# **PARCIAL FINAL POSTGRES**

**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** 3](#_Toc192270171)

[**Justificación** 5](#_Toc192270172)

[**Desarrollo** 6](#_Toc192270173)

[**Identificación de las Interrelaciones entre el Conjunto de Entidades** 11](#_Toc192270174)

[**Diagrama Entidad Relación App Cine(DER):** 12](#_Toc192270175)

[**Pasar el DER(Diagrama Entidad Relacion) a MR (Modelo Relacional)** 13](#_Toc192270176)

[**Modelo Relacional App Cine (MR)** 16](#_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)

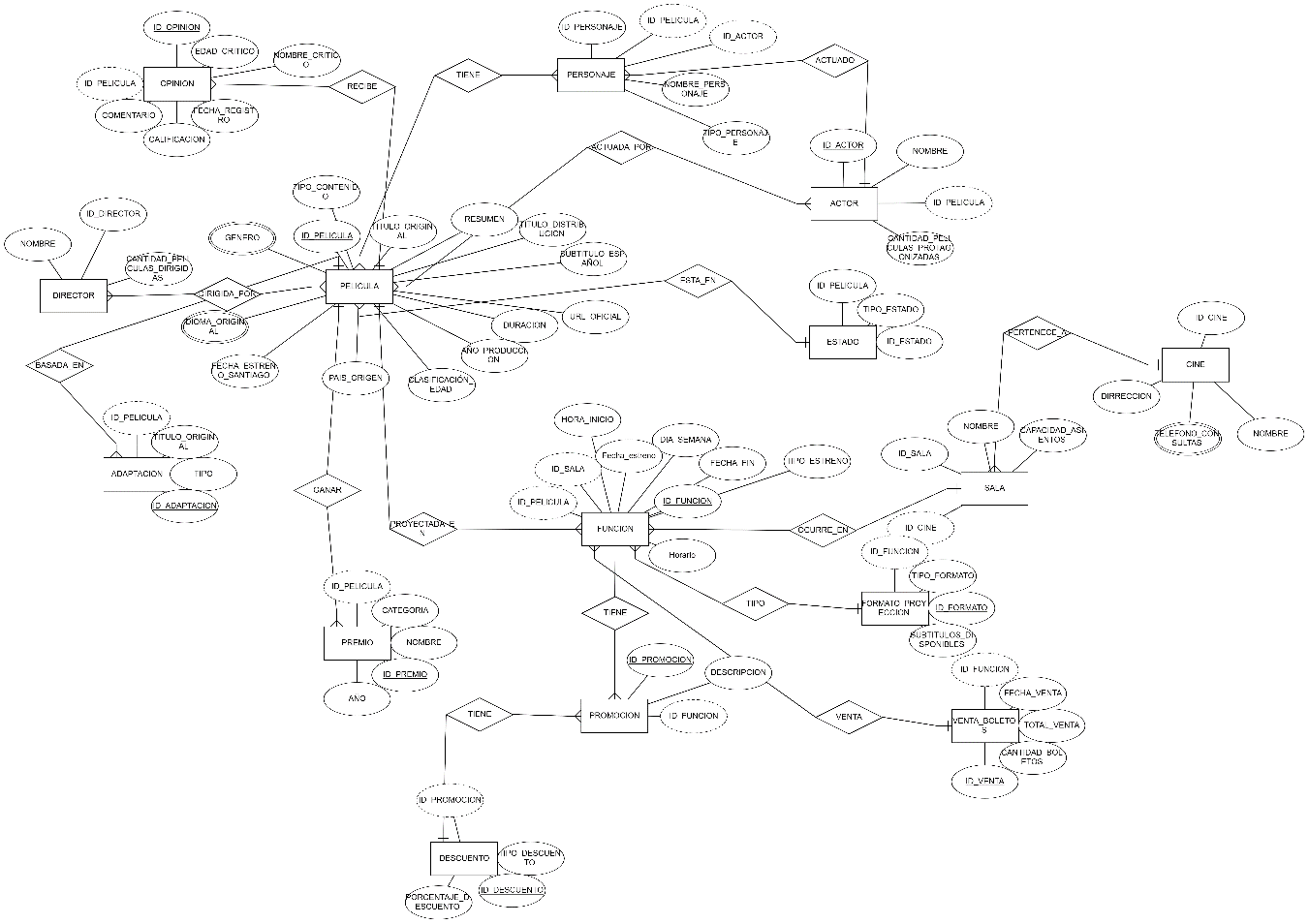
 **Promocion** (ID\_Promocion, ID\_Funcion, Descripcion)

