

Σύνολα Ασκήσεων

2^ο Σύνολο Ασκήσεων

Άσκηση 3

Οι λύσεις είναι ενδεικτικές – υπάρχουν και άλλες σωστές SQL ερωτήσεις για τα ερωτήματα της άσκησης.

Άσκηση 3 (SQL)

```
course(course_id, courseLevel)
student (s_id, inPhase, yearsinProgram)
professor(p_id, hasPosition)
taughtby(course_id, u_id)
advisedby(s_id, p_id)
```

- (γ)(i) Τους φοιτητές που έχουν παραπάνω από 5 χρόνια σπουδών και δεν έχουν σύμβουλο καθηγητή.
- (γ)(ii) Τους καθηγητές που διδάσκουν το ίδιο μάθημα με (τουλάχιστον) έναν φοιτητή του οποίου είναι σύμβουλοι.
- (γ)(iii) Το κενό πίνακα αν και μόνο αν η συμμετοχή των φοιτητών στη σχέση advisedby δεν είναι ολική.
- (γ)(iv) Το μέσο χρόνο σπουδών των φοιτητών που διδάσκουν τουλάχιστον δύο μαθήματα.
- (γ)(v) Για κάθε καθηγητή τον αριθμό των φοιτητών που επιβλέπει. Για τους καθηγητές που δεν επιβλέπουν κανένα καθηγητή να εμφανίζεται ο αριθμός 0.
- (γ)(vi) [προαιρετικό ερώτημα] Για κάθε στάδιο σπουδών, το μέσο όρο ετών σπουδών των φοιτητών που βρίσκονται σε αυτό σε φθίνουσα διάταξη ως προς αυτόν τον μέσο όρο.

Άσκηση 3 (SQL)

course(course_id, courseLevel)
student (s_id, inPhase, yearsinProgram)
professor(p_id, hasPosition)
taughtby(course_id, u_id)
advisedby(s_id, p_id)

(γ)(i) Τους φοιτητές που έχουν παραπάνω από 5 χρόνια σπουδών και δεν έχουν σύμβουλο καθηγητή.

```
select ST.s_id
from (select s_id from student where yearsinProgram > 5) as ST
where ST.s_id not in (select s_id
                      from advisedby);
```

```
select student.s_id
from student left join advisedby on student.s_id = advisedby.s_id
      (ή student natural left join student)
where student.yearsinProgram > 5 and advisedby.s_id = null;
```

Άσκηση 3 (SQL)

course(course_id, courseLevel)
student (s_id, inPhase, yearsinProgram)
professor(p_id, hasPosition)
taughtby(course_id, u_id)
advisedby(s_id, p_id)

(γ)(ii) Τους καθηγητές που διδάσκουν το ίδιο μάθημα με (τουλάχιστον) έναν φοιτητή του οποίου είναι σύμβουλοι.

```
select distinct A.pid  
from taughtby as T1, taughtby as T2, advisedby as A  
where T1.course_id = T2.course_id and T1.u_id = A.s_id and T2.u_id = A.p_id;
```

Άσκηση 3 (SQL)

```
course(course_id, courseLevel)
student (s_id, inPhase, yearsinProgram)
professor(p_id, hasPosition)
taughtby(course_id, u_id)
advisedby(s_id, p_id)
```

(γ)(iii) Το κενό πίνακα αν και μόνο αν η συμμετοχή των φοιτητών στη σχέση advisedby δεν είναι ολική.

```
select *
from student
where not exists (select *
                  from student
                  where s_id not in (select s_id
                                     from advised by);
```

Το εσωτερικό κενό αν και μόνο αν η σχέση είναι μερική, δηλαδή υπάρχει κάποιος φοιτητής που δεν εμφανίζεται στην advisedby
not exists TRUE αν κενή η εσωτερική (δηλαδή, μερική) FALSE αν ολική

Άσκηση 3 (SQL)

course(course_id, courseLevel)
student (s_id, inPhase, yearsinProgram)
professor(p_id, hasPosition)
taughtby(course_id, u_id)
advisedby(s_id, p_id)

