ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ: ΜΕΛΕΤΗ ΚΑΙ ΚΑΤΑΓΡΑΦΗ ΑΠΑΙΤΗΣΕΩΝ ΣΥΣΤΗΜΑΤΟΣ ΒΔ ΜΙΚΡΩΝ ΕΠΙΧΕΙΡΗΣΕΩΝ ΚΑΙ ΚΑΤΑΓΡΑΦΗ ΠΡΟΤΕΙΝΟΜΕΝΗΣ ΛΥΣΗΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΔΒΔ

ΣΠΟΥΔΑΣΤΕΣ: ΜΙΧΑΛΟΠΟΥΛΟΣ ΓΙΩΡΓΟΣ – Α.Μ. 6229

ΤΟΥΜΑΖΑΤΟΣ ΧΡΗΣΤΟΣ - Α.Μ. 6241

ΕΙΣΗΓΗΤΗΣ: κ. ΓΙΟΥΡΟΥΚΟΣ ΝΙΚΟΣ

ΠΑΤΡΑ – ΙΟΥΛΙΟΣ 2005

ΠΕΡΙΕΧΟΜΕΝΑ

1 ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ ΚΕΦΑΛΑΙΩΝ ΠΤΥΧΙΑΚΗΣ

Στην πτυχιακή εργασία που δημιουργήσαμε με θέμα τη ΜΕΛΕΤΗ ΚΑΙ ΚΑΤΑΓΡΑΦΗ ΑΠΑΙΤΗΣΕΩΝ ΕΝΟΣ ΣΥΣΤΗΜΑΤΟΣ ΒΔ ΜΙΚΡΩΝ ΕΠΙΧΕΙΡΗΣΕΩΝ ΚΑΙ ΚΑΤΑΓΡΑΦΗ ΜΙΑΣ ΠΡΟΤΕΙΝΟΜΕΝΗΣ ΛΥΣΗΣ (ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΔΒΔ) δημιουργήσαμε τα ακόλουθα κεφάλαια:

- Στο 2° κεφάλαιο κάνουμε μια μικρή εισαγωγή στις αρχές των βάσεων δεδομένων όπου περιγράφουμε τα χαρακτηριστικά τους, τα πλεονεκτήματα και τα μειονεκτήματα που παρουσιάζουν και περιγράφουμε τα βασικά χαρακτηριστικά των οντοτήτων, των πινάκων και των σχέσεων μεταξύ τους και δίνουμε κάποια παραδείγματα σχέσεων. Τέλος περιγράφουμε τη διαδικασία της κανονικοποίησης και τις κανονικές μορφές των πινάκων

- Στο 3° κεφάλαιο περιγράφουμε την SQL. Συγκεκριμένα περιγράφουμε τους τύπους δεδομένων που χρησιμοποιεί, αναφέρουμε με παραδείγματα τις εντολές διαχείρισης βάσεων δεδομένων, πινάκων και εγγραφών. Τέλος περιγράφουμε αναλυτικά όλους τους βασικούς τύπος ερωτημάτων (απλά, αριθμητικά και δευτερεύοντα) και περιγράφουμε τους μηχανισμούς αποθήκευσης στη MYSQL, τις συναλλαγές και τα προβλήματα της ταυτόχρονης χρήσης εγγραφών.

- Στο 4° κεφάλαιο περιγράφουμε το σκοπό δημιουργίας της πτυχιακής εργασίας

- Στο 5° κεφάλαιο περιγράφουμε τη μικρομεσαία επιχείρηση για την οποία κατασκευάσαμε το πληροφοριακό σύστημα σε MYSQL καθώς επίσης τις απαιτήσεις και τις ανάγκες της έτσι όπως μας τις περιέγραψαν οι υπεύθυνοι της

- Στο 6° κεφάλαιο περιγράφουμε τις οντότητες της μικρομεσαία επιχείρησης, τις συσχετίσεις ανάμεσα τους και σχεδιάζουμε το διάγραμμα Οντοτήτων –Συσχετίσεων που περιγράφει την επιχείρηση
- Στο 7° κεφάλαιο δημιουργούμε τη βάση δεδομένων για τη μικρομεσαία επιχείρηση και όλους τους πίνακες που την αποτελούν
- Στο 8° κεφάλαιο δημιουργούμε όλα τα απαραίτητα ερωτήματα που εξάγουν πληροφορίες από τους πίνακες της βάσης δεδομένων της μικρομεσαίας επιχείρησης
- Στο 9° κεφάλαιο περιγράφουμε την ασφάλεια δεδομένων
- Στο 10° κεφάλαιο παρουσιάζουμε τα συμπεράσματα μας από την πτυχιακή εργασίας

2 ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Ολοένα και περισσότερο η καθημερινή μας ζωή επηρεάζεται από κάποια βάση δεδομένων. Πίσω από κάθε μεγάλο όγκο δεδομένων και πληροφορικών υπάρχει πάντοτε μια, εμφανής ή όχι, βάση δεδομένων.

Βάση Δεδομένων είναι μια συλλογή δεδομένων τα οποία σχετίζονται μεταξύ τους με υψηλό βαθμό οργάνωσης. Τα δεδομένα δεν περιέχουν πλεονασμούς και ο τρόπος οργάνωσής τους είναι ανεξάρτητος από τις εφαρμογές που τα χρησιμοποιούν.

Όλα τα δεδομένα βρίσκονται σε μια "δεξαμενή" δεδομένων η οποία μπορεί να αποτελείται από έναν ή περισσότερους δίσκους.

Ο πλέον γνωστός τρόπος οργάνωσης και αποθήκευσης στοιχείων είναι με τη μορφή ενός αρχείου. Επειδή υπάρχουν πολλών ειδών αρχεία, εδώ το αρχείο ορίζεται γενικά σαν μια συλλογή ομοειδών στοιχείων.

Η Βάση Δεδομένων λοιπόν περιέχει ένα σύνολο από αρχεία. Τα αρχεία αυτά πρέπει όσο είναι δυνατόν να μην περιέχουν περιττά επαναλαμβανόμενα στοιχεία. Ορισμένα στοιχεία επιβάλλεται να επαναλαμβάνονται σε διαφορετικά αρχεία για να μπορούν να συνδυάζονται μεταξύ τους.

Ως παράδειγμα μιας βάσης δεδομένων μπορούμε να αναφέρουμε τα στοιχεία (ονόματα, διευθύνσεις και τηλέφωνα) τα οποία αφορούν τα πρόσωπα που περιλαμβάνει η προσωπική μας ατζέντα. Βάση Δεδομένων μπορούν να αποτελέσουν και τα φορολογικά στοιχεία των ελλήνων πολιτών, όπως και τα στοιχεία που αφορούν τα βιβλία και την διαχείριση μιας βιβλιοθήκης. Επίσης βάση δεδομένων μπορούν αν αποτελέσουν τα στοιχεία ενός οργανισμού, όπως αυτά που αναφέρονται στους εργαζόμενους, το λογιστήριο, την αποθήκη και γενικά στις δραστηριότητες του οργανισμού.

Για την οργάνωση και τη διαχείριση βάσεων δεδομένων μέσω ηλεκτρονικού υπολογιστή απαιτείται εξειδικευμένο λογισμικό. Η συλλογή των προγραμμάτων, που επιτρέπουν την δημιουργία και τη διαχείριση βάσεων δεδομένων, ονομάζεται Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS-ΣΔΒΔ). Το ΣΔΒΔ συνεργάζεται στενά με το λειτουργικό σύστημα του υπολογιστή, στις δυνατότητες του οποίου και στηρίζεται, ώστε να πραγματοποιήσει τους στόχους του.

Οι κύριοι στόχοι ενός ΣΔΒΔ είναι η συστηματική αποθήκευση, αναζήτηση και συντήρηση των δεδομένων που περιλαμβάνει μια βάση δεδομένων. Το ΣΔΒΔ επιτρέπει στους διάφορους χρήστες μιας βάσης δεδομένων να εκφράσουν τις απαιτήσεις τους για πληροφόρηση, χωρίς να απασχολούνται καθόλου με τον τρόπο φύλαξης της βάσης δεδομένων. Το ΣΔΒΔ λειτουργεί ως φίλτρο και διαμεσολαβητής μεταξύ χρήστη και δεδομένων. Καμία πρόσβαση ή χρήση της βάσης δεδομένων δεν μπορεί να πραγματοποιηθεί με παράκαμψη του ΣΔΒΔ, τουλάχιστον από τους απλούς χρήστες.

Τα επιθυμητά χαρακτηριστικά ενός ΣΔΒΔ, επιπλέον εκείνων που το ορίζουν ως ΣΔΒΔ, είναι:

- η αποτελεσματικότητα
- η φιλικότητα στο χρήστη
- η ταχύτητα
- η ύπαρξη περιβάλλοντος ανάπτυξης εφαρμογών
- η διαχείριση εικόνων και κειμένων
- η γραφική απεικόνιση των αποτελεσμάτων
- η επικοινωνία με άλλα προγράμματα

2.1 ΠΛΕΟΝΕΚΤΗΜΑΤΑ - ΜΕΙΟΝΕΚΤΗΜΑΤΑ

Ένα ΣΔΒΔ (σύστημα διαχείρισης βάσεων δεδομένων) αναμένεται να διαθέτει ένα σύνολο από δυνατότητες στην οργάνωση και διαχείριση βάσεων δεδομένων οι οποίες αποτελούν και πλεονεκτήματα του ΣΔΒΔ έναντι των κλασικών εφαρμογών. Στη συνέχεια αναφέρουμε τις κυριότερες από αυτές:

- Έλεγχος Πλεονασμού Δεδομένων
- Κοινή Χρήση Δεδομένων
- Επιβολή Μηχανισμών Ασφαλείας και Ελέγχου της Πρόσβασης
- Επιβολή Κανόνων Ακεραιότητας: Μπορούν να οριστούν κανόνες σχετικά με τις τιμές που μπορούν να πάρουν τα δεδομένα και τις μεταξύ τους σχέσεις. Οι κανόνες ορίζονται στην φάση της δημιουργίας της βάσης δεδομένων από τους σχεδιαστές της.
- Επαναφορά της Βάσης Δεδομένων και Αντίγραφα Ασφαλείας.
- Πιο Γρήγορη Ανάπτυξη Εφαρμογών: Η ανάπτυξη εφαρμογών γίνεται γρήγορα λόγω των εργαλείων του περιβάλλοντος ανάπτυξης εφαρμογών και των ήδη οργανωμένων στοιχείων.
- Άμεσα Διαθέσιμες Ενημερωμένες Πληροφορίες: Η ύπαρξη της βάσης δεδομένων και οι ευκολίες που προσφέρει το ΣΔΒΔ ενθαρρύνουν τη συγκέντρωση στοιχείων και την ενημέρωσή τους.

Όπως είναι φυσικό, πέρα από τα πλεονεκτήματα που προσφέρει η χρησιμοποίηση των ΣΔΒΔ, υπάρχουν σε κάποιες περιπτώσεις και κάτω από ορισμένες προϋποθέσεις και κάποια μειονεκτήματα:

- Στην ανάλυση και το σχεδιασμό βάσεων δεδομένων απαιτείται προσωπικό με σύνθετες και εξειδικευμένες γνώσεις

- Χρειάζεται εξειδικευμένο λογισμικό, που έχει κάποιο σοβαρό κόστος, και απαιτείται εξειδικευμένο προσωπικό να το χρησιμοποιήσει
- Υπάρχουν αυξημένες απαιτήσεις σε υλικό. Χρειάζονται υπολογιστές με μεγάλη, κατά κανόνα, χωρητικότητα σε κεντρική και περιφερειακή μνήμη, μεγάλη ταχύτητα επεξεργασίας και πρόσβασης στα δεδομένα. Η ανάπτυξη της τεχνολογίας υλικού έχει δώσει ικανοποιητικές λύσεις σε αυτό τον τομέα, με το ανάλογο βέβαια κόστος
- Η ανάλυση και ο σχεδιασμός βάσεων δεδομένων απαιτεί χρόνο και χρήμα, στην αρχική τουλάχιστον φάση
- Στο σχεδιασμό μιας βάσης δεδομένων, συγκερασμός πολλών και διαφορετικών απαιτήσεων είναι δύσκολη υπόθεση και απαιτεί κόπο, χρόνο και ικανότητες από μέρους των σχεδιαστών της βάσης δεδομένων
- Οι έλεγχοι ακεραιότητας και ασφάλειας των δεδομένων, ο συγχρονισμός πρόσβασης σε αυτά και οι λειτουργίες επαναφοράς της βάσης δεδομένων, έχουν το ανάλογο κόστος στην απόδοση των εφαρμογών.

2.2 ΠΙΝΑΚΕΣ ΣΤΟ ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ

Ο πίνακας αποτελείται από ένα αριθμό γραμμών οι οποίες αντιστοιχούν στις εγγραφές (records) ενός αρχείου. Κάθε γραμμή αποτελείται από ένα αριθμό πεδίων τα οποία αντιστοιχούν στα πεδία (fields) της εγγραφής. Οι γραμμές ενός πίνακα αναφέρονται σαν πολλαπλότητες ή πλειάδες (tuples) ενώ οι στήλες του πίνακα σαν ιδιότητες (attribute).

Η ιδιότητα (πεδίο) είναι η μικρότερη μονάδα δεδομένων στο σχεσιακό μοντέλο. Κάθε ιδιότητα μιας σχέσης (πίνακα) παίρνει τιμές μέσα από ένα ορισμένο σύνολο τιμών. Άρα το σύνολο τιμών μιας ορισμένης ιδιότητας περιέχει όλες τις δυνατές τιμές που

μπορεί αυτή να πάρει. Όλα τα σύνολα τιμών που ορίζονται με τον τρόπο αυτό αποτελούν το πεδίο ορισμού της σχέσης.

Υπάρχουν ορισμένοι περιορισμοί στην κατασκευή ενός πίνακα. Αυτοί είναι οι ακόλουθοι:

1. Το όνομα ενός πίνακα πρέπει να είναι μοναδικό και να μην ταυτίζεται με το όνομα άλλου πίνακα
2. Οι γραμμές ενός πίνακα πρέπει να περιέχουν όλες τον ίδιο αριθμό πεδίων. Μερικά από τα πεδία είναι δυνατόν να μην έχουν περιεχόμενο.
3. Οι γραμμές ενός πίνακα πρέπει να είναι διαφορετικές μεταξύ τους. Δηλαδή πρέπει να διαφέρουν έστω και σε ένα πεδίο.
4. Η σειρά τοποθέτησης των γραμμών ή των στηλών δεν έχει καμία σημασία.
5. Κάθε στήλη ενός πίνακα πρέπει να έχει ένα όνομα. Το όνομα αυτό πρέπει να είναι μοναδικό και να μην ταυτίζεται με το όνομα άλλης στήλης του ίδιου πίνακα. Επιτρέπεται όμως διαφορετικοί πίνακες να έχουν τα ίδια ονόματα στηλών. Επίσης το όνομα μιας στήλης δεν πρέπει να ταυτίζεται με κάποιο από τα περιεχόμενά της.
6. Μια στήλη πρέπει πάντα να περιέχει δεδομένα του ίδιου τύπου. Για παράδειγμα αριθμούς, χαρακτήρες, ημερομηνίες κ.λ.π.

2.3 ΚΑΝΟΝΕΣ ΑΚΕΡΑΙΟΤΗΤΑΣ

- **Πρωτεύον Κλειδί** (primary key) ονομάζουμε κάποιο ή κάποια από τα πεδία του πίνακα τα οποία έχουν μοναδικές τιμές και μας επιτρέπουν να ξεχωρίσουμε μια εγγραφή του πίνακα από μια άλλη κατά μοναδικό τρόπο
- **Δευτερεύον Κλειδί** (secondary key) είναι το πρωτεύον κλειδί ενός άλλου πίνακα που έχει αντιγραφεί στον πίνακα που βρισκόμαστε

Όταν σχεδιάζουμε μια νέα Βάση Δεδομένων πρέπει να ικανοποιούνται οι ακόλουθοι κανόνες: (ονομάζονται και κανόνες ακεραιότητας)

- **Ο Πρώτος Κανόνας Ακεραιότητας ή Ακεραιότητα Οντότητας** (Entity Integrity) αναφέρει ότι κανένα από τα χαρακτηριστικά που συνθέτουν το κλειδί δεν μπορεί να έχει απροσδιόριστη τιμή. Κάθε διαφορετική τιμή του κλειδιού αντιστοιχεί σε κάποια διαφορετική οντότητα και δεν είναι λογικό να προσπαθούμε να καταγράψουμε πληροφορίες για κάποια οντότητα για κάποια οντότητα που δεν μπορούμε να προσδιορίσουμε.
- **Ο Δεύτερος Κανόνας Ακεραιότητας ή Ακεραιότητα Αναφοράς** (Referential Integrity) αναφέρει ότι για κάθε τιμή ενός ξένου κλειδιού μιας σχέσης Β, που αντιστοιχεί στο πρωτεύον κλειδί κάποιας σχέσης Α πρέπει:
  - ✓ είτε να συμφωνεί με την τιμή του πρωτεύοντος κλειδιού κάποιας πλειάδας στη σχέση Α
  - ✓ είτε να είναι απροσδιόριστη (null). Η απροσδιόριστη τιμή χρειάζεται για πληροφορίες που είτε δεν υπάρχουν, είτε δεν είναι διαθέσιμες τη δεδομένη στιγμή.

Οι **πράξεις μεταβολής** δεδομένων, **εισαγωγή**, **ενημέρωση** και **διαγραφή** θα πρέπει να ελέγχονται ώστε να μην παραβιάζουν τους κανόνες ακεραιότητας του μοντέλου. Τον έλεγχο για την τήρηση των κανόνων ακεραιότητας διεκπεραιώνει το αντίστοιχο ΣΔΒΔ.

2.4 ΟΝΤΟΤΗΤΕΣ

Οντότητα είναι ένα αντικείμενο ή μια έννοια που έχει αυτοτελή και πρωτεύουσα σημασία για τον οργανισμό που εξετάζουμε και για την οποία υπάρχει η ανάγκη να συγκεντρωθούν στοιχεία και πληροφορίες. Μια οντότητα θα πρέπει να μπορεί να διακριθεί από τις υπόλοιπες και περιγράφεται από ένα σύνολο ιδιοτήτων και χαρακτηριστικών. **Κάθε συγκεκριμένη τιμή μιας οντότητας ονομάζεται στιγμιότυπο.**

Ως παράδειγμα οντοτήτων μπορούμε να αναφέρουμε τις επενδύσεις, τα δάνεια, τις ασφάλειες, τους εργαζόμενους ή τα τμήματα που αναφέρονται σε ένα οργανισμό ή επιχείρηση. Ο εργαζόμενος που έχει το επώνυμο Παπαδόπουλος με όλα τα χαρακτηριστικά και τις σχέσεις του αποτελεί ένα στιγμιότυπο της οντότητας ΕΡΓΑΖΟΜΕΝΟΣ.

Τα στοιχεία που χρειάζεται να συγκεντρωθούν σχετικά με την περιγραφή και την πλήρη τεκμηρίωση μιας οντότητας είναι τα εξής:

1. Η περιγραφή του τι αντιπροσωπεύει
2. Οι εναλλακτικές ονομασίες ή τα ψευδώνυμα της οντότητας. Π.χ. για μια τράπεζα ή οντότητα Πελάτης είναι γνωστή και ως κάτοχος λογαριασμού.
3. Τα ονόματα των χαρακτηριστικών που συγκροτούν την οντότητα και των ιδιοτήτων της κ.λ.π.

2.5 ΣΥΣΧΕΤΙΣΕΙΣ

Ανάμεσα στα διάφορα σύνολα οντοτήτων υπάρχουν διάφορες σχέσεις και συγγένειες Αυτές οι σχέσεις ονομάζονται **συσχετίσεις**.

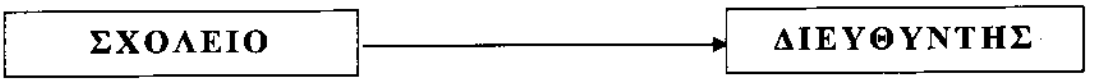
Η καλύτερη οργάνωση αυτών των οντοτήτων και ο ορισμός **σχέσεων** (**Relationship**) μεταξύ τους γίνεται με τον ορισμό των ιδιοτήτων (attributes) που θα περιέχουν. Για παράδειγμα κάθε είδος θα έχει ένα κωδικό, ένα όνομα (περιγραφή), χρώμα, βάρος και τιμή. Ένας προμηθευτής πρέπει να έχει ένα κωδικό ένα όνομα, μια διεύθυνση, ένα τηλέφωνο κ.λ.π. Κάποιο από τα πεδία αυτά πρέπει να χρησιμεύσει για κλειδί του πίνακα που θα προκύψει. Για το σκοπό αυτό αν δεν υπάρχει ένα κατάλληλο πεδίο, πρέπει εμείς να δημιουργήσουμε ένα. Για παράδειγμα με την εισαγωγή ενός κωδικού.

Ακολούθως πρέπει να ορισθούν οι σχέσεις μεταξύ των οντοτήτων. Οι πραγματικές σχέσεις των οντοτήτων μπορεί να είναι υπαρκτικές, λειτουργικές ή απλώς σχέσεις γεγονότων. Για παράδειγμα η οντότητα "ΥΠΑΛΛΗΛΟΣ΄" και η οντότητα "ΠΤΥΧΙΟ" συνδέονται με μια υπαρκτική σχέση. Ο υπάλληλος έχει (ή δεν έχει) ορισμένα πτυχία. Ενώ η οντότητα "ΥΠΑΛΛΗΛΟΣ" και η οντότητα "ΕΡΓΑΣΙΑ" συνδέονται με μια λειτουργική σχέση. Ο υπάλληλος εκτελεί κάποια ή ορισμένες εργασίες. Ακόμα οι οντότητες "ΑΠΟΘΗΚΗ" και "ΠΑΡΑΛΑΒΕΣ" συνδέονται με μια σχέση γεγονότων. Η αποθήκη δέχεται ορισμένες παραλαβές.

Οι σχέσεις αυτές για να χρησιμοποιηθούν πρέπει να μετατραπούν σε μαθηματικές της μορφής **1:1** (ένα προς ένα), **1:N** (ένα προς πολλά) ή **N:M** (πολλά προς πολλά). Για διευκρίνιση των παραπάνω ακολουθούν ορισμένα παραδείγματα.

