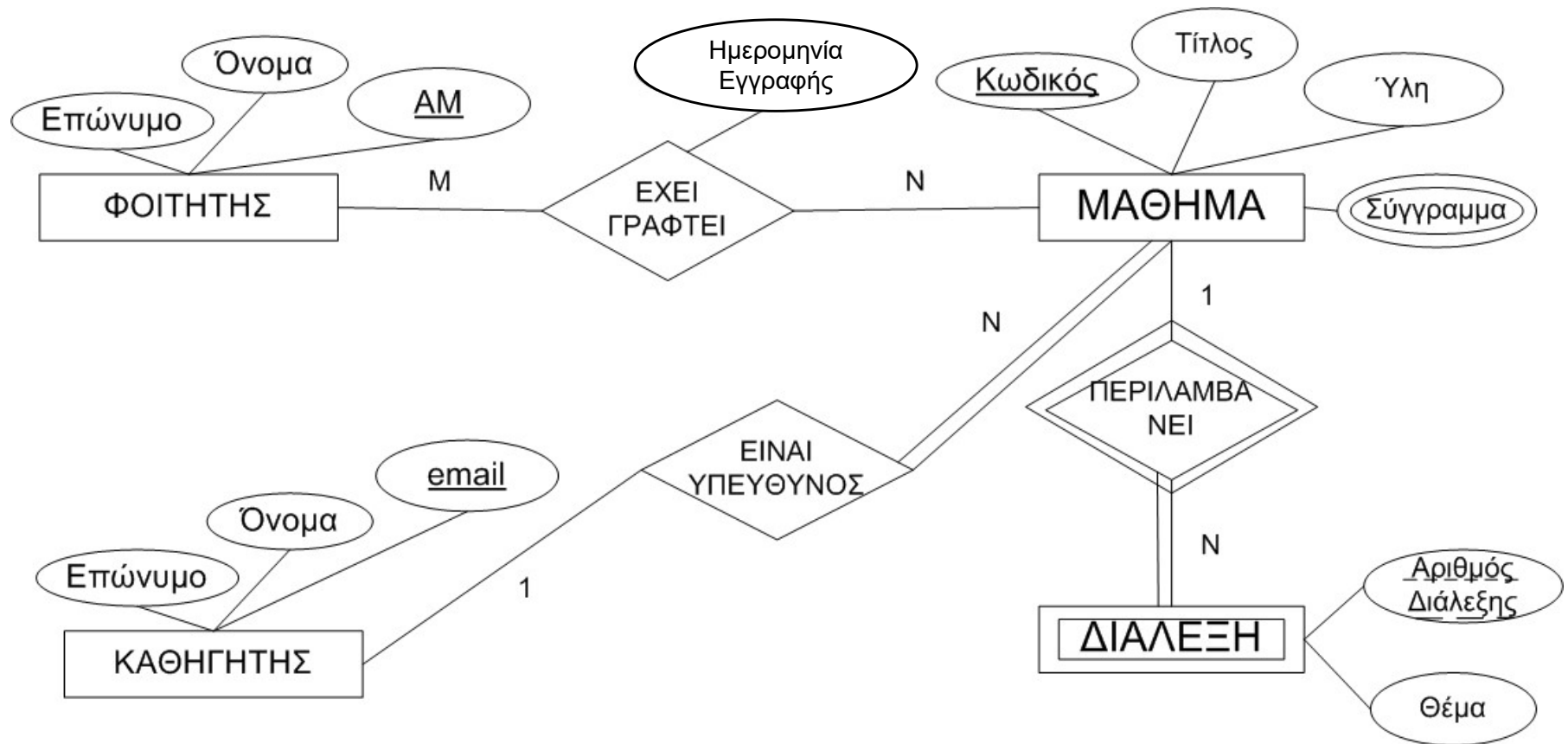


Εργαστήριο Βάσεων Δεδομένων

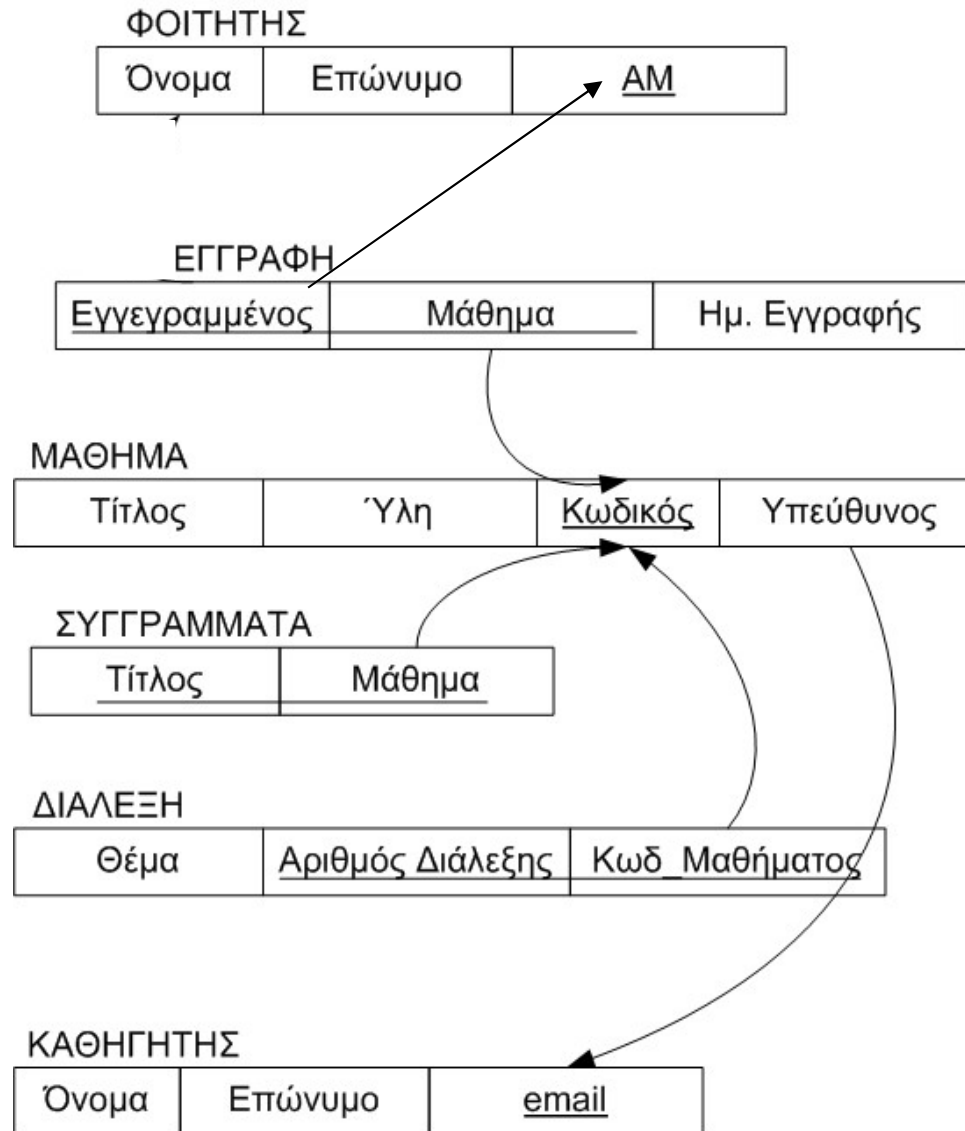
MySQL queries



Παράδειγμα - ER



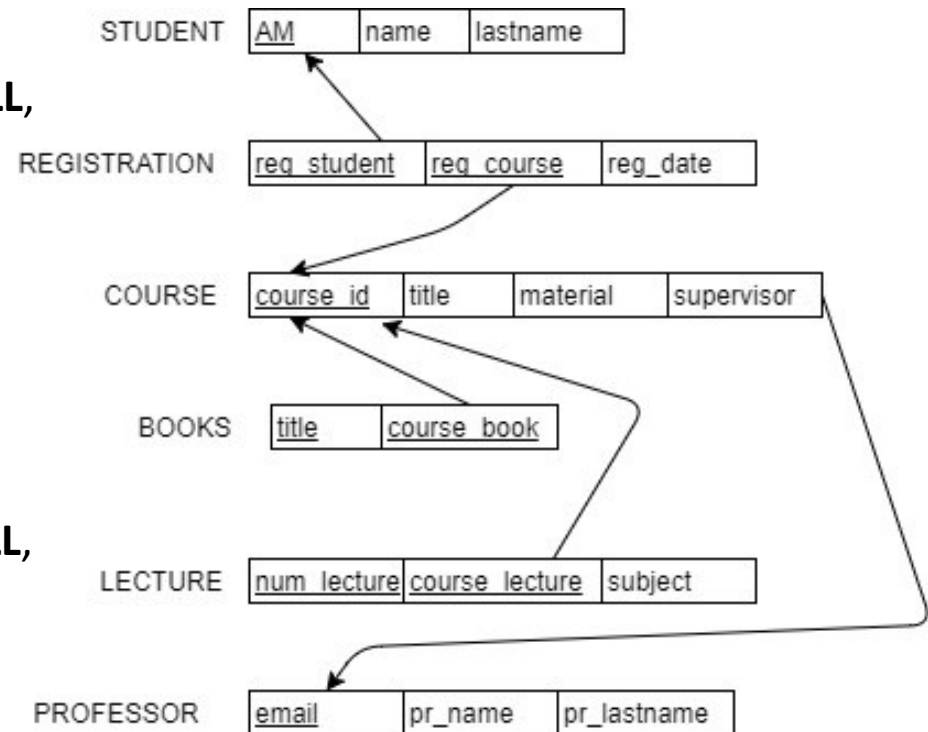
Παράδειγμα-Σχεσιακό



Παράδειγμα – Δημιουργία Πινάκων

```
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)  
);
```



```

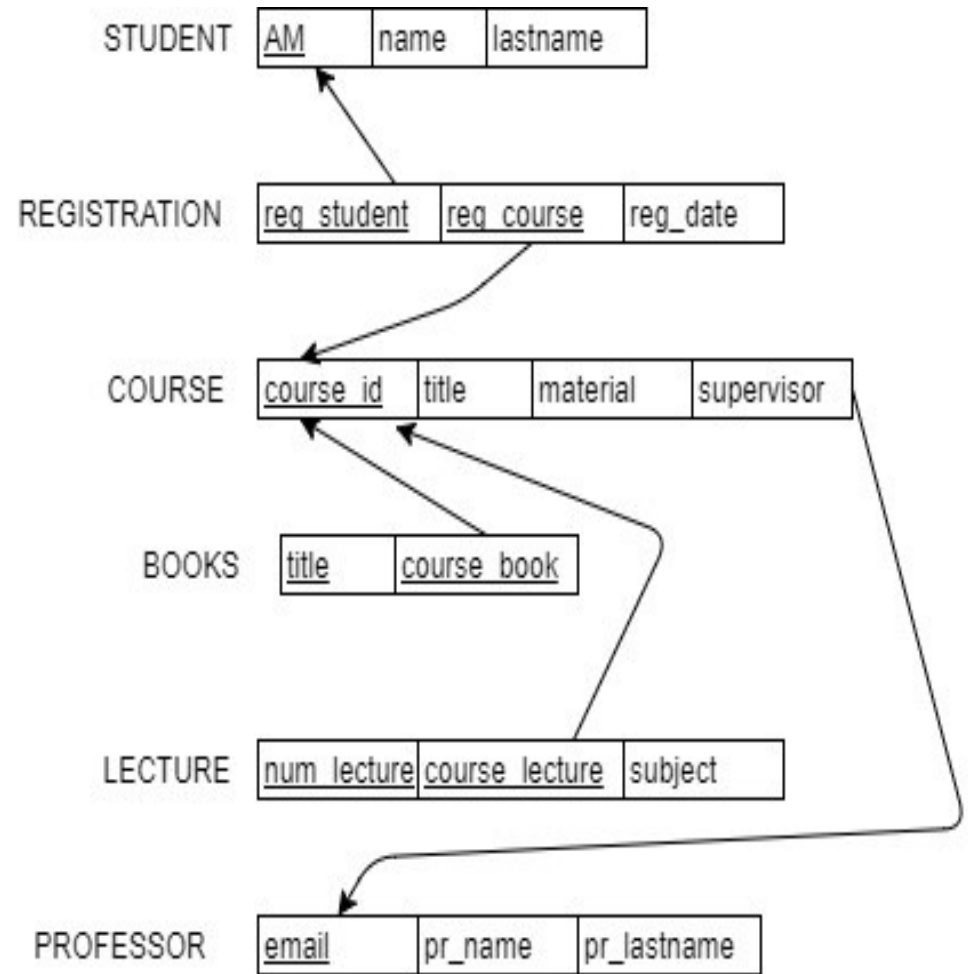
CREATE TABLE course (
  title VARCHAR(255) DEFAULT 'unknown' NOT NULL,
  material TEXT,
  course_id INT(4) NOT NULL AUTO_INCREMENT,
  supervisor VARCHAR(255) NOT NULL,
  PRIMARY KEY(course_id),
  UNIQUE(title),
  CONSTRAINT SUPERVISED
  FOREIGN KEY (supervisor)
  REFERENCES professor(email)
  ON DELETE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

