

--1 Proporciona una SQL que muestre los siguientes datos: - Nombre Actor - Apellido Actor

```
select first_name, last_name from actor;
```

```
select count(*) from actor;
```

-- 200

The screenshot shows the pgAdmin 4 web interface. The left sidebar displays a tree view of the database structure, with the 'actor' table selected under the 'Tables (15)' category. The main panel is divided into two sections: the 'Query Editor' and the 'Data Output'.

Query Editor: Contains the following SQL queries:

```
--1 Proporciona una SQL que muestre los siguientes datos: - Nombre Actor - Apellido Actor  
  
select first_name, last_name from actor;  
  
select count(*) from actor;
```

Data Output: Displays the results of the queries. The first query's results are not visible, but the second query's results are shown in a table:

count
200

-- 2 Proporciona una SQL que muestre los siguientes datos: - Nombre Actor - Titulo de la Película

```
select a.first_name, f.title from actor as a
inner join film_actor as fa on a.actor_id=fa.actor_id
inner join film as f on fa.film_id=f.film_id;
select count(*) from actor as a
inner join film_actor as fa on a.actor_id=fa.actor_id
inner join film as f on fa.film_id=f.film_id;
-- 5462
```

The screenshot shows the pgAdmin 4 web interface. The left sidebar displays a tree view of the database structure, with the 'actor' table selected under the 'Tables (15)' category. The main window is divided into two panes. The top pane, titled 'Query Editor', contains the following SQL query:

```
-- 2 Proporciona una SQL que muestre los siguientes datos: - Nombre Actor - Titulo de la Película

select a.first_name, f.title from actor as a
inner join film_actor as fa on a.actor_id=fa.actor_id
inner join film as f on fa.film_id=f.film_id;
select count(*) from actor as a
inner join film_actor as fa on a.actor_id=fa.actor_id
inner join film as f on fa.film_id=f.film_id;
-- 5462
```

The bottom pane, titled 'Data Output', displays the results of the query in a table format:

count
5462

-- 3 Proporciona una SQL que muestre los siguientes datos: - Nombre Actor - Número de películas - Ordenar de mayor a menor

```
select a.first_name,a.last_name,count(fa.actor_id) as numero_pelic from actor as a
```

```
inner join film_actor as fa on a.actor_id=fa.actor_id
```

```
inner join film as f on fa.film_id=f.film_id
```

```
group by a.actor_id order by 3 desc;
```

```
select count(*) from (select a.first_name,a.last_name,count(fa.actor_id) as numero_pelic from actor as a
```

```
inner join film_actor as fa on a.actor_id=fa.actor_id
```

```
inner join film as f on fa.film_id=f.film_id
```

```
group by a.actor_id)as t;
```

```
-- 200
```

The screenshot shows the pgAdmin 4 web interface. On the left, a tree view displays the database structure, with 'Tables (15)' expanded and 'actor' selected. The main panel is divided into 'Query Editor' and 'Data Output'. The 'Query Editor' contains the following SQL code:

```
-- 3 Proporciona una SQL que muestre los siguientes datos: - Nombre Actor - Número de películas - Ordenar de mayor a menor

select a.first_name,a.last_name,count(fa.actor_id) as numero_pelic from actor as a
inner join film_actor as fa on a.actor_id=fa.actor_id
inner join film as f on fa.film_id=f.film_id
group by a.actor_id order by 3 desc;

select count(*) from (select a.first_name,a.last_name,count(fa.actor_id) as numero_pelic from actor as a
inner join film_actor as fa on a.actor_id=fa.actor_id
inner join film as f on fa.film_id=f.film_id
group by a.actor_id)as t;

-- 200
```

The 'Data Output' tab shows the results of the query in a table with two columns: 'count' and 'bigint'. The first row shows a value of 1 in the 'count' column and 200 in the 'bigint' column.

count	bigint
1	200

At the bottom right, a green message box states: 'Successfully run. Total query runtime: 956 msec. 1 rows affected.'

