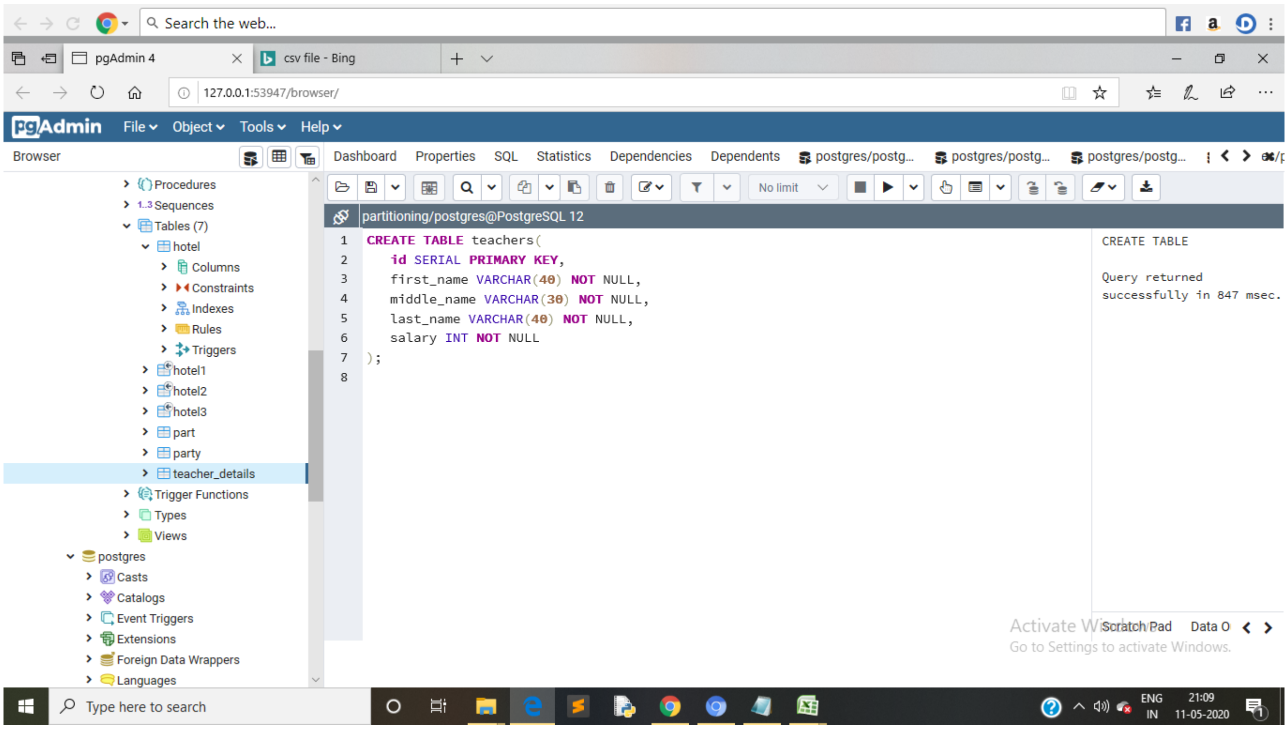
BABIKA A
KloudOne
babianandhan@gmail.com