 **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**

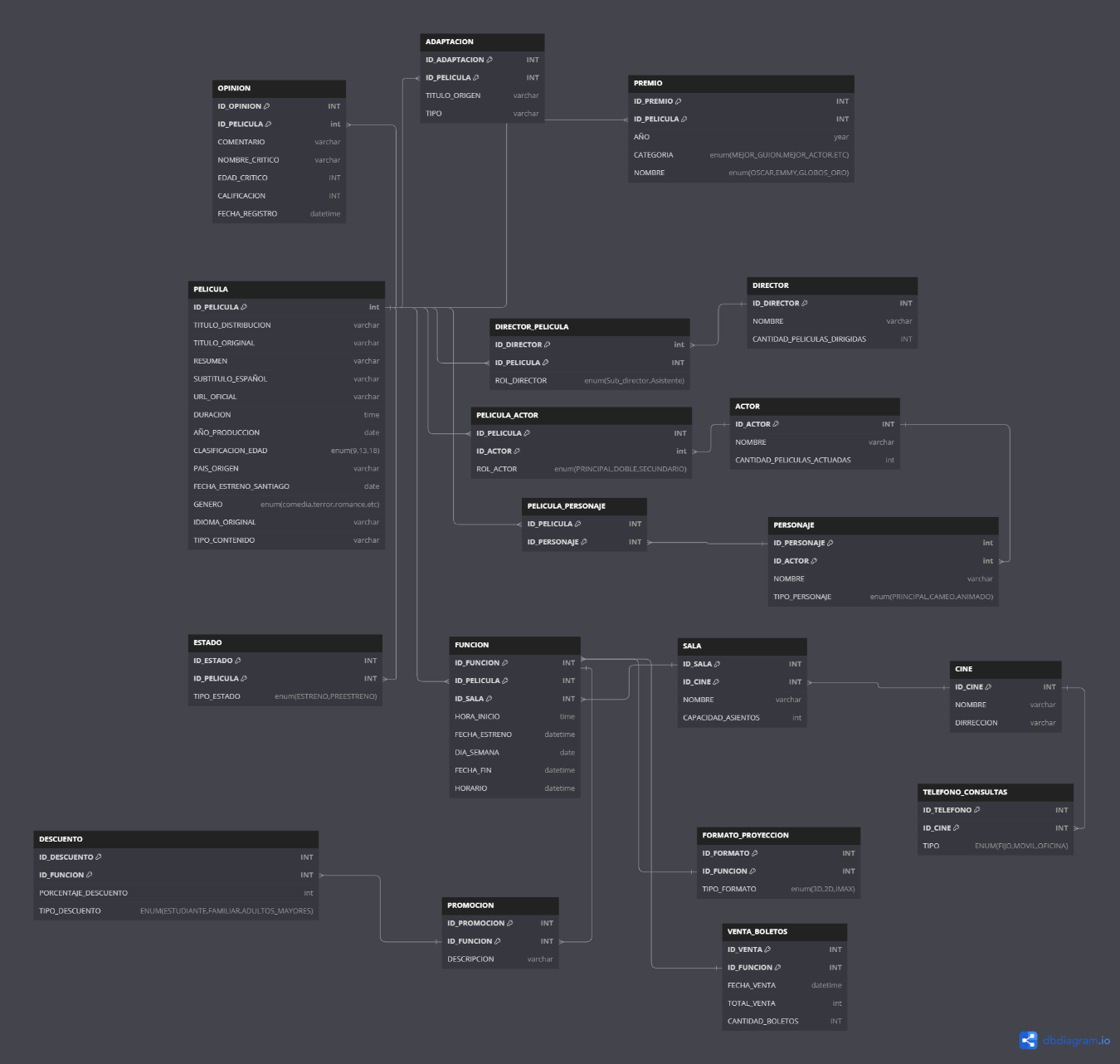
#### **Promocion**

* **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)**

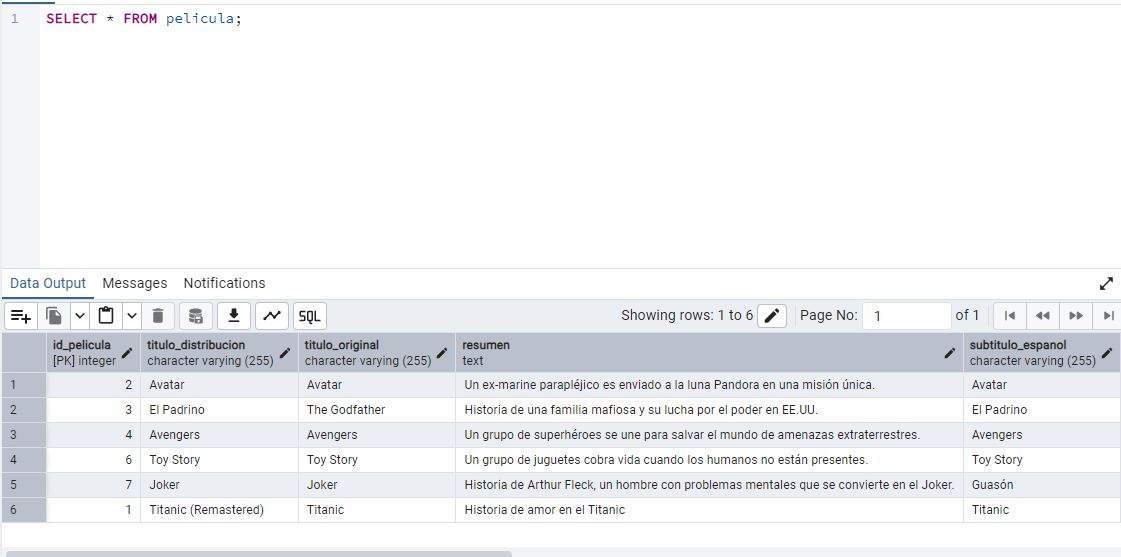


Tomar alguna de las tablas (TablaX, TablaY, TablaZ) de la base de datos bd\_grupo\_xyz y realizar las siguientes operaciones.

1. Mostrar **todos los registros** de la TablaX:

**INSTRUCCIÓN SQL**:

SELECT \* FROM pelicula;



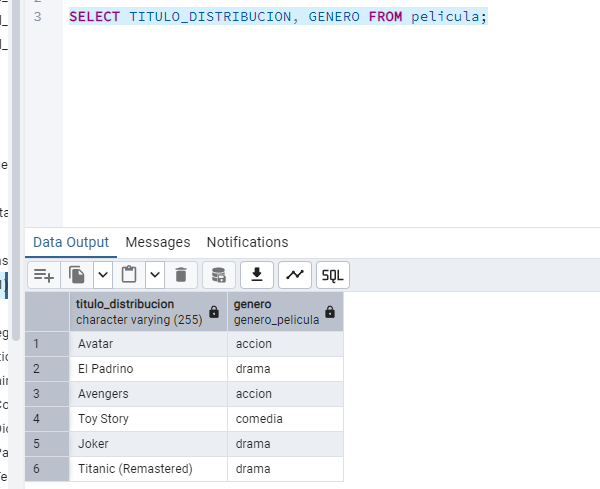
Explicación:

Mostramos todos los registros de la tabla pelicula

1. **Mostrar el valor solo de 2 columnas** de todos los registros de la TablaX

**INSTRUCCIÓN SQL**:

SELECT TITULO\_DISTRIBUCION, GENERO FROM pelicula;

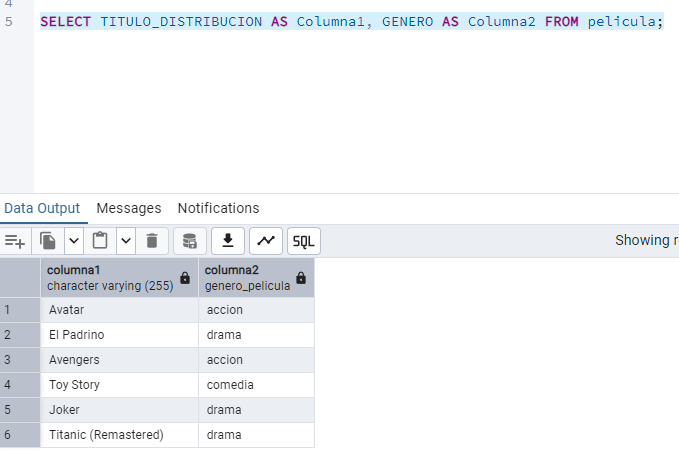
Explicación:

Esta vez lo que hacemos es escoger solamente dos columnas de la tabla películas y mostrarla

1. Mostrar el valor de 2 columnas de todos los registros de la TablaX, pero en el resultado de la consulta el **nombre de esas dos columnas debe ser** Columna1, Column2 respectivamente:

**INSTRUCCIÓN SQL**:

SELECT TITULO\_DISTRIBUCION AS Columna1, GENERO AS Columna2 FROM pelicula;



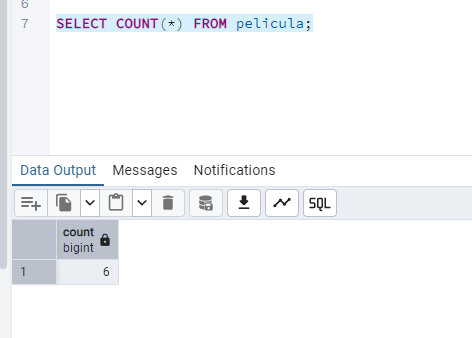
Explicación:

Volvemos a mostrar las columnas anteriores pero renombras a “columna1” y “columna2”

1. Mostrar el **número de registros** almacenados en la TablaX:

**INSTRUCCIÓN SQL**:

SELECT COUNT(\*) FROM pelicula;

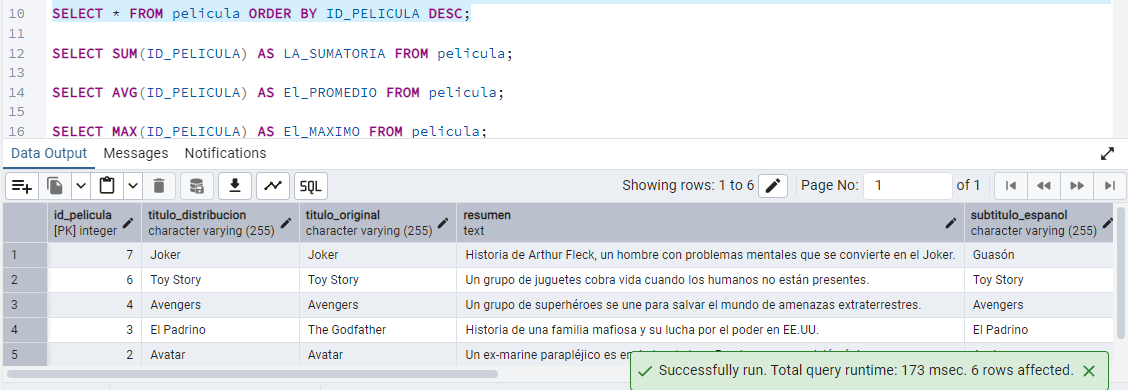


Explicación: Mostramos todos los registros que hay en nuestra tabla pelicula-

1. Mostrar todos los registros de la TablaX **ordenados descendentemente**:

**INSTRUCCIÓN SQL**:

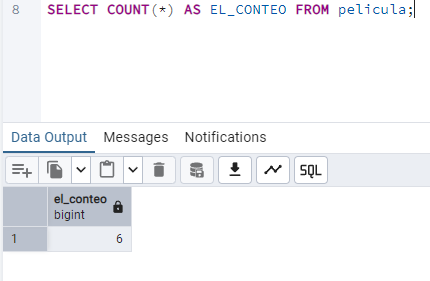
SELECT \* FROM pelicula ORDER BY ID\_PELICULA DESC;



1. Mostrar el **número de registros** almacenados en la TablaX, pero el valor del resultado debe estar almacenado en una columna llamada EL\_CONTEO.

**INSTRUCCIÓN SQL**:

SELECT COUNT(\*) AS EL\_CONTEO FROM pelicula;

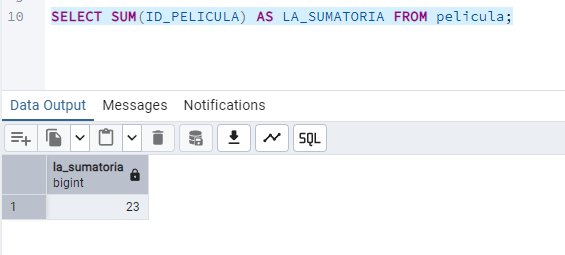


Explicación: Modificamos el nombre de la columna “EL\_CONTEO” y mostramos.

