LAB REPORT – 1

For

Database Systems for Analytics

Group – 11

Shazid Shaik
Sri Karthik Nandam
Sai Nikhil Juluri

ABSTRACT

The report concerns the application for finding the latest movies with the help of a database called TMDB. This application has a database which shows you popular stuff, gives you custom movie recommendations, and it also simulates users by using their stats and data which they give to the application. The report gives a clear picture of the benefits of using the database and moving the app to Amazon cloud services, what the application needs, and what the application can do and limitations. It also comprises of the planning procedures for the database design, identification of the functions required, moving the database from the .CSV files to Amazon's servers and lastly, the connection of the App to the database through the utilization of the Python Programming Language. In brief, it is about going through the implementation of a cloud-based, movie analysis and recommendation application, covering the points of reason, design, migration, and integration.

1. INTRODUCTION

1.1. PROPOSED SYSTEM OVERVIEW

The proposed system aims at the study of the movie industry profoundly by utilizing the TMDB (The Movie Database) database. This system will make it possible to generate the trends and personalized advice with movies and model user behavior based on the movie information and comments.

1.2. SYSTEM LIMITATIONS

This system has one limitation that consists in the lack of real-time data availability because it is grounded on the periodical updates to the TMDB data set that sometimes may is behind the most current movie releases or user comments. On the other hand, users will operate with the system mainly by a pre-defined interface or the script, rather than touching data directly or searching randomly.

2. METHODOLOGY

2.1. THE NEED FOR A DATABASE.

2.1.1. Data Storage

Thus, data storage requires the use of databases. TMDB hold this enormous collection of structured information which include movies data and user ratings. Organization of the data will make it easy to retrieve and analyze it.

2.1.2. Data Integrity

Data integrity will be provided by storing data inside the database, using the constraints to primary keys and the foreign keys that defines the connections between various entities (e.g., movies, credits, and ratings).

2.1.3. Query and Analysis

As we move forward Databases, equip us with a set structure for doing complicated queries and analytics and this makes it possible to extract significant insights and to see patterns neatly from a set of data.

2.2. JUSTIFICATION FOR MIGRATING TO AWS.

2.2.1. Flexibility

Switching to AWS gives you the opportunity to be flexible, as AWS has a variety of services under its umbrella which means you can choose the most efficient service for placing your database, dataset files in storage facilities, and analyzing the data. This will turn out to be an adjustable model that regards utilization of the right scenarios and its unique specifications.

2.2.2. Scalability

AWS delivers scalable solutions for both storage and computing resources that are flexible. As the dataset gets bigger or the analytical tasks get more complex, the infrastructure can be just scaled up to handle the workload.

2.2.3. Resilience

AWS maintains an in-built redundancy and high availability properties which ensures that the database is accessible and stable in case of hardware failures or other disturbances thus resulting in longer period of the analytical system.

2.2.4. Cost-effectiveness

You can save money using Amazon's "pay-as-you-go" feature. With this mode, you are charged only for the computational power and resources that you utilize. This feature enables you to scale up or down according to your requirements at a specific instant in time as a substitute for a fixed payment. By merely paying for what you use and when you need to, you will be able to lower substantially your operations costs.

3. SOLUTION REQUIREMENTS

3.1. Data Storage and Management

The solution includes the ability to organize and manage TMDB files into a structured way and guarantee the integrity of the database schema design as well as define restrictions for the users.

3.2. Analysis Capabilities

This solution is supposed to be able to perform complicated queries and analytic functions on the data set. Also, it provides functionalities for trend analysis, suggestion generating and user behavior modeling.

3.3. Ability to Grow & Improve Speed

It must be built so that the time will come when it can handle extremely large amounts of real-world data besides the advanced algorithms it requires to run. Furthermore, the system must be capable of boosting the speed of database triggering searches and analysis. Therefore, the users need not to wait for a long time, instead of being prompt.

3.4. Accessibility and Security

This solution shall provide secure access to the database and the dataset files through proper authorization and admissions procedures and at distance query and analysis for authorized users.

3.5. Monitoring & Maintenance

This method reflects periodical monitoring of the database performance, resource use and system state, as well as regular maintenance functions like data backups, upgrades, and optimization projects.

4. System Design

4.1. System Functionality

4.1.1. Trend Analysis

Identifying patterns in movie release dates, genre popularity, and more.

4.1.2. Recommendation Generation

Creating movie suggestions based on user interests and watching history.

4.1.3. User Behavior Modeling

Analyzing user ratings and interactions to better understand watching habits and preferences.

4.2. Limitations

The drawbacks of the system include the absence of real-time updates, low range of interaction due to the fact that users mostly use predefined queries or scripts, and scalability issues on the managing of either large data sets or doing many analytical processes at the same time.

5. User Interaction.

5.1. Accessing the System

Authorized users with correct login parameters will work respectively with the system via web interface or command line interface. These interfaces are hosted on the Amazon Web Service (AWS) infrastructure.

5.2. Interact with the System:

The users of such database can learn from the data using either prepared queries or analytic scripts that include filtering criteria such as movie title, genre, release year, etc. The results can be seen in a visual form through the interface or saved for further analysis.

6. Database Design

6.1. Conceptual Database Design

6.1.1. Database Requirements for the Application System:

The proposed application system needs a solid well-structured database to process and keep movie information in the best possible way. The database should have great information processing capabilities maintaining the data integrity, consistency, and performance.

6.1.2. Entity Relationship Diagram (ERD)

The image seen here represents the logical view of the database that includes the Entity Relationship Diagram for the proposed database schema. The ERD displays entities (tables) and how they interrelate with each other inside the system. The primary entities include \`movies\`, \`credits_crew\`, \`credits_cast\`, \`production_companies\`, \`genre\`, \`keywords\`, and \`ratings\`. The partner relations between different entities are shown by the use of correct notations, e.g. one-to-many, and many-to-many. For example, \`movies\` entity has one-to-many relationship with \`credits_crew\` and \`credits_cast\` to cover situations, in which each movie can be attributed to multiple crew and cast members as well.

