2. Indique cuáles son las relaciones uno a uno, uno a muchos, muchos a muchos

Relaciones 1 - 1
NO se encontraron

Relaciones M - n
Film - Category
Film - Actor
Film - Language

Relaciones 1 - N

Film - Inventory
Store - Inventory
Inventory - Rental
Rental - Payment
Customer - Payment
Customer - Rental
Staff - Payment
Staff - Rental
Store - Customer
Address - Customer
Address - Store
Staff - Store (Manager_staff_id)
Store - Staff
Country - City
City - Address

3. Cuales son las claves primarias y foráneas de las tablas: Actor, Film, Customer, Staff

ACTOR
PK actor_id

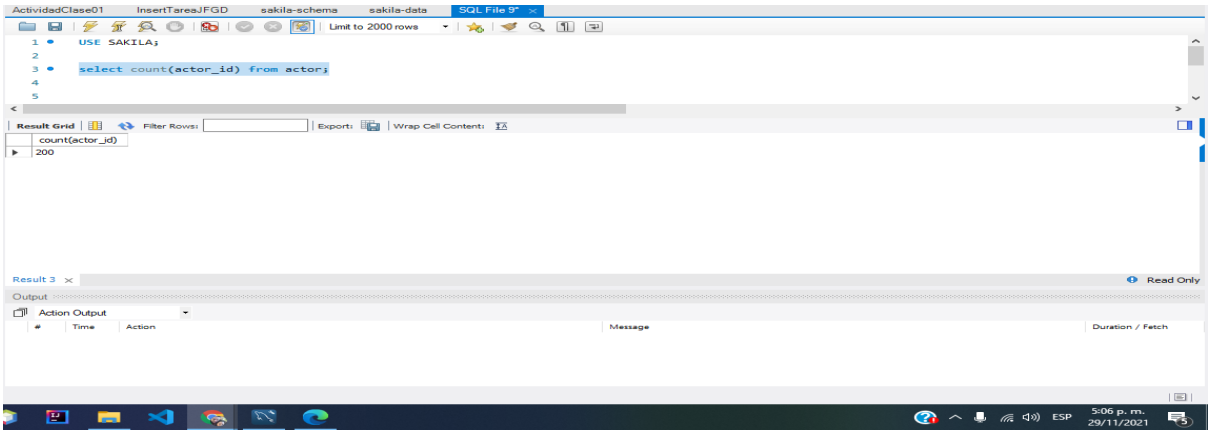
Film
PK film_id
FK language_id
FK language_language_id

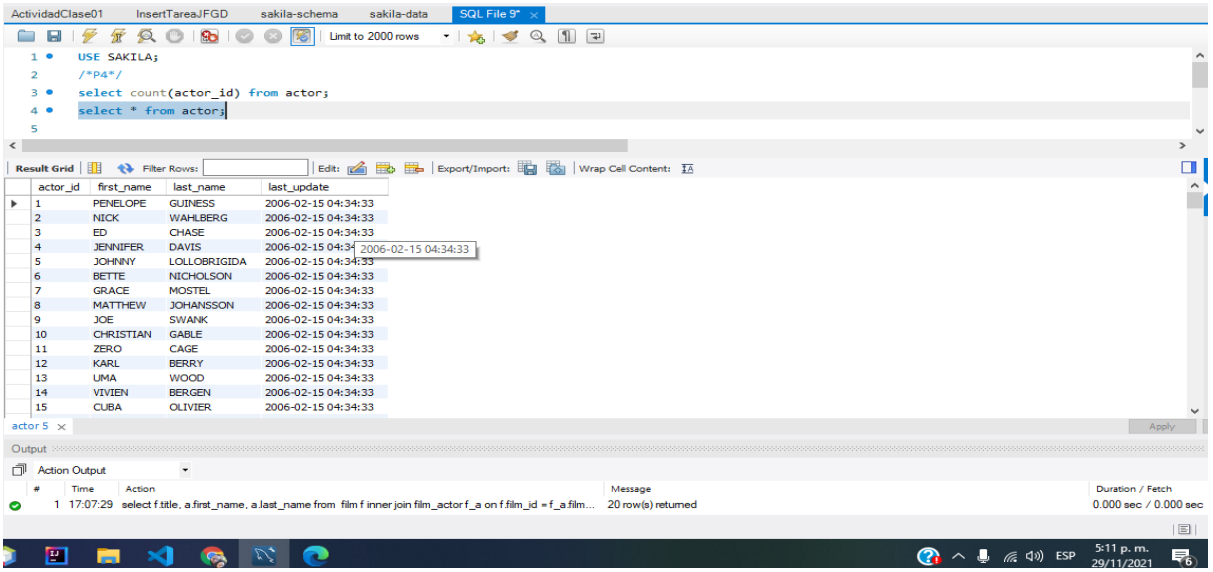
Customer
PK customer_id
FK store_id
FK address_id

