



**Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης.**

**Σχολή Τεχνολογικών εφαρμογών  
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων.**



## **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

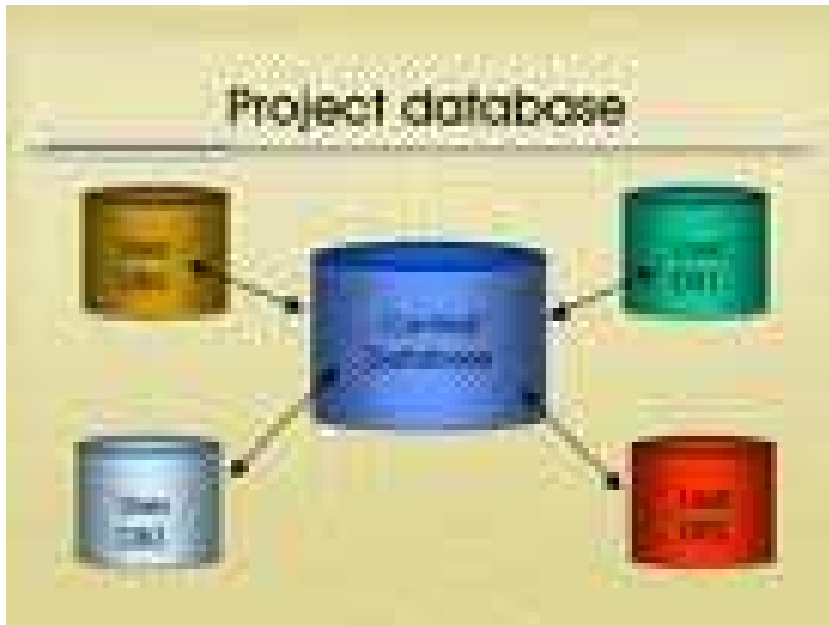
### **ΤΙΤΛΟΣ:**

**Διαχείριση Βάσεων Δεδομένων. Ανάπτυξη ευφυών μεθόδων αναζήτησης και παρουσίαση πληροφοριών.**

**ΣΠΟΥΔΑΣΤΡΙΑ: Γκιουλή Χριστίνα (Α.Μ. : 327)**

**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Μαυρικάκης Ιωάννης**

## Project database



# Περιεχόμενα

Μέρος Α' – Γενικά Θέματα Βάσεων Δεδομένων .....	6
Έννοιες και Ορισμοί.....	6
Στοιχεία μιας Βάσης Δεδομένων.....	7
Εργαλεία Χειρισμού Πληροφοριών .....	8
Οι Γλώσσες 4ης Γενιάς (4GL) .....	8
Ο ρόλος του DBA.....	8
Μοντέλα & Εξέλιξη Βάσεων Δεδομένων .....	10
Οι Ιεραρχικές Βάσεις Δεδομένων .....	10
Οι Δικτυωτές Βάσεις Δεδομένων .....	10
Οι Σχεσιακές Βάσεις Δεδομένων .....	10
Το Σχεσιακό Μοντέλο .....	11
Κανόνες Ακεραιότητας.....	11
Συστήματα Διαχείρισης Βάσεων Δεδομένων.....	13
Γενικές έννοιες .....	13
Παράδειγμα .....	14
Αρχιτεκτονική ενός τυπικού ΣΔΒΔ.....	15
Πλεονεκτήματα χρήσης ΒΔ.....	17
Αρχές σχεδιασμού ΒΔ .....	18
Μέρος Β' – Βάση Δεδομένων για Ηλεκτρονικό Αρχείο Άρθρων Περιοδικών .....	20
Γενική περιγραφή της ΒΔ.....	20
Εικόνα 1 - γενική όψη της ΒΔ.....	22
Πίνακες της ΒΔ.....	23
Article .....	23
Author.....	24
ArticleAuthor.....	25
Publisher .....	26
Magazine .....	27
ArticleMagazine .....	28
Reference .....	29
ArticleReference.....	30

Παρουσίαση Παραδειγμάτων .....	31
Article .....	31
Author .....	31
ArticleAuthor .....	32
Reference .....	32
ArticleReference .....	32
Publisher .....	33
Magazine .....	33
ArticleMagazine .....	33
Παράρτημα Α' – Οδηγός Εγκατάστασης MySQL .....	36
Εγκατάσταση MySQL Community Server .....	36
Ρύθμιση Παραμέτρων Λειτουργίας .....	39
Εγκατάσταση Γραφικών Εργαλείων .....	44
Παράρτημα Β – Κώδικας Δημιουργίας Βάσης Δεδομένων .....	49

*Μέρος Α΄*

*Γενικά Θέματα Βάσεων*

*Δεδομένων*

## Μέρος Α΄

# Γενικά Θέματα Βάσεων Δεδομένων

### Έννοιες και Ορισμοί

Με τον όρο Βάση Δεδομένων (ΒΔ) εννοείται μία συλλογή από συστηματικά οργανωμένα (formatted) σχετιζόμενα δεδομένα. Ένας τηλεφωνικός κατάλογος, για παράδειγμα, θεωρείται βάση δεδομένων, καθώς αποθηκεύει και οργανώνει σχετιζόμενα τμήματα πληροφορίας, όπως είναι το όνομα και ο αριθμός τηλεφώνου. Ωστόσο, στον κόσμο των υπολογιστών, με τον όρο βάση δεδομένων αναφερόμαστε σε μια συλλογή σχετιζόμενων δεδομένων - τμημάτων πληροφορίας - ηλεκτρονικά αποθηκευμένων.

Πέρα από την εγγενή της ικανότητα να αποθηκεύει δεδομένα, η βάση δεδομένων παρέχει βάσει του σχεδιασμού και του τρόπου ιεράρχησης των δεδομένων της σε προγράμματα ή συλλογές προγραμμάτων, τα αποκαλούμενα συστήματα διαχείρισης περιεχομένου, τη δυνατότητα γρήγορης άντλησης και ανανέωσης των δεδομένων. Η ηλεκτρονική βάση δεδομένων χρησιμοποιεί ιδιαίτερου τύπου λογισμικό προκειμένου να οργανώσει την αποθήκευση των δεδομένων της. Το διακριτό αυτό λογισμικό είναι γνωστό ως Σύστημα διαχείρισης βάσης δεδομένων συντομευμένα (DBMS)

Το Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS, DataBase Management System) είναι ένα σύνολο από προγράμματα που επιτρέπουν τον χειρισμό των δεδομένων μιας ή περισσοτέρων βάσεων δεδομένων που ανήκουν στο ίδιο σύστημα. Το DBMS περιέχει κάποια εργαλεία γενικής

χρήσης για να μπορούμε να δημιουργούμε και να χειριζόμαστε τα δεδομένα.

### Οι στόχοι μιας βάσης δεδομένων είναι οι εξής:

- ❖ Ο περιορισμός της πολλαπλής αποθήκευσης των ίδιων στοιχείων (redundancy).
- ❖ Ο καταμερισμός (sharing) των ίδιων στοιχείων σ' όλους τους χρήστες.
- ❖ Η ομοιομορφία (uniformity) στον χειρισμό και την αναπαράσταση των δεδομένων.
- ❖ Η επιβολή κανόνων ασφαλείας (security).
- ❖ Η διατήρηση της ακεραιότητας (integrity) και της αξιοπιστίας (reliability) των δεδομένων.
- ❖ Η ανεξαρτησία των δεδομένων (data independence) και των προγραμμάτων από τον φυσικό τρόπο αποθήκευσης των δεδομένων.

### Στοιχεία μιας Βάσης Δεδομένων

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.
- **Πρωτεύον Κλειδί (Primary Key)**, είναι ένα πεδίο ή συνδυασμός πεδίων που χαρακτηρίζει μοναδικά μια εγγραφή.
- **Ξένο Κλειδί (Foreign Key)**, είναι ένα πεδίο που έχει το ίδιο σύνολο τιμών με το πρωτεύον κλειδί ενός άλλου αρχείου.

## Εργαλεία Χειρισμού Πληροφοριών

Τα εργαλεία χειρισμού πληροφοριών μιας βάσης δεδομένων είναι γνωστά και σαν "Γλώσσες Εντολών" και με τη βοήθειά τους μπορούμε να δώσουμε εντολές χειρισμού των δεδομένων. Η πιο γνωστή και ευρέως διαδεδομένη γλώσσα εντολών για τις σύγχρονες βάσεις δεδομένων είναι η Δομημένη Γλώσσα Ερωτήσεων (Structured Query Language - SQL), η οποία αποτελείται από τα εξής μέρη :

- DDL (Data Definition Language, Γλώσσα Ορισμού Δεδομένων), με την οποία καθορίζουμε τις δομές και τα τμήματα μιας βάσης δεδομένων.
- DML (Data Manipulation Language, Γλώσσα Χειρισμού Δεδομένων), με την οποία επεξεργαζόμαστε τα δεδομένα μιας βάσης δεδομένων.
- DCL (Data Control Language, Γλώσσα Ελέγχου Δεδομένων), με την οποία εξασφαλίζουμε την ασφάλεια και την ακεραιότητα των δεδομένων μιας βάσης δεδομένων.

### Οι Γλώσσες 4ης Γενιάς (4GL)

Οι Γλώσσες 4ης Γενιάς 4GL (4th Generation Languages) είναι γλώσσες προγραμματισμού υψηλού επιπέδου, δηλ. έχουμε την δυνατότητα να κάνουμε πολλές και σύνθετες λειτουργίες με την βοήθειά τους και με λίγες μόνο εντολές.

Οι γλώσσες αυτές στηρίζουν την αποθήκευση των δεδομένων στην θεωρία των βάσεων δεδομένων και έτσι εξασφαλίζεται η ακεραιότητα των πληροφοριών ενώ παρέχουν τα βασικά εργαλεία αναζήτησης και χειρισμού των δεδομένων.

### Ο ρόλος του DBA

Ο Διαχειριστής μιας Βάσης Δεδομένων (DBA, DataBase Administrator) είναι αυτός που έχει την ευθύνη για τον σωστό, αποδοτικό και αξιόπιστο τρόπο δημιουργίας και λειτουργίας μια βάσης δεδομένων. Οι αρμοδιότητές του είναι οι εξής :



- Η απόφαση για το είδος των πληροφοριών που πρέπει να αποθηκευθούν.
- Η απόφαση για τον τρόπο αποθήκευσης και πρόσβασης στις πληροφορίες αυτές.
- Η συνεργασία με τους τελικούς χρήστες.
- Η απόφαση για τον τρόπο εξασφάλισης των πληροφοριών.
- Η απόφαση για το κάθε πότε θα γίνονται αντίγραφα ασφαλείας (backup) των αρχείων.
- Η παρακολούθηση της σωστής λειτουργίας της βάσης δεδομένων και η απαιτούμενη προσαρμογή της.

## **Μοντέλα & Εξέλιξη Βάσεων Δεδομένων**

### Οι Ιεραρχικές Βάσεις Δεδομένων

Στις Ιεραρχικές (Hierarchical) βάσεις δεδομένων τα δεδομένα αναπαρίστανται με δενδρικής μορφής δομές δεδομένων και συνδέονται μεταξύ τους με συνδέσμους (links). Η κάθε εγγραφή μπορεί να συνδέεται προς τα πάνω μόνο με μία άλλη εγγραφή (γονέας), ενώ μπορεί να έχει έως δύο εγγραφές που να εξαρτώνται απ' αυτήν (παιδιά). Υπάρχει μία μόνο εγγραφή ρίζα (root), απ' την οποία εξαρτώνται όλες οι άλλες εγγραφές της βάσης δεδομένων. Έχουν το μειονέκτημα ότι είναι πολύπλοκες στην επεξεργασία των εγγραφών τους (προσθήκη, διαγραφή, τροποποίηση).

### Οι Δικτυωτές Βάσεις Δεδομένων

Στις Δικτυωτές (Network) βάσεις δεδομένων τα δεδομένα αναπαρίστανται με δενδρικής μορφής δομές δεδομένων και συνδέονται μεταξύ τους με συνδέσμους (links), όπως ακριβώς και στις ιεραρχικές βάσεις δεδομένων, με την διαφορά ότι μια εγγραφή μπορεί να συνδέεται προς τα πάνω με περισσότερες από μία γονικές εγγραφές (parent records). Είναι λογικά πιο δύσχρηστες αλλά και πιο γρήγορες από τις ιεραρχικές βάσεις δεδομένων.

### Οι Σχεσιακές Βάσεις Δεδομένων

Στις Σχεσιακές (Relational) βάσεις δεδομένων, τα δεδομένα συνδέονται μεταξύ τους με σχέσεις (relations), οι οποίες προκύπτουν από τα κοινά πεδία που υπάρχουν σε διαφορετικά αρχεία. Τα αρχεία αποκαλούνται πίνακες (tables), οι εγγραφές γραμμές (rows) και τα πεδία στήλες (columns). Η ύπαρξη μιας κοινής τιμής στα πεδία δύο αρχείων καθορίζει και μια σχέση μεταξύ των γραμμών διαφορετικών πινάκων.

Οι σχεσιακές βάσεις δεδομένων έχουν το πλεονέκτημα ότι είναι λογικά κατανοητές και πολύ ευέλικτες και δεκτικές σε αλλαγές.