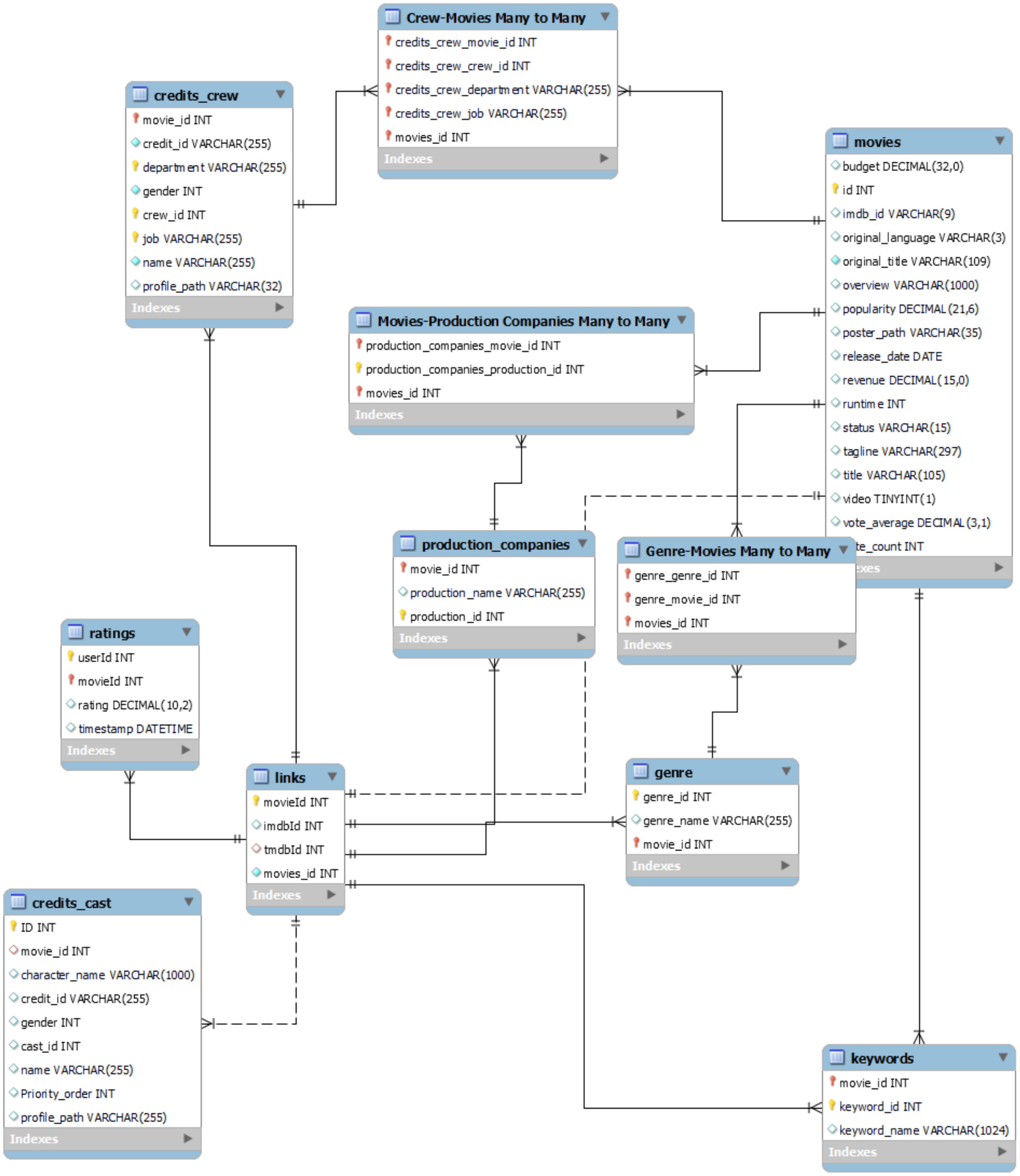


Fig - 6.1.2: ER Diagram of the database with the representation of all the relations with the Primary and Foreign keys.

6.1.3. Normalization and Key Representation

Normalization is the process of arranging data in a database to eliminate the duplicates and to better the data consistency. As the ERD provided follows the principles of normalization, no data redundancy is given, and it is stored as effectively as possible. Primary keys are represented in the ERD as unique identifiers for each entity. For example, the `movies` table has an `id` field marked as the primary key, ensuring that each movie record is uniquely identifiable. Foreign keys are used to establish relationships between entities. For instance, the `credits_crew` table has a `movie_id` field, which serves as a foreign key referencing the primary key (`id`) of the `movies` table. This relationship allows the system to associate crew members with specific movies.

