

IT 214 DBMS

Lab 9

Prepared by: Group S6_T9

ID	Name
201901076	Utsav Ladani
201901090	Pandar Mayur
201901131	Bhavya Solanki
201901304	Dev Joshi

**Dhirubhai Ambani Institute of Information and Communication
Technology**

16 Nov, 2021

1)

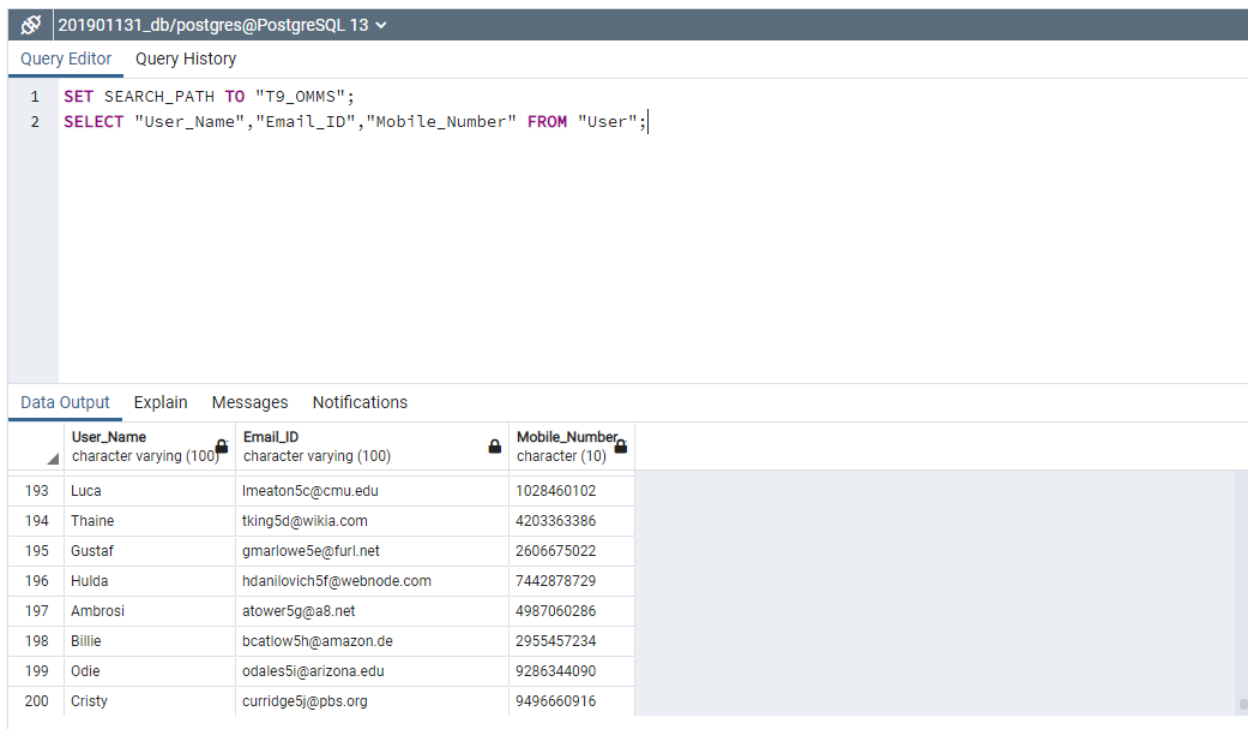
Plain English Query :

Show Details of Users (user name, email ID, phone number).

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
SELECT "User_Name", "Email_ID", "Mobile_Number" FROM "User";
```

Snapshot :



The screenshot shows a PostgreSQL query editor interface. At the top, the database name is '201901131_db/postgres@PostgreSQL 13'. Below the title bar, there are tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";  
2 SELECT "User_Name", "Email_ID", "Mobile_Number" FROM "User";
```

Below the query editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with 3 columns: 'User_Name', 'Email_ID', and 'Mobile_Number'. The table contains 8 rows of data, with row numbers 193 through 200 visible on the left. The 'User_Name' column is of type 'character varying (100)', 'Email_ID' is 'character varying (100)', and 'Mobile_Number' is 'character (10)'. The data is as follows:

	User_Name character varying (100)	Email_ID character varying (100)	Mobile_Number character (10)
193	Luca	lmeaton5c@cmu.edu	1028460102
194	Thaine	tking5d@wikia.com	4203363386
195	Gustaf	gmarlowe5e@furl.net	2606675022
196	Hulda	hdanilovich5f@webnode.com	7442878729
197	Ambrosi	atower5g@a8.net	4987060286
198	Billie	bcatlow5h@amazon.de	2955457234
199	Odie	odales5i@arizona.edu	9286344090
200	Cristy	curridge5j@pbs.org	9496660916

Result Contains : 200 Tuples.

2)

Plain English Query :

Show Names of the All Admins.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
SELECT "Admin_Name" FROM "Admin";
```

Snapshot :

The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is set to '201901131_db/postgres@PostgreSQL 13'. Below this, the 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";  
2 SELECT "Admin_Name" FROM "Admin";
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with the following structure:

	Admin_Name character varying (100)
1	Utsav
2	Mayur
3	Bhavya
4	Dev

At the bottom right of the interface, a green status bar indicates: '✓ Successfully run. Total query runtime: 631 msec. 4 rows affected.'

Result Contains : 4 Tuples.

3)

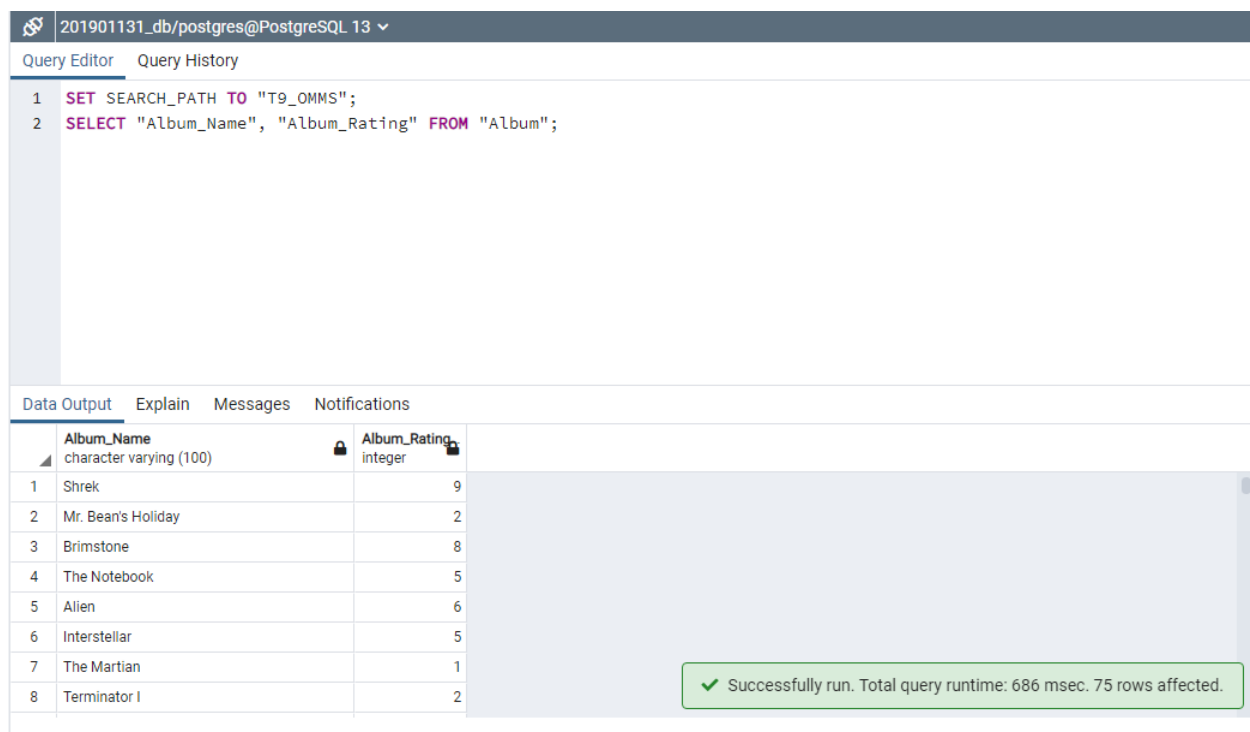
Plain English Query :

Show Rating of the Albums with Album Name.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
SELECT "Album_Name", "Album_Rating" FROM "Album";
```

Snapshot :



	Album_Name character varying (100)	Album_Rating integer
1	Shrek	9
2	Mr. Bean's Holiday	2
3	Brimstone	8
4	The Notebook	5
5	Alien	6
6	Interstellar	5
7	The Martian	1
8	Terminator I	2

✓ Successfully run. Total query runtime: 686 msec. 75 rows affected.

Result Contains : 75 Tuples.

4)

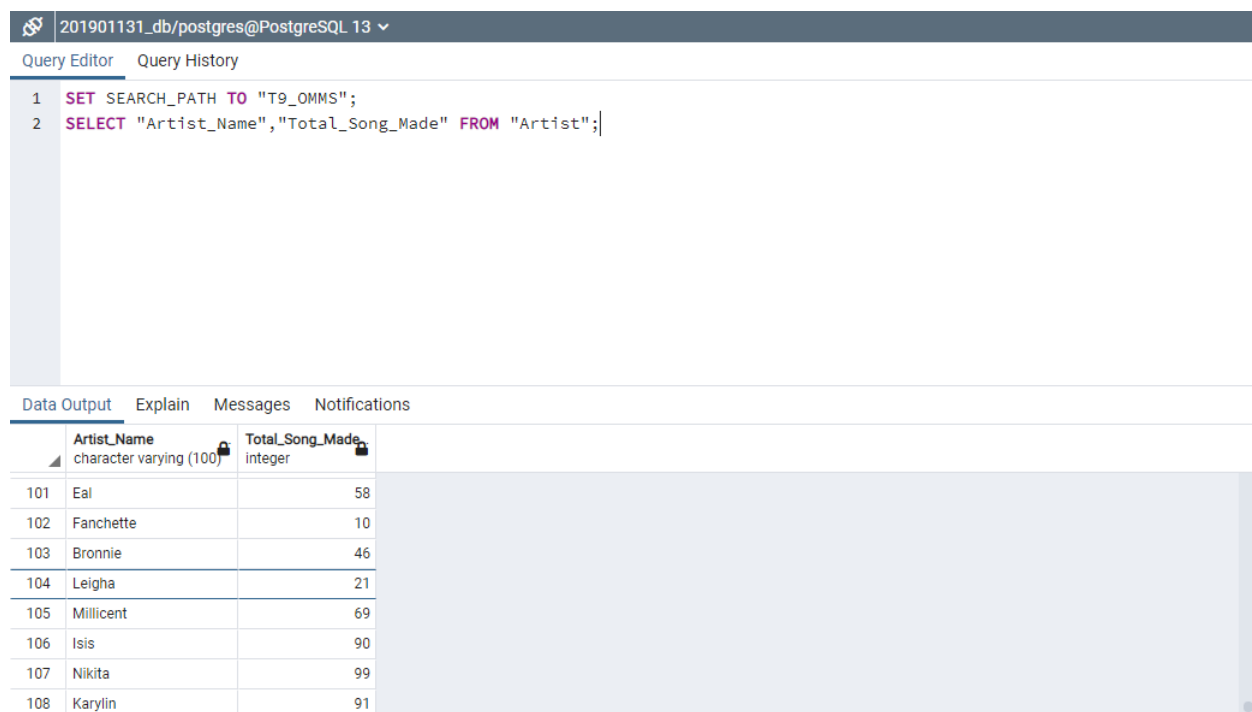
Plain English Query :

Show Count of How many songs an Artist has made along with Artist Name.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
SELECT "Artist_Name", "Total_Song_Made" FROM "Artist";
```

Snapshot :



The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is '201901131_db/postgres@PostgreSQL 13'. Below the connection bar, there are tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";  
2 SELECT "Artist_Name", "Total_Song_Made" FROM "Artist";
```

Below the query editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing the results of the query in a table format. The table has two columns: 'Artist_Name' (character varying (100)) and 'Total_Song_Made' (integer). The results are as follows:

	Artist_Name	Total_Song_Made
101	Eal	58
102	Fanchette	10
103	Bronnie	46
104	Leigha	21
105	Millicent	69
106	Isis	90
107	Nikita	99
108	Karylin	91

Result Contains : 108 Tuples.