PostgreSQL Drop trigger

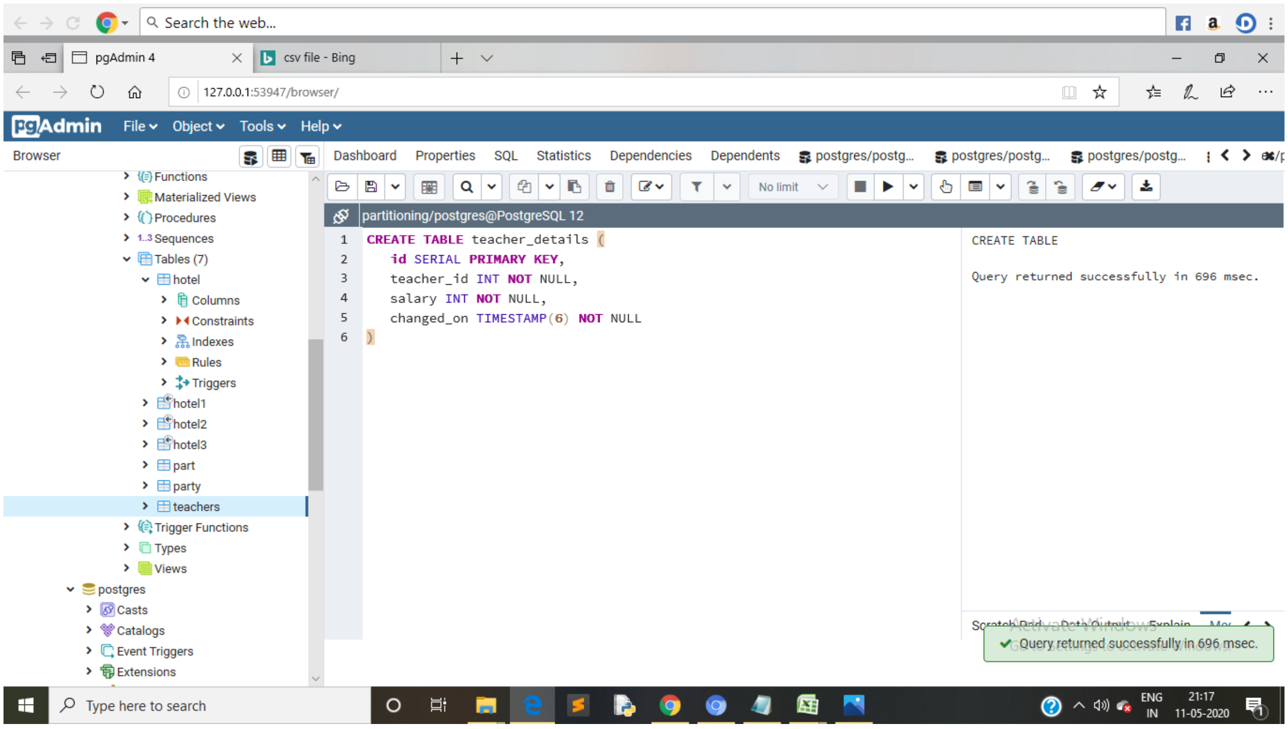
Step1: Creating two tables teacher and teacher_details and insert values



partitioning/postgres@PostgreSQL 12

```
CREATE TABLE teachers(
   id SERIAL PRIMARY KEY,
   first_name VARCHAR(40) NOT NULL,
   middle_name VARCHAR(30) NOT NULL,
   last_name VARCHAR(40) NOT NULL,
   salary INT NOT NULL
);
```

```
CREATE TABLE

Query returned
successfully in 847 msec.
```



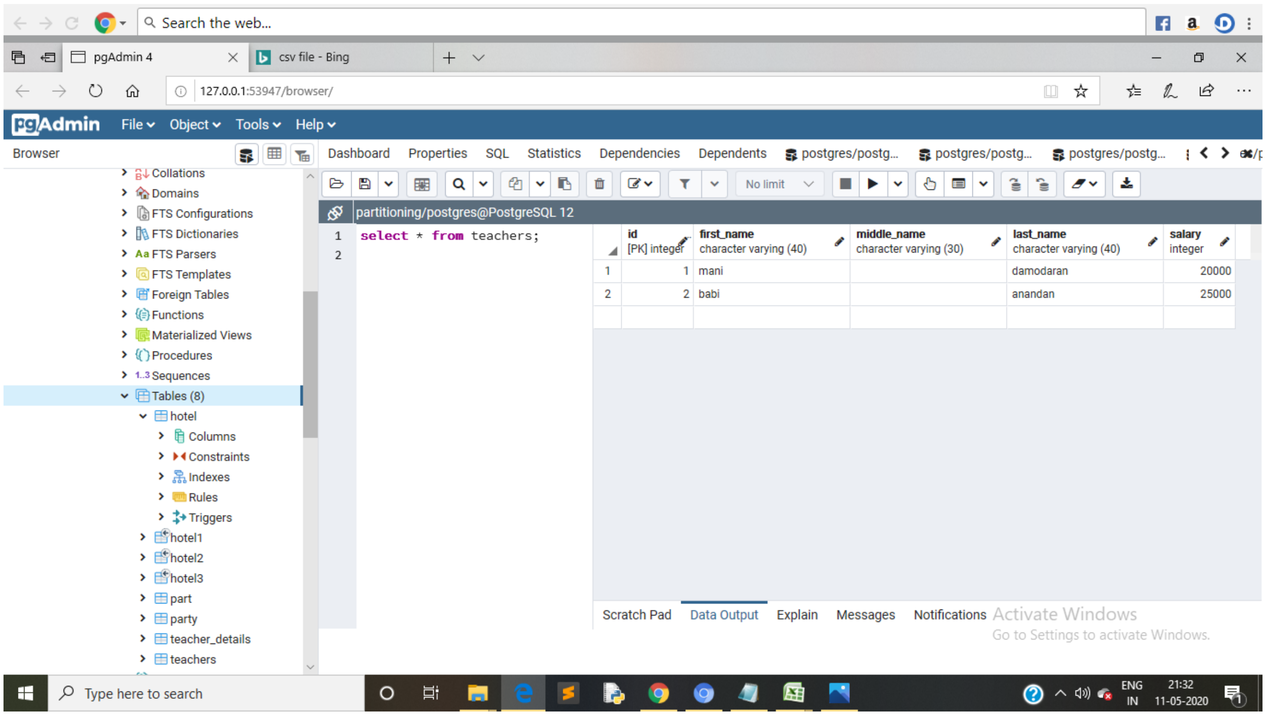
partitioning/postgres@PostgreSQL 12

```
CREATE TABLE teacher_details (
   id SERIAL PRIMARY KEY,
   teacher_id INT NOT NULL,
   salary INT NOT NULL,
   changed_on TIMESTAMP(6) NOT NULL
)
```

```
CREATE TABLE

Query returned successfully in 696 msec.
```