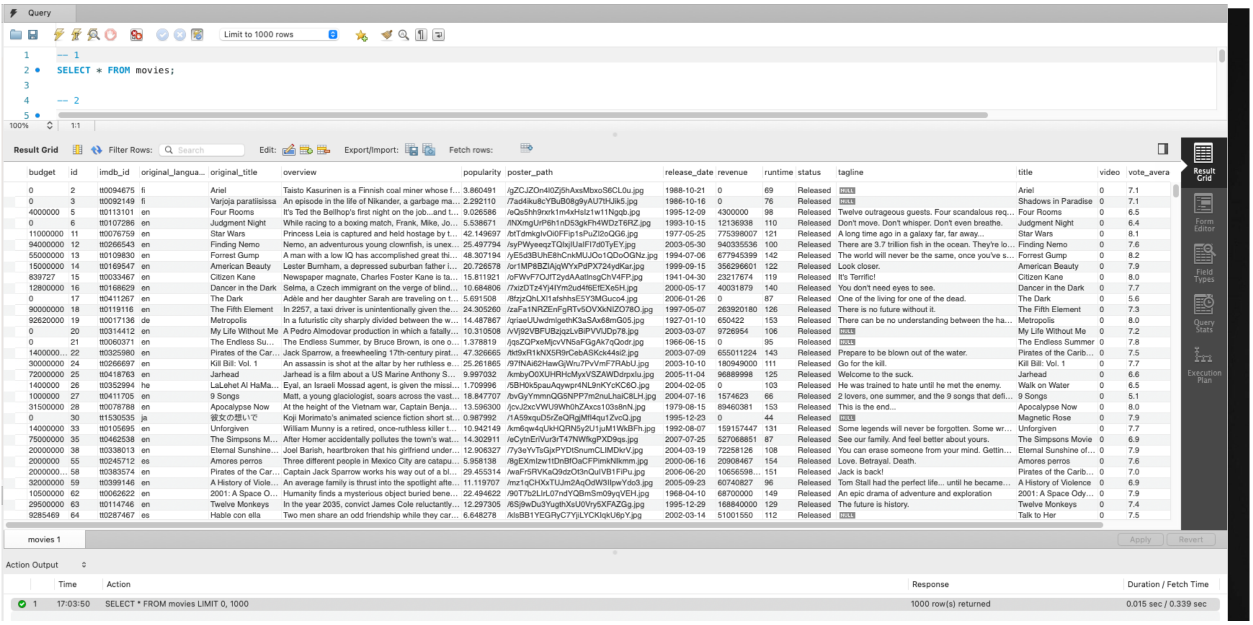
6.2. Functional Analysis

The developed application will blend smoothly for the purpose of providing a platform for storing movie related data; in this case, the efficiency of the system and the quality of service would be considered. The functionally different components that compose the CMS consist of data management, precise search, with a strong security system. The system will provide all the necessary capabilities that are useful in medium to high-end businesses. This is through use of user-friendly interfaces and effortless integration to make it a comprehensive and all-inclusive solution for handling movie data across different needs and settings.

Movie Management:

- Table Structure: `movies`, `credits_crew`, `credits_cast`, `production_companies`, `genre`, `keywords`
- Access Privileges: Administrative users may have full access for CRUD (Create, Read, Update, Delete) operations, while regular users may have read-only access.

- SQL Queries: Retrieve movie details, cast, and crew information, production companies, genres, and keywords associated with a movie.

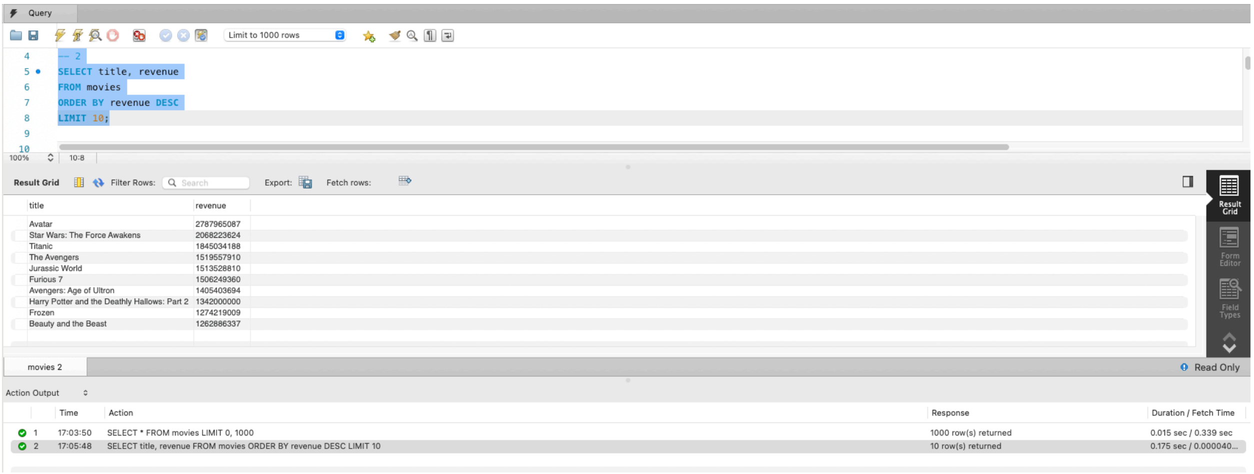


```
-- 1
SELECT * FROM movies;

-- 2
```