```



```

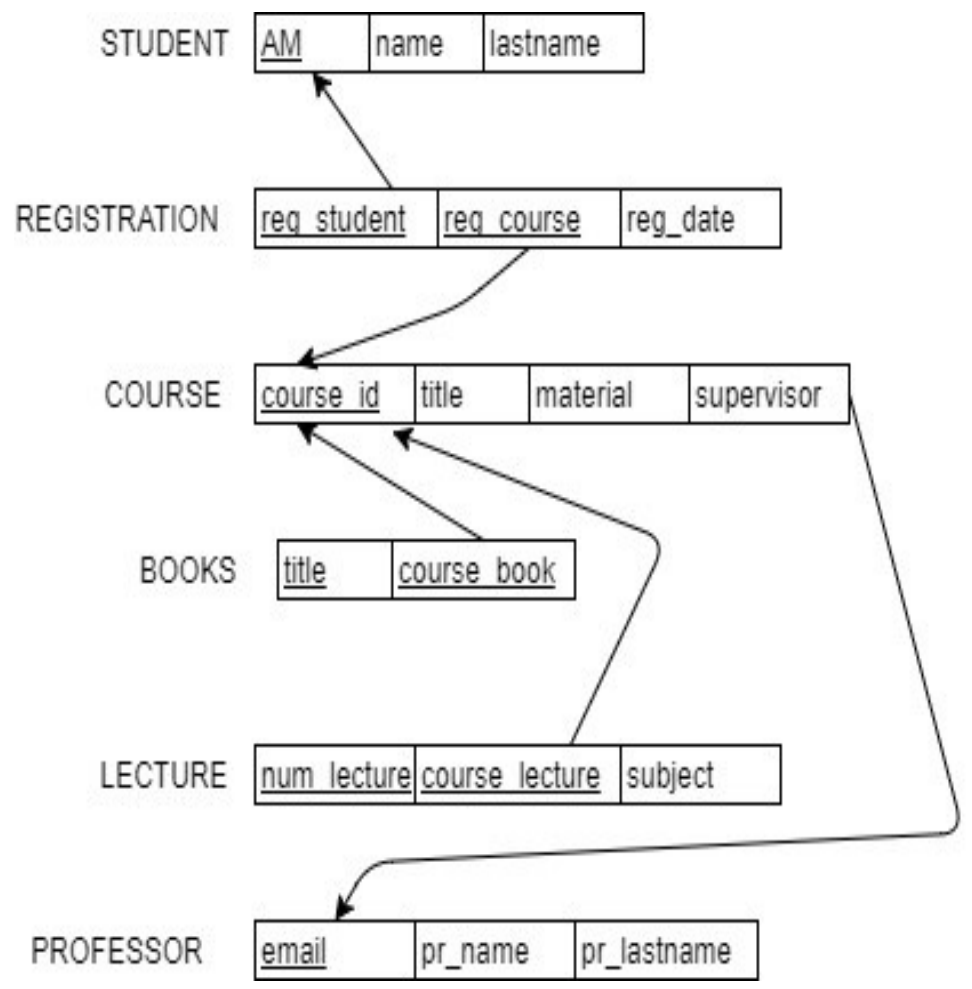
CREATE TABLE lecture (
  subject VARCHAR(128),
  num_lecture INT(2) NOT NULL,
  course_lecture INT(4) NOT NULL,
  PRIMARY KEY(num_lecture,course_lecture),
  CONSTRAINT CRSLECTURE
  FOREIGN KEY (course_lecture)
  REFERENCES course(course_id)
  ON DELETE CASCADE ON UPDATE CASCADE);

```

```

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

```



Ανάκτηση πληροφορίας από τη βάση

- Η ανάκτηση πληροφοριών από τη βάση γίνεται με την εντολή `select`.
- Απάντηση σε μία ερώτηση `select` είναι ένα σύνολο εγγραφών (γραμμών) που πληρούν τις προϋποθέσεις (ένας **προσωρινός πίνακας**).
- Για το σχηματισμό του πίνακα η `select` καθορίζει:
 - Ποιες στήλες θα περιέχει η απάντηση
 - Από ποιόν πίνακα θα πάρει αυτές τις στήλες
 - Ποιες γραμμές θα περιέχει η απάντηση (συνθήκη επιλογής)
- Η βασική σύνταξη της `select` είναι:
SELECT <λίστα στηλών>
FROM <λίστα πινάκων>
WHERE <συνθήκη επιλογής>

Select: γενική σύνταξη

- **SELECT** <Λίστα στηλών>:
 - Ονόματα στηλών χωρισμένα με κόμμα
 - Μόνο αυτές οι στήλες θα υπάρχουν στην απάντηση
 - Αν τις θέλουμε όλες χρησιμοποιούμε το *
- **FROM** <Λίστα πινάκων>:
 - Ονόματα των πινάκων που θα συμμετέχουν στην επιλογή δεδομένων
 - Αν είναι περισσότεροι από ένας, υπάρχουν εναλλακτικοί τρόποι συνδυασμού τους (join)

