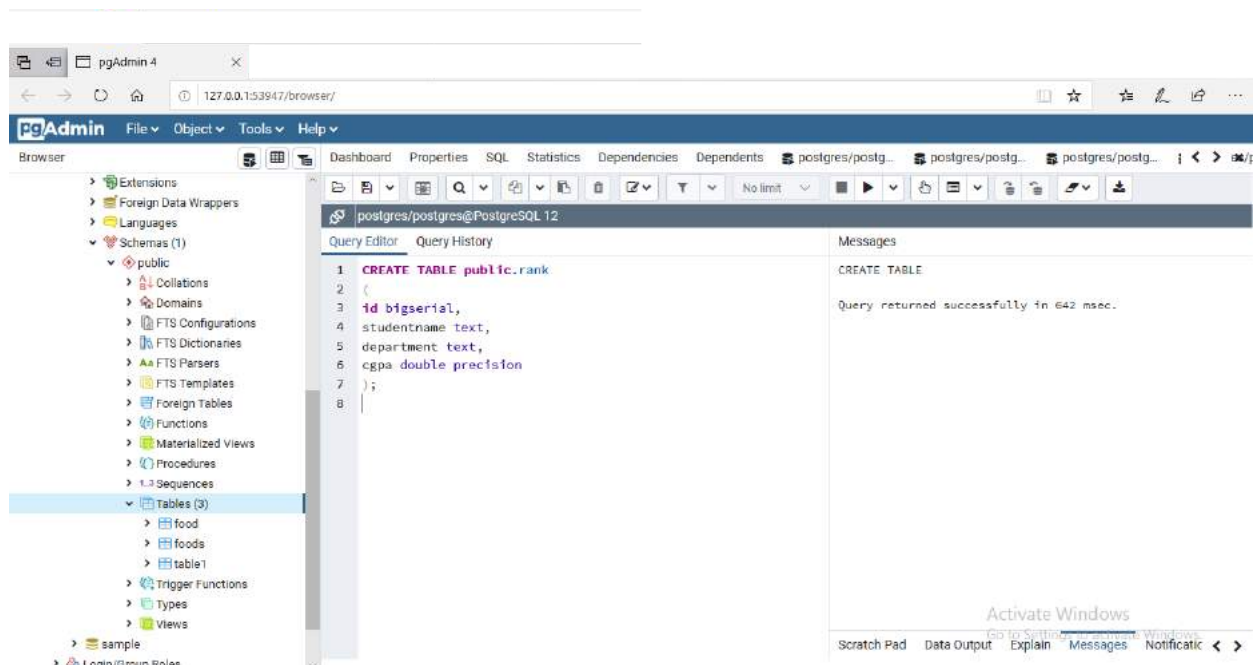
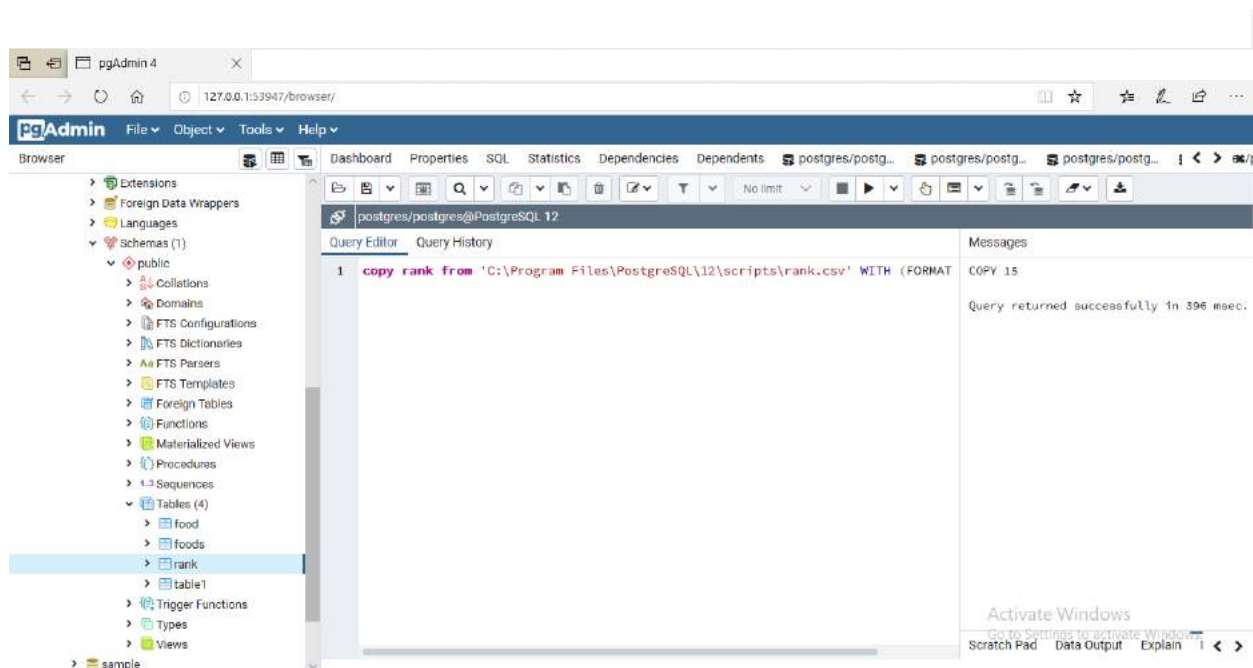


