Q2. Find top 5 students (get their names and department) by tot_cred (i.e, those top 5 who have completed highest total credits).

Solution : select s.id,s.name,s.dept_name,creditList.total_credits from (select t.id,sum(c.credits) as total_credits from takes t,course c where t.course_id=c.course_id group by t.id order by total_credits desc LIMIT 5) as creditList inner join student as s on s.id=creditList.id;

Q5 . Use begin transaction SQL statement along with commit/rollback.

You need to write insert statements for the following:
i) add a course
ii) create a section for it with some year/semester etc.
iii) assign a teacher to it
Create a file (.sql type) containing the above statements with begin,commit or rollback transactions as per the following cases.Create separate .sql file for each of the following cases and execute :

Note : Load different data in each case as you might get errors if the data gets inserted and may cause insert rejection.

1. Put commit after all three insert statements.
2. Put rollback after second insert statement and commit after third insert statement.
3. Put rollback after all three insert statements.

After each case specify if the data is inserted or not and the reason for the same in your submission along with the screenshot of each .sql file.

Solution :

1. **Putting commit after all 3 insert statements**

```
BEGIN;
insert into course values('303', 'DBMS', 'Comp. Sci.', 3);
insert into section values('303', '1', 'Fall', 2010, 'Chandler', '804', 'N');
insert into teaches values('34175', '303', '1', 'Fall', 2010);
COMMIT;
```

All data will be inserted into tables.

2. **Now putting rollback after second statement and commit after third statement**

```
BEGIN;
insert into course values('303', 'DBMS', 'Comp. Sci.', 3);
insert into section values('303', '1', 'Fall', 2010, 'Chandler', '804', 'N');
ROLLBACK;
insert into teaches values('34175', '735', '2', 'Spring', 2010);
COMMIT;
```

Only the third insert statement is executed if there is no foriegn key constraint with respect to first and second statements .First and second are not inserted due to rollback.

3. **Now putting rollback after all 3 insert statements**

```
BEGIN;
insert into course values('303', 'DBMS', 'Comp. Sci.', 3);
insert into section values('303', '1', 'Fall', 2010, 'Chandler', '804', 'N');
insert into teaches values('34175', '303', '1', 'Fall', 2010);
ROLLBACK;
```

Nothing is inserted due to rollback after all insert statements.

QUIZ QUESTION

Q 1. Use this feature of postgresql to create a materialized view TotalCredits which will have only two columns : department name and total credits offered in that department.

After creating this m-view, use it to query to obtain the department names having the largest total credits (output will be department name and total credits).
Write SQL query on base tables to check if this answer is correct !

Solution :

Query on base table for checking

select dept_name,sum(credits) from course group by dept_name having sum(credits)=(Select max(total_credits) from (select dept_name,sum(credits) as total_credits from course group by dept_name)as c2);

Using Materialized view

CREATE materialized VIEW TotalCredits AS select dept_name,sum(credits) from course group by dept_name;

select dept_name,sum from TotalCredits where sum=(Select max(sum) from TotalCredits);