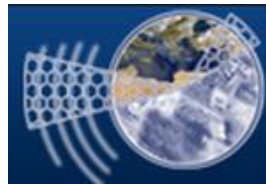




Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

**Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Μηχανικών Πληροφορικής**



Πτυχιακή Εργασία

**Κρατήσεις δωματίων σε ξενοδοχεία και ενοικιάσεις
αυτοκινήτων**

**Κοντάκης Γεώργιος Α.Μ.: 2574
Συντυχάκης Ηλίας Α.Μ.: 3005**

Επιβλέπων Καθηγητής: Παπαδάκης Νικόλαος

**Ηράκλειο
2015**

Abstract

Nowadays, where technological development increasing rapidly, people's desire appears for improving their lifestyle using technological means to facilitate their operations. The internet serves the needs of users and helps in achieving their goal.

This paper describes the technologies and tools needed to create a comprehensive system for booking rooms at hotels and car rentals.

Σύνοψη

Στις μέρες μας όπου η τεχνολογική ανάπτυξη αυξάνεται ραγδαία, εμφανίζεται η επιθυμία των ανθρώπων για βελτίωση του τρόπου ζωής τους χρησιμοποιώντας τεχνολογικά μέσα με σκοπό την διευκόλυνση των ενεργειών τους. Το διαδίκτυο εξυπηρετεί τις ανάγκες των χρηστών και βοηθάει στην επίτευξη των στόχο τους.

Στην εργασία αυτή περιγράφονται οι τεχνολογίες και τα μέσα που χρειάζονται για την δημιουργία ενός πλήρους συστήματος κρατήσεων δωματίων σε ξενοδοχεία και ενοικιάσεων αυτοκινήτων.

Πίνακας Περιεχομένων

1 Εισαγωγή	- 5-
1.1 Περίληψη	- 5-
1.2 Κίνητρο για την διεξαγωγή της εργασίας.....	- 5-
1.3 Σκοπός και στόχοι της εργασίας.....	- 5-
1.4 Δομή της εργασίας	- 5-
2 Συστήματα διαχείρισης βάσεων δεδομένων και χρήσιμες τεχνολογίες.....	- 6-
2.1 Εισαγωγή στις βάσεις δεδομένων και στα συστήματα διαχείρισης.....	- 6-
2.2 Εισαγωγή HTML.....	- 71-
2.3 Εισαγωγή PHP.....	- 73-
2.4 Εισαγωγή CSS.....	- 74-
3 Μηχανές αναζήτησης και ιστοσελίδες κρατήσεων.....	- 79-
3.1 Trivago.....	- 79-
3.2 Hotelz	- 80-
3.3 Mega Travel	- 81-
3.4 Viva	- 82-
3.5 Booking	- 83-
3.6 Hotels	- 85-
3.7 Hertz	- 86-
3.8 Sixt	- 87-
3.9 Avis	- 88-
4 Ανάλυση και παρουσίαση της εργασίας.....	- 90-
5 Συμπεράσματα – Αποτελέσματα.....	- 96-
Βιβλιογραφία	- 97-
ΠΑΡΑΡΤΗΜΑ Α	- 98-

1 Εισαγωγή

1.1 Περίληψη

Στην παρούσα εργασία θα παρουσιάσουμε και θα αναλύσουμε τους τρόπους και τις μεθόδους που χρειάζονται για την δημιουργία ενός συστήματος διαχείρισης κρατήσεων δωματίων σε ξενοδοχεία και ενοικιάσεων αυτοκινήτων. Αρχικά, θα ασχοληθούμε με τις τεχνολογίες και τα μέσα που είναι αναγκαία για την επιτυχή ολοκλήρωση ενός τέτοιου συστήματος. Στην συνέχεια, θα καταγράψουμε τις απαιτήσεις του συστήματος. Τέλος, θα συγκρίνουμε τα αποτελέσματα με τα ζητούμενα και θα εμφανίσουμε τα τελικά συμπεράσματα για την σωστή λειτουργία του συστήματος.

1.2 Κίνητρο για την διεξαγωγή της εργασίας

Τα τελευταία χρόνια, η ανάπτυξη της τεχνολογίας εμφάνισε στην κοινωνία μας το ηλεκτρονικό εμπόριο με το οποίο οι αγορές γίνονται πλέον μέσω διαδικτύου προσφέροντας στους χρήστες εξοικονόμηση χρόνου και ενέργειας. Όπως αντιλαμβανόμαστε ένα τέτοιο εμπόριο απλουστεύει τις διαδικασίες αγοράς προϊόντων και υπηρεσιών.

Σύμφωνα με τα παραπάνω, είναι πολύ σημαντικό να ερευνήσουμε, να αναλύσουμε και να υλοποιήσουμε ένα σύστημα με αποτέλεσμα την επιτυχή δημιουργία ενός ηλεκτρονικού καταστήματος (e-shop) με αποτέλεσμα την ικανοποίηση των αναγκών των χρηστών.

1.3 Σκοπός και στόχοι της εργασίας

Ο κύριος σκοπός της εργασίας είναι η σωστή δημιουργία και κατασκευή ενός ηλεκτρονικού καταστήματος (e-shop) με την χρήση προγραμμάτων, τεχνολογιών και μεθόδων (HTML, PHP, CSS, MYSQL) έχοντας στόχο την αποτελεσματική παρουσίαση του τρόπου λειτουργίας ενός συστήματος διαχείρισης.

1.4 Δομή της εργασίας

- Στο πρώτο κεφάλαιο παρουσιάζουμε την περίληψη, το κίνητρο, τον σκοπό και τους στόχους της εργασίας.
- Στο δεύτερο κεφάλαιο αναφέρουμε και αναλύουμε τις μεθόδους, τις τεχνολογίες και τα προγράμματα που χρησιμοποιήθηκαν για την ολοκλήρωση της εργασίας.
- Στο τρίτο κεφάλαιο παρουσιάζουμε γνωστές ιστοσελίδες κρατήσεων και μηχανές αναζήτησης και τον τρόπο λειτουργίας τους.
- Στο τέταρτο κεφάλαιο αναλύουμε τα στάδια για την δημιουργία και την ολοκλήρωση της εργασίας.
- Στο πέμπτο κεφάλαιο συγκρίνουμε τα αποτελέσματα της εργασίας με τα ζητούμενα του προβλήματος.

2 Συστήματα διαχείρισης βάσεων δεδομένων και τεχνολογίες

2.1 Εισαγωγή στις βάσεις δεδομένων και στα συστήματα διαχείρισης

Εισαγωγή στις Βάσεις Δεδομένων

Η αλματώδης ανάπτυξη της επιστήμης της πληροφορικής και των επικοινωνιών τα τελευταία χρόνια έχει καταστήσει την πληροφορία ως ένα από τα βασικότερα και πολυτιμότερα αγαθά. Είναι κοινός τόπος σήμερα η εκτίμηση ότι το αγαθό της πληροφορίας είναι επιθυμητό απ' όλους τους εργαζόμενους αλλά και τους εκπαιδευόμενους, ώστε να είναι πιο αποδοτικοί, ανταγωνιστικοί αλλά και παραγωγικοί στην εργασία τους.

Τα συστήματα βάσεων δεδομένων τα χρησιμοποιούμε για να μπορούμε να αποθηκεύσουμε, να επεξεργαστούμε αλλά και να εκμεταλλευτούμε αποδοτικά αυτόν τον τεράστιο όγκο των πληροφοριών που αυξάνονται με αλματώδεις ρυθμούς καθημερινά.

Τα Δεδομένα και οι Πληροφορίες

Με τον όρο πληροφορία αναφερόμαστε συνήθως σε ειδήσεις, γεγονότα και έννοιες που αποκτάμε από την καθημερινή μας επικοινωνία και τα θεωρούμε ως αποκτηθείσα γνώση, ενώ τα δεδομένα μπορούν να είναι μη κατάλληλα επεξεργασμένα και μη ταξινομημένα σύνολα πληροφοριών. Ένας αυστηρός ορισμός για το τι είναι δεδομένα και τι είναι πληροφορία, σύμφωνα με την επιτροπή ANSI των ΗΠΑ, είναι ο εξής :

- Δεδομένα (data) είναι μια παράσταση, όπως γράμματα, αριθμοί, σύμβολα κ.ά. στα οποία μπορούμε να δώσουμε κάποια σημασία (έννοια).
- Πληροφορία (information) είναι η σημασία που δίνουμε σ' ένα σύνολο από δεδομένα, τα οποία μπορούμε να επεξεργαστούμε βάσει προκαθορισμένων κανόνων και να βγάλουμε έτσι κάποια χρήσιμα συμπεράσματα. Με τις πληροφορίες περιορίζεται η αβεβαιότητα που έχουμε για διάφορα πράγματα και βοηθιόμαστε έτσι στο να λάβουμε σωστές αποφάσεις.

Τα δεδομένα μπορούν να θεωρηθούν ως τρόποι αναπαράστασης εννοιών και γεγονότων που μπορούν να υποστούν διαχείριση και επεξεργασία. Η συλλογή και αποθήκευση ενός τεράστιου όγκου δεδομένων όπως απαιτούν οι κοινωνικές συνθήκες σήμερα, δεν λύνει τελείως το πρόβλημα της σωστής οργάνωσης και ταξινόμησης των δεδομένων. Τα δεδομένα θα πρέπει να οργανωθούν με τέτοιο τρόπο έτσι ώστε να μπορούμε να τα εντοπίζουμε και να τα αξιοποιούμε εύκολα και γρήγορα και τη στιγμή που τα χρειαζόμαστε.

Ένα κλασικό παράδειγμα μη σωστής οργάνωσης δεδομένων θα ήταν για παράδειγμα ο τηλεφωνικός κατάλογος της πόλης Σερρών, όπου οι συνδρομητές δεν θα ήταν καταχωρημένοι αλφαβητικά σύμφωνα με το επώνυμο και το όνομά τους, αλλά εντελώς τυχαία. Ένας τέτοιος τηλεφωνικός κατάλογος θα περιείχε μια τεράστια ποσότητα δεδομένων αλλά θα ήταν ουσιαστικά άχρηστος.

Εκτός λοιπόν από τη μόνιμη αποθήκευση των δεδομένων, χρειαζόμαστε και κάποιους τρόπους ευέλικτης και αποδοτικής οργάνωσής τους.

Προβλήματα στην Οργάνωση Αρχείων

Στα αρχικά στάδια της οργάνωσης αρχείων, μια συνηθισμένη πρακτική ήταν η δημιουργία ξεχωριστών εφαρμογών (προγραμμάτων) και ξεχωριστών αρχείων, όπως για παράδειγμα η δημιουργία ενός αρχείου πελατών και ενός άλλου ανεξάρτητου αρχείου για τις παραγγελίες των πελατών. Τα προβλήματα που προέκυψαν από την πρακτική αυτή είναι τα εξής :

- Πλεονασμός των δεδομένων (data redundancy). Υπάρχει η περίπτωση να έχουμε επανάληψη των ίδιων δεδομένων σε αρχεία διαφορετικών εφαρμογών. Για παράδειγμα, αν έχουμε ένα αρχείο

πελατών και ένα αρχείο παραγγελιών αυτών των πελατών, είναι σχεδόν σίγουρο ότι θα υπάρχουν κάποια στοιχεία των πελατών που θα υπάρχουν και στα δύο αρχεία.

- Ασυνέπεια των δεδομένων (data inconsistency). Αυτό μπορεί να συμβεί όταν υπάρχουν τα ίδια στοιχεία των πελατών (πλεονασμός) και στο αρχείο πελατών και στο αρχείο παραγγελιών και χρειασθεί να γίνει κάποια αλλαγή στη διεύθυνση ή στα τηλέφωνα κάποιου πελάτη, οπότε είναι πολύ πιθανό να γίνει η διόρθωση μόνο στο ένα αρχείο και όχι και στο άλλο.
- 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. Έχει να κάνει με την ανομοιομορφία και με την διαφορετική αναπαράσταση και οργάνωση των δεδομένων στα αρχεία των εφαρμογών. Η αδυναμία αυτή δημιουργεί προβλήματα προσαρμογής των χρηστών καθώς και προβλήματα στην ανταλλαγή δεδομένων μεταξύ διαφορετικών συστημάτων.

Βάσεις Δεδομένων

Συστήματα βάσης δεδομένων είναι ένα ηλεκτρονικό σύστημα τήρησης εγγραφών, δηλαδή, είναι ένα σύστημα για υπολογιστές, που ο γενικός σκοπός του είναι να τηρεί πληροφορίες και να δίνει αυτές τις πληροφορίες όταν του ζητούνται. Οι πληροφορίες που τηρούνται σε ένα τέτοιο σύστημα μπορεί να είναι οτιδήποτε έχει σημασία για το άτομο ή τον οργανισμό που εξυπηρετεί το συγκεκριμένο σύστημα, με άλλα λόγια οτιδήποτε χρειάζεται για την υποβοήθηση των εργασιών αυτού του ατόμου ή οργανισμού.

Η ίδια η βάση δεδομένων μπορεί να θεωρηθεί ένα είδος ηλεκτρονικής αρχειοθήκης, ένα χώρος για την αποθήκευση μιας συλλογής ηλεκτρονικών αρχείων δεδομένων.

Ο χρήστης του συστήματος έχει στη διάθεση του ορισμένα βοηθήματα για να εκτελεί στα αρχεία Β.Δ. διάφορες εργασίες, στις οποίες συγκαταλέγονται ,ανάμεσα σε άλλες, και οι εξής:

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

Ένα σύστημα βάσης δεδομένων απαρτίζεται από τέσσερα βασικά στοιχεία: τα δεδομένα, το υλικό, το λογισμικό και τους χρήστες.

Τα πλεονεκτήματα ενός συστήματος βάσης δεδομένων, σε σύγκριση με τις παραδοσιακές μεθόδους παρακολούθησης (χαρτί και μολύβι), αρχικά για μια μικρή βάση δεδομένων είναι πολλά

- Οικονομία χώρου (Καταργούνται τα ογκώδη παραδοσιακά αρχεία με φακέλους και έγγραφα).
- Ταχύτητα (το σύστημα μπορεί να ανακαλεί και να αλλάζει τα δεδομένα γρηγορότερα από τον άνθρωπο)
- Λιγότερος κόπος
- Άμεση πληροφόρηση (Ακριβείς και ενημερωμένες πληροφορίες είναι διαθέσιμες κάθε στιγμή).

Επιπλέον, σε ένα περιβάλλον πολλών χρηστών, το σύστημα βάσης δεδομένων παρέχει στην επιχείρηση κεντρικό έλεγχο των δεδομένων της ώστε να προκύπτουν τα εξής πλεονεκτήματα:

- Ο πλεονασμός μειώνεται στο ελάχιστο . Στα συμβατικά συστήματα (εκείνα που δεν είναι συστήματα βάσεων δεδομένων), η κάθε εφαρμογή έχει τα δικά της αρχεία. Αυτό το γεγονός οδηγεί πολύ συχνά σε υψηλό βαθμό πλεονασμού (επανάληψης) για τα αποθηκευμένα δεδομένα, με αποτέλεσμα τη σπατάλη αποθηκευτικού χώρου. Θα πρέπει εδώ να ξεκαθαρίσουμε ότι αυτό δε σημαίνει πως είναι πάντα δυνατό να εξαλειφθούν όλοι οι πλεονασμοί, ούτε πως είναι πάντα επιθυμητό. Μερικές φορές υπάρχουν σοβαροί επιχειρηματικοί ή τεχνικοί λόγοι που επιβάλλουν να τηρούνται ξεχωριστά αντίγραφα των ίδιων αποθηκευμένων δεδομένων.
- Η ασυνέπεια μπορεί να αποφευχθεί (ως ένα βαθμό). Το DBMS θα μπορεί να εγγυηθεί ότι η βάση δεδομένων δε θα είναι ποτέ ασυνεπής στα μάτια του χρηστή ,εξασφαλίζοντας ότι κάθε αλλαγή που θα γίνεται σε οποιαδήποτε από δυο όμοιες καταχωρίσεις θα γίνεται αυτόματα και στην άλλη. Αυτή η διαδικασία είναι γνωστή με το όνομα διάδοση ενημερώσεων.
- Τα δεδομένα μπορούν να είναι κοινόχρηστα. Ο μερισμός δε σημαίνει μόνο ότι οι υπάρχουσες εφαρμογές μπορούν να μοιράζονται τα δεδομένα της βάσης δεδομένων αλλά και ότι είναι δυνατή η ανάπτυξη νέων εφαρμογών που θα μπορούν να χρησιμοποιούν τα ίδια αποθηκευμένα δεδομένα.
- Μπορούν να επιβάλλονται πρότυπα. Η τυποποίηση της αναπαράστασης των δεδομένων διευκολύνει ιδιαίτερα την ανταλλαγή δεδομένων. Τα πρότυπα ονομασίας και τεκμηρίωσης των δεδομένων είναι επίσης πολύ επιθυμητά για να διευκολύνεται ο μερισμός και η καλύτερη κατανόηση των δεδομένων.
- Μπορούν να εφαρμόζονται περιορισμοί ασφαλείας .Έχοντας πλήρη δικαιοδοσία πάνω στη βάση δεδομένων, ο DBA-Database Administrator (α) μπορεί να εξασφαλίσει ότι η πρόσβαση στη βάση δεδομένων θα μπορεί να γίνεται μόνο μέσω των κατάλληλων καναλιών και, κατά συνέπεια, (β) μπορεί να ορίσει κανόνες ασφαλείας με βάση τους οποίους θα γίνεται έλεγχος κάθε φορά που θα υπάρχει απόπειρα προσπέλασης εμπιστευτικών δεδομένων. Βέβαια ένα σύστημα βάσης δεδομένων απαιτεί την ύπαρξη ενός καλού συστήματος ασφαλείας.
- Μπορεί να διατηρείται η ακεραιότητα .Το πρόβλημα της ακεραιότητας είναι να εξασφαλίζεται ότι τα δεδομένα της βάσης δεδομένων είναι ακριβή. Η ασυμφωνία μεταξύ δυο καταχωρίσεων που υποτίθεται ότι αντιπροσωπεύουν το ίδιο “γεγονός” είναι ένα παράδειγμα έλλειψης ακεραιότητας φυσικά, αυτό το συγκεκριμένο πρόβλημα μπορεί να παρουσιαστεί μόνο αν υπάρχει πλεονασμός στα αποθηκευμένα δεδομένα. Ακόμη και αν δεν υπάρχει πλεονασμός όμως, πάλι υπάρχει περίπτωση να περιέχει η βάση δεδομένων λανθασμένες πληροφορίες. Αξίζει να επισημάνουμε ότι η ακεραιότητα των δεδομένων έχει πολύ μεγαλύτερη σημασία σε ένα σύστημα βάσης δεδομένων πολλών χρηστών από ότι σε ένα περιβάλλον “ιδιωτικών αρχείων “, ακριβώς επειδή η βάση δεδομένων είναι μεριζομενη. Αυτό συμβαίνει γιατί, χωρίς τους κατάλληλους ελέγχους, μπορεί ένας χρήστης να ενημερώσει τη βάση δεδομένων με εσφαλμένο τρόπο, δημιουργώντας με αυτό τον τρόπο λανθασμένα δεδομένα και “ μολύνοντας “ τους υπόλοιπους “
- Οι αντικρουόμενες απαιτήσεις μπορούν να εξισορροπούνται. Γνωρίζοντας τις συνολικές απαιτήσεις της επιχείρησης, σε αντιδιαστολή με τις απαιτήσεις των μεμονωμένων χρηστών, ο DBA (πάντα με τις οδηγίες του υπεύθυνου διαχείρισης δεδομένων) μπορεί να δομήσει το σύστημα με τέτοιο τρόπο ώστε να παρέχει γενικές υπηρεσίες που να είναι “ βέλτιστες για την επιχείρηση”. Για παράδειγμα μπορεί να επιλεγεί μια αναπαράσταση των αποθηκευμένων δεδομένων που να παρέχει γρήγορη πρόσβαση στις σημαντικές εφαρμογές ίσως σε βάρος της απόστασης άλλων εφαρμογών.

Σύστημα Βάσης Δεδομένων (ΣΒΔ)

Ένα Σύστημα Βάσης Δεδομένων (ΣΒΔ) ή DBS (Data Base System) αποτελείται από το υλικό, το λογισμικό, τη βάση δεδομένων και τους χρήστες. Είναι δηλαδή ένα σύστημα με το οποίο μπορούμε να αποθηκεύσουμε και να αξιοποιήσουμε δεδομένα με τη βοήθεια ηλεκτρονικού υπολογιστή. Αναλυτικά :

- Το υλικό (hardware) αποτελείται όπως είναι γνωστό από τους ηλεκτρονικούς υπολογιστές, τα περιφερειακά, τους σκληρούς δίσκους, τις μαγνητικές ταινίες κ.ά., όπου είναι αποθηκευμένα τα αρχεία της βάσης δεδομένων αλλά και τα προγράμματα που χρησιμοποιούνται για την επεξεργασία τους.
- Το λογισμικό (software) είναι τα προγράμματα που χρησιμοποιούνται για την επεξεργασία των δεδομένων (στοιχείων) της βάσης δεδομένων.
- Η βάση δεδομένων (database) αποτελείται από το σύνολο των αρχείων όπου είναι αποθηκευμένα τα δεδομένα του συστήματος. Τα στοιχεία αυτά μπορεί να βρίσκονται αποθηκευμένα σ' έναν φυσικό υπολογιστή αλλά και σε περισσότερους. Όμως, στον χρήστη δίνεται η εντύπωση ότι βρίσκονται συγκεντρωμένα στον ίδιο υπολογιστή. Τα δεδομένα των αρχείων αυτών είναι ενοποιημένα (data integration), δηλ. δεν υπάρχει πλεονασμός (άσκοπη επανάληψη) δεδομένων και μερισμένα (data sharing), δηλ. υπάρχει δυνατότητα ταυτόχρονης προσπέλασης των δεδομένων από πολλούς χρήστες. Ο κάθε χρήστης έχει διαφορετικά δικαιώματα και βλέπει διαφορετικό κομμάτι της βάσης δεδομένων, ανάλογα με τον σκοπό για τον οποίο συνδέεται.
- Οι χρήστες (users) μιας βάσης δεδομένων χωρίζονται στις εξής κατηγορίες :
- Τελικοί χρήστες (end users). Χρησιμοποιούν κάποια εφαρμογή για να παίρνουν στοιχεία από μια βάση δεδομένων, έχουν τις λιγότερες δυνατότητες επέμβασης στα στοιχεία της βάσης δεδομένων, χρησιμοποιούν ειδικούς κωδικούς πρόσβασης και το σύστημα τους επιτρέπει ανάλογα πρόσβαση σε συγκεκριμένο κομμάτι της βάσης δεδομένων.
- Προγραμματιστές εφαρμογών (application programmers). Αναπτύσσουν τις εφαρμογές του ΣΒΔ σε κάποια από τις γνωστές γλώσσες προγραμματισμού.
- Διαχειριστής δεδομένων (data administrator – DA). Έχει τη διοικητική αρμοδιότητα και ευθύνη για την οργάνωση της βάσης δεδομένων και την απόδοση δικαιωμάτων πρόσβασης στους χρήστες.
- Διαχειριστής βάσης δεδομένων (database administrator – DBA). Λαμβάνει οδηγίες από τον διαχειριστή δεδομένων και είναι αυτός που διαθέτει τις τεχνικές γνώσεις και αρμοδιότητες για τη σωστή και αποδοτική λειτουργία του ΣΔΒΔ.

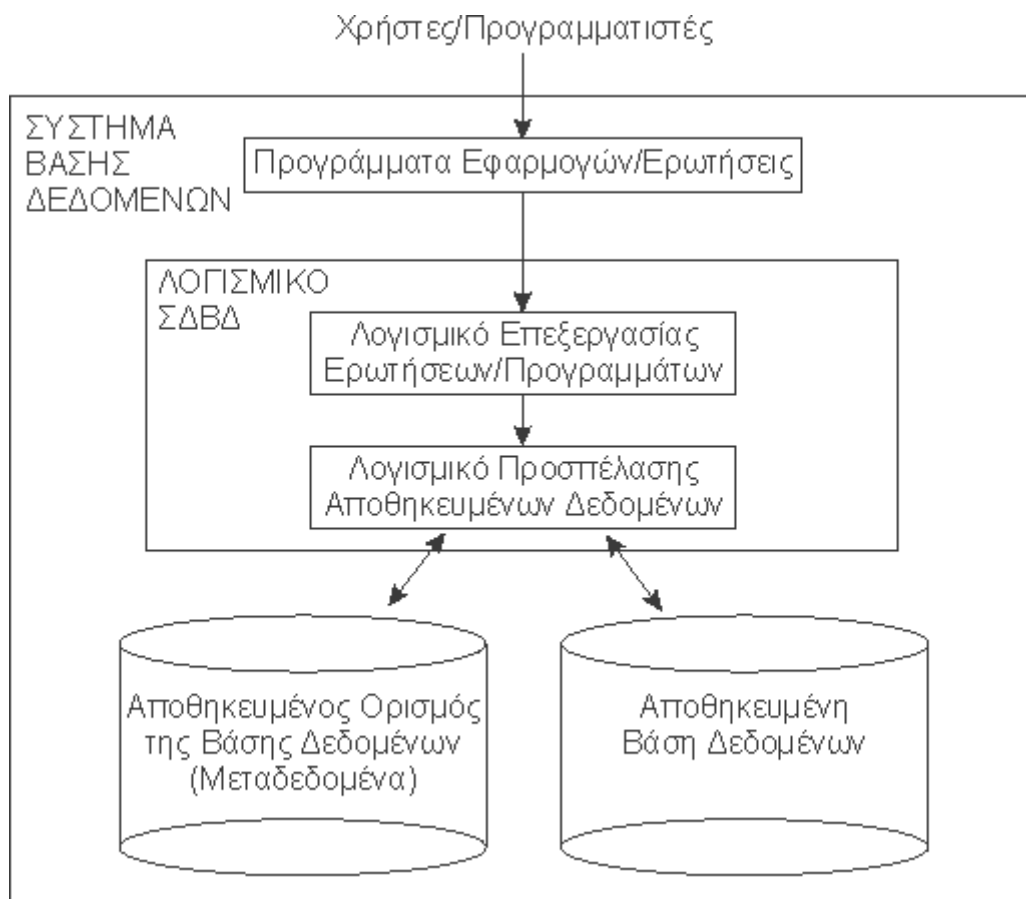
Ο υπεύθυνος διαχείρισης βάσεων δεδομένων (DBA)

Ο υπεύθυνος διαχείρισης βάσεων δεδομένων (DBA) είναι το άτομο που παρέχει την απαιτούμενη τεχνική υποστήριξη για την υλοποίηση αυτών των αποφάσεων. Ο DBA είναι λοιπόν υπεύθυνος για το συνολικό έλεγχο του συστήματος σε τεχνικό επίπεδο. Γενικά ,αυτές οι λειτουργίες είναι οι εξής:

- Ορισμός του εννοιολογικού σχήματος. Αφού ο υπεύθυνος διαχείρισης δεδομένων καθορίσει τη λογική ή μερικές φορές εννοιολογική σχεδίαση της βάσης δεδομένων εντοπίζοντας τις σημαντικές οντότητες και άρα και τις σχετικές με αυτές πληροφορίες που θα τηρούνται, ο DBA δημιουργεί το αντίστοιχο εννοιολογικό σχήμα, χρησιμοποιώντας την εννοιολογική DDL(γλώσσα ορισμού δεδομένων).
- Ορισμός του εσωτερικού σχήματος. Αποφασίζει για τη φυσική σχεδίαση της βάσης δεδομένων, και δημιουργεί τον αντίστοιχο ορισμό αποθηκευτικής δομής, χρησιμοποιώντας την εσωτερική DDL. Ο DBA πρέπει επίσης να ορίσει και τη σχετική απεικόνιση μεταξύ του εσωτερικού και του εννοιολογικού σχήματος. Οι αντίστοιχες απεικονίσεις θα υπάρχουν και σε μορφή πηγαίου κώδικα και σε μορφή αντικειμένου κώδικα.
- Επαφή με τους χρήστες. Εξασφαλίζει ότι τα δεδομένα που χρειάζονται είναι διαθέσιμα. Γράφει τα απαραίτητα εξωτερικά σχήματα, χρησιμοποιώντας την κατάλληλη εξωτερική DDL ώστε να οριστεί η απεικόνιση ανάμεσα σε οποιοδήποτε δεδομένο εξωτερικό σχήμα και στο εννοιολογικό σχήμα. Συγχρόνως παρέχει συμβουλές για την σχεδίαση των εφαρμογών ,τεχνική εκπαίδευση, εντόπιση και επίλυση προβλημάτων.

- Ορισμός κανόνων ασφάλειας και ακεραιότητας. Οι κανόνες ασφάλειας και ακεραιότητας μπορούν να θεωρηθούν μέρος του εννοιολογικού σχήματος. Η εννοιολογική DDL θα πρέπει να διαθέτει λειτουργίες για τον καθορισμό τέτοιων κανόνων.
- Ορισμός διαδικασιών για τη λήψη εφεδρικών αντιγράφων και την ανάκαμψη. Από τη στιγμή που μια επιχείρηση υιοθετεί ένα σύστημα βάσης δεδομένων, εξαρτάται σε κρίσιμο βαθμό από την επιτυχημένη λειτουργία αυτού του συστήματος. Σε περίπτωση που παρουσιαστεί μια αστοχία σε οποιοδήποτε μέρος της βάσης δεδομένων που μπορεί να οφείλεται είτε σε ανθρώπινο σφάλμα είτε σε κάποια αστοχία του υλικού ή του λειτουργικού συστήματος είναι απαραίτητο να μπορούν να αποκατασταθούν τα δεδομένα που επηρεάστηκαν, με ελάχιστη δυνατή καθυστέρηση και με τις ελάχιστες δυνατές επιπτώσεις στο υπόλοιπο σύστημα. Ο DBA θα πρέπει να ορίσει και να υλοποιήσει έναν κατάλληλο μηχανισμό ανάκαμψης.

Σύστημα διαχείρισης βάσεων δεδομένων (ΣΔΒΔ) (database management system - DBMS) είναι μια συλλογή από προγράμματα που επιτρέπουν στους χρήστες να δημιουργήσουν και να συντηρήσουν μία βάση δεδομένων. Επομένως, το ΣΔΒΔ είναι ένα γενικής χρήσης σύστημα λογισμικού που διευκολύνει τις διαδικασίες ορισμού, κατασκευής και χειρισμού βάσεων δεδομένων για διάφορες εφαρμογές.



Εικόνα 1.1 : Σύστημα διαχείρισης βάσεων δεδομένων

Το ΣΔΒΔ παρέχει στους χρήστες και τις εφαρμογές που αναπτύσσουν οι προγραμματιστές ένα επίπεδο λογισμικού για την επεξεργασία ερωτήσεων και προγραμμάτων. Για την πρόσβαση στον

ορισμό της βάσης και τα δεδομένα χρησιμοποιείται το επίπεδο προσπέλασης αποθηκευμένων δεδομένων.

Τα επιθυμητά χαρακτηριστικά ενός ΣΔΒΔ είναι τα ακόλουθα:

- μείωση πλεοναζόντων δεδομένων
- ευκολία στην ανάπτυξη νέων εφαρμογών
- μηχανισμοί ασφαλείας
- υποστήριξη δοσοληψιών (transactions)
- ανεξαρτησία δεδομένων από τις εφαρμογές τόσο στο φυσικό επίπεδο της αποθήκευσης, όσο και στις διαφορετικές παρεχόμενες όψεις

Η χρήση ΣΔΒΔ σε σχέση με αποθήκευση σε απλά αρχεία πλεονεκτεί:

- στην ασφάλεια δεδομένων
- στη διαφύλαξη της ακεραιότητας των δεδομένων
- στην ταυτόχρονη πρόσβαση από πολλούς χρήστες και εφαρμογές
- στην καταγραφή στοιχείων χρήσης
- στη ρύθμιση του τρόπου αποθήκευσης σε διαφορετικούς δίσκους και υπολογιστές
- στη ρύθμιση της οργάνωσης της αποθήκευσης των δεδομένων για την βελτιστοποίηση της απόδοσης (π.χ. χρήση δομών δέντρων)
- στη γρήγορη πρόσβαση στα δεδομένα

Οι Πίνακες Βάσεων Δεδομένων (Database Tables)

Οι βάσεις δεδομένων (databases) περιέχουν αντικείμενα (objects) που ονομάζονται Πίνακες (Tables). Οι Εγγραφές (Records) των δεδομένων αποθηκεύονται σ' αυτούς τους πίνακες. Οι Πίνακες αναγνωρίζονται με τα ονόματά τους, όπως "Persons", "Orders", "Suppliers" κ.ά.

Οι Πίνακες περιέχουν Στήλες (Columns) και Γραμμές (Rows) με δε-δομένα. Οι Γραμμές (Rows) περιέχουν εγγραφές (records), όπως μία εγγραφή για κάθε άτομο. Οι Στήλες (Columns) περιέχουν δεδομένα, όπως First Name, Last Name, Address και City.

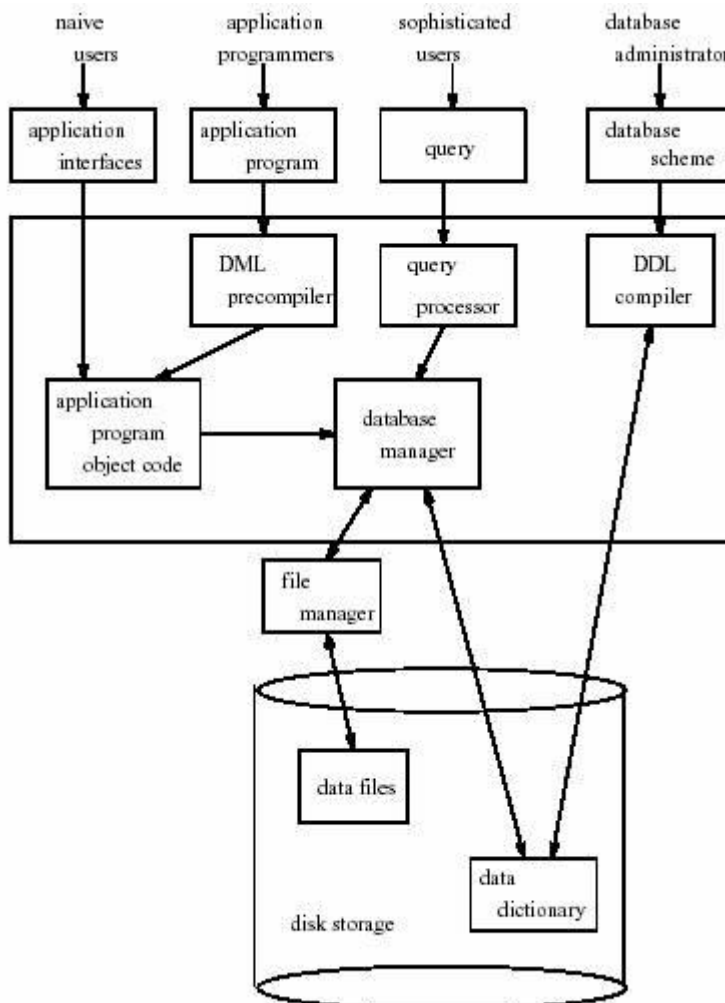
Τα Στοιχεία ενός πίνακα

Τα δεδομένα μιας βάσης δεδομένων αποθηκεύονται (οργανώνονται) στις εξής στοιχειώδεις μορφές :

- Πεδίο (Field), είναι το μικρότερο κομμάτι δεδομένων στο οποίο μπορούμε να αναφερθούμε και περιέχει ένα μόνο χαρακτηριστικό ή ιδιότητα ενός στοιχείου της βάσης δεδομένων.
- Εγγραφή (Record), είναι ένα σύνολο από διαφορετικά πεδία που περιέχει όλες τις πληροφορίες για ένα στοιχείο της βάσης δεδομένων.
- Αρχείο (File), είναι ένα σύνολο από πολλά παρόμοια στοιχεία (εγγραφές) της βάσης δεδομένων.
- Υπερ-Κλειδί (Super Key), είναι ένα πεδίο ή συνδυασμός περισσότερων πεδίων που χαρακτηρίζει μοναδικά μια εγγραφή.
- Υποψήφιο-Κλειδί (Candidate Key), είναι το υπερ-κλειδί με το λιγότερο αριθμό πεδίων που χαρακτηρίζουν μοναδικά μια εγγραφή.
- Πρωτεύον Κλειδί (Primary Key), είναι ένα πεδίο ή συνδυασμός πεδίων που χαρακτηρίζει μοναδικά μια εγγραφή. Επιλέγουμε ένα από τα διαθέσιμα υποψήφια κλειδιά ως πρωτεύον κλειδί.
- Ξένο Κλειδί (Foreign Key), είναι ένα πεδίο που έχει το ίδιο σύνολο τιμών με το πρωτεύον κλειδί ενός άλλου πίνακα.

Γενική Δομή Συστημάτων Βάσεων Δεδομένων

Τα συστατικά μέρη ενός συστήματος διαχείρισης βάσης δεδομένων (DBMS) είναι λειτουργικές μονάδες που συνεργάζονται μεταξύ τους όπως φαίνεται στην Εικόνα 2 και αναλαμβάνουν τον έλεγχο για την σωστή λειτουργία της βάσης.



Εικόνα 1.2: Δομή συστημάτων βάσεων δεδομένων.

Για μερικές λειτουργίες τους (π.χ. συστήματα αρχείων) μπορείτε να ανατρέξετε και να ενημερωθείτε από τα λειτουργικά συστήματα.

Τα λειτουργικά συστατικά μέρη ενός συστήματος Βάσεων Δεδομένων μπορούν να διαιρεθούν γενικά σε δύο μεγάλες κατηγορίες α. στα συστατικά του διαχειριστή αποθήκευσης και β. στα συστατικά του επεξεργαστή ερωτημάτων.

A. Διαχειριστής Αποθήκευσης (Storage manager)

Είναι υπεύθυνος για την ανάκληση αποθήκευση, ενημέρωση των δεδομένων της βάσης και παρέχει την διασύνδεση μεταξύ των δεδομένων που έχουν αποθηκευθεί στον δίσκο και των ερωτημάτων που στέλνονται στο σύστημα βάσης δεδομένων από την εκάστοτε εφαρμογή.

Ο διαχειριστής αρχείων (File manager): διαχειρίζεται την κατανομή του διαστήματος δίσκων και των δομών δεδομένων που χρησιμοποιούνται για να αντιπροσωπεύσουν τις πληροφορίες για δίσκος.

Διαχειριστής βάσεων δεδομένων (database manager): αποτελεί τη διεπαφή μεταξύ των προγραμμάτων χαμηλού επιπέδου δεδομένων και εφαρμογής καθώς και των ερωτήσεων.

Διαχειριστής buffer (Buffer manager): είναι υπεύθυνος για την τροφοδοσία των δεδομένων από τον δίσκο προς την κύρια μνήμη. Αποφασίζει για την προτεραιότητα των δεδομένων που θα προσπελαστούν από την κύρια μνήμη. Αποτελεί ένα από τα σημαντικότερα μέρη του συστήματος καθώς καλείται να διαχειριστεί όγκο δεδομένων που τις περισσότερες φορές είναι πολύ μεγαλύτερος από το μέγεθος της κύριας μνήμης.

Διαχειριστής συναλλαγών (transaction manager): είναι υπεύθυνος για την διατήρηση της βάσης σε συνεπή κατάσταση είναι μετά από ακάκωχη από αστοχία του μέσου ή του συστήματος. Επίσης διατηρεί την συνέπεια της βάσης και κατά την διάρκεια ταυτόχρονων συναλλαγών από πολλαπλούς χρήστες χωρίς διενέξεις και προβλήματα.

Διαχειριστής ελέγχου ταυτότητας χρηστών και ακεραιότητας (Authorization and integrity control manager): είναι υπεύθυνος για τον έλεγχο της ακεραιότητας της βάσης δηλ. τα δεδομένα που εισάγονται στην βάση να είναι σωστά και σύμφωνα με τους αντίστοιχους κανόνες. Επίσης ελέγχει τους χρήστες εάν έχουν πρόσβαση στα δεδομένα.

B. Ο επεξεργαστής ερωτημάτων (query processor)

Μεταφράζει τις δηλώσεις από μια γλώσσα διατύπωσης ερωτημάτων στις χαμηλού επιπέδου εντολές.

Ο προ-μεταγλωττιστής (precompiler dml) μετατρέπει τις δηλώσεις ορισμού δεδομένων που ενσωματώνονται σε ένα πρόγραμμα εφαρμογής στην κανονική διαδικασία κλήσης σε μια γλώσσα οικοδεσποτών (host). Ο Precompiler αλληλεπιδρά με τον επεξεργαστή ερωτημάτων.

Ο μεταγλωττιστής ddl (ddl compiler) μετατρέπει τις δηλώσεις ddl σε ένα σύνολο πινάκων που περιέχουν τα μεταδεδομένα και αποθηκεύονται σε ένα λεξικό δεδομένων.

Ο διαχειριστής αποθήκευσης χειρίζεται τις διάφορες δομές δεδομένων ως μέρος του φυσικού χειρισμού των δεδομένων με αποτέλεσμα την φυσική καταγραφή των δεδομένων σε μέσα αποθήκευσης.

Φυσική αποθήκευση –Δίσκος (Physical storage)

Αρχεία δεδομένων (data files) αποθηκεύεται η ίδια τη βάση δεδομένων σε μια περιοχή του δίσκου.

Λεξικό δεδομένων (data dictionary) πληροφορίες μεταδεδομένων σχετικές με τη δομή της βάσης δεδομένων. Μεγάλη έμφαση πρέπει να δοθεί στην ανάπτυξη ενός καλού σχεδίου ώστε να υπαρχει αποδοτική εφαρμογή του λεξικού.

Ευρετήρια-Δείκτες (index) Ταξινομημένα πινακάκια των δεδομένων με την βοήθεια των ποίων επιτυγχάνεται η γρήγορη πρόσβαση στα δεδομένα της βάσης.

Συνοπτικά μπορούμε να παρουσιάσουμε την παρακάτω δεδνρική δομή των λειτουργικών μονάδων ενός συστήματος βάσεων δεδομένων και των συστατικών τους.