## Το Σχεσιακό Μοντέλο

Κάθε πεδίο ή στήλη ενός πίνακα, δέχεται τιμές οι οποίες ανήκουν σε ένα συγκεκριμένο και καθορισμένο εκ των προτέρων σύνολο τιμών (domain). Το είδος των τιμών αυτού του συνόλου καθορίζεται από τον τύπο δεδομένων του πεδίου του πίνακα, ο οποίος με τη σειρά του ορίζεται κατά το στάδιο της λογικής σχεδίασης της εφαρμογής. Έτσι, ένα πεδίο που εκφράζει την ηλικία, θα αναπαρίσταται από μια ακέραια ποσότητα. Όσον αφορά το σύνολο των επιτρεπτών τιμών για αυτό το πεδίο, αυτό εξαρτάται και πάλι από τη φύση του προβλήματος.

Σε ορισμένες περιπτώσεις είναι δυνατό να καθορίσουμε όχι μόνο το εύρος των τιμών για ένα πεδίο, αλλά και τη μορφή που θα έχουν οι τιμές που καταχωρούνται σε αυτό. Για παράδειγμα, ένας αριθμός αυτοκινήτου αποτελείται από τρία γράμματα και τέσσερα ψηφία – π.χ. ΤΚΕ-6960. Στην περίπτωση αυτή μπορούμε να γράψουμε τη μορφή των τιμών του πεδίου ως TTT-DDDD όπου το σύμβολο T υποδηλώνει χαρακτήρα (text) και παίρνει τιμές στο διάστημα A-Z, ενώ το σύμβολο D, υποδηλώνει ψηφίο (digit) και παίρνει τιμές στο διάστημα 0-9. Άλλες φορές πάλι είναι δυνατό να καθορίσουμε και κάποιες επιπλέον πληροφορίες. Για παράδειγμα σε μια εφαρμογή που καταχωρούμε αριθμητικές τιμές για διάφορες περιπτώσεις πεδίων, ίσως χρειαστεί να καθορίσουμε και τη μονάδα μέτρησης της τιμής, κάτι που συνηθίζεται σε όλα σχεδόν τα προγράμματα που χρησιμοποιούνται στο χώρο των επιχειρήσεων για εφαρμογές μηχανογράφησης.

### Κανόνες Ακεραιότητας

Οι κανόνες ακεραιότητας (integrity constraints) ενός σχεσιακού σχήματος μιας βάσης δεδομένων, διασφαλίζουν τη συνέπεια των δεδομένων της βάσης, και ισχύουν για κάθε στιγμιότυπό της.

Ο πρώτος από αυτούς τους κανόνες (key constraint) επιβάλλει την απόδοση μοναδικών τιμών σε όλα τα πεδία του πίνακα που έχουν χαρακτηριστεί ως υποψήφια κλειδιά (candidate keys). Με άλλα λόγια δεν θα πρέπει να υπάρχουν στον ίδιο πίνακα, δύο εγγραφές που να έχουν την ίδια τιμή στα υποψήφια κλειδιά τους – και προφανώς στο πρωτεύον κλειδί τους που είναι κάποιο από αυτά.

Ο δεύτερος κανόνας (entity integrity constraint) απαγορεύει την απόδοση της τιμής NULL, στο πρωτεύον κλειδί της κάθε εγγραφής του

πίνακα. Αυτό γίνεται επειδή ως γνωστόν το πρωτεύον κλειδί μας επιτρέπει να ξεχωρίσουμε τις εγγραφές του πίνακα, και επομένως εάν δύο εγγραφές έχουν τιμή NULL στο πρωτεύον κλειδί τους, δεν είναι δυνατή η διάκριση ανάμεσά τους.

Τέλος, ο τρίτος κανόνας (referential integrity constraint) αναφέρεται στις συσχετίσεις που υφίστανται ανάμεσα στους πίνακες της βάσης. Ο κανόνας αυτός λέει ότι εάν η τιμή κάποιου πεδίου μιας από τις εγγραφές ενός πίνακα αναφέρεται σε εγγραφή κάποιου άλλου πίνακα, αυτή θα πρέπει να είναι μια υπάρχουσα εγγραφή.

Η συσχέτιση ανάμεσα σε δύο πίνακες μιας σχεσιακής βάσης δεδομένων, μας επιτρέπει να ορίσουμε ένα επιπλέον τύπο κλειδιού, που ονομάζεται ξένο κλειδί (foreign key). Ένα πεδίο ενός πίνακα A ονομάζεται ξένο κλειδί, όταν αποτελεί πρωτεύον κλειδί κάποιου άλλου πίνακα B και έχει τοποθετηθεί στον πίνακα A ως αποτέλεσμα της συσχέτισης που υφίσταται ανάμεσα στους πίνακες A και B σύμφωνα με το μοντέλο οντοτήτων – συσχετίσεων.

# Συστήματα Διαχείρισης Βάσεων Δεδομένων

## Γενικές έννοιες

Ένα Σύστημα Διαχείρισης Βάσης Δεδομένων (ΣΔΒΔ) [Database Management System] είναι ένα πακέτο λογισμικού που στοχεύει στη συστηματική αποθήκευση, αναζήτηση και διαχείριση των δεδομένων.

Ένα τυπικό περιβάλλον λειτουργίας ΣΔΒΔ αποτελείται από τέσσερα κύρια μέρη:

1. Τα δεδομένα (data)
2. Το υλικό (hardware)
3. Το λογισμικό (software)
4. Τους χρήστες (users)

Μία Βάση Δεδομένων είναι μια συλλογή συσχετιζόμενων δεδομένων. Τα δεδομένα που αποθηκεύονται σε μια βάση μπορεί να είναι κάποια πεδία ή γραμμές που απαρτίζουν λόγου χάρη μια ατζέντα ονομάτων, διευθύνσεων και τηλεφωνικών αριθμών. Επίσης, μια Βάση Δεδομένων μπορεί να περιέχει χιλιάδες εγγραφές που περιγράφουν τον κατάλογο, τις προσκλήσεις, τις παραγγελίες και τις μισθοδοσίες μιας μεγάλης εταιρίας.

Η Βάση Δεδομένων (ΒΔ) είναι το τμήμα του ΣΔΒΔ που αποθηκεύει τα δεδομένα του οργανισμού. Πριν από την ανάπτυξη των ΒΔ τα δεδομένα αποθηκεύονταν σε ένα σύνολο διαφορετικών αρχείων. Οι ΒΔ δίνουν τη δυνατότητα αποθήκευσης των δεδομένων με ολοκληρωμένο (integrated) και καταμερισμένο (shared) τρόπο. Ο όρος ολοκληρωμένος αναφέρεται στο γεγονός ότι η ΒΔ αποτελείται από την ένωση διακεκριμένων αρχείων στα οποία η αποθήκευση των πλεοναζόντων δεδομένων έχει περιοριστεί στο ελάχιστο δυνατό. Ο όρος καταμερισμένος αναφέρεται στο γεγονός ότι τα ίδια δεδομένα μπορούν να τα χρησιμοποιούν διαφορετικοί χρήστες.

Τα δεδομένα μιας ΒΔ αφορούν:

- Τις *οντότητες* (*entities*) που απασχολούν τον οργανισμό: ως οντότητα ορίζουμε την πληροφορία πραγμάτων του αληθινού κόσμου (δηλαδή κάτι πραγματικό και όχι κάτι φανταστικό ή ιδεατό) που θα αποθηκεύσουμε στη ΒΔ. Πχ μπορεί να επιλέξουμε να αποθηκεύσουμε πληροφορία σχετικά με υπαλλήλους και τα τμήματα για τα οποία εργάζονται. Σε αυτήν τη περίπτωση ένας υπάλληλος θεωρείται μία οντότητα και ένα τμήμα μία άλλη.

και

- Τις *συσχετίσεις (relationships)* μεταξύ των οντοτήτων. Σύμφωνα με το παραπάνω παράδειγμα οι συσχετίσεις είναι οι σύνδεσμοι μεταξύ της οντότητας «υπάλληλος» και της οντότητας «τμήμα». Για παράδειγμα: ένας υπάλληλος δουλεύει για ένα τμήμα. Η φράση «δουλεύει για» αναπαριστά τη συσχέτιση των οντοτήτων «υπάλληλος» και «τμήμα». Οι συσχετίσεις έχουν διάφορες διαβαθμίσεις: από μία οντότητα σε μία άλλη, από μία σε πολλές ή από πολλές σε μία ή από πολλές σε πολλές.

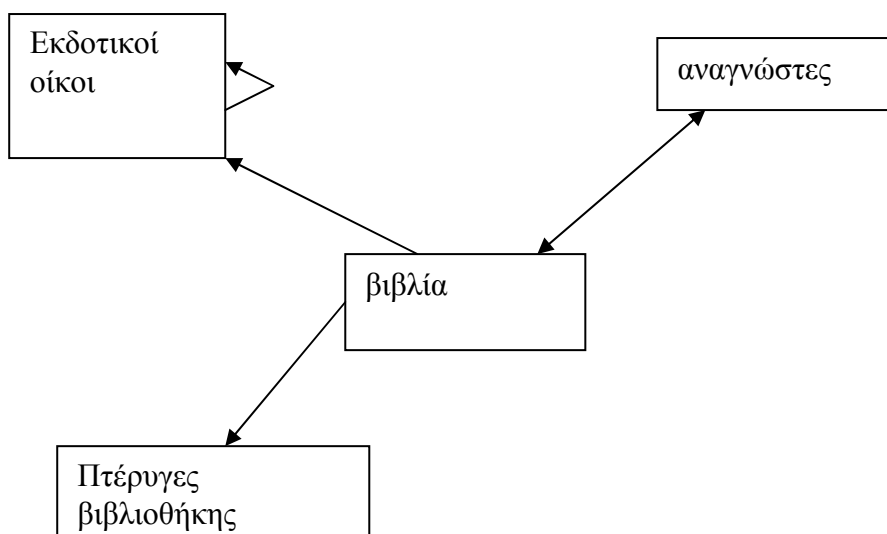
### Παράδειγμα

Το σχήμα 1 παρακάτω περιγράφει τα δεδομένα μιας ΒΔ που αφορούν σε μία δανειστική βιβλιοθήκη. Τα παραλληλόγραμμα αντιστοιχούν στις οντότητες «βιβλία», «αναγνώστες», «εκδοτικοί οίκοι» και «πτέρυγες της βιβλιοθήκης». Τα βέλη αντιστοιχούν στις συσχετίσεις και είναι διπλής ή μονής κατεύθυνσης για να δείξουν την κατεύθυνση στην οποία μπορεί να λάβει χώρα η συσχέτιση. Πχ. η σχέση «αναγνώστες» - «βιβλία» μπορεί να χρησιμοποιηθεί για να απαντήσουμε στις ερωτήσεις:

1. Βρες τα βιβλία που έχει δανειστεί κάποιος αναγνώστης
2. Βρες τους αναγνώστες που έχουν δανειστεί κάποιο βιβλίο

Τα βέλη συνήθως συσχετίζουν 2 οντότητες, αλλά μπορεί να συσχετίσουν και περισσότερες.

- Υπάρχουν βέλη που συσχετίζουν μια οντότητα με τον εαυτό της. Οι συσχετίσεις αυτές αντιστοιχούν σε αναδρομικούς (recursive) τύπους δεδομένων. Πχ. το βέλος που συσχετίζει την οντότητα «εκδοτικοί οίκοι» με τον εαυτό της θέλει να δείξει ότι ορισμένοι εκδοτικοί οίκοι ανήκουν σε κάποιους άλλους.



### Σχήμα 1: Παράδειγμα οντοτήτων και συσχετίσεων

Τέλος, ένα ΣΔΒΔ απευθύνεται σε μια ποικιλία χρηστών που περιλαμβάνει:

- α) προγραμματιστές εφαρμογών (Application Programmers),
- β) τελικούς χρήστες (End users) και
- γ) διαχειριστές της ΒΔ (Database Administrators)

Η δουλειά του προγραμματιστή εφαρμογών είναι η ανάπτυξη προγραμμάτων που επεξεργάζονται τη ΒΔ. Ο χρήστης ενδιαφέρεται για την αναζήτηση δεδομένων της ΒΔ μέσα από τη χρήση μιας γλώσσας υψηλού επιπέδου που αναφέρεται σαν γλώσσα ερωτήσεων (query language). Ο ρόλος του διαχειριστή της ΒΔ είναι ο καθορισμός των δεδομένων της ΒΔ, η επιλογή της φυσικής δομής και του τρόπου προσπέλασης στη ΒΔ, η επικοινωνία με τους χρήστες και ο καθορισμός διαφόρων ελέγχων και η καταγραφή διαφόρων στατιστικών χρήσης.

