

BASE DE DATOS PARA EL ANÁLISIS DE PELÍCULAS CON LA FINALIDAD DE CONOCER LA RENTABILIDAD Y VISIBILIZACIÓN

Autora

Ing. Adriana Silvio

Academia

Coderhouse

Fecha de presentación

Septiembre/2024

Índice

Introducción	2
Objetivo	2
Situación problemática	3
Modelo de negocio	3
Diagrama entidad-relación de la base de datos	3
Listado de columna de cada tabla	4
Script de creación de cada objeto de la base de datos	8
TABLAS	8
VISTAS	12
STORED PROCEDURES	15
FUNCIONES	17
TRIGGERS	18
Script en SQL de inserción de datos	20
Informe	33
Herramientas y tecnologías usadas	33

Introducción

En el mundo del entretenimiento moderno, la industria cinematográfica ha evolucionado significativamente, especialmente con el crecimiento de las plataformas de transmisión en línea y los servicios de streaming. A medida que las películas y series se distribuyen en múltiples plataformas, la gestión de datos relacionados con producción, actores, directores, ingresos, reseñas y premios se vuelve cada vez más compleja. Para abordar este desafío, el diseño de una base de datos bien estructurada y optimizada es fundamental.

Este proyecto presenta el diseño e implementación de una base de datos de películas utilizando MySQL Workbench, que permite gestionar datos relacionados con películas, compañías productoras, actores, directores, plataformas de streaming, ingresos, premios y calificaciones. Con esta base de datos, se busca facilitar el análisis de la rentabilidad, la popularidad y el rendimiento de las películas en diferentes plataformas y, al mismo tiempo, proporcionar un sistema para rastrear actores, directores y sus respectivas contribuciones a lo largo del tiempo.

Objetivo

El objetivo principal de este proyecto es diseñar una base de datos relacional que permita almacenar y analizar información relevante sobre películas, su desempeño financiero, su distribución en plataformas de streaming, las calificaciones que reciben, y la participación de actores y directores. Este sistema proporcionará una estructura centralizada para la consulta y análisis de datos en la industria cinematográfica, mejorando la toma de decisiones en torno a la rentabilidad y popularidad de las películas.

Situación problemática

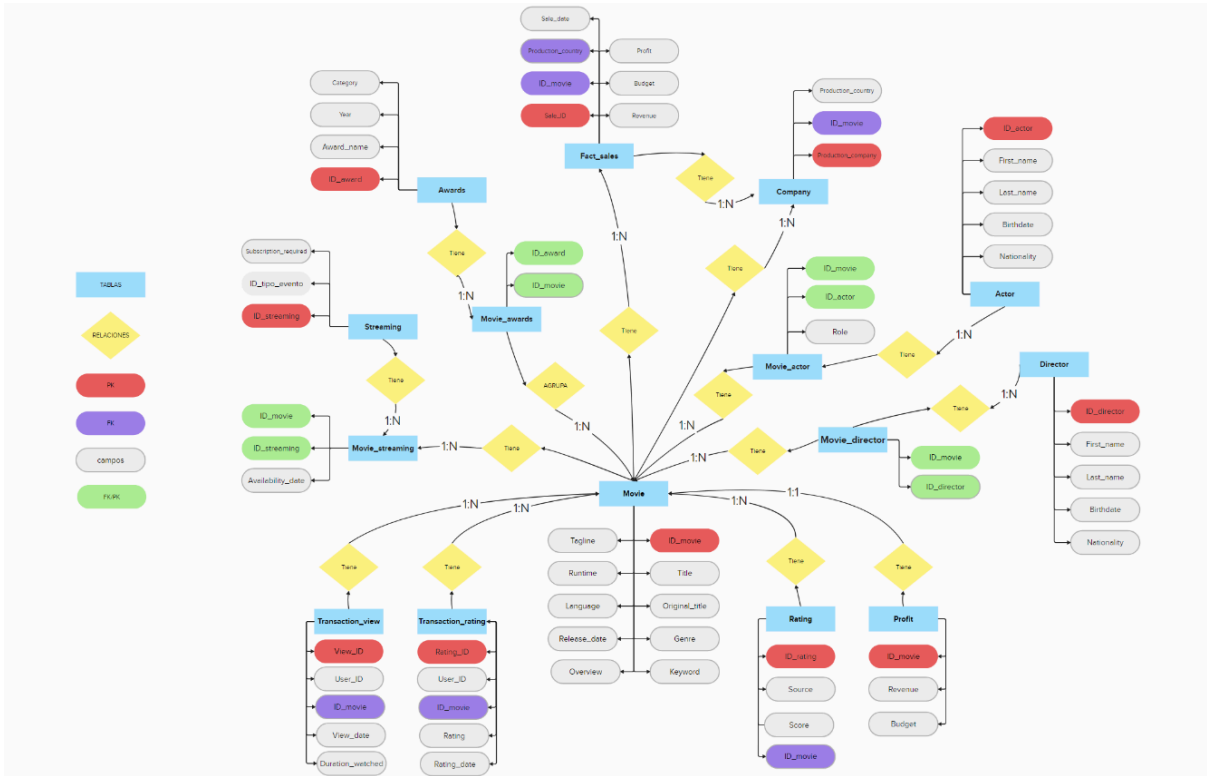
El creciente número de películas y su disponibilidad en múltiples plataformas digitales presenta un desafío para la gestión de grandes volúmenes de datos. En la actualidad, muchos estudios cinematográficos, plataformas de streaming y productores tienen dificultades para analizar y rastrear adecuadamente la rentabilidad de sus películas, la recepción del público y su distribución en plataformas globales. Este proyecto busca resolver esta problemática mediante un sistema centralizado que facilita la gestión y análisis de datos en la industria cinematográfica.

Modelo de negocio

Este modelo de negocio se apoya en la centralización y análisis de datos para mejorar la toma de decisiones en la industria cinematográfica, optimizar la

rentabilidad y aumentar la visibilidad de las películas a través de plataformas de distribución globales.

Diagrama entidad-relación de la base de datos



[Link DER en Mural](#)

Listado de columna de cada tabla

MOVIE		
Campo	Tipo de campo	Tipo de clave
ID_movie	int	PK
Title	varchar (100)	-
Original_title	varchar (100)	-
Genre	varchar (100)	-
Keyword	varchar (100)	-
Overview	text	-
Tagline	text	-
Runtime	time	-

Language	varchar (100)	-
Release_date	datetime	-

COMPANY		
Campo	Tipo de campo	Tipo de clave
ID_movie	int	FK
Production_company	varchar (100)	PK
Production_country	varchar (100)	-

PROFIT		
Campo	Tipo de campo	Tipo de clave
ID_movie	int	FK
Revenue	int	-
Budget	int	-

ACTOR		
Campo	Tipo de campo	Tipo de clave
ID_actor	int	PK
First_name	varchar (50)	-
Last_name	varchar (50)	-
Birthdate	date	-
Nationality	varchar (50)	-

DIRECTOR		
Campo	Tipo de campo	Tipo de clave
ID_director	int	PK
First_name	varchar (50)	-
Last_name	varchar (50)	-
Birthdate	date	-
Nationality	varchar (50)	-

MOVIE ACTOR		
Campo	Tipo de campo	Tipo de clave
ID_movie	int	PK, FK
ID_actor	int	PK, FK
Role	varchar (100)	-

MOVIE DIRECTOR		
Campo	Tipo de campo	Tipo de clave
ID_movie	int	PK, FK
ID_director	int	PK, FK

USER ACCOUNT		
Campo	Tipo de campo	Tipo de clave
ID_user	int	PK
Username	varchar (50)	Unique
Email	varchar (100)	Unique
Password	varchar (255)	-
Signup_date	timestamp	default current_timestamp

STREAMING		
Campo	Tipo de campo	Tipo de clave
ID_streaming	int	PK
Platform_name	varchar (100)	-
Subscription_required	boolean	-

MOVIE STREAMING		
Campo	Tipo de campo	Tipo de clave
ID_movie	int	PK, FK
ID_streaming	int	PK, FK
Availability_date	date	-

RATING		
Campo	Tipo de campo	Tipo de clave
ID_rating	int	PK
Source	varchar (100)	-
Score	decimal (3,1)	-
ID_movie	int	FK

AWARDS		
Campo	Tipo de campo	Tipo de clave
ID_award	int	PK
Award_name	varchar (100)	-
Year	int	-
Category	varchar (100)	-

MOVIE AWARDS		
Campo	Tipo de campo	Tipo de clave
ID_movie	int	PK, FK
ID_award	int	PK, FK

FACT SALES		
Campo	Tipo de campo	Tipo de clave
Sale_ID	int	PK
ID_movie	int	FK
Production_company	varchar (100)	FK
Sale_Date	date	-
Revenue	decimal (15,2)	-
Budget	decimal (15,2)	-
Profit	decimal (15,2)	Generado

TRANSACTION VIEW

Campo	Tipo de campo	Tipo de clave
View_ID	int	PK
User_ID	int	
ID_movie	int	FK
View_Date	datetime	-
Duration_watched	time	-

