Εργαστήριο Βάσεων Δεδομένων

Εισαγωγή στη MySQL (1)

SQL

- Η SQL (Structured Query Language) είναι μια πλήρης γλώσσα Βάσεων Δεδομενων
- Είναι δομημένη σε βάσεις που περιέχουν πίνακες
- Οι πίνακες αποτελούνται από γραμμές (πλειάδες) και στήλες (γνωρίσματα)

Διαχείριση Βάσεων

Δημιουργία βάσης:

CREATE DATABASE [IF NOT EXISTS] dbname;

Επιλογή Βάσης για χρήση:

USE dbname;

Διαγραφή βάσης:

DROP DATABASE [IF EXISTS] dbname;

Περιγραφή βάσης:

SHOW TABLES;

Παραδείγματα

- mysql>CREATE DATABASE IF NOT EXISTS dblab;
  Query OK, 1 row affected (0.00 sec)
- mysql>USE dblab;
  Database changed
- mysql>SHOW TABLES;
  Empty set(0.00 sec)
- mysql>DROP DATABASE IF EXISTS dblab;
  Query OK, 1 row affected (0.01 sec)

Πίνακες

- Κάθε πίνακας αποτελείται από **πεδία** τα οποία αντιπροσωπεύουν τα γνωρίσματα
- Για κάθε πεδίο πρέπει να οριστεί ο **τύπος δεδομένων** του
- Θα πρέπει πάντα να χρησιμοποιείται ο τύπος δεδομένων που χρειάζεται για να αποθηκευτούν οι **τιμές** που μπορεί να πάρει το γνώρισμα
- Επίσης, πρέπει να ορίζεται και το **μέγεθος του πεδίου**, σύμφωνα με τις ανάγκες

Αριθμητικοί

- **INT**: ακέραιος μέγιστου μήκους 11 ψηφίων
- **TINYINT**: ακέραιος μέγιστου μήκους 4 ψηφίων
- **SMALLINT**: ακέραιος μέγιστου μήκους 5 ψηφίων
- **MEDIUMINT**: ακέραιος μέγιστου μήκους 9 ψηφίων
- **BIGINT**: ακέραιος μέγιστου μήκους 20 ψηφίων
- **FLOAT(M,D)**: κινητής υποδιαστολής M ψηφίων και D δεκαδικών(μέχρι 24), είναι πάντα προσημασμένος
- **DOUBLE(M,D)**:διπλής ακρίβειας, M ψηφίων και D δεκαδικών(μέχρι 53), είναι πάντα προσημασμένος

Ημερομηνίες & Ώρες

- **DATE**: Τιμή ημερομηνίας με μορφή ΕΕΕΕ-ΜΜ-ΗΗ
- **DATETIME**: Ημερομηνία και ώρα με μορφή ΕΕΕΕ-ΜΜ-ΗΗ ΩΩ:ΛΛ:ΔΔ
- **TIME**: Ώρα με μορφή ΩΩ:ΛΛ:ΔΔ
- **YEAR(M)**: Αποθηκεύει έτος σε διψήφια ή τετραψήφια μορφή. Στη διψήφια μορφή περιλαμβάνονται τα έτη από 1970-2069. Στην τετραψηφια από 1901-2155
- **TIMESTAMP**: Χρονική ένδειξη με μορφές 14(ΕΕΕΕΜΜΗΗΩΩΛΛΔΔ) 12(ΕΕΜΜΗΗΩΩΛΛΔΔ) 8(ΕΕΕΕΜΜΗΗ) 6(ΕΕΜΜΗΗ)

Αλφαριθμητικά 1/2

- **CHAR(M)**: Αλφαριθμητικό σταθερού μεγέθους μεταξύ 1-255 χαρακτήρες. Αν η τιμή δεν φτάνει, το μέγεθος συμπληρώνεται με κενά ως το προσδιοριζόμενο μήκος
- **VARCHAR(M)**: Αλφαριθμητικό μεταβλητού μεγέθους, από 1-255 χαρακτήρες
- **TEXT**: Πεδίο κειμένου με μέγιστο μήκος 65535
- **TINYTEXT**: Πεδίο κειμένου με μέγιστο μήκος 255
- **MEDIUMTEXT**: Πεδίο κειμένου με μέγιστο μήκος 16777215
- **LONGTEXT**: Πεδίο κειμένου με μέγιστο μήκος 4294967295