5)

Plain English Query :

Count Total Number Of Production Companies.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
SELECT COUNT(*) FROM "Production_Company";
```

Snapshot :

The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is set to '201901131_db/postgres@PostgreSQL 13'. Below this, the 'Query Editor' tab is active, displaying two lines of SQL code: '1 SET SEARCH_PATH TO "T9_OMMS";' and '2 SELECT COUNT(*) FROM "Production_Company";'. The 'Data Output' tab is also visible, showing a single row of results with the column 'count bigint' and the value '36'. A green status bar at the bottom right indicates 'Successfully run. Total query runtime: 314 msec. 1 rows affected.'

	count bigint
1	36

Result Contains : Count = 36 (1 Tuple).

6)

Plain English Query :

List Name of All Premium Songs

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
SELECT "Music_Name" FROM "Music"  
WHERE "Premium_Information" = 'TRUE';
```

Snapshot :

The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is set to '201901131_db/postgres@PostgreSQL 13'. Below this, the 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";  
2 SELECT "Music_Name" FROM "Music"  
3 WHERE "Premium_Information" = 'TRUE';
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with the following columns:

	Music_Name character varying (100)
1	Flashdance...What A Feeling
2	Mentiroso
3	Every Breath You Take
4	Cheri cheri lady
5	Shadow Dancing
6	Foolish Games
7	Tonights The Night
8	Cant touch this

A green status bar at the bottom right of the results table indicates: 'Successfully run. Total query runtime: 368 msec. 256 rows affected.'

Result Contains : 256 Tuples.

7)

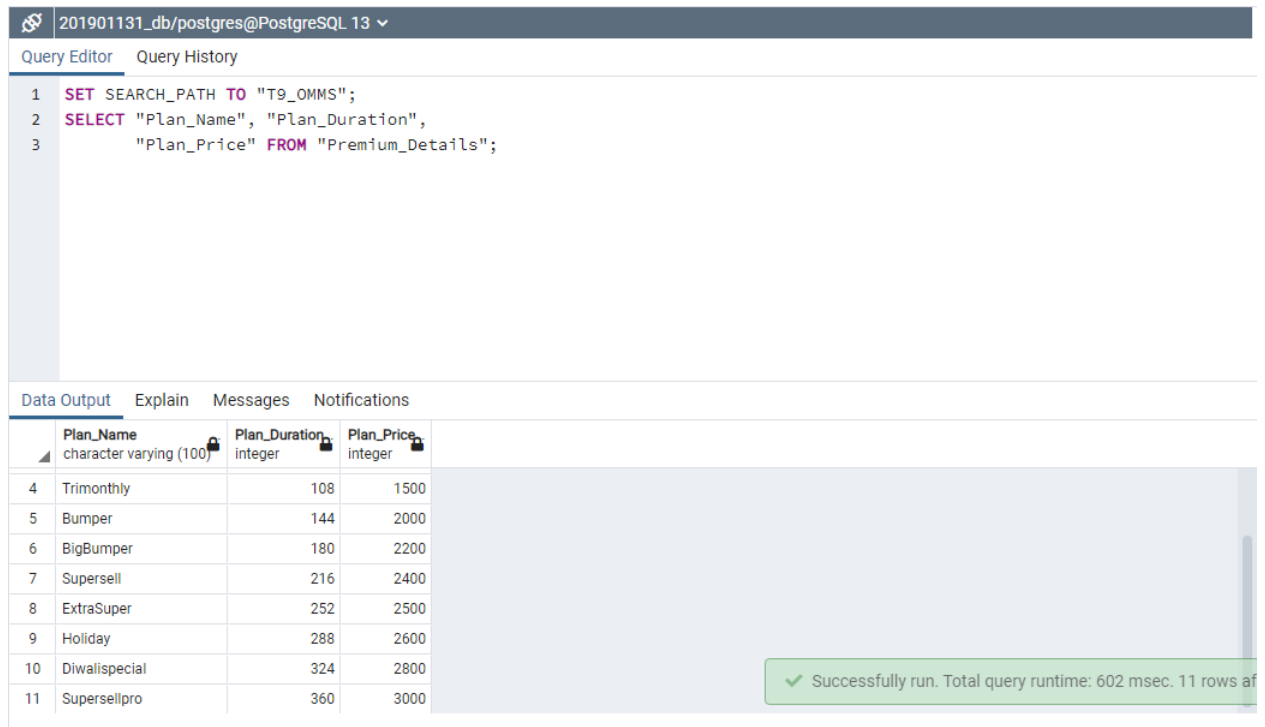
Plain English Query :

Show Details of all the Plans.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
SELECT "Plan_Name", "Plan_Duration",
       "Plan_Price" FROM "Premium_Details";
```

Snapshot :



The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is '201901131_db/postgres@PostgreSQL 13'. Below the connection bar, there are tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";
2 SELECT "Plan_Name", "Plan_Duration",
3        "Plan_Price" FROM "Premium_Details";
```

Below the query editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing the results of the query in a table format. The table has three columns: 'Plan_Name' (character varying (100)), 'Plan_Duration' (integer), and 'Plan_Price' (integer). The results are as follows:

	Plan_Name	Plan_Duration	Plan_Price
4	Trimonthly	108	1500
5	Bumper	144	2000
6	BigBumper	180	2200
7	Supersell	216	2400
8	ExtraSuper	252	2500
9	Holiday	288	2600
10	Diwalispecial	324	2800
11	Supersellpro	360	3000

At the bottom right of the table, there is a green status bar that reads: '✓ Successfully run. Total query runtime: 602 msec. 11 rows af'.

Result Contains : 11 Tuples.

8)

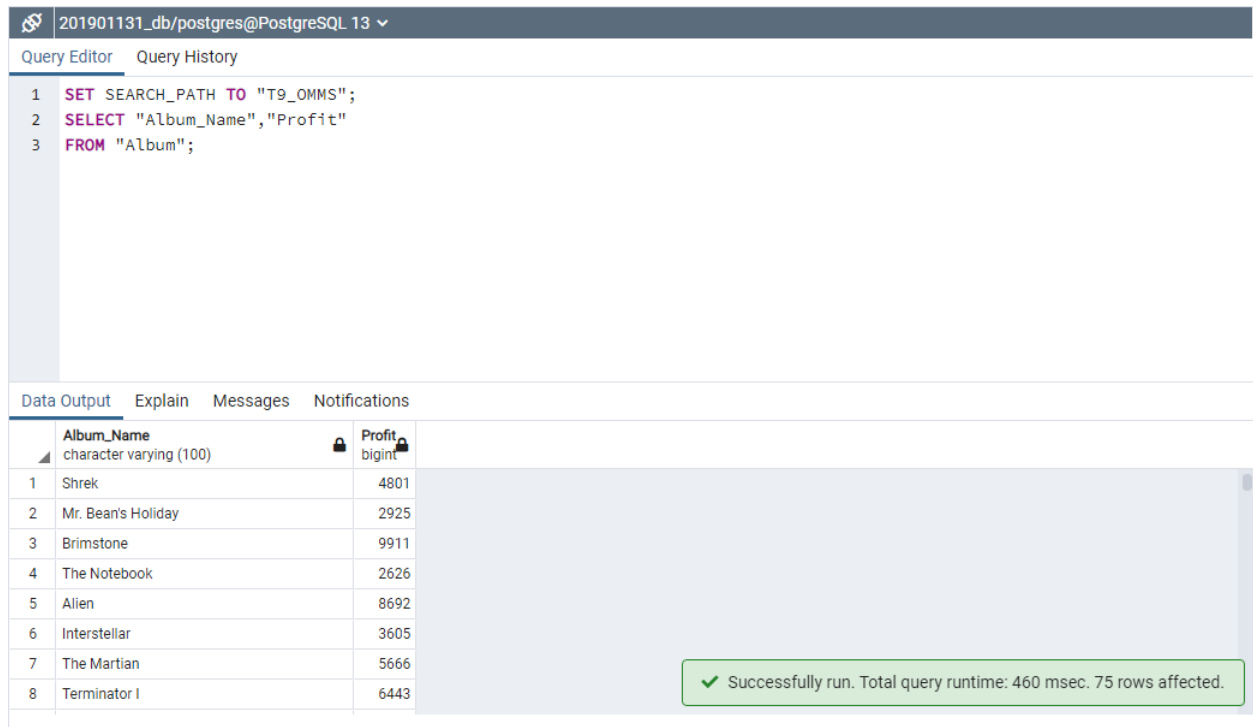
Plain English Query :

Show Money made by each Album.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
SELECT "Album_Name", "Profit"
FROM "Album";
```

Snapshot :



The screenshot displays a PostgreSQL query editor interface. At the top, the database connection is identified as '201901131_db/postgres@PostgreSQL 13'. Below this, the 'Query Editor' tab is active, showing the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";
2 SELECT "Album_Name", "Profit"
3 FROM "Album";
```

Below the query editor, the 'Data Output' tab is selected, displaying the results of the query in a table format. The table has two columns: 'Album_Name' (character varying (100)) and 'Profit' (bigint). The results are as follows:

	Album_Name	Profit
1	Shrek	4801
2	Mr. Bean's Holiday	2925
3	Brimstone	9911
4	The Notebook	2626
5	Alien	8692
6	Interstellar	3605
7	The Martian	5666
8	Terminator I	6443

A green status bar at the bottom right of the table indicates: '✓ Successfully run. Total query runtime: 460 msec. 75 rows affected.'

Result Contains : 75 Tuples.