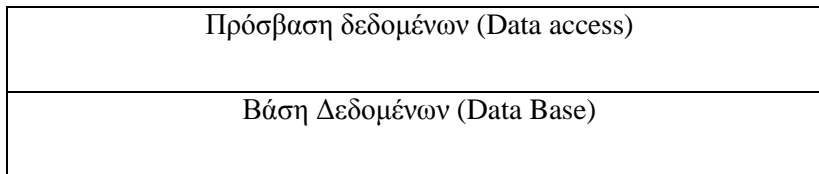
#### Αρχιτεκτονική ενός τυπικού ΣΔΒΔ

Το κύριο πλεονέκτημα των ΣΔΒΔ είναι ότι επιτρέπουν στους χρήστες να χρησιμοποιήσουν τις αποθηκευμένες πληροφορίες στη ΒΔ χωρίς να απασχολούνται με τον τρόπο φύλαξής της στα αποθηκευτικά μέσα του Η/Υ. Κατά συνέπεια το ΣΔΒΔ λειτουργεί σαν μεταφραστής που δέχεται σαν είσοδο τις απαιτήσεις του χρήστη και παράγει σαν αποτέλεσμα τον τρόπο που θα εξυπηρετηθεί η απαίτηση αυτή από τον Η/Υ.

Ένα ΣΔΒΔ είναι ένα σύνολο από συστατικά στοιχεία που χρησιμεύουν στον καθορισμό, τη δημιουργία και τη διαχείριση μιας ΒΔ. Όταν αναφερόμαστε σε ένα ΣΔΒΔ συνήθως εννοούμε ένα Σχεσιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων (ΣΣΔΒΔ). Οι Σχεσιακές ΒΔ αποθηκεύουν και διαχειρίζονται τις συσχετίσεις μεταξύ των δεδομένων.

Στο σχήμα 2 που ακολουθεί διαφαίνεται η αρχιτεκτονική ενός τυπικού ΣΔΒΔ και ακολουθεί η επεξήγηση των συστατικών μερών που τη συνθέτουν.

Επιφάνεια εφαρμογών (Application Interface)
«Μεταφραστής-Διερμηνευτής» (Interpreter)
Ελεγκτής ερωτήματος (Query evaluator)



**Σχήμα 2: Αρχιτεκτονική του ΣΔΒΔ**

- *Επιφάνεια εφαρμογών (Application Interface)*: η επιφάνεια η οποία δέχεται την απαίτηση του χρήστη.
- *Μεταφραστής-Διερμηνευτής (Interpreter)*: ένας parser που ελέγχει τη σύνταξη των δηλώσεων του εισερχόμενου ερωτήματος και τις μεταφράζει/ μετατρέπει σε μια εσωτερική αναπαράσταση. Όταν θέτουμε ένα ερώτημα στη βάση εμείς βλέπουμε κάποια λέξη ή λέξεις γραμμένες σε φυσική γλώσσα. Ωστόσο, το σύστημα αντιλαμβάνεται το ερώτημα (δηλ. τις λέξεις-εντολές που έχουμε εισάγει) σαν κώδικα της γλώσσας προγραμματισμού που χρησιμοποιεί η βάση. Έτσι, ο parser μεταφράζει στο σύστημα αυτό που εμείς έχουμε πληκτρολογήσει σε φυσική γλώσσα δημιουργώντας για αυτό (το σύστημα δηλαδή) μια εσωτερική αναπαράσταση σε κώδικα.
- *Ελεγκτής ερωτήματος (Query evaluator)*: ένα σύστημα αξιολόγησης που ελέγχει αν το ερώτημα (query) είναι σωστό και το μεταφράζει για να εκτελεστεί.
- *Πρόσβαση δεδομένων (Data access)*: ένα πρόγραμμα που ελέγχει τη σωστή και ασφαλή πρόσβαση στα δεδομένα.
- *Βάση Δεδομένων (Data Base)*: Τα δεδομένα αυτά καθαυτά που φυλάσσονται στη ΒΔ.



## Πλεονεκτήματα χρήσης ΒΔ

Παρακάτω παραθέτουμε ορισμένα από τα πλεονεκτήματα που επιφέρει η χρήση των ΒΔ:

- Μείωση πλεοναζόντων δεδομένων
- Εύκολη επεκτασιμότητα
- Εύκολη ανάπτυξη μηχανισμών ασφαλείας
- Βελτιστοποίηση μεθόδων ανάκτησης δεδομένων
- Ανεξαρτησία δεδομένων: ορίζουμε σαν ανεξαρτησία δεδομένων την απαίτηση να μη μεταβάλλονται τα προγράμματα εφαρμογών όταν αλλάζει ο τρόπος οργάνωσης και προσπέλασης των δεδομένων. Ένα ΣΔΒΔ και μια ΒΔ παρέχουν ανεξαρτησία δεδομένων – προγραμμάτων που σημαίνει ότι η μέθοδος της αποθήκευσης των δεδομένων, η ταξινόμηση της αποθηκευμένης πληροφορίας και ο τρόπος που ο σκληρός δίσκος διαχειρίζεται τα δεδομένα είναι ανεξάρτητα από το λογισμικό που προσπελάει τα δεδομένα (Williams and Lane, 2002). Επίσης, το ΣΔΒΔ υποστηρίζει διαφορετικές απεικονίσεις των ίδιων δεδομένων από διαφορετικές εφαρμογές.
- Ευκολία διαχείρισης σύνθετων συσχετίσεων.
- Ταχύτητα αναζήτησης συσχετιζόμενων δεδομένων.

## Αρχές σχεδιασμού ΒΔ

Όταν σχεδιάζουμε μια ΒΔ πρέπει να λαμβάνουμε υπ' όψιν τα εξής:

- Τί πληροφορία χρειάζεται να αποθηκευτεί; Δηλαδή, τίνος πράγματος ή ποιιάς οντότητας την πληροφορία είναι απαραίτητο να αποθηκεύσουμε;
- Τι συσχετίσεις υπάρχουν ανάμεσα στα δεδομένα;
- Τί ερωτήσεις θα κάνουμε στη ΒΔ; (αυτό ονομάζεται ερωτήματα, queries). Όταν ασχολούμαστε με τα ερωτήματα πρέπει να έχουμε κατά νου τους κανονισμούς του οργανισμού για τον οποίο σχεδιάζουμε τη ΒΔ: δηλαδή, ποια είναι εκείνα τα πράγματα των οποίων τα δεδομένα πρέπει να αποθηκεύσουμε;

Αναλογιζόμενοι πάντα τους παραπάνω προβληματισμούς θα πρέπει να δομήσουμε με τέτοιο τρόπο τη ΒΔ μας ώστε να αποφύγουμε προβλήματα δομής όπως την απώλεια και τις ανωμαλίες δεδομένων.

*Μέρος Β΄*

*Βάση Δεδομένων για*

*Ηλεκτρονικό Αρχείο Άρθρων*

*Περιοδικών*

# Μέρος Β΄

## Βάση Δεδομένων για Ηλεκτρονικό Αρχείο Άρθρων Περιοδικών

### Γενική περιγραφή της ΒΔ.

Παρουσιάζουμε μια Βάση Δεδομένων (ΒΔ) για την υλοποίηση ενός ηλεκτρονικού αρχείου για άρθρα περιοδικών. Στην ΒΔ περιέχονται πληροφορίες για:

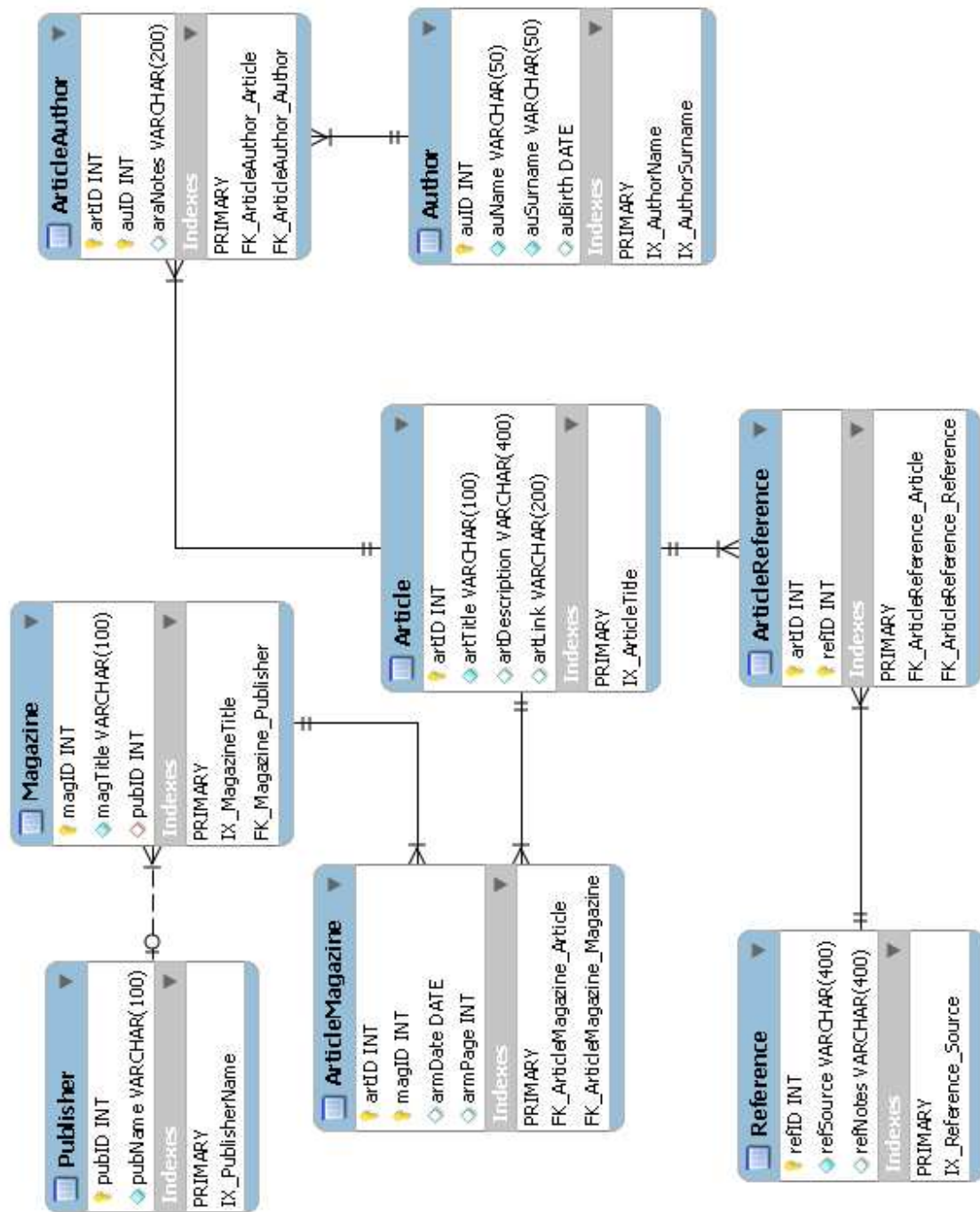
- Τα άρθρα (πίνακας *Article*)
- Τους συγγραφείς των άρθρων (πίνακας *Author*)
- Τις πηγές-αναφορές που εμφανίζονται στα άρθρα (πίνακας *Reference*)
- Τα περιοδικά στα οποία εμφανίζονται τα άρθρα (πίνακας *Magazine*)
- Τους εκδότες των περιοδικών (πίνακας *Publisher*)

Πέρα από την καταγραφή πληροφοριών για τα αντικείμενα της εφαρμογής, είναι πολύ σημαντικό να περιγράψουμε και τις σχέσεις που υπάρχουν ανάμεσά τους. Πιο συγκεκριμένα:

- *Συγγραφή άρθρων*: Κάθε συγγραφέας μπορεί να έχει συγγράψει 0 ή περισσότερα άρθρα. Κάθε άρθρο έχει συγγραφεί από 1 ή περισσότερους συγγραφείς. Δηλαδή ανάμεσα στα άρθρα και τους συγγραφείς υπάρχει μια συσχέτιση πολλά-προς-πολλά. Η συσχέτιση αυτή υλοποιείται μέσω του πίνακα *ArticleAuthor*.
- *Δημοσίευση άρθρων σε περιοδικά*: Ένα περιοδικό περιλαμβάνει 1 ή περισσότερα άρθρα. Παρότι ένα άρθρο τυπικά δημοσιεύεται σε ένα

περιοδικό, δεν υπάρχει κάτι που να αποκλείει την αναδημοσίευση του άρθρου σε ένα δεύτερο (τρίτο, κ.λπ.) περιοδικό. Κατά συνέπεια και η συσχέτιση ανάμεσα στα άρθρα και στα περιοδικά είναι πολλά-προς-πολλά και αυτό πρέπει να αντικατοπτριστεί στην ΒΔ. Η συσχέτιση υλοποιείται μέσω του πίνακα *ArticleMagazine*.

- *Εμφάνιση πηγών σε άρθρα:* Όπως είναι προφανές, κάθε άρθρο μπορεί να περιλαμβάνει 0 ή περισσότερες πηγές (references) ενώ και κάθε πηγή μπορεί να περιλαμβάνεται σε πολλά άρθρα. Δηλαδή, όπως και τις προηγούμενες περιπτώσεις η συσχέτιση ανάμεσα στα άρθρα και τις πηγές είναι τύπου πολλά-προς-πολλά. Η συσχέτιση υλοποιείται μέσω του πίνακα *ArticleReference*.
- *Έκδοση περιοδικών:* Κάθε εκδοτικός οργανισμός μπορεί να εκδίδει πολλά περιοδικά, ενώ κάθε περιοδικό εκδίδεται από ένα μόνο εκδότη. Η συσχέτιση είναι ένα-προς-πολλά και για την υλοποίησή της χρησιμοποιείται ένα πεδίο (*pubID*) στον πίνακα *Magazine*.

