

## Βάση Δεδομένων για το τμήμα Λαμίας

- Student (StAM, name, surname, semester)
- Phones (StAM, phonenum)
- Professor (profID, name, surname)
- Course (courseID, title, profID)
- Grades (StAM, courseID, grade)
- Requires (courseID1, courseID2)

1. Αναζητήστε τους σπουδαστές του 'Ε' Εξαμήνου.
2. Βρείτε τα IDs των σπουδαστών με κινητά τηλέφωνα.
3. Αναζητήστε τα IDs των σπουδαστών που έχουν περάσει τουλάχιστον ένα μάθημα.
4. Αναζητήστε τα IDs όλων των αποτυχόντων που έγραψαν πάνω από 3.
5. Αναζητήστε τους καθηγητές των οποίων τα επώνυμα αρχίζουν από 'L' και τελειώνουν σε 'os'.
6. Αναζητήστε τους καθηγητές των οποίων τα επώνυμα αρχίζουν από 'L' αλλά δεν τελειώνουν σε 'os'.

7. Αναζητήστε τα ονόματα των σπουδαστών που κατέχουν κινητά τηλέφωνα.
8. Αναζητήστε τους φοιτητές που έδωσαν το μάθημα με κωδικό '5'.
9. Εμφανίστε το σύνολο των φοιτητών που πέρασαν.
10. Εμφανίστε τους φοιτητές με ID<3345 που κόπηκαν στο μάθημα με κωδικό '10'.
11. Να εμφανιστούν οι κωδικοί των προαπαιτούμενων του μαθήματος 'vaseis dedomenwn'.

12. Να εμφανιστούν τα τηλέφωνα και τα ονόματα των επιτυχόντων στο μάθημα με κωδικό '5'.
13. Να εμφανιστούν οι βαθμοί που έγραψαν οι φοιτητές στα μαθήματα του διδάσκοντα 'Karanikas', μαζί με τους αντίστοιχους τίτλους μαθημάτων.
14. Να εμφανιστούν τα IDs των φοιτητών που κόπηκαν σε κάποιο από τα προαπαιτούμενα των μαθημάτων 'vaseis dedomenwn' και 'maths1'.

### Εξωτερικές Συνενώσεις – Outer Joins

15. Εμφανίστε όλους τους φοιτητές μαζί με τα τηλέφωνα τους.
16. Εμφανίστε όλα τα μαθήματα μαζί με τους κωδικούς τυχόν προαπαιτούμενων που έχουν.
17. Εμφανίστε τις αναλυτικές βαθμολογίες όλων των φοιτητών δηλ., όλους τους φοιτητές και για κάθε φοιτητή όλους τους τίτλους των μαθημάτων μαζί με τους βαθμούς που έγραψε.

## Υπενθυμίσεις από θεωρία

Η βασική εντολή της SQL με την οποία θέτουμε ερωτήματα είναι η:

```
SELECT <πεδίο1, πεδίο2, ...>  
FROM <πίνακας1, πίνακας2, ...>  
WHERE <εκφράσεις-συνθήκες>
```

Υπάρχουν δύο τρόποι για να συνδυάσουμε πίνακες. Ο πρώτος είναι χρησιμοποιώντας καρτεσιανό γινόμενο και ο δεύτερος είναι με την πράξη της συνένωσης (JOIN). Αν η συνένωση γίνεται σε όλα τα κοινά πεδία των πινάκων (απαιτώντας να είναι οι αντίστοιχες τιμές ίσες), τότε έχουμε την πράξη της φυσικής ισοσυνένωσης (NATURAL JOIN).

Οι συνενώσεις που είδαμε ως τώρα (NATURAL JOIN, JOIN...ON..) είναι εσωτερικές συνενώσεις (INNER JOIN). Στις εσωτερικές συνενώσεις χάνεται πληροφορία. Για παράδειγμα στο ακόλουθο ερώτημα:

```
SELECT name, surname  
FROM student NATURAL JOIN takes;
```

δεν εμφανίζονται οι φοιτητές που δεν έχουν δώσει τουλάχιστον ένα μάθημα. Η λύση αν θέλαμε να εμφανίσουμε όλους τους φοιτητές μαζί με τα μαθήματα που έχουν δώσει (συμπεριλαμβανομένων αυτών που δεν έχουν δώσει κανένα μάθημα) είναι να χρησιμοποιήσουμε εξωτερικές συνενώσεις. Τα είδη των εξωτερικών συνενώσεων είναι τα: LEFT OUTER JOIN...ON... (ή μόνο LEFT JOIN), RIGHT OUTER JOIN και FULL OUTER JOIN (δεν υποστηρίζεται στην τρέχουσα έκδοση της mysql). Αν πρόκειται για εξωτερικές *φυσικές* συνενώσεις (δηλ. σε όλα τα κοινά γνωρίσματα), τότε γράφουμε: NATURAL LEFT JOIN, NATURAL RIGHT JOIN κλπ.