Query returned successfully in 696 msec.

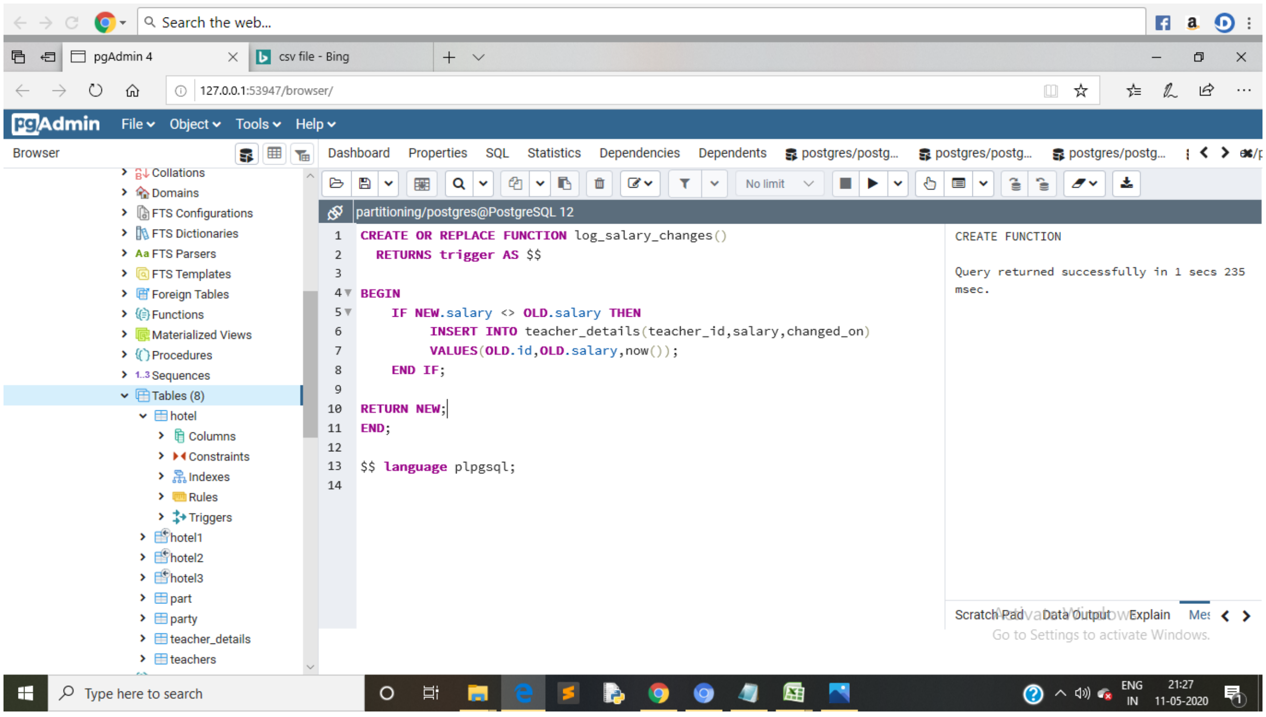
BABIKA A
KloudOne
babianandhan@gmail.com



```
select * from teachers;
```

	id [PK] integer	first_name character varying (40)	middle_name character varying (30)	last_name character varying (40)	salary integer
1	1	mani		damodaran	20000
2	2	babi		anandan	25000

