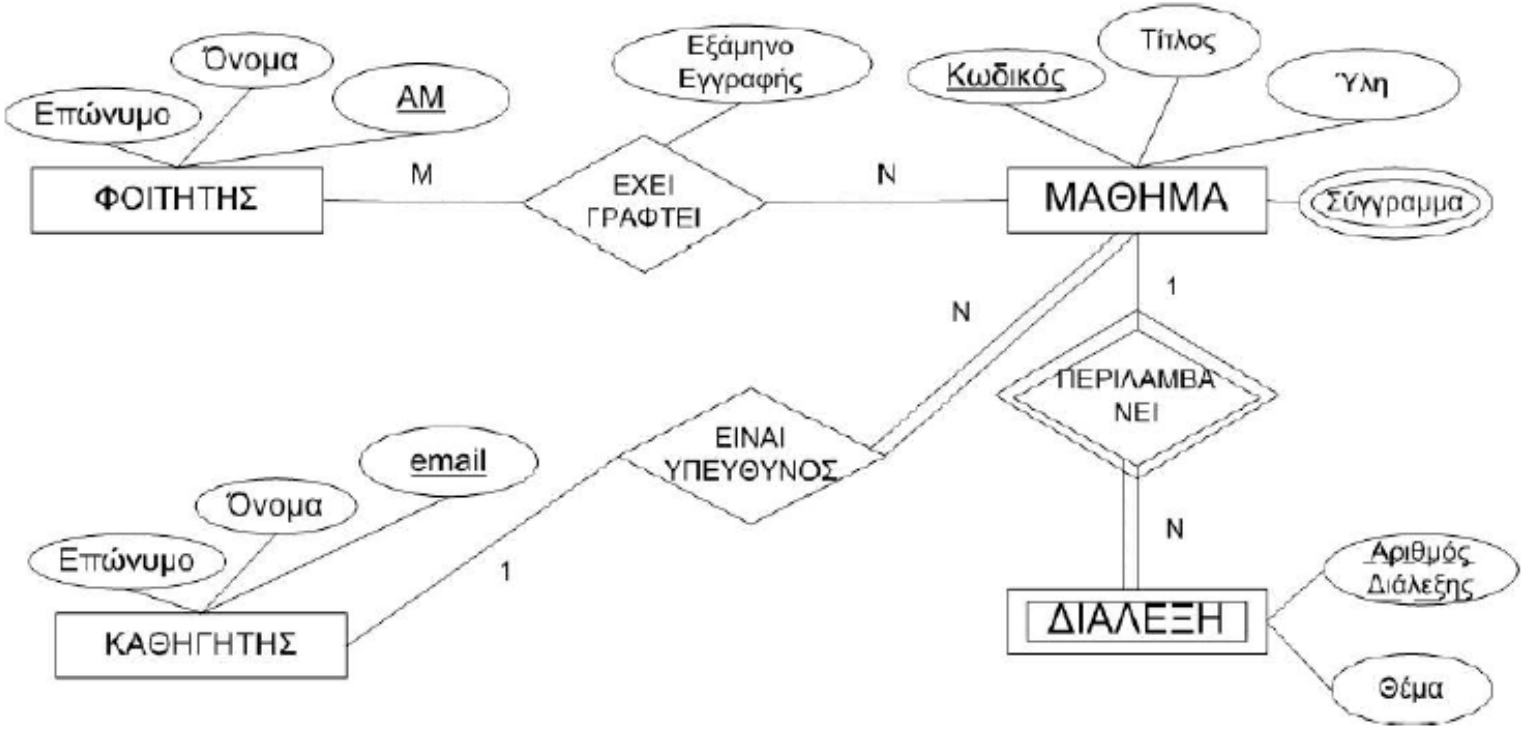
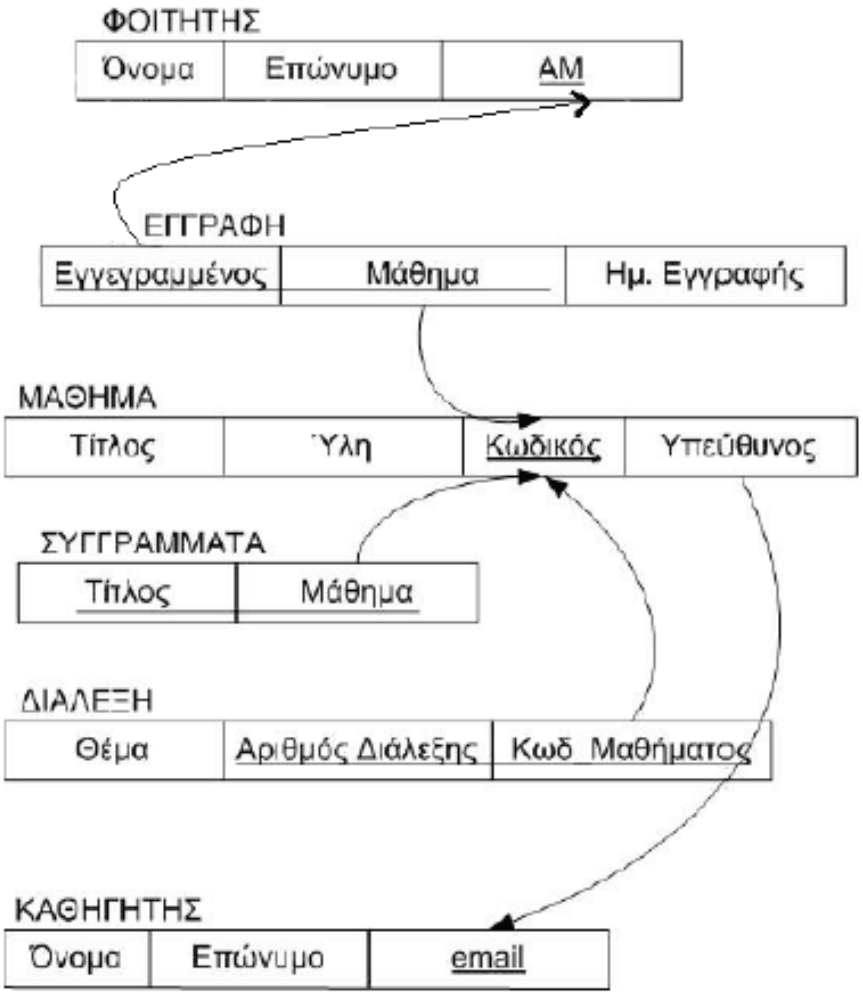
Εργαστήριο Βάσεων Δεδομένων