budget	id	imdb_id	original_langua...	original_title	overview	popularity	poster_path	release_date	revenue	runtime	status	tagline	title	video	vote_avera
0	2	tt0094675	fi	Ariel	Taisto Kasurinen is a Finnish coal miner whose f...	3.860491	/gZCJZOn4l0Zj5hAxsMbxoS6CL0u.jpg	1988-10-21	0	69	Released	NULL	Ariel	0	7.1
0	3	tt0092149	fi	Varjoja paratiisissa	An episode in the life of Nikander, a garbage ma...	2.292110	/7ad4iku8cYBuB08g9yAU7tHJik5.jpg	1986-10-16	0	76	Released	NULL	Shadows in Paradise	0	7.1
4000000	5	tt0113101	en	Four Rooms	It's Ted the Bellhop's first night on the job...and t...	9.026586	/eQs5hh9rxrk1m4xHslz1w11Ngqb.jpg	1995-12-09	4300000	98	Released	Twelve outrageous guests. Four scandalous req...	Four Rooms	0	6.5
0	6	tt0107286	en	Judgment Night	While racing to a boxing match, Frank, Mike, Jo...	5.538671	/lNXmgUrP6h1nD53gkFh4WDzT6RZ.jpg	1993-10-15	12136938	110	Released	Don't move. Don't whisper. Don't even breathe.	Judgment Night	0	6.4
11000000	11	tt0076759	en	Star Wars	Princess Leia is captured and held hostage by t...	42.149697	/btTdmkgIvOi0FFip1sPuZI2oQG6.jpg	1977-05-25	775398007	121	Released	A long time ago in a galaxy far, far away...	Star Wars	0	8.1
94000000	12	tt0266543	en	Finding Nemo	Nemo, an adventurous young clownfish, is unex...	25.497794	/syPWyeeqzTQIxjIUaIFI7d0TyEY.jpg	2003-05-30	940335536	100	Released	There are 3.7 trillion fish in the ocean. They're lo...	Finding Nemo	0	7.6
55000000	13	tt0109830	en	Forrest Gump	A man with a low IQ has accomplished great thi...	48.307194	/yE5d3BUhE8hCnkMUJOo1QDoOGNz.jpg	1994-07-06	677945399	142	Released	The world will never be the same, once you've s...	Forrest Gump	0	8.2
15000000	14	tt0169547	en	American Beauty	Lester Burnham, a depressed suburban father i...	20.726578	/or1MP8BZIAjqWYxPdPX724ydKar.jpg	1999-09-15	356296601	122	Released	Look closer.	American Beauty	0	7.9
839727	15	tt0033467	en	Citizen Kane	Newspaper magnate, Charles Foster Kane is ta...	15.811921	/oFWvF7OJfT2ydAAatlnsgChV4FP.jpg	1941-04-30	23217674	119	Released	It's Terrific!	Citizen Kane	0	8.0
12800000	16	tt0168629	en	Dancer in the Dark	Selma, a Czech immigrant on the verge of blind...	10.684806	/7xizDTz4Yj4IYm2ud4f6EfEXe5H.jpg	2000-05-17	40031879	140	Released	You don't need eyes to see.	Dancer in the Dark	0	7.7
0	17	tt0411267	en	The Dark	Adèle and her daughter Sarah are traveling on t...	5.691508	/8fzjzQhLXI1afshhsE5Y3MGuco4.jpg	2006-01-26	0	87	Released	One of the living for one of the dead.	The Dark	0	5.6
90000000	18	tt0119116	en	The Fifth Element	In 2257, a taxi driver is unintentionally given the...	24.305260	/zaFa1NRZEnFgRTv5OVXkNIZO78O.jpg	1997-05-07	263920180	126	Released	There is no future without it.	The Fifth Element	0	7.3
92620000	19	tt0017136	de	Metropolis	In a futuristic city sharply divided between the w...	14.487867	/qriaeUUwdmlgethK3aSAx68mG05.jpg	1927-01-10	650422	153	Released	There can be no understanding between the ha...	Metropolis	0	8.0
0	20	tt0314412	en	My Life Without Me	A Pedro Almodovar production in which a fatally...	10.310508	/vVj92VBFUBzjqzLvBiPVVlJDp78.jpg	2003-03-07	9726954	106	Released	NULL	My Life Without Me	0	7.2
0	21	tt0060371	en	The Endless Su...	The Endless Summer, by Bruce Brown, is one o...	1.378819	/jqsZQPxeMjcvVN5aFGgAk7qQodr.jpg	1966-06-15	0	95	Released	NULL	The Endless Summer	0	7.8
1400000...	22	tt0325980	en	Pirates of the Car...	Jack Sparrow, a freewheeling 17th-century pirat...	47.326665	/tkt9xR1kNX5R9rCebASKck44si2.jpg	2003-07-09	655011224	143	Released	Prepare to be blown out of the water.	Pirates of the Carib...	0	7.5
30000000	24	tt0266697	en	Kill Bill: Vol. 1	An assassin is shot at the altar by her ruthless e...	25.261865	/97fNAi62HawGjWru7PvVmF7RAbU.jpg	2003-10-10	180949000	111	Released	Go for the kill.	Kill Bill: Vol. 1	0	7.7
72000000	25	tt0418763	en	Jarhead	Jarhead is a film about a US Marine Anthony S...	9.997032	/kmbyO0XUHRHcMyxVSZAWDdrpxlu.jpg	2005-11-04	96889998	125	Released	Welcome to the suck.	Jarhead	0	6.6
1400000	26	tt0352994	he	LaLehet Al HaMa...	Eyal, an Israeli Mossad agent, is given the missi...	1.709996	/5BH0k5pauAqywpr4NL9nKYcKC6O.jpg	2004-02-05	0	103	Released	He was trained to hate until he met the enemy.	Walk on Water	0	6.5
1000000	27	tt0411705	en	9 Songs	Matt, a young glaciologist, soars across the vast...	18.847707	/bvGyYmmnQG5NPP7m2nuLhaiC8LH.jpg	2004-07-16	1574623	66	Released	2 lovers, one summer, and the 9 songs that defi...	9 Songs	0	5.1
31500000	28	tt0078788	en	Apocalypse Now	At the height of the Vietnam war, Captain Benja...	13.596300	/jcvJ2xcVWU9Wh0hZAxcs103s8nN.jpg	1979-08-15	89460381	153	Released	This is the end...	Apocalypse Now	0	8.0
0	30	tt1530535	ja	彼女の想いで	Koji Morimato's animated science fiction short st...	0.987992	/1A59xquD5rZeQRgjMfl4qu1ZvcQ.jpg	1995-12-23	0	44	Released	NULL	Magnetic Rose	0	7.9
14000000	33	tt0105695	en	Unforgiven	William Munny is a retired, once-ruthless killer t...	10.942149	/km6qw4qUkHQRN5y2U1juM1WkBFh.jpg	1992-08-07	159157447	131	Released	Some legends will never be forgotten. Some wr...	Unforgiven	0	7.7
75000000	35	tt0462538	en	The Simpsons M...	After Homer accidentally pollutes the town's wat...	14.302911	/eCytnEriVur3rT47NWfkgPXD9qs.jpg	2007-07-25	527068851	87	Released	See our family. And feel better about yours.	The Simpsons Movie	0	6.9
20000000	38	tt0338013	en	Eternal Sunshine...	Joel Barish, heartbroken that his girlfriend under...	12.906327	/7y3eYvTsGjxPYDtSnumCLIMDkrV.jpg	2004-03-19	72258126	108	Released	You can erase someone from your mind. Gettin...	Eternal Sunshine of...	0	7.9
2000000	55	tt0245712	es	Amores perros	Three different people in Mexico City are catapu...	5.958138	/8gEXmlzw1tDnBfOaCFPimkNlkmm.jpg	2000-06-16	20908467	154	Released	Love. Betrayal. Death.	Amores perros	0	7.6
2000000...	58	tt0383574	en	Pirates of the Car...	Captain Jack Sparrow works his way out of a bl...	29.455314	/waFr5RVKaQ9dzOt3nQuIVB1FiPu.jpg	2006-06-20	10656598...	151	Released	Jack is back!	Pirates of the Carib...	0	7.0
32000000	59	tt0399146	en	A History of Viole...	An average family is thrust into the spotlight afte...	11.119707	/mz1qCHXxTUJm2AqOdW3IlpwYdo3.jpg	2005-09-23	60740827	96	Released	Tom Stall had the perfect life... until he became...	A History of Violence	0	6.9
10500000	62	tt0062622	en	2001: A Space O...	Humanity finds a mysterious object buried bene...	22.494622	/90T7b2LIrL07ndYQBmSm09yqVEH.jpg	1968-04-10	68700000	149	Released	An epic drama of adventure and exploration	2001: A Space Ody...	0	7.9
29500000	63	tt0114746	en	Twelve Monkeys	In the year 2035, convict James Cole reluctantly...	12.297305	/6Sj9wDu3YugthXsU0Vry5XFAZGg.jpg	1995-12-29	168840000	129	Released	The future is history.	Twelve Monkeys	0	7.4
9285469	64	tt0287467	es	Hable con ella	Two men share an odd friendship while they car...	6.648278	/klsBB1YEGRyC7YjiLYCKlqkU6pY.jpg	2002-03-14	51001550	112	Released	NULL	Talk to Her	0	7.5