TRANSACTION RATING		
Campo	Tipo de campo	Tipo de clave
Rating_ID	int	PK
User_ID	int	
ID_movie	int	FK
Rating	decimal (2,1)	-
Rating_Date	date	-

Script de creación de cada objeto de la base de datos

TABLAS

```
-- Crear la base de datos
CREATE DATABASE IF NOT EXISTS BaseDeDatosPeliculas;

-- Usar la base de datos
USE BaseDeDatosPeliculas;

-- Crear la tabla 1 de MOVIE
CREATE TABLE MOVIE (
  ID_movie INT AUTO_INCREMENT PRIMARY KEY,
  Title VARCHAR(100),
  Original_title VARCHAR(100),
  Genre VARCHAR(100),
  Keyword VARCHAR(100),
  Overview TEXT,
  Tagline TEXT,
  Runtime TIME,
  Language VARCHAR(100),
  Release_date DATETIME
);
```


-- Crear la tabla 2 de COMPANY

```
CREATE TABLE COMPANY (  
    ID_movie INT,  
    Production_company VARCHAR(100),  
    Production_country VARCHAR(100),  
    PRIMARY KEY (Production_company),  
    FOREIGN KEY (ID_movie) REFERENCES MOVIE(ID_movie)  
);
```

-- Crear la tabla 3 de PROFIT

```
CREATE TABLE PROFIT (  
    ID_movie INT,  
    Revenue INT,  
    Budget INT,  
    FOREIGN KEY (ID_movie) REFERENCES MOVIE(ID_movie)  
);
```

-- Crear la tabla 4 de hechos

```
CREATE TABLE FACT_SALES (  
    Sale_ID INT AUTO_INCREMENT PRIMARY KEY,  
    ID_movie INT, -- FK referencing MOVIE  
    Production_company VARCHAR(100), -- FK referencing COMPANY  
    Sale_Date DATE,  
    Revenue DECIMAL(15,2),  
    Budget DECIMAL(15,2),  
    Profit DECIMAL(15,2) GENERATED ALWAYS AS (Revenue - Budget),  
    FOREIGN KEY (ID_movie) REFERENCES MOVIE(ID_movie),  
    FOREIGN KEY (Production_company) REFERENCES COMPANY(Production_company)  
);
```

-- Crear tabla 5 transaccional 1

```
CREATE TABLE TRANSACTION_VIEW (  
    View_ID INT AUTO_INCREMENT PRIMARY KEY,  
    User_ID INT, -- ID of the user who viewed the movie  
    ID_movie INT, -- FK referencing MOVIE  
    View_Date DATETIME,  
    Duration_Watched TIME, -- How long the movie was watched  
    FOREIGN KEY (ID_movie) REFERENCES MOVIE(ID_movie)  
);
```

-- Crear tabla 6 transaccional 2

```
CREATE TABLE TRANSACTION_RATING (  
    Rating_ID INT AUTO_INCREMENT PRIMARY KEY,  
    User_ID INT, -- ID of the user who rated the movie  
    ID_movie INT, -- FK referencing MOVIE  
    Rating DECIMAL(2,1), -- Rating given to the movie, e.g., 8.5  
    Rating_Date DATE,  
    FOREIGN KEY (ID_movie) REFERENCES MOVIE(ID_movie)
```

```
);
```

```
-- Crear tablas 7,8 principales
```

```
CREATE TABLE ACTOR (  
    ID_actor INT PRIMARY KEY AUTO_INCREMENT,  
    First_name VARCHAR(50) NOT NULL,  
    Last_name VARCHAR(50) NOT NULL,  
    Birthdate DATE,  
    Nationality VARCHAR(50)  
);
```

```
CREATE TABLE DIRECTOR (  
    ID_director INT PRIMARY KEY AUTO_INCREMENT,  
    First_name VARCHAR(50) NOT NULL,  
    Last_name VARCHAR(50) NOT NULL,  
    Birthdate DATE,  
    Nationality VARCHAR(50)  
);
```

```
-- Crear tablas 9, 10 intermedias para actores y directores
```

```
CREATE TABLE MOVIE_ACTOR (  
    ID_movie INT,  
    ID_actor INT,  
    Role VARCHAR(100),  
    PRIMARY KEY (ID_movie, ID_actor),  
    FOREIGN KEY (ID_movie) REFERENCES MOVIE(ID_movie),  
    FOREIGN KEY (ID_actor) REFERENCES ACTOR(ID_actor)  
);
```

```
CREATE TABLE MOVIE_DIRECTOR (  
    ID_movie INT,  
    ID_director INT,  
    PRIMARY KEY (ID_movie, ID_director),  
    FOREIGN KEY (ID_movie) REFERENCES MOVIE(ID_movie),  
    FOREIGN KEY (ID_director) REFERENCES DIRECTOR(ID_director)  
);
```

```
-- Crear tabla 11 de usuarios
```

```
CREATE TABLE USER_ACCOUNT (  
    ID_user INT PRIMARY KEY AUTO_INCREMENT,  
    Username VARCHAR(50) UNIQUE NOT NULL,  
    Email VARCHAR(100) UNIQUE NOT NULL,  
    Password VARCHAR(255) NOT NULL,  
    Signup_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
-- Crear tablas 12, 13 de streaming
```

```
CREATE TABLE STREAMING (  
    ID_streaming INT PRIMARY KEY AUTO_INCREMENT,  
    Name VARCHAR(50) NOT NULL,  
    Genre VARCHAR(50) NOT NULL,  
    Release_date DATE,  
    Duration INT,  
    Rating INT,  
    Description VARCHAR(255) NOT NULL  
);
```

```

    ID_streaming INT PRIMARY KEY AUTO_INCREMENT,
    Platform_name VARCHAR(100) NOT NULL,
    Subscription_required BOOLEAN DEFAULT TRUE
);

CREATE TABLE MOVIE_STREAMING (
    ID_movie INT,
    ID_streaming INT,
    Availability_date DATE,
    PRIMARY KEY (ID_movie, ID_streaming),
    FOREIGN KEY (ID_movie) REFERENCES MOVIE(ID_movie),
    FOREIGN KEY (ID_streaming) REFERENCES STREAMING(ID_streaming)
);

-- Crear tabla 14 de ratings
CREATE TABLE RATING (
    ID_rating INT PRIMARY KEY AUTO_INCREMENT,
    Source VARCHAR(100) NOT NULL,
    Score DECIMAL(3,1) NOT NULL,
    ID_movie INT,
    FOREIGN KEY (ID_movie) REFERENCES MOVIE(ID_movie)
);

-- Crear tablas 15, 16 de premios
CREATE TABLE AWARDS (
    ID_award INT PRIMARY KEY AUTO_INCREMENT,
    Award_name VARCHAR(100) NOT NULL,
    Year INT,
    Category VARCHAR(100)
);

CREATE TABLE MOVIE_AWARDS (
    ID_movie INT,
    ID_award INT,
    PRIMARY KEY (ID_movie, ID_award),
    FOREIGN KEY (ID_movie) REFERENCES MOVIE(ID_movie),
    FOREIGN KEY (ID_award) REFERENCES AWARDS(ID_award)
);

```

[Comentarios tablas:](#)

Realicé la creación de nuevas tablas para complementar las ya creadas en la entrega anterior, agregué las siguientes:

- Tabla de hecho - Fact_sales: para generar informes sobre los ingresos totales, rentabilidad por película o compañía productora, etc.
- Tablas transaccionales

- Transaction_view: almacena todas las visualizaciones de películas por parte de los usuarios para entender la popularidad.
- Transaction_rating: Almacena las calificaciones dadas por los usuarios a las películas para medir las recomendaciones.

Luego creé el resto de las tablas:

- Actor: para guardar información sobre los actores que participaron en cada película.
- Director: para guardar información sobre los directores.
 - Tablas intermedias: Movie_actor y movie_director para las relaciones muchos a muchos, ya que una película puede tener varios actores y un actor puede estar en varias películas por ejemplo.
- User: para registrar información de usuarios que califican o comentan sobre las películas.
- Streaming: para rastrear en qué plataformas de streaming están disponibles las películas.
 - Tabla intermedia: movie_streaming para manejar la relación de muchos a muchos entre películas y plataformas de streaming.
- Rating: para guardar diferentes calificaciones (ej, IMDb, Rotten Tomatoes). Hago la aclaración que eliminé la tabla "preference" creada en la entrega anterior, ya que me gustó más esta.
- Awards: para almacenar premios ganados por las películas.
 - Tabla intermedia: movie_awards para manejar la relación de muchos a muchos entre películas y premios.