Staff
PK staff_id
FK address_id
FK store_id

4 Contar y consultar todos los actores que existen en la BD usando sentencias SQL.

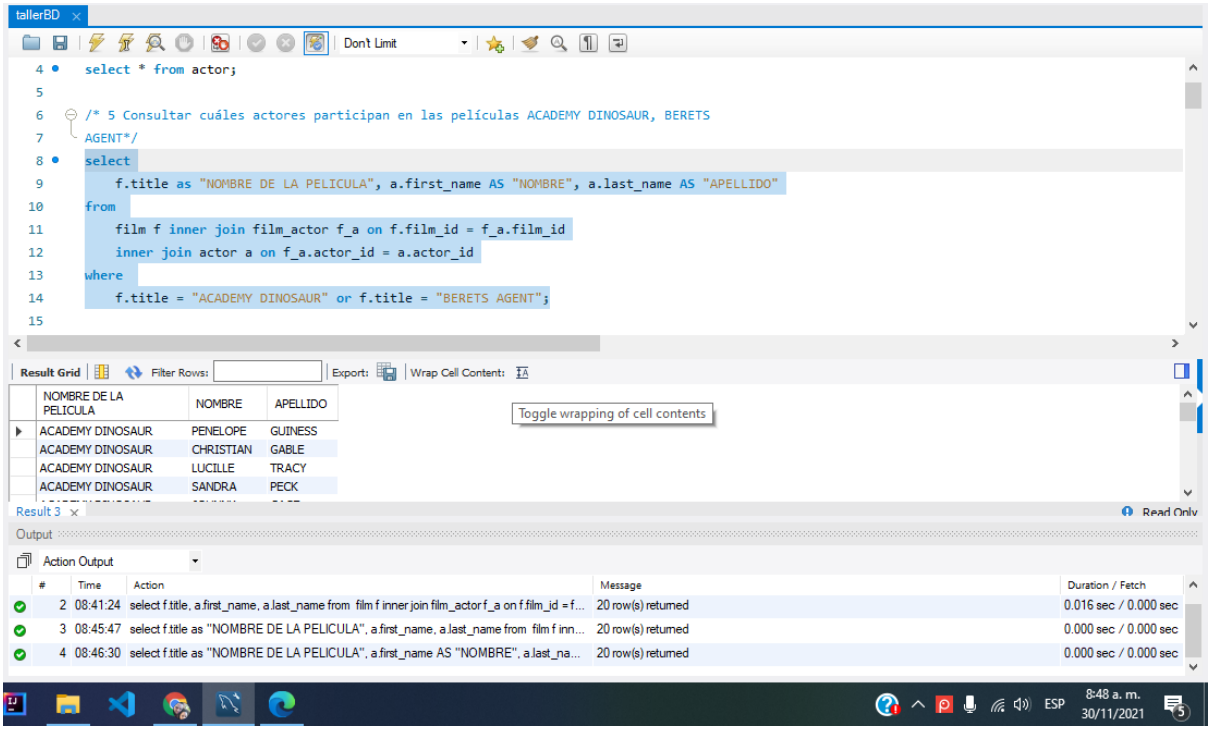
select count(actor_id) from actor;