(γ)(iv) Το μέσο χρόνο σπουδών των φοιτητών που διδάσκουν τουλάχιστον δύο μαθήματα.

```
select average(S.yearinProgram)
from (select s_id, count(course_id)
      from taughtby as T, student as S
      where T.u_id = s_id
      group by s_id
      having count(course_id) >=2) as T, student as S
where S_s_id = T_s_id;
```

Φοιτητές που διδάσκουν
τουλάχιστον 2 μαθήματα



```
select average(S.yearinProgram)
from (select s_id, count(course_id)
      from taughtby as T, student as S
      where T.u_id = s_id
      group by s_id
      having count(course_id) >=2) natural join S;
```

Άσκηση 3 (SQL)

```
course(course_id, courseLevel)
student(s_id, inPhase, yearsinProgram)
professor(p_id, hasPosition)
taughtby(course_id, u_id)
advisedby(s_id, p_id)
```

(γ)(ν) Για κάθε καθηγητή τον αριθμό των φοιτητών που επιβλέπει. Για τους καθηγητές που δεν επιβλέπουν κανένα καθηγητή να εμφανίζεται ο αριθμός 0.

```
(select p_id, count(s_id)
from advisedby
group by p_id)
union
(select p_id, 0
from professor
where p_id not in (select p_id
                    from advisedby));
```

Θα μπορούσε και με εξωτερική συνένωση, αλλά όχι τόσο «καθαρή» η ερμηνεία του count

```
select p_id, count(s_id)
from advisedby natural right join professor
group by professor.pid;
```

Άσκηση 3 (SQL)

```
course(course_id, courseLevel)
student (s_id, inPhase, yearsinProgram)
professor(p_id, hasPosition)
taughtby(course_id, u_id)
advisedby(s_id, p_id)
```

(γ)(vi) [προαιρετικό ερώτημα] Για κάθε στάδιο σπουδών, το μέσο όρο ετών σπουδών των φοιτητών που βρίσκονται σε αυτό σε φθίνουσα διάταξη ως προς αυτόν τον μέσο όρο.

```
select inPhase, average(yearsinProgram)
from student
group by inPhase
ordered by average(yearsinProgram) desc;
```

Άσκηση 3 (SQL)

course(course_id, courseLevel)
student (s_id, inPhase, yearsinProgram)
professor(p_id, hasPosition)
taughtby(course_id, u_id)
advisedby(s_id, p_id)

(δ) Δώστε την SQL έκφραση που να έχει ως αποτέλεσμα αύξηση κατά ένα των ετών σπουδών των φοιτητών που δεν διδάσκουν κάποιο μάθημα.

```
update student
set yearsinProgram = yearsinProgram +1
where student.s_id not in (select u_id
                           from taughtby);
```

Άσκηση 3 (SQL)

course(course_id, courseLevel)
student (s_id, inPhase, yearsinProgram)
professor(p_id, hasPosition)
taughtby(course_id, u_id)
advisedby(s_id, p_id)

(ε) Τα παρακάτω ερωτήματα αφορούν στην χρήση όψεων.

(i) Ορίστε μια όψη professor-count(p_id, c_count) που να περιέχει για κάθε καθηγητή που διδάσκει ένα μάθημα τον αριθμό των μαθημάτων που διδάσκει. Δείτε το περιεχόμενο αυτής της όψης, χρησιμοποιώντας το select * from professor-count.

(ii) Εισάγετε δυο πλειάδες στη σχέση taughtby. Δείτε το περιεχόμενο της όψης μετά την εισαγωγή χρησιμοποιώντας πάλι το select * from professor-count.

(ii) Είναι η παραπάνω όψη τροποποιήσιμη (updatable); Εξηγείστε.

```
create view professorcount(p_id, c_count) as  
select u_id, count(course_id)  
from taughtby  
where u_id in (select p_id  
               from professor)  
group by u_id;
```

```
create view professorcount(p_id, c_count) as  
select u_id, count(course_id)  
from taughtby inner join professor on taughtby.u_id = professor.p_id  
group by u_id;
```