Αλφαριθμητικά 2/2

- **ENUM**: Απαρίθμηση, δηλαδή μία λίστα. Όταν ορίζεται ένας τύπος ENUM δημιουργείται μία λίστα στοιχείων από τα οποία μπορεί να επιλεχθεί μία τιμή (ή μπορεί να είναι και NULL)
    * Π.χ. αν θέλουμε το πεδίο να περιέχει τιμές A,B ή C τότε ορίζουμε τον τύπο τους ως: ENUM('A','B','C')
- **SET**: Παρομοίως με τον ENUM αλλά η τιμή ενός γνωρίσματος μπορεί να είναι ένα σύνολο με περισσότερες από μία τιμές του πεδίου ορισμού
    * Π.χ. για SET('A','B','C') κάποιο πεδίο μπορεί να πάρει την τιμή A,B ή την τιμή A,B,C

Δημιουργία πίνακα 1/3

- Η δημιουργία ενός πίνακα γίνεται με την εντολή **CREATE TABLE**
- Η CREATE TABLE συντάσσεται ως εξής:

CREATE TABLE

CREATE TABLE [IF NOT EXISTS] table_name(
column_name column_type[options],
column_name2 column_type2[options2],etc..
PRIMARY KEY(column_key_name1,...)
UNIQUE(index_col_name,...)
[**CONSTRAINT** symbol]
FOREIGN KEY(index_col_name,...)
REFERENCES tbl_name [(index_col_name,...)]
[**ON DELETE** reference_option]
[**ON UPDATE** reference_option]

Δημιουργία πίνακα 2/3

- **options**=[NOT NULL|NULL][DEFAULT default_value][AUTO_INCREMENT]
  - NOT NULL: δεν επιτρέπεται η τιμή NULL
  - NULL: επιτρέπεται η τιμή NULL
  - DEFAULT: ορισμός προκαθορισμένης τιμής που τίθεται αυτόματα όταν δεν δίνεται ρητά τιμή στο γνώρισμα
  - AUTO_INCREMENT: αυτόματη αύξηση της τιμής κατά 1

Δημιουργία πίνακα 3/3

- **reference option**=RESTRICT|CASCADE|SET NULL|NO ACTION|SET DEFAULT
  - RESTRICT: απορρίπτει τη διαγραφή ή την ανανέωση από τον πίνακα πατέρα
  - CASCADE: διαγραφή ή ανανέωση στον πίνακα πατέρα, αυτόματα διαγράφει ή ανανεώνει και τις αντίστοιχες πλειάδες στον πίνακα παιδί
  - SET NULl: διαγραφή ή ανανέωση στον πίνακα πατέρα, θέτει το πεδίο του εξωτερικού κλειδιού του πίνακα παιδιού NULL
  - NO ACTION: διαγραφή ή ανανέωση της τιμής του πρωτεύοντος κλειδιού, απορρίπτεται εάν υπάρχει ξένο κλειδί που "δείχνει" στον πίνακά μας

Στήλες

- Έστω ότι ορίζουμε ένα γνώρισμα που περιέχει τον κωδικό ενός προϊόντος. Είναι ακέραιος αριθμός 9 ψηφίων. Θέλουμε να μην είναι ποτέ null και να αυξάνεται αυτόματα η τιμή του με την προσθήκη κάθε νέου προϊόντος

CREATE TABLE

product_id INT(9) NOT NULL AUTO_INCREMENT

- Έστω ότι θέλουμε κι άλλο ένα γνώρισμα το οποίο είναι ο αριθμός ταυτότητας του αγοραστή.Είναι αριθμητικό, μήκους 7 χαρακτήρων

buyer_id VARCHAR(7)