Εισαγωγή στη MySQL (3)

ER


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

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 1/4

```
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_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
);
```

Δεδομένα 1/2

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

student	name	lastname	AM
	Βιβή	Τζέκου	2191
	unknown	Ντούρου	2192
	Αθανασία	Κουμπούρη	2193
	Βάσω	unknown	1845

course	title	material	course_id	supervisor
	Βάσεις Δεδομένων	Εισαγωγή σε σχεσιακές βάσεις δεδομένων	2	pap@ceid.upatras.gr
	Βάσεις Δεδομένων II	Προχωρημένα θέματα βάσεων δεδομένων	3	alex@ceid.upatras.gr

Δεδομένα 2/2

books	title	course_book
	Databases 1	2
	Databases 1 2nd volume	2
	Databases 2	3

lecture	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

registration	reg_date	reg_student	reg_course
	2010-08-09	1845	2
	2011-02-11	2191	2
	2012-06-03	2192	3
	2010-01-12	2193	3
	2011-04-19	2191	3

Ανάκτηση πληροφορίας από τη βάση

- Η ανάκτηση πληροφοριών από τη βάση γίνεται με την εντολή **select**
  - Απάντηση σε μια ερώτηση select είναι ένας προσωρινός πίνακας
  - Για το σχηματισμό του πίνακα η select καθορίζει:
    - Ποιες στήλες θα περιέχει η απάντηση (λίστα στηλών)
    - Ποιες γραμμές θα περιέχει η απάντηση (εφαρμογή της συνθήκης επιλογής)

- Η βασική σύνταξη της select είναι ως εξής:

SELECT <λίστα στηλών>
FROM <λίστα πινάκων>
WHERE <συνθήκη>

Select: Γενική Σύνταξη

- Λίστα πινάκων:
  - Ονόματα των πινάκων οι οποίοι θα συμμετάσχουν στην επιλογή δεδομένων
  - Αν είναι περισσότεροι από ένας, τότε υπάρχουν διάφοροι τρόποι συνδυασμού τους (join)
- Λίστα στηλών:
  - Λίστα ονομάτων στηλών που χωρίζονται με κόμμα
  - Μόνο αυτές οι στήλες περιλαμβάνονται στην απάντηση
  - Αν τις θέλουμε όλες, μπορούμε να χρησιμοποιήσουμε το *

Παραδείγματα:

SELECT name, lastname
FROM student;
SELECT *
FROM professor;

Select: Συνθήκη WHERE

- Λογική συνθήκη μεταξύ πεδίων του πίνακα και αριθμών
- Περιλαμβάνονται στην απάντηση μόνο οι γραμμές που την ικανοποιούν
- Χρησιμοποιούνται οι τελεστές:

Boolean AND, OR, NOT, ...

