# **PARCIAL FINAL MYSQL**

**Nombre del Taller:**

PARCIAL FINAL DB

**Estudiante:**

ADDEL JOSE PEREZ OSPINO

**Programa Académico:**

Ing. Sistemas 5to Semestre

**Asignatura:**

BASES DE DATOS

Semestre 5

**Docente:**

JOHN CARLOS ARRIETA ARRIETA

**Institución Educativa**

Fundación Universitaria Colombo Internacional UNICOLOMBO

**Fecha:** 28/02/2025

TABLA DE CONTENIDO

[**PARCIAL FINAL MYSQL** 1](#_Toc192270169)

[**Introducción** 2](#_Toc192270170)

[**Objetivos** 2](#_Toc192270171)

[**Justificación** 2](#_Toc192270172)

[**Desarrollo** 2](#_Toc192270173)

[**Identificación de las Interrelaciones entre el Conjunto de Entidades** 2](#_Toc192270174)

[**Diagrama Entidad Relación App Cine(DER):** 2](#_Toc192270175)

[**Pasar el DER(Diagrama Entidad Relacion) a MR (Modelo Relacional)** 2](#_Toc192270176)

[**Modelo Relacional App Cine (MR)** 2](#_Toc192270177)

# **Introducción**

En el contexto de las bases de datos, representar la información de forma adecuada es esencial para el buen funcionamiento de cualquier entidad. En el caso de un cine, la administración de la cartelera, la gestión de funciones y salas, así como la administración de promociones y opiniones de los espectadores, son actividades críticas que demandan un manejo preciso y eficiente de los datos.

Este modelo de base de datos tiene como objetivo facilitar la gestión de la información relacionada con las películas en proyección, sus horarios, salas, promociones y críticas de los espectadores, garantizando así un correcto funcionamiento del cine y una experiencia satisfactoria para los usuarios.

.

# **Objetivos**

**Objetivo Principal**

Diseñar un modelo de base de datos relacional en los diferentes motores de bases de datos propuestos, que sea capaz de manejar la información de las películas en cartelera, funciones, cines, promociones y opiniones de los usuarios de forma precisa y eficiente.

**Objetivos Secundarios**

* Construir un **Diagrama Entidad-Relación (DER)** y transformarlo en el **Modelo Relacional (MR)**, asegurando la integridad y normalización de los datos.
* Documentar adecuadamente la estructura de la base de datos mediante diagramas y descripciones detalladas para facilitar su comprensión y mantenimiento.
* Identificar y definir claramente las **entidades** presentes en el sistema, como películas, funciones, salas, cines, promociones y usuarios, así como sus interrelaciones.
* Garantizar que el modelo de base de datos permita realizar consultas eficientes sobre la cartelera, funciones disponibles, promociones y opiniones de los usuarios.
* Diseñar la base de datos de manera escalable y flexible, permitiendo la incorporación de nuevas funcionalidades en el futuro.

# **Justificación**

La elaboración de un modelo de base de datos para la gestión de un cine es una tarea esencial que responde a necesidades críticas en la administración diaria de la cartelera, funciones y servicios ofrecidos. La razón fundamental para llevar a cabo este trabajo radica en la importancia de contar con un sistema preciso y eficiente que administre la información clave relacionada con las películas en proyección, horarios, salas, promociones y opiniones de los espectadores.

# **Desarrollo**

**Gestión del Cine**

**Inicio del Proceso**

Un cine requiere gestionar la cartelera, las funciones y los servicios ofrecidos a los espectadores. Para ello, se debe registrar información detallada sobre las películas en proyección, los horarios de las funciones, las salas disponibles y las promociones aplicables.

* Registro de Películas en la Cartelera:

Cada película debe ser registrada con la siguiente información:

• ID de película (identificador único de la película en la cartelera).

• Título de distribución.

• Título original.

• Resumen.

• Subtítulo en español.

• URL oficial.

• Duración.

• Año de producción.

• Clasificación por edad.

• País de origen.

• Fecha de estreno en Santiago.

• Género.

• Idioma original.

• Tipo de contenido.

* Registro de Directores:

Cada director debe ser registrado con la siguiente información: • ID de director (identificador único).

• Nombre.

• Cantidad de películas dirigidas.

* Asignación de Directores a Películas:

Solo ciertos directores pueden dirigir una película, y pueden tener diferentes roles. Se debe registrar:

• ID del director.

• ID de la película.

• Rol del director (Subdirector, Asistente, etc.).

* Registro de Actores:

Cada actor debe ser registrado con la siguiente información:

• ID de actor (identificador único).

• Nombre completo.

• Cantidad de películas en las que ha actuado.

* Asignación de Actores a Películas:

Cada actor tiene un rol específico en cada película. Se debe registrar:

• ID de película.

• ID de actor.

• Rol del actor (Principal, Doble, Secundario).

* Registro de Personajes:

Cada personaje debe ser registrado con la siguiente información:

• ID del personaje.

• ID del actor que lo interpreta.

• Nombre del personaje.

• Tipo de personaje (Principal, Cameo, Animado).

* Opiniones de Críticos:

Cada opinión sobre una película debe ser registrada con la siguiente información:

• ID de opinión (identificador único).

• ID de película.

• Comentario.

• Nombre del crítico. • Edad del crítico.

• Calificación.

• Fecha de registro.

**Consultas Clave para el Sistema**

Para garantizar el correcto funcionamiento del sistema, la base de datos debe permitir responder las siguientes consultas:

1. Mostrar las películas en cartelera para el día de hoy.
2. Mostrar las películas en cartelera para la próxima semana.
3. Buscar películas por título.
4. Buscar películas por género.
5. Mostrar las películas más populares en este momento.
6. Mostrar las películas mejor calificadas por los usuarios.
7. Mostrar las películas con promociones especiales.
8. Mostrar las funciones disponibles para una película específica.
9. Mostrar las películas que están próximamente en cartelera.
10. Mostrar las películas que se proyectan en un cine específico.
11. Ver detalles de una película específica, como el elenco, director y resumen.
12. Mostrar las opiniones y calificaciones de los usuarios para una película específica.
13. Mostrar las promociones disponibles para un cine específico.
14. Ver los horarios de las funciones para una película específica.
15. Mostrar las películas que están en 3D o IMAX.
16. Ver los detalles de una función específica, como la sala y la hora de inicio.
17. Filtrar las películas por clasificación por edades.
18. Ver las películas que están subtituladas en español.
19. Mostrar las películas disponibles en un idioma específico.
20. Ver las películas que están en versión original.
21. Ver las películas que tienen descuentos para estudiantes.

**Identificación del conjunto de entidades y sus  
atributos**

 **Pelicula** (ID\_Pelicula, Titulo\_Distribucion, Titulo\_Original, Resumen, Subtitulo\_Español, URL\_Oficial, Duracion, Año\_Produccion, Clasificacion\_Edad, Pais\_Origen, Fecha\_Estreno\_Santiago, Genero, Idioma\_Original, Tipo\_Contenido)

 **Director** (ID\_Director, Nombre, Cantidad\_Peliculas\_Dirigidas)

 **Director\_Pelicula** (ID\_Director, ID\_Pelicula, Rol\_Director)

 **Actor** (ID\_Actor, Nombre, Cantidad\_Peliculas\_Actuadas)

 **Pelicula\_Actor** (ID\_Pelicula, ID\_Actor, Rol\_Actor)

 **Personaje** (ID\_Personaje, ID\_Actor, Nombre, Tipo\_Personaje)

 **Pelicula\_Personaje** (ID\_Pelicula, ID\_Personaje)

 **Opinion** (ID\_Opinion, ID\_Pelicula, Comentario, Nombre\_Critico, Edad\_Critico, Calificacion, Fecha\_Registro)

 **Adaptacion** (ID\_Adaptacion, ID\_Pelicula, Titulo\_Origen, Tipo)

 **Premio** (ID\_Premio, ID\_Pelicula, Año, Categoria, Nombre)

 **Estado** (ID\_Estado, ID\_Pelicula, Tipo\_Estado)

 **Funcion** (ID\_Funcion, ID\_Pelicula, ID\_Sala, Hora\_Inicio, Fecha\_Estreno, Dia\_Semana, Fecha\_Fin, Horario)

 **Sala** (ID\_Sala, ID\_Cine, Nombre, Capacidad\_Asientos)

 **Cine** (ID\_Cine, Nombre, Direccion)

 **Telefono\_Consultas** (ID\_Telefono, ID\_Cine, Tipo)

 **Formato\_Proyeccion** (ID\_Formato, ID\_Funcion, Tipo\_Formato)

 **Venta\_Boletos** (ID\_Venta, ID\_Funcion, Fecha\_Venta, Total\_Venta, Cantidad\_Boletos)

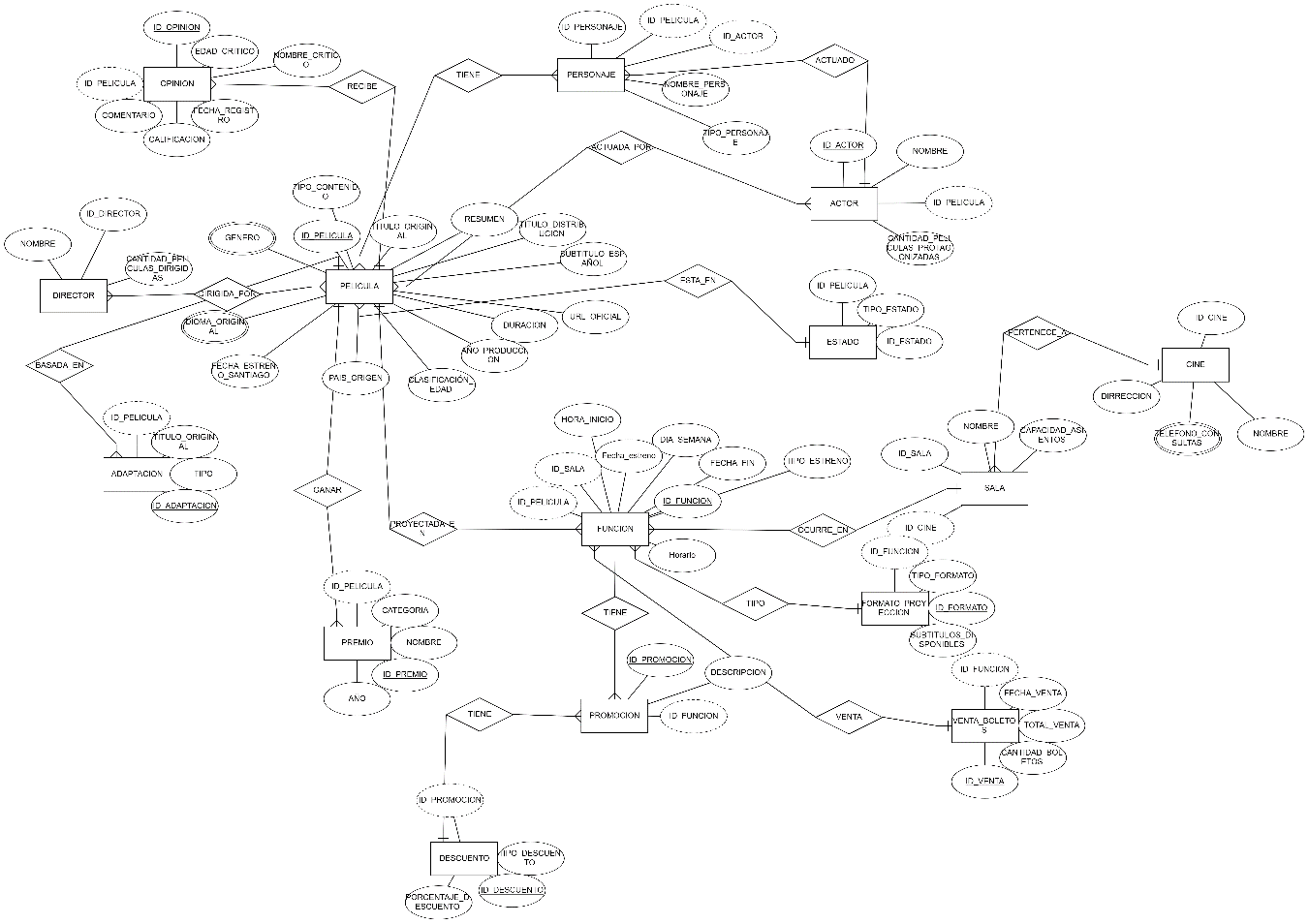
 **Promocion** (ID\_Promocion, ID\_Funcion, Descripcion)