	Time	Action	Response	Duration / Fetch Time
✓ 1	17:03:50	SELECT * FROM movies LIMIT 0, 1000	1000 row(s) returned	0.015 sec / 0.339 sec

Fig – 6.2.1: SQL query

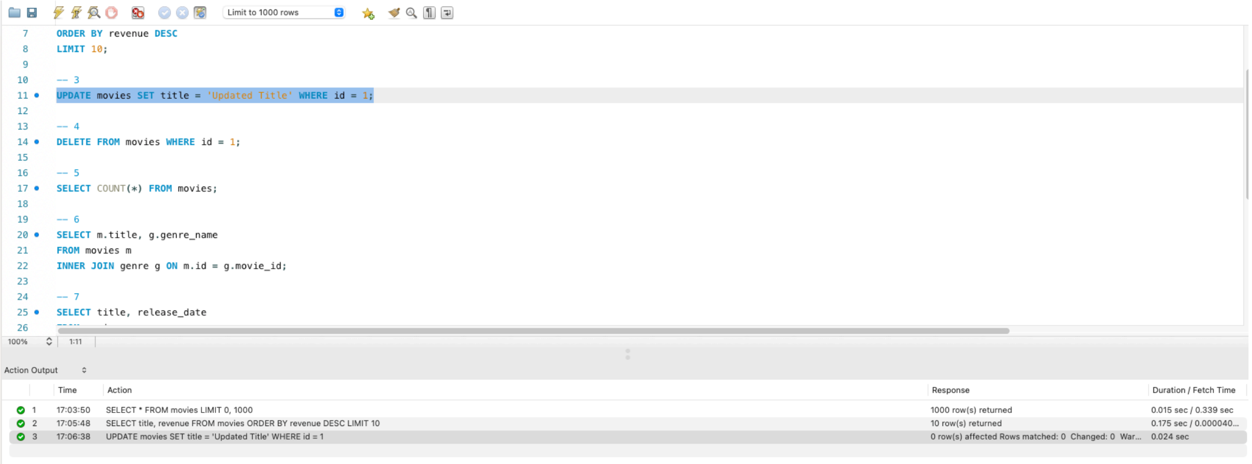


```
-- 2
SELECT title, revenue
FROM movies
ORDER BY revenue DESC
LIMIT 10;
```

title	revenue
Avatar	2787965087
Star Wars: The Force Awakens	2068223624
Titanic	1845034188
The Avengers	1519557910
Jurassic World	1513528810
Furious 7	1506249360
Avengers: Age of Ultron	1405403694
Harry Potter and the Deathly Hallows: Part 2	1342000000
Frozen	1274219009
Beauty and the Beast	1262886337

	Time	Action	Response	Duration / Fetch Time
✓ 1	17:03:50	SELECT * FROM movies LIMIT 0, 1000	1000 row(s) returned	0.015 sec / 0.339 sec
✓ 2	17:05:48	SELECT title, revenue FROM movies ORDER BY revenue DESC LIMIT 10	10 row(s) returned	0.175 sec / 0.000040...

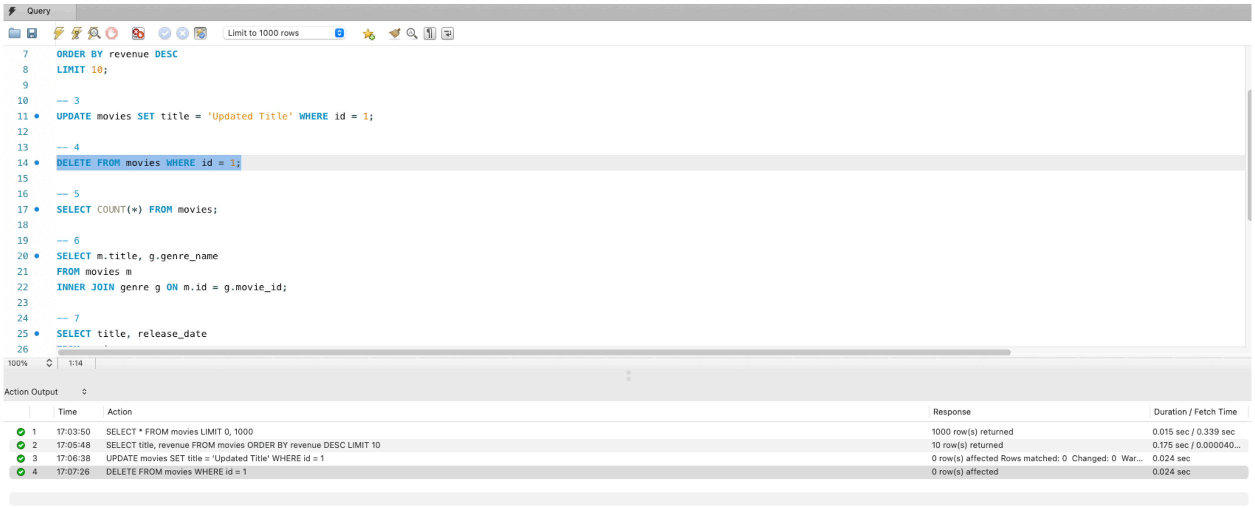
Fig – 6.2.2: SQL query



```
ORDER BY revenue DESC
LIMIT 10;

-- 3
UPDATE movies SET title = 'Updated Title' WHERE id = 1;

-- 4
DELETE FROM movies WHERE id = 1;

-- 5
SELECT COUNT(*) FROM movies;

-- 6
SELECT m.title, g.genre_name
FROM movies m
INNER JOIN genre g ON m.id = g.movie_id;

-- 7
SELECT title, release_date
```

```
   Time      Action                                                        Response                                               Duration / Fetch Time
1  17:03:50  SELECT * FROM movies LIMIT 0, 1000                            1000 row(s) returned                                   0.015 sec / 0.339 sec
2  17:05:48  SELECT title, revenue FROM movies ORDER BY revenue DESC LIMIT 10  10 row(s) returned                                 0.175 sec / 0.000040...
3  17:06:38  UPDATE movies SET title = 'Updated Title' WHERE id = 1        0 row(s) affected Rows matched: 0  Changed: 0  War...  0.024 sec
```