Comperative Σύγκριση μεταξύ πεδίων και κυριολεκτικών

Σύγκριση βαθμωτών =,<>,>,>=,<,<=

Σύγκριση με το NULL IS NULL, IS NOT NULL

Σύγκριση αλφαριθμητικών LIKE (με wildcards)

% Αντικατάσταση μηδέν ή περισσότερων χαρακτήρων

_ Αντικατάσταση ακριβώς ενός χαρακτήρα

Select: Συνθήκη WHERE - Παραδείγματα 1/2

- Επιλογή των ονομάτων των φοιτητών με ΑΜ μεγαλύτερο του 2000

```
SELECT name, lastname FROM student WHERE am>2000;
```

- Επιλογή των εισαγωγικών μαθημάτων (=αυτών που έχουν τη λέξη εισαγωγή στην περιγραφή τους)

```
SELECT * FROM course WHERE material LIKE '%Εισαγωγή%';
```

- Επιλογή των μαθημάτων που έχει εγγραφεί ο φοιτητής με κωδικό 2191

```
SELECT reg_course FROM registration WHERE reg_student=2191;
```

- Επιλογή των τίτλων των μαθημάτων που διδάσκονται από την Μ. Παπαδοπούλου και αφορούν στις βάσεις δεδομένων

```
SELECT title FROM course
WHERE supervisor='pap@ceid.upatras.gr' AND material LIKE '%βάσεις δεδομένων%';
```

Select: Συνθήκη WHERE - Παραδείγματα 2/2

- Επιλογή των στοιχείων των φοιτητών που το επώνυμό τους αρχίζει από Τ

```
SELECT * FROM student WHERE lastname LIKE 'Τ%';
```

- Επιλογή των μαθημάτων που δεν έχουν περιγραφή ύλης

```
SELECT * FROM course WHERE material IS NULL;
```

- Επιλογή των μαθημάτων στα οποία έχει γίνει κάποια εγγραφή φέτος

```
SELECT reg_course FROM registration
WHERE reg_date>='2012-01-01';
```

- Επιλογή των φοιτητών που έχουν εγγραφεί μέσα στο 2011 στο μάθημα με κωδικό 3

```
SELECT reg_student FROM registration
WHERE reg_date>='2011-01-01' AND reg_date<'2012-01-01' AND reg_course=3;
```

Select: Order By

- Ταξινόμηση σύμφωνα με ένα ή περισσότερα πεδία
- Για περισσότερα πεδία ταξινόμησης, τότε ισχύει προτεραιότητα από αριστερά
- Οι τελεστές ASC και DESC ορίζουν αύξουσα ή φθίνουσα σειρά αντίστοιχα
- Παραδείγματα:
  - Επιστροφή των στοιχείων των φοιτητών σε αύξουσα σειρά ως προς το επώνυμό τους

```
SELECT * FROM student ORDER BY lastname ASC;
```

  - Επιστροφή των φετινών εγγραφών από την πιο πρόσφατη στην παλιότερη

```
SELECT * FROM registration WHERE reg_date>='2012-01-01' ORDER BY reg_date DESC;
```

  - Επιστροφή των μαθημάτων ταξινομημένων ανά email καθηγητή και δευτερευόντως ανά τίτλο

```
SELECT * FROM course ORDER BY supervisor ASC, title ASC;
```

Select: Limit

- Καθορίζει ρητά τον αριθμό των εγγραφών που επιστρέφονται
- Σύνταξη: LIMIT m,n
  - Το m είναι η θέση εκκίνησης στο σύνολο των εγγραφών που θα επιστρέφονταν κανονικά (με αρίθμηση από το 0)
  - Το n είναι το πλήθος των εγγραφών που θα επιστραφούν. Αν παραλείπεται επιστρέφονται όλες από το m και μετά
- Συνήθως είναι χρήσιμη μαζί με ταξινόμηση των εγγραφών
- Παραδείγματα:
  - Επιστροφή του φοιτητή με το μεγαλύτερο ΑΜ

```
SELECT * FROM student ORDER BY AM DESC LIMIT 0,1;
```

