθάει ,εκτός από το να διορθώσετε κάποιο λάθος, να δείτε όλα τα στοιχεία μιας εγγραφής (πεδία, τιμές) ταυτόχρονα σε μορφή φόρμας.

ΕΙΚΟΝΑ4.8 : Διόρθωση στοιχείων μιας εγγραφής για τον πίνακα ΕΓΓΡΑΦΗ



ΚΩΔ_ΕΓΓΡΑΦΗΣ	4	ΚΩΔ_ΜΑΘΗΤΗ	4	F2-GO	
ΑΙΘΟΥΣΑ	A-4	ΚΩΔ_ΤΜΗΜΑΤΟΣ	2	F3-GO	
ΗΜ_ΕΓΓΡΑΦΗΣ	3/9/2003	F4-GO	ΗΜ_ΕΝΑΡΞΗΣ	8/9/2003	F5-GO

Enter-OK Esc-Cancel

Το πεδίο ΑΙΘΟΥΣΑ του πίνακα ΕΓΓΡΑΦΗ έχει τώρα την τιμή "Α-4" και όχι "Α-5" που ήταν πριν. Αν θέλετε να καταχωρηθεί η εγγραφή με την διόρθωση που έχει γίνει πατάτε το κουμπί Enter-OK. Ενώ αν θέλετε να ακυρωθεί η ενέργεια που κάνατε πατήστε Esc-Cancel.

Στους πίνακες ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ, ΠΛΗΡΩΘΕΝΤΑ ΔΙΔΑΚΤΡΑ και ΛΟΓΑΡΙΑΣΜΟΙ δεν μπορείτε να κάνετε καμία διόρθωση. Γι' αυτό απαιτείται ιδιαίτερη προσοχή κατά την εισαγωγή δεδομένων σ' αυτούς.

4.4 ΔΙΑΓΡΑΦΗ ΕΓΓΡΑΦΗΣ ΑΠΟ ΤΗΝ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Η διαγραφή μιας εγγραφής από την βάση δεδομένων είναι μια πολύ απλή διαδικασία, αρκεί μόνο να προσέχετε ποια εγγραφή επιλέγετε για να διαγράψετε.

Έστω ότι θέλετε να διαγράψετε μια εγγραφή του πίνακα ΕΓΓΡΑΦΗ (ΕΙΚΟΝΑ4.2). Απλά επιλέγετε την εγγραφή που επιθυμείτε και στην συνέχεια πατάτε το κουμπί F5-ΔΙΑΓΡΑΦΗ. Πριν το πρόγραμμα διαγράψει την εγγραφή από την βάση σας εμφανίζει ένα μήνυμα το οποίο σας ζητάει να επιβεβαιώσετε ότι θέλετε να διαγράψετε οπωσδήποτε αυτή την εγγραφή.

Από την τη στιγμή που θα διαγράψετε τελείως την εγγραφή δεν υπάρχει περίπτωση επαναφοράς αυτής στην βάση δεδομένων.

4.5 ΕΜΦΑΝΙΣΕΙΣ/ΕΚΤΥΠΩΣΕΙΣ

Κάποια στοιχεία (δεδομένα) της Βάσης χρειάζονται, από τον Εκπαιδευτικό Οργανισμό, να είναι τυπωμένα σε χαρτί και όχι μόνο αποθηκευμένα στον υπολογιστή. Μιλάμε βέβαια για τις Αναφορές της εφαρμογής Project1. Σκοπός αυτών είναι να βοηθούν (το Διοικητικό προσωπικό κυρίως) στο να ενημερώνεται ο οργανισμός για την λειτουργική και οικονομική του κατάσταση.

Στο αρχικό μενού (ΕΙΚΟΝΑ4.1) της εφαρμογής Project1 η επιλογή "Εμφανίσεις/Εκτυπώσεις" περιέχει το υπομενού με όλες τις επιλογές των Αναφορών που θέλετε να εκτυπώσετε:

Στοιχεία Μαθητών
Εγγραφές >>

Βαθμοί >>

Στοιχεία Προσωπικού :

Διοικητικό

Καθηγητές

Υπάλληλοι

Συμφωνηθέντα Δίδακτρα

Λογαριασμοί

Έσοδα / Έξοδα

Βιβλία και Τμήματα.

Εκτός από το να εκτυπώσετε τις Αναφορές αυτές, μπορείτε απλά να τις χρησιμοποιήσετε για να δείτε τα αποθηκευμένα δεδομένα σε διαμόρφωση σελίδας, και όχι σε ένα πίνακα ή μια φόρμα.

4.6 ΕΙΔΙΚΕΣ ΠΕΡΙΠΤΩΣΕΙΣ

Η επεξεργασία κάποιων δεδομένων ή κάποιων ενεργειών της εφαρμογής Project1 διαφέρει από τα παραδείγματα που αναλύθηκαν παραπάνω. Στην ενότητα αυτή θα παρουσιαστούν όλες αυτές τις περιπτώσεις.

1. Στην εισαγωγή δεδομένων του πίνακα ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ το πεδίο ΠΟΣΟ ΔΟΣΗΣ είναι κλειδωμένο(δεν μπορείτε να του δώσετε τιμή από το πληκτρολόγιο). Το πεδίο αυτό παίρνει τιμή από την διαίρεση των πεδίων ΣΥΝΟΛΙΚΟ ΠΟΣΟ / ΑΡ_ΔΟΣΕΩΝ.

Το **Συνολικό Ποσό** δεν πρέπει να ξεπερνάει τα 3000 €

Ο μέγιστος **Αριθμός Δόσεων** είναι το 18.

Επίσης στον πίνακα αυτό δεν μπορείτε να διορθώσετε καμία εγγραφή.

2. Όταν εισάγετε δεδομένα στον πίνακα ΠΛΗΡΩΘΕΝΤΑ ΔΙΔΑΚΤΡΑ ενός μαθητή, το πεδίο ΥΠΟΛΟΙΠΟ παίρνει τιμή από την αφαίρεση των πεδίων ΣΥΝΟΛΙΚΟ ΠΟΣΟ (πίνακας ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ) - ΠΟΣΟ (τρέχον πίνακα).

Την δεύτερη φορά που θα καταχωρείται εγγραφή για τον ίδιο μαθητή το πεδίο ΥΠΟΛΟΙΠΟ θα παίρνει τιμή από την αφαίρεση του προηγούμενου

Υπολοίπου μείον την τρέχουσα τιμή του πεδίου ΠΟΣΟ.

Σαφώς το πεδίο ΥΠΟΛΟΙΠΟ δεν μπορεί να πάρει τιμή από το πληκτρολόγιο.

Όταν μηδενιστεί το πεδίο ΥΠΟΛΟΙΠΟ σημαίνει ότι ο μαθητής δεν οφείλει άλλο χρηματικό ποσό προς τον οργανισμό.

Επίσης σ' αυτόν τον πίνακα δεν μπορεί να γίνει καμία διόρθωση

3. Για να δημιουργήσετε ένα νέο Λογαριασμό πρέπει να επιλέξετε από τον πίνακα ΑΙΤΙΟΛΟΓΙΑ_1 την τιμή ΕΣΟΔΑ ή ΕΞΟΔΑ. Καμία άλλη τιμή δεν γίνεται δεκτή. Επίσης δεν μπορείτε να επέμβετε στο πεδίο ΛΟΓΑΡΙΑΣΜΟΣ.

Δεν μπορείτε να κάνετε καμία διόρθωση σε εγγραφή του πίνακα ΛΟΓΑΡΙΑΣΜΟΙ.

4. Στους πίνακες ΑΠΟΥΣΙΕΣ και ΕΣΟΔΑ_ΕΞΟΔΑ υπάρχει μια επιπλέον επιλογή , εκτός από Δημιουργία, Αναζήτηση, Εμφάνιση/Διόρθωση, Διαγραφή. Η επιλογή ΣΥΝΟΛΟ ΑΠΟΥΣΙΩΝ και ΣΥΝΟΛΙΚΟ ΠΟΣΟ αντίστοιχα σε κάθε πίνακα. Η πρώτη εμφανίζει το σύνολο απουσιών ενός Μαθητή ενώ η δεύτερη το συνολικό ποσό ενός Λογαριασμού.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Φέρνοντας σε πέρας την πτυχιακή εργασία μου (μια ολοκληρωμένη εφαρμογή διαχείρισης βάσης δεδομένων ενός Εκπαιδευτικού Οργανισμού Ξένων Γλωσσών) δημιούργησα ένα σύστημα το οποίο μπορεί να επεξεργάζεται τα δεδομένα μιας βάσης με τον καλύτερο δυνατό τρόπο, προς αποφυγή λαθών και παροχή σωστών αποτελεσμάτων (πληροφορίες). Ωστόσο παρατήρησα ότι κάποια πράγματα θα μπορούσαν να είχαν γίνει διαφορετικά και κάποια άλλα να προστεθούν στην εφαρμογή για την καλύτερη απόδοση του συστήματος.