# Import csv files into a table using pgAdmin

## Step1: Creating a table



## Step2: Import csv file into the table



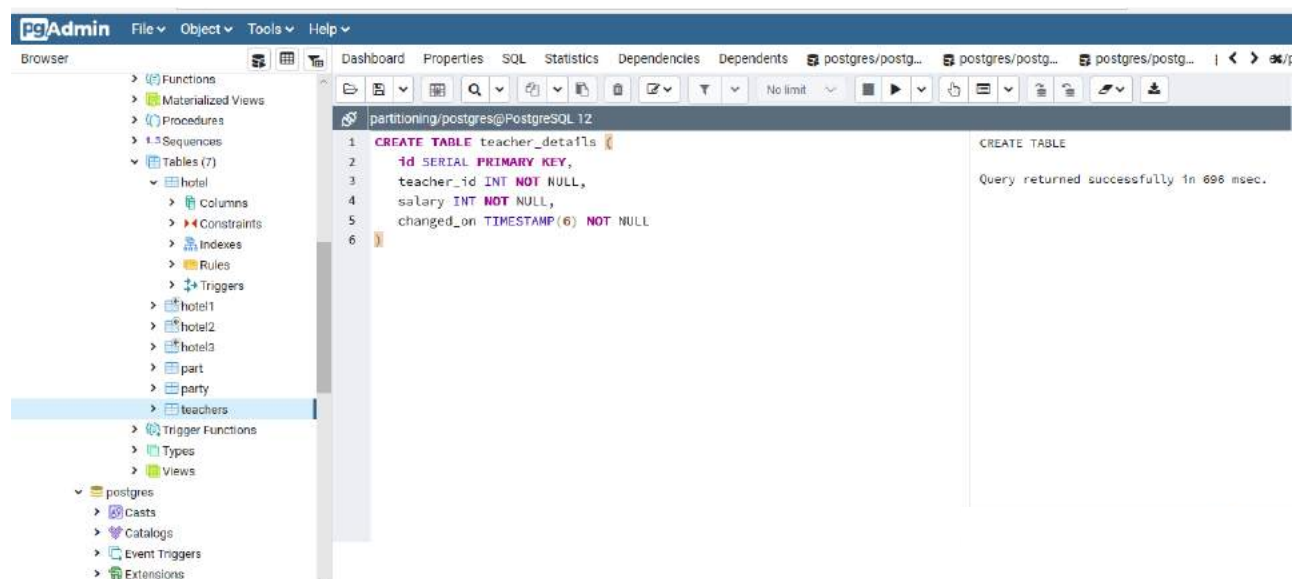
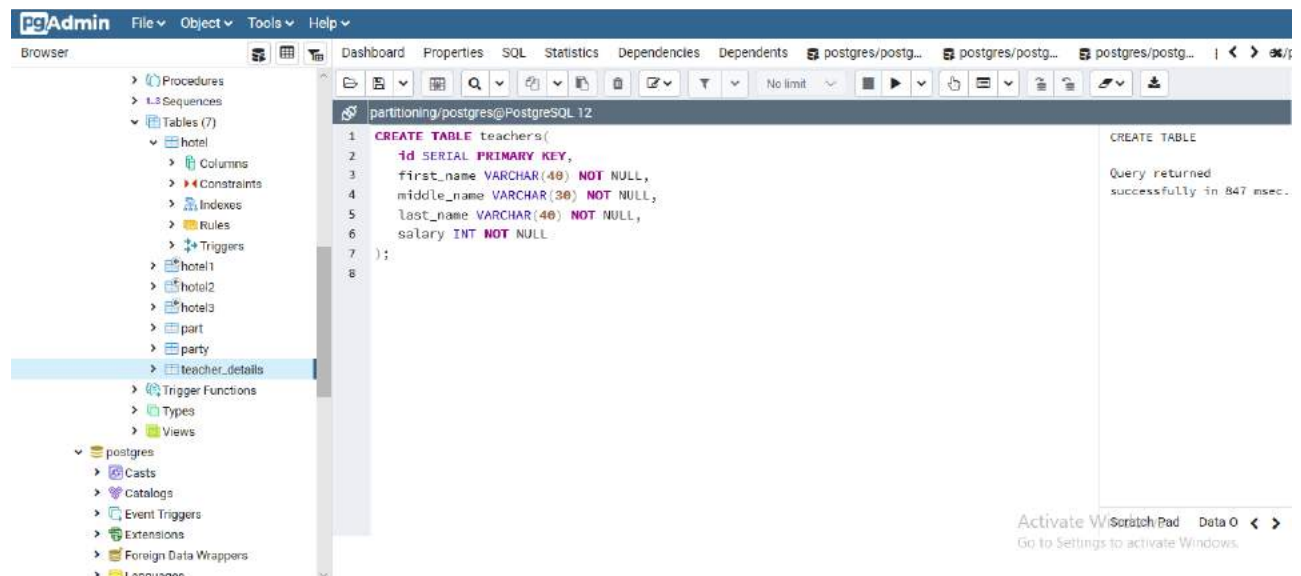
Step3: Display the imported data by select command.

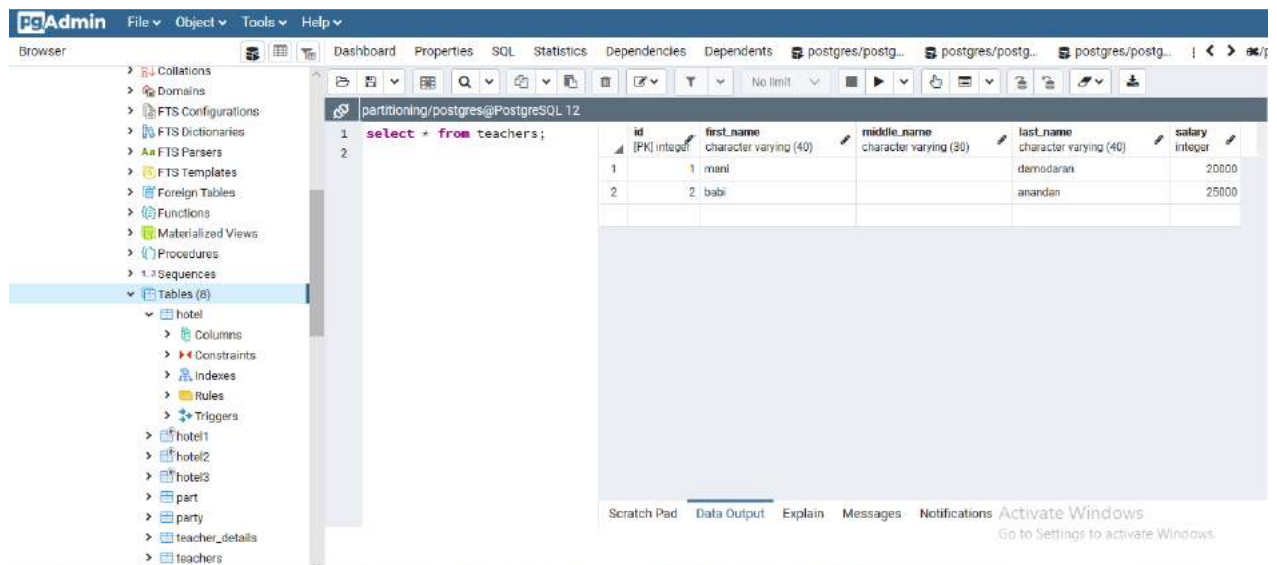
The screenshot shows the pgAdmin 4 interface. On the left, the 'Browser' pane displays a tree view of database objects. The 'public' schema is expanded, showing 'Tables (4)' with 'rank' selected. The 'Query Editor' pane contains the SQL command: `1 select * from rank;`. The 'Data Output' pane displays the results of the query in a table format.