1. Realizar una consulta en la TabalX o la Tabla Y o la TablaZ en la que se pueda obtener una **sumatoria** de valores de alguna columna numérica, pero el valor del resultado debe estar almacenado en una columna llamada LA\_SUMATORIA.

**INSTRUCCIÓN SQL**:

SELECT SUM(ID\_PELICULA) AS LA\_SUMATORIA FROM pelicula;

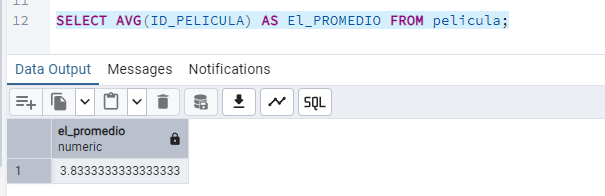


Explicación: Seleccionamos la columna “ID\_PELICULA” ya que la es la unica en la que podemos hacer una sumatoria y mostramos el resultado.

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ en la que se pueda obtener el **promedio** de valores de alguna columna numérica, pero el valor del resultado debe estar almacenado en una columna llamada El\_PROMEDIO.

**INSTRUCCIÓN SQL**:

SELECT AVG(ID\_PELICULA) AS El\_PROMEDIO FROM pelicula;

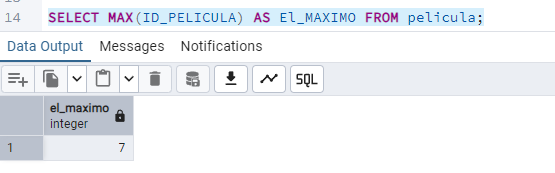


Explicación: Ahora obtenemos el promedio de esa misma columna y mostramos.

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ en la que se pueda obtener el **el valor máximo** de alguna columna numérica, pero el valor del resultado debe estar almacenado en una columna llamada El\_MAXIMO.

**INSTRUCCIÓN SQL**:

SELECT MAX(ID\_PELICULA) AS El\_MAXIMO FROM pelicula;



Explicación:De la misma columna ID obtenemos el valor maximo y modificamos el nombre el como se va a mostrar.

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ en la que se pueda obtener el **el valor mínimo** de alguna columna numérica, pero el valor del resultado debe estar almacenado en una columna llamada El\_MINIMO.

**INSTRUCCIÓN SQL**:

SELECT MIN(ID\_PELICULA) AS El\_MINIMO FROM pelicula;

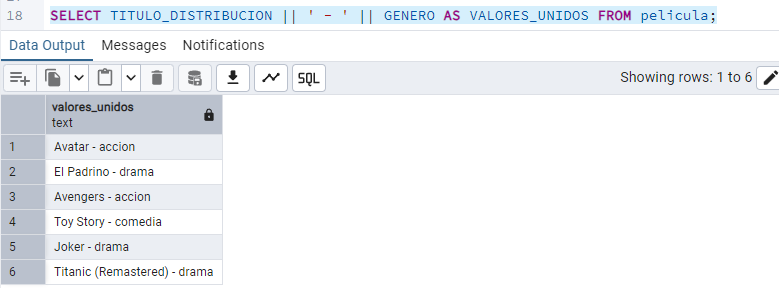


Explicación:De la misma columna ID obtenemos el valor minimo y modificamos el nombre el como se va a mostrar.

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar los valores de dos columnas **concatenados** en una sola columna, pero el valor del resultado debe estar almacenado en una columna llamada VALORES\_UNIDOS.

**INSTRUCCIÓN SQL**:

SELECT TITULO\_DISTRIBUCION || ' - ' || GENERO AS VALORES\_UNIDOS FROM pelicula;



Explicación: Concatenamos la columna titulo y genero y lo mostramos como valores\_unidos

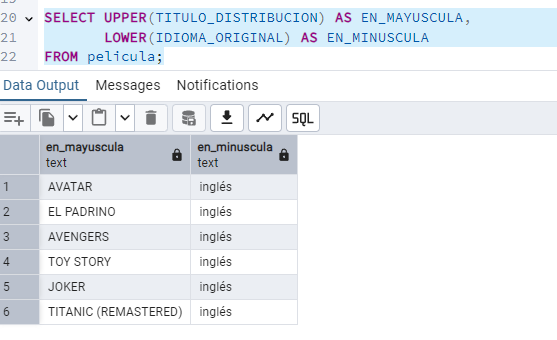
1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar el **MAYÚSCULAS** los valores de una columna, y en **Minúsculas** los valores de otra columna, pero en el resultado estas columnas deben llamarse EN\_MAYUSCULA, EN\_MINUSCULA respectivamente.

**INSTRUCCIÓN SQL**:

SELECT UPPER(TITULO\_DISTRIBUCION) AS EN\_MAYUSCULA,

LOWER(IDIOMA\_ORIGINAL) AS EN\_MINUSCULA

FROM pelicula;

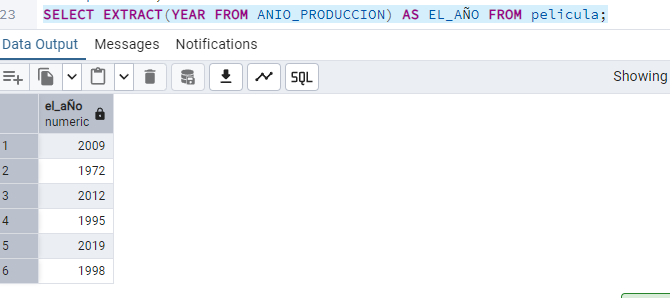


Explicación: En este caso mostramos el titulo en mayuscula y el idioma en miniscula y renombramos las columnas.

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar sólo **el AÑO** de alguna columna Tipo Fecha, pero en el resultado esta columna debe llamarse EL\_AÑO.

