

# STREAMLYTICS

Movie Streaming Analytics using SQL 

Unlocking Insights from User Behaviour, Movie Trends and Revenue patterns

# TABLE OF CONTENT

- Project Overview
- Objectives
- Data Model
- Structure of Table
- Content of Table
- SQL Queries and Analysis
- Key Insights
- Conclusion

# PROJECT OVERVIEW

- The Movie Streaming Analytics Project (Streamlytics) is designed to analyze data from a streaming platform that manages users, subscriptions, movies, watch history, and ratings.
- The project simulates how real-world platforms like Netflix or Prime Video track content performance, user engagement, and revenue generation.

# OBJECTIVES



01

## User Behavior:

Identify most-watched movies & genres.

02

## Content Insights:

Find top-rated & trending movies.

03

## Subscriptions:

Track active users & revenue contribution.

04

## Business Value:

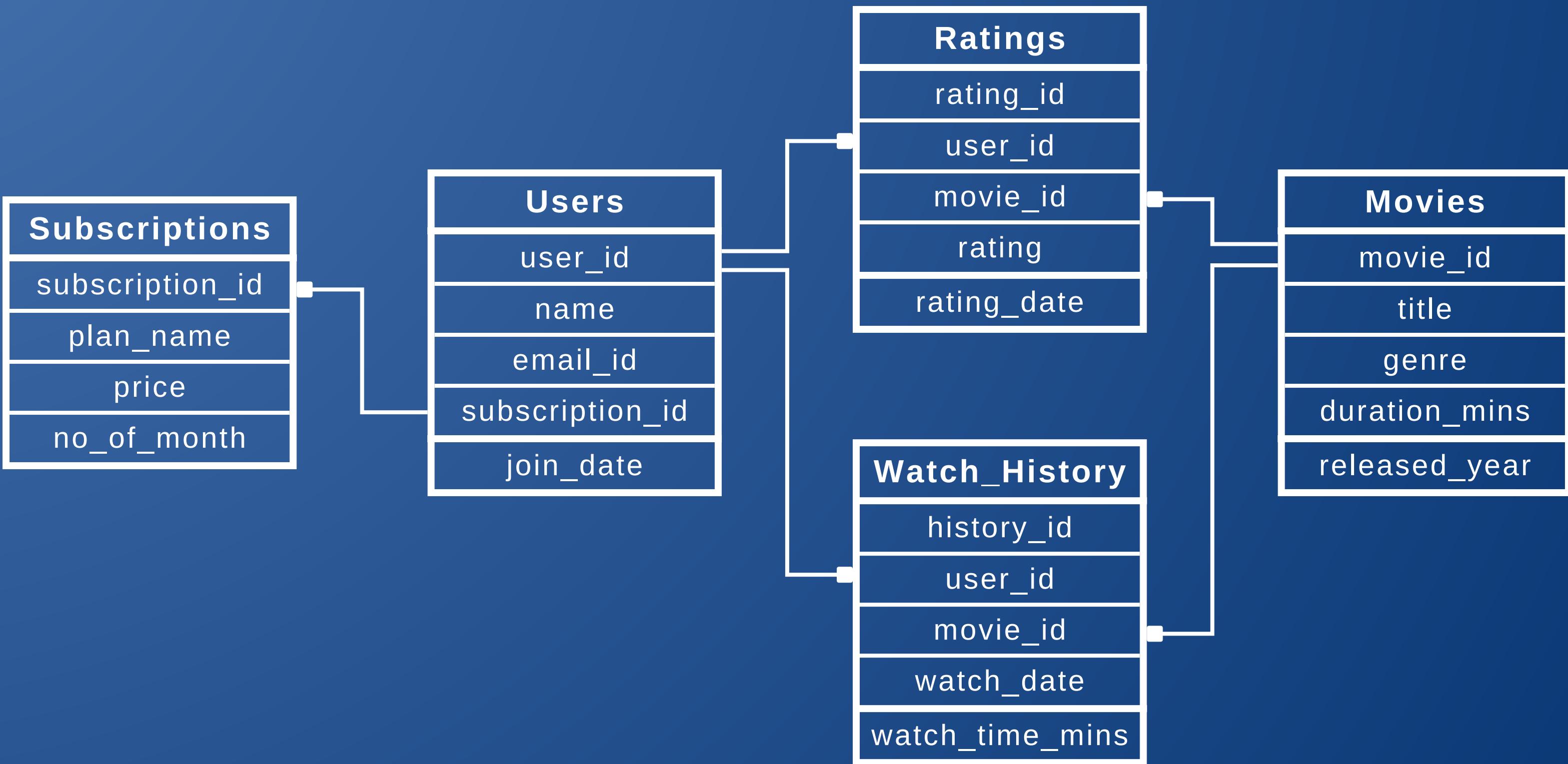
Guide decisions in content, marketing & pricing.

05

## SQL Practice:

Master joins, aggregates, subqueries & window functions.