Για παράδειγμα, οι πληροφορίες που διαχειρίζεται η εφαρμογή για την βιβλιοθήκη του Εκπαιδευτικού Οργανισμού είναι μόνο ποια βιβλία περιέχονται σ' αυτή και σε ποιες ποσότητες το καθένα. Θα μπορούσε επιπλέον η εφαρμογή να επεξεργάζεται τους δανεισμούς των βιβλίων αυτών, δηλ. ποιος μαθητής και σε ποια ημερομηνία δανείστηκε κάποιο βιβλίο. Έτσι θα ήταν τελειοποιημένη η διαχείριση της βάσης για την βιβλιοθήκη του οργανισμού.

Ένα ακόμα σημείο της εφαρμογής το οποίο θα μπορούσε να είχε σχεδιαστεί διαφορετικά είναι οι πληρωμές των διδάκτρων των μαθητών. Όπως είναι τώρα η εφαρμογή αποθηκεύονται στην βάση δεδομένων οι πληρωμές, με ημερομηνίες, ποσά κ.τ.λ. και μετά πιστώνετε ο αντίστοιχος λογαριασμός από τον χρήστη. Θα ήταν καλύτερα αν αυτές οι δύο ενέργειες εκτελούνταν ταυτόχρονα. Αυτή η αλλαγή θα έκανε την εφαρμογή ταχύτερη.

Τέλος σε μία μελλοντική τροποποίηση της εφαρμογής αυτό που θα προσπαθήσω να αλλάξω/προσθέσω πρώτο είναι οι εκτυπώσεις αναφορών με βάση αναζητήσεις εγγραφών. Για παράδειγμα, να εκτυπωθούν α) τα στοιχεία του μαθητή με τον Κωδικό 8, β) οι μαθητές που απουσίασαν στις 18/11/2003 ή γ) οι Λογαριασμοί με μηδενικό υπόλοιπο.

Κλείνοντας, αυτό που μπορώ να πω είναι ότι η εφαρμογή διαχείρισης της βάσης δεδομένων ενός Εκπαιδευτικού Οργανισμού Ξένων Γλωσσών που έχω δημιουργήσει μπορεί να δουλέψει με πολύ καλή απόδοση για τον οργανισμό και χωρίς τις αλλαγές που προανέφερα. Ωστόσο κάθε εφαρμογή πρέπει να εξελίσσεται και να βελτιώνεται ώστε να ανταποκρίνεται στα νέα δεδομένα ενός οργανισμού. Γι' αυτό το λόγο είναι απαραίτητη η δημιουργία και η ενοποίηση των παραπάνω στην εφαρμογή.

ΠΑΡΑΡΤΗΜΑ

MDIForm1

```
Option Explicit
Public Con As ADODB.Connection
Public RS As ADODB.Recordset

Private Sub MDIForm_Load()

'Access.OpenAccessProject ("c:\programming\job_control.mdb")

Dim strMDB As String
Dim strTableNameKeys As String
Dim strPrimaryKey As String
Set Module1.colPrimaryKeys = New Collection
strMDB = "c:\TED\ITYXIAKH\EKPAIDEYTHRIO.mdb"

Set Con = New ADODB.Connection
Set RS = New ADODB.Recordset
With Con
.Provider = "Microsoft.Jet.OLEDB.3.51"
.ConnectionString = strMDB
.Open
End With
Set RS = Con.OpenSchema(adSchemaPrimaryKeys)

Dim fld As ADODB.field
For Each fld In RS.fields
Debug.Print fld.Name
Next
While Not RS.EOF

strTableNameKeys = RS("TABLE_NAME")
strPrimaryKey = RS("COLUMN_NAME")
Debug.Print RS("TABLE_NAME"), RS("COLUMN_NAME")
Module1.colPrimaryKeys.Add strPrimaryKey, strTableNameKeys
RS.MoveNext
Wend
End Sub

Private Sub MDIForm_Resize()
MDIForm1.WindowState = 2
End Sub

Private Sub MDIForm_Unload(Cancel As Integer)
If Con.State = adStateOpen Then
Con.Close
End If
Set Con = Nothing
```

End Sub

```
Private Sub mnuApousies_Click()  
If Module1.boolGnrIFrmLoaded Then Unload frmGeneralTblFunctions  
Module1.strTableName = "ΑΠΟΥΣΙΕΣ"  
frmGeneralTblFunctions.Caption = "ΑΠΟΥΣΙΕΣ"  
End Sub
```

```
Private Sub mnuBathmoi_Click()  
If Module1.boolGnrIFrmLoaded Then Unload frmGeneralTblFunctions  
Module1.strTableName = "ΒΑΘΜΟΙ"  
frmGeneralTblFunctions.Caption = "ΒΑΘΜΟΙ"  
End Sub
```

```
Private Sub mnuBathmoi2_Click()  
DataReport3.Show
```

End Sub

```
Private Sub mnuBiblia2_Click()  
DataReport10.Show
```

End Sub

```
Private Sub mnuBiblio_Click()  
If Module1.boolGnrIFrmLoaded Then Unload frmGeneralTblFunctions  
Module1.strTableName = "ΒΙΒΛΙΑ"  
frmGeneralTblFunctions.Caption = "ΒΙΒΛΙΑ"
```

End Sub

```
Private Sub mnuDimioyrgiaEpexergasia_Click()  
If Module1.boolGnrIFrmLoaded Then Unload frmGeneralTblFunctions  
Module1.strTableName = "ΤΜΗΜΑΤΑ"  
frmGeneralTblFunctions.Caption = "ΤΜΗΜΑΤΑ"  
End Sub
```

```
Private Sub mnuDioikitiko_Click()  
If Module1.boolGnrIFrmLoaded Then Unload frmGeneralTblFunctions  
Module1.strTableName = "ΔΙΟΙΚΗΤΙΚΟ ΠΡΟΣΩΠΙΚΟ"  
frmGeneralTblFunctions.Caption = "ΔΙΟΙΚΗΤΙΚΟ ΠΡΟΣΩΠΙΚΟ"  
End Sub
```

```
Private Sub mnuDioikitiko2_Click()  
DataReport4.Show  
End Sub
```

```
Private Sub mnuEfragh2_Click()  
DataReport1.Show  
End Sub
```

```
Private Sub mnuEgraph_Click()  
If Module1.boolGnrIFrmLoaded Then Unload frmGeneralTblFunctions  
Module1.strTableName = "ΕΓΓΡΑΦΗ"
```

```
frmGeneralTblFunctions.Caption = "ΕΓΓΡΑΦΗ"  
End Sub
```

```
Private Sub mnuEkdoseis_Click()  
If Module1.boolGnrIFrmLoaded Then Unload frmGeneralTblFunctions  
Module1.strTableName = "ΕΚΔΟΣΕΙΣ"  
frmGeneralTblFunctions.Caption = "ΕΚΔΟΣΕΙΣ"  
End Sub
```

```
Private Sub mnuEsodaExoda2_Click()  
DataReport8.Show
```

```
End Sub
```

```
Private Sub mnuKathigites_Click()  
If Module1.boolGnrIFrmLoaded Then Unload frmGeneralTblFunctions  
Module1.strTableName = "ΚΑΘΗΓΗΤΕΣ"  
frmGeneralTblFunctions.Caption = "ΚΑΘΗΓΗΤΕΣ"  
End Sub
```

```
Private Sub mnuKathigites2_Click()  
DataReport5.Show  
End Sub
```

```
Private Sub mnuLogariasmoi2_Click()  
DataReport9.Show
```

```
End Sub
```

```
Private Sub mnuPlir_Click()  
If Module1.boolGnrIFrmLoaded Then Unload frmGeneralTblFunctions  
Module1.strTableName = "ΠΛΗΡΩΘΕΝΤΑ ΔΙΔΑΚΤΡΑ"  
frmGeneralTblFunctions.Caption = "ΠΛΗΡΩΘΕΝΤΑ ΔΙΔΑΚΤΡΑ"  
End Sub
```

```
Private Sub mnuStoixeia2_Click()  
DataReport2.Show
```

```
End Sub
```

```
Private Sub mnuSyggrafeas_Click()  
If Module1.boolGnrIFrmLoaded Then Unload frmGeneralTblFunctions  
Module1.strTableName = "ΣΥΓΓΡΑΦΕΙΣ"  
frmGeneralTblFunctions.Caption = "ΣΥΓΓΡΑΦΕΙΣ"  
End Sub
```

```
Private Sub mnuEsodaExoda_Click()  
If Module1.boolGnrIFrmLoaded Then Unload frmGeneralTblFunctions  
Module1.strTableName = "ΕΣΟΔΑ_ΕΞΟΔΑ"  
frmGeneralTblFunctions.Caption = "ΕΣΟΔΑ_ΕΞΟΔΑ"  
End Sub
```

```
Private Sub mnuLogariasmoi_Click()
If Module1.boolGnrIFrmLoaded Then Unload frmGeneralTblFunctions
Module1.strTableName = "ΛΟΓΑΡΙΑΣΜΟΙ"
frmGeneralTblFunctions.Caption = "ΛΟΓΑΡΙΑΣΜΟΙ"
End Sub
```

```
Private Sub mnuSTOIXEIA_Click()
If Module1.boolGnrIFrmLoaded Then Unload frmGeneralTblFunctions
Module1.strTableName = "ΣΤΟΙΧΕΙΑ ΜΑΘΗΤΗ"
frmGeneralTblFunctions.Caption = "ΣΤΟΙΧΕΙΑ ΜΑΘΗΤΗ"
```

```
End Sub
```

```
Private Sub mssysrelationships_Click()
If Module1.boolGnrIFrmLoaded Then Unload frmGeneralTblFunctions
Module1.strTableName = "MyRelationships"
frmGeneralTblFunctions.Caption = "MyRelationships"
End Sub
```

```
Private Sub mnuSymf_Click()
If Module1.boolGnrIFrmLoaded Then Unload frmGeneralTblFunctions
Module1.strTableName = "ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ"
frmGeneralTblFunctions.Caption = "ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ"
End Sub
```

```
Private Sub mnuSymfonithenta2_Click()
    DataReport7.Show
End Sub
```

```
Private Sub mnuTmhmata2_Click()
    DataReport11.Show
End Sub
```

```
Private Sub mnuYpallhloi2_Click()
    DataReport6.Show
End Sub
```

```
Private Sub mnuYpalliloi_Click()
If Module1.boolGnrIFrmLoaded Then Unload frmGeneralTblFunctions
Module1.strTableName = "ΥΠΑΛΛΗΛΟΙ"
frmGeneralTblFunctions.Caption = "ΥΠΑΛΛΗΛΟΙ"
End Sub
```

