New submission for: Lecture 16 Exercise

Due: 10/30/2021 @ 04:00 PM EDT

Gradeable Time Remaining: 11 mins 55 seconds

Database Setup. Suppose you have three unprivileged use database who all can connect to a database named testare

Assume the following commands all execute correctly:

```
$ psql -U postgres -- means login as user postgres

psql> create role dbs_grp1;
psql> grant dbs_grp1 to dbs1;
psql> grant dbs_grp1 to dbs2;
psql> create role dbs_grp2;
psql> grant dbs_grp2 to dbs3;
psql> \q

$ psql testarea -U dbs1 -- means login to database testa

psql> create table tmp1 (id1 int);
psql> create table tmp2 (id1 int);
psql> create table tmp3 (id1 int);

psql> grant select on tmp1 to dbs_grp1 ;
psql> grant select on tmp1 to dbs_grp2 with grant optior
psql> grant select on tmp2 to dbs2 with grant option ;
psql> grant select on tmp3 to dbs_grp2 ;
```

Question 1: Check all the commands below that will su testarea as shown below:

```
$ psql testarea -U dbs2 -- means login to database testa
```

You may select many:

- ✔ select * from tmp1 ;
- ✔ select * from tmp2 ;
- select * from tmp3 ;
- grant select on tmp1 to dbs3;
- ✔ grant select on tmp2 to dbs3;
- grant select on tmp3 to dbs3;
- None of these choices

Clear Use Most Recent Submission

Question 2: Assuming you start from the original set of p
check all the commands below that will succeed (not raise e
below:

```
$ psql testarea -U dbs3 -- means login to database testa
```

You may select many:

- ✔ select * from tmp1 ;
- select * from tmp2 ;
- ✔ select * from tmp3 ;
- ✔ grant select on tmp1 to dbs2;
- grant select on tmp2 to dbs2;
- grant select on tmp3 to dbs2;
- None of these choices

Clear Use Most Recent Submission

Trigger Setup. You are given the following two tables wit

```
CREATE TABLE a(id INT PRIMARY KEY, name VARCHAR(20)) ;
CREATE TABLE b(id INT PRIMARY KEY, name VARCHAR(20)) ;

CREATE FUNCTION a_trg1 () RETURNS trigger AS $$
    BEGIN
        IF NEW.id < 4 THEN
            NEW.name = upper(NEW.name);
            DELETE FROM b WHERE id = NEW.id ;
   END IF ;
        RETURN NEW;
    END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER a_trg1 BEFORE INSERT ON a
    FOR EACH ROW EXECUTE FUNCTION a_trg1();

CREATE FUNCTION b_trg1 () RETURNS trigger AS $$
    BEGIN
        NEW.name = lower(NEW.name);
        INSERT INTO a VALUES(OLD.id, OLD.name);
        RETURN NEW;
    END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER b_trg1 AFTER UPDATE ON b
    FOR EACH ROW EXECUTE FUNCTION b_trg1();
```

Check all tuples in the database after the following transact

```
BEGIN ;
INSERT INTO a VALUES(1,'red') ;
INSERT INTO b VALUES(2,'GREEN') ;
INSERT INTO a VALUES(3,'violet') ;
INSERT INTO b VALUES(4,'blue') ;
UPDATE b SET name = 'teal' WHERE id = 4 ;
UPDATE a SET name = 'PURPLE' WHERE id = 1 ;
UPDATE b SET name = 'ORANGE' WHERE id = 2 ;
COMMIT;
```

You may select many:

- Table a (1,'red')
- Table a (1,'RED')
- Table a (1,'purple')
- ✔ Table a (1,'PURPLE')
- Table a (2,'green')
- ✔ Table a (2,'GREEN')
- Table a (3,'violet')
- ✔ Table a (3,'VIOLET')
- Table a (3,'orange')
- Table a (3,'ORANGE')
- ✔ Table a (4,'blue')
- Table a (4,'BLUE')
- Table a (4,'teal')
- Table a (4,'TEAL')
- Table b (1,'red')
- Table b (1,'RED')
- Table b (1,'purple')
- Table b (1,'PURPLE')
- Table b (2,'green')
- Table b (2,'GREEN')
- Table b (3,'violet')
- Table b (3,'VIOLET')
- Table b (3,'orange')
- Table b (3,'ORANGE')
- Table b (4,'blue')
- Table b (4,'BLUE')
- ✔ Table b (4,'teal')
- Table b (4,'TEAL')

Clear Use Most Recent Submission

By clicking "Submit" you are confirming that you have read, understand, and agree to follow the Academic

Integrity Policy.

Submit

Select Submission Version:

Version #4 GRADE THIS VERSION

Do Not Grade This Assignment

Note: This version of your assignment will be graded by the instructor/TAs and the score recorded in the gradebook.

Submitted Files

l16ex_1.txt (0.07kb)	Download
l16ex_2.txt (0.07kb)	Download
l16ex_3.txt (0.10kb)	Download
Download all files:	

First access timestamp:	10/28/2021 @ 07:36:14 PM EDT
Submission timestamp:	10/28/2021 @ 08:46:32 PM EDT
Days late:	0 (before extensions)
Grading time:	0 seconds
Queue wait time:	0 seconds
Gradeable access duration:	1 hours 10 minutes 18 seconds

No Autograding for this Gradeable