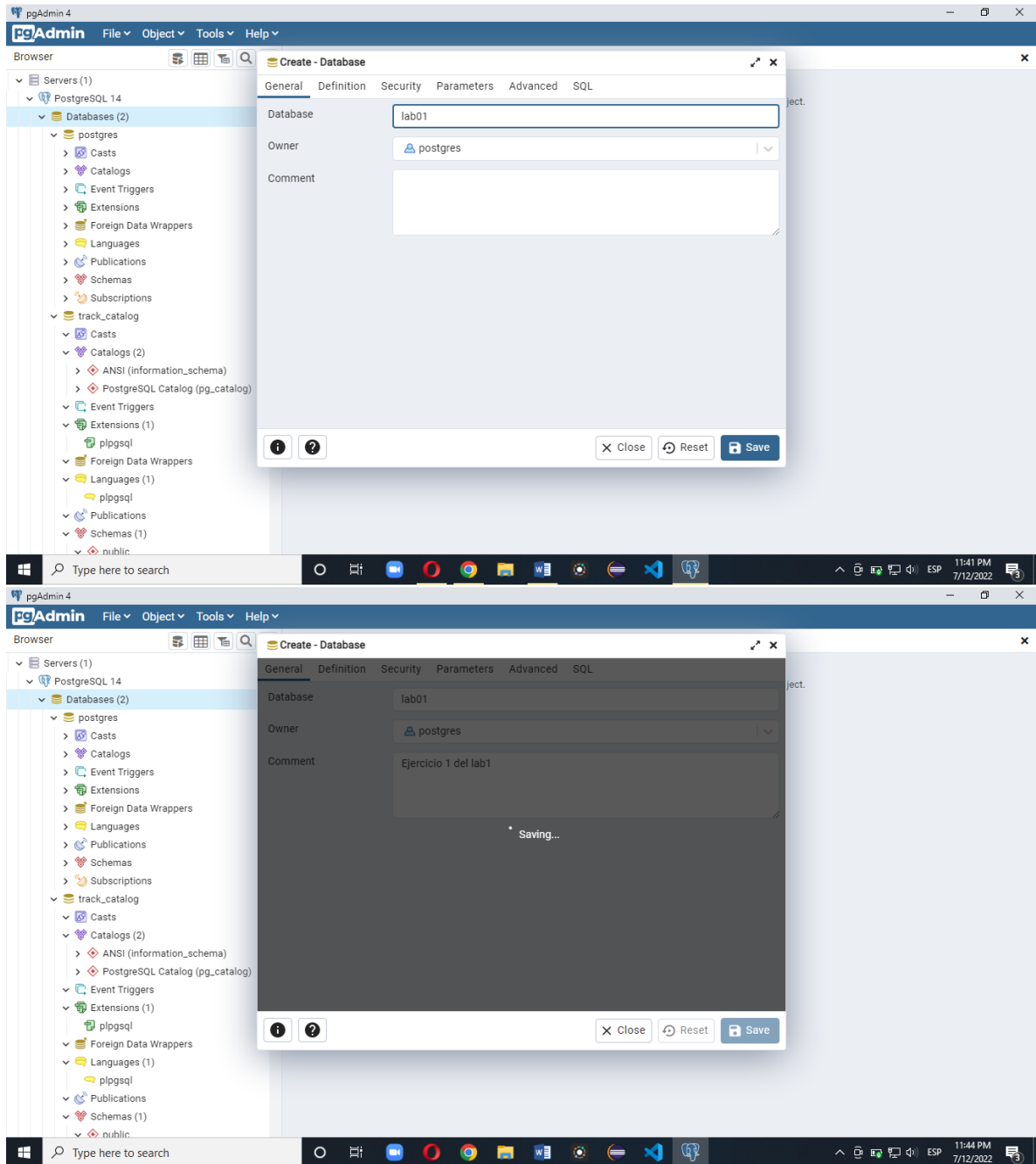


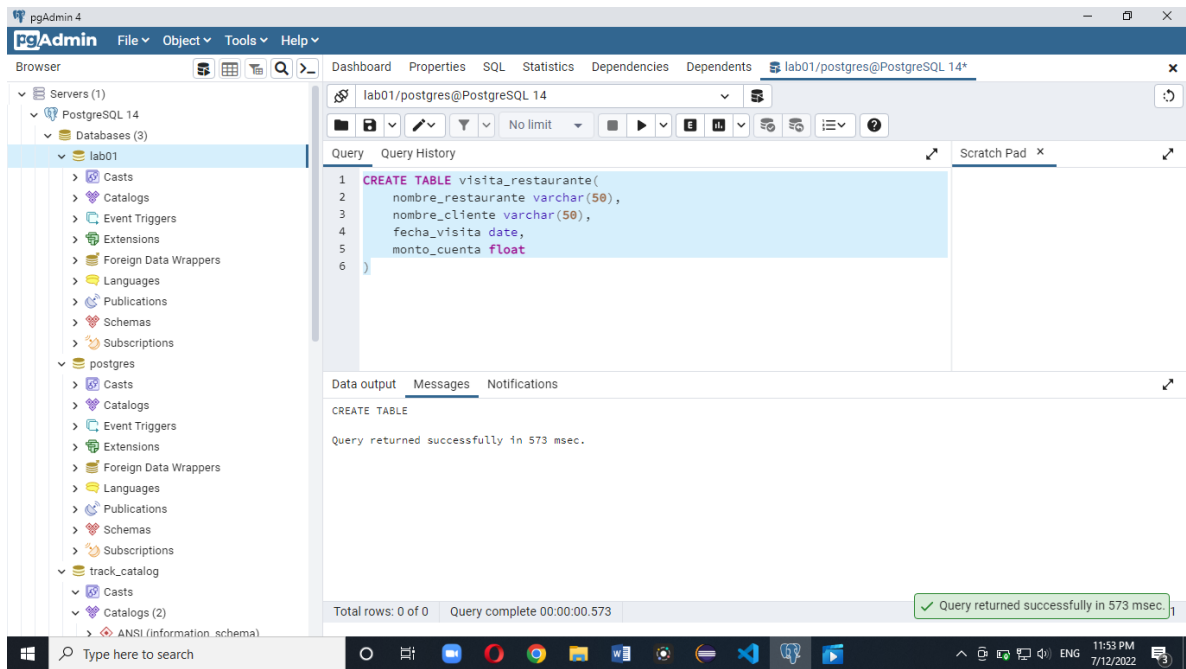
Laboratorio 01 Base de datos

EJERCICIO 1

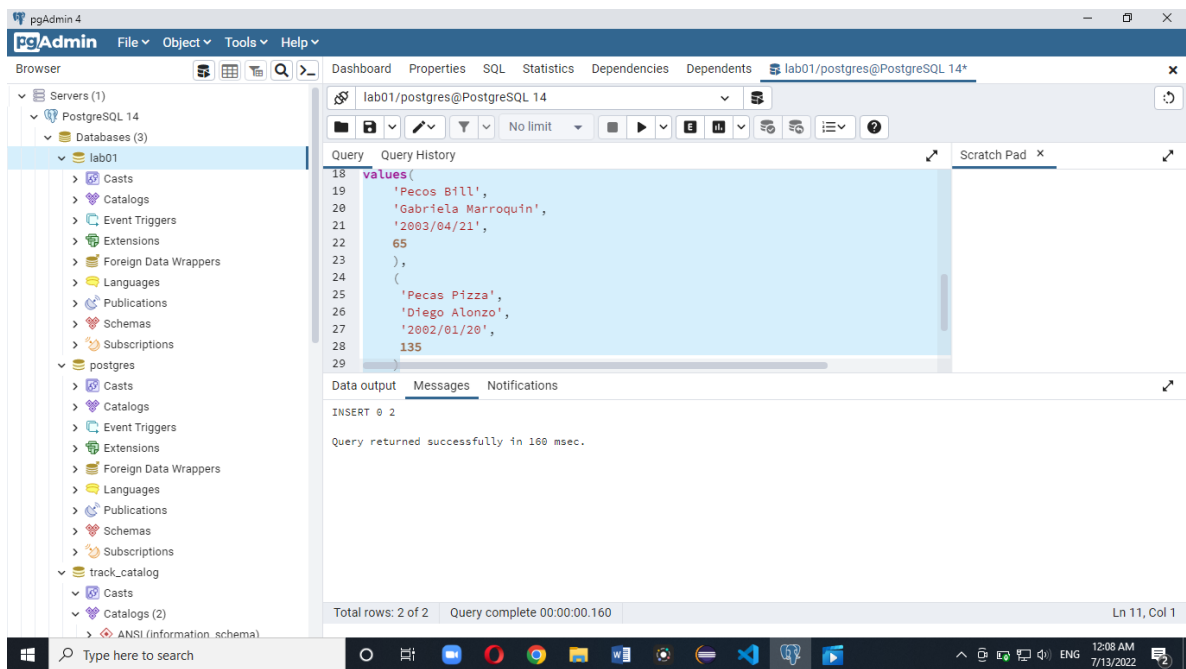
1.1 Crear la base de datos lab01