# DATA MODEL



# STRUCTURE OF TABLES

## MOVIES TABLE

DESC movies;

Field	Type	Null	Key	Default	Extra
movie_id	int	NO	PRI	NULL	auto_increment
title	varchar(50)	NO		NULL	
genre	varchar(20)	YES		NULL	
duration_mins	int	YES		NULL	
released_year	year	YES		NULL	

## USERS TABLE

DESC users;

Field	Type	Null	Key	Default	Extra
user_id	int	NO	PRI	NULL	auto_increment
name	varchar(50)	NO		NULL	
email_id	varchar(100)	NO	UNI	NULL	
subscription_id	int	YES	MUL	NULL	
join_date	date	YES		NULL	

# STRUCTURE OF TABLES

## WATCH HISTORY TABLE

```
DESC watch_history;
```

Field	Type	Null	Key	Default	Extra
history_id	int	NO	PRI	NULL	auto_increment
user_id	int	YES	MUL	NULL	
movie_id	int	YES	MUL	NULL	
watch_date	date	YES		NULL	
watch_time_mins	int	YES		NULL	

## RATINGS TABLE

```
DESC ratings;
```

Field	Type	Null	Key	Default	Extra
rating_id	int	NO	PRI	NULL	auto_increment
user_id	int	YES	MUL	NULL	
movie_id	int	YES	MUL	NULL	
rating	decimal(2,1)	YES		NULL	
rating_date	date	YES		NULL	

# STRUCTURE OF TABLES

## SUBSCRIPTIONS TABLE

DESC subscriptions;

Field	Type	Null	Key	Default	Extra
subscription_id	int	NO	PRI	NULL	auto_increment
plan_name	varchar(20)	NO		NULL	
price	decimal(10,2)	NO		NULL	
no_of_month	int	YES		NULL	

# CONTENT OF THE TABLE

## MOVIES TABLE

SELECT \* FROM movies;

	movie_id	title	genre	duration_mins	released_year
▶	1	The Matrix Reloaded	Sci-Fi	88	2003
	2	Fast & Furious 6	Action	169	2013
	3	Mad Max: Fury Road	Action	132	2015
	4	Interstellar	Sci-Fi	81	2014
	5	Antonement	Drama	163	2007
	6	The Hangover Part III	Comedy	171	2013
	7	The Lost City	Comedy	139	2022
	8	Superbad	Comedy	150	2007
	9	The Invisible Man	Sci-Fi	123	2020
	10	Jurassic World	Action	87	2015
	11	Prometheus	Sci-Fi	126	2012
	12	Blade Runner 2049	Sci-Fi	114	2017
	13	The Fault in Our Stars	Romance	157	2014
	14	Tenet	Action	160	2020
	15	Love Again	Romance	115	2023
	16	28 Weeks Later	Horror	129	2012
	17	The Dark Knight	Action	83	2008
	18	Captain America: The...	Action	81	2014
	19	The Avengers	Action	85	2012
	20	Meet the Parents	Comedy	133	2000
	21	Devil wears the Prada	Comedy	83	2006
	22	Twilight	Romance	133	2008
	23	Red, White & Royal B...	Romance	172	2023
	24	Mission: Impossible II	Action	142	2000
	25	Crazy, Stupid, Love	Romance	97	2011

## USERS TABLE

SELECT \* FROM users;

	user_id	name	email_id	subscription_id	join_date
▶	1	Allison Hill	dam@gmail.com	3	2024-08-06
	2	Noah Rhodes	calebsmith@hall.com	1	2023-10-14
	3	Angie Henderson	whiteheadmichele@palmer.info	3	2023-09-26
	4	Daniel Wagner	jenniferkhan@kennedy.com	3	2025-05-24
	5	Cristian Santos	hopkinsmichael@owens-daniel.com	1	2023-11-11
	6	Connie Lawrence	zchandler@wright.net	1	2024-11-24
	7	Abigail Shaffer	robertramirez@gmail.com	3	2024-09-28
	8	Gina Moore	jmorton@williams.com	2	2025-05-07
	9	Gabrielle Davis	gabrieltucker@hancock.com	3	2023-09-13
	10	Ryan Munoz	gallowayjoseph@yahoo.com	3	2023-11-07
	11	Monica Herrera	uhorton@hotmail.com	3	2024-07-21
	12	Jamie Arnold	jamesrobinson@gmail.com	3	2023-01-05
	13	Lisa Hensley	brian97@calhoun.net	1	2023-11-24
	14	Michele Williams	caseyjones@powell.com	3	2023-09-06
	15	Dylan Miller	barnesbrandy@stewart.com	2	2023-11-03
	16	Brian Ramirez	randy47@hickman-walls.com	1	2024-02-10
	17	Holly Wood	jonesjason@yahoo.com	2	2024-01-16
	18	Derek Zuniga	nicole35@moore-bass.com	2	2024-12-24
	19	Lisa Jackson	sarah12@wilson-rodriguez.net	2	2023-01-10
	20	Carla Gray	juan35@watts.net	2	2023-04-09
	21	Margaret Hawki...	rebecca01@hotmail.com	1	2023-04-29
	22	Patty Perez	bradley52@hotmail.com	1	2023-09-01
	23	Ethan Adams	joycearnold@yahoo.com	2	2024-04-24
	24	Tommy Walter	jonesnicole@gardner.org	2	2025-08-20
	25	Matthew Foster	tanderson@nolan-flynn.com	1	2023-05-02