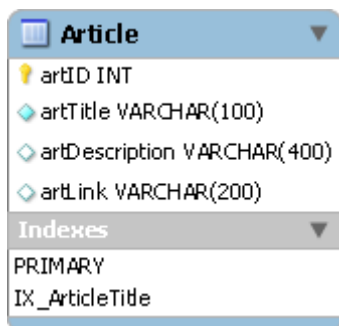


Εικόνα 1 - γενική όψη της ΒΔ

## Πίνακες της ΒΔ

Παρακάτω περιγράφουμε αναλυτικά τους πίνακες της ΒΔ.

### Article



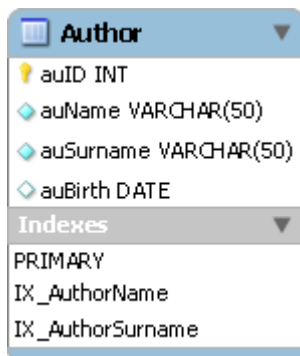
Article	
artID	INT
artTitle	VARCHAR(100)
artDescription	VARCHAR(400)
artLink	VARCHAR(200)
Indexes	
PRIMARY	
IX_ArticleTitle	

Ο πίνακας *Article* περιέχει πληροφορίες για τα άρθρα. Κάθε εγγραφή του πίνακα περιγράφει ένα ξεχωριστό-διαφορετικό άρθρο. Τα δεδομένα που μας ενδιαφέρουν και καταγράφονται σε αυτόν είναι:

- artID – Πρόκειται για έναν ακέραιο που αποτελεί τον μοναδικό κωδικό του κάθε άρθρου. Το πεδίο αυτό είναι και το *Πρωτεύον Κλειδί* (*Primary Key*) του πίνακα.
- artTitle – Είναι ο τίτλος του άρθρου. Πρόκειται για αλφαριθμητικό του οποίου το μέγιστο μήκος είναι 100 χαρακτήρες. Η συμπλήρωση του πεδίου αυτού είναι υποχρεωτική (NOT NULL).
- artDescription – Πρόκειται για αλφαριθμητικό του οποίου το μέγιστο μήκος είναι 400 χαρακτήρες. Στο πεδίο αυτό περιέχονται πιο αναλυτικές πληροφορίες για το άρθρο. Για παράδειγμα θα μπορούσαμε να εισάγουμε μια περίληψη του άρθρου. Καθώς η πληροφορία αυτή δεν είναι πάντα διαθέσιμη το πεδίο επιτρέπεται να μείνει και κενό.
- artLink – Σε αυτό το πεδίο μπορούμε να εισάγουμε έναν σύνδεσμο προς την τοποθεσία του άρθρου στο Διαδίκτυο εφόσον αυτή η πληροφορία είναι διαθέσιμη. Η εισαγωγή δεδομένων σε αυτό το πεδίο (όπως και για το artDescription) είναι προαιρετική.

Στον πίνακα έχει δημιουργηθεί το ευρετήριο (index) *IX\_ArticleTitle* το οποίο είναι ορισμένο πάνω στο πεδίο *artTitle*. Κατά συνέπεια οι ερωτήσεις με κριτήρια βασισμένα στον τίτλο ενός άρθρου θα απαντώνται πιο αποδοτικά.

## Author



Author	
auID	INT
auName	VARCHAR(50)
auSurname	VARCHAR(50)
auBirth	DATE
Indexes	
PRIMARY	
IX_AuthorName	
IX_AuthorSurname	

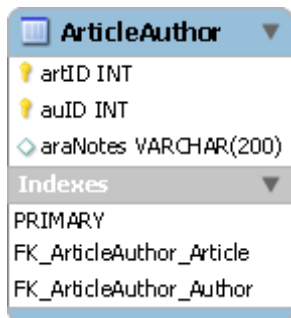
Ο πίνακας **Author** περιέχει πληροφορίες για τους συγγραφείς. Κάθε εγγραφή του πίνακα περιγράφει ένα ξεχωριστό-διαφορετικό συγγραφέα. Τα δεδομένα που μας ενδιαφέρουν και καταγράφονται σε αυτόν είναι:

- auID – Πρόκειται για έναν ακέραιο που αποτελεί τον μοναδικό κωδικό κάθε συγγραφέα. Το πεδίο αυτό είναι και το *Πρωτεύον Κλειδί* (*Primary Key*) του πίνακα.
- auName – Το όνομα του συγγραφέα. Πρόκειται για αλφαριθμητικό του οποίου το μέγιστο μήκος είναι 50. Η συμπλήρωση του πεδίου αυτού είναι υποχρεωτική (NOT NULL).
- auSurname – Το επίθετο του συγγραφέα. Πρόκειται για αλφαριθμητικό του οποίου το μέγιστο μήκος είναι 50. Η συμπλήρωση του πεδίου αυτού είναι υποχρεωτική (NOT NULL).
- auBirth – Η ημερομηνία γέννησης του συγγραφέα. Δεδομένου ότι η πληροφορία αυτή μπορεί να μην είναι πάντα διαθέσιμη, η συμπλήρωση του πεδίου είναι προαιρετική.

Στον πίνακα έχουν δημιουργηθεί τα ευρετήρια (index) *IX\_AuthorName* και *IX\_AuthorSurname* τα οποία είναι ορισμένα πάνω στο πεδία *auName* και *auSurname* αντίστοιχα. Κατά συνέπεια οι ερωτήσεις με κριτήρια βασισμένα στο όνομα ή το επίθετο του συγγραφέα θα απαντώνται πιο αποδοτικά.



## ArticleAuthor



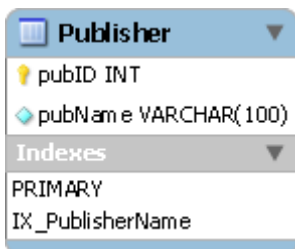
ArticleAuthor	
artID	INT
auID	INT
araNotes	VARCHAR(200)
Indexes	
PRIMARY	
FK_ArticleAuthor_Article	
FK_ArticleAuthor_Author	

Πρόκειται για έναν ενδιάμεσο πίνακα που χρησιμοποιείται για την υλοποίηση της συσχέτισης πολλά-προς-πολλά ανάμεσα στους πίνακες *Article* και *Author*. Κάθε εγγραφή του πίνακα περιγράφει το γεγονός ότι ο καθοριζόμενος συγγραφέας έχει συμμετάσχει στην συγγραφή του καθοριζόμενου άρθρου. Τα πεδία που περιέχει είναι:

- artID – Καθορίζει το άρθρο μέσω του κωδικού του. Μαζί με το *auID* αποτελεί το *Πρωτεύον Κλειδί (Primary Key)* του πίνακα.
- auID – Καθορίζει τον συγγραφέα μέσω του κωδικού του. Μαζί με το *artID* αποτελεί το *Πρωτεύον Κλειδί (Primary Key)* του πίνακα.
- araNotes – Επιπλέον πληροφορία που σχετίζεται με τη συγγραφή. Για παράδειγμα μπορεί να περιγράφει το ρόλο του συγγραφέα (μετάφραση, κ.λπ.).

Στον πίνακα έχουν καθοριστεί δύο *Περιορισμοί Ξένου Κλειδιού (Foreign Key Constraints)*. Τα *FK\_ArticleAuthor\_Article* και *FK\_ArticleAuthor\_Author* προς τους πίνακες *Article* και *Author* αντίστοιχα. Οι περιορισμοί αυτοί σε συνδυασμό με τους αντίστοιχους κανόνες Αναφορικής Ακεραιότητας (Referential Integrity Rules) εξασφαλίζουν την εγκυρότητα των δεδομένων που υπάρχουν στον πίνακα.

## Publisher



Publisher	
pubID	INT
pubName	VARCHAR(100)
Indexes	
PRIMARY	
IX_PublisherName	

Ο πίνακας **Publisher** περιέχει πληροφορίες για τους εκδοτικούς οργανισμούς. Κάθε εγγραφή του πίνακα περιγράφει ένα ξεχωριστό-διαφορετικό εκδότη. Τα δεδομένα που μας ενδιαφέρουν και καταγράφονται σε αυτόν είναι:

- pubID – Πρόκειται για έναν ακέραιο που αποτελεί τον μοναδικό κωδικό κάθε εκδότη. Το πεδίο αυτό είναι και το *Πρωτεύον Κλειδί* (*Primary Key*) του πίνακα.
- pubName – Το όνομα του εκδοτικού οργανισμού. Πρόκειται για αλφαριθμητικό του οποίου το μέγιστο μήκος είναι 100 χαρακτήρες. Η συμπλήρωση του πεδίου αυτού είναι υποχρεωτική (NOT NULL).

Στον πίνακα έχει δημιουργηθεί το ευρετήριο (index) *IX\_PublisherTitle* το οποίο είναι ορισμένο πάνω στο πεδίο *pubName*. Κατά συνέπεια οι ερωτήσεις με κριτήρια βασισμένα στο όνομα ενός εκδοτικού οργανισμού θα απαντώνται πιο αποδοτικά.

## Magazine



Magazine	
magID	INT
magTitle	VARCHAR(100)
pubID	INT
Indexes	
PRIMARY	
IX_MagazineTitle	
FK_Magazine_Publisher	

Ο πίνακας *Magazine* περιέχει πληροφορίες για τα περιοδικά. Κάθε εγγραφή του πίνακα περιγράφει ένα ξεχωριστό-διαφορετικό περιοδικό. Τα δεδομένα που μας ενδιαφέρουν και καταγράφονται σε αυτόν είναι:

- magID – Πρόκειται για έναν ακέραιο που αποτελεί τον μοναδικό κωδικό κάθε περιοδικού. Το πεδίο αυτό είναι και το *Πρωτεύον Κλειδί* (*Primary Key*) του πίνακα.
- magTitle – Ο τίτλος του περιοδικού. Πρόκειται για αλφαριθμητικό του οποίου το μέγιστο μήκος είναι 100 χαρακτήρες. Η συμπλήρωση του πεδίου αυτού είναι υποχρεωτική (NOT NULL).
- pubID – Ο κωδικός του εκδότη που εκδίδει το περιοδικό. Μέσω αυτού του πεδίου υλοποιείται η συσχέτιση ένα-προς-πολλά ανάμεσα στους πίνακες *Publisher* και *Magazine*.

Στον πίνακα έχει δημιουργηθεί το ευρετήριο (index) *IX\_MagazineTitle* το οποίο είναι ορισμένο πάνω στο πεδίο *magTitle*. Κατά συνέπεια οι ερωτήσεις με κριτήρια βασισμένα στο όνομα ενός περιοδικού θα απαντώνται πιο αποδοτικά.

## ArticleMagazine



Column	Type
artID	INT
magID	INT
armDate	DATE
armPage	INT

Indexes

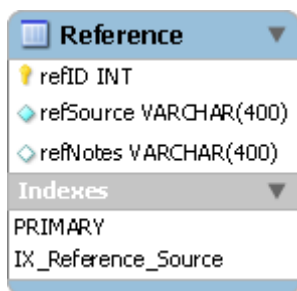
Index Name	Type
PRIMARY	PRIMARY
FK_ArticleMagazine_Article	FOREIGN KEY
FK_ArticleMagazine_Magazine	FOREIGN KEY

Πρόκειται για έναν ενδιάμεσο πίνακα που χρησιμοποιείται για την υλοποίηση της συσχέτισης πολλά-προς-πολλά ανάμεσα στους πίνακες *Article* και *Magazine*. Κάθε εγγραφή του πίνακα περιγράφει το γεγονός ότι το καθοριζόμενο άρθρο εμφανίστηκε στο καθοριζόμενο περιοδικό. Τα πεδία που περιέχει είναι:

- artID – Καθορίζει το άρθρο μέσω του κωδικού του. Μαζί με το *magID* αποτελεί το *Πρωτεύον Κλειδί (Primary Key)* του πίνακα.
- magID – Καθορίζει το περιοδικό μέσω του κωδικού του. Μαζί με το *artID* αποτελεί το *Πρωτεύον Κλειδί (Primary Key)* του πίνακα.
- armDate – Η ημερομηνία στην οποία το άρθρο εκδόθηκε στο συγκεκριμένο περιοδικό. Μπορεί να χρησιμοποιηθεί για να καθοριστεί επακριβώς το τεύχος του περιοδικού στο οποίο εμφανίστηκε το άρθρο. Η συμπλήρωση του πεδίου είναι προαιρετική καθώς η πληροφορία μπορεί να μην είναι διαθέσιμη ή να μην είναι απαραίτητη.
- armPage – Η σελίδα του περιοδικού στην οποία εμφανίζεται το άρθρο. Στις περισσότερες περιπτώσεις όπου το άρθρο καλύπτει περισσότερες από μια σελίδες, το πεδίο αυτό αναφέρεται στην σελίδα στην οποία ξεκινά το άρθρο. Όπως και για το *armData*, η συμπλήρωση του πεδίου είναι προαιρετική καθώς η πληροφορία μπορεί να μην είναι διαθέσιμη ή να μην είναι απαραίτητη.

Στον πίνακα έχουν καθοριστεί δύο *Περιορισμοί Ξένου Κλειδιού (Foreign Key Constraints)*. Τα *FK\_ArticleMagazine\_Article* και *FK\_ArticleMagazine\_Magazine* προς τους πίνακες *Article* και *Magazine* αντίστοιχα. Οι περιορισμοί αυτοί σε συνδυασμό με τους αντίστοιχους κανόνες Αναφορικής Ακεραιότητας (Referential Integrity Rules) εξασφαλίζουν την εγκυρότητα των δεδομένων που υπάρχουν στον πίνακα.

## Reference



Column	DataType
refID	INT
refSource	VARCHAR(400)
refNotes	VARCHAR(400)

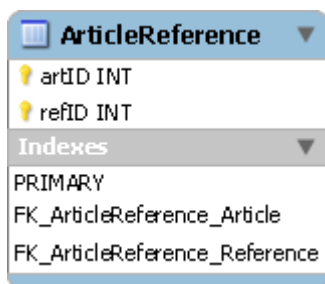
Index Name	Index Type
IX_Reference_Source	PRIMARY

Ο πίνακας **Reference** περιέχει πληροφορίες για τις πηγές-αναφορές που εμφανίζονται στα άρθρα. Κάθε εγγραφή του πίνακα περιγράφει μια ξεχωριστή-διαφορετική πηγή. Τα δεδομένα που μας ενδιαφέρουν και καταγράφονται σε αυτόν είναι:

- refID – Πρόκειται για έναν ακέραιο που αποτελεί τον μοναδικό κωδικό κάθε πηγής. Το πεδίο αυτό είναι και το *Πρωτεύον Κλειδί* (*Primary Key*) του πίνακα.
- refSource – Το κείμενο της πηγής-αναφοράς. Πρόκειται για αλφαριθμητικό του οποίου το μέγιστο μήκος είναι 400 χαρακτήρες. Η συμπλήρωση του πεδίου αυτού είναι υποχρεωτική (NOT NULL).
- refNotes – Επιπλέον πληροφορίες που μπορεί να υπάρχουν για κάποια πηγή-αναφορά. Πρόκειται για αλφαριθμητικό του οποίου το μέγιστο μήκος είναι 400 χαρακτήρες. Η συμπλήρωση του πεδίου είναι προαιρετική.

Στον πίνακα έχει δημιουργηθεί το ευρετήριο (index) *IX\_Reference\_Source* το οποίο είναι ορισμένο πάνω στο πεδίο *refSource*. Κατά συνέπεια οι ερωτήσεις με κριτήρια βασισμένα στο κείμενο μιας πηγής θα απαντώνται πιο αποδοτικά.

## ArticleReference



ArticleReference	
artID	INT
refID	INT
Indexes	
PRIMARY	
FK_ArticleReference_Article	
FK_ArticleReference_Reference	

Πρόκειται για έναν ενδιάμεσο πίνακα που χρησιμοποιείται για την υλοποίηση της συσχέτισης πολλά-προς-πολλά ανάμεσα στους πίνακες *Article* και *Reference*. Κάθε εγγραφή του πίνακα περιγράφει το γεγονός ότι η καθοριζόμενη πηγή εμφανίζεται στο καθοριζόμενο άρθρο. Τα πεδία που περιέχει είναι:

- artID – Καθορίζει το άρθρο μέσω του κωδικού του. Μαζί με το *refID* αποτελεί το *Πρωτεύον Κλειδί (Primary Key)* του πίνακα.
- refID – Καθορίζει την πηγή μέσω του κωδικού της. Μαζί με το *artID* αποτελεί το *Πρωτεύον Κλειδί (Primary Key)* του πίνακα.

Στον πίνακα έχουν καθοριστεί δύο *Περιορισμοί Ξένου Κλειδιού (Foreign Key Constraints)*. Τα *FK\_ArticleReference\_Article* και *FK\_ArticleReference\_Reference* προς τους πίνακες *Article* και *Reference* αντίστοιχα. Οι περιορισμοί αυτοί σε συνδυασμό με τους αντίστοιχους κανόνες Αναφορικής Ακεραιότητας (*Referential Integrity Rules*) εξασφαλίζουν την εγκυρότητα των δεδομένων που υπάρχουν στον πίνακα.

## Παρουσίαση Παραδειγμάτων

Παρακάτω παρουσιάζουμε μερικά απλά παραδείγματα και τον ακριβή τρόπο με τον οποίο αυτά θα περιγραφούν και θα αποθηκευτούν στην ΒΔ. Ξεκινάμε με μια σύντομη περιγραφή του συνολικού παραδείγματος και στην συνέχεια δείχνουμε αναλυτικά πως τα δεδομένα εμφανίζονται στους διάφορους πίνακες.

*Παράδειγμα: Επιθυμούμε να καταχωρήσουμε στην ΒΔ πληροφορία σχετικά με 5 άρθρα (ένα μείγμα από ερευνητικά papers αλλά και απλά άρθρα). Μερικά από αυτά τα άρθρα έχουν πολλαπλούς συγγραφείς. Θέλουμε επίσης να αποθηκεύσουμε την πληροφορία σχετικά με το σε ποιο περιοδικό εμφανίστηκαν και πότε. Ένα από τα ερευνητικά papers περιέχει αναφορές σε παλιότερες εργασίες και παρουσιάζουμε τον τρόπο αποθήκευσης αυτής της πληροφορίας.*

### Article

artID	artTitle	artDescription	artLink
1	A Relational Model of Data for Large Shared Data Banks	Περιγράφει το σχεσιακό μοντέλο.	<a href="http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1.1.1.1">http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1.1.1.1</a>
2	On the semantics of the relational data model	NULL	<a href="http://portal.acm.org/ft_gateway.cfm?id=500110&amp;type=journal_article">http://portal.acm.org/ft_gateway.cfm?id=500110&amp;type=journal_article</a>
3	Service Congestion: The Problem, and an Optimized Service Composition Architecture as a Solution	NULL	NULL
4	Unlocking Android's Multimedia Muscle	Επιλογές για την διαχείριση πολυμεσικού περιεχομένου στην πλατφόρμα της Google.	NULL
5	Windows Phone 7: Microsoft Ups the Ante	Παρουσίαση των Windows Phone 7.	NULL

Στο παραπάνω απόσπασμα από τον πίνακα *Article* εμφανίζονται τα 5 άρθρα του παραδείγματος. Όπως προαναφέρθηκε τα 3 πρώτα άρθρα είναι ερευνητικά papers στον τομέα των Βάσεων Δεδομένων και γενικότερα της Πληροφορικής, ενώ τα 2 τελευταία είναι άρθρα από ένα περιοδικό υπολογιστών (συγκεκριμένα το τεύχος Δεκεμβρίου 2010 της αγγλικής έκδοσης του PC World©).

### Author

auID	auName	auSurname	auBirth
1	Edgar	Codd	1923-08-23
2	Hans Albrecht	Schmid	NULL
3	J Richard	Swenson	NULL
4	Robert	Anthony	NULL
5	JR	Raphael	NULL

Στο παραπάνω απόσπασμα από τον πίνακα *Author* εμφανίζονται οι συγγραφείς των άρθρων χωρίς ωστόσο να καθορίζεται πουθενά ποιο άρθρο έχει γραφεί από ποιον συγγραφέα. Η συσχέτιση αυτή (ανάμεσα σε άρθρα και συγγραφείς) γίνεται στον πίνακα *ArticleAuthor* που παρουσιάζεται παρακάτω.

## ArticleAuthor

artID	auID	araNotes
1	1	NULL
2	2	NULL
2	3	NULL
3	2	NULL
4	5	NULL
5	4	NULL

Η σύνδεση των άρθρων με τους συγγραφείς γίνεται με βάση τους κωδικούς τους (*artID* και *auID* αντίστοιχα). Για παράδειγμα στο παραπάνω απόσπασμα από τον πίνακα *ArticleAuthor*:

- Η πρώτη εγγραφή δηλώνει ότι το άρθρο με κωδικό 1 έχει γραφτεί από τον συγγραφέα με κωδικό 1. Μεταφράζοντας και ανατρέχοντας στους πίνακες *Article* και *Author* το άρθρο 'A Relational Model of Data for Large Shared Data Banks' (που αποτελεί στην ουσία τον ορισμό του Σχεσιακού Μοντέλου) έχει γραφτεί από τον *Edgar Codd*.
- Η δεύτερη και τρίτη εγγραφή έχουν τον ίδιο κωδικό άρθρου (2) υποδηλώνοντας έτσι, ότι το άρθρο έχει γραφτεί από δύο συγγραφείς (2 και 3). Ή μεταφράζοντας και ανατρέχοντας στους πίνακες *Article* και *Author* το άρθρο 'On the semantics of the relational data model' έχει γραφτεί από τους *Hans Albrecht Schmid* και *J Richard Swenson*.
- Η δεύτερη και τέταρτη εγγραφή έχουν τον ίδιο κωδικό συγγραφέα (2) υποδηλώνοντας ότι ο ίδιος συγγραφέας έχει γράψει και τα δύο άρθρα (2 και 3) αν και όχι απαραίτητα μόνος του (όπως είδαμε παραπάνω για το άρθρο 2).

## Reference

refID	refSource	refNotes
1	LEVEIN, R. E., AND MARON, M. E. A computer system for inference execution and data retrieval. Comm. ACM 10	NULL
2	CHURCH, A. An Introduction to Mathematical Logic I. Princeton U. Press, Princeton, N.J., 1956	NULL

Στο παραπάνω απόσπασμα από τον πίνακα *Reference* περιέχονται οι αναφορές που εμφανίζονται στα διάφορα άρθρα. Όπως και με τους συγγραφείς όμως, πουθενά δεν καθορίζεται ποια αναφορά εμφανίζεται σε ποιο άρθρο. Η συσχέτιση αυτή (ανάμεσα σε άρθρα και αναφορές) γίνεται στον πίνακα *ArticleReference* που παρουσιάζεται παρακάτω.

## ArticleReference

artID	refID
1	1
1	2

Η σύνδεση των άρθρων με τους συγγραφείς γίνεται με βάση τους κωδικούς τους (*artID* και *refID* αντίστοιχα). Στο παραπάνω απόσπασμα οι εγγραφές



δείχνουν ότι και οι δύο αναφορές (1 και 2) εμφανίζονται στο άρθρο 1. Αντίστοιχα (αν και δεν παρουσιάζεται εδώ) θα μπορούσαμε να έχουμε αναφορές που να εμφανίζονταν σε περισσότερα από ένα άρθρα.

### *Publisher*

	pubID	pubName
▶	1	Association for Computing Machinery
	2	IEEE Computer Society
	3	International Data Group (IDG)

Στο παραπάνω απόσπασμα από τον πίνακα *Publisher* εμφανίζονται στοιχεία για 3 εκδοτικούς οργανισμούς.

### *Magazine*

	magID	magTitle	pubID
▶	1	Communications of the ACM	1
	2	SIGMOD '75 Proceedings	1
	3	ICWS '06 Proceedings of the IEEE International Conference on Web Services	2
	4	PC World	3

Στο παραπάνω απόσπασμα από τον πίνακα *Magazine* εμφανίζονται στοιχεία για 4 περιοδικά. Στον πίνακα φαίνεται (πέρα από τον κωδικό και τον τίτλο του περιοδικού) και ποιος είναι ο εκδότης κάθε περιοδικού (πεδίο *pubID*).

- Για παράδειγμα, οι δύο πρώτες εγγραφές αναφέρουν ότι τα περιοδικά με κωδικούς 1 και 2 εκδίδονται από τον εκδότη με κωδικό 1. Μεταφράζοντας και ανατρέχοντας στον πίνακα *Publisher* τα περιοδικά ‘*Communications of the ACM*’ και ‘*SIGMOD '75 Proceedings*’ εκδίδονται από το ‘*Association for Computer Machinery*’.

Η συσχέτιση ανάμεσα στα άρθρα και τα περιοδικά (σε ποιο περιοδικό και πότε εμφανίζεται ένα άρθρο) γίνεται στον πίνακα *ArticleMagazine* που παρουσιάζεται παρακάτω.