- Query processor
 - DML compiler
 - Embedded DML precompiler
 - DDL interpreter
 - Query processing unit
- Storage manager
 - Authorization and integrity control
 - Transactions management
 - File management

- Buffer management
- Physical storage
- Data files,
- Meta-data (data dictionary),
- Index, statistics

Εισαγωγή Στην Έννοια Client-Server Computing

Client-server computing

Γενικά, το client-server computing αναφέρεται σε μια βασική αλλαγή στο στυλ των υπολογιστών, την αλλαγή από τα συστήματα που βασίζονται στα μηχανήματα στα συστήματα που βασίζονται στον χρήστη.

Ειδικότερα, ένα σύστημα client-server είναι ένα σύστημα στο οποίο το δίκτυο ενώνει διάφορους υπολογιστικούς πόρους, ώστε οι clients (ή αλλιώς front end) να μπορούν να ζητούν υπηρεσίες από έναν server (ή αλλιώς back end), ο οποίος προσφέρει πληροφορίες ή επιπρόσθετη υπολογιστική ισχύ. Με άλλα λόγια, στο client-server μοντέλο, ο client θέτει μια αίτηση και ο server επιστρέφει μια ανταπόκριση ή κάνει μια σειρά από ενέργειες. Ο server μπορεί να ενεργοποιείται άμεσα για την αίτηση αυτή ή να προσθέτει την αίτηση σε μια ουρά. Η άμεση ενεργοποίηση για την αίτηση μπορεί, για παράδειγμα, να σημαίνει ότι ο server υπολογίζει έναν αριθμό και τον επιστρέφει αμέσως στον client. Η τοποθέτηση της αίτησης σε μια ουρά μπορεί να σημαίνει ότι η αίτηση πρέπει να τεθεί σε αναμονή για να εξυπηρετηθεί. Ένα καλό παράδειγμα για αυτό είναι όταν εκτυπώνουμε ένα κείμενο σε ένα εκτυπωτή δικτύου. Ο server τοποθετεί την αίτηση σε μια ουρά μαζί με αιτήσεις εκτυπώσεων και από άλλους clients. Μετά επεξεργάζεται την αίτηση με βάση την σειρά προτεραιότητας, η οποία, σε αυτή την περίπτωση, καθορίζεται από τη σειρά με την οποία ο server παρέλαβε την απαίτηση.

Το client-server computing είναι πολύ σημαντικό, διότι επιτυγχάνει τα εξής:

- Αποτελεσματική χρήση της υπολογιστικής ισχύος.
- Μείωση του κόστους συντήρησης, δημιουργώντας συστήματα client-server που απαιτούν λιγότερη συντήρηση και κοστίζουν λιγότερο στην αναβάθμιση.
- Αύξηση της παραγωγικότητας, προσφέροντας στους χρήστες ξεκάθαρη πρόσβαση στις αναγκαίες πληροφορίες μέσω σταθερών και εύκολων στην χρήση διασυνδέσεων.
- Αύξηση της ευελιξίας και της δυνατότητας δημιουργίας συστημάτων που υποστηρίζουν πολλά περιβάλλοντα.

Server

Ο server απαντάει στις αιτήσεις που γίνονται από τους clients. Ένας client μπορεί να ενεργεί ως server εάν λαμβάνει, επεξεργάζεται αιτήσεις και τις στέλνει (για παράδειγμα, ένας σταθμός εργασίας που χρησιμοποιείται και ως server εκτυπώσεων από άλλους). Οι server δεν ξεκινάνε τις επικοινωνίες -περιμένουν τις αιτήσεις των clients.

Επιστρέφοντας στο παράδειγμα του server εκτυπώσεων ενός δικτύου, ο client ζητάει από τον server να εκτυπώσει ένα κείμενο σε έναν συγκεκριμένο εκτυπωτή και ο server προσθέτει την εκτύπωση σε μια ουρά και ενημερώνει τον client όταν το κείμενο εκτυπωθεί επιτυχημένα. Η διαδικασία του client μπορεί να ανήκει φυσικά στον ίδιο σταθμό εργασίας με την διαδικασία του server. Στο παράδειγμα εδώ, μια εντολή εκτύπωσης μπορεί να εκδίδεται στον server του σταθμού εργασίας του δικτύου, χρησιμοποιώντας την διαδικασία του server εκτυπώσεων σε αυτόν τον σταθμό εργασίας.

Μια server μηχανή πρέπει να μπορεί να κάνει τα ακόλουθα :

- Να αποθηκεύει, να ανακτά και να προστατεύει πληροφορίες.
- Να επιθεωρεί τις αιτήσεις των clients.

- Να δημιουργεί εφαρμογές διαχείρισης πληροφοριών, όπως δημιουργία αντιγράφων, ασφάλεια κτλ.
- Να διαχειρίζεται πληροφορίες

Τύποι των Servers

Οι servers μπορούν να διαιρεθούν σε έξι τύπους:

- Server Εφαρμογών (Application servers).
- Server Πληροφοριών (Data servers).
- Server Υπολογισμών (Computing servers).
- Server Βάσεων Δεδομένων (Database servers).
- Server Πόρων ή Επικοινωνιών (Resource or Communications servers).

Ο τύπος του server που χρησιμοποιείται εξαρτάται από την απαιτούμενη εργασία. Επίσης, αυτοί οι έξι ρόλοι μπορούν να συνδυαστούν σε ένα σύστημα ή να διαιρεθούν σε περισσότερα. Για παράδειγμα, η ίδια μηχανή μπορεί να εξυπηρετήσει σαν ένας server εφαρμογών και ένας server βάσεων δεδομένων.

Οι περισσότεροι servers που χρησιμοποιούνται σήμερα στις επιχειρήσεις είναι servers αρχείων (file servers). Οι servers αρχείων επιτρέπουν στους clients να προσπελάσουν αρχεία και να μοιραστούν πληροφορίες και λογισμικό. Αυτοί οι servers είναι συνήθως ένας προσωπικός υπολογιστής ή ένα UNIX σύστημα με έναν επεξεργαστή. Πολλοί άνθρωποι μπορούν να προσπελάσουν τον server αρχείων την ίδια στιγμή, που σημαίνει ότι ο server έχει πολλαπλές μονάδες δίσκων και κάρτες προσαρμογής δικτύου, αλλά μόνο ένα άτομο μπορεί να προσπελάσει ένα συγκεκριμένο αρχείο εκείνη τη στιγμή.

Server Εφαρμογών (Application servers)

Οι servers εφαρμογών (application servers) τρέχουν λογισμικό εφαρμογών, που είναι πολύ σημαντικό όταν διανέμονται λογικές εφαρμογών μεταξύ του client και του server. Η τοποθέτηση εφαρμογών στον server σημαίνει ότι αυτές οι εφαρμογές είναι διαθέσιμες σε πολλούς clients. Πολλοί clients μπορούν να χρησιμοποιήσουν τα RPCs (Remote Procedure Calls) για να θέσουν σε λειτουργία μια επεξεργασία στον server. Πολλοί servers εφαρμογών μπορούν ακόμα και να εργαστούν μαζί για να απαντήσουν στην απαίτηση του client. Κάθε server μπορεί να τρέξει ένα διαφορετικό λειτουργικό σύστημα σε μια διαφορετική πλατφόρμα υλικού, αλλά αυτές οι λεπτομέρειες είναι ξεκάθαρες στον client -ο client μπορεί να κάνει αιτήσεις χωρίς να υπολογίζει τον τύπο της μηχανής που θα ανταποκριθεί.

Server Πληροφοριών (Data servers)

Οι servers πληροφοριών (data servers) χρησιμοποιούνται μόνο για αποθήκευση και διαχείριση πληροφοριών και χρησιμοποιούνται σε συνδυασμό με servers υπολογισμών (compute servers). Αυτοί οι servers ερευνούν και ελέγχουν την αξιοπιστία των πληροφοριών, αλλά γενικά δεν μεταβιβάζουν μεγάλη ποσότητα πληροφοριών στο δίκτυο.

Server Υπολογισμών (Computing servers)

Οι servers υπολογισμών (compute servers) παίρνουν τις αιτήσεις των clients για πληροφορίες στον server πληροφοριών και μετά προωθούν τα αποτελέσματα των αιτήσεων πίσω στον client.

Server Βάσεων Δεδομένων (Database servers)

Οι servers βάσεων δεδομένων (database servers) είναι τυπικά client-server συστήματα, και έχουν να κάνουν την ίδια εργασία με αυτή που κάνουν οι servers πληροφοριών και υπολογισμών μαζί. Οι servers βάσεων δεδομένων τρέχουν DBMS (Database Management System) λογισμικό και πολύ πιθανό και κάποια λογική client-server εφαρμογή, που σημαίνει ότι αυτός ο τύπος του server χρειάζεται περισσότερη ισχύ. Τα DBMS προσφέρουν εξειδικευμένες υπηρεσίες: την δυνατότητα να ανακτά πληροφορίες και να διαχειρίζεται πληροφορίες. Οι servers που συνδυάζουν τις λειτουργίες του server βάσεων δεδομένων και του server εφαρμογών είναι επίσης γνωστοί ως server συναλλαγών (transaction servers).

Server Πόρων ή Επικοινωνιών (Resource or Communication servers)

Οι servers πόρων (resource servers), που περικλείουν τους servers επικοινωνιών (communications servers) επιτρέπουν σε πολλούς clients την προσπέλαση συγκεκριμένων πόρων, που είναι ουσιαστικά πολύ ακριβοί για να βρίσκονται σε έναν client. Για παράδειγμα, οι servers εκτυπώσεις (print servers) συνδέουν πολλούς clients με πολλούς εκτυπωτές. Οι servers επικοινωνιών συνδέουν απομακρυσμένα συστήματα. Άλλοι servers πόρων μπορούν να συνδέσουν clients με άλλες συσκευές, όπως πολυμέσα. Συνήθως από τη στιγμή που οι servers πόρων είναι συνδεδεμένοι σε μια συγκεκριμένη συσκευή, δεν απαιτείται τόσο πολύ ισχύ, όση αυτή των servers που προσφέρουν περισσότερο περιπλοκές υπηρεσίες.

Ένας εύκολος τρόπος για να ξεχωρίσουμε τους servers εφαρμογών, βάσεων δεδομένων και συναλλαγών είναι πώς ο client κάνει αιτήσεις στον server. Οι servers δέχονται τους παρακάτω τύπους απαιτήσεων από τους clients:

- Οι servers εφαρμογών ενεργούν κάπως πιο αποκεντρωμένα από τη βάση δεδομένων σε απάντηση του client.
- Οι servers βάσεων δεδομένων επιστρέφουν πληροφορία σαν απάντηση σε μια αίτηση του client, που γίνεται σε SQL.
- Οι servers συναλλαγών επιστρέφουν πληροφορία σαν απάντηση σε ένα μήνυμα που αποτελείται από ένα σύνολο εντολών SQL. Αυτό το σύνολο επιτυγχάνει ή αποτυγχάνει σαν μια μονάδα.

Από τους έξι τύπους, οι client-server εφαρμογές συνήθως χρησιμοποιούν πιο πολύ servers εφαρμογών, βάσεων δεδομένων και συναλλαγών ή κάποιο συνδυασμό αυτών των τριών.

Ο ρόλος του server

Η client-server διαδικασία μπορεί να απλοποιηθεί στα παρακάτω βήματα:

- 1) Ο χρήστης στέλνει μια αίτηση ή ένα ερώτημα, μέσω του client, στον server.
- 2) server ακούει την αίτηση του client.
- 3) Από τη στιγμή που ο server ακούει την αίτηση, ελέγχει την δυνατότητα πρόσβασης του χρήστη.
- 4) Ο server επεξεργάζεται το ερώτημα.
- 5) Ο server επιστρέφει τα αποτελέσματα στον client.
- 6) client δέχεται τα αποτελέσματα και τα παρουσιάζει στον χρήστη.

Από αυτά τα έξι βήματα, ο server παίζει τέσσερις σημαντικούς ρόλους. Είναι φανερό ότι ο server είναι η καρδιά της client-server εφαρμογής. Ο server υπάρχει για να απαντήσει στις ανάγκες του client, και ο client εξαρτάται από την αξιοπιστία και την έγκαιρη απάντηση του server.

Ο server πρέπει να εκτελέσει τις ακόλουθες λειτουργίες:

- Να ακούσει την αίτηση του client .
- Να ελέγξει την δυναμικότητα πρόσβασης του χρήστη.
- Να επεξεργαστεί την αίτηση.
- Να επιστρέψει τα αποτελέσματα.

Ο server δεν «εγκαινιάζει» καμιά ενέργεια. Αντίθετα, ο server περιμένει παθητικά να φτάσουν οι αιτήσεις του client μέσω του δικτύου. Ο server πρέπει πάντα να απαντάει στους clients, ακόμα και όταν πολλοί clients κάνουν ταυτόχρονες αιτήσεις.

Από την στιγμή που ο server δέχεται από τον client την απαίτηση, ο server πρέπει να βεβαιωθεί ότι ο client είναι εξουσιοδοτημένος να λάβει την πληροφορία ή την απάντηση. Αν ο client δεν είναι εξουσιοδοτημένος, ο server απορρίπτει την αίτηση και στέλνει μήνυμα στον client. Εάν ο client είναι εξουσιοδοτημένος, ο server συνεχίζει και επεξεργάζεται την αίτηση.

Η επεξεργασία της αίτησης περιλαμβάνει την παραλαβή της αίτησης του client, την μετατροπή του σε μια μορφή που μπορεί ο server να χρησιμοποιήσει και την επεξεργασία της ίδιας της αίτησης.

Όταν η επεξεργασία ολοκληρώνεται, ο server στέλνει τα αποτελέσματα πίσω στον client. Μετά, ο client μπορεί να μεταφράσει και να χρησιμοποιήσει τις πληροφορίες.

Δεν υπάρχει προκαθορισμένος διαχωρισμός στις ευθύνες για τις client-server εφαρμογές. Ανάλογα με τις ανάγκες μας, μπορούμε και να διαχωρίσουμε την εφαρμογή. Το ισχυρό client μοντέλο δίνει περισσότερες λειτουργίες στον client, ενώ το ισχυρό server μοντέλο δίνει περισσότερες λειτουργίες στον server. Οι servers εφαρμογών και συναλλαγών τείνουν να είναι ισχυροί servers, ενώ οι servers βάσεων δεδομένων και αρχείων τείνουν να έχουν ισχυρούς clients.

Ανεξάρτητα του πώς διαχωρίζουμε την εφαρμογή, η βασική ευθύνη του server παραμένει η ίδια: να εξυπηρετεί τους clients που κάνουν αιτήσεις.

Ακούγοντας την αίτηση του client

Ο server δεν ξεκινάει καμιά αλληλεπίδραση με τον client, ο server απλά περιμένει τον client για να κάνει την αίτηση του. Όταν ο client κάνει την αίτηση, ο server ανταποκρίνεται το συντομότερο δυνατό.

Η κάρτα προσαρμογής στο δίκτυο συνδέει φυσικά τον server, με το δίκτυο και καθορίζει εάν οι εισερχόμενες αιτήσεις είναι κατανοητές για τον κόμβο του προσαρμογέα. Εάν ναι, το πρωτόκολλο τις αποδέχεται και τις αποκωδικοποιεί ώστε μετά ο server να μπορεί να τις επεξεργαστεί.

Ελέγχοντας την δυνατότητα πρόσβασης του χρήστη

Από την στιγμή που ο server δέχεται την αίτηση από τον client, ο server πρέπει να βεβαιωθεί ότι ο χρήστης είναι εξουσιοδοτημένος να λάβει την πληροφορία ή την ανταπόκριση από το server. Εάν ο χρήστης δεν είναι εξουσιοδοτημένος, ο server απορρίπτει την αίτηση και στέλνει ένα μήνυμα στον client. Εάν ο χρήστης είναι εξουσιοδοτημένος, ο server συνεχίζει και επεξεργάζεται την αίτηση.

Επεξεργάζοντας την αίτηση

Ο server πρέπει να είναι ικανός να ανταποκριθεί στην αίτηση του client αμέσως. Εάν πολλοί clients κάνουν αιτήσεις ταυτόχρονα, ο server πρέπει να είναι ικανός να βάζει σε προτεραιότητα τις αιτήσεις των clients, και να επεξεργάζεται πολλές αιτήσεις την στιγμή.

Από την στιγμή, που ο server επιβεβαιώνει ότι ο χρήστης είναι εξουσιοδοτημένος να κάνει αιτήσεις στον server, ο server μπορεί να αποκαλύψει την απαίτηση και να την επεξεργαστεί.

Η αίτηση μπορεί να έχει μια από τις ακόλουθες τέσσερις μορφές:

- Μια απόμακρη αίτηση είναι μια απλή αίτηση για πληροφορίες από έναν απλό client.
- Μια απόμακρη συναλλαγή περιλαμβάνει πολλαπλές αιτήσεις για πληροφορίες από έναν απλό client.
- Μια κατανεμημένη συναλλαγή περιλαμβάνει πολλαπλές αιτήσεις για πληροφορίες από έναν απλό client, οι οποίες πληροφορίες ανήκουν σε πολλούς server.
- Μια κατανεμημένη αίτηση είναι μια συναλλαγή που σχηματίζεται από πολλαπλές αιτήσεις για πληροφορίες από πολλαπλούς clients, οι οποίες πληροφορίες ανήκουν σε πολλαπλούς servers.

Αυτές οι αιτήσεις πρέπει να περάσουν από το λεγόμενο ACID τεστ: Ατομικότητα (Atomicity), Συνέπεια (Consistency), Απομόνωση (Isolation) και Αντοχή (Durability). Η ατομικότητα σημαίνει ότι ολόκληρη η συναλλαγή πρέπει να πετύχει ή να αποτύχει, δεν μπορεί να ολοκληρωθεί ως προς ένα

κομμάτι της. Η συνέπεια σημαίνει ότι το σύστημα πάει από ένα σταθερό σημείο σε ένα άλλο σταθερό σημείο. Η απομόνωση σημαίνει ότι, από τη στιγμή που μια συναλλαγή ολοκληρώνεται με επιτυχία, τα αποτελέσματα της δεν είναι ορατά σε άλλες συναλλαγές. Η αντοχή σημαίνει ότι από τη στιγμή που η συναλλαγή ολοκληρώνεται με επιτυχία, δεσμεύεται μόνιμα από το σύστημα και επακόλουθες αποτυχίες δεν θα το επηρεάσουν. Εάν η συναλλαγή αποτύχει, το σύστημα οπισθοχωρεί στο σημείο που ήταν πριν προσπαθήσει να επεξεργαστεί την συναλλαγή.

Η διαχείριση της συναλλαγής ελέγχεται είτε από το DBMS είτε από το TPM (Transaction Processing Manager). Οι διαχειριστές της συναλλαγής προστατεύουν την ακεραιότητα των πληροφοριών που είναι μια απόλυτη αξίωση. Ο server είναι υπεύθυνος για προστασία και την διατήρηση της ακρίβειας των πληροφοριών.

Όταν ο server τελειώνει την επεξεργασία των αποτελεσμάτων και είναι έτοιμος να επιστρέψει τα αποτελέσματα στον client, πρέπει να μορφοποιήσει τα αποτελέσματα και να τα στείλει με ένα τρόπο που μπορεί ο client να καταλάβει.

Ο server παραδίδει τις πληροφορίες στο πρωτόκολλο, που διευθύνει ένα πακέτο, μορφοποιεί τις πληροφορίες για να τις τοποθετήσει στο πακέτο και περνάει το πακέτο στο δίκτυο. Το δίκτυο μετά βεβαιώνεται ότι το πακέτο πηγαίνει στον client.

Client

Ο client είναι ο αιτών των υπηρεσιών. Ο client δεν μπορεί παρά να είναι ένας υπολογιστής. Οι υπηρεσίες που ζητούνται από τον client μπορεί να υπάρχουν στους ίδιους σταθμούς εργασίας ή σε απομακρυσμένους σταθμούς εργασίας που συνδέονται μεταξύ τους μέσω ενός δικτύου. Ο client ξεκινάει πάντα την επικοινωνία.

Τα συστατικά του client είναι πολύ απλά. Μια client μηχανή πρέπει να μπορεί να κάνει τα ακόλουθα:

- Να τρέχει το λογισμικό των γραφικών διεπαφών χρηστών (GUIs).
- Να δημιουργεί τις αιτήσεις για πληροφορίες και να τις στέλνει στον server.
- Να αποθηκεύει τις επιστρεφόμενες πληροφορίες.

Αυτές οι αιτήσεις καθορίζουν πόση μνήμη χρειάζεται, ποια ταχύτητα επεξεργασίας θα μπορούσε να βελτιώσει τον χρόνο ανταπόκρισης, και πόση χωρητικότητα αποθήκευσης απαιτείται.

Η διαδικασία client-server μπορεί να απλοποιηθεί στα ακόλουθα βήματα:

- 1) Ο χρήστης δημιουργεί μια αίτηση ή ένα ερώτημα.
- 2) Ο client μορφοποιεί το ερώτημα και το στέλνει στο server.
- 3) Ο server ελέγχει την δυνατότητα πρόσβασης του χρήστη.
- 4) Ο server επεξεργάζεται το ερώτημα και επιστρέφει τα αποτελέσματα.
- 5) Ο client λαμβάνει την ανταπόκριση και τη μορφοποιεί για τον χρήστη.
- 6) Ο χρήστης βλέπει και χειρίζεται την πληροφορία.

Πέρα από τα έξι αυτά βήματα, ο client παίζει τέσσερις βασικούς ρόλους. Ο client είναι στην πραγματικότητα το κέντρο της client-server εφαρμογής. Ο χρήστης αλληλεπιδρά με τον client, ο client ξεκινάει το μεγαλύτερο μέρος της ανάπτυξης της εφαρμογής, και ο server υπάρχει για να απαντάει στις ανάγκες του client.

Ο client εκτελεί τις ακόλουθες λειτουργίες :

- Προσφέρει μια εύκολη στη χρήση διασύνδεση χρηστών.
- Στέλνει απαιτήσεις.
- Δέχεται ανταποκρίσεις.
- Επιτρέπει στον χρήστη να βλέπει και να χειρίζεται τις πληροφορίες.

Για κάθε έναν από τους τέσσερις ρόλους, δηλαδή της παροχής μιας εύκολης στη χρήση διασύνδεσης, της αποστολής απαιτήσεων, της αποδοχής ανταποκρίσεων και της δυνατότητας στο χρήστη να παίρνει και να χειρίζεται πληροφορίες, ο client έχει συγκεκριμένες ευθύνες.

Η ανάπτυξη του client βασίζεται σε αρχές σχεδίασης εστιασμένες στον χρήστη. Αυτές οι αρχές είναι οι ακόλουθες:

- Διατηρεί τη διασύνδεση συνεπή, ώστε οι χρήστες να πάρουν μια οικεία όψη και αίσθηση από τις εφαρμογές και τις πλατφόρμες.
- Δεν ξεχνά ότι ο υπολογιστής εξυπηρετεί τον χρήστη. Ο χρήστης θα πρέπει να ελέγχει την σειρά των εργασιών. Ο υπολογιστής δεν θα πρέπει ποτέ να αγνοεί τον χρήστη, αλλά να διατηρεί τον χρήστη ενήμερο και να προσφέρει άμεσες απαντήσεις.
- Χρησιμοποιεί μεταφορές, τόσο φραστικές, όσο και οπτικές για να βοηθήσει τους χρήστες να αναπτύξουν θεμελιώδεις απεικονίσεις. Για παράδειγμα, η αποθήκευση αρχείων σε φακέλους στον υπολογιστή, ώστε ο χρήστης να μπορεί να συγκεντρωθεί στη δουλειά παρά να αποκαλύπτει πως λειτουργεί ο υπολογιστής.
- Δεν ζητάει από τον χρήστη να θυμάται εντολές. Οι εντολές μπορούν να είναι διαθέσιμες στον χρήστη για να τις επιλέγει, ώστε ο χρήστης να μπορεί να βασίζεται στην αναγνώριση, παρά στην απομνημόνευση.
- Επιτρέπει στην διασύνδεση να συγχωρεί τα λάθη του χρήστη. Οι καταστροφικές ενέργειες απαιτούν επιβεβαίωση, και οι χρήστες μπορούν να ανατρέψουν ή να ακυρώσουν την τελευταία ενέργεια.

Δίκτυα

Τα δίκτυα είναι τα πιο άγνωστα συστατικά στην εξίσωση των client-server. Γενικά οι άνθρωποι δεν ξέρουν πολλά για το πώς λειτουργούν τα δίκτυα στα συστήματα client-server, διότι τα συστήματα αυτά είναι σχεδιασμένα για να κάνουν τα δίκτυα διάφανα στον χρήστη. Επιπλέον, τα δίκτυα πρέπει να είναι αξιόπιστα. Πρέπει να μπορούν να υποστηρίξουν την επικοινωνία, να ελέγχουν σφάλματα και να ξεπερνούν αμέσως τις αποτυχίες.

Τα δίκτυα ελέγχονται από το λογισμικό λειτουργικών συστημάτων και διαχείρισης για να ελέγχουν τις υπηρεσίες επικοινωνίας του server και να προστατεύουν τα προγράμματα του client και του server από το να έχουν άμεση σύνδεση μεταξύ τους. Το λογισμικό διαχείρισης εστιάζεται στη παροχή αξιόπιστων υπηρεσιών, στην ελαχιστοποίηση των προβλημάτων στο δίκτυο και στην ελαχιστοποίηση των χρόνων «πτώσης» του δικτύου.

Η Σύνδεση

Εκείνο που παίζει σημαντικό ρόλο στο client server computing είναι η σύνδεση του client με τον server, δηλαδή ουσιαστικά το δίκτυο στο οποίο εντάσσονται. Οι χρήστες θέλουν να αισθάνονται ότι οι υπηρεσίες που χρειάζονται οι ίδιοι, είναι διαθέσιμες και προσπελάσιμες στο δίκτυο, χωρίς να πρέπει να λαμβάνουν υπόψη μόνο την τεχνολογία. Όταν χρειάζεται να χρησιμοποιήσουν client-server εφαρμογές, είναι απαραίτητο να προσδιορίζεται το θέμα της σύνδεσης. Τα δίκτυα Lan προσφέρουν μεγαλύτερες ταχύτητες σε σχέση με το Internet γεγονός που είναι υψίστης σημασίας στον τρόπο σχεδιασμού των εφαρμογών .

Ισχυροί servers και ισχυροί clients (fat servers και fat clients)

Εκτός από τον διαχωρισμό ανάλογα με τις υπηρεσίες που προσφέρουν, οι client-server εφαρμογές μπορούν να διακριθούν ανάλογα με το πώς η εφαρμογή κατανέμεται μεταξύ του client και του server. Το μοντέλο του ισχυρού server προσδίδει περισσότερες λειτουργίες στον server. Το μοντέλο του ισχυρού client προσδίδει περισσότερες λειτουργίες στον client. Οι Web servers είναι παραδείγματα ισχυρών servers. Οι servers βάσεων δεδομένων και αρχείων είναι παραδείγματα ισχυρών clients.

Οι ισχυροί clients είναι οι πιο παραδοσιακοί τύποι των clients-servers. Το κύριο σώμα της εφαρμογής τρέχει στην πλευρά της εξίσωσης, που ανήκει στον client. Ταυτόχρονα στον server αρχείων και τον server βάσεων δεδομένων, οι clients γνωρίζουν το πώς είναι οργανωμένες και αποθηκευμένες οι πληροφορίες στην πλευρά του server. Προσφέρουν ευλυγισία και ευκαιρίες για δημιουργία εργαλείων που επιτρέπουν στους τελικούς χρήστες να δημιουργήσουν τις δικές τους εφαρμογές.

Οι εφαρμογές των ισχυρών servers είναι πιο εύκολο να διαχειρίζονται και να αναπτύσσονται στο δίκτυο διότι το μεγαλύτερο μέρος του κώδικα τρέχει στους servers. Οι ισχυροί servers προσπαθούν να ελαχιστοποιήσουν τις ανταλλαγές στο δίκτυο δημιουργώντας πιο ουσιώδη επίπεδα υπηρεσιών. Οι servers «συναλλαγών», για παράδειγμα, συμπυκνώνουν τη βάση δεδομένων. Αντί να εξάγουν ανεπεξέργαστες πληροφορίες, εξάγουν τις διαδικασίες που χειρίζονται αυτές τις πληροφορίες. Ο client στο μοντέλο του ισχυρού client προσφέρει το GUI και αλληλεπιδρά με τον server μέσω των RPCs (Remote Procedure Calls).

Κάθε client-server μοντέλο έχει την χρησιμότητά του. Σε πολλές περιπτώσεις, τα μοντέλα αλληλοσυμπληρώνονται και δεν είναι ασυνήθιστο να συνυπάρχουν σε μια εφαρμογή. Για παράδειγμα, μια εφαρμογή θα μπορούσε να απαιτεί έναν server, ο οποίος να συνδυάζει τους servers αρχείων, βάσεων δεδομένων και συναλλαγών.

Αρχιτεκτονικές Two-tier (2-στρωμάτων) και Three-tier (3-στρωμάτων) client-server

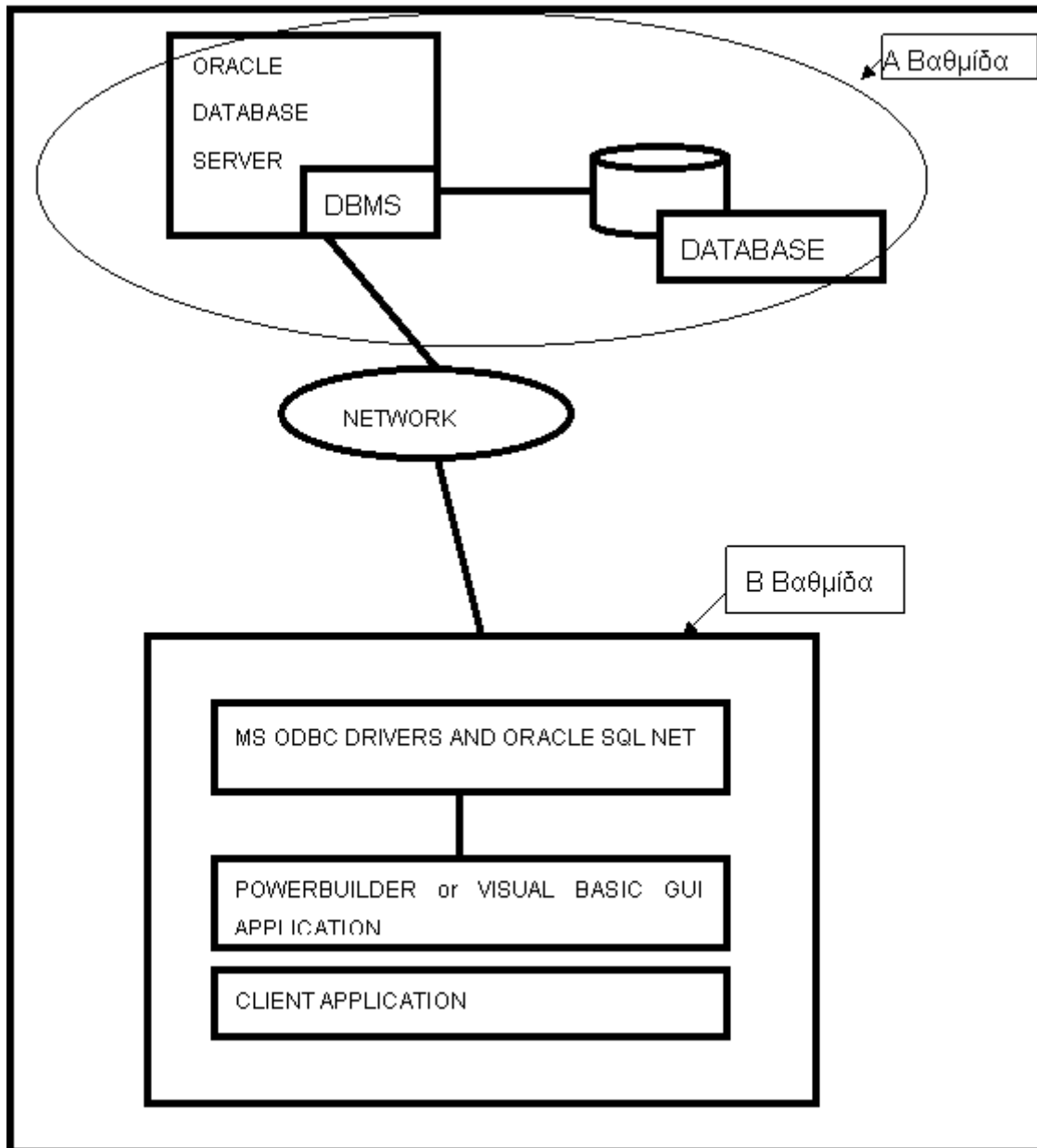
Πολλές φορές προτιμούνται να χρησιμοποιούνται όροι, όπως 2-tier, 3-tier client-server αρχιτεκτονικές αντί των όρων ισχυροί clients και ισχυροί servers. Αλλά ουσιαστικά αυτοί οι όροι βασίζονται στην ίδια βασική ιδέα. Έχουν να κάνουν με το πώς διαιρείται η client-server εφαρμογή σε λειτουργικές ενότητες, οι οποίες μετά μπορούν να ανατεθούν είτε στον client, είτε σε έναν ή περισσότερους servers.

Two-tier client-server computing

Οι client-server εφαρμογές πρώτης γενιάς εκτελούνταν γενικά με δυο λογικά στρώματα. Αυτό το μοντέλο έχει συχνά δυο στρώματα υλικού. Αυτή δεν είναι η περίπτωση όπου ο client και ο server τρέχουν ταυτόχρονα στον ίδιο υπολογιστή. Ο two-tier client-server διαιρεί την εφαρμογή σε δυο συγκεκριμένα τμήματα (τα tiers), όπου ένα τμήμα τρέχει στον client υπολογιστή και ένα ξεχωριστό τμήμα τρέχει στον server. Αξίζει να σημειωθεί ότι ο κώδικας του client και του server δεν ενημερώνει, ούτε καν γνωρίζει εάν αυτοί τρέχουν στον ίδιο υπολογιστή ή όχι. Επιπλέον, η εφαρμογή διαιρείται κατά μήκος του client και του server δηλαδή χωρίζεται σε 2 τμήματα που το καθένα τρέχει στον client και server αντίστοιχα.

Η ποσότητα της λογικής εφαρμογής που λειτουργεί στον client ή στον server καθορίζει εάν αυτό είναι αδύνατο ή ισχυρό. Το αδύνατο υποδηλώνει ότι παρουσιάζεται μικρή ανάπτυξη της εφαρμογής και το ισχυρό ότι παρουσιάζεται ένα μεγάλο τμήμα της λογικής της εφαρμογής. Υπάρχουν ποικίλλες διαβαθμίσεις αδυνάτου και ισχυρού. Οι αδύνατοι client είναι ελκυστικοί όταν ο client υπολογιστής έχει περιορισμένη επίδοση. Οι two-tier clients-server εμφανίζονται να είναι πιο δύσκολο να αναπτυχθούν και να διατηρηθούν από ό,τι κανονικά προσδοκάται. Οι two-tier εφαρμογές δεν κλιμακώνουν καλά. Επίσης, τα εργαλεία των client-server πήραν χρόνο για να αναπτυχθούν.

Στην επόμενη σελίδα ακολουθεί ένα διάγραμμα όπου διαφαίνεται η διάρθρωση μιας 2 tier εφαρμογής .



Εικόνα 1.3 : Αντιπροσωπευτικό παράδειγμα Client/Server (“TWO-TIER”) αρχιτεκτονικής

Three-tier Client Server Computing

Ο πιο πρόσφατος τύπος client-server computing που αναπτύσσεται είναι ο three-tier. Μερικοί άνθρωποι επίσης χρησιμοποιούν πολλαπλούς tiers για να περιλάβουν οποιαδήποτε προσέγγιση που χρησιμοποιεί περισσότερους από δυο λογικούς tiers.

Μια πρωταρχική διαφορά μεταξύ two-tier και three-tier εφαρμογών είναι η επιπλέον επίστρωση λογισμικού στο server. Όπου οι two-tier εφαρμογές τείνουν να τοποθετούν την λογική στον client και να περνούν εγγραφές στη βάση δεδομένων (ισχυρό client μοντέλο) ή να περνούν δεδομένα στη βάση δεδομένων, όπου αποθηκευμένες διαδικασίες εκτελούν την λογική της εφαρμογής (αδύνατο client μοντέλο), οι three-tier εφαρμογές τείνουν να περνούν μήνυμα μεταξύ των client και των server τμημάτων του κώδικα της εφαρμογής. Το τμήμα του server εφαρμόζει τη λογική της εφαρμογής, κατόπιν την στέλνει στη βάση δεδομένων. Η λογική της εφαρμογής συνήθως καλείται «Business Rules» στη χώρα των client-server.

Το τμήμα του server της three-tier αρχιτεκτονικής προσθέτει κάτι σε όλη την πολυπλοκότητα της εφαρμογής. Ωστόσο υπάρχουν ορισμένα πλεονεκτήματα σε μια three-tier client-server προσέγγιση. Αυτά περιλαμβάνουν:

- Κλιμάκωση
- Γενικότερα πιο χαμηλά προβλήματα στα δίκτυα διανομής
- Ευλυγισία

Η κλιμάκωση βελτιώθηκε, διότι ο κώδικας του server και η βάση δεδομένων είναι χωρισμένα, μπορούν να ξεκινήσουν από ένα απλό «υπολογιστή-οικοδεσπότη» και αργότερα να χωριστούν. Πολλαπλές εφαρμογές server μπορούν να επικοινωνήσουν με μια κεντρική βάση δεδομένων ή μια εφαρμογή server μπορεί ακόμα να εξυπηρετήσει τους clients ενώ προσπελάζονται οι πολλαπλές βάσεις δεδομένων όσο το σύστημα αυξομειώνεται.

Τα χαμηλότερα προβλήματα στα δίκτυα απορρέουν από το πέρασμα μικρών μηνυμάτων στην εφαρμογή παρά από ολόκληρες εγγραφές πληροφοριών.

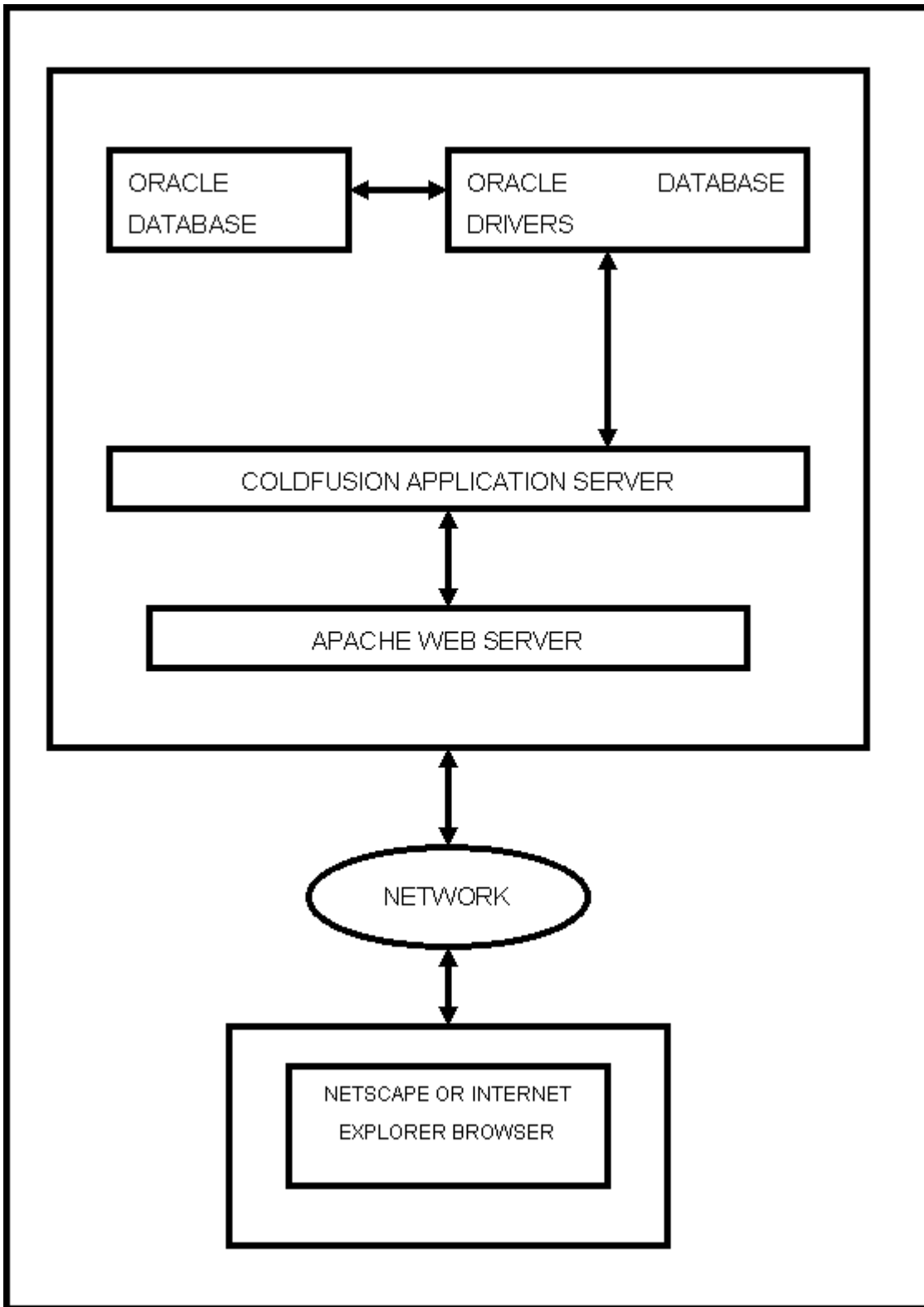
Η ευλυγισία κερδίζεται, διότι ο client, ο server και τα συστήματα βάσεων δεδομένων μπορούν το καθένα να αντικατασταθούν χωρίς να επηρεάζουν τα άλλα κομμάτια, δεδομένου ότι η διασύνδεση επίσης δεν αλλάζει. Για παράδειγμα, μετατρέποντας τη βάση δεδομένων από Sybase σε Oracle επηρεάζεται μόνο το τμήμα του server της εφαρμογής, όχι το client. Το να ξαναδιατυπώνεις έναν client από Visual Basic σε Delphi δεν έχει επίδραση στο υπόλοιπο τμήμα της εφαρμογής, δεδομένου ότι υπάρχει αλληλεπίδραση ανάμεσα στον κώδικα του client και τον κώδικα του server.