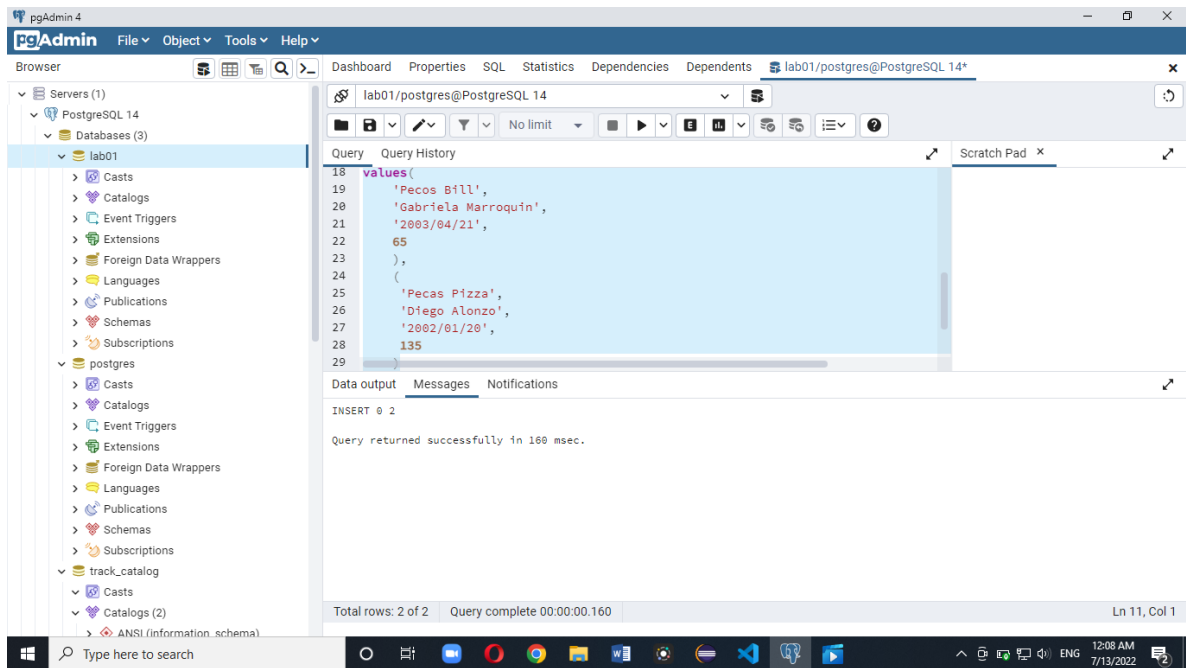
1.2 Crear una tabla de visita al restaurante.



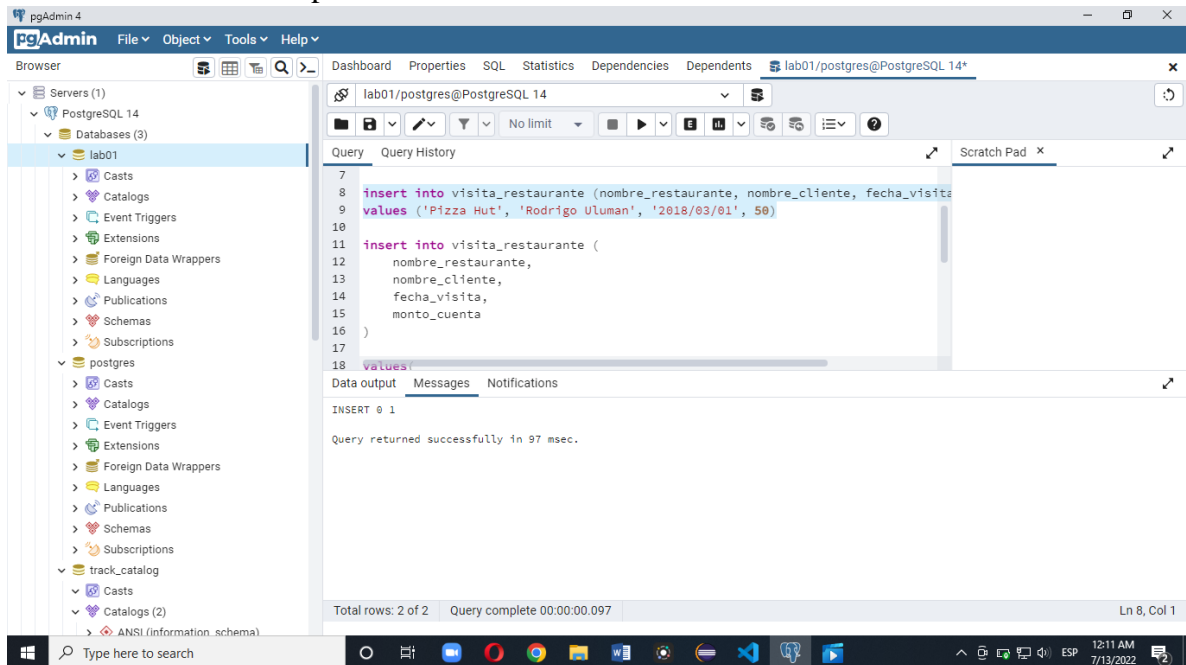
1.3 Inserción de datos en la tabla de visita a restaurante.