	id	studentname	department	cgpa
	bigint	text	text	double precision
1	1	asa	cse	9.1
2	2	bbb	eee	9
3	3	ccc	eee	8.9
4	4	dod	cse	8.8
5	5	eee	eee	8.7
6	6	fff	mech	8.69
7	7	ggg	it	8.68
8	8	hhh	cse	8.67
9	9	iii	cse	8.5
10	10	jjj	civil	8.49
11	11	kkk	civil	8.43
12	12	lll	mech	8.42
13	13	mmm	cse	8.41
14	14	nnn	cse	8.4
15	15	ooo	cse	8

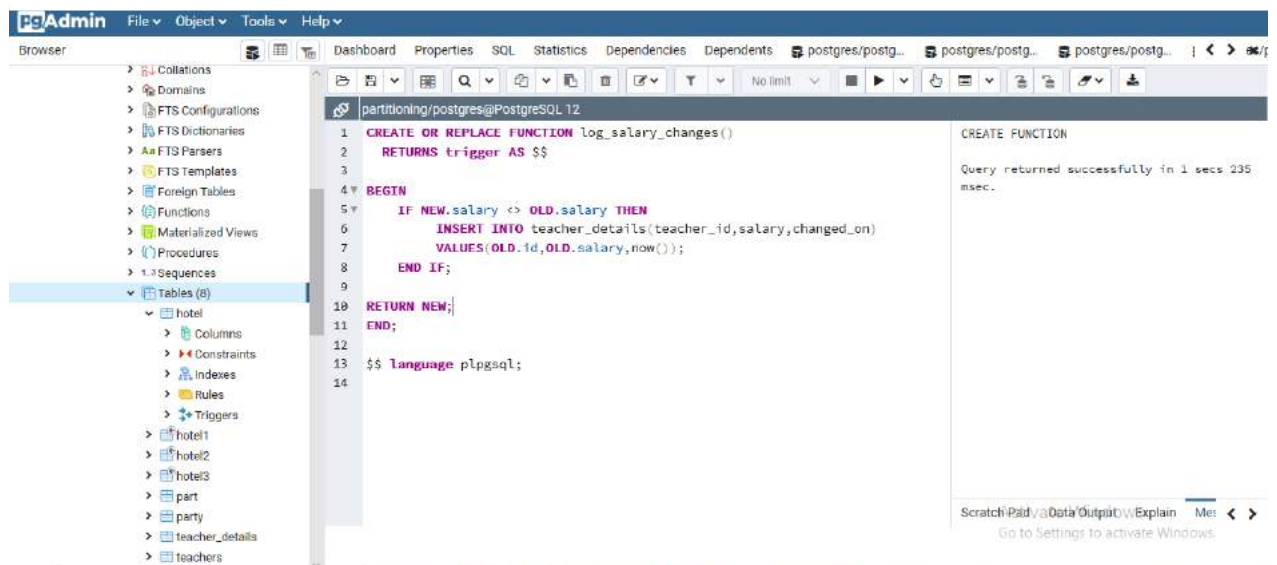
# PostgreSQL Drop trigger

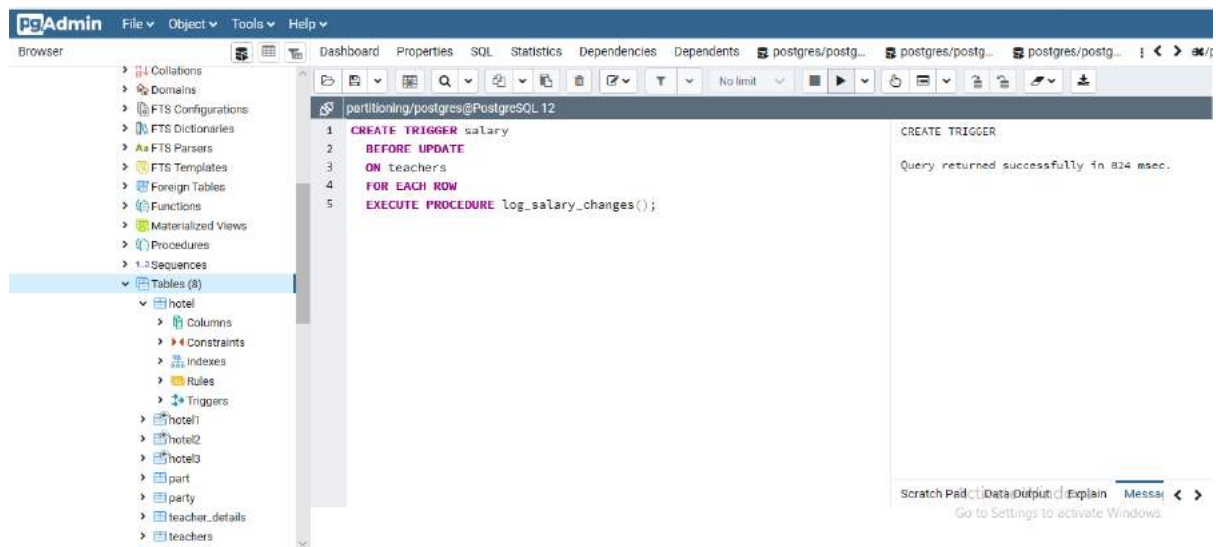
Step1: Creating two tables teacher and teacher\_details and insert values