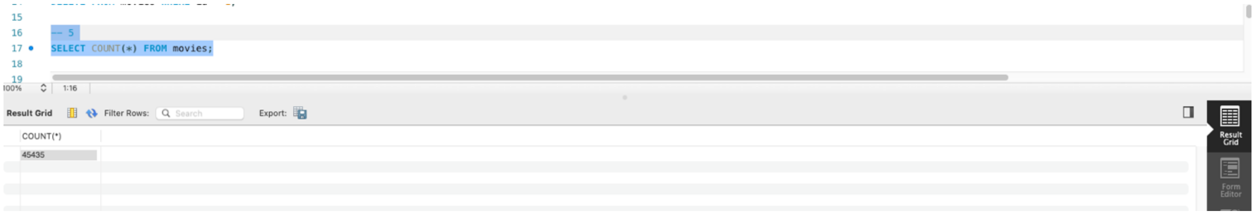
Fig – 6.2.3: SQL query



```
ORDER BY revenue DESC
LIMIT 10;

-- 3
UPDATE movies SET title = 'Updated Title' WHERE id = 1;

-- 4
DELETE FROM movies WHERE id = 1;

-- 5
SELECT COUNT(*) FROM movies;

-- 6
SELECT m.title, g.genre_name
FROM movies m
INNER JOIN genre g ON m.id = g.movie_id;

-- 7
SELECT title, release_date
```

```
   Time      Action                                                        Response                                               Duration / Fetch Time
1  17:03:50  SELECT * FROM movies LIMIT 0, 1000                            1000 row(s) returned                                   0.015 sec / 0.339 sec
2  17:05:48  SELECT title, revenue FROM movies ORDER BY revenue DESC LIMIT 10  10 row(s) returned                                 0.175 sec / 0.000040...
3  17:06:38  UPDATE movies SET title = 'Updated Title' WHERE id = 1        0 row(s) affected Rows matched: 0  Changed: 0  War...  0.024 sec
4  17:07:26  DELETE FROM movies WHERE id = 1                               0 row(s) affected                                      0.024 sec
```

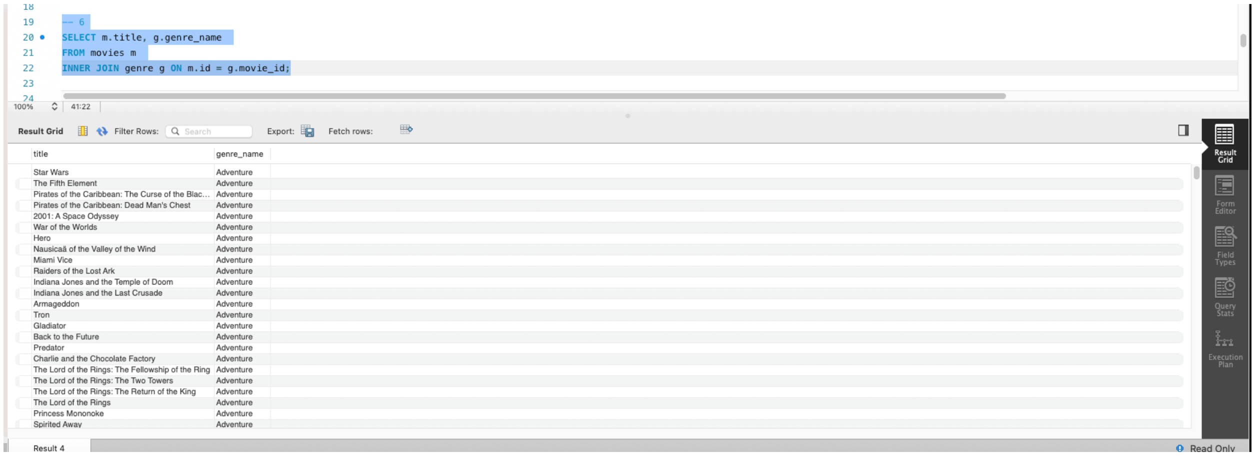
Fig – 6.2.4: SQL query



```
-- 5
SELECT COUNT(*) FROM movies;
```

```
COUNT(*)
45435
```

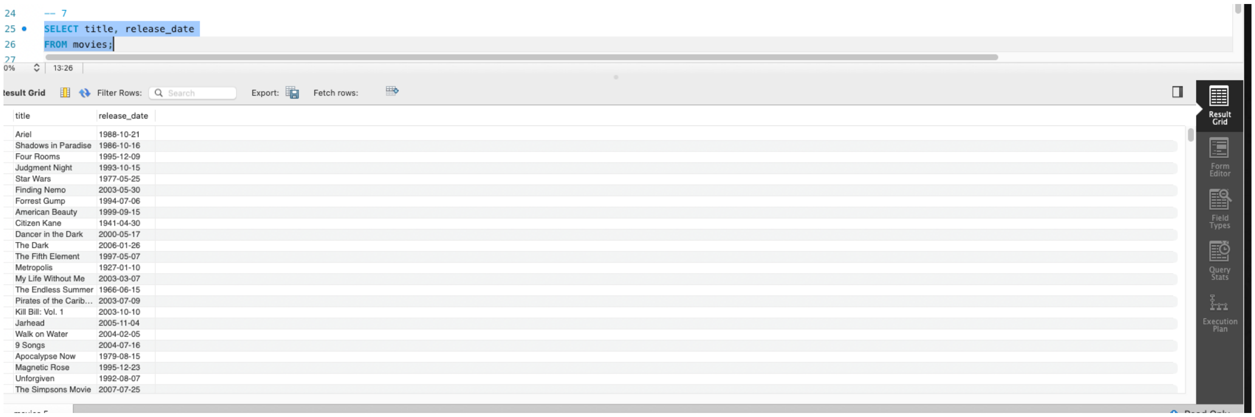
Fig – 6.2.5: SQL query



```sql
-- 6
SELECT m.title, g.genre_name
FROM movies m
INNER JOIN genre g ON m.id = g.movie_id;
```

