

CS313, Lab – 5

Rishit Saiya, 180010027

October 3, 2020

Q.2]

Query:

```
SELECT st.name, st.dept_name, SUM(c.credits) FROM course as c, student AS st, takes AS t WHERE st.id =t.id AND t.course_id=c.course_id GROUP BY st.id ORDER BY SUM(c.credits) DESC LIMIT 5;
```

Output:

```
university=# SELECT st.name, st.dept_name, SUM(c.credits) FROM course as c, student AS st, takes AS t WHERE st.id =t.id AND t.course_id=c.course_id GROUP BY st.id ORDER BY SUM(c.credits) DESC LIMIT 5;
 name | dept_name | sum
-----|-----|-----
Knutson | Languages | 93
Nguyen | Astronomy | 90
Godfrey | English | 90
Schmitz | Elec. Eng. | 90
Wakamiya | Comp. Sci. | 86
(5 rows)
```

Q.5]

i]

Query: Putting COMMIT after all three INSERT statements.

```
BEGIN TRANSACTION;
INSERT INTO course VALUES ('060', 'Capture The Flag', 'Comp. Sci.', 6);
INSERT INTO section VALUES ('060', '1', 'Spring', '2019', 'Gates', '707', 'K');
INSERT INTO teaches VALUES ('3200', '060', '1', 'Spring', '2019');
COMMIT TRANSACTION;
```

SQL file Screenshot:

```
BEGIN TRANSACTION;
INSERT INTO course VALUES ('060', 'Capture The Flag', 'Comp. Sci.', 6);
INSERT INTO section VALUES ('060', '1', 'Spring', '2019', 'Gates', '707', 'K')
;
INSERT INTO teaches VALUES ('3199', '060', '1', 'Spring', '2019');
COMMIT TRANSACTION;
```

Output:

```
university=# \i 'C:\\Users\\Rishit Saiya\\Desktop\\SEMESTER V\\CS 313 Database System Lab\\Assignments\\Assignment-5\\query1.sql'
BEGIN
INSERT 0 1
INSERT 0 1
INSERT 0 1
COMMIT
```

We do not get any errors here and all the tuples are inserted, because we just INSERT the tuples here and finally we commit them.

ii]

Query: Put ROLLBACK after second INSERT statement and COMMIT after third INSERT statement.

```
BEGIN TRANSACTION;
INSERT INTO course VALUES ('061', 'Reverse Engineering', 'Comp. Sci.', 6);
INSERT INTO section VALUES ('061', '1', 'Spring', '2019', 'Gates', '707', 'K');
ROLLBACK TRANSACTION;
INSERT INTO teaches VALUES ('3199', '061', '1', 'Spring', '2019');
COMMIT TRANSACTION;
```

SQL file Screenshot:

```
BEGIN TRANSACTION;
INSERT INTO course VALUES ('061', 'Reverse Engineering', 'Comp. Sci.', 6);
INSERT INTO section VALUES ('061', '1', 'Spring', '2019', 'Gates', '707', 'K')
;
ROLLBACK TRANSACTION;
INSERT INTO teaches VALUES ('3199', '061', '1', 'Spring', '2019');
COMMIT TRANSACTION;
```

Output:

```
university=# \i "C:\Users\Rishit Saiya\Desktop\SEMESTER V\CS 313 Database System Lab\Assignments\Assignment-5\query2.sql"
BEGIN
INSERT 0 1
INSERT 0 1
ROLLBACK
psql:C:/Users/Rishit Saiya/Desktop/SEMESTER V/CS 313 Database System Lab/Assignments/Assignment-5/query2.sql:5: ERROR: insert or update on table "teaches" violates foreign key constraint "teaches_course_id_fkey"
DETAIL:  Key (course_id, sec_id, semester, year)=(061, 1, Spring, 2019) is not present in table "section".
psql:C:/Users/Rishit Saiya/Desktop/SEMESTER V/CS 313 Database System Lab/Assignments/Assignment-5/query2.sql:6: WARNING: there is no transaction in progress
COMMIT
```

Here after INSERT of first 2 tuples, we give a ROLLBACK statement, which instates an error because 3rd tuple has a foreign key-id of Course Code (061) referring to first 2 tuples which we just removed. Hence we get error for the 3rd INSERT query.

iii]

Query: Put ROLLBACK after all three INSERT statements.

```
BEGIN TRANSACTION;
INSERT INTO course VALUES ('062', 'Binary Exploitation', 'Comp. Sci.', 6);
INSERT INTO section VALUES ('062', '1', 'Spring', '2019', 'Gates', '314', 'K');
INSERT INTO teaches VALUES ('79081', '062', '1', 'Spring', '2019');
ROLLBACK TRANSACTION;
```

SQL file Screenshot:

```
BEGIN TRANSACTION;
INSERT INTO course VALUES ('062', 'Binary Exploitation', 'Comp. Sci.', 6);
INSERT INTO section VALUES ('062', '1', 'Spring', '2019', 'Gates', '314', 'K')
;
INSERT INTO teaches VALUES ('79081', '062', '1', 'Spring', '2019');
ROLLBACK TRANSACTION;
```

Output:

```
university=# \i 'C:\\Users\\Rishit Saiya\\Desktop\\SEMESTER V\\CS 313 Database System Lab\\Assignments\\Assignment-5\\query3.sql'
BEGIN
INSERT 0 1
INSERT 0 1
INSERT 0 1
ROLLBACK
```

Here we don't get any errors because we INSERT all the tuples sequentially and then remove them at the end, so there is no scope of NULL referencing as it happened in previous query. So finally all of them were added and removed at the end.

Q. QUIZ]

Base Table Query:

```
WITH T1 AS (SELECT SUM(course.credits) AS total_credits_offered, department.dept_name AS department_name FROM department, course WHERE department.dept_name = course.dept_name GROUP BY department.dept_name) SELECT * FROM T1 WHERE (SELECT max(total_credits_offered) FROM T1) = total_credits_offered;
```

Output:

```
university=# WITH T1 AS (SELECT SUM(course.credits) AS total_credits_offered, department.dept_name AS department_name FROM department, course WHERE department.dept_name = course.dept_name GROUP BY department.dept_name) SELECT * FROM T1 WHERE (SELECT max(total_credits_offered) FROM T1) = total_credits_offered;
 total_credits_offered | department_name 
-----
67 | Cybernetics
(1 row)
```

Materialized View:

```
CREATE MATERIALIZED VIEW TotalCredits AS SELECT dept_name, SUM(credits) AS total_credits_offered FROM course GROUP BY dept_name;
```

Materialized View Query:

```
SELECT * FROM TotalCredits WHERE total_credits_offered IN (SELECT MAX(total_credits_offered) FROM TotalCredits);
```

Output:

```
university=# SELECT * FROM TotalCredits WHERE total_credits_offered IN (SELECT MAX(total_credits_offered) FROM TotalCredits);
 department_name | total_credits_offered 
-----+-----
 Cybernetics      |                      67
(1 row)
```