# CONTENT OF THE TABLE

## WATCH HISTORY TABLE

SELECT \* FROM watch\_history;

	history_id	user_id	movie_id	watch_date	watch_time_mins
▶	1	27	28	2025-02-07	175
	2	15	20	2025-09-06	86
	3	29	14	2025-06-10	127
	4	38	7	2024-11-22	90
	5	18	30	2024-10-05	28
	6	1	18	2025-04-23	12
	7	49	25	2025-06-01	127
	8	11	24	2025-07-25	169
	9	45	23	2025-05-07	154
	10	28	7	2024-12-09	35
	11	22	23	2024-10-15	28
	12	18	10	2025-06-03	147
	13	10	13	2025-07-03	64
	14	14	22	2025-01-01	139
	15	49	21	2025-08-02	77
	16	22	12	2025-05-29	43
	17	7	15	2025-04-27	99
	18	6	29	2025-07-15	27
	19	25	17	2024-10-23	72
	20	7	15	2025-07-12	104
	21	23	4	2025-04-13	82
	22	23	8	2025-07-06	50
	23	39	8	2024-09-21	122
	24	17	3	2025-01-18	149
	25	3	11	2025-03-31	87

## RATINGS TABLE

SELECT \* FROM ratings;

	rating_id	user_id	movie_id	rating	rating_date
▶	1	43	1	4.7	2025-08-06
	2	6	30	2.1	2025-03-10
	3	19	13	2.8	2024-10-20
	4	33	19	4.0	2025-07-22
	5	43	19	3.3	2024-10-01
	6	41	22	3.7	2025-02-08
	7	40	1	2.3	2025-03-15
	8	22	3	3.0	2024-10-04
	9	6	21	2.3	2025-04-18
	10	49	14	3.0	2025-05-08
	11	16	5	2.4	2025-05-30
	12	44	28	2.0	2024-09-26
	13	20	15	2.1	2024-11-03
	14	15	6	3.4	2024-11-02
	15	13	2	4.5	2025-07-18
	16	10	9	3.2	2025-04-12
	17	2	13	3.1	2025-01-01
	18	3	11	1.8	2024-10-09
	19	16	7	2.0	2024-12-11
	20	31	15	2.6	2024-10-22
	21	40	11	3.2	2025-04-10
	22	50	11	2.0	2025-06-29
	23	5	25	2.9	2025-02-14
	24	30	29	3.8	2025-04-29
	25	27	13	2.8	2025-04-25

# CONTENT OF THE TABLE

## SUBSCRIPTIONS TABLE

```
SELECT * FROM subscriptions;
```

	subscription_id	plan_name	price	no_of_month
▶	1	Basic	299.00	3
	2	Standard	499.00	6
	3	Premium	799.00	12
●	NULL	NULL	NULL	NULL

# SQL QUERIES & ANALYSIS

- CONTENT INSIGHTS
- USER INSIGHTS
- SUBSCRIPTION AND REVENUE INSIGHTS

# CONTENT INSIGHTS

## 1] TOTAL NO. OF MOIVES

```
SELECT COUNT(*) AS  
Total_No_of_Movies FROM movies;
```

	Total_No_of_Movies
▶	30

## 2] TOTAL NO. OF MOVIES RELEASED IN EACH YEAR

```
SELECT Released_Year,  
COUNT(movie_id) AS Total_Movies  
FROM movies  
GROUP BY released_year  
ORDER BY released_year;
```

	Released_Year	Total_Movies
▶	2000	2
	2003	1
	2006	1
	2007	3
	2008	2
	2010	1
	2011	1
	2012	3
	2013	2
	2014	3
	2015	2
	2016	1
	2017	1
	2018	1
	2020	2
	2022	1
	2023	3

# CONTENT INSIGHTS

## 3] TOTAL NO. OF MOVIES IN EACH GENRE

```
SELECT Genre, COUNT(movie_id) AS  
Total_Movies  
FROM movies  
GROUP BY genre  
ORDER BY total_movies DESC;
```

	Genre	Total_Movies
▶	Sci-Fi	8
	Action	8
	Comedy	6
	Romance	6
	Drama	1
	Horror	1

## 4] RETRIEVE TOTAL MOVIES RELEASED IN EACH GENRE COMPARING WITH TOTAL MOVIES RELEASED IN A SPECIFIC YEAR (EX: YEAR-2013)

```
SELECT Genre, COUNT(movie_id) AS  
Movies_in_Genre,  
(SELECT COUNT(*) FROM Movies  
WHERE released_year = 2013) AS  
Total_Movies  
FROM movies  
WHERE released_year = 2013  
GROUP BY genre  
ORDER BY movies_in_genre DESC;
```

	Genre	Movies_in_Genre	Total_Movies
▶	Sci-Fi	1	3
	Romance	1	3
	Action	1	3

# CONTENT INSIGHTS

## 5] LIST ALL MOVIES IN A SPECIFIC GENRE (EX: GENRE-ACTION)

```
SELECT Title, Released_Year,  
Duration_mins  
FROM movies  
WHERE genre = 'Action'  
ORDER BY released_year;
```

	Title	Released_Year	Duration_mins
▶	Mission: Impossible II	2000	142
	The Dark Knight	2008	83
	The Avengers	2012	85
	Fast & Furious 6	2013	169
	Captain America: The Winter Soldier	2014	81
	Mad Max: Fury Road	2015	132
	Jurassic World	2015	87
	Tenet	2020	160