Παραδείγματα:

```
SELECT name, lastname
```

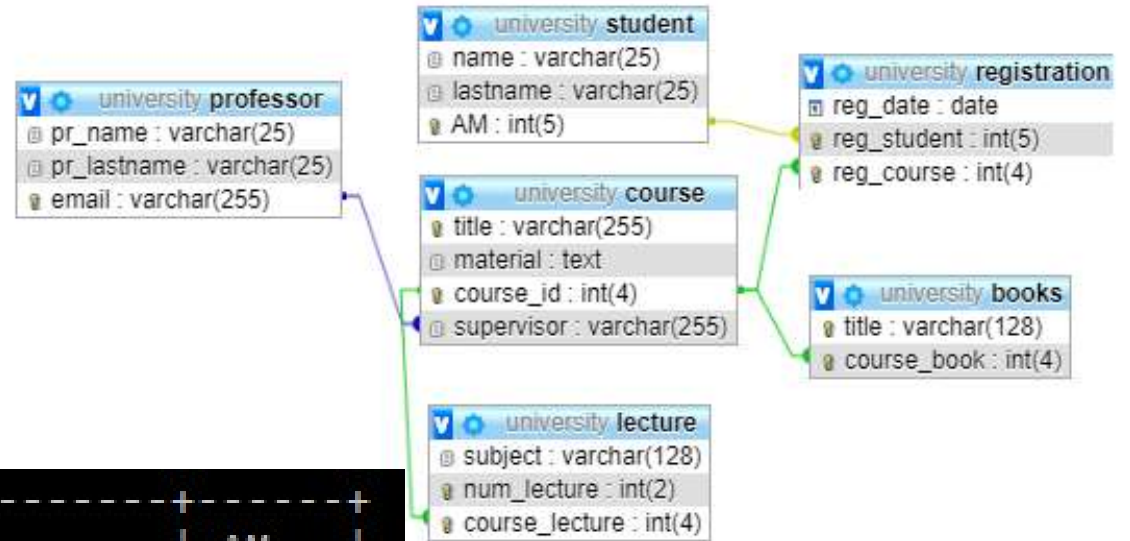
```
FROM student;
```

```
SELECT *
```

```
FROM professor;
```


Select name, lastname, AM
From student;

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



Select * from professor;

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

Select: συνθήκη WHERE

- Λογική συνθήκη μεταξύ των στηλών του πίνακα και αριθμών
- Στην απάντηση μόνο οι γραμμές που την ικανοποιούν
- Χρησιμοποιούνται οι τελεστές:
 - Boolean: AND, OR, NOT
 - Σύγκριση βαθμωτών: =, <>, >, <, >=, <=
 - Σύγκριση με το NULL: IS NULL, IS NOT NULL
 - Σύγκριση αλφαριθμητικών: LIKE (με wildcards)
 - % μηδέν ή περισσότεροι χαρακτήρες
 - _ ακριβώς ένας χαρακτήρας

Select - παραδείγματα

- Επίλεξε τα ονόματα των φοιτητών με AM μεγαλύτερο του 2000

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

- Επίλεξε τα μαθήματα που έχουν τη λέξη 'Εισαγωγή' στην περιγραφή τους

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

- Επίλεξε τα ids των μαθημάτων που έχει εγγραφεί ο φοιτητής με AM 2191

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

- Επίλεξε τους τίτλους μαθημάτων που διδάσκονται από την Παπαδοπούλου (pap@ceid.upatras.gr) και αφορούν στις βάσεις δεδομένων

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

Select - παραδείγματα

- Επίλεξε τα στοιχεία των φοιτητών με επώνυμο που αρχίζει από 'Τ'

```
SELECT * FROM student WHERE lastname LIKE 'T%';
```

- Επίλεξε τα μαθήματα που δεν έχουν περιγραφή ύλης

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

- Επίλεξε τα μαθήματα στα οποία έχει γίνει εγγραφή από το 2012 και μετά

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

- Επίλεξε τους φοιτητές 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 * from Student;

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

Select name, lastname

From student

Where AM>2000;

name	lastname
Βιβή	Τζέκου
unknown	Ντούρου
Αθανασία	Κουμπούρη

```
SELECT * FROM `student` WHERE name like "%B%";
```

name	lastname	AM
Βάσω	unknown	1845
Βιβή	Τζέκου	2191

```
SELECT * FROM `student` WHERE name like "_άσ%";
```

name	lastname	AM
Βάσω	unknown	1845

1 ακριβώς χαρακτήρας και μετά άσ%

SELECT * FROM student WHERE name like "%__σ%";

name	lastname	AM
Βάσω	unknown	1845
Αθανασία	Κουμπούρη	2193

↑
2 ακριβώς
χαρακτήρες
και μετά σ%

SELECT * FROM student WHERE name like "%___σ%";

name	lastname	AM
Αθανασία	Κουμπούρη	2193

↑
3 ακριβώς
χαρακτήρες
και μετά σ%

SELECT * FROM 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

SELECT * FROM course WHERE material like '%σαγ%';

title	material	course_id	supervisor
Βάσεις Δεδομένων	Εισαγωγή σε σχεσιακές βάσεις δεδομένων	2	pap@ceid.upatras.gr

**SELECT title FROM course
WHERE supervisor like '%ceid%' and title like '%σεις%';**

title
Βάσεις Δεδομένων
Βάσεις Δεδομένων II

Σημείωση για τα παραδείγματα που ακολουθούν:

το περιεχόμενο του πίνακα course το αλλάξαμε με διαδοχικά Update όπως πχ

```
UPDATE `course` SET `material` = 'Book Big Data'  
WHERE `course`.`course_id` = 4;
```

Το τρέχον περιεχόμενο το βρίσκω με :

```
Select * from course;
```

title	material	course_id	supervisor
Baseis dedomenwn	Book and presentations	1	vasi@ceid
erg Baseis dedomenwn	Presentations and Lab exams	2	ngou@ceid
tpy	Book Notes presentations	3	evoyia@ceid
baseis dedomwnwn II	Book Big Data	4	vasi@ceid

Πλήρης μορφή SELECT

Select a1, a2, .., an

From r1, r2, .., rm

Where P

[**group by**]

[**having**...]