- Έστω ότι υπάρχει κι άλλο όρισμα, το όνομα προϊόντος, αλφαριθμητικό 20 χαρακτήρων που δεν είναι ποτέ NULL. Έχει προεπιλεγμένη τιμή "unknown"

product_name VARCHAR(20) DEFAULT 'unknown' NOT NULL

Keys

- Έστω ότι θέλουμε να ορίσουμε ότι το όρισμα product_id είναι πρωτεύον κλειδί στον πίνακα. Προσθέτουμε την γραμμή:

```
PRIMARY KEY(product_id)
```

- Έστω ότι ξέρουμε ότι το όνομα προϊόντος είναι μοναδικό και μπορεί να χρησιμοποιηθεί σαν εναλλακτικό κλειδί. Το δηλώνουμε ως εξής:

```
UNIQUE(product_name)
```

Foreign keys

- Έστω ότι το buyer_id είναι ένα ξένο κλειδί που "δείχνει" στο όρισμα ar_taut του πίνακα "Πελάτης". Βάζουμε ένα περιορισμό με όνομα PRDCTBR και δηλώνουμε τον περιορισμό αναφορικής ακεραιότητας:

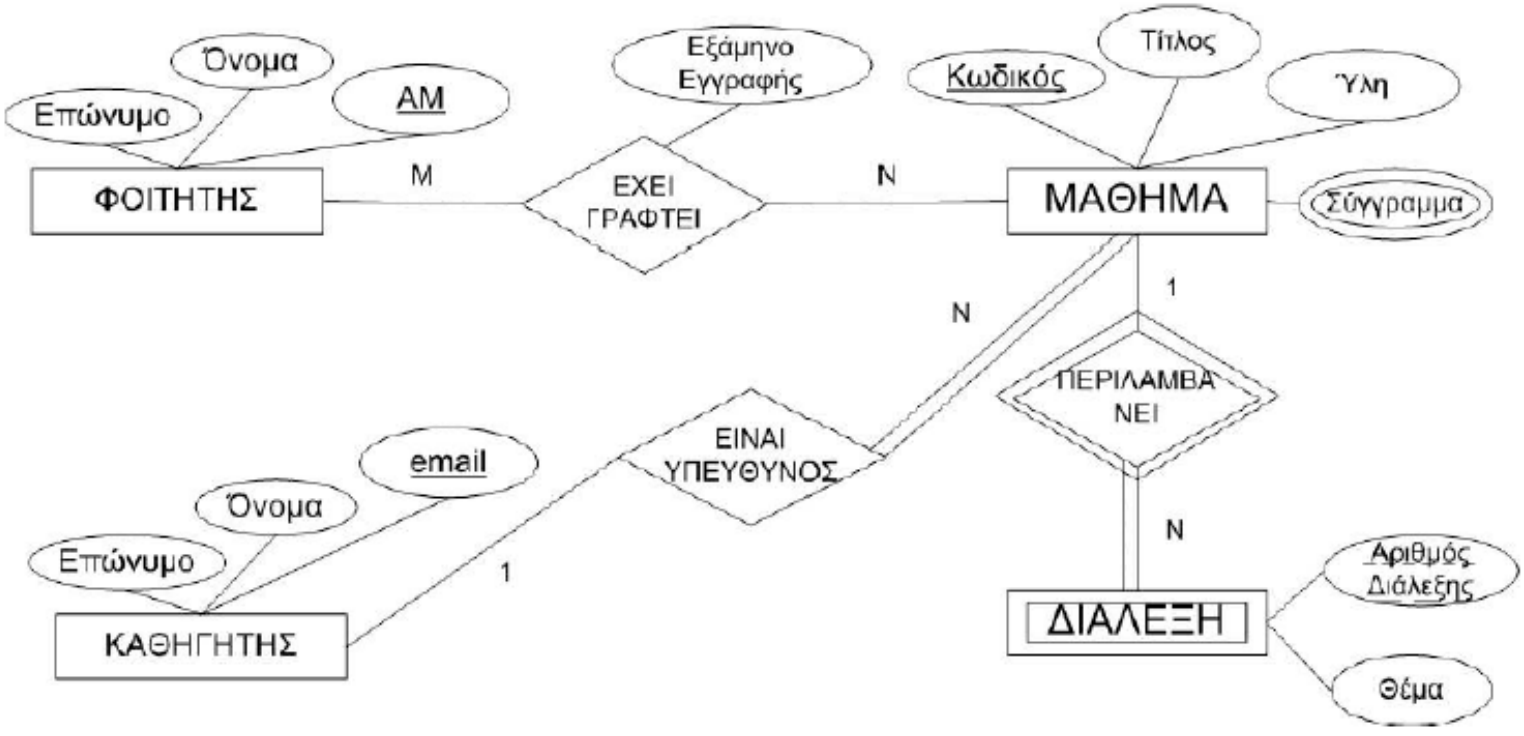
CONSTRAINT PRDCTBR
FOREIGN KEY (buyer_id) REFERENCES pelatis (ar_taut)
ON DELETE SET NULL ON UPDATE CASCADE