Πολλές από τις σημερινές εμπορικές εφαρμογές, που βασίζονται σε βάσεις δεδομένων, χρησιμοποιούν το three-tiered client-server μοντέλο για να κερδίσουν τη δυνατότητα αυξομείωσης και ευλυγισίας.

Επίσης οι Web εφαρμογές είναι γενικά three-tier client-server εφαρμογές.

Δηλαδή τα 3 επίπεδα μια 3 tier εφαρμογής είναι συνήθως τα εξής :

1. Βάση Δεδομένων
2. Server
3. Client



Εικόνα 1.4 : Αντιπροσωπευτικό παράδειγμα Client/Server (“THREE-TIER”) αρχιτεκτονικής

Μοντέλο Οντοτήτων-Συσχετίσεων

Το μοντέλο οντοτήτων-συσχετίσεων (entity-relationship model) προτάθηκε από τον P.P. Chen το 1976 για να περιγράψει με γραφικά σύμβολα τα δεδομένα ως οντότητες, συσχετίσεις και γνωρίσματα. Από τότε έχει αποτελέσει το αντικείμενο εκτεταμένων ερευνών με αποτέλεσμα τη διαρκή ανάπτυξή του. Σήμερα θεωρείται ένα από τα πιο σημαντικά εργαλεία κατασκευής **εννοιολογικών μοντέλων** βάσεων δεδομένων, σε περιβάλλοντα με ποικίλες απαιτήσεις, και αποτελεί βασική συνιστώσα μεθοδολογιών ανάπτυξης πληροφοριακών συστημάτων. Χρησιμοποιείται κατά το στάδιο του **εννοιολογικού σχεδιασμού (conceptual design)** και του **λογικού σχεδιασμού** της βάσης δεδομένων.

Ένα μοντέλο δεδομένων είναι ένα εργαλείο που επιτρέπει την αφαίρεση περιττών πληροφοριών από το σύνολο των πραγματικών δεδομένων. Κάθε μοντέλο βασίζεται σε ορισμένους κανόνες και διαδικασίες.

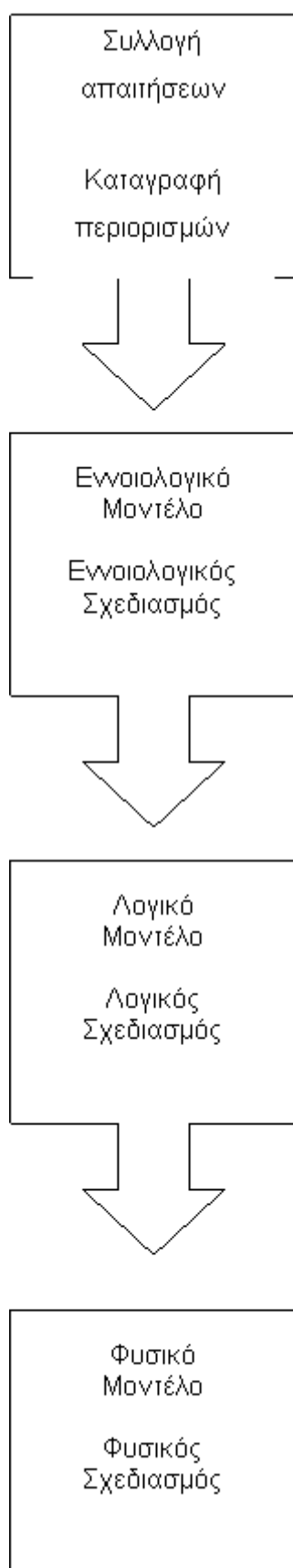
Ένα μοντέλο δεδομένων έχει τρεις βασικές συνιστώσες:

1. **Δομές (Structures):** Είναι τα αντικείμενα του μικρόκοσμου.
2. **Πράξεις (operations):** Είναι οι ενέργειες που γίνονται στις δομές
3. **Περιορισμοί (constraints):** Είναι οι περιορισμοί που μπαίνουν πάνω στις δομές.

Άρα τα μοντέλα δεδομένων είναι μία συλλογή από εργαλεία για την περιγραφή δομών δεδομένων, τη σημασιολογία (semantics) δεδομένων και τους περιορισμούς δεδομένων. Υπάρχουν τρεις διαφορετικές ομάδες που αντιστοιχούν στις αντίστοιχες φάσεις σχεδιασμού:

- **Εννοιολογικά μοντέλα** βασισμένα σε αντικείμενα
- **Λογικά μοντέλα** βασισμένα σε εγγραφές
- **Φυσικά μοντέλα** δεδομένων βασίζονται στις πραγματικές τιμές.

Παρακάτω, στην Εικόνα 2.1, φαίνονται σχηματικά οι τρεις αυτές ομάδες.



Εικόνα 2.1 Βήματα σχεδιασμού μιας Εφαρμογής

Ένα από τα βασικότερα βήματα για την επίτευξη μιας σωστά σχεδιασμένης βάσης δεδομένων είναι η καταγραφή των απαιτήσεων και των περιορισμών της εφαρμογής. Αυτό επιτυγχάνεται με συνεντεύξεις από ομάδες ανθρώπων που είναι σχετικές με το αντικείμενο της εφαρμογής, με μελέτη του επιστημονικού υπόβαθρου της εφαρμογής και με σωστή ταξινόμηση και ομαδοποίηση όλων όσων συλλεχθούν. Το βήμα αυτό συχνά παραλείπεται λόγω βιασύνης και στο τέλος αντί για οικονομία χρόνου υπάρχει λανθασμένος σχεδιασμός με ελλιπή στοιχεία και χάσιμο πολύτιμου χρόνου στο τέλος για διορθώσεις και επανασχεδιασμό.

Αφού γίνει σωστά η συλλογή των απαιτήσεων και των απαραίτητων πληροφοριών υπάρχει η ανάγκη να μπου όλες αυτές οι πληροφορίες σε μια τάξη, αφαιρώντας περιττές λεπτομέρειες και κρατώντας την ουσία. Την απαίτηση αυτή την καλύπτουν τα εννοιολογικά μοντέλα τα οποία με την βοήθεια σχηματικής αναπαράστασης βοηθούν τον αναλυτή να αποδώσει την σκέψη του με σύντομο και κατανοητό τρόπο. Τέτοιο μοντέλο είναι και μοντέλο Οντοτήτων και Συσχετίσεων που θα παρουσιάσουμε στις επόμενες παραγράφους.

Μετά τη σχηματική απόδοση που ολοκληρώνεται με τη χρήση των εννοιολογικών μοντέλων ακολουθεί η προσθήκη περισσότερων λεπτομερειών και η παρουσίαση των σχημάτων σε πιο λογικές δομές. Αυτό επιτυγχάνεται με τη χρήση των λογικών μοντέλων και στην περίπτωση των βάσεων δεδομένων, με το Σχεσιακό μοντέλο που θα επεξηγηθεί σε επόμενες παραγράφους. Το μοντέλο αυτό είναι επίσης αφαιρετικό αλλά περιέχει επιπλέον στοιχεία που βοηθούν στην κατανόηση του σχήματος της βάσης που θα αναπτυχθεί.

Τέλος πρέπει να γίνει σωστή επιλογή του συστήματος διαχείρισης βάσεων δεδομένων (RDBMS) στο οποίο θα γίνει ο φυσικός σχεδιασμός. Το βήμα αυτό είναι εξίσου σημαντικό όπως και τα προηγούμενα διότι εάν επιλεγεί λάθος RDBMS τα αποτελέσματα θα είναι δυσάρεστα με πολύ κόπο ανάπτυξης της εφαρμογής να πηγαίνει χαμένος. πχ. Να πρέπει έχουν πρόσβαση στην βάση πολλοί χρήστες ταυτόχρονα και η εφαρμογή να αναπτυχθεί σε ACCESS είναι λάθος.

2.2 Μοντέλο E-R (Οντοτήτων Συσχετίσεων).

Το μοντέλο **οντοτήτων-συσχετίσεων** είναι βασισμένο στην αντίληψη ότι ο κάθε μικρόκοσμος που θα παρασταθεί με τη βάση δεδομένων αποτελείται από μια συλλογή από βασικά αντικείμενα τις **οντότητες** και **συσχετίσεις** μεταξύ αυτών των αντικειμένων.

μια οντότητα είναι ένα διακριτό αντικείμενο.

κάθε οντότητα περιγράφεται από ένα σύνολο από ιδιότητες (γνωρίσματα)

μια συσχέτιση είναι μία σύνδεση μεταξύ διάφορων οντοτήτων.

Το σύνολο των οντοτήτων ίδιου τύπου καλείται **τύπος οντότητας** και το σύνολο των συσχετίσεων ίδιου τύπου καλείται **τύπος συσχέτισης**.

Ένα άλλο απαραίτητο στοιχείο του διαγράμματος του ER είναι οι απεικονίσεις του πλήθους των οντοτήτων οι οποίες συμμετέχουν σε κάθε συσχέτιση.

Η εννοιολογική δομή μιας βάσης δεδομένων μπορεί να εκφραστεί γραφικά από ένα **διάγραμμα οντοτήτων-συσχετίσεων** το οποίο αποτελείται από :

- **ορθογώνια:** αντιπροσωπεύουν τους τύπους οντοτήτων.
- **ρόμβοι:** αντιπροσωπεύουν τους τύπους συσχετίσεων μεταξύ των τύπων οντοτήτων.
- **ελλείψεις:** αντιπροσωπεύουν τις ιδιότητες.
- **γραμμές:** συνδέουν τους τύπους οντοτήτων με τα γνωρίσματά τους και με τους τύπους συσχετίσεών τους.

Τόσο οι οντότητες όσο και οι συσχετίσεις μπορούν να έχουν **γνωρίσματα** (attributes), τα οποία είναι ιδιότητες που τις χαρακτηρίζουν. Τα είδη των γνωρισμάτων είναι τα απλά όπου η οντότητα έχει

ατομική τιμή για αυτό, τα σύνθετα όπου το γνώρισμα αποτελείται από περισσότερες τιμές και τα πλειότιμα όπου οι πολλαπλές τιμές είναι το χαρακτηριστικό τους.

Βασικό χαρακτηριστικό ενός τύπου οντοτήτων είναι το γνώρισμα κλειδί (key). Ένα γνώρισμα ή (σύνολο γνωρισμάτων) που χαρακτηρίζει μοναδικά κάθε οντότητα (αντίστοιχα συσχέτιση) μέσα στον τύπο οντοτήτων (συσχετίσεων) ονομάζεται υπέρ – κλειδί. Δεν μπορούν δηλαδή δύο διαφορετικές οντότητες (συσχετίσεις) μέσα στον ίδιο τύπο να έχουν ίδια τιμή,

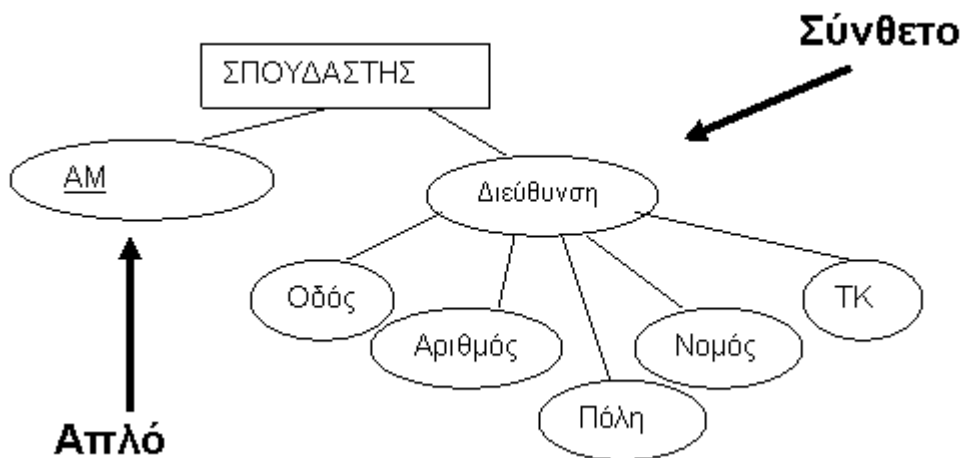
Γνωρίσματα

Τα γνωρίσματα περιγράφουν τις οντότητες και τις συσχετίσεις και μπορούν να παίρνουν τιμές. Ειδικότερα;

Απλά (Ατομικά) Γνωρίσματα

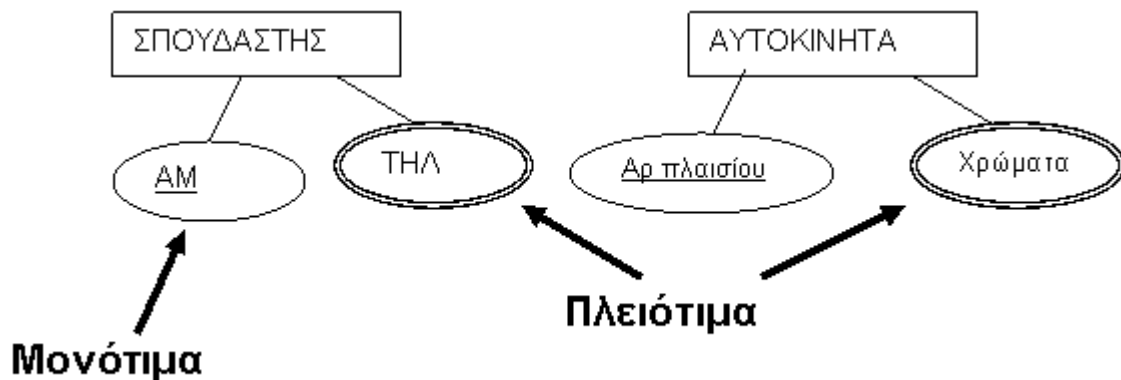
Τα γνωρίσματα αυτά (atomic attributes) όπως λέει και το όνομά τους μπορούν να πάρουν ατομικές τιμές και συμβολίζονται με μία έλλειψη.

Σύνθετα Γνωρίσματα (composite attributes) μπορούν να χωριστούν σε μικρότερα τμήματα που αναπαριστούν βασικότερα γνωρίσματα με τη δική του ανεξάρτητη σημασία το καθένα. Για παράδειγμα, το γνώρισμα Διεύθυνση της οντότητας ΣΠΟΥΔΑΣΤΗΣ μπορεί να υποδιαιρεθεί σε Οδό, Αριθμό, Πόλη, Νομό, και Ταχ_Κωδ . Γνωρίσματα που δεν υποδιαιρούνται λέγονται **απλά** (simple) ή **ατομικά** γνωρίσματα (atomic attributes). Η τιμή ενός σύνθετου γνωρίσματος είναι η συνένωση των τιμών των απλών γνωρισμάτων που το αποτελούν.

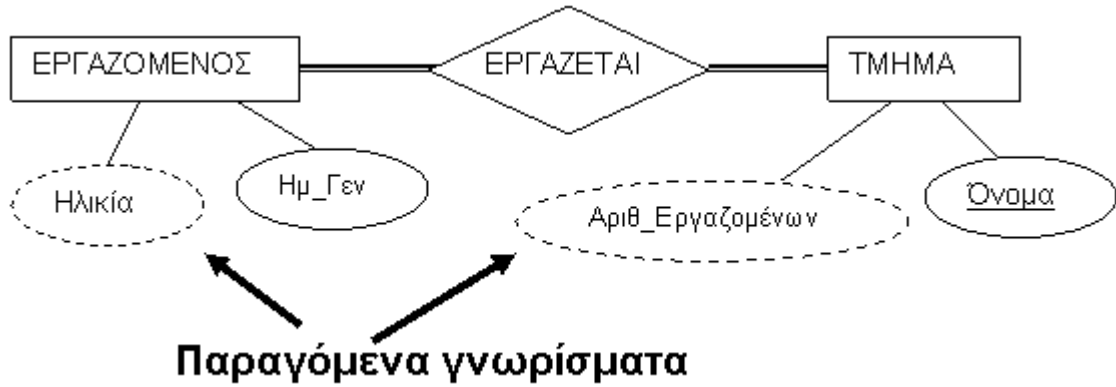


Εικόνα 2.2 . Σύνθετα και απλά γνωρίσματα.

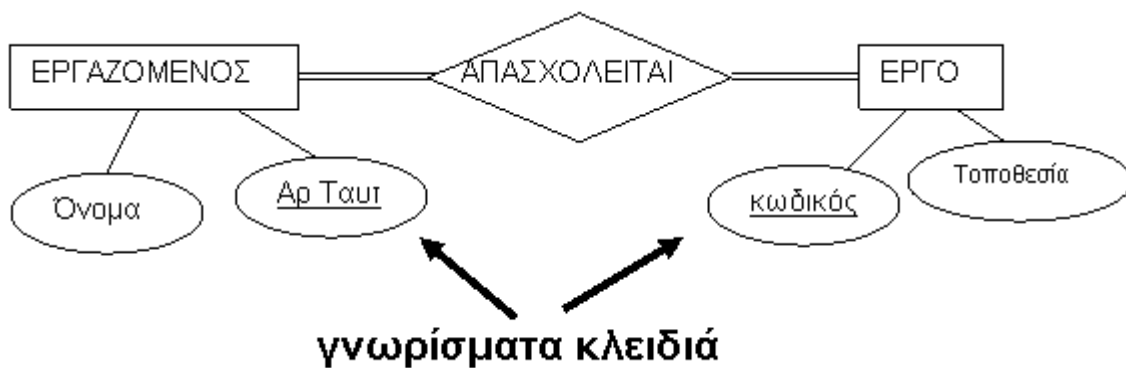
Μονότιμα έναντι Πλειότιμων Γνωρισμάτων. Τα περισσότερα γνωρίσματα έχουν μία και μόνο τιμή για μια συγκεκριμένη οντότητα: τέτοια γνωρίσματα λέγονται **μονότιμα** (single-valued). Για παράδειγμα, ο Αριθμός Ταυτότητας και η Ημερομηνία Γέννησης ενός ατόμου είναι μονότιμα γνωρίσματα. Σε μερικές περιπτώσεις ένα γνώρισμα μπορεί να έχει ένα σύνολο από τιμές για την ίδια οντότητα -για παράδειγμα το γνώρισμα Χρώματα για ένα αυτοκίνητο ή το γνώρισμα ΤΗΛΕΦΩΝΟ για έναν άνθρωπο. Τα μονόχρωμα αυτοκίνητα έχουν μία τιμή ενώ τα δίχρωμα αυτοκίνητα δύο τιμές για το γνώρισμα Χρώματα. Ομοίως, ένα άτομο μπορεί να μην έχει κανένα τηλέφωνο, ένα άλλο άτομο να έχει έναν αριθμό τηλεφώνου ενώ κάποιος άλλος να διαθέτει δύο ή περισσότερα τηλέφωνα όπως κινητό, τηλέφωνο οικίας, τηλέφωνο εργασίας, φαξ κλπ. Τέτοια γνωρίσματα ονομάζονται **πλειότιμα** (multivalued) και συμβολίζονται με δύο ομόκεντρες ελλείψεις.



Παραγόμενα Γνωρίσματα. Σε η τιμή ενός γνωρίσματος μπορεί να υπολογισθεί από την τιμή ενός άλλου γνωρίσματος. Π.χ Η τιμή του γνωρίσματος Ηλικία μπορεί να υπολογιστεί από την τρέχουσα ημερομηνία και την τιμή του Ημερομηνία Γέννησης ενός ατόμου. Επομένως το γνώρισμα Ηλικία λέγεται **παραγόμενο γνώρισμα** (derived attribute). Οι τιμές μερικών γνωρισμάτων μπορούν να υπολογιστούν από σχετιζόμενες οντότητες: για παράδειγμα, ένα γνώρισμα Αριθμος_Υπαλλήλων ενός τμήματος μπορεί να παραχθεί μετρώντας τον αριθμό των υπαλλήλων που σχετίζονται με (εργάζονται στο) τμήμα αυτό.



Γνωρίσματα-Κλειδιά ενός Τύπου Οντοτήτων. Όπως θα δούμε και παρακάτω ένας σημαντικός δομικός περιορισμός για τις οντότητες ενός τύπου οντοτήτων είναι ο **περιορισμός κλειδιού** (key constraint) ή **μοναδικότητας** (uniqueness constraint) για τα γνωρίσματα. Ένας τύπος οντοτήτων συνήθως έχει κάποιο γνώρισμα του οποίου οι τιμές είναι **μοναδικές** για κάθε ξεχωριστή οντότητα. Ένα τέτοιο γνώρισμα λέγεται **γνωρίσμα-κλειδί** (key attribute) και οι τιμές του μπορούν να χρησιμοποιηθούν για να προσδιορίζεται κάθε οντότητα μονοσήμαντα. Για τους τύπους οντοτήτων ΑΤΟΜΟ, ΕΡΓΑΖΟΜΕΝΟΣ ένα τυπικό γνώρισμα-κλειδί είναι ο αριθμός ταυτότητας.

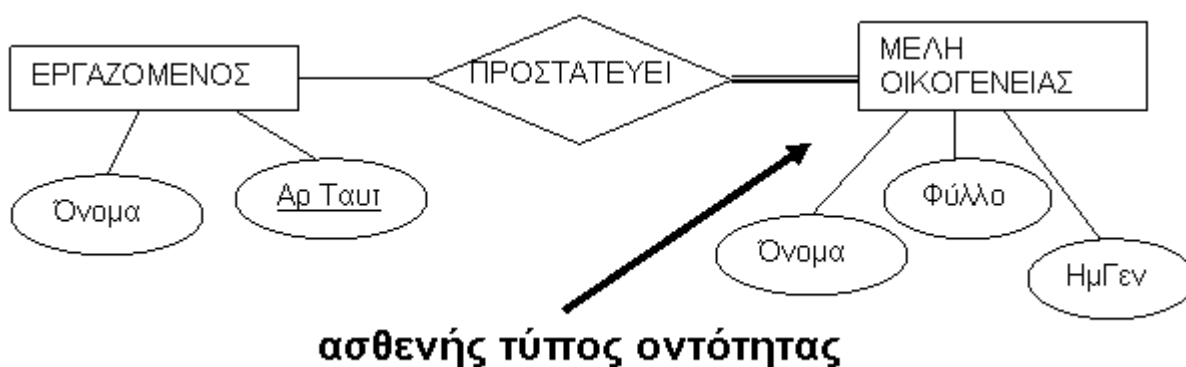


Μερικές φορές, διαφορετικά γνωρίσματα μαζί σχηματίζουν ένα κλειδί, με την έννοια ότι ο *συνδυασμός* των τιμών των γνωρισμάτων αυτών πρέπει να είναι διαφορετικός για κάθε ατομική οντότητα. Ένα σύνολο από γνωρίσματα που έχει αυτή την ιδιότητα μπορεί να ομαδοποιηθεί σε ένα *σύνθετο γνώρισμα*, το οποίο γίνεται γνώρισμα-κλειδί του τύπου οντοτήτων. Στο συμβολισμό των διαγραμμάτων ΟΣ, κάθε γνώρισμα-κλειδί έχει το όνομα του υπογραμμισμένο μέσα στην έλλειψη.

Μερικοί τύποι οντοτήτων έχουν *περισσότερα από ένα* γνωρίσματα-κλειδιά. Για παράδειγμα, στον τύπο οντοτήτων ΑΥΤΟΚΙΝΗΤΟ, τόσο ο Αριθμός_ Πλαισίου όσο και ο Αριθμός_Κυκλοφορίας είναι από μόνος του ο καθένας κλειδί.



Από την άλλη πλευρά ένας τύπος οντότητας μπορεί να μην έχει κανένα κλειδί και σ' αυτή την περίπτωση ονομάζεται **μη ισχυρός τύπος οντότητας (ασθενής τύπος)**. Για παράδειγμα σε μία εταιρεία θέλουμε για ασφαλιστικούς λόγους να τηρούμε στοιχεία για τα ΜΕΛΗ ΟΙΚΟΓΕΝΕΙΑΣ ενός ΕΡΓΑΖΟΜΕΝΟΥ, για να υπολογίζουμε τα επιδόματα μισθού. Είναι προφανές ότι αν δεν υπάρχει εργαζόμενος δεν υπάρχουν και μέλη οικογένειας (δεν είναι ανεξάρτητη οντότητα) γι' αυτό και λέγεται μη ισχυρός τύπος οντότητας.

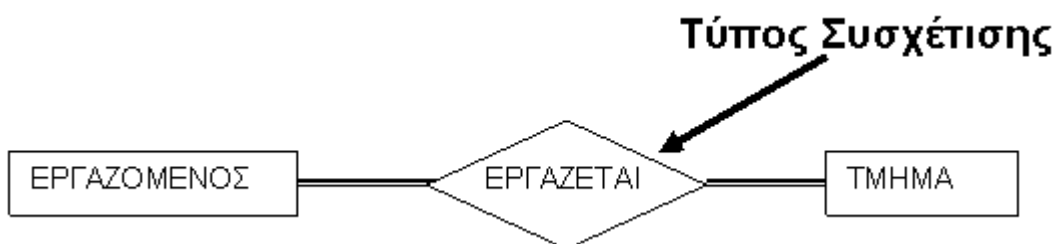


Σύνολα Τιμών (Πεδία ορισμού) των Γνωρισμάτων. Τα σύνολα τιμών δεν παρουσιάζονται στα διαγράμματα ΟΣ. Κάθε απλό γνώρισμα ενός τύπου οντοτήτων συνδέεται βέβαια με ένα **σύνολο τιμών** ή **πεδίο ορισμού** (domain), που προσδιορίζει τις επιτρεπτές τιμές που μπορεί να πάρει το γνώρισμα αυτό.

Null τιμές. Σε μερικές περιπτώσεις μια συγκεκριμένη οντότητα μπορεί να μην έχει καμιά δυνατή τιμή για ένα γνώρισμα. Για παράδειγμα, το γνώρισμα Αριθμος_Διαμερισματος μιας διεύθυνσης υφίσταται μόνον για διευθύνσεις κτιρίων με διαμερίσματα και όχι για άλλους τύπους οικιών όπως οι μονοκατοικίες. Παρόμοια, το γνώρισμα Τηλέφωνο πιθανόν να μην είναι γνωστό για κάποιο άτομο και να μην έχουμε προσωρινά την τιμή του τηλεφώνου του. Για τέτοιες περιπτώσεις δημιουργείται μια ειδική τιμή που λέγεται null. Η διεύθυνση μιας μονοκατοικίας θα είχε την τιμή null για το γνώρισμα Αριθμος_Διαμερισματος και ένα άτομο χωρίς τηλέφωνο θα είχε την τιμή null για το γνώρισμα τηλέφωνο.

Τύποι Συσχετίσεων

Ένας **τύπος συσχέτισης** (relationship type) μεταξύ δύο τύπων οντοτήτων ορίζει ένα σύνολο συνδέσεων -ή ένα **σύνολο συσχετίσεων**- μεταξύ αυτών των τύπων οντοτήτων. Στα διαγράμματα ΟΣ οι τύποι συσχετίσεων παρουσιάζονται ως ρόμβοι οι οποίοι συνδέονται με ευθείες γραμμές με τα παραλληλόγραμμα που παριστάνουν τους συμμετέχοντες τύπους οντοτήτων. Το όνομα μιας συσχέτισης γράφεται μέσα στον αντίστοιχο ρόμβο.

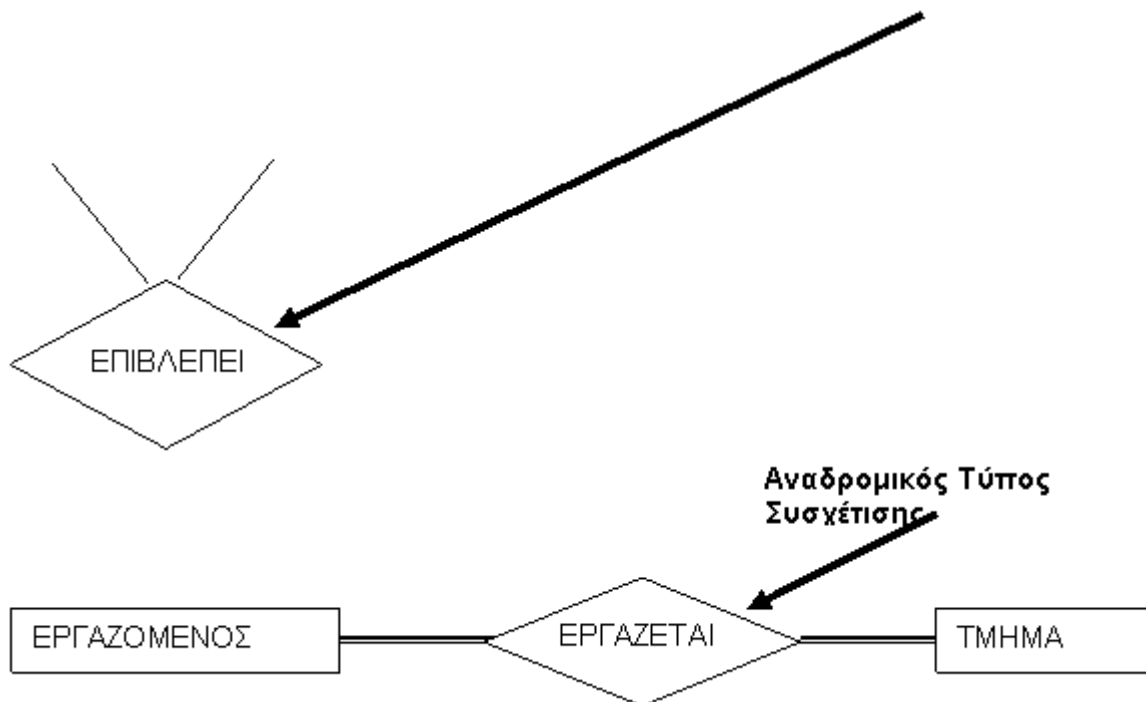


– Μεταξύ των ιδίων τύπων οντοτήτων πιθανόν να υπάρχουν περισσότεροι του ενός τύποι συσχετίσεων (multiple relationships).

π.χ., Εργαζομενος-Εργαζεται-Τμήμα, Εργαζόμενος-Διευθύνει-Τμήμα.

–Μια συσχέτιση μπορεί να συνδέει δύο οντότητες που ανήκουν στον ίδιο τύπο οντοτήτων (αποκαλείται, ένας **αναδρομικός τύπος συσχετίσεων** - recursive relationship type)

π.χ., ο ΕΠΙΒΛΕΠΕΙ τύπος συσχετίσεων συνδέει ΕΡΓΑΖΟΜΕΝΟ (στον ρόλο αυτού που επιβλέπεται) με κάποιο άλλο ΕΡΓΑΖΟΜΕΝΟ (στον ρόλο του επιβλέποντα)

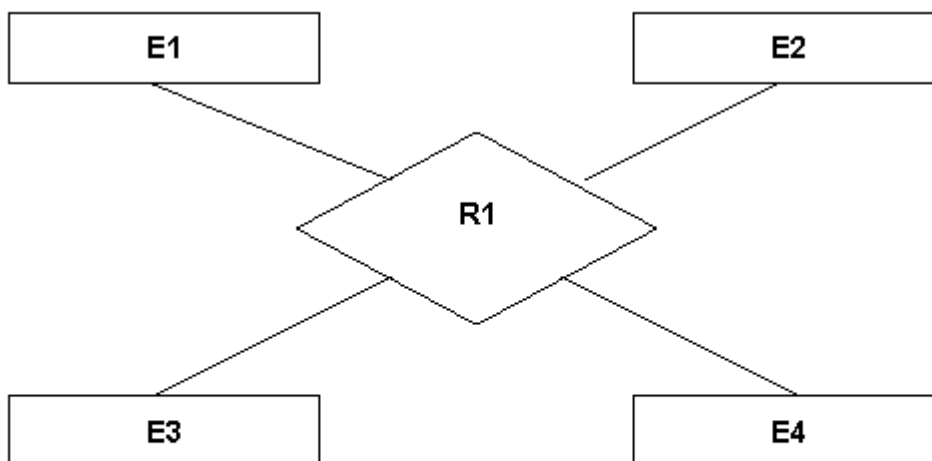


Δομικοί Περιορισμοί Στο Μοντέλο E-R.

Υπάρχουν δομικοί περιορισμοί (α) στους τύπους συσχετίσεων και (β) στα γνωρίσματα των τύπων οντοτήτων ή τύπων συσχετίσεων.

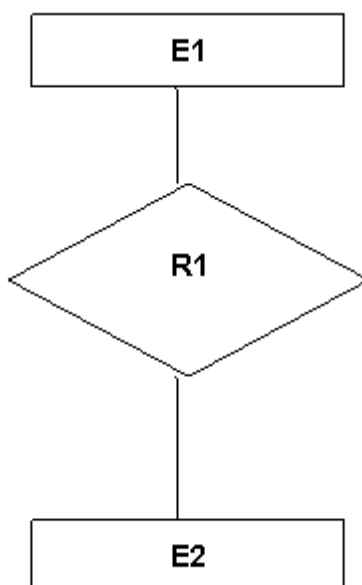
Βαθμός τύπου συσχετίσεων (relationship degree)

Βαθμός (degree) ενός τύπου συσχέτισης είναι το πλήθος των τύπων οντοτήτων που συμμετέχουν. Άρα υπάρχουν τύποι συσχετίσεων μεταξύ 2 τύπων οντοτήτων, μεταξύ 3 τύπων οντοτήτων, γενικά μεταξύ N τύπων οντοτήτων (N-ary relationship). Επομένως, ο τύπος συσχέτισης. Ένας τύπος συσχέτισης βαθμού δύο λέγεται **δυναδικός** (binary), ένας βαθμού τρία **τριαδικός** (ternary) και ένας βαθμού τέσσερα **τετραδικός**.



Εικόνα 2.3 : Τύπος συσχέτισης με βαθμό 4 όπου συμμετέχουν οι οντότητες E1..E4 στην συσχέτιση R1.

Οι συσχετίσεις μπορεί να είναι οποιοδήποτε βαθμού αλλά αυτές που εμφανίζονται πιο συχνά θα πρέπει να επιδιώκουμε είναι οι **δυναδικές συσχετίσεις**.



Εικόνα 2.4 : Δυναδική συσχέτιση μεταξύ των οντοτήτων E1 και E2 στην συσχέτιση R1.

Περιορισμός Συμμετοχής και Εξαρτήσεις Ύπαρξης.

Ο περιορισμός Συμμετοχής ορίζει αν η συμμετοχή μιας οντότητας στον τύπο συσχέτισης είναι ολική (total) ή μερική (partial). Με άλλα λόγια, αν όλες οι οντότητες του συγκεκριμένου τύπου οντοτήτων συμμετέχουν υποχρεωτικά σ' ένα τύπο συσχετίσεων ή όχι. Για παράδειγμα, όλοι οι υπάλληλοι του τύπου οντοτήτων ΕΡΓΑΖΟΜΕΝΟΣ εργάζονται υποχρεωτικά σ' ένα ΤΜΗΜΑ, άρα έχουν ολική συμμετοχή στον τύπο συσχετίσεων ΕΡΓΑΖΕΤΑΙ. Αντίθετα, δεν είναι όλοι διευθυντές τμημάτων,

οπότε η συμμετοχή του τύπου οντοτήτων ΕΡΓΑΖΟΜΕΝΟΣ στον τύπο συσχετίσεων ΔΙΕΥΘΥΝΕΙ είναι μερική. Η ολική συμμετοχή μερικές φορές λέγεται **εξάρτηση ύπαρξης** (existence dependency).

Η **ολική συμμετοχή** αναπαριστάται ως διπλή γραμμή που ενώνει το συμμετέχοντα τύπο οντοτήτων με την συσχέτιση, ενώ η **μερική συμμετοχή** ως απλή γραμμή.

Ένας εύκολος τρόπος για να διαπιστώσετε αν η συμμετοχή μιας οντότητας στην συσχέτιση είναι μερική ή ολική είναι η εξής:

Να ξεκινάτε την ερώτηση πάντα με τη λέξη ΟΛΕΣ οι οντότητες που μελετάτε συμμετέχουν στην συσχέτιση; Αν ΝΑΙ είναι ΟΛΙΚΗ συμμετοχή αν ΟΧΙ είναι ΜΕΡΙΚΗ συμμετοχή.

Λόγος πληθικότητας (cardinality)

Ο **Λόγος πληθικότητας** ενός τύπου συσχετίσεων ορίζει πόσες φορές μια οντότητα από έναν τύπο οντοτήτων μπορεί να εμπλακεί στο συγκεκριμένο τύπο συσχετίσεων.

Αποτελεί ένα από τα σημαντικότερα σημεία που πρέπει να δαθεί προσοχή κατά τον σχεδιασμό του διαγράμματος E-R, διότι ο λανθασμένος ορισμός του λόγου πληθικότητας μεταξύ δύο τύπων οντοτήτων θα έχει λανθασμένη συνέχεια κατά την μεταφορά του μοντέλου αυτού στο επόμενο βήμα που είναι το σχεσιακό μοντέλο.

Έχουμε τις εξής περιπτώσεις:

1:1 : μια οντότητα από τη μια πλευρά συνδέεται το πολύ με μια οντότητα από την άλλη πλευρά (συσχέτιση ένα-με-ένα)

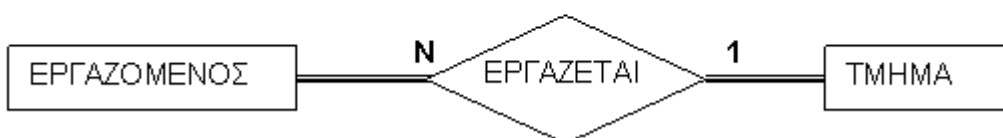
1:N : μια οντότητα από την πρώτη πλευρά μπορεί να συνδέεται με περισσότερες από μια οντότητες από τη δεύτερη πλευρά **N** ενώ μια οντότητα από τη δεύτερη πλευρά συνδέεται το πολύ με μια οντότητα από την πρώτη πλευρά **1** (συσχέτιση ένα-με-πολλά). Ακριβώς το ανάποδο ισχύει στο λόγο πληθικότητας **N:1**.

N:M : μια οντότητα από τη μια πλευρά μπορεί να συνδέεται με περισσότερες από μια οντότητες από την άλλη πλευρά. (συσχέτιση πολλά-με-πολλά)

Οι λόγοι πληθικότητας για τις δυαδικές σχέσεις παρουσιάζονται στα διαγράμματα ΟΣ με χρήση των συμβόλων 1, M και N στους ρόμβους

Όπως στο παρακάτω παράδειγμα ο δυαδικός τύπος συσχέτισης ΕΡΓΑΖΕΤΑΙ μεταξύ των ΤΜΗΜΑ και ΕΡΓΑΖΟΜΕΝΟΣ έχει λόγο πληθικότητας 1 :N (ένα-με-πολλά), που σημαίνει ότι κάθε τμήμα μπορεί να σχετίζεται με πολλούς εργαζόμενους, αλλά ένας εργαζόμενος μπορεί να σχετίζεται με (να εργάζεται σε) ένα μόνο τμήμα.

Λόγος πληθικότητας N:1



Παράδειγμα ενός 1:1 δυαδικού τύπου συσχέτισης είναι ο ΔΙΕΥΘΥΝΕΙ που σχετίζει μια οντότητα τμήματος με τον εργαζόμενο που διευθύνει το τμήμα αυτό. Ο αντίστοιχος περιορισμός λόγου πληθικότητας αναπαριστά τον περιορισμό του μικρόκοσμου ότι ένας εργαζόμενος μπορεί να διευθύνει μόνο ένα τμήμα και ένα τμήμα έχει έναν και μόνο διευθυντή.

Λόγος πληθικότητας 1:1



Λόγος πληθικότητας M:N



Ο τύπος συσχέτισης ΑΠΑΣΧΟΛΕΙΤΑΙ έχει λόγο πληθικότητας M:N, επειδή ένας εργαζόμενος μπορεί να εργάζεται σε διαφορετικά έργα και ένα έργο μπορεί να έχει πολλούς εργαζόμενους.

ΠΡΟΣΟΧΗ:

Ένας εύκολος τρόπος για να κατανοήσετε τον λόγο πληθικότητας σε μια δυαδική συσχέτιση είναι ο εξής, να ξεκινάτε πάντα με τις παρακάτω ερωτήσεις:

ΜΙΑ οντότητα από την δεξιά πλευρά συσχετίζεται με πολλές οντότητες από την αριστερή πλευρά της συσχέτισης;

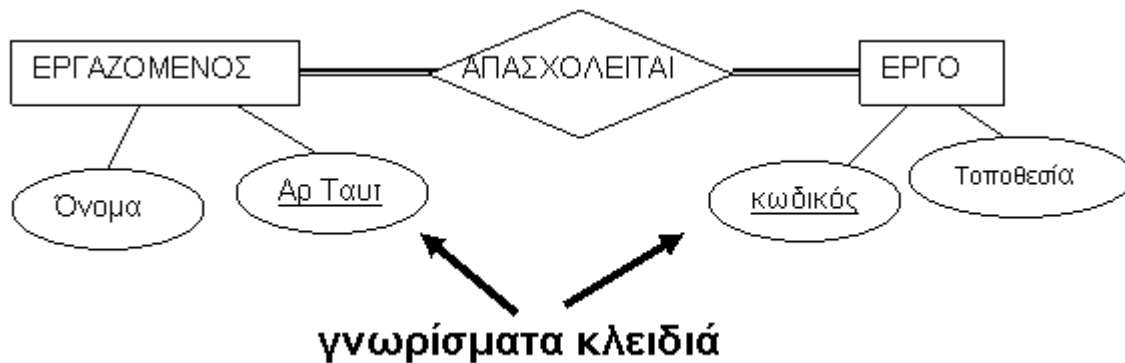
ΜΙΑ οντότητα από την αριστερή πλευρά συσχετίζεται με πολλές οντότητες από την δεξιά πλευρά της συσχέτισης;

- ✚ Αν η ερώτηση ισχύει μόνο από την μία πλευρά τότε ο λόγος πληθικότητας είναι 1:N από την πλευρά που ισχύει η ερώτηση.
- ✚ Αν η ερώτηση ισχύει και από τις δύο πλευρές τότε ο λόγος πληθικότητας είναι N:M .
- ✚ Αν η ερώτηση δεν ισχύει σε καμία από τις δύο πλευρές τότε ο λόγος πληθικότητας είναι 1:1

Δομικοί Περιορισμοί στα Γνωρίσματα

Βασικό χαρακτηριστικό ενός τύπου οντοτήτων ή συσχετίσεων είναι τα **γνωρίσματα κλειδιά** ή απλά κλειδιά (keys): Ένα γνώρισμα (ή σύνολο γνωρισμάτων) που χαρακτηρίζει μοναδικά κάθε οντότητα (αντίστοιχα, συσχέτιση) μέσα στον τύπο οντοτήτων (συσχετίσεων), δεν επιτρέπει δηλαδή δυο διαφορετικές οντότητες (συσχετίσεις) μέσα στον ίδιο τύπο να έχουν ίδια τιμή, ονομάζεται υπερ-κλειδί (super-key).