9)

Plain English Query :

List all song names that non-premium users can listen.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
SELECT "Music_Name" FROM "Music"  
WHERE "Premium_Information" = 'False';
```

Snapshot :

The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is set to '201901131_db/postgres@PostgreSQL 13'. Below this, the 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";  
2 SELECT "Music_Name" FROM "Music"  
3 WHERE "Premium_Information" = 'False';
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with the following columns: 'Music_Name' (character varying (100)). The table contains 8 rows of data:

	Music_Name
1	Demons
2	Familiar
3	Livin la vida loca
4	Corazon
5	We are young
6	Eye Of The Tiger
7	I Just Want To Be Your Everything
8	Corazon

At the bottom right of the interface, a green status bar indicates: 'Successfully run. Total query runtime: 435 msec. 244 rows affected.'

Result Contains : 244 Tuples.

10)

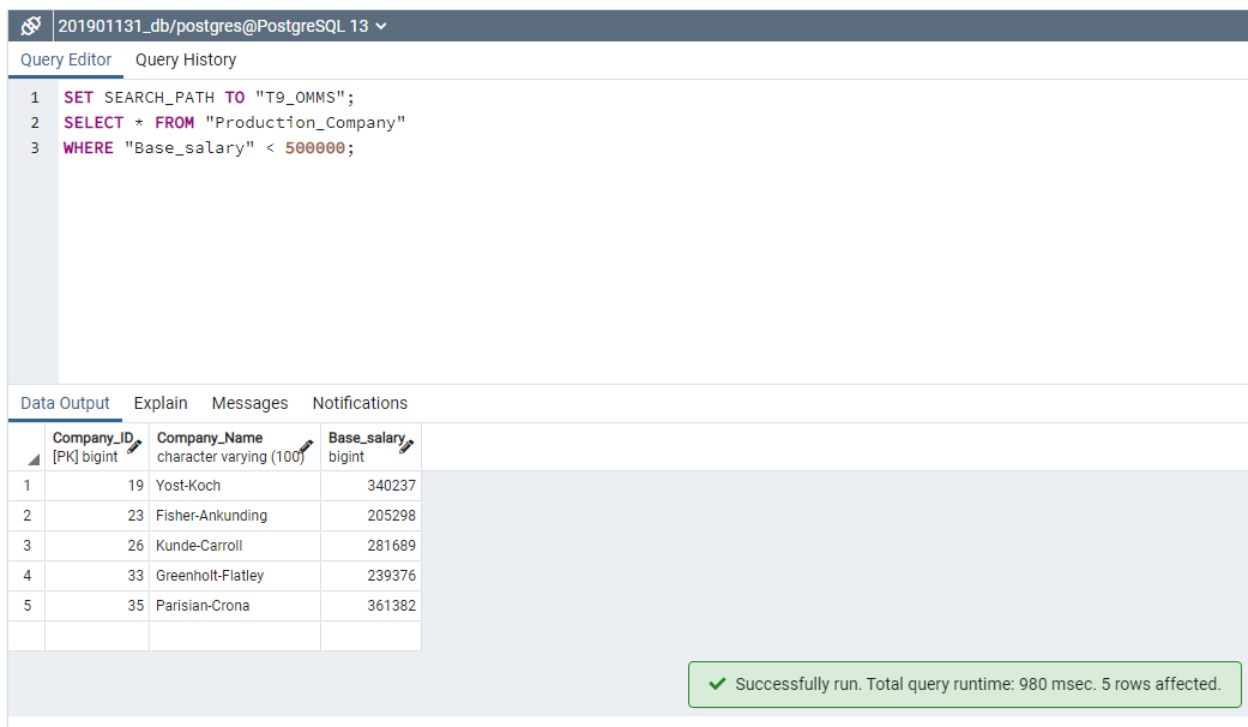
Plain English Query :

List all the Company names which give the base salary less than 500000 .

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
SELECT * FROM "Production_Company"  
WHERE "Base_salary" < 500000;
```

Snapshot :



201901131_db/postgres@PostgreSQL 13 ▾

Query Editor Query History

```
1 SET SEARCH_PATH TO "T9_OMMS";  
2 SELECT * FROM "Production_Company"  
3 WHERE "Base_salary" < 500000;
```

Data Output Explain Messages Notifications

	Company_ID [PK] bigint	Company_Name character varying (100)	Base_salary bigint
1	19	Yost-Koch	340237
2	23	Fisher-Ankunding	205298
3	26	Kunde-Carroll	281689
4	33	Greenholt-Flatley	239376
5	35	Parisian-Crona	361382

✓ Successfully run. Total query runtime: 980 msec. 5 rows affected.

Result Contains : 5 Tuples.

11)

Plain English Query :

List all the Music names with English language.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
SELECT "Music_Name" FROM "Music"  
WHERE "Music_Language"='English';
```

Snapshot :

The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is set to '201901131_db/postgres@PostgreSQL 13'. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";  
2 SELECT "Music_Name" FROM "Music"  
3 WHERE "Music_Language"='English';
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with the following columns and data:

	Music_Name character varying (100)
1	How Do I Live
2	I want to break free
3	I wanna dance with somebody
4	Flashdance...What A Feeling
5	Physical
6	Demons
7	More than you know

At the bottom right of the interface, a green status bar indicates: 'Successfully run. Total query runtime: 788 msec. 7 rows affected.'

Result Contains : 7 Tuples.

12)

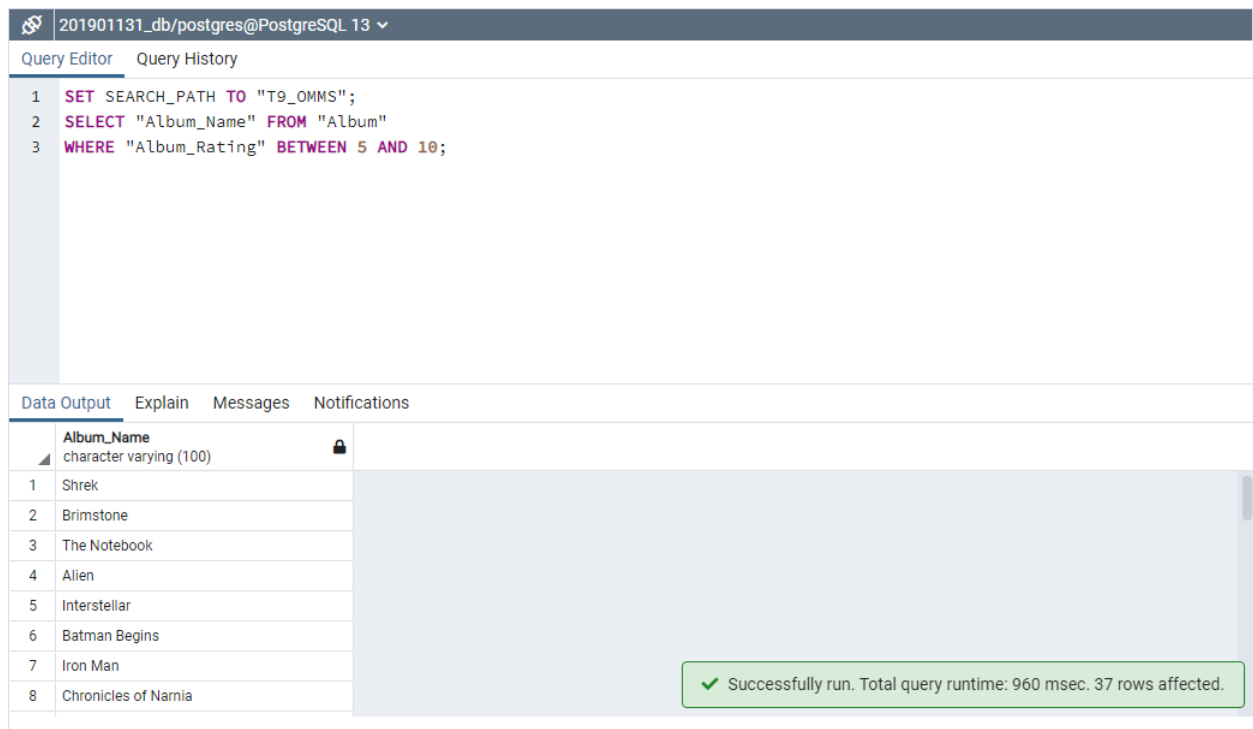
Plain English Query :

List all the album names which rating between 6 to 10.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
SELECT "Album_Name" FROM "Album"  
WHERE "Album_Rating" BETWEEN 5 AND 10;
```

Snapshot :



The screenshot shows a PostgreSQL query editor interface. The top bar indicates the database is '201901131_db/postgres@PostgreSQL 13'. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";  
2 SELECT "Album_Name" FROM "Album"  
3 WHERE "Album_Rating" BETWEEN 5 AND 10;
```

Below the query editor, the 'Data Output' tab is selected, showing the results of the query. The results are displayed in a table with the following columns:

Album_Name
Shrek
Brimstone
The Notebook
Alien
Interstellar
Batman Begins
Iron Man
Chronicles of Narnia

A green notification box at the bottom right of the results area states: 'Successfully run. Total query runtime: 960 msec. 37 rows affected.'

Result Contains : 37 Tuples.

13)

Plain English Query :

Find the average base salary of all the companies.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
SELECT AVG("Base_salary") FROM "Production_Company";
```

Snapshot :

The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is '201901131_db/postgres@PostgreSQL 13'. Below this, the 'Query Editor' tab is active, displaying two lines of SQL code: '1 SET SEARCH_PATH TO "T9_OMMS";' and '2 SELECT AVG("Base_salary") FROM "Production_Company";'. Below the query editor, the 'Data Output' tab is active, showing a single row of results. The first column is labeled 'avg' with a data type of 'numeric'. The value in the row is '2293146.11111111111111'. At the bottom right of the interface, a green status bar indicates 'Successfully run. Total query runtime: 760 msec. 1 rows affected.'

	avg numeric
1	2293146.11111111111111

Result Contains : 1 Tuple.

14)

Plain English Query :

Find out the average profit of albums.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
SELECT AVG("Profit") FROM "Album";
```

Snapshot :

The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is set to '201901131_db/postgres@PostgreSQL 13'. Below this, the 'Query Editor' tab is active, displaying two lines of SQL code: '1 SET SEARCH_PATH TO "T9_OMMS";' and '2 SELECT AVG("Profit") FROM "Album";'. The 'Data Output' tab is also visible, showing a single row of results with the column 'avg numeric' and the value '5447.3066666666666667'. A green status bar at the bottom right indicates 'Successfully run. Total query runtime: 610 msec. 1 rows affected.'

avg numeric
5447.3066666666666667

Result Contains : 1 Tuple.

15)

Plain English Query :

List the music names in ascending order of likes.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
SELECT "Music_Name" FROM "Music"  
ORDER BY "Likes";
```

Snapshot :

The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is '201901131_db/postgres@PostgreSQL 13'. Below the connection bar, there are tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";  
2 SELECT "Music_Name" FROM "Music"  
3 ORDER BY "Likes";
```

Below the query editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with the following data:

	Music_Name character varying (100)
1	Hero
2	You Light Up My Life
3	You are the one that I want
4	Un-Break My Heart
5	(Everything I Do) I Do It For You
6	Por una cabeza
7	Dance dance
8	The Twist

At the bottom right of the 'Data Output' tab, there is a green status message: '✓ Successfully run. Total query runtime: 1 secs 235 msec. 500 rows affected.'

Result Contains : 500 Tuples.

16)

Plain English Query :

List the music names in ascending order of views.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
SELECT "Music_Name" FROM "Music"  
ORDER BY "Views";
```

Snapshot :

The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is '201901131_db/postgres@PostgreSQL 13'. Below the connection bar, there are two tabs: 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";  
2 SELECT "Music_Name" FROM "Music"  
3 ORDER BY "Views";
```

Below the query editor, there are four tabs: 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with the results of the query. The table has a single column 'Music_Name' with a data type of 'character varying (100)'. The results are as follows:

	Music_Name
1	Bette Davis Eyes
2	Hey Jude
3	Cheap thrills
4	Strongest
5	Mad Love
6	Yeah!
7	Demons
8	Le Freak

At the bottom right of the 'Data Output' tab, there is a green status box that reads: '✓ Successfully run. Total query runtime: 784 msec. 500 rows affected.'

Result Contains : 500 Tuples.

17)


Plain English Query :

List the albums in descending order of rating.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
SELECT * FROM "Album"
ORDER BY "Album_Rating" DESC;
```







Snapshot :

 201901131_db/postgres@PostgreSQL 13 ▾

Query Editor Query History

```
1 SET SEARCH_PATH TO "T9_OMMS";
2 SELECT * FROM "Album"
3     ORDER BY "Album_Rating" DESC;
```

Data Output Explain Messages Notifications

	 Album_ID [PK] bigint	 Album_Name character varying (100)	 Album_Rating integer	 Profit bigint	 Artist_ID bigint	 Company_ID bigint
1	35	Brimstone	9	1699	92	32
2	49	The Wolf Of Wallstreet	9	2496	46	33
3	39	Octav	9	1047	24	17
4	1	Shrek	9	4801	49	3
5	23	Love	9	8719	102	27
6	74	Shrek	9	2271	4	18
7	48	The Dark Tower	9	9846	8	2
8	72	The Martian	9	3278	8	2

✓ Successfully run. Total query runtime: 322 msec. 75 rows affected.