## 6] TOP 5 MOVIES BY AVG\_RATING

```
SELECT m.Title, m.Genre,  
ROUND(AVG(r.rating),1) AS Avg_Rating  
FROM movies m INNER JOIN ratings r  
ON m.movie_id=r.movie_id  
GROUP BY m.title, m.genre  
ORDER BY Avg_Rating DESC  
LIMIT 5;
```

	Title	Genre	Avg_Rating
▶	Jurassic World	Action	4.0
	Red, White & Royal Blue	Romance	3.8
	The Matrix Reloaded	Sci-Fi	3.6
	Devil wears the Prada	Comedy	3.6
	Ready Player One	Sci-Fi	3.6

# CONTENT INSIGHTS

## 7] TOP 5 MOVIES BY TOTAL\_WATCH\_TIME

```
SELECT m.Title, m.Genre,  
SUM(w.watch_time_mins) AS  
Total_Watch_Time  
FROM movies m INNER JOIN  
watch_history w  
ON m.movie_id=w.movie_id  
GROUP BY m.title, m.genre  
ORDER BY Total_Watch_Time DESC  
LIMIT 5;
```

	Title	Genre	Total_Watch_Time
▶	Fast & Furious 6	Action	1297
	Inception	Sci-Fi	1237
	Mad Max: Fury Road	Action	1183
	The Avengers	Action	1120
	Superbad	Comedy	1057

## 8] LIST THE MOVIES THAT ARE BELOW AVG AND COUNT OF USER WHO WATCH THEM

```
SELECT m.Movie_id, m.Title,  
ROUND(AVG(r.rating),1) AS No_Ratings,  
COUNT(r.rating_id) AS Count_of_user  
FROM movies m LEFT JOIN ratings r  
ON m.movie_id=r.movie_id  
GROUP BY m.title, m.movie_id  
HAVING NO_Ratings < (SELECT AVG(r.rating)  
FROM ratings r)  
ORDER BY count_of_user;
```

	Movie_id	Title	No_Ratings	Count_of_user
▶	17	The Dark Knight	2.7	2
	18	Captain America: The Winter Soldier	1.9	3
	6	The Hangover Part III	2.1	4
	7	The Lost City	2.7	4
	16	28 Weeks Later	2.5	4
	26	Knocked Up	2.8	4
	28	Inception	2.3	4
	3	Mad Max: Fury Road	2.6	5
	8	Superbad	2.5	5
	12	Blade Runner 2049	1.4	5
	4	Interstellar	2.2	6
	14	Tenet	2.2	7
	25	Crazy, Stupid, Love	2.8	8
	27	The Creator	2.7	9

# CONTENT INSIGHTS

## 9] SHOW THE LEAST WATCHED MOVIES BY WATCH\_TIME

```
SELECT m.Title, m.Genre, m.Duration_mins,  
SUM(w.watch_time_mins) AS  
"Total_Watched_Time(mins)"  
FROM movies m LEFT JOIN watch_history w  
ON m.movie_id=w.movie_id  
GROUP BY m.title, m.genre, m.duration_mins  
ORDER BY SUM(w.watch_time_mins);
```

	Title	Genre	Duration_mins	Total_Watched_Time(mins)
▶	Blade Runner 2049	Sci-Fi	114	55
	The Creator	Sci-Fi	123	66
	Crazy, Stupid, Love	Romance	97	215
	Devil wears the Prada	Comedy	83	338
	Jurassic World	Action	87	344
	The Hangover Part III	Comedy	171	418
	Mission: Impossible II	Action	142	421
	The Matrix Reloaded	Sci-Fi	88	425
	The Dark Knight	Action	83	435
	Meet the Parents	Comedy	133	454

# CONTENT INSIGHTS

	Released_Year	Top_Rated_Movie	Avg_Rating
▶	2000	Meet the Parents	3.2
	2003	The Matrix Reloaded	3.6
	2006	Devil wears the Prada	3.6
	2007	Antonement	3.0
	2008	Twilight	3.2
	2010	Inception	2.3
	2011	Crazy, Stupid, Love	2.8
	2012	The Avengers	3.1
	2013	Fast & Furious 6	3.5
	2014	The Fault in Our Stars	3.0
	2015	Jurassic World	4.0
	2016	La La Land	3.2
	2017	Blade Runner 2049	1.4
	2018	Ready Player One	3.6
	2020	The Invisible Man	3.1
	2022	The Lost City	2.7
	2023	Red, White & Royal ...	3.8

## 10] TOP RATED MOVIE FROM EACH YEAR

```
SELECT Released_Year, title AS Top_Rated_Movie,  
Avg_Rating  
FROM  
(SELECT m.released_year, m.title,  
ROUND(AVG(r.rating), 1) AS avg_rating,  
ROW_NUMBER()  
OVER (PARTITION BY m.released_year ORDER BY  
AVG(r.rating) DESC) AS rn  
FROM movies m JOIN ratings r  
ON m.movie_id = r.movie_id  
GROUP BY m.released_year, m.title) AS ranked  
WHERE rn = 1;
```