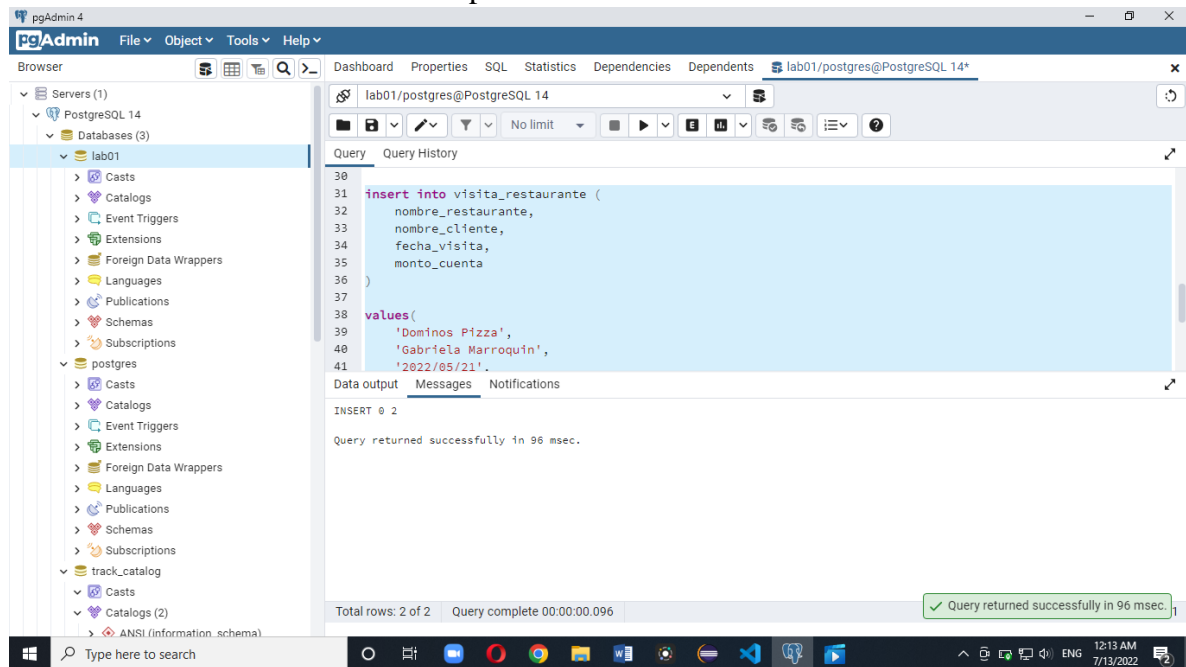
Inserción de múltiples datos a la vez.



Inserción de una sola tupla a la vez.



Inserción de últimos dos elementos para tener 5 visitas a restaurantes.



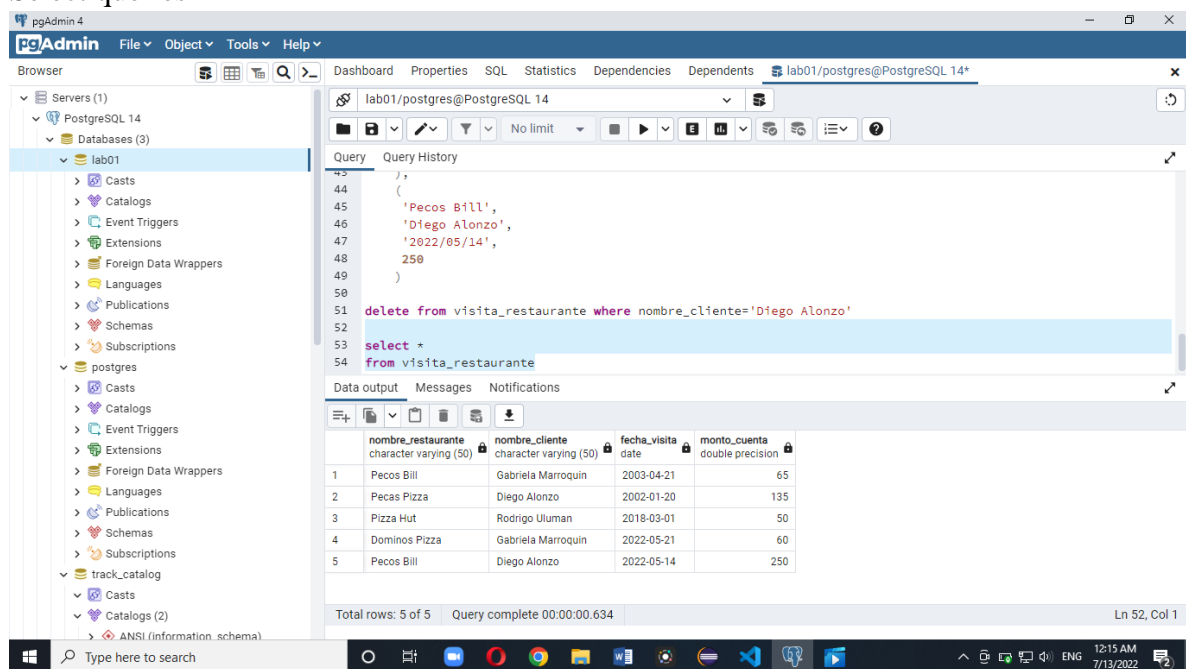
```
insert into visita_restaurante (
  nombre_restaurante,
  nombre_cliente,
  fecha_visita,
  monto_cuenta
)
values(
  'Dominos Pizza',
  'Gabriela Marroquin',
  '2022/05/21',
  60
)
```

INSERT 0 2

Query returned successfully in 96 msec.

Total rows: 2 of 2 Query complete 00:00:00.096

1.4 Select queries



```
delete from visita_restaurante where nombre_cliente='Diego Alonzo'

select *
from visita_restaurante
```

Data output Messages Notifications

	nombre_restaurante character varying (50)	nombre_cliente character varying (50)	fecha_visita date	monto_cuenta double precision
1	Pecos Bill	Gabriela Marroquin	2003-04-21	65
2	Pecas Pizza	Diego Alonzo	2002-01-20	135
3	Pizza Hut	Rodrigo Ullman	2018-03-01	50
4	Dominos Pizza	Gabriela Marroquin	2022-05-21	60
5	Pecos Bill	Diego Alonzo	2022-05-14	250

Total rows: 5 of 5 Query complete 00:00:00.634 Ln 52, Col 1

Query para seleccionar a aquellas personas que hayan ido a comer a Pecos Bill y en qué fecha lo fueron a visitar.