[**order by**]

[**limit**]

Select – Order By

- Ταξινόμηση σύμφωνα με ένα ή περισσότερα πεδία
- Για περισσότερα πεδία, ισχύει προτεραιότητα από αριστερά
- Οι τελεστές ASC και DESC ορίζουν αύξουσα και φθίνουσα σειρά αντίστοιχα

Εμφάνισε τα στοιχεία των φοιτητών σε αύξουσα σειρά ως προς το επώνυμό τους

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

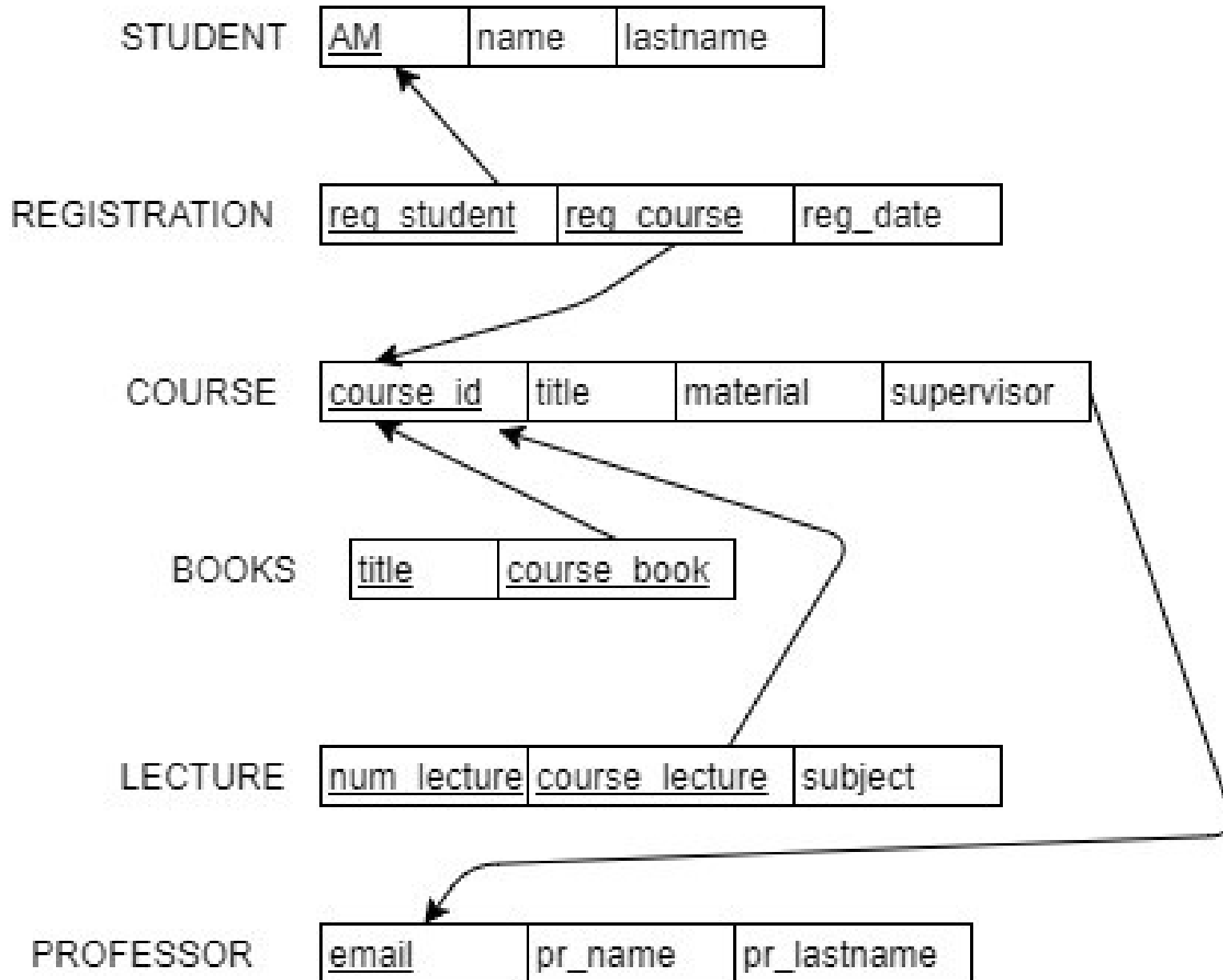
Εμφάνισε τις εγγραφές του 2012 από την πιο πρόσφατη προς την παλιότερη

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

Επέστρεψε τα μαθήματα ταξινομημένα ανά email Καθηγητή και δευτερευόντως ανά τίτλο

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

Να θυμηθούμε τη βάση



Select * from 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

Ταξινόμηση με βάση το supervisor

select * from course

order by supervisor; default παίρνει το ASC

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

select * from course

order by supervisor DESC;

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

```
Select *  
from  
registration;
```

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

```
SELECT *  
FROM `registration`  
WHERE reg_date >  
'2011-10-01';
```

reg_date	reg_student	reg_course
2012-06-03	2192	3

```
SELECT *  
FROM registration  
order by reg_date;
```

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

Ποιο θα είναι το αποτέλεσμα αν το order by γίνει reg_date, reg_student;

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 0).
 - Το **n** είναι το πλήθος των εγγραφών που θα επιστραφούν.
 - Συνήθως η LIMIT είναι χρήσιμη μαζί με ταξινόμηση εγγραφών.