### *ArticleMagazine*

	artID	magID	armDate	armPage
▶	1	1	1970-06-01	377
	2	2	1975-12-31	211
	3	3	2006-09-30	505
	4	4	2010-12-01	16
	5	4	2010-12-01	18

Η σύνδεση των άρθρων με τα περιοδικά γίνεται με βάση τους κωδικούς τους (*artID* και *magID* αντίστοιχα). Για παράδειγμα στο παραπάνω απόσπασμα από τον πίνακα *ArticleMagazine*:

- Η πρώτη εγγραφή δηλώνει ότι το άρθρο με κωδικό 1 εμφανίζεται στο περιοδικό με κωδικό 1. Μεταφράζοντας και ανατρέχοντας στους πίνακες *Article* και *Magazine* το άρθρο ‘*A Relational Model of Data for Large Shared Data Banks*’ (που αποτελεί στην ουσία τον ορισμό του Σχεσιακού Μοντέλου) εμφανίζεται στο περιοδικό ‘*Communications of the ACM*’. Επιπλέον στην εγγραφή αναφέρεται ότι το άρθρο ξεκινά στη σελίδα 377 καθώς και ότι δημοσιεύτηκε τη 1/6/1970 (πληροφορία που μπορεί να είναι χρήσιμη για την περίπτωση που θέλουμε να διαχωρίσουμε ανάμεσα στα τεύχη ενός περιοδικού).
- Η τέταρτη και πέμπτη εγγραφή έχουν τον ίδιο κωδικό περιοδικού (4) υποδηλώνοντας έτσι, ότι στο ίδιο περιοδικό περιλαμβάνονται και τα δύο άρθρα (4 και 5). Ή μεταφράζοντας και ανατρέχοντας στους πίνακες *Article* και *Magazine* τα άρθρα ‘*Unlocking Android's Multimedia Muscle*’ και ‘*Windows Phone 7: Microsoft Ups the Ante*’ εμφανίζονται στις σελίδες 16 και 18 αντίστοιχα του τεύχους Δεκεμβρίου 2010 του περιοδικού *PC World*©.

Αντίστοιχα (αν και δεν παρουσιάζεται εδώ) ένα άρθρο θα μπορούσε να εμφανίζεται σε πολλά περιοδικά (οπότε θα είχαμε δύο ή περισσότερες εγγραφές με τον ίδιο κωδικό άρθρου – *artID*).

# *Παράρτημα*

# Παράρτημα Α΄

## Οδηγός Εγκατάστασης MySQL

### Εγκατάσταση MySQL Community Server

Ο MySQL Server είναι ένας από τους πιο δημοφιλείς ανοικτού κώδικα (*open source*) servers για βάσεις δεδομένων. Η έκδοση MySQL Community Server είναι δωρεάν διαθέσιμη στην διεύθυνση <http://dev.mysql.com/downloads/mysql/>. Παρακάτω δίνουμε μερικές βασικές οδηγίες εγκατάστασης του MySQL Community Server.

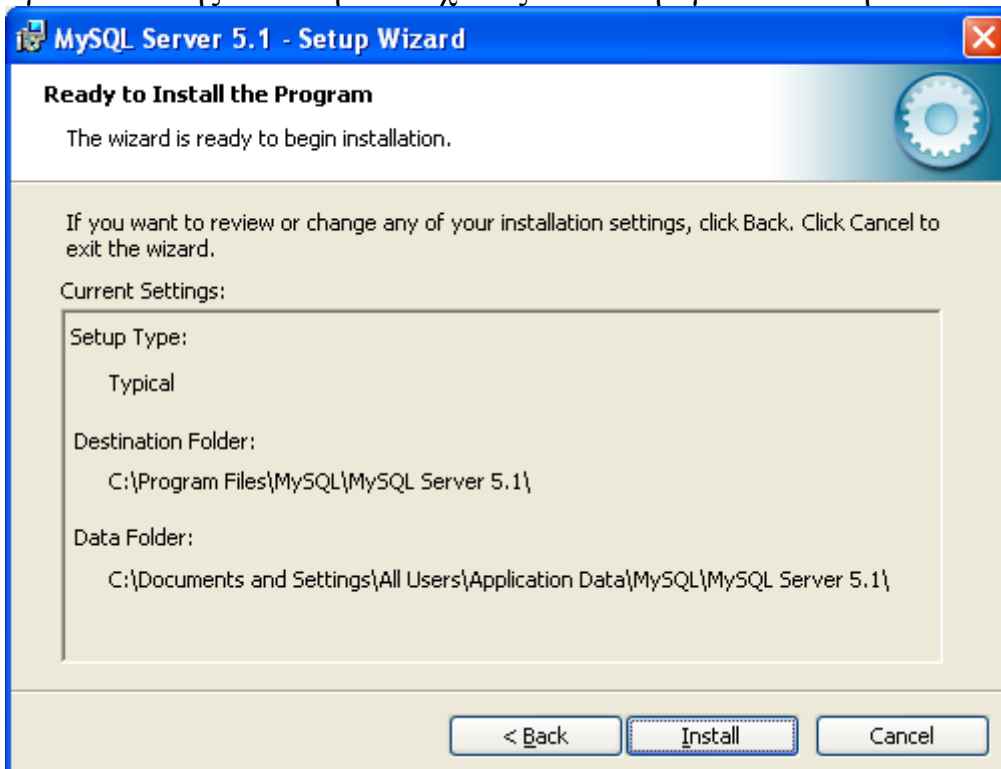
Εκτελώντας το πρόγραμμα εγκατάστασης εμφανίζεται η παρακάτω οθόνη καλωσορίσματος.



Στην δεύτερη οθόνη επιλέγουμε τύπο εγκατάστασης. (Εκτός από ειδικές περιπτώσεις η τυπική εγκατάσταση συνήθως καλύπτει τις απαιτήσεις των περισσότερων χρηστών.)



Στο επόμενο παράθυρο μπορούμε να επιθεωρήσουμε τις επιλογές εγκατάστασης και στην συνέχεια ξεκινάει η εγκατάσταση.





Παρακάτω φαίνεται η οθόνη ολοκλήρωσης της εγκατάστασης όπου έχουμε την δυνατότητα να ξεκινήσουμε και τον οδηγό ρυθμίσεων του MySQL Server.



**Ρύθμιση Παραμέτρων Λειτουργίας.** Ο οδηγός ρυθμίσεων (που συνήθως εκτελείται αμέσως μετά την ολοκλήρωση της εγκατάστασης) μας επιτρέπει να καθορίσουμε μερικές από τις βασικές παραμέτρους λειτουργίας του MySQL Community Server που αφορούν σε θέματα Απόδοσης και Ασφάλειας.

Επίσης, μας δίνει την δυνατότητα να καθορίσουμε περισσότερα από ένα instances του MySQL Server που θα τρέχουν στον υπολογιστή. Κάθε ένα από τα instances θα έχει τις δικές του παραμέτρους λειτουργίας και θα εκτελείται ανεξάρτητα από τα υπόλοιπα.

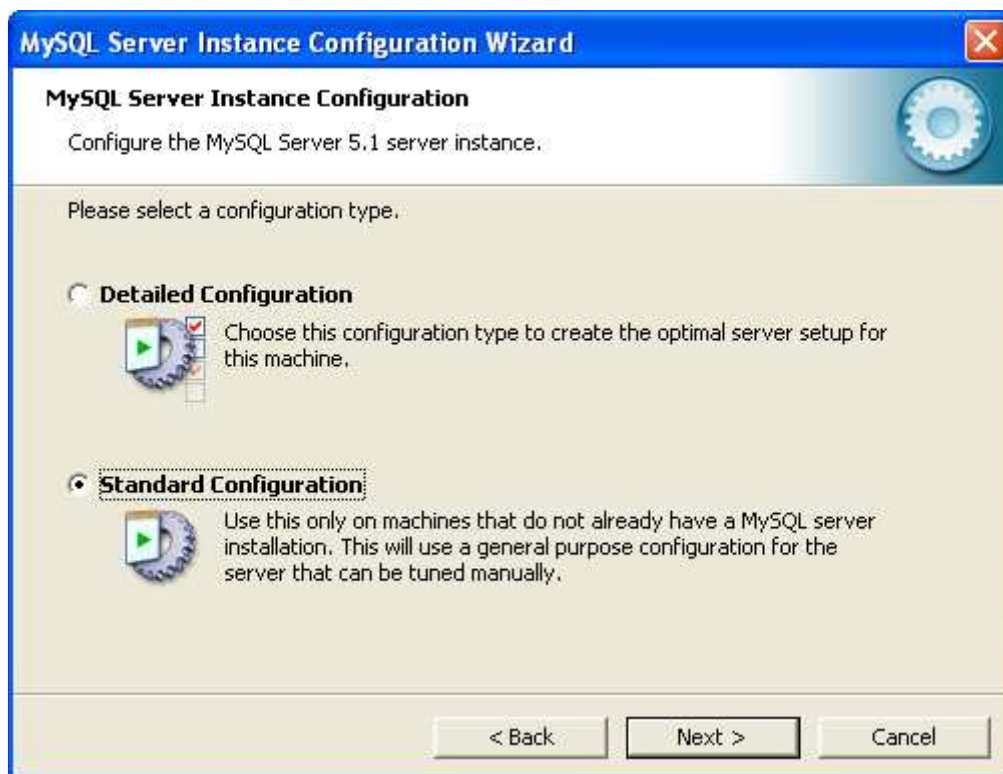
Εκτελώντας τον οδηγό ρυθμίσεων (είτε αυτόματα μετά την εγκατάσταση, είτε αυτόνομα σε κάποια αργότερη χρονική στιγμή) εμφανίζεται η παρακάτω οθόνη καλωσορίσματος.



Οι *Τυπικές Ρυθμίσεις (Standard Configuration)* προσφέρουν ένα προκαθορισμένο σύνολο παραμέτρων που είναι ικανοποιητικές για την πλειονότητα των χρηστών. Οι *Αναλυτικές Ρυθμίσεις (Detailed Configuration)* χρησιμοποιούνται κυρίως σε δύο περιπτώσεις:

- Όταν υπάρχουν ήδη ένα ή περισσότερα instances του MySQL Community Server που λειτουργούν στον υπολογιστή.
- Όταν έχουμε εξασφαλίσει την σωστή λειτουργία της Βάσης Δεδομένων και έχοντας ήδη παρατηρήσει τον τρόπο χρήσης της, θέλουμε να βελτιστοποιήσουμε την απόδοση για το συγκεκριμένο μοντέλο χρήσης.

Στην συγκεκριμένη περίπτωση και καθώς ο MySQL Community Server χρησιμοποιήθηκε για την *ανάπτυξη* μιας καινούριας ΒΔ προτιμήσαμε τις *Τυπικές Ρυθμίσεις*.





Στην επόμενη οθόνη (που φαίνεται παρακάτω) καθορίζουμε στοιχεία που αφορούν στον τρόπο λειτουργίας του συγκεκριμένου instance και στην ενσωμάτωσή του στο Λειτουργικό Σύστημα. Πιο συγκεκριμένα:

- Αν θέλουμε να εγκατασταθεί το instance σαν μια υπηρεσία (service) των Windows (προκαθορισμένη επιλογή).
  - Εγκαθιστώντας το instance σαν υπηρεσία μπορούμε να το διαχειριζόμαστε (ξεκινάμε, σταματάμε, κ.λπ.) μέσα από το κοινό περιβάλλον διαχείρισης υπηρεσιών των Windows.
  - Αν χρησιμοποιήσουμε αυτήν την δυνατότητα πρέπει επίσης να καθορίσουμε το όνομα της υπηρεσίας καθώς και το κατά πόσο αυτή θα ξεκινάει αυτόματα μαζί με τα Windows (προεπιλεγμένη ρύθμιση).
- Αν θέλουμε να συμπεριλάβουμε τον κατάλογο προγραμμάτων (bin directory) του MySQL Server στην μεταβλητή Διαδρομής (PATH) των Windows. Η συγκεκριμένη επιλογή είναι αρχικά αποεπιλεγμένη, αλλά συνίσταται ισχυρά να την ενεργοποιήσουμε, ειδικά εφ' όσον σκοπεύουμε να εργαστούμε με τα εργαλεία γραμμής εντολών(command line tools) του MySQL Server.



Η επόμενη οθόνη μας επιτρέπει να καθορίσουμε τις πληροφορίες ασφαλείας για το συγκεκριμένο instance. Συγκεκριμένα:

- Το συνθηματικό (password) του διαχειριστή του instance. (Στην ορολογία του MySQL Server ο διαχειριστής του συστήματος ονομάζεται root.)
  - Προφανώς συνίσταται να χρησιμοποιήσουμε ένα ισχυρό συνθηματικό (αφού ο διαχειριστής έχει απόλυτα δικαιώματα στο instance) καθώς και να φροντίσουμε να χρησιμοποιήσουμε ένα συνθηματικό που να μπορούμε να θυμόμαστε.
- Αν θέλουμε μπορούμε να δημιουργήσουμε ένα *ανώνυμο λογαριασμό*. Αυτό σημαίνει ότι οι χρήστες θα μπορούν να συνδέονται στο συγκεκριμένο instance χωρίς να χρειάζεται να δίνουν κάποια στοιχεία *αυθεντικοποίησης (authentication)*.
  - Γενικά δεν συνίσταται να δημιουργήσουμε ανώνυμο λογαριασμό γιατί αυτό μπορεί να οδηγήσει σε προβλήματα ασφαλείας. Ωστόσο μπορεί να είναι χρήσιμο για την περίπτωση που το instance θα χρησιμοποιηθεί σαν δημόσιος server και δεν θέλουμε να μπούμε στην διαδικασία του να μοιράζουμε ονόματα χρηστών και συνθηματικά.

MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.1 server instance.

Please set the security options.

**Modify Security Settings**

 New root password:  Enter the root password.

 Confirm:  Retype the password.