Κάθε γνώρισμα κλειδί θα φαίνεται υπογραμμισμένο στο Διάγραμμα Ο-Σ



Ορισμοί:

Ένα γνώρισμα (ή σύνολο γνωρισμάτων) ενός τύπου οντοτήτων/συσχετίσεων για το οποίο κάθε οντότητα/συσχέτιση στο σύνολο πρέπει να έχει μοναδική τιμή ονομάζεται **υπερ-κλειδί (super-key)**.

Ένα ελάχιστο υπερ-κλειδί (σύνολο γνωρισμάτων) που μπορεί να χαρακτηρίσει μοναδικά τις εγγραφές ονομάζεται **υποψήφιο κλειδί (candidate key)**.

Μεταξύ των κλειδιών επιλέγουμε και ορίζουμε ένα υποψήφιο κλειδί ως τον προσδιοριστή του τύπου οντοτήτων και το ονομάζουμε **πρωτεύον κλειδί (primary key)**

Οι τύποι συσχετίσεων μπορεί επίσης να έχουν γνωρίσματα παρόμοια με αυτά των τύπων οντοτήτων. Για παράδειγμα, για να καταγράψουμε τις εβδομαδιαίες ώρες απασχόλησης ενός εργαζομένου σ' ένα έργο μπορούμε να περιλάβουμε ένα γνώρισμα 'Ωρες' στον τύπο συσχέτισης ΑΠΑΣΧΟΛΕΙΤΑΙ.

Μη Ισχυροί Τύποι Οντοτήτων

Οι τύποι οντοτήτων που δεν έχουν γνωρίσματα-κλειδιά από μόνοι τους λέγονται **μη ισχυροί ή ασθενής τύποι οντοτήτων (weak entity types)**.

Σε αντίθεση, οι **κανονικοί** τύποι οντοτήτων που έχουν ένα γνώρισμα κλειδί ονομάζονται **ισχυροί τύποι οντοτήτων**. Οι οντότητες (π.χ. τα μέλη οικογένειας κάθε εργαζομένου) που ανήκουν σε κάποιον μη ισχυρό τύπο οντοτήτων προσδιορίζονται από τη σχέση τους με συγκεκριμένες οντότητες (π.χ. οι εργαζόμενοι) από έναν άλλο τύπο οντοτήτων, σε συνδυασμό με τις τιμές κάποιων γνωρισμάτων τους. Ονομάζουμε αυτόν τον άλλο τύπο οντοτήτων **προσδιορίζοντα ή ιδιοκτήτη τύπο οντότητας (owner entity type)** και τον τύπο συσχέτισης που συνδέει ένα μη ισχυρό τύπο οντοτήτων με τον ιδιοκτήτη του **προσδιορίζουσα συσχέτιση (identifying relationship)**.



Ένας μη ισχυρός τύπος οντοτήτων έχει πάντα περιορισμό ολικής συμμετοχής (εξάρτηση ύπαρξης) ως προς την προσδιορίζουσα του συσχετίση, διότι μια μη ισχυρή οντότητα δεν μπορεί να προσδιοριστεί χωρίς μια οντότητα-ιδιοκτήτη.

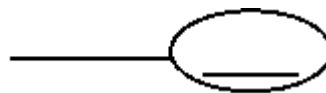
ΣΥΝΟΨΗ



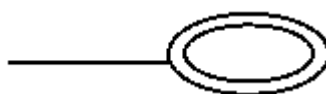
ΤΥΠΟΣ ΣΥΣΧΕΤΙΣΕΩΝ



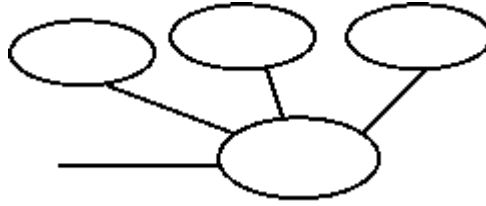
ΤΥΠΟΣ ΠΡΟΣΔΙΟΡΙΖΟΥΣΑΣ ΣΥΣΧΕΤΙΣΗΣ



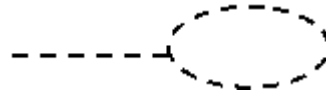
ΓΝΩΡΙΣΜΑ-ΚΛΕΙΔΙ



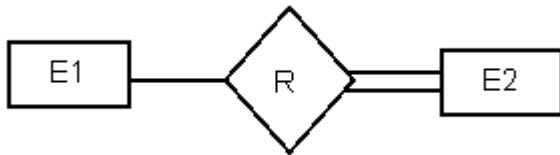
ΠΛΕΙΟΤΙΜΟ ΓΝΩΡΙΣΜΑ



ΣΥΝΘΕΤΟ ΓΝΩΡΙΣΜΑ



ΠΑΡΑΓΟΜΕΝΟ ΓΝΩΡΙΣΜΑ



ΟΛΙΚΗ ΣΥΜΜΕΤΟΧΗ ΤΗΣ E₂ ΣΤΗΝ R



ΛΟΓΟΣ ΠΛΗΘΙΚΟΤΗΤΑΣ 1:N

Εικόνα2.5 Σύνοψη του συμβολισμού των διαγραμμάτων Οντοτήτων-Συσχετίσεων

Σχεσιακό Μοντέλο

Τα λογικά μοντέλα βασισμένα σε εγγραφές χρησιμοποιούνται για να περιγράψουν τα δεδομένα από τον εννοιολογικό σχεδιασμό στον λογικό σχεδιασμό. Αντίθετα από τα εννοιολογικά μοντέλα, τα λογικά μοντέλα καθορίζουν τη γενική λογική δομή της Βάσης Δεδομένων και παρέχουν μία υψηλότερου επιπέδου περιγραφή της υλοποίησης της.

Ονομάζονται έτσι επειδή η Βάση Δεδομένων είναι δομημένη σε εγγραφές διάφορων τύπων. Κάθε τύπος εγγραφής καθορίζει έναν σταθερό αριθμό πεδίων (γνωρίσματα - ιδιότητες). Κάθε πεδίο είναι συνήθως σταθερού μήκους (αυτό απλοποιεί την εφαρμογή).

Στο σχεσιακό μοντέλο οι οντότητες και οι συσχετίσεις αντιπροσωπεύονται από μία συλλογή από κανονικοποιημένους πίνακες (σχέσεις). Κάθε κανονικοποιημένος πίνακας έχει διάφορες στήλες με μοναδικά ονόματα.

Οι σημαντικότεροι από τους όρους που χρησιμοποιούνται με τα “αντικείμενα του σχεσιακού μοντέλου” είναι η **σχέση (relation)** που αντιστοιχεί σε εκείνο που μέχρι τώρα ονομαζόταν πίνακας, μία **συστοιχία ή πλειάδα (tuple)** αντιστοιχεί σε μία γραμμή ενός τέτοιου πίνακα και ένα **γνώρισμα (attribute)** αντιστοιχεί σε μία στήλη-γνώρισμα. Το πλήθος των συστοιχιών ονομάζεται **πληθικότητα (cardinality)** και το πλήθος των γνωρισμάτων ονομάζεται **βαθμός (degree)**.

Το **πρωτεύον κλειδί (primary key)** είναι ένα μοναδικό αναγνωριστικό για τον πίνακα. Δηλαδή, μία στήλη ή συνδυασμός στηλών που έχει την ιδιότητα οι τιμές του να είναι μοναδικές.

Τέλος, ένα **πεδίο ορισμού (domain)** είναι μία δεξαμενή τιμών από την οποία τα συγκεκριμένα γνωρίσματα των συγκεκριμένων σχέσεων αντλούν τις συγκεκριμένες τιμές τους.

Μετατροπή Μοντέλου E–R Στο Σχεσιακό Μοντέλο

Από τη μία πλευρά, το μοντέλο ER διακρίνει τύπους οντοτήτων και τύπους συσχετίσεων (με τα γνωρίσματα τους) ενώ, από την άλλη, το Σχεσιακό μοντέλο υποστηρίζει **μία δομή μόνο, τις σχέσεις** (που έχουν και αυτές γνωρίσματα). Άρα η διαδικασία μετατροπής ενός διαγράμματος ER σε σχεσιακό σχήμα αφορά στη **μετατροπή των τύπων οντοτήτων και συσχετίσεων** (και των γνωρισμάτων τους) **σε σχέσεις (πίνακες)**.

Υπάρχει ο γενικός κανόνας ότι για κάθε τύπο οντοτήτων δημιουργείται μία σχέση που παίρνει το όνομα του αντίστοιχου τύπου. Πέρα από αυτόν το γενικό κανόνα, παρακάτω αναλύονται οι περιπτώσεις για τους τύπους οντοτήτων, τους τύπους συσχετίσεων και τα γνωρίσματα.

Μετατροπή Τύπων Οντοτήτων.

Για κάθε **ισχυρό τύπο οντοτήτων** δημιουργεί ο χρήστης μία σχέση R με τα ίδια γνωρίσματα, ένα για κάθε απλό γνώρισμα της οντότητας. Αν η οντότητα έχει σύνθετα γνωρίσματα, στη σχέση R υπάρχει ένα γνώρισμα για κάθε απλό γνώρισμα που απαρτίζει το σύνθετο.

Για κάθε **αδύνατο τύπο οντοτήτων** A που εξαρτάται από τον ισχυρό τύπο οντοτήτων B δημιουργείται ένα σχήμα σχέσης R με τα εξής γνωρίσματα

- (α) τα γνωρίσματα του A, και
- (β) τα γνωρίσματα του πρωτεύοντος κλειδιού του B.

Μετατροπή Τύπων Συσχετίσεων.

Με βάση το λόγο πληθικότητας, διακρίνονται τρεις περιπτώσεις:

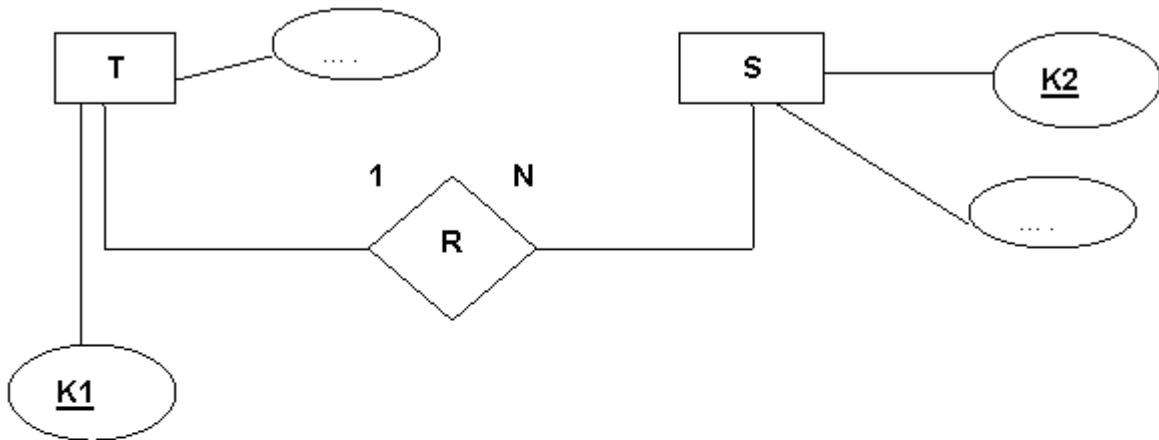
M:N πολλά προς πολλά συσχετίσεις,

1:1 ένα προς ένα δυαδικές συσχετίσεις και

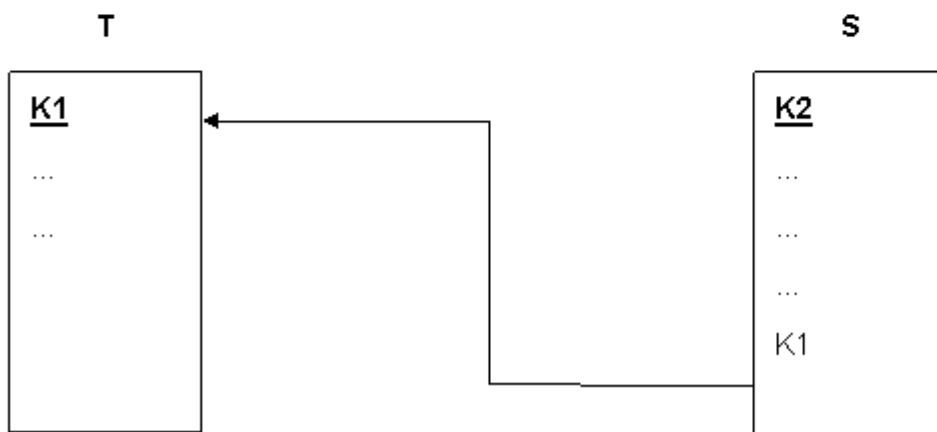
1:N ένα προς πολλά δυαδικές συσχετίσεις.

Για κάθε **1:N** δυαδική (μη αδύναμη) συσχέτιση R μεταξύ δύο τύπων οντοτήτων που αντιστοιχούν στις σχέσεις T και S, **δεν δημιουργούμε νέα σχέση R**. Στα ήδη υπάρχοντα γνωρίσματα της S προστίθενται: (α) τα γνωρίσματα της R, αν υπάρχουν, και (β) το πρωτεύον κλειδί της T, το οποίο συμμετέχει ως ξένο κλειδί στη σχέση S όπως φαίνεται στην Εικόνα 3.1.

E-R



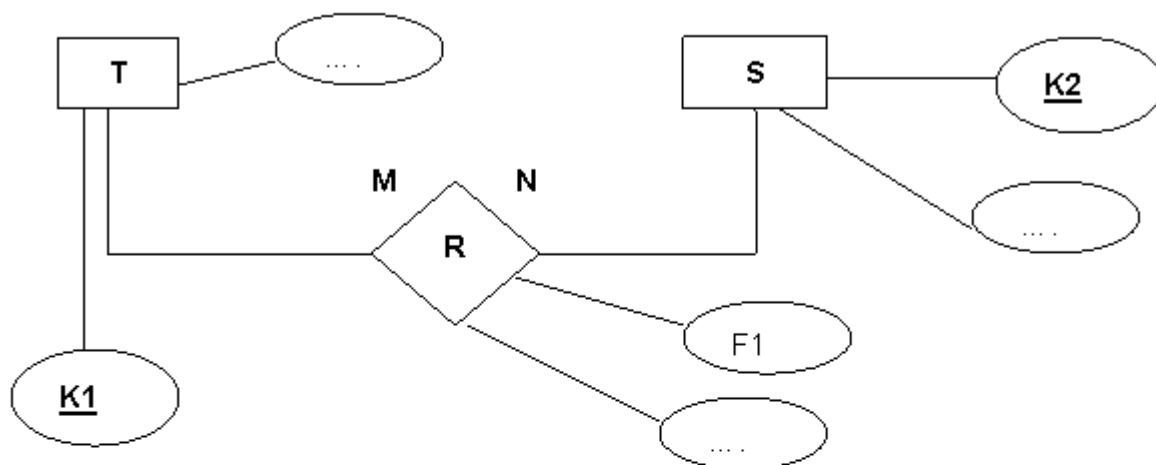
Μετατροπή E-R σε Σχεσιακό Μοντέλο



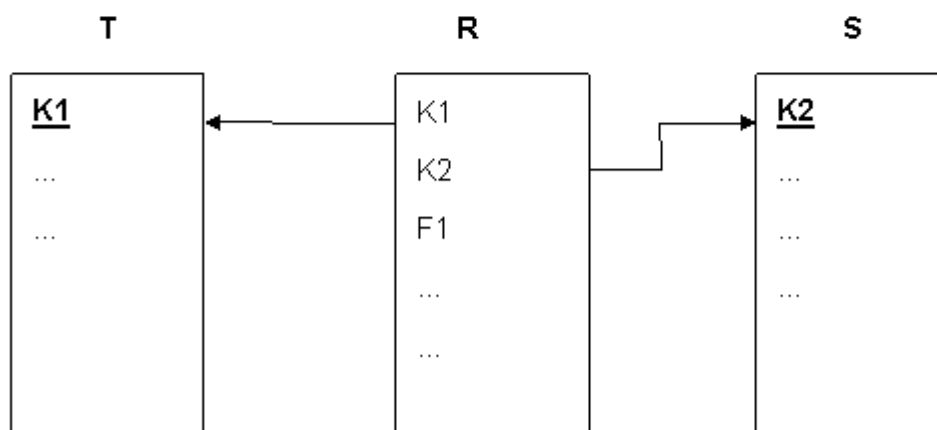
Εικόνα 3.1 Μετατροπή από E-R σε σχεσιακό για 1:N

Για κάθε $M:N$ συσχέτιση R μεταξύ τύπων οντοτήτων που αντιστοιχούν στις σχέσεις T, S , ο χρήστης δημιουργεί μία **νέα σχέση (πίνακα) R** με γνωρίσματα: (α) τα γνωρίσματα της R , αν υπάρχουν, και (β) τα γνωρίσματα του πρωτεύοντος κλειδιού κάθε συμμετέχουσας σχέσης S και T , τα οποία συμμετέχουν ως ξένα κλειδιά στη σχέση R όπως φαίνεται στην Εικόνα 3.2.

E-R



Μετατροπή E-R σε Σχεσιακό Μοντέλο



Εικόνα 3.2 Μετατροπή από E-R σε σχεσιακό για M:N

Για κάθε 1:1 δυαδική (μη αδύναμη) συσχέτιση R μεταξύ δύο τύπων οντοτήτων που αντιστοιχούν στις σχέσεις T και S, δεν δημιουργείται νέα σχέση R αλλά επιλέγεται μία εκ των T και S, έστω την S. Στα ήδη υπάρχοντα γνωρίσματα της S προσθέτουμε: (α) τα γνωρίσματα της R, αν υπάρχουν, και (β) το πρωτεύον κλειδί της T, το οποίο συμμετέχει ως ξένο κλειδί στη σχέση S όπως στην περίπτωση 1:N. Για την επιλογή T και S, κριτήριο είναι η ολική συμμετοχή του ενός ή του άλλου τύπου οντοτήτων στη συσχέτιση. Η σχέση που αντιστοιχεί στον τύπο οντοτήτων που συμμετέχει ολικά στη συσχέτιση, είναι αυτή (η σχέση S παραπάνω) που επιλέγεται να “φιλοξενήσει” ως ξένο κλειδί το πρωτεύον κλειδί της άλλης (S). Αν δε υπάρχει ολική συμμετοχή από μία πλευρά, τότε προσπαθούμε να μην έχουμε πολλές null τιμές, οπότε επιλέγουμε να προσθέτουμε ξένο κλειδί στη σχέση που αντιστοιχεί στον τύπο οντοτήτων με τη μεγαλύτερη συμμετοχή στη συσχέτιση.

Για τις 1:1, 1:N και N:1 δυαδικές (μη αδύναμες) μπορούμε, εναλλακτικά, να ακολουθήσουμε το γενικό κανόνα (δημιουργία νέας σχέσης με ξένα κλειδιά, τα πρωτεύοντα κλειδιά των εμπλεκομένων σχέσεων), ειδικά όταν δε υπάρχει ολική συμμετοχή από καμία πλευρά. Στην περίπτωση αυτή, κερδίζουμε ως προς την εννοιολογική καθαρότητα και πληρότητα αλλά χάνουμε ως προς την

αποδοτικότητα χώρου και τη επίδοση. Συνήθως δε επιλέγεται αυτός ο τρόπος αλλά η ειδική αντιμετώπιση τους, όπως παρουσιάστηκε παραπάνω.

Γνωρίσματα.

Η μετατροπή που γίνεται στα γνωρίσματα από μοντέλο οντοτήτων και συσχετίσεων στο σχεσιακό μοντέλο είναι η εξής

Ένα απλό γνώρισμα A γίνεται γνώρισμα της αντίστοιχης σχέσης,

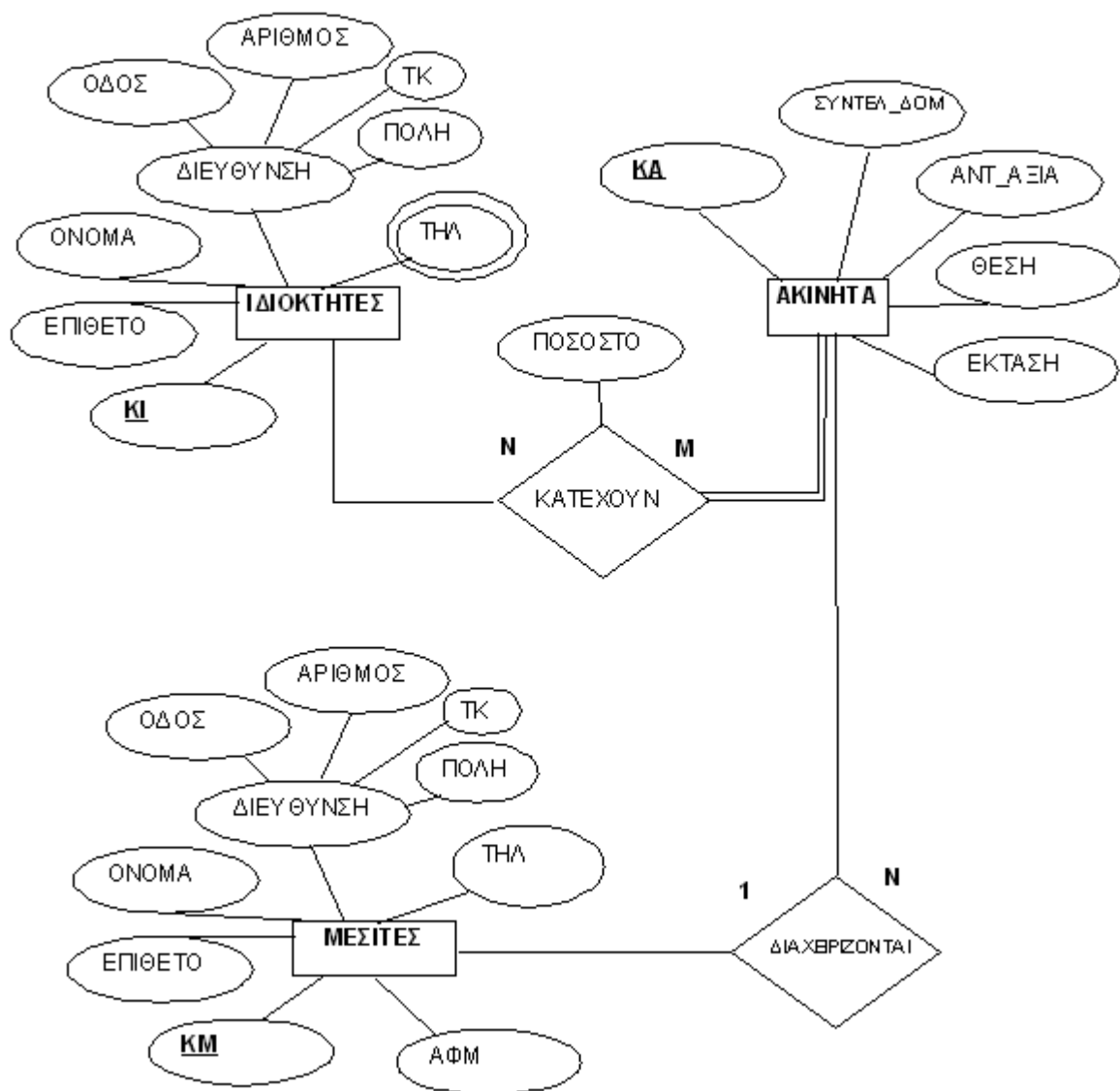
ένα σύνθετο γνώρισμα μετατρέπεται σε μία ομάδα γνωρισμάτων για την καινούρια σχέση που αποτελείται από τα επί μέρους απλά γνωρίσματα.

Στα πλειότιμα γνωρίσματα κατασκευάζεται μία σχέση R με γνωρίσματα το πρωτεύον κλειδί της αρχικής σχέσης καθώς και το γνώρισμα του πλειότιμου πεδίου

Στη συνέχεια παρουσιάζεται ένα παράδειγμα , η μετατροπής ενός διαγράμματος ER σε σχεσιακό σχήμα.

Παράδειγμα 3.1

Μετατρέψτε το παρακάτω διάγραμμα E-R σε σχεσιακό σχήμα. Το διάγραμμα αναφέρεται σε Ιδιοκτήτες που κατέχουν ακίνητα σε διαφορετικό ποσοστό τα οποία ακίνητα τα διαχειρίζονται μεσίτες.



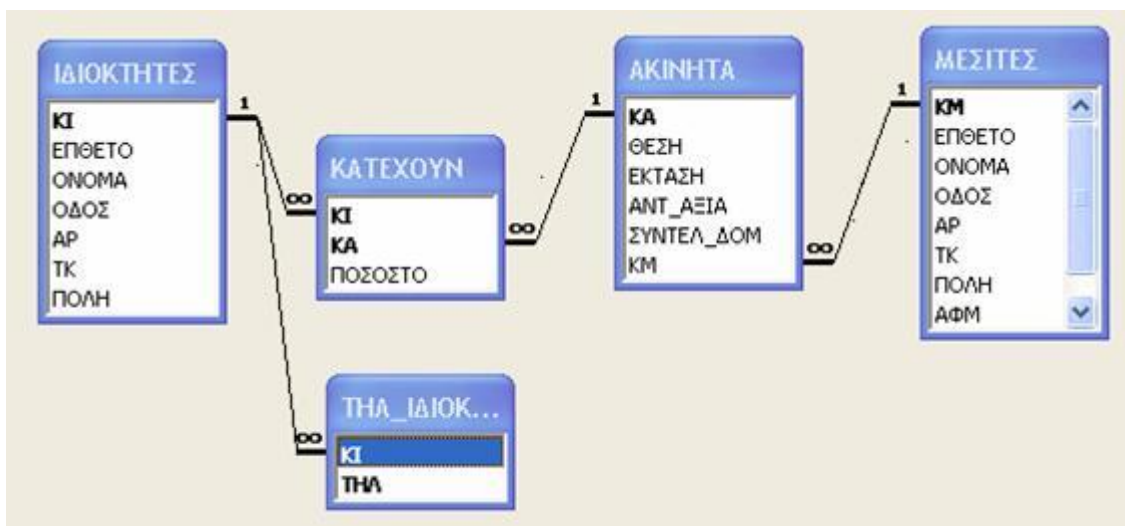
Λύση

Κάθε οντότητα θα γίνει πίνακας δηλ. σε πρώτο βήμα θα δημιουργηθούν οι πίνακες ΙΔΙΟΚΤΗΤΕΣ, ΑΚΙΝΗΤΑ, ΜΕΣΙΤΕΣ.

Στη συνέχεια οι συσχετίσεις που είναι N:M θα γίνουν νέοι πίνακες, δηλ. θα δημιουργηθεί ο πίνακας ΚΑΤΕΧΟΥΝ που έχει το ίδιο όνομα με την συσχέτιση.

Ακολουθεί η μετατροπή της συσχέτισης ΔΙΑΧΕΙΡΙΖΟΝΤΑΙ που είναι 1:N. Σε αυτή την περίπτωση το πρωτεύον κλειδί του πίνακα ΜΕΣΙΤΕΣ θα γίνει ξένο κλειδί στον πίνακα ΑΚΙΝΗΤΑ.

Τέλος για τα πλειότιμα γνωρίσματα θα δημιουργηθούν νέοι πίνακες όπως για το πεδίο ΤΗΛ ιδιοκτήτη θα γίνει νέος πίνακας με ξένο κλειδί το πρωτεύον κλειδί του πίνακα ΙΔΙΟΚΤΗΤΕΣ.



Οι ολικές συμμετοχές προαναγγέλλουν ότι τα ξένα κλειδιά που συμμετέχουν στη συσχέτιση θα παίρνουν υποχρεωτικά τιμές. Δηλ. αν δεν καταχωρηθεί τιμή σε αυτό το πεδίο δεν πρόκειται να προχωρήσει σε επόμενο πεδίο για καταχώριση τιμής. Τέτοια πεδία θα είναι το ΚΑ στον πίνακα ΚΑΤΕΧΟΥΝ.

Θεωρία Κανονικοποίησης

Είδαμε στα προηγούμενα κεφάλαια ότι το σχεσιακό μοντέλο αποτελείται από σχέσεις που δημιουργούνται από γνωρίσματα. Μία σχέση είναι ένα σύνολο από γνωρίσματα με τιμές για κάθε γνώρισμα τέτοιες ώστε να ισχύουν οι παρακάτω ιδιότητες

1. Κάθε όνομα γνωρίσματος είναι μοναδικό.
2. Όλες οι τιμές κάθε γνωρίσματος είναι ίδιου τύπου (ή πεδίου ορισμού).
3. Κάθε τιμή γνωρίσματος είναι ατομική (μία τιμή και όχι ομάδα πολλών τιμών).
4. Τα γνωρίσματα δεν έχουν διάταξη από τα αριστερά προς τα δεξιά..
5. Οι συστοιχίες (σειρές) δεν έχουν διάταξη από επάνω προς τα κάτω.
6. Δεν υπάρχουν δύο ίδιες σειρές (συστοιχίες) σε μία σχέση.

Από τη συζήτηση της μοντελοποίησης με το μοντέλο οντοτήτων-συσχετίσεων είδαμε ότι μία οντότητα τυπικά αντιστοιχεί σε μία σχέση και ότι τα γνωρίσματα της οντότητας γίνονται γνωρίσματα της σχέσης.

Η διαδικασία που ακολουθούμε είναι η ακόλουθη:

- 1 Συγκεντρώνουμε τις απαιτήσεις της επιχείρησης και των χρηστών.
- 2 Σχεδιάζουμε το μοντέλο οντοτήτων-συσχετίσεων βασισμένο στις παραπάνω απαιτήσεις
- 3 Μετατρέπουμε το διάγραμμα οντοτήτων-συσχετίσεων της επιχείρησης σε ένα σύνολο από σχέσεις (πίνακες) με το σχεσιακό μοντέλο.
- 4 Κανονικοποιούμε τις σχέσεις για να απομακρύνουμε τυχόν ανωμαλίες ενημέρωσης-διαγραφής-εισαγωγής στοιχείων.
- 5 Υλοποιούμε τη βάση δεδομένων δημιουργώντας ένα πίνακα για κάθε κανονικοποιημένη σχέση.

Προβλήματα Τροποποιήσεων

Μόλις το μοντέλο οντοτήτων-συσχετίσεων μετατραπεί σε σχέσεις μπορεί να ανακαλύψουμε ότι μερικές σχέσεις δεν καθορίζονται σωστά. Έτσι μπορεί να εμφανιστούν κάποια προβλήματα όπως:

Ανωμαλίες διαγραφής: διαγράφοντας μία σχέση κάποιες πληροφορίες που συνδέονται με μία άλλη σχέση χάνονται και αυτές.

Ανωμαλίες Εισαγωγής: Για να εισάγουμε μία καινούργια σχέση απαιτείται να έχουμε πληροφορίες από δύο ή περισσότερες οντότητες-αυτή η κατάσταση μπορεί να μην είναι εφαρμόσιμη.

Οι παραπάνω ανωμαλίες εισαγωγής ή διαγραφής σχέσεων προκύπτουν επειδή σε μία σχεσιακή βάση δεδομένων δεν μπορεί να υπάρχει σχέση που δεν σχετίζεται με καμία άλλη.

Συναρτησιακές Εξαρτήσεις

Έστω R μια σχέση και έστω ότι τα X και Y είναι τυχαία υποσύνολα του συνόλου των γνωρισμάτων της R . Τότε, λέμε ότι το Y είναι συναρτησιακά εξαρτημένο από το X , ή συμβολικά (διαβάζεται "το X καθορίζει συναρτησιακά το Y , ή απλώς $X \square Y$), εάν και μόνο εάν η κάθε τιμή X της σχέσης R αντιστοιχεί σε μία ακριβώς τιμή Y της R .

Μία συναρτησιακή εξάρτηση περιγράφει τη συσχέτιση ($X \square Y$) μεταξύ γνωρισμάτων της ίδιας σχέσης. Ένας δεύτερος ορισμός μπορεί να διατυπωθεί ως εξής:

Ένα γνώρισμα Y είναι συναρτησιακά εξαρτώμενο από ένα γνώρισμα X αν και μόνο αν κάθε τιμή του X σχετίζεται με μία και μόνο μία τιμή του Y . Δηλαδή ένα γνώρισμα είναι συναρτησιακά εξαρτώμενο από ένα άλλο αν μπορούμε να χρησιμοποιήσουμε την τιμή του ενός για να προσδιορίσουμε την τιμή του άλλου (να βρούμε την τιμή του μέσα στη σχέση).

Με άλλα λόγια, δύο οποιεσδήποτε συστοιχίες της R που συμφωνούν στην τιμή X , συμφωνούν επίσης στην τιμή Y .

Οι συναρτησιακές εξαρτήσεις διαθέτουν πλούσιο σύνολο από ενδιαφέρουσες τυπικές ιδιότητες. Με την χρήση αυτών των ιδιοτήτων γίνεται αντιμετώπιση των προβλημάτων με τυπικό και αυστηρό τρόπο.

Με τον όρο εξάρτηση, αποδίδονται δύο διαφορετικοί όροι, οι όροι dependence και dependency. Ο 1ος όρος θα έπρεπε να σημαίνει εξάρτηση και ο 2ος το εξαρτημένο αντικείμενο για μας και οι δύο αγγλικοί όροι θα σημαίνουν εξάρτηση.

Αν το X είναι υποψήφιο κλειδί της σχέσης R — και ειδικότερα, αν είναι το πρωτεύον κλειδί — τότε όλα τα γνωρίσματα Y της σχέσης R πρέπει κατ' ανάγκη να είναι συναρτησιακά εξαρτημένα από το X .

Μάλιστα, αν η σχέση R ικανοποιεί τη συναρτησιακή εξάρτηση $A \square B$ και το A δεν είναι υποψήφιο κλειδί, τότε η R θα έχει κάποιον πλεονασμό (redundancy).

Για παράδειγμα στη σχέση ΕΡΓΑΖΟΜΕΝΟΙ το Όνομα-Εργαζομένου είναι συναρτησιακά εξαρτώμενο από τον Αριθμό-Ταυτότητας επειδή ο Αριθμό-Ταυτότητας μπορεί να χρησιμοποιηθεί για να προσδιορισθεί μοναδικά η τιμή του Ονόματος του Εργαζομένου. Τότε ο Αριθμό-Ταυτότητας ονομάζεται **προσδιοριστικό** (ή ορίζουσα).

Εδώ θα χρησιμοποιήσουμε λοιπόν το σύμβολο \square για να δείχνουμε μερικές τέτοιες συναρτησιακές εξαρτήσεις (το σύμβολο \square διαβάζεται "προσδιορίζει συναρτησιακά")

ΚωδΦοιτητή \square Κατεύθυνση-σπουδών

ΚωδΦοιτητή, Μάθημα, Εξάμηνο \square Βαθμός

ΚωδΜαθήματος, Τμήμα \square Καθηγητής, Αίθουσα, Αριθμός-φοιτητών

Μοντέλο, Τρόπος-Πληρωμής, φόρος \square Τιμή αυτοκινήτου

Τα γνωρίσματα στα αριστερά ονομάζονται **προσδιοριστικά**. Αν γενικά $X \sqsubseteq A$ σημαίνει ότι το X προσδιορίζει το A .

ΜΕΤΑΒΑΤΙΚΗ ΣΥΝΑΡΤΗΣΙΑΚΗ ΕΞΑΡΤΗΣΗ

Υποθέστε ότι έχουμε μια σχέση R με τρία γνωρίσματα, A , B , και C , τέτοια ώστε οι συναρτησιακές εξαρτήσεις $A \sqsubseteq B$ και $B \sqsubseteq C$ να ισχύουν και οι δύο στην R .

Τότε ισχύει επίσης η συναρτησιακή εξάρτηση $A \sqsubseteq C$ στην R .

Εδώ, η συναρτησιακή εξάρτηση $A \sqsubseteq C$ είναι ένα παράδειγμα μεταβατικής συναρτησιακής εξάρτησης. Λέμε ότι η C εξαρτάται από την A μεταβατικά, μέσω της B .

Κλειδιά και μοναδικότητα

Όπως εξηγήσαμε και στο Κεφ.1 με τη λέξη **κλειδί** ονομάζουμε ένα ή περισσότερα γνωρίσματα που προσδιορίζουν μοναδικά μία συστοιχία (πλειάδα ή ολοκληρωμένη εγγραφή)

Η επιλογή των κλειδιών εξαρτάται πάντα από τη συγκεκριμένη εφαρμογή. Οι χρήστες της εφαρμογής μπορούν να προσφέρουν οδηγίες ως προς το ποια γνωρίσματα θα δημιουργήσουν ένα κατάλληλο κλειδί. Επειδή δεν είναι δυνατόν δύο σχέσεις να έχουν ακριβώς τις ίδιες τιμές, ένα υποψήφιο κλειδί μπορεί αν αποτελείται από όλα τα γνωρίσματα μίας σχέσης. Ένα κλειδί προσδιορίζει συναρτησιακά μία συστοιχία και δεν είναι όλα τα προσδιοριστικά να είναι κλειδιά.

ΚΑΝΟΝΕΣ ΣΥΝΑΓΩΓΗΣ –ΑΞΙΩΜΑΤΑ- ΤΟΥ ARMSTRONG

Έστω ότι A , B , και C είναι τυχαία υποσύνολα του συνόλου των γνωρισμάτων της δεδομένης σχέσης R , και έστω ο συμβολισμός AB σημαίνει την ένωση των A και B . Τότε:

Ανακλαστικότητα (reflexivity): αν το B είναι υποσύνολο του A , τότε $A \sqsubseteq B$.

Επαύξηση (augmentation): αν $A \sqsubseteq B$, τότε $AC \sqsubseteq BC$.

Μεταβατικότητα (transitivity): αν $A \sqsubseteq B$ και $B \sqsubseteq C$, τότε $A \sqsubseteq C$.

Μερικοί ακόμα κανόνες μπορούν να προκύψουν από τους τρεις προηγούμενους :

Αυτοκαθορισμός (self-determination): $A \sqsubseteq A$.

Ανάλυση (decomposition): αν $A \sqsubseteq BC$, τότε $A \sqsubseteq B$ και $A \sqsubseteq C$.

Ένωση (union): αν $A \sqsubseteq B$ και $A \sqsubseteq C$, τότε $A \sqsubseteq BC$.

Σύνθεση (composition): αν $A \sqsubseteq B$ και $C \sqsubseteq D$, τότε $AC \sqsubseteq BD$.

Κανονικοποίηση-Κανονικές Μορφές

Για τον καλύτερο σχεδιασμό μιας βάσης δεδομένων, οι πίνακες που απαρτίζουν μια βάση πρέπει να ελέγχονται σύμφωνα με κάποιους γενικούς κανόνες ώστε να υπάρχει μια ενιαία αντιμετώπιση στον σχεδιασμό για όλους τους προγραμματιστές. Για τον λόγο αυτό οι σχέσεις/ πίνακες μπορούν να υπάγονται σε μία ή περισσότερες κατηγορίες (ή κλάσεις) που ονομάζονται **Κανονικές Μορφές** (KM-normal forms-NF).

Κανονική Μορφή: Ονομάζεται μία κλάση σχέσεων απαλλαγμένων από συγκεκριμένα προβλήματα τροποποιήσεων.

Οι κανονικές Μορφές που θα εξηγήσουμε παρακάτω έχουν ονόματα όπως:

- Πρώτη κανονική μορφή (1NF-1KM)
- Δεύτερη κανονική μορφή (2NF-2KM)
- Τρίτη κανονική μορφή (3NF-3KM)
- Boyce-Codd κανονική μορφή (BC NF-KM)
- Τέταρτη κανονική μορφή (4NF-4KM)
- Πέμπτη κανονική μορφή (5NF-5KM)
- Κανονική μορφή πεδίου ορισμού- κλειδιού (Domain-Key/NF)

Αυτές οι κανονικές μορφές είναι αθροιστικές. Μία σχέση που βρίσκεται σε Τρίτη κανονική μορφή είναι επίσης και σε δεύτερη και σε πρώτη.

Οι τρεις πρώτες κανονικές μορφές (1KM, 2KM, 3KM) ορίστηκαν από τον Codd. Όλες οι κανονικοποιημένες σχέσεις είναι σε 1KM. Με άλλα λόγια, "κανονικοποιημένη" και "σε 1KM" σημαίνει ακριβώς το ίδιο πράγμα. Μερικές σχέσεις 1KM είναι επίσης σε 2KM, και μερικές σχέσεις 2KM είναι επίσης σε 3KM. Το σκεπτικό των ορισμών του Codd ήταν ότι η 2KM είναι "πιο επιθυμητή" από την 1KM, και ότι η 3KM είναι με τη σειρά της πιο επιθυμητή από την 2KM. Αυτό σημαίνει ότι σχεδιαστής της βάσης δεδομένων θα πρέπει γενικά να στοχεύει σε μια σχεδίαση που να έχει σχέσης σε 3KM, και όχι απλώς σε 2KM ή σε 1KM.

