

DBMS Assignment 3: SQL DDL and Updates

Name: Omkar Oak

MIS: 112103099

Div: 2

Batch: T1

Statement : Write the DDL and DML statements for the following.

- 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.

```
• CREATE TABLE IF NOT EXISTS TEACHING_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  
);
```

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.

```
12  -- To remove foreign key constraints.  
13 • ALTER TABLE advisor DROP constraint advisor_ibfk_1;  
14 • ALTER TABLE advisor DROP constraint advisor_ibfk_2;  
15  
16  -- To drop primary key  
17 • ALTER TABLE advisor DROP primary key;  
18  
19  -- to add multiple primary keys and add foreign key constraints  
20 • ALTER TABLE advisor ADD PRIMARY key (s_ID, i_ID);  
21 • ALTER TABLE advisor ADD foreign key (i_ID) references instructor (ID),  
22   ADD foreign key (s_ID) references student (ID) on delete cascade;  
23  
24  -- inserting new dvalues (contains mutiple advisors)  
25 • INSERT INTO advisor (s_ID, i_ID) VALUES ('23121', '76766');  
26 • INSERT INTO advisor (s_ID, i_ID) VALUES ('23121', '45565');  
27 • INSERT INTO advisor (s_ID, i_ID) VALUES ('54321', '22222');
```

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

- Find all students who have more than 3 advisors

```
30
31 • SELECT s_ID FROM advisor GROUP BY s_ID HAVING COUNT(i_ID) >= 3;
32
33
```

Result Grid	
s_ID	
▶	23121

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

```
33 • SELECT s_ID FROM advisor WHERE i_ID IN
34 (SELECT ID FROM instructor WHERE name IN ('Srinivasan', 'Ashok')) GROUP BY s_ID;
35
```

Result Grid	
Filter Rows:	Export: Wrap Cell Content:
s_ID	
▶	12345

- Find students advised by instructors from different departments. etc.

```
36 • SELECT a.s_ID FROM advisor a JOIN instructor i ON a.i_ID = i.ID
37 GROUP BY a.s_ID HAVING COUNT(DISTINCT i.dept_name) > 1;
--
```

Result Grid	
Filter Rows:	Export: Wrap Cell Content:
s_ID	
▶	23121
	54321

Write SQL queries for the following:

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

```
39 • DELETE FROM takes WHERE year < (EXTRACT(YEAR FROM CURRENT_DATE) - 10);
40 • DELETE FROM teaches WHERE year < (EXTRACT(YEAR FROM CURRENT_DATE) - 10);
41 • DELETE FROM section WHERE year < (EXTRACT(YEAR FROM CURRENT_DATE) - 10);
42
```

- 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.

```
43 • ALTER TABLE prereq ADD CONSTRAINT fk_prereq_course
44 FOREIGN KEY (prereq_id) REFERENCES course(course_id) ON DELETE CASCADE;
```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
course_id	prereq_id			
BIO-301	BIO-101			
BIO-399	BIO-101			
CS-190	CS-101			
CS-315	CS-101			
CS-319	CS-101			
CS-347	CS-101			
EE-181	PHY-101			
NULL	NULL			

```
46 • DELETE FROM course WHERE course_id = 'CS-101';
```

Result Grid	Filter Rows:	Edit:	Export/Import:	W
course_id	prereq_id			
BIO-301	BIO-101			
BIO-399	BIO-101			
EE-181	PHY-101			
NULL	NULL			