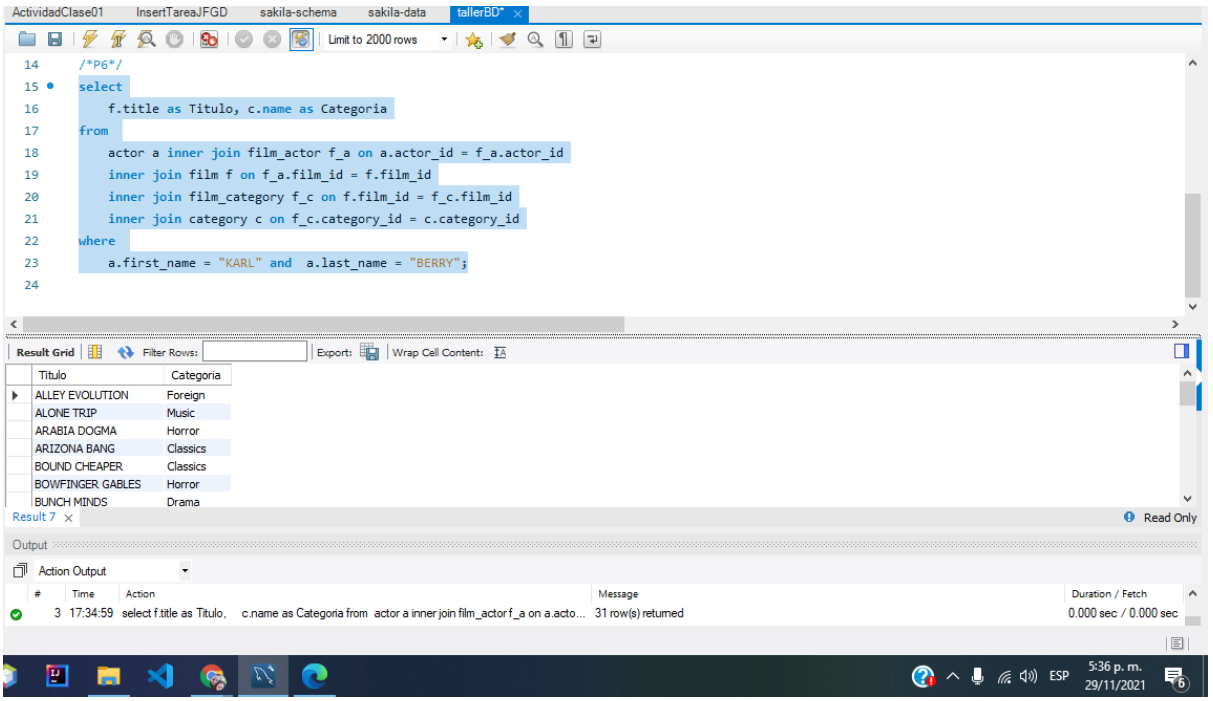
5 Consultar cuáles actores participan en las películas ACADEMY DINOSAUR, BERETS AGENT

```
select
        f.title as "NOMBRE DE LA PELICULA", a.first_name AS "NOMBRE", a.last_name AS
"APELLIDO"
from
        film f inner join film_actor f_a on f.film_id = f_a.film_id
        inner join actor a on f_a.actor_id = a.actor_id
where
        f.title = "ACADEMY DINOSAUR" or f.title = "BERETS AGENT";
```



