

DATABASE MANAGEMENT SYSTEMLAB-7**Answer the following SQL queries**

1. Each offering of a course (i.e. a section) can have many Teaching assistants; each teaching assistant is a student. Extend the existing schema (Add/Alter tables) to accommodate this requirement.

```
varsha_university=# create table assistant(
ID varchar(5),
course_id varchar(8),
sec_id varchar(8),
semester varchar(6),
year numeric(4,0),
PRIMARY KEY (ID,course_id,sec_id,semester,year),
FOREIGN KEY (course_id,sec_id,semester,year)
references section(course_id,sec_id,semester,year) on delete cascade,
FOREIGN KEY(ID) references student(ID) on delete cascade);
CREATE TABLE
```

```
varsha_university=# select *from assistant;
 id   | course_id | sec_id | semester | year
-----+-----+-----+-----+-----
00306 | BIO-101   | 1      | Summer   | 2009
00306 | BIO-301   | 1      | Summer   | 2010
00306 | CS-101    | 1      | Fall     | 2009
00339 | BIO-101   | 1      | Summer   | 2009
00339 | BIO-301   | 1      | Summer   | 2010
00339 | CS-101    | 1      | Fall     | 2009
00420 | CS-101    | 1      | Fall     | 2009
00314 | BIO-101   | 1      | Summer   | 2009
00314 | BIO-301   | 1      | Summer   | 2010
00314 | CS-101    | 1      | Fall     | 2009
(10 rows)
```

2. According to the existing schema, one student can have only one advisor. Alter the schema to allow a student to have multiple advisors and make sure that you are able to insert multiple advisors for a student.

```
varsha_university=# alter table advisor add constraint advisor_pkey primary key(s_ID,i_ID);
ALTER TABLE
```

```
varsha_university=# select *from advisor;
 s_id | i_id
-----+-----
00412 | 12121
00412 | 10101
00056 | 22222
00339 | 76766
00339 | 83821
00420 | 10101
00525 | 22222
00339 | 12121
00339 | 10101
00525 | 12345
00258 | 67891
00412 | 22222
00412 | 76766
(13 rows)
```

**Write SQL queries on the modified schema. You will need to insert data to ensure the query results are not empty.**

1. Find all students who have more than 3 advisors

```
varsha_university=# select student.name,student.ID
varsha_university=# from student
varsha_university=# where student.ID in(
varsha_university=# select s_ID
varsha_university=# from advisor
varsha_university=# group by s_ID having count(s_ID)>3);
 name | id
-----+-----
varsha | 00339
Swati  | 00412
(2 rows)
```

2. Find all students who are co-advised by Prof. Srinivas and Prof. Ashok.

```
varsha_university=# select distinct(s_ID)
from advisor
where i_ID in(
select id
from instructor
where name='Srinivas' or name='Ashok'
);
 s_id
-----
00258
00525
(2 rows)
```

3. Find students advised by instructors from different departments. Etc.

```
varsha_university=# select distinct student.ID,student.name
from (select advisor.s_ID,advisor.i_ID,dept_name
from instructor,advisor
where instructor.id=advisor.i_ID) as s,
(select advisor.s_ID,advisor.i_ID,dept_name
from instructor,advisor
where instructor.id=advisor.i_ID
) as t,student where s.s_id=student.ID and
s.s_ID=t.s_ID and s.dept_name<>t.dept_name
;
 id | name
-----+-----
00525 | Tanaka
00339 | varsha
00412 | Swati
(3 rows)
```

4. Delete all information in the database which is more than 10 years old. Add data as necessary to verify your query.

```
varsha_university=# delete from takes where year<(select extract(year from CURRENT_DATE)-10);
DELETE 9
varsha_university=# delete from teaches where year<(select extract(year from CURRENT_DATE)-10);
DELETE 15
varsha_university=# delete from section where year<(select extract(year from CURRENT_DATE)-10);
DELETE 15
```

5. Delete the course CS 101. Any course which has CS 101 as a prereq should remove CS 101 from its prereq set. Create a cascade constraint to enforce the above rule, and verify that it is working.

‘ON DELETE CASCADE’

Specification is present as part of the foreign key from prereq.prereq\_id referencing course.

```
varsha_university=# create table prereq
varsha_university=# (course_id
varsha_university(# varchar(8),
varsha_university(# prereq_id
varsha_university(# varchar(8),
varsha_university(# primary key (course_id, prereq_id),
varsha_university(# foreign key (course_id) references course
varsha_university(# on delete cascade,
varsha_university(# foreign key (prereq_id) references course on delete cascade
varsha_university(# );
CREATE TABLE
```

```
varsha_university=# DELETE from course where course_id='CS-101';
DELETE 1
```

Verification:

```
varsha_university=# DELETE from prereq where prereq_id='CS-101';
DELETE 4
varsha_university=# DELETE from course where course_id='CS-101';
DELETE 1
```