

# Informe de Evidencias Reto Bases de Datos

Nombre: Julian Alberto Avila Navarro

Fecha: 02/12/2021

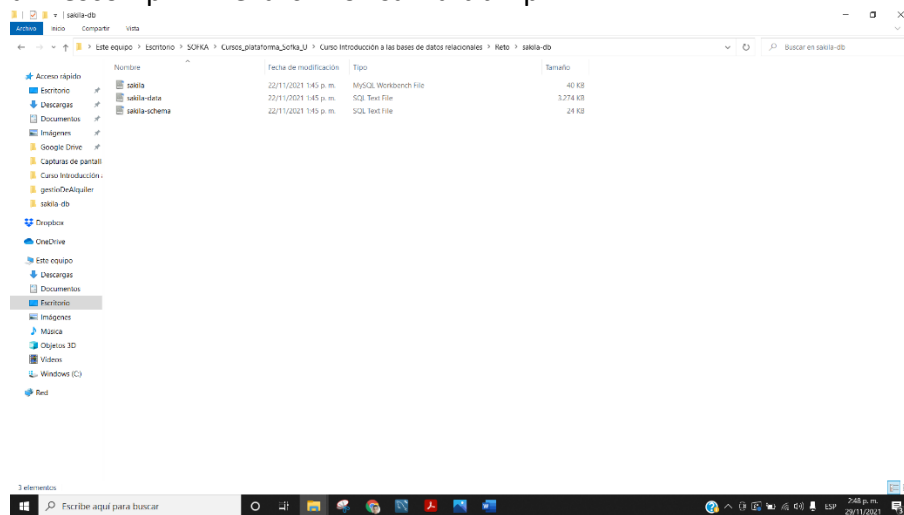
## Training QA C4

Nota: Se anexan los archivos necesarios para el desarrollo de los diferentes puntos y la creación de la base de datos.

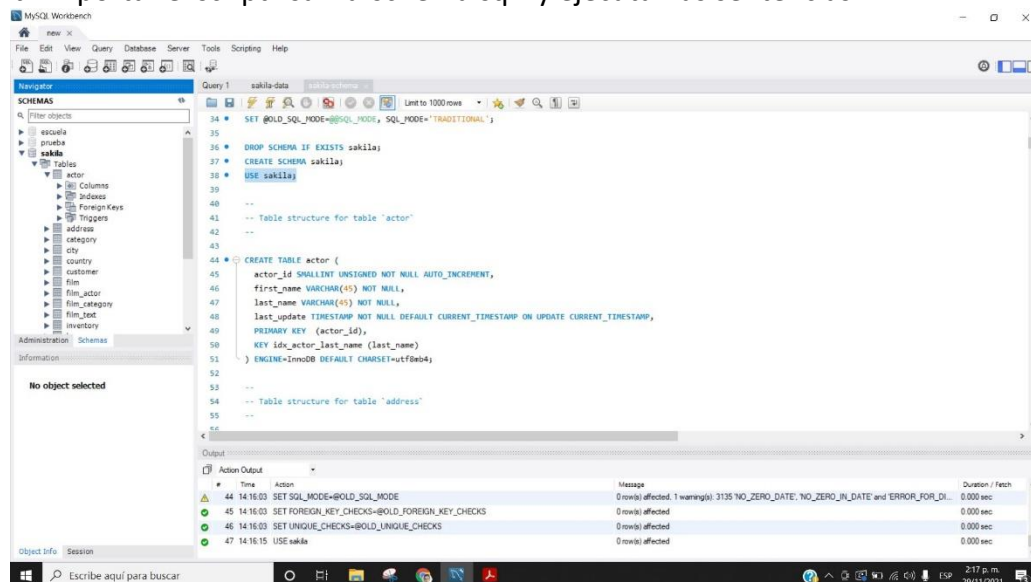
### Punto 1

En una conexión local de base de datos, crear la base de datos llamada Sakila:

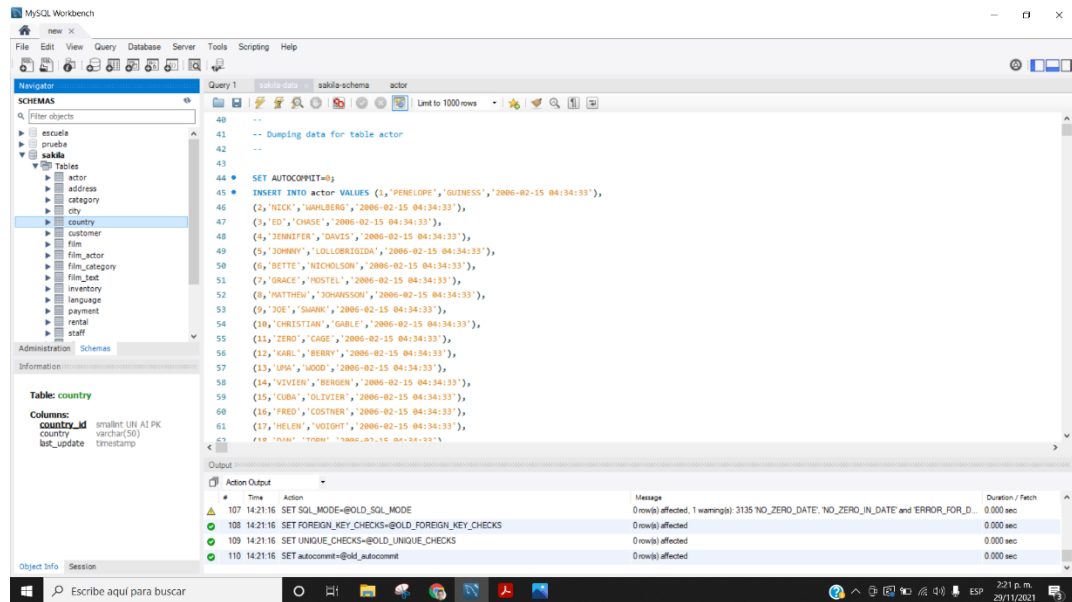
a. Descomprimir el archivo “sakila-db.zip”.