 **Descuento** (ID\_Descuento, ID\_Funcion, Porcentaje\_Descuento, Tipo\_Descuento)

# **Identificación de las Interrelaciones entre el Conjunto de Entidades**

* **Película, Director\_Película:** Una película puede tener varios directores, y un director puede dirigir varias películas (N:M).
* **Director, Director\_Película:** Un director puede estar asociado a varias películas y una película puede tener múltiples directores (N:M).
* **Película, Película\_Actor:** Una película puede contar con múltiples actores, y un actor puede participar en varias películas (N:M).
* **Actor, Personaje:** Un actor puede interpretar varios personajes, pero un personaje solo puede ser interpretado por un actor (1:N).
* **Película, Opinión:** Una película puede tener muchas opiniones, pero cada opinión pertenece a una sola película (1:N).
* **Película, Película\_Personaje:** Una película puede incluir varios personajes, y un personaje puede aparecer en varias películas (N:M).
* **Película, Adaptación:** Una película puede tener varias adaptaciones, pero cada adaptación está asociada a una sola película (1:N).
* **Película, Premio:** Una película puede ganar varios premios, pero cada premio está asociado a una sola película (1:N).
* **Película, Estado:** Una película puede estar en diferentes estados a lo largo del tiempo (1:N).
* **Película, Función:** Una película puede tener múltiples funciones en cines, pero cada función pertenece a una sola película (1:N).
* **Sala, Función:** Una sala puede proyectar varias funciones, pero cada función se lleva a cabo en una sola sala (1:N).
* **Actor, Película\_Actor:** Un actor puede aparecer en varias películas, y una película puede contar con múltiples actores (N:M).
* **Personaje, Película\_Personaje:** Un personaje puede aparecer en varias películas, y una película puede incluir múltiples personajes (N:M).
* **Cine, Sala:** Un cine puede tener varias salas, pero cada sala pertenece a un solo cine (1:N).
* **Cine, Teléfono\_Consultas:** Un cine puede tener varios teléfonos de consulta, pero cada teléfono pertenece a un solo cine (1:N).
* **Formato\_Proyección, Función:** Un formato de proyección puede aplicarse a varias funciones, pero cada función solo tiene un formato de proyección (1:N).
* **Venta\_Boletos, Función:** Una función puede vender múltiples boletos, pero cada venta de boleto pertenece a una sola función (1:N).
* **Función, Promoción:** Una función puede tener varias promociones, pero cada promoción está asociada a una sola función (1:N).

# **Diagrama Entidad Relación App Cine (DER):**



# **Pasar el DER(Diagrama Entidad Relacion) a MR (Modelo Relacional)**

#### **Pelicula**

* **ID\_Pelicula (PK)**
* **Titulo\_Distribucion**
* **Titulo\_Original**
* **Resumen**
* **Subtitulo\_Español**
* **URL\_Oficial**
* **Duracion**
* **Año\_Produccion**
* **Clasificacion\_Edad**
* **Pais\_Origen**
* **Fecha\_Estreno\_Santiago**
* **Genero**
* **Idioma\_Original**
* **Tipo\_Contenido**

#### **Director**

* **ID\_Director (PK)**
* **Nombre**
* **Cantidad\_Peliculas\_Dirigidas**

#### **Director\_Pelicula**

* **ID\_Director (FK → Director.ID\_Director)**
* **ID\_Pelicula (FK → Pelicula.ID\_Pelicula)**
* **Rol\_Director**
* **PK compuesta (ID\_Director, ID\_Pelicula)**

#### **Actor**

* **ID\_Actor (PK)**
* **Nombre**
* **Cantidad\_Peliculas\_Actuadas**

#### **Pelicula\_Actor**

* **ID\_Pelicula (FK → Pelicula.ID\_Pelicula)**
* **ID\_Actor (FK → Actor.ID\_Actor)**
* **Rol\_Actor**
* **PK compuesta (ID\_Pelicula, ID\_Actor)**

#### **Personaje**

* **ID\_Personaje (PK)**
* **ID\_Actor (FK → Actor.ID\_Actor)**
* **Nombre**
* **Tipo\_Personaje**

#### **Pelicula\_Personaje**

* **ID\_Pelicula (FK → Pelicula.ID\_Pelicula)**
* **ID\_Personaje (FK → Personaje.ID\_Personaje)**
* **PK compuesta (ID\_Pelicula, ID\_Personaje)**

#### **Opinion**

* **ID\_Opinion (PK)**
* **ID\_Pelicula (FK → Pelicula.ID\_Pelicula)**
* **Comentario**
* **Nombre\_Critico**
* **Edad\_Critico**
* **Calificacion**
* **Fecha\_Registro**

#### **Adaptacion**

* **ID\_Adaptacion (PK)**
* **ID\_Pelicula (FK → Pelicula.ID\_Pelicula)**
* **Titulo\_Origen**
* **Tipo**

#### **Premio**

* **ID\_Premio (PK)**
* **ID\_Pelicula (FK → Pelicula.ID\_Pelicula)**
* **Año**
* **Categoria**
* **Nombre**

#### **Estado**

* **ID\_Estado (PK)**
* **ID\_Pelicula (FK → Pelicula.ID\_Pelicula)**
* **Tipo\_Estado**

#### **Funcion**

* **ID\_Funcion (PK)**
* **ID\_Pelicula (FK → Pelicula.ID\_Pelicula)**
* **ID\_Sala (FK → Sala.ID\_Sala)**
* **Hora\_Inicio**
* **Fecha\_Estreno**
* **Dia\_Semana**
* **Fecha\_Fin**
* **Horario**

#### **Sala**

* **ID\_Sala (PK)**
* **ID\_Cine (FK → Cine.ID\_Cine)**
* **Nombre**
* **Capacidad\_Asientos**

#### **Cine**

* **ID\_Cine (PK)**
* **Nombre**
* **Direccion**

#### **Telefono\_Consultas**

* **ID\_Telefono (PK)**
* **ID\_Cine (FK → Cine.ID\_Cine)**
* **Tipo**

#### **Formato\_Proyeccion**

* **ID\_Formato (PK)**
* **ID\_Funcion (FK → Funcion.ID\_Funcion)**
* **Tipo\_Formato**

#### **Venta\_Boletos**

* **ID\_Venta (PK)**
* **ID\_Funcion (FK → Funcion.ID\_Funcion)**
* **Fecha\_Venta**
* **Total\_Venta**
* **Cantidad\_Boletos**

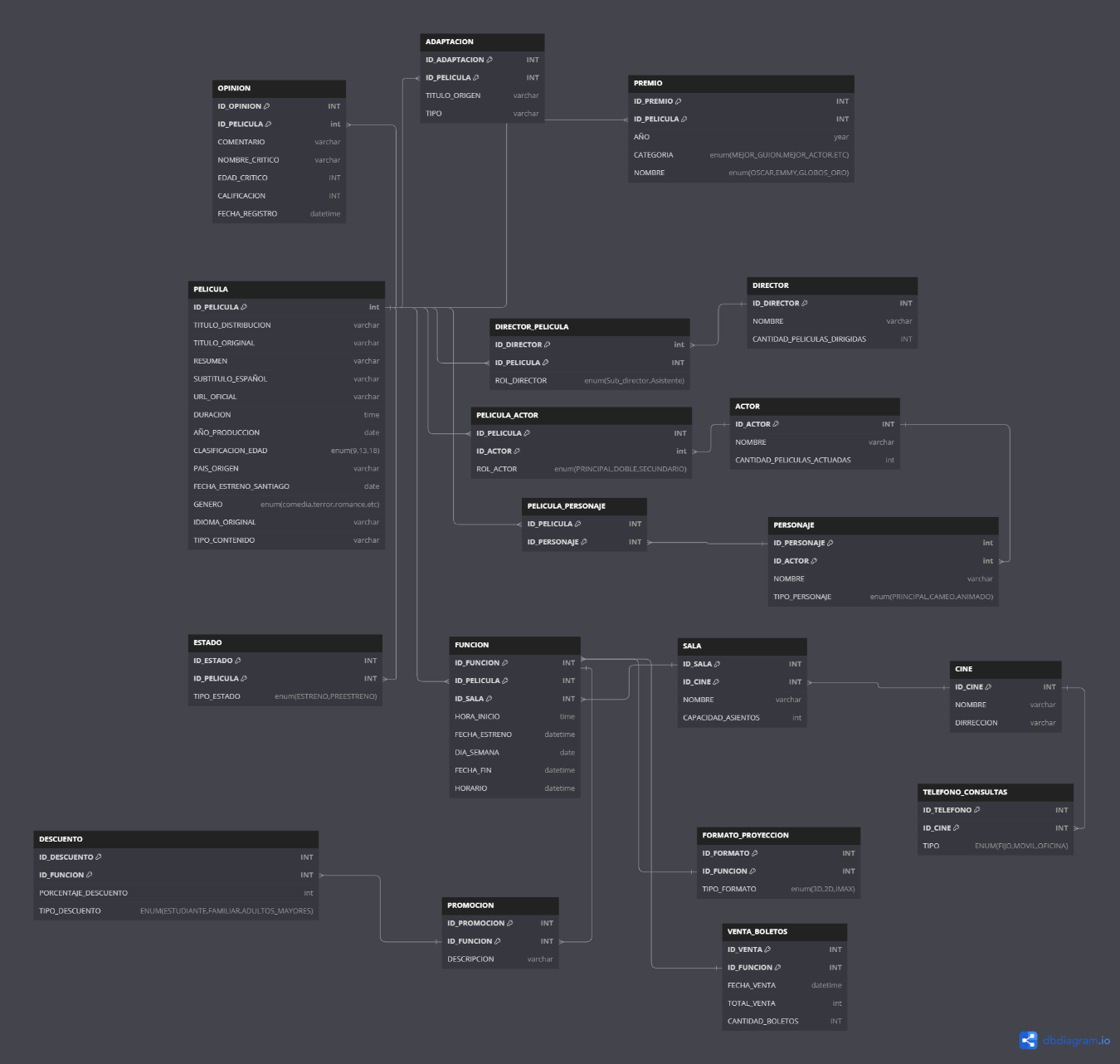
#### **Promocion**

* **ID\_Promocion (PK)**
* **ID\_Funcion (FK → Funcion.ID\_Funcion)**
* **Descripcion**

#### **Descuento**

* **ID\_Descuento (PK)**
* **ID\_Funcion (FK → Funcion.ID\_Funcion)**
* **Porcentaje\_Descuento**
* **Tipo\_Descuento**

# **Modelo Relacional Comedor Infantil (MR)**



1. **Entrar al Motor de Bases** de datos usando la **Terminal (consola)** de Línea de Comandos usando el usuario creado durante la instrucción o el usuario por defecto con rol de super usuario.

INSTRUCCIÓN SQL:

* Sqlcmd -S .\SQLEXPRESS -U SA -P 2006

RECORTE DE PANTALLA



EXPLICACIÓN.