The screenshot shows the pgAdmin 4 interface. On the left, the 'Servers' tree is expanded to 'PostgreSQL 14' > 'lab01'. The main pane displays a SQL query in the 'Query' tab. The query is as follows:

```
47 2022/05/14',
48 250
49 )
50
51 delete from visita_restaurante where nombre_cliente='Diego Alonzo'
52
53 select *
54 from visita_restaurante
55
56 select nombre_cliente, fecha_visita
57 from visita_restaurante
58 where nombre_restaurante = 'Pecos Bill'
```

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

	nombre_cliente	fecha_visita
1	Gabriela Marroquin	2003-04-21
2	Diego Alonzo	2022-05-14

The status bar at the bottom indicates 'Total rows: 2 of 2' and 'Query complete 00:00:00.092'.

Query para verificar cuál fue la tupla que tuvo el gasto máximo.

The screenshot shows the pgAdmin 4 interface. On the left, the 'Servers' tree is expanded to 'PostgreSQL 14' > 'lab01'. The main pane displays a SQL query in the 'Query' tab. The query is as follows:

```
51 delete from visita_restaurante where nombre_cliente='Diego Alonzo'
52
53 select *
54 from visita_restaurante
55
56 select nombre_cliente, fecha_visita
57 from visita_restaurante
58 where nombre_restaurante = 'Pecos Bill'
59
60 select nombre_cliente, fecha_visita, monto_cuenta
61 from visita_restaurante
62 where monto_cuenta= (select max(monto_cuenta) from visita_restaurante)
```

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

	nombre_cliente	fecha_visita	monto_cuenta
1	Diego Alonzo	2022-05-14	250

The status bar at the bottom indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.100'.

Los queries del ejercicio 1:

```
CREATE TABLE visita_restaurante(
    nombre_restaurante varchar(50),
    nombre_cliente varchar(50),
    fecha_visita date,
```

monto_cuenta float

)

insert into visita_restaurante (nombre_restaurante, nombre_cliente, fecha_visita,
monto_cuenta)

values ('Pizza Hut', 'Rodrigo Uluman', '2018/03/01', 50)

insert into visita_restaurante (

nombre_restaurante,

nombre_cliente,

fecha_visita,

monto_cuenta

)

values(

'Pecos Bill',

'Gabriela Marroquin',

'2003/04/21',

65

),

(

'Pecas Pizza',

'Diego Alonzo',

'2002/01/20',

135

)

insert into visita_restaurante (

nombre_restaurante,

```
nombre_cliente,  
fecha_visita,  
monto_cuenta  
)
```

```
values(  
  'Dominos Pizza',  
  'Gabriela Marroquin',  
  '2022/05/21',  
  60  
)  
(  
  'Pecos Bill',  
  'Diego Alonzo',  
  '2022/05/14',  
  250  
)
```

```
delete from visita_restaurante where nombre_cliente='Diego Alonzo'
```

```
select *  
from visita_restaurante
```

```
select nombre_cliente, fecha_visita  
from visita_restaurante  
where nombre_restaurante = 'Pecos Bill'
```

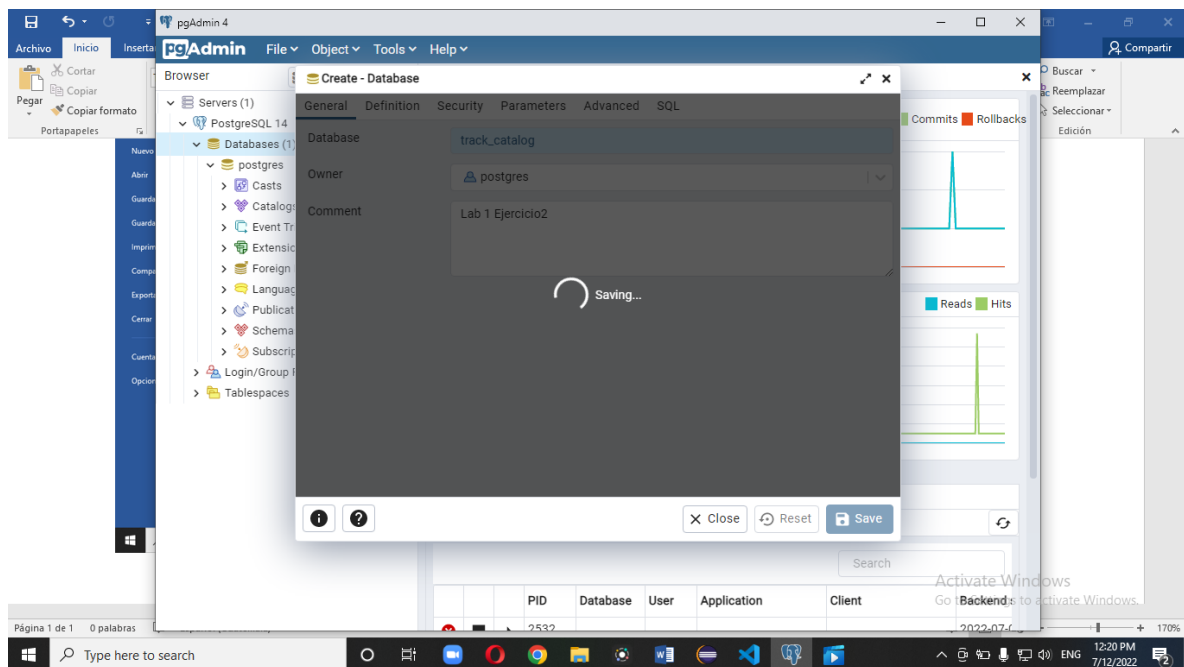
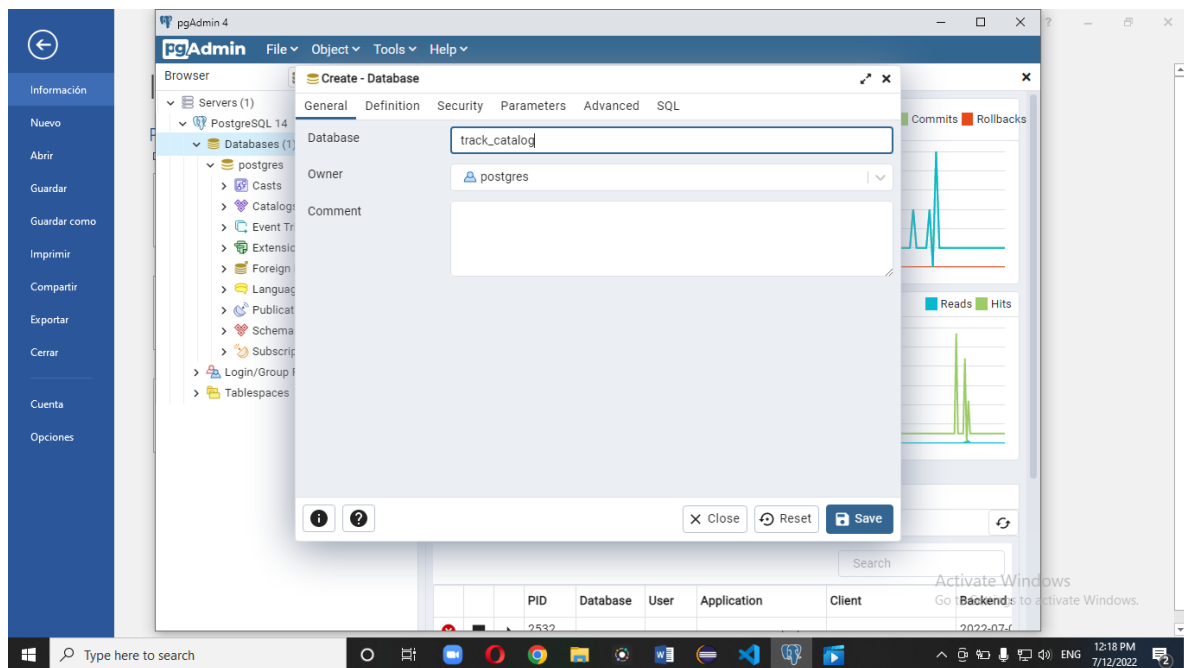
```
select nombre_cliente, fecha_visita, monto_cuenta
```