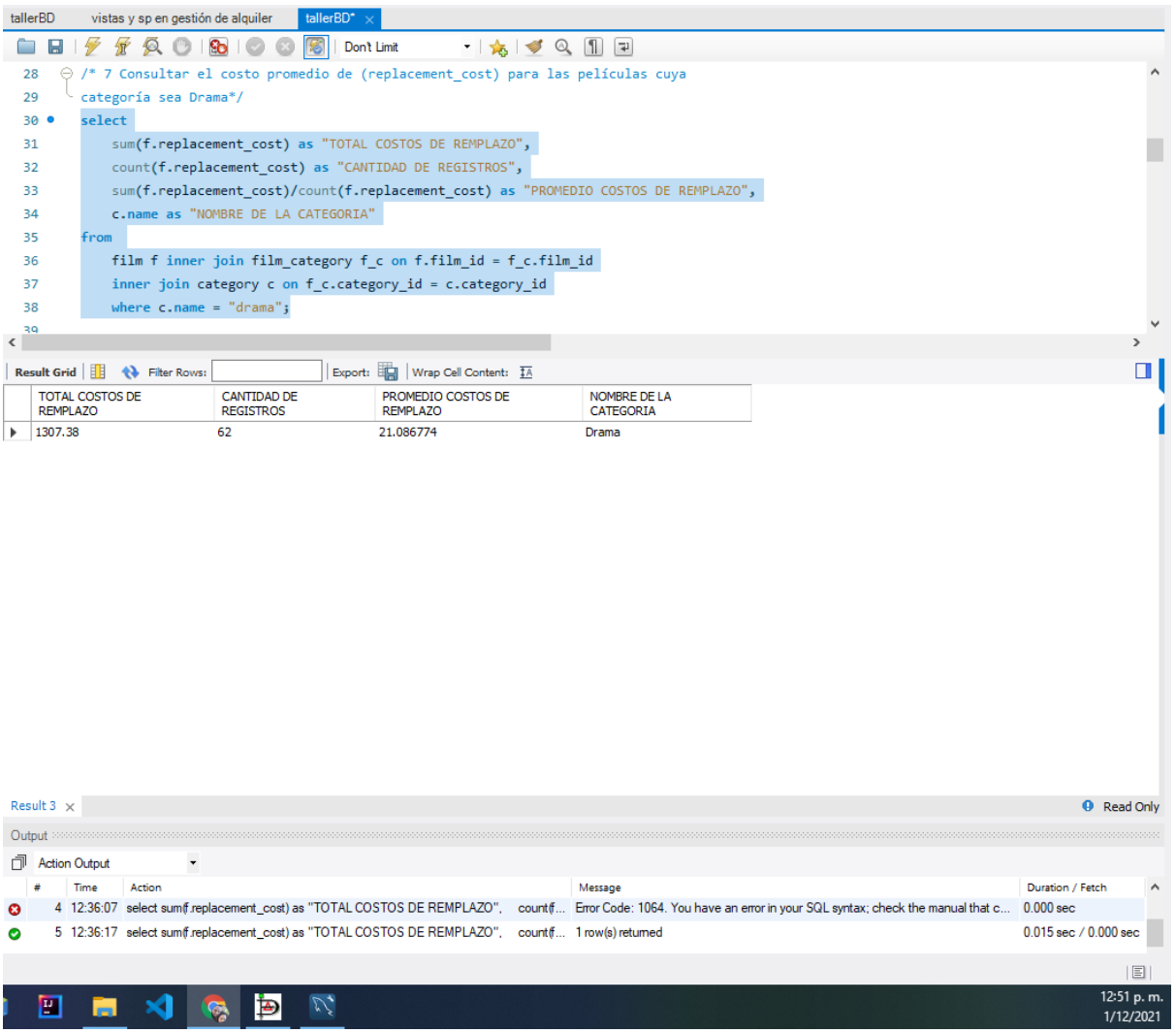
6 Consultar en cuáles películas ha participado KARL BERRY y cual es la categoría de dichas películas.

```
select
        f.title as Titulo, c.name as Categoria
from
        actor a inner join film_actor f_a on a.actor_id = f_a.actor_id
    inner join film f on f_a.film_id = f.film_id
    inner join film_category f_c on f.film_id = f_c.film_id
    inner join category c on f_c.category_id = c.category_id
where
        a.first_name = "KARL" and  a.last_name = "BERRY";
```



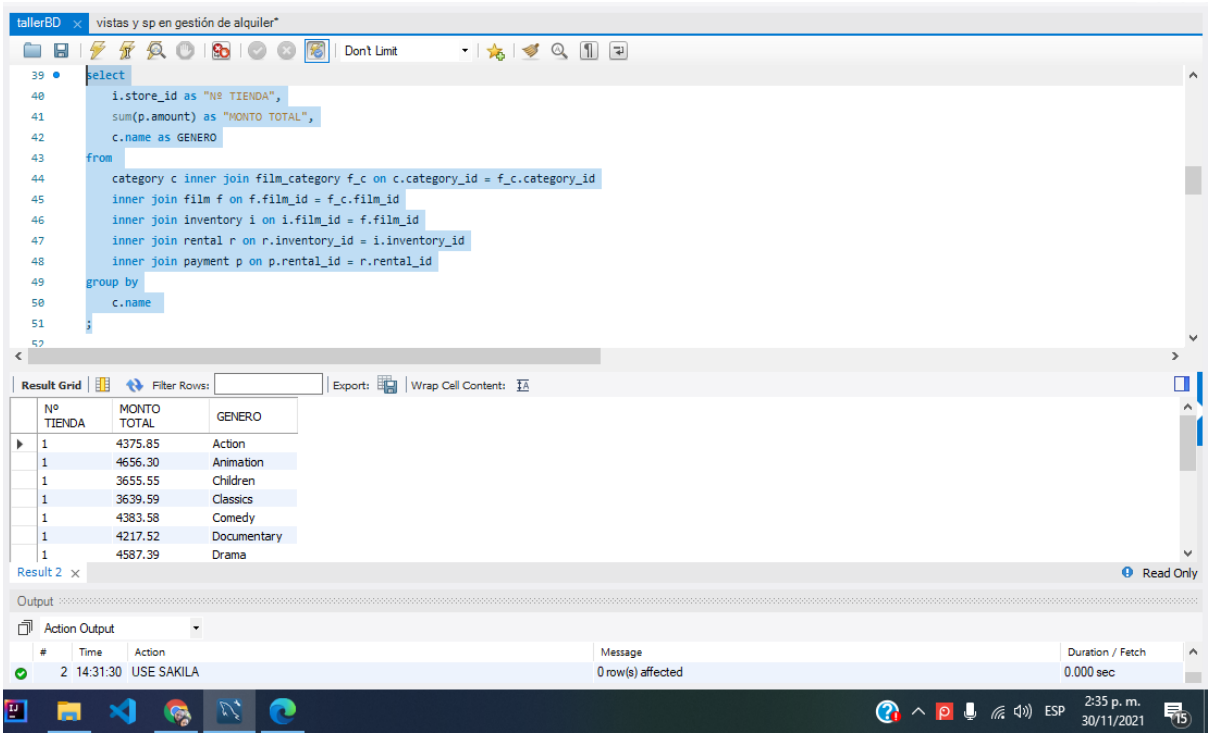
7. Consultar el costo promedio de (replacement_cost) para las películas cuya categoría sea Drama.

```
select
        sum(f.replacement_cost) as "TOTAL COSTOS DE REMPLAZO",
        count(f.replacement_cost) as "CANTIDAD DE REGISTROS",
        sum(f.replacement_cost)/count(f.replacement_cost) as "PROMEDIO COSTOS DE
REMPLAZO",
        c.name as "NOMBRE DE LA CATEGORIA"
from
        film f inner join film_category f_c on f.film_id = f_c.film_id
        inner join category c on f_c.category_id = c.category_id
        where c.name = "drama";
```