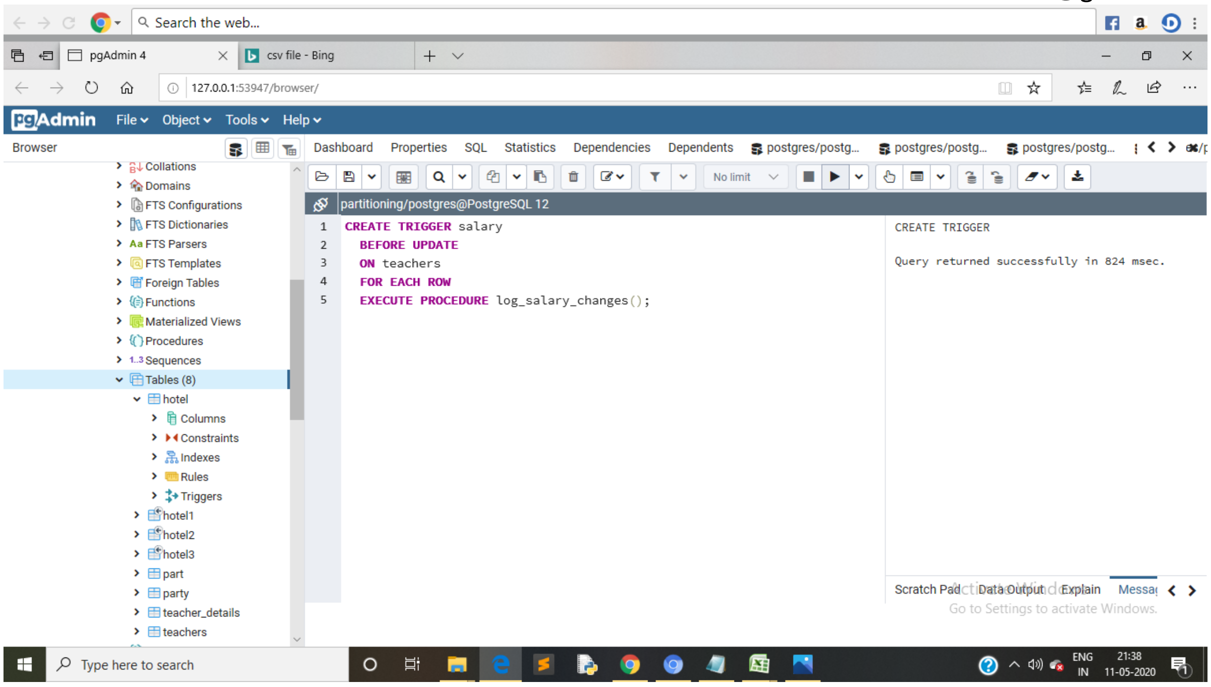
Step2: Creating Function and Trigger



```
CREATE OR REPLACE FUNCTION log_salary_changes()
  RETURNS trigger AS $$

BEGIN
    IF NEW.salary <> OLD.salary THEN
        INSERT INTO teacher_details(teacher_id,salary,changed_on)
        VALUES(OLD.id,OLD.salary,now());
    END IF;

RETURN NEW;
END;

$$ language plpgsql;
```

```
CREATE FUNCTION

Query returned successfully in 1 secs 235
msec.
```

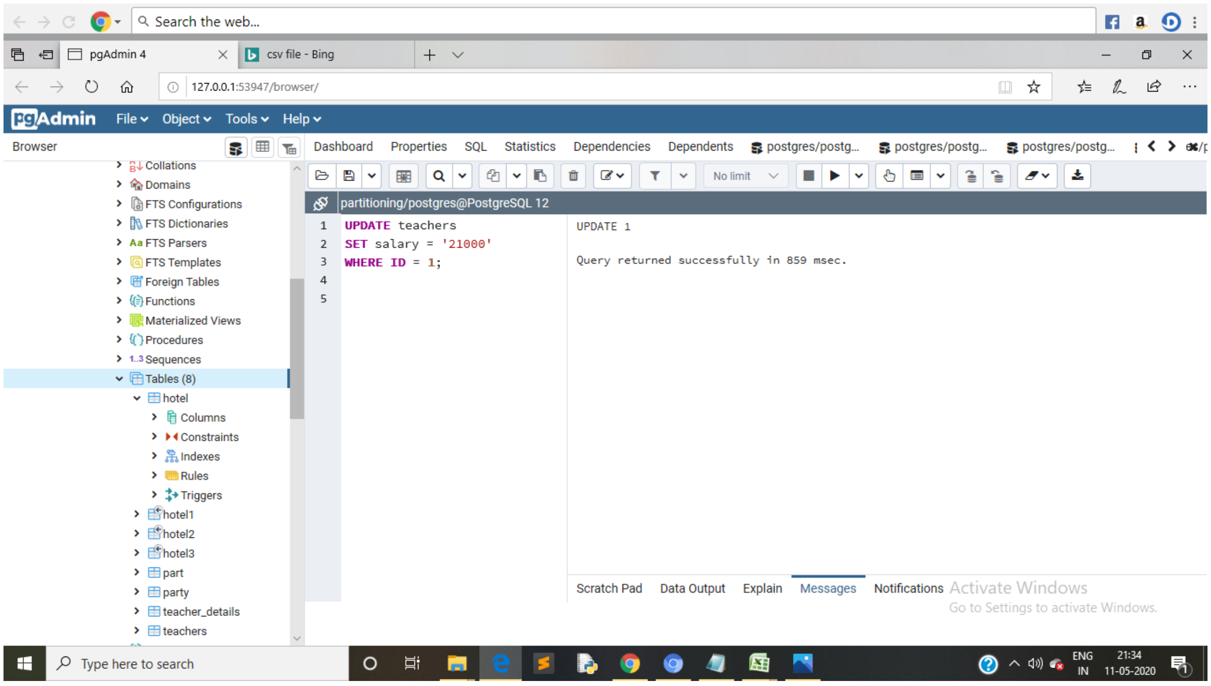
BABIKA A
KloudOne
babianandhan@gmail.com



```
CREATE TRIGGER salary
  BEFORE UPDATE
  ON teachers
  FOR EACH ROW
  EXECUTE PROCEDURE log_salary_changes();
```

```
CREATE TRIGGER

Query returned successfully in 824 msec.
```

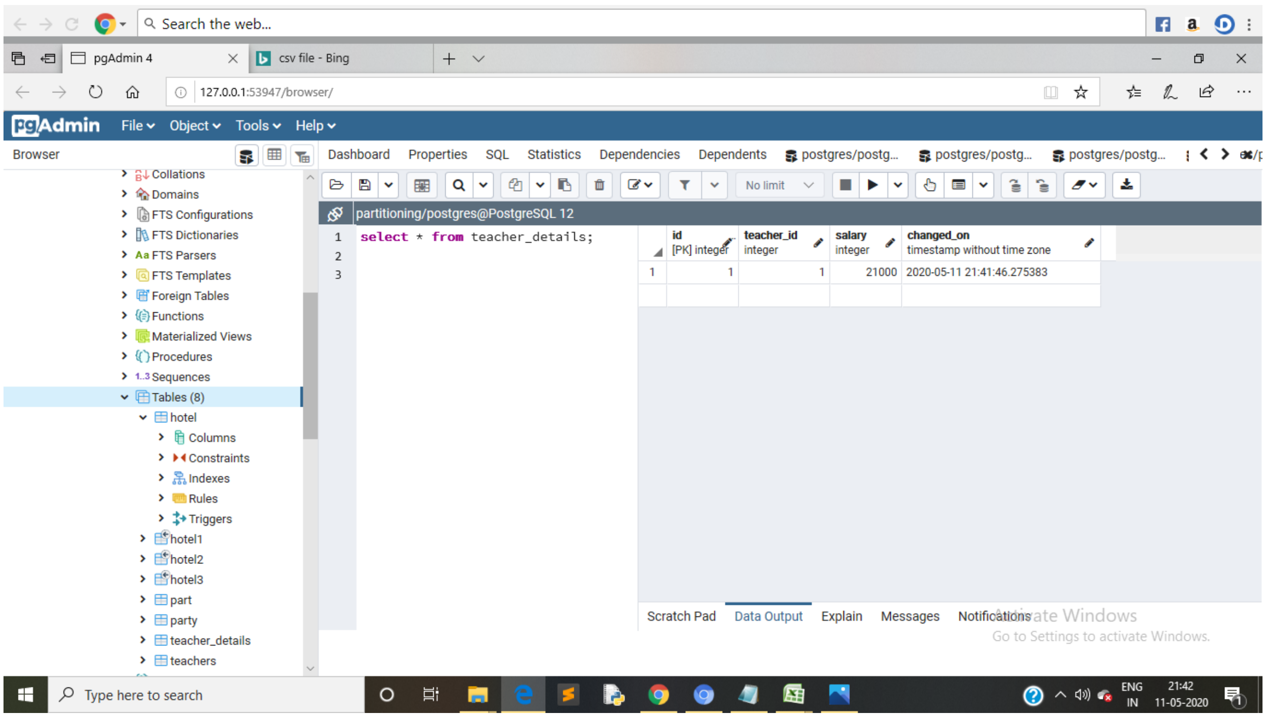
Step3: Update the values



```
UPDATE teachers
SET salary = '21000'
WHERE ID = 1;
```

```
UPDATE 1

Query returned successfully in 859 msec.
```