from visita_restaurante

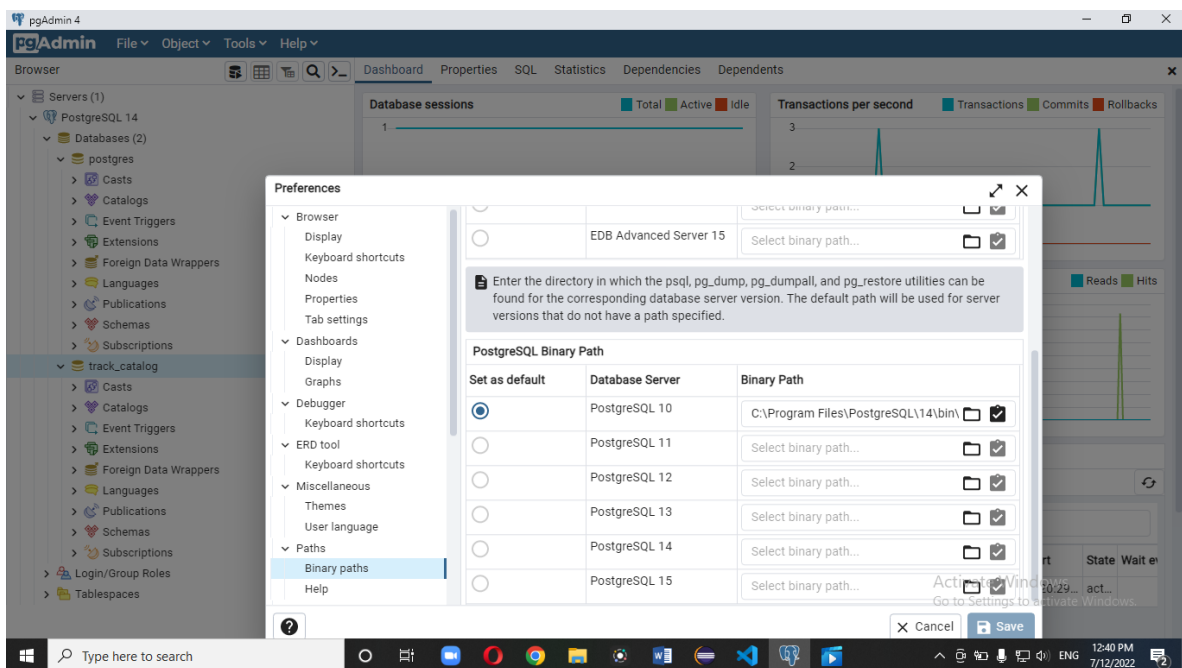
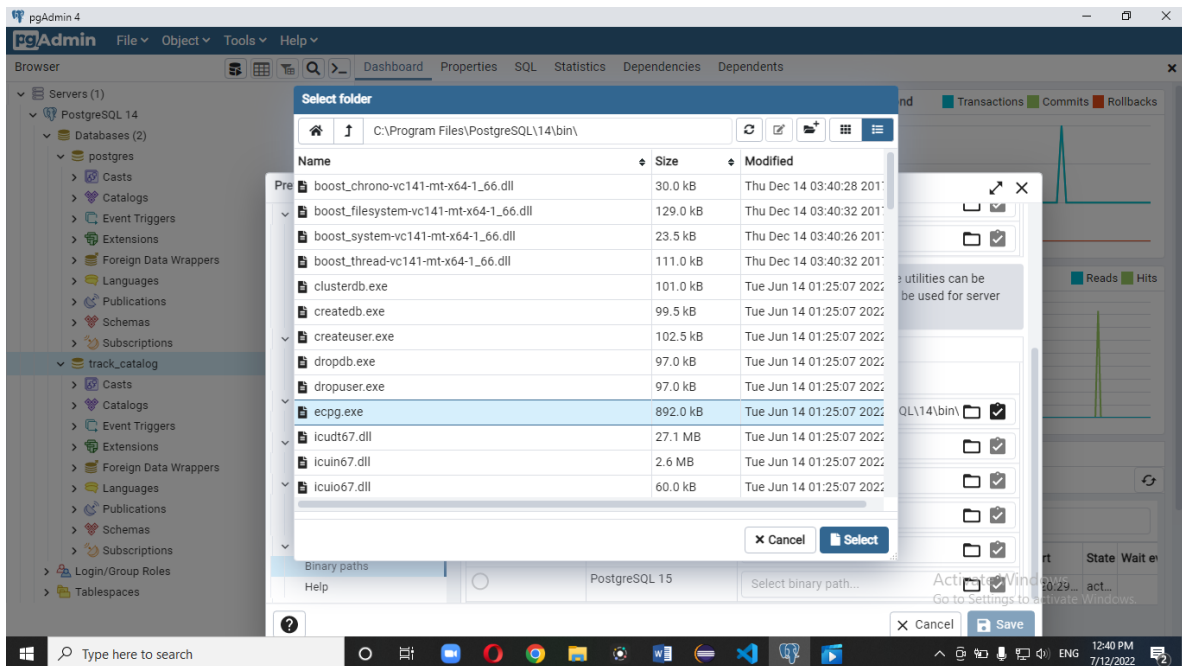
where monto_cuenta= (select max(monto_cuenta) from visita_restaurante)