VISTAS

-- Vista 1 rentabilidad

```
CREATE VIEW V_MOST_PROFITABLE_MOVIES AS
SELECT M.Title, M.Release_date, FS.Revenue, FS.Budget, FS.Profit
FROM MOVIE M
JOIN FACT_SALES FS ON M.ID_movie = FS.ID_movie
ORDER BY FS.Profit DESC;
```

-- Vista 2 popularidad streaming

```
CREATE VIEW V_POPULAR_MOVIES_ON_STREAMING AS
SELECT M.Title, S.Platform_name, AVG(R.Score) AS Average_Rating
FROM MOVIE M
JOIN MOVIE_STREAMING MS ON M.ID_movie = MS.ID_movie
JOIN STREAMING S ON MS.ID_streaming = S.ID_streaming
JOIN RATING R ON M.ID_movie = R.ID_movie
GROUP BY M.Title, S.Platform_name
ORDER BY Average_Rating DESC;
```

```
-- Vista 3 elenco
CREATE VIEW V_MOVIE_CREW AS
SELECT M.Title, CONCAT(A.First_name, ' ', A.Last_name) AS Actor,
CONCAT(D.First_name, ' ', D.Last_name) AS Director
FROM MOVIE M
LEFT JOIN MOVIE_ACTOR MA ON M.ID_movie = MA.ID_movie
LEFT JOIN ACTOR A ON MA.ID_actor = A.ID_actor
LEFT JOIN MOVIE_DIRECTOR MD ON M.ID_movie = MD.ID_movie
LEFT JOIN DIRECTOR D ON MD.ID_director = D.ID_director;
```

```
-- Vista 4 premios
CREATE VIEW V_AWARD_WINNING_MOVIES AS
SELECT M.Title, A.Award_name, A.Year, A.Category
FROM MOVIE M
JOIN MOVIE_AWARDS MA ON M.ID_movie = MA.ID_movie
JOIN AWARDS A ON MA.ID_award = A.ID_award
ORDER BY A.Year DESC;
```

```
-- Vista 5 usuarios
CREATE VIEW V_USER_ACTIVITY AS
SELECT UA.Username, M.Title, TR.Rating, TR.Rating_Date, TV.View_Date,
TV.Duration_Watched
FROM USER_ACCOUNT UA
JOIN TRANSACTION_RATING TR ON UA.ID_user = TR.User_ID
JOIN MOVIE M ON TR.ID_movie = M.ID_movie
LEFT JOIN TRANSACTION_VIEW TV ON UA.ID_user = TV.User_ID AND M.ID_movie =
TV.ID_movie
ORDER BY TR.Rating_Date DESC;
```

Comentarios Vistas:

V_MOST_PROFITABLE_MOVIES: Muestra las películas más rentables, calculando el beneficio de cada una y ordenando por el más alto.

- Tablas utilizadas:
 - MOVIE: Para obtener el título de la película (Title) y la fecha de lanzamiento (Release_date).
 - FACT_SALES: Para obtener los datos de ingresos (Revenue), presupuesto (Budget), y la rentabilidad (Profit).

Result Grid					
Filter Rows:					
Export:					
Wrap Cell Content:					
	Title	Release_date	Revenue	Budget	Profit
▶	The Last Kingdom	2022-01-01 00:00:00	1000000000.00	2200000000.00	7800000000.00
	The Forgotten War	2020-09-10 00:00:00	9200000000.00	1500000000.00	7700000000.00
	The Haunted House	2019-10-31 00:00:00	9800000000.00	2100000000.00	7700000000.00
	Echoes of the Past	2019-05-03 00:00:00	9200000000.00	1500000000.00	7700000000.00
	The Great Adventure	2020-03-15 00:00:00	9500000000.00	2000000000.00	7500000000.00
	Chasing the Dream	2019-12-25 00:00:00	8800000000.00	1400000000.00	7400000000.00
	City of Lights	2022-02-14 00:00:00	9300000000.00	2100000000.00	7200000000.00
	Underworld	2021-11-11 00:00:00	8600000000.00	1800000000.00	6800000000.00
	Mystery in the Fog	2019-06-22 00:00:00	8500000000.00	1800000000.00	6700000000.00
	Beneath the Waves	2021-06-30 00:00:00	8500000000.00	2000000000.00	6500000000.00

V_POPULAR_MOVIES_ON_STREAMING: Proporciona una lista de películas en plataformas de streaming, mostrando su popularidad basada en el promedio de ratings.

- Tablas utilizadas:
 - MOVIE: Para obtener el título de la película (Title).
 - MOVIE_STREAMING: Para vincular las películas con las plataformas de streaming (ID_movie y ID_streaming).
 - STREAMING: Para obtener el nombre de la plataforma de streaming (Platform_name).
 - RATING: Para calcular el promedio de ratings (Score) de cada película.

Result Grid			
Filter Rows:			
Export:			
Wrap Cell Content:			
	Title	Platform_name	Average_Rating
▶	The Infinite Chase	FandangoNOW	67.43333
	Shadows in the Mist	Acorn TV	66.73333
	The Rise of AI	Pluto TV	66.00000
	The Haunted House	Hulu	65.70000
	Through the Fire	The Criterion Channel	63.93333
	Echoes of the Past	Sling TV	63.23333
	The Final Countdown	AMC+	63.20000
	The Last Kingdom	Paramount+	63.16667
	Love Beyond Time	Amazon Prime Video	62.10000
	To the Name of Honor	Kanopy	62.03333

V_MOVIE_CREW: Esta vista combina la información de actores y directores, proporcionando una visión general del equipo detrás de cada película.

- Tablas utilizadas:
 - MOVIE: Para obtener el título de la película (Title).
 - MOVIE_ACTOR: Para vincular las películas con los actores (ID_movie y ID_actor).
 - ACTOR: Para obtener los nombres y apellidos de los actores (First_name, Last_name).
 - MOVIE_DIRECTOR: Para vincular las películas con los directores (ID_movie y ID_director).
 - DIRECTOR: Para obtener los nombres y apellidos de los directores (First_name, Last_name).

Result Grid			
Filter Rows:		Export:	Wrap Cell Content:
	Title	Actor	Director
▶	The Great Adventure	John Doe	Steven Spielberg
	Mystery in the Fog	Emily Blunt	Christopher Nolan
	Love Beyond Time	Chris Hemsworth	Quentin Tarantino
	The Forgotten War	Scarlett Johansson	James Cameron
	Into the Stars	Tom Hiddleston	Martin Scorsese
	The Haunted House	Zendaya Coleman	David Fincher
	The Big Heist	Dwayne Johnson	Peter Jackson
	The Island	Gal Gadot	Ridley Scott
	The Last Kingdom	Robert Downey Jr.	Denis Villeneuve
	The Hidden Truth	Ewan Reynolds	Guillermo del Toro

V_MOVIE_CREW 37 x Read Only

V_AWARD_WINNING_MOVIES: Muestra una lista de películas que han ganado premios, detallando el nombre del premio y el año.

- Tablas utilizadas:
 - MOVIE: Para obtener el título de la película (Title).
 - MOVIE_AWARDS: Para vincular las películas con los premios (ID_movie y ID_award).
 - AWARDS: Para obtener el nombre del premio (Award_name), el año (Year), y la categoría (Category).

Result Grid				
Filter Rows:		Export:	Wrap Cell Content:	
	Title	Award_name	Year	Category
▶	The Hidden Truth	Academy Award	2024	Best Picture
	Chasing the Dream	Golden Globe	2024	Best Director
	Beneath the Waves	BAFTA Award	2024	Best Actor
	City of Lights	Academy Award	2024	Best Actress
	Breaking the Code	Golden Globe	2024	Best Screenplay
	The Rise of AI	BAFTA Award	2024	Best Cinematography
	Frozen Heart	Academy Award	2024	Best Supporting Actor
	Echoes of the Past	Golden Globe	2024	Best Supporting Actress
	On the Edge	BAFTA Award	2024	Best Visual Effects
	Underworld	Cannes Film Festival	2024	Palme d'Or

V_AWARD_WINNING_MOVIES 38 x Read Only

V_USER_ACTIVITY: Esta vista proporciona una visión detallada de la actividad de los usuarios en la plataforma, incluyendo las calificaciones y la duración de visualización.

- Tablas utilizadas:
 - USER_ACCOUNT: Para obtener el nombre de usuario (Username).
 - TRANSACTION_RATING: Para obtener la calificación dada por los usuarios a las películas (Rating), y la fecha de la calificación (Rating_Date).
 - MOVIE: Para obtener el título de la película (Title).
 - TRANSACTION_VIEW: Para obtener la fecha en que se vio la película (View_Date) y la duración que se visualizó (Duration_Watched).