Se ingresa al motor de bases de datos mysql

1. **Crear una base de datos** (tablespace en oracle) llamada **bd\_tu\_nombre\_tu\_apellido\_xyz**, donde debe reemplazarse numero\_nombre\_del\_archivo del ejercicio que le fue asignado.

INSTRUCCIÓN SQL:

* CREATE DATABASE bd\_addel\_perez\_xyz;
* GO

RECORTE DE PANTALLA



EXPLICACIÓN.

Se crea una base de datos llamada bd\_addel\_perez\_xyz

1. **Crear** un nuevo **usuario** para poder entrar al Motor de BD, los datos del nuevo usuario deben ser:

Nombre: **user\_tu\_nombre\_tu\_apellido**

Clave: **AbcdeUdeC**

El nuevo usuario podrá conectarse desde cualquier dirección IP

INSTRUCCIÓN SQL:

* CREATE LOGIN user\_addel\_perez WITH PASSWORD = ‘AbcdeUdeC’;
* GO

RECORTE DE PANTALLA



EXPLICACIÓN.

Se crea un nuevo usuario llamado user\_addeL\_perez

1. Al nuevo usuario **user\_tu\_nombre\_tu\_apellido** creado anteriormente , se le deben **asignar Privilegios de Superusuario** sobre la base de datos **bd\_tu\_nombre\_tu\_apellido\_xyz**

INSTRUCCIÓN SQL:

* ALTER ROLE db\_owner ADD MEMBER user\_addel\_perez;
* GO

RECORTE DE PANTALLA



EXPLICACIÓN.

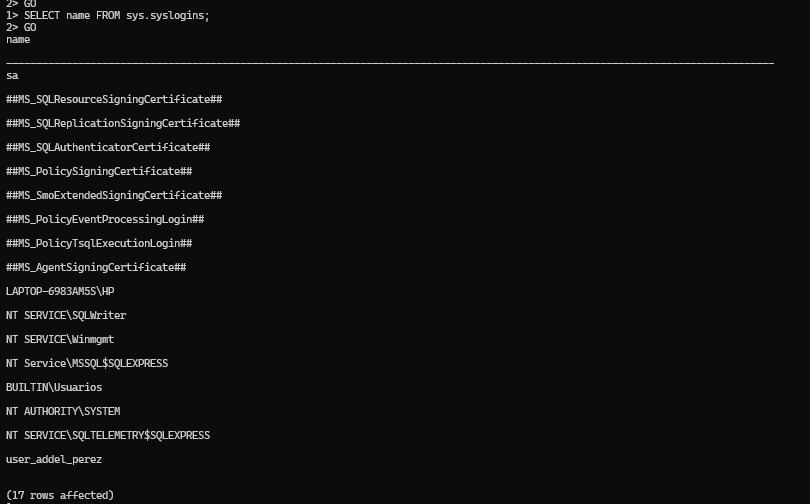
Se le da privilegios de superusuario a user\_addel\_perez sobre la base de datos bd\_addel\_perez

1. **Mostrar los usuarios** existentes

INSTRUCCIÓN SQL:

* SELECT name FROM sys.syslogins;
* GO

RECORTE DE PANTALLA



EXPLICACIÓN

Muestra los usuarios existentes

1. **Salir de la terminal** o consola del Motor de BD

INSTRUCCIÓN SQL:

EXIT

RECORTE DE PANTALLA



EXPLICACIÓN

Salir de la consola del motor de BD

1. Volver a **entrar** la Motor de Bases de datos **desde la terminal o consola** pero ahora usando las c**redenciales del nuevo usuario** **user\_tu\_nombre\_tu\_apellido** creado anteriormente-

INSTRUCCIÓN SQL:

* Sqlcmd – S localhost -U sa -P “2006” -d bd\_addel\_perez\_xyz

RECORTE DE PANTALLA



EXPLICACIÓN

Se ingresa con el usuario user\_addel\_perez

Las siguientes operaciones SQL deben ser realizadas utilizando el nuevo usuario **user\_tu\_nombre\_tu\_apellido**.

1. **Crear** una **nueva BD** llamada **bd\_tmp\_tu\_nombre\_tu\_apellido**

INSTRUCCIÓN SQL:

* CREATE DATABASE bd\_tmp\_addel\_perez;
* GO

RECORTE DE PANTALLA



EXPLICACIÓN

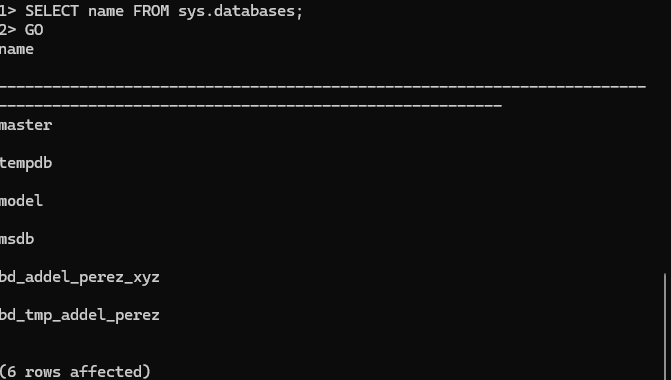
Se crea una nueva base de datos llamada bd\_tmp\_addel\_perez

1. **Mostrar** las **Bases de datos** que pertenecen al **user\_tu\_nombre\_tu\_apellido**

INSTRUCCIÓN SQL:

* SELECT name FROM sys.databases;
* GO

RECORTE DE PANTALLA



EXPLICACIÓN

Se muestran las bases de datos pertenecientes a user\_addel\_perez

1. Entrar o usar la BD **bd\_tmp\_tu\_nombre\_tu\_apellido**

INSTRUCCIÓN SQL:

* USE bd\_tmp\_addel\_perez
* GO

RECORTE DE PANTALLA



EXPLICACIÓN

Se entra/usa la base de datos bd\_tmp\_addel\_perez

1. **Crear una tabla** llamada **bd\_tmp\_tu\_nombre\_tu\_apellido**con las siguiente estructura

**tabla\_tempora**

**id** auto\_incremental y clave primera

**columna2** de tipo Texto, no nula y con longitud de 100

**columna3** de tipo Entero

**columna4** de tipo Real

**columna5** de tipo Fecha

**columna6** de tipo FechaHora

**columna7** de tipo hora

**columna8** de tipo FechaHora automatica

**columna9** de tipo Texto con longitud variable

**columna10** de tipo archivo binario

INSTRUCCIÓN SQL:

CREATE TABLE tabla\_temporal (

id INT IDENTITY(1,1) PRIMARY KEY,

columna2 NVARCHAR(100) NOT NULL,

columna3 INT,

columna4 DECIMAL(10,2),

columna5 DATE,

columna6 DATETIME,

columna7 TIME,

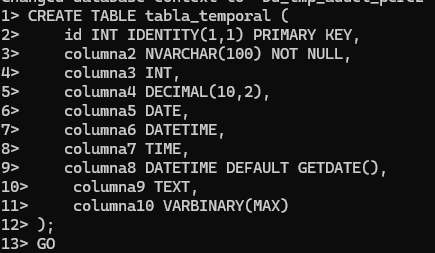
columna8 DATETIME DEFAULT GETDATE(),

columna9 TEXT,

columna10 VARBINARY(MAX)

);

RECORTE DE PANTALLA



EXPLICACIÓN

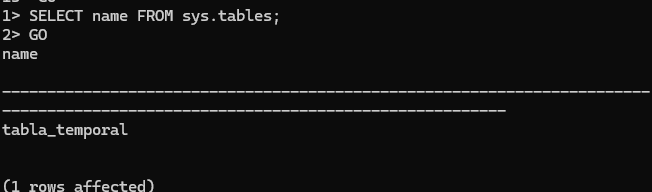
Se crea una nueva tabla con las especificaciones anteriormente dadas

1. **Mostrar las tablas** de la **bd\_tmp\_tu\_nombre\_tu\_apellido**

INSTRUCCIÓN SQL:

* SELECT name FROM sys.tables;
* GO

RECORTE DE PANTALLA



EXPLICACIÓN

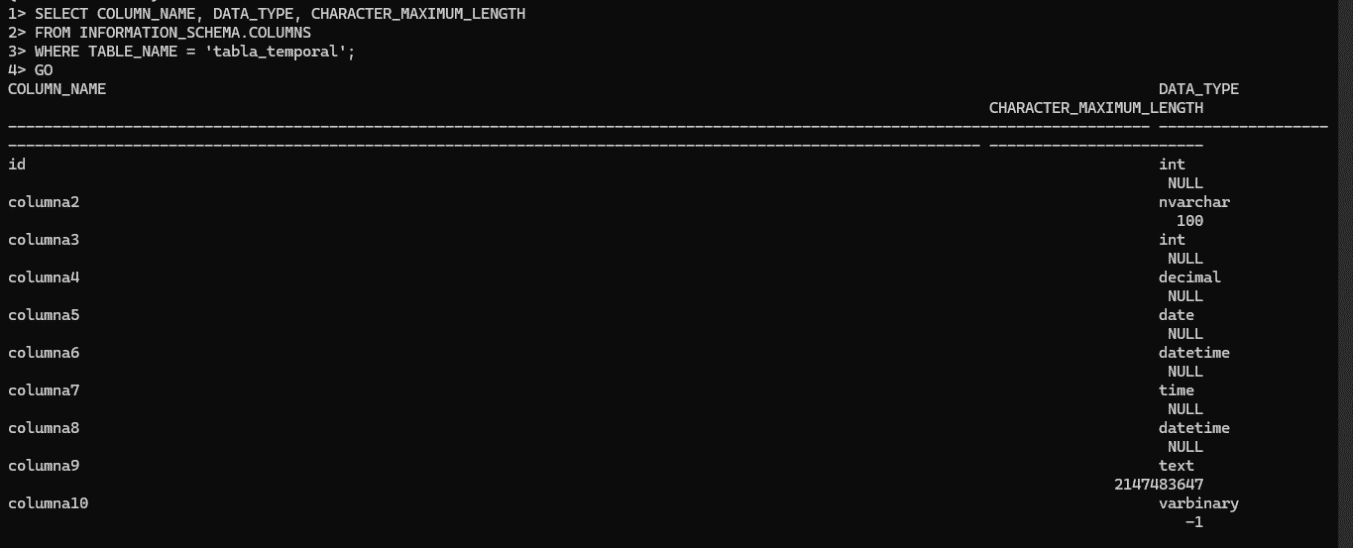
Se muestran las tablas de bd\_tmp\_addel\_perez

1. **Describir** la estructura de la **tabla\_tempora**

INSTRUCCIÓN SQL:

* SELECT COLUMN\_NAME, DATA\_TYPE, CHARACTER\_MAXIMUM\_LENGTH
* FROM INFORMATION\_SCHEMA.COLUMNS
* WHERE TABLE\_NAME = ‘tabla\_temporal’;
* GO

RECORTE DE PANTALLA



EXPLICACIÓN

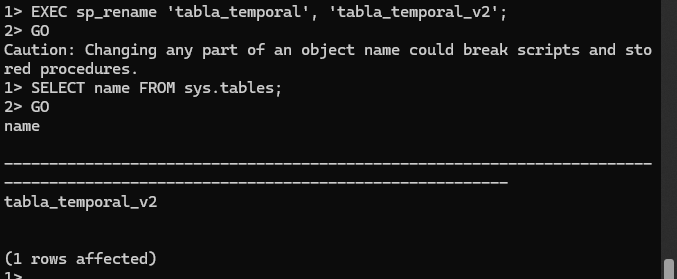
Se describe la tabla tabla\_temporal

1. **Cambiar** el **nombre** de la **tabla\_tempora** por **tabla\_tempora\_v2**

INSTRUCCIÓN SQL:

* EXEC sp\_rename ‘tabla\_temporal’, ‘tabla\_temporal\_v2’;
* GO

RECORTE DE PANTALLA



EXPLICACIÓN

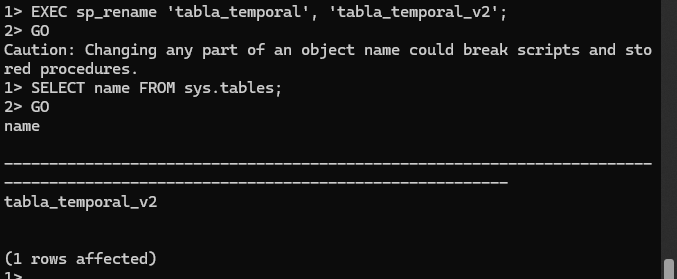
Se cambia el nombre de la tabla tabla\_temporal por tabla\_temporal\_v2

1. **Mostrar** las **tablas** de la **bd\_temporal**

INSTRUCCIÓN SQL:

* SELECT name FROM sys.tables;
* GO

RECORTE DE PANTALLA



EXPLICACIÓN

Se muestran las tablas en la base de tatos bd\_tmp\_addel\_perez

1. **Describir** la Estructura de la **tabla\_tempora\_v2** para observar los cambios

INSTRUCCIÓN SQL:

* SELECT c.name AS Nombre\_Columna, t.name AS Tipo\_Dato, c.max\_length AS Tamaño
* FROM sys.columns c
* JOIN sys.types t ON c.user\_type\_id = t.user\_type\_id
* WHERE c.object\_id = OBJECT\_ID(‘tabla\_temporal\_v2’);
* GO

RECORTE DE PANTALLA



EXPLICACIÓN

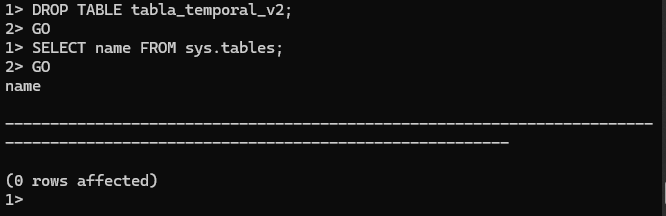
Se describe la tabla tabla\_temporla\_v2 para observar los cambios realizados en esta

1. **Eliminar** la **tabla\_tempora\_v2**

INSTRUCCIÓN SQL:

* DROP TABLE tabla\_temporal\_v2;
* GO

RECORTE DE PANTALLA



EXPLICACIÓN

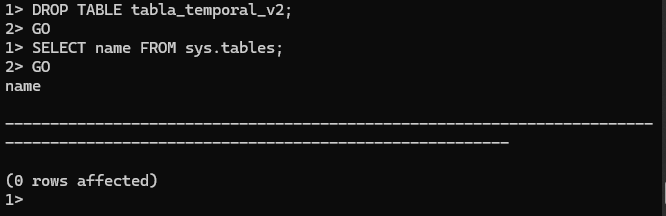
Se elimina la tabla tabla\_temporal\_v2

1. **Mostrar** las **tablas** de la **bd\_tmp\_tu\_nombre\_tu\_apellido**

INSTRUCCIÓN SQL:

* SELECT name FROM sys.tables;

RECORTE DE PANTALLA



EXPLICACIÓN

Se muestran las tablas

1. **Cambiarle** el **nombre** de la **base de datos** **bd\_tmp\_tu\_nombre\_tu\_apellido** por **bd\_tmp\_tu\_nombre\_tu\_apellido\_v2**

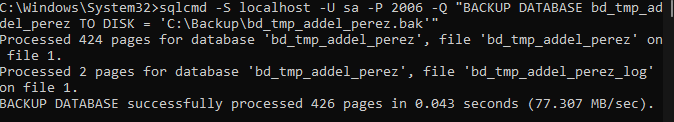
Según el motor de BD utilizado esta operación no se realiza por CLI, si ese es su caso, entonces, debe explicar el proceso necesario para cambiar el nombre a la BD.

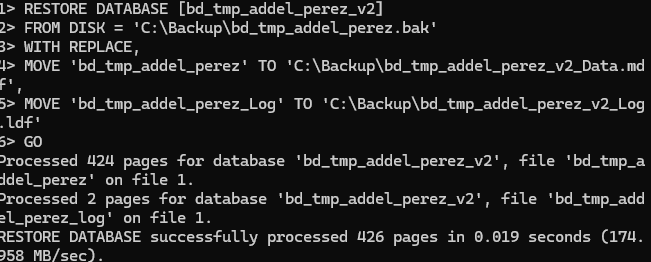
INSTRUCCIÓN SQL:

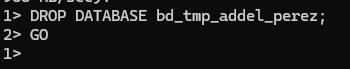
* CREATE DATABASE bd\_tmp\_addel\_perez\_v2;
* GO
* Sqlcmd -S localhost -U sa -P 2006 -Q “BACKUP DATABSE bd\_tmp\_addel\_perez TO DISK = ‘C:\Backup\bd\_addel\_perez.bak’”
* RESTORE DATABASE [bd\_tmp\_addel\_perez\_v2]
* FROM DISK = ‘C:\Backup\bd\_tmp\_addel\_perez.bak’
* WITH REPLACE,
* MOVE ‘bd\_tmp\_addel\_perez’ TO ‘C:\Backup\bd\_tmp\_addel\_perez\_v2\_Data.mdf’,
* MOVE ‘bd\_tmp\_addel\_perez\_Log’ TO ‘C:\Backup\bd\_tmp\_addel\_perez\_v2\_Log.ldf’
* GO

RECORTE DE PANTALLA









EXPLICACIÓN

SQL SERVER no permite directamente el cambio de nombre a una base de datos por ende se creo una nueva base de datos bd\_tmp\_addel\_perez\_v2 y se hizo un respaldo de bd\_tmp\_addel\_perez para luego usar ese respaldo en bd\_tmp\_addel\_perez\_v2 y finalmente eliminar bd\_tmp\_addel\_perez. (bd\_tmp\_addel\_perez no tenia ninguna tabla en su interior debido a que se borro en puntos anteriores)

1. Salid de la CLI de la terminal del motor de BD

INSTRUCCIÓN SQL:

* exit

RECORTE DE PANTALLA



EXPLICACIÓN

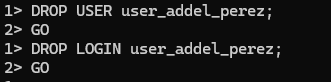
Salir de la terminal del motor de BD

1. Eliminar el usuario **user\_tmp\_tu\_nombre\_tu\_apellido**

INSTRUCCIÓN SQL:

* DROP USER user\_addel\_perez;
* GO
* DROP LOGIN user\_addel\_perez;
* GO

RECORTE DE PANTALLA



EXPLICACIÓN

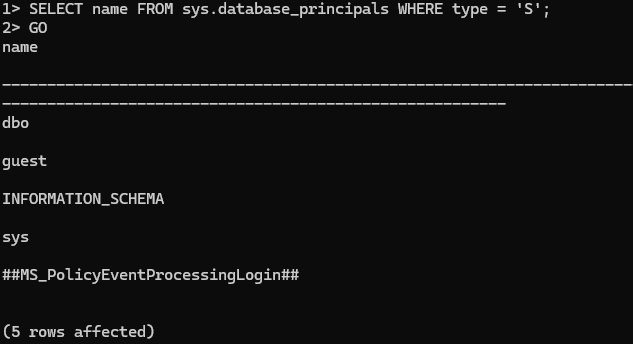
Se elimina al usuario user\_addel\_perez(el ejercicio dice que el usuario  
eliminar debe ser “**user\_tmp\_tu\_nombre\_tu\_apellido”** pero en ningún lugar antes de este punto ese usuario fue mandado a crear por ende se asumió que el usuario a eliminar seria “**user\_tu\_nombre\_tu\_apellido”**)

1. Mostrar los usuarios activos en el motor de bases de datos.

INSTRUCCIÓN SQL:

* SELECT name FROM sys.database\_principals WHERE type = ‘S’;
* GO

RECORTE DE PANTALLA



EXPLICACIÓN

Se muestran los usuarios activos del motor de DB

Las siguientes operaciones SQL se deben realizar tomando como base **Modelo Relacional** del ejercicio que le fue asignado:

1. **Entrar** o usar la base de datos **bd\_tu\_nombre\_tu\_apellido\_xyz** creada anteriormente.

INSTRUCCIÓN SQL:

* USE bd\_addel\_perez\_xyz;

RECORTE DE PANTALLA



EXPLICACIÓN

Se ingresa a la base de datos bd\_addel\_perez\_xyz

1. **Mostrar** las **tablas** de la **bd\_tu\_nombre\_tu\_apellido\_xyz**

INSTRUCCIÓN SQL:

* SHOW TABLES;

RECORTE DE PANTALLA



EXPLICACIÓN

Se muestran las tablas de la base de datos bd\_addel\_perez\_xyz

1. **Crear** las tablasde acuerdo a lo reflejado en el Modelo Relacional del ejercicio.

INSTRUCCIONES SQL PARA CREAR LAS TABLAS:

 **CREATE TABLE OPINION** (ID\_OPINION INT PRIMARY KEY AUTO\_INCREMENT, ID\_PELICULA INT, COMENTARIO VARCHAR(255), NOMBRE\_CRITICO VARCHAR(100), EDAD\_CRITICO INT, CALIFICACION INT, FECHA\_REGISTRO DATETIME, FOREIGN KEY (ID\_PELICULA) REFERENCES PELICULA(ID\_PELICULA) ON DELETE CASCADE).

 **CREATE TABLE PELICULA** (ID\_PELICULA INT PRIMARY KEY AUTO\_INCREMENT, TITULO\_DISTRIBUCION VARCHAR(255), TITULO\_ORIGINAL VARCHAR(255), RESUMEN TEXT, SUBTITULO\_ESPANOL VARCHAR(255), URL\_OFICIAL VARCHAR(255), DURACION TIME, ANIO\_PRODUCCION DATE, CLASIFICACION\_EDAD ENUM('G', 'PG', 'PG-13', 'R', 'NC-17'), PAIS\_ORIGEN VARCHAR(100), FECHA\_ESTRENO\_SANTIAGO DATE, GENERO ENUM('comedia', 'drama', 'accion', 'terror'), IDIOMA\_ORIGINAL VARCHAR(100), TIPO\_CONTENIDO VARCHAR(255)).

 **CREATE TABLE ESTADO** (ID\_ESTADO INT PRIMARY KEY AUTO\_INCREMENT, ID\_PELICULA INT, TIPO\_ESTADO ENUM('ESTRENO', 'PREESTRENO'), FOREIGN KEY (ID\_PELICULA) REFERENCES PELICULA(ID\_PELICULA) ON DELETE CASCADE).

 **CREATE TABLE DESCUENTO** (ID\_DESCUENTO INT PRIMARY KEY AUTO\_INCREMENT, PORCENTAJE\_DESCUENTO INT, TIPO\_DESCUENTO ENUM('ESTUDIANTE', 'FAMILIAR', 'ADULTOS', 'MAYORES')).

 **CREATE TABLE ADAPTACION** (ID\_ADAPTACION INT PRIMARY KEY AUTO\_INCREMENT, ID\_PELICULA INT, TITULO\_ORIGEN VARCHAR(255), TIPO VARCHAR(100), FOREIGN KEY (ID\_PELICULA) REFERENCES PELICULA(ID\_PELICULA) ON DELETE CASCADE).

 **CREATE TABLE PREMIO** (ID\_PREMIO INT PRIMARY KEY AUTO\_INCREMENT, ID\_PELICULA INT, ANIO INT, CATEGORIA ENUM('OSCAR', 'GOLDEN\_GLOBE', 'ACTOR'), NOMBRE VARCHAR(255), FOREIGN KEY (ID\_PELICULA) REFERENCES PELICULA(ID\_PELICULA) ON DELETE CASCADE).