# CONTENT INSIGHTS

## 11] MOST WATCHED MOVIE FROM EACH YEAR

```
SELECT Year_Released, title AS Most_Watched_Movie,  
Watch_Count  
FROM  
(SELECT m.released_year AS year_released,m.title,  
COUNT(w.history_id) AS watch_count,  
ROW_NUMBER()  
OVER (PARTITION BY m.released_year ORDER BY  
COUNT(w.history_id) DESC) AS rn  
FROM movies m JOIN watch_history w  
ON m.movie_id = w.movie_id  
GROUP BY m.released_year, m.title) AS ranked  
WHERE rn = 1;
```

	Year_Released	Most_Watched_Movie	Watch_Count
	2000	Mission: Impossible II	7
	2003	The Matrix Reloaded	4
	2006	Devil wears the Prada	5
	2007	Superbad	13
	2008	Twilight	9
	2010	Inception	12
	2011	Crazy, Stupid, Love	2
	2012	The Avengers	10
	2013	Fast & Furious 6	12
	2014	Captain America: The Winter Soldier	8
	2015	Mad Max: Fury Road	10
	2016	La La Land	7
	2017	Blade Runner 2049	2
	2018	Ready Player One	5
	2020	Tenet	9
	2022	The Lost City	10
	2023	Love Again	7

# CONTENT INSIGHTS

## 12] TOP RATED MOVIE BY EACH GENRE

```
SELECT Genre, title AS Top_Rated_Movies, Avg_Rating  
FROM (SELECT m.genre, m.title,  
ROUND(AVG(r.rating), 2) AS avg_rating,  
ROW_NUMBER()  
OVER (PARTITION BY m.genre ORDER BY  
AVG(r.rating) DESC) AS rn  
FROM Movies m JOIN Ratings r  
ON m.movie_id = r.movie_id  
GROUP BY m.genre, m.title) AS ranked  
WHERE rn = 1;
```

	Genre	Top_Rated_Movies	Avg_Rating
▶	Action	Jurassic World	3.97
	Comedy	Devil wears the Prada	3.56
	Drama	Antonement	3.00
	Horror	28 Weeks Later	2.48
	Romance	Red, White & Royal Blue	3.83
	Sci-Fi	The Matrix Reloaded	3.60

## 13] MOST WATCHED MOVIE FROM EACH GENRE

```
SELECT Genre, title AS Most_Watched_Movie,  
Watch_Count  
FROM (SELECT m.genre, m.title,  
COUNT(w.history_id) AS watch_count,  
ROW_NUMBER()  
OVER (PARTITION BY m.genre ORDER BY  
COUNT(w.history_id) DESC) AS rn  
FROM Movies m JOIN Watch_History w  
ON m.movie_id = w.movie_id  
GROUP BY m.genre, m.title) AS ranked WHERE rn = 1;
```

	Genre	Most_Watched_Movie	Watch_Count
▶	Action	Fast & Furious 6	12
	Comedy	Superbad	13
	Drama	Antonement	4
	Horror	28 Weeks Later	7
	Romance	Twilight	9
	Sci-Fi	Inception	12

# USER INSIGHTS

	User_ID	Name	Genres_Watched
▶	1	Allison Hill	Action,Comedy
	4	Daniel Wagner	Comedy,Horror,Sci-Fi
	5	Cristian Santos	Action,Comedy,Sci-Fi
	6	Connie Lawrence	Action,Comedy,Romance,Sci-Fi
	7	Abigail Shaffer	Horror,Romance
	9	Gabrielle Davis	Comedy,Romance,Sci-Fi
	10	Ryan Munoz	Action,Romance,Sci-Fi
	11	Monica Herrera	Action,Comedy,Romance,Sci-Fi
	14	Michele Williams	Action,Comedy,Horror,Romance
	15	Dylan Miller	Action,Comedy,Romance
	16	Brian Ramirez	Action,Sci-Fi
	17	Holly Wood	Action,Drama,Romance
	18	Derek Zuniga	Action,Comedy,Horror,Romance
	19	Lisa Jackson	Action,Comedy
	20	Carla Gray	Action,Comedy
	21	Margaret Hawki...	Action,Sci-Fi
	22	Patty Perez	Action,Romance,Sci-Fi
	23	Ethan Adams	Action,Comedy,Sci-Fi
	24	Tommy Walter	Action,Comedy,Romance
	25	Matthew Foster	Action,Comedy,Romance,Sci-Fi
	26	Judy Baker	Action,Comedy
	28	Stephanie Ross	Action,Comedy,Sci-Fi
	30	Anthony Rodrig...	Action,Comedy,Drama,Romance
	32	James Ferrell	Action,Comedy,Romance,Sci-Fi

## 1] USERS WHO HAVE WATCHED MOVIES IN MULTIPLE GENRE WITH GENRES

```
SELECT u.User_ID, u.Name,
GROUP_CONCAT(DISTINCT m.genre ORDER BY
m.genre) AS Genres_Watched
FROM users u JOIN watch_history w
ON u.user_id = w.user_id
JOIN Movies m ON w.movie_id = m.movie_id
GROUP BY u.user_id, u.name
HAVING COUNT(DISTINCT m.genre) > 1;
```

