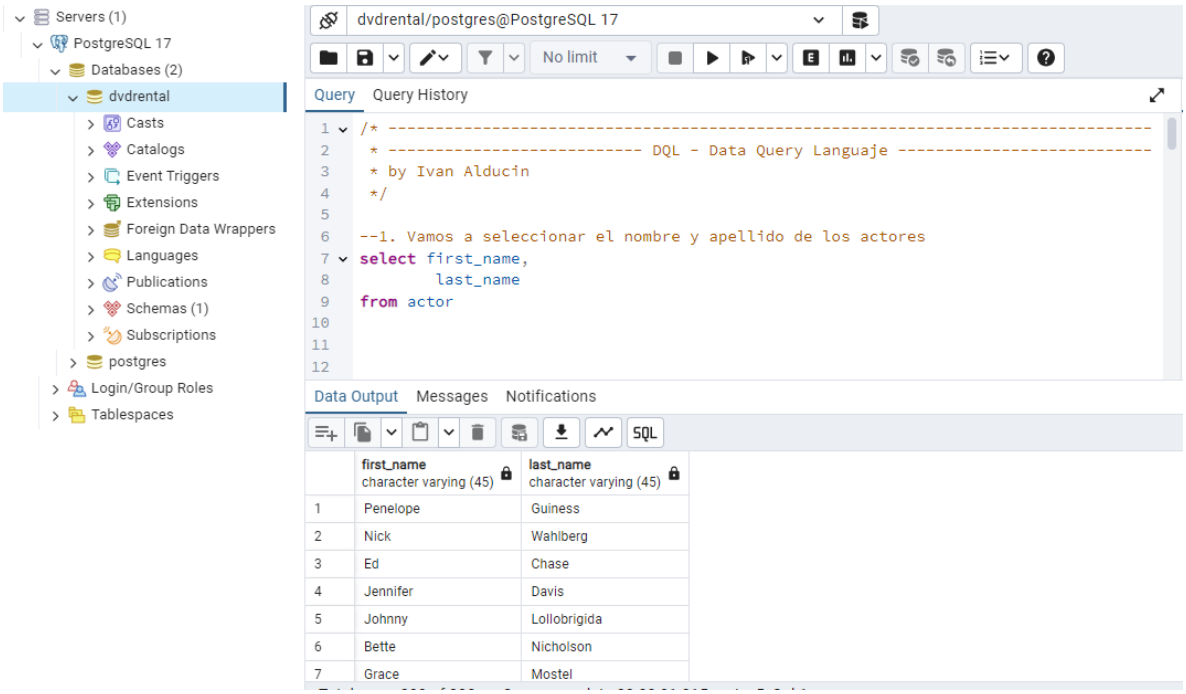


```sql
--2. Vamos a seleccionar el nombre completo del actor en una sola columna
select first_name || ' ' || last_name AS NombreCompleto
from actor
```

Output Messages Notifications

SQL

nombrecompleto text
Penelope Guiness
Nick Wahlberg
Ed Chase
Jennifer Davis
Johnny Lollobrigida
Bette Nicholson
Grace Mostel

```sql
--3. Selecciona los actores que su nombre empieza con "D"
select first_name,
       last_name
from actor
where first_name like 'D%'
```

Data Output | Messages | Notifications

	first_name character varying (45)	last_name character varying (45)
1	Dan	Torn
2	Dan	Harris
3	Dustin	Tautou
4	Daryl	Wahlberg
5	Dan	Streep
6	Daryl	Crawford
7	Debbie	Akroyd

Total rows: 7 of 7 Query complete 00:00:00.842 Ln 19, Col 27

```sql
--4. ¿Tenemos algún actor con el mismo nombre?
select first_name,
       count(*) As Name_Dup
from actor
group by first_name
having count(*) > 1;
```

Data Output | Messages | Notifications

	first_name character varying (45)	name_dup bigint
1	Christian	3
2	Kenneth	4
3	Cameron	3
4	Spencer	2
5	Humphrey	2
6	Audrey	2
7	Fay	3

```sql
--5. ¿Cuál es el costo máximo de renta de una película?

select max(amount) as CostoMaximo
from payment
```

Data Output | Messages | Notifications

	costomaximo numeric
1	11.99

--6. ¿Cuáles son las películas que fueron rentadas con ese costo?