Result Contains : 75 Tuples.

18)

Plain English Query :

List the company in descending order of base salary.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
SELECT * FROM "Production_Company"  
ORDER BY "Base_salary" DESC;
```

Snapshot :

The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is '201901131_db/postgres@PostgreSQL 13'. Below the connection bar, there are tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";  
2 SELECT * FROM "Production_Company"  
3 ORDER BY "Base_salary" DESC;
```

Below the query editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing the results of the query in a table format. The table has three columns: 'Company_ID' (PK) bigint, 'Company_Name' character varying (100), and 'Base_salary' bigint. The results are ordered by 'Base_salary' in descending order.

	Company_ID [PK] bigint	Company_Name character varying (100)	Base_salary bigint
1	10	Zulauf-Abernathy	4882455
2	9	Lynch Inc	4756547
3	27	Watsica-Wehner	4634836
4	17	Prohaska-Grant	4603867
5	24	Farrell, Stoltenberg and Barrows	4484165
6	18	Zemlak and Sons	4388603
7	28	Quigley LLC	4044463
8	15	Mante, Wolf and Moore	3885691

At the bottom right of the 'Data Output' tab, there is a green box with a checkmark and the text: 'Successfully run. Total query runtime: 1 secs 84 msec. 36 rows affected.'

Result Contains : 36 Tuples.

19)

Plain English Query :

List the User in alphabetical order.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
SELECT * FROM "User"  
ORDER BY "User_Name";
```

Snapshot :

201901131_db/postgres@PostgreSQL 13 ▾							
Query Editor Query History							
<pre>1 SET SEARCH_PATH TO "T9_OMMS"; 2 SELECT * FROM "User" 3 ORDER BY "User_Name";</pre>							
Data Output Explain Messages Notifications							
	User_ID [PK] bigint	User_Name character varying (100)	Password character varying (128)	Email_ID character varying (100)	Mobile_Number character (10)	Plan_ID integer	User_Type character varying (60)
1	44	Ahmed	NZ8t7n	allince17@people.com.cn	3744610498	1	Non_Premium
2	129	Alf	XHw7u4ne	ashovelbottom3k@hostgator.com	3974145718	1	Non_Premium
3	68	Allsha	y2tjwOdjN	acesco1v@ed.gov	6526042701	1	Non_Premium
4	197	Ambrosi	9MHlocbC0m	atower5g@a8.net	4987060286	1	Non_Premium
5	186	Andy	8vB01HrFV	amerrgen55@illinois.edu	5459862279	1	Non_Premium
6	180	Angelica	hfhX8T564mr9	arussam4z@flavors.me	4056926694	1	Non_Premium
7	161	Anny	H8tANDTzg	aoliphard4g@xrea			
8	98	Arabele	ulbL26CyKS	abond2p@usatod			
✓ Successfully run. Total query runtime: 1 secs 433 msec. 200 rows affected.							

Result Contains : 200 Tuples.

20)

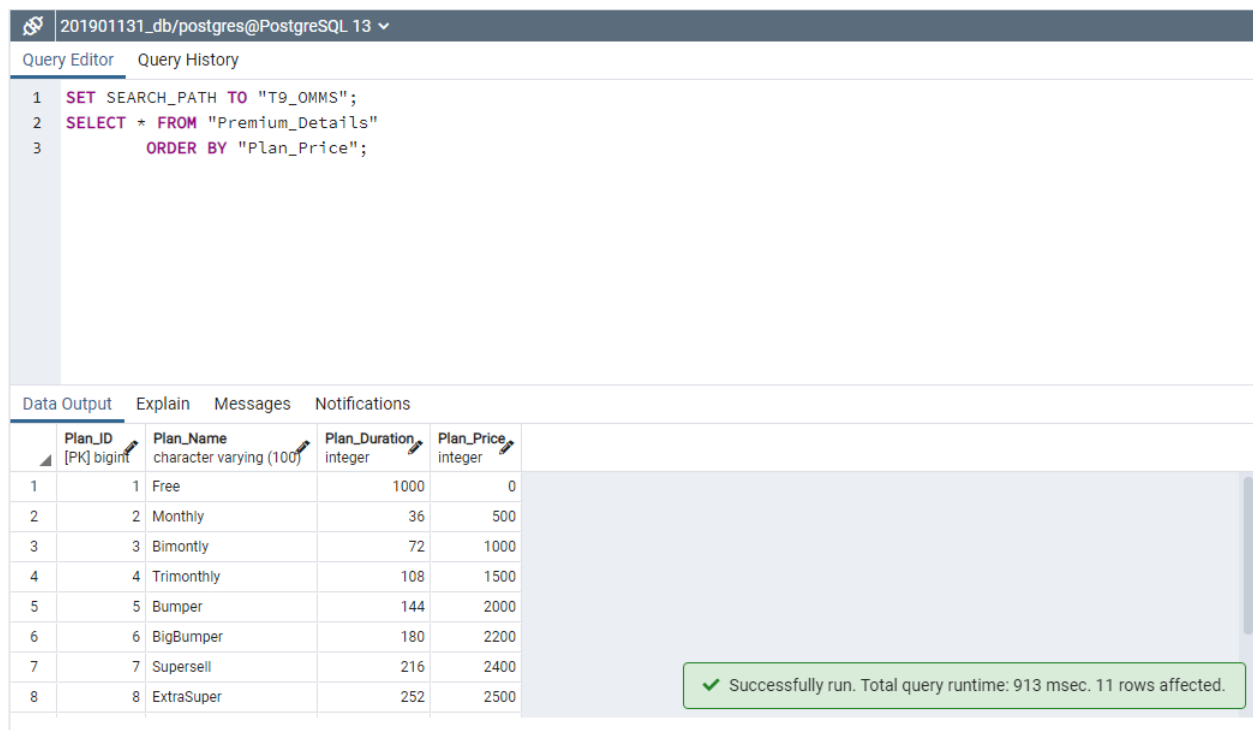
Plain English Query :

List the Plan in ascending order of plan price.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
SELECT * FROM "Premium_Details"  
ORDER BY "Plan_Price";
```

Snapshot :



The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is '201901131_db/postgres@PostgreSQL 13'. Below this are tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";  
2 SELECT * FROM "Premium_Details"  
3 ORDER BY "Plan_Price";
```

Below the query editor are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with 4 columns: 'Plan_ID [PK] bigint', 'Plan_Name character varying (100)', 'Plan_Duration Integer', and 'Plan_Price Integer'. The table contains 8 rows of data:

Plan_ID [PK] bigint	Plan_Name character varying (100)	Plan_Duration Integer	Plan_Price Integer
1	Free	1000	0
2	Monthly	36	500
3	Bimonthly	72	1000
4	Trimonthly	108	1500
5	Bumper	144	2000
6	BigBumper	180	2200
7	Supersell	216	2400
8	ExtraSuper	252	2500

At the bottom right of the 'Data Output' tab, there is a green status box that reads: '✓ Successfully run. Total query runtime: 913 msec. 11 rows affected.'

Result Contains : 11 Tuples.

21)

Plain English Query :

List the Plan in descending order of plan duration.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
SELECT * FROM "Premium_Details"
ORDER BY "Plan_Duration" DESC
```

Snapshot :

The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is set to '201901131_db/postgres@PostgreSQL 13'. Below this, the 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";
2 SELECT * FROM "Premium_Details"
3     ORDER BY "Plan_Duration" DESC;
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with the following columns: Plan_ID, Plan_Name, Plan_Duration, and Plan_Price. The table contains 11 rows of data, sorted by Plan_Duration in descending order.

Plan_ID	Plan_Name	Plan_Duration	Plan_Price
1	Free	1000	0
2	Supersellpro	360	3000
3	Diwalispecial	324	2800
4	Holiday	288	2600
5	ExtraSuper	252	2500
6	Supersell	216	2400
7	BigBumper	180	2200
8	Bumper	144	2000

At the bottom right of the results area, a green message box indicates: 'Successfully run. Total query runtime: 1 secs 124 msec. 11 rows affected.'

Result Contains : 11 Tuples.

22)

Plain English Query :

List the albums which have maximum rating.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
WITH Temptable(mvalue) AS
(SELECT MAX("Album_Rating") FROM "Album")
SELECT * FROM "Album",Temptable WHERE
"Album"."Album_Rating" = mvalue;
```

Snapshot :

201901131_db/postgres@PostgreSQL 13							
Query Editor Query History							
<pre>1 SET SEARCH_PATH TO "T9_OMMS"; 2 WITH Temptable(mvalue) AS 3 (SELECT MAX("Album_Rating") FROM "Album") 4 SELECT * FROM "Album",Temptable WHERE 5 "Album"."Album_Rating" = mvalue; 6</pre>							
Data Output Explain Messages Notifications							
	Album_ID [PK] bigint	Album_Name character varying (100)	Album_Rating integer	Profit bigint	Artist_ID bigint	Company_ID bigint	mvalue integer
1	1	Shrek	9	4801	49	3	9
2	18	Limitless	9	2897	30	19	9
3	23	Love	9	8719	102	27	9
4	35	Brimstone	9	1699	92	32	9
5	39	Octav	9	1047	24	17	9
6	48	The Dark Tower	9	9846	89	31	9
7	49	The Wolf Of Wallstreet	9	2496	46	33	
8	72	The Martian	9	3278	89	19	

✓ Successfully run. Total query runtime: 363 msec. 9 rows affected.

Result Contains : 9 Tuples.

23)

Plain English Query :

List songs which have more likes than average likes.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
WITH TEMPTABLE(colom) AS
(SELECT AVG("Likes") FROM "Music")
SELECT * FROM "Music",TEMPTABLE WHERE
"Music"."Likes" > colom;
```

Snapshot :

201901131_db/postgres@PostgreSQL 13 ▾										
Query Editor Query History										
<pre>1 SET SEARCH_PATH TO "T9_OMMS"; 2 WITH TEMPTABLE(colom) AS 3 (SELECT AVG("Likes") FROM "Music") 4 SELECT * FROM "Music",TEMPTABLE WHERE 5 "Music"."Likes" > colom;</pre>										
Data Output Explain Messages Notifications										
	Music_ID [PK] bigint	Music_Name character varying (100)	Music_Language character varying (100)	Premium_Information boolean	Views bigint	Likes bigint	Music_Type character varying (60)	Artist_ID bigint	Album_ID bigint	
1	6	Cheri cheri lady	Polish	true	5563525	95258941	Dubstep	58		
2	7	Livin la vida loca	French	false	62721593	55140963	Jazz	103		
3	10	We are young	Hindi	false	65607905	75099000	Jazz	15		
4	14	Cant touch this	Fula	true	71923189	75040357	Baroque	88		
5	15	Foolish Games	Portuguese	true	40036595	52711666	Hip-Hop	41		
6	16	I Just Want To Be Your Everything	Hakka	false	8842410	92868697	Baroque	104		
7	17	Sorry seems to be the hardest word	Greek	true	22864718	92864675	Pop	38		
8	18	Corazon	Malaveian	false	22864718	92864675	Pop	38		

✓ Successfully run. Total query runtime: 431 msec. 255 rows affected.