33	Tricia Valencia	Action,Comedy
34	Nathan Maldonado	Comedy,Romance,Sci-Fi
35	Debra Davidson	Action,Romance,Sci-Fi
36	Jeffrey Chavez	Action,Comedy,Sci-Fi
37	Sherri Baker	Action,Comedy,Romance
38	Cassandra Gaines	Action,Comedy,Sci-Fi
39	Elizabeth Fowler	Action,Comedy,Romance,Sci-Fi
41	Paula Moreno	Action,Comedy,Romance,Sci-Fi
42	Fred Smith	Action,Comedy,Sci-Fi
43	Sherry Decker	Comedy,Sci-Fi
44	Anthony Humph...	Action,Comedy,Romance,Sci-Fi
45	Angelica Tucker	Horror,Romance
46	Philip Cannon	Drama,Sci-Fi
47	John Pierce	Action,Sci-Fi
48	Shane Henderson	Comedy,Sci-Fi
49	Joshua Blair	Action,Comedy,Horror,Romance
50	Eric Carney	Action,Comedy,Drama,Romance

# USER INSIGHTS

## 2] USERS WITH "WATCHED AND RATED MOVIE'S GENRE" AND "WATCHED AND NON-RATED MOVIES'S GENRE"

```
SELECT u.User_ID, u.Name,  
GROUP_CONCAT(DISTINCT CASE WHEN  
r.rating_id IS NOT NULL THEN m.genre END) AS  
Watched_and_Rated_Genres,  
GROUP_CONCAT(DISTINCT CASE WHEN  
r.rating_id IS NULL THEN m.genre END) AS  
Watched_and_not_Rated_Genres  
FROM users u JOIN watch_history w  
ON u.user_id = w.user_id  
JOIN movies m ON w.movie_id = m.movie_id  
LEFT JOIN Ratings r ON u.user_id = r.user_id AND  
m.movie_id = r.movie_id  
GROUP BY u.user_id, u.name  
ORDER BY u.user_id;
```

	User_ID	Name	Watched_and_Rated_Genres	Watched_and_not_Rated_Genres
▶	1	Allison Hill	NULL	Action,Comedy
	2	Noah Rhodes	NULL	Action
	3	Angie Henderson	Sci-Fi	Sci-Fi
	4	Daniel Wagner	NULL	Comedy,Horror,Sci-Fi
	5	Cristian Santos	Comedy	Action,Sci-Fi
	6	Connie Lawrence	NULL	Action,Comedy,Romance,Sci-Fi
	7	Abigail Shaffer	NULL	Horror,Romance
	8	Gina Moore	NULL	Comedy
	9	Gabrielle Davis	Sci-Fi	Comedy,Romance
	10	Ryan Munoz	NULL	Action,Romance,Sci-Fi
	11	Monica Herrera	NULL	Action,Comedy,Romance,Sci-Fi
	12	Jamie Arnold	NULL	Horror
	13	Lisa Hensley	NULL	Action
	14	Michele Williams	NULL	Action,Comedy,Horror,Romance
	15	Dylan Miller	Comedy	Action,Comedy,Romance
	16	Brian Ramirez	NULL	Action,Sci-Fi
	17	Holly Wood	NULL	Action,Drama,Romance
	18	Derek Zuniga	Comedy	Action,Comedy,Horror,Romance
	19	Lisa Jackson	NULL	Action,Comedy
	20	Carla Gray	NULL	Action,Comedy
	21	Margaret Hawki...	NULL	Action,Sci-Fi
	22	Patty Perez	Sci-Fi	Action,Romance
	23	Ethan Adams	NULL	Action,Comedy,Sci-Fi
	24	Tommy Walter	NULL	Action,Comedy,Romance
	25	Matthew Foster	NULL	Action,Comedy,Romance,Sci-Fi

# USER INSIGHTS

## 2] USERS WITH "WATCHED AND RATED MOVIE'S GENRE" AND "WATCHED AND NON-RATED MOVIES'S GENRE"

```
SELECT u.User_ID, u.Name,  
GROUP_CONCAT(DISTINCT CASE WHEN  
r.rating_id IS NOT NULL THEN m.genre END) AS  
Watched_and_Rated_Genres,  
GROUP_CONCAT(DISTINCT CASE WHEN  
r.rating_id IS NULL THEN m.genre END) AS  
Watched_and_not_Rated_Genres  
FROM users u JOIN watch_history w  
ON u.user_id = w.user_id  
JOIN movies m ON w.movie_id = m.movie_id  
LEFT JOIN Ratings r ON u.user_id = r.user_id AND  
m.movie_id = r.movie_id  
GROUP BY u.user_id, u.name  
ORDER BY u.user_id;
```