title	genre_name
Star Wars	Adventure
The Fifth Element	Adventure
Pirates of the Caribbean: The Curse of the Blac…	Adventure
Pirates of the Caribbean: Dead Man's Chest	Adventure
2001: A Space Odyssey	Adventure
War of the Worlds	Adventure
Hero	Adventure
Nausicaä of the Valley of the Wind	Adventure
Miami Vice	Adventure
Raiders of the Lost Ark	Adventure
Indiana Jones and the Temple of Doom	Adventure
Indiana Jones and the Last Crusade	Adventure
Armageddon	Adventure
Tron	Adventure
Gladiator	Adventure
Back to the Future	Adventure
Predator	Adventure
Charlie and the Chocolate Factory	Adventure
The Lord of the Rings: The Fellowship of the Ring	Adventure
The Lord of the Rings: The Two Towers	Adventure
The Lord of the Rings: The Return of the King	Adventure
The Lord of the Rings	Adventure
Princess Mononoke	Adventure
Spirited Away	Adventure

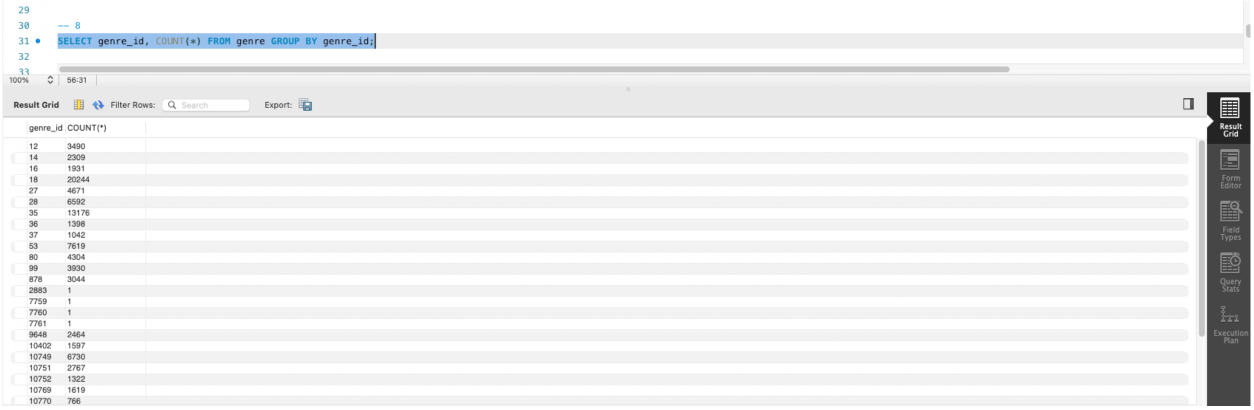
Fig – 6.2.6: SQL query



```sql
-- 7
SELECT title, release_date
FROM movies;
```

title	release_date
Ariel	1988-10-21
Shadows in Paradise	1986-10-16
Four Rooms	1995-12-09
Judgment Night	1993-10-15
Star Wars	1977-05-25
Finding Nemo	2003-05-30
Forrest Gump	1994-07-06
American Beauty	1999-09-15
Citizen Kane	1941-04-30
Dancer in the Dark	2000-05-17
The Dark	2006-01-26
The Fifth Element	1997-05-07
Metropolis	1927-01-10
My Life Without Me	2003-03-07
The Endless Summer	1966-06-15
Pirates of the Carib…	2003-07-09
Kill Bill: Vol. 1	2003-10-10
Jarhead	2005-11-04
Walk on Water	2004-02-05
9 Songs	2004-07-16
Apocalypse Now	1979-08-15
Magnetic Rose	1995-12-23
Unforgiven	1992-08-07
The Simpsons Movie	2007-07-25

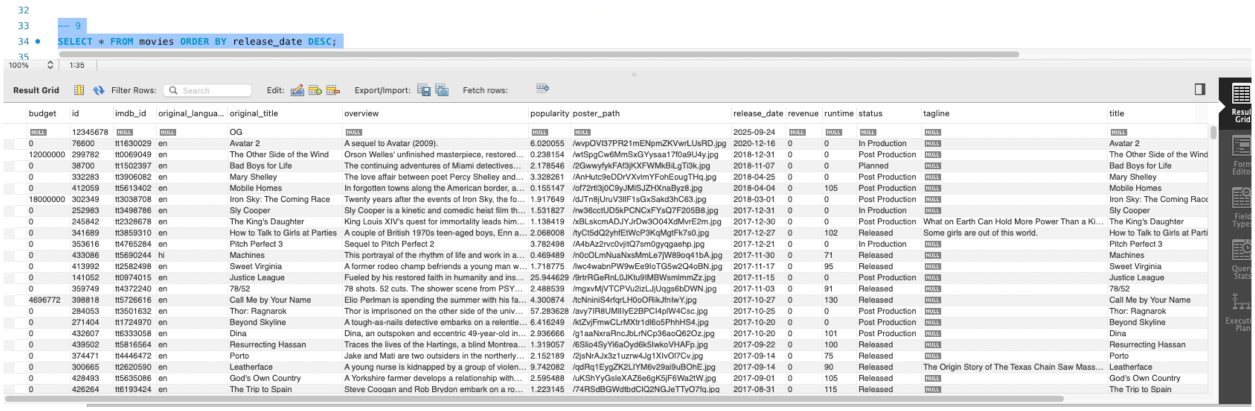
Fig – 6.2.7: SQL query



```sql
-- 8
SELECT genre_id, COUNT(*) FROM genre GROUP BY genre_id;
```

genre_id	COUNT(*)
12	3490
14	2309
16	1931
18	20244
27	4671
28	6592
35	13176
36	1398
37	1042
53	7619
80	4304
99	3930
878	3044
2883	1
7759	1
7760	1
7761	1
9648	2464
10402	1597
10749	6730
10751	2767
10752	1322
10769	1619
10770	766