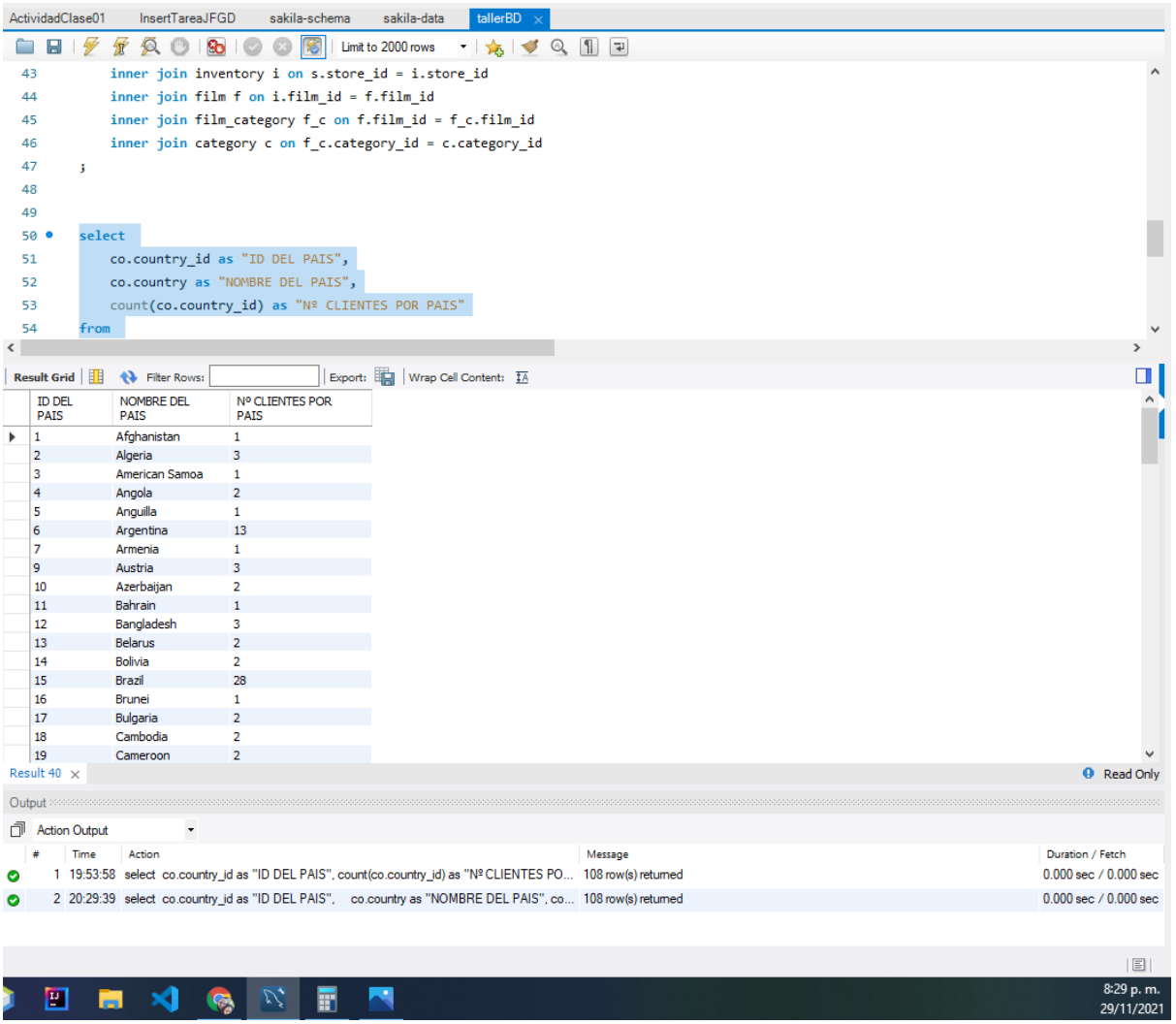
8. Consultar el total de ventas de cada store por categoría de película.

```
select
        i.store_id as "N° TIENDA",
        sum(p.amount) as "MONTO TOTAL",
        c.name as GENERO
from
        category c inner join film_category f_c on c.category_id = f_c.category_id
        inner join film f on f.film_id = f_c.film_id
        inner join inventory i on i.film_id = f.film_id
         inner join rental r on r.inventory_id = i.inventory_id
        inner join payment p on p.rental_id = r.rental_id
        group by
        c.name
        order by i.store_id
;
```