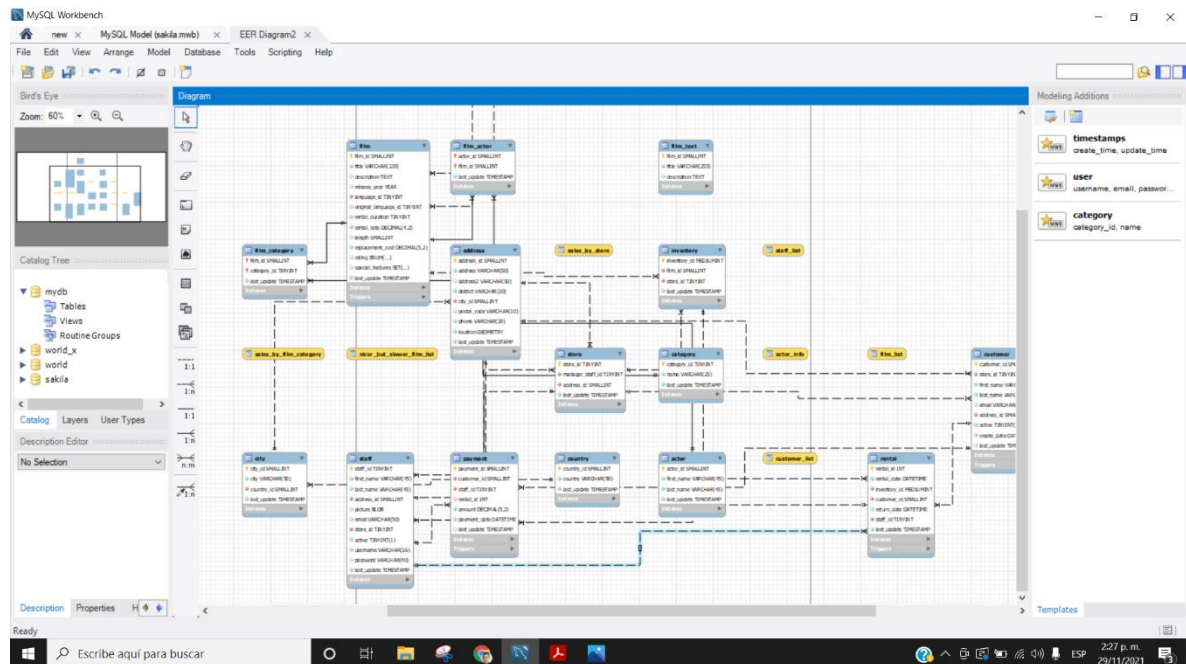
b. Importar el script “sakila-schema.sql” y ejecutar las sentencias.



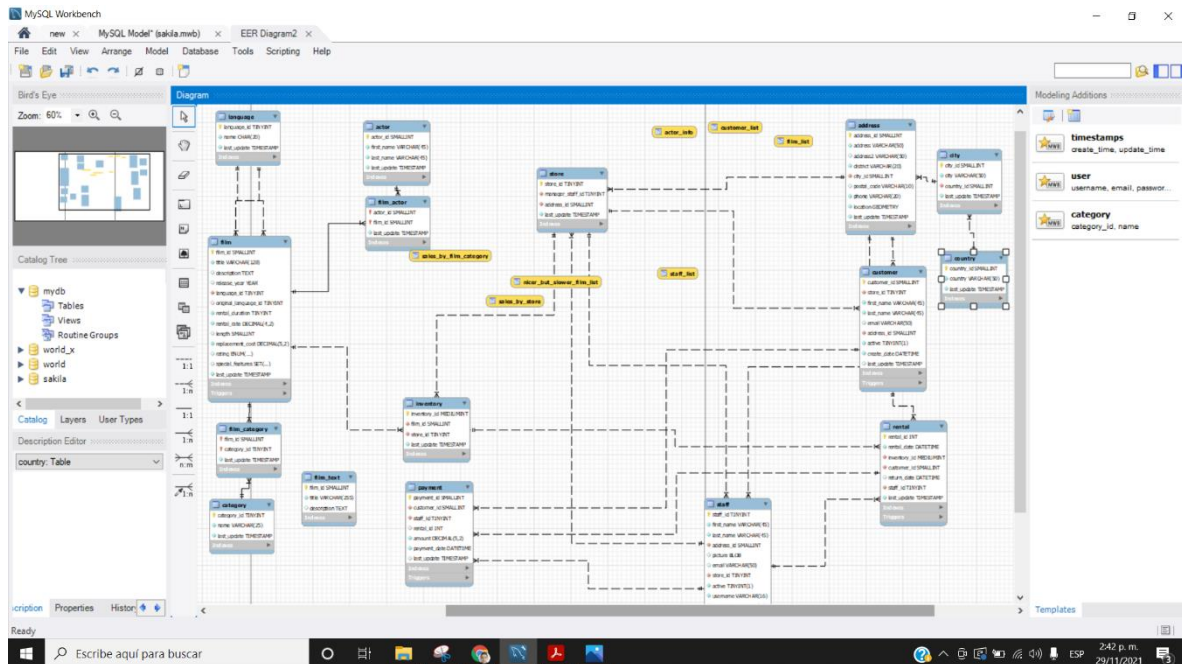
c. Importar el script “sakila-data.sql” y ejecutar las sentencias.



d. Ejecutar el archivo “sakila.mwb” para análisis posteriores.



Ajuste versión



## Punto 2.

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

### Relaciones M:N

- Film y category (Tabla intermedia film\_category)
- Film y actor (Tabla intermedia film\_actor)

### Relaciones 1 a muchos

- Inventory a film
- Inventory y store
- Inventory y Rental
- Rental y Payment
- Rental y staff
- Rental y customer
- Staff y Payment
- Staff y Store
- Staff y Address
- Customer y Store
- Store y Address
- Country y city
- Language y Film
- Address y Customer
- Store y Staff
- Customer y payment