Εμφάνισε το φοιτητή με το μεγαλύτερο AM

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

Εμφάνισε τις τρεις πρώτες εγγραφές από το 2012 μέχρι σήμερα

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

Select * from registration

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

SELECT * FROM registration LIMIT 3;

reg_date	reg_student	reg_course
2010-08-09	1845	2
2011-02-11	2191	2
2011-04-19	2191	3

SELECT * FROM `registration` order by reg_date, reg_student Limit 3;

reg_date	reg_student	reg_course
2010-01-12	2193	3
2010-08-09	1845	2
2011-02-11	2191	2

Select - Group By

- Ομαδοποιεί τις εγγραφές σύμφωνα με ένα πεδίο ή πολλά πεδία
- Για τα κριτήρια ομαδοποίησης δημιουργούνται ομάδες με κοινές τιμές στα πεδία που ορίζει η GROUP BY
- Αν είναι πολλά τα πεδία ομαδοποίησης οι ομάδες σχηματίζονται από εγγραφές με κοινές τιμές σε όλα τα πεδία

Εμφάνισε για κάθε φοιτητή το ΑΜ του και τον αριθμό των εγγραφών που έχει πραγματοποιήσει

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

reg_student	count(*)
1845	1
2191	2
2192	1
2193	1

Select * from registration;

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

```
SELECT reg_student, count(*)  
FROM `registration`  
group by reg_student;
```

reg_student	count(*)
1845	1
2191	2
2192	1
2193	1

```
SELECT reg_student, count(reg_date)  
FROM `registration`  
group by reg_student;
```

reg_student	count(reg_date)
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 (group)
 - SUM: άθροισμα των τιμών του πεδίου που δέχεται σαν όρισμα
 - COUNT: μέτρηση πλήθους
 - Για να μετρήσουμε πλήθος μοναδικών τιμών χρησιμοποιούμε το DISTINCT. Σε ένα SELECT η χρήση του DISTINCT επιστρέφει μόνο πλειάδες με μοναδικές τιμές.
 - MAX: μέγιστο
 - MIN: ελάχιστο
 - AVG: μέσος όρος της τιμή του πεδίου που δέχεται σαν όρισμα

Select - Having

- Συνθήκη που εφαρμόζεται στα αποτελέσματα της group by (στις ομάδες)
- **Προσοχή:** Αν η συνθήκη αφορά στα αρχικά δεδομένα τότε πρέπει να τοποθετηθεί στο where
- **Having** : λειτουργεί όπως το **Where** **ΑΛΛΑ** για ομάδες *(όχι για συγκεκριμένες πλειάδες)*

Εμφάνισε τους φοιτητές με περισσότερες από 1 εγγραφές

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

reg_student	count(*)
2191	2

Εμφάνισε τους φοιτητές και τον αριθμό των εγγραφών τους λαμβάνοντας υπόψη μόνο τις εγγραφές από 2012 και μετά

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

Πίνακας *Registration*

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

```
SELECT reg_student, count(*)  
FROM registration  
WHERE reg_date >= '2010-10-01'  
Group by reg_student;
```

reg_student	count(*)
2191	2
2192	1

Αυτό δεν υλοποιείται με having διότι το αποτέλεσμα της

```
SELECT reg_student, count(*)  
FROM registration  
Group by reg_student
```

Δεν έχει στήλη reg_date

reg_student	count(*)
1845	1
2191	2
2192	1
2193	1

Πίνακας *Registration*

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

```
SELECT reg_student, count( reg_course)
FROM `registration`
group by reg_student;
```



Μέτρησε τα courses
για κάθε φοιτητή

reg_student	count(reg_course)
1845	1
2191	2
2192	1
2193	1

```
SELECT reg_student, count(reg_course)
FROM `registration`
group by reg_student
having count(reg_course)>1;
```



Από τα παραπάνω
επέλεξε αυτά που είναι
περισσότερα από 1

reg_student	count(reg_course)
2191	2

Select – Group By

- Εμφάνισε τον αριθμό των μαθημάτων που διδάσκει κάθε καθηγητής

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

- Εμφάνισε για κάθε μάθημα τον αριθμό των διαλέξεων που έχουν θέμα σχετικό με βάσεις (=περιέχουν στο θέμα τη λέξη)