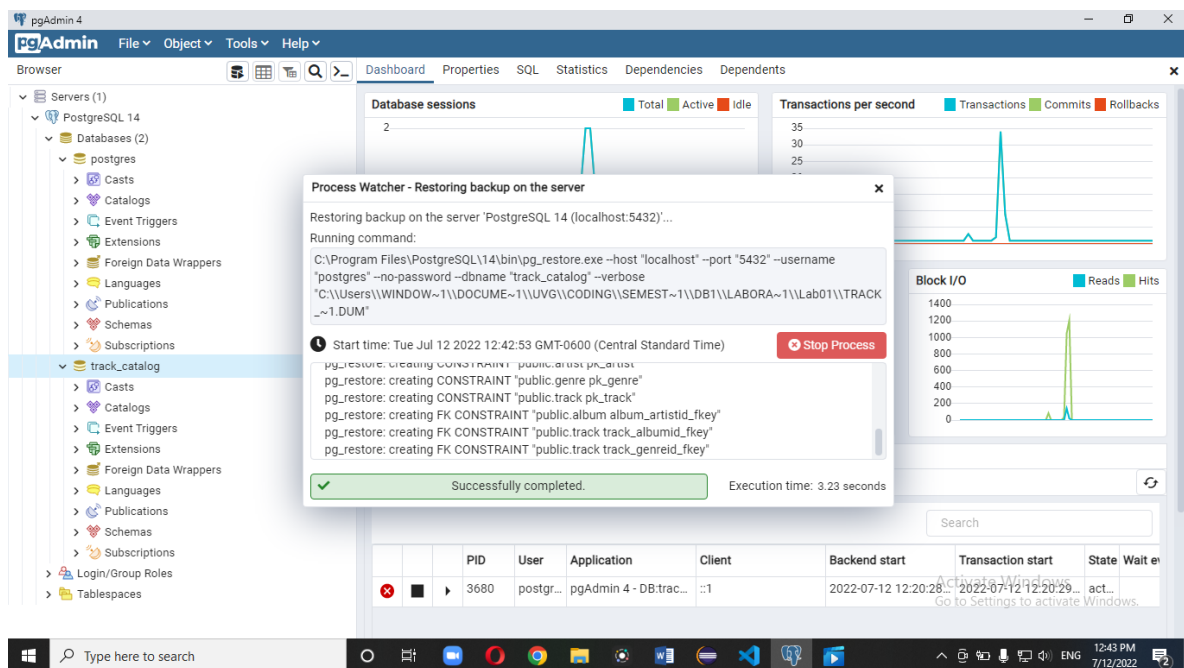
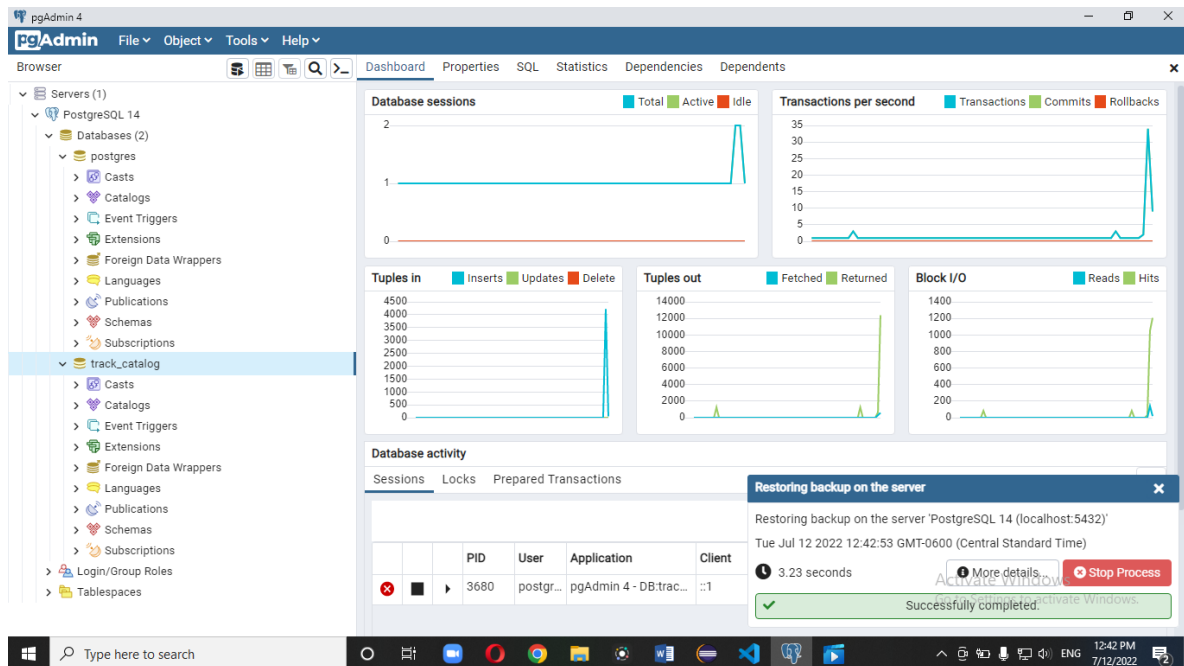
EJERCICIO 2

2.1 Crear base de datos track_catalog



2.2 Levantar el backup de la base de datos





2.3.1 Las tablas disponibles para esta base de datos

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for PostgreSQL 14, including the 'track_catalog' database. The main pane shows a SQL query executed in the 'track_catalog/postgres@PostgreSQL 14*' connection:

```
1 SELECT *
2 FROM pg_catalog.pg_tables
3 WHERE schemaname != 'pg_catalog' AND
4       schemaname != 'information_schema';
```

The 'Data output' tab shows the results of the query as a table with 4 rows and 9 columns:

	schemaname	tablename	tableowner	tablespace	hasindexes	hasrules	hastriggers	rowsecurity
1	public	artist	postgres	[null]	true	false	true	false
2	public	album	postgres	[null]	true	false	true	false
3	public	track	postgres	[null]	true	false	true	false
4	public	genre	postgres	[null]	true	false	true	false

The status bar at the bottom indicates 'Total rows: 4 of 4' and 'Query complete 00:00:00.591'.

The screenshot shows the pgAdmin 4 interface with the 'Dependents' tab selected for the 'track' table. The main pane displays a message: 'No dependant information is available for the selected object.' The left sidebar shows the database structure, including the 'track' table and its dependencies.

2.3.2 El esquema de la tabla artista es: Artista ID es de tipo entero, Nombre es de tipo Carácter Variante.

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, including the 'track_catalog' database. The main pane shows a SQL query executed on the 'track_catalog' database. The query filters for tables in the 'pg_catalog' schema and the 'information_schema'.

```
1 SELECT +
2 FROM pg_catalog.pg_tables
3 WHERE schemaname != 'pg_catalog' AND
4      schemaname != 'information_schema';
5
6 SELECT +
7 FROM artist
8
9
```

The 'Data output' tab shows the results of the query, displaying a table with columns 'artistid' and 'name'.

artistid	name
1	ac/dc
2	accept
3	aerosmith
4	alanis morissette
5	alice in chains
6	antônio carlos jobim

Total rows: 275 of 275. Query complete 00:00:00.323.

2.3.3 ¿Cuántas tracks hay registradas en la base de datos?

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, including the 'track_catalog' database. The main pane shows a SQL query executed on the 'track_catalog' database. The query filters for tables in the 'pg_catalog' schema and the 'information_schema', and then counts the number of tracks in the 'TRACK' table.

```
3 WHERE schemaname != 'pg_catalog' AND
4      schemaname != 'information_schema';
5
6 SELECT +
7 FROM artist
8
9 SELECT +
10 FROM TRACK
11
12 SELECT count(trackid)
13 from TRACK
14
```

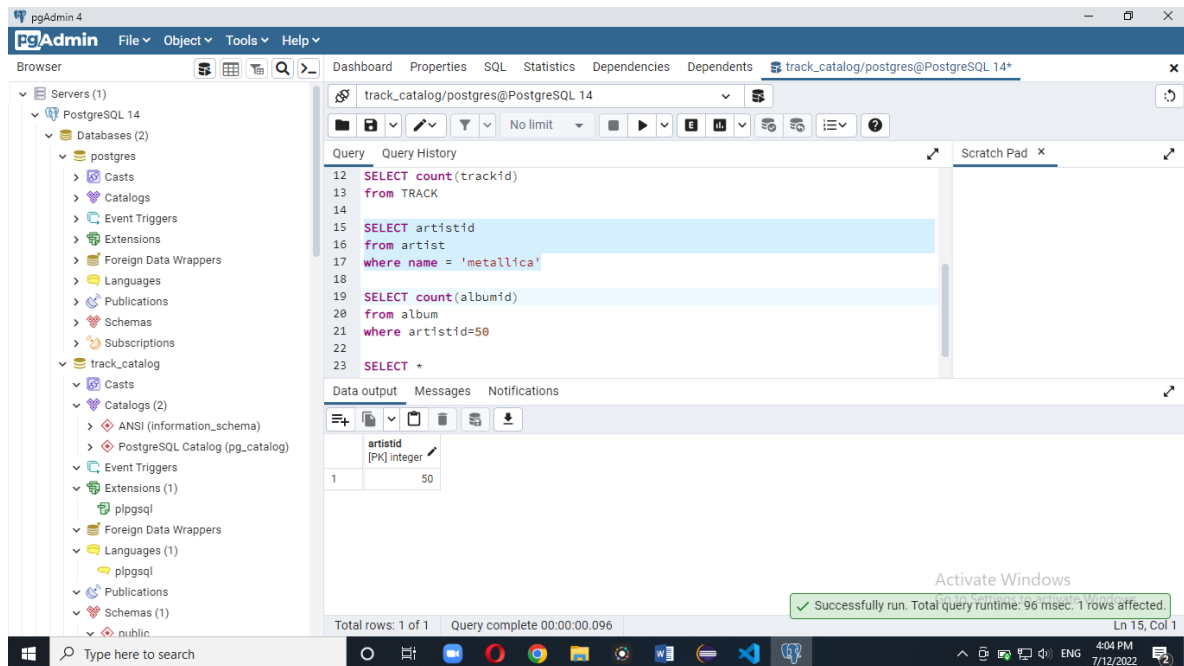
The 'Data output' tab shows the results of the query, displaying a table with columns 'count' and 'bigint'.

count	bigint
1	3503

Total rows: 1 of 1. Query complete 00:00:00.213.

Hay 3503 tracks registradas.

2.3.4 ¿Qué álbumes de la banda metallica se encuentran registrados en la base de datos?



pgAdmin 4

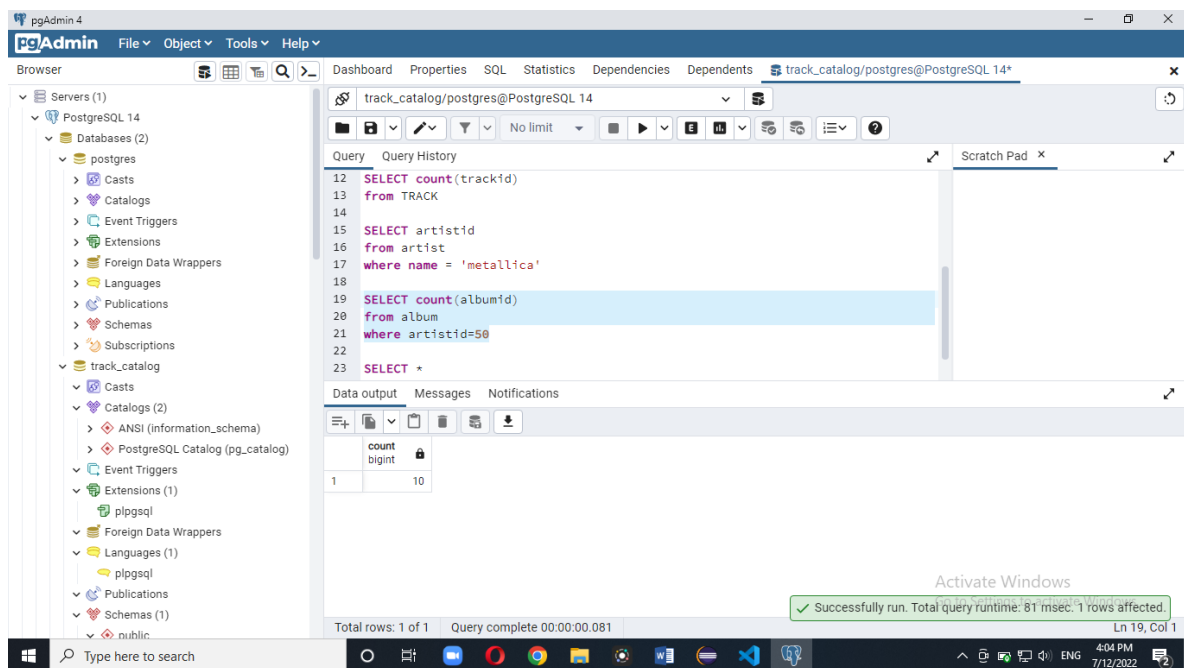
track_catalog/postgres@PostgreSQL 14*

```
12 SELECT count(trackid)
13 from TRACK
14
15 SELECT artistid
16 from artist
17 where name = 'metallica'
18
19 SELECT count(albumid)
20 from album
21 where artistid=50
22
23 SELECT *
```

artistid	[PK] integer
1	50

Successfully run. Total query runtime: 96 msec. 1 rows affected.

Ln 15, Col 1



pgAdmin 4

track_catalog/postgres@PostgreSQL 14*

```
12 SELECT count(trackid)
13 from TRACK
14
15 SELECT artistid
16 from artist
17 where name = 'metallica'
18
19 SELECT count(albumid)
20 from album
21 where artistid=50
22
23 SELECT *
```

count	bigint
1	10

Successfully run. Total query runtime: 81 msec. 1 rows affected.

Ln 19, Col 1

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, including the 'track_catalog' database. The main pane shows a SQL query in the 'Query' tab:

```
28 FROM album o
29 INNER JOIN artist i
30 ON o.artistid = i.artistid
31 WHERE i.name='metallica'
32 )
33 SELECT o.artistid, i.artistid, i.name
34 FROM album o
35 INNER JOIN artist i
36 ON o.artistid = i.artistid
37 WHERE i.name='metallica'
38
39
```

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

	artistid integer	artistid integer	name character varying (120)
4	50	50	metallica
5	50	50	metallica
6	50	50	metallica
7	50	50	metallica
8	50	50	metallica
9	50	50	metallica
10	50	50	metallica

Total rows: 10 of 10 Query complete 00:00:00.102

2.3.5 ¿Cuántas canciones duran igual o más de 5 minutos ?