Result Contains : 255 Tuples.

24)

Plain English Query :

List songs which have more views than average views.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
WITH TEMPTABLE(colom) AS
(SELECT AVG("Views") FROM "Music")
SELECT * FROM "Music",TEMPTABLE WHERE
"Music"."Views" > colom;
```

Snapshot :

201901131_db/postgres@PostgreSQL 13 ▾										
Query Editor Query History										
<pre>1 SET SEARCH_PATH TO "T9_OMMS"; 2 WITH TEMPTABLE(colom) AS 3 (SELECT AVG("Views") FROM "Music") 4 SELECT * FROM "Music",TEMPTABLE WHERE 5 "Music"."Views" > colom;</pre>										
Data Output Explain Messages Notifications										
	Music_ID [PK] bigint	Music_Name character varying (100)	Music_Language character varying (100)	Premium_Information boolean	Views bigint	Likes bigint	Music_Type character varying (60)	Artist_ID bigint	Album_ID bigint	
1	4	Demons	Serbo-Croatian	false	70809651	36163732	Latin	85	6	
2	5	Familiar	Polish	false	70929967	28086316	Techno	56		
3	7	Livin la vida loca	French	false	62721593	55140963	Jazz	103	4	
4	10	We are young	Hindi	false	65607905	75099000	Jazz	15	4	
5	14	Cant touch this	Fula	true	71923189	75040357	Baroque	88	5	
6	18	Corazon	Malaysian	false	76695614	69066986	Techno	101	3	
7	19	Cest la vie	Uyghur	true						
8	20	We are young	Turkish	true						
✓ Successfully run. Total query runtime: 646 msec. 247 rows affected.										

Result Contains : 247 Tuples.

25)

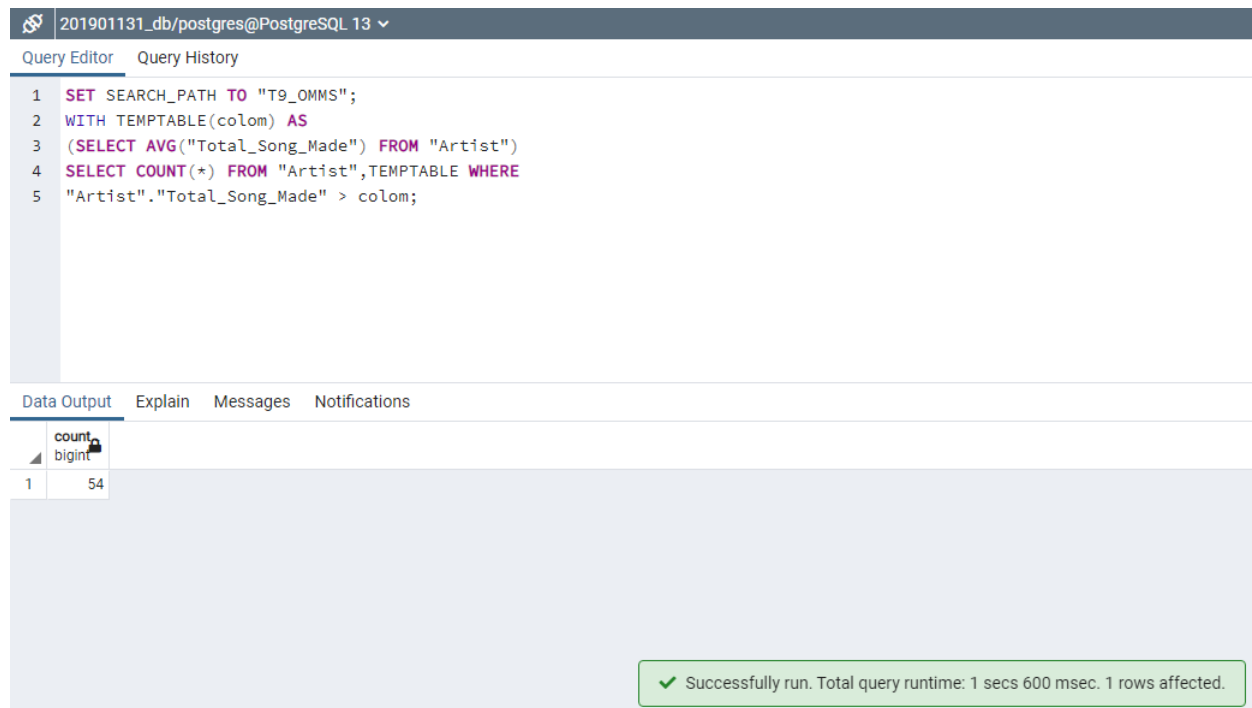
Plain English Query :

List the number of artists which made the songs greater than average songs.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
WITH TEMPTABLE(colom) AS  
(SELECT AVG("Total_Song_Made") FROM "Artist")  
SELECT COUNT(*) FROM "Artist",TEMPTABLE WHERE  
"Artist"."Total_Song_Made" > colom;
```

Snapshot :



The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is set to '201901131_db/postgres@PostgreSQL 13'. Below this, the 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";  
2 WITH TEMPTABLE(colom) AS  
3 (SELECT AVG("Total_Song_Made") FROM "Artist")  
4 SELECT COUNT(*) FROM "Artist",TEMPTABLE WHERE  
5 "Artist"."Total_Song_Made" > colom;
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with two columns: 'count' and 'bigint'. The first row shows a count of 54.

	count	bigint
1	54	

At the bottom right of the interface, a green status bar indicates: 'Successfully run. Total query runtime: 1 secs 600 msec. 1 rows affected.'

Result Contains : 1 Tuple.

26)

Plain English Query :

List the name of albums which have profit greater than average profit.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
WITH TEMPTABLE(colom) AS  
(SELECT AVG("Profit") FROM "Album")  
SELECT "Album_Name" FROM "Album",TEMPTABLE WHERE  
"Album"."Profit" > colom;
```

Snapshot :

The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is '201901131_db/postgres@PostgreSQL 13'. Below this are tabs for 'Query Editor' and 'Query History'. The query editor contains the following SQL code:

```
1 SET SEARCH_PATH TO "T9_OMMS";  
2 WITH TEMPTABLE(colom) AS  
3 (SELECT AVG("Profit") FROM "Album")  
4 SELECT "Album_Name" FROM "Album",TEMPTABLE WHERE  
5 "Album"."Profit" > colom;
```

Below the query editor are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with the following data:

	Album_Name character varying (100)
1	Brimstone
2	Alien
3	The Martian
4	Terminator I
5	Fantastic Beasts and Where To Find Them
6	Avatar
7	Casablanca
8	Iron Man

At the bottom right of the 'Data Output' tab, there is a green status box that says: '✓ Successfully run. Total query runtime: 578 msec. 36 rows affected.'

Result Contains : 36 Tuples.

27)

Plain English Query :

List the Companies which have a base salary greater than average base salary.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
WITH TEMPTABLE(colom) AS
(SELECT AVG("Base_salary") FROM "Production_Company")
SELECT "Company_Name" FROM "Production_Company",TEMPTABLE WHERE
"Production_Company"."Base_salary" > colom;
```

Snapshot :

The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is '201901131_db/postgres@PostgreSQL 13'. Below the connection bar, there are tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";
2 WITH TEMPTABLE(colom) AS
3 (SELECT AVG("Base_salary") FROM "Production_Company")
4 SELECT "Company_Name" FROM "Production_Company",TEMPTABLE WHERE
5 "Production_Company"."Base_salary" > colom;
```

Below the query editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with the following columns: 'Company_Name' (character varying (100)). The table contains 8 rows of data:

Company_Name
Lynch Inc
Zulauf-Abernathy
Heidenreich-Murphy
Rippin-Shields
Hammes and Sons
Conroy, Kilback and Kreiger
Mante, Wolf and Moore
Prohaska-Grant

At the bottom right of the 'Data Output' tab, there is a green status box that says: '✓ Successfully run. Total query runtime: 1 secs 184 msec. 17 rows affected.'

Result Contains : 17 Tuples.

28)

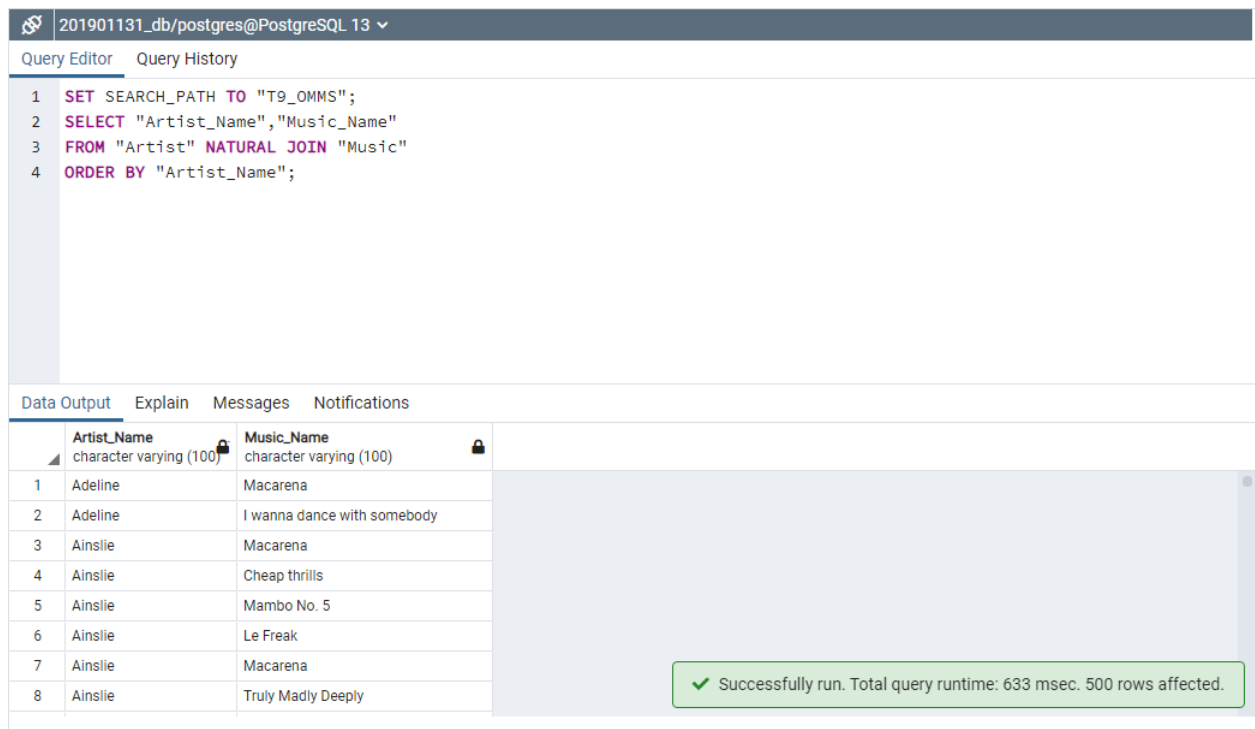
Plain English Query :

List the Artist name along with the song name he has made and order them by Artist name.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
SELECT "Artist_Name", "Music_Name"
FROM "Artist" NATURAL JOIN "Music"
ORDER BY "Artist_Name";
```

Snapshot :



201901131_db/postgres@PostgreSQL 13

Query Editor Query History

```
1 SET SEARCH_PATH TO "T9_OMMS";
2 SELECT "Artist_Name", "Music_Name"
3 FROM "Artist" NATURAL JOIN "Music"
4 ORDER BY "Artist_Name";
```

Data Output Explain Messages Notifications

	Artist_Name character varying (100)	Music_Name character varying (100)
1	Adeline	Macarena
2	Adeline	I wanna dance with somebody
3	Ainslie	Macarena
4	Ainslie	Cheap thrills
5	Ainslie	Mambo No. 5
6	Ainslie	Le Freak
7	Ainslie	Macarena
8	Ainslie	Truly Madly Deeply

✓ Successfully run. Total query runtime: 633 msec. 500 rows affected.