Ο Fagin όρισε την τέταρτη κανονική μορφή. Μετέπειτα και πάλι ο Fagin όρισε άλλη μία κανονική μορφή, την κανονική μορφή προβολής-σύζευξης (projection join), που αργότερα έγινε γνωστή και ως πέμπτη κανονική μορφή (5KM). Όπως φαίνεται μερικές σχέσεις που είναι σε KM-BC είναι επίσης σε 4KM, και μερικές σχέσεις που είναι σε 4KM είναι επίσης σε 5KM.

Πρώτη κανονική μορφή (1NF-1KM)

Η Πρώτη κανονική μορφή θεωρείται γενικώς να είναι μέρος του ορισμού μιας σχέσης.

Μία σχέση βρίσκεται σε πρώτη κανονική μορφή αν ικανοποιεί όλες τις ιδιότητες του ορισμού της σχέσης:

- ✚ Κάθε όνομα γνωρίσματος (στήλης) είναι μοναδικό.
- ✚ Όλες οι τιμές κάθε γνωρίσματος είναι ίδιου τύπου (ή πεδίου ορισμού).
- ✚ Κάθε τιμή γνωρίσματος είναι ατομική (μία τιμή και όχι ομάδα πολλών τιμών).
- ✚ Τα γνωρίσματα δεν έχουν διάταξη από τα αριστερά προς τα δεξιά..
- ✚ Οι συστοιχίες (σειρές) δεν έχουν διάταξη από επάνω προς τα κάτω.
- ✚ Δεν υπάρχουν δύο ίδιες σειρές (συστοιχίες) σε μία σχέση.

Εάν υπάρχει καθορισμένο **κλειδί** για τη σχέση τότε ικανοποιείται η απαίτηση της μοναδικότητας των συστοιχιών (σειρών).

Ένας πίνακας σε πρώτη κανονική μορφή λέγεται κανονικοποιημένος πίνακας και τότε και μόνο τότε αντιστοιχεί σε μία σχέση (οι σχέσεις του σχεσιακού μοντέλου είναι στην 1η κανονική μορφή). Μια γενικότερη συμβουλή για να καταλάβει κάποιος πότε ένας πίνακας δεν βρίσκεται σε πρώτη κανονική μορφή, θα πρέπει να προσέξει ώστε να μην υπάρχουν επαναλαμβανόμενα πεδία.

Τα βήματα που πρέπει να ακολουθηθούν ώστε να επέλθει ένας πίνακας με επαναλαμβανόμενα πεδία σε 1ΚΜ, είναι το σπάσιμο του αρχικού πίνακα σε δύο επί μέρους πίνακες. Ο ένας θα περιλαμβάνει τα βασικά στοιχεία του πίνακα και ο δεύτερος τα ονόματα των πεδίων που επαναλαμβάνονται μαζί με το πρωτεύον κλειδί του πρώτου πίνακα.

Παράδειγμα 4.1

Ο παρακάτω πίνακας βρίσκεται σε 1ΚΜ;

ΣΠΟΥΔΑΣΤΕΣ (ΑΜ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΤΗΛ, ΟΔΟΣ, ΑΡ, ΤΚ, ΠΟΛΗ, ΚΩΔ_ΜΑΘ1, ΜΑΘΗΜΑ1, ΒΙΒΛΙΟ1, ΒΑΘΜΟΣ1..., ΚΩΔ_ΜΑΘn, ΜΑΘΗΜΑn, ΒΙΒΛΙΟn ΒΑΘΜΟΣn)

Τι θα κάνανε ώστε να επέλθει σε 1ΚΜ;

Λύση

Ο πίνακας δεν είναι σε 1ΚΜ διότι έχει μια ομάδα επαναλαμβανόμενων πεδίων και έρχεται σε αντίθεση με τον πρώτο όρο που πρέπει να έχει ένας πίνακας για να είναι σε 1ΚΜ που λέει ότι το όνομα κάθε γνωρίσματος είναι μοναδικό.

Τα βήματα που θα ακολουθηθούν για να επέλθει σε 1ΚΜ είναι να τον σπάσουμε σε δύο επί μέρους πίνακες ως εξής :

ΣΠΟΥΔΑΣΤΕΣ (ΑΜ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΤΗΛ, ΟΔΟΣ, ΑΡ, ΤΚ, ΠΟΛΗ)

ΜΑΘΗΜΑΤΑ_ΒΑΘΜΟΙ (ΚΩΔ_ΜΑΘ, ΑΜ, ΜΑΘΗΜΑ, ΒΙΒΛΙΟ ΒΑΘΜΟΣ)

Δεύτερη κανονική μορφή (2NF-2ΚΜ)

Μία σχέση βρίσκεται σε δεύτερη κανονική μορφή εάν κάθε ένα από τα γνωρίσματά της που δεν είναι κλειδιά προσδιορίζονται από ολόκληρο το πρωτεύον κλειδί και όχι μόνο από ένα τμήμα του.

Οι σχέσεις που έχουν μόνο ένα **γνωρίσμα σαν πρωτεύον κλειδί** βρίσκονται αυτόματα και στη **δεύτερη κανονική μορφή**.

Αυτός είναι ένας λόγος για τον οποίο χρησιμοποιούμε συχνά τεχνητά αναγνωριστικά σαν κλειδιά.

Η λύση που ακολουθείται όπως και σε όλες τις περιπτώσεις που ένας πίνακας δεν είναι κανονικοποιημένος, είναι το σπάσιμο του πίνακα σε επιμέρους πίνακες.

Στον έναν πίνακα περιλαμβάνονται τα πεδία του κλειδιού και τα πεδία που εξαρτώνται και περιγράφουν όλο το σύνθετο κλειδί.

Στον άλλο πίνακα μπαίνουν τα πεδία που εξαρτώνται και περιγράφουν μέρος από το σύνθετο κλειδί καθώς και το μέρος του σύνθετου κλειδιού.

Παράδειγμα 4.2

Ελέγξτε τους πίνακες που προέκυψαν από το Παράδειγμα 1. αν είναι σε 2ΚΜ.

ΣΠΟΥΔΑΣΤΕΣ (ΑΜ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΤΗΛ, ΟΔΟΣ, ΑΡ, ΤΚ, ΠΟΛΗ)

ΜΑΘΗΜΑΤΑ_ΒΑΘΜΟΙ (ΚΩΔ_ΜΑΘ, ΑΜ, ΜΑΘΗΜΑ, ΒΙΒΛΙΟ, ΒΑΘΜΟΣ)

Λύση

Κατ' αρχήν εξετάζουμε το πρώτο πίνακα ΣΠΟΥΔΑΣΤΕΣ. Αυτός ο πίνακας είναι σε 2ΚΜ γιατί έχει ένα πεδίο ως πρωτεύον κλειδί. Οπότε όλα τα υπόλοιπα πεδία που δεν είναι κλειδιά προσδιορίζονται από ολόκληρο το πρωτεύον κλειδί.

Στον δεύτερο πίνακα όμως ΜΑΘΗΜΑΤΑ_ΒΑΘΜΟΙ παρατηρούμε ότι το πρωτεύον κλειδί είναι σύνθετο και αποτελείται από δύο πεδία τον Κωδικό μαθήματος και τον Αριθμό μητρώου του σπουδαστή (ΚΩΔ_ΜΑΘ, ΑΜ). Οι συναρτησιακές εξαρτήσεις που προκύπτουν είναι οι παρακάτω:

ΚΩΔ_ΜΑΘ, ΑΜ \square ΒΑΘΜΟΣ

ΚΩΔ_ΜΑΘ \square ΜΑΘΗΜΑ

ΚΩΔ_ΜΑΘ \square ΒΙΒΛΙΟ

Παρατηρούμε ότι οι δύο τελευταίες συναρτησιακές εξαρτήσεις παραβιάζουν τον κανόνα για τη δεύτερη κανονική μορφή.

Το επόμενο βήμα είναι ο διαχωρισμός του πίνακα ΜΑΘΗΜΑΤΑ_ΒΑΘΜΟΙ σε δύο πίνακες:

Οπότε προκύπτουν οι παρακάτω σχέσεις.

ΣΠΟΥΔΑΣΤΕΣ (ΑΜ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΤΗΛ, ΟΔΟΣ, ΑΡ, ΤΚ, ΠΟΛΗ)

ΜΑΘΗΜΑΤΑ_ΒΑΘΜΟΙ (ΚΩΔ_ΜΑΘ, ΑΜ, ΒΑΘΜΟΣ)

ΜΑΘΗΜΑΤΑ (ΚΩΔ_ΜΑΘ, ΜΑΘΗΜΑ, ΒΙΒΛΙΟ)

Τρίτη κανονική μορφή (3NF-3ΚΜ)

Μία σχέση βρίσκεται σε Τρίτη κανονική μορφή εάν είναι σε δεύτερη και δεν περιέχει **μεταβατικές εξαρτήσεις**.

Θεωρήστε για παράδειγμα τη σχέση R που έχει γνωρίσματα τα A, B και Γ. Εάν $A \square B$ και $B \square \Gamma$ τότε θα ισχύει και $A \square \Gamma$. Αν το πεδίο A προσδιορίζει συναρτησιακά το πεδίο B. Το πεδίο B προσδιορίζει συναρτησιακά το πεδίο Γ. Τότε το πεδίο A προσδιορίζει συναρτησιακά και το πεδίο Γ. Δεν θέλουμε τέτοιες εξαρτήσεις σε μία σχέση.

Για να λύσουμε το πρόβλημα σε περίπτωση που υπάρχει μεταβατική συναρτησιακή εξάρτηση σε έναν πίνακα σπάμε τον πίνακα σε δύο άλλους πίνακες. Ο ένας πίνακας περιέχει τα σταθερά στοιχεία και ο άλλος περιλαμβάνει τα πεδία με την συναρτησιακές εξαρτήσεις συν το πεδίο κλειδί του αρχικού πίνακα.

Παράδειγμα 4.3

Ο παρακάτω πίνακας είναι κανονικοποιημένος ως προς την 3ΚΜ; Εξηγήστε αν υπάρχουν μεταβατικές εξαρτήσεις και σε ποια πεδία. Δώστε λύση ώστε να εξαλειφτεί το πρόβλημα.

ΣΠΟΥΔΑΣΤΕΣ (ΑΜ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΤΗΛ, ΕΠΙΘΕΤΟ_ΚΗΔΕΜΟΝΑ ΟΝ_ ΚΗΔΕΜΟΝΑ, ΕΠΑΓΓΕΛΜΑ_ ΚΗΔΕΜΟΝΑ, ΤΗΛ_ ΚΗΔΕΜΟΝΑ, ΟΔΟΣ, ΑΡ, ΤΚ, ΠΟΛΗ)

Λύση:

Ο πίνακας ΣΠΟΥΔΑΣΤΕΣ δεν είναι κανονικοποιημένος ως προς 3ΚΜ διότι περιέχει μεταβατικές εξαρτήσεις στα παρακάτω πεδία:

ΑΜ \square ΕΠΙΘΕΤΟ_ΚΗΔΕΜΟΝΑ, ΟΝ_ΚΗΔΕΜΟΝΑ

ΕΠΙΘΕΤΟ_ΚΗΔΕΜΟΝΑ, ΟΝ_ΚΗΔΕΜΟΝΑ \square ΕΠΑΓΓΕΛΜΑ_ΚΗΔΕΜΟΝΑ

ΕΠΙΘΕΤΟ_ΚΗΔΕΜΟΝΑ, ΟΝ_ΚΗΔΕΜΟΝΑ \square ΤΗΛ_ΚΗΔΕΜΟΝΑ

AM □ ΕΠΑΓΓΕΛΜΑ_ ΚΗΔΕΜΟΝΑ

AM □ ΤΗΛ_ ΚΗΔΕΜΟΝΑ

Για να εξαληφθεί η μεταβατική συναρτησιακή εξάρτηση των παραπάνω πεδίων διαιρούμε τον πίνακα σε δύο επι μέρους πίνακες.

ΣΠΟΥΔΑΣΤΕΣ (AM, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΤΗΛ, ΕΠΙΘΕΤΟ_ΚΗΔΕΜΟΝΑ ΟΝ_ΚΗΔΕΜΟΝΑ, ΟΔΟΣ, ΑΡ, ΤΚ, ΠΟΛΗ)

ΚΗΔΕΜΟΝΕΣ (ΕΠΙΘΕΤΟ ΚΗΔΕΜΟΝΑ, ΟΝ ΚΗΔΕΜΟΝΑ, ΕΠΑΓΓΕΛΜΑ_ ΚΗΔΕΜΟΝΑ, ΤΗΛ_ ΚΗΔΕΜΟΝΑ , AM,)

Το πρωτεύον κλειδί του αρχικού πίνακα γίνεται ξένο κλειδί στον νέο πίνακα που προκύπτει από τον διαμελισμό του αρχικού πίνακα.

Boyce-Codd κανονική μορφή (BCNF)

Η Boyce-Codd Κανονική Μορφή είναι μια πιο περιοριστική μορφή της 3KM, με την έννοια ότι κάθε σχέση στην **BCNF** είναι επίσης στην 3KM, αλλά δεν ισχύει πάντα το αντίστροφο.

Η 3KM δεν αντιμετωπίζει τις παρακάτω περιπτώσεις:

1. Μία σχέση να έχει δύο ή περισσότερα υποψήφια κλειδιά
2. Τα δύο υποψήφια κλειδιά να είναι σύνθετα
3. Να επικαλύπτονται (να έχουν τουλάχιστον ένα γνώρισμα κοινό)

Μία σχέση βρίσκεται σε κανονική μορφή Boyce-Codd εάν κάθε προσδιοριστικό της σχέσης είναι ένα υποψήφιο κλειδί. Δηλαδή τα μόνα βέλη εξαρτήσεων που εμφανίζονται στο διάγραμμα εξαρτήσεων είναι βέλη που ξεκινούν από υποψήφια κλειδιά.

Θυμηθείτε ότι δεν είναι κλειδιά όλα τα προσδιοριστικά. Αυτά τα προσδιοριστικά που μπορούν να είναι κλειδιά τα καλούμε υποψήφια κλειδιά. Τελικά επιλέγουμε ένα υποψήφιο κλειδί για το πρωτεύον κλειδί μιας σχέσης. Στην πραγματικότητα, οι περισσότερες σχέσεις οι οποίες είναι στην 3NF είναι επίσης στην BCNF. Οποιαδήποτε σχέση που έχει μόνο δύο γνώρισμα είναι σεBCNF.

Τα βήματα που ακολουθούνται ώστε να επιτευχθεί κανονικοποίηση μία σχέση ως προς KM BC είναι τα ακόλουθα:

- 1 Δημιουργία μίας λίστας με όλα τα γνώρισμα
- 2 Έλεγχος εάν κάθε προσδιοριστικό γνώρισμα μπορεί να είναι υποψήφιο κλειδί
- 3 Για τα προσδιοριστικά γνώρισμα που δεν είναι κλειδιά δημιουργείται μία σχέση για την κάθε συναρτησιακή εξάρτηση.
- 4 Γίνεται συσχέτιση των γνωρισμάτων που δεν είναι κλειδιά με την αρχική σχέση

Παράδειγμα 4.4

Στο παρακάτω παράδειγμα η σχέση βρίσκεται σε Τρίτη κανονική μορφή αλλά όχι σε κανονική μορφή Boyce-Codd. Θεωρείστε ότι ισχύουν οι εξής περιορισμούς.

- 1 Οι Χρηματιστηριακές Συναλλαγές γίνονται σε πολλούς Τύπους Μετοχών
- 2 Οι Συναλλαγές διαχειρίζονται από έναν ή περισσότερους Χρηματιστές
- 3 Οι Τύποι Μετοχών μπορούν έχουν έναν ή πολλούς Χρηματιστές
- 4 Οι Χρηματιστές μπορούν να συναλλάσσονται σε έναν Τύπο Μετοχών

ΠΙΝΑΚΑΣ_Α

:

ΚωδΣυναλλαγής	ΤύποςΜετοχής	Χρηματιστής
15-03	ΚοινέςΜετοχές	Παπάς
15-03	Ομόλογα Δημοσίου	Γιώτης
3-103	ΚοινέςΜετοχές	Δημητρίου
2-234	ΠρονομακέςΜετοχές	Δήμου
56-117	ΚοινέςΜετοχές	Παπάς

Οι συναρτησιακές εξαρτήσεις που προκύπτουν σύμφωνα με τα παραπάνω είναι:

ΚωδΣυναλλαγής, ΤύποςΜετοχής □ *Χρηματιστής*

ΚωδΣυναλλαγής, Χρηματιστής □ *ΤύποςΜετοχής*

Χρηματιστής □ *ΤύποςΜετοχής*

Τι θα συνέβαινε εάν διαγραφόταν η εγγραφή με ΚωδΣυναλλαγής 2-234.

Κανονικοποιήστε τον παραπάνω πίνακα ως προς KM-BC.

Λύση

Εάν διαγραφόταν η εγγραφή με ΚωδΣυναλλαγής 2-234 θα χανόταν και το γεγονός ότι ο Δήμος διαχειρίζεται τις Προνομακές Μετοχές.

Ο συνδυασμός ΚωδΣυναλλαγής, και ΤύποςΜετοχής συνθέτουν ένα υποψήφιο κλειδί γιατί χρησιμοποιώντας τον συνδυασμό αυτό μπορούμε να αναγνωρίσουμε με μοναδικό τρόπο μια πλειάδα μέσα στη σχέση.

Όμοια ο συνδυασμός ΚωδΣυναλλαγής, και Χρηματιστής συνθέτουν ένα υποψήφιο κλειδί για τι χρησιμοποιώντας τον συνδυασμό αυτό μπορούμε να αναγνωρίσουμε με μοναδικό τρόπο μια πλειάδα μέσα στη σχέση

Το πεδίο χρηματιστής μόνο του δεν είναι υποψήφιο κλειδί διότι με αυτό δεν μπορούμε να αναγνωρίσουμε με μοναδικό τρόπο μια πλειάδα μέσα στη σχέση

Εξετάζουμε εάν η σχέση ΠΙΝΑΚΑΣ_Α(ΚωδΣυναλλαγής, Τύπος Μετοχής, Χρηματιστής) βρίσκεται σε 1KM, 2KM ή 3 KM.

Εάν ορίσουμε ως πρωτεύον κλειδί το ζεύγος γνωρισμάτων *ΚωδΣυναλλαγής, Τύπος Μετοχής* τότε σίγουρα είναι σε 1KM. Είναι και σε 2KM επειδή τα υπόλοιπα γνωρίσματα εξαρτώνται από το σύνολο του πρωτεύοντος κλειδιού και όχι μόνο από τμήμα του. Είναι και σε 3KM επειδή δεν υπάρχουν μεταβατικές εξαρτήσεις.

Στο προηγούμενο παράδειγμα ποια είναι η λύση:

ΠινακαςΑ (ΚωδΣυναλλαγής, Τύπος Μετοχής, Χρηματιστής)

Τα προσδιοριστικά γνωρίσματα είναι:

ΚωδΣυναλλαγής, Τύπος Μετοχής

ΚωδΣυναλλαγής, Χρηματιστής

Χρηματιστής

Ποια προσδιοριστικά γνωρίσματα μπορούν να λειτουργήσουν σαν υποψήφια κλειδιά;

ΚωδΣυναλλαγής, Τύπος Μετοχής **ΝΑΙ**

ΚωδΣυναλλαγής, Χρηματιστής **ΝΑΙ**

Χρηματιστής **ΟΧΙ**

Δημιουργήστε μία νέα σχέση για κάθε συναρτησιακή εξάρτηση:

ΠΙΝΑΚΑΣ_A(Χρηματιστής, Τύπος Μετοχής)

ΠΙΝΑΚΑΣ_B(ΚωδΣυναλλαγής, Χρηματιστής)

Σε αυτό το τελευταίο βήμα κρατήσαμε το προσδιοριστικό γνώρισμα " Χρηματιστής " στην αρχική σχέση ΠινακαςΑ.

Τέταρτη κανονική μορφή (4NF)

Μία σχέση είναι σε **4KM** εάν είναι σε **BC-NF** και δεν περιέχει εξαρτήσεις πολλαπλών τιμών (*multivalued dependencies*).

Με άλλα λόγια μια σχέση είναι σε 4KM εάν και μόνο εάν οι εξαρτήσεις πολλαπλών τιμών που ικανοποιεί είναι στην πραγματικότητα συναρτησιακές εξαρτήσεις που ξεκινούν από υποψήφια κλειδιά. Για να γίνει κατανοητό αυτό θα πρέπει πρώτα να δώσουμε τον ορισμό της εξάρτησης πολλαπλών τιμών.

Εξάρτηση Πολλαπλών Τιμών (Multivalued Dependency):

Έστω R μία σχέση, και A, B, C, τυχαία υποσύνολα του συνόλου των γνωρισμάτων της R. τότε το πεδίο B είναι πολλαπλά εξαρτημένο με το A ή $B \twoheadrightarrow A$ ή αλλιώς το A καθορίζει πολλαπλά το B, εάν το σύνολο τιμών του B που αντιστοιχούν σε δεδομένο ζεύγος (A,C) στη σχέση R, εξαρτάται μόνο από την τιμή A και είναι ανεξάρτητο από την τιμή C

Μπορούμε να πούμε ότι υπάρχουν τρία κριτήρια που πρέπει να προσεχθούν

- 1 Θα πρέπει να υπάρχουν τουλάχιστον 3 πεδία σε μια σχέση. Πχ A, B, και Γ
- 2 Γνωρίζοντας το A μπορεί κάποιος να καθορίσει πολλές τιμές για το B και γνωρίζοντας το A μπορεί κάποιος να καθορίσει πολλές τιμές για το Γ
- 3 Το B και το Γ είναι ανεξάρτητα μεταξύ τους

Πέμπτη κανονική μορφή (5NF)

Μία σχέση είναι σε πέμπτη κανονική μορφή (5KM) εάν και μόνο εάν οι μόνες εξαρτήσεις σύζευξης που ικανοποιεί είναι στην πραγματικότητα συναρτησιακές εξαρτήσεις που ξεκινούν από υποψήφια κλειδιά

Εξάρτηση Σύζευξης: Έστω R μια σχέση, και A,B,..Z, τυχαία υποσύνολα του συνόλου των γνωρισμάτων της R. τότε η R ικανοποιεί εξάρτηση σύζευξης *(A,B,..Z) εάν και μόνο εάν η R είναι ίση με τη σύζευξη των προβολών της πάνω στα A ,B...Z

Επίσης η πέμπτη κανονική μορφή απομακρύνει εξαρτήσεις σύζευξης.

Παράδειγμα 4.6

Στον παρακάτω πίνακα υπάρχουν εξαρτήσεις σύζευξης που ξεκινούν από το υποψήφια κλειδί A . Ποιες είναι αυτές οι εξαρτήσεις και τι λύση θα δίνετε;

ΠΙΝΑΚΑΣ_A (A,B,Γ,Δ,E,Z,H)

Λύση

Οι προβολές πάνω στα πεδία του πίνακα είναι. AB, AΓ, AΔ

Οι πίνακες που προκύπτουν είναι:

R1(A,B), R2(A,Γ), R3(A,Δ), R4(A,E), R5(A,Z), R6(A,H)

Η σύζευξη των πινάκων R1, R2,R3,R4,R5,R6 δίνει το ίδιο αποτέλεσμα με τον ΠΙΝΑΚΑ_A. Δηλ. με την εντολή

```
SELECT      R1.A, B,Γ,Δ,E,Z,H
FROM        R1,R2,R3,R4,R5,R6
WHERE       R1.A=R2.A AND R1.A=R3.A AND
            R1.A=R4.A AND R1.A=R5.A AND R1.A=R6.A
```

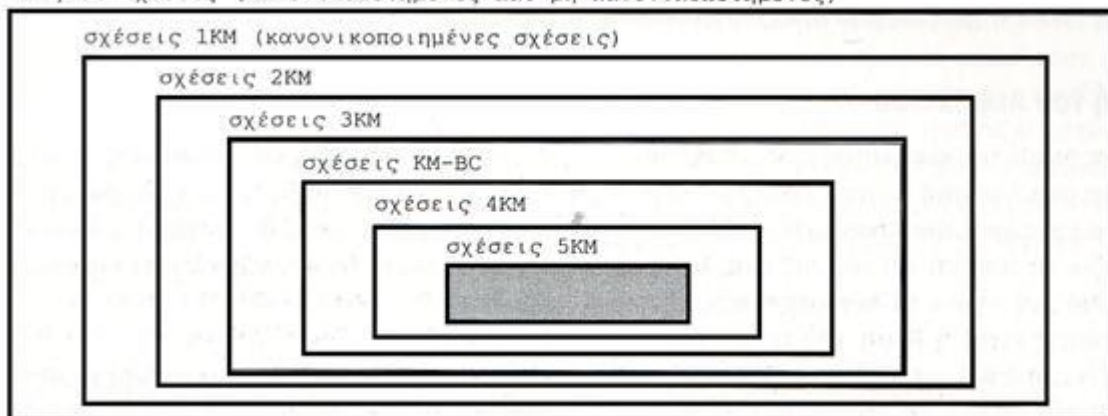
Κανονική μορφή Πεδίου ορισμού/ Κλειδιού (ΚΜ ΠΟ/Κ)

Κανονική μορφή Πεδίου ορισμού : Μια δέσμευση που λέει ότι οι τιμές ενός δεδομένου γνωρίσματος παίρνονται από ένα καθορισμένο πεδίο ορισμού

Κανονική μορφή Κλειδιού: Μια δέσμευση που ορίζει ότι κάποιο γνώρισμα ή συνδυασμός γνωρισμάτων είναι υποψήφιο κλειδί

Δέσμευση: Είναι ένας κανόνας που αναγκάζει τις στατικές τιμές ενός γνωρίσματος έτσι ώστε ο καθορισμός της τιμής να γίνεται μόνο όταν αυτός ο κανόνας είναι αληθής.

όλες οι σχέσεις (κανονικοποιημένες και μη κανονικοποιημένες)



Εικόνα4.1 . Οι κανονικές μορφές

Κανονική Μορφή Περιορισμού- Ένωσης

Δίνει απαντήσεις σε ερωτήσεις όπου η θεωρία της κανονικοποίησης δεν δίνει απαντήσεις. Το αποτέλεσμα είναι κακοί σχεδιασμοί σχέσεων.

Απο-Κανονικοποίηση (De-Normalization)

Υπάρχουν περιπτώσεις όπου μπορεί γίνει να από-κανονικοποίηση των σχέσεων ώστε να επιτευχθεί καλύτερη απόδοση της βάσης

Θεωρήστε την παρακάτω σχέση: ΠΕΛΑΤΕΣ (ΚΜ, Επίθετο, Όνομα, Διεύθυνση, Πόλη, , ΤΚ)

Η σχέση αυτή περιέχει συναρτησιακές εξαρτήσεις ΤΚ □ Πόλη

Μπορούμε να κάνουμε κανονικοποίηση ΚΜ ΠΟ/Κ χωρίζοντας την σχέση ΠΕΛΑΤΕΣ στα δύο ΠΕΛΑΤΕΣ (ΚΜ, Επίθετο, Όνομα, Διεύθυνση, ΤΚ) και

ΤαχΚΩΔΙΚΟΙ (ΤΚ, Πόλη)

Αυτό θεωρείτε λάθος γιατί όταν θελήσουμε την διεύθυνση ενός πελάτη θα ψάχνουμε σε δύο σχέσεις (πίνακες).

Το ίδιο ισχύει και στις περιπτώσεις των πεδίων ΤΗΛΕΦΩΝΟ. Είναι προτιμότερο να έχουμε στον ίδιο πίνακα 2 και 3 πεδία τηλεφώνου (ΤΗΛ_ΚΙΝ, ΤΗΛ_ΕΡΓΑΣΙΑΣ, ΤΗΛ_ΟΙΚΙΑΣ, ΤΗΛ_ΦΑΞ), γεγονός που έρχεται σε αντίθεση με ΙΚΜ, παρά να ψάχνουμε σε δύο πίνακες για να βρούμε το τηλέφωνο ενός ατόμου.

Σε τέτοιες περιπτώσεις η από-κανονικοποίηση αποφέρει μεγαλύτερη απόδοση τη βάσης.

SQL

Για να μπορέσουμε να δημιουργήσουμε και να διαχειριστούμε μια βάση δεδομένων, μπορούμε να χρησιμοποιήσουμε ειδικές γλώσσες προγραμματισμού, τις λεγόμενες γλώσσες ερωταπαντήσεων (query languages). Είναι γλώσσες μη διαδικαστικές, τέταρτης γενιάς (4th generation languages). Εμείς απλά διατυπώνουμε με απλές και κατανοητές εντολές το τι πληροφορίες ζητάμε και το ΣΔΒΔ (Σύστημα Διαχείρισης Βάσεων Δεδομένων) αναλαμβάνει να μας απαντήσει. Η SQL (Structured Query Language, δηλ. Δομημένη Γλώσσα Ερωτημάτων) είναι σήμερα η πιο δημοφιλής και πιο διαδεδομένη γλώσσα ανάπτυξης και διαχείρισης σχεσιακών βάσεων δεδομένων.

- Η SQL μάς δίνει τη δυνατότητα να έχουμε πρόσβαση σε μια βάση δεδομένων (database).
- Η SQL αποτελεί μια στάνταρτ γλώσσα του ANSI (ANSI standard language).
- Η SQL μπορεί να εκτελέσει ερωτήματα (queries) και να αναζητήσει πληροφορίες σε μια βάση δεδομένων.
- Η SQL μπορεί να ανακτήσει δεδομένα από μια βάση δεδομένων.
- Η SQL μπορεί να εισαγάγει νέες εγγραφές σε μια βάση δεδομένων.
- Η SQL μπορεί να διαγράψει εγγραφές από μια βάση δεδομένων.
- Η SQL μπορεί να ενημερώσει εγγραφές σε μια βάση δεδομένων.
- Η SQL είναι πολύ εύκολη στην εκμάθηση.

Η SQL αποτελείται από εντολές με τα ορίσματά τους, τις οποίες μπορούμε να χρησιμοποιήσουμε με συγκεκριμένους κανόνες σύνταξης για να πάρουμε τα αποτελέσματα που θέλουμε. Με την SQL μπορούμε να δημιουργήσουμε μια βάση δεδομένων και τους πίνακές της με τα αντίστοιχα πεδία, να καταχωρήσουμε δεδομένα στους πίνακες, να τροποποιήσουμε και να διαγράψουμε τα δεδομένα αυτά, να αλλάξουμε τη δομή των πινάκων με προσθήκη και διαγραφή πεδίων και να εμφανίσουμε πληροφορίες (συνδυασμούς από δεδομένα).

Η SQL έχει διάφορα τμήματα, τα πιο βασικά είναι τα παρακάτω :

- Τη Γλώσσα Ορισμού Δεδομένων (**DDL, Data Definition Language**), η οποία περιέχει τις απαραίτητες εντολές για τον ορισμό και την τροποποίηση του σχεσιακού σχήματος καθώς και για τη δημιουργία, την τροποποίηση και τη διαγραφή σχέσεων. Περιέχει ακόμη τις εντολές δημιουργίας και επεξεργασίας όψεων και ορισμού περιορισμών ακεραιότητας.
- Τη Γλώσσα Χειρισμού Δεδομένων (**DML, Data Manipulation Language**), η οποία περιέχει τις απαραίτητες εντολές για την εμφάνιση (αναζήτηση) δεδομένων καθώς και για την καταχώρηση, τροποποίηση και διαγραφή των εγγραφών (πλειάδων) μιας σχέσης.
- Τέλος, περιέχει εντολές για τον ορισμό και την επεξεργασία συναλλαγών (transactions) και εντολές για την ασφάλεια (authentication).

Τα διάφορα τμήματα της SQL παρουσιάζονται συνοπτικά στον παρακάτω πίνακα:

	REATE DROP ALTER TABLE (base table)
--	---

Γλώσσα Ορισμού Δεδομένων (DDL)	CREATE DROP ALTER VIEW (virtual table) CREATE DROP ALTER INDEX (index table)
Γλώσσα Χειρισμού Δεδομένων (DML)	SELECT αναζήτηση INSERT εισαγωγή εγγραφής DELETE διαγραφή εγγραφής UPDATE τροποποίηση εγγραφής
Προσωποποιημένη Γλώσσα Χειρισμού Δεδομένων (PL/SQL) (Static SQL) (Dynamic SQL)	DECLARE Sinfo CURSOR FOR SELECT S_name, S_sity FROM SUPPLIERS ORDER BY S_name; OPEN Sinfo; LOOP { FETCH Sinfo INTO : sname, : ssity; CLOSE Sinfo; String CDataSourceSet::GetDefaultConnect() return ODBC;DSN=DataSource;"; } String CDataSourceSet::GetDefaultSQL() return "SELECT * FROM SUPPLIERS ORDER BY S_Name"; }
Ορισμός Όψεων	CREATE VIEW ATHENS_SUPPLIERS SELECT S_No, S_Name, S_City FROM SUPPLIERS WHERE S_City="ATHENS";
Προσδιορισμός δικαιωμάτων (authorization)	CREATE DROP ALTER User CREATE USER John IDENTIFIED BY Johns-password; GRANT CONNECT TO John; GRANT SELECT, UPDATE ON SUPPLIERS TO John; REVOKE SELECT ON SUPPLIERS FROM John;
Προστασία (constraints)	CREATE DOMAIN S_No AS INTEGER CONSTRAINT Έλεγχος-Αριθμού ρομηθευτή CHECK (S_No >0); CREATE ASSERTION <Όνομα-δήλωσης> CHECK <Κατηγορία>; DEFINE TRIGGER <όνομα> ON UPDATE OF SUPPLIERS (...)
Έλεγχος Συναλλαγών	SQL> SET TRANSACTION; INSERT INTO CUSTOMERS VALUES ('SMITH', 'JOHN'); SQL> COMMIT;

Χειρισμός Δεδομένων της SQL (Data Manipulation Language)

Όπως δηλώνει και το όνομά της, η SQL είναι μια σύνταξη για την εκτέλεση ερωτημάτων (queries). Αλλά η γλώσσα της SQL περιλαμβάνει επίσης μια σύνταξη για την ενημέρωση εγγραφών, την εισαγωγή νέων εγγραφών και τη διαγραφή υπαρχόντων εγγραφών.

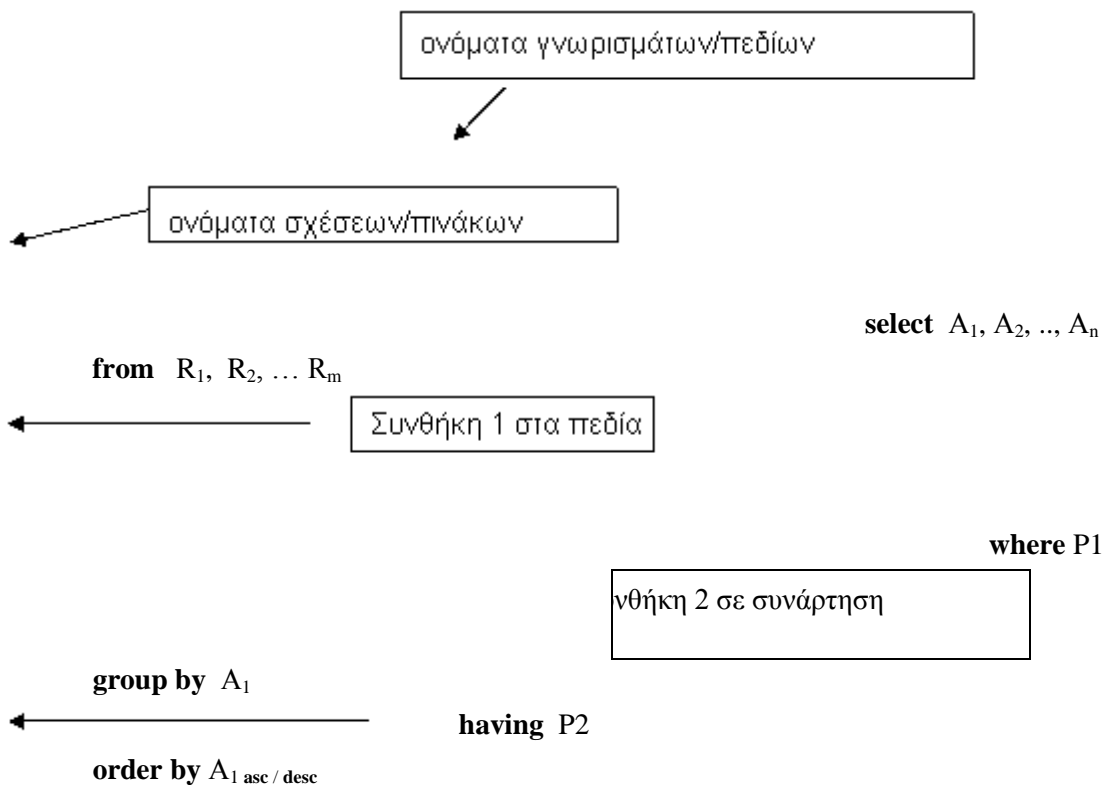
Η Γλώσσα Χειρισμού Δεδομένων (Data Manipulation Language, DML) που αποτελεί κομμάτι της SQL χρησιμοποιεί τις παρακάτω εντολές :

- SELECT**
 - προβάλλει τα δεδομένα αναζήτησης από μια βάση δεδομένων.
- INSERT**
- εισάγει νέα δεδομένα σε μια βάση δεδομένων.
 - UPDATE**
 - ενημερώνει δεδομένα σε μια βάση δεδομένων.
 - DELETE**
 - διαγράφει δεδομένα από μια βάση δεδομένων.

Στη συνέχεια θα δοθεί μη σύνταξη των παραπάνω εντολών. Στις εντολές αυτές θα αναφερθούμε συνοπτικά και θα δώσουμε μεγαλύτερη έμφαση σε ερωτήματα περίπλοκα και εμφωλιασμένα.

Select

Η εντολή **SELECT** προβάλλει δεδομένα από τις στήλες (columns) ενός πίνακα της βάσης δεδομένων. Το αποτέλεσμα προβάλλεται σε μορφή πίνακα και αποκαλείται result set. Η σύνταξη της εντολής είναι η εξής:



A_i : αναπαριστά γνωρίσματα, R_i : αναπαριστά σχέσεις(πίνακες), P : είναι συνθήκη

Αυτό το ερώτημα είναι ισοδύναμο με την έκφραση της σχεσιακής άλγεβρας

$\pi_{A_1, A_2, \dots, A_n} (\sigma_P (R_1 \times R_2 \times \dots \times R_m))$

select αντιστοιχεί στην πράξη της προβολής της σχεσιακής άλγεβρας. Τα γνωρίσματα που θέλουμε να υπάρχουν στο αποτέλεσμα της ερώτησης.

from αντιστοιχεί στην πράξη του καρτεσιανού γινομένου της σχεσιακής άλγεβρας. Ποιες σχέσεις(πίνακες) θα χρησιμοποιηθούν για τον υπολογισμό του αποτελέσματος.

where αντιστοιχεί στη συνθήκη της πράξης της επιλογής στη σχεσιακή άλγεβρα. Το κριτήριο P έχει γνωρίσματα των σχέσεων που εμφανίζονται στο **from**.

group by ο όρος χρησιμοποιείται στις περιπτώσεις ομαδοποίησης αποτελεσμάτων σύμφωνα με ένα γνώρισμα. Συνήθως χρησιμοποιείται όταν υπάρχουν συναθροιστικές συναρτήσεις (sum, avg, count, min, max) στο select.

having η πρόταση **αυτή** περιορίζει τις γραμμές που επιστρέφονται από μία πρόταση **group by** με τον ίδιο τρόπο που η πρόταση **where** περιορίζει τις γραμμές που επιστρέφονται από την πρόταση **select**. Τόσο η **where** όσο και η **having** μπορούν να συμπεριληφθούν στην ίδια πρόταση **select**. Η **where** εφαρμόζεται πριν την ομαδοποίηση και η **having** εφαρμόζεται αφού σχηματιστούν οι ομάδες (**group by**) και υπολογιστούν οι συναθροιστικές συναρτήσεις.

order by χρησιμοποιείται ώστε οι εγγραφές/ γραμμές/ πλειάδες/ συστοιχίες να είναι ταξινομημένες με βάση το αντίστοιχο γνώρισμα κατά αύξουσα σειρά asc ή κατά φθίνουσα σειρά desc.

ΠΡΟΣΟΧΗ: Δε γίνεται απαλοιφή των διπλών εμφανίσεων (διπλότυπα).

Η SQL επιτρέπει και πολλαπλές εμφανίσεις της ίδιας συστοιχίας σε μια σχέση. Μια σχέση στην SQL είναι ένα πολυσύνολο (multiset)

Η σημασιολογία των διπλότυπων στην SQL:

```
select A1, A2, ..., An
from R1, R2, ... Rm
where P
```

είναι ισότιμο με την **εκδοχή πολλαπλών συνόλων** της έκφρασης:

$\pi_{A_1, A_2, \dots, A_n}(\sigma_P(R_1 \times R_2 \times \dots \times R_m))$

Για την Απαλοιφή διπλών εμφανίσεων (διπλότυπα) χρησιμοποιούμε το **distinct**. Π.χ.

```
select distinct A1, A2, ..., An
from R1, R2, ... Rm
```

Insert

Για να εισάγουμε δεδομένα σε μια σχέση μπορούμε να το υλοποιήσουμε με την εντολή insert into, εισάγοντας (α) μεμονωμένες τιμές είτε (β) πλήθος τιμών από έναν άλλο πίνακα

(α)

```
Insert into R(A1, ..., An) values (v1, ..., vn)
```

(β)

```
Insert into R(A1, ..., An) select-from-where
```

Update

Η γενική σύνταξη είναι

```
Update όνομα-πίνακα Set όνομα-στήλης = νέα-τιμή
Where P
```

Η ενημέρωση γίνεται σε όλες τις γραμμές του πίνακα που ικανοποιούν την συνθήκη P

Delete

Με την εντολή αυτή γίνεται η διαγραφή των εγγραφών από έναν πίνακα

```
Delete from R where P
```

Σβήνει όλες τις γραμμές της R για τις οποίες ισχύει η συνθήκη P.

Όταν λείπει το **where** σβήνονται όλες οι εγγραφές ενός πίνακα

.

Ορισμός Δεδομένων της SQL (Data Definition Language)

Με τις εντολές αυτές της SQL ορίζουμε το σχήμα της βάσης δεδομένων, μπορούμε να το τροποποιήσουμε αλλά και να το διαγράψουμε.