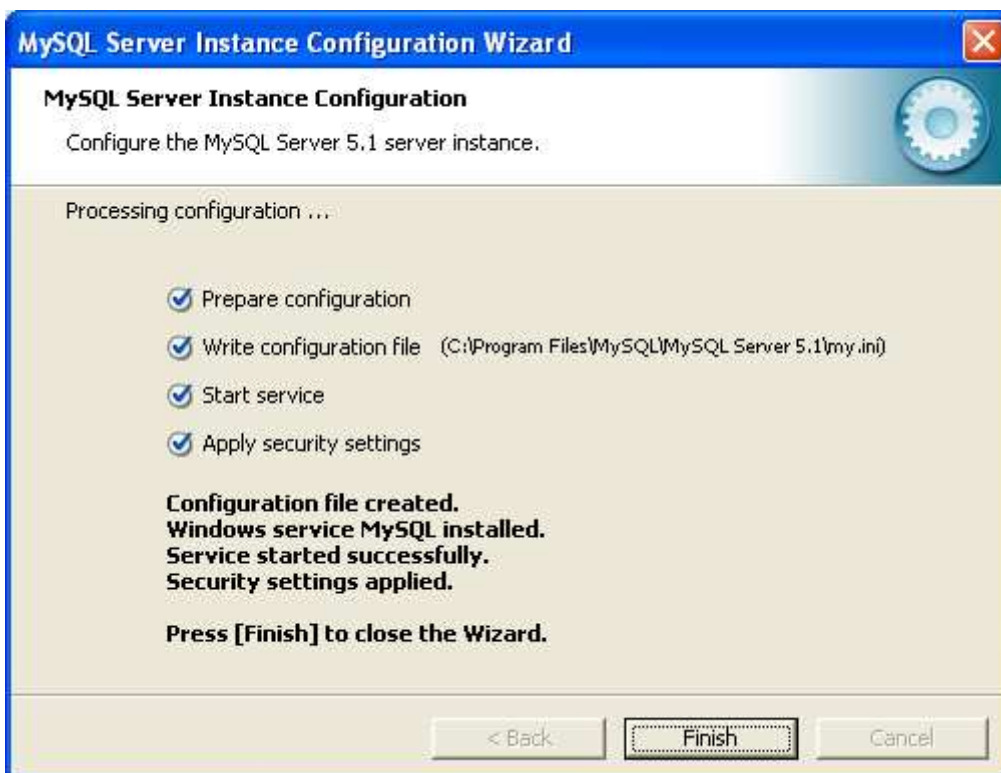
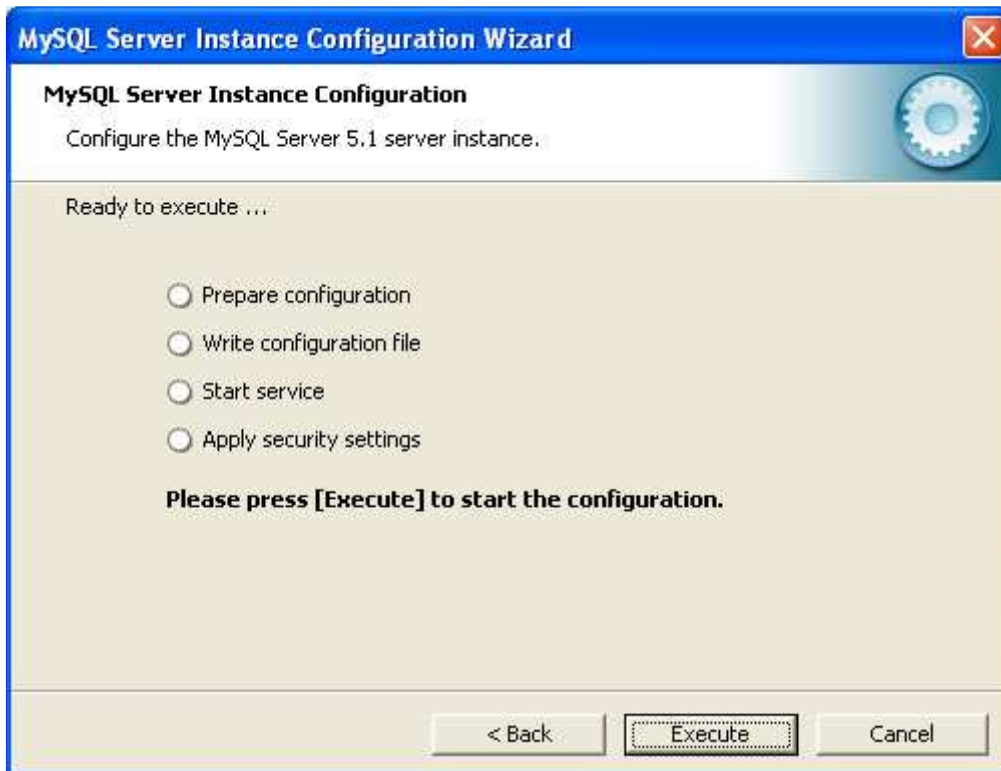
Enable root access from remote machines

Create An Anonymous Account

 This option will create an anonymous account on this server. Please note that this can lead to an insecure system.

< Back    Next >    Cancel

Η τελευταία οθόνη περιλαμβάνει μια σύνοψη των ενεργειών-ρυθμίσεων που θα πραγματοποιηθούν. Πατώντας το αντίστοιχο πλήκτρο ο οδηγός εκτελεί τις ενέργειες και στην συνέχεια εμφανίζει πληροφορίες σχετικά με την επιτυχία ή όχι εκτέλεσή τους.



## Εγκατάσταση Γραφικών Εργαλείων

Με την ολοκλήρωση της εγκατάστασης και της παραμετροποίησης του MySQL Community Server έχουμε ένα πλήρως λειτουργικό Σύστημα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) το οποίο μπορούμε να αρχίσουμε να χρησιμοποιούμε άμεσα. Ωστόσο, η χρήση του συστήματος μέσω των εργαλείων γραμμής εντολών (command line tools) δεν είναι πάντα εύκολη και οπωσδήποτε απαιτεί ένα ικανό επίπεδο γνώσεων. Γι αυτό, συνίσταται η εγκατάσταση και χρήση κάποιου πακέτου λογισμικού που να προσφέρει διαχείριση ΒΔ σε γραφικό περιβάλλον.

Ανάμεσα στα διαθέσιμα εργαλεία περιλαμβάνονται:

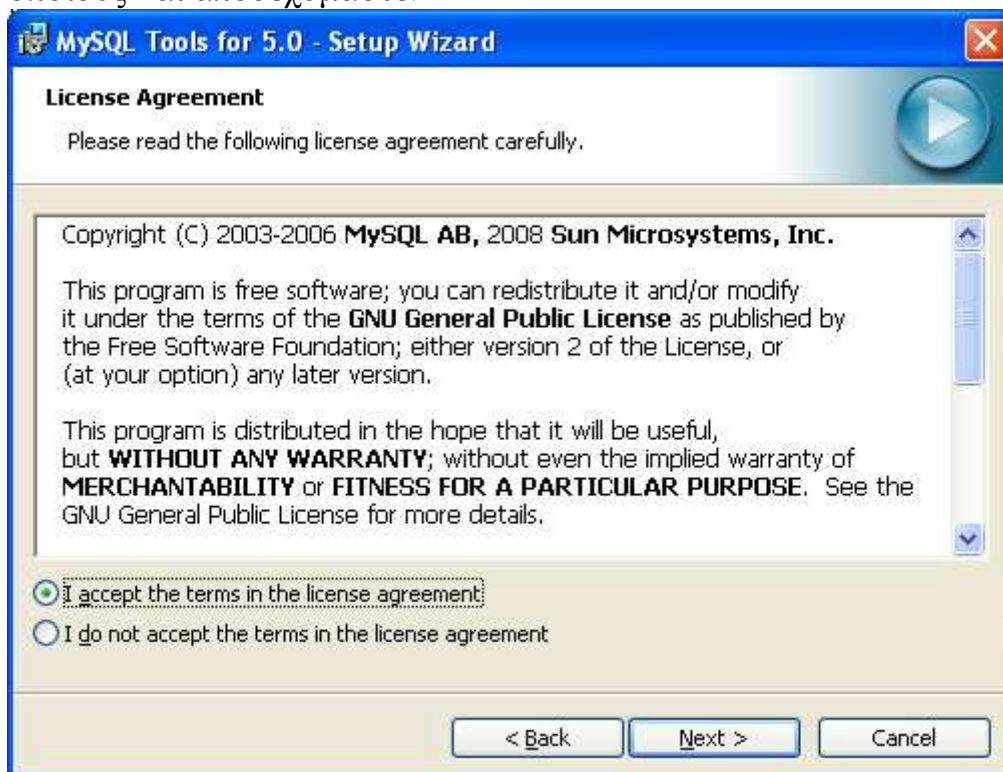
- MySQL Workbench: ενοποιημένο εργαλείο που επιτρέπει την μοντελοποίηση βάσεων δεδομένων, την σύνδεση με ΣΔΒΔ και την διαχείρισή τους. Είναι δωρεάν διαθέσιμο στην διεύθυνση <http://dev.mysql.com/downloads/workbench/5.2.html> για διάφορα λειτουργικά συστήματα.
- MySQL GUI Tools Bundle: πρόκειται για μια συλλογή από εργαλεία (MySQL Administrator, MySQL Query Browser και Migration Toolkit) που το καθένα χωριστά μπορεί να χρησιμοποιηθεί για την σύνδεση, διαχείριση και χρήση ενός MySQL Server. Είναι δωρεάν διαθέσιμα στην διεύθυνση <http://dev.mysql.com/downloads/gui-tools/5.0.html> για διάφορα λειτουργικά συστήματα.

Το πακέτο που κατά βάση χρησιμοποιήθηκε στην ανάπτυξη της ΒΔ είναι το MySQL GUI Tools Bundle. Στην συνέχεια δίνουμε οδηγίες εγκατάστασής του.

Εκτελώντας το πρόγραμμα εγκατάστασης εμφανίζεται η παρακάτω οθόνη καλωσορίσματος.



Η δεύτερη οθόνη περιλαμβάνει τους όρους χρήσης των εργαλείων τους οποίους και αποδεχόμαστε.

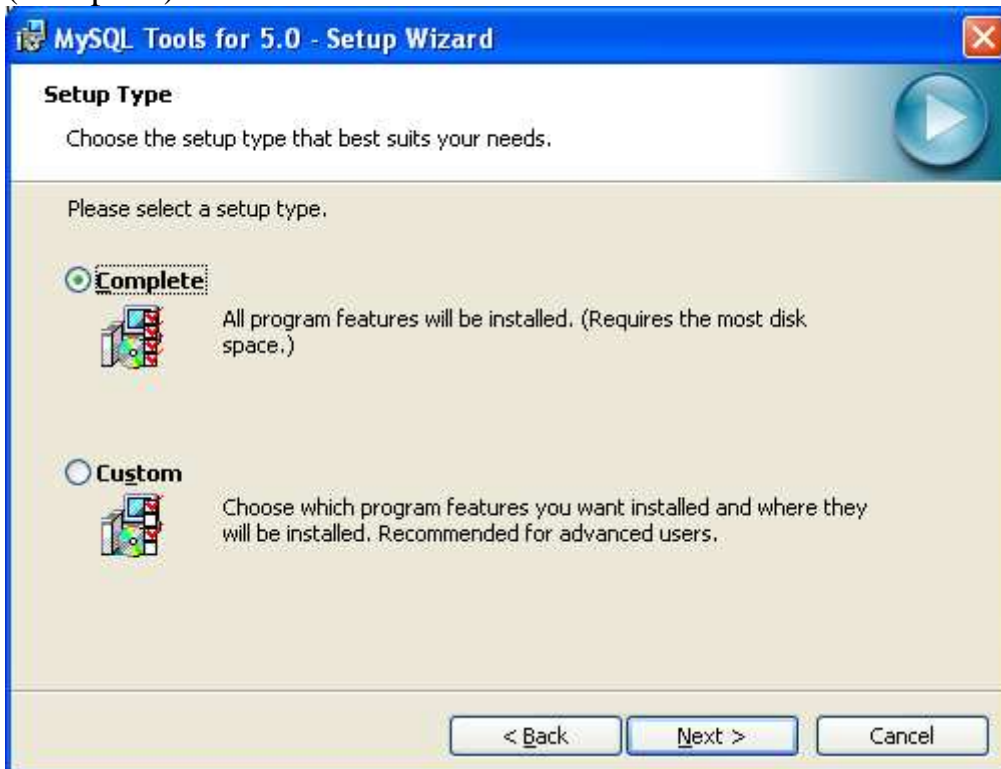




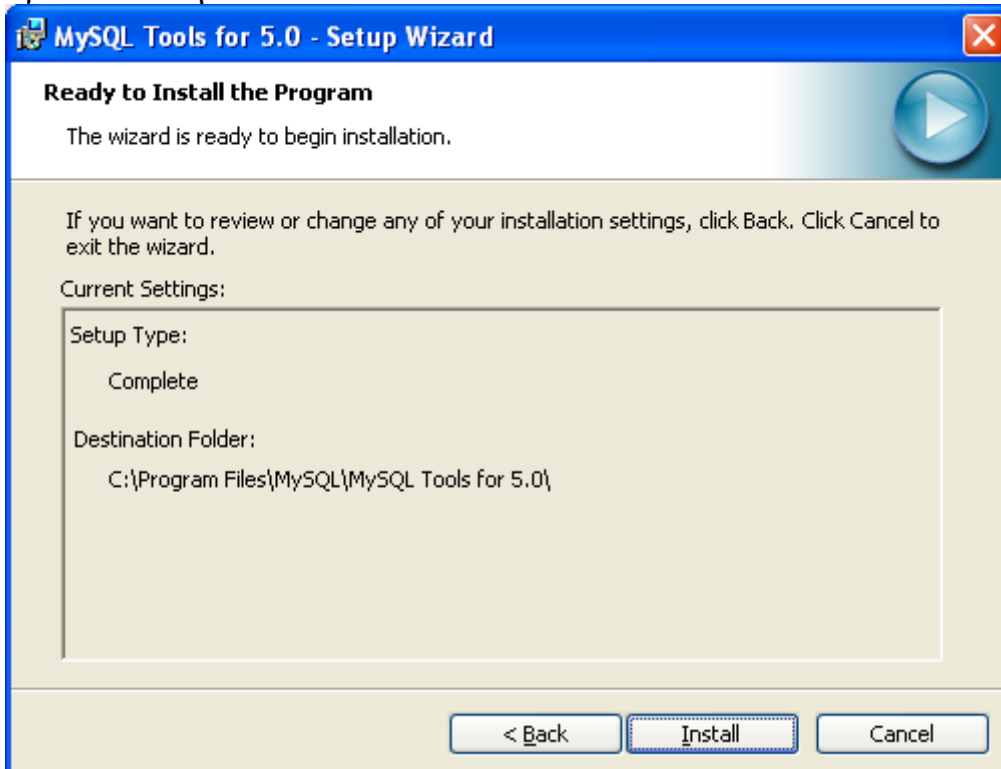
Στην τρίτη οθόνη μπορούμε να επιλέξουμε τον κατάλογο στον οποίο θα εγκατασταθούν τα εργαλεία.