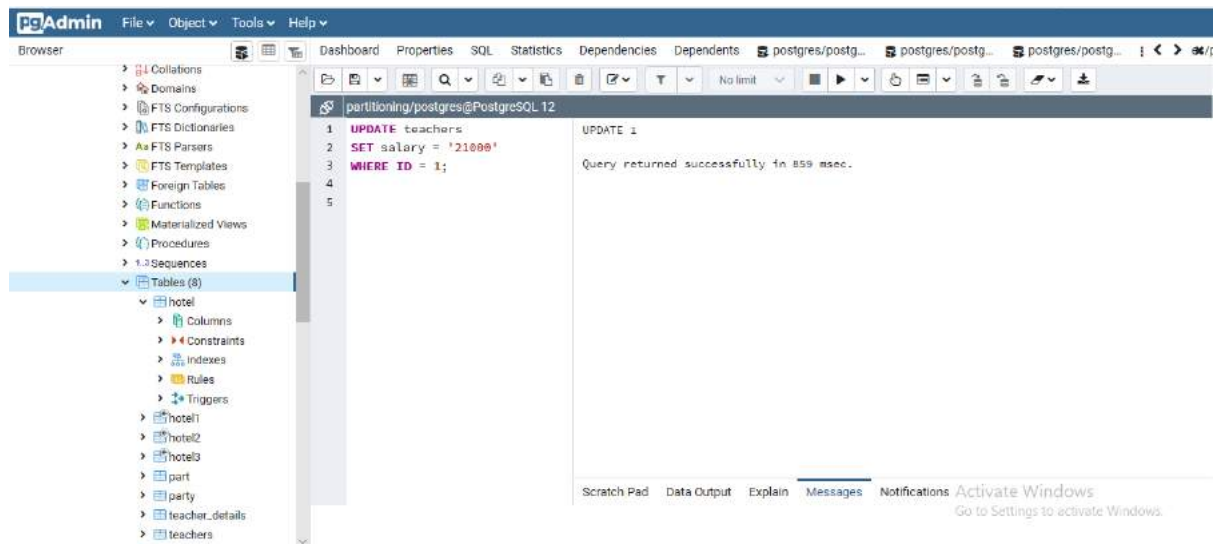


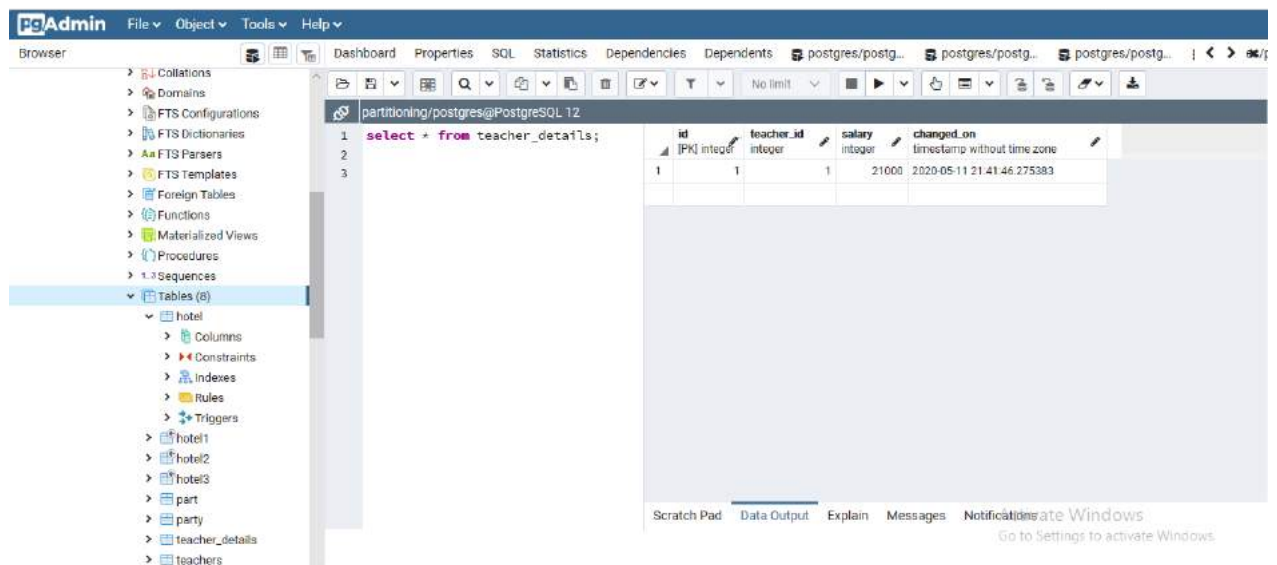
## Step2: Creating Function and Trigger



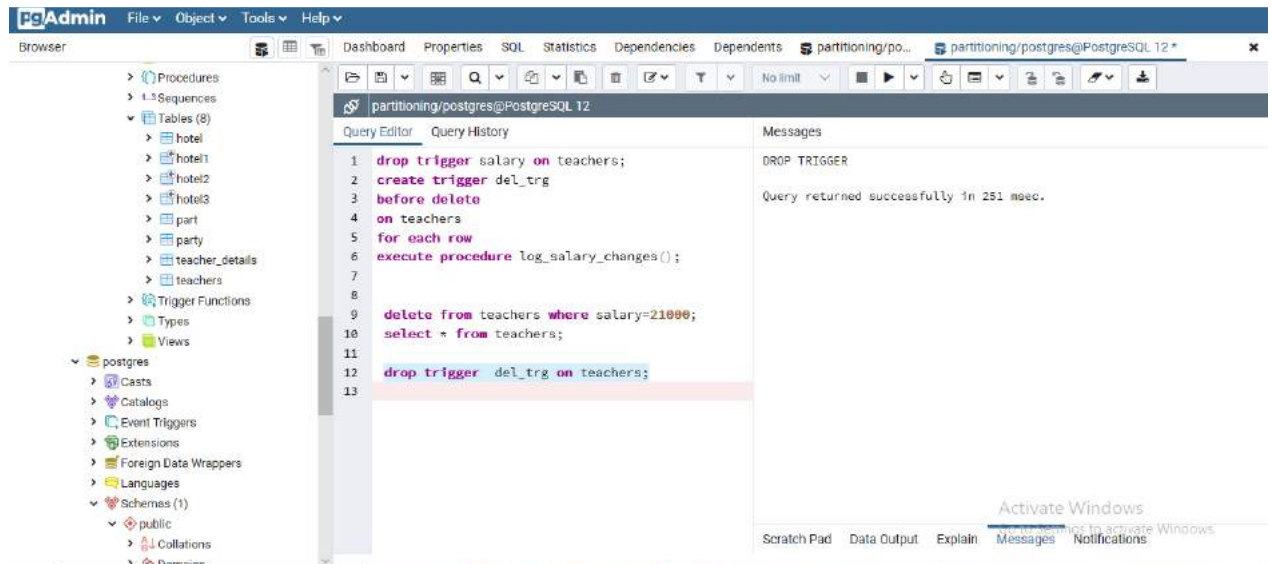


### Step3: Update the values



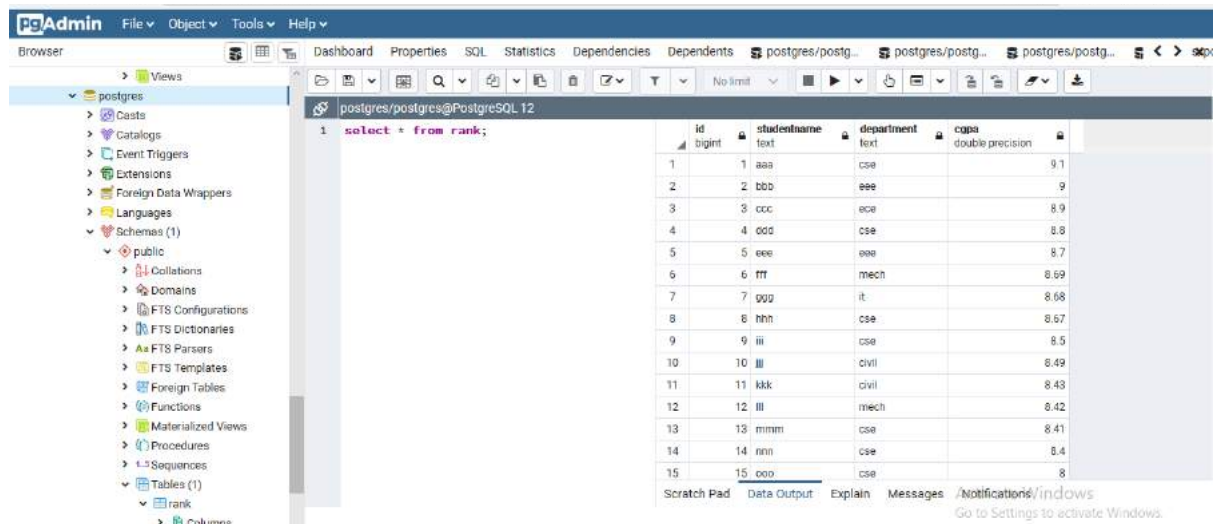


Step4 : Drop the trigger



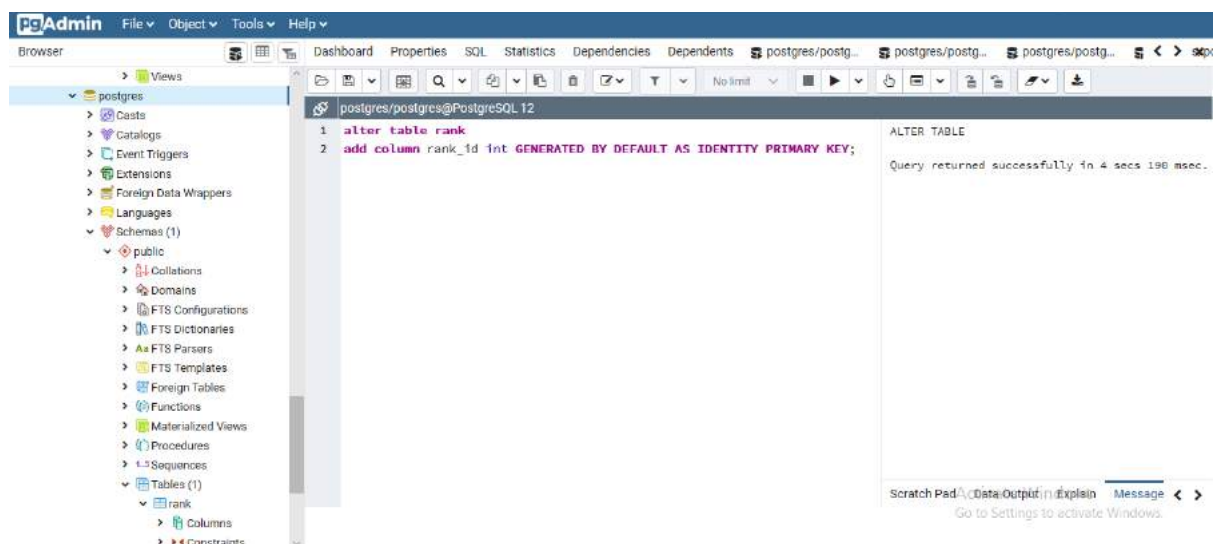


# Add an auto increment primary key to an existing table



The screenshot shows the pgAdmin interface with the 'rank' table selected in the left sidebar. The table is located in the 'public' schema. The table structure is as follows:

id	bigint	studentname	department	cgpa
1	1	aaa	cse	9.1
2	2	bbb	eee	9
3	3	ccc	ece	8.9
4	4	ddd	cse	8.8
5	5	eee	eee	8.7
6	6	fff	mech	8.69
7	7	ggg	it	8.68
8	8	hhh	cse	8.67
9	9	iii	cse	8.5
10	10	jjj	civil	8.49
11	11	kkk	civil	8.43
12	12	lll	mech	8.42
13	13	mmm	cse	8.41
14	14	nnn	cse	8.4
15	15	ooo	cse	8



The screenshot shows the pgAdmin interface with the SQL editor open. The SQL command being executed is:

```
1 alter table rank
2 add column rank_id int GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY;
```

The output of the query is displayed in the right pane:

```
ALTER TABLE
Query returned successfully in 4 secs 190 msec.
```