## Relaciones 1:1

No se tienen

## Punto 3

### Claves Primarias:

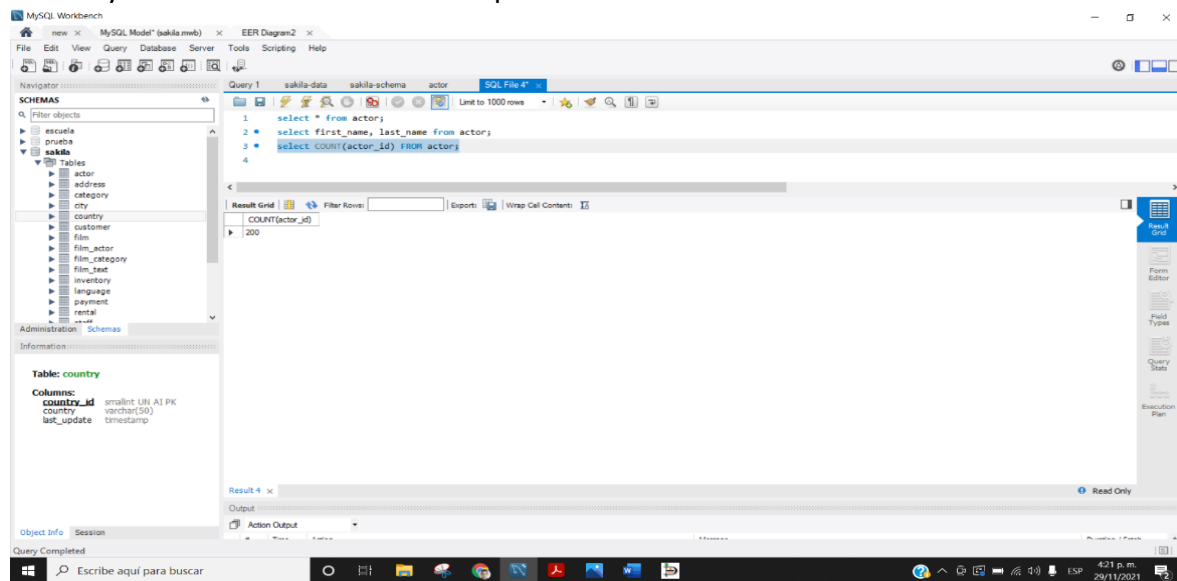
- actor\_id
- film\_id
- customer\_id
- staff\_id

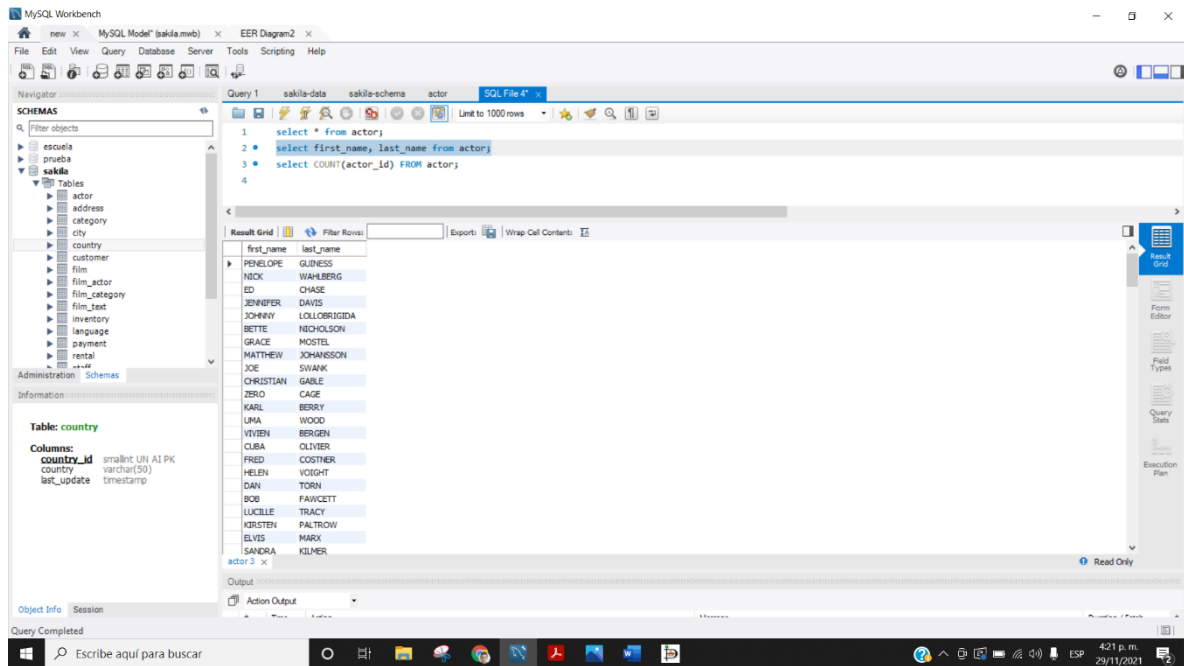
### Claves Foráneas:

- language\_id
- Original\_Language\_id
- store\_id
- address\_id

## Punto 4

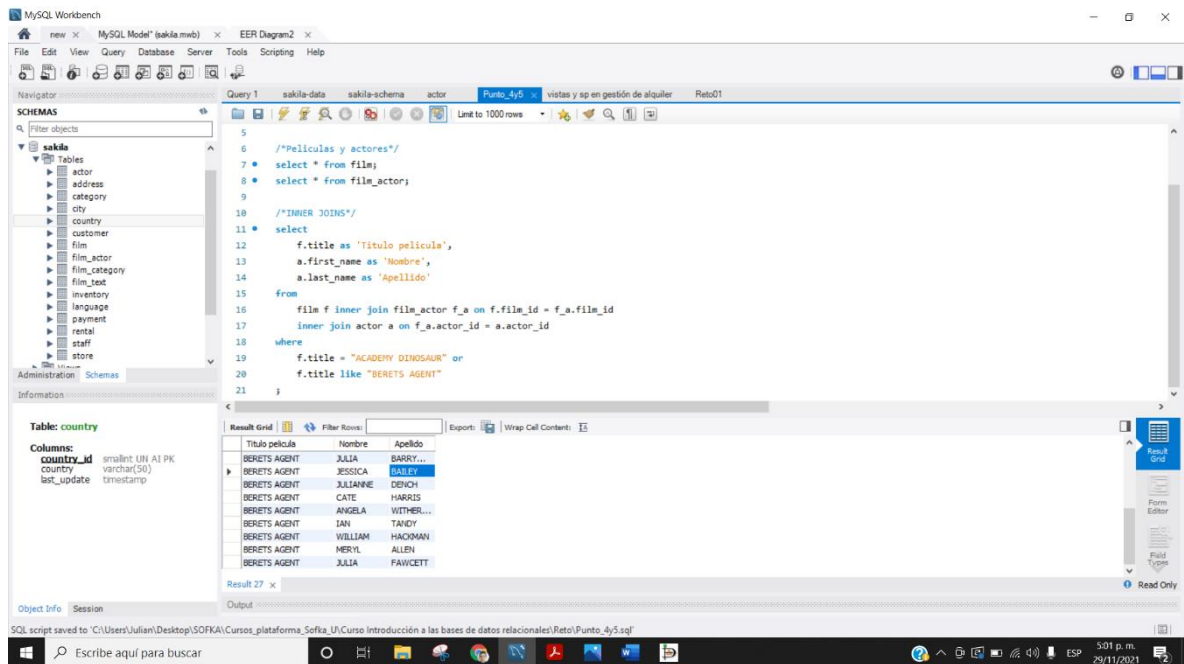
Contar y consultar todos los actores que existen en la BD usando sentencias SQL.