  - Επιστροφή των τριών πρώτων εγγραφών του τρέχοντος έτους

```
SELECT * FROM registration WHERE reg_date>='2012-01-01' ORDER BY reg_date ASC LIMIT 0,3;
```

Select: Group By

- Ομαδοποίηση εγγραφών σύμφωνα με πεδίο ή πολλαπλά πεδία
- Για τα κριτήρια ομαδοποίησης δημιουργούνται ομάδες στις οποίες ανήκουν οι εγγραφές με τις ίδιες τιμές στα πεδία αυτά
- Για περισσότερα πεδία ομαδοποίησης, οι ομάδες δημιουργούνται με ίδιες τιμές στο συνδυασμό των πεδίων
- Η ομαδοποίηση συνδυάζεται με συναρτήσεις που εφαρμόζονται στις ομάδες, πχ για μέτρηση πλήθους
- Παράδειγμα:
  - Επιστροφή για κάθε φοιτητή του αριθμού των εγγραφών που έχει πραγματοποιήσει

SELECT reg_student, count(*) **FROM** registration **GROUP BY** reg_student;

Επιστρέφει:

student AM	count(*)
1845	1
2191	2
2192	1
2193	1

Ενσωματωμένες Συναρτήσεις

- Συναρτήσεις που χρησιμοποιούνται συνήθως με την group by
- Εφαρμόζονται σε ένα πλήθος εγγραφών που ανήκουν σε μια ομάδα

SUM	Άθροισμα του πεδίου που δέχεται σαν όρισμα
COUNT	Μέτρηση πλήθους
MAX	Μέγιστο
MIN	Ελάχιστο
AVG	Μέσος όρος της τιμής του πεδίου που δέχεται σαν όρισμα

Select: Having

- Συνθήκη που εφαρμόζεται στα αποτελέσματα της group by
  - Προσοχή: αν η συνθήκη αφορά στα αρχικά δεδομένα, τότε πρέπει να εκφραστεί με where!
- Παράδειγμα:
  - Επιστροφή των φοιτητών που έχουν πραγματοποιήσει πάνω από 1 εγγραφές

```
SELECT reg_student, count(*) FROM registration
GROUP BY reg_student HAVING count(*)>1;
```

Επιστρέφει:

student	AM	count(*)
	2191	2

- Επιστροφή των φοιτητών και του αριθμού των εγγραφών τους, λαμβάνοντας υπόψη μόνο τις φετινές εγγραφές

```
SELECT reg_student, count(*) FROM registration
WHERE reg_date>='2012-01-01' GROUP BY reg_student;
```

Select: Group By - Παραδείγματα

- Επιστροφή του αριθμού των μαθημάτων που διδάσκει κάθε καθηγητής

```
SELECT supervisor, COUNT(*) FROM course GROUP BY supervisor;
```

- Επιστροφή για κάθε μάθημα του αριθμού των διαλέξεων που έχουν θέμα με βάσεις (=περιέχουν στον τίτλο τη λέξη)

```
SELECT course_lecture, COUNT(*) FROM lecture
WHERE subject LIKE '%βάσεις%' GROUP BY course_lecture;
```

- Επιστροφή των μαθημάτων που η τελευταία διάλεξη έχει αύξοντα αριθμό πάνω από 2

```
SELECT course_lecture, MAX(num_lecture) FROM lecture
GROUP BY course_lecture HAVING MAX(num_lecture)>2 ;
```

Δεδομένα από πολλαπλούς πίνακες: Πότε?

- Όταν χρειαζόμαστε να συνδυάσουμε πάνω από έναν πίνακες για να επιστρέψουμε τα αποτελέσματα που θέλουμε
- Το σημείο στο οποίο συνδυάζονται είναι συνήθως ένα ξένο κλειδί
- Στόχος είναι να δηλωθεί ποιο πεδίο πρέπει να εξισωθεί με ποιο
- Αν δεν δηλωθεί, τότε έχουμε καρτεσιανό γινόμενο των πινάκων

Παράδειγμα

Θέλουμε τα **ονοματεπώνυμα** των φοιτητών που έχουν πραγματοποιήσει εγγραφή φέτος.
Ποιοι πίνακες περιέχουν την ζητούμενη πληροφορία?