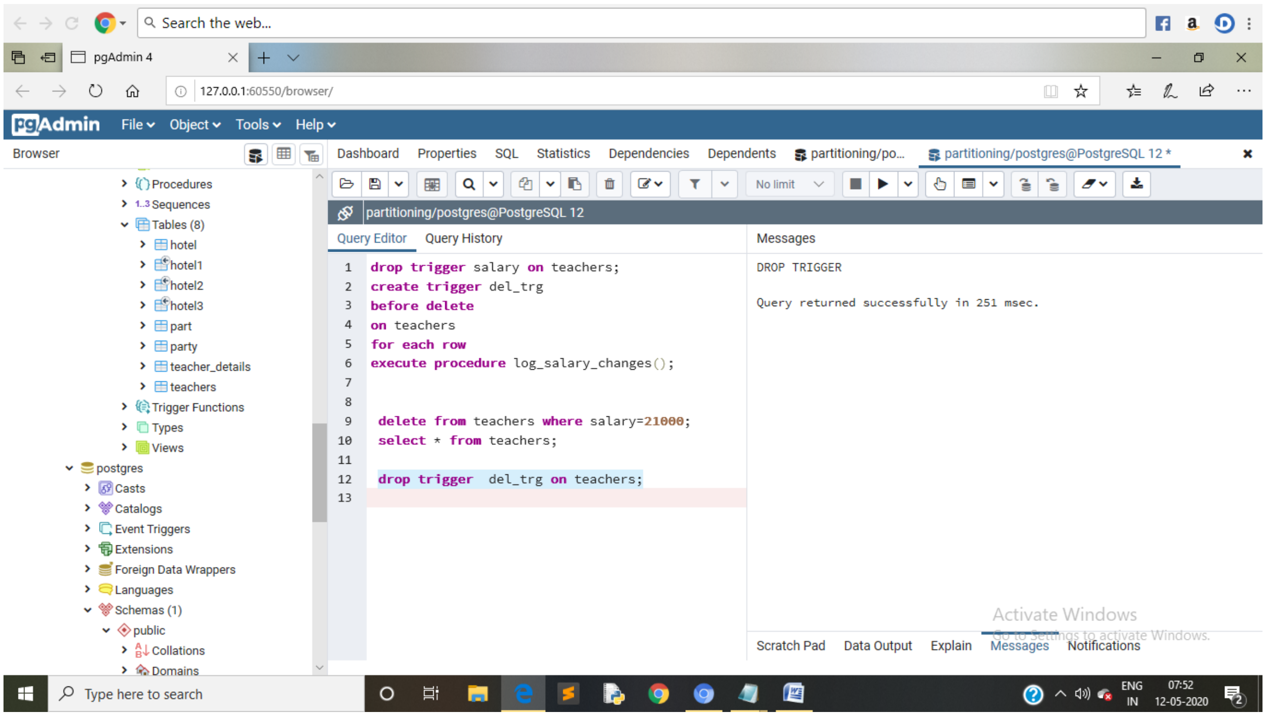
BABIKA A
KloudOne
babianandhan@gmail.com



```
select * from teacher_details;
```

id [PK] integer	teacher_id integer	salary integer	changed_on timestamp without time zone
1	1	21000	2020-05-11 21:41:46.275383

Step4 : Drop the trigger



```
drop trigger salary on teachers;
create trigger del_trg
before delete
on teachers
for each row
execute procedure log_salary_changes();


 delete from teachers where salary=21000;
 select * from teachers;

 drop trigger  del_trg on teachers;
```

```
DROP TRIGGER

Query returned successfully in 251 msec.
```