Οι βασικότερες εντολές που χρησιμοποιεί η DDL είναι οι εξής :

- CREATE TABLE**
 - δημιουργεί έναν πίνακα.
- ALTER TABLE**
- Τροποποιεί την δομή ενός πίνακα.
- DROP TABLE**
 - διαγράφει ολόκληρο πίνακα

Create Table

Η σύνταξη της εντολής για την δημιουργία ενός νέου πίνακα είναι :

Create table R(A_1 D_1 , A_2 D_2 ,, A_n D_n),

<περιορισμός-ακεραιότητας_i>,

....,

<περιορισμός-ακεραιότητας_k>

όπου **R** είναι το όνομα του πίνακα, A_i τα ονόματα των γνωρισμάτων, και D_i οι τύποι των αντίστοιχων πεδίων ορισμού.

Τύποι Πεδίων Ορισμού

Για τον ορισμό του πεδίου ορισμού, οι διαθέσιμοι built-in τύποι περιλαμβάνουν:

char(n). Αλφαριθμητικό σταθερού μήκους n που ορίζεται από τον χρήστη.

varchar(n). Αλφαριθμητικό μεταβλητού μήκους, με μέγιστο μήκος n οριζόμενο από τον χρήστη.

int. Ακέραιος (υποσύνολο που εξαρτάται από τη μηχανή)

smallint. Μικρός ακέραιος (υποσύνολο που εξαρτάται από τη μηχανή)

numeric(p,d). Αριθμός σταθερής υποδιαστολής, με ακρίβεια p ψηφίων όπου τα d είναι στα δεξιά της υποδιαστολής.

real, double precision. Πραγματικοί αριθμοί, διπλής ακρίβειας, όπου η ακρίβεια εξαρτάται από τη μηχανή.

float(n). Πραγματικοί αριθμοί κινητής υποδιαστολής, με ακρίβεια τουλάχιστον n ψηφίων οριζόμενη από τον χρήστη.

date. Ημερομηνία της μορφής Π.χ. **date** '2002-10-22'

time. Ώρα της ημέρας σε ώρες, λεπτά, δευτερόλεπτα. Π.χ. **time** '09:05:32'

timestamp. Η ημέρα συν την ώρα. Π.χ. **timestamp** '2002-10-22 09:05:32.75'

interval. Χρονικό διάστημα. Π.χ. διάστημα '1' ημέρας

Οι τιμές null επιτρέπονται στα πεδία ορισμού. Δηλώνοντας ένα πεδίο ως **not null** απαγορεύει οι τιμές του πεδίου να είναι κενές και προϋποθέτει την υποχρεωτική καταχώριση τιμής σε αυτό το πεδίο..

Περιορισμοί Ακεραιότητας

Επιτρεπτοί **περιορισμοί ακεραιότητας** είναι της μορφής:

- primary key** A_{j1} , A_{j2} , ..., A_{jn} , (δεν επιτρέπονται διπλόεγγραφές και NULL τιμές), ορίζει το πρωτεύον κλειδί του πίνακα.
- unique** A_{j1} , A_{j2} , ..., A_{jn} , (δεν επιτρέπονται διπλοεγγραφές; NULL τιμές επιτρέπονται)
- check** P έλεγχος μιας συνθήκης σε ένα πεδίο
- foreign key** (A_i) **references** A_i ορισμός ενός πεδίου ως ξένο κλειδί.

Alter Table

Η σύνταξη της εντολής αυτής μπορεί να γίνει με ένα από του παρακάτω τρεις τρόπους ανάλογα με τις αντίστοιχες απαιτήσεις

alter table R ADD όνομα_στήλης τύπος δεδομένων

alter table R DROP COLUMN όνομα_στήλης

alter table R MODIFY (όνομα_στήλης νέος τύπος δεδομένων)

- **ADD** - προσθέτει καινούργια στήλη
- **DROP** - διαγράφει μια στήλη
- **MODIFY** - τροποποιεί μια στήλη

Η εντολή **modify** μπορεί να τροποποιήσει μόνο τον τύπο δεδομένων, όχι το όνομα της στήλης. Μερικά RDBMS δεν επιτρέπουν τη διαγραφή (DROP) στήλης σε έναν πίνακα της βάσης

Drop Table

Για να σβηστεί ένας πίνακας R η σύνταξη της εντολής είναι η παρακάτω:

drop table R

Δεν πρέπει να την συγχέουμε με την εντολή **delete from R** η οποία διαγράφει μόνο τις εγγραφές από τον πίνακα. Η εντολή Drop Table διαγράφει και εξαφανίζει οριστικά ολόκληρο τον πίνακα.

Ορισμός Όψεων της SQL (View Definition Language)

Με τις εντολές αυτές της SQL ορίζουμε όψεις της βάσης δεδομένων, όπου μπορούμε να εκτελέσουμε ερωτήματα και τα αποτελέσματα τους να τα χρησιμοποιήσουμε με τον ίδιο τρόπο που χρησιμοποιούμε τους πίνακες μιας βάσης. Τις όψεις αυτές μπορούμε να τις τροποποιήσουμε αλλά και να τις διαγράψουμε. Οι όψεις αποτελούν ένα σημαντικό εργαλείο των βάσεων δεδομένων όσων αφορά την ασφάλεια των δεδομένων. Χρησιμοποιώντας όψεις αντί για πίνακες μειώνεται οι πιθανότητα να αλλοιωθούν τα δεδομένα ενός πίνακα μιας και ο τελικός χρήστης δεν θα έχει απευθείας πρόσβαση στον πίνακα και σε όλα τα στοιχεία του, αλλά θα βλέπει και θα επεξεργάζεται τα στοιχεία που θα προβάλλει η όψη.

Οι βασικότερες εντολές που χρησιμοποιεί η VDL είναι οι εξής :

- **CREATE VIEW**
 - δημιουργεί μία όψη.
- **ALTER VIEW**
 - Τροποποιεί τη δομή μιας όψεις.
- **DROP VIEW**
 - διαγράφει ολόκληρη την όψη

Create View

Η σύνταξη της εντολής για την δημιουργία μιας όψης είναι :

Create View V As

Select

From

Where

.....,

όπου V είναι το όνομα της όψης, και ακολουθεί η σύνταξη ενός ερωτήματος που ξεκινά με την εντολή select.

Alter View

Η σύνταξη της εντολής αυτής μπορεί να γίνει με ένα από του παρακάτω τρεις τρόπους ανάλογα με τις αντίστοιχες απαιτήσεις

Alter View V As

Select

Όπου V είναι το όνομα της όψης που πρόκειται να τροποποιηθεί.

Drop View

.Για να διαγραφεί μία όψη V, η σύνταξη της εντολής είναι η παρακάτω:

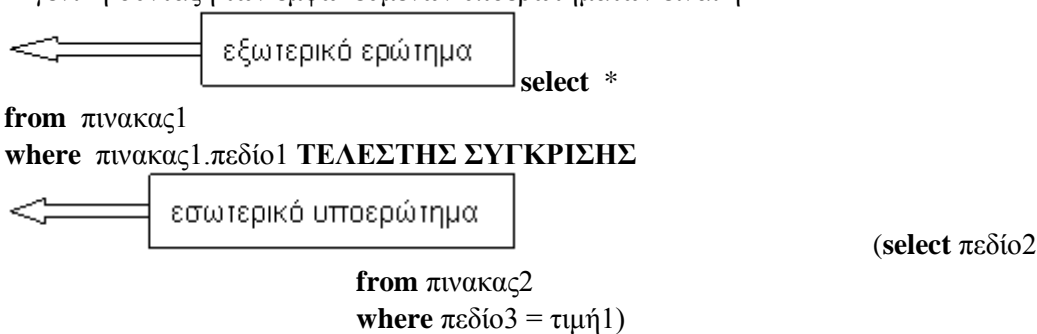
Drop View V

Εμφωλευμένα Υπο-ερωτήματα

Ένα υποερώτημα είναι μια έκφραση **select – from – where** που είναι εμφωλευμένη (ένθετη) μέσα σε μια άλλη έκφραση. Τότε το πρώτο ονομάζεται εξωτερικό ερώτημα και το δεύτερο εσωτερικό υποερώτημα.

Υπολογίζεται πρώτα το εσωτερικό υποερώτημα και κατόπιν το εξωτερικό εφαρμόζοντας σε αυτό το αποτέλεσμα του εσωτερικού ερωτήματος. Εξαίρεση αποτελεί η περίπτωση των συσχετιζόμενων υποερωτημάτων όπου το εσωτερικό υποερώτημα θα υπολογιστεί για κάθε γραμμή (συστοιχία) του εξωτερικού ερωτήματος.

Η γενική σύνταξη των εμφωλευμένων υποερωτημάτων είναι η



Σαν τελεστές σύγκρισης χρησιμοποιούνται οι < <= > >= = <>

Τα εμφωλευμένα υποερωτήματα χρησιμοποιούνται όταν **δεν γνωρίζουμε την τιμή σύγκρισης** στη συνθήκη που υπάρχει σε προτάσεις **where** ή **having**.

Στις περιπτώσεις όπου το εσωτερικό υποερώτημα θα έχει ως αποτέλεσμα την επιστροφή δύο ή περισσότερων τιμών χρησιμοποιούνται οι τελεστές **in, not in, some, all, exists, unique**.

Ο τελεστής **in (not in)** ελέγχει αν μια πλειάδα ανήκει (ή δεν ανήκει) σε ένα σύνολο από πλειάδες που έχουν προκύψει από μια έκφραση select-from-where.

Ο τελεστής **some** έχει τη σημασία του **τουλάχιστον ένα** από ένα σύνολο

Επίσης ισχύει (= **some**) ισοδύναμο. του **in**

Ωστόσο (□ **some**) όχι ισοδύναμο του **not in**

Ο τελεστής **any** έχει τη σημασία του **το πολύ ένα** από ένα σύνολο

Ο τελεστής **all** έχει τη σημασία από όλα τα στοιχεία ενός συνόλου.

Επίσης ισχύει (□ **all**) ισοδύναμο του **not in**

Το (= **all**) ελέγχει αν το υποερώτημα έχει μόνο μία τιμή, δεν είναι ισοδύναμο του **in**.

Ο τελεστής **exists** χρησιμοποιείται στον **έλεγχο για κενά σύνολα** (κενές σχέσεις) και επιστρέφει **true** αν η υποερώτηση δεν είναι κενή

Ο τελεστής **not exists**: επιστρέφει true αν η υποερώτηση είναι κενή

Ο τελεστής **unique (not unique)** ελέγχει αν ένα υποερώτημα περιέχει διπλότυπες πλειάδες στο αποτέλεσμα του (έλεγχος για διπλότυπα).

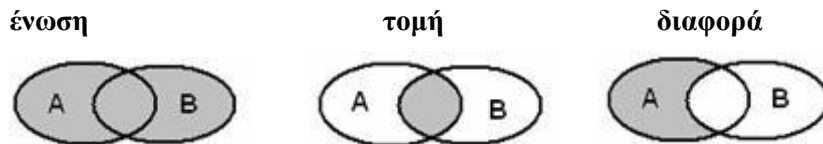
Ο τελεστής **unique**: επιστρέφει **true** αν η υποερώτηση δεν έχει διπλότυπες πλειάδες.

Ο τελεστής **not unique**: επιστρέφει **true** αν η υποερώτηση έχει διπλότυπα.

Ο πιο συχνά χρησιμοποιούμενος τελεστής από τους παραπάνω είναι ο **in (not in)**.

Πράξεις Συνόλων

Οι πράξεις δύο συνόλων A και B είναι η ένωση η τομή και διαφορά τους όπως φαίνεται στην παρακάτω εικόνα:



Εικόνα 5.1 Πράξεις συνόλων.

Οι πράξεις συνόλων εφαρμόζονται σε **πίνακες** που έχουν τον ίδιο αριθμό στηλών και ίδιο τύπο δεδομένων.

Γενική σύνταξη:

Πίνακας A

<Πράξη>

Πίνακας B

Οι τελεστές που μπορούν να χρησιμοποιηθούν για τις πράξεις συνόλων είναι

union ένωση

intersect τομή

except διαφορά

Καθένας από τους παραπάνω τελεστές αυτόματα διαγράφει τα διπλότυπα (απαλοιφή διπλών εμφανίσεων). Για την τήρηση των διπλοτύπων χρησιμοποιείται η αντίστοιχη εκδοχή πολλαπλών συνόλων **union all**, **intersect all**, **except all**.

ο τελεστής **union** υποστηρίζεται από τα περισσότερα σχεσιακά συστήματα διαχείρισης βάσεων δεδομένων.

Τύποι Σύζευξης

Ο κλασικός τρόπος εκτέλεσης ενός ερωτήματος από την στιγμή που καλούνται πάνω από ένας πίνακες που συσχετίζονται μεταξύ τους σε κάποια πεδία είναι να χρησιμοποιήσουμε μια συνθήκη στο where του ερωτήματος όπου θα ελέγχουμε τις τιμές από τα κοινά πεδία των πινάκων που μετέχουν στην έκφραση from να είναι ίσες μεταξύ τους. Σε περίπτωση που δεν κάνουμε τον παραπάνω έλεγχο, θα παρουσιάσουμε το καρτεσιανό γινόμενο όλων των εγγραφών από όλους τους πίνακες και το αποτέλεσμα θα είναι λάθος.

select R1.A1, A2, A3, A4

from R1,R2

where R1.A1=R2.A1

Αντί για τον παραπάνω τρόπο χρησιμοποιούμε καλύτερα τους παρακάτω τύπους σύζευξης πινάκων.

inner join: εσωτερική (φυσική) σύζευξη	ημιάνει τις εγγραφές από τους δύο πίνακες όπου οι τιμές του κοινού πεδίου είναι ίδιες
left outer join: αριστερή εξωτερική σύζευξη	ημιάνει όλες τις εγγραφές του αριστερού πίνακα και τις εγγραφές του δεξιού όπου οι τιμές του κοινού πεδίου είναι ίδιες
right outer join: δεξιά εξωτερική σύζευξη	ημιάνει όλες τις εγγραφές του δεξιού πίνακα και τις εγγραφές του αριστερού όπου οι τιμές του κοινού πεδίου είναι ίδιες
full outer join: ολοκληρή εξωτερική σύζευξη	ένωση (union) των δύο παραπάνω

(Η λέξη OUTER προαιρετική):

Σε περίπτωση δύο πινάκων που συσχετίζονται μεταξύ τους με το πεδίο A1

```
select R1.A1, A2, A3, A4
from R1 ΣΥΖΕΥΞΗ R2 on R1.A1=R2.A2
```

Σε περίπτωση τριών πινάκων που συσχετίζονται μεταξύ τους ο R1 και R2 με το πεδίο A1 και ο R2, R3 με το πεδίο A2

```
select R1.A1, R2.A2, A3, A4
from R1 ΣΥΖΕΥΞΗ (R2 ΣΥΖΕΥΞΗ R3 on R2.A2=R3.A2)
on R1.A1=R2.A1
```

Ο τρόπος σύζευξης που χρησιμοποιείται συχνότερα είναι η εσωτερική (φυσική) σύζευξη inner join.

Παραγόμενες Σχέσεις

Η SQL-92 δίνει τη δυνατότητα μια υποερώτηση να χρησιμοποιηθεί στο **from** αντί για το όνομα ενός πίνακα.

Τότε πρέπει να τις δοθεί ένα όνομα στα αποτελέσματα της υποερώτησης και τα γνωρίσματα της να μετονομαστούν

Αυτό γίνεται χρησιμοποιώντας το **as** και δίνεται έτσι ένα προσωρινό όνομα στην προσωρινή σχέση που προκύπτει από την υποερώτηση.

Στην παρακάτω σύνταξη μιας παραγόμενης σχέσης παρατηρούμε ότι το R2 είναι το όνομα του πίνακα που προκύπτει από την εκτέλεση ενός υποερωτήματος που υπολογίζει την μέση τιμή του πεδίου A2 από τον πίνακα R1, και την ονομάζει A3, ομαδοποιώντας τα αποτελέσματα κατά το πεδίο A1. Στο τέλος συγκρίνουμε το πεδίο A3 της παραγόμενης σχέσης με μία τιμή X.

```
select A1, A3
from (select A1, avg(A2) as A3
from R1
group by A1 )
as R2(A1, A3)
```

where A3 > X

Να σημειώσουμε ότι Οι παραγόμενες σχέσεις δεν υποστηρίζονται από όλα τα RDBMS.

Το ίδιο αποτέλεσμα θα δινόταν εάν δημιουργούσαμε πρώτα μια όψη με όνομα R2 και στη συνέχεια την χρησιμοποιούσαμε σε ένα επόμενο ερώτημα μιας άλλης όψης.

Create View R2 As

```
select A1, avg(A2) as A3
from R1
group by A1
```

Create View V1 As

```
select A1, A3
from R2
where A3 > X
```

Στη συνέχεια θα παραθέσουμε παραδείγματα SQL για όλες τις παραπάνω περιπτώσεις.

Παράδειγμα 5.1

Δημιουργήστε με εντολές SQL τον πίνακα ATZENTA με τα παρακάτω πεδία.

AA	int	4	
EPITHETO	varchar	50	
ONOMA	varchar	50	✓
ON_PATROS	varchar	50	✓
HM_GENNHISIS	smalldatetime	4	✓
ODOS	varchar	50	✓
AR	varchar	50	✓
TK	varchar	50	✓
POLH	varchar	50	✓
THL_ERGASIAS	varchar	50	✓
THL_SPITIOY	varchar	50	✓
KINHTO	varchar	50	✓
SXOLIA	varchar	50	✓

Λύση

```
CREATE TABLE ATZENTA (
    AA          INT          NOT NULL,
    EPITHETO   VARCHAR(50),
    ONOMA      VARCHAR(50),
    ON_PATROS  VARCHAR(50),
    HM_GENNHISIS SMALLDATETIME,
    ODOS       VARCHAR(50),
    AR         VARCHAR(50),
    TK         VARCHAR(50),
    .....
    ..... ,
    PRIMARY KEY (AA))
```


Παράδειγμα 5.2

Εμφανίστε όλα τα στοιχεία του πίνακα ATZENTA που το επίθετό αρχίζει από Κ.

```
ATZENTA(AA, EPIHETHO, ONOMA, ON_PATROS, HM_GENNHSIS,
ODOΣ, AR, TK, POLH, THL)
```

Λύση

```
SELECT *
FROM ATZENTA
WHERE EPIHETHO LIKE 'Κ%'
```

Παράδειγμα 5.3

Εμφανίστε Επίθετο, Όνομα και Τηλέφωνο των ατόμων του πίνακα ATZENTA που γεννήθηκαν μετά το 1997 και είναι από την πόλη Σέρρες.

```
ATZENTA(AA, EPIHETHO, ONOMA, ON_PATROS, HM_GENNHSIS,
ODOΣ, AR, TK, POLH, THL)
```

Λύση

```
SELECT EPIHETHO, ONOMA, THL
FROM ATZENTA
WHERE HM_GENNHSIS > '1-1-1197' AND POLH='Σέρρες'
```

Παράδειγμα 5.4

Ομαδοποιήστε τα άτομα του πίνακα ATZENTA ανά πόλη και μετρήστε πόσοι είναι από κάθε πόλη.

```
ATZENTA(AA, EPIHETHO, ONOMA, ON_PATROS, HM_GENNHSIS,
ODOΣ, AR, TK, POLH, THL)
```

Λύση

```
SELECT POLH, COUNT(AA) AS PLHTHOS
FROM ATZENTA
GROUP BY POLH
```

Παράδειγμα 5.5

Προβάλετε τις καταχωρήσεις του πίνακα ATZENTA που κατάγονται από την ίδια πόλη που κατάγεται ο Γεωργίου (φωλιασμένο ερώτημα)

```
ATZENTA(AA, EPIHETHO, ONOMA, ON_PATROS, HM_GENNHSIS,
ODOΣ, AR, TK, POLH, THL)
```

Λύση

```
SELECT *
FROM AZENTA
WHERE POLH = (SELCET POLH
FROM ATZENTA
WHERE EPIHETHO='Γεωργίου')
```

Παράδειγμα 5.6

Προβάλετε σε ένα ερώτημα τις καταχωρήσεις του πίνακα ATZENTA που κατάγονται από Αθήνα και αυτές από Θεσσαλονίκη.

```
ATZENTA(AA, EPIHETO, ONOMA, ON_PATROS, HM_GENNHSIS,
ODOΣ, AR, TK, POLH, THL)
```

Λύση

```
SELECT *
FROM AZENTA
WHERE POLH 'Αθήνα' OR POLH 'Θεσσαλονίκη'
```

Παράδειγμα 5.7

Εισάγετε στον πίνακα ATZENTA μια εγγραφή με τα παρακάτω στοιχεία

123, ΔΗΜΗΤΡΙΟΥ, ΙΩΑΝΝΗΣ, ΓΕΩΡΓΙΟΣ, 7-7-1974, ΕΡΜΟΥ, 12, 62122, ΣΕΡΡΕΣ.

```
ATZENTA(AA, EPIHETO, ONOMA, ON_PATROS, HM_GENNHSIS,
ODOΣ, AR, TK, POLH, THL)
```

Λύση

```
INSERT INTO ATZENTA
VALUES(123, 'ΔΗΜΗΤΡΙΟΥ', 'ΙΩΑΝΝΗΣ', 'ΓΕΩΡΓΙΟΣ', #7-7-1974#, 'ΕΡΜΟΥ', 12, '62122',
'SΕΡΡΕΣ', '23210-45678')
```

Παράδειγμα 5.8

Διορθώστε στον πίνακα ATZENTA το τηλέφωνο της εγγραφής με AA ίσο με 1234 και δώστε στο τηλέφωνο την τιμή 2321-55532

```
ATZENTA(AA, EPIHETO, ONOMA, ON_PATROS, HM_GENNHSIS,
ODOΣ, AR, TK, POLH, THL)
```

Λύση

```
UPDATE ATZENTA
SET THL= '23210-55532'
WHERE AA=1234
```

Παράδειγμα 5.9

Διορθώστε την δομή του πίνακα ATZENTA και προσθέστε ακόμη ένα πεδίο με όνομα EMAIL που θα έχει τύπο δεδομένων VARCHAR(20).

Λύση

```
ALTER TABLE ATZENTA ADD EMAIL VARCHAR(20)
```

Παράδειγμα 5.10

Διορθώστε την δομή του πίνακα ATZENTA και τροποποιήστε τον τύπο δεδομένων του πεδίου EMAIL ώστε να έχει τύπο δεδομένων VARCHAR(50).

Λύση

```
ALTER TABLE ATZENTA MODIFY EMAIL VARCHAR(50)
```

Παράδειγμα 5.11

Διορθώστε την δομή του πίνακα ATZENTA και διαγράψτε το πεδίο EMAIL

Λύση

```
ALTER TABLE ATZENTA DROP EMAIL
```

Παράδειγμα 5.12

Διαγράψτε όλες τις εγγραφές από τον πίνακα ATZENTA.

Λύση

```
DELETE FROM ATZENTA
```

Παράδειγμα 5.13

Διαγράψτε τον πίνακα ATZENTA.

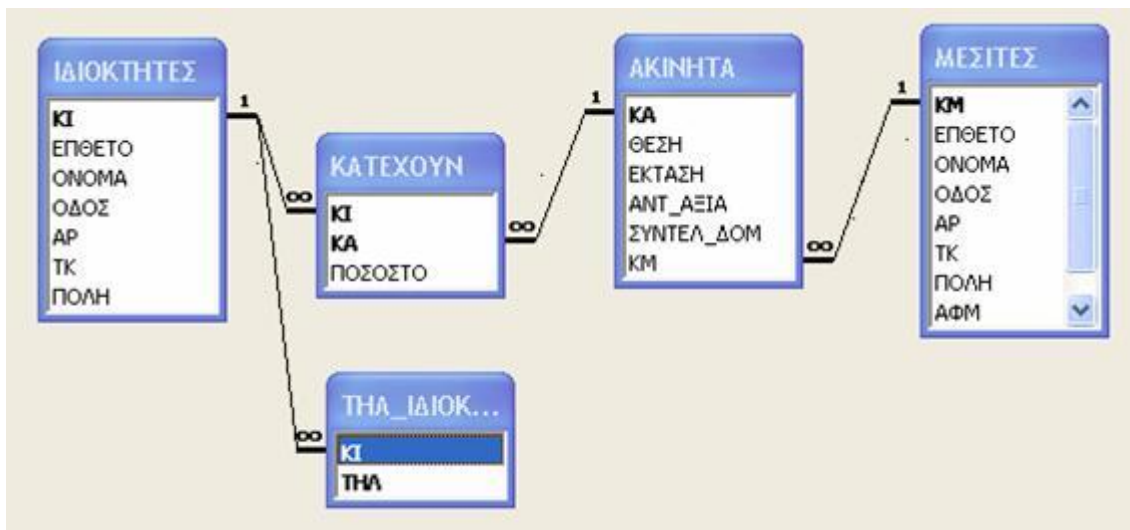
Λύση

```
DROP TABLE ATZENTA
```

Παράδειγμα 5.14

- A. Δημιουργήστε με SQL τον πίνακα KATEXOYN από το παρακάτω σχήμα βάσης.
- B. Δημιουργήστε με SQL ένα ερώτημα που θα προβάλλει αναλυτικά όλα τα απαραίτητα πεδία σχετικά με τα ακίνητα που κατέχει ο κάθε ιδιοκτήτης, και την έκταση ακριβώς που κατέχει σε κάθε ακίνητο.
- Γ. Δημιουργήστε με SQL ένα ερώτημα που θα προβάλλει τον κωδικό του ιδιοκτήτη και την συνολική έκταση που κατέχει ο κάθε ιδιοκτήτης.
- Δ. Δημιουργήστε με SQL ένα ερώτημα που θα προβάλλει τον κωδικό του ιδιοκτήτη και την συνολική έκταση που κατέχει ο κάθε ιδιοκτήτης για τους ιδιοκτήτες που κατέχουν συνολικά πάνω από 10.000 τμ.
- E. Δημιουργήστε με SQL ένα ερώτημα που θα προβάλλει κωδικό του μεσίτη και την συνολική αντικειμενική αξία που έχουν όλα τα ακίνητα που διαχειρίζεται η οποία ξεπερνά το 1.000.000 ευρώ .
- Στ. Δημιουργήστε με SQL ένα ερώτημα που θα προβάλλει τις θέσεις των ακινήτων με τον μεγαλύτερο συντελεστή δόμησης.
- Z. Προβάλετε με SQL τα στοιχεία του μεσίτη που διαχειρίζεται τα περισσότερα ακίνητα.
- H. Προβάλετε με SQL τα στοιχεία του Ιδιοκτήτη που κατέχει τα περισσότερα ακίνητα.
- Θ. Προβάλετε με SQL τα στοιχεία των Ιδιοκτητών με τον αριθμό των ακινήτων που κατέχουν ταξινομημένα κατά φθίνουσα σειρά. Δηλ από τα περισσότερα ακίνητα προς τα λιγότερα.

I. Επαναλάβετε το παραπάνω ερώτημα Θ χρησιμοποιώντας παραγόμενες σχέσεις και όχι όψεις.



Λύση

A.

```
CREATE TABLE ΚΑΤΕΧΟΥΝ (
ΚΙ          INT          NOT NULL,
ΚΑ          INT          NOT NULL,
ΠΟΣΟΣΤΟ   REAL ,
PRIMARY KEY (ΚΙ,ΚΑ),
FOREIGN KEY ΚΙ REFERENCES ΙΔΙΟΚΤΗΤΕΣ(ΚΙ),
FOREIGN KEY ΚΑ REFERENCES ΑΚΙΝΗΤΑ(ΚΑ) )
```

B.

```
SELECT  ΚΑΤΕΧΟΥΝ.ΚΙ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΤΗΛ, ΚΑΤΕΧΟΥΝ. ΚΑ, ΘΕΣΗ,
ΑΝΤ_ΑΞΙΑ, ΣΥΝΤΕΛ_ΔΟΜ, (ΕΚΤΑΣΗ * ΠΟΣΟΣΤΟ) AS ΙΔΙΟΚΤΗΤΗ_ΕΚΤΑΣΗ
FROM      ΙΔΙΟΚΤΗΤΕΣ INNER JOIN
(ΚΑΤΕΧΟΥΝ INNER JOIN  ΑΚΙΝΗΤΑ ON ΚΑΤΕΧΟΥΝ.ΚΑ= ΑΚΙΝΗΤΑ.ΚΑ)
ON ΙΔΙΟΚΤΗΤΕΣ.ΚΙ=ΚΑΤΕΧΟΥΝ.ΚΙ
```

Στο ερώτημα αυτό δίνουμε προσοχή ότι η έκταση που κατέχει ένας ιδιοκτήτης υπολογίζεται από το γινόμενο της ΕΚΤΑΣΗΣ του ακινήτου επί το ΠΟΣΟΣΤΟ που κατέχει. Δηλ. από το 100% της έκτασης ενός ακινήτου 200τμ ένας ιδιοκτήτης με ποσοστό 20% κατέχει $(20 * 200) / 100 = 4000/100 = 40$ τμ.

Γ.

```
SELECT  ΚΑΤΕΧΟΥΝ.ΚΙ, SUM(ΕΚΤΑΣΗ * ΠΟΣΟΣΤΟ) AS ΣΥΝΟΛ_ΙΔ_ΕΚΤΑΣΗ
FROM      ΚΑΤΕΧΟΥΝ INNER JOIN  ΑΚΙΝΗΤΑ
ON ΚΑΤΕΧΟΥΝ.ΚΑ= ΑΚΙΝΗΤΑ.ΚΑ
GROUP BY ΚΑΤΕΧΟΥΝ.ΚΙ
```

Δ.

```
SELECT      ΚΑΤΕΧΟΥΝ.ΚΙ, SUM(ΕΚΤΑΣΗ * ΠΟΣΟΣΤΟ) AS ΣΥΝΟΛ_ΙΔ_ΕΚΤΑΣΗ
FROM        ΚΑΤΕΧΟΥΝ INNER JOIN   ΑΚΙΝΗΤΑ
           ON ΚΑΤΕΧΟΥΝ.ΚΑ= ΑΚΙΝΗΤΑ.ΚΑ
           GROUP BY ΚΑΤΕΧΟΥΝ.ΚΙ
           HAVING SUM(ΕΚΤΑΣΗ * ΠΟΣΟΣΤΟ) >= 10.000
```

Ε.

```
SELECT      ΜΕΣΙΤΕΣ.ΚΜ, SUM(ΑΝΤ_ΑΞΙΑ) AS ΣΥΝΟΛ_ΑΞΙΑ
FROM        ΜΕΣΙΤΕΣ INNER JOIN   ΑΚΙΝΗΤΑ
           ON ΜΕΣΙΤΕΣ.ΚΜ= ΑΚΙΝΗΤΑ.ΚΜ
           GROUP BY ΜΕΣΙΤΕΣ.ΚΜ
           HAVING SUM(ΑΝΤ_ΑΞΙΑ) >= 1.000.000
```

Στ.

```
SELECT      ΘΕΣΗ
FROM        ΑΚΙΝΗΤΑ
           WHERE ΑΝΤ_ΑΞΙΑ IN (SELECT MAX(ΑΝΤ_ΑΞΙΑ)
                              FROM ΑΚΙΝΗΤΑ)
```

Ζ.

Για να υλοποιήσουμε αυτό το ερώτημα θα πρέπει να βρούμε πρώτα πόσα ακίνητα διαχειρίζεται κάθε μεσίτης και μετά να βρούμε ποιος είναι αυτός που έχει τα περισσότερα. Για να το πετύχουμε αυτό θα χρησιμοποιήσουμε δύο όψεις.

```
1. CREATE VIEW V1 AS
SELECT ΚΜ, COUNT(ΚΑ) AS ΠΛΗΘΟΣ1
FROM ΑΚΙΝΗΤΑ
GROUP BY ΚΜ
```

```
2. CREATE VIEW V2 AS
SELECT ΜΕΣΙΤΕΣ.ΚΜ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΤΗΛ, ΠΛΗΘΟΣ1
FROM ΜΕΣΙΤΕΣ INNER JOIN V1 ON ΜΕΣΙΤΕΣ.ΚΜ=V1.ΚΜ
WHERE ΠΛΗΘΟΣ1 IN (SELECT MAX(ΠΛΗΘΟΣ1)
                  FROM V1)
```

Η.

Για να υλοποιήσουμε και αυτό το ερώτημα θα πρέπει να βρούμε πρώτα πόσα ακίνητα κατέχει κάθε ιδιοκτήτης και μετά να βρούμε ποιος είναι αυτός που κατέχει τα περισσότερα. Για να το πετύχουμε αυτό θα χρησιμοποιήσουμε δύο όψεις.

```
1. CREATE VIEW V1 AS
```

```
SELECT KI, COUNT(KA) AS ΠΛΗΘΟΣ2
FROM ΚΑΤΕΧΟΥΝ
GROUP BY KI
```

```
2. CREATE VIEW V2 AS
SELECT ΙΔΙΟΚΤΗΤΕΣ.ΚΙ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΤΗΛ, ΠΛΗΘΟΣ2
FROM ΙΔΙΟΚΤΗΤΕΣ INNER JOIN V1
ON ΙΔΙΟΚΤΗΤΕΣ.ΚΙ=V1.ΚΙ
WHERE ΠΛΗΘΟΣ2 IN (SELECT MAX(ΠΛΗΘΟΣ2)
FROM V1)
```

Θ.

Για να υλοποιήσουμε και αυτό το ερώτημα θα πρέπει να βρούμε πρώτα πόσα ακίνητα κατέχει κάθε ιδιοκτήτης και μετά να ταξινομήσουμε σε φθίνουσα σειρά το πλήθος των ακινήτων. Για να το πετύχουμε αυτό θα χρησιμοποιήσουμε πάλι δύο όψεις.

```
1. CREATE VIEW V1 AS
SELECT KI, COUNT(KA) AS ΠΛΗΘΟΣ2
FROM ΚΑΤΕΧΟΥΝ
GROUP BY KI
```

```
2. CREATE VIEW V2 AS
SELECT ΙΔΙΟΚΤΗΤΕΣ.ΚΙ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΤΗΛ, ΠΛΗΘΟΣ2
FROM ΙΔΙΟΚΤΗΤΕΣ INNER JOIN V1
ON ΙΔΙΟΚΤΗΤΕΣ.ΚΙ=V1.ΚΙ
ORDER BY ΠΛΗΘΟΣ2 DESC
```

Ι.

Για να υλοποιήσουμε και αυτό το ερώτημα θα πρέπει να βρούμε πρώτα πόσα ακίνητα κατέχει κάθε ιδιοκτήτης σε μια παραγόμενη σχέση και μετά να ταξινομήσουμε σε φθίνουσα σειρά το πλήθος των ακινήτων.

ΠΑΡΑΓΟΜΕΝΗ ΣΧΕΣΗ

```
SELECT KI, MAX( ΠΛΗΘΟΣ2)
FROM (SELECT KI, COUNT(KA) AS ΠΛΗΘΟΣ2
```

```
FROM KATECHOYN
GROUP BY KI)
AS NEW_TABLE(KI,ΠΛΗΘΟΣ2)
ORDER BY ΠΛΗΘΟΣ2 DESC
```

Παράδειγμα 5.15

Δημιουργήστε με SQL ένα ερώτημα που θα προβάλει ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΤΗΛ από τους ΙΔΙΟΚΤΗΤΕΣ και τους ΜΕΣΙΤΕΣ που είναι από την πόλη ΣΕΡΡΕΣ

Λύση

Για να δώσει αποτελέσματα σωστά αυτό το ερώτημα θα πρέπει οι δύο αυτοί πίνακες να έχουν όμοια πεδία ίδιου τύπου δεδομένων, ώστε να γίνει η ένωση των συνόλων.

```
(SELECT ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΤΗΛ
FROM ΙΔΙΟΚΤΗΤΕΣ
WHERE ΠΟΛΗ='ΣΕΡΡΕΣ')
UNION
(SELECT ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΤΗΛ
FROM ΜΕΣΙΤΕΣ
WHERE ΠΟΛΗ='ΣΕΡΡΕΣ')
```

2.2 Εισαγωγή HTML

HTML είναι το ακρώνυμο από το **Hyper Text Markup Language** που σημαίνει **γλώσσα χαρακτηρισμού** υπερκειμένου. Η χρήση μιας γλώσσας χαρακτηρισμού σημαίνει ότι γράφεται πρώτα το κείμενο και έπειτα προσθέτονται ειδικά σύμβολα γύρω από τις λέξεις ή από ολόκληρες προτάσεις ώστε να καθοριστεί η εμφάνιση τους στην οθόνη. Τα ειδικά σύμβολα στην HTML λέγονται ετικέτες (tags). Η HTML διαθέτει ένα πεπερασμένο αριθμό ετικετών που μπορούμε να χρησιμοποιήσουμε. Ωστόσο ο αριθμός αυτός δεν παραμένει σταθερός. Κατά διαστήματα το W3 Consortium, το οποίο ανέπτυξε και διαχειρίζεται τα πρότυπα της HTML, δημοσιεύει νέα πρότυπα στα οποία προσθέτει καινούργιες ετικέτες που καλύπτουν ή διορθώνουν μια λειτουργία στο προηγούμενο πρότυπο. Η τελευταία αναθεώρηση του HTML προτύπου είναι η HTML5

Τι είναι οι ετικέτες;

Οι ετικέτες ελέγχουν την δομή και την μορφή του κειμένου της ιστοσελίδας. Επίσης παρέχουν πληροφορίες προς τον web browser για την σελίδα που πρόκειται να εμφανίσουν, όπως ο τίτλος της σελίδας ή ο συγγραφέας της, κ.α.

Πώς είναι μια ετικέτα ;

Οι HTML ετικέτες γράφονται ανάμεσα στα σύμβολα < και > π.χ. <όνομα-ετικέτας>
Οι περισσότερες HTML ετικέτες αποτελούνται από μια ετικέτα αρχής και μια ετικέτα τέλους και ανάμεσα σε αυτές υπάρχει το κείμενο που χαρακτηρίζεται από τις ετικέτες αυτές. Η ετικέτα τέλους περιέχει τον χαρακτήρα /πριν το όνομα της ετικέτας.
π.χ. <όνομα-ετικέτας> ... κείμενο ... </όνομα-ετικέτας>
π.χ. Κείμενο με έμφαση
Υπάρχουν ορισμένες ετικέτες που δεν έχουν ετικέτες τέλους. Στις ετικέτες αυτές, πριν από το σύμβολο >τοποθετούμε τον χαρακτήρα /
π.χ. <όνομα-ετικέτας/>
π.χ.

Το αρχείο που περιέχει HTML ετικέτες λέγεται HTML αρχείο και έχει επέκταση .html ή .htm (εκτός βέβαια κι' αν η ιστοσελίδα είναι δυναμική οπότε έχει επεκτάσεις όπως .php, asp, jsp κτλ.). Τα αρχεία αυτά είναι απλά αρχεία κειμένου σε μορφή ASCII και δεν περιέχουν πληροφορίες για το περιβάλλον ή τα προγράμματα με τα οποία θα λειτουργήσουν. Μπορείτε να ανοίξετε και να δείτε τα αρχεία htm ή html με οποιονδήποτε επεξεργαστή κειμένου, π.χ. Σημειωματάριο (Notepad) των Windows

Πώς λειτουργεί μια ετικέτα μέσα σε ένα HTML αρχείο ;

Ένα ζευγάρι ετικετών καθορίζει την μορφή ενός κειμένου ή μιας και μόνο λέξης. Για παράδειγμα αν ήθελα να γράψω στην γλώσσα HTML το κείμενο 'Μαθαίνω HTML' με έντονη γραφή, θα το έγραφα έτσι: Μαθαίνω HTML. Μόλις συναντήσαμε την πρώτη HTML ετικέτα. Το γράμμα **b** είναι το αρχικό από την λέξη bold. Το ζευγάρι των ετικετών και , δηλώνει στον web browser, ότι το κείμενο που είναι γραμμένο μεταξύ των ετικετών αυτών θα εμφανιστεί με έντονα (bold) γράμματα. Αν πάλι ήθελα να γράψω την λέξη wlearn με πλάγιους χαρακτήρες θα έγραφα: <i>wlearn</i>

Το ζευγάρι των ετικετών <i> και </i>, λέει στον browser να εμφανίσει το κείμενο που είναι γραμμένο ανάμεσα σε αυτές τις ετικέτες, με πλάγιους χαρακτήρες. Το γράμμα **i** είναι το πρώτο γράμμα από την λέξη italics.

Οι ετικέτες μέσα σε ένα HTML αρχείο μπορούν να είναι γραμμένες είτε με μικρά γράμματα (πεζά), είτε με κεφαλαία. Η ετικέτα είναι ίδια με τη ετικέτα . Μόνο πρέπει να προσέχουμε τις ετικέτες τέλους να είναι γραμμένες όπως η ετικέτες αρχής. Δηλαδή αν μια ετικέτα αρχής είναι γραμμένη με πεζά τότε και η ετικέτα τέλους πρέπει να γραφεί με πεζά.

Λάθος τρόπος:
wlearn

Σωστός τρόπος:

wlearn

ή

wlearn

2.3 Εισαγωγή PHP

Η PHP είναι μια ισχυρή γλώσσα script η οποία χρησιμοποιείται για τη δημιουργία δυναμικών και διαδραστικών διαδικτυακών εφαρμογών.

Η PHP είναι πολύ διαδεδομένη, δοκιμασμένη και ιδανική για ανάπτυξη εφαρμογών στο web.

Σχετικά με τη σύνταξη της γλώσσας, είναι παρόμοια της perl και της C ή αν προτιμάτε της Java και της JavaScript.

Η PHP κατά κανόνα τρέχει επάνω σε έναν Apache server ο οποίος με τη σειρά του μπορεί να είναι εγκατεστημένος σε ένα λειτουργικό όπως Unix, Linux, Windows κλπ.

Ένα αρχείο PHP περιέχει συνήθως ένα συνδυασμό από PHP κώδικα και html κώδικα. Ο κώδικας PHP εκτελείται στον server, αντίθετα ο κώδικας html αποστέλεται στον client ανέπαφος.