 **CREATE TABLE DIRECTOR** (ID\_DIRECTOR INT PRIMARY KEY AUTO\_INCREMENT, NOMBRE VARCHAR(255), CANTIDAD\_PELICULAS\_DIRIGIDAS INT).

 **CREATE TABLE DIRECTOR\_PELICULA** (ID\_DIRECTOR INT, ID\_PELICULA INT, ROL\_DIRECTOR ENUM('PRINCIPAL', 'ASISTENTE'), PRIMARY KEY (ID\_DIRECTOR, ID\_PELICULA), FOREIGN KEY (ID\_DIRECTOR) REFERENCES DIRECTOR(ID\_DIRECTOR) ON DELETE CASCADE, FOREIGN KEY (ID\_PELICULA) REFERENCES PELICULA(ID\_PELICULA) ON DELETE CASCADE).

 **CREATE TABLE ACTOR** (ID\_ACTOR INT PRIMARY KEY AUTO\_INCREMENT, NOMBRE VARCHAR(255), CANTIDAD\_PELICULAS\_ACTUADAS INT).

 **CREATE TABLE PELICULA\_ACTOR** (ID\_PELICULA INT, ID\_ACTOR INT, ROL\_ACTOR ENUM('PRINCIPAL', 'DOBLE', 'SECUNDARIO'), PRIMARY KEY (ID\_PELICULA, ID\_ACTOR), FOREIGN KEY (ID\_PELICULA) REFERENCES PELICULA(ID\_PELICULA) ON DELETE CASCADE, FOREIGN KEY (ID\_ACTOR) REFERENCES ACTOR(ID\_ACTOR) ON DELETE CASCADE).

 **CREATE TABLE PERSONAJE** (ID\_PERSONAJE INT PRIMARY KEY AUTO\_INCREMENT, NOMBRE VARCHAR(255), TIPO\_PERSONAJE ENUM('PRINCIPAL', 'CAMEO', 'ANIMADO')).

 **CREATE TABLE PELICULA\_PERSONAJE** (ID\_PELICULA INT, ID\_PERSONAJE INT, PRIMARY KEY (ID\_PELICULA, ID\_PERSONAJE), FOREIGN KEY (ID\_PELICULA) REFERENCES PELICULA(ID\_PELICULA) ON DELETE CASCADE, FOREIGN KEY (ID\_PERSONAJE) REFERENCES PERSONAJE(ID\_PERSONAJE) ON DELETE CASCADE).

 **CREATE TABLE CINE** (ID\_CINE INT PRIMARY KEY AUTO\_INCREMENT, NOMBRE VARCHAR(255), DIRECCION VARCHAR(255)).

 **CREATE TABLE SALA** (ID\_SALA INT PRIMARY KEY AUTO\_INCREMENT, ID\_CINE INT, NOMBRE VARCHAR(100), CAPACIDAD\_ASIENTOS INT, FOREIGN KEY (ID\_CINE) REFERENCES CINE(ID\_CINE) ON DELETE CASCADE).

 **CREATE TABLE FUNCION** (ID\_FUNCION INT PRIMARY KEY AUTO\_INCREMENT, ID\_PELICULA INT, ID\_SALA INT, HORA\_INICIO TIME, FECHA\_ESTRENO DATETIME, FECHA\_SALA DATETIME, FECHA\_FIN DATETIME, HORARIO DATETIME, FOREIGN KEY (ID\_PELICULA) REFERENCES PELICULA(ID\_PELICULA) ON DELETE CASCADE, FOREIGN KEY (ID\_SALA) REFERENCES SALA(ID\_SALA) ON DELETE CASCADE).

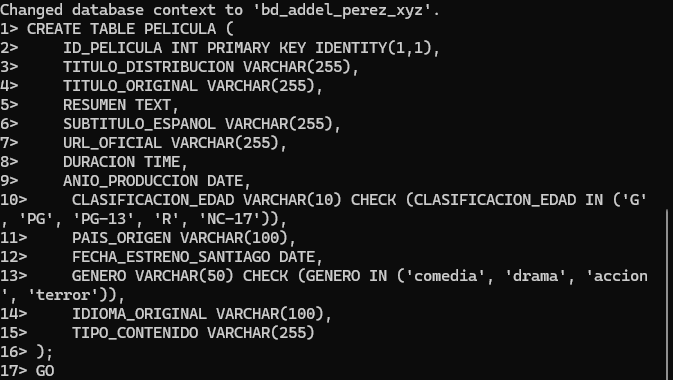
 **CREATE TABLE PROMOCION** (ID\_PROMOCION INT PRIMARY KEY AUTO\_INCREMENT, ID\_FUNCION INT, DESCRIPCION VARCHAR(255), FOREIGN KEY (ID\_FUNCION) REFERENCES FUNCION(ID\_FUNCION) ON DELETE CASCADE).

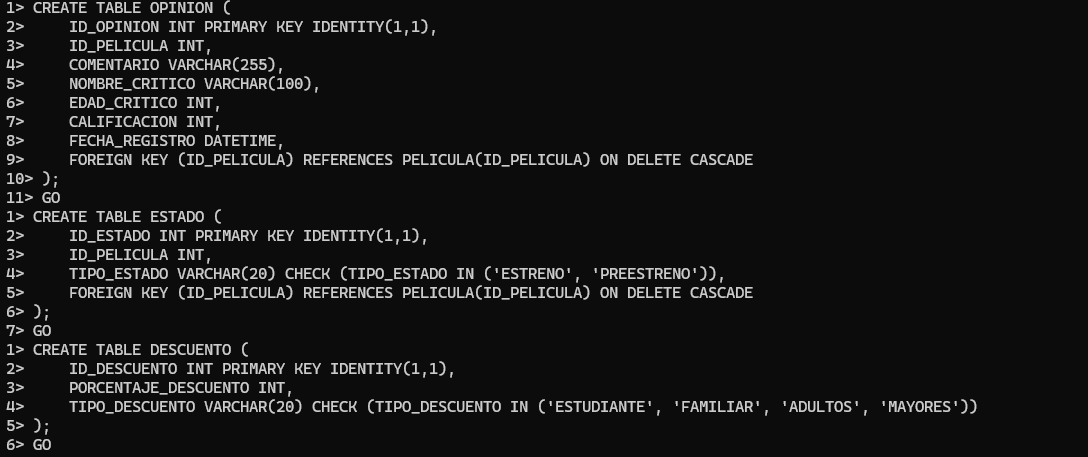
 **CREATE TABLE FORMATO\_PROYECCION** (ID\_FORMATO INT PRIMARY KEY AUTO\_INCREMENT, ID\_FUNCION INT, TIPO\_FORMATO ENUM('2D', '3D', 'IMAX'), FOREIGN KEY (ID\_FUNCION) REFERENCES FUNCION(ID\_FUNCION) ON DELETE CASCADE).

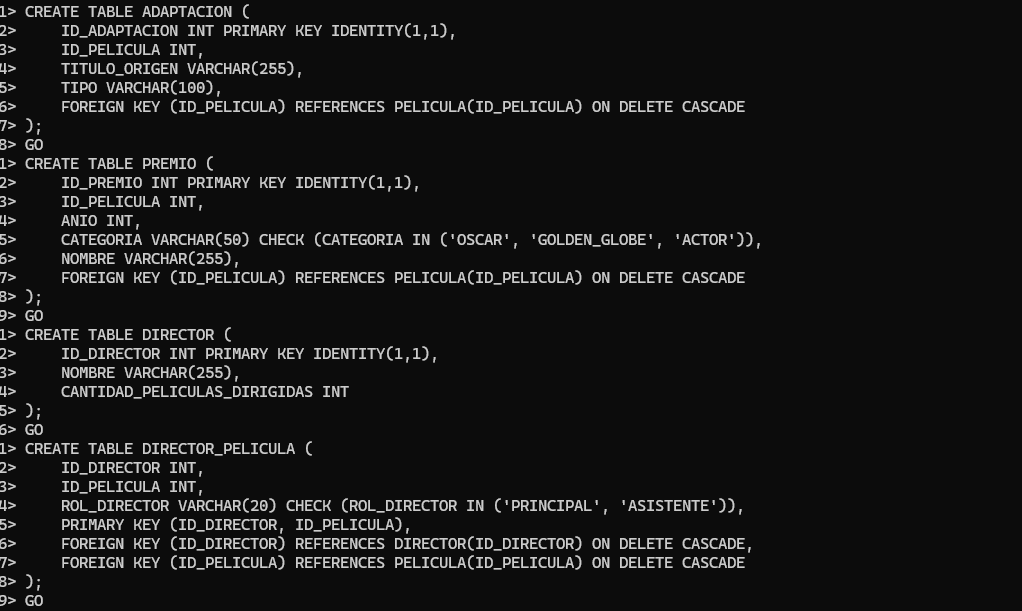
 **CREATE TABLE VENTA\_BOLETOS** (ID\_VENTA INT PRIMARY KEY AUTO\_INCREMENT, ID\_FUNCION INT, FECHA\_VENTA DATETIME, TOTAL\_VENTA INT, TOTAL\_BOLETOS INT, FOREIGN KEY (ID\_FUNCION) REFERENCES FUNCION(ID\_FUNCION) ON DELETE CASCADE).

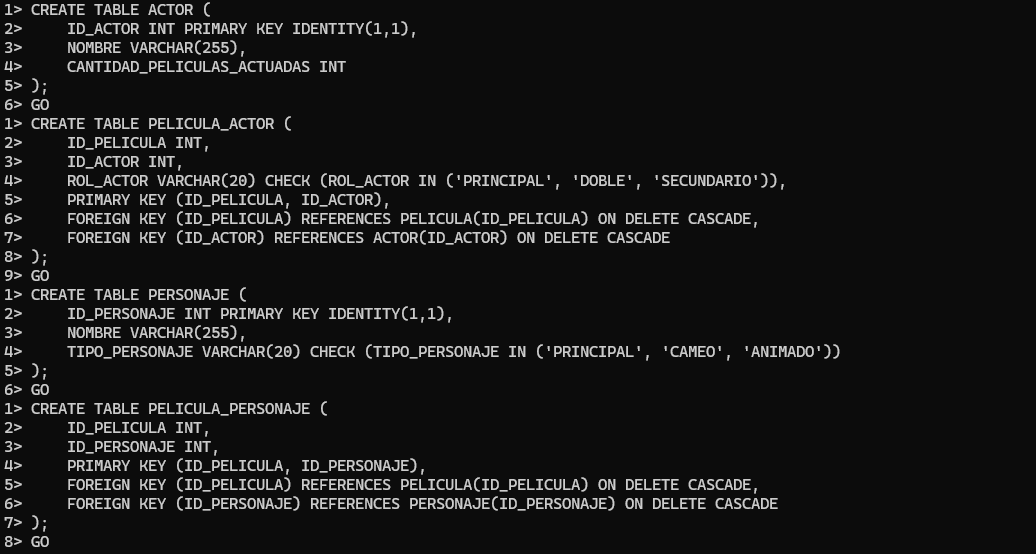
 **CREATE TABLE TELEFONO\_CONSULTAS** (ID\_TELEFONO INT PRIMARY KEY AUTO\_INCREMENT, ID\_CINE INT, TIPO ENUM('ENLACE\_WEB', 'MOVIL', 'FIJO'), FOREIGN KEY (ID\_CINE) REFERENCES CINE(ID\_CINE) ON DELETE CASCADE).