Στην τέταρτη οθόνη μπορούμε να επιλέξουμε τον τύπο εγκατάστασης. Εκτός από ελάχιστες περιπτώσεις όπου πιθανώς να θέλουμε να περιορίσουμε τον απαιτούμενο χώρο συνίσταται η πλήρης εγκατάσταση (Complete).



Η επόμενη οθόνη συνοψίζει τις επιλογές εγκατάστασης. Εφ' όσον είμαστε ικανοποιημένοι πατάμε το πλήκτρο *Install* για να ξεκινήσει η εγκατάσταση.



Τέλος, παρακάτω φαίνεται η οθόνη ολοκλήρωσης της εγκατάστασης που μας πληροφορεί για την επιτυχή (ή όχι) ολοκλήρωσή της.





# Παράρτημα Β΄

## Κώδικας Δημιουργίας Βάσης Δεδομένων

Παρακάτω δίνουμε τον απαραίτητο κώδικα (SQL εντολές) για την δημιουργία της ΒΔ μαζί με τα δεδομένα του παραδείγματος που παρουσιάστηκε παραπάνω. (Ο ίδιος κώδικας είναι διαθέσιμος και στο αρχείο *ArticleDB.sql*, ενώ το αρχείο *ArticleDB.NoData.sql* περιέχει τον κώδικα δημιουργίας της ΒΔ χωρίς καθόλου δεδομένα.)

```
CREATE DATABASE IF NOT EXISTS `ArticleDB` /*!40100 DEFAULT CHARACTER SET utf8 */;
USE `ArticleDB`;
-- MySQL dump 10.13 Distrib 5.1.40, for Win32 (ia32)
--
-- Host: 192.168.117.136 Database: ArticleDB
-----
-- Server version 5.0.51a-3ubuntu5.7

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Not dumping tablespaces as no INFORMATION_SCHEMA.FILES table on this server
--

--
-- Table structure for table `Publisher`
--

DROP TABLE IF EXISTS `Publisher`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Publisher` (
```

```

`pubID` int(11) NOT NULL,
`pubName` varchar(100) NOT NULL,
PRIMARY KEY (`pubID`),
KEY `IX_PublisherName` (`pubName`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `Publisher`
--

LOCK TABLES `Publisher` WRITE;
/*!40000 ALTER TABLE `Publisher` DISABLE KEYS */;
INSERT INTO `Publisher` VALUES (1,'Association for Computing Machinery'),(2,'IEEE Computer Society'),(3,'International Data Group (IDG)');
/*!40000 ALTER TABLE `Publisher` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `ArticleReference`
--

DROP TABLE IF EXISTS `ArticleReference`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `ArticleReference` (
  `artID` int(11) NOT NULL,
  `refID` int(11) NOT NULL,
  PRIMARY KEY (`artID`,`refID`),
  KEY `FK_ArticleReference_Article` (`artID`),
  KEY `FK_ArticleReference_Reference` (`refID`),
  CONSTRAINT `FK_ArticleReference_Article` FOREIGN KEY (`artID`) REFERENCES `Article`
(`artID`) ON UPDATE CASCADE,
  CONSTRAINT `FK_ArticleReference_Reference` FOREIGN KEY (`refID`) REFERENCES
`Reference` (`refID`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `ArticleReference`
--

LOCK TABLES `ArticleReference` WRITE;
/*!40000 ALTER TABLE `ArticleReference` DISABLE KEYS */;
INSERT INTO `ArticleReference` VALUES (1,1),(1,2);
/*!40000 ALTER TABLE `ArticleReference` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `Reference`
--

DROP TABLE IF EXISTS `Reference`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Reference` (
  `refID` int(11) NOT NULL,
  `refSource` varchar(400) NOT NULL,
  `refNotes` varchar(400) default NULL,
  PRIMARY KEY (`refID`),

```

```

KEY `IX_Reference_Source` (`refSource`(255))
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `Reference`
--

LOCK TABLES `Reference` WRITE;
/*!40000 ALTER TABLE `Reference` DISABLE KEYS */;
INSERT INTO `Reference` VALUES (1,'LEVEIN, R. E., AND MARON, M. E. A computer system for
inference execution and data retrieval. Comm. ACM 10',NULL),(2,'CHURCH, A. An Introduction to
Mathematical Logic I. Princeton U. Press, Princeton, N.J., 1956',NULL);
/*!40000 ALTER TABLE `Reference` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `Author`
--

DROP TABLE IF EXISTS `Author`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Author` (
  `auID` int(11) NOT NULL,
  `auName` varchar(50) NOT NULL,
  `auSurname` varchar(50) NOT NULL,
  `auBirth` date default NULL,
  PRIMARY KEY (`auID`),
  KEY `IX_AuthorName` (`auName`),
  KEY `IX_AuthorSurname` (`auSurname`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `Author`
--

LOCK TABLES `Author` WRITE;
/*!40000 ALTER TABLE `Author` DISABLE KEYS */;
INSERT INTO `Author` VALUES (1,'Edgar','Codd','1923-08-23'),(2,'Hans
Albrecht','Schmid',NULL),(3,'J
Richard','Swenson',NULL),(4,'Robert','Anthony',NULL),(5,'JR','Raphael',NULL);
/*!40000 ALTER TABLE `Author` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `ArticleAuthor`
--

DROP TABLE IF EXISTS `ArticleAuthor`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `ArticleAuthor` (
  `artID` int(11) NOT NULL,
  `auID` int(11) NOT NULL,
  `araNotes` varchar(200) default NULL,
  PRIMARY KEY (`artID`,`auID`),
  KEY `FK_ArticleAuthor_Article` (`artID`),
  KEY `FK_ArticleAuthor_Author` (`auID`),

```

```

    CONSTRAINT `FK_ArticleAuthor_Article` FOREIGN KEY (`artID`) REFERENCES `Article`
(`artID`) ON UPDATE CASCADE,
    CONSTRAINT `FK_ArticleAuthor_Author` FOREIGN KEY (`auID`) REFERENCES `Author`
(`auID`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `ArticleAuthor`
--

LOCK TABLES `ArticleAuthor` WRITE;
/*!40000 ALTER TABLE `ArticleAuthor` DISABLE KEYS */;
INSERT INTO `ArticleAuthor` VALUES
(1,1,NULL),(2,2,NULL),(2,3,NULL),(3,2,NULL),(4,5,NULL),(5,4,NULL);
/*!40000 ALTER TABLE `ArticleAuthor` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `Magazine`
--

DROP TABLE IF EXISTS `Magazine`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Magazine` (
  `magID` int(11) NOT NULL,
  `magTitle` varchar(100) NOT NULL,
  `pubID` int(11) default NULL,
  PRIMARY KEY (`magID`),
  KEY `IX_MagazineTitle` (`magTitle`),
  KEY `FK_Magazine_Publisher` (`pubID`),
  CONSTRAINT `FK_Magazine_Publisher` FOREIGN KEY (`pubID`) REFERENCES `Publisher`
(`pubID`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `Magazine`
--

LOCK TABLES `Magazine` WRITE;
/*!40000 ALTER TABLE `Magazine` DISABLE KEYS */;
INSERT INTO `Magazine` VALUES (1,'Communications of the ACM',1),(2,'SIGMOD \'75
Proceedings',1),(3,'ICWS \'06 Proceedings of the IEEE International Conference on Web Services
',2),(4,'PC World',3);
/*!40000 ALTER TABLE `Magazine` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `Article`
--

DROP TABLE IF EXISTS `Article`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Article` (
  `artID` int(11) NOT NULL,
  `artTitle` varchar(100) NOT NULL,
  `artDescription` varchar(400) default NULL,

```

```

`artLink` varchar(200) default NULL,
PRIMARY KEY (`artID`),
KEY `IX_ArticleTitle` (`artTitle`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `Article`
--

LOCK TABLES `Article` WRITE;
/*!40000 ALTER TABLE `Article` DISABLE KEYS */;
INSERT INTO `Article` VALUES (1,'A Relational Model of Data for Large Shared Data
Banks',Περιγράφει το σχεσιακό
μοντέλο.',http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.98.5286&rep=rep1&type=pdf),(2,'O
n the semantics of the relational data
model',NULL,http://portal.acm.org/ft_gateway.cfm?id=500110&type=pdf&CFID=114208394&CFTOK
EN=66693971),(3,'Service Congestion: The Problem, and an Optimized Service Composition
Architecture as a Solution',NULL,NULL),(4,'Unlocking Android\'s Multimedia Muscle',Επιλογές για την
διαχείριση πολυμεσικού περιεχομένου στην πλατφορμα της Google.',NULL),(5,'Windows Phone 7:
Microsoft Ups the Ante',Παρουσίαση των Windows Phone 7.',NULL);
/*!40000 ALTER TABLE `Article` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `ArticleMagazine`
--

DROP TABLE IF EXISTS `ArticleMagazine`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `ArticleMagazine` (
  `artID` int(11) NOT NULL,
  `magID` int(11) NOT NULL,
  `armDate` date default NULL,
  `armPage` int(11) default NULL,
  PRIMARY KEY (`artID`,`magID`),
  KEY `FK_ArticleMagazine_Article` (`artID`),
  KEY `FK_ArticleMagazine_Magazine` (`magID`),
  CONSTRAINT `FK_ArticleMagazine_Article` FOREIGN KEY (`artID`) REFERENCES `Article`
(`artID`) ON UPDATE CASCADE,
  CONSTRAINT `FK_ArticleMagazine_Magazine` FOREIGN KEY (`magID`) REFERENCES
`Magazine` (`magID`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `ArticleMagazine`
--

LOCK TABLES `ArticleMagazine` WRITE;
/*!40000 ALTER TABLE `ArticleMagazine` DISABLE KEYS */;
INSERT INTO `ArticleMagazine` VALUES (1,1,'1970-06-01',377),(2,2,'1975-12-31',211),(3,3,'2006-09-
30',505),(4,4,'2010-12-01',16),(5,4,'2010-12-01',18);
/*!40000 ALTER TABLE `ArticleMagazine` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;

```

```
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;  
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;  
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

# ΒΙΒΛΙΟΓΡΑΦΙΑ

- Κέντρο ΠΛΗΝΕΤ Ν. Φλώρινας
- Περιοδικό PC World (τεύχος ΔΕΚΕΜΒΡΙΟΣ 2010)

## INTERNET:

<http://dev.mysql.com/downloads/mysql/>

<http://dev.mysql.com/downloads/workbench/5.2.html>

<http://dev.mysql.com/downloads/gui-tools/5.0.html>

# Ευχαριστήρια

Πρώτα από όλους θα ήθελα να ευχαριστήσω τον καθηγητή κ. Ιωάννη Μαυρικάκη για την προθυμία και την βοήθεια που προσέφερε όσες φορές την χρειάστηκα. Πολλά ευχαριστώ σε όλους τους καθηγητές για τις γνώσεις που μου προσέφεραν σε όλα τα φοιτητικά μου χρόνια.

Ξεχωριστά ευχαριστώ στους γονείς και την αδερφή μου για την ηθική και όχι μόνο στήριξη τους όλα αυτά τα χρόνια αλλά και για την απεριόριστη υπομονή τους. Χωρίς αυτούς δεν θα είχα καταφέρει τίποτα!!!



# Παράρτημα

Η παρουσίαση σε PowerPoint.