Result Grid						
Filter Rows:						
Export:						
Wrap Cell Content:						
	Username	Title	Rating	Rating_Date	View_Date	Duration_Watched
▶	olivia_johnson	Shadows in the Mist	8.9	2024-08-09	2023-10-16 00:00:00	01:40:00
	chris_evans	Underworld	7.5	2024-07-26	NULL	NULL
	laura_james	On the Edge	8.7	2024-06-13	NULL	NULL
	david_dark	Echoes of the Past	8.5	2024-05-23	NULL	NULL
	sarah_connor	Frozen Heart	7.9	2024-04-06	NULL	NULL
	robert_brown	The Rise of AI	9.0	2024-03-19	NULL	NULL
	emily_watson	Breaking the Code	8.6	2024-02-11	NULL	NULL
	michael_jordan	City of Lights	8.1	2024-01-16	NULL	NULL
	jane_smith	Beneath the Waves	8.8	2023-12-26	NULL	NULL
	john_doe	Chasing the Dream	7.7	2023-11-13	NULL	NULL

STORED PROCEDURES

-- SP 1 Agregar nueva película e ingresos
DELIMITER //

```

CREATE PROCEDURE AddMovieAndProfit(
    IN p_Title VARCHAR(100),
    IN p_Original_title VARCHAR(100),
    IN p_Genre VARCHAR(100),
    IN p_Keyword VARCHAR(100),
    IN p_Overview TEXT,
    IN p_Tagline TEXT,
    IN p_Runtime TIME,
    IN p_Language VARCHAR(100),
    IN p_Release_date DATETIME,
    IN p_Revenue INT,
    IN p_Budget INT
)
BEGIN
    DECLARE new_movie_id INT;

    -- Insertar la nueva película en la tabla MOVIE
    INSERT INTO MOVIE (Title, Original_title, Genre, Keyword, Overview, Tagline, Runtime,
    Language, Release_date)
    VALUES (p_Title, p_Original_title, p_Genre, p_Keyword, p_Overview, p_Tagline,
    p_Runtime, p_Language, p_Release_date);

    -- Obtener el ID de la película recién insertada
    SET new_movie_id = LAST_INSERT_ID();

    -- Insertar los datos de ingresos y presupuesto en la tabla PROFIT
    INSERT INTO PROFIT (ID_movie, Revenue, Budget)
    VALUES (new_movie_id, p_Revenue, p_Budget);
END //

```


DELIMITER ;

```
CREATE PROCEDURE UpdateMovieRevenue(  
    IN p_ID_movie INT,  
    IN p_NewRevenue DECIMAL(15,2),  
    IN p_NewBudget DECIMAL(15,2)  
)  
BEGIN  
    -- Actualiza los ingresos y presupuesto de la película en la tabla PROFIT  
    UPDATE PROFIT  
    SET Revenue = p_NewRevenue,  
        Budget = p_NewBudget  
    WHERE ID_movie = p_ID_movie;  
  
    -- No es necesario actualizar Profit ya que se calcula automáticamente  
END //
```

DELIMITER ;

Comentarios SP:

Creé los siguientes Stored Procedure

- AddMovieAndProfit: Este procedimiento tiene como objetivo automatizar el proceso de agregar una nueva película a la base de datos y, al mismo tiempo, registrar sus ingresos y presupuesto en las tablas correspondientes. Permite insertar registros relacionados en varias tablas (MOVIE y PROFIT) en una sola operación, lo que minimiza errores y asegura consistencia.
 - Tablas utilizadas:
 - MOVIE: Se usa para almacenar la nueva película.
 - PROFIT: Se usa para almacenar los ingresos y el presupuesto asociado a la película recién creada.
- UpdateMovieRevenue: Tiene como objetivo actualizar los ingresos y el presupuesto de una película específica en la base de datos.
 - Tablas utilizadas:
 - PROFIT: Se actualiza para reflejar el nuevo ingreso de la película.
 - FACT_SALES: La columna Profit se calcula automáticamente como Revenue - Budget, por lo que si los ingresos o el presupuesto se actualizan, el beneficio se ajustará automáticamente en esta tabla, dado que Profit es una columna generada.

FUNCIONES

-- FUNCION 1 Promedio rating
DELIMITER //

```
CREATE FUNCTION CalculateAverageRating(p_ID_movie INT)
```

```

RETURNS DECIMAL(3,1)
DETERMINISTIC
BEGIN
    DECLARE avg_rating DECIMAL(3,1);

    -- Calcular el promedio de calificaciones para la película especificada
    SELECT AVG(Score) INTO avg_rating
    FROM RATING
    WHERE ID_movie = p_ID_movie;

    RETURN avg_rating;
END //

DELIMITER ;

-- FUNCION 2 Margen de ganancia
DELIMITER //

CREATE FUNCTION GetMovieProfitMargin(p_ID_movie INT)
RETURNS DECIMAL(5,2)
DETERMINISTIC
BEGIN
    DECLARE profit_margin DECIMAL(5,2);
    DECLARE movie_revenue INT;
    DECLARE movie_budget INT;

    -- Obtener el ingreso y presupuesto de la película
    SELECT Revenue, Budget INTO movie_revenue, movie_budget
    FROM PROFIT
    WHERE ID_movie = p_ID_movie;

    -- Calcular el margen de ganancia
    IF movie_revenue > 0 THEN
        SET profit_margin = (movie_revenue - movie_budget) / movie_revenue * 100;
    ELSE
        SET profit_margin = 0;
    END IF;

    RETURN profit_margin;
END //

DELIMITER ;

```

Comentarios funciones:

Cree las siguientes funciones:

- CalculateAverageRating: Esta función calcula el promedio de las calificaciones (ratings) para una película específica en base a las entradas en la tabla RATING. El objetivo es proporcionar una forma fácil de obtener el promedio de calificaciones de una película, lo cual es útil para reportes, recomendaciones o para mostrar en la interfaz de usuario.
 - Tablas utilizadas: RATING.
- GetMovieProfitMargin: Esta función calcula el margen de ganancia (profit margin) de una película específica en base a los datos de ingresos (revenue) y presupuesto (budget) almacenados en la tabla PROFIT. El objetivo es determinar qué tan rentable es una película en términos de margen de ganancia, lo cual es esencial para el análisis financiero y la toma de decisiones comerciales.
 - Tablas utilizadas: PROFIT.

TRIGGERS

-- Trigger 1 duración película
DELIMITER //

```
CREATE TRIGGER before_insert_transaction_view
BEFORE INSERT ON TRANSACTION_VIEW
FOR EACH ROW
BEGIN
```

```
    DECLARE movie_runtime TIME;
```

```
    -- Obtener la duración total de la película desde la tabla MOVIE
```

```
    SELECT Runtime INTO movie_runtime
```

```
    FROM MOVIE
```

```
    WHERE ID_movie = NEW.ID_movie;
```

```
    -- Validar que la duración de visualización no sea mayor que la duración total de la
    película
```

```
    IF NEW.Duration_Watched > movie_runtime THEN
```

```
        SIGNAL SQLSTATE '45000'
```

```
        SET MESSAGE_TEXT = 'Error: La duración de visualización no puede ser mayor que
    la duración total de la película.';
```

```
    END IF;
```

```
END //
```

```
DELIMITER ;
```

-- trigger 2 eliminar pelicula

DELIMITER //

```
CREATE TRIGGER before_delete_movie
BEFORE DELETE ON MOVIE
FOR EACH ROW
BEGIN
```

```

DECLARE associated_records INT;

-- Verificar si existen registros asociados en MOVIE_ACTOR
SELECT COUNT(*) INTO associated_records
FROM MOVIE_ACTOR
WHERE ID_movie = OLD.ID_movie;

IF associated_records > 0 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Error: No se puede eliminar la película porque tiene actores
asociados.';
END IF;

-- Verificar si existen registros asociados en MOVIE_DIRECTOR
SELECT COUNT(*) INTO associated_records
FROM MOVIE_DIRECTOR
WHERE ID_movie = OLD.ID_movie;

IF associated_records > 0 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Error: No se puede eliminar la película porque tiene
directores asociados.';
END IF;

-- Verificar si existen registros asociados en FACT_SALES
SELECT COUNT(*) INTO associated_records
FROM FACT_SALES
WHERE ID_movie = OLD.ID_movie;

IF associated_records > 0 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Error: No se puede eliminar la película porque tiene datos de
ventas asociados.';
END IF;

END //

DELIMITER ;

```

[Comentarios triggers:](#)

Creé los siguientes triggers:

- [before_insert_transaction_view](#): Este trigger se activa antes de que se inserte un nuevo registro en la tabla TRANSACTION_VIEW. Su función es validar que la duración de visualización (Duration_Watched) no sea mayor que la duración total de la película (Runtime) para evitar inconsistencias en los datos.
 - Tablas utilizadas:

- TRANSACTION_VIEW: El trigger opera sobre esta tabla, validando los datos antes de la inserción.
 - MOVIE: Se utiliza para obtener la duración total de la película.
- before_delete_movie: Este trigger se activa antes de que se elimine un registro de la tabla MOVIE. Su objetivo es prevenir la eliminación de películas que ya están asociadas a registros importantes en otras tablas, como MOVIE_ACTOR, MOVIE_DIRECTOR, o FACT_SALES, asegurando la integridad referencial de la base de datos.
 - Tablas utilizadas:
 - MOVIE: El trigger se aplica antes de que se elimine un registro de esta tabla.
 - MOVIE_ACTOR: Verifica si la película tiene actores asociados.
 - MOVIE_DIRECTOR: Verifica si la película tiene directores asociados.
 - FACT_SALES: Verifica si la película tiene datos de ventas asociados.