Τι πρέπει να γνωρίζετε

Για να συνεχίσετε με την PHP θα πρέπει να έχετε βασική γνώση των παρακάτω:

- HTML / XHTML
- Βασική προγραμματιστική γνώση σε γλώσσα προγραμματισμού ή σε script (όπως javascript ή κάτι σχετικό)

Χαρακτηριστικά της PHP

- Το ακρωνύμιο PHP προέρχεται από τις λέξεις: Hypertext Preprocessor.
- Η PHP είναι μία γλώσσα script που τρέχει στη μεριά του διακομιστή (server-side), όπως η jsp και η asp.
- Η PHP υποστηρίζει πολλά συστήματα διαχείρισης βάσεων δεδομένων όπως: (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, κλπ.).
- Η PHP είναι ένα λογισμικό ανοικτού κώδικα (Open Source Software ή OSS).
- Μπορείτε να κατεβάσετε και να χρησιμοποιήσετε την PHP δωρεάν και χωρίς καμία δέσμευση.
- Τα αρχεία PHP έχουν συνήθως την προέκταση .php αλλά μπορεί να έχουν και .php3 ή .phtml.

Τι είναι η MySQL?

- Η MySQL είναι ένα σύστημα διαχείρισης βάσεων δεδομένων
- Η MySQL είναι ιδανική για μικρές και μεγάλες εφαρμογές.
- Η MySQL υποστηρίζει την πρότυπη γλώσσα SQL.
- Η MySQL μπορεί να στηθεί σε διαφορετικές πλατφόρμες.
- Μπορείτε να κατεβάσετε και να χρησιμοποιήσετε την MySQL δωρεάν και χωρίς καμία δέσμευση.

PHP + MySQL

- Η PHP σε συνδυασμό με την MySQL αποτελούν το πιο δημοφιλές ζευγάρι για ανάπτυξη εφαρμογών στο διαδίκτυο.

Γιατί PHP

- Η PHP μπορεί να τρέξει σε διαφορετικές πλατφόρμες (Windows, Linux, Unix, κλπ.)
- Η PHP είναι συμβατή με του περισσότερους διακομιστές (Apache, IIS, κλπ.)
- Η PHP είναι δωρεάν και ανοικτού κώδικα (μπορείτε να κατεβάσετε και τον κώδικα από τη διεύθυνση: www.php.net)
- Η PHP είναι εύκολη ως γλώσσα για να τη μάθει κάποιος και αποδοτική όταν τρέχει σε ένα διακομιστή.

Από που να αρχίσετε

Για να αρχίσετε να δουλεύετε με την PHP θα πρέπει να έχετε εγκαταστήσει το κατάλληλο περιβάλλον το οποίο συνήθως είναι:

1. Εγκατεστημένος Apache server σε Windows ή Linux
2. Εγκατεστημένη PHP σε Windows ή Linux
3. Εγκατεστημένη MySQL σε Windows ή Linux

LAMP, WAMP ή XAMP

Συνήθως το περιβάλλον στο οποίο μπορεί κάποιος να εργαστεί είναι LAMP (Linux, Apache, MySql, Php) ή WAMP (Windows, Apache, MySql, Php). Αν η πλατφόρμα σας είναι Windows, τότε μπορείτε να κατεβάσετε και να στήσετε όλο το περιβάλλον από τη διεύθυνση: <http://www.wampserver.com/>

2.4 Εισαγωγή CSS

Τα CSS (Cascading Style Sheets), είναι ένα σύνολο από ορίσματα, διαφορετικά από αυτά που συναντάμε στην HTML, τα οποία αναπτύχθηκαν με στόχο την καλύτερη διαχείριση της εμφάνισης των ιστοσελίδων.

Μπορούμε μέσα από τα CSS, να ορίσουμε γραμματοσειρές, χρώματα, στοίχιση, backgrounds, κ.λ.π. Ουσιαστικά αποτελούν κλάσεις αντικειμένων τα οποία παρέχουν μεθόδους για υλοποίηση σε κάποια tags της HTML. Για παράδειγμα μία κλάση CSS μπορεί να είναι η `main`, η οποία ορίζει ότι η γραμματοσειρά που θα χρησιμοποιηθεί είναι η Verdana, μεγέθους 11, bold και το κείμενο θα είναι στοιχισμένο στο κέντρο.

Για να εφαρμόσουμε αυτή την κλάση και τις ιδιότητες – μεθόδους της μέσα στην HTML, χρησιμοποιούμε εντός όλων σχεδόν των tags της HTML που αφορούν σε εμφάνιση κειμένου, πινάκων, γραμμών ή στηλών πινάκων, links, φορμών κ.λ.π. την παράμετρο `class="όνομα κλάσης CSS"`.

Για παράδειγμα αν θέλουμε σε μία παράγραφο να εφαρμόσουμε τις ιδιότητες της κλάσης `main` που αναφέραμε παραπάνω πρέπει να γράψουμε:

```
<p class="main">Το κείμενό μας</p>
```

Τα ορίσματα των CSS, μπορούν να εισαχθούν ως κώδικας τόσο μέσα στην HTML, όσο και σε ξεχωριστό αρχείο με κατάληξη .css το οποίο στη συνέχεια αναφέρεται ως link στην HTML.

Εισαγωγή ορισμάτων CSS

Για να ορίσετε ότι σε μία σελίδα HTML, θα εισαχθεί κώδικας CSS πρέπει εντός της HTML και μέσα στο tag <HEAD> να εισάγετε τα εξής:

```
<style type="text/css">
```

```
<!--
```

```
Ακολουθεί ο κώδικας των CSS.
```

```
-->
```

```
</style>
```

Στην περίπτωση που ο κώδικας των CSS, έχει γραφτεί σε ξεχωριστό αρχείο με κατάληξη .css τότε αντίστοιχα πρέπει εντός της HTML και μέσα στο tag <HEAD> να εισάγετε τα εξής:

<link href=" διαδρομή και όνομα αρχείου.css " rel="stylesheet" type="text/css">

Κλάσεις CSS

Παραπάνω αναφέραμε ότι τα CSS, αποτελούνται από κλάσεις και ιδιότητες αυτών των κλάσεων. Ένα παράδειγμα μπορεί να είναι το παρακάτω το οποίο έχει δύο κλάσεις τις main και small οι οποίες ορίζουν διαφορετικές ιδιότητες.

```
.main {  
font-family: Verdana, Tahoma, Arial, Helvetica, Sans-serif;  
font-size: 12px;  
color: #FFFFFF;  
line-height: 110%;  
list-style-type: disc;  
}  
.small {  
font-family: Verdana, Tahoma, Arial, Helvetica, Sans-serif;  
font-size: 9px;  
color: #333333;  
}
```

Κοινώς για να ορίσετε μία κλάση CSS, εντός ενός αρχείου ή εντός της ίδιας της HTML σελίδας, όπως αναφέρθηκε παραπάνω πρέπει να εισάγετε τα εξής:

```
.όνομα κλάσης {  
ιδιότητες κλάσης  
}
```

Ιδιότητες κλάσεων CSS

Οι ιδιότητες κλάσεων ορίζονται ως εξής:

Ιδιότητα : Παράμετροι Ιδιότητας ;

Παρακάτω αναφέρονται οι πιο συνηθεις ιδιότητες κλάσεων που χρησιμοποιούνται στα CSS.

font-family: Verdana, Tahoma, Arial, Helvetica, Sans-serif;

Αναφέρεται στη γραμματοσειρά που θα χρησιμοποιηθεί. Παράμετροι είναι τα ονόματα των γραμματοσειρών που θα χρησιμοποιηθούν για την εμφάνιση κειμένου με προτεραιότητα από αριστερά προς δεξιά.

font-size: 10px;

Το μέγεθος της γραμματοσειράς.

font-style: italic;

Το στυλ της γραμματοσειράς. Παράμετροι είναι normal,italic,oblique.

font-weight: bold;

Το πόσο έντονη θα είναι η γραμματοσειρά. Παράμετροι είναι normal, bold, bolder, lighter.

text-decoration: underline overline line-through blink ή none;

Η διακόσμηση του κειμένου. Οι παράμετροι όπως φαίνονται στα δεξιά του ορίσματος διακοσμούν ο καθένας διαφορετικά το κείμενο στο οποίο εφαρμόζονται π.χ. overline δημιουργεί μία γραμμή πάνω από το κείμενο.

text-align: left;

Η στοίχιση του κειμένου. Παράμετροι είναι left, right, center, justify.

color: #006699;

Το χρώμα της γραμματοσειράς που θα χρησιμοποιηθεί.

background-color: #003366;

Το χρώμα του φόντου στο σημείο που θα χρησιμοποιηθεί η κλάση.

background-image: url(διαδρομή και όνομα αρχείου);

Η εικόνα που θα χρησιμοποιηθεί ως φόντο στο σημείο που θα χρησιμοποιηθεί η κλάση.

background-repeat: repeat;

Αν χρησιμοποιηθεί εικόνα ως φόντο, αν η εικόνα θα επαναλαμβάνεται ή όχι για να επικαλύψει όλη ή όχι την περιοχή όπου εφαρμόζεται η κλάση. Για παράδειγμα σε ένα πίνακα μήκους 500 pixels αν εφαρμοστεί η κλάση που περιέχει αυτή την ιδιότητα και η παράμετρος της είναι no-repeat τότε η εικόνα θα εμφανιστεί ως φόντο μία μόνο φορά. Παράμετροι είναι repeat, no-repeat, repeat-x, repeat-y.

background-attachment: fixed;

Αν χρησιμοποιηθεί εικόνα ως φόντο, αν η εικόνα θα μετακινείται όσο ο χρήστης μετακινεί την σελίδα πάνω κάτω ή όχι. Παράμετροι είναι fixed και scroll.

background-position: left top;

Αν χρησιμοποιηθεί εικόνα ως φόντο σε πιο σημείο της περιοχής που εφαρμόζεται η κλάση θα τοποθετηθεί το φόντο. Παράμετροι είναι Left, center, right ή απόσταση σε pixels για την οριζόντια τοποθέτηση και top, center, bottom ή απόσταση σε pixels για την κάθετη τοποθέτηση. Οι παραπάνω είναι οι πιο συνηθισμένες ιδιότητες κλάσεων. Για μία πλήρη λίστα των ιδιοτήτων των CSS αναφερθείτε στην τυποποίησή τους στο <http://www.w3c.org>

Τα CSS (Cascading Style Sheets), είναι ένα σύνολο από ορίσματα, διαφορετικά από αυτά που συναντάμε στην HTML, τα οποία αναπτύχθηκαν με στόχο την καλύτερη διαχείριση της εμφάνισης των ιστοσελίδων.

Μπορούμε μέσα από τα CSS, να ορίσουμε γραμματοσειρές, χρώματα, στοίχιση, backgrounds, κ.λ.π. Ουσιαστικά αποτελούν κλάσεις αντικειμένων τα οποία παρέχουν μεθόδους για υλοποίηση σε κάποια tags της HTML.

Για παράδειγμα μία κλάση CSS μπορεί να είναι η main, η οποία ορίζει ότι η γραμματοσειρά που θα χρησιμοποιηθεί είναι η Verdana, μεγέθους 11, bold και το κείμενο θα είναι στοιχισμένο στο κέντρο.

Για να εφαρμόσουμε αυτή την κλάση και τις ιδιότητες – μεθόδους της μέσα στην HTML, χρησιμοποιούμε εντός όλων σχεδόν των tags της HTML που αφορούν σε εμφάνιση κειμένου, πινάκων, γραμμών ή στηλών πινάκων, links, φορμών κ.λ.π. την παράμετρο class="όνομα κλάσης CSS".

Για παράδειγμα αν θέλουμε σε μία παράγραφο να εφαρμόσουμε τις ιδιότητες της κλάσης main που αναφέραμε παραπάνω πρέπει να γράψουμε:

```
<p class="main">Το κείμενό μας</p>
```

Τα ορίσματα των CSS, μπορούν να εισαχθούν ως κώδικας τόσο μέσα στην HTML, όσο και σε ξεχωριστό αρχείο με κατάληξη .css το οποίο στη συνέχεια αναφέρεται ως link στην HTML.

Εισαγωγή ορισμάτων CSS

Για να ορίσετε ότι σε μία σελίδα HTML, θα εισαχθεί κώδικας CSS πρέπει εντός της HTML και μέσα στο tag <HEAD> να εισάγετε τα εξής:

```
<style type="text/css">
```

```
<!--
```

Ακολουθεί ο κώδικας των CSS.

```
-->
```

```
</style>
```

Στην περίπτωση που ο κώδικας των CSS, έχει γραφτεί σε ξεχωριστό αρχείο με κατάληξη .css τότε αντίστοιχα πρέπει εντός της HTML και μέσα στο tag <HEAD> να εισάγετε τα εξής:

```
<link href=" διαδρομή και όνομα αρχείου.css " rel="stylesheet" type="text/css">
```

Κλάσεις CSS

Παραπάνω αναφέραμε ότι τα CSS, αποτελούνται από κλάσεις και ιδιότητες αυτών των κλάσεων.

Ένα παράδειγμα μπορεί να είναι το παρακάτω το οποίο έχει δύο κλάσεις τις main και small οι οποίες ορίζουν διαφορετικές ιδιότητες.

```
.main {  
font-family: Verdana, Tahoma, Arial, Helvetica, Sans-serif;  
font-size: 12px;  
color: #FFFFFF;  
line-height: 110%;  
list-style-type: disc;  
}
```

```
.small {  
font-family: Verdana, Tahoma, Arial, Helvetica, Sans-serif;  
font-size: 9px;  
color: #333333;  
}
```

Κοινώς για να ορίσετε μία κλάση CSS, εντός ενός αρχείου ή εντός της ίδιας της HTML σελίδας, όπως αναφέρθηκε παραπάνω πρέπει να εισάγετε τα εξής:

```
.όνομα κλάσης {  
ιδιότητες κλάσης  
}
```

Ιδιότητες κλάσεων CSS

Οι ιδιότητες κλάσεων ορίζονται ως εξής:

Ιδιότητα : Παράμετροι Ιδιότητας ;

Παρακάτω αναφέρονται οι πιο συνήθεις ιδιότητες κλάσεων που χρησιμοποιούνται στα CSS.

font-family: Verdana, Tahoma, Arial, Helvetica, Sans-serif;

Αναφέρεται στη γραμματοσειρά που θα χρησιμοποιηθεί. Παράμετροι είναι τα ονόματα των γραμματοσειρών που θα χρησιμοποιηθούν για την εμφάνιση κειμένου με προτεραιότητα από αριστερά προς δεξιά.

font-size: 10px;

Το μέγεθος της γραμματοσειράς.

font-style: italic;

Το στυλ της γραμματοσειράς. Παράμετροι είναι normal, italic, oblique.

font-weight: bold;

Το πόσο έντονη θα είναι η γραμματοσειρά. Παράμετροι είναι normal, bold, bolder, lighter.

text-decoration: underline overline line-through blink ή none;

Η διακόσμηση του κειμένου. Οι παράμετροι όπως φαίνονται στα δεξιά του ορίσματος διακοσμούν ο καθένας διαφορετικά το κείμενο στο οποίο εφαρμόζονται π.χ. overline δημιουργεί μία γραμμή πάνω από το κείμενο.

text-align: left;

Η στοίχιση του κειμένου. Παράμετροι είναι left, right, center, justify.

color: #006699;

Το χρώμα της γραμματοσειράς που θα χρησιμοποιηθεί.

background-color: #003366;

Το χρώμα του φόντου στο σημείο που θα χρησιμοποιηθεί η κλάση.

background-image: url(διαδρομή και όνομα αρχείου);

Η εικόνα που θα χρησιμοποιηθεί ως φόντο στο σημείο που θα χρησιμοποιηθεί η κλάση.

background-repeat: repeat;

Αν χρησιμοποιηθεί εικόνα ως φόντο, αν η εικόνα θα επαναλαμβάνεται ή όχι για να επικαλύψει όλη ή όχι

την περιοχή όπου εφαρμόζεται η κλάση. Για παράδειγμα σε ένα πίνακα μήκους 500 pixels αν εφαρμοστεί η κλάση που περιέχει αυτή την ιδιότητα και η παράμετρος της είναι no-repeat τότε η εικόνα θα εμφανιστεί ως φόντο μία μόνο φορά. Παράμετροι είναι repeat, no-repeat, repeat-x, repeat-y.

background-attachment: fixed;

Αν χρησιμοποιηθεί εικόνα ως φόντο, αν η εικόνα θα μετακινείται όσο ο χρήστης μετακινεί την σελίδα πάνω κάτω ή όχι. Παράμετροι είναι fixed και scroll.

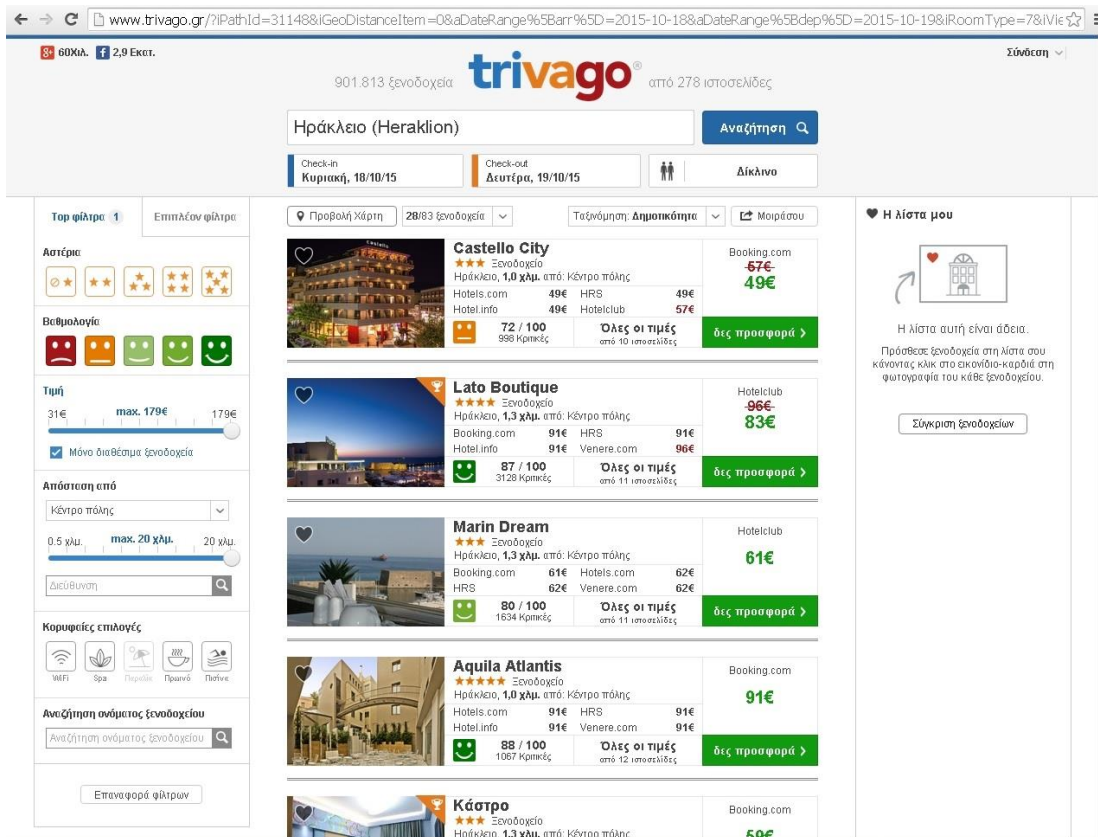
background-position: left top;

Αν χρησιμοποιηθεί εικόνα ως φόντο σε πιο σημείο της περιοχής που εφαρμόζεται η κλάση θα

τοποθετηθεί το φόντο. Παράμετροι είναι Left, center, right ή απόσταση σε pixels για την οριζόντια τοποθέτηση και top, center, bottom ή απόσταση σε pixels για την κάθετη τοποθέτηση. Οι παραπάνω είναι οι πιο συνηθεις ιδιότητες κλάσεων. Για μία πλήρη λίστα των ιδιοτήτων των CS αναφερθείτε στην τυποποίησή τους στο <http://www.w3c.org>

3 Μηχανές αναζήτησης και ιστοσελίδες κρατήσεων

3.1 Trivago



Η trivago είναι μια μηχανή αναζήτησης ξενοδοχειακών τιμών και πληροφοριών. Η ιστοσελίδα συγκρίνει τις τιμές 600.000 ξενοδοχείων, προερχόμενων από περισσότερες από 100 ιστοσελίδες ξενοδοχειακών κρατήσεων, όπως για παράδειγμα την booking.com και η hotels.com. Τα κεντρικά γραφεία της εταιρείας βρίσκονται στο Ντίσελντορφ της Γερμανίας και η ιστοσελίδα φιλοξενεί 18 εκατομμύρια χρήστες κάθε μήνα, στις 47 διεθνείς πλατφόρμες της.

Η ιδέα της trivago γεννήθηκε το 2004 στο Ντίσελντορφ της Γερμανίας, από τους τρεις ιδρυτές της Ρολφ Σρέμγκενς (Rolf Schromgens), Μαλτ Ζίβερτ (Malte Siewert) και Πίτερ Βιννεμάιερ (Peter Vinnemeier) και η πρώτη έκδοση της ιστοσελίδας ήταν στον αέρα το 2005. Το 2007 ξεκίνησε να λειτουργεί η trivago Ισπανίας, Γαλλίας και Αγγλίας, ενώ έναν χρόνο αργότερα λανσάρονται οι πλατφόρμες της Πολωνίας και της Σουηδίας. Η Ελληνική πλατφόρμα της trivago είναι στον αέρα από το 2008. Το 2009, ήταν διαθέσιμες και οι πρώτες μη ευρωπαϊκές πλατφόρμες σε άλλες χώρες όπως στις Ηνωμένες Πολιτείες, Κίνα, Ιαπωνία, Βραζιλία και Μεξικό.

Η trivago συγκρίνει σε πραγματικό χρόνο τιμές ξενοδοχείων για τον επιλεγμένο, από τον χρήστη, προορισμό, αναζητώντας σε πάνω από 100 διαφορετικές ιστοσελίδες ξενοδοχειακών κρατήσεων πληροφορίες όπως κριτικές, φωτογραφίες, περιγραφές και τιμές. Ο κάθε επισκέπτης μπορεί να βρει το ιδανικό για τις ανάγκες του ξενοδοχείο, βασισμένος στη σύγκριση τιμών αλλά και στις εντυπώσεις άλλων χρηστών που το έχουν ήδη επισκεφτεί. Η σύγκριση ξενοδοχειακών τιμών παρέχει πληροφορίες προερχόμενες από διάφορες πηγές, όπως ιστοσελίδες ξενοδοχειακών κρατήσεων, χρήστες και ξενοδόχους. Αυτές οι πληροφορίες είναι έτσι δομημένες, ώστε να παρέχουν στον χρήστη την καλύτερη δυνατή εικόνα κάθε ξενοδοχείου.

Μια ακόμα υπηρεσία που προσφέρεται από την trivago είναι το trivago Hotel Manager και απευθύνεται στους ιδιοκτήτες ξενοδοχείων. Παρέχεται χωρίς οικονομική επιβάρυνση και επιτρέπει στους ξενοδόχους να βελτιώσουν το ξενοδοχειακό τους προφίλ, να διαχειριστούν κριτικές, σχόλια και βαθμολογίες πελατών, ενισχύοντας έτσι τη συνολική τους παρουσία στην trivago και, συνεπώς, τις κρατήσεις τους. Η trivago οργανώνει ανά τακτά χρονικά διαστήματα webinars και άλλα ενημερωτικά προγράμματα για την εκπαίδευση των ξενοδόχων στη χρήση των διάφορων εργαλείων του trivago Hotel Manager.

3.2 Hotelz

The screenshot shows the Hotelz.gr search results page for Heraklion, Greece. The search criteria are: 1 week (03 Nov - 04 Nov) for 2 adults. The results are sorted by 'Price' (Ταξινόμηση κατά: Προτεινόμενα, Αστέρια, Απόσταση από, Τιμή, Βαθμολογία επισκεπτών).

Hotelz.gr Search Results:

- Κατηγορίες:** Ηπειρωτική Ελλάδα / Ελλάδα / Ελληνικά Νησιά / Κρήτη / Ηράκλειο / Αναζήτηση Αποτελεσμάτων (24.409, 17.250, 4.366, 55, 21 διαθέσιμα)
- Αναζήτηση ξενοδοχείων:**
 - Περιοχή/Πόλη: Ηράκλειο, Ελλάδα
 - Άφιξη: Τρι 3, Νοέμβριος 2015
 - Αναχώρηση: Τετ 4, Νοέμβριος 2015
 - Επισκέπτες: 2 ενήλικες σε 1 δωμάτιο
- Βελτίωση αναζήτησης:**
 - Αστέρια: 2 (★), 9 (★★), 6 (★★★), 1 (★★★★), 3 (★★★★★)
- Hotel Listings:**
 - Kastro Hotel Ηράκλειο** (3★):
 - Double Or Twin Standard - Breakfast Included: €47 (Προβόλη Προσφοράς)
 - Standard Double Room: €50
 - Standard Double Room: €51
 - Διπλό, 1 Ή 2 Κρεβάτια Standard...: €52
 - Triple Standard - Breakfast Included ΔΩΡΕΑΝ σκόρπωση, Πρωινό περιλαμβάνεται: €64
 - Ξενοδοχείο ΕΛ Γκρέκο Ηράκλειο** (2★):
 - Economy Δωμάτιο: €32 (Έχουν μείνει 2 δωμάτια)
 - Economy Room: €32 (Προβόλη Προσφοράς)
 - Economy Room: €32 (Έχουν μείνει 2 δωμάτια)
 - Standard Δωμάτιο ΔΩΡΕΑΝ σκόρπωση, Πρωινό περιλαμβάνεται: €42 (Έχουν μείνει 3 δωμάτια)

Το Hotelz.gr είναι μία εξελιγμένη μηχανή αναζήτησης η οποία επικοινωνεί με τις μεγαλύτερες εταιρίες Online κρατήσεων δωματίων και μπορεί εύκολα και άμεσα να παρουσιάσει τα αποτελέσματα στους χρήστες του. Η αναζήτηση των τιμών – δωματίων γίνεται σε πραγματικό χρόνο και η κράτηση σας είναι γρήγορη και άμεση.

Στο Hotelz.gr μπορείτε να αναζητήσετε δωμάτιο στον αγαπημένο σας προορισμό σε οποιοδήποτε μέρος του κόσμου και αν βρίσκεται αυτός, να δείτε φωτογραφίες του δωματίου – ξενοδοχείου, την τιμή αλλά και να κάνετε την κράτηση σας σε πραγματικό χρόνο.

Το Hotelz.gr ΔΕΝ είναι ταξιδιωτικό γραφείο. Οι κρατήσεις των δωματίων πραγματοποιούνται μέσω των εταιριών που τα διαχειρίζονται. Έτσι την στιγμή που θα επιλέξετε το δωμάτιο που θέλετε θα μεταφερθείτε στο site της εταιρίας και θα ολοκληρώσετε την συναλλαγή σας γρήγορα και φυσικά με ασφάλεια.

3.3 MegaTravel

search.megatravel.gr/Hotels/Search?destination=place%3dHeraklion&resultID=0&languageCode=EL&checkin=2015-09-30&checkout=2015-10-01&Rooms

Ελληνικά English

megatravel.gr Αρχική σελίδα Αναζήτηση Συχνές ερωτήσεις Επικοινωνία

Κεντρική σελίδα / Ελλάδα / Ελληνικά Νησιά / Κρήτη / Ηράκλειο / Αναζήτηση Αποσιλεκσμάτων

24.409 17.250 4.366 55 21 δωμάτια

Αναζήτηση ξενοδοχείων

Περιοχή/Πόλη
Ηράκλειο, Ελλάδα

Αφιή
Τετ 30 Σεπτέμβριος 2015

Αναχώρηση
Πεμ 1 Οκτώβριος 2015

Δεν έχω συγκεκριμένες ημερομηνίες ακούμνη

Επισκέπτες
2 ενήλικες σε 1 δωμάτιο

Αναζήτηση

Βελτίωση αναζήτησης

Αστέρις
★ 3
★★ 11
★★★ 4
★★★★ 0
★★★★★ 3

Τιμή
€ - €39 6
€40 - €89 12

Ηράκλειο
1 βράδυ (30 Σεπ - 01 Οκτ) για 2 ενήλικες [Άλλεγή αναζήτησης](#)

Τελευταίωμα κατά Προτεινόμενο Αστέρις Απόσταση από Τιμή Βαθμολογία επισκεπών

Ξενοδοχείο ΕΛ Γκρέκο Ηράκλειο ★★

Ηράκλειο, Ελλάδα Προβολή χάρτη

Economy Room Hotels.com	€36	Έχει μείνει 1 δωμάτιο
Economy Δωμάτιο Venere.com	€36	Έχει μείνει 1 δωμάτιο
Standard Room Hotels.com	€42	Έχουν μείνει 3 δωμάτια
Standard Δωμάτιο Venere.com	€42	Έχουν μείνει 3 δωμάτια
Διπλό, 1 Ή 2 Κρεβάτια Hotels.gr	€57	Έχουν μείνει 3 δωμάτια

Σύγκριση 3 ιστοτόπων

5.8
Σύμφωνα με 251 κριτικές

Προβολή Προσφοράς
Venere.com

Διαβάστε περισσότερα

Lato Boutique Hotel Ηράκλειο

Ηράκλειο, Ελλάδα Προβολή χάρτη

Φανταστικό, 8,7
Σύμφωνα με 1.276 κριτικές

"Εμπορικά καταστήματα", "Υπέροχη θέα", "Εξαιρετικό φαγητό"

Twin/Double/Double Room Περιλαμβάνεται πρωινό Otel.com	€74	
Standard Double Room - Non-Refundable Rate Περιλαμβάνεται πρωινό Prestigia.com	€79	Έχει μείνει 1 δωμάτιο
Double - Standard Room - Non Refundable Περιλαμβάνεται πρωινό HotelsClick.com	€89	Έχει μείνει 1 δωμάτιο

Προβολή Προσφοράς

Το MegaTravel.gr έχει δημιουργήσει μια από τις καλύτερες μηχανές αναζήτησης σύγκρισης τιμών ξενοδοχείων στον κόσμο με περισσότερες 900.000 προσφορές ξενοδοχείων σε 20.000 προορισμούς σε 195 χώρες.

3.4 Viva

The screenshot shows the VIVA.gr website interface. At the top, there's a navigation bar with categories like 'Αεροπορικά', 'Ακτοπορικά', 'Θεάματα', 'Αθλητικά', 'Ξενοδοχεία', 'Τηλεφωνία', 'Ασφάλειες', and 'Πληρωμές'. The main content area is titled 'Ξενοδοχεία σε Κρήτη (GREECE)'. It shows search criteria: 'Check in: 30 Σεπτεμβρίου 2015 | Check Out: 03 Οκτωβρίου 2015' and 'Δωμάτιο 1: 2 ενήλικες'. A filter sidebar on the left lists various hotels with their room counts. The main results show two hotels: Galaxy Hotel Iraklio (5 stars, 95.07 value score) and Iraklion (2 stars, 69.52 value score). Each hotel card includes a photo, location, amenities (like NO-SMOKE-AREA, GYM, WI-FI, BREAKFAST), and pricing for 3 and 1 nights.

Το Website έχει αναπτυχθεί και λειτουργεί με σκοπό την παροχή από το VIVA προς τους χρήστες του website της δυνατότητας λήψης των ακόλουθων υπηρεσιών/αγαθών:

Τη συλλογή πληροφοριών σχετικά με τουριστικά αγαθά, όπως οι υπηρεσίες μεταφοράς από/προς και διαμονής σε διάφορους προορισμούς παγκοσμίως (πχ ξενοδοχείων και καταλυμάτων εν γένει), η διαθεσιμότητα εισιτηρίων, κλινών ή/και οχημάτων και η κράτηση ή/και αγορά των αγαθών που παρέχουν Τρίτοι Πάροχοι.

Τη λήψη υπηρεσιών Μεταφοράς που παρέχουν Τρίτοι Πάροχοι (όπως ενδεικτικά, μεταφορά με Ταξί, Λεωφορεία ΚΤΕΛ κ.ο.κ.).

Τη συλλογή πληροφοριών σχετικά με αγαθά ψυχαγωγίας, όπως ενδεικτικά, πολιτιστικά θεάματα (π.χ. συναυλίες, θεατρικές παραστάσεις, προβολές κινηματογραφικών ταινιών κ.λπ.), αθλητικά θεάματα κ.ά., έλεγχος διαθεσιμότητας εισιτηρίων και κράτηση ή/και αγορά εισιτηρίων των θεαμάτων αυτών που παρέχουν Τρίτοι Πάροχοι.

Την προμήθεια κωδικών ανανέωσης χρόνου ομιλίας κινητής τηλεφωνίας.

Την επιλογή ασφαλιστικών προϊόντων κλάδου ασφάλισης οχημάτων των συνεργαζόμενων με το VIVA ασφαλιστικών επιχειρήσεων.

Την απόκτηση δυνατότητας χρήσης ενός συνόλου καινοτόμων τηλεπικοινωνιακών υπηρεσιών.

Η διάθεση των εν λόγω υπηρεσιών από το VIVA πραγματοποιείται ή/και ολοκληρώνεται από το σύνολο των καναλιών του δικτύου του, δηλαδή είτε από το website είτε μέσω άλλων σημείων παρουσίας του, φυσικών ή μη, ιδιόκτητων ή συνεργατών και υποπρακτόρων του, σύμφωνα με τις οδηγίες, τους όρους και προϋποθέσεις που γνωστοποιούνται στο χρήστη κατά την πραγματοποίηση της αγοράς του.

Μέρος των ανωτέρω υπηρεσιών παρέχεται και μέσω εφαρμογής κινητού τηλεφώνου (Viva Wallet), σύμφωνα με τις εκάστοτε οδηγίες που γνωστοποιούνται στο χρήστη κατά την εγγραφή του στην εφαρμογή, στην οποία αντίστοιχα εφαρμόζονται αναλογικά οι όροι του παρόντος.

3.5 Booking

The screenshot shows the Booking.com search results page for Heraklion, Crete. The search criteria include the destination 'Ηράκλειο Πόλη', dates from 29 Sep to 30 Sep, and 2 adults. The results list three hotels:

- Ξενοδοχείο Κάστρο** (Kastros Hotel): 3 stars, 976 reviews, 8.9 rating. Description: Located in an exceptional location for archaeological sites, 100m from the city center. Price: €27.
- Iraklion Hotel**: 2 stars, 418 reviews, 8.2 rating. Description: Modern family-run hotel with Wi-Fi, parking, and 24-hour reception. Price: €27.
- GDM Megaron Hotel**: 5 stars, 800 reviews, 9.5 rating. Description: Historic hotel from 1920, near the Old Harbor. Price: €27.

Η Booking.com B.V., μέλος του Priceline Group (Nasdaq: PCLN), κατέχει και διαχειρίζεται την Booking.com™, την κορυφαία εταιρεία online κρατήσεων καταλυμάτων στον κόσμο. Κάθε μέρα γίνονται κρατήσεις για περισσότερες από 900.000 διανυκτερεύσεις μέσω της Booking.com. Ο ιστοχώρος της Booking.com και οι εφαρμογές προσελκύουν επισκέπτες τόσο από την αγορά τουρισμού όσο και των επιχειρήσεων παγκοσμίως.

Από το 1996, η Booking.com B.V. εγγυάται τις καλύτερες τιμές για κάθε τύπο καταλύματος, από μικρά, οικογενειακά διοικούμενα bed and breakfast μέχρι πολυτελή διαμερίσματα και σουίτες πολυτελείας 5 αστερών. Ειδικά για αυτόνομα καταλύματα επισκεφτείτε το εξειδικευμένο Villas.com, έναν νέο ιστοχώρο της Booking.com. Πραγματικά διεθνής, η Booking.com είναι διαθέσιμη σε περισσότερες από 40 γλώσσες και προσφέρει πάνω από 804.427 ενεργά καταλύματα σε 221 χώρες και περιφέρειες.

Η Booking.com B.V. έχει την έδρα της στο Άμστερνταμ, Κάτω Χώρες, και υποστηρίζεται διεθνώς από 170 γραφεία σε περισσότερες από 60 χώρες σε όλο τον κόσμο: Αγία Πετρούπολη - Άγιος Δομίνικος - Αθήνα - Αμβούργο - Άμστερνταμ - Ανόι - Ατλάντα - Αττάλεια - Βανκούβερ - Βαρκελώνη - Βαρσοβία - Βενετία - Βερολίνο - Βιέννη - Βίλνιους - Βοστώνη - Βουδαπέστη - Βουκουρέστι - Βρυξέλλες - Γιανγκόν - Γιογκιακάρτα - Γιοχάνεσμπουργκ - Γκουανγκζού - Γκραντ Ράπιντς - Δουβλίνο - Εδιμβούργο - Ελσίνκι - Ζάγκρεμπ - Ζυρίχη - Ηράκλειο -

Θεσσαλονίκη - Ίνσμπρουκ - Καζαμπλάνκα - Κάιρο - Κάλγκαρι - Κανκούν - Κέμπριτζ - Κέιπ Τάουν - Κίεβο - Κολόμπο - Κοπεγχάγη - Κο Σαμουί - Κουάλα Λουμπούρ - Κούτα (Μπαλί) - Κρακοβία - Κωνσταντινούπολη - Λάρνακα - Λας Βέγκας - Λας Πάλμας ντε Γκραν Κανάρια - Λιλ - Λίμα - Λιουμπλιάνα - Λισαβόνα - Λονδίνο - Λος Άτζελες - Λυών - Μαδρίτη - Μαϊάμι - Μάλαγα - Μανίλα - Μάντσεστερ - Μαρακές - Μελβούρνη - Μεξικό - Μιλάνο - Μόναχο - Μονπελιέ - Μόντρεαλ - Μόσχα - Μουμπάι - Μπανγκαλόρ - Μπανγκόκ - Μπογκοτά - Μπολζάνο - Μπορντώ - Μπουένος Άιρες - Μπρατισλάβα - Ναϊρόμπι - Νατάλ - Νέα Ορλεάνη - Νέα Υόρκη - Νέο Δελχί - Νίκαια - Νόργουοκ - Ντάλας - Ντένβερ - Ντίσελντορφ - Ντουμπάι - Ορλάντο - Οσάκα - Όσλο - Ουάσιγκτον - Πάλμα ντε Μαγιόρκα - Παρίσι - Πεκίνο - Πόλη του Παναμά - Πόρτο Αλέγκρε - Πουκέτ - Πράγα - Ρέικιαβικ - Ρεν - Ρίγα - Ρίο ντε Τζανέιρο - Ρώμη - Σαλάνς - Σάλτσμπουργκ - Σανγκάη - Σαντιάγο - Σαν Ντιέγκο - Σαν Φρανσίσκο - Σάο Πάολο - Σαπόρο - Σεούλ - Σιάτλ - Σιγκαπούρη - Σίδνεϊ - Σιέμ Ριπ - Σικάγο - Σορέντο - Σότσι - Σόφια - Στοκχόλμη - Στρασβούργο - Ταϊπέι - Ταλίν - Τελ Αβίβ - Τζακάρτα - Τζέντα - Τόκιο - Τορόντο - Τσεγκκντού - Φάρο - Φλωρεντία - Φουκουόκα - Φράμπουργκ μ Μπράισγκαου - Φρανκφούρτη - Χάικου - Χιούστον - Χο Τσι Μινγκ - Χογκ-Κονγκ - Χονολουλού - Φοίνιξ - Ωκλαντ

Υπάρχουν περισσότεροι από 10000 υπάλληλοι παγκοσμίως που είναι αφοσιωμένοι στην εξυπηρέτηση των πελατών και των συνεργαζόμενων καταλυμάτων της Booking.com.

Η Booking.com είναι ένας φιλικός προς το χρήστη ιστοχώρος που εγγυάται τις καλύτερες τιμές. Στόχος της είναι να παρέχει σε όλους τους ταξιδιώτες, είτε ταξιδεύουν για αναψυχή είτε για επαγγελματικούς λόγους, τον πιο απλό και οικονομικό τρόπο να ανακαλύψουν και να κάνουν κράτηση σε πληθώρα καταλυμάτων, σε κάθε γωνιά του κόσμου.

Οι ομάδες υποστήριξης καταλυμάτων και περιεχομένου συνεργάζονται από τα τοπικά γραφεία με τους υπεύθυνους καταλυμάτων για να διασφαλίσουν διαφάνεια, διαθεσιμότητα και τις καλύτερες τιμές για τους πελάτες μας. Η πολύγλωσση ομάδα εξυπηρέτησης πελατών προσφέρει άμεση βοήθεια και υποστήριξη σε όλους τους ενδιαφερόμενους, καθημερινά, όλο το 24ωρο, ώστε η διαμονή τους να πληροί ή και να υπερβαίνει τις προσδοκίες τους.

Η Booking.com εγγυάται τις καλύτερες τιμές, τόσο σε πόλεις, όσο και σε παραλιακούς ή εξοχικούς προορισμούς.

Η υπηρεσία κρατήσεων της Booking.com είναι δωρεάν. Δεν χρεώνονται τέλη κράτησης ή έξοδα συναλλαγής, ενώ σε πολλές περιπτώσεις μπορείτε να ακυρώσετε την κράτηση δωματίου χωρίς επιβάρυνση.

Το σύστημα κρατήσεων της Booking.com είναι ασφαλές και τα προσωπικά σας στοιχεία, καθώς και τα στοιχεία της πιστωτικής σας κάρτας, κρυπτογραφούνται.

3.6 Hotels

The screenshot shows the Hotels.com website interface. At the top, the search bar indicates the location as Heraklion, Greece, with dates from October 1, 2015, to October 2, 2015, for 1 adult and 2 children. The main content area displays search results for Heraklion, Greece, with filters for 'Ταξινόμηση κατά' (Sort by), 'Κατηγορία αστέρων' (Star category), 'Απόσταση' (Distance), 'Αξιολόγηση επισκεπτή' (Guest rating), and 'Τιμή' (Price). Three hotel listings are visible:

- Galaxy Hotel Iraklio:** 75 Dimokratias Ave, Heraklion, Crete Island, 71 306 Ελλάδα, 442030247879. Rating: 4.6/5 (Άριστο). Price: 440€ ~~83€~~. Features: Κορυφαίο ξενοδοχείο, 323 σχόλια πελατών.
- Hotel Olympic:** Kornarou Square 43, Heraklion, Crete Island, 71 201 Ελλάδα, 442030247879. Rating: 3.7/5 (Καλό). Price: 51€. Features: 119 σχόλια πελατών.
- Atrion Hotel:** 9 Chronaki Street, Heraklion, Crete Island, 71202 Ελλάδα, 442030247879. Rating: 3.7/5 (Πολύ καλό). Price: 76€ ~~53€~~. Features: Ειδική προσφορά.