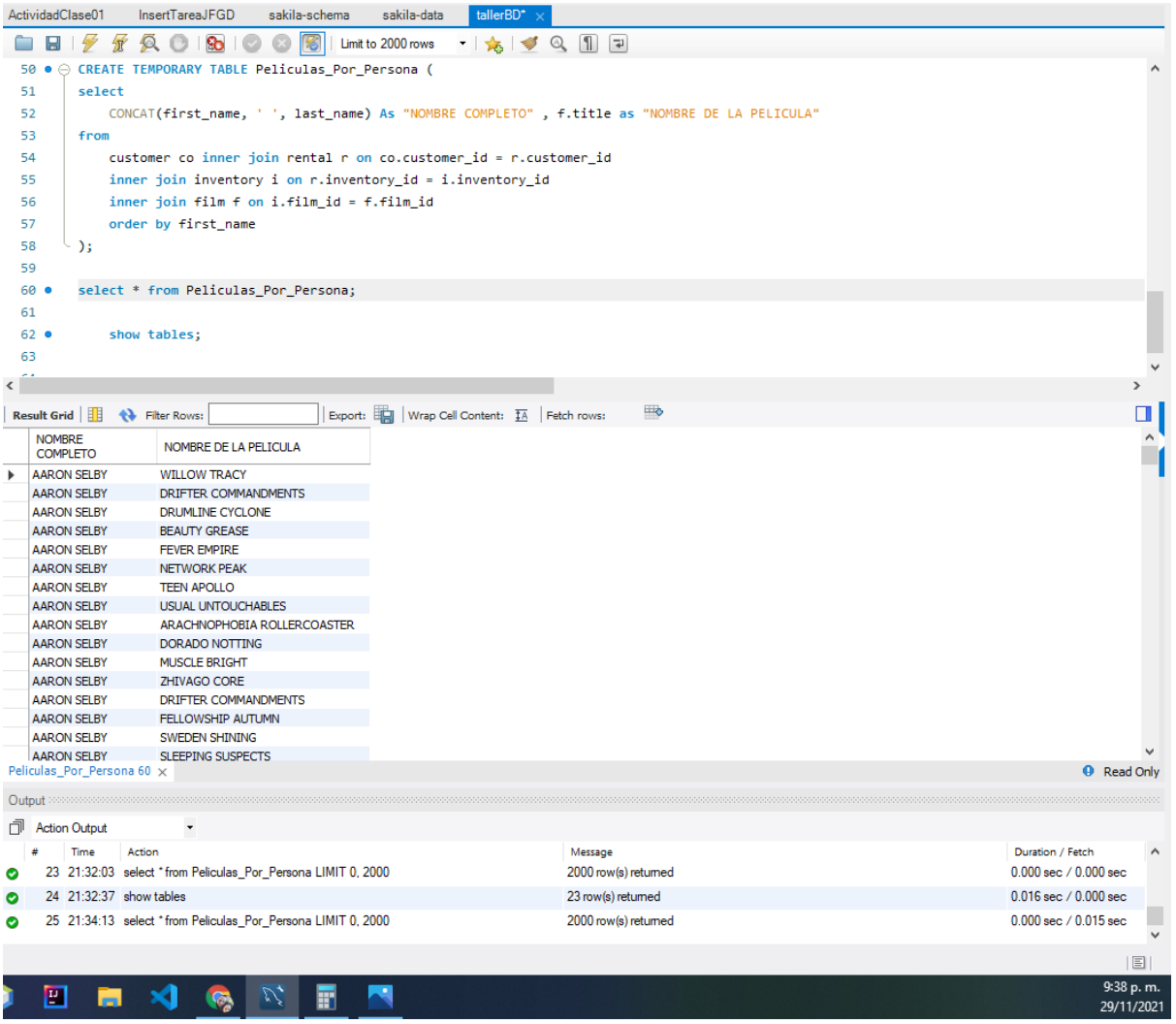
9. Contar todos los clientes existentes por País

```
select
        co.country_id as "ID DEL PAIS",
        count(co.country_id) as "Nº CLIENTES POR PAIS",
   co.country as "NOMBRE DEL PAIS"

from
        country co inner join city ci on co.country_id = ci.country_id
   inner join address ad on ci.city_id = ad.city_id
   inner join customer cu on ad.address_id = cu.address_id
   group by co.country_id;
```



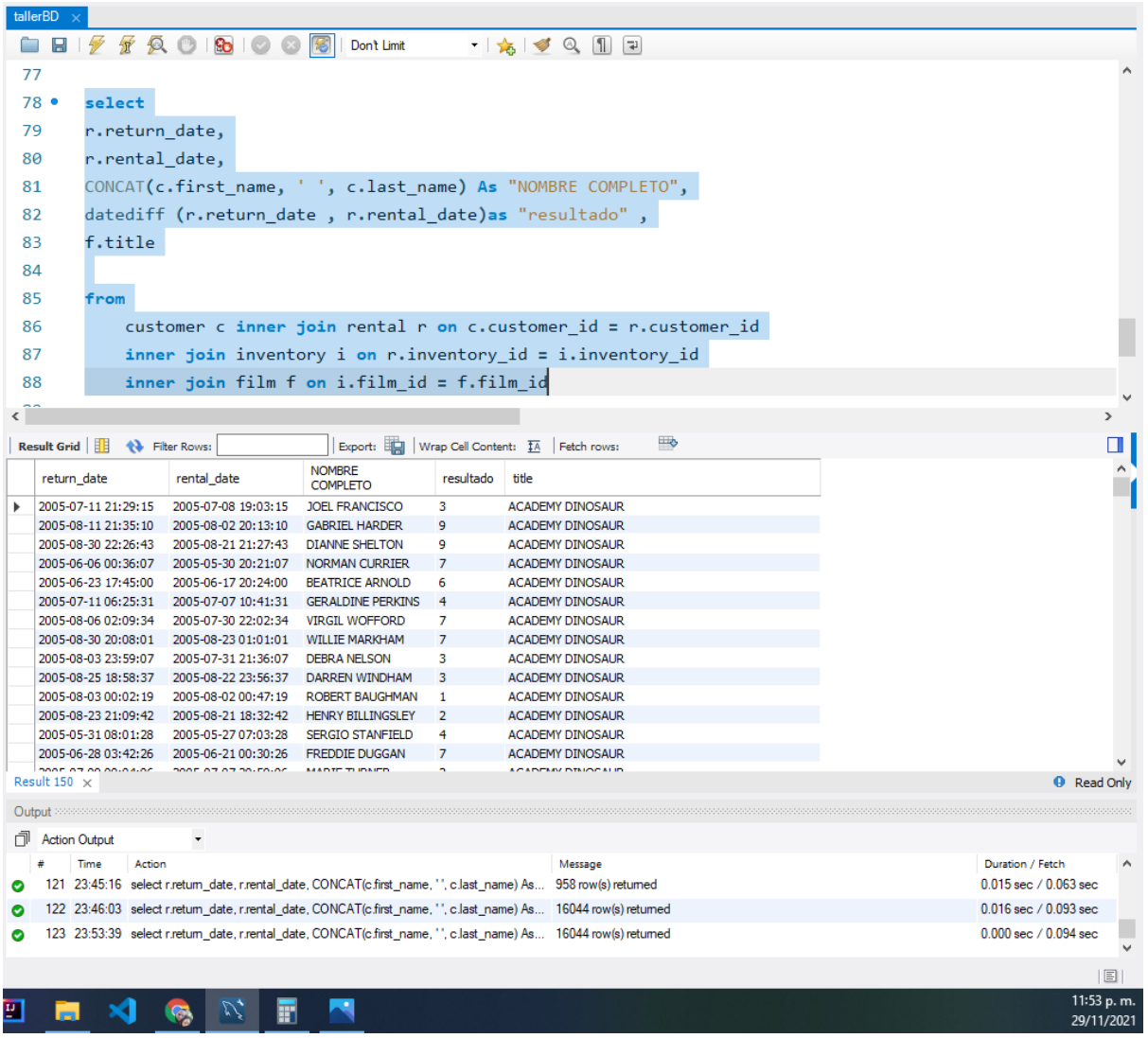
10 . Crear una tabla temporal que contenga las películas que cada cliente ha alquilado.

```
CREATE TEMPORARY TABLE Peliculas_Por_Persona (
select
        CONCAT(first_name, ' ', last_name) As "NOMBRE COMPLETO" , f.title as "NOMBRE DE LA PELICULA"
from
        customer co inner join rental r on co.customer_id = r.customer_id
        inner join inventory i on r.inventory_id = i.inventory_id
        inner join film f on i.film_id = f.film_id
        order by first_name
);

select * from Peliculas_Por_Persona;
```



11. Consulte el tiempo máximo en días que cada cliente ha tenido una película.

```
select
r.return_date,
r.rental_date,
CONCAT(c.first_name, ' ', c.last_name) As "NOMBRE COMPLETO",
datediff (r.return_date , r.rental_date)as "resultado" ,
f.title

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
  inner join film f on i.film_id = f.film_id
```



return_date	rental_date	NOMBRE COMPLETO	resultado	title
2005-07-11 21:29:15	2005-07-08 19:03:15	JOEL FRANCISCO	3	ACADEMY DINOSAUR
2005-08-11 21:35:10	2005-08-02 20:13:10	GABRIEL HARDER	9	ACADEMY DINOSAUR
2005-08-30 22:26:43	2005-08-21 21:27:43	DIANNE SHELTON	9	ACADEMY DINOSAUR
2005-06-06 00:36:07	2005-05-30 20:21:07	NORMAN CURRIER	7	ACADEMY DINOSAUR
2005-06-23 17:45:00	2005-06-17 20:24:00	BEATRICE ARNOLD	6	ACADEMY DINOSAUR
2005-07-11 06:25:31	2005-07-07 10:41:31	GERALDINE PERKINS	4	ACADEMY DINOSAUR
2005-08-06 02:09:34	2005-07-30 22:02:34	VIRGIL WOFFORD	7	ACADEMY DINOSAUR
2005-08-30 20:08:01	2005-08-23 01:01:01	WILLIE MARKHAM	7	ACADEMY DINOSAUR
2005-08-03 23:59:07	2005-07-31 21:36:07	DEBRA NELSON	3	ACADEMY DINOSAUR
2005-08-25 18:58:37	2005-08-22 23:56:37	DARREN WINDHAM	3	ACADEMY DINOSAUR
2005-08-03 00:02:19	2005-08-02 00:47:19	ROBERT BAUGHMAN	1	ACADEMY DINOSAUR
2005-08-23 21:09:42	2005-08-21 18:32:42	HENRY BILLINGSLEY	2	ACADEMY DINOSAUR
2005-05-31 08:01:28	2005-05-27 07:03:28	SERGIO STANFIELD	4	ACADEMY DINOSAUR
2005-06-28 03:42:26	2005-06-21 00:30:26	FREDDIE DUGGAN	7	ACADEMY DINOSAUR

#	Time	Action	Message	Duration / Fetch
121	23:45:16	select r.return_date, r.rental_date, CONCAT(c.first_name, ' ', c.last_name) As...	958 row(s) returned	0.015 sec / 0.063 sec
122	23:46:03	select r.return_date, r.rental_date, CONCAT(c.first_name, ' ', c.last_name) As...	16044 row(s) returned	0.016 sec / 0.093 sec
123	23:53:39	select r.return_date, r.rental_date, CONCAT(c.first_name, ' ', c.last_name) As...	16044 row(s) returned	0.000 sec / 0.094 sec