## Δημιουργία πινάκων

```
create table Student(StAM int(20) not null,  
name varchar(20) not null,  
surname varchar(20) not null,  
semester ENUM('A','B','C','D','E','ST','PT') DEFAULT 'A',  
primary key(StAM));
```

```
create table Phones(StAM int(20) not null,  
phonenumber varchar(20) not null,  
primary key(StAM,phonenumber),  
foreign key(StAM) references student(StAM));
```

```
create table professor(profID int(20) not null,  
name varchar(20) not null,  
surname varchar(20) not null,  
primary key(profID));
```

```
create table Course(courseID int(20) not null,  
title varchar(20) not null,  
profID int(20) not null,  
primary key(courseID),  
foreign key(profID) references professor(profID));
```

```
create table Grades(StAM int(20) not null,  
courseID int(20) not null,  
grade int(20) not null,  
primary key(StAM, courseID),  
foreign key(StAM) references student(StAM),  
foreign key(courseID) references Course(courseID));
```

```
create table Requires(  
courseID1 int(20) not null,  
courseID2 int(20) not null,  
foreign key(courseID1) references Course(courseID),  
foreign key(courseID2) references Course(courseID));
```

## Εισαγωγή δεδομένων

```
insert into Student values(2341,'maria','gewgriou','B'),
(2437,'tasos','trikka','E'),
(3257,'kostas','antwniou','D'),
(3345,'nikos','makos','B'),
(2654,'alex','rizou','A'),
(3451,'george','kouletas','E'),
(3392,'marina','swthriou','ST'),
(1267,'vasilis','dhmou','ST'),
(3375,'panos','lamprou','PT'),
(3764,'stauros','alexiou','A');
```

```
insert into Phones values (2341,6978667892),
(2437,6945287959),
(3257,6932211703),
(3345,6985462314),
(2654,6978952146),
(3451,6975736251),
(3392,'-'),
(1267,'-'),
(3375,6972001136),
(3764,6946057154);
```

```
insert into Professor values (792, 'Haralampos', 'Karanikas'),
(654, 'Vaggelis', 'Spyrou'),
(872, 'Arhs', 'Keriniwths'),
(531, 'Dimitris', 'Lakovidis'),
(237, 'Georgios', 'Laknakos');
```

```
insert into Course values (8,'maths1',872),
(4,'grammikh algevra',872),
(2,'psyfiakal',654),
(1,'grammika systhmata',792),
(0,'automata',531),
(3,'java2',531),
(5,'e.ereuna',654),
(10,'vaseis dedomenwn',792),
(7,'physics2',237),
(14,'domes dedomenwn',654);
```

```
insert into Grades values (3375,5,8),
(3764,2,10),
(1267,14,0),
(3392,5,1),
(2654,10,2),
(3451,5,5),
```

(3345,7,10),  
(3257,5,7),  
(2437,4,10),  
(2341,2,3);

insert into Requires values (8,4),  
(10,14),  
(4,1);

## Ερωτήματα στην βάση

1. `select * from Student where semester ='E';`
2. `select StAM from Phones where phonenumber like '69%';`
3. `select StAM from grades where grade >=5;`
4. `select StAMfrom grades where grade<5 and grade >3;`
5. `select name, Surname from Professor where surname LIKE 'L%os';`
6. `select name from Professor where surname LIKE 'L%' and surname NOT LIKE '%os';`
7. `select student.name as Stud_name, Phones.phonenumber as Phone_num from Student,Phones where Student.StAM=Phones.StAM and Phones.phonenumber Like '69%';`
8. `select Student.name as Stud_name from Student Natural Join Grades where Grades.courseID='5';`
9. `select Student.name as Stud_name,Grades.grade as Grades_grade from Student Natural Join Grades where grade>=5;`
10. `select Student.name as Stud_name,Student.StAM as Student_ID, Grades.courseID as Course_ID,Grades.grade as Grades_grade from Student Natural Join Grades where Student.Stam <3345 and Grades.grade<5 and Grades.CourseID='10';`
11. `select Requires.courseID2 as Rec_ID,Course.title as Course_Title from Requires,Course where Requires.courseID1=Course.courseID and title='vaseis dedomenwn';`
12. `select Student.name, Phones.phonenumber, Grades.grade,Course.courseID from Student Natural Join Phones Natural Join Grades Natural Join Course where Grades.grade>=5 and Course.courseID='5';`
13. `select Grades.grade, Professor.name, Course.title from Grades Natural Join Course Natural Join Professor where Professor.surname='Karanikas';`
14. `SELECT Grades.StAM FROM grades, course, requires WHERE course.courseID=requires.courseID1 AND (title='vaseis dedomenwn' or title='maths1') AND Grades.courseID=requires.courseID2 AND grades.grade<5;`
15. `select student.name,student.surname, phones.phonenumber from student left outer join phones on student.StAM=phones.StAM;`
16. `select Course.title,requires.courseID2 from Course left outer join requires on Course.courseID=requires.courseID1;`
17. `select student.name,student.surname,Course.title,grades.grade from grades right outer join student on student.StAM=grades.StAM left outer join Course on Course.courseID=grades.courseID;`