**INSTRUCCIÓN SQL**:

SELECT EXTRACT(YEAR FROM ANIO\_PRODUCCION) AS EL\_AÑO FROM pelicula;

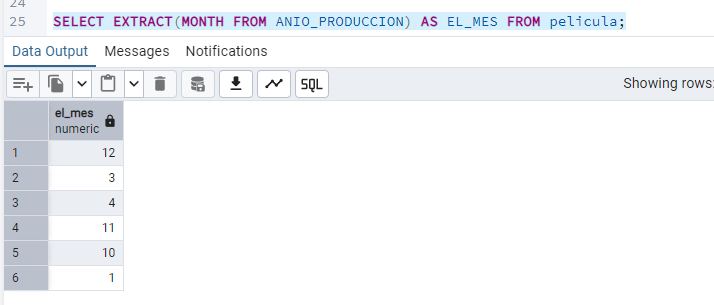


Explicación: Escogemos mi columna ANIO\_PRODUCCION y mostramos solamente el año y renombramos la columna en el que se mostrara.

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar sólo **el MES** de alguna columna Tipo Fecha, pero en el resultado esta columna debe llamarse EL\_MES:

**INSTRUCCIÓN SQL**:

SELECT EXTRACT(MONTH FROM ANIO\_PRODUCCION) AS EL\_MES FROM pelicula;



Explicación: Un caso parecido al anterior instrucción solo que ca mostraremos solo el mes y se renombra la columna a EL\_MES.

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar sólo **el DIA** de alguna columna Tipo Fecha, pero en el resultado esta columna debe llamarse EL\_DIA..

**INSTRUCCIÓN SQL**:

SELECT EXTRACT(DAY FROM ANIO\_PRODUCCION) AS EL\_DIA FROM pelicula;

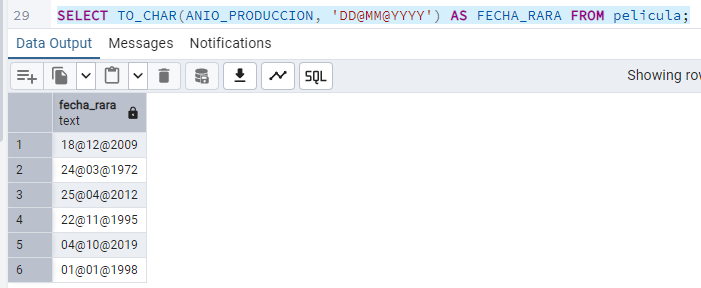


Explicación: Lo mismo que los ejemplos anteriores pero solo mostrando el dia y renombrando la columna a EL\_DIA

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar sólo **la fecha en formato personalizado,** así **(dia@mes@año)** de alguna columna Tipo Fecha, pero en el resultado esta columna debe llamarse FECHA\_RARA.:

**INSTRUCCIÓN SQL**:

SELECT TO\_CHAR(ANIO\_PRODUCCION, 'DD@MM@YYYY') AS FECHA\_RARA FROM pelicula;

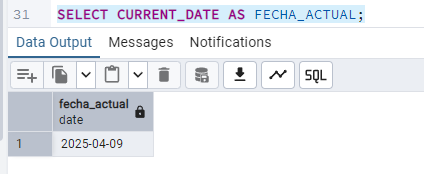


Explicación: Personalizamos el como queremos que muestre la fecha y renombramos la columna a FECHA\_RARA

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar sólo **la FECHA ACTUAL** en una nueva columna**,** pero en el resultado esta columna debe llamarse FECHA\_ACTUAL.

**INSTRUCCIÓN SQL**:

SELECT CURRENT\_DATE AS FECHA\_ACTUAL;



Explicación: Mostraremos en una columna nueva la fecha actual.

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar en 3 nuevas columnas el resultado de **SUMAR**, **RESTAR** Y **MULTIPLICAR** los valores de 2 columnas Tipo numérico, pero en el resultado esta columna debe llamarse LA\_SUMA, LA\_RESTA, LA\_MULTIPLICACION respectivamente.

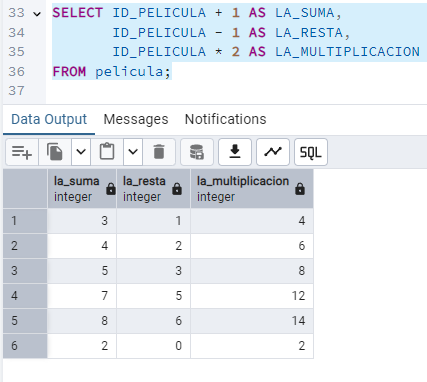
**INSTRUCCIÓN SQL**:

SELECT ID\_PELICULA + 1 AS LA\_SUMA,

ID\_PELICULA - 1 AS LA\_RESTA,

ID\_PELICULA \* 2 AS LA\_MULTIPLICACION

FROM pelicula;



Explicación: Sumamos, restamos y multiplicamos a ID\_PELICULA

1. Investigar cual es el objetivo de la Palabra DISTINCT de SQL

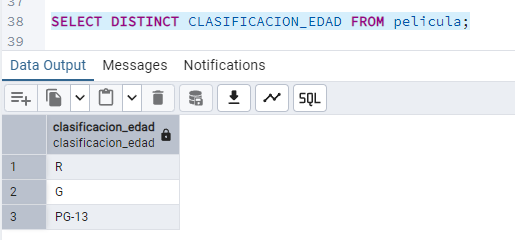
**INSTRUCCIÓN SQL**:

La palabra clave DISTINCT se utiliza en SQL para **eliminar los valores duplicados** en el resultado de una consulta. Es decir, cuando se realiza una consulta que puede retornar filas con valores repetidos en una o más columnas, DISTINCT asegura que en el resultado solo aparezcan **valores únicos**.