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

- Εμφάνισε τα ids των μαθημάτων και το μέγιστο αύξοντα αριθμό των διαλέξεων ανά μάθημα, 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 ;
```

Select *

from lecture;

subject	num_lecture	course_lecture
Εισαγωγή σε βάσεις	1	2
Κανονικοποίηση βάσεων	1	3
Ανάλυση απαιτήσεων	2	2
Βελτιστοποίηση	2	3
ER-Σχεσιακό	3	2

SELECT course_lecture,

max(num_lecture)

from lecture

group by course_lecture;



Βρες το μέγιστο num_lecture

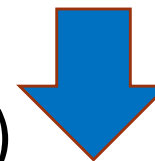
course_lecture	max(num_lecture)
2	3
3	2

SELECT course_lecture, max(num_lecture)

from lecture

group by course_lecture

having max(num_lecture)>2;



Επέλεξε αυτά που το μέγιστο είναι μεγαλύτερο από 2

course_lecture	max(num_lecture)
2	3

Select – Group By

- Για σωστή σύνταξη select με group by, προσέχουμε

1. Στη λίστα στηλών στο select χρησιμοποιούμε πεδία

- που συμμετέχουν στο group by
- με τα οποία υπάρχει συναρτησιακή εξάρτηση με πεδία που υπάρχουν στο group by

2. Πεδία

- που δεν συμμετέχουν στο group by
- δεν εξαρτώνται συναρτησιακά από τα πεδία που υπάρχουν στο group by

μπορούν να χρησιμοποιηθούν στη λίστα στηλών στο select μόνο ως ορίσματα μιας ενσωματωμένης συνάρτησης, π.χ. count .

Select – Group By

- Σωστή σύνταξη select με group by

1. SELECT course_lecture FROM lecture group by course_lecture;

course_lecture
2
3

2. SELECT course_lecture, sum(num_lecture) FROM lecture group by course_lecture;

course_lecture	sum(num_lecture)
2	6
3	3

- Παράδειγμα με λάθος σύνταξη select με group by;

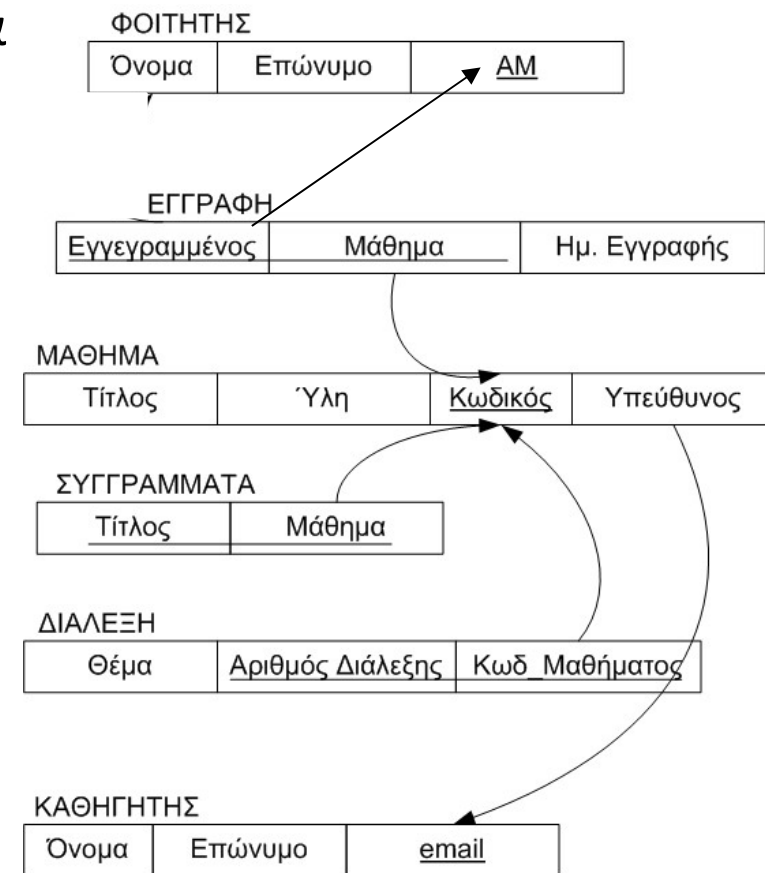
Δεδομένα από πολλούς πίνακες: ΠΟΤΕ;

- Όταν χρειαζόμαστε να συνδυάσουμε δεδομένα από δύο ή παραπάνω 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 δε δηλωθεί τότε έχουμε το καρτεσιανό γινόμενο των πινάκων

Παράδειγμα

Έστω ότι ψάχνουμε τα ονόματα των φοιτητών που έχουν πραγματοποιήσει εγγραφή από το 2012 και μετά

- Ποιοί πίνακες περιέχουν τη ζητούμενη πληροφορία;



Δεδομένα από πολλούς πίνακες:

1. Εμφωλευμένες select

- Η εκτέλεση του ερωτήματος ξεκινά από την πιο εσωτερική select και επιστρέφει τα αποτελέσματά της στην αμέσως εξωτερική της, κ.ο.κ.

Εμφάνισε τα ονόματα και τα επίθετα των φοιτητών που έκαναν εγγραφή από το 2012 και μετά

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

Επιστρέφει:

name	lastname
unknown	Ντούρου

Δεδομένα από πολλούς πίνακες:

2. JOIN

- JOIN: τρόπος δήλωσης της συνένωσης πινάκων για το συνδυασμό δεδομένων, κυρίως όταν έχουμε συσχέτιση

Εμφάνισε τον αριθμό της κάθε διάλεξης, το θέμα της και τον τίτλο του μαθήματος στο οποίο ανήκει (ταξινομημένες κατά...)

```
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;
```

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	Βάσεις Δεδομένων II

- INNER JOIN: μια εγγραφή συμπεριλαμβάνεται μόνο αν υπάρχει εγγραφή που της 'ταιριάζει' από τον άλλο πίνακα.
- LEFT JOIN: συμπεριλαμβάνονται όλες οι εγγραφές του αριστερού πίνακα, ανεξάρτητα από το αν υπάρχει εγγραφή στο δεξιό πίνακα που να τους ταιριάζει
- RIGHT JOIN: Ομοίως για τον δεξιό πίνακα

Δεδομένα από πολλούς πίνακες:

3. Παράθεση πινάκων

- Παραθέτουμε στη λίστα πινάκων όλους όσους χρειαζόμαστε
- Για να μη γίνει καρτεσιανό γινόμενο, στη WHERE ορίζουμε τη συνθήκη συνένωσης

Εμφάνισε τον αριθμό της κάθε διάλεξης, το θέμα της και τον τίτλο του μαθήματος στο οποίο ανήκει (ταξινομημένες κατά...)

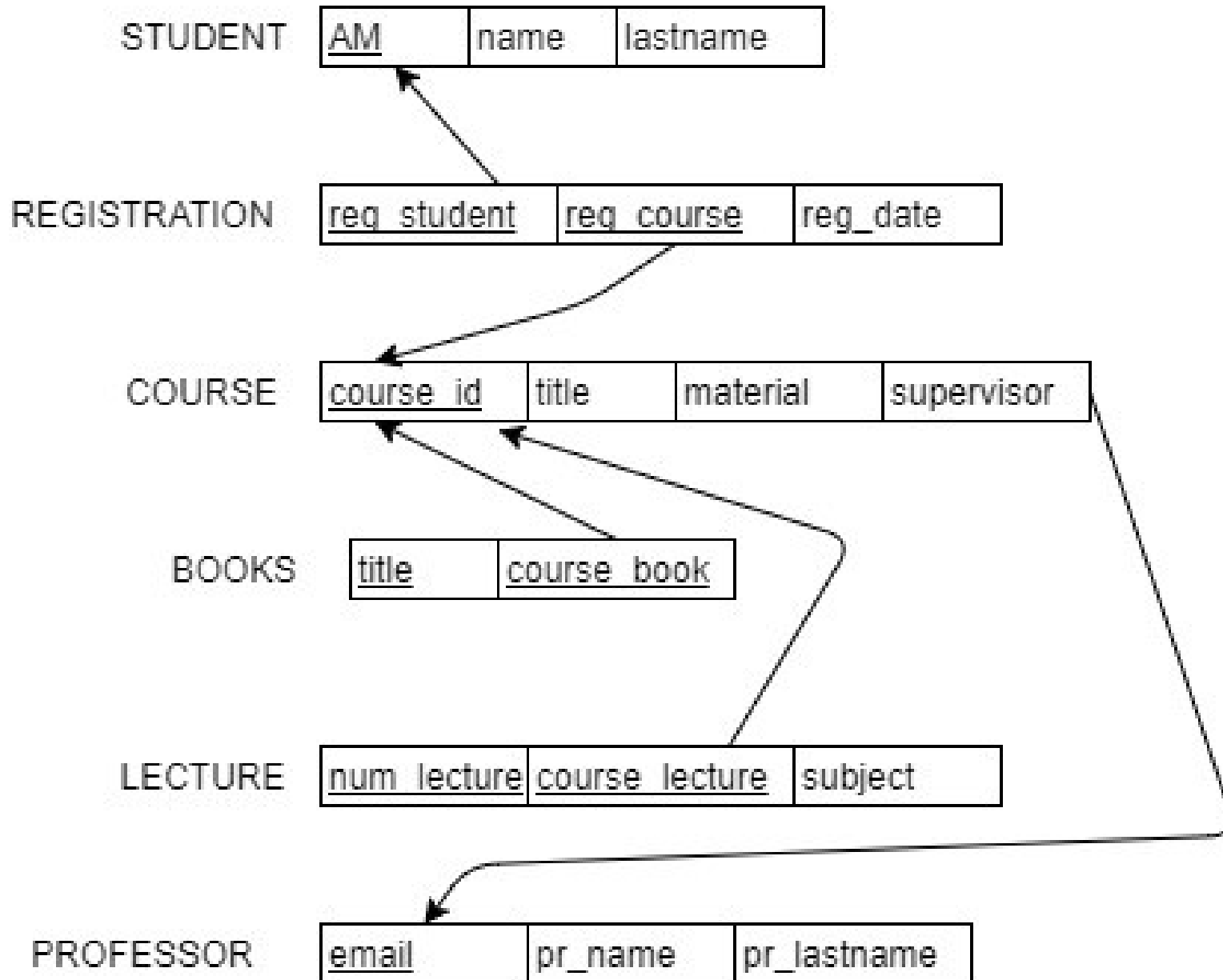
```
SELECT lecture.num_lecture, lecture.subject, course.title
FROM lecture, course
WHERE course.course_id=lecture.course_lecture
ORDER BY course.course_id ASC, lecture.num_lecture ASC;
```

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	Βάσεις Δεδομένων II

Ερώτηση σε M:N

Εμφάνισε το ονοματεπώνυμο των φοιτητών και το όνομά του μαθήματος στα οποία έχουν εγγραφεί για κάθε εγγραφή που έχουν κάνει

Να θυμηθούμε τη βάση



Ερώτηση σε M:N

Εμφάνισε το ονοματεπώνυμο των φοιτητών και το όνομά 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;
```

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

Παραδείγματα σε JOIN

- Εμφάνισε τον τίτλο κάθε βιβλίου και τον τίτλο του μαθήματος στο οποίο διδάσκεται

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

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

- Εμφάνισε το όνομα του καθηγητή που διδάσκει το μάθημα με κωδικό 2 και τον τίτλο του μαθήματος

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

Παραδείγματα σε JOIN

- Εμφάνισε το όνομα των καθηγητών που διδάσκουν κάποιο μάθημα και τον αριθμό των μαθημάτων που διδάσκουν

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

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

- Το όνομα του κάθε καθηγητή και τον αριθμό των μαθημάτων που διδάσκει

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

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

Aliases

- Με χρήση του keyword **as** η MySQL μπορούμε να δίνουμε προσωρινά ονόματα σε πίνακες και στήλες μόνο για την εκτέλεση του query
- Τα πολύπλοκα queries γίνονται πιο ευανάγνωστα, εύχρηστα και 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;
```

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

Aliases για join πίνακα με τον εαυτό του (1/3)

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

```
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

Aliases για join πίνακα με τον εαυτό του (2/3)

- Θέλουμε να εμφανιστούν τα ονόματα των κατηγοριών που έχουν γονικές κατηγορίες και τα ονόματα 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

Aliases για join πίνακα με τον εαυτό του (3/3)

- Θέλουμε να εμφανιστούν τα ονόματα όλων των κατηγοριών και τα ονόματα των γονικών τους

```
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;
```

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

Προσεχώς

- Σήμερα θα αναρτηθεί στο eclass το 3^ο σετ εξάσκησης (οι λύσεις θα ανακοινωθούν σύντομα)
- 3^η Εξέταση στο Υπολογιστικό Κέντρο : Δευτέρα 18/11
 - Θα σας δωθεί μια έτοιμη ΒΔ και το σχεσιακό της και θα ζητηθεί να συντάξετε ερωτήματα (select)
 - **60 λεπτά** καθαρός χρόνος εξέτασης -1^η Ομάδα ξεκινά 10.15
 - Πρέπει να θυμάστε τα στοιχεία πρόσβασης στο CEID webmail και στο eclass
- Επόμενο μάθημα στην Αίθουσα Γ: Δευτέρα 25/11
- Έχει αναρτηθεί η Προπαρασκευαστική Φάση του Πρότζεκτ
 - Σας δίνετε το σχεσιακό μοντέλο της ΒΔ του Πρότζεκτ
 - Πρέπει να δημιουργήσετε τους αντίστοιχους πίνακες με εντολές CREATE. **Χρησιμοποιήστε τις ονομασίες της εκφώνησης.**
 - Πρέπει να εισαγάγετε δικές σας τιμές στους παραπάνω πίνακες με εντολές INSERT.
 - **ΠΡΟΣΟΧΗ:** Θα το χρειαστείτε για την 5^η Εξέταση στα Triggers!!!!

Βάση Δεδομένων για τα παραδείγματα

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

Βάση Δεδομένων για τα παραδείγματα

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

Ευχαριστώ για την προσοχή σας!

- Απορίες;
- Διευκρινήσεις;