

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΛΟΓΙΣΤΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Ανάπτυξη Βάσεων Δεδομένων επιχειρησιακού**

**ενδιαφέροντος σε σύστημα MySQL.**

**Ανάλυση και σχεδιασμός.**

ΕΚΠΟΝΗΣΗ:

ΔΟΥΦΕΞΗ ΑΝΑΣΤΑΣΙΑ

ΠΑΓΑΝΙΑ ΣΠΥΡΙΔΟΥΛΑ

ΣΟΥΝΔΙΑ ΓΕΩΡΓΙΑ

ΕΠΟΠΤΕΥΩΝ ΚΑΘΗΓΗΤΗΣ

ΜΙΧΑΛΗΣ ΦΩΤΕΙΝΟΠΟΥΛΟΣ

ΠΑΤΡΑ - 2018

## Περίληψη

Σκοπός της εργασίας αυτής είναι να αναλυθεί η σημασία των Βάσεων Δεδομένων, ιδιαίτερα αυτών που χρησιμοποιούνται από τις επιχειρήσεις. Αυτό γίνεται με ανάλυση τόσο της θεωρία που σχετίζεται με τις Βάσεις Δεδομένων αλλά και την πρακτική εφαρμογή ερωτημάτων με χρήση της γλώσσας SQL.

Ξεκινώντας, στην εργασία γίνεται μια ανάλυση της θεωρίας που αφορά την επιστήμη των Η/Υ αλλά και των Βάσεων Δεδομένων. Οι Βάσεις Δεδομένων έχουν ευρύτατη εφαρμογή στο σημερινό επιχειρηματικό περιβάλλον και η πλειοψηφία των επιχειρήσεων χρησιμοποιεί κάποιου είδους Βάση Δεδομένων ανεξάρτητα από το μέγεθος της. Τα οφέλη από τη χρήση μιας Βάσης Δεδομένων είναι πολλά. Η ταχύτητα διεκπεραίωσης των εργασιών, η αξιοπιστία και η ευελιξία που παρέχουν στο χρήστη είναι μόνο μερικά από αυτά. Επίσης το γεγονός ότι ταυτόχρονα μπορούν πολλοί χρήστες να αξιοποιούν τα δεδομένα της αποτελεί άλλο ένα χαρακτηριστικό που αυξάνει την αποδοτικότητα των χρηστών μιας Βάσης Δεδομένων.

Στην συνέχεια της εργασίας γίνεται ανάλυση τόσο της θεωρίας που αφορά τις Βάσεις Δεδομένων γενικά αλλά επικεντρώνεται στην γλώσσα Βάσεων Δεδομένων SQL. Η SQL (Structured Query Language) είναι μία πλήρης γλώσσα Βάσεων Δεδομένων. Έχει σχεδιαστεί για την διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων και αποτελεί έναν ιδιαίτερα διαδεδομένο τρόπο για την δημιουργία και την επεξεργασία Βάσεων Δεδομένων.

Στο τελευταίο τμήμα της εργασίας γίνεται η πρακτική εφαρμογή ερωτημάτων με την χρήση του ερωτήματος SELECT σε μια Βάση Δεδομένων που δημιουργήθηκε με την χρήση MySQL και αφορούν την βάση δεδομένων που αναφέρθηκε προηγουμένως. Η εντολή SELECT επιλέγει στήλες (columns) δεδομένων από μια βάση δεδομένων. Το αποτέλεσμα αποθηκεύεται σε μορφή πίνακα και αποκαλείται result set.

## Περιεχόμενα

Περίληψη .....	2
Περιεχόμενα Εικόνων .....	5
1. Η Επιστήμη των Η/Υ .....	7
1.1 Ιστορία της επιστήμης των υπολογιστών.....	8
1.2 Πληροφοριακά Συστήματα.....	9
1.3 Κατηγορίες πληροφοριακών συστημάτων.....	12
1.4 Οργάνωση των Αρχείων .....	14
1.5 Δεδομένα και Πληροφορίες .....	16
1.6 Προβλήματα της Οργάνωσης Αρχείων .....	18
2. Βάσεις Δεδομένων .....	20
2.1 Συστήματα Διαχείρισης Βάσεων Δεδομένων .....	21
2.2 Διαφορά Βάσης Δεδομένων με Σύστημα Διαχείρισης Βάσεων Δεδομένων.....	22
2.3 Οφέλη από τη χρήση των Βάσεων Δεδομένων .....	24
2.4 Λόγοι για την αποφυγή χρήσης Βάσεων Δεδομένων .....	26
2.5 Επίπεδα χρηστών .....	27
3. Η γλώσσα Βάσεων Δεδομένων SQL .....	29
3.1 Θεωρία Σχεσιακών Βάσεων Δεδομένων .....	30
3.2 Κανόνες για την σχεδίαση των πινάκων .....	31
3.3 Κανόνες για την εκτέλεση εντολών και την εισαγωγή τιμών.....	31

3.4 Εντολές της MySQL.....	32
3.5 Εφαρμογή σε MySQL.....	33
3.6 Εκτέλεση ερωτημάτων με την χρήση της εντολής SELECT .....	36
Συμπεράσματα .....	57
Βιβλιογραφία.....	59
Ηλεκτρονικές πηγές .....	59

## Περιεχόμενα Εικόνων

Εικόνα 1 Μέρη πληροφοριακού συστήματος (Πηγή: <a href="http://ebooks.edu.gr">http://ebooks.edu.gr</a> ).....	10
Εικόνα 2 Η πληροφορία ως προϊόν επεξεργασμένων δεδομένων (Πηγή: <a href="http://www.differencebetween.info/difference-between-data-and-information">http://www.differencebetween.info/difference-between-data-and-information</a> ).....	16
Εικόνα 3 Διαφορά δεδομένων και πληροφορίας (Πηγή: <a href="http://effectualsystems.com/data-need-information/">http://effectualsystems.com/data-need-information/</a> ).....	17
Εικόνα 4 Απεικόνιση Βάσης Δεδομένων (Πηγή: <a href="https://www.kbmanage.com/concept/information-database-idb">https://www.kbmanage.com/concept/information-database-idb</a> ) .....	20
Εικόνα 5 Λογότυπο MySQL (Πηγή: <a href="http://www.wikipedia.com">www.wikipedia.com</a> ) .....	29
Εικόνα 6: Πίνακας δεδομένων .....	30
Εικόνα 7: Δημιουργία Βάσης Δεδομένων.....	33
Εικόνα 8: Δημιουργία Πίνακα .....	34
Εικόνα 9: Καταχώρηση δεδομένων σε ένα πίνακα.....	34
Εικόνα 10: Δημιουργία πίνακα .....	35
Εικόνα 11: Καταχώρηση δεδομένων σε ένα πίνακα.....	35
Εικόνα 12: Ερώτημα 1.....	36
Εικόνα 13: Ερώτημα 2.....	37
Εικόνα 14: Ερώτημα 3.....	38
Εικόνα 15: Ερώτημα 4.....	39
Εικόνα 16: Ερώτημα 5.....	40
Εικόνα 17: Ερώτημα 6.....	41
Εικόνα 18: Ερώτημα 7.....	42
Εικόνα 19 : Ερώτημα 8.....	43

Εικόνα 20 : Ερώτημα 9.....	44
Εικόνα 21 : Ερώτημα 10.....	45
Εικόνα 22 : Ερώτημα 20.....	47
Εικόνα 23: Ερώτημα 21.....	48
Εικόνα 24 : Ερώτημα 22.....	49
Εικόνα 25 : Ερώτημα 23.....	50
Εικόνα 26: Ερώτημα 24.....	51
Εικόνα 27: Ερώτημα 25.....	52
Εικόνα 28 : Ερώτημα 26.....	53
Εικόνα 29: Ερώτημα 27.....	54
Εικόνα 30 : Ερώτημα 28.....	55
Εικόνα 31 : Ερώτημα 29.....	56

## 1. Η Επιστήμη των Η/Υ

Σαν επιστήμη υπολογιστών ορίζεται «η θετική και εφαρμοσμένη επιστήμη η οποία έχει ως στόχο να ερευνήσει τα θεωρητικά θεμέλια και τη φύση των δομών δεδομένων, των αλγορίθμων και των υπολογισμών, από τη σκοπιά της σχεδίασης, της ανάπτυξης, της υλοποίησης, της διερεύνησης, της ανάλυσης και της προδιαγραφής τους» (Πηγή:wikipedia). Ένας ακόμα ορισμός που δίνεται από το Πανεπιστήμιο του Oxford αναφέρει ότι: «η επιστήμη των υπολογιστών είναι αυτή που αφορά την κατανόηση των συστημάτων υπολογιστών και δικτύων. Οι υπολογιστές και τα προγράμματα που εκτελούν είναι από τα πιο σύνθετα προϊόντα που δημιουργήθηκαν ποτέ. ο σχεδιασμός και η χρήση τους παρουσιάζει τεράστιες προκλήσεις. Οι θεωρίες που αναδύονται για να αντιμετωπιστούν αυτές οι προκλήσεις, μπορούν να εφαρμοστούν άμεσα για το σχεδιασμό νέων υπολογιστών, προγραμμάτων, δικτύων και συστημάτων που μεταμορφώνουν την επιστήμη, την επιχειρηματικότητα, τον πολιτισμό και όλες τις άλλες πτυχές της ζωής» (Πηγή: <https://www.ox.ac.uk>).

Η επιστήμη των υπολογιστών χωρίζεται τόσο σε πρακτικά όσο και σε θεωρητικά πεδία. Ένα παράδειγμα θεωρητικού πεδίου είναι η θεωρία υπολογιστικής πολυπλοκότητας η οποία αφορά την μελέτη θεμελιωδών ιδιοτήτων των υπολογιστικών και ανυπέβλητων προβλημάτων ενώ ένα παράδειγμα πρακτικού πεδίου είναι τα γραφικά υπολογιστών τα οποία χρησιμοποιούνται σε πάρα πολλές εφαρμογές. Ωστόσο υπάρχουν και άλλοι τομείς όπως η θεωρία της γλώσσας προγραμματισμού η οποία εξετάζει διάφορες προσεγγίσεις στην περιγραφή του υπολογισμού, ενώ η μελέτη του προγραμματισμού ηλεκτρονικών υπολογιστών εξετάζει τις διάφορες πτυχές της χρήσης της γλώσσας προγραμματισμού και των σύνθετων συστημάτων.

Η επιστήμη υπολογιστών και κατ' επέκταση ένας επιστήμονας υπολογιστών διαφέρει από έναν προγραμματιστή ηλεκτρονικών υπολογιστών, καθώς ο μεν προγραμματιστής ασχολείται με την ανάπτυξη λογισμικού υπολογιστών ενώ ο επιστήμονας των υπολογιστών ασχολείται με την επιστημονική μελέτη των αλγορίθμων και της σχεδίασης, ανάλυσης και κατασκευής υλικού ή λογισμικού υπολογιστών και αποδοτικών, αυτοματοποιημένων υπολογιστικών

υποδομών για τη μηχανική εκτέλεση αλγορίθμων, αν και ο προγραμματιστής έχει σίγουρα σε ένα βαθμό διδαχθεί την επιστήμη των υπολογιστών. Το ίδιο ισχύει και για τους χρήστες διάφορων προγραμμάτων λογισμικού καθώς η χρήση αυτών δεν συνεπάγεται γνώση της επιστήμης των υπολογιστών. Από τα παραπάνω προκύπτει ότι η επιστήμη υπολογιστών χρησιμοποιεί τον προγραμματισμό όχι βασικό εργαλείο για ένα από τα αντικείμενα της.

## **1.1 Ιστορία της επιστήμης των υπολογιστών**

Τα πρώτα θεμέλια του τι είναι η επιστήμη της πληροφορικής προηγούνται της εφεύρεσης του σύγχρονου ψηφιακού υπολογιστή και τοποθετούνται ήδη στην αρχαιότητα. Η επιστήμη υπολογιστών εμφανίστηκε ταυτόχρονα με τις πρώτες υπολογιστικές μηχανές την δεκαετία του 1940. Ταυτόχρονα τότε τέθηκαν και τα πρώτα θεωρητικά θεμέλια του υπολογισμού. Η συνεχής τεχνολογική εξέλιξη αλλά και η εδραίωση της τεχνολογίας των πληροφοριών και της επικοινωνίας σε διεθνές επίπεδο ιδιαίτερα μετά την δεκαετία του 1970 οδήγησαν στην ευρεία αποδοχή της επιστήμης αυτής.

Το 1673, ο Gottfried Leibniz παρουσίασε μια ψηφιακή μηχανική αριθμομηχανή, που ονομάζεται ενώ μπορεί να θεωρηθεί ο πρώτος επιστήμονας πληροφορικής διότι, μεταξύ άλλων, τεκμηριώνει το σύστημα δυαδικών αριθμών. Ο Charles Babbage ήταν Βρετανός μαθηματικός, φιλόσοφος, εφευρέτης και μηχανικός ο οποίος επινόησε τον προγραμματίσιμο υπολογιστή. Θεωρείται ο «πατέρας του υπολογιστή» ενώ του αποδίδεται η εφεύρεση του πρώτου μηχανικού υπολογιστή. Ξεκίνησε το σχεδιασμό της πρώτης αυτόματης μηχανικής αριθμομηχανής, της Μηχανής Διαφορών το 1822, που του έδωσε τελικά την ιδέα της πρώτης προγραμματιζόμενης μηχανικής αριθμομηχανής, της επονομαζόμενης «Αναλυτικής Μηχανής» του. Το 1937, εκατό χρόνια μετά το αδύνατο όνειρο του Babbage, ο Howard Aiken έπεισε την IBM, η οποία έκανε όλα τα είδη εξοπλισμού διάτρησης καρτών και ήταν επίσης στην επιχείρηση αριθμομηχανών να αναπτύξει την τεράστια προγραμματιζόμενη αριθμομηχανή της, το ASCC / Harvard Mark I, με βάση την Αναλυτική Μηχανή του Babbage, η οποία χρησιμοποιούσε κάρτες και μια κεντρική μονάδα επεξεργασίας δεδομένων.



Κατά τη διάρκεια της δεκαετίας του 1940, καθώς αναπτύχθηκαν νέες και ισχυρότερες υπολογιστικές μηχανές, ο όρος ηλεκτρονικός υπολογιστής κατέστη σαφές ότι θα μπορούσε να χρησιμοποιηθεί για περισσότερους από απλούς μαθηματικούς υπολογισμούς. Ο χρόνος έχει σημειώσει σημαντικές βελτιώσεις στη χρηστικότητα και την αποτελεσματικότητα της τεχνολογίας των υπολογιστών. Η σύγχρονη κοινωνία έχει δει μια σημαντική μετατόπιση των χρηστών της τεχνολογίας των ηλεκτρονικών υπολογιστών σε μια σχεδόν πανταχού παρούσα βάση χρηστών.

Η επιστήμη υπολογιστών ταυτόχρονα έχει σχέση και με άλλους κλάδους της επιστήμης ακόμα και αν φαινομενικά έχουν σημαντικές διαφορές, έχουν κάποια πεδία τα οποία επικαλύπτονται. Κάποιες από αυτές είναι οι ακόλουθες:

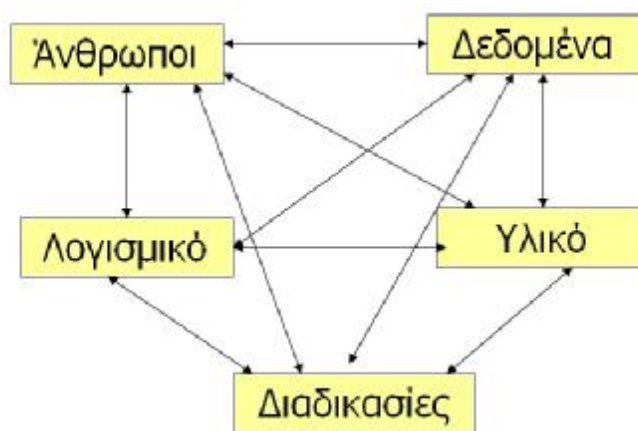
- Μαθηματική λογική
- Στατιστική
- Εφαρμοσμένα μαθηματικά
- Ηλεκτρονική
- Υπολογιστικά μαθηματικά
- Υπολογιστική φυσική κλπ.

## **1.2 Πληροφοριακά Συστήματα**

Πληροφοριακό σύστημα (Information System) είναι ένα σύνολο από μέσα συλλογής, αποθήκευσης, επεξεργασίας, ανάλυσης και διάχυσης της πληροφόρησης με στόχο την επίτευξη ενός συγκεκριμένου σκοπού κι αποτελέσματος. Ένα πληροφοριακό σύστημα μπορεί να είναι ανεξάρτητο, οπότε και να έχει υλοποιηθεί για ένα να εκτελέσει συγκεκριμένες εργασίες σε ένα καθαρά αυτόνομο περιβάλλον. Επίσης, μπορεί να λειτουργεί ως ανεξάρτητη μονάδα, αλλά στα πλαίσια ενός στόχου να λειτουργεί ως μέρος μιας ομάδας, οπότε και μιλάμε για ομαδοποιημένο πληροφοριακό σύστημα. Ο όρος πληροφοριακό σύστημα έχει επικρατήσει να αναφέρεται στα πληροφοριακά συστήματα που βασίζονται στην χρήση ηλεκτρονικών υπολογιστών. Για παράδειγμα, τα στοιχεία εξόδου ενός Π.Σ. μπορεί κάλλιστα να αποτελούν τα στοιχεία εισόδου ενός άλλου.

Τα πληροφοριακά συστήματα αποτελούνται από τα παρακάτω στοιχεία:

- Υλικό ηλεκτρονικών υπολογιστών (Hardware)
- Λογισμικό (Software)
- Δεδομένα
- Διαδικασίες
- Άνθρωποι



Εικόνα 1 Μέρη πληροφοριακού συστήματος (Πηγή: <http://ebooks.edu.gr>)

Θα μπορούσε κάποιος να πει πως τα πληροφοριακά συστήματα αποτελούν απαραίτητο συστατικό προκειμένου να μπορέσει ο άνθρωπος να χρησιμοποιήσει τα δεδομένα που έχει διαθέσιμα ώστε να είναι σε θέση να παράγει πληροφορίες τις οποίες στην συνέχεια θα επεξεργαστεί προς όφελός του. Ένας πιο ευρύς ορισμός των υπολογιστικών συστημάτων είναι ότι τα υπολογιστικά συστήματα αναφέρονται γενικά στην τεχνολογία της πληροφορίας και της επικοινωνίας που ένα άτομο ή ένας οργανισμός χρησιμοποιεί, αλλά επίσης και στο τρόπο με τον οποίο ο άνθρωπος αλληλοεπιδρά με την τεχνολογία για την υποστήριξη διάφορων διαδικασιών και την λήψη αποφάσεων. Από αυτό γίνεται σαφές ότι τα πληροφοριακά συστήματα σχετίζονται με τα συστήματα διαχείρισης βάσεων δεδομένων καθώς ένα

πληροφοριακό σύστημα αποτελεί μια μορφή επικοινωνίας του συστήματος εντός του οποίου τα δεδομένα δέχονται επεξεργασία.

### 1.3 Κατηγορίες πληροφοριακών συστημάτων

Τα πληροφοριακά συστήματα χωρίζονται σε διάφορες κατηγορίες ανάλογα με το είδος της πληροφόρησης και της υποστήριξης που έχουν οι χρήστες τους.

Ξεκινώντας, η πρώτη κατηγορία πληροφοριακών συστημάτων, είναι τα **Συστήματα Διεκπεραίωσης Επεξεργασιών** (Transaction Processing Systems – TPS). Αυτά τα πληροφοριακά συστήματα ξεκίνησαν να χρησιμοποιούνται στις επιχειρήσεις. Με τα συστήματα αυτά μπορεί να γίνει συλλογή, αποθήκευση, επεξεργασία, επίβλεψη- παρακολούθηση και διάχυση των αποτελεσμάτων κάποιων επιχειρηματικών ενεργειών. Τα δεδομένα συγκεντρώνονται σε ημερήσια βάση ή ακόμα και σε πραγματικό χρόνο και αποθηκεύονται σε βάσεις δεδομένων. Τα δεδομένα αυτά τελούν υπό επεξεργασία και τα αποτελέσματα είτε διοχετεύονται υπό τύπο αναφορών, καταστάσεων, εγγράφων σε κάθε ενδιαφερόμενο μέλος της επιχείρησης, είτε αποθηκεύονται για μελλοντική χρήση.

Μια άλλη κατηγορία πληροφοριακών συστημάτων είναι τα **Συστήματα αυτοματισμού γραφείου** (Office Automation Systems – OAS) που καλύπτουν τις γραμματειακές ανάγκες μιας επιχείρησης μέσω της υποστήριξης της επικοινωνίας και της παραγωγικότητας της επιχείρησης μέσω της αυτοματοποίησης της ροής της εργασίας και της ελαχιστοποίησης των προβλημάτων που προκύπτουν. Τα συστήματα αυτοματισμού γραφείου μπορούν να εφαρμοστούν σε όλα τα επίπεδα διοίκησης.

τα **Συστήματα υποστήριξης ομάδων** (Group Support Systems – GSS) είναι συστήματα τα οποία εξυπηρετούν εργαζομένους που δουλεύουν ομαδικά για την επίτευξη ενός κοινού στόχου. Κύριος σκοπός τους είναι να υποστηρίξουν την ομαδική εργασία.

Τα **Συστήματα υποστήριξης αποφάσεων** (Decision Support Systems – DSS) αναπτύσσονται για την κάλυψη αυτών των αναγκών κατά τη λήψη αποφάσεων σε στρατηγικό επίπεδο. Αντικείμενο τους είναι η υποστήριξη των διευθυντικών στελεχών της επιχείρησης στην εργασία τους και κυρίως στην λήψη αποφάσεων.

Τα **Διοικητικά πληροφοριακά συστήματα** (Executive Information Systems – EIS) είναι συστήματα τα οποία δίνουν πληροφορίες, συνήθως σε μορφή γραφημάτων, που προορίζονται για τα ανώτατα στελέχη διοικήσεως της επιχειρήσεως. Είναι μια σειρά από μηχανογραφικά εργαλεία με δυνατότητες χρωμάτων, γραφημάτων, φωνητικών εντολών, διασυνδέσεως φυσικής γλώσσας, κ.λπ. που βοηθούν τους διευθυντές σε ταχύτατη προσπέλαση, ανάλυση, σύνοψη και διάδοση μεγάλων όγκων στοιχείων (data) και πληροφοριών.

Τελευταία κατηγορία είναι τα **Ευφυή Συστήματα** υποστήριξης (Intelligent Support System – ISS) που είναι και τα πιο σύγχρονα και απευθύνονται σε άτομα που εργάζονται σαν μάνατζερ ή σαν εσωτερικοί εξειδικευμένοι σύμβουλοι και έχουν ανάγκη υποστήριξης σε κρίσιμα θέματα από τεχνικές της τεχνητής νοημοσύνης.

Οι διάφοροι τύποι πληροφοριακών συστημάτων μπορούν να λειτουργούν τόσο σαν ανεξάρτητες μονάδες όσο και σαν συστήματα που το ένα υποστηρίζει και υποβοηθά τις λειτουργίες του άλλου. Τα βασικά στοιχεία που καθορίζουν αυτού του είδους τις συνεργασίες είναι οι ανάγκες της επιχείρησης, ο χρόνος ανάπτυξης, το κόστος και οι ανάγκες των χρηστών.

## 1.4 Οργάνωση των Αρχείων

Η επεξεργασία και η οργάνωση δεδομένων όταν αναφερόμαστε σε Η/Υ γίνεται με την χρήση αρχείων εγγραφών. Τα στοιχεία οργανώνονται σε καρτέλες και περιέχουν όσες από αυτές χρειάζονται προκειμένου να είναι σε θέση ο χρήστης της Βάσης Δεδομένων να έχει συνολική πληροφόρηση η οποία θα τον βοηθά στο να κάνει τις εργασίες του. Σε μια εμπορική επιχείρηση για παράδειγμα ένα αρχείο πελατών θα περιείχε ενδεικτικά τα παρακάτω στοιχεία:

- Αναγνωριστικός Αριθμός Πελάτη
- Επωνυμία
- Όνομα
- Διεύθυνση
- ΤΚ
- Πόλη
- Τηλέφωνο
- ΑΦΜ
- Νομική μορφή εταιρείας
- ΔΟΥ
- Email

Πριν να χρησιμοποιηθούν οι Η/Υ στην οργάνωση των αρχείων, αυτή γινόταν με χειρόγραφο τρόπο από υπαλλήλους που είχαν αυτή την αρμοδιότητα. Η διαδικασία αυτή ήταν χρονοβόρα και απαιτούσε ιδιαίτερη προσοχή. Επίσης απαιτούσε και ειδικό χώρο για την αποθήκευση όλων αυτών των εγγράφων ενώ σε περίπτωση απώλειας τους δεν υπήρχε δυνατότητα ανάκτησης και γι' αυτό έπρεπε να κρατούνται αντίγραφα. Πλέον σχεδόν στο σύνολο των επιχειρήσεων αυτός ο τρόπος οργάνωσης έχει εκλείψει και πλέον το χώρο όπου φυλάσσονταν τα έγγραφα τον έχουν αντικαταστήσει τα αρχεία δεδομένων, την καρτέλα πελάτη οι εγγραφές του αρχείου δεδομένων και τα στοιχεία της καρτέλας τα πεδία της εγγραφής.

Σαν αρχείο ορίζεται ένα σύνολο που από οργανωμένα ομοειδή στοιχεία. Στη συνέχεια τα στοιχεία αυτά που περιλαμβάνονται σε ένα αρχείο τα οργανώνουμε σε λογικές ενότητες με το

κάθε σύνολο να ονομάζεται εγγραφή με το κάθε στοιχείο της εγγραφής να ονομάζεται πεδίο. Μήκος πεδίου ονομάζεται ο μέγιστος επιτρεπόμενος αριθμός χαρακτήρων που επιτρέπονται για να γίνει μια καταχώρηση.

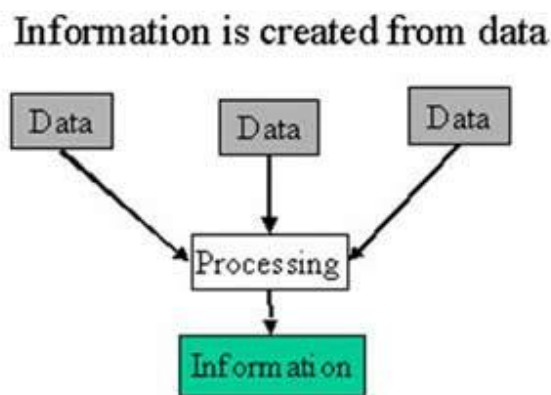
Σε μια οργάνωση αρχείου όπως είναι οι πελάτες μιας εμπορικής επιχείρησης που αναφέρθηκε προηγουμένως, τα αντίστοιχα πεδία όλων των εγγραφών καταλαμβάνουν τον ίδιο αριθμό σε bytes που είναι αυτός που έχουμε ορίσει κατά τη δημιουργία του αρχείου. Για παράδειγμα, αν ορίσαμε ότι το πεδίο Επώνυμο θα έχει μήκος 15 χαρακτήρες, τότε το πεδίο της εγγραφής του πελάτη με επώνυμο Παπαδόπουλος, αλλά και το πεδίο της εγγραφής του πελάτη με επώνυμο Ιωάννου θα καταλαμβάνουν από 15 bytes στη μνήμη του υπολογιστή, ενώ αν ένας πελάτης έχει όνομα με περισσότερα από 15 γράμματα, θα καταχωρηθούν στη μνήμη του υπολογιστή μόνο τα 15 πρώτα γράμματα.

Ένα πεδίο χαρακτηρίζεται ακόμη και από το είδος των δεδομένων που μπορεί να περιέχει, όπως:

- Αλφαριθμητικό, το οποίο μπορεί να περιέχει γράμματα, ψηφία ή και ειδικούς χαρακτήρες.
- Αριθμητικό, το οποίο περιέχει μόνο αριθμούς.
- Αλφαβητικό, το οποίο περιέχει μόνο γράμματα
- Ημερομηνίας, το οποίο περιέχει μόνο ημερομηνίες.
- Δυαδικό, στο οποίο μπορεί να περιέχονται άλλου είδους δεδομένα π.χ. ήχος
- Λογικό, στο οποίο μπορούν να καταχωρηθούν μόνο δυο τιμές. Με βάση την καταχωρούμενη τιμή χωρίζεται σε αληθής ή ψευδής η καταχώρηση.
- Σημειώσεων, στο οποίο μπορεί να γραφτεί κείμενο

## 1.5 Δεδομένα και Πληροφορίες

Σημαντικό είναι πριν την ανάλυση των Βάσεων Δεδομένων να διακρίνουμε τι είναι οι πληροφορίες και τι τα δεδομένα καθώς είναι σύνηθες να συγχέονται. Όπως αναφέρεται στο βιβλίο των Ξένου, Χριστουδουλάκη (2002) «Ως δεδομένα ορίζονται στοιχεία, έννοιες, και οδηγίες τυποποιημένα σε καθορισμένη μορφή κατάλληλα για επεξεργασία από ανθρώπους ή μηχανές», ενώ « η πληροφορία εξάγεται από τα δεδομένα και χρησιμοποιείται στην επίλυση προβλημάτων». Παρακάτω παρουσιάζονται και σχηματικά οι παραπάνω ορισμοί.



Εικόνα 2 Η πληροφορία ως προϊόν επεξεργασμένων δεδομένων (Πηγή:<http://www.differencebetween.info/difference-between-data-and-information>)

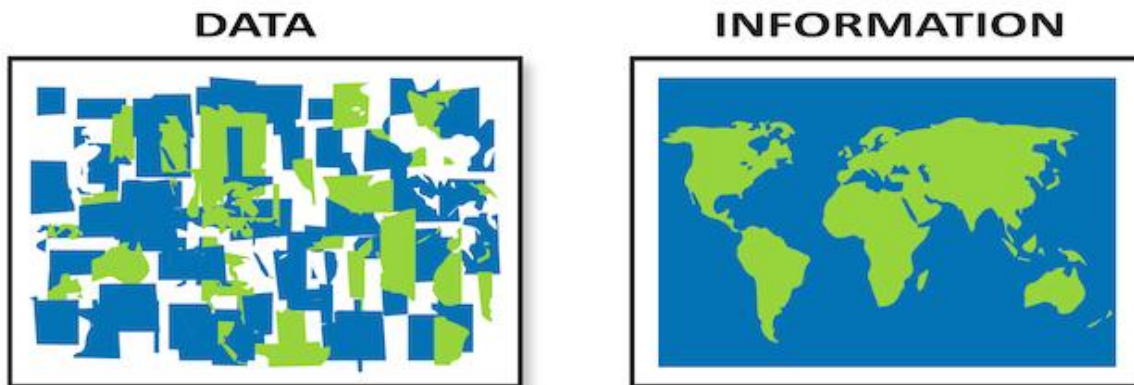
Ένας άλλος ορισμός για τα δεδομένα και τις πληροφορίες που δίνει ο Παπαθανασίου Ε. (2003) είναι ο παρακάτω:

«Η λέξη πληροφορία σημαίνει γνώση, είδηση, αλλά και βεβαίωση ή πιστοποίηση στοιχείων περί πραγμάτων, γεγονότων, ιδιοτήτων αντικειμένων, ή έμψυχων όντων και φαινομένων, τα οποία είναι διαμορφωμένα σε μια κατανοητή και διαχειρίσιμη μορφή. Τα στοιχεία, τα γεγονότα, οι ιδιότητες των αντικειμένων, καθώς και των έμψυχων όντων και φαινομένων συνηθίζεται να αποκαλούνται πληροφοριακά δεδομένα (data) ή πληροφοριακά στοιχεία.

Από τα παραπάνω προκύπτει ότι η πληροφορία και τα δεδομένα δεν είναι έννοιες ταυτόσημες. Αν και πολλές φορές στην καθημερινότητα των ανθρώπων οι έννοιες αυτές



χρησιμοποιούνται ως όμοιες εντούτοις η βασική τους διαφορά είναι ότι η πληροφορία πηγάζει από τα δεδομένα ή αλλιώς μια σύνθεση δεδομένων ύστερα από επεξεργασία γεννά την πληροφορία.



Εικόνα 3 Διαφορά δεδομένων και πληροφορίας (Πηγή: <http://effectualsystems.com/data-need-information/>)

Οι σύγχρονες Βάσεις Δεδομένων μέσα σε άλλα έχουν δημιουργεί γιατί βοηθούν στην καλύτερη οργάνωση των δεδομένων. Η μη σωστή οργάνωση των δεδομένων δημιουργεί πολλά προβλήματα όπως η αδυναμία εύρεσης των κατάλληλων πληροφοριών. Για παράδειγμα μια βιβλιοθήκη στην οποία τα βιβλία δεν είναι χωρισμένα σε κατηγορίες και εν συνεχεία με βάση τα ονόματα των συγγραφέων αλφαβητικά καθιστά την εύρεση ενός βιβλίου δύσκολη έως και αδύνατη. Μια τέτοια βιβλιοθήκη θα περιείχε μια τεράστια ποσότητα δεδομένων αλλά θα ήταν ουσιαστικά άχρηστος.

Όσον αφορά τις επιχειρήσεις και τις Βάσεις Δεδομένων που χρησιμοποιούν, ένα χαρακτηριστικό παράδειγμα δεδομένων για τα οποία απαιτείται να οργανώνονται με τρόπο σωστό και αποδοτικό είναι τα παρακάτω:

- Στοιχεία υπαλλήλων
- Στοιχεία πελατών
- Στοιχεία προμηθευτών
- Παραγγελίες.

- Υλικά σε μια αποθήκη
- Δαπάνες
- Έσοδα κλπ.

## 1.6 Προβλήματα της Οργάνωσης Αρχείων

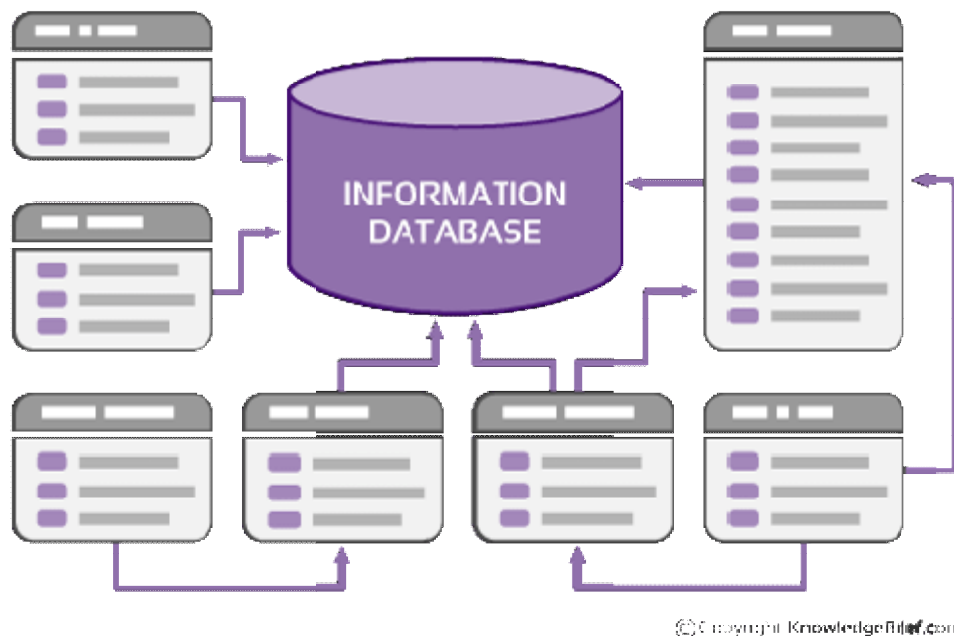
Είναι συνηθισμένο κατά την οργάνωση των αρχείων, να δημιουργούνται ξεχωριστά προγράμματα και ξεχωριστά αρχεία. Χαρακτηριστικό παράδειγμα είναι η δημιουργία διαφορετικών αρχείων για τα στοιχεία των πελατών και άλλου για τις παραγγελίες τους ή τους λογαριασμούς τους. Αυτός ο τρόπος οργάνωσης είναι προβληματικός και δημιουργεί τα παρακάτω προβλήματα:

- **Πλεονασμός δεδομένων (data redundancy).** Αυτό το πρόβλημα υφίσταται όταν τα ίδια δεδομένα επαναλαμβάνονται σε διαφορετικά αρχεία. Όπως αναφέρθηκε και προηγουμένως είναι σχεδόν βέβαιο ότι το αρχείο παραγγελιών των πελατών θα περιλαμβάνονται τα στοιχεία τους όπως και σε αυτό των λογαριασμών τους.
- **Ασυνέπεια των δεδομένων (data inconsistency).** Το πρόβλημα αυτό αναφέρεται στη περίπτωση που εμφανίζονται διαφορές στα καταχωρημένα δεδομένα. Τέτοιες αλλαγές μπορεί να υπάρξουν αν υπάρχουν τα ίδια στοιχεία π.χ. των πελατών τόσο στο αρχείο που περιέχει τα στοιχεία τους όσο και στο αρχείο με τις παραγγελίες τους και στην περίπτωση αλλαγής ενός στοιχείου δεν γίνουν οι αλλαγές και στα δυο αρχεία.
- **Αδυναμία μερισμού δεδομένων (data sharing).** Μερισμός δεδομένων σημαίνει δυνατότητα για κοινή χρήση των στοιχείων κάποιων αρχείων. Για παράδειγμα, ο μερισμός δεδομένων θα ήταν χρήσιμος αν με την παραγγελία ενός πελάτη μπορούμε να έχουμε πρόσβαση την ίδια στιγμή στο αρχείο πελατών για να δούμε το υπόλοιπο του πελάτη και μετά στο αρχείο της αποθήκης για να δούμε αν είναι διαθέσιμα τα προϊόντα που παρήγγειλε ο συγκεκριμένος πελάτης. Η αδυναμία μερισμού δεδομένων δημιουργεί καθυστέρηση στη λήψη αποφάσεων και στην εξυπηρέτηση των χρηστών.

- **Αδυναμία προτυποποίησης.** Αυτό το πρόβλημα έχει να κάνει με τον ανόμοιο τρόπο με τον οποίο παρουσιάζονται και οργανώνονται τα δεδομένα στα αρχεία. Ιδιαίτερα όταν αυτά έχουν δημιουργηθεί σε διαφορετικά προγράμματα υπάρχει πρόβλημα για τον χρήστη να αναζητήσει τα δεδομένα που θέλει αλλά και σε μια πιθανή ανταλλαγή δεδομένων μεταξύ αυτών των συστημάτων.

## 2. Βάσεις Δεδομένων

Ξεκινώντας την ανάλυση όσον αφορά τις βάσεις δεδομένων είναι σημαντικό να ξεκινήσουμε με τον ορισμό τους. Ο ορισμός όπως αναφέρεται στο βιβλίο των Ξένου, Χριστοδουλάκη (2002) αναφέρει ότι «βάση δεδομένων είναι μια αυστηρά τυποποιημένη συλλογή από σχετικά μεταξύ τους δεδομένα, που παρέχει ένα αυτόματο κεντρικό και κοινό τρόπο χειρισμού τους». Από τον ορισμό αυτό καταλαβαίνει κανείς ότι για να δημιουργηθεί και να είναι σημαντική μια βάση δεδομένων πρέπει να έχει δεδομένα που σχετίζονται μεταξύ τους. Αυτό σημαίνει ότι όταν απλά αποθηκεύονται δεδομένα που δεν έχουν σχέση μεταξύ τους σε έναν υπολογιστή αυτά δεν αποτελούν μια βάση δεδομένων. Όπως επίσης συμπληρώνουν οι συγγραφείς «μια Βάση Δεδομένων πρέπει να αντικατοπτρίζει ένα περιβάλλον του πραγματικού κόσμου» και «να έχουν λογική συνέχεια και νόημα». Πέρα από το γεγονός ότι τα δεδομένα πρέπει να έχουν μεταξύ τους συνάφεια ένα ακόμα σημαντικό χαρακτηριστικό των Βάσεων Δεδομένων είναι το γεγονός ότι αυτά τα δεδομένα μπορούν να χρησιμοποιηθούν και να επεξεργαστούν από το χρήστη με έναν αυτόματο κοινό τρόπο.



Εικόνα 4 Απεικόνιση Βάσης Δεδομένων (Πηγή: <https://www.kbmanage.com/concept/information-database-idb>)

Με βάση τα παραπάνω μπορούμε να πούμε ότι οι Βάσεις Δεδομένων δημιουργούνται και χρησιμοποιούνται προκειμένου να διευκολύνουν ένα ή περισσότερους χρήστες στην απεικόνιση και την λύση ενός προβλήματος. Τα δεδομένα που περιέχονται κάθε φορά σε μια Βάση Δεδομένων ωστόσο δε χρειάζεται να είναι στατικά, δηλαδή να είναι μόνο ιστορικά στοιχεία αλλά μπορούν να εμπλουτίζονται καθημερινά και να αλλάζουν προκειμένου η Βάση Δεδομένων να είναι επίκαιρη και να μην χάνει την χρησιμότητά της. Αυτή την καθημερινή καταχώρηση των στοιχείων την αναλαμβάνουν οι χρήστες (users) οι οποίοι πέρα από την χρήση της Βάσης Δεδομένων αναλαμβάνουν και όποιες ενέργειες απαιτούνται για την ομαλή λειτουργία της. Ο ορισμός που δίνεται για τους χρήστες από τους Ξένο και Χριστοδουλάκη (2002) αναφέρει ότι «χρήστες μιας Βάσης Δεδομένων είναι όσοι χρησιμοποιούν μια Βάση Δεδομένων, είτε για απόκτηση πληροφορίας είτε για συντήρηση της Βάσης Δεδομένων».

## **2.1 Συστήματα Διαχείρισης Βάσεων Δεδομένων**

Από την αρχική δημιουργία τους μέχρι και σήμερα οι Βάσεις Δεδομένων έχουν εξελιχθεί και αλλάξει αρκετά και αυτό οδήγησε στη δημιουργία των Συστημάτων Διαχείρισης Βάσεων Δεδομένων (Database Management Systems ή DBMS).

Αρχικά μπορούμε να πούμε ότι ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων είναι ένα εργαλείο το οποίο διευκολύνει τους χρήστες να εργάζονται με Βάσεις Δεδομένων. Μέσω αυτού του Συστήματος, οι χρήστες μπορούν να κατασκευάσουν και να χρησιμοποιήσουν Βάσεις Δεδομένων. Συνολικά ένας ορισμός που θα μπορούσε να δοθεί είναι ότι Σύστημα Διαχείρισης Βάσεων Δεδομένων είναι ένα σύστημα με βάση το οποίο έχουν δημιουργηθεί πολλές Βάσεις Δεδομένων από διαφορετικούς χρήστες. Οι δυνατότητες που έχει ένα τέτοιο Σύστημα Διαχείρισης Βάσεων Δεδομένων είναι (Ξένος, Χριστοδουλάκης 2002):

- Ορισμός της Βάσης Δεδομένων
- Κατασκευή της Βάσης Δεδομένων
- Διαγραφή της Βάσης Δεδομένων

- Χρήση της Βάσης Δεδομένων

-Όταν ο χρήστης ορίζει μια Βάση Δεδομένων ο χρήστης μπορεί να καθορίσει το μοντέλο της Βάσης Δεδομένων, να ορίσει τους τύπους δεδομένων που θα χρησιμοποιήσει και να ελέγξει τη Βάση Δεδομένων χωρίς να προχωρήσει σε κατασκευή της.

-Στην κατασκευή της Βάσης Δεδομένων οι τύποι των δεδομένων και τα δεδομένα αποθηκεύονται στο υλικό (hardware) του ηλεκτρονικού υπολογιστή με διαδικασίες που ελέγχονται από το Σύστημα Διαχείρισης Βάσεων Δεδομένων και δεν απασχολούν τον χρήστη.

-Στη διαγραφή της Βάσης Δεδομένων ο χρήστης αποφασίζει τον τερματισμό μιας Βάσης Δεδομένων και την απομάκρυνση των δεδομένων από το υλικό.

-Τέλος, κατά τη χρήση της Βάσης Δεδομένων ο χρήστης, είτε χειρίζεται τα δεδομένα (εισάγει νέα δεδομένα, τροποποιεί δεδομένα, ή διαγράφει δεδομένα), είτε υποβάλει ερωτήσεις στη Βάση Δεδομένων με στόχο την εξαγωγή πληροφοριών.

## **2.2 Διαφορά Βάσης Δεδομένων με Σύστημα Διαχείρισης Βάσεων Δεδομένων**

Σε αυτό το σημείο πρέπει να είναι ξεκάθαρη σε εσάς η διαφορά μεταξύ ενός Συστήματος Διαχείρισης Βάσεων Δεδομένων και μίας Βάσης Δεδομένων. Το Σύστημα Διαχείρισης Βάσεων Δεδομένων είναι λογισμικό το οποίο διευκολύνει τους χρήστες να υλοποιήσουν Βάσεις Δεδομένων. Αντίθετα η Βάση Δεδομένων υλοποιείται με τη βοήθεια ενός Συστήματος Διαχείρισης Βάσεων Δεδομένων που χειρίζεται δεδομένα τα οποία αποθηκεύονται στο υλικό του υπολογιστή. Η όλη διαδικασία χειρισμού και αποθήκευσης των δεδομένων διευκολύνεται από το Σύστημα Διαχείρισης Βάσεων Δεδομένων, το οποίο αναλαμβάνει τη μετατροπή των εντολών του χρήστη σε εντολές προς τον υπολογιστή και λειτουργίες που σχετίζονται με το χειρισμό των δεδομένων στο υλικό του υπολογιστή.

Ο χρήστης μιας Βάσης Δεδομένων επιτελεί λειτουργίες που μπορούν να ομαδοποιηθούν στις παρακάτω: εισαγωγή δεδομένων, διαγραφή δεδομένων, αλλαγή δεδομένων και ανάκληση δεδομένων. Όλες οι παραπάνω λειτουργίες προϋποθέτουν επικοινωνία με το υλικό του υπολογιστή (π.χ. το μαγνητικό δίσκο του υπολογιστή). Αυτή η επικοινωνία γίνεται μέσω του λογισμικού του Συστήματος Διαχείρισης Βάσεων Δεδομένων. Η επικοινωνία αυτή συνήθως δεν είναι ορατή από τον χρήστη της Βάσης Δεδομένων, ο οποίος απλά βλέπει τα αποτελέσματα από τις λειτουργίες που ζήτησε να επιτελούνται. Αυτό είναι

και το σημείο που κάνει αρκετούς χρήστες να συγχέουν τις έννοιες Βάση Δεδομένων και Σύστημα Διαχείρισης Βάσεων Δεδομένων.

## 2.3 Οφέλη από τη χρήση των Βάσεων Δεδομένων

### 1. Ταυτόχρονη προσπέλαση

Το κύριο κέρδος από την χρήση Βάσεων Δεδομένων είναι το γεγονός ότι ο χρήστης ή οι χρήστες μπορούν να έχουν ταυτόχρονη πρόσβαση στα δεδομένα. Επίσης οι χρήστες των Βάσεων μπορούν ταυτόχρονα να εργάζονται πάνω σε αυτή γνωρίζοντας ότι η ταυτόχρονη προσπέλαση δεν δημιουργεί προβλήματα σε αυτή. Αυτό συμβαίνει διότι το Σύστημα Διαχείρισης της Βάσης Δεδομένων είναι έτσι προγραμματισμένο ώστε να αποκλείει ταυτόχρονες αλλαγές στα δεδομένα από δύο ή περισσότερους χρήστες. Μέσω του Συστήματος Διαχείρισης της Βάσης Δεδομένων γίνεται επίσης έλεγχος για το ποιος χρήστης έχει πρόσβαση στα εκάστοτε δεδομένα. Δηλαδή φροντίζει ώστε μη εξουσιοδοτημένα άτομα να μην έχουν πρόσβαση στα δεδομένα ή να έχουν σε ένα μέρος αυτών. Επιπλέον το κάθε άτομο έχει πρόσβαση μόνο στα δεδομένα για τα οποία έχει εξουσιοδότηση και του επιτρέπονται του μόνο οι πράξεις (π.χ. διαγραφή, ή νέα εγγραφή) αυτές για τις οποίες έχει εξουσιοδότηση.

### 2. Ταχύτητα εξαγωγής απαντήσεων

Ένας από τους κυριότερους λόγους για τους οποίους δημιουργήθηκαν οι βάσεις δεδομένων είναι για να αποφευχθεί ένα σημαντικό μέρος προσωπικής εργασίας από τον άνθρωπο στην αναζήτηση και επεξεργασία πληροφοριών. Η χρήση μιας Βάσης Δεδομένων του εξασφαλίζει ότι με εύκολο τρόπο θα μπορεί κάποιος να πάρει την απαιτούμενη πληροφορία από ένα μεγάλο πλήθος πληροφοριών χωρίς να απαιτείται κόπος. Για παράδειγμα στην περίπτωση που κάποιος θέλει να βρει ποια ημέρα και σε ποια πόλη υπήρξε το μεγαλύτερο ύψος βροχόπτωσης θα έπρεπε δίχως την ύπαρξη μιας βάσης δεδομένων να ανοίξει πολλές χιλιάδες έγγραφα και να τα επεξεργαστεί. Τα αποτελέσματα αυτής της μεθόδου θα αργούσαν πολύ να εξαχθούν και θα απαιτούσαν πάρα πολύ μεγάλο κόπο ενώ θα υπήρχε και μεγάλη πιθανότητα να έχει γίνει κάποιο λάθος. Συνεπώς η χρήση παραδοσιακών μοντέλων οργάνωσης δεν είναι εξυπηρετική, αφού η παροχή τέτοιου είδους πληροφορίας είναι πρακτικά ανέφικτη. Ο λόγος για τον οποίο αυτού του είδους η οργάνωση δεν είναι εξυπηρετική είναι διότι ο τρόπος που οργανώνονται



τα αρχεία αυτά δεν εξυπηρετούν την γρήγορη εξαγωγή απαντήσεων σε διάφορα ερωτήματα που τίθενται.

Αντίθετα, η χρήση μιας Βάσης Δεδομένων δίνει ακριβώς αυτή τη δυνατότητα. Ο χρήστης μπορεί άμεσα και σε πολύ μικρό χρόνο να ζητήσει και να πάρει όποιες πληροφορίες επιθυμεί. Αυτές οι πληροφορίες μπορεί να είναι κάποιο απλό ερώτημα, π.χ. ταξινόμηση από το μικρότερο στο μεγαλύτερο ή μπορεί να είναι κάποιο πολύ σύνθετο ερώτημα. Και στις δυο περιπτώσεις τα αποτελέσματα θα είναι άμεσα διαθέσιμα στο χρήστη και μάλιστα με αξιοπιστία.

### 3. Ευελιξία στις αλλαγές

Τα δεδομένα που έχουν συλλεχθεί πολλές φορές χρειάζεται να εμπλουτιστούν ή να διαφοροποιηθούν καθώς αλλάζει και στην πραγματικότητα το είδος των πληροφοριών που χαρακτηρίζει μια κατάσταση ή ένα μοντέλο. Αυτές οι αλλαγές σε ένα παραδοσιακό σύστημα καταγραφής είναι δύσκολο έως αδύνατο να πραγματοποιηθούν. Αντίθετα, ο τρόπος οργάνωσης των δεδομένων σε μία Βάση Δεδομένων, μπορεί στις περισσότερες περιπτώσεις να δεχτεί αλλαγές και μπορεί να προσαρμοστεί με τέτοιο τρόπο ώστε να εξυπηρετεί τις ανάγκες του χρήστη.

### 4. Ποιότητα και αξιοπιστία δεδομένων

Σε μια Βάση Δεδομένων υπάρχει η δυνατότητα μέσω του Συστήματος Διαχείρισης Βάσεων Δεδομένων να ελεγχθεί το περιεχόμενο και η ορθότητα των δεδομένων όπως και να γίνουν εξειδικευμένοι έλεγχοι π.χ. για διπλοεγγραφές ή παραβιάσεις της ορθότητας των δεδομένων. Με αυτό τον τρόπο τα δεδομένα διατηρούν την ποιότητα και την αξιοπιστία τους καθώς δεν είναι δυνατόν να υπάρξουν λάθος καταχωρίσεις. Έτσι η οργάνωση των δεδομένων είναι καλύτερη και ο χρήστης έχει την σιγουριά ότι τα δεδομένα που του επιστρέφει η Βάση Δεδομένων ως απάντηση στις ερωτήσεις του είναι σωστά.

## 2.4 Λόγοι για την αποφυγή χρήσης Βάσεων Δεδομένων

Αν και είναι δεδομένο πως οι Βάσεις Δεδομένων παρέχουν πάρα πολλά πλεονεκτήματα στον χρήστη κατά την χρήση τους σε αντίθεση με οποιοδήποτε άλλο σύστημα καταχώρησης δεδομένων π.χ. χειρόγραφο ή απλό αρχείο στον Η/Υ, παρόλα αυτά υπάρχουν και περιπτώσεις στις οποίες η χρήση Βάσης Δεδομένων δεν είναι η καλύτερη επιλογή. Είναι σαφές πως μια Βάση Δεδομένων είναι προτιμότερη όταν κάποιος έχει να επεξεργαστεί μεγάλο όγκο δεδομένων ή δεδομένα που χωρίζονται σε πολλές κατηγορίες ή δεδομένα που συνεχώς ενημερώνονται και καταχωρούνται συνεχώς νέα. Ωστόσο, υπάρχουν και περιπτώσεις που η χρήση μιας Βάσης Δεδομένων δεν αποτελεί ιδανική επιλογή. Η πιο συνηθισμένη τέτοια περίπτωση είναι το κόστος. Καθώς η υλοποίηση μίας Βάσης Δεδομένων περιέχει κάποιο κόστος τόσο για την αγορά όσο και για την συντήρηση της, το κόστος χρήσης συνολικά ενός Συστήματος Διαχείρισης Βάσεων Δεδομένων μπορεί να αποτελέσει σημαντικό παράγοντα για την μη χρήση ενός τέτοιου συστήματος. Ωστόσο με τα σημερινά δεδομένα, το κόστος των Συστημάτων Διαχείρισης Βάσεων Δεδομένων έχει μειωθεί σημαντικά και πλέον είναι δυνατόν κάποιος να λειτουργεί μια Βάση Δεδομένων σε έναν απλό Η/Υ με πολύ μικρό κόστος.

Μια άλλη περίπτωση στην οποία δεν χρειάζεται η χρήση μιας Βάσης Δεδομένων είναι όταν τα δεδομένα είναι στατικά. Αυτό σημαίνει ότι τα δεδομένα είναι ήδη διαθέσιμα και δεν πρόκειται να αλλάξουν. Ωστόσο αυτή η περίπτωση δεν πρέπει να συγχέεται με την περίπτωση όπου σε στατικά δεδομένα γίνεται επεξεργασία με σκοπό την ανάκτηση πληροφοριών. Επίσης όταν υπάρχει μικρός όγκος δεδομένων ή για δεδομένα που έχει πρόσβαση μόνο ένας χρήστης συνήθως δεν είναι απαραίτητη η υλοποίηση Βάσης Δεδομένων.

## 2.5 Επίπεδα χρηστών

Τα επίπεδα χρηστών σε μια Βάση Δεδομένων είναι 3: Ο Διαχειριστής του Συστήματος Διαχείρισης Βάσεων Δεδομένων, ο Ιδιοκτήτης μιας Βάσης Δεδομένων και οι Χρήστες οι οποίοι χωρίζονται σε χρήστες υψηλής και κατώτερης διαβάθμισης.

Ξεκινώντας, ο Διαχειριστής του Συστήματος Διαχείρισης Βάσεων Δεδομένων είναι ένα άτομο που είναι επιφορτισμένο με τον έλεγχο και την συντήρηση του συστήματος και γι' αυτό απαιτείται να είναι ένα άτομο με υψηλή κατάρτιση πάνω στο αντικείμενο αυτό. Στις αρμοδιότητες του είναι επίσης ο έλεγχος των πόρων του συστήματος, η δημιουργία ή εγκατάσταση εφαρμογών που χρησιμοποιούνται για τη δημιουργία ή τη σχεδίαση Βάσεων Δεδομένων, η επίβλεψη της δημιουργίας ή της διαγραφής Βάσης Δεδομένων και η ανάθεση πόρων του συστήματος για την καλή λειτουργία των Βάσεων Δεδομένων. Συνοπτικά μπορούμε να πούμε ότι ο Διαχειριστής έχει ως κύριο αντικείμενο την ομαλή λειτουργία του Συστήματος Διαχείρισης Βάσεων Δεδομένων από τους χρήστες και την υποστήριξη αυτών στο έργο τους καθώς πρέπει να είναι σε θέση να τους παρέχει τα κατάλληλα εργαλεία για να χειρίζονται τις Βάσεις Δεδομένων αλλά και τυχόν άλλων εφαρμογών που τις συνοδεύουν.

Η δεύτερη κατηγορία χρηστών είναι αυτή των Ιδιοκτητών μιας Βάσης Δεδομένων. Αυτοί συχνά ονομάζονται και Διαχειριστές Βάσης Δεδομένων. Ο Ιδιοκτήτης μιας Βάσης Δεδομένων είναι το άτομο το οποίο έχει δημιουργήσει την Βάση Δεδομένων όπως και τις εφαρμογές που απαιτούνται για την προσπέλασή της. Έχει σαν αρμοδιότητα να φροντίζει για τους πόρους υλικού και λογισμικού σε συνεργασία με τον Διαχειριστή του Συστήματος. Επίσης, είναι αυτός που ορίζει τους χρήστες της Βάσης Δεδομένων του και καθορίζει τα επίπεδα διαβάθμισης του κάθε χρήστη, παρέχει δικαιώματα, και αναλαμβάνει και την διαγραφή των χρηστών οι οποίοι ολοκλήρωσαν τις δραστηριότητές τους στη Βάση Δεδομένων. Ο Ιδιοκτήτης μιας Βάσης Δεδομένων συχνά έχει μαζί του και μια ομάδα τεχνικής υποστήριξης με την οποία συνεργάζεται για την υλοποίηση των εφαρμογών προσπέλασης της Βάσης Δεδομένων, δηλαδή των προγραμμάτων λογισμικού μέσω των οποίων οι υπόλοιποι χρήστες θα μπορούν να έχουν πρόσβαση στα δεδομένα της Βάσης Δεδομένων.

Συνεχίζοντας, το τρίτο επίπεδο χρηστών μιας Βάσης Δεδομένων είναι οι Χρήστες Υψηλής Διαβάθμισης. Πρόκειται για χρήστες στους οποίους ο Ιδιοκτήτης της Βάσης Δεδομένων έχει δώσει αυξημένες αρμοδιότητες. Οι δυνατότητες τους είναι η εισαγωγή δεδομένων, η διαγραφή δεδομένων, η δυνατότητα αλλαγών στα δεδομένα αλλά και η ανάγνωση δεδομένων. Η δυνατότητα των χρηστών υψηλής διαβάθμισης να προκαλούν αλλαγές στα δεδομένα της Βάσης Δεδομένων είναι το κύριο χαρακτηριστικό που τους διαφοροποιεί από τους άλλους χρήστες όπως θα δούμε στη συνέχεια. Επίσης σε μερικούς χρήστες υψηλής διαβάθμισης δίνονται από τον ιδιοκτήτη της Βάσης περισσότερα δικαιώματα για να τον διευκολύνουν στο έργο του. Για παράδειγμα, εκχωρείται το δικαίωμα χειρισμού χρηστών σε κάποιον επιλεγμένο Χρήστη Υψηλής Διαβάθμισης, έτσι ώστε να τον διευκολύνει στη δημιουργία και διαγραφή χρηστών.

Η τελευταία κατηγορία χρηστών είναι οι Χρήστες Κατώτερης Διαβάθμισης. Πρόκειται για χρήστες που χρησιμοποιούν τη Βάση Δεδομένων χωρίς όμως να έχουν δικαίωμα να κάνουν αλλαγές στα δεδομένα. Συνήθως αυτοί λαβαίνουν πληροφορίες από τη Βάση Δεδομένων τις οποίες έχουν εισάγει οι Χρήστες Υψηλής Διαβάθμισης. Υπάρχει δυνατότητα και σε αυτούς τους χρήστες να χωριστούν σε διαφορετικά επίπεδα διαβάθμισης τα οποία καθορίζει συνήθως ο Ιδιοκτήτης της Βάσης Δεδομένων. Η κύρια εργασία των χρηστών αυτών περιορίζεται σε ότι έχει να κάνει με την πληροφόρησή τους από τα δεδομένα που περιέχονται στη Βάση Δεδομένων.

### 3. Η γλώσσα Βάσεων Δεδομένων SQL

Η SQL (Structured Query Language) είναι μία πλήρης γλώσσα Βάσεων Δεδομένων. Έχει σχεδιαστεί για την διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (Relational Database Management System, RDBMS).



Εικόνα 5 Λογότυπο MySQL (Πηγή: [www.wikipedia.com](http://www.wikipedia.com))

Η SQL είναι δομημένη σε βάσεις οι οποίες περιέχουν πίνακες οι οποίοι στη συνέχεια αποτελούνται από γραμμές και στήλες. Κάθε πίνακας στη συνέχεια, περιέχει πεδία για το κάθε ένα πρέπει να οριστεί ο τύπος δεδομένων που θα περιέχει. Προκειμένου να είναι δυνατή η αποθήκευση τιμών στα πεδία θα πρέπει αυτές να καταχωρούνται σύμφωνα με τον τύπο δεδομένων που έχει προηγουμένως οριστεί. Επίσης όπως θα αναλυθεί και στην συνέχεια ορίζεται και το μέγεθος του πεδίου σύμφωνα με τις ανάγκες του χρήστη της βάσης.

### 3.1 Θεωρία Σχισιακών Βάσεων Δεδομένων

Όπως αναφέρουν οι Welling και Thompson (2009) οι σχεσιακές Βάσεις Δεδομένων είναι ο πιο συνηθισμένος τύπος Βάσης Δεδομένων. Αυτές αποτελούνται από σχέσεις ή αλλιώς πίνακες. Ένας πίνακας περιέχει δεδομένα με διάφορα στοιχεία πχ για τους πελάτες όπως παρουσιάζεται στην παρακάτω εικόνα.

```
1 • SELECT name, surname, address, phone FROM Employee ORDER BY surname, name DESC; |
```

result Grid Filter Rows:  Export: Wrap Cell Content:

#	name	surname	address	phone
1	Γιώργος	Αλεξίου	Αμερικής 62	2108964362
2	Νικόλαος	Ανδρεάδης	Τσαμαδού Ανα...	2610456728
3	Νικόλαος	Βασιλείου	Κορίνθου 55	2610456728
4	Γιάννης	Γουδής	Παντανάσσης ...	2102376542
5	Γιάννης	Δημητρίου	Βεάκη 12	2102376542
6	Σπύρος	Καλάρης	Πουκεβίλ Φρα...	2107834665
7	Λευτέρης	Κοντολέων	Κανάρη 48	2610896741
8	Σπύρος	Μαρινάκης	Μενελάου 38	2107834665
9	Γιώργος	Οικονομάκης	Καραόλη 124	2108964362
10	Λευτέρης	Παπανίκος	Αγ. Ανδρέου 200	2610896741

Εικόνα 6: Πίνακας δεδομένων

Στον παραπάνω πίνακα περιέχεται το όνομα, το επώνυμο, η διεύθυνση και το τηλέφωνο των πελατών. Αυτά τα στοιχεία περιέχονται σε διαφορετικές στήλες όπου κάθε στήλη αντιστοιχεί σε διαφορετικά δεδομένα και γραμμές οι οποίες αντιστοιχούν σε διαφορετικούς πελάτες. Κάθε στήλη του πίνακα έχει διαφορετικό όνομα και περιέχει διαφορετικά δεδομένα. Επίσης, κάθε στήλη περιέχει διαφορετικό τύπο δεδομένων. Οι στήλες επίσης μπορούν να ονομαστούν πεδία ή ιδιότητες. Συνεχίζοντας, οι γραμμές αναφέρονται η κάθε μια σε διαφορετικό πελάτη. Οι γραμμές ονομάζονται επίσης εγγραφές.

### 3.2 Κανόνες για την σχεδίαση των πινάκων

Στην SQL ακολουθούνται οι παρακάτω απλοί κανόνες για την σχεδίαση των πινάκων σε μια Βάση Δεδομένων.

- Χρησιμοποιούνται αλφαριθμητικοί χαρακτήρες και όσον αφορά τα γράμματα αυτά είναι μόνο πεζά.
- Δεν χρησιμοποιούνται κενά και προκειμένου να χωρίζονται οι λέξεις χρησιμοποιείται ο χαρακτήρας « \_ ».
- Τα ονόματα στηλών πρέπει να είναι περιγραφικά και μοναδικά για κάθε πίνακα.
- Δεσμευμένες λέξεις της MySQL δεν πρέπει να χρησιμοποιούνται (π.χ. SELECT).

### 3.3 Κανόνες για την εκτέλεση εντολών και την εισαγωγή τιμών

Προκειμένου να είναι δυνατή η εκτέλεση εντολών και να μπορεί ο χρήστης να εισάγει τιμές στις Βάσεις Δεδομένων της SQL ακολουθούνται συνοπτικά οι παρακάτω κανόνες.

- Οι αριθμητικές τιμές δεν έχουν κόμματα και δεν μπαίνουν μέσα σε εισαγωγικά.
- Οι αλφαριθμητικές τιμές μπαίνουν πάντα σε εισαγωγικά (char, varchar και text).
- Οι ημερομηνίες και ώρες καταχωρούνται πάντα με εισαγωγικά.
- Στις συναρτήσεις δεν μπορούμε να βάλουμε εισαγωγικά.
- Όταν ολοκληρώνεται μια εντολή ή ένα ερώτημα βάζουμε ελληνικό ερωτηματικό.
- Οι τιμές που καταχωρούνται στα ερωτήματα ή στις εντολές μπαίνουν σε παρένθεση και χωρίζονται μεταξύ τους με κόμμα.
- Δεν βάζουμε κόμμα ανάμεσα σε εντολές, ονόματα πινάκων ή συναρτήσεις.
- Η τιμή NULL δεν μπαίνει σε εισαγωγικά.
- Για να Δει ο χρήστης τις εντολές και τα ερωτήματα που έχει πληκτρολογήσει πρέπει να πατήσει το πλήκτρο F1 και μετά μετακινείται με τα βέλη πάνω και κάτω.

### 3.4 Εντολές της MySQL

Κάποιες από εντολές που χρησιμοποιούμε στη MySQL φαίνονται πιο κάτω.

- CREATE δημιουργεί πίνακες η Βάσεις Δεδομένων.
- DELETE διαγράφει τιμές από πίνακες.
- DROP διαγράφει ολόκληρο τον πίνακα ή την Βάση Δεδομένων.
- UPDATE ενημερώνει τον πίνακα.
- INSERT εισάγει τιμές στον πίνακα.
- ALTER αλλάζει την δομή του πίνακα.
- SHOW εμφανίζει στοιχεία της δομής του πίνακα ή της Βάσης Δεδομένων.
- SELECT επιλέγει δεδομένα από τον πίνακα.



### 3.5 Εφαρμογή σε MySQL

Σε αυτή την ενότητα παρουσιάζονται τα ερωτήματα τα οποία έγιναν με την χρήση MySQL και αφορούν την βάση δεδομένων που αναφέρθηκε προηγουμένως. Τα ερωτήματα γίνονται με την χρήση της εντολής SELECT. Η εντολή SELECT επιλέγει στήλες (columns) δεδομένων από μια βάση δεδομένων. Το αποτέλεσμα αποθηκεύεται σε μορφή πίνακα και αποκαλείται result set. Την χρησιμοποιούμε για να εμφανίζουμε (επιλέγουμε) πληροφορίες από έναν πίνακα ως εξής :

```
SELECT ονόματα_στηλών FROM όνομα_πίνακα
```

Πριν ξεκινήσουμε την ανάλυση των ερωτημάτων παρουσιάζεται συνοπτικά ο τρόπος δημιουργίας της Βάσης Δεδομένων. Στην παρακάτω εικόνα παρουσιάζεται η εντολή CREATE με την οποία δίνεται η εντολή για την δημιουργία της Βάσης Δεδομένων με όνομα "company"

```
mysql> CREATE DATABASE company;  
Query OK, 1 row affected (0.01 sec)
```

*Εικόνα 7: Δημιουργία Βάσης Δεδομένων*

Στην συνέχεια ακολουθεί η δημιουργία των πινάκων. Για λόγους συντομίας θα παρατεθούν ενδεικτικά δύο από τους πίνακες που δημιουργήθηκαν. Ο πρώτος πίνακας που δημιουργήθηκε είναι ο πίνακας "Employee" ο οποίος αναφέρεται στους υπαλλήλους της επιχείρησης. Όπως φαίνεται στην εικόνα που ακολουθεί αρχικά με την εντολή CREATE δίνεται η εντολή για την δημιουργία του πίνακα. Στην συνέχεια δημιουργούνται τα πεδία του εν λόγω πίνακα. Σε αυτά είναι το επώνυμο, το όνομα, η διεύθυνση, ο ΑΦΜ, το τηλέφωνο καθώς επίσης το είδος των δεδομένων που μπορούν να καταχωρηθούν στα πεδία οι οποίες είναι εδώ αλφαριθμητικές και το ότι τα πεδία αυτά δεν δέχονται κενές τιμές.

```
mysql> CREATE TABLE `Employee` (
-> `surname` varchar(45) NOT NULL,
-> `name` varchar(45) NOT NULL,
-> `address` varchar(45) NOT NULL,
-> `afm` varchar(9) NOT NULL,
-> `phone` varchar(45) NOT NULL
-> ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.05 sec)
```

*Εικόνα 8: Δημιουργία Πίνακα*

Μετά την δημιουργία του πίνακα ακολουθεί η εισαγωγή δεδομένων σε αυτόν. Η εντολή INSERT INTO χρησιμοποιείται για να επιτευχθεί αυτό. Όπως φαίνεται και στην παρακάτω εικόνα η εντολή ξεκινά διαλέγοντας τον πίνακα (Employee) και στην συνέχεια τα πεδία στα οποία θα καταχωρηθούν τα δεδομένα ('surname', 'name', 'address', 'afm', 'phone'). Στην συνέχεια προχωρούμε στην καταχώρηση γραμμών δεδομένων στην Βάση Δεδομένων μας.

```
mysql> INSERT INTO `Employee` (`surname`, `name`, `address`, `afm`, `phone`) VALUES
-> ('Γουδής', 'Γιάννης', 'Παντανάσσης 61', '023451630', '2102376542'),
-> ('Καλάρης', 'Σπύρος', 'Πουκεβίλ Φρανσουά 22', '033461187', '2107834665'),
-> ('Αλεξίου', 'Γιώργος', 'Αμερικής 62', '034127690', '2108964362'),
-> ('Ανδρεάδης', 'Νικόλαος', 'Τσαμαδού Αναστάσιου 39', '098742199', '2610456728'),
-> ('Κοντολέων', 'Λευτέρης', 'Κανάρη 48', '099765422', '2610896741'),
-> ('Δημητρίου', 'Γιάννης', 'Βεάκη 12', '023451630', '2102376542'),
-> ('Μαρινάκης', 'Σπύρος', 'Μενελάου 38', '033461187', '2107834665'),
-> ('Οικονομάκης', 'Γιώργος', 'Καραόλη 124', '034127690', '2108964362'),
-> ('Βασιλείου', 'Νικόλαος', 'Κορίνθου 55', '098742199', '2610456728'),
-> ('Παπανίκος', 'Λευτέρης', 'Αγ. Ανδρέου 200', '099765422', '2610896741');
Query OK, 10 rows affected (0.01 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

*Εικόνα 9: Καταχώρηση δεδομένων σε ένα πίνακα*

Συνεχίζοντας με τον πίνακα που αφορά τους πελάτες (customer). Όπως φαίνεται στην εικόνα που ακολουθεί αρχικά με την εντολή CREATE δίνεται η εντολή για την δημιουργία του πίνακα. Στην συνέχεια δημιουργούνται τα πεδία του εν λόγω πίνακα. Σε αυτά είναι ο κωδικός πελάτη, το όνομα, η νομική μορφή της επιχείρησης, η διεύθυνση κλπ. Επίσης ορίζεται το είδος των δεδομένων που μπορούν να καταχωρηθούν στα πεδία οι οποίες είναι εδώ αλφαριθμητικές και το ότι τα πεδία αυτά δεν δέχονται κενές τιμές.

```
mysql> CREATE TABLE `customer` (  
-> `customer_id` int(11) NOT NULL,  
-> `name` varchar(45) NOT NULL,  
-> `type` varchar(45) NOT NULL,  
-> `address` varchar(45) NOT NULL,  
-> `doy` varchar(45) NOT NULL,  
-> `afmEmployee` varchar(9) NOT NULL  
-> ) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
Query OK, 0 rows affected (0.03 sec)
```

Εικόνα 10: Δημιουργία πίνακα

Μετά την δημιουργία του πίνακα ακολουθεί η εισαγωγή δεδομένων σε αυτόν. Όπως φαίνεται και στην παρακάτω εικόνα η εντολή ξεκινά διαλέγοντας τον πίνακα (customer) και στην συνέχεια τα πεδία στα οποία θα καταχωρηθούν τα δεδομένα. Στην συνέχεια προχωρούμε στην καταχώρηση γραμμών δεδομένων στην Βάση Δεδομένων μας.

```
mysql> INSERT INTO `customer` (`customer_id`, `name`, `type`, `address`, `doy`,  
`afmEmployee`) VALUES (1,'BOOK STOP CAFE','ΑΕ','Αλίμου 79','Α Αθηνών','033461187'  
'),(2,'ΘΕΜΕΛΙΟ ΓΝΩΣΗΣ','ΟΕ','Λ. Στρατού 300','Β Θεσ/νίκης','034127690'),(3,'COPY  
- COPY','ΑΕ','Αγ. Ανδρέου 201','Β Πατρών','098742199'),(4,'ΕΠΙΠΕΔΟ ΒΙΒΛΙΟΠΩΛΕΙΟ  
' ,'ΕΕ','Μαρκοπούλου 21','Α Πατρών','099765422'),(5,'ΠΑΛΕΤΑ','ΑΕ','Κεφαλληνίας 20  
' ,'Γ Πατρών','023451630'),(6,'ΠΙΞΕΛ','ΑΕ','Πολυτεχνείου 300','Α Πατρών','0987421  
99');  
Query OK, 6 rows affected (0.01 sec)  
Records: 6 Duplicates: 0 Warnings: 0
```

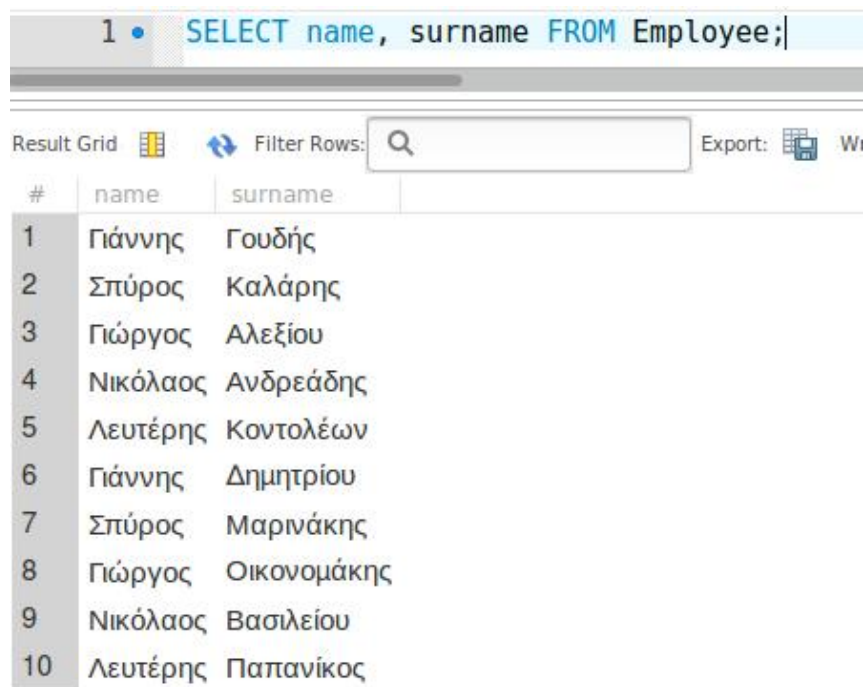
Εικόνα 11: Καταχώρηση δεδομένων σε ένα πίνακα

Με τον ίδιο τρόπο δημιουργούνται και οι υπόλοιποι πίνακες της Βάσης που χρησιμοποιείται εδώ και καταχωρούνται επίσης τα δεδομένα σε αυτούς.

### 3.6 Εκτέλεση ερωτημάτων με την χρήση της εντολής SELECT

**ΕΡΩΤΗΜΑ 1:** SELECT ΥΠΑΛΛΗΛΟΣ.ΟΝΟΜΑ, ΥΠΑΛΛΗΛΟΣ.ΕΠΩΝΥΜΟ FROM ΥΠΑΛΛΗΛΟΣ;

Παίρνουμε ως αποτέλεσμα τον παρακάτω πίνακα ο οποίος περιέχει το όνομα και το επώνυμο των υπαλλήλων από τον πίνακα που περιέχει τα στοιχεία των υπαλλήλων.



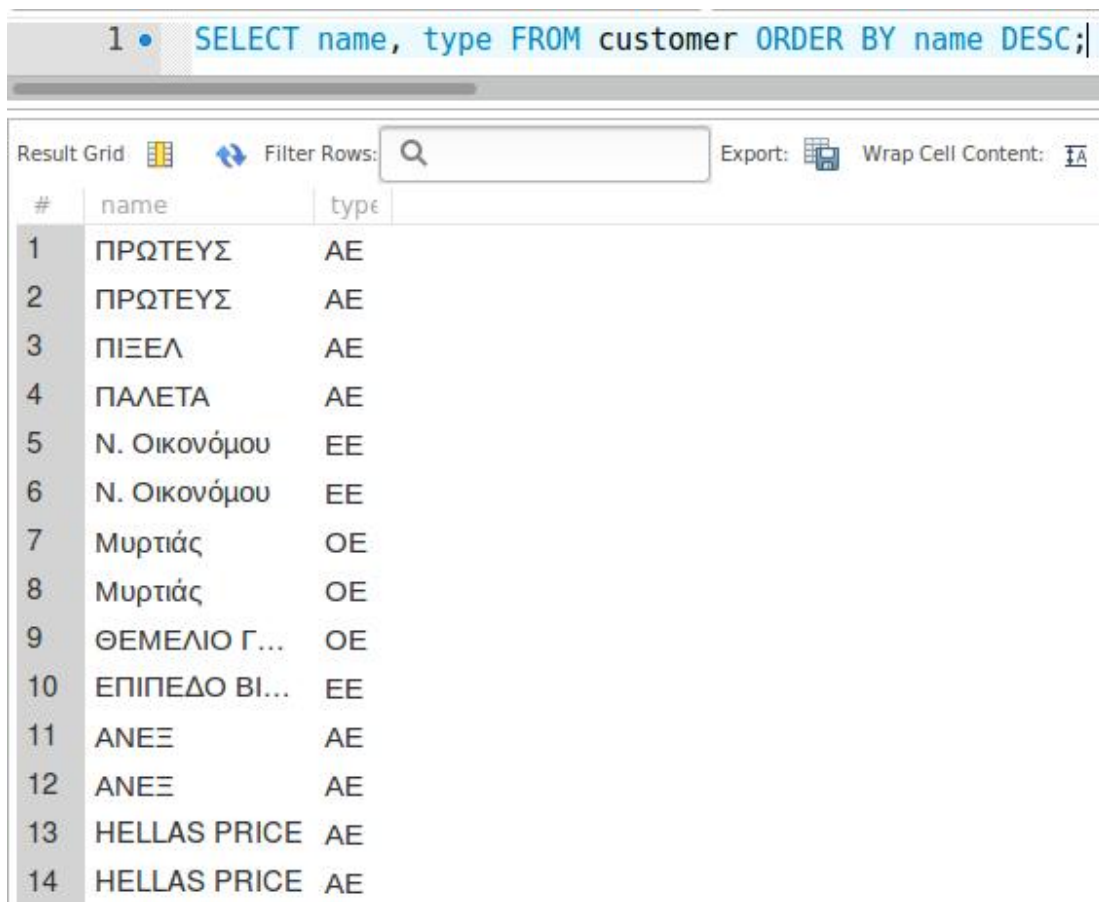
The screenshot shows a database query execution interface. At the top, the SQL statement is displayed: `1 • SELECT name, surname FROM Employee;`. Below the statement, there is a toolbar with options for 'Result Grid', 'Filter Rows', and 'Export'. The main area displays a table with 10 rows of results. The table has three columns: '#', 'name', and 'surname'. The data is as follows:

#	name	surname
1	Γιάννης	Γουδής
2	Σπύρος	Καλάρης
3	Γιώργος	Αλεξίου
4	Νικόλαος	Ανδρεάδης
5	Λευτέρης	Κοντολέων
6	Γιάννης	Δημητρίου
7	Σπύρος	Μαρινάκης
8	Γιώργος	Οικονομάκης
9	Νικόλαος	Βασιλείου
10	Λευτέρης	Παπανίκος




Εικόνα 12: Ερώτημα 1

**ΕΡΩΤΗΜΑ 2:** SELECT ΠΕΛΑΤΗΣ.ΕΠΩΝΥΜΙΑ, ΠΕΛΑΤΗΣ.ΝΟΜΙΚΜΟΡΦΗ FROM ΠΕΛΑΤΗΣ ORDER BY ΠΕΛΑΤΗΣ.ΕΠΩΝΥΜΙΑ;

Παίρνουμε ως αποτέλεσμα τον παρακάτω πίνακα ο οποίος περιέχει την επωνυμία των πελατών καθώς και την νομική μορφή της εταιρείας τους. Στο πίνακα αυτό η ταξινόμηση των αποτελεσμάτων έχει γίνει με βάση την επωνυμία των πελατών.



```
1 • SELECT name, type FROM customer ORDER BY name DESC;
```

Result Grid  Filter Rows:  Export:  Wrap Cell Content: 

#	name	type
1	ΠΡΩΤΕΥΣ	ΑΕ
2	ΠΡΩΤΕΥΣ	ΑΕ
3	ΠΙΞΕΛ	ΑΕ
4	ΠΑΛΕΤΑ	ΑΕ
5	Ν. Οικονόμου	ΕΕ
6	Ν. Οικονόμου	ΕΕ
7	Μυρτιάς	ΟΕ
8	Μυρτιάς	ΟΕ
9	ΘΕΜΕΛΙΟ Γ...	ΟΕ
10	ΕΠΙΠΕΔΟ ΒΙ...	ΕΕ
11	ΑΝΕΞ	ΑΕ
12	ΑΝΕΞ	ΑΕ
13	HELLAS PRICE	ΑΕ
14	HELLAS PRICE	ΑΕ




Εικόνα 13: Ερώτημα 2

**ΕΡΩΤΗΜΑ 3:** SELECT ΥΠΑΛΛΗΛΟΣ.ΟΝΟΜΑ, ΥΠΑΛΛΗΛΟΣ.ΕΠΩΝΥΜΟ, ΥΠΑΛΛΗΛΟΣ.ΔΙΕΥΘΥΝΣΗ, ΥΠΑΛΛΗΛΟΣ.ΤΗΛΕΦΩΝΟ FROM ΥΠΑΛΛΗΛΟΣ ORDER BY ΥΠΑΛΛΗΛΟΣ.ΟΝΟΜΑ, ΥΠΑΛΛΗΛΟΣ.ΕΠΩΝΥΜΟ;

Παίρνουμε ως αποτέλεσμα τον παρακάτω πίνακα ο οποίος περιέχει τα στοιχεία των υπαλλήλων στα οποία περιέχονται όνομα, επώνυμο, διεύθυνση και τηλέφωνο. Η ταξινόμηση γίνεται με βάση το επώνυμο και το όνομα.

1 • SELECT name, surname, address, phone FROM Employee ORDER BY surname, name DESC; |

---

result Grid  Filter Rows:  Export:  Wrap Cell Content: 

#	name	surname	address	phone
1	Γιώργος	Αλεξίου	Αμερικής 62	2108964362
2	Νικόλαος	Ανδρεάδης	Τσαμαδού Ανα...	2610456728
3	Νικόλαος	Βασιλείου	Κορίνθου 55	2610456728
4	Γάβνης	Γουδής	Παντανάσσης ...	2102376542
5	Γάβνης	Δημητρίου	Βεάκη 12	2102376542
6	Σπύρος	Καλάρης	Πουκεβίλ Φρα...	2107834665
7	Λευτέρης	Κοντολέων	Κανάρη 48	2610896741
8	Σπύρος	Μαρινάκης	Μενελάου 38	2107834665
9	Γιώργος	Οικονομάκης	Καραόλη 124	2108964362
10	Λευτέρης	Παπανίκος	Αγ. Ανδρέου 200	2610896741





Εικόνα 14: Ερώτημα 3



**ΕΡΩΤΗΜΑ 4:** SELECT ΠΕΛΑΤΗΣ.ΕΠΩΝΥΜΙΑ, ΠΕΛΑΤΗΣ.ΝΟΜΙΚΜΟΡΦΗ, ΠΕΛΑΤΗΣ.ΑΡΜΟΔΔΟΥ  
FROM ΠΕΛΑΤΗΣ WHERE (((ΠΕΛΑΤΗΣ.ΑΡΜΟΔΔΟΥ)="Α ΠΑΤΡΩΝ"));

Παίρνουμε ως αποτέλεσμα τον παρακάτω πίνακα ο οποίος περιέχει την επωνυμία και την νομική μορφή των επιχειρήσεων των πελατών οι οποίοι έχουν ως αρμόδια ΔΟΥ την Α' ΔΟΥ Πατρών.

```
1 • SELECT name, type, doy FROM customer WHERE doy = 'Α Πατρών';
```

result Grid   Filter Rows:  Export:  Wrap Cell Content: 

#	name	type	doy
1	ΕΠΙΠΕΔΟ ΒΙ...	ΕΕ	Α Πατρών
2	ΠΙΞΕΛ	ΑΕ	Α Πατρών
3	Ν. Οικονόμου	ΕΕ	Α Πατρών
4	AdventurePatras	ΑΕ	Α Πατρών
5	Ν. Οικονόμου	ΕΕ	Α Πατρών
6	AdventurePatras	ΑΕ	Α Πατρών

Εικόνα 15: Ερώτημα 4

**ΕΡΩΤΗΜΑ 5:** SELECT ΥΠΑΛΛΗΛΟΣ.ΟΝΟΜΑ, ΥΠΑΛΛΗΛΟΣ.ΕΠΩΝΥΜΟ, ΥΠΑΛΛΗΛΟΣ.ΔΙΕΥΘΥΝΣΗ, ΥΠΑΛΛΗΛΟΣ.ΤΗΛΕΦΩΝΟ FROM ΥΠΑΛΛΗΛΟΣ WHERE (((ΥΠΑΛΛΗΛΟΣ.ΤΗΛΕΦΩΝΟ) Like "210\*"));

Παίρνουμε ως αποτέλεσμα τον παρακάτω πίνακα ο οποίος περιέχει τα στοιχεία των υπαλλήλων (όνομα, επώνυμο, διεύθυνση, τηλέφωνο) των οποίων το τηλέφωνο αρχίζει από «210».

```
1 • SELECT name, surname, address, phone FROM Employee WHERE phone like '210%';|
```

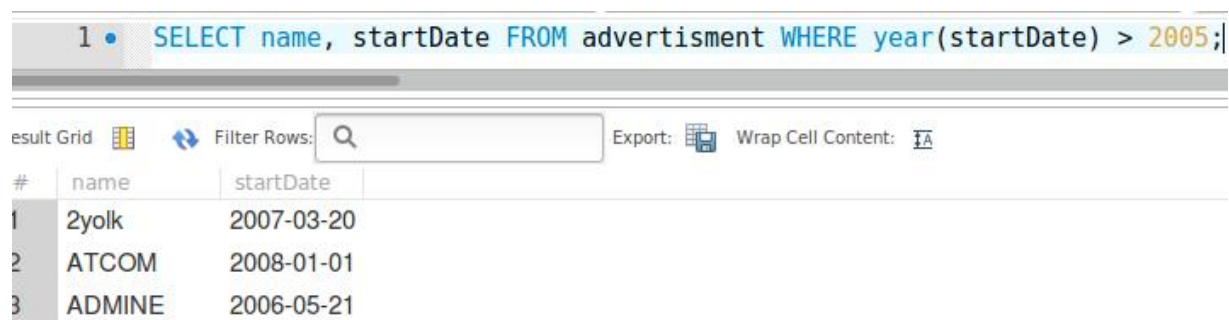
#	name	surname	address	phone
1	Γάννης	Γουδής	Παντανάσσης ...	2102376542
2	Σπύρος	Καλάρης	Πουκεβίλ Φρα...	2107834665
3	Γιώργος	Αλεξίου	Αμερικής 62	2108964362
4	Γάννης	Δημητρίου	Βεάκη 12	2102376542
5	Σπύρος	Μαρινάκης	Μενελάου 38	2107834665
6	Γιώργος	Οικονομάκης	Καραόλη 124	2108964362

Εικόνα 16: Ερώτημα 5



**ΕΡΩΤΗΜΑ 6:** SELECT ΔΙΑΦΗΜΙΣΗ.ΕΠΩΝΥΜΙΑ, ΔΙΑΦΗΜΙΣΗ.ΧΡΟΝΕΝΑΡΔΙΑΦ FROM ΔΙΑΦΗΜΙΣΗ WHERE (((Year([ΧΡΟΝΕΝΑΡΔΙΑΦ])>2005));

Παίρνουμε ως αποτέλεσμα τον παρακάτω πίνακα ο οποίος περιέχει τα ονόματα των διαφημιστικών εταιρειών καθώς και την ημερομηνία που ξεκίνησαν οι διαφημίσεις όσον όμως ξεκινούν μετά το 2005.



The screenshot shows a SQL query editor with the following query: `1 • SELECT name, startDate FROM advertisement WHERE year(startDate) > 2005;`

Below the query editor, there is a toolbar with options: "Result Grid" (checked), "Filter Rows" (with a search icon), "Export" (with a document icon), and "Wrap Cell Content" (with a text icon).

The results are displayed in a table with the following columns: "#", "name", and "startDate".

#	name	startDate
1	2yolk	2007-03-20
2	ATCOM	2008-01-01
3	ADMINE	2006-05-21

Εικόνα 17: Ερώτημα 6

**ΕΡΩΤΗΜΑ 7:** SELECT ΠΡΟΙΟΝ.ΚΠΡΟΙΟΝΤΟΣ, ΠΡΟΙΟΝ.ΟΝΟΜΠΡΟΙΟΝΤΟΣ, ΠΡΟΙΟΝ.ΤΙΜΠΡΤΕΜΑΧ, [ΤΙΜΠΡΤΕΜΑΧ]\*0.2 AS ΦΠΑ FROM ΠΡΟΙΟΝ;

Παίρνουμε ως αποτέλεσμα τον παρακάτω πίνακα ο οποίος περιέχει από τον πίνακα ΠΡΟΙΟΝ τις στήλες που αναγράφουν τον κωδικό του προϊόντος, το όνομα του, την τιμή και το ΦΠΑ το οποίο είναι η τιμή πολλαπλασιασμένη επί 0,2.

```
1 • SELECT product_id, name, price, (price * 0.2) fpa FROM product;
```

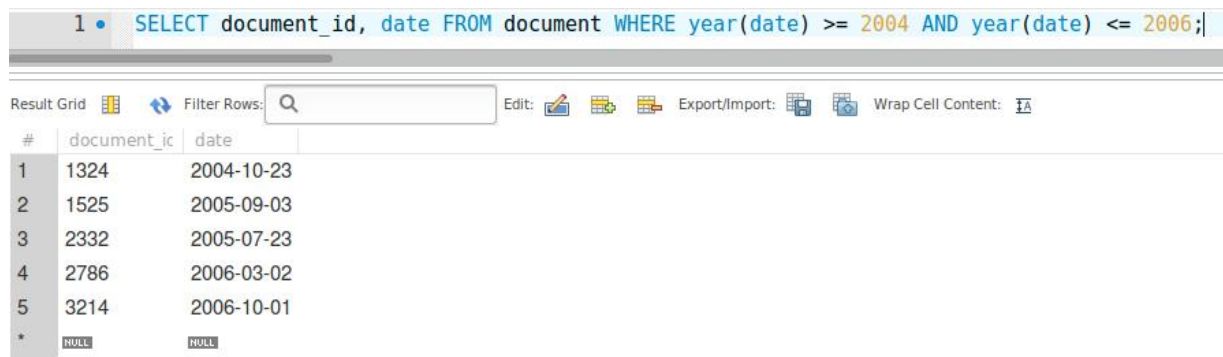
Result Grid Filter Rows:  Export: Wrap Cell Content:

#	product_id	name	price	fpa
1	500	ΤΣΑΚΙΣΜΕΝΗ ΑΚΤΗ	2.40	0.480
2	501	ΜΙΧΑΕΛ ΚΟΛΧΑΑΣ	1.20	0.240
3	502	ΣΗΜΕΙΟ ΩΜΕΓΑ	6.00	1.200
4	503	ΤΑ ΠΡΩΤΑ ΒΗΜΑΤΑ ΤΟΥ ΨΥ...	1.20	0.240
5	504	ΤΑ ΠΡΟΣΩΠΑ ΤΟΥ ΞΕΝΟΥ	6.00	1.200
6	505	ΜΕΓΑΛΟΙ ΡΩΣΟΙ ΣΥΓΓΡΑΦΕΙΣ	6.00	1.200
7	506	ΑΓΑΠΩ, ΡΗΜΑ ΑΜΕΤΑΒΑΤΟ	7.20	1.440
8	507	ΨΗΛΑ ΠΑΝΩ ΑΠΟ ΤΟΝ ΚΟ...	8.40	1.680

Εικόνα 18: Ερώτημα 7

**ΕΡΩΤΗΜΑ 8:** SELECT ΠΑΡΑΣΤΑΤΙΚΟ.ΑΡΠΑΡΑΣΤΑΤΙΚΟΥ, ΠΑΡΑΣΤΑΤΙΚΟ.ΗΜΕΡΟΜΕΚΔΟΣΗΣ FROM ΠΑΡΑΣΤΑΤΙΚΟ WHERE (((Year([ΗΜΕΡΟΜΕΚΔΟΣΗΣ])>=2004 And (Year([ΗΜΕΡΟΜΕΚΔΟΣΗΣ])<=2006));

Παίρνουμε ως αποτέλεσμα τον παρακάτω πίνακα ο οποίος περιέχει από τον πίνακα των παραστατικών τον αριθμό του παραστατικού και την ημερομηνία έκδοσης για τα παραστατικά που εκδόθηκαν από το 2004 έως το 2006



```
1 • SELECT document_id, date FROM document WHERE year(date) >= 2004 AND year(date) <= 2006;
```

#	document_id	date
1	1324	2004-10-23
2	1525	2005-09-03
3	2332	2005-07-23
4	2786	2006-03-02
5	3214	2006-10-01
*	NULL	NULL

Εικόνα 19 : Ερώτημα 8

**ΕΡΩΤΗΜΑ 9:** SELECT ΠΕΛΑΤΗΣ.ΕΠΩΝΥΜΙΑ, ΠΕΛΑΤΗΣ.ΔΙΕΥΘΕΔΡΑΣ, ΠΕΛΑΤΗΣ.ΑΡΜΟΔΔΟΥ  
FROM ΠΕΛΑΤΗΣ WHERE (((ΠΕΛΑΤΗΣ.ΝΟΜΙΚΜΟΡΦΗ)<>"ΑΕ"));

Παίρνουμε ως αποτέλεσμα τον παρακάτω πίνακα ο οποίος περιέχει την επωνυμία, την νομική μορφή, την διεύθυνση και την ΔΟΥ των εταιρειών των πελατών με κριτήριο η νομική μορφή να μην είναι Α.Ε.

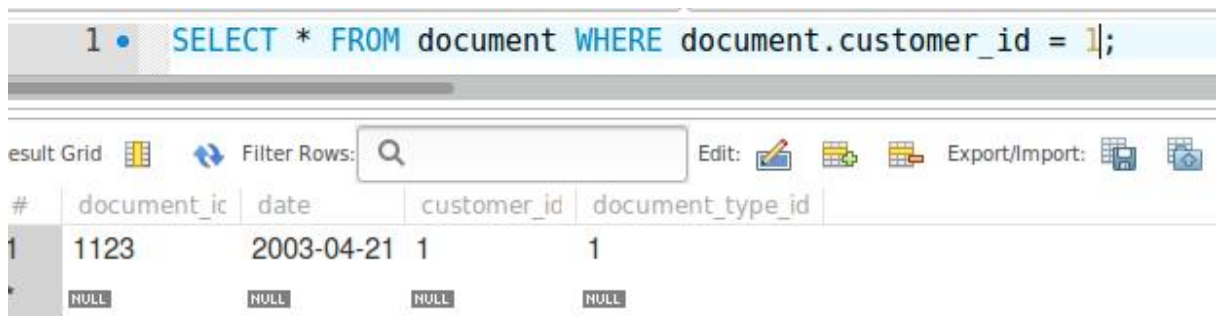
```
1 • SELECT name,type, address, doy FROM customer WHERE type != 'ΑΕ';|
```

#	name	type	address	doy
1	ΘΕΜΕΛΙΟ Γ...	ΟΕ	Λ. Στρατού 300	Β Θεσ/νίκης
2	ΕΠΙΠΕΔΟ ΒΙ...	ΕΕ	Μαρκοπούλου 21	Α Πατρών
3	Μυρτιάς	ΟΕ	Λ. Στρατού 300	Β Θεσ/νίκης
4	Ν. Οικονόμου	ΕΕ	Μαρκοπούλου 21	Α Πατρών
5	Μυρτιάς	ΟΕ	Λ. Στρατού 300	Β Θεσ/νίκης
6	Ν. Οικονόμου	ΕΕ	Μαρκοπούλου 21	Α Πατρών

Εικόνα 20 : Ερώτημα 9

**ΕΡΩΤΗΜΑ 10:** SELECT ΠΑΡΑΣΤΑΤΙΚΟ.ΑΡΠΑΡΑΣΤΑΤΙΚΟΥ, ΠΑΡΑΣΤΑΤΙΚΟ.ΗΜΕΡΟΜΕΚΔΟΣΗΣ, ΠΑΡΑΣΤΑΤΙΚΟ.ΚΠΕΛΑΤΗ, ΠΑΡΑΣΤΑΤΙΚΟ.ΚΕΙΔΟΥΣΠΑΡ FROM ΠΑΡΑΣΤΑΤΙΚΟ WHERE (((ΠΑΡΑΣΤΑΤΙΚΟ.ΚΠΕΛΑΤΗ)=[ΠΛΗΚΤΡΟΛΟΓΗΣΤΕ ΚΠΕΛΑΤΗ]));

Παίρνουμε ως αποτέλεσμα τον παρακάτω πίνακα ο οποίος περιέχει τον αριθμό παραστατικού, την ημερομηνία έκδοσης του, τον κωδικό πελάτη και το είδος παραστατικού σύμφωνα με τον κωδικό πελάτη που θα εισάγει ο χρήστης



#	document_ic	date	customer_id	document_type_id
1	1123	2003-04-21	1	1
	NULL	NULL	NULL	NULL

Εικόνα 21 : Ερώτημα 10

Τα επόμενα ερωτήματα αφορούν κάποιες εντολές της SQL οι οποίες παρουσιάζονται στη συνέχεια. Η εντολή DELETE χρησιμοποιείται για να διαγράψουμε γραμμές από έναν πίνακα. Η σύνταξή της είναι ως εξής :

DELETE FROM όνομα\_πίνακα

WHERE όνομα\_στήλης = τιμή

Η εντολή INSERT INTO εισάγει νέες γραμμές σ' έναν πίνακα. Η σύνταξή της είναι ως εξής :

INSERT INTO όνομα\_πίνακα

VALUES (τιμή1, τιμή2, ...)

Μπορούμε επίσης να καθορίσουμε τις στήλες για τις οποίες θέλουμε να εισάγουμε δεδομένα :

INSERT INTO όνομα\_πίνακα(στήλη1, στήλη2, ...)

VALUES (τιμή1, τιμή2, ...)

**ΕΡΩΤΗΜΑ 12:** DELETE ΔΙΑΦΗΜΙΣΗ.ΧΡΟΝΕΝΑΡΔΙΑΦ, Year([ΧΡΟΝΕΝΑΡΔΙΑΦ]) FROM ΔΙΑΦΗΜΙΣΗ WHERE (((Year([ΧΡΟΝΕΝΑΡΔΙΑΦ]))<2006));

Διαγράφει όλες τις Διαφημίσεις με Ημερομηνία Έναρξης πριν το 2006.

**ΕΡΩΤΗΜΑ 13:** . DELETE ΠΑΡΑΣΤΑΤΙΚΟ.ΚΠΕΛΑΤΗ FROM ΠΑΡΑΣΤΑΤΙΚΟ WHERE (((ΠΑΡΑΣΤΑΤΙΚΟ.ΚΠΕΛΑΤΗ)="304"));

Διαγράφει όλα τα Παραστατικά του Πελάτη με κωδικό 304

**ΕΡΩΤΗΜΑ 14:** SELECT \* INTO ΑΝΤΙΓΡΑΦΟ FROM ΠΕΛΑΤΗΣ;

Δημιουργεί στην τρέχουσα βάση δεδομένων ένα αντίγραφο του πίνακα customer με όνομα copyEmployee

**ΕΡΩΤΗΜΑ 15:** SELECT \* INTO ΑΝΤΙΓΡΑΦΟΥΠΑΛΛΗΛΟΣ FROM ΥΠΑΛΛΗΛΟΣ;

Δημιουργεί στην τρέχουσα βάση δεδομένων ένα αντίγραφο του πίνακα Employee με όνομα copyEmployee.

**ΕΡΩΤΗΜΑ 16:** UPDATE ΠΡΟΙΟΝ SET ΠΡΟΙΟΝ.ΤΙΜΠΡΤΕΜΑΧ = [ΤΙΜΠΡΤΕΜΑΧ]\*"1,20";

Πραγματοποιεί αύξηση της Τιμής(price) όλων των Προϊόντων (product) κατά 20%

**ΕΡΩΤΗΜΑ 17:** UPDATE ΕΞΕΙΔΙΚΕΥΣΗ SET ΕΞΕΙΔΙΚΕΥΣΗ.ΕΞΕΙΔΙΚΕΥΣΗ = "ΛΟΓΙΣΤΗΣ Α" WHERE (((ΕΞΕΙΔΙΚΕΥΣΗ.ΕΞΕΙΔΙΚΕΥΣΗ)="ΛΟΓΙΣΤΗΣ"));

Πραγματοποιεί τροποποίηση της Εξειδίκευσης(specialisation.specialisation) «Λογιστής» σε «Λογιστής Α».

**ΕΡΩΤΗΜΑ 18:** INSERT INTO ΠΕΛΑΤΗΣ SELECT \* FROM ΑΝΤΙΓΡΑΦΟ;

Προσαρτά τις εγγραφές του πίνακα copy στον πίνακα customer.

**ΕΡΩΤΗΜΑ 19:** INSERT INTO ΥΠΑΛΛΗΛΟΣ SELECT \* FROM ΑΝΤΙΓΡΑΦΟΥΠΑΛΛΗΛΟΣ;

Προσαρτά τις εγγραφές του πίνακα copyEmployee στον πίνακα Employee.

**ΕΡΩΤΗΜΑ 20:** SELECT ΥΠΑΛΛΗΛΟΣ.ΑΦΜ, ΥΠΑΛΛΗΛΟΣ.ΟΝΟΜΑ, ΥΠΑΛΛΗΛΟΣ.ΕΠΩΝΥΜΟ, ΕΞΕΙΔΙΚΕΥΣΗ.ΕΞΕΙΔΙΚΕΥΣΗ FROM ΥΠΑΛΛΗΛΟΣ INNER JOIN ΕΞΕΙΔΙΚΕΥΣΗ ΟΝ ΥΠΑΛΛΗΛΟΣ.ΑΦΜ = ΕΞΕΙΔΙΚΕΥΣΗ.ΑΦΜ ORDER BY ΥΠΑΛΛΗΛΟΣ.ΕΠΩΝΥΜΟ;

Εμφανίζει ως αποτέλεσμα τον παρακάτω πίνακα ο οποίος περιέχει το ΑΦΜ, το όνομα, το επώνυμο όλων των Υπαλλήλων, καθώς και την εξειδίκευσή τους, ταξινομημένα ανά επώνυμο.

```
1 • SELECT e.afm, e.name, e.surname, GROUP_CONCAT(DISTINCT(s.specialisation) SEPARATOR ', ') spec
2 FROM Employee AS e
3 LEFT JOIN specialisation AS s ON s.afm = e.afm GROUP BY e.afm ORDER BY e.surname;
```

afm	name	surname	spec
098742199	Νικόλαος	Ανδρεάδης	Συντηρητής Ιατρ. Συντηρητής Η/Υ
023451630	Γάννης	Γουδής	Λογιστής Α, Πωλητής
033461187	Σπύρος	Καλάρης	Λογιστής Α, Ταμίας
034127690	Γιώργος	Οικονομάκης	Γραμματέας, Δακτυλογράφος
099765422	Λευτέρης	Παπανίκος	Πωλητής

Εικόνα 22 : Ερώτημα 20

**ΕΡΩΤΗΜΑ 21:** SELECT ΠΡΟΙΟΝ.ΟΝΟΜΠΡΟΙΟΝΤΟΣ, ΠΕΡΙΛΑΜΒΑΝΕΙ.ΑΡΠΑΡΑΣΤΑΤΙΚΟΥ FROM ΠΡΟΙΟΝ INNER JOIN (ΠΑΡΑΣΤΑΤΙΚΟ INNER JOIN ΠΕΡΙΛΑΜΒΑΝΕΙ ON ΠΑΡΑΣΤΑΤΙΚΟ.ΑΡΠΑΡΑΣΤΑΤΙΚΟΥ = ΠΕΡΙΛΑΜΒΑΝΕΙ.ΑΡΠΑΡΑΣΤΑΤΙΚΟΥ) ON ΠΡΟΙΟΝ.ΚΠΡΟΙΟΝΤΟΣ = ΠΕΡΙΛΑΜΒΑΝΕΙ.ΚΠΡΟΙΟΝΤΟΣ WHERE (((ΠΡΟΙΟΝ.ΟΝΟΜΠΡΟΙΟΝΤΟΣ)="ΣΗΜΕΙΟ ΩΜΕΓΑ"));

Εμφανίζει ως αποτέλεσμα τον παρακάτω πίνακα ο οποίος περιέχει το όνομα προϊόντος και τον αριθμό παραστατικού στο οποίο εμπεριέχεται, για το προϊόν με ονομασία 'ΣΗΜΕΙΟ ΩΜΕΓΑ'.

The screenshot shows a SQL query editor with the following query:

```
1 • SELECT o.document_id, p.name FROM `order` AS o
2 | JOIN product AS p ON o.product_id = p.product_id AND p.name = 'ΣΗΜΕΙΟ ΩΜΕΓΑ';
```

Below the query editor, there is a table with the following data:

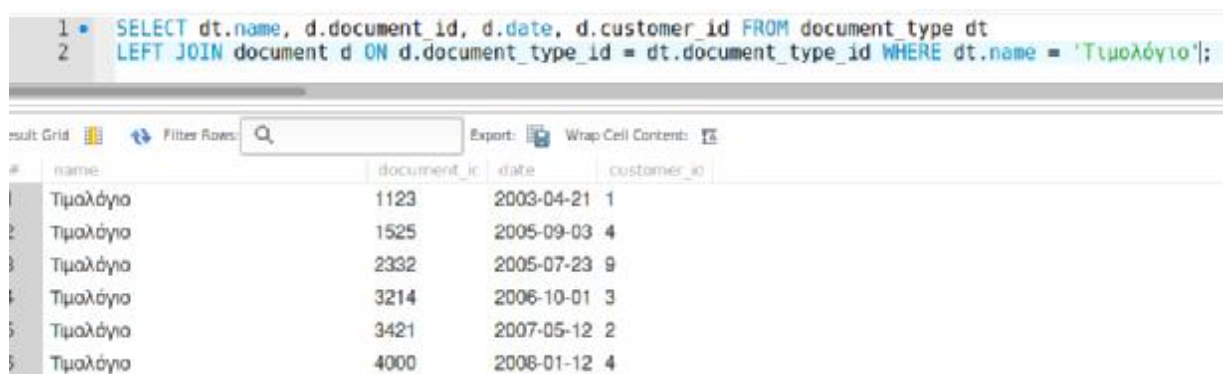
document_id	name
2786	ΣΗΜΕΙΟ ΩΜΕΓΑ
1525	ΣΗΜΕΙΟ ΩΜΕΓΑ

Εικόνα 23: Ερώτημα 21



**ΕΡΩΤΗΜΑ 22:** SELECT ΕΙΔΟΣΠΑΡΑΣΤΑΤΙΚΟΥ.ΟΝΟΜΑΣΕΙΔΟΥΣ, ΠΑΡΑΣΤΑΤΙΚΟ.ΑΡΠΑΡΑΣΤΑΤΙΚΟΥ, ΠΑΡΑΣΤΑΤΙΚΟ.ΗΜΕΡΟΜΕΚΔΟΣΗΣ, ΠΑΡΑΣΤΑΤΙΚΟ.ΚΠΕΛΑΤΗ FROM ΕΙΔΟΣΠΑΡΑΣΤΑΤΙΚΟΥ INNER JOIN ΠΑΡΑΣΤΑΤΙΚΟ ON ΕΙΔΟΣΠΑΡΑΣΤΑΤΙΚΟΥ.ΚΕΙΔΟΥΣ = ΠΑΡΑΣΤΑΤΙΚΟ.ΚΕΙΔΟΥΣΠΑΡ WHERE (((ΕΙΔΟΣΠΑΡΑΣΤΑΤΙΚΟΥ.ΟΝΟΜΑΣΕΙΔΟΥΣ)=[ΠΛΗΚΤΡΟΛΟΓΗΣΤΕ ΟΝΟΜΑ ΕΙΔΟΥΣ]));

Εμφανίζει ως αποτέλεσμα τον παρακάτω πίνακα ο οποίος εμφανίζει το Όνομα Είδους Παραστατικού, τον Αριθμό Παραστατικού, την Ημερομηνία Έκδοσης, και τον Κωδικό Πελάτη, όταν το Όνομα Είδους Παραστατικού καταχωρείται από τον χρήστη της βάσης.



```
1 SELECT dt.name, d.document_id, d.date, d.customer_id FROM document type dt
2 LEFT JOIN document d ON d.document_type_id = dt.document_type_id WHERE dt.name = 'Τιμολόγιο';
```

#	name	document_id	date	customer_id
1	Τιμολόγιο	1123	2003-04-21	1
2	Τιμολόγιο	1525	2005-09-03	4
3	Τιμολόγιο	2332	2005-07-23	9
4	Τιμολόγιο	3214	2006-10-01	3
5	Τιμολόγιο	3421	2007-05-12	2
6	Τιμολόγιο	4000	2006-01-12	4

Εικόνα 24 : Ερώτημα 22

**ΕΡΩΤΗΜΑ 23:** SELECT ΠΑΡΑΣΤΑΤΙΚΟ.ΗΜΕΡΟΜΕΚΔΟΣΗΣ, ΠΕΡΙΛΑΜΒΑΝΕΙ.ΑΡΤΕΜΑΧΙΩΝ, ΠΡΟΙΟΝ.ΤΙΜΠΡΤΕΜΑΧ, ΠΡΟΙΟΝ.ΚΠΡΟΙΟΝΤΟΣ FROM ΠΡΟΙΟΝ INNER JOIN (ΠΑΡΑΣΤΑΤΙΚΟ INNER JOIN ΠΕΡΙΛΑΜΒΑΝΕΙ ON ΠΑΡΑΣΤΑΤΙΚΟ.ΑΡΠΑΡΑΣΤΑΤΙΚΟΥ = ΠΕΡΙΛΑΜΒΑΝΕΙ.ΑΡΠΑΡΑΣΤΑΤΙΚΟΥ) ON ΠΡΟΙΟΝ.ΚΠΡΟΙΟΝΤΟΣ = ΠΕΡΙΛΑΜΒΑΝΕΙ.ΚΠΡΟΙΟΝΤΟΣ WHERE (((ΠΡΟΙΟΝ.ΤΙΜΠΡΤΕΜΑΧ)=[ΠΛΗΚΤΡΟΛΟΓΗΣΤΕ ΤΙΜΗ ΤΕΜΑΧΙΟΥ]) AND ((ΠΡΟΙΟΝ.ΚΠΡΟΙΟΝΤΟΣ)=[ΠΛΗΚΤΡΟΛΟΓΗΣΤΕ ΚΩΔΙΚΟ ΠΡΟΙΟΝΤΟΣ]));

Εμφανίζει ως αποτέλεσμα τον παρακάτω πίνακα ο οποίος εμφανίζει την Ημερομηνία Έκδοσης Παραστατικού, τον Αριθμό Τεμαχίων, την Τιμή Τεμαχίου, και τον Κωδικό Προϊόντος, όταν τόσο η Τιμή Τεμαχίου, όσο και ο Κωδικός Προϊόντος δίνονται από το χρήστη.

```

1 • SELECT d.date, o.num_of_items, p.price, p.product_id FROM document AS d
2 LEFT JOIN `order` AS o ON d.document_id = o.document_id
3 LEFT JOIN product AS p ON o.product_id = p.product_id
4 WHERE p.price = 1.44 AND p.product_id = 501;

```

ult Grid Filter Rows:  Export: Wrap Cell Content:

date	num_of_items	price	product_id
2007-10-10	145	1.44	501

Εικόνα 25 : Ερώτημα 23

**ΕΡΩΤΗΜΑ 24:** SELECT ΠΑΡΑΣΤΑΤΙΚΟ.ΗΜΕΡΟΜΕΚΔΟΣΗΣ, Sum(ΠΕΡΙΛΑΜΒΑΝΕΙ.ΑΡΤΕΜΑΧΙΩΝ)  
AS ΣΥΝΟΛΟΤΟΥ1525 FROM ΠΑΡΑΣΤΑΤΙΚΟ INNER JOIN ΠΕΡΙΛΑΜΒΑΝΕΙ ON  
ΠΑΡΑΣΤΑΤΙΚΟ.ΑΡΠΑΡΑΣΤΑΤΙΚΟΥ = ΠΕΡΙΛΑΜΒΑΝΕΙ.ΑΡΠΑΡΑΣΤΑΤΙΚΟΥ GROUP BY  
ΠΑΡΑΣΤΑΤΙΚΟ.ΗΜΕΡΟΜΕΚΔΟΣΗΣ, ΠΕΡΙΛΑΜΒΑΝΕΙ.ΑΡΠΑΡΑΣΤΑΤΙΚΟΥ HAVING  
(((ΠΕΡΙΛΑΜΒΑΝΕΙ.ΑΡΠΑΡΑΣΤΑΤΙΚΟΥ)="1525"));



Εμφανίζεται ως αποτέλεσμα ο παρακάτω πίνακας ο οποίος εμφανίζει την ημερομηνία έκδοσης παραστατικού, και τον συνολικό αριθμό τεμαχίων, για το παραστατικό με αριθμό παραστατικού «1525»

```

1 • SELECT o.document_id, d.date, sum(o.num_of_items)
2 FROM document AS d
3 JOIN `order` AS o ON d.document_id = o.document_id
4 WHERE o.document_id = 1525 |
5 GROUP BY d.date, o.document_id;

```

---

Result Grid  Filter Rows:  Export:  Wrap Cell Content:

#	document_id	date	sum(o.num_of_items)
1	1525	2005-09-03	408

Εικόνα 26: Ερώτημα 24

**ΕΡΩΤΗΜΑ 25:** SELECT ΔΙΑΦΗΜΙΣΗ.ΕΠΩΝΥΜΙΑ, ΠΡΟΙΟΝ.ΟΝΟΜΠΡΟΙΟΝΤΟΣ, (2011-Year([ΧΡΟΝΕΝΑΡΔΙΑΦ]))\*500 AS ΚΟΣΤΟΣ FROM ΠΡΟΙΟΝ INNER JOIN ΔΙΑΦΗΜΙΣΗ ON ΠΡΟΙΟΝ.ΚΠΡΟΙΟΝΤΟΣ=ΔΙΑΦΗΜΙΣΗ.ΚΠΡΟΙΟΝΤΟΣ WHERE (((ΔΙΑΦΗΜΙΣΗ.ΕΠΩΝΥΜΙΑ)="ADMINE") AND ((ΠΡΟΙΟΝ.ΟΝΟΜΠΡΟΙΟΝΤΟΣ)="ΜΙΧΑΕΛ ΚΟΛΧΑΑΣ"));

Εμφανίζει το κόστος διαφήμισης, στην διαφημιστική με επωνυμία «ADMINE», για το προϊόν με ονομασία «ΜΙΧΑΕΛ ΚΟΛΧΑΑΣ», αν υποθέσουμε ότι η διαφήμιση θα συνεχίζεται για όλο το 2010, και η ετήσια χρέωση είναι 500 Ευρώ.

The screenshot shows a SQL query editor with the following text:

```

1 • SELECT advertisement.name, product.name, (2011-year(advertisement.startDate))*500 AS cost
2 FROM product INNER JOIN advertisement ON advertisement.product_id = advertisement.product_id
3 WHERE (((advertisement.name)="ADMINE") AND ((product.name)="ΜΙΧΑΕΛ ΚΟΛΧΑΑΣ"));

```

Below the query editor, there is a table with the following data:

name	name	cost
ADMINE	ΜΙΧΑΕΛ ΚΟΛΧΑΑΣ	2500

Εικόνα 27: Ερώτημα 25

**ΕΡΩΤΗΜΑ 26:** SELECT ΠΑΡΑΣΤΑΤΙΚΟ.ΗΜΕΡΟΜΕΚΔΟΣΗΣ, ΠΕΛΑΤΗΣ.ΕΠΩΝΥΜΙΑ, ΥΠΑΛΛΗΛΟΣ.ΟΝΟΜΑ, ΥΠΑΛΛΗΛΟΣ.ΕΠΩΝΥΜΟ FROM ΥΠΑΛΛΗΛΟΣ INNER JOIN (ΠΕΛΑΤΗΣ INNER JOIN ΠΑΡΑΣΤΑΤΙΚΟ ON ΠΕΛΑΤΗΣ.ΚΠΕΛΑΤΗ = ΠΑΡΑΣΤΑΤΙΚΟ.ΚΠΕΛΑΤΗ) ON ΥΠΑΛΛΗΛΟΣ.ΑΦΜ = ΠΕΛΑΤΗΣ.ΑΦΜΥΠΑΛ WHERE (((ΥΠΑΛΛΗΛΟΣ.ΑΦΜ)=[ΠΛΗΚΤΡΟΛΟΓΗΣΤΕ ΑΦΜ]));

Εμφανίζεται ως αποτέλεσμα ο παρακάτω πίνακας ο οποίος περιέχει την ημερομηνία έκδοσης παραστατικού, την επωνυμία πελάτη στον οποίο εκδόθηκε, το όνομα, το επώνυμο του υπαλλήλου που έχει αναλάβει τον πελάτη, όταν το ΑΦΜ του τελευταίου δίνεται παραμετρικά.

```

1 • SELECT document.date, customer.name, Employee.name, Employee.surname FROM Employee
2 INNER JOIN ( customer INNER JOIN document ON customer.customer_id = document.customer_id)
3 ON Employee.afm = customer.afmEmployee
4 WHERE (Employee.afm)=099765422;

```

#	date	name	name	surname
1	2005-09-03	ΕΠΙΠΕΔΟ ΒΙ...	Λευτέρης	Κοντολέων
2	2005-09-03	ΕΠΙΠΕΔΟ ΒΙ...	Λευτέρης	Παπανίκος
3	2005-09-03	N. Οικονόμου	Λευτέρης	Κοντολέων
4	2005-09-03	N. Οικονόμου	Λευτέρης	Παπανίκος
5	2005-09-03	N. Οικονόμου	Λευτέρης	Κοντολέων
6	2005-09-03	N. Οικονόμου	Λευτέρης	Παπανίκος
7	2006-03-02	ΕΠΙΠΕΔΟ ΒΙ...	Λευτέρης	Κοντολέων
8	2006-03-02	ΕΠΙΠΕΔΟ ΒΙ...	Λευτέρης	Παπανίκος
9	2006-03-02	N. Οικονόμου	Λευτέρης	Κοντολέων
10	2006-03-02	N. Οικονόμου	Λευτέρης	Παπανίκος

Εικόνα 28 : Ερώτημα 26

**ΕΡΩΤΗΜΑ 27:** SELECT ΠΑΡΑΣΤΑΤΙΚΟ.ΑΡΠΑΡΑΣΤΑΤΙΚΟΥ, ΠΑΡΑΣΤΑΤΙΚΟ.ΗΜΕΡΟΜΕΚΔΟΣΗΣ, ΠΕΡΙΛΑΜΒΑΝΕΙ.ΑΡΤΕΜΑΧΙΩΝ, 1.17\*[ΑΡΤΕΜΑΧΙΩΝ] AS ΒΑΡΟΣ FROM ΠΑΡΑΣΤΑΤΙΚΟ INNER JOIN ΠΕΡΙΛΑΜΒΑΝΕΙ ON ΠΑΡΑΣΤΑΤΙΚΟ.ΑΡΠΑΡΑΣΤΑΤΙΚΟΥ = ΠΕΡΙΛΑΜΒΑΝΕΙ.ΑΡΠΑΡΑΣΤΑΤΙΚΟΥ WHERE (((ΠΑΡΑΣΤΑΤΙΚΟ.ΑΡΠΑΡΑΣΤΑΤΙΚΟΥ)=[ΠΛΗΚΤΡΟΛΟΓΗΣΤΕ ΑΡΠΑΡΑΣΤΑΤΙΚΟΥ]));

Ο παρακάτω πίνακας εμφανίζει τον αριθμό παραστατικού, την ημερομηνία έκδοσής του, τον αριθμό τεμαχίων που περιέχει ανά προϊόν και το βάρος των επιμέρους τεμαχίων, αν υποθέσουμε ότι το κάθε τεμάχιο έχει βάρος 1,17 γραμμάρια ενώ ζητείται ο αριθμός παραστατικού να δίνεται παραμετρικά.

```
1 • SELECT d.document_id, d.date, o.num_of_items, 1.17*o.num_of_items AS weight
2 FROM document AS d
3 INNER JOIN `order` AS o ON d.document_id = o.document_id
4 WHERE (d.document_id)=1525;|
```

ult Grid Filter Rows:  Export: Wrap Cell Content:

document_id	date	num_of_items	weight
1525	2005-09-03	300	351.00
1525	2005-09-03	40	46.80
1525	2005-09-03	68	79.56

Εικόνα 29: Ερώτημα 27

**ΕΡΩΤΗΜΑ 28:** SELECT ΠΕΛΑΤΗΣ.ΕΠΩΝΥΜΙΑ, ΥΠΑΛΛΗΛΟΣ.ΟΝΟΜΑ, ΕΞΕΙΔΙΚΕΥΣΗ.ΕΞΕΙΔΙΚΕΥΣΗ FROM (ΥΠΑΛΛΗΛΟΣ INNER JOIN ΕΞΕΙΔΙΚΕΥΣΗ ON ΥΠΑΛΛΗΛΟΣ.ΑΦΜ = ΕΞΕΙΔΙΚΕΥΣΗ.ΑΦΜ) INNER JOIN ΠΕΛΑΤΗΣ ON ΥΠΑΛΛΗΛΟΣ.ΑΦΜ = ΠΕΛΑΤΗΣ.ΑΦΜΥΠΑΛ WHERE (((ΕΞΕΙΔΙΚΕΥΣΗ.ΕΞΕΙΔΙΚΕΥΣΗ)="ΠΩΛΗΤΗΣ");

Εμφανίζεται ως αποτέλεσμα ο παρακάτω πίνακας ο οποίος περιέχει την επωνυμία του πελάτη, το όνομα και την εξειδίκευση του υπαλλήλου που τον εξυπηρετεί, για τους υπαλλήλους που είναι πωλητές.

```

1 • SELECT c.name, e.name, s.specialisation
2 FROM (Employee AS e INNER JOIN specialisation AS s ON e.afm = s.afm)
3 INNER JOIN customer AS c
4 ON e.afm = c.afmEmployee
5 WHERE (s.specialisation)="ΠΩΛΗΤΗΣ";

```

name	name	specialisation
ΕΠΙΠΕΔΟ ΒΙ...	Λευτέρης	Πωλητής
ΕΠΙΠΕΔΟ ΒΙ...	Λευτέρης	Πωλητής
ΠΑΛΕΤΑ	Γάννης	Πωλητής
ΠΑΛΕΤΑ	Γάννης	Πωλητής
Ν. Οικονόμου	Λευτέρης	Πωλητής
Ν. Οικονόμου	Λευτέρης	Πωλητής
ΠΡΩΤΕΥΣ	Γάννης	Πωλητής
ΠΡΩΤΕΥΣ	Γάννης	Πωλητής
Ν. Οικονόμου	Λευτέρης	Πωλητής

Εικόνα 30 : Ερώτημα 28

**ΕΡΩΤΗΜΑ 29:** SELECT ΠΕΛΑΤΗΣ.ΕΠΩΝΥΜΙΑ, ΠΕΛΑΤΗΣ.ΑΡΜΟΔΔΟΥ, ΠΑΡΑΣΤΑΤΙΚΟ.ΗΜΕΡΟΜΕΚΔΟΣΗΣ, ΕΙΔΟΣΠΑΡΑΣΤΑΤΙΚΟΥ.ΟΝΟΜΑΣΕΙΔΟΥΣ FROM ΠΕΛΑΤΗΣ INNER JOIN (ΕΙΔΟΣΠΑΡΑΣΤΑΤΙΚΟΥ INNER JOIN ΠΑΡΑΣΤΑΤΙΚΟ ΟΝ ΕΙΔΟΣΠΑΡΑΣΤΑΤΙΚΟΥ.ΚΕΙΔΟΥΣ = ΠΑΡΑΣΤΑΤΙΚΟ.ΚΕΙΔΟΥΣΠΑΡ) ΟΝ ΠΕΛΑΤΗΣ.ΚΠΕΛΑΤΗ = ΠΑΡΑΣΤΑΤΙΚΟ.ΚΠΕΛΑΤΗ WHERE (((ΠΕΛΑΤΗΣ.ΑΡΜΟΔΔΟΥ)="Α ΠΑΤΡΩΝ") AND ((ΕΙΔΟΣΠΑΡΑΣΤΑΤΙΚΟΥ.ΟΝΟΜΑΣΕΙΔΟΥΣ)="ΤΙΜΟΛΟΓΙΟ") AND ((Year([ΗΜΕΡΟΜΕΚΔΟΣΗΣ]))<2008));

Εμφανίζει την επωνυμία και τη ΔΟΥ του πελάτη, την ημερομηνία έκδοσης και το όνομα είδους του παραστατικού, όταν η ΔΟΥ είναι η Α΄ Πατρών, το είδος του παραστατικού είναι τιμολόγιο, για τα Παραστατικά που εκδόθηκαν πριν το 2008.

The screenshot shows a SQL query editor with the following code:

```

1 • SELECT c.name, c.doy, d.date, dt.name
2 FROM customer AS c
3 INNER JOIN
4 (document_type AS dt INNER JOIN document AS d ON dt.document_type_id = d.document_type_id)
5 ON c.customer_id = d.customer_id
6 WHERE |(c.doy) = 'Α ΠΑΤΡΩΝ' AND (dt.name) = 'ΤΙΜΟΛΟΓΙΟ' AND YEAR(date) < 2008;

```

Below the query editor is a "Result Grid" with the following data:

#	name	doy	date	name
1	ΕΠΙΠΕΔΟ ΒΙ...	Α Πατρών	2005-09-03	Τιμολόγιο
2	N. Οικονόμου	Α Πατρών	2005-09-03	Τιμολόγιο
3	N. Οικονόμου	Α Πατρών	2005-09-03	Τιμολόγιο

Εικόνα 31 : Ερώτημα 29



## Συμπεράσματα

Στην εργασία αυτή έγινε μια ανάλυση της σημασίας των Βάσεων Δεδομένων, και πιο συγκεκριμένα αυτών που χρησιμοποιούνται από τις επιχειρήσεις. Το γενικό συμπέρασμα που προκύπτει είναι ότι οι Βάσεις Δεδομένων ιδιαίτερα στην σημερινή εποχή όπου ο όγκος των πληροφοριών είναι ιδιαίτερα μεγάλος και δύσκολα γίνεται διαχειρίσιμος μια επιχείρηση που επιθυμεί να παραμείνει ανταγωνιστική πρέπει να είναι σε θέση να αξιοποιεί αποτελεσματικά τα δεδομένα τα οποία συνεχώς λαμβάνει. Αυτό γίνεται με την χρήση συγκεκριμένων εργαλείων με ένα από αυτά να είναι οι Βάσεις Δεδομένων.

Η χρήση μιας Βάσης Δεδομένων επιτρέπει την σωστή και αποτελεσματική συλλογή δεδομένων με τρόπο οργανωμένο και αξιόπιστο. Εν συνεχεία τα δεδομένα μπορούν να αποτελέσουν αντικείμενο επεξεργασίας από τον χρήστη της Βάσης ή κάποιο άλλο πρόσωπο με σκοπό να εξαγάγει χρήσιμες πληροφορίες. Η ταχύτητα διεκπεραίωσης των εργασιών, η αξιοπιστία και η ευελιξία που παρέχει στο χρήστη μια Βάση Δεδομένων εγγυάται ότι θα αποτελέσει εργαλείο αύξησης της ανταγωνιστικότητας και βελτίωσης της λειτουργίας της επιχείρησης. Επίσης, η ταυτόχρονη προσπέλαση των δεδομένων της αποτελεί άλλο ένα στοιχείο που αυξάνει την αποδοτικότητα των χρηστών μιας Βάσης Δεδομένων.

Δεν θα ήταν υπερβολή να πει κανείς ότι πλέον με τον τόσο έντονο ανταγωνισμό που επικρατεί μεταξύ των επιχειρήσεων είναι αδύνατο μια επιχείρηση να έχει αφήσει τα δεδομένα της χωρίς αξιοποίηση με σκοπό την εξαγωγή πληροφοριών. Προκειμένου αυτά τα χαρακτηριστικά και η αξία των Βάσεων Δεδομένων να γίνουν καλύτερα αντιληπτά σχεδιάστηκε στα πλαίσια της εργασίας αυτής μια Βάση Δεδομένων επιχειρησιακού ενδιαφέροντος με την χρήση MySQL και στην συνέχεια εκτελέστηκαν αρκετά ερωτήματα με την χρήση του ερωτήματος SELECT.

Από τα ερωτήματα αυτά και τα αποτελέσματα που φέρνουν στο χρήστη προκύπτει ότι οι πληροφορίες που μπορεί να έχει ο χρήστης στα χέρια του είναι πολλές και ιδιαίτερα ενδιαφέρουσες. Με δεδομένο ότι ο χρήστης μπορεί να ζητήσει σχεδόν οποιοδήποτε στοιχείο θελήσει από την Βάση και μάλιστα ακριβώς με τον τρόπο που επιθυμεί (π.χ. ταξινόμηση) το βάρος για την ορθή αξιοποίηση της πέφτει σε αυτόν. Αυτό σημαίνει ότι ο χρήστης της Βάσης πρέπει να είναι σε θέση να συντάξει τα κατάλληλα ερωτήματα προκειμένου να πάρει τις πληροφορίες που επιθυμεί.

Συνολικά μπορούμε να πούμε ότι μια Βάση Δεδομένων αποτελεί πλέον αναπόσπαστο τμήμα μιας σύγχρονης επιχείρησης καθώς είναι ο μόνος τρόπος για να διατηρήσει η επιχείρηση την ποιότητα των δεδομένων της σε ένα υψηλό επίπεδο αλλά και να εξάγει από αυτήν χρήσιμες πληροφορίες.

## Βιβλιογραφία

- 1) Welling L. Thompson L., (2009), «Ανάπτυξη εφαρμογών με PHP και MySQL», Εκδόσεις Μ. Γκιούρδας - Γ' έκδοση
- 2) Ξένος Μ. Χριστοδουλάκης Δ. (2002), «Εισαγωγή στις Βάσεις Δεδομένων», Εκδόσεις Παπασωτηρίου
- 3) Παπαθανασίου Α. Ελευθέριος (2003), «Στοιχεία Υπολογιστικών Συστημάτων», Εκδόσεις Μπένου
- 4) Στραγαλινός Ε. (2010), «Επέκταση του OLAP μοντέλου με σημαντικά δίκτυα» Πανεπιστήμιο Πατρών

## Ηλεκτρονικές πηγές

- 1) <https://www.mysql.com/>
- 2) [www.wikipedia.com](http://www.wikipedia.com)