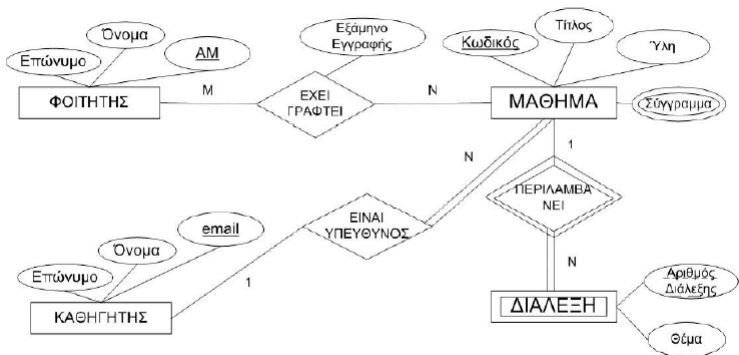
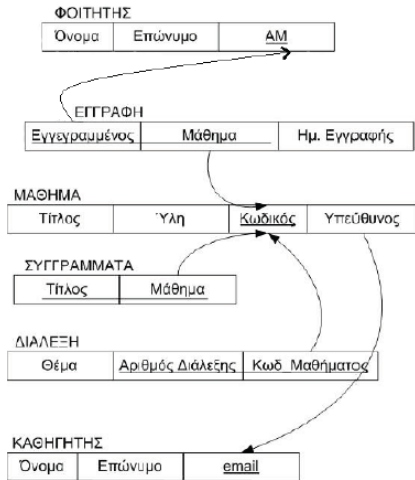


## Εργαστήριο Βάσεων Δεδομένων

Εισαγωγή στη MySQL (2)



# Σχεσιακό



## Create Tables 1/4

```
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 2/4

```
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 Tables 3/4

```
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 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 Tables 4/4

```

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 STDN'TREGISTRATION
  FOREIGN KEY (reg_student) REFERENCES student(AM)
  ON DELETE CASCADE ON UPDATE CASCADE
);
  
```

## Δεδομένα 1/2

<b>professor</b>	pr_name	pr_lastname	email
	Μαρία	Παπαδοπούλου	pap@ceid.upatras.gr
	unknown	Δημητρίου	dim@ceid.upatras.gr
	Αλεξάνδρα	unknown	alex@ceid.upatras.gr

<b>student</b>	name	lastname	AM
	Βιβή	Τζέκου	2191
	unknown	Ντούρου	2192
	Αθανασία	Κουμπούρη	2193
	Βάσω	unknown	2194

<b>course</b>	title	material	course_id	supervisor
	Βάσεις Δεδομένων	Εισαγωγή σε σχεσια- κές βάσεις δεδομέ- νων	2	pap@ceid.upatras.gr
	Βάσεις Δεδομένων II	Προχωρημένα θέματα βάσεων δεδομένων	3	alex@ceid.upatras.gr



## Δεδομένα 2/2

<b>books</b>	title	course_book
	Databases 1 1st volume	2
	Databases 1 2nd volume	2
	Databases 2 1st volume	3

<b>lecture</b>	subject	num_lecture	course_lecture
	Εισαγωγή σε βάσεις	1	2
	Ανάλυση απαιτήσεων	2	2
	ER-Σχεσιακό	3	2
	Κανονικοποίηση βάσεων	1	3
	Βελτιστοποίηση	2	3

<b>registration</b>	reg_date	reg_student	reg_course
	2010-08-09	2191	2
	2011-02-11	2193	2
	2012-06-03	2191	3
	2010-01-12	2192	3
	2011-04-19	2194	3

## Εισαγωγή δεδομένων 1/2

- Η εισαγωγή μιας εγγραφής σε πίνακα γίνεται με την εντολή **insert**
- Η εντολή insert συντάσσεται ως εξής:

```
INSERT INTO table_name (column1, column2, column3,...)  
VALUES (value1, value2, value3,...)
```

- Όταν συμπεριλαμβάνουμε τιμές για **όλες** τις στήλες δεν χρειάζεται να αναφέρουμε τη λίστα των τιμών

```
INSERT INTO table_name VALUES (value1, value2, value3,...)
```

## Εισαγωγή δεδομένων 2/2

- 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_name VALUES
```

```
(value1, value12, ..., value1n),
```

```
(value21, value22, ..., value2n),
```

```
...
```

```
(valuen1, valuen2, ..., valuen3);
```

## Παράδειγμα

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

**Ή**

```
INSERT INTO professor VALUES  
(DEFAULT, 'Παπαδόπουλος', 'pap@ceid.gr') ;
```

## Ανανέωση εγγραφής

- Η ανανέωση μιας εγγραφής σε πίνακα γίνεται με την εντολή **update**
- Η εντολή update συντάσσεται ως εξής:

```
UPDATE table_name SET column1=value1, column2=value2,...  
WHERE some_column=some_value;
```

- Η συνθήκη **where** προσδιορίζει ποια/ποιες εγγραφές θα ανανεωθούν
- Εάν παραλείψουμε το where θα ανανεωθούν **όλες** οι εγγραφές

## Παράδειγμα

```
UPDATE professor SET pr_name='Νίκος',pr_lastname='Δασκαλάκης'  
WHERE pr_lastname='unknown' ;
```

## Διαγραφή εγγραφής

- Η διαγραφή μιας εγγραφής σε πίνακα γίνεται με την εντολή **delete**
- Η εντολή delete συντάσσεται ως εξής:

```
DELETE FROM table_name WHERE some_column=some_value ;
```

- Η συνθήκη **where** προσδιορίζει ποια/ποιες εγγραφές θα διαγραφούν
- Εάν παραλείψουμε το where θα διαγραφούν **όλες** οι εγγραφές

## Παράδειγμα

```
DELETE FROM professor;
```

```
DELETE FROM professor WHERE pr_lastname='Γεωργίου';
```



## Τροποποίηση πίνακα

- Η προσθήκη, διαγραφή και τροποποίηση μιας στήλης ενός πίνακα γίνεται με την εντολή **alter**
- Η εντολή alter συντάσσεται ως εξής:

```
ALTER TABLE table_name ADD||DROP||ALTER column_name  
datatype;
```

- Παράδειγμα:

```
ALTER TABLE professor DROP pr_name, ADD address varchar(70)  
not null;
```