frmGeneralTblFunctions

Option Explicit

Public Con As ADODB.Connection

Public RS As ADODB.Recordset

Public objdragsource As Object

Private Sub Command2_Click()

 Text1.Visible = True

 Combo1.Visible = True

 Dim field As ADODB.field

 Combo1.Clear

 For Each field In RS.fields

 Combo1.AddItem field.Name

 Next

 Combo1.Text = Combo1.List(0)

 Text1.SetFocus

End Sub

Private Sub Command4_Click()

If Not RS.EOF Then

 Dim a As Long

 a = CLng(RS.Bookmark)

 Dim strWhere As String

 If Module1.strTableName = "BIBΛIA" Then

 strWhere = "[" & RS.fields(0).Name & "]" = " & "" & DataGrid1.Columns(0) & ""

 Else

 strWhere = "[" & RS.fields(0).Name & "]" = " & DataGrid1.Columns(0) & ""

 End If

 Dim ind As Integer

 ind = 0

 If Module1.colPrimaryKeys(Module1.strTableName) = RS.fields(0).Name Then

 Debug.Print "primkey"

 Else

 colprerr:

 Dim fld As ADODB.field

 For ind = 1 To RS.fields.Count - 1

 If DataGrid1.Columns(ind) <> "" Then

 Debug.Print RS.fields(ind).Type

 If RS.fields(ind).Type = adInteger Then

 strWhere = strWhere & " AND [" & RS.fields(ind).Name & "]" = " &

 DataGrid1.Columns(ind) & ""

 Else

 strWhere = strWhere & " AND [" & RS.fields(ind).Name & "]" = " &

 DataGrid1.Columns(ind) & ""

 End If

 End If

 ind = ind + 1

 Next

 End If

 If MsgBox("Επιβεβαίωση Διαγραφής", vbOKCancel, "ΠΙΡΟΣΟΧΗ") = vbOK Then

 Call Module1.Delete(Module1.strTableName, strWhere, " ", Con)

```

Call Form_Load
If Not RS.EOF Then RS.Move a - 1
End If

```

```

End If
End Sub

```

```

Private Sub Command5_Click()
On Error GoTo view_error
Dim root As String
Dim node As String
Dim obj As Object
Dim i As Integer
Dim a As Long

```

```

a = CLng(RS.Bookmark)

```

```

Dim rs3 As New ADODB.Recordset
With rs3
.ActiveConnection = Con
.CursorLocation = adUseClient
.CursorType = adOpenDynamic
.LockType = adLockReadOnly
End With

```

```

If Module1.strTableName = "ΑΠΟΥΣΙΕΣ" Then
If Combo1.Text <> "" Then
rs3.Open "SELECT SUM([ΩΡΕΣ]) FROM [ΑΠΟΥΣΙΕΣ] WHERE ([ " & Combo1.Text
& "] LIKE " & VBA.Trim(Text1.Text) & "%' OR [ " & Combo1.Text & "] LIKE %" &
VBA.Trim(Text1.Text) & "%'"
MsgBox "ΣΥΝΟΛΟ ΑΠΟΥΣΙΩΝ ΜΑΘΗΤΗ: " & rs3(0).Value, , "ΣΥΝΟΛΟ
ΑΠΟΥΣΙΩΝ"
rs3.Close
Else
MsgBox "Απαιτείται Αναζήτηση Του ΚΩΔ_ΜΑΘΗΤΗ", vbCritical, "ΜΗΝΥΜΑ"
End If
End If

```

```

If Module1.strTableName = "ΕΣΟΔΑ_ΕΞΟΔΑ" Then
If Combo1.Text <> "" Then
rs3.Open "SELECT SUM([ΠΟΣΟ]) FROM [ΕΣΟΔΑ_ΕΞΟΔΑ] WHERE ([ " &
Combo1.Text & "] LIKE " & VBA.Trim(Text1.Text) & "%' OR [ " & Combo1.Text & "]
LIKE %" & VBA.Trim(Text1.Text) & "%'"
MsgBox "ΣΥΝΟΛΙΚΟ ΠΟΣΟ ΛΟΓΑΡΙΑΣΜΟΥ: " & rs3(0).Value & " €", ,
"ΣΥΝΟΛΙΚΟ ΠΟΣΟ"
rs3.Close
Else
MsgBox "Απαιτείται Αναζήτηση Του ΛΟΓΑΡΙΑΣΜΟΥ", vbCritical, "ΜΗΝΥΜΑ"
End If
End If

```

```

Exit Sub
view_error:

```

```
MsgBox "Δεν υπάρχουν εγγραφές", vbCritical, "ΜΗΝΥΜΑ"  
Call Form_Load
```

```
End Sub
```

```
Private Sub DataGrid1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As  
Single)
```

```
If Button = vbLeftButton Then
```

```
    Call DataGrid1_MouseUp(0, 0, X, Y)
```

```
    Set objdragsource = DataGrid1.Columns(1)
```

```
    DataGrid1.Drag vbBeginDrag
```

```
End If
```

```
End Sub
```

```
Private Sub DataGrid1_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As  
Single)
```

```
'cc
```

```
End Sub
```

```
Private Sub Form_Activate()
```

```
'Call Form_Load
```

```
End Sub
```

```
Private Sub Form_GotFocus()
```

```
Call Form_Load
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    WindowState = 0
```

```
    Me.Move 100, 100
```

```
Dim SqlSelect As String
```

```
SqlSelect = "SELECT * FROM [" & Module1.strTableName & "]"
```

```
Set RS = New ADODB.Recordset
```

```
Set Con = MDIForm1.Con
```

```
With RS
```

```
    .ActiveConnection = Con
```

```
    .CursorLocation = adUseClient
```

```
    .CursorType = adOpenDynamic
```

```
    .LockType = adLockReadOnly
```

```
    .Open SqlSelect
```

```
End With
```

```
If Module1.strTableName = "ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ" Then
```

```
    Command3.Caption = "F4-ΕΜΦΑΝΙΣΗ"
```

```
End If
```

```
If Module1.strTableName = "ΠΛΗΡΩΘΕΝΤΑ ΔΙΔΑΚΤΡΑ" Then
```

```
    Command3.Caption = "F4-ΕΜΦΑΝΙΣΗ"
```

```
End If
```

```

If Module1.strTableName = "ΛΟΓΑΡΙΑΣΜΟΙ" Then
    Command3.Caption = "F4-ΕΜΦΑΝΙΣΗ"
End If
If Module1.strTableName = "ΕΣΟΔΑ_ΕΞΟΔΑ" Then
    Command5.Visible = True
    Command5.Caption = "ΣΥΝΟΛΙΚΟ ΠΟΣΟ"
End If
If Module1.strTableName = "ΑΠΟΥΣΙΕΣ" Then
    Command5.Visible = True
    Command5.Caption = "ΣΥΝΟΛΟ ΑΠΟΥΣΙΩΝ"
End If

Set DataGrid1.DataSource = RS

DataGrid1.Refresh
DataGrid1.MarqueeStyle = dbgHighlightRow
Dim col As Object
For Each col In DataGrid1.Columns
    col.Width = col.Width + 100
Next
Module1.boolGnrlFrmLoaded = True

End Sub
Private Sub DataGridTitles()
    Dim intColId As Integer
    Dim strfield As ADODB.field
    intColId = 0
    For Each strfield In RS.fields
        DataGrid1.Columns(intColId).Caption = strfield.Name
        frmTblInsert.Label1(0).Caption = strfield.Name
        DataGrid1.Columns(intColId).Width = TextWidth(strfield.Name)
    Next
End Sub
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyF2 Then
        Call Command1_Click
    End If
    If KeyCode = vbKeyF3 Then
        Call Command2_Click
    End If
    If KeyCode = vbKeyF4 Then
        Call Command3_Click
    End If
    If KeyCode = vbKeyF5 Then
        Call Command4_Click
    End If
    If KeyCode = vbKeyEscape Then
        Unload Me
    End If
End Sub
Private Sub DataGrid1_KeyDown(KeyAscii As Integer, Shift As Integer)
    'checking if Fnctn keys pressed
    If KeyAscii = VBRUN.vbKeyF2 Then
        Call Command1_Click
    End If