	User_ID	Name	Watched_and_Rated_Genres	Watched_and_not_Rated_Genres
26	Judy Baker	NULL		Action,Comedy
27	Justin Baker	NULL		Sci-Fi
28	Stephanie Ross	NULL		Action,Comedy,Sci-Fi
29	Zachary Hicks	NULL		Action
30	Anthony Rodrig...	NULL		Action,Comedy,Drama,Romance
31	Rebecca Hender...	NULL		Comedy
32	James Ferrell	NULL		Action,Comedy,Romance,Sci-Fi
33	Tricia Valencia	NULL		Action,Comedy
34	Nathan Maldonado	NULL		Comedy,Romance,Sci-Fi
35	Debra Davidson	Action,Sci-Fi		Action,Romance,Sci-Fi
36	Jeffrey Chavez	Sci-Fi		Action,Comedy,Sci-Fi
37	Sherri Baker	NULL		Action,Comedy,Romance
38	Cassandra Gaines	Sci-Fi		Action,Comedy
39	Elizabeth Fowler	Action		Comedy,Romance,Sci-Fi
40	Brittany Farmer	NULL		Comedy
41	Paula Moreno	NULL		Action,Comedy,Romance,Sci-Fi
42	Fred Smith	NULL		Action,Comedy,Sci-Fi
43	Sherry Decker	NULL		Comedy,Sci-Fi
44	Anthony Humph...	NULL		Action,Comedy,Romance,Sci-Fi
45	Angelica Tucker	NULL		Horror,Romance
46	Philip Cannon	NULL		Drama,Sci-Fi
47	John Pierce	Action,Sci-Fi		Action,Sci-Fi
48	Shane Henderson	NULL		Comedy,Sci-Fi
49	Joshua Blair	Romance		Action,Comedy,Horror,Romance
50	Eric Carney	NULL		Action,Comedy,Drama,Romance

# USER INSIGHTS

	User_ID	Name	Total_Watch_Time	Total_Avg_Watch_Time
▶	1	Allison Hill	361	379.32
	2	Noah Rhodes	180	379.32
	3	Angie Henderson	267	379.32
	4	Daniel Wagner	302	379.32
	5	Cristian Santos	368	379.32
	6	Connie Lawrence	515	379.32
	7	Abigail Shaffer	460	379.32
	8	Gina Moore	483	379.32
	9	Gabrielle Davis	404	379.32
	10	Ryan Munoz	339	379.32
	11	Monica Herrera	899	379.32
	12	Jamie Arnold	120	379.32
	13	Lisa Hensley	123	379.32
	14	Michele Williams	309	379.32
	15	Dylan Miller	577	379.32
	16	Brian Ramirez	379	379.32
	17	Holly Wood	684	379.32
	18	Derek Zuniga	725	379.32
	19	Lisa Jackson	433	379.32
	20	Carla Gray	336	379.32
	21	Margaret Hawki...	332	379.32
	22	Patty Perez	110	379.32
	23	Ethan Adams	212	379.32
	24	Tommy Walter	470	379.32
	25	Matthew Foster	781	379.32

## 3] CALCULATE EACH USER'S TOTAL\_WATCH\_TIME COMPARING WITH AVG\_WATCH\_TIME OF ALL USERS

```
SELECT u.User_ID, u.Name,  
SUM(w.watch_time_mins) AS Total_Watch_Time,  
ROUND(AVG(SUM(w.watch_time_mins))  
OVER (), 2) AS 'Avg_Watch_Time(All Users)'  
FROM users u  
JOIN watch_history w ON u.user_id = w.user_id  
GROUP BY u.user_id, u.name;
```

# USER INSIGHTS

	User_ID	Name	Total_Watch_Time	Total_Avg_Watch_Time
	26	Judy Baker	255	379.32
	27	Justin Baker	175	379.32
	28	Stephanie Ross	333	379.32
	29	Zachary Hicks	127	379.32
	30	Anthony Rodrig...	375	379.32
	31	Rebecca Hender...	123	379.32
	32	James Ferrell	247	379.32
	33	Tricia Valencia	333	379.32
	34	Nathan Maldonado	214	379.32
	35	Debra Davidson	691	379.32
	36	Jeffrey Chavez	509	379.32
	37	Sherri Baker	342	379.32
	38	Cassandra Gaines	281	379.32
	39	Elizabeth Fowler	485	379.32
	40	Brittany Farmer	44	379.32
	41	Paula Moreno	525	379.32
	42	Fred Smith	402	379.32
	43	Sherry Decker	215	379.32
	44	Anthony Humph...	532	379.32
	45	Angelica Tucker	356	379.32
	46	Philip Cannon	265	379.32
	47	John Pierce	451	379.32
	48	Shane Henderson	97	379.32
	49	Joshua Blair	794	379.32
	50	Eric Carney	626	379.32

**3] CALCULATE EACH USER'S TOTAL\_WATCH\_TIME COMPARING WITH AVG\_WATCH\_TIME OF ALL USERS**

```
SELECT u.User_ID, u.Name,  
SUM(w.watch_time_mins) AS Total_Watch_Time,  
ROUND(AVG(SUM(w.watch_time_mins))  
OVER (), 2) AS 'Avg_Watch_Time(All Users)'  
FROM users u  
JOIN watch_history w ON u.user_id = w.user_id  
GROUP BY u.user_id, u.name;
```

# USER INSIGHTS

## 4] DISPLAY TOTAL WATCH\_SESSION EACH YEAR

```
SELECT YEAR(watch_date) AS Year,  
COUNT(*) AS Total_Watch_Session  
FROM watch_history  
GROUP BY Year  
ORDER BY Year;
```

	Year	Total_Watch_Session
▶	2024	62
	2025	138

## 5] DISPLAY TOTAL USERS JOINED EACH YEAR

```
SELECT YEAR(join_date) AS Year,  
COUNT(*) AS Total_Users_joined  
FROM users  
GROUP BY year  
ORDER BY Year;
```