Script en SQL de inserción de datos

-- INSERCIÓN DE DATOS

-- TABLA MOVIE

INSERT INTO MOVIE (Title, Original_title, Genre, Keyword, Overview, Tagline, Runtime, Language, Release_date)

VALUES

('The Great Adventure', 'La Gran Aventura', 'Action', 'Adventure', 'A man sets out on a dangerous journey.', 'The ultimate adventure awaits.', '02:15:00', 'English', '2020-03-15'),
 ('Mystery in the Fog', 'Misterio en la Niebla', 'Mystery', 'Detective', 'A detective unravels a complex case.', 'Nothing is as it seems.', '01:45:00', 'Spanish', '2019-06-22'),
 ('Love Beyond Time', 'Amor Más Allá del Tiempo', 'Romance', 'Time Travel', 'Two lovers find each other through time.', 'Love knows no boundaries.', '02:05:00', 'English', '2021-02-14'),
 ('The Forgotten War', 'La Guerra Olvidada', 'History', 'War', 'A soldier recounts a forgotten battle.', 'A battle that should be remembered.', '02:30:00', 'French', '2020-09-10'),
 ('Into the Stars', 'Hacia las Estrellas', 'Science Fiction', 'Space', 'A team of astronauts embarks on a space mission.', 'The stars are calling.', '02:20:00', 'English', '2022-07-05'),
 ('The Haunted House', 'La Casa Embrujada', 'Horror', 'Ghost', 'A family moves into a haunted house.', 'Beware of the spirits.', '01:50:00', 'Spanish', '2019-10-31'),
 ('The Big Heist', 'El Gran Robo', 'Crime', 'Robbery', 'A group plans the ultimate heist.', 'The score of a lifetime.', '02:10:00', 'English', '2021-05-20'),
 ('The Island', 'La Isla', 'Thriller', 'Survival', 'A group stranded on a mysterious island.', 'Survive at any cost.', '02:00:00', 'French', '2019-11-11'),
 ('The Last Kingdom', 'El Último Reino', 'Drama', 'Royalty', 'A young king struggles to hold his kingdom.', 'Rule or be ruled.', '02:25:00', 'English', '2022-01-01'),
 ('The Hidden Truth', 'La Verdad Oculta', 'Documentary', 'Conspiracy', 'An investigation into hidden truths.', 'What they don't want you to know.', '01:55:00', 'English', '2020-04-18'),
 ('Chasing the Dream', 'Persiguiendo el Sueño', 'Sports', 'Football', 'An underdog team fights to win the championship.', 'It's more than just a game.', '02:05:00', 'English', '2019-12-25'),

('Beneath the Waves', 'Debajo de las Olas', 'Adventure', 'Ocean', 'A diver discovers hidden secrets under the sea.', 'The ocean's mysteries await.', '02:20:00', 'English', '2021-06-30'),
 ('City of Lights', 'Ciudad de Luces', 'Romance', 'Paris', 'Two strangers meet in Paris and fall in love.', 'Love is always brighter in the city of lights.', '01:50:00', 'French', '2022-02-14'),
 ('Breaking the Code', 'Rompiendo el Código', 'Thriller', 'Hacker', 'A hacker uncovers a global conspiracy.', 'The truth is encrypted.', '01:55:00', 'English', '2020-07-21'),
 ('The Rise of AI', 'El Ascenso de la IA', 'Science Fiction', 'Artificial Intelligence', 'A world controlled by artificial intelligence.', 'Can humanity survive?', '02:25:00', 'English', '2021-09-15'),
 ('Frozen Heart', 'Corazón Congelado', 'Romance', 'Winter', 'A couple finds love during a harsh winter.', 'Love can melt even the coldest heart.', '01:45:00', 'English', '2020-12-10'),
 ('Echoes of the Past', 'Ecos del Pasado', 'Drama', 'Family', 'A woman returns to her childhood home to confront her past.', 'Some memories never fade.', '02:10:00', 'Spanish', '2019-05-03'),
 ('On the Edge', 'Al Límite', 'Action', 'Extreme Sports', 'An extreme sports athlete pushes the limits.', 'Only the fearless survive.', '01:55:00', 'English', '2022-03-25'),
 ('Underworld', 'Inframundo', 'Fantasy', 'Mythology', 'A warrior ventures into the underworld to save a loved one.', 'The gods have other plans.', '02:20:00', 'Greek', '2021-11-11'),
 ('Shadows in the Mist', 'Sombras en la Niebla', 'Horror', 'Supernatural', 'A town plagued by mysterious supernatural events.', 'Beware what lurks in the mist.', '01:50:00', 'Spanish', '2019-09-30'),
 ('The Quantum Leap', 'El Salto Cuántico', 'Science Fiction', 'Time Travel', 'Scientists discover the key to time travel.', 'Time is not what you think.', '02:15:00', 'English', '2020-10-05'),
 ('In the Name of Honor', 'En Nombre del Honor', 'War', 'Soldiers', 'A soldier fights to protect his country.', 'For honor, for glory.', '02:20:00', 'English', '2022-05-17'),
 ('The Secret Ingredient', 'El Ingrediente Secreto', 'Comedy', 'Food', 'A chef discovers the secret ingredient to success.', 'It's all in the recipe.', '01:40:00', 'French', '2020-06-10'),
 ('Through the Fire', 'A Través del Fuego', 'Drama', 'Firefighters', 'A firefighter's struggle to save lives.', 'Bravery in the face of danger.', '02:05:00', 'English', '2019-08-14'),
 ('Whispers in the Dark', 'Susurros en la Oscuridad', 'Mystery', 'Detective', 'A detective investigates a series of mysterious murders.', 'The truth hides in the shadows.', '02:00:00', 'Spanish', '2021-01-20'),
 ('Dawn of the Empire', 'El Amanecer del Imperio', 'Historical', 'Empire', 'The rise and fall of an ancient empire.', 'Empires are not built in a day.', '02:35:00', 'Greek', '2020-02-22'),
 ('The Final Countdown', 'La Cuenta Final', 'Action', 'Bomb', 'A team must stop a deadly bomb from detonating.', 'The clock is ticking.', '01:55:00', 'English', '2021-12-24'),
 ('A New Beginning', 'Un Nuevo Comienzo', 'Drama', 'Rebirth', 'A woman rebuilds her life after a tragedy.', 'Every ending is a new beginning.', '02:10:00', 'Spanish', '2021-07-18'),
 ('The Infinite Chase', 'La Persecución Infinita', 'Action', 'Chase', 'A cop chases down a dangerous fugitive.', 'The chase never ends.', '02:05:00', 'English', '2022-09-09'),
 ('The Silent Echo', 'El Eco Silencioso', 'Horror', 'Ghost', 'A woman haunted by the echoes of her past.', 'Some voices never leave.', '01:50:00', 'French', '2021-03-11');

-- TABLA COMPANY

INSERT INTO COMPANY (ID_movie, Production_company, Production_country)

VALUES

(1, 'Adventure Studios', 'USA'),

(2, 'Mystery Films', 'Spain'),

(3, 'Timeless Productions', 'USA'),
 (4, 'Historical Films', 'France'),
 (5, 'Cosmic Studios', 'USA'),
 (6, 'Horror House', 'Spain'),
 (7, 'Heist Productions', 'USA'),
 (8, 'Survival Studios', 'France'),
 (9, 'Royal Films', 'UK'),
 (10, 'Truth Seekers', 'USA'),
 (11, 'Dream Chasers', 'USA'),
 (12, 'Ocean Deep', 'USA'),
 (13, 'Paris Films', 'France'),
 (14, 'Hacker Studios', 'USA'),
 (15, 'AI World', 'USA'),
 (16, 'Winter Productions', 'Canada'),
 (17, 'Memory Films', 'Spain'),
 (18, 'Extreme Sports Studios', 'USA'),
 (19, 'Mythical Studios', 'Greece'),
 (20, 'Supernatural Films', 'Spain'),
 (21, 'Time Travel Studios', 'USA'),
 (22, 'Honor Films', 'USA'),
 (23, 'Comedy House', 'France'),
 (24, 'Bravery Films', 'USA'),
 (25, 'Detective Studios', 'Spain'),
 (26, 'Empire Films', 'Greece'),
 (27, 'Bomb Squad', 'USA'),
 (28, 'Rebirth Films', 'Spain'),
 (29, 'Chase Studios', 'USA'),
 (30, 'Echo Films', 'France');

-- TABLA PROFIT

INSERT INTO PROFIT (ID_movie, Revenue, Budget) VALUES

(1, 150000000, 70000000),
 (2, 95000000, 55000000),
 (3, 35000000, 15000000),
 (4, 25000000, 12000000),
 (5, 80000000, 60000000),
 (6, 22000000, 9000000),
 (7, 180000000, 95000000),
 (8, 45000000, 28000000),
 (9, 140000000, 70000000),
 (10, 65000000, 45000000),
 (11, 105000000, 50000000),
 (12, 76000000, 40000000),
 (13, 132000000, 80000000),
 (14, 30000000, 15000000),
 (15, 94000000, 62000000),
 (16, 24000000, 10000000),
 (17, 170000000, 110000000),

```
(18, 52000000, 30000000),
(19, 125000000, 85000000),
(20, 99000000, 60000000),
(21, 81000000, 51000000),
(22, 60000000, 30000000),
(23, 45000000, 25000000),
(24, 20000000, 10000000),
(25, 92000000, 50000000),
(26, 78000000, 45000000),
(27, 142000000, 70000000),
(28, 50000000, 20000000),
(29, 108000000, 60000000),
(30, 88000000, 55000000);
```