1. Seleccionar alguna de las tablas TablaX o la TablaY o la TablaZ y realizar una consulta donde se pueda apreciar el uso de la palabra **DISTINCT**

**INSTRUCCIÓN SQL**:

SELECT DISTINCT CLASIFICACION\_EDAD FROM pelicula;

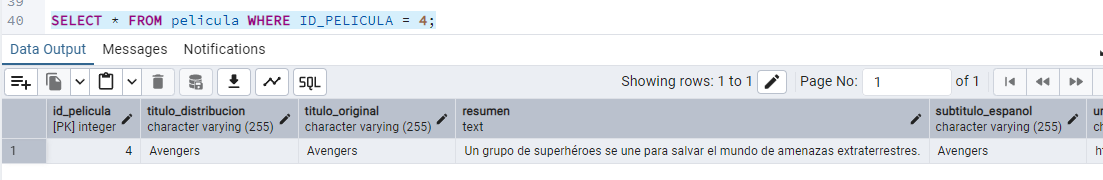


Explicación: Elimina todos los duplicados que puedan existir.

1. Realizar una consulta en la TablaX que permita mostrar los datos de un registro cuya columna llave primaria sea **igual** al valorY

**INSTRUCCIÓN SQL**:

SELECT \* FROM pelicula WHERE ID\_PELICULA = 4;

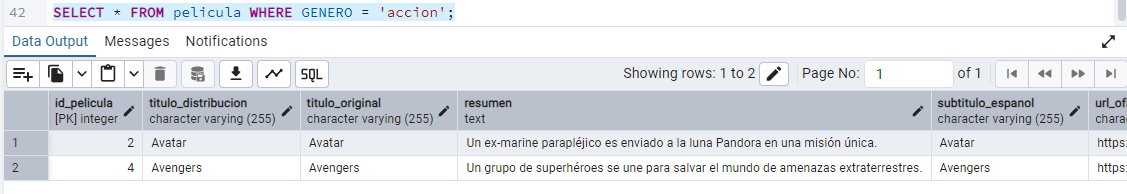


Explicación: Buscamos por llave primaria.

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **igual** datoZ

**INSTRUCCIÓN SQL**:

SELECT \* FROM pelicula WHERE GENERO = 'accion';

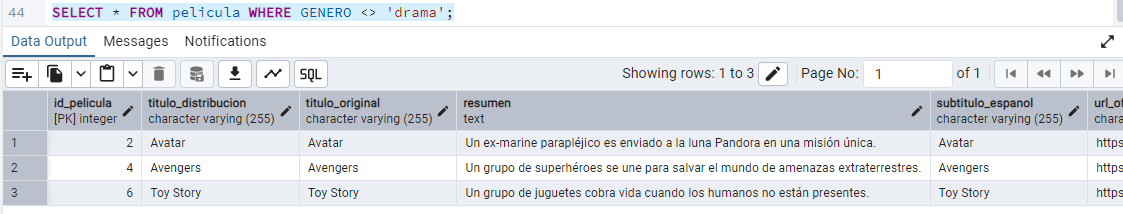


Explicación: Buscamos en la tabla pelicula resultados en base a Genero = accion

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **diferente**  datoZ

**INSTRUCCIÓN SQL**:

SELECT \* FROM pelicula WHERE GENERO <> 'drama';

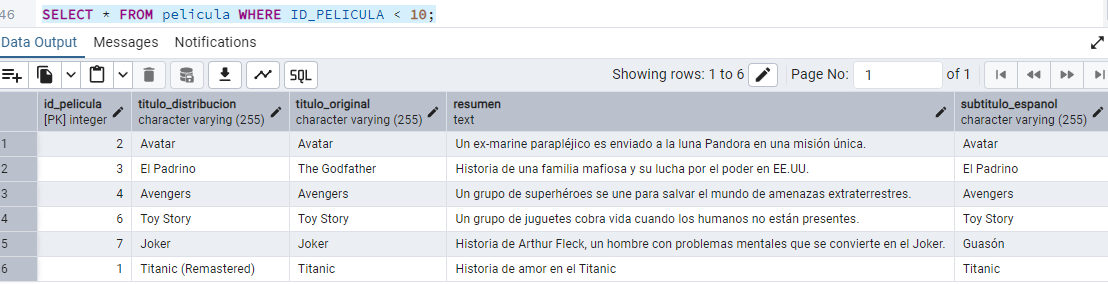


Explicacion: Ahora buscamos resultado diferentes a Genero <> Terror

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **menor**  datoZ

**INSTRUCCIÓN SQL**:

SELECT \* FROM pelicula WHERE ID\_PELICULA < 10;

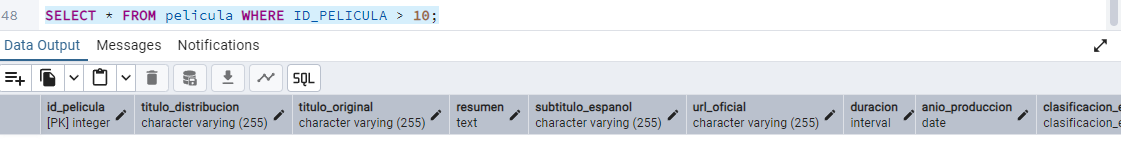


Explicación: Buscamos resultados menores al id 10.

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **mayor** datoZ

**INSTRUCCIÓN SQL**:

SELECT \* FROM pelicula WHERE ID\_PELICULA > 10;

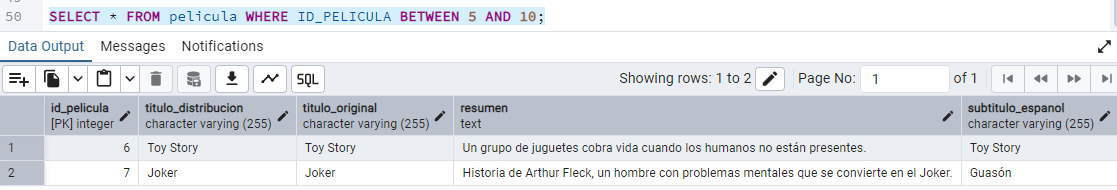


Explicación: Buscamos resultados mayores al id 10.