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, including the 'track_catalog' database. The main pane shows a SQL query in the 'Query' tab:

```
33
34 SELECT o.artistid, i.artistid, i.name
35 FROM album o
36 INNER JOIN artist i
37 ON o.artistid = i.artistid
38 WHERE i.name='metallica'
39
40 Select count(milliseconds)
41 from TRACK
42 WHERE milliseconds>=300000
43
44
```

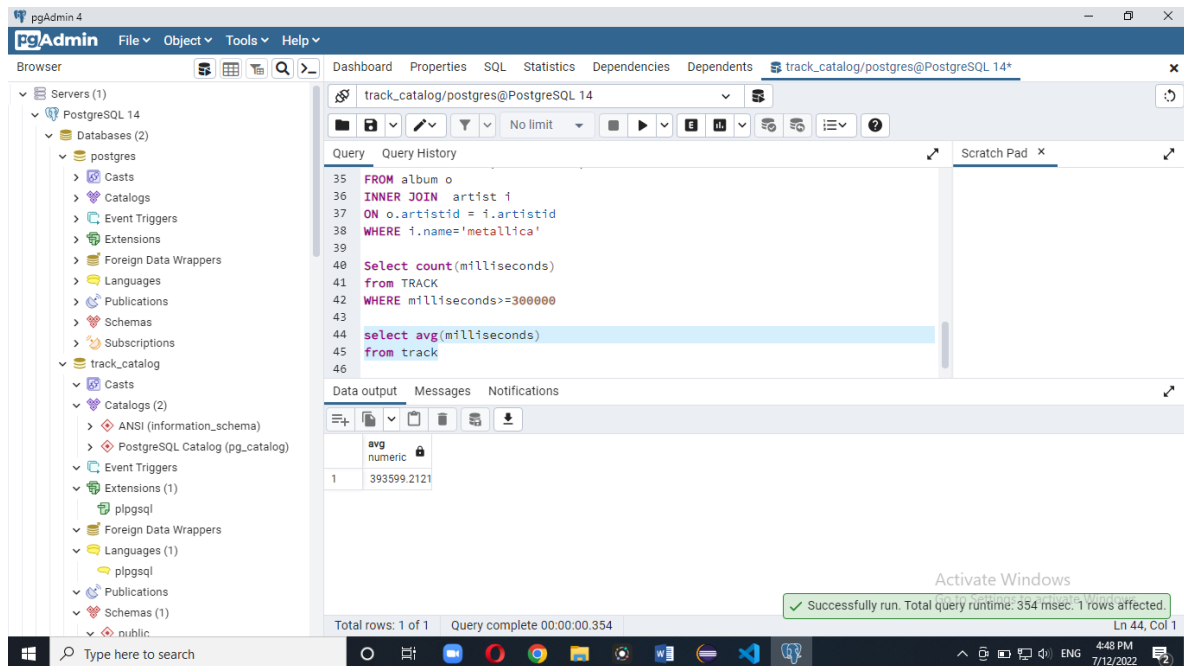
The 'Data output' tab shows the results of the query:

	count bigint
1	1069

Total rows: 1 of 1 Query complete 00:00:00.124

1069 canciones duran al menos 5 minutos.

2.3.6 ¿Cuál es el promedio en milisegundos de las tracks?



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, including the 'track_catalog' database. The main pane shows a SQL query in the 'Query' tab:

```
35 FROM album o
36 INNER JOIN artist i
37 ON o.artistid = i.artistid
38 WHERE i.name='metallica'
39
40 Select count(milliseconds)
41 from TRACK
42 WHERE milliseconds>=300000
43
44 select avg(milliseconds)
45 from track
46
```

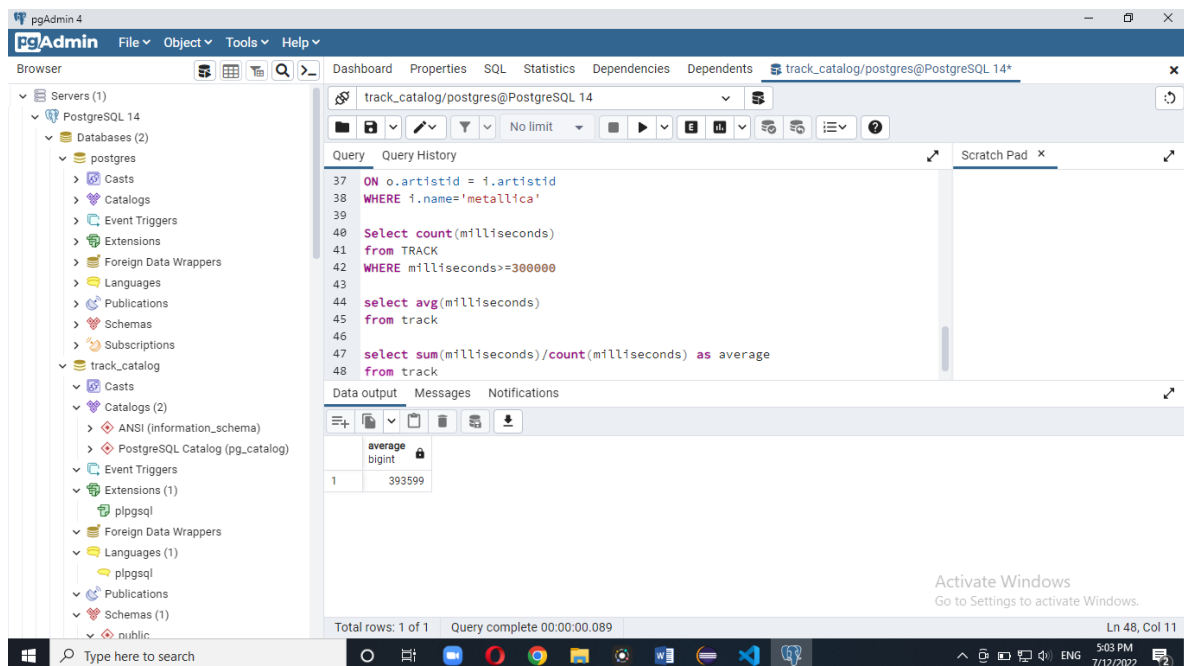
The 'Data output' tab shows the results of the query:

avg numeric
393599.2121

The status bar at the bottom indicates: 'Total rows: 1 of 1 Query complete 00:00:00.354 Ln 44, Col 1'. A green message box at the bottom right says: 'Successfully run. Total query runtime: 354 msec. 1 rows affected.'

El promedio de las canciones en milisegundos es de 393599.2121.

2.3.7



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, including the 'track_catalog' database. The main pane shows a SQL query in the 'Query' tab:

```
37 ON o.artistid = i.artistid
38 WHERE i.name='metallica'
39
40 Select count(milliseconds)
41 from TRACK
42 WHERE milliseconds>=300000
43
44 select avg(milliseconds)
45 from track
46
47 select sum(milliseconds)/count(milliseconds) as average
48 from track

```

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

average bigint
393599

The status bar at the bottom indicates: 'Total rows: 1 of 1 Query complete 00:00:00.089 Ln 48, Col 11'. A green message box at the bottom right says: 'Successfully run. Total query runtime: 89 msec. 1 rows affected.'

El promedio de tiempo de canciones sin utilizar la función AVG es 393599 milisegundos.