Δεδομένα από πολλαπλούς πίνακες: Εμφωλευμένες Select

- Κάθε select από την πιο εσωτερική εκτελείται και επιστρέφει τα αποτελέσματά της στην εξωτερική της
  - Παράδειγμα: Επιστρέφουμε τα ονοματεπώνυμα των φοιτητών που έχουν πραγματοποιήσει εγγραφή φέτος

```
SELECT name, lastname FROM student
WHERE am IN
(SELECT reg_student FROM registration
WHERE reg_date>='2012-01-01');
```

Επιστρέφει:

student	name	lastname
	unknown	Ντούρου

Δεδομένα από πολλαπλούς πίνακες: Join

- JOIN: Τρόπος δήλωσης της συνένωσης πινάκων για το συνδυασμό πληροφοριών, κυρίως όταν έχουμε συσχέτιση.
  - Παράδειγμα: Επιστρέφουμε τον αριθμό της κάθε διάλεξης, το θέμα της καθώς και τον τίτλο του μαθήματος στο οποίο ανήκει.

```
SELECT lecture.num_lecture, lecture.subject, course.title
FROM lecture
INNER JOIN course ON course.course_id=lecture.course_lecture
ORDER BY course.course_id ASC, lecture.num_lecture ASC;
```

Επιστρέφει:

num_lecture	subject	title
1	Εισαγωγή σε βάσεις	Βάσεις Δεδομένων
2	Ανάλυση απαιτήσεων	Βάσεις Δεδομένων
3	ER-Σχεσιακό	Βάσεις Δεδομένων
1	Κανονικοποίηση Βάσεων	Βάσεις Δεδομένων II
2	Βελτιστοποίηση Βάσεων	Βάσεις Δεδομένων II

Δεδομένα από πολλαπλούς πίνακες: Παράθεση πινάκων

- Παραθέτουμε στη λίστα πινάκων όλους όσους χρειαζόμαστε
- Για να μη γίνει καρτεσιανό γινόμενο, στη where ορίζουμε τη συνθήκη συνένωσης
    - Για το προηγούμενο παράδειγμα:

```
SELECT lecture.num_lecture, lecture.subject, course.title
FROM lecture, course
WHERE course.course_id=lecture.course_lecture
ORDER BY course.course_id ASC, lecture.num_lecture ASC;
```

Επιστρέφει:

num_lecture	subject	title
1	Εισαγωγή σε βάσεις	Βάσεις Δεδομένων
2	Ανάλυση απαιτήσεων	Βάσεις Δεδομένων
3	ER-Σχεσιακό	Βάσεις Δεδομένων
1	Κανονικοποίηση Βάσεων	Βάσεις Δεδομένων II
2	Βελτιστοποίηση Βάσεων	Βάσεις Δεδομένων II

Είδη Join

INNER JOIN

Μια εγγραφή συμπεριλαμβάνεται μόνο αν υπάρχει εγγραφή που να της "ταιριάζει".

LEFT JOIN

Οι εγγραφές του αριστερού πίνακα συμπεριλαμβάνονται πάντα, ανεξάρτητα από το αν υπάρχει εγγραφή στο δεξιό πίνακα που να τους ταιριάζει.

RIGHT JOIN

Ομοίως για τον δεξιό πίνακα.

Case Study: Ερώτηση σε M-N

- Θέλουμε το ονοματεπώνυμο των φοιτητών και το όνομα του μαθήματος στα οποία έχουν εγγραφεί, για κάθε εγγραφή που έχουν κάνει.

```
SELECT name, lastname, title
FROM student
INNER JOIN registration ON am=reg_student
INNER JOIN course ON course_id=reg_course
ORDER BY lastname ASC, name ASC, title ASC;
```

Επιστρέφει:

name	lastname	title
unknown	Ντούρου	Βάσεις Δεδομένων II
Αθανασία	Κουμπούρη	Βάσεις Δεδομένων II
Βάσω	unknown	Βάσεις Δεδομένων
Βιβή	Τζέκου	Βάσεις Δεδομένων
Βιβή	Τζέκου	Βάσεις Δεδομένων II

Συνένωση: Παραδείγματα 1/2

- Ο τίτλος κάθε βιβλίου και ο τίτλος του μαθήματος στο οποίο διδάσκεται.

```
SELECT course.title, books.title
FROM books
LEFT JOIN course ON course_id=course_book
ORDER BY books.title;
```

Επιστρέφει:

title	title
Βάσεις Δεδομένων	Databases 1
Βάσεις Δεδομένων	Databases 1 2nd volume
Βάσεις Δεδομένων II	Databases 2

- Το όνομα του καθηγητή που διδάσκει το μάθημα με κωδικό 2 και ο τίτλος του μαθήματος.

```
SELECT pr_name, pr_lastname,course.title
FROM professor INNER JOIN course ON email=supervisor
WHERE course_id=2
ORDER BY pr_lastname, pr_name, course.title;
```

Επιστρέφει:

pr_name	pr_lastname	title
Μαρία	Παπαδοπούλου	Βάσεις Δεδομένων

Συνένωση: Παραδείγματα 2/2

- Το όνομα των καθηγητών που διδάσκουν κάποιο μάθημα και ο αριθμός των μαθημάτων τα οποία διδάσκουν.

```
SELECT pr_name, pr_lastname, count(*)
FROM professor
INNER JOIN course ON supervisor=email
GROUP BY supervisor;
```

Επιστρέφει:

pr_name	pr_lastname	count(*)
Αλεξάνδρα	unknown	1
Μαρία	Παπαδοπούλου	1

- Το όνομα **κάθε καθηγητή** και ο αριθμός των μαθημάτων τα οποία διδάσκει.

```
SELECT pr_name, pr_lastname, count(course_id)
FROM professor
LEFT JOIN course ON supervisor=email
GROUP BY supervisor;
```

Επιστρέφει:

pr_name	pr_lastname	count(course_id)
unknown	Δημητρίου	0
Αλεξάνδρα	unknown	1
Μαρία	Παπαδοπούλου	1

Aliases

- Με χρήση του keyword **as** η MySql μας επιτρέπει να δίνουμε ψευδώνυμα σε πίνακες και στήλες μέσα σε ένα query
- Τα πολύπλοκα queries γίνονται πιο ευανάγνωστα, εύχρηστα, κατανοητά

Παράδειγμα

```
SELECT p.pr_name AS 'Professor Name', p.pr_lastname AS 'Professor
Lastname', count(c.course_id) AS 'Number of Courses'
FROM professor AS p
LEFT JOIN course AS c ON c.supervisor=p.email
GROUP BY c.supervisor;
```

Επιστρέφει:

Professor Name	Professor Lastname	Number of Courses
unknown	Δημητρίου	0
Αλεξάνδρα	unknown	1
Μαρία	Παπαδοπούλου	1

Case Study: Aliases για join πίνακα με τον εαυτό του 1/3

- Η συνένωση σε αναδρομική συσχέτιση χρειάζεται aliases για να υλοποιηθεί
- Για παράδειγμα έστω ο πίνακας:

```
CREATE TABLE category(
cat_id INT NOT NULL AUTO_INCREMENT,
cat_name VARCHAR(10) NOT NULL,
cat_parent INT,
PRIMARY KEY(cat_id),
FOREIGN KEY (cat_parent) REFERENCES category(cat_id)
ON DELETE SET NULL ON UPDATE CASCADE
)ENGINE='InnoDb';
```

- και δεδομένα:

category	cat_id	cat_name	cat_parent
	1	sports	NULL
	2	football	1
	3	basketball	1
	4	art	NULL
	5	painting	4
	6	dancing	4

Case Study: Aliases για join πίνακα με τον εαυτό του 2/3

- Θέλουμε να επιστραφούν τα ονόματα των κατηγοριών και το όνομα των γονικών τους.

```
SELECT a.cat_name AS Name, b.cat_name AS Parent
FROM category AS a
INNER JOIN category AS b ON b.cat_id = a.cat_parent;
```

Επιστρέφει:

Name	Parent
football	sports
basketball	sports
painting	art
dancing	art

Case Study: Aliases για join πίνακα με τον εαυτό του 3/3

- Θέλουμε να επιστραφούν τα ονόματα **όλων των κατηγοριών** και το όνομα των γονικών τους αν υπάρχουν.

```
SELECT a.cat_name AS Name, b.cat_name AS Parent
FROM category AS a
LEFT JOIN category AS b ON b.cat_id = a.cat_parent;
```

Επιστρέφει:

Name	Parent
sports	NULL
football	sports
basketball	sports
art	NULL
painting	art
dancing	art