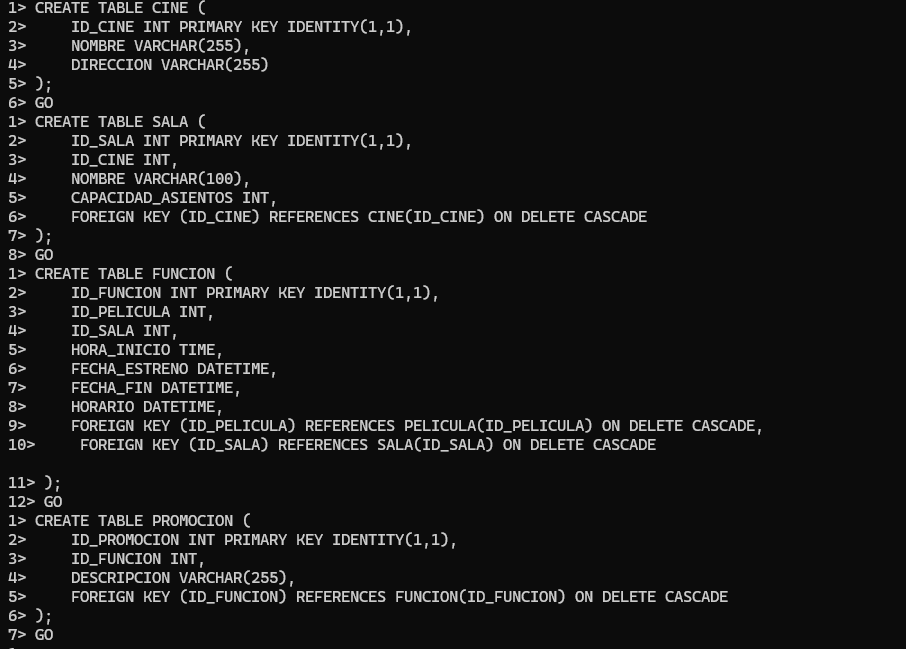
RECORTES DE PANTALLAS

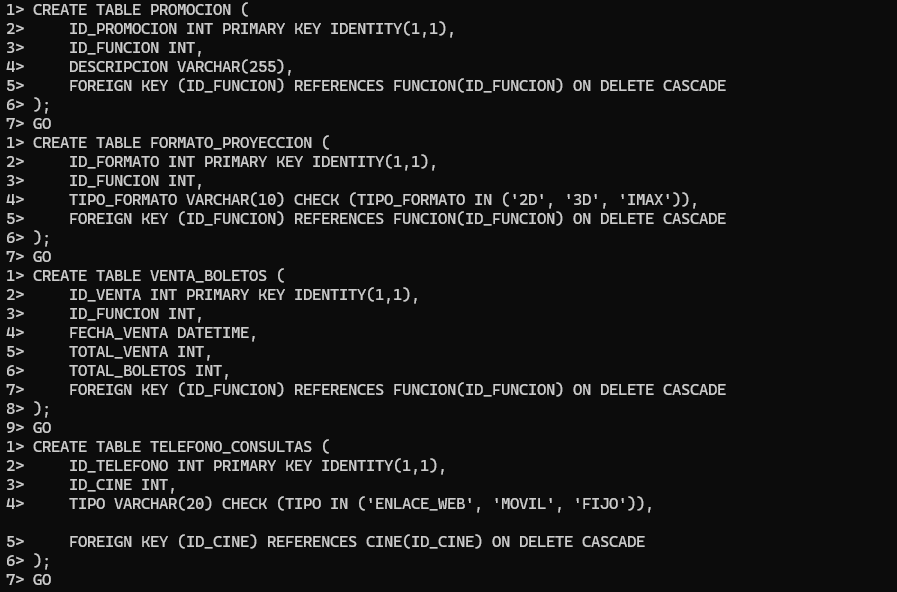
















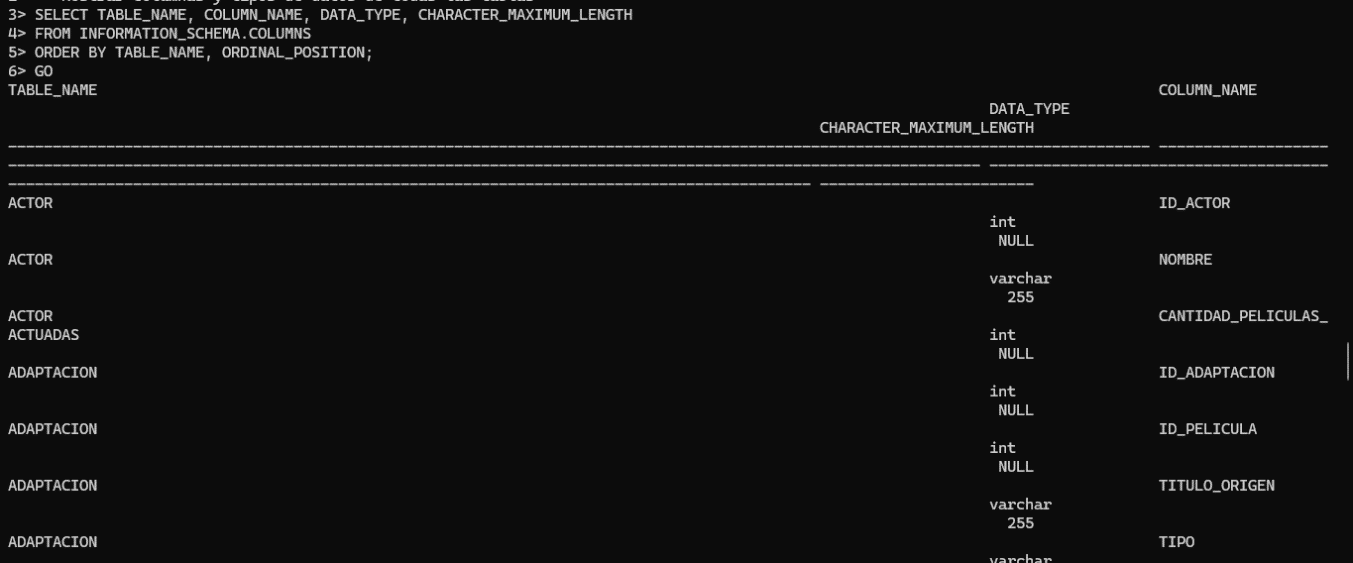
EXPLICACIÓN

Se crearon todas las tablas descritas en el modelo relacional .

**Describir** cada una de las **tablas** creadas creadas en la BD **bd\_tu\_nombre\_tu\_apellido\_xyz**

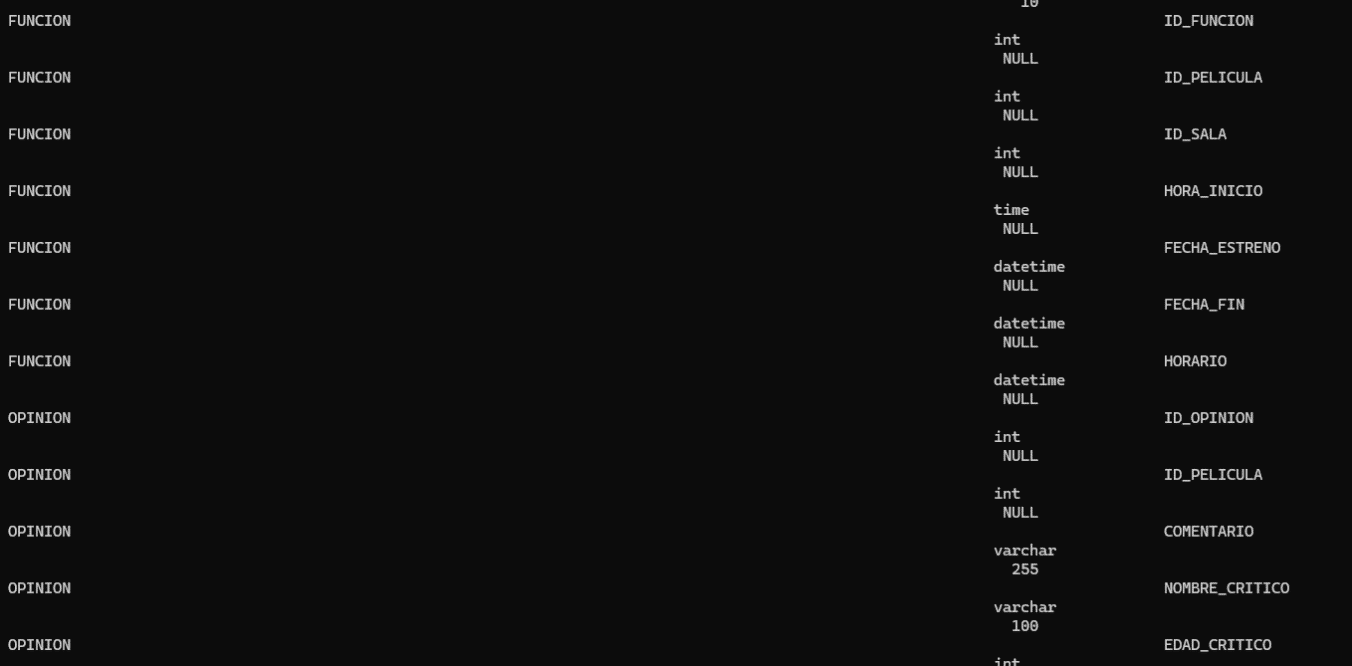
INSTRUCCIONES SQL PARA DESCRIBIR LAS TABLAS:

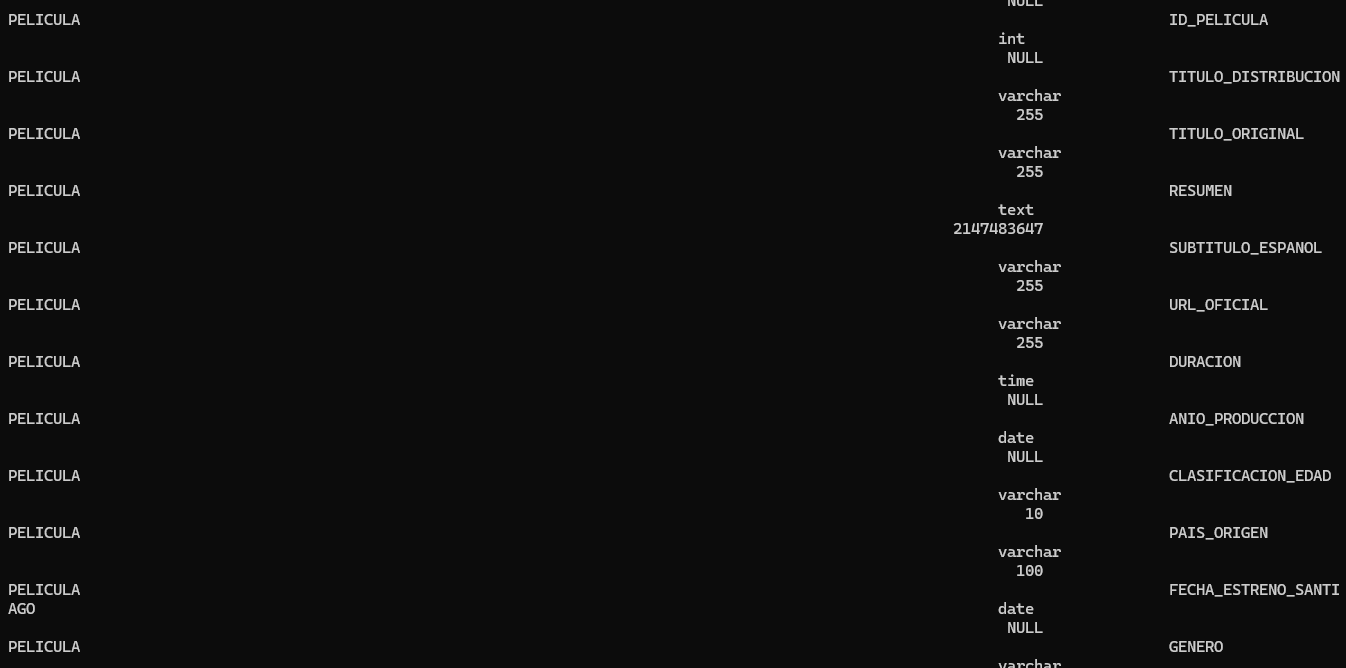
RECORTES DE PANTALLAS



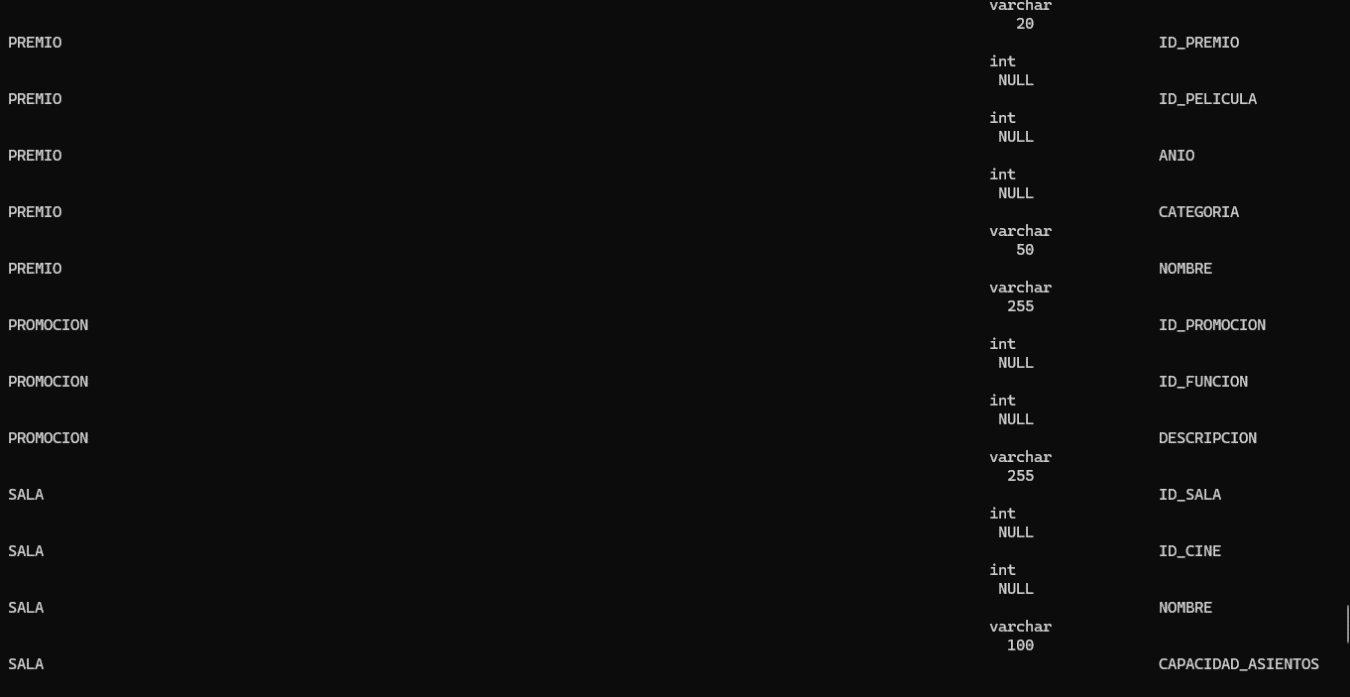














SELECT TABLE\_NAME, COLUMN\_NAME, DATA\_TYPE, CHARACTER\_MAXIMUM\_LENGTH FROM INFORMATION\_SCHEMA.COLUMNS ORDER BY TABLE\_NAME, ORDINAL\_POSITION;

GO

EXPLICACIÓN

Se describieron cada una de las tablas creadas anteriormente

1. Si las tablas aún no tienen las columnas que deben actuar como llaves foráneas, entonces, usted debe agregar dichas columnas a en respectivas tablas.

INSTRUCCIONES SQL PARA AGREGAR LAS COLUMNAS:

Las tablas al crearse si hicieron junto a sus llaves foráneas en los puntos anteriores

RECORTES DE PANTALLAS

Las tablas al crearse si hicieron junto a sus llaves foráneas en los puntos anteriores

EXPLICACIÓN

Las tablas al crearse si hicieron junto a sus llaves foráneas en los puntos anteriores

1. **Agregar** los **índices** en las columnas que actuarán como llaves foráneas

INSTRUCCIONES SQL PARA CREAR ÍNDICES:

CREATE NONCLUSTERED INDEX IDX\_OPINION\_PELICULA ON OPINION(ID\_PELICULA);

CREATE NONCLUSTERED INDEX IDX\_ESTADO\_PELICULA ON ESTADO(ID\_PELICULA);

CREATE NONCLUSTERED INDEX IDX\_DESCUENTO\_TIPO ON DESCUENTO(TIPO\_DESCUENTO);

CREATE NONCLUSTERED INDEX IDX\_PREMIO\_PELICULA ON PREMIO(ID\_PELICULA);

CREATE NONCLUSTERED INDEX IDX\_ADAPTACION\_PELICULA ON ADAPTACION(ID\_PELICULA);

CREATE NONCLUSTERED INDEX IDX\_FUNCION\_PELICULA ON FUNCION(ID\_PELICULA);

CREATE NONCLUSTERED INDEX IDX\_FUNCION\_SALA ON FUNCION(ID\_SALA);

CREATE NONCLUSTERED INDEX IDX\_PELICULA\_TITULO ON PELICULA(TITULO\_DISTRIBUCION);

CREATE NONCLUSTERED INDEX IDX\_PELICULA\_CLASIFICACION ON PELICULA(CLASIFICACION\_EDAD);

CREATE NONCLUSTERED INDEX IDX\_PELICULA\_GENERO ON PELICULA(GENERO);

CREATE NONCLUSTERED INDEX IDX\_PELICULA\_IDIOMA ON PELICULA(IDIOMA\_ORIGINAL);

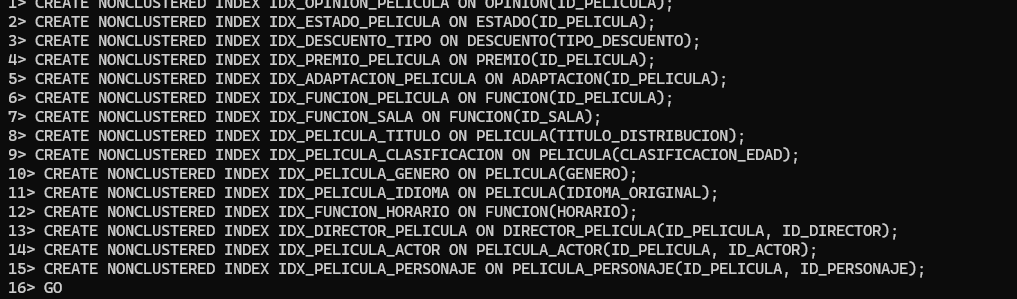
CREATE NONCLUSTERED INDEX IDX\_FUNCION\_HORARIO ON FUNCION(HORARIO);

CREATE NONCLUSTERED INDEX IDX\_DIRECTOR\_PELICULA ON DIRECTOR\_PELICULA(ID\_PELICULA, ID\_DIRECTOR);

CREATE NONCLUSTERED INDEX IDX\_PELICULA\_ACTOR ON PELICULA\_ACTOR(ID\_PELICULA, ID\_ACTOR);

CREATE NONCLUSTERED INDEX IDX\_PELICULA\_PERSONAJE ON PELICULA\_PERSONAJE(ID\_PELICULA, ID\_PERSONAJE);

GO;RECORTES DE PANTALLAS



EXPLICACIÓN

Se agregaron los índices necesarios para optimizar las consultas descritas en el ejercicio

1. **Agregar** restricciones de **llave foránea** en cada una de las tablas a las que haya lugar según el **Modelo Relacional**, cada llave foránea debe aplicar restricción de integridad referencial Restrictiva para operaciones de Eliminar y Cascada para Operaciones de Actualizar.

**(Estudiar e investigar qué es integridad referencial, como se aplica esto en las llaves foráneas)**

INSTRUCCIONES SQL PARA AGREGAR LAS LLAVES FORÁNEAS:

Esta restricción se le aplico a todas las llaves foráneas al momento de ser creadas en puntos anteriores en puntos

ON UPDATE CASCADE ON DELETE CASCADE

RECORTES DE PANTALLAS

Esta restricción se le aplico a todas las llaves foráneas al momento de ser creadas en puntos anteriores en puntos

ON UPDATE CASCADE ON DELETE CASCADE

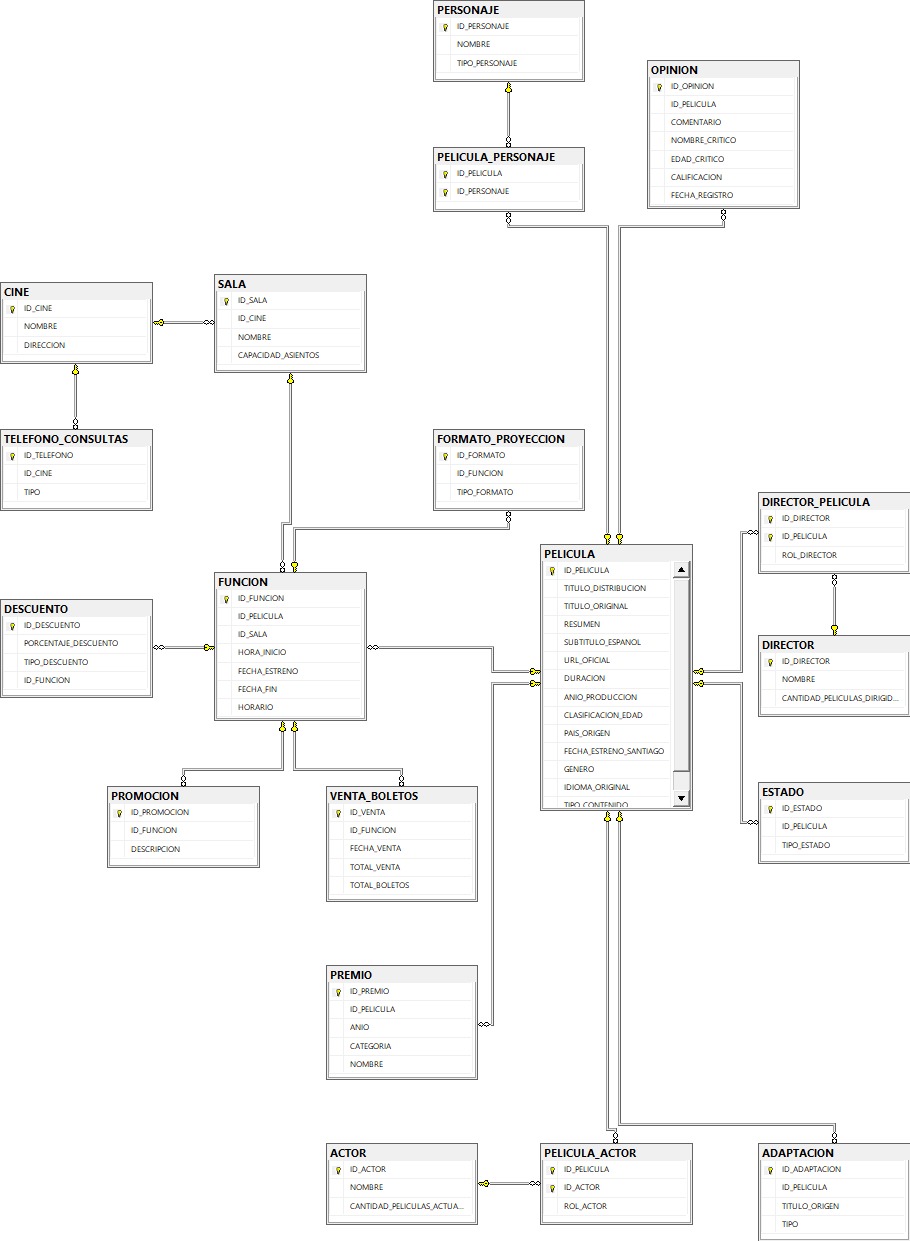
EXPLICACIÓN

Esta restricción se le aplico a todas las llaves foráneas al momento de ser creadas en puntos anteriores en puntos

ON UPDATE CASCADE ON DELETE CASCADE

1. Utilizar alguna herramienta para visualizar el Modelo Relacional a partir de la BD ya creada. Los tres motores de BD que usaron para el desarrollo de esta actividad ofrecen una herramienta que nos permite ingresar instrucciones SQL al motor de BD de forma rápida e intuitiva, gracias a que están desarrolladas usando Interfaz Gráfica de Usuario (GUI), también nos permiten gestionar cómodamente todos los componentes del servidor y los elementos de cualquier BD dentro de dicho motor, así como también, nos permiten generar automáticamente el MR (Modelo Relacional) seleccionado una BD previamente creará, o viceversa, nos permiten crear la BD a partir de un MR.