The screenshot shows the pgAdmin interface with the 'rank' table selected in the browser. The table has the following columns: id (bigint), studentname (text), department (text), cgpa (double precision), and rank\_id (integer [PK]). The data is as follows:

id	studentname	department	cgpa	rank_id
1	aaa	cse	9.1	1
2	bbb	eee	9	2
3	ccc	eee	8.9	3
4	ddd	cse	8.8	4
5	eee	eee	8.7	5
6	fff	mech	8.69	6
7	ggg	it	8.68	7
8	hhh	cse	8.67	8
9	iii	cse	8.5	9
10	jjj	civil	8.49	10
11	kkk	civil	8.43	11
12	lll	mech	8.42	12
13	mmm	cse	8.41	13
14	nnn	cse	8.4	14
15	ooo	cse	8	15

## Add an auto increment foreign key to an existing table

The screenshot shows the pgAdmin interface with the SQL Editor open. The query being executed is:

```

1 select * from rank;
2 select * from two;
3 alter table two add constraint rank_fk
4 foreign key(two_id) references rank(rank_id);
5
6

```

The Messages pane shows the following output:

```

ALTER TABLE
Query returned successfully in 278 msec.

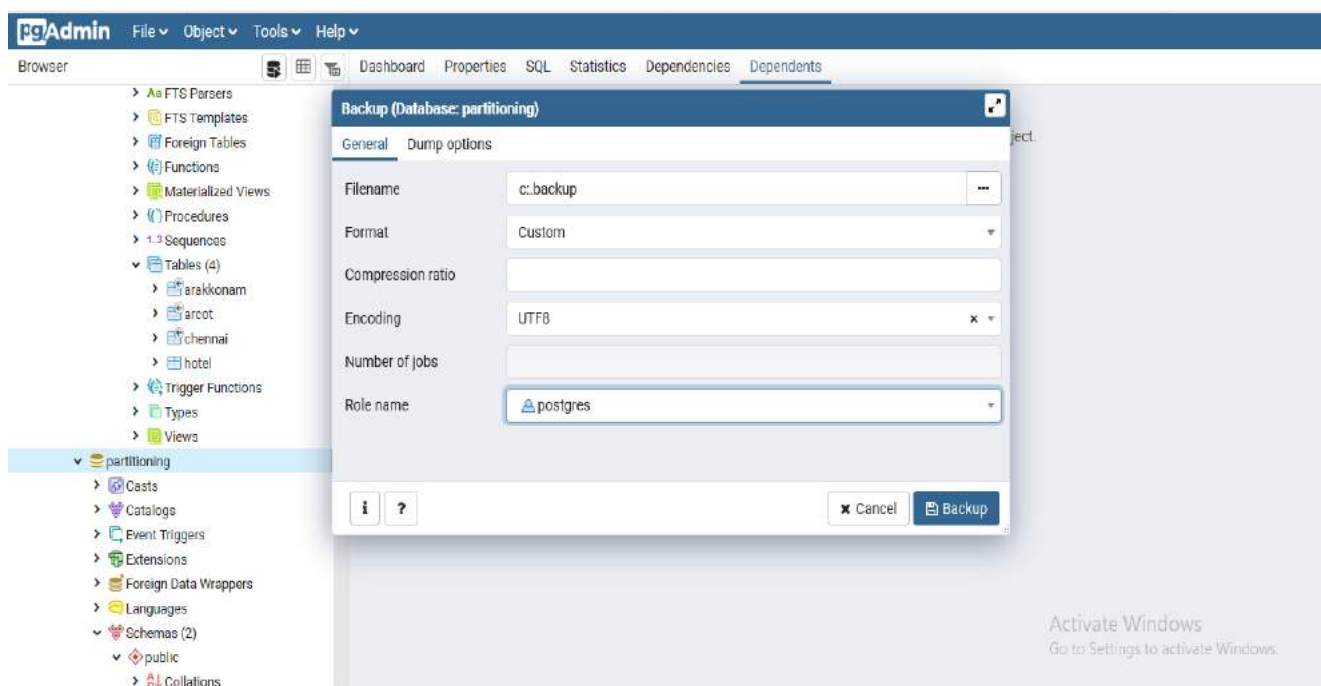
```