- Η πράξη ON DELETE SET NULL σημαίνει ότι στην περίπτωση που διαγραφεί η αντίστοιχη εγγραφή του πελάτη,το πεδίο τίθεται NULL. Η πράξη ON UPDATE CASCADE σημαίνει ότι αν αλλάξει ο αριθμός ταυτότητας του πελάτη,ενημερώνεται αυτόματα το πεδίο buyer_id

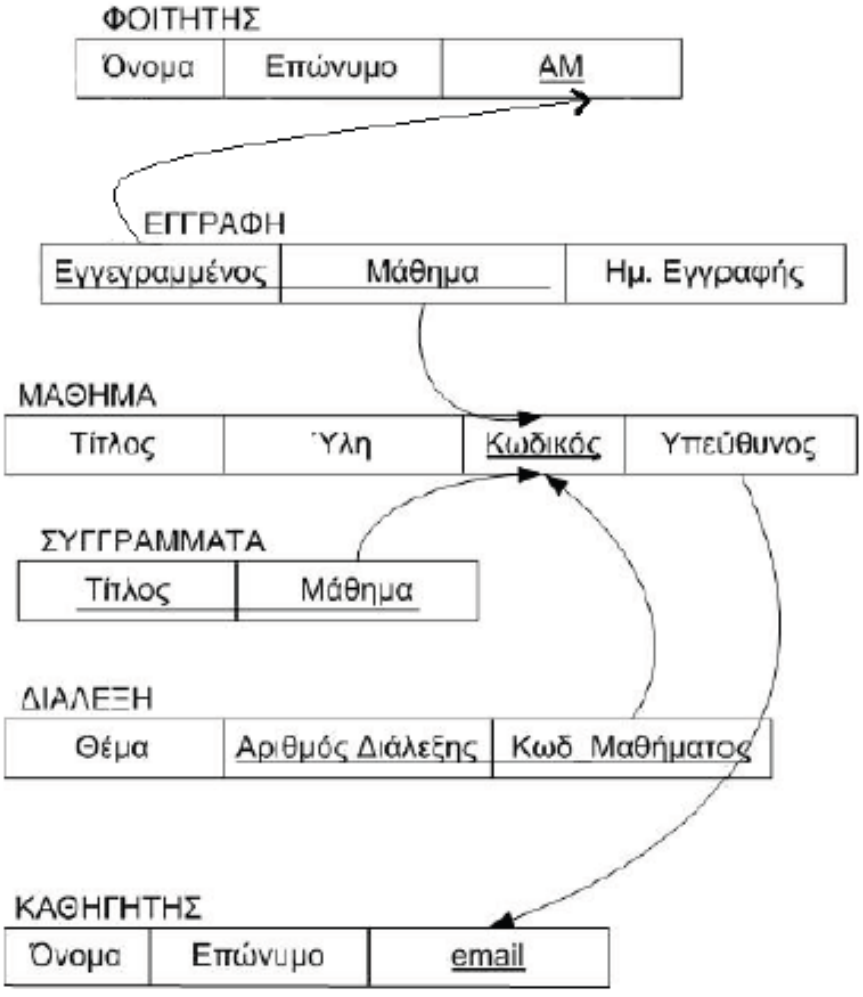
Create Table

```
mysql> CREATE TABLE product(
 -> product_id INT(9) NOT NULL AUTO_INCREMENT,
 -> buyer_id VARCHAR(7),
 -> product_name VARCHAR(20) DEFAULT 'unknown' NOT
NULL,
 -> PRIMARY KEY(product_id),
 -> UNIQUE name_index(product_name),
 -> CONSTRAINT PRDCTBR
 -> FOREIGN KEY(buyer_id) REFERENCES pelatis(Ar_Tayt)
 -> ON DELETE SET NULL ON UPDATE CASCADE
 -> );
Query OK, 0 rows affected (0.08 sec)
```

ER


Όνομα
Επώνυμο
ΑΜ
ΦΟΙΤΗΤΗΣ
Εξάμηνο Εγγραφής
ΕΧΕΙ ΓΡΑΦΤΕΙ
Κωδικός
Τίτλος
Ύλη
ΜΑΘΗΜΑ
Σύγγραμμα
ΠΕΡΙΛΑΜΒΑΝΕΙ
ΕΙΝΑΙ ΥΠΕΥΘΥΝΟΣ
email
Όνομα
Επώνυμο
ΚΑΘΗΓΗΤΗΣ
ΔΙΑΛΕΞΗ
Αριθμός Διάλεξης
Θέμα
M
N
N
1
1
N

Σχεσιακό


ΦΟΙΤΗΤΗΣ
Όνομα
Επώνυμο
ΑΜ
ΕΓΓΡΑΦΗ
Εγγεγραμμένος
Μάθημα
Ημ. Εγγραφής
ΜΑΘΗΜΑ
Τίτλος
Ύλη
Κωδικός
Υπεύθυνος
ΣΥΓΓΡΑΜΜΑΤΑ
Τίτλος
Μάθημα
ΔΙΑΛΕΞΗ
Θέμα
Αριθμός Διάλεξης
Κωδ Μαθήματος
ΚΑΘΗΓΗΤΗΣ
Όνομα
Επώνυμο
email

Create Tables 1/3

```
CREATE TABLE student(
name VARCHAR(25) DEFAULT 'unknown' NOT NULL,
lastname VARCHAR(25) DEFAULT 'unknown' NOT NULL,
AM INT(5) NOT NULL AUTO_INCREMENT,
PRIMARY KEY(AM)
);

CREATE TABLE professor(
pr_name VARCHAR(25) DEFAULT 'unknown' NOT NULL,
pr_lastname VARCHAR(25) DEFAULT 'unknown' NOT NULL,
email VARCHAR(255) NOT NULL,
PRIMARY KEY(email)
);
```

Create Tables 2/3

```
CREATE TABLE course(
title VARCHAR(255) DEFAULT 'unknown' DEFAULT,
material TEXT,
course_id INT(4) NOT NULL AUTO_INCREMENT,
supervisor VARCHAR(255) NOT NULL,
PRIMARY KEY(course_id),
UNIQUE(title),
CONSTRAINT SUPERVISED
FOREIGN KEY (supervisor) REFERENCES professor(email)
ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE books(
title VARCHAR(128) DEFAULT 'Title' NOT NULL,
course_book INT(4) NOT NULL,
PRIMARY KEY(title,course_book),
CONSTRAINT CRSBOOK
FOREIGN KEY (course_book) REFERENCES course(course_id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

Create Tables 3/3

```
CREATE TABLE lecture(
subject VARCHAR(128),
num_lecture INT(2) NOT NULL,
course_lecture INT(4) NOT NULL,
PRIMARY KEY(num_lecture,course_lecture),
CONSTRAINT CRSLECTURE
FOREIGN KEY (course_lecture) REFERENCES course(course_id)
ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE registration(
reg_date DATE NOT NULL,
reg_student INT(5) NOT NULL,
reg_courseINT(4) NOT NULL,
PRIMARY KEY(reg_student,reg_course),
CONSTRAINT CRSREGISTRATION
FOREIGN KEY (reg_course) REFERENCES course(course_id)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT STDNTREGISTRATION
FOREIGN KEY (reg_student) REFERENCES student(AM)
ON DELETE CASCADE ON UPDATE CASCADE

);
```