Result Contains : 500 Tuples.

29)

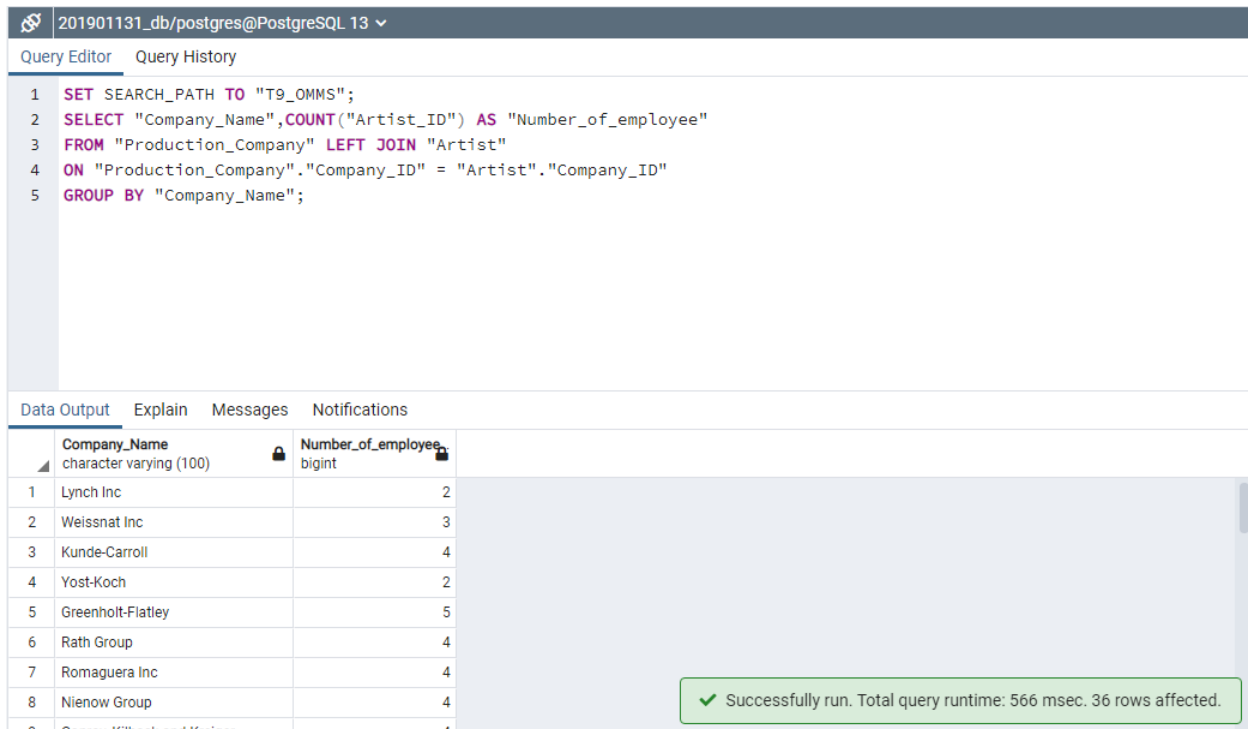
Plain English Query :

List the company name along with number of employees.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
SELECT "Company_Name",COUNT("Artist_ID") AS "Number_of_employee"
FROM "Production_Company" LEFT JOIN "Artist"
ON "Production_Company"."Company_ID" = "Artist"."Company_ID"
GROUP BY "Company_Name";
```

Snapshot :



201901131_db/postgres@PostgreSQL 13

Query Editor Query History

```
1 SET SEARCH_PATH TO "T9_OMMS";
2 SELECT "Company_Name",COUNT("Artist_ID") AS "Number_of_employee"
3 FROM "Production_Company" LEFT JOIN "Artist"
4 ON "Production_Company"."Company_ID" = "Artist"."Company_ID"
5 GROUP BY "Company_Name";
```

Data Output Explain Messages Notifications

	Company_Name character varying (100)	Number_of_employee bigint
1	Lynch Inc	2
2	Weissnat Inc	3
3	Kunde-Carroll	4
4	Yost-Koch	2
5	Greenholt-Flatley	5
6	Rath Group	4
7	Romaguera Inc	4
8	Nienow Group	4
9	Conroy Kilback and Kriger	4

✓ Successfully run. Total query runtime: 566 msec. 36 rows affected.

Result Contains : 36 Tuples.

30)

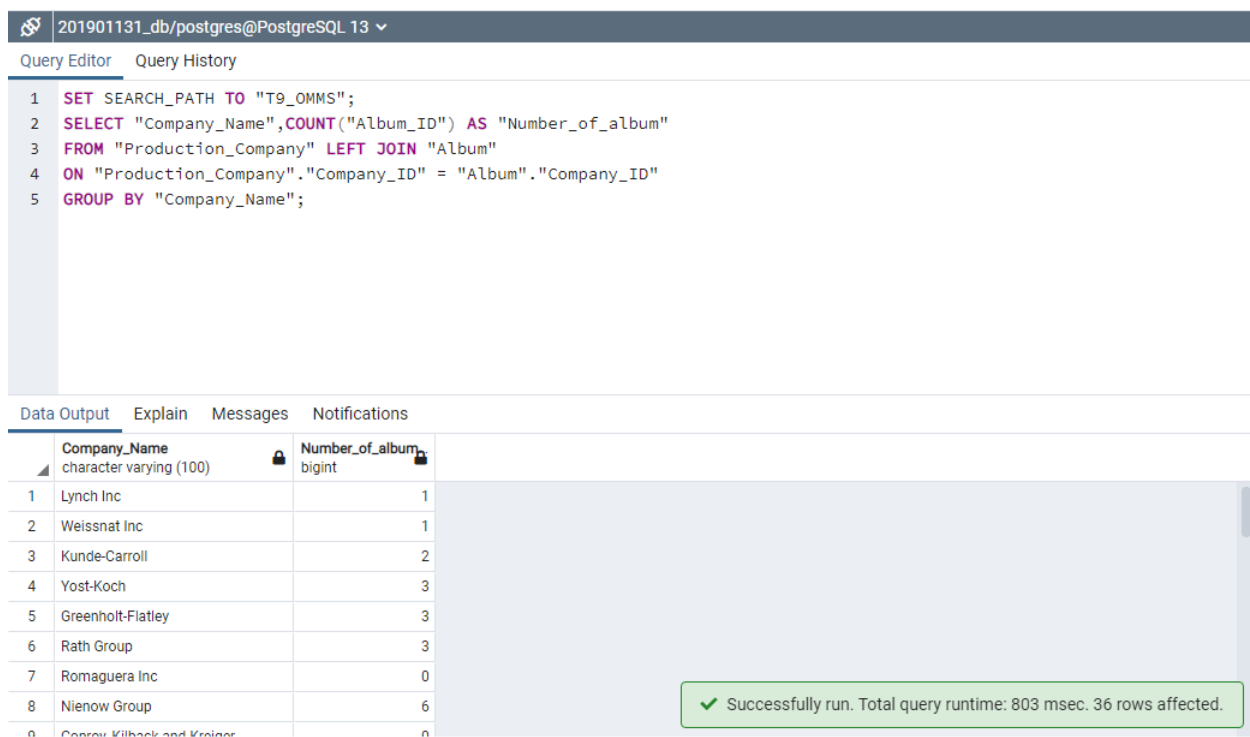
Plain English Query :

List the company name along with the number of albums.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
SELECT "Company_Name",COUNT("Album_ID") AS "Number_of_album"
FROM "Production_Company" LEFT JOIN "Album"
ON "Production_Company"."Company_ID" = "Album"."Company_ID"
GROUP BY "Company_Name";
```

Snapshot :



201901131_db/postgres@PostgreSQL 13

Query Editor Query History

```
1 SET SEARCH_PATH TO "T9_OMMS";
2 SELECT "Company_Name",COUNT("Album_ID") AS "Number_of_album"
3 FROM "Production_Company" LEFT JOIN "Album"
4 ON "Production_Company"."Company_ID" = "Album"."Company_ID"
5 GROUP BY "Company_Name";
```

Data Output Explain Messages Notifications

	Company_Name character varying (100)	Number_of_album bigint
1	Lynch Inc	1
2	Weissnat Inc	1
3	Kunde-Carroll	2
4	Yost-Koch	3
5	Greenholt-Flatley	3
6	Rath Group	3
7	Romaguera Inc	0
8	Nienow Group	6
9	Conroy, Kilback and Krueger	0

✓ Successfully run. Total query runtime: 803 msec. 36 rows affected.

Result Contains : 36 Tuples.