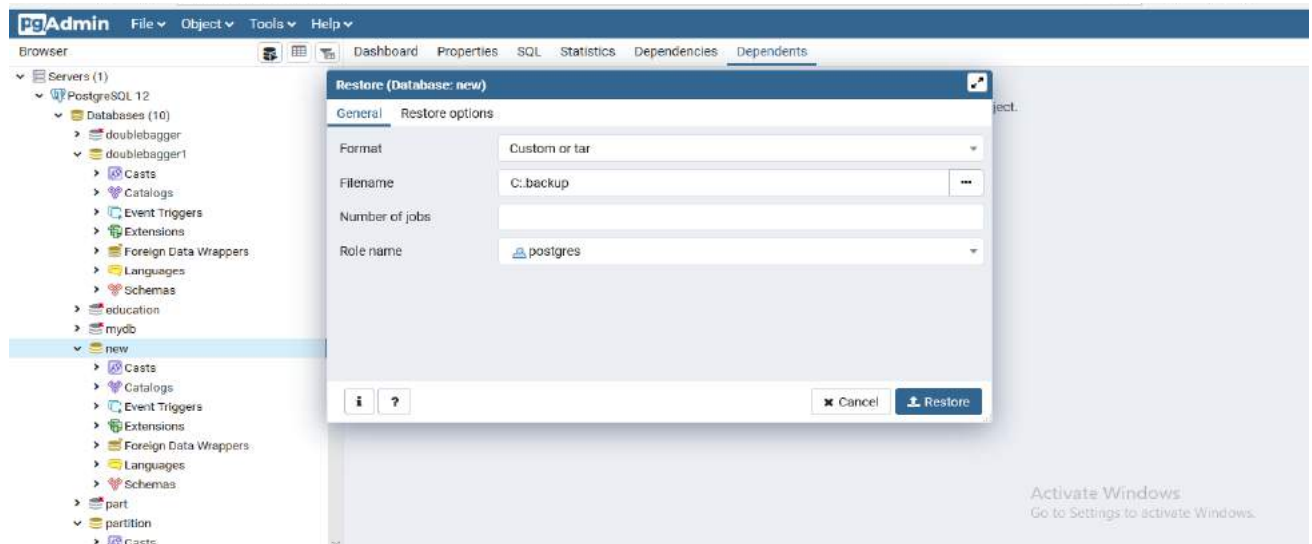
## BACKUP and RESTORE the files

Select a database to backup the data and click right click-> backup

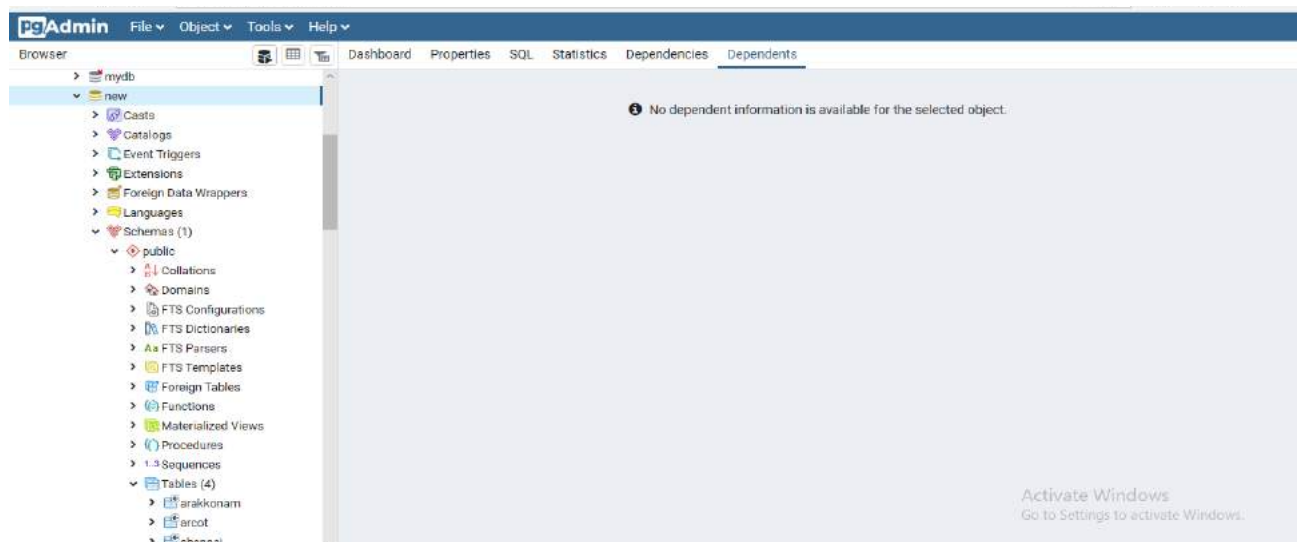


Create a new database and restore the backup into in the new database

click right click-> restore

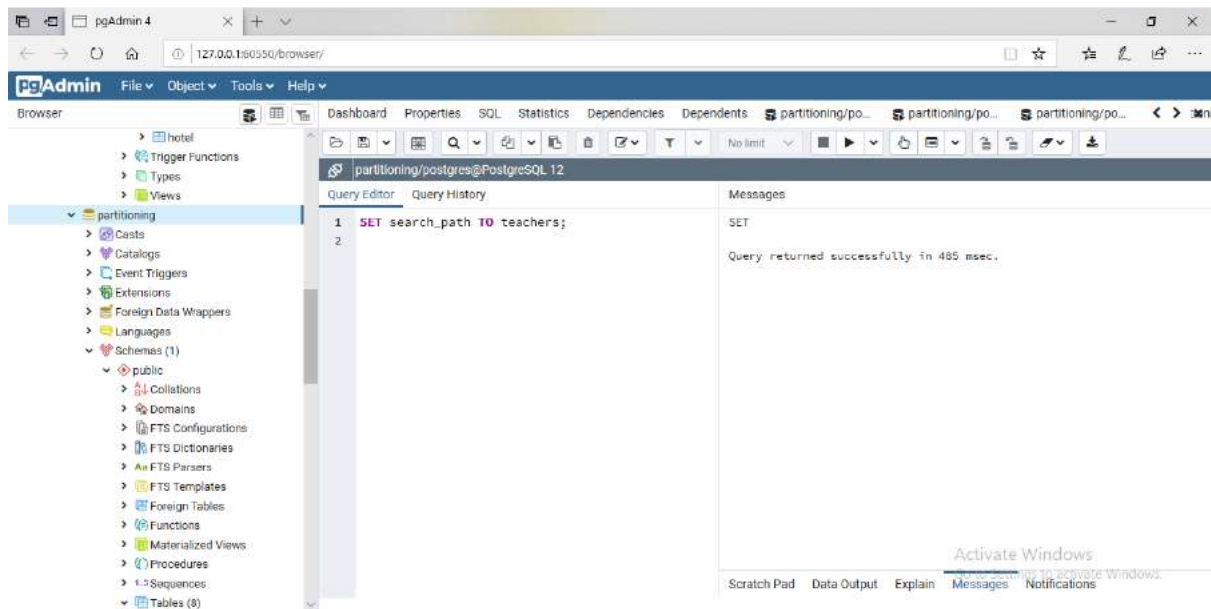


Finally in the new database .we add the backup database tables.