1. Conectar payment con rental a través de rental_id:

La tabla payment tiene información sobre los pagos. Podemos usar el campo rental_id para unirla con la tabla rental:

```sql
SELECT *
FROM payment p
INNER JOIN rental r ON p.rental_id = r.rental_id
WHERE p.amount = 11.99;
```

Data Output | Messages | Notifications

	payment_id integer	customer_id smallint	staff_id smallint	rental_id integer	amount numeric (5,2)	payment_date timestamp without time zone	rental_id integer	rental_date timestamp without time zone	inventory_id integer
3	23757	116	2	14763	11.99	2007-03-21 22:02:26.996577	14763	2005-08-21 23:34:00	1480
4	24553	195	2	16040	11.99	2007-03-23 20:47:59.996577	16040	2005-08-23 22:19:33	3524
5	24866	237	2	11479	11.99	2007-03-02 20:46:39.996577	11479	2005-08-02 22:18:13	4077

Resultado de esta consulta: **Obtendremos todas las rentas (rental_id) que tienen un costo de 11.99.**

2. Conectar rental con inventory a través de inventory_id:

Ahora, usando el campo inventory_id de la tabla rental, podemos unirla con la tabla inventory, que contiene la información de qué película se rentó:

```
SELECT *
FROM payment p
INNER JOIN rental r ON p.rental_id = r.rental_id
INNER JOIN inventory i ON r.inventory_id = i.inventory_id
WHERE p.amount = 11.99;
```

Data Output | Messages | Notifications

	[illegible]	rental_date timestamp without time zone	inventory_id integer	customer_id smallint	return_date timestamp without time zone	staff_id smallint	last_update timestamp without time zone	inventory_id integer	film_id smallint
1	759	2005-08-21 23:28:58	3871	362	2005-08-31 00:35:58	2	2006-02-16 02:30:53	3871	[illegible]
2	415	2005-08-22 23:48:56	4176	204	2005-09-01 02:05:56	1	2006-02-16 02:30:53	4176	[illegible]
3	763	2005-08-21 23:34:00	1480	116	2005-08-31 03:58:00	2	2006-02-16 02:30:53	1480	[illegible]
4	040	2005-08-23 22:19:33	3524	195	2005-09-02 02:19:33	2	2006-02-16 02:30:53	3524	[illegible]

Resultado de esta consulta: **Ahora sabemos qué inventario (inventory_id) está relacionado con las rentas de 11.99.**

3. Conectar inventory con film a través de film_id:

Finalmente, con la tabla inventory, tenemos acceso al campo film_id, que nos dice qué película específica fue rentada. Podemos unir esta tabla con film para obtener el título de la película:

```
SELECT f.title
FROM payment p
INNER JOIN rental r ON p.rental_id = r.rental_id
INNER JOIN inventory i ON r.inventory_id = i.inventory_id
INNER JOIN film f ON i.film_id = f.film_id
WHERE p.amount = 11.99;
```

Data Output | Messages | Notifications

	title character varying (255)
1	Sting Personal
2	Trap Guys
3	Flintstones Happiness
4	Scorpion Apollo
5	Ties Hunger
6	Mine Titans
7	Midsummer Groundhog

Total rows: 8 of 8 Query complete 00:00:00.251 Ln 54, Col 1

```sql
--7. ¿Cuantás películas hay por el tipo de audencia (rating)?

select rating,
     count(*) as CantidadPeliculasAudiencia
from film
Group by rating
```

Data Output | Messages | Notifications

	rating mpaa_rating	cantidadpeliculasaudiencia bigint
1	PG	194
2	R	195
3	G	178
4	PG-13	223
5	NC-17	210