31)

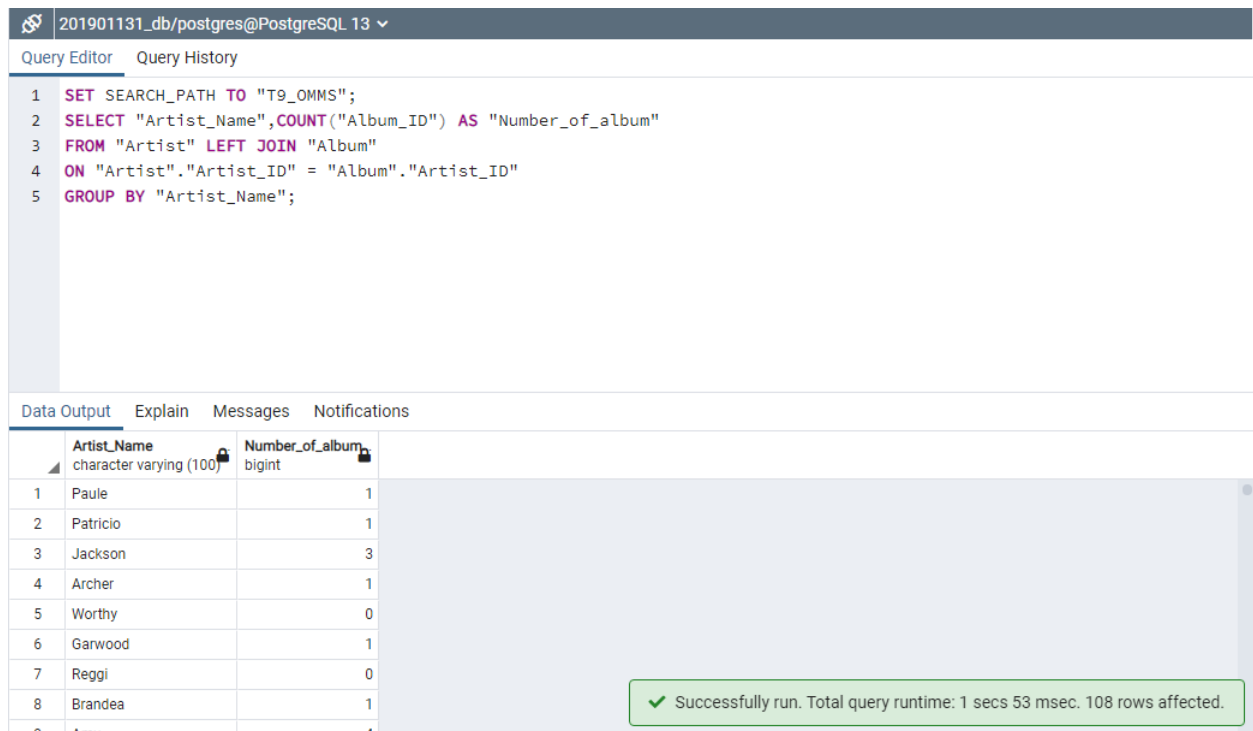
Plain English Query :

List artist along with number of albums.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
SELECT "Artist_Name",COUNT("Album_ID") AS "Number_of_album"
FROM "Artist" LEFT JOIN "Album"
ON "Artist"."Artist_ID" = "Album"."Artist_ID"
GROUP BY "Artist_Name";
```

Snapshot :



201901131_db/postgres@PostgreSQL 13

Query Editor Query History

```
1 SET SEARCH_PATH TO "T9_OMMS";
2 SELECT "Artist_Name",COUNT("Album_ID") AS "Number_of_album"
3 FROM "Artist" LEFT JOIN "Album"
4 ON "Artist"."Artist_ID" = "Album"."Artist_ID"
5 GROUP BY "Artist_Name";
```

Data Output Explain Messages Notifications

	Artist_Name character varying (100)	Number_of_album bigint
1	Paule	1
2	Patricio	1
3	Jackson	3
4	Archer	1
5	Worthy	0
6	Garwood	1
7	Reggi	0
8	Brandea	1
9	Arroy	4

✓ Successfully run. Total query runtime: 1 secs 53 msec. 108 rows affected.

Result Contains : 108 Tuples.

32)

Plain English Query :

List artist along with number of albums which have ratings greater than average rating.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
WITH tablename(colom) AS
(SELECT AVG("Album_Rating") FROM "Album")
SELECT "Artist_Name", COUNT("Album_ID")
FROM "Artist","Album",tablename
WHERE "Album"."Album_Rating" > colom AND
"Artist"."Artist_ID" = "Album"."Artist_ID"
GROUP BY "Artist_Name";
```

Snapshot :

The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is '201901131_db/postgres@PostgreSQL 13'. Below the connection bar, there are tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying the SQL query. Below the query editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with two columns: 'Artist_Name' (character varying (100)) and 'count' (bigint). The table contains 8 rows of data. A green status bar at the bottom right indicates 'Successfully run. Total query runtime: 1 secs 151 msec. 32 rows affected.'

	Artist_Name	count
1	Ambros	2
2	Paule	1
3	Jackson	1
4	Archer	1
5	Garwood	1
6	Brandea	1
7	Amy	2
8	Albrecht	1
9	Dutch	1

Result Contains : 32 Tuples.

33)

Plain English Query :

List artist along with number of profit which have profit greater than average profit.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
WITH tablename(colom) AS
(SELECT AVG("Profit") FROM "Album")
SELECT "Artist_Name", COUNT("Album_ID")
FROM "Artist","Album",tablename
WHERE "Album"."Profit" > colom AND
"Artist"."Artist_ID" = "Album"."Artist_ID"
GROUP BY "Artist_Name";
```

Snapshot :

The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is set to '201901131_db/postgres@PostgreSQL 13'. Below the connection bar, there are tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";
2 WITH tablename(colom) AS
3 (SELECT AVG("Profit") FROM "Album")
4 SELECT "Artist_Name", COUNT("Album_ID")
5 FROM "Artist","Album",tablename
6 WHERE "Album"."Profit" > colom AND
7 "Artist"."Artist_ID" = "Album"."Artist_ID"
8 GROUP BY "Artist_Name";
```

Below the query editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing the results of the query in a table format. The table has two columns: 'Artist_Name' (character varying (100)) and 'count' (bigint). The results are as follows:

	Artist_Name	count
1	Ambros	1
2	Jackson	2
3	Brandea	1
4	Amy	2
5	Gil	1
6	Ainslie	1
7	Erda	1
8	Lorelei	1
9	Milliecent	1

At the bottom right of the 'Data Output' tab, there is a green status box that reads: '✓ Successfully run. Total query runtime: 1 secs 554 msec. 30 rows affected.'

Result Contains : 30 Tuples.

34)

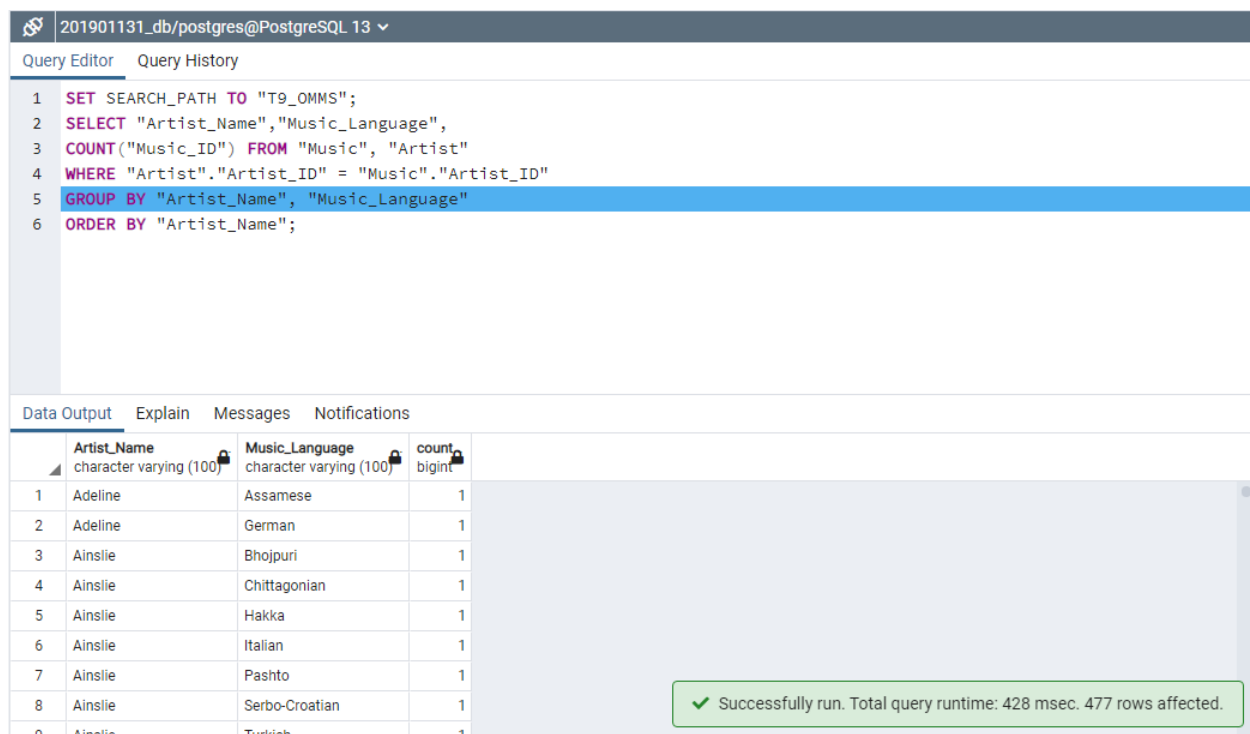
Plain English Query :

List artist along with number of music made in each language.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
SELECT "Artist_Name", "Music_Language",
COUNT("Music_ID") FROM "Music", "Artist"
WHERE "Artist"."Artist_ID" = "Music"."Artist_ID"
GROUP BY "Artist_Name", "Music_Language"
ORDER BY "Artist_Name";
```

Snapshot :



The screenshot shows a PostgreSQL query editor interface. The top bar indicates the database is '201901131_db/postgres@PostgreSQL 13'. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";
2 SELECT "Artist_Name", "Music_Language",
3 COUNT("Music_ID") FROM "Music", "Artist"
4 WHERE "Artist"."Artist_ID" = "Music"."Artist_ID"
5 GROUP BY "Artist_Name", "Music_Language"
6 ORDER BY "Artist_Name";
```

The 'Data Output' tab is also active, showing the results of the query. The results are displayed in a table with three columns: 'Artist_Name', 'Music_Language', and 'count'. The table contains 8 rows of data:

	Artist_Name	Music_Language	count
1	Adeline	Assamese	1
2	Adeline	German	1
3	Ainslie	Bhojpuri	1
4	Ainslie	Chittagonian	1
5	Ainslie	Hakka	1
6	Ainslie	Italian	1
7	Ainslie	Pashto	1
8	Ainslie	Serbo-Croatian	1

A green notification box at the bottom right of the results area states: '✓ Successfully run. Total query runtime: 428 msec. 477 rows affected.'

Result Contains : 477 Tuples.

35)

Plain English Query :

List album along with number of music in each language.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
SELECT "Album_Name", "Music_Language",
COUNT("Music_ID") FROM "Music", "Album"
WHERE "Album"."Album_ID" = "Music"."Album_ID"
GROUP BY "Album_Name", "Music_Language"
ORDER BY "Album_Name";
```

Snapshot :

The screenshot shows a PostgreSQL query editor interface. The top bar indicates the database is '201901131_db/postgres@PostgreSQL 13'. Below the bar, there are tabs for 'Query Editor' and 'Query History'. The query editor contains the following SQL code:

```
1 SET SEARCH_PATH TO "T9_OMMS";
2 SELECT "Album_Name", "Music_Language",
3 COUNT("Music_ID") FROM "Music", "Album"
4 WHERE "Album"."Album_ID" = "Music"."Album_ID"
5 GROUP BY "Album_Name", "Music_Language"
6 ORDER BY "Album_Name";
```

Below the query editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with the following columns: 'Album_Name' (character varying (100)), 'Music_Language' (character varying (100)), and 'count' (bigint). The table contains 9 rows of data:

	Album_Name	Music_Language	count
1	Alien	Arabic	1
2	Alien	Chinese	1
3	Alien	Chittagonian	1
4	Alien	Gan Chinese	1
5	Alien	Greek	1
6	Alien	Italian	2
7	Alien	Malaysian	1
8	Alien	Russian	1
9	Alien	Serbo-Croatian	1

At the bottom right of the results window, there is a green status bar that reads: '✓ Successfully run. Total query runtime: 910 msec. 430 rows affected.'

Result Contains : 430 Tuples.

36)

Plain English Query :

List albums along with a number of music types(pop, rope, Hlp-Hop, etc).

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
SELECT "Album_Name", "Music_Type",
COUNT("Music_ID") FROM "Music", "Album"
WHERE "Album"."Album_ID" = "Music"."Album_ID"
GROUP BY "Album_Name", "Music_Type"
ORDER BY "Album_Name";
```