	Year	Total_Users_joined
▶	2022	4
	2023	17
	2024	22
	2025	7

# Subscription & Revenue Insights

## 1] ACTIVE USERS IN EACH SUBSCRIPTION PLAN

```
SELECT s.plan_name AS Subscription_Plan,  
COUNT(u.user_id) AS Active_Users  
FROM users u JOIN subscriptions s  
ON u.subscription_id = s.subscription_id  
WHERE CURRENT_DATE  
BETWEEN u.join_date AND  
DATE_ADD(u.join_date, INTERVAL s.no_of_month  
MONTH)  
GROUP BY s.plan_name  
ORDER BY active_users DESC;
```

	Subscription_Plan	Active_Users
▶	Premium	4
	Standard	2
	Basic	1

## 2] MOST USED SUBSCRIPTION PLAN

```
SELECT s.plan_name AS  
Subscription_Plan, COUNT(u.user_id) AS  
User_Count  
FROM subscriptions s  
JOIN users u ON s.subscription_id =  
u.subscription_id  
GROUP BY s.plan_name  
ORDER BY user_count DESC;
```

	Subscription_Plan	User_Count
▶	Premium	20
	Basic	15
	Standard	15

# Subscription & Revenue Insights

## 3] TOTAL REVENUE GENERATED FROM SUBSCRIPTIONS

```
SELECT SUM(s.price) AS Total_Revenue  
FROM subscriptions s  
JOIN users u  
ON s.subscription_id = u.subscription_id;
```

	Total_Revenue
▶	27950.00

## 4] TOTAL REVENUE GENERATED PER YEAR FROM SUBSCRIPTIONS

```
SELECT YEAR(u.join_date) AS Year,  
SUM(s.price) AS Revenue  
FROM subscriptions s  
JOIN users u  
ON s.subscription_id = u.subscription_id  
GROUP BY YEAR(u.join_date)  
ORDER BY year;
```

	Year	Revenue
▶	2022	1896.00
	2023	9183.00
	2024	13178.00
	2025	3693.00

# Subscription & Revenue Insights

## 5] TOTAL REVENUE GENERATED FROM EACH SUBSCRIPTION PLAN IN RS AND PERCENTAGE

```
SELECT s.plan_name AS Subscription_Plan,  
       (COUNT(u.user_id) * s.price) AS Revenue,  
       ROUND(((COUNT(u.user_id)*s.price)*100.0) /  
             SUM(COUNT(u.user_id)*s.price))  
             OVER (), 2) AS Percentage_Contribution  
FROM subscriptions s  
JOIN users u ON s.subscription_id =  
u.subscription_id  
GROUP BY s.subscription_id, s.plan_name, s.price  
ORDER BY revenue DESC;
```

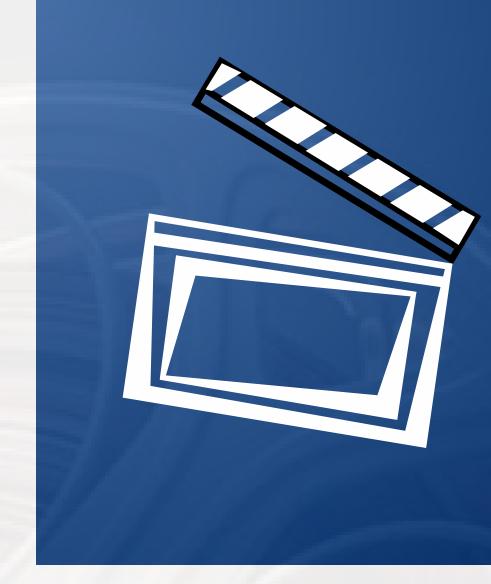
	Subscription_Plan	Revenue	Percentage_Contribution
▶	Premium	15980.00	57.17
	Standard	7485.00	26.78
	Basic	4485.00	16.05

# KEY INSIGHTS

HOUR MINUTE SECOND

DATE

DIRECTOR



## Content Insights

- Top 5 most-watched movies across all years.
- Top-rated movies by genre and by year.
- Popularity of genres based on watch sessions and ratings.
- Yearly release trends showing growth/decline of genres.

## User Insights

- Identification of active vs inactive users by subscription duration.
- Power users: Users who watch more than average watch time.
- Viewing diversity: Users who watch movies from multiple genres.
- Ratio of watched-only vs watched-and-rated movies per user.

## Subscription & Revenue Insights

- Active users per subscription plan.
- Most used subscription plan (Basic, Standard, Premium).
- Total revenue and yearly revenue trends.
- Revenue contribution (%) of each subscription plan.

# CONCLUSION

## SUMMARY OF FINDINGS

- Identified top-performing movies and most popular genres across years.
- Discovered user engagement patterns, including watch time and multi-genre viewing habits.
- Analyzed subscription trends, active users, and revenue contribution per plan.
- Generated actionable business insights for content acquisition, marketing, and pricing.

## FUTURE ENHANCEMENT

- Integrate real-time analytics for live user behavior tracking.
- Add recommendation system using machine learning for personalized movie suggestions.
- Expand dataset with regional movies, TV shows, and series for deeper insights.
- Incorporate visual dashboards (Power BI/Tableau) for interactive exploration.
- Perform predictive analysis for revenue forecasting and churn prediction.

# THANK YOU

PROD.

ROLL

DIRECTOR:

CAMERA:

DATE:

SCENE

TA

SOUND: