

Εργαστήριο Βάσεων Δεδομένων

SQL (μέρος 1)

Create, Insert, Delete, Update, Alter



SQL

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

Διαχείριση Βάσεων

- Δημιουργία βάσης:

```
CREATE DATABASE dbname;
```

- Επιλογή Βάσης για χρήση:

```
USE dbname;
```

- Διαγραφή βάσης:

```
DROP DATABASE dbname;
```

- Προβολή όλων των πινάκων στην τρέχουσα βάση:

```
SHOW TABLES;
```

- Προβολή των πεδίων ενός πίνακα:

```
DESCRIBE table_name;
```

- Προβολή της εντολής create table:

```
SHOW CREATE TABLE table_name;
```

- Προβολή του περιεχομένου των πεδίων του πίνακα

```
SELECT * FROM table_name;
```

Διαχείριση βάσεων - Παραδείγματα

```
mysql> CREATE DATABASE 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 dblab;
```

```
Query OK, 1 row affected (0.01 sec)
```

Διαχείριση βάσεων - Παραδείγματα

- Διαθέσιμες βάσεις:

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| maria     |
| maria_2   |
+-----+
3 rows in set (0.03 sec)
```

- Επιλογή Βάσης για χρήση:

```
mysql> use maria
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

Διαχείριση βάσεων - Παραδείγματα

- Προβολή όλων των πινάκων στην τρέχουσα βάση:
SHOW TABLES;

```
mysql> show tables;
+-----+
| Tables_in_maria |
+-----+
| APPLIES_FOR      |
| CANDIDATE        |
| Course           |
| JOB              |
| Professors       |
| Staff            |
| SubjectCoverAreas |
| SupportingStaff   |
| Teach            |
| books            |
| course           |
| lecture          |
| product          |
| professor        |
| registration     |
| student          |
+-----+
16 rows in set (0.02 sec)
```

Διαχείριση βάσεων - Παραδείγματα

- Προβολή των πεδίων ενός πίνακα:

DESCRIBE table_name;

```
mysql> describe course  
-> ;
```

Field	Type	Null	Key	Default	Extra
title	varchar(255)	NO	UNI	unknown	
material	text	YES		NULL	
course_id	int(4)	NO	PRI	NULL	auto_increment
supervisor	varchar(255)	NO	MUL	NULL	

Διαχείριση βάσεων - Παραδείγματα

- Προβολή της εντολής create table:

SHOW CREATE TABLE table name;

```
mysql> show create table course;
```

```
+-----+-----+
| Table | Create Table
+-----+-----+
| course | CREATE TABLE `course` (
  `title` varchar(255) NOT NULL DEFAULT 'unknown',
  `material` text,
  `course_id` int(4) NOT NULL AUTO_INCREMENT,
  `supervisor` varchar(255) NOT NULL,
  PRIMARY KEY (`course_id`),
  UNIQUE KEY `title` (`title`),
  KEY `SUPERVISED` (`supervisor`),
  CONSTRAINT `SUPERVISED` FOREIGN KEY (`supervisor`) REFERENCES `professor` (`email`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1 |
+-----+-----+
1 row in set (0.00 sec)
```

Διαχείριση βάσεων - Παραδείγματα

- Προβολή του περιεχομένου των πεδίων του πίνακα

SELECT * FROM table_name;

```
mysql> select * from course;
```

```
+-----+-----+-----+-----+
| title      | material                | course_id | supervisor                |
+-----+-----+-----+-----+
| Databases  | Intro to relational DBs | 2         | pap@ceid.upatras.gr      |
| Databases II | Advanced DBs           | 3         | alex@ceid.upatras.gr     |
+-----+-----+-----+-----+
2 rows in set (0.00 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.

Τύποι Δεδομένων-Αριθμητικοί

- **INT**: ακέραιος μέγιστου μήκους 11 ψηφίων
 - (4 bytes) -2.147.483.648 μέχρι 2.147.483.647 normal. 0 μέχρι 4.294.967.295 UNSIGNED
- **TINYINT**: ακέραιος μέγιστου μήκους 4 ψηφίων
 - (1 byte) -128 to 127 normal. 0 to 255 UNSIGNED
- **SMALLINT**: ακέραιος μέγιστου μήκους 5 ψηφίων
 - (2 bytes) -32768 to 32767 normal. 0 to 65535 UNSIGNED
- **MEDIUMINT**: (3 bytes) ακέραιος μέγιστου μήκους 9 ψηφίων
- **BIGINT**: (8 bytes) ακέραιος μέγιστου μήκους 20 ψηφίων
- **FLOAT(M,D)**: κινητής υποδιαστολής M ψηφίων και D δεκαδικών (μέχρι 23), είναι πάντα προσημασμένος
- **DOUBLE(M,D)**: διπλής ακρίβειας, M ψηφίων και D δεκαδικών (μέχρι 53), είναι πάντα προσημασμένος

Τύποι Δεδομένων – Ημερομηνίες & Ώρες

- **DATE:** Τιμή ημερομηνίας με μορφή
EEEE-MM-HH (1973-12-30, 2017-01-26)
- **TIME:** Ώρα με μορφή
ΩΩ:ΛΛ:ΔΔ (18:46:05)
- **DATETIME:** Ημερομηνία και ώρα με μορφή
EEEE-MM-HH ΩΩ:ΛΛ:ΔΔ (1992-12-30 15:30:00)
- **YEAR(M):** Αποθηκεύει έτος σε διψήφια ή τετραψήφια μορφή. Στη διψήφια μορφή περιλαμβάνονται τα έτη από 1970-2069. Στην τετραψήφια από 1901-2155.
- **TIMESTAMP:** Χρονική ένδειξη με μορφές
14(EEEEEMMHΩΩΛΛΔΔ) 12(EEMMHΩΩΛΛΔΔ)
8(EEEEEMMH) 6(EEMMH)
 - Π.χ. 19731230101550

Τύποι Δεδομένων - Αλφαριθμητικά

- **CHAR(M)**: Αλφαριθμητικό (letters, numbers, special characters) σταθερού μεγέθους μεταξύ 1-255 χαρακτήρες. *Αν η τιμή δεν φτάνει το μέγεθος M συμπληρώνεται αριστερά με κενά. Default M=1*
- **VARCHAR(M)**: Αλφαριθμητικό μεταβλητού μεγέθους, από 1-255 χαρακτήρες. *Πρέπει να καθορίζουμε το M.*
- **TEXT**: Πεδίο κειμένου με μέγιστο μήκος 65.535
- **TINYTEXT**: Πεδίο κειμένου με μέγιστο μήκος 255
- **MEDIUMTEXT**: Πεδίο κειμένου με μέγιστο μήκος 16.777.215
- **LONGTEXT**: Πεδίο κειμένου με μέγιστο μήκος 4.294.967.295

MySQL DATA TYPES

DATE TYPE	SPEC	DATA TYPE	SPEC
CHAR	String (0 - 255)	INT	Integer (-2147483648 to 2147483647)
VARCHAR	String (0 - 255)	BIGINT	Integer (-9223372036854775808 to 9223372036854775807)
TINYTEXT	String (0 - 255)	FLOAT	Decimal (precise to 23 digits)
TEXT	String (0 - 65535)	DOUBLE	Decimal (24 to 53 digits)
BLOB	String (0 - 65535)	DECIMAL	"DOUBLE" stored as string
MEDIUMTEXT	String (0 - 16777215)	DATE	YYYY-MM-DD
MEDIUMBLOB	String (0 - 16777215)	DATETIME	YYYY-MM-DD HH:MM:SS
LONGTEXT	String (0 - 4294967295)	TIMESTAMP	YYYYMMDDHHMMSS
LOBLOB	String (0 - 4294967295)	TIME	HH:MM:SS
TINYINT	Integer (-128 to 127)	ENUM	One of preset options
SMALLINT	Integer (-32768 to 32767)	SET	Selection of preset options
MEDIUMINT	Integer (-8388608 to 8388607)	BOOLEAN	TINYINT(1)

Τύποι Δεδομένων - Αλφαριθμητικά

- **ENUM:** Απαρίθμηση, δηλαδή μία λίστα. Όταν ορίζεται ένας τύπος ENUM δημιουργείται μία λίστα στοιχείων από τα οποία μπορεί να επιλεγθεί 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 του ως: ENUM('A', 'B', 'C')
- **SET:** Παρόμοιος 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 ή την τιμή B ή την τιμή A,B,C

Παράδειγμα: Δημιουργία πίνακα

- 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.
- Έστω ότι θέλουμε κι άλλο ένα γνώρισμα το οποίο είναι ο αριθμός ταυτότητας του αγοραστή. Είναι αλφαριθμητικό, μήκους 7 χαρακτήρων.
- Έστω ότι υπάρχει κι άλλο όρισμα, το όνομα προϊόντος, αλφαριθμητικό 20 χαρακτήρων που δεν είναι ποτέ NULL. Έχει προεπιλεγμένη τιμή “unknown”.

Παράδειγμα: Δημιουργία πίνακα-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.

product_id **INT**(9)

- Έστω ότι θέλουμε κι άλλο ένα γνώρισμα το οποίο είναι ο αριθμός ταυτότητας του αγοραστή. Είναι αλφαριθμητικό, 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)

Δημιουργία πίνακα

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

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

Δημιουργία πίνακα

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