Snapshot :

The screenshot shows a PostgreSQL query editor interface. The top bar indicates the database is '201901131_db/postgres@PostgreSQL 13'. Below the bar are tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";
2 SELECT "Album_Name", "Music_Type",
3 COUNT("Music_ID") FROM "Music", "Album"
4 WHERE "Album"."Album_ID" = "Music"."Album_ID"
5 GROUP BY "Album_Name", "Music_Type"
6 ORDER BY "Album_Name";
```

Below the query editor are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with the following data:

	Album_Name character varying (100)	Music_Type character varying (60)	count bigint
1	Alien	Baroque	1
2	Alien	Blues	2
3	Alien	Country	2
4	Alien	Drum & Bass	2
5	Alien	Dubstep	1
6	Alien	Hard Rock	1
7	Alien	House	2
8	Alien	Opera	2
9	Alien	Pop	1

A green status bar at the bottom right of the output window indicates: '✓ Successfully run. Total query runtime: 647 msec. 296 rows affected.'

Result Contains : 296 Tuples.

37)

Plain English Query :

List company name along with number of produced songs.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
SELECT "Company_Name", COUNT("Music_ID")
FROM "Production_Company", "Album", "Music"
WHERE
"Production_Company"."Company_ID" =
"Album"."Company_ID" AND
"Album"."Album_ID" = "Music"."Album_ID"
GROUP BY "Company_Name"
ORDER BY "Company_Name";
```

Snapshot :

201901131_db/postgres@PostgreSQL 13

Query Editor

Query History

1

SET SEARCH_PATH TO "T9_OMMS";

2

SELECT "Company_Name", COUNT("Music_ID")

3

FROM "Production_Company", "Album", "Music"

4

WHERE

5

"Production_Company"."Company_ID" =

6

"Album"."Company_ID" AND

7

"Album"."Album_ID" = "Music"."Album_ID"

8

GROUP BY "Company_Name"

9

ORDER BY "Company_Name";

Data Output

Explain

Messages

Notifications

	Company_Name character varying (100)	count bigint
1	Bins Inc	20
2	Bins LLC	14
3	Christiansen and Sons	3
4	Dooley, Conn and Kessler	12
5	Durgan Inc	5
6	Farrell, Stoltenberg and Barrows	10
7	Fisher-Ankunding	13
8	Greenholt-Flatley	26
9	Hammer and Sons	12
10	Hartman and Sons	10

✓ Successfully run. Total query runtime: 242 msec. 30 rows affected.

Result Contains : 30 Tuples.

38)

Plain English Query :

List artist along with a number of music types(pop, rope, Hlp-Hop, etc).

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
SELECT "Artist_Name", "Music_Type",
COUNT("Music_ID") FROM "Music", "Artist"
WHERE "Artist"."Artist_ID" = "Music"."Artist_ID"
GROUP BY "Artist_Name", "Music_Type"
ORDER BY "Artist_Name";
```

Snapshot :

The screenshot shows a PostgreSQL query editor interface. At the top, the database connection is '201901131_db/postgres@PostgreSQL 13'. Below the connection bar, there are tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO "T9_OMMS";
2 SELECT "Artist_Name", "Music_Type",
3 COUNT("Music_ID") FROM "Music", "Artist"
4 WHERE "Artist"."Artist_ID" = "Music"."Artist_ID"
5 GROUP BY "Artist_Name", "Music_Type"
6 ORDER BY "Artist_Name";
7
```

Below the query editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing the results of the query in a table format. The table has three columns: 'Artist_Name' (character varying (100)), 'Music_Type' (character varying (60)), and 'count' (bigint). The results are as follows:

	Artist_Name	Music_Type	count
1	Adeline	House	1
2	Adeline	Soul	1
3	Ainslie	Hip-Hop	1
4	Ainslie	Jazz	1
5	Ainslie	Latin	1
6	Ainslie	Opera	1
7	Ainslie	Pop	2
8	Ainslie	Techno	1
9	Altkrecht	Disco	1

At the bottom right of the 'Data Output' tab, there is a green status message: '✓ Successfully run. Total query runtime: 1 secs 260 msec. 428 rows affected.'

Result Contains : 428 Tuples.

39)

Plain English Query :

Show plan details using a function.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";  
CREATE OR REPLACE FUNCTION "T9_OMMS".show_plans()  
RETURNS TABLE(PID BIGINT, PName CHAR VARYING(100), Duration INT, Price INT)  
LANGUAGE 'plpgsql'
```

```
AS $BODY$  
DECLARE  
tempvar record;
```

```
BEGIN
```

```
FOR tempvar in (  
    SELECT "Plan_ID", "Plan_Name", "Plan_Duration", "Plan_Price" FROM  
    "T9_OMMS"."Premium_Details"  
)  
LOOP  
    PID := tempvar."Plan_ID";  
    PName := tempvar."Plan_Name";  
    Duration := tempvar."Plan_Duration";  
    Price := tempvar."Plan_Price";  
    Return next;  
end loop;  
END;  
$BODY$;  
SELECT * FROM show_plans();
```


Snapshot :

201901131_db/postgres@PostgreSQL 13 ▾

Query Editor Query History

```
1 SET SEARCH_PATH TO "T9_OMMS";
2 CREATE OR REPLACE FUNCTION "T9_OMMS".show_plans()
3 RETURNS TABLE(PID BIGINT, PName CHAR VARYING(100), Duration INT, Price INT)
4 LANGUAGE 'plpgsql'
5
6 AS $BODY$
7 DECLARE
8   tempvar record;
9
10 BEGIN
11
12   FOR tempvar IN (
```

Data Output Explain Messages Notifications

	pid bigint	pname character varying	duration integer	price integer
1	1	Free	1000	0
2	2	Monthly	36	500
3	3	Bimonthly	72	1000
4	4	Trimonthly	108	1500
5	5	Bumper	144	2000
6	6	BigBumper	180	2200
7	7	Supersell	216	2400
8	8	ExtraSuper	252	2500
9	9	Holiday	288	2600

✓ Successfully run. Total query runtime: 521 msec. 11 rows affected.

Result Contains : 11 Tuples

41)

Plain English Query :

Create Trigger function to do transaction.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
CREATE OR REPLACE FUNCTION buy_plan()
RETURNS TRIGGER
LANGUAGE 'plpgsql'
AS $BODY$
DECLARE
price integer;
number_of_transaction integer;

BEGIN

SET SEARCH_PATH TO "T9_OMMS";
```

```
SELECT "Plan_Price" INTO price
FROM "Premium_Details"
WHERE "Plan_ID" = NEW."Plan_ID";
RAISE NOTICE '%', price;
```

```
SELECT COUNT("Transaction_ID") INTO
number_of_transaction FROM "Transaction";
RAISE NOTICE '%', number_of_transaction;
```

```
INSERT INTO "Transaction"(
"Transaction_ID","Transaction_Mode",
"Sender_ID", "Receiver_ID",
"Sender_Type","Receiver_Type", "Price")
VALUES (number_of_transaction+1,'offline',NEW."User_ID",201901076,'User','Admin',price);
RETURN NULL;
END;
$BODY$;
```

```
CREATE TRIGGER "Buy_Plan" AFTER INSERT OR UPDATE
ON "T9_OMMS"."User" FOR EACH ROW EXECUTE FUNCTION "T9_OMMS".buy_plan();
```

Check on update

```
SET SEARCH_PATH TO "T9_OMMS";
UPDATE "T9_OMMS"."User"
SET "Plan_ID" = 4
WHERE "User_ID" = 1;
SELECT * FROM "Transaction";
```

Snapshot :

201901131_db/postgres@PostgreSQL 13

Query EditorQuery History

1 UPDATE "T9_OMMS"."User"

2 SET "Plan_ID" = 4

3 WHERE "User_ID" = 1;

Data Output

Explain

Messages

Notifications

NOTICE: 1500

NOTICE: 1

UPDATE 1

Query returned successfully in 211 msec.

✓ Query returned successfully in 211 msec.

201901131_db/postgres@PostgreSQL 13

Query EditorQuery History

1 SET SEARCH_PATH TO "T9_OMMS";

2 UPDATE "T9_OMMS"."User"

3 SET "Plan_ID" = 4

4 WHERE "User_ID" = 1;

5 SELECT * FROM "Transaction";

Data Output

Explain

Messages

Notifications

	Transaction_ID [PK] bigint	Transaction_Mode character varying (20)	Sender_ID bigint	Receiver_ID bigint	Sender_Type character varying (20)	Receiver_Type character varying (20)	Price bigint
1	1	offline	1	201901076	User	Admin	1500
2	2	offline	1	201901076	User	Admin	1500

✓ Successfully run. Total query runtime: 398 msec. 2 rows affected.

Result Contains : 2 Rows affected.

41)

Plain English Query :

Create trigger function to update User Type.

SQL Query :

```
SET SEARCH_PATH TO "T9_OMMS";
CREATE OR REPLACE FUNCTION set_user_type()
RETURNS TRIGGER
LANGUAGE 'plpgsql'
AS $BODY$
BEGIN

IF (NEW."Plan_ID" > 1) THEN
    NEW."User_Type" = 'Premium';
ELSE
    NEW."User_Type" = 'Free';
END IF;

RETURN NEW;
END;
$BODY$;

CREATE TRIGGER "Set_User_Type"
BEFORE UPDATE ON "T9_OMMS"."User"
FOR EACH ROW EXECUTE FUNCTION "T9_OMMS".set_user_type();
```

Check on Update

```
SET SEARCH_PATH TO "T9_OMMS";
UPDATE "T9_OMMS"."User"
SET "User_Type" = 'Premium'
WHERE "User_ID" = 1;
```

Snapshot :

201901131_db/postgres@PostgreSQL 13 ▾

Query Editor Query History

```
1 SET SEARCH_PATH TO "T9_OMMS";
2 UPDATE "T9_OMMS"."User"
3 SET "User_Type" = 'Premium'
4 WHERE "User_ID" = 1;
```

Data Output Explain Messages Notifications

NOTICE: 1500
NOTICE: 4
UPDATE 1

Query returned successfully in 240 msec.

✓ Query returned successfully in 240 msec.

201901131_db/postgres@PostgreSQL 13 ▾

Query Editor Query History

```
1 SET SEARCH_PATH TO "T9_OMMS";
2 SELECT * FROM "User";
```

Data Output Explain Messages Notifications

	User_ID [PK] bigint	User_Name character varying (100)	Password character varying (128)	Email_ID character varying (100)	Mobile_Number character (10)	Plan_ID integer	User_Type character varying (60)
194	195	Gustaf	XLMM6tFsn	gmarlowe5e@furl.net	2606675022	1	Non_Premium
195	196	Hulda	xLtOT23l	hdanilovich5f@webnode.com	7442878729	1	Non_Premium
196	197	Ambrosi	9MHlocbC0m	atower5g@a8.net	4987060286	1	Non_Premium
197	198	Billie	mrfrBag1T0	bcatlow5h@amazon.de	2955457234	1	Non_Premium
198	199	Odie	R2tMcK39	odales5i@arizona.edu	9286344090	1	Non_Premium
199	200	Cristy	HcHMSN	curridge5j@pbs.org	9496660916	1	Non_Premium
200	1	Melodee	7zj0m93xWj	martrick0@jigsy.com	2276321528	4	Premium

Result Contains : 200 Tupels..