-- TABLA ACTOR

INSERT INTO ACTOR (ID_actor, First_name, Last_name, Birthdate, Nationality) VALUES

```
(1, 'John', 'Doe', '1980-05-14', 'USA'),
(2, 'Emily', 'Blunt', '1983-02-23', 'UK'),
(3, 'Chris', 'Hemsworth', '1983-08-11', 'Australia'),
(4, 'Scarlett', 'Johansson', '1984-11-22', 'USA'),
(5, 'Tom', 'Hiddleston', '1981-02-09', 'UK'),
(6, 'Zendaya', 'Coleman', '1996-09-01', 'USA'),
(7, 'Dwayne', 'Johnson', '1972-05-02', 'USA'),
(8, 'Gal', 'Gadot', '1985-04-30', 'Israel'),
(9, 'Robert', 'Downey Jr.', '1965-04-04', 'USA'),
(10, 'Ryan', 'Reynolds', '1976-10-23', 'Canada'),
(11, 'Hugh', 'Jackman', '1968-10-12', 'Australia'),
(12, 'Anne', 'Hathaway', '1982-11-12', 'USA'),
(13, 'Mark', 'Ruffalo', '1967-11-22', 'USA'),
(14, 'Natalie', 'Portman', '1981-06-09', 'Israel'),
(15, 'Chris', 'Pratt', '1979-06-21', 'USA'),
(16, 'Samuel', 'Jackson', '1948-12-21', 'USA'),
(17, 'Vin', 'Diesel', '1967-07-18', 'USA'),
(18, 'Margot', 'Robbie', '1990-07-02', 'Australia'),
(19, 'Leonardo', 'DiCaprio', '1974-11-11', 'USA'),
(20, 'Will', 'Smith', '1968-09-25', 'USA'),
(21, 'Emma', 'Stone', '1988-11-06', 'USA'),
(22, 'Jennifer', 'Lawrence', '1990-08-15', 'USA'),
(23, 'Christian', 'Bale', '1974-01-30', 'UK'),
(24, 'Brad', 'Pitt', '1963-12-18', 'USA'),
(25, 'George', 'Clooney', '1961-05-06', 'USA'),
(26, 'Meryl', 'Streep', '1949-06-22', 'USA'),
(27, 'Sandra', 'Bullock', '1964-07-26', 'USA'),
(28, 'Halle', 'Berry', '1966-08-14', 'USA'),
(29, 'Tom', 'Cruise', '1962-07-03', 'USA'),
(30, 'Joaquin', 'Phoenix', '1974-10-28', 'USA');
```

-- TABLA DIRECTOR


```
INSERT INTO DIRECTOR (ID_director, First_name, Last_name, Birthdate, Nationality)
VALUES
```

```
(1, 'Steven', 'Spielberg', '1946-12-18', 'USA'),
(2, 'Christopher', 'Nolan', '1970-07-30', 'UK'),
(3, 'Quentin', 'Tarantino', '1963-03-27', 'USA'),
(4, 'James', 'Cameron', '1954-08-16', 'Canada'),
(5, 'Martin', 'Scorsese', '1942-11-17', 'USA'),
(6, 'David', 'Fincher', '1962-08-28', 'USA'),
(7, 'Peter', 'Jackson', '1961-10-31', 'New Zealand'),
(8, 'Ridley', 'Scott', '1937-11-30', 'UK'),
(9, 'Denis', 'Villeneuve', '1967-10-03', 'Canada'),
(10, 'Guillermo', 'del Toro', '1964-10-09', 'Mexico'),
(11, 'Patty', 'Jenkins', '1971-07-24', 'USA'),
(12, 'Wes', 'Anderson', '1969-05-01', 'USA'),
(13, 'Sofia', 'Coppola', '1971-05-14', 'USA'),
(14, 'Clint', 'Eastwood', '1930-05-31', 'USA'),
(15, 'James', 'Wan', '1977-02-26', 'Australia'),
(16, 'George', 'Miller', '1945-03-03', 'Australia'),
(17, 'Taika', 'Waititi', '1975-08-16', 'New Zealand'),
(18, 'Greta', 'Gerwig', '1983-08-04', 'USA'),
(19, 'Spike', 'Lee', '1957-03-20', 'USA'),
(20, 'Alejandro', 'González Iñárritu', '1963-08-15', 'Mexico'),
(21, 'Francis', 'Ford Coppola', '1939-04-07', 'USA'),
(22, 'Tim', 'Burton', '1958-08-25', 'USA'),
(23, 'Lana', 'Wachowski', '1965-06-21', 'USA'),
(24, 'Lilly', 'Wachowski', '1967-12-29', 'USA'),
(25, 'Michael', 'Bay', '1965-02-17', 'USA'),
(26, 'Damien', 'Chazelle', '1985-01-19', 'USA'),
(27, 'Paul', 'Thomas Anderson', '1970-06-26', 'USA'),
(28, 'Edgar', 'Wright', '1974-04-18', 'UK'),
(29, 'Ang', 'Lee', '1954-10-23', 'Taiwan'),
(30, 'Bong', 'Joon-ho', '1969-09-14', 'South Korea');
```

-- TABLAS INTERMEDIAS ACTOR Y DIRECTOR

```
INSERT INTO MOVIE_ACTOR (ID_movie, ID_actor, Role) VALUES
```

```
(1, 1, 'John Connor'),
(2, 2, 'Ellen Ripley'),
(3, 3, 'Rick Dalton'),
(4, 4, 'Jake Sully'),
(5, 5, 'Henry Hill'),
(6, 6, 'Mark Zuckerberg'),
(7, 7, 'Frodo Baggins'),
(8, 8, 'Maximus'),
(9, 9, 'K'),
(10, 10, 'Elisa Esposito'),
(11, 11, 'Diana Prince'),
(12, 12, 'Mr. Fox'),
(13, 13, 'Charlotte'),
```

(14, 14, 'Walt Kowalski'),
(15, 15, 'Elise Rainier'),
(16, 16, 'Max Rockatansky'),
(17, 17, 'Korg'),
(18, 18, 'Jo March'),
(19, 19, 'Mookie'),
(20, 20, 'Ursúa'),
(21, 21, 'Vito Corleone'),
(22, 22, 'Edward Scissorhands'),
(23, 23, 'Neo'),
(24, 24, 'Agent Smith'),
(25, 25, 'Marcus Burnett'),
(26, 26, 'Sebastian'),
(27, 27, 'Reynolds Woodcock'),
(28, 28, 'Baby'),
(29, 29, 'Pi Patel'),
(30, 30, 'Kim Ki-taek');

INSERT INTO MOVIE_DIRECTOR (ID_movie, ID_director) VALUES

(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5),
(6, 6),
(7, 7),
(8, 8),
(9, 9),
(10, 10),
(11, 11),
(12, 12),
(13, 13),
(14, 14),
(15, 15),
(16, 16),
(17, 17),
(18, 18),
(19, 19),
(20, 20),
(21, 21),
(22, 22),
(23, 23),
(24, 24),
(25, 25),
(26, 26),
(27, 27),
(28, 28),
(29, 29),

(30, 30);

-- TABLA USUARIOS

```
INSERT INTO USER_ACCOUNT (Username, Email, Password, Signup_date) VALUES
('john_doe', 'john.doe@example.com', 'password123', '2023-01-05 10:34:56'),
('jane_smith', 'jane.smith@example.com', 'securepass456', '2023-02-10 15:28:33'),
('michael_jordan', 'michael.jordan@example.com', 'basketball7', '2023-03-12 18:45:22'),
('emily_watson', 'emily.watson@example.com', 'watsonrules89', '2023-04-08 12:12:12'),
('robert_brown', 'robert.brown@example.com', 'supersecret!88', '2023-05-14 09:34:11'),
('sarah_connor', 'sarah.connor@example.com', 'terminator82', '2023-06-22 14:56:23'),
('david_clark', 'david.clark@example.com', 'clarkkent2000', '2023-07-02 17:45:09'),
('laura_james', 'laura.james@example.com', 'jameslaura3$', '2023-08-01 16:38:56'),
('chris_evans', 'chris.evans@example.com', 'america$hero', '2023-09-05 11:24:10'),
('olivia_johnson', 'olivia.johnson@example.com', 'olivia$22', '2023-10-12 13:49:45'),
('jack_white', 'jack.white@example.com', 'whiterabbit123', '2023-11-03 08:12:34'),
('emma_brooks', 'emma.brooks@example.com', 'emmazing91', '2023-12-17 19:56:22'),
('ethan_cole', 'ethan.cole@example.com', 'coletrain1991', '2024-01-09 20:10:10'),
('natalie_adams', 'natalie.adams@example.com', 'adamsnatalie$55', '2024-02-16 10:25:37'),
('daniel_wright', 'daniel.wright@example.com', 'wrightway888', '2024-03-18 12:36:48'),
('julia_ross', 'julia.ross@example.com', 'rossandroll!', '2024-04-20 14:42:19'),
('thomas_green', 'thomas.green@example.com', 'greenthomas77', '2024-05-22 09:15:54'),
('sophia_lee', 'sophia.lee@example.com', 'leesophia', '2024-06-30 16:11:03'),
('henry_hughes', 'henry.hughes@example.com', 'hughesrocks789', '2024-07-13 18:25:45'),
('amelia_wright', 'amelia.wright@example.com', 'wrightstuff2021', '2024-08-09 20:10:44'),
('william_scott', 'william.scott@example.com', 'scotty2hottie', '2024-09-18 11:48:57'),
('charlotte_brown', 'charlotte.brown@example.com', 'charlottebrown2$', '2024-10-25
13:17:38'),
('jackson_davis', 'jackson.davis@example.com', 'davisjackson99', '2024-11-30 14:44:21'),
('madison_turner', 'madison.turner@example.com', 'turnmad2020', '2024-12-13 16:02:39'),
('logan_hill', 'logan.hill@example.com', 'hilltime321', '2024-12-25 18:37:58'),
('hannah_young', 'hannah.young@example.com', 'younggoddess', '2025-01-05 11:20:07'),
('liam_roberts', 'liam.roberts@example.com', 'liamwinning44', '2025-02-14 13:48:36'),
('zoe_thompson', 'zoe.thompson@example.com', 'zozo$thom', '2025-03-22 15:26:44'),
('noah_allen', 'noah.allen@example.com', 'allensavage!', '2025-04-30 17:12:05'),
('ella_bennett', 'ella.bennett@example.com', 'bennettstrong12', '2025-05-18 19:28:11');
```

-- TABLA STREAMING

```
INSERT INTO STREAMING (Platform_name, Subscription_required) VALUES
('Netflix', TRUE),
('Amazon Prime Video', TRUE),
('Disney+', TRUE),
('HBO Max', TRUE),
('Hulu', TRUE),
('Apple TV+', TRUE),
('Peacock', TRUE),
('Paramount+', TRUE),
('YouTube', FALSE),
('Google Play Movies', FALSE),
```

```

('Vudu', FALSE),
('Tubi', FALSE),
('Crackle', FALSE),
('Pluto TV', FALSE),
('FuboTV', TRUE),
('Sling TV', TRUE),
('Shudder', TRUE),
('CuriosityStream', TRUE),
('Acorn TV', TRUE),
('BritBox', TRUE),
('Kanopy', FALSE),
('Mubi', TRUE),
('The Criterion Channel', TRUE),
('Starz', TRUE),
('Showtime', TRUE),
('AMC+', TRUE),
('Discovery+', TRUE),
('FandangoNOW', FALSE),
('Rakuten TV', FALSE),
('Plex', FALSE);

```

-- TABLA INTERMEDIA STREAMING

INSERT INTO MOVIE_STREAMING (ID_movie, ID_streaming, Availability_date) VALUES

```

(1, 1, '2023-01-01'),
(2, 1, '2023-02-05'),
(3, 2, '2023-03-10'),
(4, 3, '2023-04-15'),
(5, 4, '2023-05-20'),
(6, 5, '2023-06-25'),
(7, 6, '2023-07-30'),
(8, 7, '2023-08-10'),
(9, 8, '2023-09-15'),
(10, 9, '2023-10-20'),
(11, 10, '2023-11-25'),
(12, 11, '2023-12-30'),
(13, 12, '2024-01-05'),
(14, 13, '2024-02-10'),
(15, 14, '2024-03-15'),
(16, 15, '2024-04-20'),
(17, 16, '2024-05-25'),
(18, 17, '2024-06-30'),
(19, 18, '2024-07-10'),
(20, 19, '2024-08-15'),
(21, 20, '2024-09-20'),
(22, 21, '2024-10-25'),
(23, 22, '2024-11-30'),
(24, 23, '2024-12-05'),
(25, 24, '2025-01-10'),

```

(26, 25, '2025-02-15'),
(27, 26, '2025-03-20'),
(28, 27, '2025-04-25'),
(29, 28, '2025-05-30'),
(30, 29, '2025-06-10');

-- TABLA DE RATING

INSERT INTO RATING (Source, Score, ID_movie) VALUES

('IMDb', 7.8, 1),
(('Rotten Tomatoes', 85.0, 1),
(('Metacritic', 70.0, 1),
(('IMDb', 6.9, 2),
(('Rotten Tomatoes', 72.0, 2),
(('Metacritic', 65.0, 2),
(('IMDb', 8.3, 3),
(('Rotten Tomatoes', 90.0, 3),
(('Metacritic', 88.0, 3),
(('IMDb', 7.2, 4),
(('Rotten Tomatoes', 80.0, 4),
(('Metacritic', 75.0, 4),
(('IMDb', 6.5, 5),
(('Rotten Tomatoes', 68.0, 5),
(('Metacritic', 60.0, 5),
(('IMDb', 9.1, 6),
(('Rotten Tomatoes', 95.0, 6),
(('Metacritic', 93.0, 6),
(('IMDb', 7.9, 7),
(('Rotten Tomatoes', 85.0, 7),
(('Metacritic', 80.0, 7),
(('IMDb', 6.3, 8),
(('Rotten Tomatoes', 60.0, 8),
(('Metacritic', 58.0, 8),
(('IMDb', 8.5, 9),
(('Rotten Tomatoes', 92.0, 9),
(('Metacritic', 89.0, 9),
(('IMDb', 7.0, 10),
(('Rotten Tomatoes', 75.0, 10),
(('Metacritic', 70.0, 10),
(('IMDb', 8.2, 11),
(('Rotten Tomatoes', 88.0, 11),
(('Metacritic', 85.0, 11),
(('IMDb', 5.9, 12),
(('Rotten Tomatoes', 55.0, 12),
(('Metacritic', 52.0, 12),
(('IMDb', 7.6, 13),
(('Rotten Tomatoes', 80.0, 13),
(('Metacritic', 78.0, 13),
(('IMDb', 6.8, 14),

('Rotten Tomatoes', 70.0, 14),
('Metacritic', 65.0, 14),
('IMDb', 9.0, 15),
('Rotten Tomatoes', 95.0, 15),
('Metacritic', 94.0, 15),
('IMDb', 7.5, 16),
('Rotten Tomatoes', 82.0, 16),
('Metacritic', 78.0, 16),
('IMDb', 8.7, 17),
('Rotten Tomatoes', 91.0, 17),
('Metacritic', 90.0, 17),
('IMDb', 6.2, 18),
('Rotten Tomatoes', 60.0, 18),
('Metacritic', 55.0, 18),
('IMDb', 7.8, 19),
('Rotten Tomatoes', 85.0, 19),
('Metacritic', 80.0, 19),
('IMDb', 9.2, 20),
('Rotten Tomatoes', 96.0, 20),
('Metacritic', 95.0, 20),
('IMDb', 6.7, 21),
('Rotten Tomatoes', 65.0, 21),
('Metacritic', 60.0, 21),
('IMDb', 8.1, 22),
('Rotten Tomatoes', 90.0, 22),
('Metacritic', 88.0, 22),
('IMDb', 7.3, 23),
('Rotten Tomatoes', 82.0, 23),
('Metacritic', 77.0, 23),
('IMDb', 8.8, 24),
('Rotten Tomatoes', 92.0, 24),
('Metacritic', 91.0, 24),
('IMDb', 7.4, 25),
('Rotten Tomatoes', 80.0, 25),
('Metacritic', 75.0, 25),
('IMDb', 6.1, 26),
('Rotten Tomatoes', 55.0, 26),
('Metacritic', 52.0, 26),
('IMDb', 8.6, 27),
('Rotten Tomatoes', 91.0, 27),
('Metacritic', 90.0, 27),
('IMDb', 7.0, 28),
('Rotten Tomatoes', 72.0, 28),
('Metacritic', 70.0, 28),
('IMDb', 9.3, 29),
('Rotten Tomatoes', 97.0, 29),
('Metacritic', 96.0, 29),
('IMDb', 6.9, 30),

('Rotten Tomatoes', 65.0, 30),
('Metacritic', 60.0, 30);

-- TABLA FACT SALES

INSERT INTO FACT_SALES (ID_movie, Production_company, Sale_Date, Revenue,
Budget)