Total rows: 5 of 5 Query complete 00:00:00.263 Ln 54, Col 1

-- NOT IN selecciona todas las películas cuyo rating no esté en la lista que contiene R y NC-17.

```sql
--8. Selecciona las películas que no tienen un rating R o NC-17
-- NOT IN selecciona todas las películas cuyo rating no esté en la lista que contiene R y NC-17.

select title,
       rating
from film
where rating NOT IN ('R', 'NC-17');
```

Data Output | Messages | Notifications

	title character varying (255)	rating mpaa_rating
1	Bright Encounters	PG-13
2	Academy Dinosaur	PG
3	Ace Goldfinger	G
4	Affair Prejudice	G
5	African Egg	G
6	Agent Truman	PG
7	Airplane Sierra	PG-13

Total rows: 595 of 595 Query complete 00:00:00.325 Ln 57, Col 1

```sql
--9. ¿Cuantos clientes hay en cada tienda?

select store_id, count(customer_id) as ClientesXTienda
from customer
group by store_id
```

Data Output | Messages | Notifications

	store_id smallint	clientesxtienda bigint
1	1	326
2	2	273

```sql
--10. ¿Cuál es la pelicula que mas veces se rento?

SELECT f.title, COUNT(i.film_id) AS PeliculaMasRentada
FROM film f
INNER JOIN inventory i
    ON f.film_id = i.film_id
INNER JOIN rental r
    ON i.inventory_id = r.inventory_id
GROUP BY f.title
ORDER BY PeliculaMasRentada DESC
LIMIT 1;
```

Data Output | Messages | Notifications

	title character varying (255)	peliculamasrentada bigint
1	Bucket Brotherhood	34

```sql
--11. ¿Qué peliculas no se han rentado?

SELECT f.title
FROM film f
LEFT JOIN inventory i
    ON f.film_id = i.film_id
LEFT JOIN rental r
    ON i.inventory_id = r.inventory_id
WHERE r.rental_id IS NULL;
```

Data Output | Messages | Notifications

	title character varying (255)
1	Academy Dinosaur
2	Sky Miracle
3	Kill Brotherhood
4	Sister Freddy
5	Gladiator Westward
6	Floats Garden
7	Apollo Teen

Total rows: 43 of 43 Query complete 00:00:00.219 Ln 86, Col 29

```sql
--12. ¿Qué clientes no han rentado ninguna película?

SELECT c.customer_id, c.first_name, c.last_name
FROM customer c
LEFT JOIN rental r
    ON c.customer_id = r.customer_id
WHERE r.rental_id IS NULL;
```

Data Output | Messages | Notifications

	customer_id [PK] integer	first_name character varying (45)	last_name character varying (45)

Por el resultado de la consulta, todos los clientes por lo menos han rentado 1 película.

```sql
--13. ¿Qué actores han actuado en más de 30 películas?

select a.first_name, count(fa.film_id) as Mas30Peliculas
from  actor a
    inner join film_actor fa
        on
        fa.actor_id = a.actor_id
group by a.first_name
having count(fa.film_id) > 30
order by Mas30Peliculas desc;
```

Data Output | Messages | Notifications

	first_name character varying (45)	mas30peliculas bigint
1	Kenneth	103
2	Penelope	102
3	Jayne	90
4	Matthew	89
5	Julia	88
6	Groucho	86
7	Ed	83

Total rows: 71 of 71 Query complete 00:00:00.630 Ln 108, Col 30

Query | Query History

```sql
--15. Muestra los clientes que rentaron una pelicula más de una vez
with rental_counts as (
    select c.customer_id,
           c.first_name,
           count(r.rental_id) as rental_count
    from customer c
    inner join rental r on c.customer_id = r.customer_id
    inner join inventory i on r.inventory_id = i.inventory_id
    group by c.customer_id, c.first_name,i.film_id
)
select customer_id,
       first_name
from rental_counts
where rental_count > 1;

```

Data Output | Messages | Notifications

	customer_id [PK] integer	first_name character varying (45)
1	513	Duane
2	280	Tracey
3	245	Courtney
4	516	Elmer

Total rows: 212 of 212 Query complete 00:00:00.271 Ln 135, Col 1