Η σχέση της οντότητας "ΣΧΟΛΗ" και της οντότητας "ΔΙΕΥΘΥΝΤΗΣ" είναι 1:1 (ένα προς ένα) γατί κάθε τμήμα έχει μόνο ένα

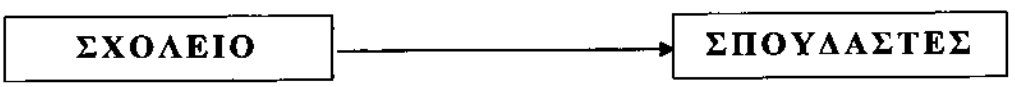
διευθυντή. Η σχέση 1:1 αναπαριστάνεται με ένα βέλος για παράδειγμα:



Εικόνα 1 - Σχέση 1:1

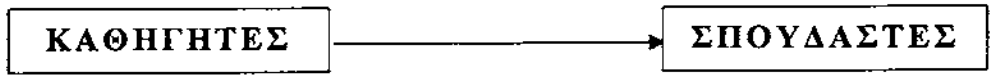
Αν έχουν ορισθεί και τα πεδία της οντότητας, τότε το βέλος συνδέει το κοινό πεδίο που χρησιμοποιείται για την σύνδεση των δύο αυτών οντοτήτων.

Αντίθετα η σχέση της οντότητας "ΣΧΟΛΗ" με την οντότητα "ΣΠΟΥΔΑΣΤΕΣ" είναι της μορφής **1:N** (ένα προς πολλά) γιατί ένα τμήμα μπορεί να έχει πολλούς σπουδαστές. Η σχέση **1:N** παρίσταται με ένα βέλος το οποίο διαθέτει δύο "μύτες" για παράδειγμα:



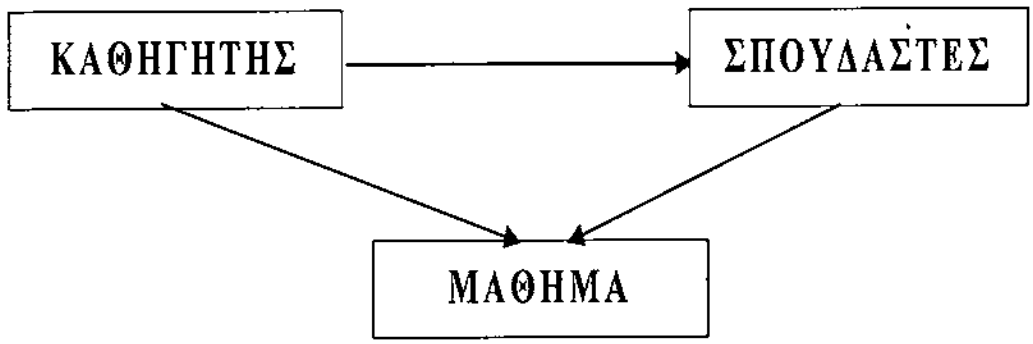
Εικόνα 2 - Σχέση 1:N

Η σχέση N:M (πολλά προς πολλά) είναι η πιο δύσκολη σχέση στην αναπαράσταση της. Για παράδειγμα η σχέση μεταξύ των οντοτήτων ΚΑΘΗΓΗΤΕΣ και ΣΠΟΥΔΑΣΤΗΣ είναι της μορφής N:M γιατί ένας καθηγητής διδάσκει (έχει) πολλούς σπουδαστές και αντίθετα ένας σπουδαστής διδάσκεται (έχει) από πολλούς Καθηγητές. Η σχέση N:M παρίσταται με ένα βέλος το οποίο έχει και από τις δύο άκρες του διπλές "μύτες" για παράδειγμα:



Εικόνα 3 - Σχέση N:M

Επειδή η σχέση αυτή είναι δύσκολη στην αναπαράσταση και το χειρισμό της, πρέπει να διασπασθεί σε δύο σχέσεις της μορφής 1:N. Η διάσπαση αυτή γίνεται με τον ορισμό μιας νέας ενδιάμεσης οντότητας. Στο παραπάνω παράδειγμα η οντότητα αυτή μπορεί να είναι το ΜΑΘΗΜΑ. Ένας καθηγητής διδάσκει πολλά μαθήματα (1:N) και ένας σπουδαστής παρακολουθεί πολλά μαθήματα (1:N). Παραστατικά η σχέση σημειώνεται:

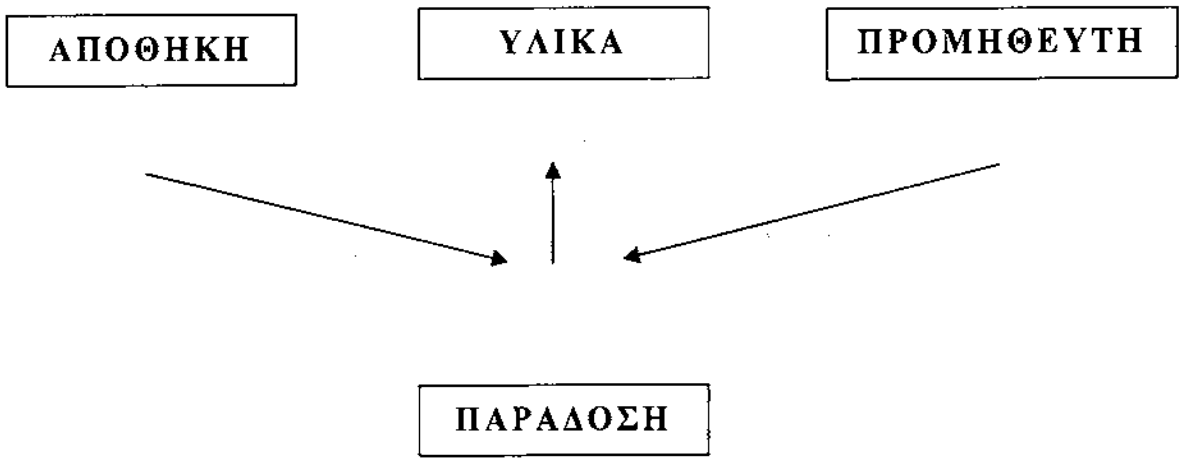


Εικόνα 4.

Εκτός από τις παραπάνω σχέσεις μπορεί να παρουσιασθεί το φαινόμενο μια σχέση να συνδέει τρεις ή και περισσότερες οντότητες. Μια τέτοια σχέση ονομάζεται **σύνθετη (complex relationship)**.

Για παράδειγμα μπορούμε να αναφέρουμε το εξής: Σε μια αποθήκη γίνεται συχνά παράδοση υλικών από τους προμηθευτές της εταιρείας, άρα έχουμε μια σύνθετη σχέση μεταξύ των οντοτή-

των ΑΠΟΘΗΚΗ, ΥΛΙΚΑ, ΠΡΟΜΗΘΕΥΤΗΣ. Μια τέτοια σχέση πρέπει να τη διασπάσουμε προσθέτοντας μία ακόμη σχετική οντότητα π.χ. την ονομάζουμε ΠΑΡΑΔΟΣΗ και ακολούθως εξετάζουμε τις σχέσεις των τριών πρώτων οντοτήτων ως προς τη νέα οντότητα.



Εικόνα 5.

Η σχέση ΑΠΟΘΗΚΗΣ και ΠΑΡΑΔΟΣΗΣ είναι **1:N**, διότι μια αποθήκη μπορεί να δεχθεί (την ίδια ημέρα) πολλές παραδόσεις. Η σχέση ΥΛΙΚΑ και ΠΑΡΑΔΟΣΗ αντίθετα είναι **N:1**, γιατί μια παράδοση μπορεί να περιλαμβάνει πολλά υλικά. Η σχέση της οντότητας ΠΡΟΜΗΘΕΥΤΗΣ και ΠΑΡΑΔΟΣΗ είναι **1:1**, αν δεχθούμε ότι ένας προμηθευτής σε μια ημέρα πραγματοποιεί μόνο μια παράδοση. Αν δεχθούμε ότι πραγματοποιεί πολλές, η σχέση είναι και εδώ 1:N.

Όταν γίνεται διάσπαση μιας οντότητας και προκύπτει μια "θυγατρική" οντότητα, ένας πραγματικός και χρήσιμος κανόνας αναφέρει ότι τα πεδία που δεν ανήκουν σε κάποιο πρωτεύον κλειδί πρέπει να παραμένουν στην ίδια "πατρική" οντότητα.

Για παράδειγμα: Θεωρούμε ότι η αρχική οντότητα είναι τα ΜΑΘΗΜΑΤΑ με κλειδί τον κωδικό μαθήματος και η θυγατρική οντότητα τα ΑΠΟΤΕΛΕΣΜΑΤΑ με σύνθετο κλειδί τον κωδικό του

μαθήματος και την ημερομηνία εξέτασης. Τότε το όνομα του μαθήματος, το οποίο δεν είναι κλειδί σε καμία οντότητα, δεν πρέπει να τοποθετηθεί στην οντότητα ΑΠΟΤΕΛΕΣΜΑΤΑ αλλά στην αρχική ΜΑΘΗΜΑΤΑ. Ένα άμεσο πλεονέκτημα της παραπάνω τοποθέτησης είναι ο περιορισμός αποθήκευσης επαναλαμβανόμενων στοιχείων. Το όνομα, αν ήταν στην θυγατρική οντότητα, θα έπρεπε να επαναλαμβάνεται για κάθε σπουδαστή που δίνει το σχετικό μάθημα. Με την τοποθέτησή του στην αρχική οντότητα γράφεται μόνο μία φορά και αν χρειάζεται μπορεί εύκολα να βρεθεί μέσω του κωδικού του.

Στην οριστικοποίηση των οντοτήτων και των πεδίων τους πολύ χρήσιμη είναι μια διαδικασία που ονομάζεται **κανονικοποίηση** δεδομένων και η οποία περιγράφεται παρακάτω, Σε κάθε περίπτωση δεν πρέπει να ξεχνάμε ότι σε κάθε οντότητα είμαστε υποχρεωμένοι να καθορίζουμε όλα τα κλειδιά.

Αρχικά βέβαια το πρωτεύον κλειδί ακολούθως τα εναλλακτικά και τέλος τα ξένα κλειδιά. Φυσικά ο ορισμός των κλειδιών πρέπει να ικανοποιεί τους δύο κανόνες ακεραιότητας.

Για τη μετατροπή των οντοτήτων σε πίνακες βασικό ρόλο παίζουν οι σχέσεις μεταξύ των οντοτήτων. Χρησιμοποιώντας τις παραπάνω αναφερόμενες σχέσεις θα προσπαθήσουμε να ορίσουμε τους βασικούς πίνακες.

2.6 ΚΑΝΟΝΙΚΟΠΟΙΗΣΗ

Η κανονικοποίηση αποτελεί ένα εργαλείο και οδηγό στην προσπάθεια μας για σχεδιασμό της βάσης δεδομένων. Βελτιώνει το σχεδιασμό της βάσης δεδομένων ώστε να περιοριστούν οι ανωμαλίες ενημέρωσης και αποτελεί ένα συστηματικό τρόπο να συμπεριλάβουμε στο σχεδιασμό της βάσης δεδομένων ένα μικρό τμήμα της σημασίας που έχουν τα δεδομένα στο φυσικό κόσμο. Η διαδικασία της κανονικοποίησης ελέγχει μία προς μία τις σχέσεις που θα αποτελέσουν τη βάση δεδομένων ώστε να ικανοποιούν κάποια κριτήρια.

Οι κανονικές μορφές είναι σχεδιασμένες ώστε:

- να προστατεύον από ανωμαλίες στην ενημέρωση των δεδομένων
- να ελαττώνουν τα πλεονάζοντα δεδομένα
- να αυξάνουν την ανεξαρτησία των δεδομένων
- να περιορίζουν τις πιθανότητες μη συμβατών δεδομένων
- να απομακρύνουν τις ανεπιθύμητες λογικές εξαρτήσεις

Στόχος της κανονικοποίησης είναι όλες οι σχέσεις που συμμετέχουν στην βάση δεδομένων να εντάσσονται στην **Τρίτη Κανονική μορφή**. Η πρώτη και η δεύτερη κανονική μορφή είναι ενδιάμεσα βήματα που βοηθούν στην υλοποίηση του τελικού στόχου.

- **Πρώτη κανονική μορφή** → Μια σχέση ανήκει στην **πρώτη κανονική μορφή** (ΠΚΜ) αν δεν περιέχει σύνθετα ή επαναλαμβανόμενα χαρακτηριστικά. Δηλαδή το κάθε χαρακτηριστικό περιέχει μία και μοναδική τιμή.

- **Δεύτερη κανονική μορφή** → Μια σχέση που βρίσκεται στη **δεύτερη κανονική μορφή** (ΔΚΜ) όταν έχει σύνθετο κλειδί, δηλαδή αποτελείται από περισσότερα από ένα χαρακτηριστικά.

- **Τρίτη κανονική μορφή** → Μια σχέση βρίσκεται στην **τρίτη κανονική μορφή** (ΤΚΜ) όταν βρίσκεται στη δεύτερη κανονική μορφή και τα χαρακτηριστικά που δεν αποτελούν μέρος του κλειδιού είναι ανεξάρτητα μεταξύ τους και αναφέρονται στο κλειδί και όχι σε άλλα χαρακτηριστικά. Με άλλα λόγια κάθε σχέση αποτελείται από το κλειδί, που ορίζει κάποια οντότητα και ένα σύνολο από χαρακτηριστικά, ανεξάρτητα μεταξύ τους, που περιγράφουν με κάποιο τρόπο την οντότητα.

3 SQL

Στο κεφάλαιο αυτό περιγράφουμε τη σύνταξη και τη λειτουργία όλων των βασικών εντολών της SQL και παρουσιάζουμε παραδείγματα εφαρμογής των εντολών στο περιβάλλων της MySQL.

3.1 ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ (DATA TYPES) ΣΤΗ MYSQL

Τύπος (Type)	Μέγεθος σε bytes	Περιγραφή
TINYINT (length)	1	Ακέραιοι με απρόσημες τιμές στο διάστημα από 0-255 και προσημασμένες τιμές στο διάστημα από -128 έως 127
SMALLINT (length)	2	Ακέραιοι με απρόσημες τιμές στο διάστημα από 0-65535 και προσημασμένες τιμές στο διάστημα από -32768-32767
MEDIUMINT(length)	3	Ακέραιοι με απρόσημες τιμές στο διάστημα από 0-16777215 και προσημασμένες τιμές στο διάστημα από -8388608-8388607
INT(length)	4	Ακέραιοι με απρόσημες τιμές στο διάστημα από 0-429467295 και προσημασμένες τιμές στο διάστημα από -2147483648-2147483647
BIGINT(length)	8	Ακέραιοι με απρόσημες τιμές στο διάστημα από 0-18446744 και προσημασμένες τιμές στο διάστημα από -9223372036854775808-9223372036854775807
FLOAT(length, decimal)	4	Αριθμοί κινητής υποδιαστολής με μέγιστη τιμή +/-3.402823466E38 και ελάχιστη τιμή (μη μηδενική) +/11.175494351E-38
DOUBLEPRECISION(length, decimal)	8	Αριθμοί κινητής υποδιαστολής με μέγιστη τιμή +/- -1.7976931348623157E308 και ελάχιστη τιμή (μη μηδενική) +/-2.2250738585072014E-308
DECIMAL(length, decimal)	length	Αριθμοί κινητής υποδιαστολής στην κλίμακα του τύπου DOUBLE που αποθηκεύονται ως τύπου CHAR
TIMESTAMP(length)	4	YYYYMMDDHHMMSS ή YYMMDDHHMMSS ή YYYYMMDD, YYMMDD. Μια τιμή Timestamp ενημερώνεται κάθε φορά που μια γραμμή αλλάζει τιμή. Μια τιμή NULL θέτει το πεδίο στην τρέχουσα ώρα

Τύπος (Type)	Μέγεθος σε bytes	Περιγραφή
DATE	3	YYYY-MM-DD
TIME	3	HH:MM:DD
DATETIME	8	YYYY-MM-DD HH:MM:SS
YEAR	1	YYYY ή YY
CHAR(length)	length	Μια συμβολοσειρά σταθερού μήκους (fixed length text string) όπου τα πεδία που είναι μικρότερα από το εξορισμού μέγεθος γεμίζουν με κενά διαστήματα στο τέλος (trailing spaces)
VARCHAR(length)	length	Μια συμβολοσειρά σταθερού μήκους (που έχει το πολύ 255 χαρακτήρες) όπου τα αχρησιμοποίητα κενά στο τέλος απομακρύνονται (καταργούνται) πριν την αποθήκευση.
TINYTEXT	length+1	Μια συμβολοσειρά που έχει το πολύ μήκος 255 χαρακτήρων.
TINYBLOB	length+1	Ένα δυαδικό πεδίο (binary field) που έχει το πολύ 255 χαρακτήρες.
TEXT	length+1	64Kb κειμένου
BLOB	length+1	64Kb δεδομένων
MEDIUMTEXT	length+3	16Mb κειμένου
MEDIUMBLOB	length+3	16 Mb δεδομένων
LONGTEXT	length+4	4GB κειμένου
LONGBLOB	length+4	4GB δεδομένων
ENUM	1,2	Αυτό το πεδίο μπορεί να περιέχει μια από 65535 πιθανές τιμές π.χ. ENUM('abc', 'def', 'ghi')
SET	1-8	Αυτός ο τύπος μπορεί να περιέχει ένα οποιοδήποτε αριθμό ενός συνόλου προκαθορισμένων πιθανών τιμών

3.2 ΕΝΤΟΛΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

3.2.1 ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Για να δημιουργήσουμε μια νέα βάση δεδομένων γράφουμε την ακόλουθη εντολή:

Εντολή	Τρόπος λειτουργίας
CREATE DATABASE όνομα	Δημιουργείται μια νέα βάση δεδομένων με το όνομα που προσδιορίζουμε

3.2.2 ΕΠΙΛΟΓΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Για να επιλέξουμε μια βάση δεδομένων που έχουμε δημιουργήσει γράφουμε την εντολή:

Εντολή	Τρόπος λειτουργίας
USE όνομα βάσης δεδομένων	Επιλέγεται η βάση δεδομένων που προσδιορίζουμε.

3.3 ΕΝΤΟΛΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΠΙΝΑΚΩΝ

ΔΗΜΙΟΥΡΓΙΑ ΠΙΝΑΚΑ

Για να δημιουργήσουμε ένα νέο πίνακα στην τρέχουσα βάση δεδομένων γράφουμε την ακόλουθη εντολή:

Εντολή	Τρόπος λειτουργίας
CREATE TABLE όνομα (πεδία πίνακα)	Δημιουργείται ένας νέος πίνακας με το όνομα και τα πεδία που προσδιορίζουμε στην τρέχουσα βάση δεδομένων.

ΕΝΤΟΛΗ ΤΡΟΠΟΠΟΙΗΣΗΣ ΠΙΝΑΚΩΝ

Για να τροποποιήσουμε τη δομή ενός πίνακα προσθέτοντας, αφαιρώντας, μετονομάζοντας πεδία κ.λ.π. γράφουμε την ακόλουθη εντολή:

Εντολή	Τρόπος λειτουργίας
ALTER TABLE όνομα πίνακα επιλογές - τροποποίησης	Τροποποιείται η δομή του πίνακα σύμφωνα με τις επιλογές που χρησιμοποιούμε

ΕΝΤΟΛΗ ΟΡΙΣΜΟΥ ΠΡΩΤΕΥΟΝΤΟΣ ΚΛΕΙΔΙΟΥ ΠΙΝΑΚΑ

Το πρωτεύον κλειδί ενός πίνακα είναι ένα πεδίο με μοναδικές τιμές. Για να ορίσουμε ένα ή περισσότερα πεδία ως πρωτείων κλειδί, γράφουμε την ακόλουθη εντολή:

Εντολή	Τρόπος λειτουργίας
PRIMARY KEY (στήλη ή στήλες)	Δημιουργούμε ως πρωτεύον κλειδί το πεδίο ή τα πεδία που βρίσκονται μέσα στην παρένθεση.

ΕΝΤΟΛΗ ΟΡΙΣΜΟΥ ΔΕΥΤΕΡΕΥΟΝΤΟΣ ΚΛΕΙΔΙΟΥ ΠΙΝΑΚΑ

Για να ορίσουμε ένα ή περισσότερα πεδία ως δευτερεύον κλειδί, γράφουμε την ακόλουθη εντολή:

Εντολή	Τρόπος λειτουργίας
FOREIGN KEY (στήλη ή στήλες) REFERENCES βασικός πίνακας	Δημιουργούμε ως δευτερεύον κλειδί το πεδίο ή τα πεδία που βρίσκονται μέσα στην παρένθεση και με την εντολή REFERENCES προσδιορίζουμε τον βασικό πίνακα από τον οποίο προέρχονται.

ΕΝΤΟΛΗ ΔΙΑΓΡΑΦΗΣ ΠΙΝΑΚΑ

Για να διαγράψουμε ένα πίνακα από μια βάση δεδομένων ο οποίος για κάποιο λόγο δε μας χρειάζεται γράφουμε την ακόλουθη εντολή:

Εντολή	Τρόπος λειτουργίας
DROP TABLE όνομα πίνακα	Διαγράφεται ο πίνακας που προσδιορίζουμε.

ΕΝΤΟΛΗ ΕΜΦΑΝΙΣΗΣ ΤΩΝ ΠΙΝΑΚΩΝ ΜΙΑΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Για να εμφανίσουμε όλους τους πίνακες μιας βάσης δεδομένων γράφουμε την ακόλουθη εντολή:

Εντολή	Τρόπος λειτουργίας
SHOW TABLES	Εμφανίζονται τα ονόματα όλων των πινάκων της τρέχουσας βάσης δεδομένων.

ΕΝΤΟΛΗ ΕΜΦΑΝΙΣΗΣ ΤΩΝ ΠΕΔΙΩΝ ΕΝΟΣ ΠΙΝΑΚΑ

Για να εμφανίσουμε όλα τα πεδία ενός πίνακα γράφουμε την ακόλουθη εντολή:

Εντολή	Τρόπος λειτουργίας
SHOW FULL COLUMNS FROM (όνομα πίνακα)	Εμφανίζονται όλα τα πεδία του πίνακα που προσδιορίζουμε.

3.4 ΕΝΤΟΛΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΔΕΔΟΜΕΝΩΝ ΠΙΝΑΚΩΝ

ΕΝΤΟΛΗ ΠΡΟΣΘΗΚΗΣ ΕΓΓΡΑΦΩΝ ΣΕ ΠΙΝΑΚΑ

Για να προσθέσουμε νέες εγγραφές σε ένα πίνακα υπάρχουν βασικά δυο τρόποι: Ο ένας είναι να προσθέτουμε μια εγγραφή κάθε φορά ενώ ο άλλος είναι να προσθέτουμε πολλές εγγραφές.

Η εντολή για να προσθέτουμε μια εγγραφή κάθε φορά είναι η ακόλουθη:

Εντολή	Τρόπος λειτουργίας
INSERT INTO όνομα-πίνακα (πεδίο1, πεδίο2, …) VALUES (τιμή1, τιμή2,…)	Προστίθεται στον πίνακα που προσδιορίζουμε η εγγραφή με τις συγκεκριμένες τιμές που γράφουμε. Με τον τρόπο αυτό μπορούμε να προσθέσουμε μόνο μια εγγραφή στον πίνακα 1

Η εντολή για να προσθέτουμε πολλές εγγραφές είναι η ακόλουθη:

Εντολή	Τρόπος λειτουργίας
INSERT INTO όνομα-πίνακα1 (πεδίο1, πεδίο2, …) SELECT πεδίο1, πεδίο2,…FROM όνομα πίνακα2	Προστίθενται στον πίνακα 1 όλες οι εγγραφές που ικανοποιούν τα κριτήρια από τον πίνακα 2. Με τον τρόπο αυτό μπορούμε να προσθέσουμε περισσότερες από μια εγγραφές στον πίνακα 1

ΕΝΤΟΛΗ ΔΙΑΓΡΑΦΗΣ ΕΓΓΡΑΦΩΝ ΑΠΟ ΠΙΝΑΚΑ

Για να διαγράψουμε εγγραφές από ένα πίνακα γράφουμε την ακόλουθη εντολή:

Εντολή	Τρόπος λειτουργίας
DELETE FROM όνομα πίνακα WHERE συνθήκη	Διαγράφονται από τον πίνακα που προσδιορίζουμε όλες οι εγγραφές που ικανοποιούν τη συνθήκη που γράφουμε

ΕΝΤΟΛΗ ΤΡΟΠΟΠΟΙΗΣΗΣ ΕΓΓΡΑΦΩΝ ΠΙΝΑΚΑ

Για να ενημερώσουμε τις εγγραφές ενός πίνακα γράφουμε την ακόλουθη εντολή:

Εντολή	Τρόπος λειτουργίας
UPDATE όνομα πίνακα SET εντολή καταχώρησης WHERE συνθήκη	Όλες οι εγγραφές του αναφερόμενου πίνακα που ικανοποιούν τη συνθήκη που γράφουμε ενημερώνονται σύμφωνα με την εντολή καταχώρησης που χρησιμοποιούμε.

3.5 ΕΝΤΟΛΕΣ ΕΠΙΛΟΓΗΣ ΕΓΓΡΑΦΩΝ

Μια από τις βασικές λειτουργίες της SQL είναι να θέτουμε κριτήρια στους πίνακες και να επιλέγουμε συγκεκριμένες εγγραφές. Αυτό γίνεται με την εντολή SELECT η οποία χρησιμοποιείται για επιλογή στοιχείων από μια βάση δεδομένων, για δημιουργία αντιγράφων πινάκων (copies of tables), για δημιουργία όψεων (create views), και για προσδιορισμό εγγραφών πινάκων προς ενημέρωση.

Η εντολή SELECT έχει την ακόλουθη σύνταξη (σε πλήρη μορφή):

Εντολή	Τρόπος λειτουργίας
SELECT λίστα πεδίων FROM λίστα πινάκων WHERE συνθήκες	Το αποτέλεσμα της εντολής SELECT είναι η επιλογή μιας λίστας δεδομένων από ένα σύνολο πινάκων της βάσης δεδομένων οι οποίοι προσδιορίζονται με την εντολή FROM. Η λίστα πεδίων που γράφεται δεξιά από την εντολή SELECT προσδιορίζει τα πεδία που επιλέγονται. Η εντολή GROUP BY χρησιμοποιείται όταν θέλουμε να συνοψίζουμε τα δεδομένα που επιλέγουμε και η εντολή HAVING χρησιμοποιείται για να προσδιορίσει ποιες ομάδες δεδομένων θέλουμε να συμπεριλαμβάνονται. Η εντολή ORDER BY χρησιμοποιείται για να ταξινομήσει τις εγγραφές των δεδομένων που επιλέγονται. Οι εντολές WHERE, GROUP BY, HAVING, ORDER BY, είναι προαιρετικές.
GROUP BY πεδία ομαδοποίησης	
HAVING έκφραση ομαδοποίησης	
ORDERED BY λίστα πεδίων;	

3.5.1 ΑΠΛΗ ΕΝΤΟΛΗ SELECT

Η εντολή SELECT στην πιο απλή της μορφή έχει την ακόλουθη σύνταξη:

Εντολή	Τρόπος λειτουργίας
SELECT λίστα πεδίων FROM λίστα πινάκων	Επιλέγονται όλες οι εγγραφές του αναφερόμενου πίνακα ή των αναφερόμενων πινάκων.

3.5.2 SELECT DISTINCT

Η εντολή SELECT μας επιτρέπει να συλλέξουμε όλες τις πληροφορίες από ένα ή περισσότερα πεδία ενός ή περισσότερων πινάκων. Αυτό φυσικά μπορεί να δημιουργήσει πλεονασμούς δεδομένων. Αν θέλουμε να επιλέξουμε μοναδικά στοιχεία τότε χρησιμοποιούμε την ακόλουθη εντολή:

Εντολή	Τρόπος λειτουργίας
SELECT DISTINCT λίστα πεδίων FROM λίστα πινάκων	Επιλέγονται οι μοναδικές εγγραφές του αναφερόμενου πίνακα ή των αναφερόμενων πινάκων.

3.5.3 SELECT WHERE

Η εντολή SELECT μας επιτρέπει να συλλέξουμε όλες τις πληροφορίες από ένα ή περισσότερα πεδία ενός ή περισσότερων πινάκων. Αν όμως θέλουμε να επιλέξουμε συγκεκριμένες εγγραφές από ένα η περισσότερους πίνακες τότε χρησιμοποιούμε την ακόλουθη εντολή:

Εντολή	Τρόπος λειτουργίας
SELECT λίστα πεδίων FROM λίστα πινάκων WHERE συνθήκη	Επιλέγονται οι εγγραφές του αναφερόμενου πίνακα ή των αναφερόμενων πινάκων που ικανοποιούν τη συνθήκη που θέτουμε.

3.5.4 SELECT ΜΕ ΛΟΓΙΚΟΥΣ ΤΕΛΕΣΤΕΣ AND ΚΑΙ OR

Η εντολή SELECT μπορεί να χρησιμοποιηθεί και για έλεγχο σύνθετων συνθηκών οι οποίες κατασκευάζονται από απλούστερες συνθήκες χρησιμοποιώντας τους λογικούς τελεστές AND και OR. Στην περίπτωση αυτή η εντολή SELECT έχει την ακόλουθη σύνταξη:

Εντολή	Τρόπος λειτουργίας
ELECT λίστα πεδίων ROM λίστα πινάκων HERE συνθήκη1 AND/OR συνθήκη2 AND/OR συνθήκη3…	Εφόσον χρησιμοποιείται ο λογικός τελεστής AND επιλέγονται όλες οι εγγραφές που ικανοποιούν ταυτόχρονα όλες τις συνθήκες ενώ όταν χρησιμοποιείται ο λογικός τελεστής OR επιλέγονται οι εγγραφές που ικανοποιούν τουλάχιστον μία από όλες τις συνθήκες

3.5.5 SELECT ΜΕ ΤΟΝ ΤΕΛΕΣΤΗ IN

Ο τελεστής IN μπορεί να χρησιμοποιηθεί με δύο διαφορετικούς τρόπους. Στο κεφάλαιο αυτό θα περιγράψουμε τον πρώτο τρόπο χρήσης του που σχετίζεται με την εντολή WHERE.

Εντολή	Τρόπος λειτουργίας
SELECT λίστα πεδίων FROM λίστα πινάκων WHERE όνομα πεδίου IN ('τιμή1', 'τιμή2', ….)	Οι τιμές μέσα στην παρένθεση μπορεί να είναι μια ή περισσότερες και διαχωρίζονται μεταξύ τους με κόμμα. Οι τιμές αυτές μπορεί να είναι αριθμητικές ή χαρακτήρες. Η λογική του τελεστή IN μοιάζει αρκετά με αυτή του τελεστή OR, δηλαδή επιλέγονται οι εγγραφές των οποίων η τιμή στο συγκεκριμένο πεδίο ισούται με κάποια από την τιμή1 ή την τιμή2 κ τ λ.

3.5.6 SELECT ΜΕ BETWEEN..AND

Ο τελεστής BETWEEN..AND επιλέγει δεδομένα που βρίσκονται ανάμεσα σε ένα διάστημα που προσδιορίζεται από δυο τιμές. Οι τιμές αυτές μπορεί να είναι αριθμοί ,κείμενο ή ημερομηνίες. Η σύνταξη του τελεστή BETWEEN..AND είναι η ακόλουθη:

Εντολή	Τρόπος λειτουργίας
SELECT λίστα πεδίων FROM πίνακας WHERE όνομα πεδίου BETWEEN τιμή1 AND τιμή2;	Επιλέγονται όλες οι εγγραφές του πίνακα που οι τιμές του πεδίου βρίσκονται μέσα στο διάστημα από την τιμή1 μέχρι την τιμή2

3.5.7 SELECT ΜΕ ΤΟΝ ΤΕΛΕΣΤΗ LIKE

Ο τελεστής LIKE χρησιμοποιείται για να συγκρίνουμε μία τιμή με παρόμοιες τιμές χρησιμοποιώντας δύο χαρακτήρες μπαλαντέρ: τον χαρακτήρα ο οποίος αντιπροσωπεύει κανένα, ένα ή πολλούς χαρακτήρες και τον χαρακτήρα υπογράμμιση (underscore) ο οποίος αντιπροσωπεύει ένα μόνο χαρακτήρα. Ο τελεστής LIKE χρησιμοποιείται για να κάνουμε μία αναζήτηση που θα βασίζεται επάνω σε ένα υπόδειγμα (pattern) και όχι ακριβής αναζήτηση όπως γίνεται με τον τελεστή IN ή επιλογή σε ένα διάστημα τιμών όπως γίνεται με τον τελεστή BETWEEN. Η σύνταξη του τελεστή LIKE είναι η ακόλουθη:

Εντολή	Τρόπος λειτουργίας
SELECT λίστα πεδίων FROM πίνακας WHERE όνομα πεδίου LIKE υπόδειγμα (pattern);	Επιλέγονται όλες οι εγγραφές του πίνακα που οι τιμές του πεδίου μοιάζουν με το υπόδειγμα. Το υπόδειγμα αυτό περιλαμβάνει εκτός των άλλων και τους χαρακτήρες μπαλαντέρ που αναφέραμε.

3.5.8 SELECT με τον τελεστή ORDER BY

Στα μέχρι τώρα παραδείγματα έχουμε χρησιμοποιήσει τις Η σύνταξη του τελεστή αυτού είναι η ακόλουθη:

Εντολή	Τρόπος λειτουργίας
SELECT λίστα πεδίων FROM πίνακας [WHERE συνθήκη] ORDER BY όνομα πεδίου [ASC, DESC]	Επιλέγονται όλες οι εγγραφές του πίνακα που οι τιμές τους ικανοποιούν την συνθήκη. Οι αγκύλες υποδηλώνουν ότι η συνθήκη είναι προαιρετική. Οι εγγραφές που εμφανίζουμε, ταξινομούνται είτε σε αύξουσα σειρά (ASC) είτε σε φθίνουσα σειρά (DESC)- Αν δεν καθορίσουμε σειρά ταξινόμησης, η εξ ορισμού σειρά ταξινόμησης είναι η αύξουσα. Γι' αυτό έχουμε τοποθετήσει και τη σειρά ταξινόμησης μέσα σε αγκύλες.

3.6 ΑΡΙΘΜΗΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ SQL

Η SQL προσφέρει αρκετές αριθμητικές συναρτήσεις με τις οποίες μπορούμε να κάνουμε πολλούς υπολογισμούς. Οι συναρτήσεις αυτές είναι οι ακόλουθες:

- SUM
- AVG
- MAX
- MIN
- COUNT

Η σύνταξη των αριθμητικών συναρτήσεων είναι η ακόλουθη:

Εντολή	Τρόπος λειτουργίας
SELECT όνομα συνάρτησης (όνομα στήλης) FROM όνομα πίνακα	Εφαρμόζεται η αριθμητική συνάρτηση που αναφέρεται στη στήλη του πίνακα που προσδιορίζουμε.

3.6.1 SELECT με τον τελεστή GROUP BY

Ο τελεστής GROUP BY χρησιμοποιείται με την εντολή SELECT όταν επιλέγουμε πολλές στήλες (πεδία) από ένα ή περισσότερους πίνακες και τουλάχιστον ένας αριθμητικός τελεστής εμφανίζεται στην εντολή SELECT. Η σύνταξη της εντολής είναι η ακόλουθη:

Εντολή	Τρόπος λειτουργίας
SELECT όνομα πεδίου, όνομα συνάρτησης (όνομα πεδίου2) FROM όνομα πίνακα GROUP BY όνομα πεδίου	Επιλέγουμε τις εγγραφές, εφαρμόζοντας σε αυτές την αριθμητική συνάρτηση που αναφέρεται και οι εγγραφές που εμφανίζονται στην απάντηση ομαδοποιούνται σύμφωνα με το πεδίο που καθορίζουμε στο GROUP BY.

3.6.2 SELECT με τον τελεστή HAVING

Ο τελεστής HAVING χρησιμοποιείται όταν θέλουμε να περιορίσουμε τα δεδομένα εξόδου που παίρνουμε συνήθως με την εντολή GROUP BY (αλλά και χωρίς αυτή) θέτοντας μία συνθήκη. Η σύνταξη της εντολής είναι η ακόλουθη:

Εντολή	Τρόπος λειτουργίας
SELECT όνομα πεδίου, όνομα συνάρτησης(όνομα πεδίου2) FROM όνομα πίνακα GROUP BY όνομα πεδίου HAVING (αριθμητική συνθήκη);	Επιλέγουμε πρώτα τις εγγραφές, εφαρμόζοντας σε αυτές την αριθμητική συνάρτηση που αναφέρεται. Αν υπάρχει ο τελεστής GROUP BY τότε οι εγγραφές που εμφανίζονται στην απάντηση ομαδοποιούνται σύμφωνα με το πεδίο που καθορίζουμε στο GROUP BY, αλλιώς όχι. Στο τέλος εφαρμόζεται στις εγγραφές αυτές μία αριθμητική συνθήκη που καθορίζουμε με την εντολή HAVING

3.6.3 SELECT με τον τελεστή JOIN

Υπάρχει δυνατότητα να συνενώνουμε δυο ή περισσότερους πίνακες προκειμένου να παίρνουμε δεδομένα από όλους μαζί. Υπάρχουν διάφοροι τύποι συνενώσεων αλλά οι πιο βασικοί είναι οι ακόλουθοι:

- INNER JOIN
- NATURAL JOIN
- OUTER JOIN

Ας μελετήσουμε διεξοδικότερα αυτούς τους τύπους συνενώσεων.

3.7 ΣΥΝΕΝΩΣΗ ΙΣΟΤΗΤΑΣ (INNER JOIN)

Ο πιο χρησιμοποιούμενος τρόπος συνένωσης είναι η συνένωση ισότητας. Σύμφωνα με αυτή συνενώνουμε δυο πίνακες οι οποίοι έχουν ένα κοινό πεδίο. Η σύνταξη της INNER JOIN είναι η εξής:

Εντολή	Τρόπος λειτουργίας
SELECT TABLE1.COLUMN1, TABLE2.COLUMN2,… FROM TABLE1, TABLE2,… WHERE TABLE1.COLUMN=TABLE2.COLUMN (AND (περισσότεροι πίνακες)…)	Επιλέγονται από τους πίνακες TABLE1, TABLE2,… όλες οι εγγραφές που έχουν ίδια τιμή στο κοινό τους πεδίο.

3.7.1 ΦΥΣΙΚΗ ΣΥΝΕΝΩΣΗ (NATURAL JOIN)

Η φυσική συνένωση είναι σχεδόν ίδια με τη συνένωση ισότητας. Η διαφορά είναι ότι εξαλείφει τις ίδιες στήλες στα πεδία που συνενώνονται. Η σύνταξη είναι η εξής:

Εντολή	Τρόπος λειτουργίας
SELECT TABLE1.*, TABLE2.COLUMN, FROM TABLE1, TABLE2, WHERE TABLE1.COLUMN = TABLE2.COLUMN (AND (περισσότεροι πίνακες))	Επιλέγονται από το TABLE1 όλα τα πεδία και από το TABLE2 μόνο το πεδίο που καθορίζουμε. Αν οι πίνακες έχουν κοινά πεδία, τότε επιλέγονται μόνο μια φορά

3.7.2 ΕΞΩΤΕΡΙΚΗ ΣΥΝΕΝΩΣΗ (OUTER JOIN)

Η εξωτερική συνένωση (outer join) χρησιμοποιείται προκειμένου να επιστρέψει όλες τις γραμμές ενός πίνακα ανεξάρτητα από το αν βρίσκονται στον δεύτερο πίνακα. Το σύμβολο της πρόσθεσης χρησιμοποιείται (στην oracle) στην εντολή WHERE στην πλευρά του πίνακα που θέλουμε να συμπεριλάβουμε όλες τις γραμμές του. Η σύνταξή της είναι η εξής:

Εντολή	Τρόπος λειτουργίας
SELECT TABLE1.COLUMN, TABLE2.COLUMN, FROM TABLE1, TABLE2, [TABLE3] WHERE TABLE1.COLUMN [(+)]=TABLE2.COLUMN [(+)] [AND TABLE1.COLUMN [(+)]=TABLE3.COLUMN [(+)]	Επιλέγονται όλες οι εγγραφές του πίνακα στον οποίο είναι το σύμβολο '+' ανεξάρτητα από το αν υπάρχει κοινή τιμή στον άλλο πίνακα.

3.8 ΔΕΥΤΕΡΕΥΟΝΤΑ ΕΡΩΤΗΜΑΤΑ (SUBQUERIES)

Ένα δευτερεύον ερώτημα είναι ένα ερώτημα το οποίο ενσωματώνεται μέσα στην εντολή WHERE ενός άλλου ερωτήματος. Το δευτερεύον ερώτημα χρησιμοποιείται για να επιστρέψει δεδομένα τα οποία θα χρησιμοποιηθούν στο κύριο ερώτημα σαν μια συνθήκη για να περιορίσει τα δεδομένα που θα εμφανιστούν. Για την κατασκευή ενός δευτερεύοντος ερωτήματος ισχύουν οι ακόλουθοι κανόνες.

- Το δευτερεύον ερώτημα πρέπει να περικλείεται σε παρενθέσεις
- Το δευτερεύον ερώτημα δεν μπορεί να περικλείεται άμεσα σε συνάρτηση.
- Ο τελεστής BETWEEN ...AND μπορεί να χρησιμοποιηθεί σε ένα δευτερεύον ερώτημα αλλά δε μπορεί να χρησιμοποιηθεί με ένα δευτερεύον ερώτημα.
- Δευτερεύοντα ερωτήματα που επιστρέφουν περισσότερες από μια γραμμές μπορούν να χρησιμοποιηθούν μόνο με τελεστές πολλαπλών τιμών όπως για παράδειγμα ο τελεστής IN.
- Μια εντολή ORDER BY δε μπορεί να χρησιμοποιηθεί μέσα σε ένα δευτερεύον ερώτημα παρ' όλο που το κυρίως ερώτημα μπορεί να χρησιμοποιεί μια ORDER BY.

Αξίζει να σημειώσουμε ότι τα δευτερεύοντα ερωτήματα υποστηρίζονται από την έκδοση MYSQL 4.1

3.8.1 ΕΡΩΤΗΜΑΤΑ ΜΕ ΤΗΝ ΕΝΤΟΛΗ SELECT

Σε αυτές τις περιπτώσεις το δευτερεύον ερώτημα χρησιμοποιεί την εντολή SELECT για να επιλέξει στοιχεία τα οποία δίνει ως δεδομένα στο βασικό ερώτημα προκειμένου αυτό να επιλυθεί

Η βασική σύνταξη ενός δευτερεύοντος ερωτήματος με την εντολή SELECT είναι η ακόλουθη:

Εντολή	Τρόπος λειτουργίας
SELECT όνομα στήλης FROM πίνακας WHERE όνομα στήλης=(SELECT όνομα στήλης FROM πίνακας WHERE συνθήκη);	Τα δεδομένα που επιλέγονται από το δευτερεύον ερώτημα χρησιμοποιούνται ως βάση για να επιλεγούν δεδομένα από το κύριο ερώτημα.

Παραδείγματα

- Έστω ότι δίνονται οι ακόλουθοι πίνακες:

Πίνακας πωλήσεων (SALES-TABLE)

STORE_REGION	SALES	SALES_DATE
Patras	1000	2004-11-14
Athens	2000	2004-11-15
Patras	3000	2004-11-16
Kalamata	1000	2004-11-17
Athens	4000	2004-12-15
Pyrgos	5000	2004-12-20

Πίνακας περιοχών (AREA-TABLE)

REGION AREA	STORE_REGION
Notia Ellada	Patras
Notia Ellada	Kalamata
Atiki	Athens

- Γράφοντας την εντολή:

```
SELECT STORE_REGION, SALES
FROM SALES_TABLE
```

```
WHERE STORE_REGION = (SELECT STORE_REGION
FROM AREA_TABLE
WHERE REGION_AREA= 'Notia Ellada');
```

εμφανίζεται η ακόλουθη απάντηση:

STORE_REGION	SALES
Patras	1000
Patras	3000
Kalamata	1000

Δηλαδή η τοποθεσία και οι πωλήσεις όλων των καταστημάτων που η περιοχή τους είναι η Νότια Ελλάδα.

3.8.2 ΕΡΩΤΗΜΑΤΑ ΜΕ ΤΗΝ ΕΝΤΟΛΗ INSERT

Σε αυτές τις περιπτώσεις το δευτερεύον ερώτημα χρησιμοποιεί την εντολή INSERT έτσι ώστε τα δεδομένα που επιστρέφονται από το δευτερεύον ερώτημα να εισαχθούν σε έναν άλλο πίνακα.

Η βασική σύνταξη ενός δευτερεύοντος ερωτήματος με την εντολή INSERT είναι η ακόλουθη:

Εντολή	Τρόπος λειτουργίας
INSERT INTO πίνακας SELECT όνομα στήλης FROM πίνακας WHERE όνομα στήλης=(SELECT όνομα στήλης FROM πίνακας WHERE συνθήκη);	Τα δεδομένα που επιλέγονται από το δευτερεύον ερώτημα προστίθενται σε ένα νέο πίνακα.

Παράδειγμα

- Έστω ότι στον πίνακα BOOKS της βάσης δεδομένων LIBRARY έχουμε καταχωρήσει τις ακόλουθες εγγραφές:

ISBN	TITLE	SUBJECT	COST	DATE_PUB
2000-200	Visual	Programming	10€	2004-7-25

	Basic			
1523-145	C++	Technology	15€	2003-8-2
2000-207	ACCESS	Databases	20€	2002-10-10
1523-145	ORGANIC CHEMISTRY	Chemistry	25€	1998-5-20
1000-107	INORGANIC CHEMISTRY	Chemistry	30€	2000-5-2
2000-300	SQL	Databases	15€	1999-11-25

και στον πίνακα DATABASES_BOOKS της ίδιας βάσης δεδομένων έχουμε καταχωρήσει τις ακόλουθες εγγραφές:

ISBN	TITLE	SUBJECT	COST	DATE_PUB
2000-207	ACCESS	Databases	20€	2/5/2000

- Γράφοντας την εντολή:

```
INSERT INTO DATABASES_BOOKS
SELECT *
FROM BOOKS
WHERE SUBJECT='DATABASES' AND COST=(SELECT *
FROM BOOKS
WHERE COST=15);
```

εμφανίζεται η ακόλουθη απάντηση:

ISBN	TITLE	SUBJECT	COST	DATE_PUB
2000-207	ACCESS	Databases	20€	2/5/2000
2000-300	SQL	Databases	15€	1999-11-25

δηλαδή προστίθεται στον πίνακα DATABASES_BOOKS η εγγραφή της SQL η οποία επιλέγεται από τον πίνακα BOOKS με το δευτερεύον ερώτημα.

3.8.3 ΕΡΩΤΗΜΑΤΑ ΜΕ ΤΗΝ ΕΝΤΟΛΗ DELETE

Σε αυτές τις περιπτώσεις το δευτερεύον ερώτημα χρησιμοποιεί την εντολή DELETE έτσι ώστε τα δεδομένα που επι-

στρέφονται από το δευτερεύον ερώτημα να διαγραφούν από έναν άλλο πίνακα.

Η βασική σύνταξη ενός δευτερεύοντος ερωτήματος με την εντολή INSERT είναι η ακόλουθη:

Εντολή	Τρόπος λειτουργίας
DELETE FROM πίνακας WHERE όνομα στήλης=(SELECT όνομα στήλης FROM πίνακας WHERE συνθήκη);	Τα δεδομένα που επιλέγονται από το δευτερεύον ερώτημα διαγράφονται από τον πίνακα.

Παράδειγμα

- Έστω ότι ο πίνακα BOOKS της βάσης δεδομένων LIBRARY περιέχει τις ακόλουθες εγγραφές:

ISBN	TITLE	SUBJECT	COST	DATE_PUB
2000-200	Visual Basic	Programming	10€	2004-7-25
1523-145	C++	Technology	15€	2003-8-2
2000-207	ACCESS	Databases	20€	2002-10-10
1523-145	ORGANIC CHEMISTRY	Chemistry	25€	1998-5-20
1000-107	INORGANIC CHEMISTRY	Chemistry	30€	2000-5-2
2000-300	SQL	Databases	15€	1999-11-25

και ο πίνακας DATABASES_BOOKS της ίδιας βάσης δεδομένων:

ISBN	TITLE	SUBJECT	COST	DATE_PUB
2000-207	ACCESS	Databases	20€	2/5/2000
2000-300	SQL	Databases	15€	1999-11-25

- Γράφοντας την εντολή:

```
DELETE FROM DATABASES_BOOKS
WHERE SUBJECT='DATABASES' AND COST=(SELECT *
FROM BOOKS
WHERE COST=15);
```

εμφανίζεται η ακόλουθη απάντηση:

ISBN	TITLE	SUBJECT	COST	DATE_PUB
2000-207	ACCESS	Databases	20€	2/5/2000

Δηλαδή διαγράφεται η εγγραφή της SQL από τον πίνακα DATABASES_BOOKS.

3.8.4 ΕΡΩΤΗΜΑΤΑ ΜΕ ΤΗΝ ΕΝΤΟΛΗ UPDATE

Σε αυτές τις περιπτώσεις το δευτερεύον ερώτημα χρησιμοποιεί την εντολή UPDATE έτσι ώστε τα δεδομένα που επιστρέφονται από το δευτερεύον ερώτημα να ενημερώνουν έναν άλλο πίνακα.

Η βασική σύνταξη ενός δευτερεύοντος ερωτήματος με την εντολή UPDATE είναι η ακόλουθη:

Εντολή	Τρόπος λειτουργίας
UPDATE πίνακας SET όνομα στήλης= WHERE όνομα στήλης= (SELECT όνομα στήλης FROM πίνακας WHERE συνθήκη);	Τα δεδομένα που επιλέγονται από το δευτερεύον ερώτημα ενημερώνουν τον πίνακα.

Παράδειγμα

- Έστω ο πίνακας BOOKS της βάσης δεδομένων LIBRARY περιέχει τις ακόλουθες εγγραφές:

ISBN	TITLE	SUBJECT	COST	DATE_PUB
2000-200	Visual Basic	Programming	10€	2004-7-25
1523-145	C++	Technology	15€	2003-8-2
2000-207	ACCESS	Databases	20€	2002-10-10

1523-145	ORGANIC CHEMISTRY	Chemistry	25€	1998-5-20
1000-107	INORGANIC CHEMISTRY	Chemistry	30€	2000-5-2
2000-300	SQL	Databases	15€	1999-11-25

- και ο πίνακας DATABASES_BOOKS της ίδιας βάσης δεδομένων περιέχει τις ακόλουθες εγγραφές:

ISBN	TITLE	SUBJECT	COST	DATE_PUB
2000-207	ACCESS	Databases	20€	2/5/2000
2000-300	SQL	Databases	15€	1999-11-25

- Γράφοντας την εντολή:

```
UPDATE DATABASES_BOOKS
SET COST=COST*1,1
WHERE SUBJECT='DATABASES' AND COST= (SELECT *
FROM BOOKS
WHERE COST=15);
```

εμφανίζεται η ακόλουθη απάντηση:

ISBN	TITLE	SUBJECT	COST	DATE_PUB
2000-300	SQL	Databases	16,5€	1999-11-25

3.9 ΜΗΧΑΝΙΣΜΟΙ ΑΠΟΘΗΚΕΥΣΗΣ - ΤΥΠΟΙ ΠΙΝΑΚΩΝ

Ο εξορισμού μηχανισμός αποθήκευσης στη MYSQL είναι ο MyISAM (στην έκδοση 3.23). Βασίζεται στον κώδικα ISAM αλλά έχει πολλές χρήσιμες επεκτάσεις.

Κάθε πίνακας MyISAM αποθηκεύεται στο δίσκο σε 3 αρχεία. Τα αρχεία αυτά έχουν ονόματα τα οποία αρχίζουν με το όνομα του πίνακα και διαθέτουν επίσης και μια κατάληξη (επέκταση) η οποία δείχνει τον τύπο του αρχείου. Ένα αρχείο με κατάληξη .FRM αποθηκεύει τον ορισμό του πίνακα. Το αρχείο δεδομένων έχει κατάληξη MYD (MYData). Τα αρχεία δεικτών έχουν κατάληξη .MYI (MYIndex).

Για να προσδιορίσουμε ότι θέλουμε να δημιουργήσουμε ένα πίνακα MyISAM γράφουμε την ακόλουθη εντολή:

```
CREATE TABLE t (i INT) ENGINE = MYISAM;
CREATE TABLE t (i INT) TYPE = MYISAM;
```

Ο μηχανισμός αποθήκευσης MERGE εισήχθηκε για πρώτη φορά στην έκδοση 3.23.25 της MySQL. Είναι γνωστός επίσης και ως μηχανισμός MRG_MyISAM.

Ένας πίνακας MERGE είναι μια συλλογή παρόμοιων πινάκων MyISAM οι οποίοι μπορούν να χρησιμοποιηθούν ως ένας πίνακας. «Παρόμοιοι» πίνακες σημαίνει ότι οι πίνακες έχουν τις ίδιες στήλες και ίδιες πληροφορίες ευρετηρίων. Δεν μπορούμε να συνενώσουμε στήλες πινάκων οι οποίες είναι σε διαφορετική σειρά ή δεν έχουν ακριβώς τις ίδιες στήλες ή τα ευρετήρια είναι σε διαφορετική σειρά.

Όταν δημιουργούμε ένα πίνακα MERGE η MySQL δημιουργεί 2 αρχεία στο δίσκο. Τα αρχεία αυτά έχουν ονόματα που αρχίζουν με το όνομα του πίνακα και έχουν μια κατάληξη

η οποία δείχνει τον τύπο του κάθε αρχείου. Ένα αρχείο με κατάληξη .frm αποθηκεύει τον ορισμό του πίνακα. Ένα αρχείο με κατάληξη .MRG file περιέχει τα ονόματα των πινάκων που πρέπει να χρησιμοποιηθούν σαν ένα. (Αρχικά όλοι οι πίνακες που χρησιμοποιούνται πρέπει να είναι μέσα στην ίδια βάση δεδομένων με τον πίνακα MERGE. Αυτός ο περιορισμός εισήχθηκε στην έκδοση MySQL 4.1.1.)

Τα ακόλουθα παραδείγματα δείχνουν τη δημιουργία ενός πίνακα MERGE:

```
mysql> CREATE TABLE t1 (
    ->     a INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    ->     message CHAR(20));
mysql> CREATE TABLE t2 (
    ->     a INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    ->     message CHAR(20));
mysql> INSERT INTO t1 (message) VALUES ('Test-
ing'),('table'),('t1');
mysql> INSERT INTO t2 (message) VALUES ('Test-
ing'),('table'),('t2');
mysql> CREATE TABLE total (
    ->     a INT NOT NULL AUTO_INCREMENT,
    ->     message CHAR(20), INDEX(a))
    ->     TYPE=MERGE UNION=(t1,t2) INSERT_METHOD=LAST
```

3.9.1 ΜΗΧΑΝΙΣΜΟΣ ΑΠΟΘΗΚΕΥΣΗΣ INNODB

Ο μηχανισμός InnoDB δίνει στην MySQL δυνατότητες για commit, rollback και επανάκαμψη από κατάρρευση (crash recovery). Ο μηχανισμός InnoDB κάνει κλείδωμα σε επίπεδο γραμμής και προσφέρει ασφαλές διάβασμα χωρίς κλείδωμα (non-locking read) στις εντολές select. Αυτές οι δυνατότητες αυξάνουν αφενός την απόδοση του συστήματος και αφετέρου διευκολύνουν την ταυτόχρονη πρόσβαση πολλών χρηστών. Τα κλειδώματα σε επίπεδο γραμμής του μηχανισμού InnoDB δεσμεύουν πολύ μικρό χώρο.

Ο μηχανισμός InnoDB προσφέρει επίσης περιορισμούς για το δευτερεύον κλειδί (foreign key). Στα ερωτήματα της SQL μπορούμε να αναμίξουμε ελεύθερα τύπους πινάκων του InnoDB (InnoDB type tables) με άλλους τύπους πινάκων της

MYSQL (αναφέρονται παρακάτω) ακόμα και εντός του ίδιου ερωτήματος.

Ο μηχανισμός InnoDB έχει σχεδιαστεί για μέγιστη απόδοση στην επεξεργασία μεγάλου όγκου δεδομένων. Η απόδοση της CPU προφανώς δεν συμπίπτει με άλλο μηχανισμό σχεσιακών βάσεων δεδομένων που βασίζεται στο δίσκο (disk-based)

Ο μηχανισμός InnoDB αποθηκεύει τους πίνακες και τα ευρετήρια του σε ένα χώρο (tablespace) ο οποίος αποτελείται από πολλά αρχεία (ή από πολλές διαμερίσεις του δίσκου). Αυτό είναι διαφορετικό π.χ. από τους πίνακες MyISAM στους οποίους ο κάθε πίνακας αποθηκεύεται χρησιμοποιώντας διαφορετικά αρχεία. Οι πίνακες InnoDB μπορεί να έχουν οποιοδήποτε μέγεθος ακόμα και σε λειτουργικά συστήματα όπου το μέγεθος του αρχείου περιορίζεται στα 2 GB.

Ξεκινώντας από την έκδοση MySQL 4.1.5. το πρόγραμμα εγκατάστασης των windows (windows installer) θέτει το InnoDB ως τον εξορισμού MySQL τύπο πινάκων στα windows.

Οι πίνακες InnoDB συμπεριλαμβάνονται στην MySQL από την έκδοση 3.23.34A και ενεργοποιήθηκαν στην έκδοση 3.23.

3.10 ΕΠΑΝΑΦΟΡΑ ΚΑΙ ΤΑΥΤΟΧΡΟΝΗ ΧΡΗΣΗ ΔΕΔΟΜΕΝΩΝ

Στο κεφάλαιο αυτό θα αναφερθούμε στους τρόπους αντιμετώπισης επαναφοράς και της ταυτόχρονης χρήσης δεδομένων από πολλούς χρήστες.

3.11 ΣΥΝΑΛΛΑΓΕΣ (TRANSACTIONS)

Μια συναλλαγή είναι μια λογική ομάδα (ενότητα) εργασιών. Τα περισσότερα συστήματα διαχείρισης βάσεων δεδομένων (RDBMS) μας δίνουν τη δυνατότητα να εκτελούμε μια σειρά από εντολές σαν μια ενότητα. Το σημαντικό είναι ότι τα αποτελέσματα μιας συναλλαγής μπορούν είτε να καταχωρηθούν μόνιμα είτε να ακυρωθούν.

3.11.1 ΕΝΤΟΛΕΣ COMMIT και ROLLBACK

Ας υποθέσουμε ότι μετά την ενεργοποίηση της διαχείρισης των συναλλαγών (TRANSACTION PROCCESING) θέλουμε να εκτελέσουμε μία ομάδα εντολών με σκοπό την αύξηση των μισθών κατά 5% όλων των υπαλλήλων που εργάζονται στο τμήμα SERVICE. Για να γίνει αυτό θα γράψουμε την ακόλουθη εντολή:

```
UPDATE EMPLOYEES SET SALARY=SALARY*1.05
```

Για να δούμε τα αποτελέσματα της εκτέλεσης αυτού του ερωτήματος γράφουμε την εντολή

```
SELECT * FROM EMPLOYEES
```

Αν για οποιονδήποτε λόγο θέλουμε να καταργήσουμε τις παραπάνω εντολές τότε γράφουμε την εντολή:

```
ROLLBACK
```

οπότε τα δεδομένα επανέρχονται στην αρχική τους κατάσταση (ακυρώνονται όλες οι αλλαγές). Αν όμως θέλουμε οι αλλαγές να αποθηκευτούν μόνιμα τότε γράφουμε την εντολή:

```
COMMIT
```

οπότε οι αλλαγές που κάνουμε είναι μόνιμες και δε μπορούμε να επαναφέρουμε τα δεδομένα στην αρχική τους κατάσταση με την εντολή ROLLBACK.

Αυτό που πρέπει να τονίσουμε είναι ότι μετά από κάθε εντολή COMMIT ή ROLLBACK η SQL θεωρεί ότι η συναλλαγή έχει ολοκληρωθεί και κάθε νέα εντολή από εκείνο το σημείο και μετά θεωρείται μέρος μιας νέας συναλλαγής μέχρις ότου χρησιμοποιηθεί πάλι κάποια από τις εντολές COMMIT ή ROLLBACK.

3.12 ΤΑΥΤΟΧΡΟΝΗ ΧΡΗΣΗ ΕΓΓΡΑΦΩΝ

Τις περισσότερες φορές μια σχεσιακή βάση δεδομένων είναι εγκατεστημένη σε δίκτυο και χρησιμοποιείται ταυτόχρονα από πολλούς χρήστες. Με αυτό τον τρόπο πολλές διαφορετικές συναλλαγές (transactions) μπορούν να εκτελούνται ταυτόχρονα και γιαυτό το λόγο πρέπει να υπάρχει ένας μηχανισμός που να συγχρονίζει όλες τις συναλλαγές που εκτελούνται την ίδια στιγμή, ώστε να μην επηρεάζει η μια την άλλη. Ο πιο γνωστός μηχανισμός είναι το κλείδωμα (locking) των εγγραφών.

Πρώτα θα περιγράψουμε τις 3 περιπτώσεις όπου δύο ή περισσότερες συναλλαγές δίνουν λάθος αποτελέσματα λόγω των εντολών που εκτελούνται σε μια από αυτές με αποτέλεσμα να επηρεάζεται η άλλη. Τα 3 προβλήματα του συγχρονισμού περιγράφονται στα ακόλουθα κεφάλαια.

3.12.1 LOST UPDATES

Υποθέτουμε ότι η συναλλαγή (transaction) A ανακτά την εγγραφή R σε χρόνο T1 και η συναλλαγή B ανακτά την ίδια εγγραφή σε χρόνο T2. Στη συνέχεια η συναλλαγή A αλλάζει (Update) την τιμή της εγγραφής R σε χρόνο T3 και η συναλλαγή B αλλάζει την τιμή της ίδιας εγγραφής σε χρόνο T4.Τότε όμως η συναλλαγή που έγινε από τη συναλλαγή A χάνεται στο χρόνο T4 διότι η συναλλαγή B την επικάλυψε (overwrite) χωρίς να έχει δει την τιμή που της έδωσε η αλλαγή από τη συναλλαγή A.

3.12.2 UNCOMMITED DEPENDENCIES

Το πρόβλημα αυτό δημιουργείται όταν μια συναλλαγή A μπορεί να αναπτύσσει ή να μεταβάλλει μια εγγραφή η οποία έχει δεχτεί αλλαγές από μια άλλη συναλλαγή B και η μεταβολή της εγγραφής δεν έχει δεσμευτεί (commit) από την συναλλαγή A. Αφού η συναλλαγή της εγγραφής παραμένει δεσμευμένη από την συναλλαγή B μπορεί κάθε στιγμή να ακυρωθεί με τη χρήση της εντολής ROLLBACK. Αν συμβεί αυτό τότε η συναλλαγή A θα έχει ανακτήσει ή θα έχει μεταβάλλει μια ανύπαρκτη εγγραφή. Για παράδειγμα έστω ότι η συναλλαγή B αλλάζει την τιμή της εγγραφής R σε χρόνο T1 στη συνέχεια η αλλαγμένη τιμή της εγγραφή R ανακτάται η μεταβάλλεται από την συναλλαγή A σε χρόνο T2. Η συναλλαγή B προχωρά σε ακύρωση της αλλαγής που έκανε με την εντολή ROLLBACK. Το πρόβλημα είναι ότι η συναλλαγή A κράτα μια τιμή της εγγραφής R που είναι ανύπαρκτη λόγω της ακύρωσης από την συναλλαγή B.

3.12.3 ΜΗ ΣΤΑΘΕΡΗ ΑΝΑΛΥΣΗ

Υποθέτουμε ότι η συναλλαγή Α αποκτά μια τιμή Table1.P1 από μια στήλη ενός πίνακα Table1 σε χρόνο t1. Κατόπιν ανακτά μια τιμή Table2.P1 από μια στήλη ενός πίνακα Table2 σε χρόνο t2. Ο στόχος της συναλλαγής Α είναι να ανακτήσει και την τιμή Table3.P1 από τον πίνακα Table3 και στο τέλος να προσθέσει τις τρεις τιμές Table1.P1, Table2.P1 και Table3.P1. Πριν από αυτό όμως και σε χρόνο Τ3 η συναλλαγή Β ανακτά την τιμή Table1.P1 από τον πίνακα Table1 και την αλλάζει σε Table1.P2. σε χρόνο t4. Το ίδιο γίνεται και για την τιμή της εγγραφής Table2.P1 η οποία μεταβάλλεται σε Table2.P2 σε αντίστοιχους χρόνους t5 και t6. Το πρόβλημα είναι ότι το άθροισμα που θα υπολογιστεί θα είναι διαφορετικό από αυτό που αναμενόταν διότι πλέον οι τιμές που θα αθροιστούν θα είναι Table1.P2, Table2.P2 και Table3.P1. Αν η τιμή του αθροίσματος πρόκειται στη συνέχεια να καταχωρηθεί σε ένα πίνακα παρουσιάζεται μια ασυνέπεια του αποτελέσματος ως προς τα δεδομένα.

3.13 ΚΛΕΙΔΩΜΑ (LOCKING)

Ο μηχανισμός του κλειδώματος είναι αρκετά απλός στη βασική του ιδέα. Αν υποθέσουμε ότι η συναλλαγή Α πρέπει να επεξεργαστεί την τιμή μιας μεταβλητής τότε καλείται ο μηχανισμός κλειδώματος και η εγγραφή δεν είναι προσπελάσιμη από καμία άλλη συναλλαγή που ίσως την καλέσει μέχρι να ολοκληρωθεί η επεξεργασία της από την συναλλαγή Α.

Ο μηχανισμός του κλειδώματος έχει τα ακόλουθα βασικά σημεία:

- Τα είδη του κλειδώματος που εφαρμόζονται από τις συναλλαγές στις εγγραφές είναι το αποκλειστικό κλείδωμα

(exclusive lock) και το διαμοιραζόμενο κλείδωμα (shared lock)

- Αν μια συναλλαγή Α εφαρμόζει αποκλειστικό κλείδωμα στην εγγραφή R τότε μια απαίτηση από τη συναλλαγή Β την ίδια χρονική στιγμή για το κλείδωμα της εγγραφής R (με οποιοδήποτε τρόπο κλειδώματος) ώστε να μπορέσει να την επεξεργαστεί, θα έχει ως αποτέλεσμα η συναλλαγή Β να μπει σε κατάσταση αναμονής μέχρι να αποδεσμευτεί η εγγραφή R από το κλείδωμα της συναλλαγής Α
- Αν η συναλλαγή Α εφαρμόζει ένα διαμοιραζόμενο κλείδωμα στην εγγραφή R τότε

    1. Μια απαίτηση από την συναλλαγή Β την ίδια χρονική στιγμή για αποκλειστικό κλείδωμα της εγγραφής R ώστε να μπορεί να την επεξεργαστεί, θα έχει ως αποτέλεσμα η συναλλαγή Β να μπει σε κατάσταση αναμονής μέχρι να αποδεσμευτεί η εγγραφή R από το κλείδωμα της συναλλαγής Α.
    2. Μια απαίτηση από τη συναλλαγή Β την ίδια χρονική στιγμή για διαμοιραζόμενο κλείδωμα της εγγραφής R ώστε να μπορεί να την επεξεργαστεί θα ικανοποιηθεί με αποτέλεσμα να εφαρμόσει και η συναλλαγή Β ένα κατανεμημένο κλείδωμα στην εγγραφή R.

- Όταν μια συναλλαγή ανακτήσει με επιτυχία μια εγγραφή τότε ασκεί αυτόματα σε αυτή διαμοιραζόμενο κλείδωμα. Αν στη συνέχεια πρέπει να εκτελέσει μια αλλαγή (UPDATE) στην τιμή της εγγραφής, τότε το κλείδωμα μετατρέπεται σε αποκλειστικό μέχρι η αλλαγή αυτή να ολοκληρωθεί. Μετά το τέλος της αλλαγής το κλείδωμα είτε παραμένει διαμοι-

ραζόμενο είτε μετατρέπεται σε αποκλειστικό είτε καταργείται ανάλογα με την εντολή που ακολουθεί.

Με το μηχανισμό κλειδώματος λύνονται αποτελεσματικά και τα 3 προβλήματα του συγχρονισμού που αναφέραμε. Συμπερασματικά μπορούμε να πούμε ότι η σωστή λειτουργία ενός συστήματος οφείλεται στην αξιόπιστη εκτέλεση των συναλλαγών.

4 ΣΚΟΠΟΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο σκοπός της πτυχιακής εργασίας μας είναι να παρουσιάσουμε αρχικά τη λειτουργία και τα χαρακτηριστικά μιας μικρομεσαίας επιχείρησης που εδρεύει στην Πάτρα και στη συνέχεια να κατασκευάσουμε ένα πληροφοριακό σύστημα σε MYSQL το οποίο θα αυτοματοποιεί όλες τις λειτουργίες αυτής της επιχείρησης.

Στο σύστημα που δημιουργήσαμε σε MYSQL προσπαθήσαμε να συμπεριλάβουμε όλες τις ανάγκες της επιχείρησης και να κατασκευάσουμε σωστούς πίνακες, σχέσεις και ερωτήματα που θα ικανοποιούν τις ανάγκες αυτές.

5 ΣΤΟΙΧΕΙΑ, ΑΝΑΓΚΕΣ ΚΑΙ ΑΠΑΙΤΗΣΕΙΣ

Η μικρομεσαία επιχείρηση την οποία περιγράφουμε και αναλύουμε είναι ένα κατάστημα εμπορίας μηχανογραφικών συσκευών (Η/Υ, εκτυπωτών, FAX κ.λ.π.) και αναλώσιμων (μελάνια, ταινίες, χαρτί κ.λ.π.) . Για ευνόητους λόγους θα ονομάσουμε και θα αναφερόμαστε σε αυτή την πραγματική επιχείρηση με το εικονικό όνομα «ΑΛΦΑ».

Η μελέτη και η καταγραφή των αναγκών της επιχείρησης έγινε κατόπιν διεξοδικής συζήτησης με τους υπεύθυνους της εταιρίας ΑΛΦΑ και ως εκ τούτου το πληροφοριακό σύστημα που κατασκευάσαμε και υλοποιήσαμε σε MYSQL πιστεύουμε ότι ικανοποιεί απόλυτα τις ανάγκες της συγκεκριμένης μικρομεσαίας επιχείρησης (τουλάχιστον σύμφωνα με αυτά που συζητήσαμε με τους υπεύθυνους της).

Το πληροφοριακό σύστημα της επιχείρησης που σχεδιάσαμε έχει ως στόχο την αυτοματοποίηση όλων των εργασιών και των λειτουργιών της επιχείρησης αυτής.

Αρχικά καταγράψαμε και αναλύσαμε λεπτομερώς όλες τις απαιτήσεις της επιχείρησης από αυτό το πληροφοριακό σύστημα. Οι απαιτήσεις που προέκυψαν από τη συγκεκριμένη επιχείρηση ύστερα από πληροφορίες που λάβαμε από τους υπεύθυνους της συνοψίζονται ακολούθως

1) Καταχώριση στον Η/Υ όλων των πελατών της με τα στοιχεία τους (π.χ. επώνυμο, όνομα, διεύθυνση κ.λ.π.). Έτσι θα μπορεί η επιχείρηση να έχει ένα οργανωμένο πελατολόγιο.
2) Καταχώριση στον Η/Υ των παραγγελιών που δίνονται από τους πελάτες της προκειμένου να διαχειρίζεται τις διαχειρίζεται κατάλληλα και να εμφανίζει πληροφοριακές καταστάσεις γιαυτές όπως π.χ. πότε δόθηκε μια παραγ-

γελία και από ποιόν πελάτη, ποια η αξία της, ποια είδη περιλαμβάνει κ.λ.π.

3) Καταχώριση στον Η/Υ των πληρωμών που γίνονται από τους πελάτες της προκειμένου η εταιρία να γνωρίζει π.χ. πόσα χρήματα πλήρωσε ο κάθε πελάτης, πότε πλήρωσε, με ποιο τρόπο (μετρητά, επιταγή κ.λ.π.), πόσα οφείλει ακόμα κ.λ.π.

4) Καταχώριση στον Η/Υ όλων των ειδών που εμπορεύεται η συγκεκριμένη εταιρία προκειμένου να έχει μια οργανωμένη αποθήκη. Έτσι θα μπορεί να παρακολουθεί το υπόλοιπο κάθε είδους, ποια είδη είναι σε έλλειψη και χρειάζονται παραγγελία ποια είδη παρουσιάζουν μεγαλύτερη ζήτηση κ.λ.π.

5) Εμφάνιση πληροφοριακών και στατιστικών καταστάσεων όπως:

   a. Εμφάνιση όλων των πελατών μιας συγκεκριμένης περιοχής

   b. Εμφάνιση όλων των πελατών που το υπόλοιπο τους (οφειλή τους) υπερβαίνει ένα συγκεκριμένο ποσό

   c. Εμφάνιση του συνολικού αλλά και του μέσο υπολοίπου όλων η συγκεκριμένων πελατών

   d. Εμφάνιση όλων των ειδών που πρέπει να παραγγελθούν (ή διαφορετικά είναι σε έλλειψη)

   e. Εμφάνιση όλων των παραγγελιών ενός συγκεκριμένου χρονικού διαστήματος

   f. Εμφάνιση όλων των παραγγελιών που αφορούν ένα ή περισσότερα είδη σε ένα συγκεκριμένο χρονικό διάστημα

g. Εμφάνιση όλων των παραγγελιών μιας συγκεκριμένης κατηγορίας/ομάδας ειδών

h. Εμφάνιση όλων των πληρωμών ενός συγκεκριμένου χρονικού διαστήματος

i. Εμφάνιση όλων των πληρωμών ενός συγκεκριμένου πελάτη ή κάποιων συγκεκριμένων πελατών σε ένα χρονικό διάστημα

6 ΟΝΤΟΤΗΤΕΣ - ΣΧΕΣΕΙΣ ΜΕΤΑΞΥ ΟΝΤΟΤΗΤΩΝ

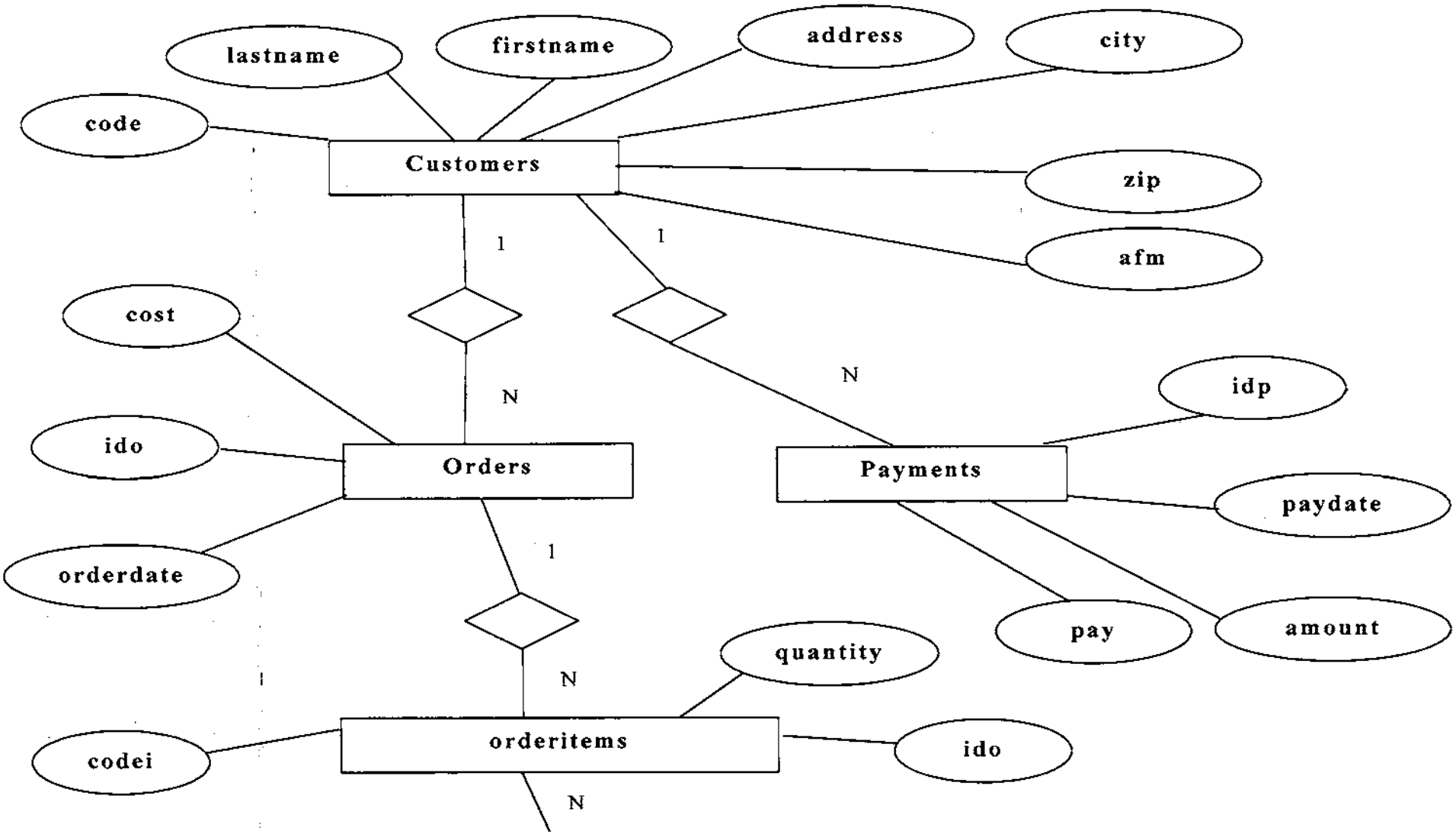
Οι οντότητες του συγκεκριμένου συστήματος είναι οι εξής:

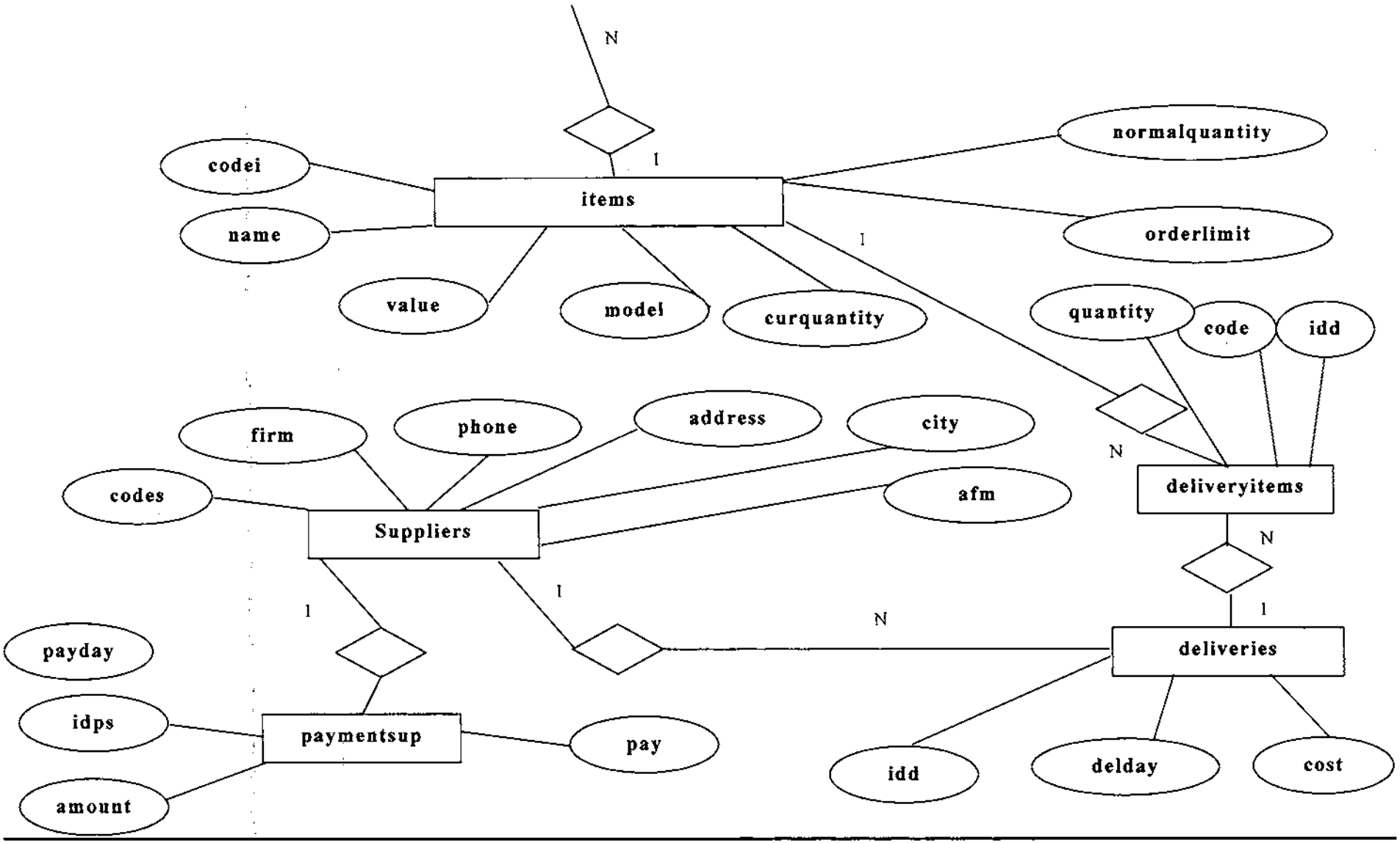
- Customers(**code**, afm, lastname, firstname, address, zip, city)
- Orders(**ido**, orderdate, cost)
- Payments(**idp**, paydate, amount, pay)
- Items(**codei**, name, value, curquantity, normalquantity, orderlimit)
- Orderitems (**codei**, **ido**, quantity)
- Suppliers(**codes**, afm, firm, address, city, phone)
- Paymentsup(**idps**, payday, amount, pay)
- Deliveries(**idd**, delday, cost, quantity)
- Models(**model id**, name)
- Deliveryitems (**codei**, **idd**, quantity)

Οι σχέσεις ανάμεσα στις οντότητες αυτές είναι οι ακόλουθες:

ΑΡΧΙΚΗ ΟΝΤΟΤΗΤΑ	ΤΕΛΙΚΗ ΟΝΤΟΤΗΤΑ	ΕΙΔΟΣ ΣΧΕΣΗΣ	ΕΡΜΗΝΕΙΑ ΣΧΕΣΗΣ
CUSTOMERS	*ORDERS*	*1-N*	*Ένας πελάτης δίνει πολλές παραγγελίες*
CUSTOMERS	*PAYMENTS*	*1-N*	*Ένας πελάτης κάνει πολλές πληρωμές*
SUPPLIERS	*DELIVERIES*	*1-N*	*Σε ένα προμηθευτή δίνονται πολλές παραγγελίες*
SUPPLIERS	*PAYMENTSUP*	*1-N*	*Σε ένα προμηθευτή γίνονται πολλές πληρωμές*
DELIVERIES	*ITEMS*	*N-N*	*Μια παραλαβή μπορεί περιλαμβάνει πολλά είδη, αλλά και ένα είδος μπορεί να περιλαμβάνεται σε πολλές παραλαβές*
ORDERS	*ITEMS*	*N-N*	*Μια παραγγελία μπορεί να περιλαμβάνει πολλά είδη, αλλά και ένα είδος μπορεί να περιλαμβάνεται σε πολλές παραγγελίες*

6.1 ΔΙΑΓΡΑΜΜΑ ΟΝΤΟΤΗΤΩΝ – ΣΥΣΧΕΤΙΣΕΩΝ




N
I
codei
name
items
value
model
curquantity
normalquantity
orderlimit
I
quantity
code
idd
N
firm
phone
address
city
codes
afm
Suppliers
deliveryitems
N
I
I
N
I
payday
deliveries
idps
paymentsup
pay
idd
delday
cost
amount

7 ΚΑΤΑΣΚΕΥΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΡΗΣΗΣ

ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Αρχικά δημιουργήσαμε μια νέα βάση δεδομένων με όνομα Company γράφοντας την εντολή:

```
CREATE DATABASE COMPANY;
```

ΕΠΙΛΟΓΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Κάθε φορά που θέλουμε να επιλέξουμε τη βάση δεδομένων company γράφουμε την εντολή:

```
USE COMPANY;
```

7.1 ΔΗΜΙΟΥΡΓΙΑ ΠΙΝΑΚΩΝ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Πίνακας 1

CUSTOMERS (ΠΕΛΑΤΕΣ)			
ΠΕΔΙΟ	**ΤΥΠΟΣ**	**ΕΡΜΗΝΕΙΑ**	**ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ**
CODE	INT	ΚΩΔΙΚΟΣ ΠΕΛΑΤΗ	ΠΡΩΤΕΥΟΝ ΚΛΕΙΔΙ ΜΕ ΜΗ ΜΗΔΕΝΙΚΕΣ ΤΙΜΕΣ ΚΑΙ ΑΥΤΟΜΑΤΗ ΑΡΙΘΜΗΣΗ
AFM	VARCHAR(9)	ΑΦΜ ΠΕΛΑΤΗ	ΚΕΙΜΕΝΟ ΜΕ 9 ΧΑΡΑΚΤΗΡΕΣ
LASTAME	VARCHAR(20)	ΕΠΩΝΥΜΟ ΠΕΛΑΤΗ	ΚΕΙΜΕΝΟ ΜΕ 20 ΧΑΡΑΚΤΗΡΕΣ ΚΑΙ ΕΙΣΑΓΩΓΗ ΣΤΟ ΕΥΡΕΤΗΡΙΟ
FIRSTNAME	VARCHAR(20)	ΟΝΟΜΑ ΠΕΛΑΤΗ	ΚΕΙΜΕΝΟ ΜΕ 20 ΧΑΡΑΚΤΗΡΕΣ
ADDRESS	VARCHAR(30)	ΔΙΕΥΘΥΝΣΗ	ΚΕΙΜΕΝΟ ΜΕ 30 ΧΑΡΑΚΤΗΡΕΣ
ZIP	VARCHAR(6)	ΤΚ	ΚΕΙΜΕΝΟ ΜΕ 6 ΧΑΡΑΚΤΗΡΕΣ
CITY	VARCHAR(10)	ΠΟΛΗ	ΚΕΙΜΕΝΟ ΜΕ 10 ΧΑΡΑΚΤΗΡΕΣ ΚΑΙ ΠΡΟΕΠΙΛΕΓΜΕΝΗ ΤΙΜΗ ΤΗ ΛΕΞΗ 'PATRAS'

Ο πίνακας Customers δημιουργήθηκε με την ακόλουθη εντολή:

```
CREATE TABLE CUSTOMERS (
        CODE INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
        AFM VARCHAR(9),
        LASTNAME VARCHAR(20),
        FIRSTNAME VARCHAR(20),
        ADDRESS VARCHAR(30),
        ZIP VARCHAR(6),
        INDEX KEYINDEX(LASTNAME))
        CITY VARCHAR(10) DEFAULT 'PATRAS'
) ENGINE=INNODB;
```

Πίνακας 2

ORDERS (ΠΑΡΑΓΓΕΛΙΕΣ)			
ΠΕΔΙΟ	**ΤΥΠΟΣ**	**ΕΡΜΗΝΕΙΑ**	**ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ**
IDO	INT	ΚΩΔΙΚΟΣ ΠΑΡΑΓΓΕΛΙΑΣ	ΠΡΩΤΕΥΟΝ ΚΛΕΙΔΙ ΜΕ ΜΗ ΜΗΔΕΝΙΚΕΣ ΤΙΜΕΣ ΚΑΙ ΑΥΤΟΜΑΤΗ ΑΡΙΘΜΗΣΗ
ORDERDATE	DATE	ΗΜΕΡΟΜΗΝΙΑ ΠΑΡΑΓΓΕΛΙΑΣ ΠΕΛΑΤΗ	ΗΜΕΡΟΜΗΝΙΑ ΜΕ ΤΟΝ ΠΕΡΙΟΡΙΣΜΟ ΟΤΙ ΔΕΝ ΜΠΟΡΕΙ ΝΑ ΥΠΕΡΒΑΙΝΕΙ ΤΗΝ ΤΡΕΧΟΥΣΑ ΗΜΕΡΟΜΗΝΙΑ
COST	DECIMAL(8,2)	ΑΞΙΑ ΠΑΡΑΓΓΕΛΙΑΣ	ΔΕΚΑΔΙΚΟΣ ΜΕ 8 ΨΗΦΙΑ ΑΠΟ ΤΑ ΟΠΟΙΑ 2 ΔΕΚΑΔΙΚΑ
CODE	INT	ΚΩΔΙΚΟΣ ΠΕΛΑΤΗ	ΔΕΥΤΕΡΕΥΟΝ ΚΛΕΙΔΙ ΑΠΟ ΠΙΝΑΚΑ CUSTOMERS

Ο πίνακας Orders δημιουργήθηκε με την ακόλουθη εντολή:

```
CREATE TABLE ORDERS (
        IDO INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
        ORDERDATE DATE,
        COST DECIMAL (8,2),
        CHECK(ORDERDATE<=CURDATE()),
        CODE INT,
        FOREIGN KEY(CODE) REFERENCES CUSTOMERS(CODE)
) ENGINE=INNODB;
```

Πίνακας 3

PAYMENTS (ΠΛΗΡΩΜΕΣ ΠΕΛΑΤΩΝ)			
ΠΕΔΙΟ	**ΤΥΠΟΣ**	**ΕΡΜΗΝΕΙΑ**	**ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ**
IDP	INT	ΚΩΔΙΚΟΣ ΠΛΗΡΩΜΗΣ	ΠΡΩΤΕΥΟΝ ΚΛΕΙΔΙ ΜΕ ΜΗ ΜΗΔΕΝΙΚΕΣ ΤΙΜΕΣ ΚΑΙ ΑΥΤΟΜΑΤΗ ΑΡΙΘΜΗΣΗ
PAYDATE	DATE	ΗΜΕΡΟΜΗΝΙΑ ΠΛΗΡΩΜΗΣ ΠΕΛΑΤΗ	ΗΜΕΡΟΜΗΝΙΑ
AMOUNT	FLOAT(8,2)	ΠΟΣΟ ΠΛΗΡΩΜΗΣ ΠΕΛΑΤΗ	ΔΕΚΑΔΙΚΟΣ ΜΕ 8 ΨΗΦΙΑ ΑΠΟ ΤΑ ΟΠΟΙΑ 2 ΔΕΚΑΔΙΚΑ
PAY	VARCHAR(20)	ΤΡΟΠΟΣ ΠΛΗΡΩΜΗΣ ΠΕΛΑΤΗ	ΚΕΙΜΕΝΟ ΜΕ 20 ΧΑΡΑΚΤΗΡΕΣ ΚΑΙ ΕΞΟΡΙΣΜΟΥ ΤΙΜΗ ΤΗ ΛΕΞΗ 'CASH'
CODE	INT	ΚΩΔΙΚΟΣ ΠΕΛΑΤΗ	ΔΕΥΤΕΡΕΥΟΝ ΚΛΕΙΔΙ ΑΠΟ ΠΙΝΑΚΑ CUSTOMERS

Ο πίνακας Payments δημιουργήθηκε με την ακόλουθη εντολή:

```
CREATE TABLE PAYMENTS (
        IDP INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
        PAYDATE DATE,
        AMOUNT FLOAT(8,2),
        PAY VARCHAR(10) DEFAULT 'CASH',
        CODE INT,
        FOREIGN KEY(CODE) REFERENCES CUSTOMERS(CODE)
) ENGINE=INNODB;
```

Πίνακας 4

ITEMS (ΕΙΔΗ)			
ΠΕΔΙΟ	**ΤΥΠΟΣ**	**ΕΡΜΗΝΕΙΑ**	**ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ**
CODEI	INT	ΚΩΔΙΚΟΣ ΕΙΔΟΥΣ	ΠΡΩΤΕΥΟΝ ΚΛΕΙΔΙ ΜΕ ΜΗ ΜΗΔΕΝΙΚΕΣ ΤΙΜΕΣ ΚΑΙ ΑΥΤΟΜΑΤΗ ΑΡΙΘΜΗΣΗ
NAME	VARCHAR(20)	ΟΝΟΜΑΣΙΑ ΕΙΔΟΥΣ	ΚΕΙΜΕΝΟ ΜΕ 20 ΧΑΡΑΚΤΗΡΕΣ ΚΑΙ ΕΙΣΑΓΩΓΗ ΣΤΟ ΕΥΡΕΤΗΡΙΟ
VALUE	FLOAT(8,2)	ΚΟΣΤΟΣ ΕΙΔΟΥΣ	ΔΕΚΑΔΙΚΟΣ ΜΕ 8 ΨΗΦΙΑ ΑΠΟ ΤΑ ΟΠΟΙΑ ΤΑ 2 ΕΙΝΑΙ ΔΕΚΑΔΙΚΑ
MODEL	VARCHAR(10)	ΜΟΝΤΕΛΟ ΕΙΔΟΥΣ	ΚΕΙΜΕΝΟ ΜΕ 10 ΧΑΡΑΚΤΗΡΕΣ
CURQUANTITY	INT	ΤΡΕΧΟΝ ΥΠΟΛΟΙΠΟ	ΑΚΕΡΑΙΟΣ ΜΕ ΜΗ ΜΗΔΕΝΙΚΕΣ ΤΙΜΕΣ
NORMALQUANTITY	INT	ΕΠΙΘΥΜΗΤΟ ΥΠΟΛΟΙΠΟ	ΑΚΕΡΑΙΟΣ ΜΕ ΜΗ ΜΗΔΕΝΙΚΕΣ ΤΙΜΕΣ
ORDERLIMIT	INT	ΟΡΙΟ ΠΑΡΑΓΓΕΛΙΑΣ	ΑΚΕΡΑΙΟΣ ΜΕ ΜΗ ΜΗΔΕΝΙΚΕΣ ΤΙΜΕΣ

Ο πίνακας Items δημιουργήθηκε με την ακόλουθη εντολή:

```
CREATE TABLE ITEMS (
        CODEI INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
        NAME VARCHAR(20),
        VALUE FLOAT(8,2),
        MODEL VARCHAR(10),
        CURQUANTITY INT NOT NULL,
        NORMALQUANTITY INT NOT NULL,
        ORDERLIMIT INT NOT NULL
) ENGINE=INNODB;
```

Πίνακας 5

ORDERITEMS (ΕΝΔΙΑΜΕΣΟΣ ΠΙΝΑΚΑΣ ΠΑΡΑΓΓΕΛΙΕΣ- ΕΙΔΗ)		
ΟΝΟΜΑ ΠΕΔΙΟΥ	**ΤΥΠΟΣ**	**ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ**
CODEI	INT	ΑΠΟ ΚΟΙΝΟΥ ΠΡΩΤΕΥΟΝΤΑ ΚΛΕΙΔΙΑ ΚΑΙ ΤΑΥΤΟΧΡΟΝΑ ΤΟ ΚΑΘΕΝΑ ΑΠΟ ΑΥΤΑ ΕΙΝΑΙ ΔΕΥΤΕΡΕΥΟΝ ΚΛΕΙΔΙ ΑΠΟ ΤΟΥΣ ΠΙΝΑΚΕΣ ITEMS ΚΑΙ ORDERS ΑΝΤΙΣΤΟΙΧΑ
IDO	INT	
QUANTITY	INT	ΑΡΙΘΜΗΤΙΚΟ ΠΕΔΙΟ ΠΟΥ ΠΕΡΙΛΑΜΒΑΝΕΙ ΤΗΝ ΠΟΣΟΤΗΤΑ ΠΑΡΑΓΓΕΛΙΑΣ

Ο πίνακας Orderltems δημιουργήθηκε με την ακόλουθη εντολή:

```
CREATE TABLE ORDERITEMS (
        CODEI INT,
        IDO INT,
        QUANTITY INT,
        PRIMARY KEY (CODEI, IDO),
        FOREIGN KEY (CODEI) REFERENCES ITEMS (CODEI),
        FOREIGN KEY (IDO) REFERENCES ORDERS (IDO)
) ENGINE=INNODB;
```

Πίνακας 6

SUPPLIERS (ΠΡΟΜΗΘΕΥΤΕΣ)			
ΠΕΔΙΟ	**ΤΥΠΟΣ**	**ΕΡΜΗΝΕΙΑ**	**ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ**
CODES	INT	ΚΩΔΙΚΟΣ ΠΡΟΜΗΘΕΥΤΗ	ΠΡΩΤΕΥΟΝ ΚΛΕΙΔΙ ΜΕ ΜΗ ΜΗΔΕΝΙΚΕΣ ΤΙΜΕΣ ΚΑΙ ΑΥΤΟΜΑΤΗ ΑΡΙΘΜΗΣΗ
AFM	VARCHAR(9)	ΑΦΜ ΠΡΟΜΗΘΕΥΤΗ	ΚΕΙΜΕΝΟ ΜΕ 9 ΧΑΡΑΚΤΗΡΕΣ
FIRM	VARCHAR(20)	ΕΠΩΝΥΜΟ ΠΡΟΜΗΘΕΥΤΗ	ΚΕΙΜΕΝΟ ΜΕ 20 ΧΑΡΑΚΤΗΡΕΣ ΚΑΙ ΕΙΣΑΓΩΓΗ ΣΤΟ ΕΥΡΕΤΗΡΙΟ
ADDRESS	VARCHAR(30)	ΔΙΕΥΘΥΝΣΗ ΠΡΟΜΗΘΕΥΤΗ	ΚΕΙΜΕΝΟ ΜΕ 30 ΧΑΡΑΚΤΗΡΕΣ
CITY	VARCHAR(10)	ΠΟΛΗ ΠΡΟΜΗΘΕΥΤΗ	ΚΕΙΜΕΝΟ ΜΕ 10 ΧΑΡΑΚΤΗΡΕΣ ΚΑΙ ΠΡΟΕΠΙΛΕΓΜΕΝΗ ΤΙΜΗ ΤΗ ΛΕΞΗ 'ATHENS'

Ο πίνακας Suppliers δημιουργήθηκε με την ακόλουθη εντολή:

```
CREATE TABLE SUPPLIERS (
        CODES INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
        AFM VARCHAR(9),
        FIRM VARCHAR(20),
        ADDRESS VARCHAR(20),
        CITY VARCHAR(20) DEFAULT 'ATHENS',
        PHONE VARCHAR(15),
        INDEX KEYINDEX(FIRM)
) ENGINE=INNODB;
```

Πίνακας 7

DELIVERIES (ΠΑΡΑΛΑΒΕΣ ΑΠΟ ΠΡΟΜΗΘΕΥΤΕΣ)			
ΠΕΔΙΟ	**ΤΥΠΟΣ**	**ΕΡΜΗΝΕΙΑ**	**ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ**
IDD	INT	ΚΩΔΙΚΟΣ ΠΑΡΑΛΑΒΗΣ	ΠΡΩΤΕΥΟΝ ΚΛΕΙΔΙ ΜΕ ΜΗ ΜΗΔΕΝΙΚΕΣ ΤΙΜΕΣ ΚΑΙ ΑΥΤΟΜΑΤΗ ΑΡΙΘΜΗΣΗ
DELDAY	DATE	ΗΜΕΡΟΜΗΝΙΑ ΠΑΡΑΛΑΒΗΣ ΑΠΟ ΠΡΟΜΗΘΕΥΤΗ	ΗΜΕΡΟΜΗΝΙΑ ΠΑΡΑΛΑΒΗΣ ΜΕ ΤΟΝ ΠΕΡΙΟΡΙΣΜΟ ΟΤΙ ΔΕΝ ΜΠΟΡΕΙ ΝΑ ΥΠΕΡΒΑΙΝΕΙ ΤΗΝ ΤΡΕΧΟΥΣΑ ΗΜΕΡΟΜΗΝΙΑ
COST	FLOAT(8,2)	ΑΞΙΑ ΠΑΡΑΛΑΒΗΣ	ΔΕΚΑΔΙΚΟΣ ΜΕ 8 ΨΗΦΙΑ ΑΠΟ ΤΑ ΟΠΟΙΑ 2 ΔΕΚΑΔΙΚΑ
QUANTITY	INT	ΠΟΣΟΤΗΤΑ ΠΑΡΑΛΑΒΗΣ	ΑΚΕΡΑΙΟΣ
CODES	INT	ΚΩΔΙΚΟΣ ΠΡΟΜΗΘΕΥΤΗ	ΔΕΥΤΕΡΕΥΟΝ ΚΛΕΙΔΙ ΑΠΟ ΠΙΝΑΚΑ SUPPLIERS

Ο πίνακας Deliveries δημιουργήθηκε με την ακόλουθη εντολή:

```
CREATE TABLE DELIVERIES (
        IDD INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
        DELDAY DATE,
        COST FLOAT(8,2),
        QUANTITY INT,
        CHECK(,DELDAY<=CURDATE()),
        CODES INT,
        FOREIGN KEY(CODES) REFERENCES SUPPLIERS(CODES)
) ENGINE=INNODB;
```

Πίνακας 8

DELIVERYITEMS (ΕΝΔΙΑΜΕΣΟΣ ΠΙΝΑΚΑΣ ΠΑΡΑΛΑΒΕΣ- ΕΙΔΗ)			
ΠΕΔΙΟ	**ΤΥΠΟΣ**	**ΕΡΜΗΝΕΙΑ**	**ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ**
CODEI	INT	ΚΩΔΙΚΟΣ ΕΙΔΟΥΣ	ΑΠΟ ΚΟΙΝΟΥ ΠΡΩΤΕΥΟΝΤΑ ΚΛΕΙΔΙΑ ΚΑΙ ΤΑΥΤΟΧΡΟΝΑ ΤΟ ΚΑΘΕΝΑ ΑΠΟ ΑΥΤΑ ΕΙΝΑΙ ΔΕΥΤΕΡΕΥΟΝ ΚΛΕΙΔΙ ΠΡΟΕΡΧΟΜΕΝΑ ΑΠΟ ΤΟΥΣ ΠΙΝΑΚΕΣ ITEMS ΚΑΙ DELIVERIES ΑΝΤΙΣΤΟΙΧΑ
IDD	INT	ΚΩΔΙΚΟΣ ΠΑΡΑΛΑΒΗΣ	
QUANTITY	INT	ΠΟΣΟΤΗΤΑ ΠΑΡΑΛΑΒΗΣ	ΑΚΕΡΑΙΟΣ ΜΕ ΤΟΝ ΠΕΡΙΟΡΙΣΜΟ ΟΤΙ ΛΑΜΒΑΝΕΙ ΜΟΝΟ ΘΕΤΙΚΕΣ ΤΙΜΕΣ

Ο πίνακας DeliveryItems δημιουργήθηκε με την ακόλουθη εντολή:

```
CREATE TABLE DELIVERYITEMS (
        CODEI INT,
        IDD INT,
        QUANTITY INT,
        CHECK (QUANTITY>0),
        PRIMARY KEY(CODEI, IDD),
        FOREIGN KEY (CODEI) REFERENCES ITEMS(CODEI),
        FOREIGN KEY (IDD) REFERENCES DELIVERIES(IDD)
) ENGINE=INNODB;
```

Πίνακας 9

DELIVERYITEMS (ΕΝΔΙΑΜΕΣΟΣ ΠΙΝΑΚΑΣ ΠΑΡΑΛΑΒΕΣ- ΕΙΔΗ)			
ΠΕΔΙΟ	**ΤΥΠΟΣ**	**ΕΡΜΗΝΕΙΑ**	**ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ**
MODEL_ID	INT	ΚΩΔΙΚΟΣ ΜΟΝΤΕΛΟΥ	ΠΡΩΤΕΥΟΝ ΚΛΕΙΔΙ ΜΕ ΜΗ ΜΗΔΕΝΙΚΕΣ ΤΙΜΕΣ ΚΑΙ ΑΥΤΟΜΑΤΗ ΑΡΙΘΜΗΣΗ
NAME	VARCHAR(20)	ΟΝΟΜΑΣΙΑ ΜΟΝΤΕΛΟΥ	ΠΕΔΙΟ ΚΕΙΜΕΝΟΥ ΜΕ 20 ΧΑΡΑΚΤΗΡΕΣ

Ο πίνακας Models δημιουργήθηκε με την ακόλουθη εντολή:

```
CREATE TABLE MODELS (
        MODEL_ID INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
        NAME VARCHAR(20)
) ENGINE=INNODB;
```

Πίνακας 10

PAYMENTSUP (ΠΛΗΡΩΜΕΣ ΣΕ ΠΡΟΜΗΘΕΥΤΕΣ)			
ΠΕΔΙΟ	**ΤΥΠΟΣ**	**ΕΡΜΗΝΕΙΑ**	**ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ**
IDPS	INT	ΚΩΔΙΚΟΣ ΠΛΗΡΩΜΗΣ	ΠΡΩΤΕΥΟΝ ΚΛΕΙΔΙ ΜΕ ΜΗ ΜΗΔΕΝΙΚΕΣ ΤΙΜΕΣ ΚΑΙ ΑΥΤΟΜΑΤΗ ΑΡΙΘΜΗΣΗ
PAYDAY	DATE	ΗΜΕΡΟΜΗΝΙΑ ΠΛΗΡΩΜΗΣ ΠΡΟΜΗΘΕΥΤΗ	ΗΜΕΡΟΜΗΝΙΑ ΜΕ ΤΟΝ ΠΕΡΙΟΡΙΣΜΟ ΟΤΙ ΔΕΝ ΜΠΟΡΕΙ ΝΑ ΥΠΕΡΒΑΙΝΕΙ ΤΗΝ ΤΡΕΧΟΥΣΑ ΗΜΕΡΟΜΗΝΙΑ
AMOUNT	FLOAT(8,2)	ΠΟΣΟ ΠΛΗΡΩΜΗΣ	ΔΕΚΑΔΙΚΟΣ ΜΕ 8 ΨΗΦΙΑ ΑΠΟ ΤΑ ΟΠΟΙΑ 2 ΔΕΚΑΔΙΚΑ
PAY	VARCHAR(20)	ΤΡΟΠΟΣ ΠΛΗΡΩΜΗΣ ΠΡΟΜΗΘΕΥΤΗ	ΚΕΙΜΕΝΟ ΜΕ 20 ΧΑΡΑΚΤΗΡΕΣ ΚΑΙ ΕΞΟΡΙΣΜΟΥ ΤΙΜΗ ΤΗ ΛΕΞΗ 'CARD'
IDD	INT	ΚΩΔΙΚΟΣ ΠΑΡΑΓΓΕΛΙΑΣ	ΔΕΥΤΕΡΕΥΟΝ ΚΛΕΙΔΙ ΑΠΟ ΠΙΝΑΚΑ DELIVERIES

Ο πίνακας PaymentSup δημιουργήθηκε με την ακόλουθη εντολή:

```
CREATE TABLE PAYMENTSUP(
      IDPS INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
      PAYDAY DATE,
      AMOUNT FLOAT(8,2),
      PAY VARCHAR(20) DEFAULT 'CARD',
      CHECK (PAYDAY<=CURDATE()),
      CODES INT,
      FOREIGN KEY(IDD) REFERENCES DELIVERIES(IDD)
)ENGINE=INNODB;
```

8 ΕΡΩΤΗΜΑΤΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

8.1 ΕΜΦΑΝΙΣΗ ΕΓΓΡΑΦΩΝ ΠΙΝΑΚΩΝ

Με τα ακόλουθα ερωτήματα εμφανίζουμε τις εγγραφές όλων των πινάκων της βάσης δεδομένων.

Εγγραφές Πίνακα Customers (Πελάτες)

```
mysql> select * from customers;
+------+-----------+-----------+-----------+--------------+-------+---------+
| CODE | AFM       | LASTNAME  | FIRSTNAME | ADDRESS      | ZIP   | CITY    |
+------+-----------+-----------+-----------+--------------+-------+---------+
|    1 | 084102397 | NIKOLAOU  | GIORGOS   | GOUNARH 1    | 26222 | PATRAS  |
|    2 | 093030303 | APOSTOLOU | PAVLOS    | KORINTHOU 28 | 26221 | PATRAS  |
|    3 | 09876543  | GEORGIOU  | MARIA     | KALABRYTON 1 | 25200 | PYRGOS  |
|    4 | 09876521  | MARATOU   | NTINA     | SOLONOS 2    | 25300 | PYRGOS  |
|    5 | 09876522  | MARATOU   | ELENH     | SOLOMOU 40   | 23500 | AIGIO   |
+------+-----------+-----------+-----------+--------------+-------+---------+
5 rows in set (0.00 sec)
```

Εγγραφές Πίνακα Payments (Πληρωμές πελατών)

```
mysql> select * from payments;
+-----+------------+--------+-------+------+
| idp | paydate    | amount | pay   | code |
+-----+------------+--------+-------+------+
|   1 | 2005-05-30 | 200.00 | CASH  |    1 |
|   2 | 2005-05-31 | 250.00 | CHECK |    2 |
|   3 | 2005-06-01 | 150.00 | CASH  |    3 |
|   4 | 2005-06-03 | 180.00 | CHECK |    3 |
|   5 | 2005-06-08 |  30.00 | CASH  |    4 |
|   6 | 2005-06-10 | 150.00 | CHECK |    5 |
|   7 | 2005-06-15 | 300.00 | CHECK |    5 |
+-----+------------+--------+-------+------+
7 rows in set (0.00 sec)
```

Εγγραφές Πίνακα Orders (Παραγγελίες)

```
mysql> use company;
Database changed
mysql> select * from orders;
+-----+------------+---------+------+
| ido | orderdate  | cost    | code |
+-----+------------+---------+------+
|   1 | 2005-05-02 |  550.00 |    1 |
|   2 | 2005-05-10 | 1000.00 |    2 |
|   3 | 2005-06-01 |  800.00 |    1 |
|   4 | 2005-06-02 | 1100.00 |    3 |
|   5 | 2005-06-03 |  100.00 |    3 |
|   6 | 2005-06-09 |  250.00 |    4 |
|   7 | 2005-06-10 |  300.00 |    5 |
|   8 | 2005-06-12 |  400.00 |    5 |
+-----+------------+---------+------+
8 rows in set (0.07 sec)
```

Εγγραφές Πίνακα Items (Είδη)

```
mysql> select * from items;
+-------+----------+--------+-------+-------------+----------------+-----------
+
| CODEI | NAME     | VALUE  | MODEL | CURQUANTITY | NORMALQUANTITY | ORDERLIMIT
|
+-------+----------+--------+-------+-------------+----------------+-----------
+
|     1 | COMPUTER | 300.00 |   343 |          15 |             10 |          5
|     2 | COMPUTER | 200.00 |   343 |           8 |              8 |          3
|     3 | PRINTER  | 100.00 |  1465 |           7 |              9 |          3
|     4 | PRINTER  | 250.00 |  1465 |          12 |             10 |          4
|     5 | FAX      |  80.00 |   545 |           5 |              4 |          2
+-------+----------+--------+-------+-------------+----------------+-----------
+
```

Εγγραφές Πίνακα Παραγγελίες - Είδη (Orderitems)

```
mysql> use company;
Database changed
mysql> select * from orderitems;
+-------+-----+----------+
| codei | ido | quantity |
+-------+-----+----------+
|     1 |   1 |        2 |
|     1 |   2 |        3 |
|     2 |   3 |        4 |
|     2 |   4 |        1 |
|     2 |   8 |        3 |
|     3 |   2 |        1 |
|     3 |   3 |        5 |
|     3 |   4 |        1 |
|     4 |   4 |        3 |
|     4 |   5 |        2 |
|     4 |   6 |        2 |
|     4 |   7 |        1 |
|     4 |   8 |        5 |
|     5 |   2 |        1 |
|     5 |   5 |        6 |
+-------+-----+----------+
15 rows in set (0.06 sec)
```

Εγγραφές Πίνακα Suppliers (Προμηθευτές)

```
mysql> select * from suppliers;
+-------+-----------+----------+--------------------+--------+------------+
| CODES | AFM       | FIRM     | ADDRESS            | CITY   | PHONE      |
+-------+-----------+----------+--------------------+--------+------------+
|     1 | 082345678 | MICROCOM | OTHONOS 20         | ATHENS | 2103344700 |
|     2 | 082345321 | DYNACOMP | PATRON-ATHINON 100 | PATRAS | 2610320456 |
|     3 | 082335610 | SOFTWARE | STADIOU 100        | ATHENS | 2101896107 |
|     4 | 082309090 | HARDWARE | ARGYROKASTROU 28   | ATHENS | 2104220500 |
+-------+-----------+----------+--------------------+--------+------------+
4 rows in set (0.05 sec)
```

Εγγραφές Πίνακα Deliveries (Παραλαβές από Προμηθευτές)

```
mysql> use company;
Database changed
mysql> select * from deliveries;
+-----+------------+---------+-------+
| idd | delday     | cost    | codes |
+-----+------------+---------+-------+
|   1 | 2005-06-01 |  300.00 |     1 |
|   2 | 2005-06-05 |  200.00 |     2 |
|   3 | 2005-06-10 |  400.00 |     2 |
|   4 | 2005-06-11 |  150.00 |     3 |
|   5 | 2005-05-31 |  600.00 |     1 |
|   6 | 2005-06-20 |  850.00 |     4 |
|   7 | 2005-06-22 | 1500.00 |     4 |
|   8 | 2005-06-30 | 1800.00 |     2 |
|   9 | 2005-06-30 | 1200.00 |     3 |
+-----+------------+---------+-------+
9 rows in set (0.07 sec)
```

Εγγραφές Πίνακα Paymentsup (Πληρωμές σε προμηθευτές)

```
mysql> select * from paymentsup;
+------+------------+---------+-------+------+
| IDPS | PAYDAY     | AMOUNT  | PAY   | idd  |
+------+------------+---------+-------+------+
|    1 | 2005-07-01 |  300.00 | CASH  |    1 |
|    2 | 2005-06-30 |  180.00 | CASH  |    2 |
|    3 | 2005-06-28 |  100.00 | CHECK |    3 |
|    4 | 2005-07-01 |  500.00 | CHECK |    4 |
|    5 | 2005-07-10 | 1000.00 | CHECK |    4 |
|    6 | 2005-07-15 | 1000.00 | CASH  |    3 |
|    7 | 2005-07-20 |  700.00 | CASH  |    3 |
+------+------------+---------+-------+------+
7 rows in set (0.00 sec)
```

Εγγραφές Πίνακα Deliveryitems (Ενδιάμεσος Πίνακας Παραλαβές-Είδη)

```
mysql> use company;
Database changed
mysql> select * from deliveryitem
+-------+-----+----------+
| codei | idd | quantity |
+-------+-----+----------+
|     1 |   1 |        3 |
|     1 |   2 |        5 |
|     1 |   5 |        4 |
|     2 |   1 |        2 |
|     2 |   3 |        4 |
|     2 |   4 |        1 |
|     3 |   3 |        5 |
|     3 |   4 |        1 |
|     4 |   5 |        5 |
|     5 |   5 |        5 |
+-------+-----+----------+
10 rows in set (0.05 sec)

mysql>
```

Εγγραφές Πίνακα Models (Μοντέλα)

```
mysql> select * from models;
+----------+-------+
| MODEL_ID | NAME  |
+----------+-------+
|      343 | IBM   |
|      545 | CANON |
|     1465 | HP    |
+----------+-------+
3 rows in set (0.05 sec)
```

8.2 ΕΡΩΤΗΜΑΤΑ ΕΠΙΛΟΓΗΣ

Τα ερωτήματα επιλογής που δημιουργήσαμε τα οποία βασίζονται σε ένα πίνακα δίνουν απάντηση σε απλά προβλήματα που μας έθεσαν οι υπεύθυνοι της μικρομεσαίας επιχείρησης.

ΕΚΦΩΝΗΣΗ

1. Να εμφανιστούν όλοι οι πελάτες της Πάτρας με τα πλήρη στοιχεία τους

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select * from customers where city='Patras';
+------+-----------+-----------+-----------+-------------+-------+--------+
| CODE | AFM       | LASTNAME  | FIRSTNAME | ADDRESS     | ZIP   | CITY   |
+------+-----------+-----------+-----------+-------------+-------+--------+
|    1 | 084102397 | NIKOLAOU  | GIORGOS   | GOUNARH 1   | 26222 | PATRAS |
|    2 | 093030303 | APOSTOLOU | PAVLOS    | KORINTHOU 28| 26221 | PATRAS |
+------+-----------+-----------+-----------+-------------+-------+--------+
2 rows in set (0.03 sec)
```

ΕΚΦΩΝΗΣΗ

2. Να εμφανιστούν όλοι οι προμηθευτές της Αθήνας με όλα τα στοιχεία τους εκτός από το ΑΦΜ τους

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select * from suppliers where city='Athens';
+-------+----------+------------------+--------+-------------+
| codes | firm     | address          | city   | phone       |
+-------+----------+------------------+--------+-------------+
|     1 | Microcom | Othonos 20       | Athens | 210-3344700 |
|     3 | Software | Stadiou 100      | Athens | 210-1896107 |
|     4 | Hardware | Argirokastrou 28 | Athens | 210-4220500 |
+-------+----------+------------------+--------+-------------+
3 rows in set (0.08 sec)
```

ΕΚΦΩΝΗΣΗ

3. Να εμφανιστούν όλες οι παραγγελίες πελατών με κόστος μικρότερο των 500€

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select * from orders where cost<500;
+-----+------------+--------+------+
| ido | orderdate  | cost   | code |
+-----+------------+--------+------+
|   5 | 2005-06-03 | 100.00 |    3 |
|   6 | 2005-06-09 | 250.00 |    4 |
|   7 | 2005-06-10 | 300.00 |    5 |
|   8 | 2005-06-12 | 400.00 |    5 |
+-----+------------+--------+------+
4 rows in set (0.00 sec)
```

4. Να εμφανιστούν όλες οι παραλαβές από προμηθευτές με κόστος από 1.000 μέχρι 1.500€. Για την κάθε παρλαβή να εμφανίζονται όλα τα στοιχεία της

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select * from deliveries where cost between 1000 and 1500;
+-----+------------+---------+-------+
| idd | delday     | cost    | codes |
+-----+------------+---------+-------+
|   7 | 2005-06-22 | 1500.00 |     4 |
|   9 | 2005-06-30 | 1200.00 |     3 |
+-----+------------+---------+-------+
2 rows in set (0.00 sec)
```

5. Να εμφανιστούν όλες οι πληρωμές πελατών με ημερομηνία πληρωμής μετά τις 06/05/2005 και τρόπο πληρωμής μετρητά

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select * from payments where paydate>'2005-06-05' and
    -> pay='cash';
+-----+------------+--------+------+------+
| idp | paydate    | amount | pay  | code |
+-----+------------+--------+------+------+
|   5 | 2005-06-08 |  30.00 | CASH |    4 |
+-----+------------+--------+------+------+
1 row in set (0.00 sec)
```

6. Να εμφανιστούν όλοι οι προμηθευτές που το αρχικό γράμμα της εταιρείας στην οποία ανήκουν να αρχίζει από 'D' ή 'S'. Για τον κάθε προμηθευτή να εμφανίζονται όλα τα στοιχεία του εκτός από το ΑΦΜ.

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select * from suppliers where firm like 'D%' or firm like 'S%';
+-------+----------+--------------------+--------+-------------+
| codes | firm     | address            | city   | phone       |
+-------+----------+--------------------+--------+-------------+
|     2 | Dynacomp | Patron-Athinon 100 | Patras | 2610-320456 |
|     3 | Software | Stadiou 100        | Athens | 210-1896107 |
+-------+----------+--------------------+--------+-------------+
2 rows in set (0.00 sec)
```

7. Να εμφανιστούν όλοι οι πελάτες με κωδικό από 1 μέχρι 3 σε φθίνουσα σειρά κατά κωδικό. Για τον κάθε πελάτη να φαίνονται τα πλήρη στοιχεία του

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select * from customers where code>=1 and code<=3
    -> order by code desc;
+------+-----------+-----------+-----------+--------------+-------+--------+
| CODE | AFM       | LASTNAME  | FIRSTNAME | ADDRESS      | ZIP   | CITY   |
+------+-----------+-----------+-----------+--------------+-------+--------+
|    2 | 093030303 | APOSTOLOU | PAVLOS    | KORINTHOU 28 | 26221 | PATRAS |
|    1 | 084102397 | NIKOLAOU  | GIORGOS   | GOUNARH 1    | 26222 | PATRAS |
+------+-----------+-----------+-----------+--------------+-------+--------+
2 rows in set (0.03 sec)
```

8.3 ΕΡΩΤΗΜΑΤΑ ΕΠΙΛΟΓΗΣ ΠΕΡΙΣΣΟΤΕΡΩΝ ΠΙΝΑΚΩΝ

Τα ερωτήματα επιλογής που δημιουργήσαμε και τα οποία βασίζονται σε περισσότερους από ένα πίνακες δίνουν απάντηση σε πιο εξειδικευμένα προβλήματα που μας έθεσαν οι υπεύθυνοι της μικρομεσαίας επιχείρησης.

ΕΚΦΩΝΗΣΗ

1. Να εμφανιστούν όλες οι παραγγελίες των πελατών. Για τον κάθε πελάτη να φαίνονται όνομα και επώνυμο και για την κάθε παραγγελία ο κωδικός και η ημερομηνία της.

ΕΡΩΤΗΜΑ

```
SELECT LASTNAME, FIRSTNAME, IDO, ORDERDATE, FROM
CUSTOMERS, ORDERS
WHERE CUSTOMERS.CODE=ORDERS.CODE;
```

ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ ΕΡΩΤΗΜΑΤΟΣ

LASTNAME	FIRSTNAME	IDO	ORDERDATE
NIKOLAOU	GIORGOS	1	2005-05-02
NIKOLAOU	GIORGOS	3	2005-06-01
APOSTOLOU	PAVLOS	2	2005-05-10
GEORGIOU	MARIA	4	2005-06-03
GEORGIOU	MARIA	5	2005-06-03
MARATOY	NTINA	6	2005-06-09
MARATOY	ELENH	7	2005-06-10
MARATOY	ELENH	8	2005-06-12

ΕΚΦΩΝΗΣΗ

2. Να εμφανιστούν οι παραγγελίες όλων των ειδών. Στην κάθε παραγγελία να φαίνεται ο κωδικός της, η ημερομηνία παραγγελίας καθώς και ο κωδικός, η ονομασία και το μοντέλο όλων των ειδών που περιλαμβάνει.

ΕΡΩΤΗΜΑ

```
SELECT ORDERS.IDO, ORDERS.ORDERDATE,
ITEMS.CODEI, ITEMS.NAME, ITEMS.MODEL
FROM ORDERS, ITEMS, ORDERITEMS
WHERE ORDERS.IDO=ORDERITEMS.IDO AND
ORDERITEMS.CODEI = ITEMS.CODE;
```

ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ ΕΡΩΤΗΜΑΤΟΣ

IDO	ORDERDATE	CODEI	NAME	MODEL
1	2005-05-02	1	COMPUTER	PENTIUM V
2	2005-05-10	1	COMPUTER	PENTIUM V
3	2005-06-01	2	COMPUTER	PENTIUM IV
4	2005-06-03	2	COMPUTER	PENTIUM IV
8	2005-06-12	2	COMPUTER	PENTIUM IV
2	2005-05-10	3	PRINTER	HP
3	2005-06-01	3	PRINTER	HP
4	2005-06-03	3	PRINTER	HP
4	2005-06-03	4	PRINTER	EPSON
5	2005-06-03	4	PRINTER	EPSON
6	2005-06-09	4	PRINTER	EPSON
7	2005-06-10	4	PRINTER	EPSON
8	2005-06-12	4	PRINTER	EPSON
2	2005-05-10	5	FAX	NOKIA
5	2005-06-03	5	FAX	NOKIA

ΕΚΦΩΝΗΣΗ

3. Να εμφανιστούν όλοι οι πελάτες με κωδικό, επώνυμο και όνομα και όλες οι παραγγελίες με κωδικό και ημερομηνία παραγγελίας για τις παραγγελίες που δόθηκαν το Μάιο του 2005. Οι πελάτες να εμφανίζονται αλφαβητικά ταξινομημένοι σε αύξουσα σειρά πρώτα κατά επώνυμο και μετά κατά όνομα.

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select customers.code,customers.lastname,customers.firstname,
    -> orders.ido,orders.orderdate
    -> from customers,orders
    -> where customers.code=orders.code
    -> and orders.orderdate
    -> between '2005-05-01' and '2005-05-31'
    -> group by customers.lastname,customers.firstname asc;
+------+-----------+-----------+-----+-------------+
| code | lastname  | firstname | ido | orderdate   |
+------+-----------+-----------+-----+-------------+
|    2 | APOSTOLOU | PAVLOS    |   2 | 2005-05-10 |
|    1 | NIKOLAOU  | GIORGOS   |   1 | 2005-05-02 |
+------+-----------+-----------+-----+-------------+
2 rows in set (0.00 sec)
```

ΕΚΦΩΝΗΣΗ

4. Να επιλεχθούν όλοι οι προμηθευτές με τον κωδικό και την εταιρία στην οποία ανήκουν και όλες οι παραλαβές με ημερομηνία παραλαβής μετά-την 1/6/2005. Οι προμηθευτές να εμφανίζονται αλφαβητικά ταξινομημένοι σε φθίνουσα σειρά κατά εταιρία.

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select suppliers.codes,suppliers.firm,deliveries.delday
    -> from suppliers,deliveries
    -> where deliveries.delday>'2005-06-01'
    -> group by suppliers.firm desc;
+-------+----------+------------+
| codes | firm     | delday     |
+-------+----------+------------+
|     3 | Software | 2005-06-05 |
|     1 | microcom | 2005-06-05 |
|     4 | Hardware | 2005-06-05 |
|     2 | Dynacomp | 2005-06-05 |
+-------+----------+------------+
4 rows in set (0.00 sec)
```

ΕΚΦΩΝΗΣΗ

5. Να επιλεχθούν όλοι οι πελάτες με το επώνυμο και το όνομα τους και όλες οι παραγγελίες με τον κωδικό τους για τους πελάτες της Πάτρας ή του Πύργου ταξινομημένους κατά επώνυμο σε αύξουσα σειρά.

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select customers.firstname,customers.lastname,
    -> orders.ido
    -> from customers,orders
    -> where customers.code=orders.code and
    -> customers.city in ('Patras','Pyrgos')
    -> order by customers.lastname;
+-----------+-----------+-----+
| firstname | lastname  | ido |
+-----------+-----------+-----+
| PAVLOS    | APOSTOLOU |   2 |
| MARIA     | GEORGIOU  |   4 |
| MARIA     | GEORGIOU  |   5 |
| NTINA     | MARATOY   |   6 |
| GIORGOS   | NIKOLAOU  |   1 |
| GIORGOS   | NIKOLAOU  |   3 |
+-----------+-----------+-----+
6 rows in set (0.00 sec)
```

ΕΚΦΩΝΗΣΗ

6. Να επιλεχθούν όλοι οι πελάτες με επώνυμο και όνομα, όλες οι παραγγελίες με τον κωδικό τους και όλες οι πληρωμές με τον κωδικό τους και με τα εξής κριτήρια: οι πληρωμές να είναι τοις μετρητοίς (cash) και το κόστος των παραγγελιών να είναι μεγαλύτερο των 1000€

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select customers.firstname,customers.lastname,
    -> orders.ido,payments.idp
    -> from customers,orders,payments
    -> where customers.code=orders.code and
    -> customers.code=payments.code and
    -> payments.pay='cash' and
    -> orders.cost>=1000;
+-----------+----------+-----+-----+
| firstname | lastname | ido | idp |
+-----------+----------+-----+-----+
| MARIA     | GEORGIOU |   4 |   3 |
+-----------+----------+-----+-----+
1 row in set (0.00 sec)
```

ΕΚΦΩΝΗΣΗ

7. Να επιλεχθούν όλοι οι προμηθευτές με τον κωδικό και το όνομα της εταιρίας τους και οι παραλαβές από αυτούς μόνο με τον κωδικό τους και με τα εξής κριτήρια: Το όνομα της εταιρίας του προμηθευτή να αρχίζει από το χαρακτήρα 'd' και το κόστος της παραλαβής να είναι μεγαλύτερο από 800€.

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select suppliers.codes,suppliers.firm,
    -> deliveries.idd
    -> from suppliers,deliveries
    -> where suppliers.codes=deliveries.codes and
    -> suppliers.firm like 'd%'
    -> and deliveries.cost>800
    -> group by deliveries.cost desc;
+-------+----------+-----+
| codes | firm     | idd |
+-------+----------+-----+
|     2 | Dynacomp |   8 |
+-------+----------+-----+
1 row in set (0.02 sec)
```

8.4 ΑΡΙΘΜΗΤΙΚΑ ΕΡΩΤΗΜΑΤΑ

Τα αριθμητικά ερωτήματα που δημιουργήσαμε δίνουν απάντηση σε συγκεκριμένα προβλήματα που μας έθεσαν οι υπεύθυνοι της επιχείρησης και τα οποία επιθυμούσαν να λύνονται από το πληροφοριακό σύστημα. Στα ερωτήματα που ακολουθούν χρησιμοποιούμε τις αριθμητικές συναρτήσεις:

- Sum
- Avg
- Max
- Min
- Count

σε συνδυασμό με τους τελεστές Group by και Having.

ΕΚΦΩΝΗΣΗ

1. Να εμφανιστεί το πλήθος των παραγγελιών του κάθε πελάτη. Στον κάθε πελάτη να εμφανίζεται μόνο το ονοματεπώνυμο του.

ΕΡΩΤΗΜΑ

```
SELECT LASTNAME, FIRSTNAME, COUNT(IDO) AS
PLITHOS_PARAGELION FROM CUSTOMERS, ORDERS WHERE
CUSTOMERS.CODE=ORDERS.CODE
GROUP BY CUSTOMERS.CODE;
```

ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ ΕΡΩΤΗΜΑΤΟΣ

LASTNAME	FIRSTNAME	PLITHOS_PARAGELION
NIKOLAOU	GIORGOS	2
APOSTOLOU	PAVLOS	1
GEORGIOU	MARIA	2
MARATOY	NTINA	1
MARATOY	ELENH	2

ΕΚΦΩΝΗΣΗ

2. Να εμφανιστεί το συνολικό ποσό πληρωμής του κάθε πελάτη. Στον κάθε πελάτη να εμφανίζεται μόνο το ονοματεπώνυμο του.

ΕΡΩΤΗΜΑ

```
SELECT LASTNAME, FIRSTNAME, SUM(AMOUNT) AS
SYNOLIKO_POSO FROM CUSTOMERS, PAYMENTS WHERE
CUSTOMERS.CODE=PAYMENTS.CODE
GROUP BY CUSTOMERS.CODE;
```

ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ ΕΡΩΤΗΜΑΤΟΣ

LASTNAME	FIRSTNAME	SYNOLIKO_POSO
NIKOLAOU	GIORGOS	200.00
APOSTOLOU	PAVLOS	250.00
GEORGIOU	MARIA	330.00
MARATOY	NTINA	30.00
MARATOY	ELENH	450.00

ΕΚΦΩΝΗΣΗ

3. Να βρεθεί το συνολικό ποσό πληρωμής των πελατών στο πρώτο δεκαήμερο του Ιουνίου 2005. Στον κάθε πελάτη να εμφανίζεται μόνο το ονοματεπώνυμο του.

ΕΡΩΤΗΜΑ

```
SELECT LASTNAME, FIRSTNAME, SUN(AMOUNT) AS
SYNOLIKO_POSO FROM CUSTOMERS, PAYMENTS WHERE
CUSTOMERS.CODE=PAYMENTS.CODE AND PAYMENTS.PAYDATE
BETWEEN '2005-06-01' AND '2005-06-10'
GROUP BY CUSTOMERS.CODE;
```

ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ ΕΡΩΤΗΜΑΤΟΣ

LASTNAME	FIRSTNAME	SYNOLIKO_POSO
GEORGIOU	MARIA	330.00
MARATOY	NTINA	30.00
MARATOY	ELENH	150.00

ΕΚΦΩΝΗΣΗ

4. Να βρεθεί το πλήθος των ειδών που περιλαμβάνει η κάθε παραγγελία από πελάτη καθώς και η συνολική αξία τους. Στην κάθε παραγγελία να εμφανίζεται μόνο ο κωδικός της.

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select deliveries.idd,count(deliveryitems.codei),
    -> sum(deliveries.cost)
    -> from deliveries,deliveryitems
    -> where deliveries.idd=deliveryitems.idd
    -> group by deliveries.idd;
+-----+-----------------------------+----------------------+
| idd | count(deliveryitems.codei) | sum(deliveries.cost) |
+-----+-----------------------------+----------------------+
|   1 |                           2 |               600.00 |
|   2 |                           1 |               200.00 |
|   3 |                           2 |               800.00 |
|   4 |                           2 |               300.00 |
|   5 |                           3 |              1800.00 |
+-----+-----------------------------+----------------------+
```

ΕΚΦΩΝΗΣΗ

5. Να εμφανιστεί η φτηνότερη (μικρότερη σε αξία) παραγγελία που δόθηκε από πελάτη

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select min(cost) from orders;
+-----------+
| min(cost) |
+-----------+
|    100.00 |
+-----------+
1 row in set (0.04 sec)
```

6. Να εμφανιστεί η ακριβότερη παραλαβή από όλους τους προμηθευτές

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select max(cost) from deliveries;
+-----------+
| max(cost) |
+-----------+
|   1800.00 |
+-----------+
1 row in set (0.02 sec)
```

ΕΚΦΩΝΗΣΗ

7. Αν το όριο παραγγελίας είναι 10 τεμάχια να εμφανιστούν ποια είδη πρέπει να παραγγείλουμε. Για το κάθε είδος που πρέπει να παραγγελθεί να εμφανίζεται μόνο ο κωδικός του

ΕΡΩΤΗΜΑ

```
SELECT ITEMS.CODEI FROM ITEMS

WHERE ORDERLIMIT < 10;
```

ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ ΕΡΩΤΗΜΑΤΟΣ

CODEI
1
2
3
4
5

ΕΚΦΩΝΗΣΗ

8. Να εμφανιστεί η συνολική αξία παραλαβών από όλους τους προμηθευτές της Αθήνας κατά τον Ιούνιο του 2005. Για την κάθε παραλαβή να φαίνεται ο κωδικός της και η εταιρία του προμηθευτή

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select deliveries.idd,suppliers.firm,suppliers.city,
    -> deliveries.delday,sum(cost) as axia_paralabhs
    -> from suppliers,deliveries
    -> where suppliers.codes=deliveries.codes
    -> and delday between '2005-06-01' and '2005-06-30'
    -> and city='Athens'
    -> group by deliveries.idd;
+-----+----------+--------+------------+----------------+
| idd | firm     | city   | delday     | axia_paralabhs |
+-----+----------+--------+------------+----------------+
|   1 | Microcom | Athens | 2005-06-01 |         300.00 |
|   4 | Software | Athens | 2005-06-11 |         150.00 |
|   6 | Hardware | Athens | 2005-06-20 |         850.00 |
|   7 | Hardware | Athens | 2005-06-22 |        1500.00 |
|   9 | Software | Athens | 2005-06-30 |        1200.00 |
+-----+----------+--------+------------+----------------+
5 rows in set (0.00 sec)
```

ΕΚΦΩΝΗΣΗ

9. Να εμφανιστεί το συνολικό ποσό πληρωμών για τον κάθε προμηθευτή. Ο προμηθευτής να εμφανίζεται με τον κωδικό και την εταιρία στην οποία ανήκει

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select suppliers.codes,suppliers.firm,sum(amount) as synoliko_poso
uppliers,paymentsup where suppliers.codes=paymentsup.codes group by suppl
des;
+-------+----------+---------------+
| codes | firm     | synoliko_poso |
+-------+----------+---------------+
|     1 | microcom |        300.00 |
|     2 | Dynacomp |       1180.00 |
|     3 | Software |        800.00 |
|     4 | Hardware |       1500.00 |
+-------+----------+---------------+
4 rows in set (0.00 sec)
```

10. Να εμφανιστεί ο μέσος όρος οφειλής (κόστους παραλαβών) για κάθε προμηθευτή. Ο προμηθευτής να εμφανίζεται με τον κωδικό και την εταιρία στην οποία ανήκει

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select suppliers.codes,suppliers.firm,avg(cost)
    -> as MO_kostous
    -> from suppliers,deliveries
    -> where suppliers.codes=deliveries.codes
    -> group by suppliers.codes;
+-------+----------+-------------+
| codes | firm     | MO_kostous  |
+-------+----------+-------------+
|     1 | Microcom |  450.000000 |
|     2 | Dynacomp |  800.000000 |
|     3 | Software |  675.000000 |
|     4 | Hardware | 1175.000000 |
+-------+----------+-------------+
4 rows in set (0.05 sec)
```

ΕΚΦΩΝΗΣΗ

11. Να βρεθεί το υπόλοιπο οφειλής σε κάθε προμηθευτή

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select suppliers.codes,sum(cost)-sum(amount)
    -> as Ypoloipo
    -> from suppliers,deliveries,paymentsup
    -> where suppliers.codes=deliveries.codes and
    -> suppliers.codes=paymentsup.codes
    -> group by suppliers.codes;
+-------+----------+
| codes | Ypoloipo |
+-------+----------+
|     1 |   300.00 |
|     2 |  1260.00 |
|     3 |  1100.00 |
|     4 |  1700.00 |
+-------+----------+
4 rows in set (0.00 sec)
```

ΕΚΦΩΝΗΣΗ

12. Να επιλεχθεί το επώνυμο και το όνομα των πελατών που το συνολικό ποσό πληρωμής τους υπερβαίνει τα 400€

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select lastname,firstname,sum(amount) from customers,payments
    -> where customers.code=payments.code group by customers.code having
    -> sum(amount)>400;
+----------+-----------+-------------+
| lastname | firstname | sum(amount) |
+----------+-----------+-------------+
| MARATOY  | ELENH     |      450.00 |
+----------+-----------+-------------+
1 row in set (0.00 sec)
```

8.5 ΔΕΥΤΕΡΕΥΟΝΤΑ ΕΡΩΤΗΜΑΤΑ

Τα δευτερεύοντα ερωτήματα που δημιουργήσαμε χρησιμοποιούν άλλα ερωτήματα για να επιστρέψουν δεδομένα τα οποία θα χρησιμοποιήσουμε στο κύριο ερώτημα ως συνθήκη για να περιορίσουμε τα δεδομένα που θα εμφανιστούν.

ΕΚΦΩΝΗΣΗ

1. Να εμφανιστεί το επώνυμο και το όνομα κάθε πελάτη που ο κωδικός του βρίσκεται σε παραγγελίες με αξία ίση με 1000€

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select firstname,lastname from customers
    -> where code=(select code
    -> from orders where cost=1000);
+-----------+-----------+
| firstname | lastname  |
+-----------+-----------+
| PAVLOS    | APOSTOLOU |
+-----------+-----------+
1 row in set (0.00 sec)
```

2. Να εμφανιστεί το επώνυμο και το όνομα κάθε πελάτη που ο κωδικός του βρίσκεται σε πληρωμές με αξία ίση με 150€ και τρόπο πληρωμής 'check'

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select firstname,lastname from customers
    -> where code=(select code
    -> from payments where amount=150 and pay='check');
+-----------+----------+
| firstname | lastname |
+-----------+----------+
| ELENH     | MARATOY  |
+-----------+----------+
1 row in set (0.00 sec)
```

3. Να εμφανιστεί ο κωδικός και η εταιρεία κάθε προμηθευτή που ο κωδικός του βρίσκεται σε παραλαβές από προμηθευτή με κωδικό 1

ΕΡΩΤΗΜΑ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

```
mysql> select codes,firm from suppliers
    -> where codes=(select distinct codes from deliveries
    -> where codes=1 or codes=2);
ERROR 1242 (21000): Subquery returns more than 1 row
mysql> select codes,firm from suppliers
    -> where codes=(select distinct codes from deliveries
    -> where codes=1);
+-------+----------+
| codes | firm     |
+-------+----------+
|     1 | Microcom |
+-------+----------+
1 row in set (0.00 sec)
```

9 ΑΣΦΑΛΕΙΑ ΔΕΔΟΜΕΝΩΝ

Η ασφάλεια δεδομένων σχετίζεται με την προστασία των δεδομένων από αναζητήσεις, τροποποιήσεις ή διαγραφές από μη εξουσιοδοτημένους χρήστες. Τα δεδομένα που αποθηκεύονται στους πίνακες μιας σχεσιακής βάσης δεδομένων επηρεάζουν τον τρόπο με τον οποίο παίρνουμε αποφάσεις και γενικά τον τρόπο με τον οποίο εργαζόμαστε. Όσο πιο πολύτιμα γίνονται τα δεδομένα αυτά, τόσο μεγαλύτερη αξία αποκτά η σχεσιακή βάση δεδομένων και το σημαντικότερο πράγμα που πρέπει να κάνουμε είναι να φροντίσουμε για την ασφάλειά της Πρέπει λοιπόν να έχουμε την δυνατότητα να απαγορεύουμε ή να επιτρέπουμε στους άλλους χρήστες να κάνουν αλλαγές στα δεδομένα, να προσθέτουν νέες εγγραφές ή να διαγράφουν δεδομένα. Η ασφάλεια μιας σχεσιακής βάσης δεδομένων είναι ένα ιδιαίτερα σημαντικό πρόβλημα στην περίπτωση του δικτύου, δηλαδή όταν τη χρησιμοποιούνε ταυτόχρονα πολλοί χρήστες. Γιαυτό το λόγο η SQL διαθέτει ένα εσωτερικό μηχανισμό ασφαλείας που ενεργοποιείται με τις εντολές GRANT και REVOKE.

9.1 ΕΝΤΟΛΗ GRANT

Η εντολή GRANT χρησιμοποιείται για να ορισθούν αρχικά οι χρήστες και τα προνόμια του συστήματος που θα τους δοθούν. Η σύνταξη της εντολής είναι η εξής:

```
GRANT CONNECT TO <όνομα χρήστη>
IDENTIFIED BY <κωδικός>
```

Η εντολή αυτή προσθέτει ένα νέο χρήστη με το όνομα και τον κωδικό που του γράφουμε. Για παράδειγμα προκειμένου να προσθέσουμε το χρήστη με όνομα Nikos και κωδικό 250 στο σύστημα γράφουμε την εντολή:

```
GRANT CONNECT TO Nikos IDENTIFIED BY 250
```

Αφού ορίσουμε τα προνόμια του συστήματος που θα δοθούν στους χρήστες, χρησιμοποιούμε την εντολή GRANT με διαφορετικό τρόπο σύνταξης για να ορίσουμε τα δικαιώματα ενός ή περισσότερων χρηστών πάνω στους πίνακες και τις κατόψεις σχεσιακών βάσεων δεδομένων. Καθορίζουμε έτσι τη δυνατότητα που έχει ο χρήστης να εκτελεί μια ή περισσότερες από τις εντολές SELECT, INSERT, DELETE, UPDATE και REFERENCES σε ένα πίνακα ή κάτοψη. . Η σύνταξη της εντολής είναι τώρα η εξής:

```
GRANT <δικαίωμα> ON <πίνακας>
TO <χρήστης>,…. [WITH GRANT OPTION]
```

Αν για παράδειγμα θέλουμε να δώσουμε το δικαίωμα στο χρήστη Nikos να μπορεί να εκτελεί ερωτήματα στον πίνακα CUSTOMERS (Πελάτες) γράφουμε την ακόλουθη εντολή:

```
GRANT SELECT ON CUSTOMERS TO Nikos
```

Μετά την εκτέλεση αυτής της εντολής ο χρήστης Nikos έχει το δικαίωμα να συντάσσει ερωτήματα στον πίνακα CUSTOMERS χρησιμοποιώντας την εντολή SELECT. Αν προσπαθήσει να εκτελέσει την ακόλουθη εντολή:

```
DELETE FROM CUSTOMERS WHERE CODE=100
```

Το σύστημα διαχείρισης βάσεων δεδομένων θα εμφανίσει μήνυμα ότι ο συγκεκριμένος χρήστης δεν έχει το δικαίωμα να διαγράφει εγγραφές από τον πίνακα CUSTOMERS δηλαδή να

χρησιμοποιεί την εντολή DELETE. Ανάλογα μηνύματα θα εμφανιστούν για τις εντολές UPDATE και INSERT. Η εντολή REFERENCES αφορά τον ορισμό των εξωτερικών κλειδιών. Ο ρόλος του σαν προνόμιο είναι ότι δίνει την δυνατότητα στο χρήστη να ορίσει τα πρωτεύοντα κλειδιά του πίνακα ως εξωτερικά κλειδιά ενός άλλου πίνακα. Για παράδειγμα αν το πεδίο CODE είναι πρωτεύον κλειδί του πίνακα SALESMEN τότε για να δώσουμε το δικαίωμα στο χρήστη Nikos να το ορίσει ως εξωτερικό κλειδί ενός άλλου πίνακα π.χ του πίνακα CUSTOMERS θα πρέπει να γράψουμε την ακόλουθη εντολή:

```
GRANT REFERENCES ON CUSTOMERS TO Nikos
```

Η εντολή GRANT μπορεί επίσης να χρησιμοποιηθεί για να ορίσει περισσότερα από ένα δικαιώματα χρήσης ταυτόχρονα. Για παράδειγμα αν θέλουμε να δώσουμε στον χρήστη Nikos το δικαίωμα INSERT και DELETE για τον πίνακα CUSTOMERS τότε γράφουμε την ακόλουθη εντολή:

```
GRANT INSERT, DELETE ON CUSTOMERS TO Nikos
```

Το δικαίωμα χρήσης της εντολής UPDATE μπορούμε να το ορίσουμε με δυο τρόπους. Για παράδειγμα αν θέλουμε να δώσουμε το δικαίωμα στο χρήστη Nikos να μπορεί να τροποποιεί τα στοιχεία των πωλητών του πίνακα SALESMEN θα γράψουμε την εντολή:

```
GRANT UPDATE ON SALESMEN TO Nikos
```

Ο άλλος τρόπος είναι να δώσουμε το δικαίωμα UPDATE σε συγκεκριμένα πεδία ενός πίνακα π.χ αν θέλουμε ο χρήστης Nikos να έχει το δικαίωμα UPDATE στα πεδία CODE, NAME και CITY του πίνακα SALESMEN γράφουμε την εντολή:

```
GRANT UPDATE (CODE, NAME, CITY) ON SALESMEN TO Nikos
```

Δηλαδή αρκεί να δηλώσουμε τα πεδία που θέλουμε να τροποποιήσει ο χρήστης Nikos μέσα σε παρενθέσεις. Η SQL διαθέτει ακόμα δυο όρους που μπορούμε να χρησιμοποιήσουμε μέσα στην εντολή GRANT για να επιταχύνουμε την διαδικασία ανάθεσης δικαιωμάτων στους χρήστες. Αν θέλουμε να δώσουμε όλα τα δικαιώματα σε ένα χρήστη, μπορούμε να χρησιμοποιήσουμε την εντολή ALL PRIVILEGES. Για παράδειγμα αν θέλουμε να δώσουμε όλα τα δικαιώματα για τον χρήστη Nikos τότε γράφουμε την εντολή:

```
GRANT ALL PRIVILEGES ON SALESMEN TO Nikos
```

Εναλλακτικά η προηγούμενη εντολή μπορεί να συνταχθεί και ως εξής:

```
GRANT ALL ON SALESMEN TO Nikos
```

Πολλές φορές είναι επιθυμητό μετά τη δημιουργία ενός πίνακα και την ανάθεση δικαιωμάτων στους χρήστες να θέλουμε να μεταβιβάσουμε σε μερικούς από αυτούς τη δυνατότητα να αναθέτουν και αυτοί δικαιώματα σε άλλους χρήστες για το συγκεκριμένο πίνακα. Η ακόλουθη εντολή χρησιμοποιεί τον όρο WITH GRANT OPTION και δίνει το δικαίωμα χρήσης της εντολής SELECT στο χρήστη Kostas ο οποίος με τη σειρά του μπορεί να μεταβιβάσει το δικαίωμα αυτό και σε άλλο χρήστη.

```
GRANT SELECT ON CUSTOMERS TO Kostas WITH GRANT OPTION
```

Ο χρήστης Kostas μπορεί με τη σειρά του να μεταβιβάσει το δικαίωμα που του δόθηκε στο χρήστη Giannis γράφοντας την ακόλουθη εντολή:

```
GRANT SELECT ON CUSTOMERS TO Giannis
```

9.2 ΕΝΤΟΛΗ REVOKE

Με την εντολή REVOKE έχουμε τη δυνατότητα να ανακαλέσουμε τα δικαιώματα που έχουμε αναθέσει στους χρήστες. Η σύνταξη της εντολής είναι η εξής:

```
REVOKE <δικαίωμα> FROM <χρήστης>
```

Αν για παράδειγμα θέλουμε να ανακαλέσουμε το δικαίωμα RESOURCE που είχαμε δώσει στο χρήστη Nikos πρέπει να γράψουμε την παρακάτω εντολή:

```
REVOKE RESOURCE FROM Nikos
```

Αυτό σημαίνει ότι ο χρήστης Nikos διατηρεί όλα τα προνόμια που του έχουν δοθεί μέχρι στιγμής αλλά δεν μπορεί πλέον να δημιουργήσει νέους πίνακες, κατόψεις κ.λ.π.

Με την εντολή:

```
REVOKE CONNECT FROM Kostas
```

Ανακαλούμε το προνόμιο συστήματος που είχαμε αναθέσει στο χρήστη Kostas. Ουσιαστικά όμως του αφαιρούμε το δικαίωμα να χρησιμοποιεί τη σχεσιακή βάση δεδομένων. Πρέπει όμως να παρατηρήσουμε ότι πριν αφαιρέσουμε το δικαίωμα χρήσης (REVOKE CONNECT) από οποιονδήποτε χρή-

στη, πρέπει πρώτα να ελέγξει αν έχει δημιουργήσει έναν ή περισσότερους πίνακες. Τέτοια δικαιώματα έχουν μόνο οι χρήστες με το δικαίωμα RESOURCE. Αν λοιπόν ο χρήστης Nikos έχει δημιουργήσει ένα μόνο πίνακα τότε δεν μπορούμε να του αφαιρέσουμε το προνόμιο CONNECT δηλαδή το δικαίωμα χρήσης του πίνακα αυτού καθώς και της σχεσιακής βάσης δεδομένων.

Η εντολή REVOKE χρησιμοποιείται επίσης και για την αφαίρεση των προνομίων χρήσης (OBJECT PRIVILEGES). Η γενική σύνταξη της εντολής REVOKE στην περίπτωση αυτή είναι:

```
REVOKE<προνόμιο χρήσης> ON <πίνακας> FROM <χρήστης>
```

Μπορούμε να αφαιρέσουμε ένα ή και όλα τα προνόμια που έχουμε αναθέσει σε κάποιον χρήστη. Για παράδειγμα αν θέλουμε να αφαιρέσουμε από τον χρήστη Nikos το δικαίωμα χρήσης της εντολής DELETE στον πίνακα SALESMEN τότε γράφουμε την εντολή:

```
REVOKE DELETE ON SALESMEN FROM Nikos
```

Ο χρήστης Nikos μετά την εκτέλεση της εντολής μπορεί να χρησιμοποιεί μόνο την εντολή INSERT στον πίνακα SALESMEN. Αντίστοιχα μπορούμε να ανακαλέσουμε και τα υπόλοιπα προνόμια χρήσης που μπορεί να έχουν αναθέσει σε κάποιον χρήστη. Με την εντολή REVOKE μπορούμε να χρησιμοποιήσουμε επίσης τις εντολές ALL και PUBLIC. Για παράδειγμα αν θέλουμε να αφαιρέσουμε από το χρήστη Nikos όλα τα δικαιώματα που έχει στον πίνακα SALESMEN θα μπορούσαμε αντί για την εντολή:

```
REVOKE INSERT, DELETE ON SALESMEN FROM Nikos
```

να γράψουμε την εντολή:

```
REVOKE ALL ON SALESMEN FROM Nikos
```

Αν επιθυμούσαμε να αφαιρέσουμε από όλους τους χρήστες μόνο ένα προνόμιο τότε θα γράφαμε την εντολή:

```
REVOKE SELECT ON PRODUCTS FROM PUBLIC
```


```
REVOKE SELECT ON PRODUCTS FROM PUBLIC
```


Ο διαχειριστής συστήματος είναι υπεύθυνος για την ασφάλεια της σχεσιακής βάσης δεδομένων. Καθώς αυξάνεται ο αριθμός των χρηστών αυξάνονται αντίστοιχα και οι κωδικοί ασφαλείας και τα προνόμια χρήσης ή συστήματος που αναθέτουμε σε αυτούς. Γιαυτό το λόγο τα περισσότερα συστήματα διαχείρισης βάσεων δεδομένων (RDBMS) διαθέτουν κάποιες εντολές που μας δίνουν τη δυνατότητα να βλέπουμε τις πληροφορίες που έχουν σχέση με την ασφάλεια της σχεσιακής βάσης δεδομένων. Τέτοιες πληροφορίες, που είναι διαθέσιμες μόνον στο διαχειριστή του συστήματος, είναι οι ονομασίες των χρηστών, οι κωδικοί ασφαλείας τους, τα προνόμια χρήσης και συστήματος που τους έχουν ανατεθεί. Ο διαχειριστής του συστήματος έχει τη δυνατότητα να αλλάζει τους κωδικούς ασφαλείας των χρηστών χρησιμοποιώντας την εντολή GRANT. Για παράδειγμα για να αλλάξει ο κωδικός ασφαλείας του χρήστη Nikos από Q3 σε R1ο διαχειριστής θα πρέπει να δώσει την εξής εντολή:

```
GRANT CONNECT Nikos IDENTIFIED BY R1
```

Βέβαια ο διαχειριστής θα πρέπει να ενημερώσει το χρήστη για το νέο κωδικό ασφαλείας.

10 ΣΥΜΠΕΡΑΣΜΑΤΑ ΠΤΥΧΙΑΚΗΣ

Τα συμπεράσματα από την εκπόνηση της πτυχιακής εργασίας είναι ότι με τη χρήση της MySQL ως ΣΔΒΔ (RDBMS) μπορούμε με αρκετά μεγάλη ευκολία και ευελιξία να καταχωρίσουμε στον Η/Υ όλες τις πληροφορίες που χρειάζεται μια μικρομεσαία επιχείρηση εμπορίας Η/Υ και αναλώσιμων, να συνδέσουμε αυτούς τους πίνακες ώστε να βρίσκουμε συνδυαστικές πληροφορίες από όλους αυτούς και να δημιουργούμε πολύπλοκα ερωτήματα τα οποία ικανοποιούν με ακρίβεια όλες τις ανάγκες της επιχείρησης για στατιστικές και πληροφοριακές εκτυπώσεις σχετικές με τη λειτουργία της επιχείρησης.

Πιστεύουμε ότι εξίσου χρήσιμη θα ήταν και η υλοποίηση του συγκεκριμένου πληροφοριακού συστήματος στην Microsoft Access καθώς και αυτό το ΣΔΒΔ προσφέρει πολλές δυνατότητες για περιγραφή και αυτοματοποίηση μιας τέτοιας επιχείρησης. Το πλεονέκτημα όμως της MSQL είναι ότι ο χρήστης μπορεί να δημιουργήσει πιο δυναμικά και ευέλικτα ερωτήματα και να καλύψει καλύτερα τις τρέχουσες αλλά και τις μελλοντικές ανάγκες μιας τέτοιας επιχείρησης.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Στην εκπόνηση της πτυχιακής εργασίας μας χρησιμοποιήσαμε τα βιβλία και τις δικτυακές τοποθεσίες που αναφέρονται ακολούθως:

1. A Guide to the SQL Standard – Second Edition – C. J. Date
2. Μάθετε την SQL σε 24 ώρες – Ryan Stephens - Ron Plew
3. SQL – Structured Query Language – Κων/νος Καδής
4. Σχεσιακές Βάσεις Δεδομένων – Σχεδιασμός και Υλοποίηση - Χρήστου Σκουρλά
5. PHP, MYSQL and Apache – All in one – Julie C. Meloni
6. Μάθετε PHP, MYSQL και Apache σε 24 ώρες – Julie C. Meloni
7. Δικτυακοί Τόποι (web sites):
   - http://www.mysql.com
   - http://www.w3schools.com/sql/default.asp
   - http://www.thefreecountry.com/documentation/onlinesql.shtml
   - http://www.sqlcourse.com/
   - http://www.techtutorials.info/dataSQL.html
   - http://www.programmingtutorials.com/sql.aspx
   - http://www.webdevelopersnotes.com/tutorials/sql/index.php3