```

```

If KeyAscii = VBRUN.vbKeyF3 Then
    Call Command2_Click
End If
If KeyAscii = VBRUN.vbKeyF4 Then
    Call Command3_Click
End If
If KeyAscii = VBRUN.vbKeyF5 Then
    Call Command4_Click
End If
If KeyAscii = VBRUN.vbKeyEscape Then
    Unload Me
End If

End Sub

Public Sub DataGrid1_KeyPress(KeyAscii As Integer)

    If KeyAscii = VBRUN.vbKeyF2 Then
        Call Command1_Click
    End If
    If KeyAscii = VBRUN.vbKeyF3 Then
        Call Command2_Click
    End If
    If KeyAscii = VBRUN.vbKeyF4 Then
        Call Command2_Click
    End If
    If KeyAscii = VBRUN.vbKeyF5 Then
        Call Command4_Click
    End If
    If KeyAscii = VBRUN.vbKeyEscape Then
        Unload Me
    End If
End Sub

Private Sub Form_KeyPress(KeyAscii As Integer)
    If KeyAscii = VBRUN.vbKeyF2 Then
        Call Command1_Click
    End If
    If KeyAscii = VBRUN.vbKeyF3 Then
        Text1.Visible = True
        Text1.SetFocus
    End If
    If KeyAscii = VBRUN.vbKeyF4 Then
        Call Command2_Click
    End If
    If KeyAscii = VBRUN.vbKeyEscape Then
        Unload Me
    End If
End Sub

Private Sub Command1_Click()
    Module1.boolInsert = True
    Call Insert
    frmTblInsert.Show (vbModal)
    Call Form_Load
    If Not RS.EOF Then RS.MoveLast
    Dim obj As Object

```

```
DataGrid1.Refresh
Module1.boolcolinitialized = False
```

```
End Sub
```

```
Public Sub Insert()
```

```
    Dim boolaristera As Boolean
    boolaristera = True
    Dim xdifference As Integer
    Dim ydifference As Integer
    Dim intxLabel1 As Integer
    Dim intyLabel1 As Integer
    Dim intxText1 As Integer
    Dim intyText1 As Integer
    Dim intxCommand2 As Integer
    Dim intyCommand2 As Integer
    Dim intFormHeight As Integer
    Dim intTabIndex As Integer
    intFormHeight = 0
    intTabIndex = 0
    xdifference = frmTblInsert.Label1(1).Left - frmTblInsert.Label1(0).Left
    ydifference = frmTblInsert.Label1(2).Top - frmTblInsert.Label1(0).Top
    intxLabel1 = frmTblInsert.Label1(0).Left
    intyLabel1 = frmTblInsert.Label1(0).Top
```

```
    intxText1 = frmTblInsert.Text1(0).Left
    intyText1 = frmTblInsert.Text1(0).Top
```

```
    intxCommand2 = frmTblInsert.Command2(0).Left
    intyCommand2 = frmTblInsert.Command2(0).Top
```

```
    'fx-go dynamika
    Dim field As ADODB.field
    Dim ind As Integer
    Set Module1.colFkeys = New VBA.Collection
    Set Module1.colTableNames = New VBA.Collection
    ind = 0
    Dim RS2 As ADODB.Recordset
    Dim intFunctionKey As Integer
    intFunctionKey = 2
    Dim intTzogos As Integer
    intTzogos = 0
    Set RS2 = New ADODB.Recordset
    Dim strName As String
    With RS2
        .ActiveConnection = Con
        .CursorLocation = adUseClient
        .CursorType = adOpenForwardOnly
        .LockType = adLockReadOnly
```

```
    End With
```

```
    For Each field In RS.fields
```

```

If field.Type = adLongVarChar Then
    On Error Resume Next
    If field.Name = Module1.colPrimaryKeys(Module1.strTableName) Then
        If Err.Number = 0 Then
            If Module1.boolInsert Then
                If Module1.strTableName = "BIBΛΙΑ" Then
                    frmTblInsert.Text1(ind).Text = ""
                    frmTblInsert.Text1(ind).Enabled = True
                Else
                    frmTblInsert.Text1(ind).Text = "(Αυτόματη Αρίθμηση)"
                    frmTblInsert.Text1(ind).Enabled = False
                End If
            Else
                frmTblInsert.Text1(ind).Enabled = False
            End If
        Else
            Err.Clear
        End If

        End If
        'fx-go dynamika
        RS2.Open "SELECT szReferencedObject,szReferencedColumn FROM
MyRelationships WHERE szColumn='" & field.Name & "' AND szObject='" &
Module1.strTableName & "'"
        Debug.Print "FNAME:" & field.Name
        If Not RS2.EOF Then
            frmTblInsert.Command2(ind).Visible = True
            frmTblInsert.Command2(ind).Caption = "F" & intFunctionKey & "-GO"
            frmTblInsert.Command2(ind).Width = 735
            frmTblInsert.Command2(ind).Left = intxCommand2
            frmTblInsert.Command2(ind).Top = intyCommand2
            frmTblInsert.Command2(ind).TabIndex = 100

            Module1.colFkeys.Add CStr(ind), CStr(intFunctionKey)
            intFunctionKey = intFunctionKey + 1

            strName = RS2("szReferencedObject")
            Module1.colTableNames.Add strName

        Else
            Module1.colTableNames.Add ("lala")
        End If
        intFormHeight = intFormHeight + 3 * ydifference
        frmTblInsert.Text2(ind).Text = ""
        If boolaristera Then
            frmTblInsert.Text2(ind).Left = intxText1
            frmTblInsert.Text2(ind).Top = intyText1
            frmTblInsert.Text2(ind).Visible = True
            frmTblInsert.Text2(ind).Height = 3 * frmTblInsert.Text2(ind).Height
            frmTblInsert.Text2(ind).Width = 3 * frmTblInsert.Text2(ind).Width
            frmTblInsert.Label1(ind).Caption = field.Name
            frmTblInsert.Label1(ind).AutoSize = True
            frmTblInsert.Label1(ind).Left = intxLabel1
            frmTblInsert.Label1(ind).Top = intyLabel1
            intyText1 = intyText1 + 3 * ydifference

```

```

intyLabel1 = intyLabel1 + 3 * ydifference
intyCommand2 = intyCommand2 + 3 * ydifference
boolaristera = False
Else
intxText1 = intxText1 - xdifference
intxLabel1 = intxLabel1 - xdifference
intxCommand2 = intxCommand2 - xdifference

intyText1 = intyText1 + ydifference
intyLabel1 = intyLabel1 + ydifference
intyCommand2 = intyCommand2 + ydifference

frmTblInsert.Text2(ind).Left = intxText1
frmTblInsert.Text2(ind).Top = intyText1
frmTblInsert.Text2(ind).Visible = True
frmTblInsert.Text2(ind).Height = 3 * frmTblInsert.Text2(ind).Height
frmTblInsert.Text2(ind).Width = 3 * frmTblInsert.Text2(ind).Width
frmTblInsert.Label1(ind).Caption = field.Name
frmTblInsert.Label1(ind).AutoSize = True
frmTblInsert.Label1(ind).Left = intxLabel1
frmTblInsert.Label1(ind).Top = intyLabel1
intyText1 = intyText1 + 3 * ydifference
intyLabel1 = intyLabel1 + 3 * ydifference
intyCommand2 = intyCommand2 + 3 * ydifference
boolaristera = True
End If
frmTblInsert.Label1(ind).Visible = True
frmTblInsert.Text2(ind).TabIndex = intTabIndex
intTabIndex = intTabIndex + 1
Else
intFormHeight = intFormHeight + ydifference
frmTblInsert.Text1(ind).Text = ""
On Error Resume Next
If field.Name = Module1.colPrimaryKeys(Module1.strTableName) Then
If Err.Number = 0 Then
If Module1.boolInsert Then
If Module1.strTableName = "BIBAIA" Then
frmTblInsert.Text1(ind).Text = ""
frmTblInsert.Text1(ind).Enabled = True
Else
frmTblInsert.Text1(ind).Text = "(Αυτόματη Αρίθμηση)"
frmTblInsert.Text1(ind).Enabled = False
End If
Else
frmTblInsert.Text1(ind).Enabled = False
End If
Else
Err.Clear
End If

End If
RS2.Open "SELECT szReferencedObject,szReferencedColumn FROM
MyRelationships WHERE szColumn="" & field.Name & "" AND szObject="" &
Module1.strTableName & ""
Debug.Print "FNAME:" & field.Name

```

```

If Not RS2.EOF Then
    frmTblInsert.Command2(ind).Visible = True
    frmTblInsert.Command2(ind).Caption = "F" & intFunctionKey & "-GO"
    frmTblInsert.Command2(ind).Width = 735
    frmTblInsert.Command2(ind).Left = intxCommand2
    frmTblInsert.Command2(ind).Top = intyCommand2
    frmTblInsert.Command2(ind).TabIndex = 100

    Module1.colFkeys.Add CStr(ind), CStr(intFunctionKey)
    intFunctionKey = intFunctionKey + 1

    strName = RS2("szReferencedObject")
    Module1.colTableNames.Add strName
Else
    Module1.colTableNames.Add ("lala")
End If
If field.Type = adDate Then
    frmTblInsert.Command2(ind).Visible = True
    frmTblInsert.Command2(ind).Caption = "F" & intFunctionKey & "-GO"
    frmTblInsert.Command2(ind).Width = 735
    frmTblInsert.Command2(ind).Left = intxCommand2
    frmTblInsert.Command2(ind).Top = intyCommand2
    frmTblInsert.Command2(ind).TabIndex = 100

    Module1.colFkeys.Add CStr(ind), CStr(intFunctionKey)
    intFunctionKey = intFunctionKey + 1

End If
frmTblInsert.Label1(ind).Visible = True
frmTblInsert.Label1(ind).AutoSize = True
frmTblInsert.Text1(ind).Visible = True

Debug.Print Module1.colTableNames(ind)
RS2.Close

If boolaristera Then
    frmTblInsert.Text1(ind).Left = intxText1
    frmTblInsert.Text1(ind).Top = intyText1
    frmTblInsert.Text1(ind).Visible = True
    frmTblInsert.Text1(ind).Width = 1935
    frmTblInsert.Label1(ind).Caption = field.Name
    frmTblInsert.Label1(ind).AutoSize = True
    frmTblInsert.Label1(ind).Left = intxLabel1
    frmTblInsert.Label1(ind).Top = intyLabel1

    intxText1 = intxText1 + xdifference
    intxLabel1 = intxLabel1 + xdifference
    intxCommand2 = intxCommand2 + xdifference
    boolaristera = False
Else
    frmTblInsert.Text1(ind).Left = intxText1
    frmTblInsert.Text1(ind).Top = intyText1
    frmTblInsert.Text1(ind).Visible = True
    frmTblInsert.Text1(ind).Width = 1935
    frmTblInsert.Label1(ind).Caption = field.Name

```

```

frmTblInsert.Label1(ind).AutoSize = True
frmTblInsert.Label1(ind).Left = intxLabel1
frmTblInsert.Label1(ind).Top = intyLabel1

intxText1 = intxText1 - xdifference
intxLabel1 = intxLabel1 - xdifference
intxCommand2 = intxCommand2 - xdifference
intyText1 = intyText1 + ydifference
intyLabel1 = intyLabel1 + ydifference
intyCommand2 = intyCommand2 + ydifference
boolaristera = True
End If
frmTblInsert.Text1(ind).TabIndex = intTabIndex
intTabIndex = intTabIndex + 1

End If
ind = ind + 1
Next
frmTblInsert.ind = ind - 1
Dim obj As Object
Dim a1 As Double
Dim b1 As Double
Dim c1 As Integer
c1 = ind / 2
b1 = ind
a1 = c1 * 2

If Not a1 = b1 Then
c1 = c1 + 1
End If
If Not boolaristera Then
intyLabel1 = intyLabel1 + ydifference
End If
frmTblInsert.Command1(0).Top = intyLabel1 + 300
frmTblInsert.Command1(0).TabIndex = intTabIndex
intTabIndex = intTabIndex + 1

frmTblInsert.Command1(1).Top = intyLabel1 + 300
frmTblInsert.Command1(1).TabIndex = intTabIndex
intTabIndex = intTabIndex + 1

intyLabel1 = intyLabel1 + frmTblInsert.Command1(1).Height
frmTblInsert.Height = intyLabel1 - frmTblInsert.Top + 2500

End Sub

Public Sub Edit()
Dim i As Integer
Module1.boolInsert = False
Call Insert
For i = 0 To frmTblInsert.ind
If RS.fields(i).Type = ADODB.adLongVarChar Then
frmTblInsert.Text2(i) = DataGrid1.Columns(i)
Else
frmTblInsert.Text1(i) = DataGrid1.Columns(i)

```

```

If Module1.strTableName = "ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ" Then
    frmTblInsert.Text1(i).Locked = True
    frmTblInsert.Command1(0).Visible = False
End If
If Module1.strTableName = "ΠΛΗΡΩΘΕΝΤΑ" Then
    frmTblInsert.Text1(i).Locked = True
    frmTblInsert.Command1(0).Visible = False
End If
If Module1.strTableName = "ΛΟΓΑΡΙΑΣΜΟΙ" Then
    frmTblInsert.Text1(i).Locked = True
    frmTblInsert.Command1(0).Visible = False
End If
End If
Next i
frmTblInsert.Show vbModal

End Sub
Private Sub Command3_Click()
If Not RS.EOF Then
Dim a As Long

a = CLng(RS.Bookmark)

    Call Edit
    Call Form_Load

    RS.Move a - 1
    DataGrid1.Refresh

End If

End Sub
Private Sub Form_Resize()
    DataGrid1.Height = Me.ScaleHeight - 1200
    DataGrid1.Width = Me.ScaleWidth
    Text1.Top = Me.ScaleHeight - 800
    Command1.Top = Me.ScaleHeight - 500
    Command2.Top = Me.ScaleHeight - 500
    Command3.Top = Me.ScaleHeight - 500
    Command4.Top = Me.ScaleHeight - 500
    Command5.Top = Me.ScaleHeight - 1150
    Combo1.Top = Me.ScaleHeight - 1100
End Sub

Private Sub Text1_Change()
    Dim strSQLsearch As String
    strSQLsearch = "SELECT * FROM [" & Module1.strTableName & "] WHERE ([" &
    Combo1.Text & "] LIKE '" & VBA.Trim(Text1.Text) & "%' OR [" & Combo1.Text & "]
    LIKE '%" & VBA.Trim(Text1.Text) & "%'"

```

```

Set RS = New ADODB.Recordset
With RS
    .ActiveConnection = Con
    .CursorLocation = adUseClient
    .CursorType = adOpenForwardOnly
    .LockType = adLockReadOnly
    .Open strSQLsearch
End With
Set DataGrid1.DataSource = RS
DataGrid1.Refresh
DataGrid1.MarqueeStyle = dbgHighlightRow

```

```

End Sub
Private Sub Form_Unload(Cancel As Integer)
    Module1.boolcolinitialized = False
    Set DataGrid1.DataSource = Nothing
    RS.Close
    Set RS = Nothing
End Sub

```

```

Private Sub Text1_KeyPress(KeyAscii As Integer)
    If KeyAscii = VBRUN.vbKeyF2 Then
        Call Command1_Click
    End If
    If KeyAscii = VBRUN.vbKeyF3 Then
        Call Command2_Click
    End If
    If KeyAscii = VBRUN.vbKeyF4 Then
        Call Command2_Click
    End If
    If KeyAscii = VBRUN.vbKeyF5 Then
        Call Command4_Click
    End If
    If KeyAscii = VBRUN.vbKeyEscape Then
        Unload Me
    End If

```

```

End Sub

```

frmTblInsert

```

Dim Con As ADODB.Connection
Public ind As Integer

```

```

Public colFkeys As Collection
Private Sub insert_update()
    Dim strFieldNames As String
    Dim strValues As String
    Dim strFVnames As String
    Module1.boolcommand1exited = False
    Dim lala As Integer
    Dim field As ADODB.field

    Dim err1 As Integer
    Dim indicator As Integer
    err1 = Module1.validateExpressions(Text1, frmGeneralTblFunctions.RS.fields)
    If Not err1 = -1 Then
        Module1.boolcommand1exited = True
        Exit Sub
    End If
    indicator = -1
    For lala = 0 To Me.ind
        If frmGeneralTblFunctions.RS.fields(lala).Type = ADODB.adLongVarChar Then
            If (Not Module1.colPrimaryKeys(Module1.strTableName) = Label1(lala).Caption) And
                (Text2(lala) <> "") Then
                indicator = lala
                Exit For
            End If
        Else
            If (Not Module1.colPrimaryKeys(Module1.strTableName) = Label1(lala).Caption) And
                (Text1(lala) <> "") Then
                indicator = lala
                Exit For
            End If
        End If
    Next
    If indicator = -1 Then
        MsgBox "Παρακαλώ Εισάγετε Τιμές"
        Module1.boolcommand1exited = True
        Exit Sub
    End If
    If Module1.strTableName <> "BIBΛΙΑ" Then
        strFieldNames = "[" & Label1(indicator).Caption & "]"
    Else
        strFieldNames = "[" & Label1(indicator - 1).Caption & "],[ " &
            Label1(indicator).Caption & "]"
    End If
    If frmGeneralTblFunctions.RS.fields(indicator).Type = ADODB.adLongVarChar Then
        strValues = "(" & Text2(indicator) & """"
        strFVnames = "[" & Label1(indicator) & "]" = "" & Text2(indicator) & """"
    Else
        If Module1.strTableName <> "BIBΛΙΑ" Then
            strValues = "(" & Text1(indicator) & """"
        Else
            strValues = "(" & Text1(indicator - 1) & "," & Text1(indicator) & """"
        End If

        strFVnames = "[" & Label1(indicator) & "]" = "" & Text1(indicator) & """"
    End If

```

```

End If
For i = indicator + 1 To ind

    If frmGeneralTblFunctions.RS.fields(i).Type = ADODB.adLongVarChar Then
        If (Text2(i) <> "") Then
            strFVnames = strFVnames & "[" & Label1(i) & "] = " & Text2(i) & """"
            strFieldNames = strFieldNames & "[" & Label1(i) & "]"
            strValues = strValues & "," & Text2(i) & """"
        End If
    Else
        If (Text1(i) <> "") Then
            strFVnames = strFVnames & "[" & Label1(i) & "] = " & Text1(i) & """"
            strFieldNames = strFieldNames & "[" & Label1(i) & "]"
            strValues = strValues & "," & Text1(i) & """"
        End If
    End If

Next
strFVnames = strFVnames & " "
strFieldNames = strFieldNames & " ) "
strValues = strValues & " ) "

Debug.Print strValues
Debug.Print strFieldNames
Debug.Print strFVnames

If err1 = -1 Then
    If Module1.boolInsert = True Then
        Call Module1.Insert(Me.Caption, strFieldNames, strValues, Con)
    Else
        If Module1.strTableName <> "ΒΙΒΛΙΑ" Then
            Call Module1.Update(Me.Caption, strFVnames, "[" & Me.Label1(0).Caption & "]=" &
                Me.Text1(0).Text, Con)
        Else
            Call Module1.Update(Me.Caption, strFVnames, "[" & Me.Label1(0).Caption & "]=" &
                Me.Text1(0).Text & """" , Con)
        End If
    End If

End If
End Sub

Private Sub Command1_Click(Index As Integer)
If Index = 0 Then
    el = 0
    For i = 1 To ind
        If Text1(i) = "" Then el = 1
    Next
    If Module1.strTableName = "ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ" Then
        If Text1(6) = "" Then
            MsgBox "Λάθος Συνολικό Ποσό ή Αρ.Δόσεων", , "ΜΗΝΥΜΑ"
            el = 2
        End If
    End If

```

```

End If
If Module1.strTableName = "ΛΟΓΑΡΙΑΣΜΟΙ" Then
    el = 0
    For i = 2 To ind
        If Text1(i) = "" Then el = 1
    Next
    If ((Text1(2).Text <> "") And (Text1(3).Text <> "")) Then
        Call Module1.Insert("ΛΟΓΑΡΙΑΣΜΟΙ",
"([ΛΟΓΑΡΙΑΣΜΟΣ],[ΑΙΤΙΟΛΟΓΙΑ_1],[ΑΙΤΙΟΛΟΓΙΑ_2])", "('ww','ww','ww')", Con)
        Dim rs3 As New ADODB.Recordset
        With rs3
            .ActiveConnection = Con
            .CursorLocation = adUseClient
            .CursorType = adOpenDynamic
            .LockType = adLockReadOnly
        End With
        rs3.Open "SELECT [ΑΝΑΓΝΩΡΙΣΤΙΚΟ] FROM [ΛΟΓΑΡΙΑΣΜΟΙ]"
        rs3.MoveLast
        boolInsert = False
        Text1(0).Text = rs3(0).Value
        If Text1(2).Text = "1" Then
            Text1(1).Text = "98.000.001." & rs3(0).Value
        Else
            If Text1(2).Text = "2" Then
                Text1(1).Text = "98.000.002." & rs3(0).Value
            End If
        End If
        rs3.Close
    Else
        el = 1
    End If
End If
With rs3
    .ActiveConnection = Con
    .CursorLocation = adUseClient
    .CursorType = adOpenDynamic
    .LockType = adLockReadOnly
End With
If Module1.strTableName = "ΕΣΟΔΑ_ΕΞΟΔΑ" Then
    If el = 0 Then

        If (Text1(1) < 10000) Then
            rs3.Open "SELECT [ΛΟΓΑΡΙΑΣΜΟΣ] FROM [ΛΟΓΑΡΙΑΣΜΟΙ] WHERE
([ΑΝΑΓΝΩΡΙΣΤΙΚΟ] = " & Text1(1).Text & ")"
            Text1(1).Text = rs3(0).Value
            rs3.Close
        End If
    End If
End If
If Module1.strTableName = "ΒΙΒΛΙΑ" Then

    If (el = 0) Then
        If (IsNumeric(Text1(2).Text)) Then

```

```

        rs3.Open "SELECT [ΣΥΓΓΡΑΦΕΑΣ] FROM [ΣΥΓΓΡΑΦΕΙΣ] WHERE
([ΚΩΔ_ΣΥΓΓΡΑΦΕΑ] = " & Text1(2).Text & ")"
        Text1(2).Text = rs3(0).Value
        rs3.Close
    End If
    If (IsNumeric(Text1(3))) Then
        rs3.Open "SELECT [ΕΚΔΟΣΕΙΣ] FROM [ΕΚΔΟΣΕΙΣ] WHERE
([ΚΩΔ_ΕΚΔΟΣΕΩΝ] = " & Text1(3).Text & ")"
        Text1(3).Text = rs3(0).Value
        rs3.Close
    End If
    End If
    End If
    If Module1.strTableName = "ΛΟΓΑΡΙΑΣΜΟΙ" Then

        If (e1 = 0) Then
            If (IsNumeric(Text1(2).Text)) Then

                rs3.Open "SELECT [ΑΙΤΙΟΛΟΓΙΑ] FROM [ΑΙΤΙΟΛΟΓΙΑ_1] WHERE
([ΑΝΑΓΝΩΡΙΣΤΙΚΟ] = " & Text1(2).Text & ")"
                Text1(2).Text = rs3(0).Value
                rs3.Close
            End If
            End If
            End If

            If e1 = 1 Then MsgBox "Ελληνική Στοιχεία", , "ΜΗΝΥΜΑ"
            If e1 = 0 Then
                If boolInsert And boolExtraTablesupdated Then
                    boolInsert = False
                    Call insert_update
                    boolInsert = True
                Else
                    Call insert_update
                End If
                If Not Module1.boolcommand1exited Then
                    Unload Me
                End If
            End If
        Else

            If Module1.boolExtraTablesupdated And boolInsert Then
                Dim obj As Collection
                For Each obj In Module1.colExtraTables
                    Module1.Delete Module1.strTableName, "[" & obj.Item(5) & "]" = " &
frmExtraTables.lngExtraId, "", MDIForm1.Con
                    Module1.Delete obj.Item(3), "[" & obj.Item(5) & "]" = " & frmExtraTables.lngExtraId,
"", MDIForm1.Con
                Next
            End If

            Unload Me
        End If
    End If

```

```

End Sub

Private Sub Command2_Click(Index As Integer)
    If Index < frmGeneralTblFunctions.RS.fields.Count Then
        If frmGeneralTblFunctions.RS.fields(Index).Type = adDate Then
            Form1.Show vbModal
            Text1(Index).Text = Module1.strdate

            Exit Sub
        End If
    End If
    Module1.strTableName1 = Module1.colTableNames(Index + 1)
    frmGeneralTblFunctions1.Caption = Module1.colTableNames(Index + 1)
    frmGeneralTblFunctions1.Caption = Module1.colTableNames(Index + 1)
    Module1.intIndex = Index
    frmGeneralTblFunctions1.Show vbModal
End Sub

Private Sub Form_KeyPress(KeyAscii As Integer)
    If KeyAscii = VBRUN.vbKeyEscape Then
        Call Command1_Click(1)
    End If
End Sub

Private Sub Form_Load()
    Dim strMDB As String
    Me.Caption = Module1.strTableName
    Me.Move 100, 1500
    Set Con = MDIForm1.Con
    If Module1.strTableName = "ΒΑΘΜΟΙ" Then
        Text1(1).Locked = True
    End If
    If Module1.strTableName = "ΕΓΓΡΑΦΗ" Then
        Text1(1).Locked = True
        Text1(3).Locked = True
        Text1(4).Locked = True
        Text1(5).Locked = True
    End If
    If Module1.strTableName = "ΣΤΟΙΧΕΙΑ ΜΑΘΗΤΗ" Then
        Text1(5).Locked = True
    End If
    If Module1.strTableName = "ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ" Then
        Text1(1).Locked = True
        Text1(2).Locked = True
        Text1(6).Locked = True
    End If
    If Module1.strTableName = "ΠΛΗΡΩΘΕΝΤΑ ΔΙΔΑΚΤΡΑ" Then
        Text1(1).Locked = True
        Text1(3).Locked = True
        Text1(5).Locked = True
    End If
    If Module1.strTableName = "ΑΠΟΥΣΙΕΣ" Then
        Text1(1).Locked = True
        Text1(3).Locked = True
    End If

```

```

If Module1.strTableName = "ΔΙΟΙΚΗΤΙΚΟ ΠΡΟΣΩΠΙΚΟ" Then
    Text1(6).Locked = True
End If
If Module1.strTableName = "ΚΑΘΗΓΗΤΕΣ" Then
    Text1(5).Locked = True
End If
If Module1.strTableName = "ΥΠΑΛΛΗΛΟΙ" Then
    Text1(6).Locked = True
End If

If Module1.strTableName = "ΛΟΓΑΡΙΑΣΜΟΙ" Then
    Text1(1).Locked = True
    Text1(2).Locked = True
End If
If Module1.strTableName = "ΕΣΟΔΑ_ΕΞΟΔΑ" Then
    Text1(1).Locked = True
    Text1(2).Locked = True
End If
If Module1.strTableName = "ΒΙΒΛΙΑ" Then
    Text1(2).Locked = True
    Text1(3).Locked = True
End If
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Set Module1.colTableNames = Nothing

End Sub

Private Sub Text1_Change(Index As Integer)
    Dim el As Integer
    If (Index = 3 Or Index = 4) Then
        If Module1.strTableName = "ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ" Then
            If (IsNumeric(Text1(3).Text) And IsNumeric(Text1(4).Text)) Then
                If (Text1(3).Text > 0 And Text1(3).Text < 1501) And (Text1(4).Text < 19 And
                Text1(4).Text > 0) Then
                    Text1(6).Text = CStr(CInt(Text1(3).Text) / CInt(Text1(4).Text))
                Else
                    Text1(6).Text = ""
                End If
            End If
        End If
    End If
    el = 0
    If (Index = 4) Then
        Dim rs3 As New ADODB.Recordset
        With rs3
            .ActiveConnection = Con
            .CursorLocation = adUseClient
            .CursorType = adOpenDynamic
            .LockType = adLockReadOnly
        End With
    End If
End Sub

```

```

End With
If Module1.strTableName = "ΠΙΛΗΡΩΘΕΝΤΑ ΔΙΔΑΚΤΡΑ" Then
On Error GoTo view_error
If ((IsNumeric(Text1(4).Text)) And (Text1(1) <> "")) Then
rs3.Open "SELECT [ΣΥΝΟΛΙΚΟ ΠΟΣΟ] FROM [ΣΥΜΦΩΝΗΘΕΝΤΑ ΔΙΔΑΚΤΡΑ]
WHERE ([ΚΩΔ_ΜΑΘΗΤΗ] = " & Text1(1).Text & ")"
Text1(5).Text = rs3(0).Value
rs3.Close
rs3.Open "SELECT SUM([ΠΟΣΟ]) FROM [ΠΙΛΗΡΩΘΕΝΤΑ ΔΙΔΑΚΤΡΑ] WHERE
([ΚΩΔ_ΜΑΘΗΤΗ] = " & Text1(1).Text & ")"
If rs3(0).Value <> 0 Then
Text1(5).Text = CStr(CInt(Text1(5).Text) - CInt(rs3(0).Value + Text1(4).Text))
Else
Text1(5).Text = CStr(CInt(Text1(5).Text) - CInt(Text1(4).Text))
End If
rs3.Close
End If
End If
End If
Exit Sub
view_error:
MsgBox "Δεν Έχουν Καταχωρηθεί τα Συμφωνηθέντα Δίδακτρα του Μαθητή με ΚΩΔΙΚΟ "
& Text1(1).Text, vbCritical, "ΜΗΝΥΜΑ"
Call Form_Load
End Sub

```

```

Private Sub Text1_KeyDown(Index As Integer, KeyCode As Integer, Shift As Integer)
On Error GoTo errkey
If KeyCode >= VBRUN.vbKeyF2 And KeyCode <= VBRUN.vbKeyF12 Then
Call Command2_Click(Module1.colFkeys(KeyCode - 112))
End If
Exit Sub
errkey:
Err.Clear
End Sub

```

```

Private Sub Text1_KeyPress(Index As Integer, KeyAscii As Integer)
If KeyAscii = VBRUN.vbKeyEscape Then
Call Command1_Click(1)
End If
If KeyAscii = VBRUN.vbKeyReturn Then
Call Command1_Click(0)
End If
Exit Sub
errkey:
Err.Clear
End Sub

```

```

Private Sub Text2_KeyDown(Index As Integer, KeyCode As Integer, Shift As Integer)
On Error GoTo errkey
If KeyCode >= VBRUN.vbKeyF2 And KeyCode <= VBRUN.vbKeyF12 Then
Call Command2_Click(Module1.colFkeys(KeyCode - 112))
End If
Exit Sub
errkey:

```

```
Err.Clear
End Sub
```

```
Private Sub Text2_KeyPress(Index As Integer, KeyAscii As Integer)
If KeyAscii = VBRUN.vbKeyEscape Then
    Call Command1_Click(1)
End If
If KeyAscii = VBRUN.vbKeyReturn Then
    Call Command1_Click(0)
End If
Exit Sub
errkey:
Err.Clear
End Sub
```

frmGeneralTblFunctions1

```
Option Explicit
Public Con As ADODB.Connection
Public RS As ADODB.Recordset
```

```
Private Sub Command2_Click()
    Text1.Visible = True
    Combo1.Visible = True
    Dim field As ADODB.field
    For Each field In RS.fields
        Combo1.AddItem field.Name
    Next
    Combo1.Text = Combo1.List(0)
    Text1.SetFocus
End Sub
```

```
Private Sub Command4_Click()
If MsgBox("Επιβεβαίωση Διαγραφής", vbOKCancel) = vbOK Then
Call Module1.Delete(Me.Caption, RS.fields(0).Name & "=" & DataGrid1.Columns(0), " ",
Con)
Call Form_Load
End If
End Sub
```

```
Private Sub Command5_Click()
If Not RS.EOF Then
Module1.intRecordCode = VBA.CInt(DataGrid1.Columns(0).Text)
frmTblInsert.Text1(Module1.intIndex).Text = DataGrid1.Columns(0).Text
Unload Me
End If
End Sub
```

```
Private Sub Form_Activate()
Call Form_Load
End Sub
```

```

Private Sub Form_Load()
    WindowState = 0
    Me.Move 150, 1800

    Dim SqlSelect As String
    SqlSelect = "SELECT * FROM [" & Module1.strTableName1 & "]"

    Set RS = New ADODB.Recordset
    Set Con = MDIForm1.Con
    With RS
        .ActiveConnection = Con
        .CursorLocation = adUseClient
        .CursorType = adOpenForwardOnly
        .LockType = adLockReadOnly
        .Open SqlSelect
    End With
    Set DataGrid1.DataSource = RS
    DataGrid1.Refresh
    DataGrid1.MarqueeStyle = dbgHighlightRow
    Dim col As Object
    For Each col In DataGrid1.Columns
        col.Width = col.Width + 100
    Next

End Sub

Private Sub DataGridTitles()
    Dim intColId As Integer
    Dim strfield As ADODB.field
    intColId = 0
    For Each strfield In RS.fields
        DataGrid1.Columns(intColId).Caption = strfield.Name
    Next
End Sub

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyAscii = vbKeyReturn Then
        Call Command5_Click
    End If

    If KeyCode = vbKeyF2 Then
        Call Insert
        frmTblInsert.Show
    End If
    If KeyCode = vbKeyF3 Then
        Text1.Visible = True
        Text1.SetFocus
    End If
    If KeyCode = vbKeyF4 Then
        Call Insert
        Call Edit
    End If
End Sub

Private Sub DataGrid1_KeyDown(KeyCode As Integer, Shift As Integer)
    'checking if Fnctn keys pressed
    If KeyCode = vbKeyReturn Then
        Call Command5_Click
    End If
End Sub