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **entre valor** datoX y valorZ

**INSTRUCCIÓN SQL**:

SELECT \* FROM pelicula WHERE ID\_PELICULA BETWEEN 5 AND 10;

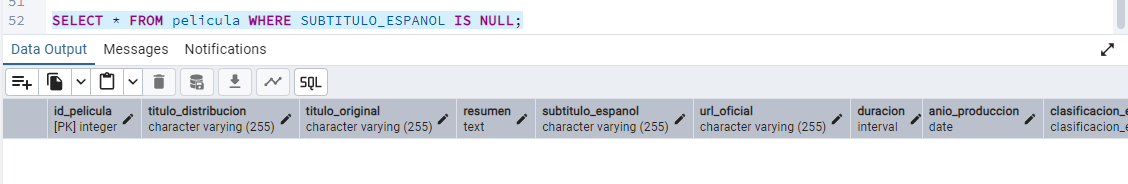


Explicación: Buscamos resultados que esten entre el id 5 yl id 10.

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **Sea NULO**

**INSTRUCCIÓN SQL**:

SELECT \* FROM pelicula WHERE SUBTITULO\_ESPANOL IS NULL;

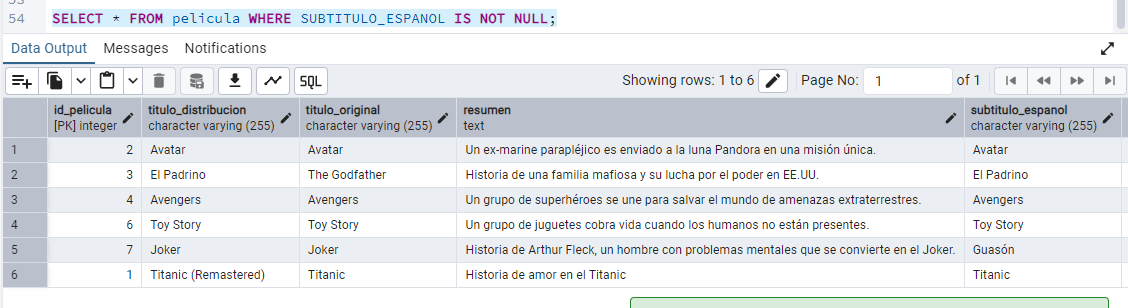


Explicación: Muestra valores q son NULL

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **No Sea NULO**

**INSTRUCCIÓN SQL**:

SELECT \* FROM pelicula WHERE SUBTITULO\_ESPANOL IS NOT NULL;

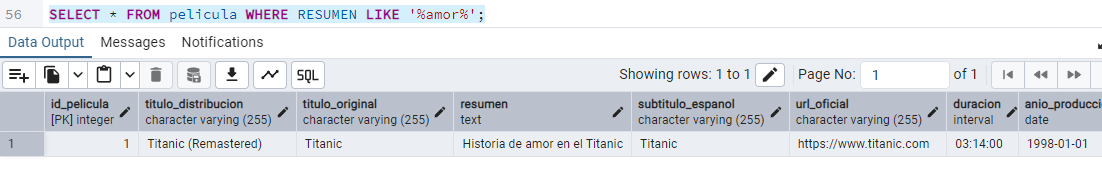


Explicación: mostramos valores que no son NULL

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **Contenga el valor** DatoX en su interior

**INSTRUCCIÓN SQL**:

SELECT \* FROM pelicula WHERE RESUMEN LIKE '%amor%';

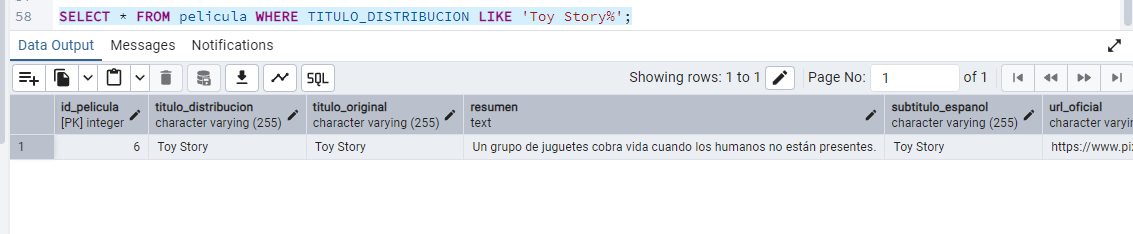


Explicación:Mostramos una pelicula que contenga amor.

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **Inicie con el valor** DatoX

**INSTRUCCIÓN SQL**:

SELECT \* FROM pelicula WHERE TITULO\_DISTRIBUCION LIKE 'Toy Story%';

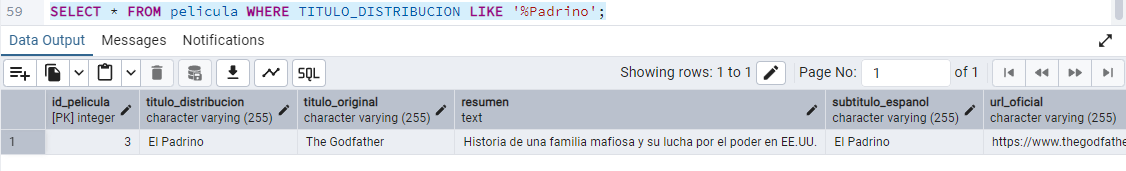


Explicación: Mostramos resultados que empiecen con “Inception”

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **Termine con el valor** DatoX

**INSTRUCCIÓN SQL**:

SELECT \* FROM pelicula WHERE TITULO\_DISTRIBUCION LIKE '%padrino';



Explicación: Mostramos resultado que terminen “knight”

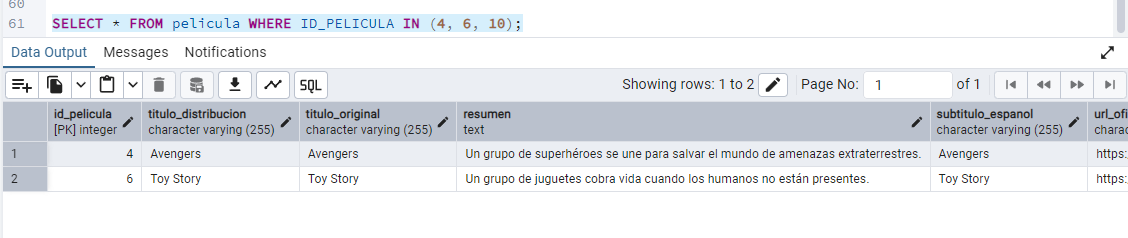
1. Investigar cual es el objetivo del comando IN.

Permite filtrar registros que coinciden con una lista de valores.

1. Realizar una consulta en la TablaX que permita mostrar el uso y aplicación de la palabra **IN**

**INSTRUCCIÓN SQL**:

SELECT \* FROM pelicula WHERE ID\_PELICULA IN (4, 6, 10);

-

Explicación: Filtramos entre los id 4 6 10.

Para realizar las siguientes consultas se asume que TablaA, TablaB, TablaC, …, TablaZ, representan relacionadas entre sí, mediante llaves foráneas, que se describen en El Modelo Relacional del ejercicio que han seleccionado.

Asumimos que TablaA y TablaB están relacionados así:

[**TablaA**] ->----<**R**>----|- [**TablaB**] ->----<**R**>----|- [**TablaC**]

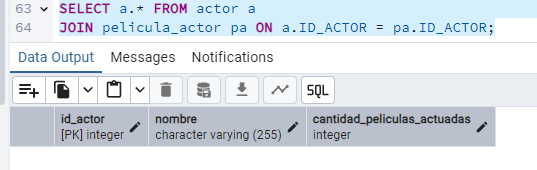
Es decir, [**TablaA**] tiene una llave foránea llamada por ejemplo **tablaB\_id**, la cual se conecta con la llave primaria **id** de la [**TablaB**], mientras que [**TablaB**] tiene una llave foránea llamada por ejemplo **tablaC\_id** la cual se conecta con la llave primaria **id** de la [**TablaC**],

1. Realizar una consulta que permitan mostrar los registros de la **TablaA que están relacionados** con la **TablaB**

**INSTRUCCIÓN SQL**:

SELECT a.\* FROM actor a

JOIN pelicula\_actor pa ON a.ID\_ACTOR = pa.ID\_ACTOR;



No me mostrara nada por falta de datos

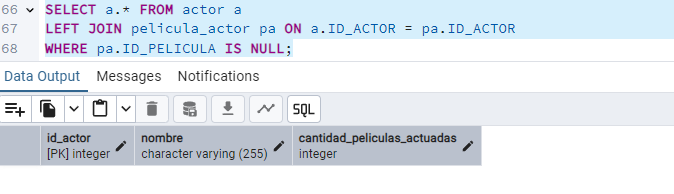
1. Realizar una consulta que permitan mostrar los registros de la **TablaA** que **NO están relacionados** con la **TablaB**

**INSTRUCCIÓN SQL**:

SELECT a.\* FROM actor a

LEFT JOIN pelicula\_actor pa ON a.ID\_ACTOR = pa.ID\_ACTOR

WHERE pa.ID\_PELICULA IS NULL;



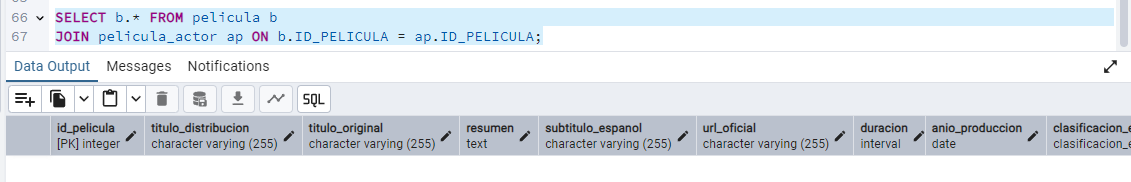
No me mostrara nada por falta de datos

1. Realizar una consulta que permitan mostrar los registros de la **TablaB** que **están relacionados** con la **TablaA**

**INSTRUCCIÓN SQL**:

SELECT b.\* FROM pelicula b

JOIN actor\_pelicula ap ON b.ID\_PELICULA = ap.ID\_PELICULA;



No me mostrara nada por falta de datos

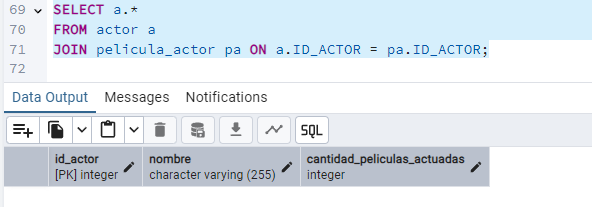
1. Realizar una consulta que permitan mostrar los registros de la **TablaB** que **NO están relacionados** con la **TablaA**

**INSTRUCCIÓN SQL**:

SELECT a.\*

FROM actor a

JOIN pelicula\_actor pa ON a.ID\_ACTOR = pa.ID\_ACTOR;



No me mostrara nada por falta de datos

https://youtu.be/l17pXEkPRz8