COLOCAR AQUÍ LA IMAGEN DEL MODELO RELACIONAL GENERADO AUTOMÁTICAMENTE



PASOS PARA GENERAR EL MODELO RELACIONAL A PARTIR DE LA BD CREADA EN EL RESPECTIVO MOTOR DE BD.

Entrar a la aplicación ssms y en la carpeta de databases, buscar nuestra base de datos y darle a la opción crear diagrama.

1. Crear un **Backup** solo **de la estructura** de la **bd\_tu\_nombre\_tu\_apellido\_xyz** y guardar el resultado en un archivo llamado **backup\_bd\_tu\_nombre\_tu\_apellido\_xyz.sql**

PASOS PARA GENERAR EL BACKUP DE LA BD CREADA EN EL RESPECTIVO MOTOR DE BD. (IDEAL UTILIZAR EL COMANDO SQL APROPIADO SEGÚN CADA MOTOR DE BD Y OPCIONALMENTE TAMBIÉN PUEDEN HACERLO Y EXPLICARLO USANDO EL MODO GRÁFICO)

Solo se ingresa el siguiente comando:

sqlcmd -S nombre\_servidor -U user\_addel\_perez -P AbcdeUdeC -Q "BACKUP DATABASE bd\_addel\_perez\_xyz TO DISK = 'C:\Backup\bd\_addel\_perez\_xyz.bak'"

1. Eliminar la Base de datos **bd\_tu\_nombre\_tu\_apellido\_xyz**

INSTRUCCIÓN SQL :

* DROP DATABASE bd\_addel\_perez\_xyz;
* GO

RECORTES DE PANTALLAS



EXPLICACIÓN

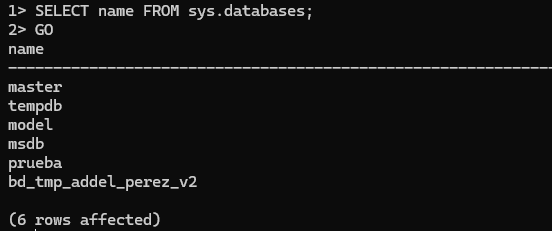
Se elimino la base de datos bd\_addel\_perez\_xyz

1. Mostrar las bases de datos del usuario **bd\_tu\_nombre\_tu\_apellido\_xyz**

INSTRUCCIÓN SQL :

* SELECT name FROM sys.databases;
* GO

RECORTES DE PANTALLAS



EXPLICACIÓN

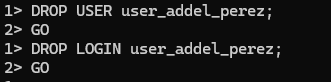
Evidencia de que se elimino la base de datos

1. Eliminar el Usuario **bd\_tu\_nombre\_tu\_apellido\_xyz**

INSTRUCCIÓN SQL :

* drop user user\_franky\_franco;

RECORTES DE PANTALLAS



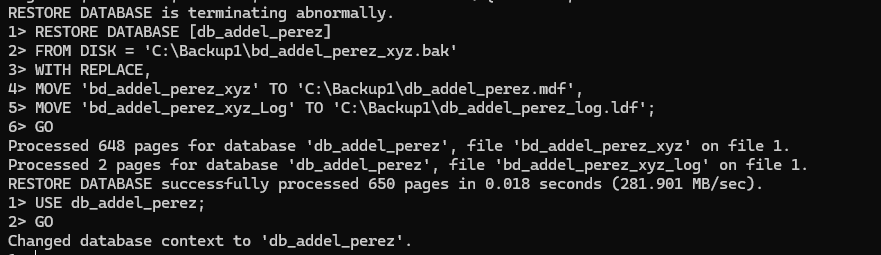
EXPLICACIÓN  
Se elimino al usuario user\_addel\_perez

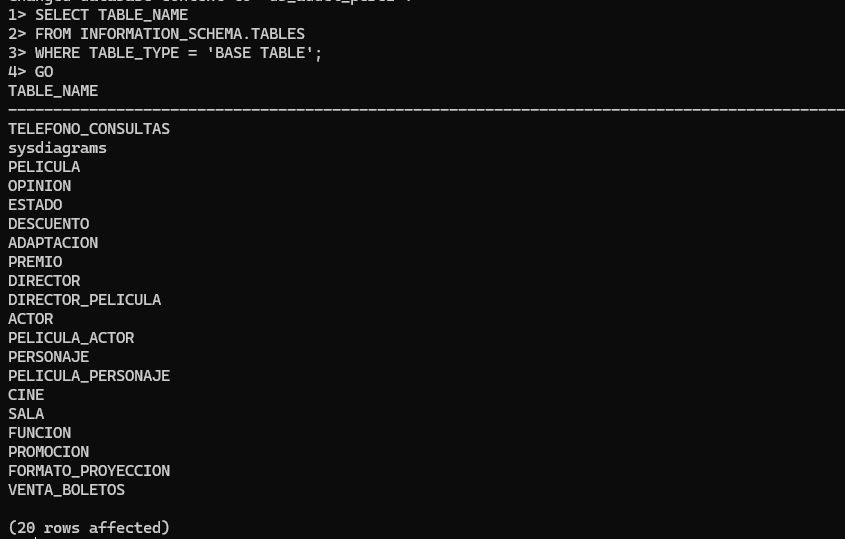
1. Utilizar el archivo de Backup para realizar restauración de la BD previamente eliminada.

INSTRUCCIÓN SQL :

RECORTES DE PANTALLAS







EXPLICACIÓN

Se recupero la base de datos recientemente borrada

1. De acuerdo a los dos motores de BD que usted ha elegido, al finalizar esta actividad usted debe estar en la capacidad de diligenciar la siguiente tabla:

|  |  |
| --- | --- |
| OPERACIÓN SQL | MySQL |
| Entrar a la terminal | sqlcmd -S localhost -U sa -P AbcdeUdeC |
| Crear un usuario | CREATE LOGIN user\_addel\_perez WITH PASSWORD = 'AbcdeUdeC'; |
| Asignar permisos a un usuario | ALTER SERVER ROLE sysadmin ADD MEMBER user\_addel\_perez; |
| Cambiar la clave a un usuario | ALTER LOGIN user\_addel\_perez WITH PASSWORD = '12345'; |
| Quitar los permisos a un usuario | ALTER SERVER ROLE sysadmin DROP MEMBER user\_addel\_perez; |
| Eliminar un usuario | DROP USER user\_addel\_perez; DROP LOGIN user\_addel\_perez; |
| Mostrar los usuarios activos en el Motor de BD | SELECT name FROM sys.syslogins; |
| Salid de la terminal | Exit |
| Mostrar las BD | SELECT name FROM sys.databases; |
| Entrar o usar una BD | USE bd\_addel\_perez\_xyz; |
| Eliminar una BD | DROP DATABASE bd\_addel\_perez\_xyz; |
| Crear la tabla del punto # 12 | CREATE TABLE tabla\_temporal ( id INT IDENTITY(1,1) PRIMARY KEY, columna2 VARCHAR(100) NOT NULL, columna3 INT, columna4 DECIMAL(10,2), columna5 DATE, columna6 DATETIME, columna7 TIME, columna8 DATETIME DEFAULT GETDATE(), columna9 TEXT, columna10 VARBINARY(MAX) -- Equivalente a BLOB en MySQL ); |
| Mostrar la estructura de una tabla | EXEC sp\_help pelicula; |
| Cambiar el nombre de una tabla | EXEC sp\_rename 'tabla\_temporal', 'tabla\_temporal\_v2'; |
| Agregar una columna a una tabla | ALTER TABLE tabla\_temporal\_v2 ADD email VARCHAR(70) UNIQUE DEFAULT 'abc@udec.edu.co'; |
| Cambiar el tipo de datos a una columna | ALTER TABLE tabla\_temporal\_v2 ALTER COLUMN columna7 DATETIME; |
| Cambiar el nombre de una columna | EXEC sp\_rename 'tabla\_temporal\_v2.columna4', 'columnaIV', 'COLUMN'; |
| Eliminar una columna | ALTER TABLE tabla\_temporal\_v2 DROP COLUMN columna6; |
| Tipos de datos Numéricos Enteros | INT |
| Tipos de datos numéricos reales o decimales | DECIMAL |
| Tipos de datos caracteres de longitud fija | CHAR |
| Tipos de datos caracteres de longitud variable | VARCHAR |
| Tipos de datos Boolean | BIT |
| Tipos de datos Fecha | DATE |
| Tipos de datos para guardar datos binarios grandes | VARBINARY(MAX) |
| Tipos de datos conjunto |  |
| Tipos de datos enumeración | CHET |
| Columnas con valor autoincremental, secuencial o serial | IDENTITY |
| Cómo Eliminar una tabla | DROP TABLE tabla\_temporal\_v2; |
| Agregar una Llave foránea al crear la tabla | CREATE TABLE funcion ( ID\_FUNCION INT PRIMARY KEY, ID\_CINE INT, FOREIGN KEY (ID\_CINE) REFERENCES CINE(ID\_CINE) ON DELETE CASCADE ); |
| Agregar una llave foránea después de creada la tabla | ALTER TABLE tabla\_hija ADD CONSTRAINT nombre\_FK FOREIGN KEY (columna\_llave\_foranea) REFERENCES tabla\_padre (columna\_llave\_primaria) ON UPDATE CASCADE ON DELETE CASCADE; |
| Crear un índice | CREATE INDEX idx\_funcion\_fecha ON FUNCION (FECHA\_ESTRENO); |
| Mostrar los índice creados | EXEC sp\_helpindex 'FUNCION'; |
| Mostrar las restricciones de llave foránea de una tabla | EXEC sp\_help 'FUNCION'; |
| Eliminar una restricción de llave foránea o cualquier otra restricción | ALTER TABLE nombre\_tabla DROP CONSTRAINT nombre\_restriccion; |
| Hacer un backup | BACKUP DATABASE bd\_addel\_perez\_xyz TO DISK = 'C:\Users\HP\OneDrive\Documentos\Backup\_bd\_addel\_perez\_xyz.bak'; |
| Restaurar la BD | RESTORE DATABASE bd\_addel\_perez\_xyz FROM DISK = 'C:\Users\HP\OneDrive\Documentos\Backup\_bd\_addel\_perez\_xyz.bak'; |

LINK VIDEOS:

<https://youtu.be/NzUPu_t-dJU>

https://youtu.be/j9h56eL8RNc