VALUES

(1, 'Adventure Studios', '2023-01-01', 950000000.00, 200000000.00),
(2, 'Mystery Films', '2023-01-15', 850000000.00, 180000000.00),
(3, 'Timeless Productions', '2023-02-01', 780000000.00, 160000000.00),
(4, 'Historical Films', '2023-02-10', 920000000.00, 150000000.00),
(5, 'Cosmic Studios', '2023-03-05', 650000000.00, 130000000.00),
(6, 'Horror House', '2023-03-20', 980000000.00, 210000000.00),
(7, 'Heist Productions', '2023-04-10', 710000000.00, 120000000.00),
(8, 'Survival Studios', '2023-04-25', 600000000.00, 100000000.00),
(9, 'Royal Films', '2023-05-15', 1000000000.00, 220000000.00),
(10, 'Truth Seekers', '2023-05-30', 750000000.00, 175000000.00),
(11, 'Dream Chasers', '2023-06-10', 880000000.00, 140000000.00),
(12, 'Ocean Deep', '2023-06-25', 850000000.00, 200000000.00),
(13, 'Paris Films', '2023-07-05', 930000000.00, 210000000.00),
(14, 'Hacker Studios', '2023-07-20', 740000000.00, 160000000.00),
(15, 'AI World', '2023-08-01', 820000000.00, 190000000.00),
(16, 'Winter Productions', '2023-08-15', 660000000.00, 130000000.00),
(17, 'Memory Films', '2023-09-05', 920000000.00, 150000000.00),
(18, 'Extreme Sports Studios', '2023-09-20', 750000000.00, 160000000.00),
(19, 'Mythical Studios', '2023-10-01', 860000000.00, 180000000.00),
(20, 'Supernatural Films', '2023-10-15', 780000000.00, 170000000.00);

-- TABLA AWARDS

INSERT INTO AWARDS (Award_name, Year, Category)

VALUES

('Academy Award', 2023, 'Best Picture'),
('Golden Globe', 2023, 'Best Director'),
('BAFTA Award', 2023, 'Best Actor'),
('Academy Award', 2023, 'Best Actress'),
('Golden Globe', 2023, 'Best Screenplay'),
('BAFTA Award', 2023, 'Best Cinematography'),
('Academy Award', 2023, 'Best Supporting Actor'),
('Golden Globe', 2023, 'Best Supporting Actress'),
('BAFTA Award', 2023, 'Best Visual Effects'),
('Academy Award', 2024, 'Best Picture'),
('Golden Globe', 2024, 'Best Director'),
('BAFTA Award', 2024, 'Best Actor'),
('Academy Award', 2024, 'Best Actress'),
('Golden Globe', 2024, 'Best Screenplay'),
('BAFTA Award', 2024, 'Best Cinematography'),
('Academy Award', 2024, 'Best Supporting Actor'),
('Golden Globe', 2024, 'Best Supporting Actress'),

```
('BAFTA Award', 2024, 'Best Visual Effects'),  
( 'Cannes Film Festival', 2024, 'Palme d'Or'),  
( 'Sundance Film Festival', 2024, 'Grand Jury Prize');
```

```
-- TABLA MOVIE AWARDS
```

```
INSERT INTO MOVIE_AWARDS (ID_movie, ID_award)
```

```
VALUES
```

```
(1, 1),  
(2, 2),  
(3, 3),  
(4, 4),  
(5, 5),  
(6, 6),  
(7, 7),  
(8, 8),  
(9, 9),  
(10, 10),  
(11, 11),  
(12, 12),  
(13, 13),  
(14, 14),  
(15, 15),  
(16, 16),  
(17, 17),  
(18, 18),  
(19, 19),  
(20, 20);
```

```
-- TABLA TRANSACTION VIEW
```

```
INSERT INTO TRANSACTION_VIEW (User_ID, ID_movie, View_Date, Duration_Watched)
```

```
VALUES
```

```
(1, 1, '2023-01-02', '01:30:00'),  
(1, 2, '2023-01-16', '01:45:00'),  
(2, 3, '2023-02-02', '02:00:00'),  
(2, 4, '2023-02-11', '01:20:00'),  
(3, 5, '2023-03-06', '01:50:00'),  
(3, 6, '2023-03-21', '01:15:00'),  
(4, 7, '2023-04-11', '02:05:00'),  
(4, 8, '2023-04-26', '00:50:00'),  
(5, 9, '2023-05-16', '01:35:00'),  
(5, 10, '2023-05-31', '01:10:00'),  
(6, 11, '2023-06-11', '01:40:00'),  
(6, 12, '2023-06-26', '01:30:00'),  
(7, 13, '2023-07-06', '01:25:00'),  
(7, 14, '2023-07-21', '01:55:00'),  
(8, 15, '2023-08-02', '02:00:00'),  
(8, 16, '2023-08-16', '00:45:00'),  
(9, 17, '2023-09-06', '01:20:00'),
```



```
(9, 18, '2023-09-21', '01:15:00'),  
(10, 19, '2023-10-02', '01:30:00'),  
(10, 20, '2023-10-16', '01:40:00');
```

-- TABLA TRANSACTION RATING

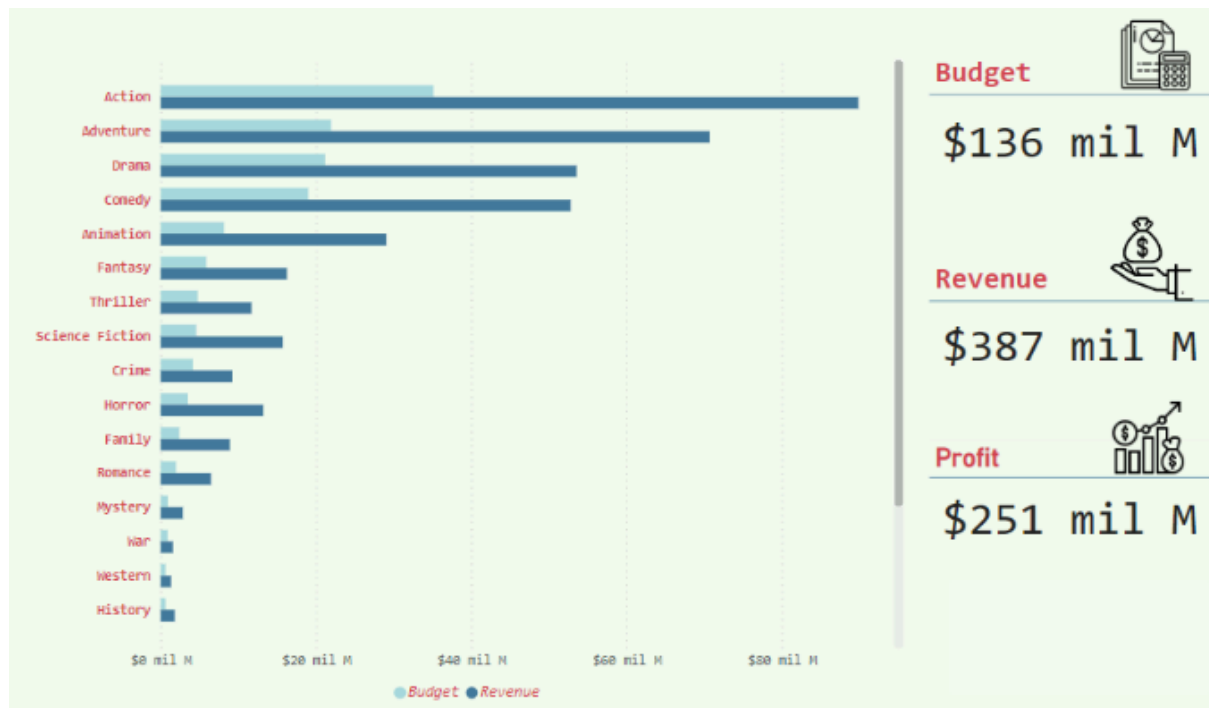
```
INSERT INTO TRANSACTION_RATING (User_ID, ID_movie, Rating, Rating_Date)  
VALUES
```

```
(1, 1, 8.5, '2023-01-16'),  
(2, 2, 7.8, '2023-02-21'),  
(3, 3, 9.0, '2023-03-26'),  
(4, 4, 8.2, '2023-04-11'),  
(5, 5, 7.5, '2023-05-13'),  
(6, 6, 8.9, '2023-06-06'),  
(7, 7, 8.4, '2023-07-19'),  
(8, 8, 9.1, '2023-08-09'),  
(9, 9, 7.6, '2023-09-11'),  
(10, 10, 8.3, '2023-10-16'),  
(1, 11, 7.7, '2023-11-13'),  
(2, 12, 8.8, '2023-12-26'),  
(3, 13, 8.1, '2024-01-16'),  
(4, 14, 8.6, '2024-02-11'),  
(5, 15, 9.0, '2024-03-19'),  
(6, 16, 7.9, '2024-04-06'),  
(7, 17, 8.5, '2024-05-23'),  
(8, 18, 8.7, '2024-06-13'),  
(9, 19, 7.5, '2024-07-26'),  
(10, 20, 8.9, '2024-08-09');
```

Informe

Informe de rentabilidad por género de película:

Este informe permite evaluar qué géneros de películas son más rentables en función del total de ingresos y presupuestos acumulados para cada género. Puede ser útil para tomar decisiones estratégicas sobre en qué géneros invertir más.



Herramientas y tecnologías usadas

- MySQL Workbench
- Github
- Google docs
- Mural
- Power BI