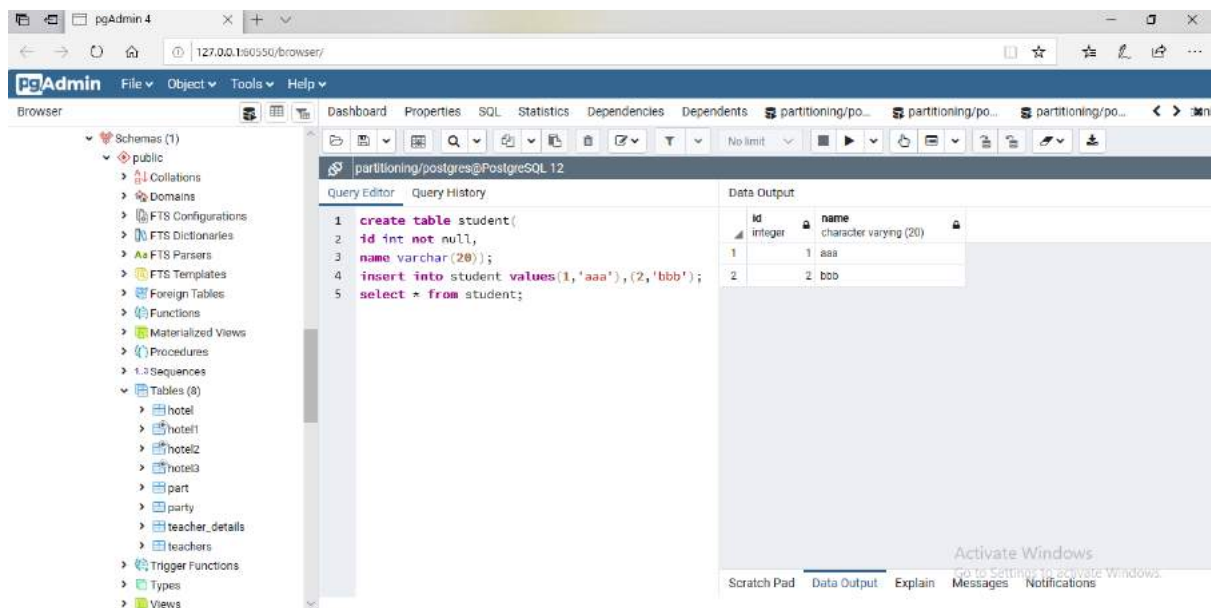


# POSTGRES DML COMMANDS

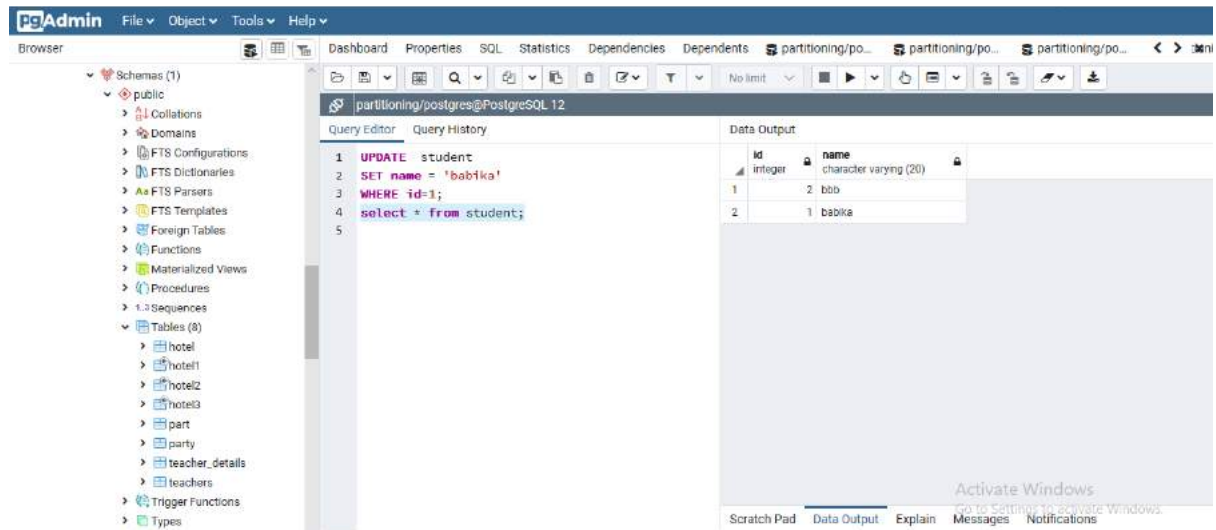
## 1.set search\_path



## 2.insert



### 3.update



The screenshot shows the pgAdmin interface with the 'Query Editor' tab active. The query being executed is:

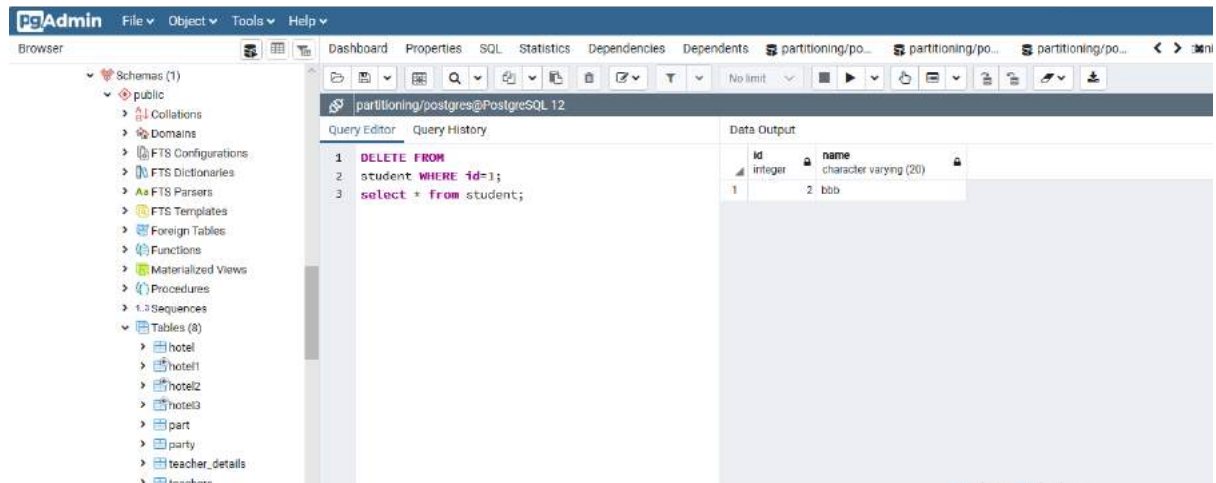
```
1 UPDATE student
2 SET name = 'babika'
3 WHERE id=1;
4 select * from student;
```

The 'Data Output' tab shows the results of the query:

id	name
1	bbb
2	babika

The interface also shows a 'Browser' pane on the left with a tree view of the database schema, including 'public' and 'Tables (8)'.

### 4.delete



The screenshot shows the pgAdmin interface with the 'Query Editor' tab active. The query being executed is:

```
1 DELETE FROM
2 student WHERE id=1;
3 select * from student;
```

The 'Data Output' tab shows the results of the query:

id	name
1	bbb

The interface also shows a 'Browser' pane on the left with a tree view of the database schema, including 'public' and 'Tables (8)'.

## 5.select