## Punto 5

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



<b>Titulo película</b>	<b>Nombre</b>	<b>Apellido</b>
ACADEMY DINOSAUR	PENELOPE	GUINESS

ACADEMY DINOSAUR	CHRISTIAN	GABLE
ACADEMY DINOSAUR	LUCILLE	TRACY
ACADEMY DINOSAUR	SANDRA	PECK
ACADEMY DINOSAUR	JOHNNY	CAGE
ACADEMY DINOSAUR	MENA	TEMPLE
ACADEMY DINOSAUR	WARREN	NOLTE
ACADEMY DINOSAUR	OPRAH	KILMER
ACADEMY DINOSAUR	ROCK	DUKAKIS
ACADEMY DINOSAUR	MARY	KEITEL
BERETS AGENT	GRACE	MOSTEL
BERETS AGENT	JULIA	BARRYMORE
BERETS AGENT	JESSICA	BAILEY
BERETS AGENT	JULIANNE	DENCH
BERETS AGENT	CATE	HARRIS
BERETS AGENT	ANGELA	WITHERSPOON
BERETS AGENT	IAN	TANDY
BERETS AGENT	WILLIAM	HACKMAN
BERETS AGENT	MERYL	ALLEN
BERETS AGENT	JULIA	FAWCETT

## Punto 6

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

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left displays the 'sakila' database structure, including tables like 'actor', 'film', and 'film\_category'. The 'Query' pane in the center contains the following SQL query:

```

33 # punto 6
34 select
35     a.first_name as 'Nombre',
36     a.last_name as 'Apellido',
37     f.title as 'Película',
38     c.name as 'Categoría'
39 from
40     actor a inner join film_actor f_a on a.actor_id = f_a.actor_id
41     inner join film f on f_a.film_id = f.film_id
42     inner join film_category f_c on f.film_id = f_c.film_id
43     inner join category c on f_c.category_id = c.category_id
44 where
45     a.first_name = "KARL" and
46     a.last_name = "BERRY"
47 ;

```

The 'Result Grid' at the bottom displays the query results for Karl Berry's filmography:

Nombre	Apellido	Película	Categoría
KARL	BERRY	ALLEY EVOLUTION	Foreign
KARL	BERRY	ALONE TRIP	Music
KARL	BERRY	ARABIA DOGMA	Horror
KARL	BERRY	ARIZONA BANG	Classics
KARL	BERRY	BOUND CHEAPER	Classics
KARL	BERRY	BOWFINGER GABLES	Horror
KARL	BERRY	BUNCH MINDS	Drama
KARL	BERRY	CLEOPATRA DEVIL	New
KARL	BERRY	CONNECTICUT TRAMP	Sci-Fi

The status bar at the bottom indicates that 34 rows were returned. The system tray shows the date as 29/11/2021.

Nombre	Apellido	Película	Categoría
KARL	BERRY	ALLEY EVOLUTION	Foreign
KARL	BERRY	ALONE TRIP	Music
KARL	BERRY	ARABIA DOGMA	Horror
KARL	BERRY	ARIZONA BANG	Classics
KARL	BERRY	BOUND CHEAPER	Classics
KARL	BERRY	BOWFINGER GABLES	Horror
KARL	BERRY	BUNCH MINDS	Drama
KARL	BERRY	CLEOPATRA DEVIL	New
KARL	BERRY	CONNECTICUT TRAMP	Sci-Fi
KARL	BERRY	DARES PLUTO	Animation
KARL	BERRY	DATE SPEED	Family
KARL	BERRY	DAY UNFAITHFUL	New
KARL	BERRY	DOORS PRESIDENT	Animation
KARL	BERRY	FURY MURDER	Children
KARL	BERRY	HARDLY ROBBERS	Documentary
KARL	BERRY	HIGHBALL POTTER	Foreign
KARL	BERRY	HOLES BRANNIGAN	Sports
KARL	BERRY	INDEPENDENCE HOTEL	Documentary
KARL	BERRY	LEATHERNECKS DWARFS	Travel
KARL	BERRY	LUCKY FLYING	Music
KARL	BERRY	MONTEREY LABYRINTH	Horror
KARL	BERRY	NOVOCAINE FLIGHT	Sports
KARL	BERRY	OKLAHOMA JUMANJI	New
KARL	BERRY	PERFECT GROOVE	Comedy
KARL	BERRY	REAP UNFAITHFUL	Horror
KARL	BERRY	REUNION WITCHES	Drama
KARL	BERRY	SMOKING BARBARELLA	Documentary
KARL	BERRY	STAGECOACH ARMAGEDDON	Action
KARL	BERRY	SWEDEN SHINING	Comedy
KARL	BERRY	TELEMARK HEARTBREAKERS	Animation
KARL	BERRY	VIRGINIAN PLUTO	Documentary

### Punto 7

Consultar el costo promedio de (replacement\_cost) para las películas cuya categoría sea Drama.