```

```

End If

If KeyCode = vbKeyF2 Then
    Call Insert
    frmTblInsert.Show
End If
If KeyCode = vbKeyF3 Then
    Text1.Visible = True
    Text1.SetFocus
End If
If KeyCode = vbKeyF4 Then
    Call Edit
End If
End Sub
Public Sub DataGrid1_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        Call Command5_Click
    End If
    If KeyAscii = 27 Then
        Unload Me
    End If
End Sub
Private Sub Form_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        Call Command5_Click
    End If

    If KeyAscii = 27 Then
        Unload Me
    End If
End Sub
Private Sub Command1_Click()
    Call Insert
    frmTblInsert.Show vbModal
    Module1.boolInsert = True
End Sub
Public Sub Insert()
    Dim field As ADO.DB.field
    Dim ind As Integer
    ind = 0
    For Each field In RS.fields
        frmTblInsert.Label1(ind).Caption = field.Name
        frmTblInsert.Label1(ind).AutoSize = True
        frmTblInsert.Text1(ind).Text = ""
        frmTblInsert.Label1(ind).Visible = True
        frmTblInsert.Text1(ind).Visible = True
        ind = ind + 1
    Next

    frmTblInsert.Command1(0).Top = frmTblInsert.Label1(ind).Top +
frmTblInsert.Label1(ind).Height + 10
    frmTblInsert.Command1(1).Top = frmTblInsert.Label1(ind).Top +
frmTblInsert.Label1(ind).Height + 10
    frmTblInsert.Height = (frmTblInsert.Label1(0).Height + 5) * ind +
frmTblInsert.Command1(1).Height + 750

```

```

        frmTblInsert.ind = ind - 1
        Module1.boolInsert = True
End Sub
Public Sub Edit()
Dim i As Integer
Call Insert
For i = 0 To frmTblInsert.ind
frmTblInsert.Text1(i) = DataGrid1.Columns(i)
Next i
frmTblInsert.Show
Module1.boolInsert = False

End Sub
Private Sub Command3_Click()
    Call Edit
End Sub
Private Sub Form_Resize()
    DataGrid1.Height = Me.ScaleHeight - 1200
    DataGrid1.Width = Me.ScaleWidth
    Text1.Top = Me.ScaleHeight - 800
    Command2.Top = Me.ScaleHeight - 500
    Command5.Top = Me.ScaleHeight - 500
    Combo1.Top = Me.ScaleHeight - 1100
End Sub
Private Sub Text1_Change()
    Dim strSQLsearch As String
    strSQLsearch = "SELECT * FROM [" & Module1.strTableName1 & "] WHERE [" &
Combo1.Text & "] LIKE '%" & VBA.Trim(Text1.Text) & "%' "
    Dim strMDB As String

    Set RS = New ADODB.Recordset
    With RS
        .ActiveConnection = Con
        .CursorLocation = adUseClient
        .CursorType = adOpenForwardOnly
        .LockType = adLockReadOnly
        .Open strSQLsearch
    End With
    Set DataGrid1.DataSource = RS
    DataGrid1.Refresh
    DataGrid1.MarqueeStyle = dbgHighlightRow
    Call DataGridTitles
End Sub
Private Sub Form_Unload(Cancel As Integer)
    Set DataGrid1.DataSource = Nothing
    RS.Close
    Set RS = Nothing

End Sub

```

Form1

```
Private Sub Calendar1_DateClick(ByVal DateClicked As Date)
    Module1.strdate = DateClicked
    Unload Me
End Sub
```

```
Private Sub Calendar1_LostFocus()
    Unload Me
End Sub
```

```
Private Sub Form_LostFocus()
    Unload Me
End Sub
```

Module1

Option Explicit

```
Public boolInsert As Boolean
Public strTableName As String
Public intRecordCode As Integer
Public strTableName1 As String
Public intIndex As Integer
Public colTableNames As Collection
Public colPrimaryKeys As Collection
Public boolGnrIFrmLoaded As Boolean
Public colFkeys As Collection
Public colExtraTables As Collection
Public insertionComplete As Boolean
Public boolcolinitialized As Boolean
Public lngExtraId As Long
Public boolcommand1exited As Boolean
Public boolExtraTablesupdated As Boolean
Public strdate As String
```

```
Public col69 As New Collection
```

```
Dim rs3 As New ADODB.Recordset
Dim rs4 As New ADODB.Recordset
```

```
Public Function Insert(strTblName As String, strFieldNames As String, strValues As String,
Con As ADODB.Connection)
```

```
    Dim sqlinsert As String
    If strValues = "" Then strValues = "()"
    sqlinsert = "Insert INTO [" & strTblName & "] " & _
strFieldNames & " VALUES " & strValues
    Debug.Print sqlinsert
    Con.BeginTrans
    Con.Execute (sqlinsert)
```

```

If Err.Number <> 0 Then
    Con.RollbackTrans
    Err.Clear
    Insert = 1
Else
    Con.CommitTrans
    Insert = 0
End If

```

End Function

```

Public Function Update(strTblName As String, strFVnames As String, strID As String, Con
As ADODB.Connection) As Integer

```

```

    Dim sqlupdate As String
    sqlupdate = "UPDATE [" & strTblName & "] SET " & strFVnames & " WHERE " &
strID
    Debug.Print sqlupdate
    Con.BeginTrans
    Con.Execute (sqlupdate)
    If Err.Number <> 0 Then
        Con.RollbackTrans
        Err.Clear
        Update = 1
    Else
        Con.CommitTrans
        Update = 0
    End If

```

End Function

```

Public Function Delete(strTblName As String, strFieldNames As String, strValues As String,
Con As ADODB.Connection) As Integer

```

```

    Dim sqlDelete As String

    sqlDelete = "DELETE [" & strTblName & "].* FROM [" & strTblName _
& "] WHERE " & strFieldNames
    Debug.Print sqlDelete
    Con.BeginTrans
    Con.Execute (sqlDelete)
    If Err.Number <> 0 Then
        Con.RollbackTrans
        Err.Clear
        Delete = 1
    Else
        Con.CommitTrans
        Delete = 0
    End If

```

End Function

```
Public Function validateExpressions(txtBoxes As Variant, fields As ADODB.fields)
```

```
    Dim ind As Integer
    Dim txtBox As TextBox
    ind = 0
    Debug.Print ADODB.adInteger, ADODB.adCurrency, ADODB.adDouble
    Dim field As ADODB.field
    For Each field In fields
    On Error Resume Next
    If txtBoxes(ind).Text <> "" Then
    If field.Name <> Module1.colPrimaryKeys(Module1.strTableName) Then
    If Err <> 0 Then Err.Clear
    On Error GoTo convert
        Select Case fields(ind).Type
        Case ADODB.adCurrency
            VBA.CCur (txtBoxes(ind).Text)
            txtBoxes(ind).Text = VBA.CStr(VBA.CCur(txtBoxes(ind).Text))
        Case ADODB.adInteger
            VBA.CInt (txtBoxes(ind).Text)
            txtBoxes(ind).Text = VBA.CStr(VBA.CInt(txtBoxes(ind).Text))
        Case ADODB.adDouble
            VBA.CDbl (txtBoxes(ind).Text)
            txtBoxes(ind).Text = VBA.CStr(VBA.CDbl(txtBoxes(ind).Text))
        Case Else
        End Select
    End If
    End If
    ind = ind + 1
Next
```

```
validateExpressions = -1
```

```
Exit Function
```

```
convert:
```

```
Dim strPrompt As String
```

```
strPrompt = "Λάθος Μορφή Εισαγόμενων Τιμών Στο Πεδίο " & fields(ind).Name
```

```
    Select Case fields(ind).Type
    Case ADODB.adCurrency
        strPrompt = strPrompt & ". Η Μορφή Πρεπει Να Είναι Νομισματική."
    Case ADODB.adInteger
        strPrompt = strPrompt & ". Η Μορφή Πρεπει Να Είναι Θετικών Ακεραίων."
    Case ADODB.adDouble
        strPrompt = strPrompt & ". Η Μορφή Πρεπει Να Είναι Θετικών Ακεραίων."
    Case Else
    End Select
```

```
MsgBox strPrompt, , "ΜΗΝΥΜΑ"
```

```
txtBoxes(ind).SetFocus
```

```
txtBoxes(ind).SelStart = 0
```

```
txtBoxes(ind).SelLength = 1000
```

```
validateExpressions = ind
```

```
End Function
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

Σχεσιακές Βάσεις Δεδομένων, Χρήστου Π. Σκουρλά, Νέων Τεχνολογιών 1992
Εισαγωγή στα Συστήματα Βάσεων Δεδομένων, C. J. Date, Κλειδάριθμος 1996
Ελληνική Access 2000, Celeste Robinson, Μ.Γκιούρδας 1999
Visual Basic 6, E. Petroutsos, Μ.Γκιούρδας 1999
Access DataBase Design & Programming, Steven Roman, Oreilly 1998
Visual Basic 6 DataBase How-To, MacMilan