Η Hotels.com είναι ένας γνωστός παροχέας ξενοδοχειακών καταλυμάτων παγκοσμίως, προσφέροντας υπηρεσίες κρατήσεων μέσω του δικτύου της που αποτελείται από μεταφρασμένες ιστοσελίδες και τηλεφωνικά κέντρα.

Η Hotels.com προσφέρει στους ταξιδιώτες μια από τις ευρύτερες επιλογές καταλυμάτων στο δίκτυο, συμπεριλαμβανομένων τόσο ανεξάρτητων ξενοδοχείων όσο και μεγάλων ξενοδοχειακών αλυσίδων, καθώς και ενοικιαζόμενων διαμερισμάτων, σε περισσότερες από 257.000 μονάδες ανά τον κόσμο. Η εταιρεία προσφέρει μια αστείρευτη αγορά για τιμές ξενοδοχείων, παροχές και διαθεσιμότητα, και επίσης ειδικεύεται στο να παρέχει στους ταξιδιώτες καταλύματα κατά τη διάρκεια περιόδων χωρίς διαθεσιμότητα.

Η Hotels.com LP είναι θυγατρική της Expedia, Inc. (NASDAQ: EXPE). CST: 2083949-50. Τα κεντρικά γραφεία της Hotels.com LP βρίσκονται στη διεύθυνση 10440 North Central Expressway, Suite 400, Dallas, Texas 75231. Ο Όμιλος Expedia διατηρεί γραφεία σε όλη την Ευρώπη.

3.7 Hertz

Στην ηλεκτρονική σελίδα μπορείτε να βρείτε πληροφορίες για την εταιρεία, τα προϊόντα και τις υπηρεσίες που προσφέρει μέσω του πιο σύγχρονου και αποτελεσματικού δικτύου εξυπηρέτησης πελατών που διαθέτει.

Η Hertz κλείνει φέτος 97 χρόνια παρουσίας στο χώρο του rent a car, με περισσότερα από 600.000 αυτοκίνητα και 7.500 σημεία εξυπηρέτησης σε 149 χώρες. Το Σεπτέμβριο του 1918, ο 22χρονος πρωτοπόρος των ενοικιάσεων αυτοκινήτων, Walter L. Jacobs, άνοιξε μια εταιρεία ενοικιάσεων αυτοκινήτων στο Σικάγο. Ξεκινώντας με περίπου δέκα αυτοκίνητα Ford Model T, τα οποία επισκεύασε και έβαψε ο ίδιος, ο Jacobs επεκτάθηκε γρήγορα και έφτασε μέσα σε πέντε χρόνια να έχει ετήσιο εισόδημα πάνω από \$1 εκατομμύριο.

Το 1923, ο Jacobs πούλησε την επιχείρηση στον John Hertz, πρόεδρο της μεταφορικής εταιρείας “Yellow Cab and Yellow Truck and Coach Manufacturing Company”. Ωστόσο, ο Jacobs παρέμεινε στην εταιρεία ως ανώτατο διοικητικό στέλεχος της Hertz. Το 1926, η επιχείρηση, η οποία αποκαλείται Hertz Drive-ur-Self System, πέρασε στα χέρια της General Motors, εφόσον είχε προηγηθεί η αγορά της Yellow Truck από τον John Hertz.

Η Hertz-Autohellas πρωτοπόρος της Ελληνικής αγοράς λειτούργησε για πρώτη φορά στην Ελλάδα το 1962 με την επωνυμία Hertz Hellas και από τότε συνεχίζει να προσφέρει στους πελάτες της υπηρεσίες υψηλής ποιότητας διεθνούς επιπέδου.

Ο επισκέπτης της ιστοσελίδας θα πρέπει να γνωρίζει ότι η Hertz είναι η μεγαλύτερη εταιρεία ενοικίασης αυτοκινήτων στον κόσμο αλλά και στην Ελλάδα όπου η πλήρης επωνυμία της είναι Hertz-Autohellas A.T.E.E. Από τον Αύγουστο του 1999 η Autohellas είναι μέλος του Χρηματιστηρίου Αξιών της Αθήνας.

Με περισσότερους από 115 σταθμούς στην Ελλάδα εκ των οποίων 21 σε αεροδρόμια η Hertz συνεχίζει να καινοτομεί προσφέροντας συνεχώς νέες υπηρεσίες, και ένα στόλο ο οποίος ξεπερνάει τα 23.000 αυτοκίνητα και έχει μέσο όρο ηλικίας μόλις 11 μήνες.

3.8 Sixt

Κρατήσεις Προσφορές Στόλος Δίκτυο MySixt Corporate/Agency Login Μακροχρόνια Μίσθωση Stay & Drive Περισσότερα Language

www.sixt.gr/gr/CarRental/ReservationsCar.aspx

1 Στοιχεία κρήσης 2 Επιλογή αυτοκινήτου 3 Τιμή & όροι

Τιμή: 2 Ημέρες Αρχική τιμή ενοικίασης
 Παραλαβή: Ηράκλειο - Αεροδρόμιο - 30.09.2015 / 16:00 **Αλλαγή**
 Επιστροφή: Ηράκλειο - Αεροδρόμιο - 02.10.2015 / 16:00

Οι καλύτερες προσφορές μας - Επιλέξτε το όχημα που επιθυμείτε

Όλες οι κατηγορίες	Saloon Cars	Mini Bus	Special Cars - Station Wagon	Special Cars - Cabrio	Special Cars - 4WD
	Renault Twingo Mini Hatchback (MCMR) Επιβάτες: 4 Πόρτες: 5 Παραδείγματα μοντέλων κατηγορίας: Renault Twingo, Chevrolet Spark	€ 59,64*	Περιλαμβάνονται: Απεριόριστο χιλιόμετρα (U.N.L.), Τέλη Κυκλοφορίας, Ασφάλεια έναντι τριτων (T.P.I.) και Μερική Απαλλαγή Ευθύνης Σημάν (C.D.W.) Δεν περιλαμβάνονται: Φόρος Αεροδρομίου (A.S.C.) και Φ.Π.Α.		
	Nissan Micra Εconom Hatchback (EDMR) Επιβάτες: 4 Πόρτες: 5 Παραδείγματα μοντέλων κατηγορίας: Nissan Micra, VW Up!	€ 65,35*	Περιλαμβάνονται: Απεριόριστο χιλιόμετρα (U.N.L.), Τέλη Κυκλοφορίας, Ασφάλεια έναντι τριτων (T.P.I.) και Μερική Απαλλαγή Ευθύνης Σημάν (C.D.W.) Δεν περιλαμβάνονται: Φόρος Αεροδρομίου (A.S.C.) και Φ.Π.Α.		
	VW Polo Compact Hatchback (CCMR) Επιβάτες: 4 Πόρτες: 5 Παραδείγματα μοντέλων κατηγορίας: VW Polo, Peugeot 208	€ 74,78*	Περιλαμβάνονται: Απεριόριστο χιλιόμετρα (U.N.L.), Τέλη Κυκλοφορίας, Ασφάλεια έναντι τριτων (T.P.I.) και Μερική Απαλλαγή Ευθύνης Σημάν (C.D.W.) Δεν περιλαμβάνονται: Φόρος Αεροδρομίου (A.S.C.) και Φ.Π.Α.		
	Peugeot 208 Diesel Compact Hatchback (CCMD) Επιβάτες: 4 Πόρτες: 5 Παραδείγματα μοντέλων κατηγορίας: Peugeot 208 Diesel	€ 80,24*	Περιλαμβάνονται: Απεριόριστο χιλιόμετρα (U.N.L.), Τέλη Κυκλοφορίας, Ασφάλεια έναντι τριτων (T.P.I.) και Μερική Απαλλαγή Ευθύνης Σημάν (C.D.W.) Δεν περιλαμβάνονται: Φόρος Αεροδρομίου (A.S.C.) και Φ.Π.Α.		
	VW Polo Compact Automatic/Hatchback (CCAR) Επιβάτες: 4 Πόρτες: 5 Παραδείγματα μοντέλων κατηγορίας: VW Polo, Peugeot 208	€ 80,24*	Περιλαμβάνονται: Απεριόριστο χιλιόμετρα (U.N.L.), Τέλη Κυκλοφορίας, Ασφάλεια έναντι τριτων (T.P.I.) και Μερική Απαλλαγή Ευθύνης Σημάν (C.D.W.) Δεν περιλαμβάνονται: Φόρος Αεροδρομίου (A.S.C.) και Φ.Π.Α.		
	VW Polo Diesel		Περιλαμβάνονται: Απεριόριστο χιλιόμετρα (U.N.L.), Τέλη Κυκλοφορίας, Ασφάλεια έναντι τριτων (T.P.I.) και Μερική Απαλλαγή Ευθύνης Σημάν (C.D.W.) Δεν περιλαμβάνονται: Φόρος Αεροδρομίου (A.S.C.) και Φ.Π.Α.		

Η Sixt rent a car είναι μια διεθνής εταιρεία ενοικίασης αυτοκινήτων που προσφέρει τις υπηρεσίες της σε 3500 τοποθεσίες σε 105 χώρες. Προσφέρει ένα ευρύ φάσμα οχημάτων από ένα μικρό οικονομικό αυτοκίνητο σε ένα μεγάλο οικογενειακό ή μίνι βαν 9 θέσεων.

Ενοικίαση αυτοκινήτου στην Αθήνα.

Οι σταθμοί ενοικίασης αυτοκινήτων έχουν εξαπλωθεί σε όλη την Αττική συμπεριλαμβανομένων των καταστημάτων στο κέντρο της Αθήνας (Λεωφόρος Συγγρού) στη Βόρεια Αττική (Νέα Χαλκηδόνα), Νότια Αττική (Βουλιαγμένη), στην Ανατολική Αττική (Σπάτα) και στη Δυτική Αττική (Περιστέρη). Επιλέξτε ενοικίαση αυτοκινήτου από τη Sixt στην Αθήνα, που βρίσκεται κοντά σε πολλούς από τους σημαντικότερους σταθμούς του υπόγειου σιδηρόδρομου στην πόλη συμπεριλαμβανομένου του Συντάγματος, της Ακρόπολης, Ξεν. Hilton, Ξεν. Grande Bretagne και πολλών άλλων.

Εκτός από τα υποκαταστήματά στο κέντρο της πόλης, μπορεί επίσης να παρέχει τις καλύτερες τιμές ενοικίασης αυτοκινήτων σε όλα τα μεγάλα αεροδρόμια όπως το Αεροδρόμιο «Μακεδονία» και το Αεροδρόμιο «Καζαντζάκης».

Ενοικίαση αυτοκινήτου στην Ελλάδα.

Έχει υποκαταστήματα σε κάθε μεγάλη πόλη σε όλη την Ελλάδα, παρέχοντας ενοικιαζόμενα αυτοκίνητα σε Θεσσαλονίκη, Κρήτη, Ρόδο και πολλών άλλων.

Προσφέρει premium ενοικιάσεις αυτοκινήτων, με τα αυτοκίνητα της Audi, BMW, Mercedes σε περισσότερες από 105 χώρες (π.χ. αυτοκίνητα για διακοπές στην Ισπανία, τη Γαλλία, την Πορτογαλία και πολλές άλλες δημοφιλείς τουριστικές περιοχές).

Ενοικίαση αυτοκινήτου για τις διακοπές σας στην Ελλάδα, Ισπανία, Γαλλία.....

Αν νοικιάσετε αυτοκίνητο στο εξωτερικό, τότε μπορείτε να βρείτε τη Sixt σε οποιοδήποτε δημοφιλή προορισμό σε όλο τον κόσμο. Μερικές από τις κορυφαίες περιοχές μας για ενοικίαση οχημάτων περιλαμβάνουν την Κρήτη, τη Ρόδο, την Κέρκυρα, τη Σκιάθο, Μάλαγα, Αλικάντε, Τενερίφη, Μόναχο, Παρίσι, Φλόριντα, το αεροδρόμιο της Γενεύης και το Άμστερνταμ.

Στη Sixt θα βρείτε οικονομικές τιμές και μια μεγάλη ποικιλία από αυτοκίνητα, σπορ αυτοκίνητα, 4x4 και πολλά άλλα οχήματα.

Υπηρεσίες Ενοικίασης αυτοκινήτου από το κινητό σας τηλέφωνο

Η Sixt είναι γνωστή για την καινοτομία της. Μια από τις τελευταίες καινοτομίες της στην προσπάθειά της να κάνει τη ζωή ευκολότερη για τους πελάτες της, είναι η γκάμα των κινητών εφαρμογών που επιτρέπουν κράτηση από το δρόμο, ακόμα κι αν έχουν μόλις φτάσει στο αεροδρόμιο.

Η κινητή υπηρεσία κρατήσεων είναι διαθέσιμη σε ένα ευρύ φάσμα από δημοφιλείς πλατφόρμες όπως το iPhone, Blackberry και Android τηλέφωνα. Αν δεν έχετε μία από αυτές τις συσκευές, τότε μπορείτε να χρησιμοποιήσετε ακόμη και μόνο το κανονικό σας τηλέφωνο ή το PDA σας, αρκεί να έχει σύνδεση στο internet.

3.9 Avis

The screenshot shows the Avis website interface. At the top, there is a navigation bar with the Avis logo and links for 'Αρχική Σελίδα', 'Βοήθεια', 'SiteMap', 'Επικοινωνία', 'H Avis', and 'Facebook'. Below this is a secondary navigation bar with links for 'Τιμές & Κρατήσεις', 'Προσφορές', 'Στάλος - Στάθμοι', 'Συνεργαστές', 'Avis Preferred', 'Υπηρεσίες', 'Μικροχρόνιες Μισθώσεις', 'Πωλήσεις Αυτιών', and 'My Avis'. The main content area is titled 'Προσφορά' and features a grid of car rental options. Each option includes a car image, the car model, the rental price in EUR, and a 'Δεπιμέριες' button. Below the car options, there is a section for 'Περιλαμβανεται' (Included) with a list of services and their prices per day.

Service	Price (per day)
Επιπλέον οδηγός	4,92 EUR
Booster (5+ ετών)	4,92 EUR
WiFi	8,61 EUR
Παιδικό κάθισμα (1-12 μηνών)	4,92 EUR
Παιδικό κάθισμα (1-4 ετών)	4,92 EUR
Παιδικό κάθισμα (4-12 ετών)	4,92 EUR

Ως πελάτης της Avis, μπορείτε να επωφεληθείτε από τα προνόμια που μπορεί να σας προσφέρει μία από τις μεγαλύτερες εταιρείες του χώρου παγκοσμίως.

Περισσότεροι από 6.000 σταθμοί ενοικίασης σε όλο τον κόσμο

Δυνατότητα επιλογής προϊόντων και υπηρεσιών που να ταιριάζουν στις ανάγκες σας

Ποικιλία αυτοκινήτων

Μία από τις ισχυρότερες εταιρίες ενοικιάσεως αυτοκινήτων

Είναι μία από τις εταιρίες του ομίλου Avis Budget EMEA Ltd. Η Avis Budget EMEA Ltd κατέχει ηγετική θέση στην ενοικίαση αυτοκινήτου σε Ευρώπη, Αφρική, Μέση Ανατολή και Ασία, με περισσότερα από 50 χρόνια εμπειρίας στο χώρο. Εξυπηρετεί τουλάχιστον 8 εκατομμύρια πελάτες ετησίως μέσα από ένα δίκτυο που εξαπλώνεται σε 168 χώρες.

Η Avis ξεκίνησε το 1946 όταν ο Warren Avis άνοιξε τον πρώτο σταθμό ενοικίαση αυτοκινήτων - "Avis Airlines Rent A Car Systems" στο Willow Run Airport του Detroit. Ήταν ο πρώτος σταθμός ενοικίασης αυτοκινήτων σε αεροδρόμιο. Το 1965, η Avis εξαπλώθηκε διεθνώς ιδρύοντας τη θυγατρική εταιρία Avis Europe η οποία δημιούργησε δίκτυο σε Ευρώπη, Αφρική και Μέση Ανατολή. Η Avis Europe κατάφερε να ηγηθεί του χώρου μόλις 8 χρόνια μετά την ίδρυσή της, το 1973 – τη θέση αυτή κατάφερε να την διατηρήσει έκτοτε.

Η συνταγή της επιτυχίας της Avis ήταν και είναι το όραμά της να δημιουργήσει την καλύτερη και ταχύτερα αναπτυσσόμενη εταιρία του κλάδου ενοικίασης αυτοκινήτου, μέσω συνεχών επενδύσεων σε σταθμούς, τεχνολογία και ειδικότερα σε ανθρώπινο δυναμικό.

4. Ανάλυση και παρουσίαση της εργασίας

The screenshot shows the 'Rooms' page of the real2 website. At the top, there is a navigation bar with links for 'Homepage', 'Information', and 'Contact us', along with flags for Greece and the UK. A yellow banner with a paperclip icon says 'Join Now for Free!'. Below this, there is a 'ΔΗΜΙΟΥΡΓΙΑ ΛΟΓΑΡΙΑΣΜΟΥ' (Create Account) section with fields for 'Όνομα Χρήστη' (Username) and 'Κωδικός' (Password), and a 'LOGIN' button. The main content area is divided into two columns. The left column is titled 'SEARCH ROOMS' and contains a search form with fields for 'Τοποθεσία:' (Location), 'Ελάχιστη Τιμή:' (Minimum Price), 'Μέγιστη Τιμή:' (Maximum Price), 'Ημερομηνία Check-in:' (Check-in Date), and 'Ημερομηνία Check-out:' (Check-out Date). The right column is titled 'ΔΗΜΟΦΙΛΗ ΞΕΝΟΔΟΧΕΙΑ' (Popular Hotels) and lists three hotels: 'CANDIA MARIS' (200.00 €), 'Rodos Palladium' (100.00 €), and 'Electra Palace Hotel' (29.00 €). Below the search section, there is a 'ΚΑΤΗΓΟΡΙΕΣ' (Categories) section with filters for star ratings: 'Ένα αστέρι', 'Δύο αστέρια', 'Τρία αστέρια', 'Τέσσερα αστέρια', and 'Πέντε αστέρια'. A 'Προτεινόμενη Λίστα' (Recommended List) section features a card for 'GDM Megaron Hotel' with a 5-star rating and a '1 Image' icon. To the right of this section is an advertisement for 'FIND YOUR Dream Car' featuring a silver car.

Ο χρήστης μπορεί να διακρίνει στην αρχική σελίδα της εργασίας πάνω αριστερά την ονομασία της ιστοσελίδας και πάνω δεξιά την φόρμα εισόδου. Υπάρχει η δυνατότητα από το ν χρήστη να εισάγει το όνομα χρήστη και τον προσωπικό του κωδικό και να εισέλθει στον λογαριασμό του. Στην περίπτωση που δεν έχει λογαριασμό μπορεί εύκολα να δημιουργήσει πατώντας την επιλογή “ΔΗΜΙΟΥΡΓΙΑ ΛΟΓΑΡΙΑΣΜΟΥ”.

Μόλις ο χρήστης πατήσει αυτήν την επιλογή, μεταφέρετε στην φόρμα εγγραφής όπου μπορεί να δώσει τα προσωπικά του στοιχεία και να πραγματοποιήσει την εγγραφή του.

The screenshot shows the 'CREATE USER' registration form on the real2 website. At the top, there is a navigation bar with links for 'Homepage', 'Information', and 'Contact us', along with flags for Greece and the UK. A yellow banner with a paperclip icon says 'Join Now for Free!'. Below this, there is a 'ΔΗΜΙΟΥΡΓΙΑ ΛΟΓΑΡΙΑΣΜΟΥ' (Create Account) section with fields for 'Όνομα Χρήστη' (Username) and 'Κωδικός' (Password), and a 'LOGIN' button. The main content area is divided into two columns. The left column is titled 'SEARCH ROOMS' and contains a search form with fields for 'Τοποθεσία:' (Location), 'Ελάχιστη Τιμή:' (Minimum Price), 'Μέγιστη Τιμή:' (Maximum Price), 'Ημερομηνία Check-in:' (Check-in Date), and 'Ημερομηνία Check-out:' (Check-out Date). The right column is titled 'ΔΗΜΟΦΙΛΗ ΞΕΝΟΔΟΧΕΙΑ' (Popular Hotels) and lists three hotels: 'CANDIA MARIS' (200.00 €), 'Rodos Palladium' (100.00 €), and 'Electra Palace Hotel' (29.00 €). Below the search section, there is a 'ΚΑΤΗΓΟΡΙΕΣ' (Categories) section with filters for star ratings: 'Ένα αστέρι', 'Δύο αστέρια', 'Τρία αστέρια', 'Τέσσερα αστέρια', and 'Πέντε αστέρια'. A 'Προτεινόμενη Λίστα' (Recommended List) section features a card for 'GDM Megaron Hotel' with a 5-star rating and a '1 Image' icon. To the right of this section is an advertisement for 'FIND YOUR Dream Car' featuring a silver car.

Με την ολοκλήρωση της εγγραφής τα στοιχεία του χρήστη αποθηκεύονται στην βάση δεδομένων και μπορεί οποιαδήποτε στιγμή να εισέλθει στον προσωπικό του λογαριασμό πληκτρολογώντας το username και τον προσωπικό του κωδικό.

MySQL Database Management Tool Screenshot:

Host: 127.0.0.1 Database: hotelrental Table: users

id	name	email	username	password	address
1	0	0	admin	1234	
2	test	info@tes.gr	test	1234	
3	giorgos	giorgakis	george	4321	
4	fol	fol@fol.gr	fol	fol	
5	wedwer	werwer	234234	234234	
6	DIMIS	DIMIS	DIMIS	DIMIS	1234
7	sdksfdjfd	jsadfkjsdfk	sdjfklsdfj	sdjfklsdfj	sdksfdkdf
8	ty	uyu	rturtu	rturtu	rturtu

Console Output:

```

41 SHOW CREATE TABLE `carrental`.`reservations`;
42 USE `hotelrental`;
43 SHOW CREATE TABLE `hotelrental`.`users`;
44 SELECT * FROM `hotelrental`.`users` LIMIT 1000;
45 SHOW CREATE TABLE `hotelrental`.`users`;

```

Όταν γίνει η είσοδος ο χρήστης μπορεί να περιηγηθεί και να πραγματοποιήσει κρατήσεις με τον προσωπικό του λογαριασμό και στην ιστοσελίδα για τις κρατήσεις δωματίων σε ξενοδοχείων αλλά και στην ιστοσελίδα για τις ενοικιάσεις αυτοκινήτων.

Web Application Interface:

Navigation: Homepage, Information, Contact us

User Status: LOGGED IN AS: FOL | logout

Rooms

Join Now for Free!

SEARCH ROOMS

Τοποθεσία: Παράδειγμα : Πόλη, Περιοχή / ή ταχυδρομικό κώδικα - 123456

Ελάχιστη τιμή:

Μέγιστη τιμή:

Ημερομηνία Check-in:

Ημερομηνία Check-out:

ΔΗΜΟΦΙΛΗ ΞΕΝΟΔΟΧΕΙΑ

- CANDIA MARIS**
ΑΜΜΟΥΔΑΡΑ
200.00 €
- Rodos Palladium**
Παραλία Φαληράκι, Φαληράκι, 85100, Ελλάδα
100.00 €
- Electra Palace Hotel**
Πλατεία Αριστοτέλους 9, Θεσσαλονίκη, 54624, Ελλάδα
29.00 €

ΚΑΤΗΓΟΡΙΕΣ | Ένα αστέρι | Δύο αστέρια | Τρία αστέρια | Τέσσερα αστέρια | Πέντε αστέρια

Προτεινόμενη λίστα

- GDM Megaron Hotel**
★★★★★
Δ. Μπισφώρ 9, Ηράκλειο Κρήτης, 71202, Ελλάδα

FIND YOUR Dream Car


Επίσης, στην αρχική σελίδα υπάρχει η αναζήτηση με την οποία ο χρήστης μπορεί να αναζητήσει ξενοδοχεία σύμφωνα με τα κριτήρια του. Στην κατηγορία με τα δημοφιλή ξενοδοχεία υπάρχει η δυνατότητα να γνωρίσει τα δημοφιλέστερα καταλύματα. Στην προτεινόμενη λίστα ο χρήστης μπορεί να επιλέξει την κατηγορία των ξενοδοχείων που επιθυμεί βάσει των αστερών και να του εμφανιστεί η σχετική λίστα.

Αναζήτηση

11 ΠΛΑΤΕΙΟ ΑΡΙΣΤΟΤΕΛΟΥΣ 9, ΘΕΣΣΑΛΟΝΙΚΗ, 54624, ΕΛΛΑΔΑ
29.00 €


ΚΑΤΗΓΟΡΙΕΣ Ένα αστέρι Δύο αστέρια Τρία αστέρια Τέσσερα αστέρια Πέντε αστέρια

Προτεινόμενη Λίστα



GDM Megaron Hotel ★★★★★
Δ. Μποφώρ 9, Ηράκλειο Κρήτης, 71202, Ελλάδα
1 Image


Λεπτομέρειες
Κράτηση



Galaxy Iraklio Hotel ★★★★★
Λεωφόρος Δημοκρατίας 75, Ηράκλειο Κρήτης, 71306, Ελλάδα
1 Image

Λεπτομέρειες
Κράτηση

Electra Palace Hotel Λεπτομέρειες




FIND YOUR Dream Car

Μόλις ο χρήστης πραγματοποιήσει κράτηση σε κάποιο ξενοδοχείο την ημερομηνία που επιθυμεί, αυτή αποθηκεύεται στην βάση δεδομένων και ολοκληρώνεται η κράτηση.

INFO

Join Now for Free!

LOGGED IN AS: FOL | logout



GDM Megaron Hotel ★★★★★
Δ. Μποφώρ 9, Ηράκλειο Κρήτης, 71202, Ελλάδα
1 Image

Από 13 Νοέμβριος 2015
Έως 1 Ιανουάριος 2016
email: gj0rg00sk17589@hotmail.com

MAKE RESERVATION

real2 Αρχική | Πληροφορίες | Επικοινωνία

Επιστροφή στην αρχή

The screenshot shows a MySQL database management tool interface. The left sidebar displays a tree view of databases and tables. The main window shows the 'reservations' table in the 'hotelrental' database. The table contains 7 rows of data. Below the table, a SQL query is visible, and the status bar at the bottom indicates the connection details.

id	hotelid	fromdate	todate	email
2	4	20.150.101	20.150.101	(NULL)
3	4	20.150.101	20.150.101	
4	1	20.150.101	20.150.101	
5	6	20.150.101	20.150.101	
6	7	20.150.905	20.150.907	info@xx.gr
7	1	20.150.911	20.150.914	info@xx.gr
8	1	20.151.113	20.160.101	gl0rg00sk17589@hotmail.com

```





33 SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM information_schema.`EVENTS` WHERE `EVENT_SCHEMA`='hotelrental';
34 USE `hotelrental`;
35 SHOW CREATE TABLE `hotelrental`.`reservations`;
36 SELECT * FROM `hotelrental`.`reservations` LIMIT 1000;
37 SHOW CREATE TABLE `hotelrental`.`reservations`;

```

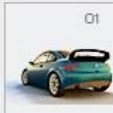
Στην αρχική σελίδα της ιστοσελίδας για τις ενοικιάσεις των αυτοκινήτων μπορεί ο χρήστης να κάνει αναζήτηση αυτοκινήτων βάση των κριτηρίων επιλογής του.

The screenshot shows the homepage of a car rental company. The main banner features a couple sitting on a yellow car. Below the banner, there is a 'Car Booking' form with several dropdown menus and a 'Submit' button. The form includes fields for 'Select State', 'Select Location', 'Pick-Up Date', and 'Drop-Off Date'. At the bottom, there is a section for 'Our Technological Partners' with logos for Thrifty, AVIS, Budget, National, Alamo, and ACE Rent a Car.

Η λίστα των διαθέσιμων αυτοκινήτων εμφανίζεται όπως στην εικόνα παρακάτω.

	FORD FOCUS	1600	2010	Heraklion	Heraklion Center
	NISSAN	1100	1999	Heraklion	Heraklion Center
	PORSCHE	3000	2011	Heraklion	Heraklion Airport
	SUZUKI	1600	2010	Chania	Chania Airport

Ο χρήστης επιλέγει το αυτοκίνητο που επιθυμεί και κάνει την κράτηση.

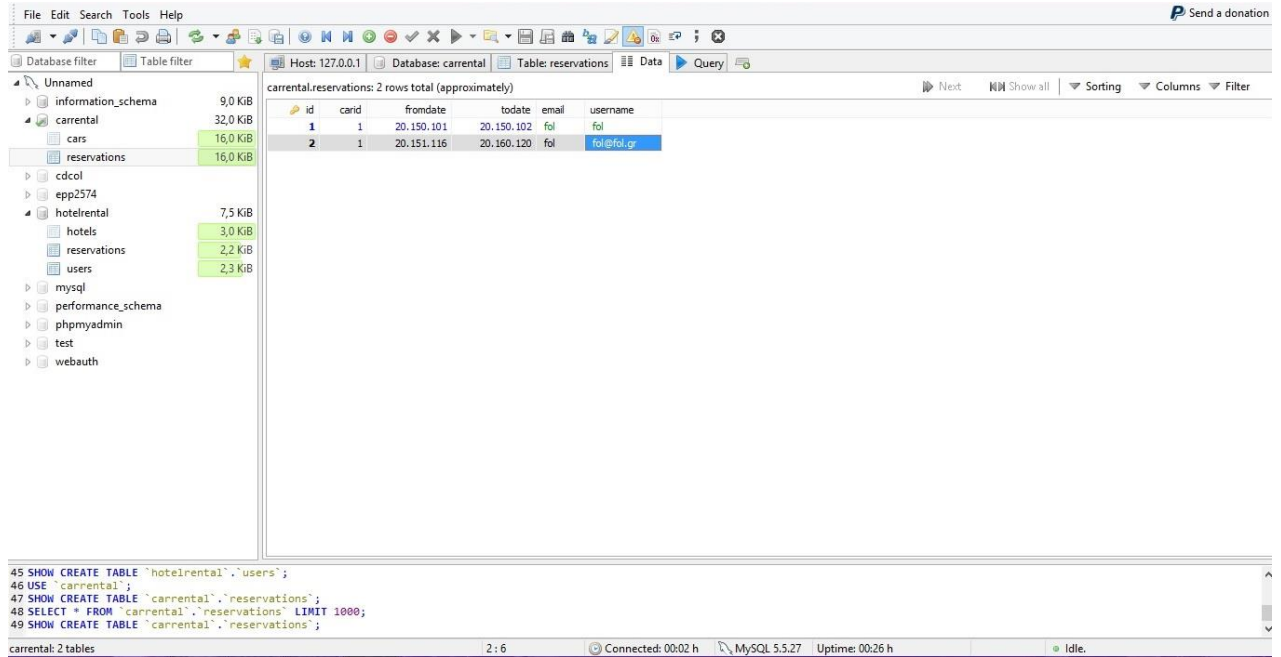
	FORD FOCUS	1600	2010	Heraklion	Heraklion Center	Κράτηση
---	------------	------	------	-----------	------------------	-------------------------

Μόλις ολοκληρωθεί η επιλογή εμφανίζεται ένα μήνυμα που επιβεβαιώνει την κράτηση.

Home | Customer Care | Faq | Contact

Thank you for your reservation!!

Τέλος, η κράτηση αποθηκεύεται στην βάση δεδομένων και ολοκληρώνεται η διαδικασία.



Στην κατηγορία “πληροφορίες” του αρχικού μενού ο χρήστης μπορεί να βρει τις απαραίτητες πληροφορίες για τις ιστοσελίδες έτσι ώστε να έχει μια ικανοποιητική γνώση για την λειτουργία των ιστοσελίδων και την κατάρτιση του προσωπικού τους.

Στην κατηγορία “επικοινωνία” ο χρήστης μπορεί να βρει τους τρόπους επικοινωνίας με τους οποίους έχει την δυνατότητα να επικοινωνήσει μαζί με το προσωπικό των ιστοσελίδων με σκοπό την επίλυση προβλημάτων, απόκτηση πληροφοριών και οδηγιών.

Να επισημάνουμε ότι υπάρχει η δυνατότητα αλλαγής γλώσσας από ελληνικά σε αγγλικά και αντίστροφα για την ευκολότερη περιήγηση στις ιστοσελίδες.

5. Συμπεράσματα – Αποτελέσματα

Η εργασία αυτή είχε σαν σκοπό την ανάλυση και επίδειξη τις λειτουργίας ιστοσελίδων που αφορούν κρατήσεις δωματίων σε ξενοδοχεία και ενοικιάσεις αυτοκινήτων. Έγιναν αναφορές σε μεθόδους και προγράμματα που χρησιμοποιούνται για την ανάπτυξη ενός συστήματος κρατήσεων. Επίσης, αναφερθήκαμε σε γνωστές ιστοσελίδες σκοπό να έχουμε ως βάση την λειτουργία τους για την δημιουργία της παρούσας εργασίας.

Το περιβάλλον της εργασίας είναι φιλικό και προσιτό στον χρήστη με στόχο την εξυπηρέτηση του. Οι υπηρεσίες που προσφέρονται είναι απλές, κατανοητές και εύκολες προς τον χρήστη.

Στην εποχή μας τα συστήματα κρατήσεων είναι ευρέως διαδομένα και πολλοί χρήστες χρησιμοποιούν ηλεκτρονικές ιστοσελίδες για τις κρατήσεις δωματίων σε ξενοδοχεία και ενοικιάσεις αυτοκινήτων. Οι επιχειρήσεις επιλέγουν δυναμικά προϊόντα και υπηρεσίες για εξοικονόμηση χρόνου και κόστους με σκοπό την αύξηση του κέρδους της εταιρείας.

Τα συστήματα κρατήσεων μπορούν να αποθηκεύσουν πολλές πληροφορίες και μεγάλο όγκο δεδομένων που τα καθιστά μία εύκολη λύση για την γρήγορη παρουσίαση και ανάλυση των υπηρεσιών.

Τέλος, συμπεραίνουμε ό,τι η εργασία πετυχαίνει τον σκοπό της σε ικανοποιητικό βαθμό όμως χρειάζονται πολλές μελλοντικές αλλαγές για να είναι πλήρης.

Βιβλιογραφία

- Κ. Π. Ν. ΦΛΩΡΙΝΑΣ Η θεωρία των βάσεων δεδομένων
Available: <http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-DataBasesTheory.html>.
- Κεφάλαιο 1 : Γενικές αρχές βάσεων δεδομένων,»
Available: http://anamorfosi.teicm.gr/ekp_yliko/e-notes/Data/database/main.htm.
- MySQL, Available: <http://www.mysql.com/>.
- W3C HTML, Available: <http://www.w3.org/html/>.
- W3C CSS , Available: <http://www.w3.org/Style/CSS/Overview.en.html>
- Node.js , Available: <http://nodejs.org/>
- <http://www.phpknowhow.com/basics/working-with-xampp/>
- <https://en.wikipedia.org/wiki/XAMPP>
- <http://php.net/>
- <http://docs.phpmyadmin.net/en/latest/>
- <http://www.phpknowhow.com/basics/working-with-xampp/>

ΠΑΡΑΡΤΗΜΑ Α

Κώδικας δημιουργίας βάσεων δεδομένων

```
CREATE DATABASE IF NOT EXISTS `hotelrental`;  
USE `hotelrental`;
```

```
CREATE TABLE IF NOT EXISTS `hotels` (  
  `id` int(11) NOT NULL AUTO_INCREMENT COMMENT 'unique id',  
  `hotelname` varchar(50) NOT NULL COMMENT 'hotel name',  
  `address` text NOT NULL COMMENT 'hotel address',  
  `stars` tinyint(4) NOT NULL COMMENT 'how many stars',  
  `featured` tinyint(4) NOT NULL COMMENT '0= not showing in home page, 1= showing in  
homepage',  
  `price` varchar(10) NOT NULL,  
  `image1` varchar(255) NOT NULL COMMENT 'images',  
  `image2` varchar(255) DEFAULT NULL COMMENT 'images',  
  `image3` varchar(255) DEFAULT NULL COMMENT 'images',  
  `image4` varchar(255) DEFAULT NULL COMMENT 'images',  
  `hits` int(11) DEFAULT NULL COMMENT 'views',  
  `checkin` datetime DEFAULT NULL COMMENT 'checkin',  
  `checkout` datetime DEFAULT NULL COMMENT 'checkout',  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM AUTO_INCREMENT=8 DEFAULT CHARSET=utf8;
```

```
INSERT INTO `hotels` (`id`, `hotelname`, `address`, `stars`, `featured`, `price`, `image1`, `image2`,  
`image3`, `image4`, `hits`, `checkin`, `checkout`) VALUES  
  (1, 'GDM Megaron Hotel', 'Δ. Μποφόρ 9, Ηράκλειο Κρήτης, 71202, Ελλάδα', 5, 1, '45',  
'2017279.jpg', NULL, NULL, NULL, 10, '2015-01-02 13:28:51', '2016-08-02 13:28:57'),  
  (2, 'Galaxy Iraklio Hotel', 'Λεωφόρος Δημοκρατίας 75, Ηράκλειο Κρήτης, 71306, Ελλάδα', 5, 1,  
'40', '5576751.jpg', NULL, NULL, NULL, 5, '2015-01-02 13:28:51', '2016-08-02 13:28:57'),  
  (3, 'Hotel Grande Bretagne', 'Βασιλέως Γεωργίου \A, 1, Αθήνα, 105 64, Ελλάδα', 1, 0, '30',  
'20533417.jpg', NULL, NULL, NULL, 30, '2015-08-02 13:56:07', '2015-08-02 13:56:08'),  
  (4, 'Electra Palace Hotel', 'Πλατεία Αριστοτέλους 9, Θεσσαλονίκη, 54624, Ελλάδα', 3, 1, '29',  
'41215031.jpg', NULL, NULL, NULL, 100, '2015-08-02 13:56:54', '2015-08-02 13:56:54'),  
  (5, 'Μακεδονία Παλλάς', 'Λεωφ. Μεγ. Αλεξάνδρου 2, Θεσσαλονίκη, Gr-546 40, Ελλάδα', 2, 0, '59',  
'37092866.jpg', NULL, NULL, NULL, 98, '2015-08-02 14:04:44', '2015-08-02 14:04:44'),  
  (6, 'Rodos Palladium', 'Παραλία Φαληράκι, Φαληράκι, 85100, Ελλάδα', 4, 0, '100', '23280173.jpg',  
NULL, NULL, NULL, 200, '2015-08-02 14:05:36', '2015-08-02 14:05:37'),  
  (7, 'CANDIA MARIS', 'AMMOYΔΑΡΑ', 4, 0, '200', '23280173.jpg', NULL, NULL, NULL, 500,  
'2015-08-28 19:00:30', '2015-08-28 19:00:35');
```

```
CREATE TABLE IF NOT EXISTS `reservations` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `hotelid` int(11) NOT NULL,  
  `fromdate` int(11) NOT NULL,  
  `todate` int(11) NOT NULL,  
  `email` varchar(150) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM AUTO_INCREMENT=8 DEFAULT CHARSET=utf8;
```

```
INSERT INTO `reservations` (`id`, `hotelid`, `fromdate`, `todate`, `email`) VALUES
```

```
(2, 4, 20150101, 20150101, NULL),
(3, 4, 20150101, 20150101, ""),
(4, 1, 20150101, 20150101, ""),
(5, 6, 20150101, 20150101, ""),
(6, 7, 20150905, 20150907, 'info@xx.gr'),
(7, 1, 20150911, 20150914, 'info@xx.gr');
```

```
CREATE TABLE IF NOT EXISTS `users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(150) NOT NULL DEFAULT '0',
  `email` varchar(100) NOT NULL DEFAULT '0',
  `username` varchar(50) NOT NULL DEFAULT '0',
  `password` varchar(50) NOT NULL DEFAULT '0',
  `address` varchar(200) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=8 DEFAULT CHARSET=utf8;
```

```
INSERT INTO `users` (`id`, `name`, `email`, `username`, `password`, `address`) VALUES
(1, '0', '0', 'admin', '1234', ''),
(2, 'test', 'info@tes.gr', 'test', '1234', ''),
(3, 'giorgos', 'giorgakis', 'george', '4321', ''),
(4, 'fol', 'fol', 'fol', 'fol', ''),
(5, 'wedwer', 'werwer', '234234', '234234', ''),
(6, 'DIMIS', 'DIMIS', 'DIMIS', 'DIMIS', '1234'),
(7, 'sdjksdjfkl', 'jsadfklsjdfk', 'sdjfklsldfj', 'sdjfklsjdfkl', 'sdkfjsdklf');
```

```
CREATE DATABASE IF NOT EXISTS `carrental`;
USE `carrental`;
```

```
CREATE TABLE IF NOT EXISTS `cars` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `marka` varchar(100) NOT NULL,
  `kybika` varchar(100) NOT NULL,
  `imerominiakataskeyis` varchar(100) NOT NULL,
  `state` varchar(100) NOT NULL,
  `image` varchar(255) NOT NULL,
  `location` varchar(100) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
```

```
INSERT INTO `cars` (`id`, `marka`, `kybika`, `imerominiakataskeyis`, `state`, `image`, `location`)
VALUES
(1, 'FORD FOCUS', '1600', '2010', 'Heraklion', 'img01.jpg', 'Heraklion Center'),
(2, 'NISSAN', '1100', '1999', 'Heraklion', 'img02.jpg', 'Heraklion Center'),
(3, 'PORCHE', '3000', '2011', 'Heraklion', 'img02_hover.jpg', 'Heraklion Airport');
```

```
CREATE TABLE IF NOT EXISTS `reservations` (
```

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`carid` int(11) NOT NULL,  
`fromdate` int(11) NOT NULL,  
`todate` int(11) NOT NULL,  
`email` varchar(150) DEFAULT NULL,  
PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```