MySQL Workbench

new x sofia-training.cpxhmdh1ok.us-e-x MySQL Model x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHEMAS

- ingreso
- paciente
- receta
- servicio
- Views
  - vista\_gastos\_paciente
  - vista\_gastos\_servicios
- Stored Procedures
  - sp\_insertar\_PACIENTES
  - sp\_listar\_pacientes
  - sp\_listar\_servicios
- Functions
  - prueba
- sakila
  - Tables
  - Views

Administration Schemas

Information

No object selected

Result Grid

Costo total	Cantidad	Suma Promedio
1307.38	62	21.086774

Result 2 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
618	18:12:37	use sakila	0 row(s) affected	0.016 sec
619	18:12:44	select sum(f.replacement_cost) as "Costo total", count(f.replacement_cost) as Cantidad, sum(f.replacement_cost)/count(f.replacement_cost) as "Suma Promedio"	1 row(s) returned	0.047 sec / 0.000 sec

Query Completed

Escribe aquí para buscar

## Punto 8

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

MySQL Workbench

new x sofia-training.cpxhmdh1ok.us-e-x MySQL Model x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHEMAS

- ingreso
- paciente
- receta
- servicio
- Views
  - vista\_gastos\_paciente
  - vista\_gastos\_servicios
- Stored Procedures
  - sp\_insertar\_PACIENTES
  - sp\_listar\_pacientes
  - sp\_listar\_servicios
- Functions
  - prueba
- sakila
  - Tables
  - Views

Administration Schemas

Information

No object selected

Result Grid

Store	Total ventas store	Categoría
1	4373.85	Action
1	4656.30	Animation
1	3655.55	Children
1	3636.59	Classics
1	4383.58	Comedy
1	4217.52	Documentary
1	4587.39	Drama
2	4226.07	Famly

Result 3 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
619	18:12:44	select sum(f.replacement_cost) as "Costo total", count(f.replacement_cost) as Cantidad, sum(f.replacement_cost)/count(f.replacement_cost) as "Suma Promedio"	1 row(s) returned	0.047 sec / 0.000 sec
620	18:14:37	select sum(p.amount) as "Total ventas store", c.name as "Categoría"	1 row(s) returned	0.000 sec / 0.000 sec

Query Completed

Escribe aquí para buscar

## Punto 9

Contar todos los clientes existentes por País.



MySQL Workbench

Schema: Scripts\_Reto1 Reto01 julianAvila\_01\_12\_2021 Datos\_DB\_julianAvila CRUD\_DB\_julianAvila CRUD\_Tablas\_Maestras\_julian... Vistas\_DB\_julianAvila Procedimientos\_Almacenados...

Filter objects

Schemas

Filter objects

Schema: Scripts\_Reto1 Reto01 julianAvila\_01\_12\_2021 Datos\_DB\_julianAvila CRUD\_DB\_julianAvila CRUD\_Tablas\_Maestras\_julian... Vistas\_DB\_julianAvila Procedimientos\_Almacenados...

Limit to 1000 rows

Result Grid

Pais	Cantidad clientes
Afghanistan	1
Algeria	3
American Samoa	1
Angola	2
Anguilla	1
Argentina	13
Armenia	1
Austria	3

Output

Action Output

Message

Duration / Fetch

0.000 sec / 0.000 sec

0.313 sec / 0.000 sec

Query Completed

Escribe aqui para buscar

## Punto 10

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

MySQL Workbench

Schema: Scripts\_Reto1 Reto01 julianAvila\_01\_12\_2021 Datos\_DB\_julianAvila CRUD\_DB\_julianAvila CRUD\_Tablas\_Maestras\_julian... Vistas\_DB\_julianAvila Procedimientos\_Almacenados...

Filter objects

Schemas

Filter objects

Schema: Scripts\_Reto1 Reto01 julianAvila\_01\_12\_2021 Datos\_DB\_julianAvila CRUD\_DB\_julianAvila CRUD\_Tablas\_Maestras\_julian... Vistas\_DB\_julianAvila Procedimientos\_Almacenados...

Limit to 1000 rows

Result Grid

Nombre Cliente	Película
AARON SELBY	DORADO NOTTING
AARON SELBY	FELLOWSHIP AUTUMN
AARON SELBY	DRIFTED COMMANDMENTS
AARON SELBY	ZHEVAGO CORE
AARON SELBY	MUSCLE BRIGHT
AARON SELBY	ARACHNOPHOBIA ROLLERCOASTER
AARON SELBY	SWEDEN SHINING
AARON SELBY	CORE SUIT

Output

Action Output

Message

Duration / Fetch

0.015 sec

0.463 sec

Query Completed

Escribe aqui para buscar

## Punto 11

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

MySQL Workbench

Schema: Scripts\_Reto01

Query:

```

100 select * from Peliculas_Alquiladas_P;
101
102 # punto 11
103
104 select
105     concat(first_name, ' ', last_name) As "Nombre Cliente",
106     datediff(r.return_date, r.rental_date) as Tiempo_Dias,
107     f.title as Titulo
108 from
109     customer c inner join rental r on c.customer_id = r.customer_id
110     inner join inventory i on r.inventory_id = i.inventory_id
111     inner join film f on i.film_id = f.film_id
112 order by first_name
113

```

Result Grid:

Nombre Cliente	Tiempo_Dias	Titulo
AARON SELBY	8	VALENTINE VANDISHING
AARON SELBY	5	SWIDEN SHINING
AARON SELBY	8	ZHIVAGO CORE
AARON SELBY	6	WILLOW TRACY
AARON SELBY	8	TEEN APOLLO
AARON SELBY	1	USUAL UNTOUCHABLES
AARON SELBY	6	SLEEPING SUSPECTS
AARON SELBY	7	PERFECT GROOVE

Output:

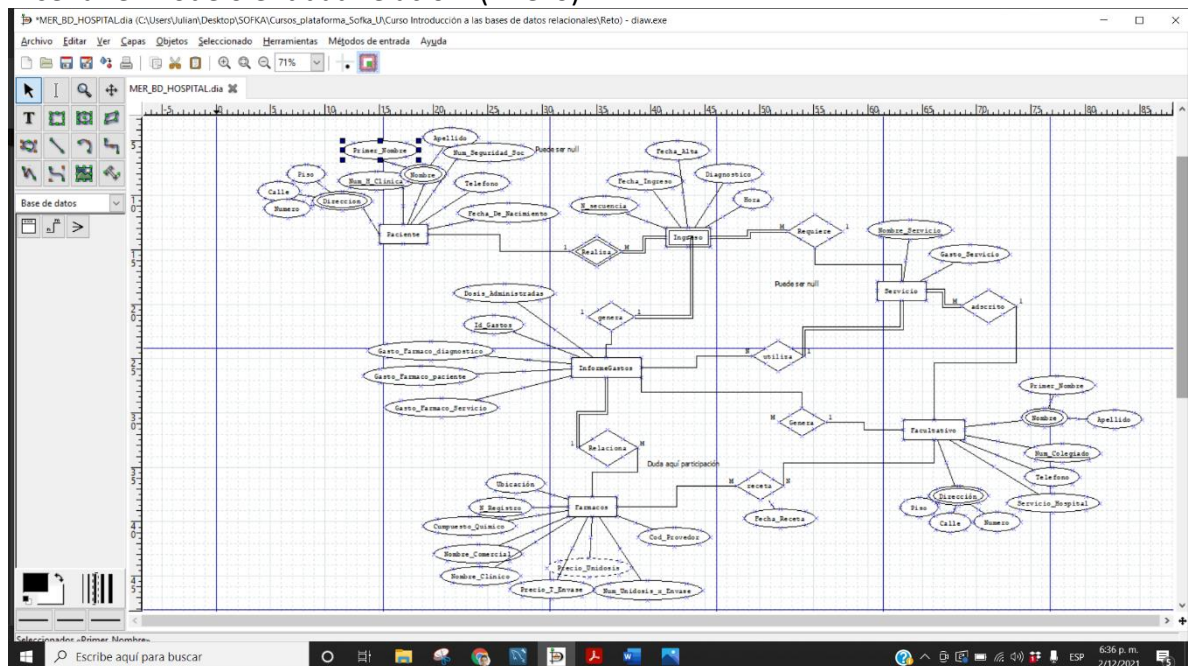
626 18:32:00 select concat(first\_name, ' ', last\_name) As "Nombre Cliente", datediff(r.return\_date, r.rental\_date) as re... 1000 row(s) returned 0.156 sec / 0.000 sec

627 18:32:09 select concat(first\_name, ' ', last\_name) As "Nombre Cliente", datediff(r.return\_date, r.rental\_date) as re... 1000 row(s) returned 0.172 sec / 0.000 sec

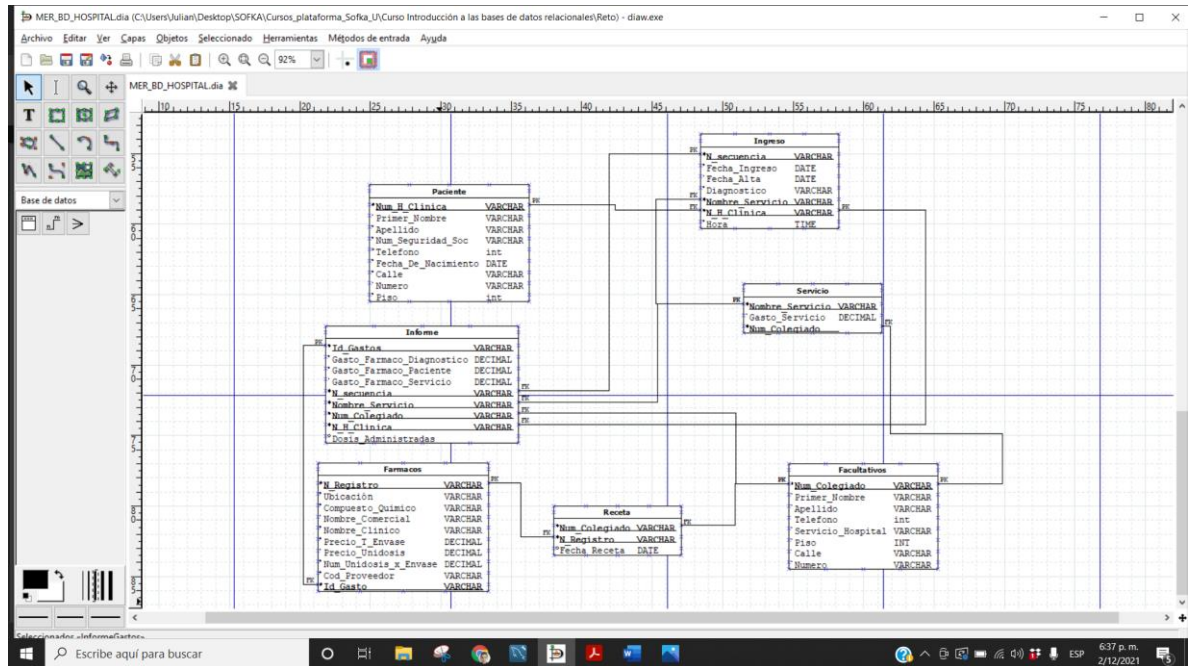
## Punto 12

Realice el siguiente ejercicio de acuerdo a la siguiente definición de negocio y tener presente:

### a. Diseñar el modelo entidad relación. (Anexo)



### b. Crear el modelo lógico debidamente normalizado.



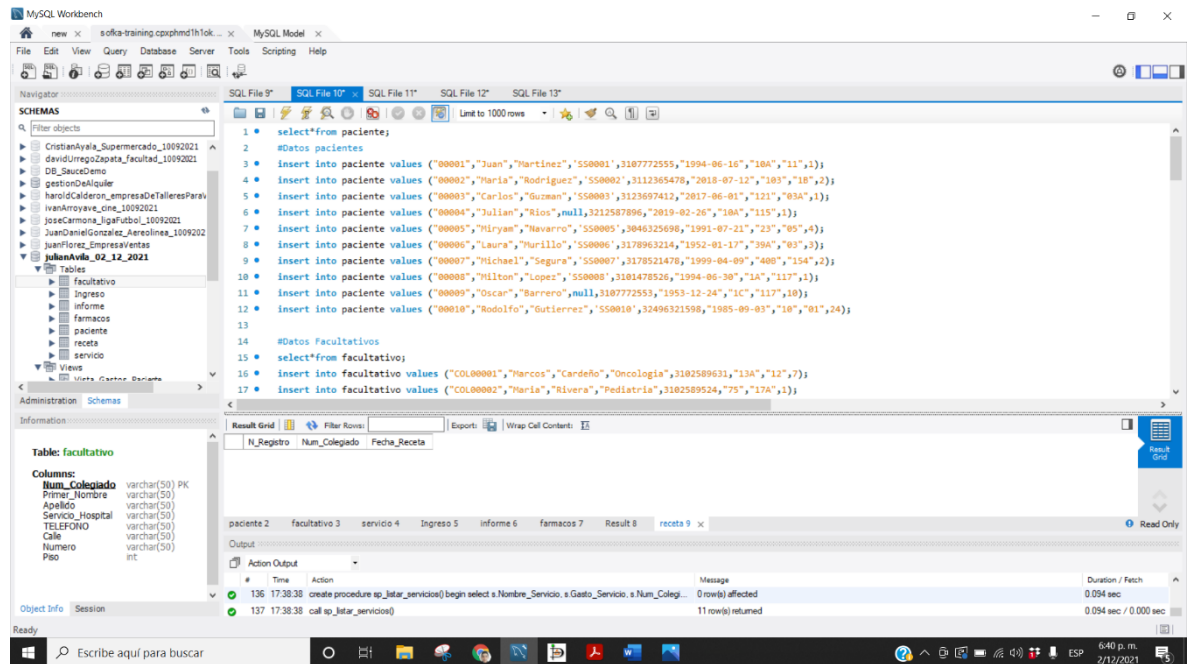
- c. Crear la base de datos en la base de datos remota de Sofka U (TENER EN CUENTA QUE EL NOMBRE DE LA BASE DE DATOS TENDRÁ COMO FORMATO EL PRIMER NOMBRE + PRIMER APELLIDO + FECHA EN FORMATO \_DD\_MM\_YYY. Ejemplo: ivanArroyave\_26\_11\_2021)

Se anexa el archivo SQL donde se crea la base de datos.

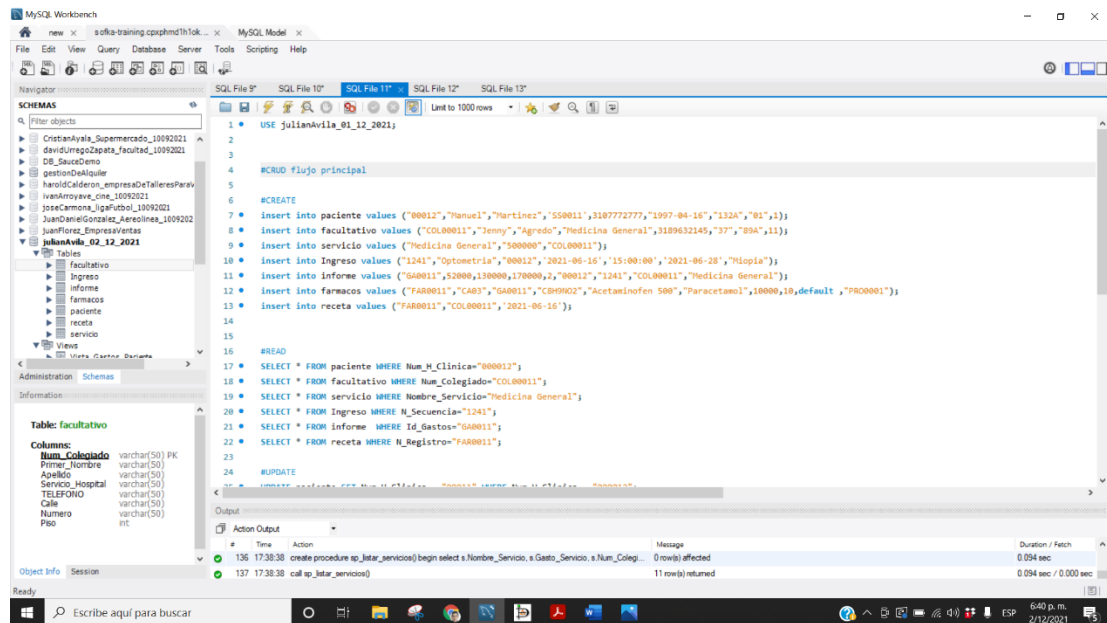
```

    1  #CREACIÓN DB HOSPITAL
    2
    3  CREATE DATABASE julianAvila_02_12_2021 CHARACTER SET utf8 COLLATE utf8_spanish_ci;
    4  USE julianAvila_02_12_2021;
    5  DROP DATABASE julianAvila_02_12_2021;
    6
    7
    8  CREATE TABLE paciente(
    9  Num_H_Clinica VARCHAR(50) PRIMARY KEY,
    10 Primer_Nombre VARCHAR(50) NOT NULL,
    11 Apellido VARCHAR(50) NOT NULL,
    12 Num_Seguridad_Social VARCHAR(50),
    13 Telefono VARCHAR(50) NOT NULL,
    14 Fecha_De_Nacimiento DATE DEFAULT '0000-00-00',
    15 Calle VARCHAR(50) NOT NULL,
    16 Numero VARCHAR(50) NOT NULL,
    17 Piso INT NOT NULL
  
```

- d. Poblar la base de datos creada. Se anexa el archivo SQL donde se ingresan los datos de las tablas.

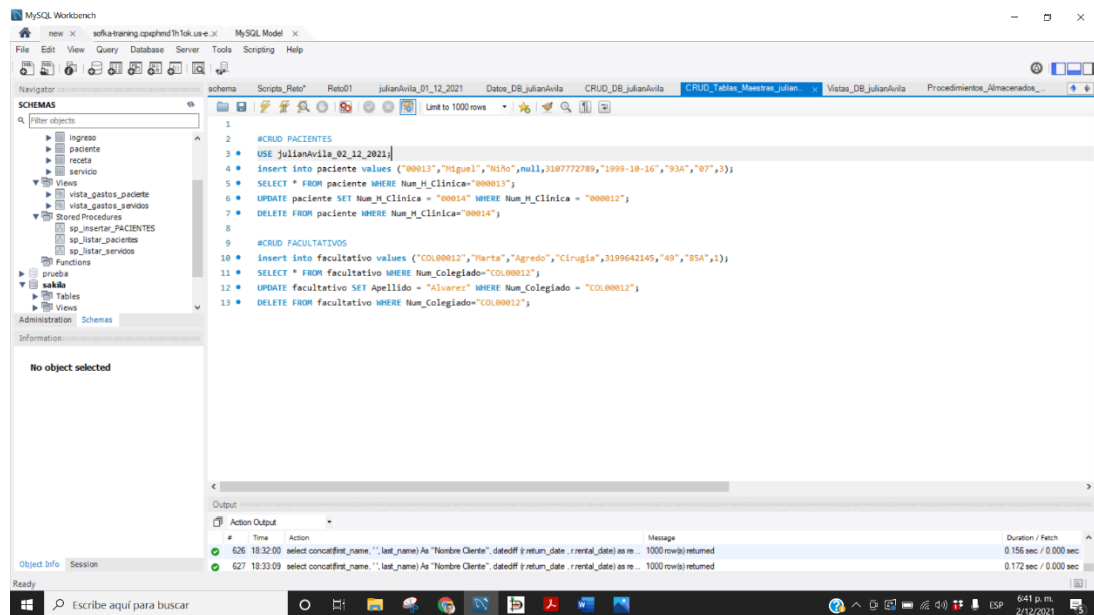


- e. Crear un CRUD para el flujo principal del ingreso de pacientes. Se anexa el archivo SQL donde se crea el CRUD del flujo principal.

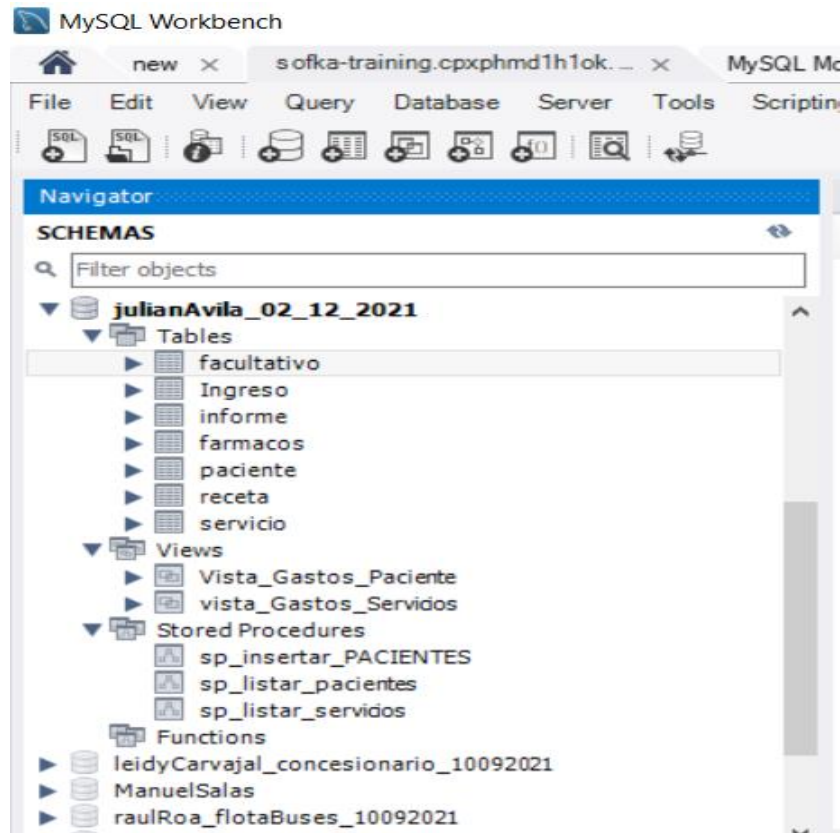


- f. Crear mínimo dos CRUD relacionados con tablas maestras. Se anexa el archivo SQL donde se crea el CRUD relacionado con tablas maestras.

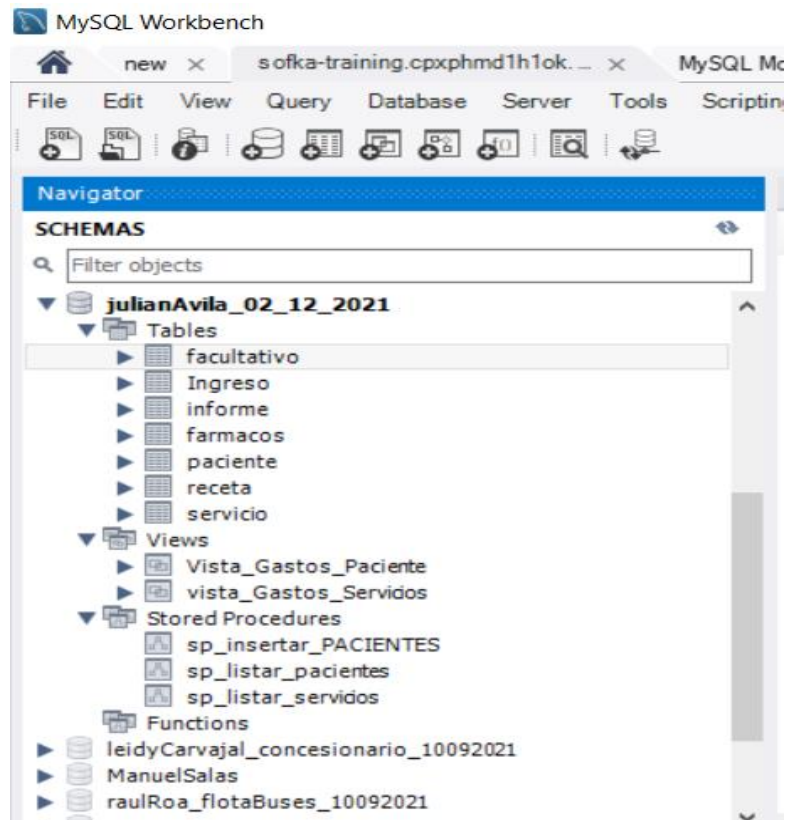




- g. Crear mínimo 3 procedimientos almacenados. Se anexa el archivo SQL donde se los procedimientos.



- h. Crear al menos 2 vistas. Se anexa el archivo SQL donde se crean las vistas.



i. En las operaciones anteriores para el uso de **select** por favor haga uso de **inner joins** donde vea que es necesario. En las vistas anexas se evidencian los **inner joins** creados.