```
CREATE TABLE [IF NOT EXISTS] table_name(  
column_name column_type [options],  
column_name2 column_type2 [options2], κλπ..
```

```
PRIMARY KEY (column_key_name1,...)
```

```
UNIQUE (index_col_name,...)
```

```
[CONSTRAINT constraint_code
```

```
FOREIGN KEY (index_col_name,...)
```

```
REFERENCES tbl_name [(index_col_name,...)]
```

```
[ON DELETE reference_option]
```

```
[ON UPDATE reference_option]
```

```
);
```

Δημιουργία πίνακα

```
CREATE TABLE [IF NOT EXISTS] table_name(  
column_name column_type [options],  
column_name2 column_type2 [options2], κλπ..  
.....
```

- **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. Ξεκινάει από το 1 εκτός κι αν ορίσουμε άλλη αρχική τιμή (π.χ. AUTO_INCREMENT = 30)

Δημιουργία πίνακα

CREATE TABLE

REFERENCES tbl_name [(index_col_name,...)]

[**ON DELETE** reference_option]

[**ON UPDATE** reference_option]

- **reference option** = RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT
 - **RESTRICT**: απορρίπτει τη διαγραφή ή την ανανέωση από τον πίνακα πατέρα (ο πίνακας που περιέχει το πρωτεύον κλειδί)
 - **CASCADE**: αν γίνει διαγραφή ή ανανέωση στον πίνακα πατέρα, αυτόματα διαγράφει ή ανανεώνει και τις αντίστοιχες πλειάδες στον πίνακα παιδί (ο πίνακας που περιέχει το ξένο κλειδί)
 - **SET NULL**: διαγραφή ή ανανέωση στον πίνακα πατέρα, θέτει το πεδίο του εξωτερικού κλειδιού του πίνακα παιδιού σε NULL
 - Εκτός αν το ξένο κλειδί έχει οριστεί σαν NOT NULL στον πίνακα παιδί
 - **NO ACTION**: διαγραφή ή ανανέωση της τιμής του πρωτεύοντος κλειδιού απορρίπτεται, εάν υπάρχει ξένο κλειδί που «δείχνει» στον πίνακα μας (ταυτίζεται με το RESTRICT στη MySQL)
 - Είναι η DEFAULT επιλογή αν δεν ορίσουμε κάτι στο ON DELETE/UPDATE

Παράδειγμα: Δημιουργία πίνακα-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. Η αντίστοιχη γραμμή στην δήλωση της CREATE TABLE θα είναι:

`product_id INT(9) NOT NULL AUTO_INCREMENT`

- Έστω ότι θέλουμε κι άλλο ένα γνώρισμα το οποίο είναι ο αριθμός ταυτότητας του αγοραστή. Είναι αλφαριθμητικό, 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`

Παράδειγμα: Δημιουργία πίνακα-κλειδιά

- 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`)

Παράδειγμα: Δημιουργία πίνακα-Foreign keys

- Έστω ότι υπάρχει ένας πίνακας Πελάτης (pelatis) με κλειδί ar_tayt. Έστω ότι το buyer_id είναι ένα ξένο κλειδί που «δείχνει» στο όρισμα ar_tayt του πίνακα Πελάτης (pelatis). Βάζουμε ένα περιορισμό με όνομα PRDCTBR και δηλώνουμε τον περιορισμό αναφορικής ακεραιότητας:

CONSTRAINT PRDCTBR

FOREIGN KEY(buyer_id) REFERENCES pelatis(Ar_Tayt)

ON DELETE SET NULL ON UPDATE CASCADE

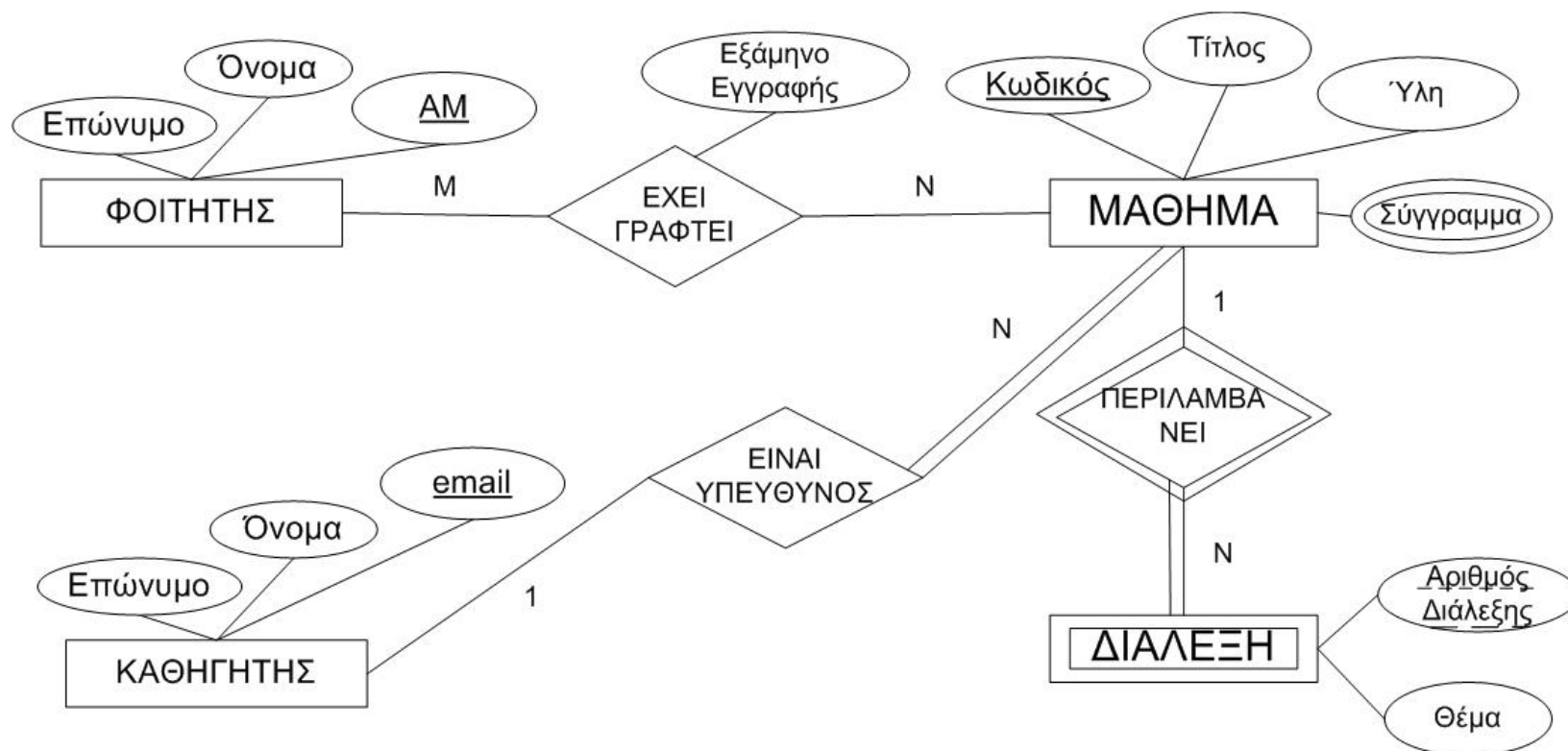
- Η πράξη ON DELETE SET NULL σημαίνει ότι στην περίπτωση που διαγραφεί η αντίστοιχη εγγραφή του πελάτη, το πεδίο τίθεται NULL.
- Η πράξη ON UPDATE CASCADE σημαίνει ότι αν αλλάξει ο αριθμός ταυτότητας του πελάτη, ενημερώνεται αυτόματα το πεδίο buyer_id.

Παράδειγμα: συνολικά η Create table

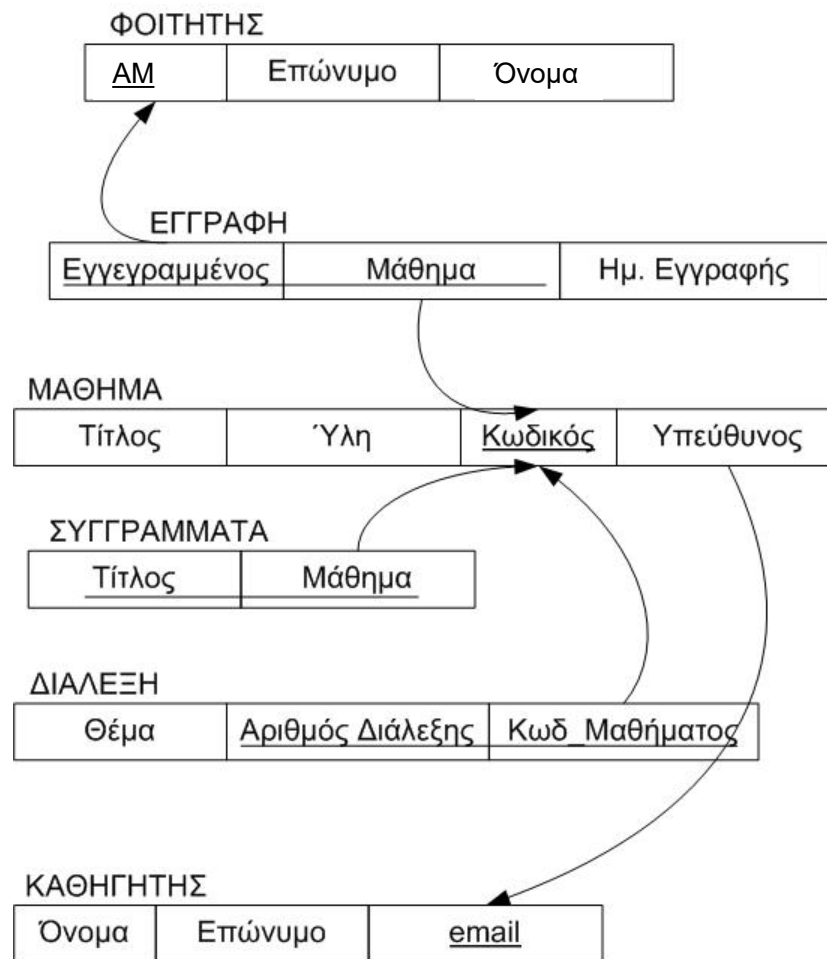
```
mysql> CREATE TABLE product(  
  -> product_id INT(9) NOT NULL AUTO_INCREMENT,  
  -> buyer_id VARCHAR(7),  
  -> product_name VARCHAR(20) DEFAULT 'unknown' NOT NULL,  
  -> PRIMARY KEY(product_id),  
  -> UNIQUE (product_name),  
  -> CONSTRAINT PRDCTBR  
  -> FOREIGN KEY(buyer_id) REFERENCES pelatis(Ar_Tayt)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

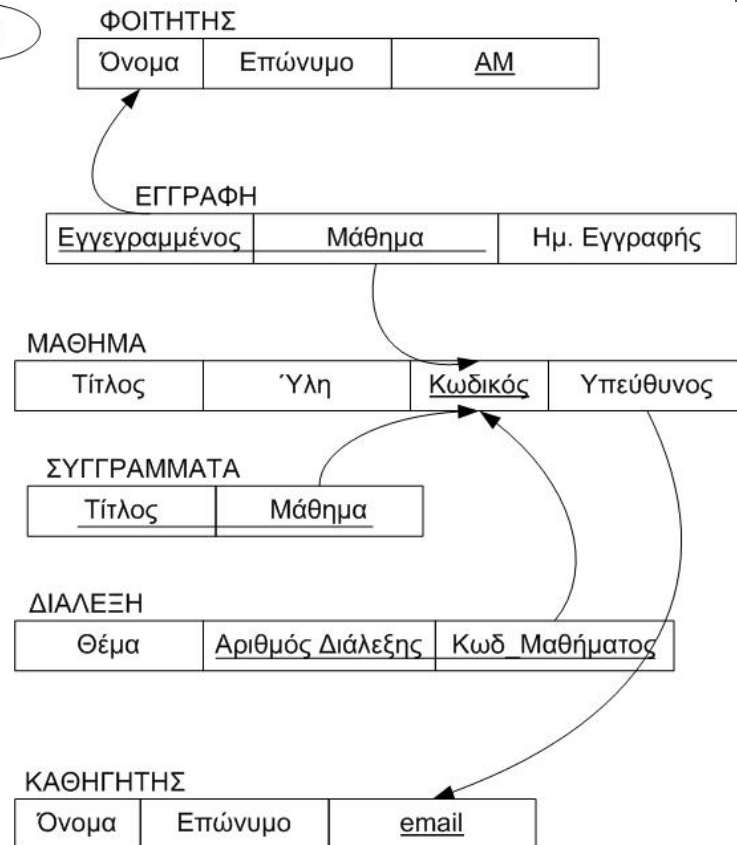
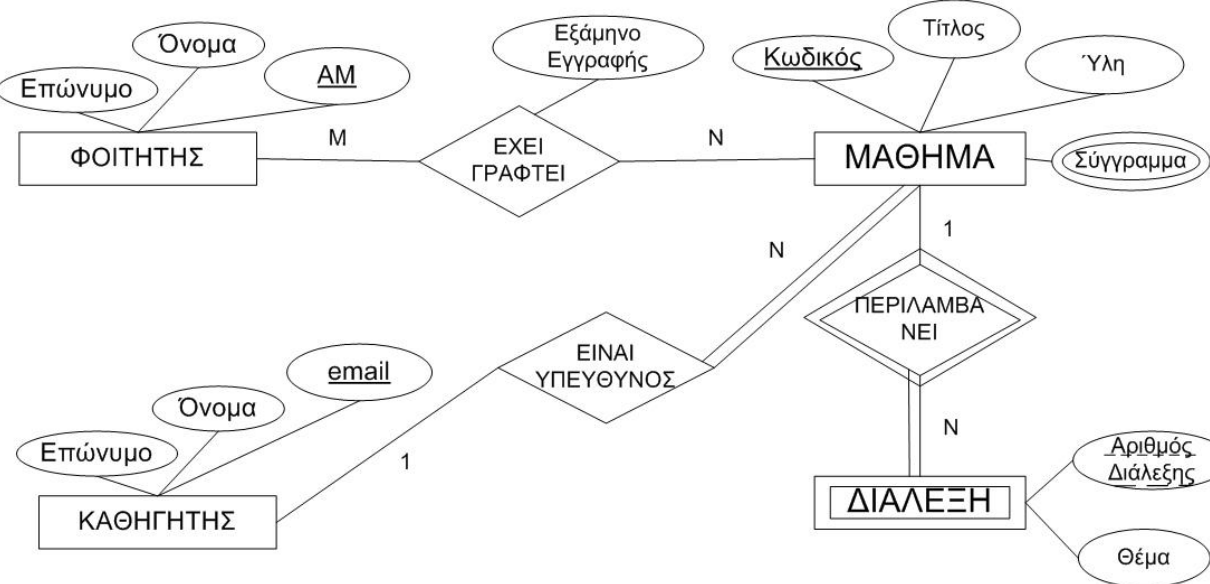
Query OK, 0 rows affected (0.08 sec)

Παράδειγμα - ER



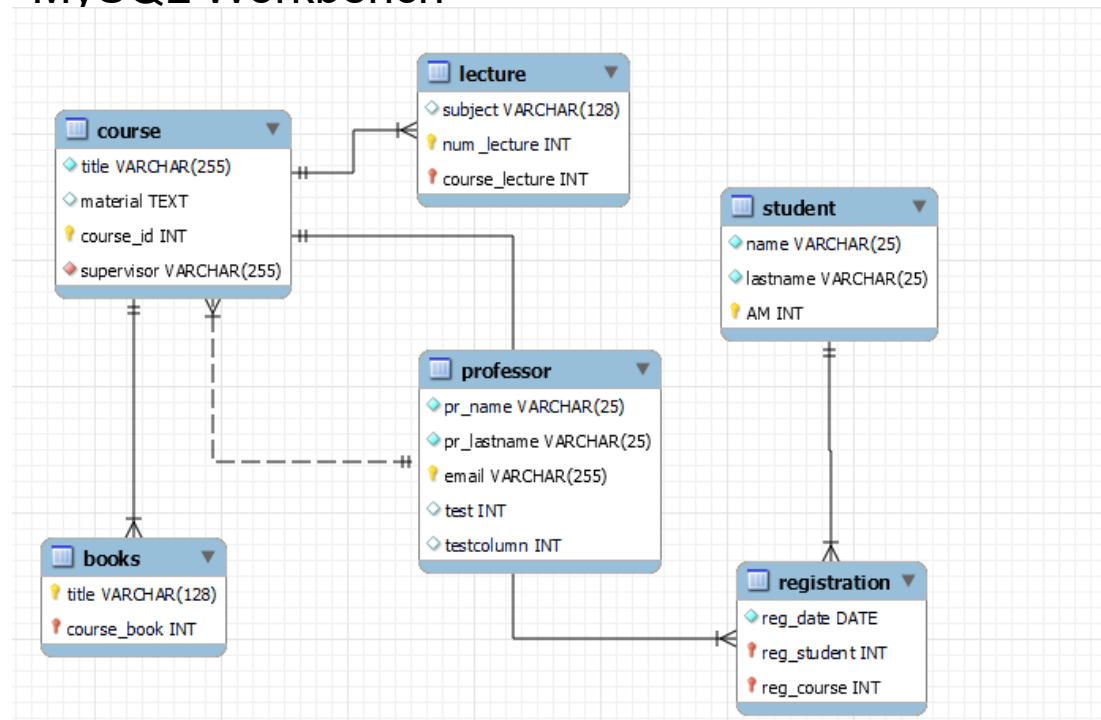
Παράδειγμα-Σχεσιακό





Εναλλακτικές Αναπαραστάσεις ΣΜ

Συνηθισμένη αναπαράσταση στα εργαλεία όπως MySQL Workbench



ΦΟΙΤΗΤΗΣ

Όνομα	Επώνυμο	<u>ΑΜ</u>
-------	---------	-----------

ΕΓΓΡΑΦΗ

<u>Εγγεγραμμένος</u>	<u>Μάθημα</u>	Ημ. Εγγραφής
----------------------	---------------	--------------

ΜΑΘΗΜΑ

Τίτλος	Ύλη	<u>Κωδικός</u>	Υπεύθυνος
--------	-----	----------------	-----------

ΣΥΓΓΡΑΜΜΑΤΑ

<u>Τίτλος</u>	<u>Μάθημα</u>
---------------	---------------

ΔΙΑΛΕΞΗ

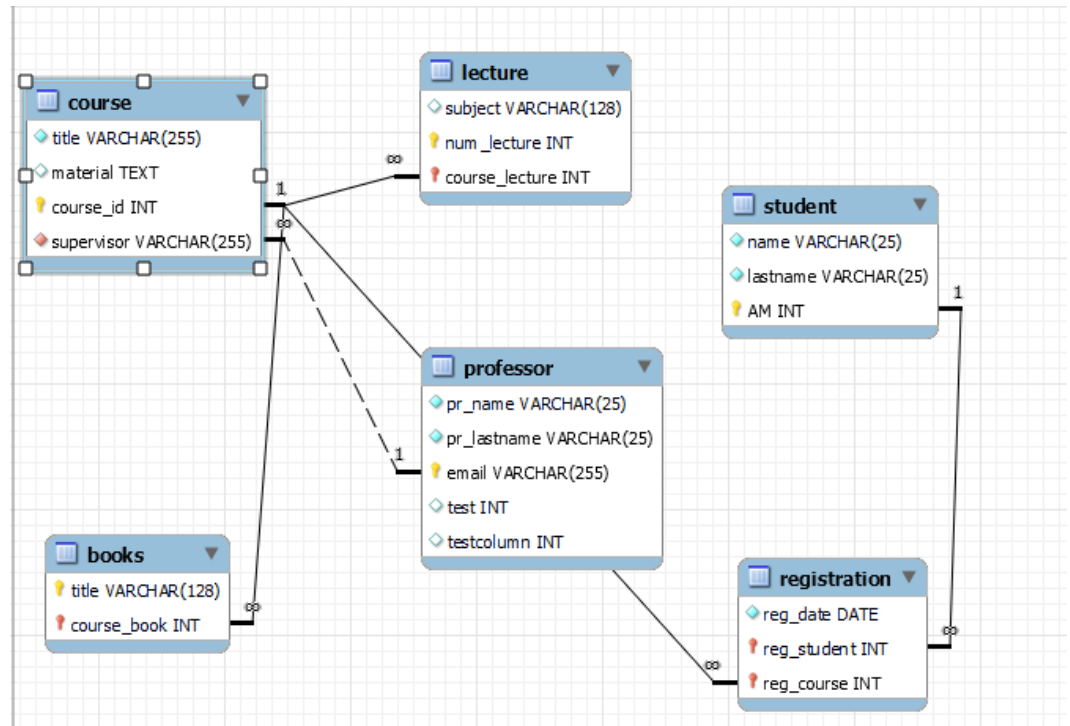
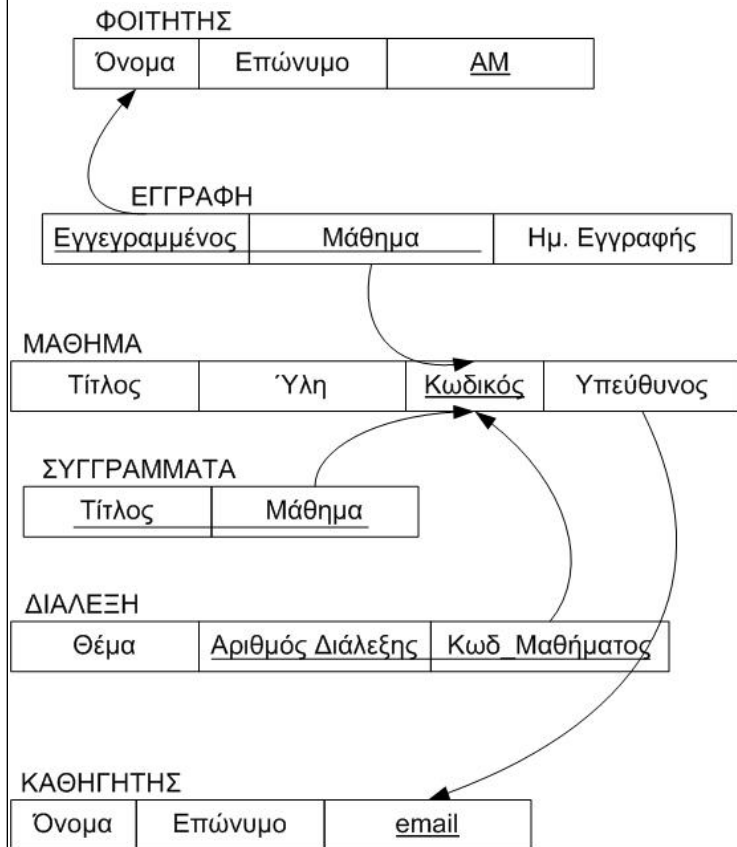
Θέμα	<u>Αριθμός Διάλεξης</u>	<u>Κωδ. Μαθήματος</u>
------	-------------------------	-----------------------

ΚΑΘΗΓΗΤΗΣ

Όνομα	Επώνυμο	<u>email</u>
-------	---------	--------------

Εναλλακτικές Αναπαραστάσεις ΣΜ

Συνηθισμένη αναπαράσταση στα εργαλεία όπως MySQL Workbench



Παράδειγμα – Δημιουργία Πινάκων

Προσοχή στη σειρά δημιουργίας των πινάκων!

ΦΟΙΤΗΤΗΣ

Όνομα	Επώνυμο	<u>AM</u>
-------	---------	-----------

```
CREATE TABLE student (
```

```
    name
```

```
    lastname
```

```
    AM
```

```
    PRIMARY KEY
```

```
);
```

```
CREATE TABLE professor (
```

```
    pr_name
```

```
    pr_lastname
```

```
    email
```

```
    PRIMARY KEY
```

```
);
```

ΚΑΘΗΓΗΤΗΣ

Όνομα	Επώνυμο	<u>email</u>
-------	---------	--------------

Παράδειγμα – Δημιουργία Πινάκων

Προσοχή στη σειρά δημιουργίας των πινάκων!

ΦΟΙΤΗΤΗΣ

Όνομα	Επώνυμο	<u>AM</u>
-------	---------	-----------

```
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  
  pr_lastname  
  email  
  PRIMARY KEY  
);
```

ΚΑΘΗΓΗΤΗΣ

Όνομα	Επώνυμο	<u>email</u>
-------	---------	--------------

Παράδειγμα – Δημιουργία Πινάκων

Προσοχή στη σειρά δημιουργίας των πινάκων!

ΦΟΙΤΗΤΗΣ

Όνομα	Επώνυμο	<u>AM</u>
-------	---------	-----------

```
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)  
);
```

ΚΑΘΗΓΗΤΗΣ

Όνομα	Επώνυμο	<u>email</u>
-------	---------	--------------

Σε command line

Δημιουργία της βάσης maria
και των πρώτων πινάκων

```
mysql> show databases;
+-----+
| Database |
+-----+
| empty    |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
| travel_agency |
+-----+
6 rows in set (0.00 sec)

mysql> create database maria;
Query OK, 1 row affected (0.01 sec)
```

```
mysql> use maria;
Database changed
mysql> 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)
  -> );
Query OK, 0 rows affected, 1 warning (0.02 sec)

mysql> show tables;
+-----+
| Tables_in_maria |
+-----+
| student          |
+-----+
1 row in set (0.00 sec)

mysql> 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)
  -> );
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-----+
| Tables_in_maria |
+-----+
| professor        |
| student          |
+-----+
2 rows in set (0.00 sec)

mysql>
```

```

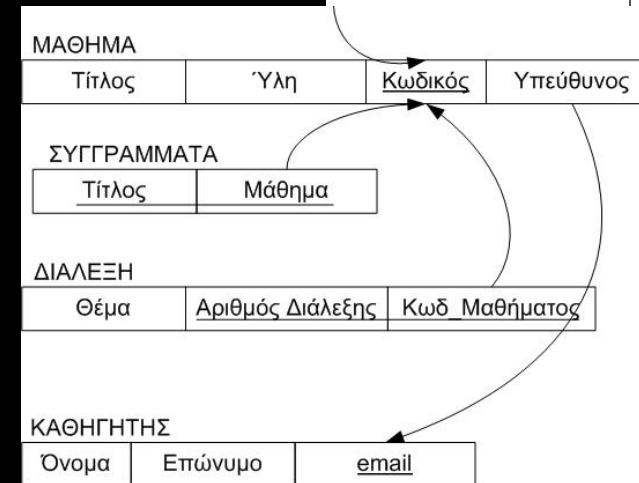
CREATE TABLE course(
    title
    material
    course_id
    supervisor
    PRIMARY KEY
    UNIQUE(title),
    CONSTRAINT SUPERVISED
    FOREIGN KEY REFERENCES
);

```

```

CREATE TABLE books(
    title
    course_book
    PRIMARY KEY
    CONSTRAINT CRSBOOK
    FOREIGN KEY REFERENCES
    ON DELETE CASCADE ON UPDATE CASCADE
);

```



ON DELETE CASCADE constraint is used in MySQL to delete the rows from the child table automatically, when the rows from the parent table are deleted.

```

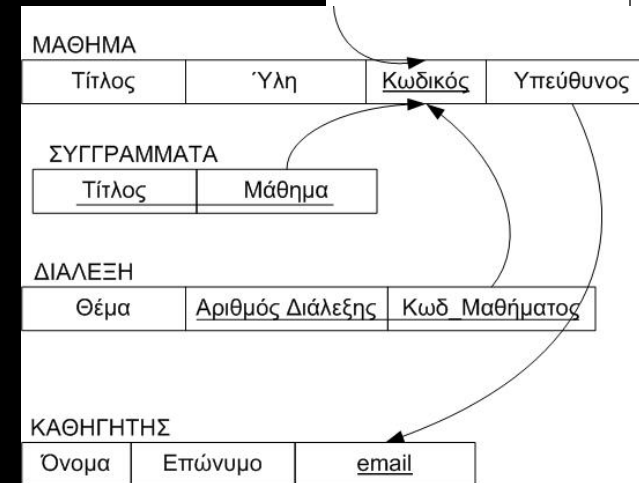
CREATE TABLE course(
    title VARCHAR(255) DEFAULT 'unknown' NOT NULL,
    material TEXT,
    course_id INT(4) NOT NULL AUTO_INCREMENT,
    supervisor VARCHAR(255) NOT NULL,
    PRIMARY KEY(course_id),
    UNIQUE(title),
    CONSTRAINT SUPERVISED
    FOREIGN KEY (supervisor) REFERENCES professor(email)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

```



ON DELETE CASCADE constraint is used in MySQL to delete the rows from the child table automatically, when the rows from the parent table are deleted.

```

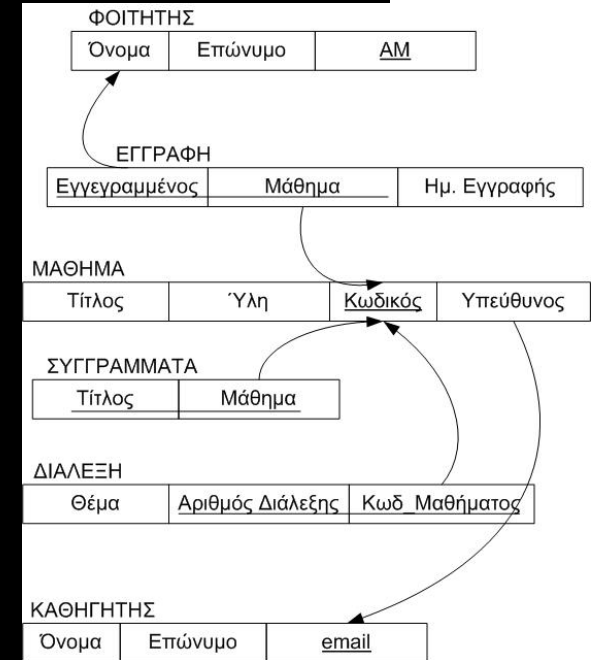
CREATE TABLE IF NOT EXISTS lecture(
    subject VARCHAR(128),
    num_lecture INT(2) NOT NULL,
    course_lecture INT(4) NOT NULL,
    PRIMARY KEY(num_lecture, course_lecture),
    CONSTRAINT CRSLECTURE FOREIGN KEY (course_lecture)
    REFERENCES course(course_id)
    ON DELETE CASCADE ON UPDATE CASCADE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

```

```

CREATE TABLE registration(
    reg_date DATE NOT NULL,
    reg_student INT(5) NOT NULL,
    reg_course INT(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
);

```



Εισαγωγή δεδομένων σε πίνακα

- Η εισαγωγή μιας εγγραφής σε πίνακα γίνεται με την εντολή insert.
INSERT INTO table_name [(column1, column2...)]
VALUES (value1, value2,...)
- Όταν συμπεριλαμβάνουμε τιμές για όλες τις στήλες δεν χρειάζεται να αναφέρουμε τα ονόματα των στηλών (not recommended!).
- Σε όποια στήλη δεν θέλουμε να προσδιορίσουμε τιμή μπορούμε αντί για τιμή να βάλουμε NULL ή DEFAULT.
- **Αν δοθεί η τιμή NULL σε ένα πεδίο με την ιδιότητα AUTO_INCREMENT, τότε η τιμή θα υπολογιστεί αυτόματα.**
- Μπορούμε να εισάγουμε πολλαπλές εγγραφές με μια insert ως εξής:
INSERT INTO table **VALUES**
 (value11,value12,..., value1n),
 (value21,value22,..., value2n),
 ...
 (valuen1,valuen2,..., valuenn);

Tips

- Προσέχω τη σειρά που κάνω insert στους πίνακες όταν έχουμε foreign keys
- Δεν εισάγω τιμή σε auto increment (εκτός αν έχω λόγο να το κάνω). Στην εντολή insert στο πεδίο που είναι auto increment βάζω **null**.
- Αν έχω πίνακα B συσχετισμένο με πίνακα A (foreign key) θα εισάγω τιμή στο πεδίο του B που συσχετίζεται με το A, την ίδια τιμή που εισήγαγα στο συσχετιζόμενο πεδίο στον A

Εισαγωγή δεδομένων σε πίνακα

```
INSERT INTO professor(pr_name,pr_lastname,email) VALUES (DEFAULT,  
'Papadopoulos','pap@ceid.gr');
```

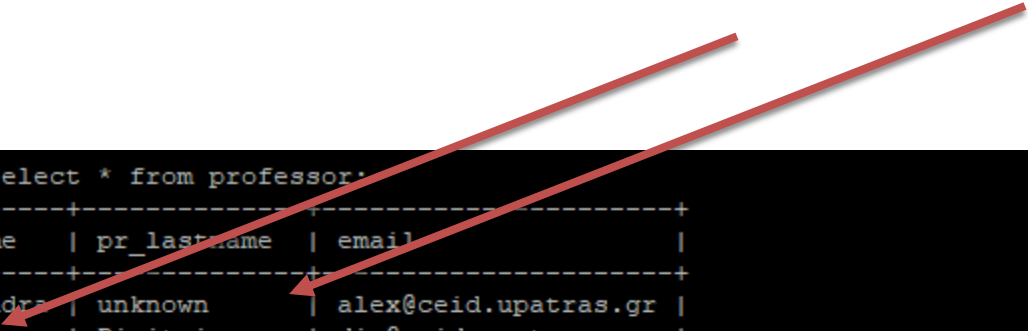
Ή

```
INSERT INTO professor VALUES (DEFAULT, 'Georgiou','geo@ceid.gr');
```

ΠΡΟΣΟΧΗ!

- **INSERT INTO professor values**(DEFAULT,'Georgiou');

ERROR 1136 (21S01): Column count doesn't match value count at row 1



```
mysql> select * from professor;
```

pr_name	pr_lastname	email
Alexandra	unknown	alex@ceid.upatras.gr
unknown	Dimitriou	dim@ceid.upatras.gr
Maria	Papadopoulou	pap@ceid.upatras.gr

```
3 rows in set (0.02 sec)
```

```
mysql> select * from course;
```

title	material	course_id	supervisor
Databases	Intro to relational DBs	2	pap@ceid.upatras.gr
Databases II	Advanced DBs	3	alex@ceid.upatras.gr

```
2 rows in set (0.00 sec)
```

Delete

- Η διαγραφή **μιας εγγραφής** σε πίνακα γίνεται με την εντολή DELETE. Γενική σύνταξη:

DELETE FROM table_name

[**WHERE** some_column=some_value];

- Η συνθήκη **where** προσδιορίζει **ποια/ποιες εγγραφές θα διαγραφούν**
- Χωρίς το WHERE θα διαγραφούν **όλες** οι εγγραφές
- Παραδείγματα:

DELETE FROM professor; // διαγράφει τα πάντα

DELETE FROM professor **WHERE** pr_lastname like '%eorgiou%';

Παράδειγμα με Delete

```
mysql> CREATE TABLE buildings (  
->     building_no INT PRIMARY KEY AUTO_INCREMENT,  
->     building_name VARCHAR(255) NOT NULL,  
->     address VARCHAR(255) NOT NULL  
-> );  
Query OK, 0 rows affected (0.12 sec)
```

```
mysql> CREATE TABLE rooms (  
->     room_no INT PRIMARY KEY AUTO_INCREMENT,  
->     room_name VARCHAR(255) NOT NULL,  
->     building_no INT NOT NULL,  
->     FOREIGN KEY (building_no)  
->         REFERENCES buildings (building_no)  
->         ON DELETE CASCADE  
-> );  
Query OK, 0 rows affected (0.11 sec)
```



```
mysql> INSERT INTO buildings(building_name,address)  
-> VALUES('ACME Headquarters','3950 North 1st Street CA 95134'),  
->         ('ACME Sales','5000 North 1st Street CA 95134');  
Query OK, 2 rows affected (0.04 sec)  
Records: 2  Duplicates: 0  Warnings: 0
```

```
mysql> SELECT * FROM buildings;  
+-----+-----+-----+  
| building_no | building_name | address |  
+-----+-----+-----+  
|          1 | ACME Headquarters | 3950 North 1st Street CA 95134 |  
|          2 | ACME Sales | 5000 North 1st Street CA 95134 |  
+-----+-----+-----+  
2 rows in set (0.00 sec)
```

```
mysql> INSERT INTO rooms(room_name,building_no)
-> VALUES('Amazon',1),
->         ('War Room',1),
->         ('Office of CEO',1),
->         ('Marketing',2),
->         ('Showroom',2);
```

Query OK, 5 rows affected (0.02 sec)
Records: 5 Duplicates: 0 Warnings: 0

```
mysql> SELECT * FROM rooms;
```

room_no	room_name	building_no
1	Amazon	1
2	War Room	1
3	Office of CEO	1
4	Marketing	2
5	Showroom	2

5 rows in set (0.00 sec)

```
mysql> DELETE FROM buildings
-> WHERE building_no = 2;
```

Query OK, 1 row affected (0.04 sec)

```
mysql> SELECT * FROM rooms;
```

room_no	room_name	building_no
1	Amazon	1
2	War Room	1
3	Office of CEO	1

3 rows in set (0.00 sec)

Παράδειγμα με DELETE

```
| name          | lastname          | AM |
+-----+-----+
| βασιλης | παπαϊωαννου | 1 |
| Spiros   | Petridis     | 2190 |
| Vivi    | Tzekou       | 2191 |
| unknown  | Dourou       | 2192 |
| Athanasia | Koumpouri    | 2193 |
| Vaso     | unknown      | 2194 |
+-----+-----+
6 rows in set (0.00 sec)
```

DELETE FROM student WHERE name like '%βασ%';

```
mysql> delete from student where name like '%βασ%';
Query OK, 1 row affected (0.08 sec)

mysql> select * from student;
+-----+-----+-----+
| name      | lastname | AM    |
+-----+-----+-----+
| Spiros    | Petridis | 2190  |
| Vivi     | Tzekou   | 2191  |
| unknown   | Dourou   | 2192  |
| Athanasia | Koumpouri | 2193  |
| Vaso      | unknown  | 2194  |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

Update

- Η ανανέωση μιας εγγραφής σε πίνακα γίνεται με την εντολή UPDATE:

UPDATE table_name

SET column1=value1, column2=value2,...

WHERE some_column=some_value;

- Η συνθήκη **where** προσδιορίζει ποιες εγγραφές θα ανανεωθούν
- **Χωρίς το where** θα ανανεωθούν **όλες** οι εγγραφές

UPDATE professor

SET pr_name='Nikos', pr_lastname='Papadakis'

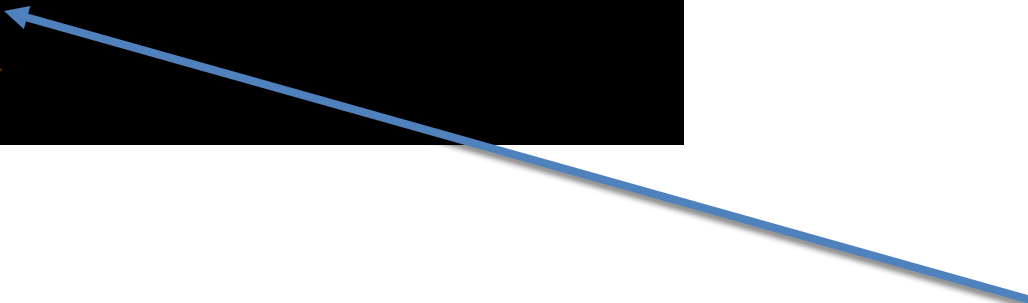
WHERE pr_lastname='unknown';

UPDATE professor

SET salary=salary*1,05

WHERE salary<2000;

```
mysql> select * from student;
+-----+-----+-----+
| name   | lastname | AM   |
+-----+-----+-----+
| Spiros | Petridis | 2190 |
| Vivi  | Tzekou  | 2191 |
| unknown | Dourou   | 2192 |
| Athanasia | Koumpouri | 2193 |
| Vaso   | unknown  | 2194 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```



UPDATE student

SET NAME = 'Eleni', lastname='Voyiatzaki',

WHERE lastname='Koumpouri';

```
mysql> update student
    -> set name = 'Eleni', lastname='Voyiatzaki' where lastname='Koumpouri';
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from student;
+-----+-----+-----+
| name   | lastname | AM   |
+-----+-----+-----+
| Spiros | Petridis | 2190 |
| Vivi  | Tzekou  | 2191 |
| unknown | Dourou   | 2192 |
| Eleni  | Voyiatzaki | 2193 |
| Vaso   | unknown  | 2194 |
+-----+-----+-----+
5 rows in set (1.42 sec)

mysql> █
```


ALTER

- Μπορούμε να αλλάξουμε τη **δομή** ενός πίνακα
 - Δηλ. να προσθέσουμε, να διαγράψουμε και να τροποποιήσουμε κάποια στήλη
- Γενική σύνταξη:
ALTER TABLE table_name **ADD** | | **DROP** | | **MODIFY**
column_name datatype;
- Παραδείγματα:
ALTER TABLE professor **DROP** pr_name, **ADD** address
VARCHAR(70) NOT NULL;
ALTER TABLE student **MODIFY** name varchar(30) not
null;

ALTER ...ADD

```
mysql> select * from professor;
```

pr_name	pr_lastname	email
Alexandra	unknown	alex@ceid.upatras.gr
unknown	Dimitriou	dim@ceid.upatras.gr
Maria	Papadopoulou	pap@ceid.upatras.gr

3 rows in set (0.02 sec)

ALTER TABLE professor DROP pr_name, ADD address VARCHAR(70) NOT NULL;

```
mysql> ALTER TABLE professor DROP pr_name, ADD address VARCHAR(70) NOT NULL;  
Query OK, 3 rows affected (0.32 sec)  
Records: 3  Duplicates: 0  Warnings: 0
```

```
mysql> select * from professor;
```

pr_lastname	email	address
unknown	alex@ceid.upatras.gr	
Dimitriou	dim@ceid.upatras.gr	
Papadopoulou	pap@ceid.upatras.gr	

3 rows in set (0.00 sec)

Can we alter primary key in MySQL?

To change the primary key of a table, delete the existing key using a DROP clause in an ALTER TABLE statement and add the new primary key.

ALTER...MODIFY

```
mysql> describe student;
```

Field	Type	Null	Key	Default	Extra
name	varchar(25)	NO		unknown	
lastname	varchar(25)	NO		unknown	
AM	int(5)	NO	PRI	NULL	auto increment

3 rows in set (0.01 sec)

- ALTER TABLE student MODIFY name varchar(30) NOT NULL;

```
mysql> alter table student modify name varchar(30) not null;
```

```
Query OK, 5 rows affected (0.21 sec)
```

```
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> describe student;
```

Field	Type	Null	Key	Default	Extra
name	varchar(30)	NO		NULL	
lastname	varchar(25)	NO		unknown	
AM	int(5)	NO	PRI	NULL	auto_increment

3 rows in set (0.01 sec)

ALTER...MODIFY

```
mysql> describe student;
```

Field	Type	Null	Key	Default	Extra
name	varchar(30)	NO		NULL	
lastname	varchar(25)	NO		unknown	
AM	int(5)	NO	PRI	NULL	auto_increment

3 rows in set (0.01 sec)

- ALTER table student MODIFY name NULL;

```
mysql> alter table student modify name varchar(30) null;
```

```
Query OK, 5 rows affected (0.25 sec)
```

```
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> describe student;
```

Field	Type	Null	Key	Default	Extra
name	varchar(30)	YES		NULL	
lastname	varchar(25)	NO		unknown	
AM	int(5)	NO	PRI	NULL	auto_increment

3 rows in set (0.00 sec)

```
mysql> CREATE TABLE course(  
-> title VARCHAR(255) DEFAULT 'unknown' NOT NULL,  
-> 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  
-> );
```

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near "unknown' N NULL,material TEXT,course_id INT(4) NOT NULL AUTO_INCREMENT,'" at line 2

Προσοχή στα ' '

```
mysql> CREATE TABLE course(  
-> title VARCHAR(255) NOT NULL,  
-> 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  
-> );
```

Query OK, 0 rows affected, 1 warning (0.07 sec)

```
mysql> alter table course modify title VARCHAR(255) Default 'unknown' not null;
```

Query OK, 0 rows affected (0.02 sec)

Records: 0 Duplicates: 0 Warnings: 0

- Δευτέρα 3/11 θα γίνει εξέταση της Create-Insert. (ομάδες από 10:30-14:30)
- Θα ανέβει στο eclass η άσκηση εξάσκησης και ακολούθως την Κυριακή μια προτεινόμενη λύση.
- **Στις ομάδες του εργαστηρίου πρέπει να εγγραφείτε έως Δευτέρα 27/10 23:00. (ΑΠΟΚΛΕΙΣΤΙΚΗ ΠΡΟΘΕΣΜΙΑ)**