-- 4 Proporciona una SQL que muestre los siguientes datos: - Película - Numero de veces alquilada

```
select f.title, count(f.title) as veces_rentada from film as f
```

```
inner join inventory as i on f.film_id=i.film_id
```

```
inner join rental as r on i.inventory_id=r.inventory_id
```

```
group by f.title;
```

```
select count(*) from (select f.title, count(f.title) as veces_rentada from film as f
```

```
inner join inventory as i on f.film_id=i.film_id
```

```
inner join rental as r on i.inventory_id=r.inventory_id
```

```
group by f.title) as t;
```

--958

The screenshot shows the pgAdmin 4 web interface. On the left is a tree view of the database schema, including tables like 'actor', 'address', 'category', 'city', 'country', 'customer', 'film', 'film_actor', 'film_category', 'inventory', 'language', 'payment', and 'rental'. The main panel displays a SQL query in the 'Query Editor' tab. The query is as follows:

```
-- 4 Proporciona una SQL que muestre los siguientes datos: - Película - Numero de veces alquilada

select f.title, count(f.title) as veces_rentada from film as f
inner join inventory as i on f.film_id=i.film_id
inner join rental as r on i.inventory_id=r.inventory_id
group by f.title;

select count(*) from (select f.title, count(f.title) as veces_rentada from film as f
inner join inventory as i on f.film_id=i.film_id
inner join rental as r on i.inventory_id=r.inventory_id
group by f.title) as t;

--958
```

Below the query editor, the 'Data Output' tab shows the results of the query. The first result is a single row with the following data:

count	bigint
1	958

A green message box at the bottom right indicates: 'Successfully run. Total query runtime: 884 msec. 1 rows affected.'

-- 5 Proporciona una SQL que muestre los siguientes datos: - Película - Dinero recaudado por película

```
select f.title, sum(amount) as dinero_recaudado from film as f
```

```
inner join inventory as i on f.film_id=i.film_id
```

```
inner join rental as r on i.inventory_id=r.inventory_id
```

```
inner join payment as p on r.rental_id=p.rental_id
```

```
group by f.title order by 2 desc;
```

```
select count(*) from (select f.title, sum(amount) as dinero_recaudado from film as f
```

```
inner join inventory as i on f.film_id=i.film_id
```

```
inner join rental as r on i.inventory_id=r.inventory_id
```

```
inner join payment as p on r.rental_id=p.rental_id
```

```
group by f.title) as t; -- 958
```

The screenshot shows the pgAdmin 4 web interface. On the left is a tree view of the database structure, with 'Tables (15)' expanded and 'rental' selected. The main pane is the 'Query Editor' for the 'dvdrental/postgres@server' database. It contains the following SQL query:

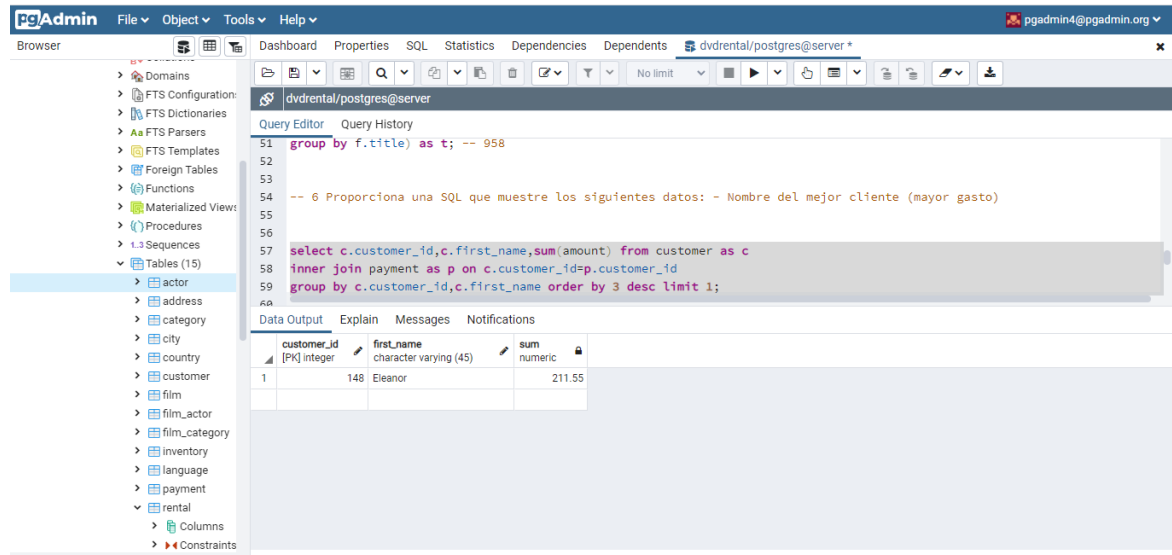
```
-- 5 Proporciona una SQL que muestre los siguientes datos: - Película - Dinero recaudado por película
select f.title, sum(amount) as dinero_recaudado from film as f
inner join inventory as i on f.film_id=i.film_id
inner join rental as r on i.inventory_id=r.inventory_id
inner join payment as p on r.rental_id=p.rental_id
group by f.title order by 2 desc;
select count(*) from (select f.title, sum(amount) as dinero_recaudado from film as f
inner join inventory as i on f.film_id=i.film_id
inner join rental as r on i.inventory_id=r.inventory_id
inner join payment as p on r.rental_id=p.rental_id
group by f.title) as t;
-- 958
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query in a table:

count	bigint
1	958

-- 6 Proporciona una SQL que muestre los siguientes datos: - Nombre del mejor cliente (mayor gasto)

```
select c.customer_id,c.first_name,sum(amount) from customer as c  
  
inner join payment as p on c.customer_id=p.customer_id  
  
group by c.customer_id,c.first_name order by 3 desc limit 1;
```



The screenshot shows the pgAdmin 4 web interface. On the left, the 'Tables (15)' folder is expanded, showing tables like 'actor', 'address', 'category', 'city', 'country', 'customer', 'film', 'film_actor', 'film_category', 'inventory', 'language', 'payment', 'rental', 'Columns', and 'Constraints'. The 'customer' table is selected. The main pane shows the 'Query Editor' with the following SQL query:

```
-- 6 Proporciona una SQL que muestre los siguientes datos: - Nombre del mejor cliente (mayor gasto)  
  
select c.customer_id,c.first_name,sum(amount) from customer as c  
inner join payment as p on c.customer_id=p.customer_id  
group by c.customer_id,c.first_name order by 3 desc limit 1;
```

Below the query editor, the 'Data Output' tab is active, displaying the results of the query in a table:

customer_id	first_name	sum
1	Eleanor	211.55

-- 7 Proporciona una SQL que muestre los siguientes datos: - Nombre del mejor cliente (mayor num alquileres)

```
select c.first_name,c.last_name,count(r.customer_id)cant_alquileres
```

```
from customer as c
```

```
inner join rental as r on c.customer_id=r.customer_id
```

```
group by c.customer_id order by 3 desc limit 1;
```

The screenshot shows the pgAdmin 4 web interface. On the left, a tree view displays the database structure, with the 'rental' table selected under the 'customer' schema. The main pane is divided into two sections: the 'Query Editor' and the 'Data Output'.

Query Editor: The SQL query is entered as follows:

```
-- 7 Proporciona una SQL que muestre los siguientes datos: - Nombre del mejor cliente (mayor num alquileres)
select c.first_name,c.last_name,count(r.customer_id)cant_alquileres
from customer as c
inner join rental as r on c.customer_id=r.customer_id
group by c.customer_id order by 3 desc limit 1;
```

Data Output: The query results are displayed in a table with the following columns: first_name, last_name, and cant_alquileres. The results show one row for the customer with the highest number of rentals.

first_name	last_name	cant_alquileres
Eleanor	Hunt	46