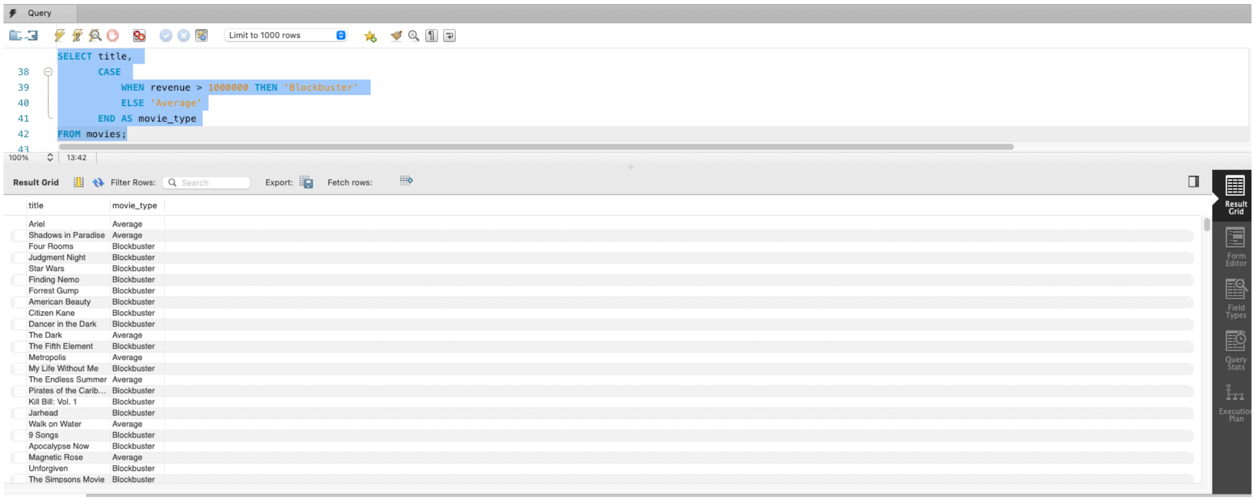
Fig – 6.2.8: SQL query



```sql
-- 9
SELECT * FROM movies ORDER BY release_date DESC;
```

budget	id	imdb_id	original_langua...	original_title	overview	popularity	poster_path	release_date	revenue	runtime	status	tagline	title
NULL	12345678	NULL	NULL	OG	NULL	NULL	NULL	2025-09-24	NULL	NULL	NULL	NULL	NULL
0	76600	tt1630029	en	Avatar 2	A sequel to Avatar (2009).	6.020055	/wvpOVI37PR21mENpmZKVwrLUsRD.jpg	2020-12-16	0	0	In Production	NULL	Avatar 2
12000000	299782	tt0069049	en	The Other Side of the Wind	Orson Welles' unfinished masterpiece, restored...	0.238154	/wtSpgCw6MmSxGYysaa17f0a9U4y.jpg	2018-12-31	0	0	Post Production	NULL	The Other Side of the Wind
0	38700	tt1502397	en	Bad Boys for Life	The continuing adventures of Miami detectives...	2.178546	/2GwwyfykFAf3jKXFWMkBiLgTi3k.jpg	2018-11-07	0	0	Planned	NULL	Bad Boys for Life
0	332283	tt3906082	en	Mary Shelley	The love affair between poet Percy Shelley and...	3.328261	/AnHutc9eDDrVXvlmYFohEougTHq.jpg	2018-04-25	0	0	Post Production	NULL	Mary Shelley
0	412059	tt5613402	en	Mobile Homes	In forgotten towns along the American border, a...	0.155147	/of72rtl3j0C9yJMlSJZHXnaByz8.jpg	2018-04-04	0	105	Post Production	NULL	Mobile Homes
18000000	302349	tt3038708	en	Iron Sky: The Coming Race	Twenty years after the events of Iron Sky, the fo...	1.917649	/dJTn8jUruV3llF1sGxSakd3hC63.jpg	2018-03-01	0	0	Post Production	NULL	Iron Sky: The Coming Race
0	252983	tt3498786	en	Sly Cooper	Sly Cooper is a kinetic and comedic heist film th...	1.531827	/rw36cctUD5kPCNCxFYsQ7F205B8.jpg	2017-12-31	0	0	In Production	NULL	Sly Cooper
0	245842	tt2328678	en	The King's Daughter	King Louis XIV's quest for immortality leads him...	1.138419	/xBLskcmADJYJrDw3O04XdMvrE2m.jpg	2017-12-30	0	0	Post Production	What on Earth Can Hold More Power Than a Ki...	The King's Daughter
0	341689	tt3859310	en	How to Talk to Girls at Parties	A couple of British 1970s teen-aged boys, Enn a...	2.068008	/tyCt5dQ2yhfEtWcP3KqMgtFk7s0.jpg	2017-12-27	0	102	Released	Some girls are out of this world.	How to Talk to Girls at Parti
0	353616	tt4765284	en	Pitch Perfect 3	Sequel to Pitch Perfect 2	3.782498	/A4bAz2rvc0vjitQ7sm0gyqgaehp.jpg	2017-12-21	0	0	In Production	NULL	Pitch Perfect 3
0	433086	tt5690244	hi	Machines	This portrayal of the rhythm of life and work in a...	0.469489	/n0cOLmNuaNxsMmLe7jW89oq41bA.jpg	2017-11-30	0	71	Released	NULL	Machines
0	413992	tt2582498	en	Sweet Virginia	A former rodeo champ befriends a young man w...	1.718775	/lwc4wabnPW9wEe9loTG5w2Q4oBN.jpg	2017-11-17	0	95	Released	NULL	Sweet Virginia
0	141052	tt0974015	en	Justice League	Fueled by his restored faith in humanity and ins...	25.944629	/9rtrRGeRnL0JKtu9IMBWsmlmmZz.jpg	2017-11-15	0	0	Post Production	NULL	Justice League
0	359749	tt4372240	en	78/52	78 shots. 52 cuts. The shower scene from PSY...	2.488539	/mgxvMjVTCPVu2izLJjUqgs6bDWN.jpg	2017-11-03	0	91	Released	NULL	78/52
4696772	398818	tt5726616	en	Call Me by Your Name	Elio Perlman is spending the summer with his fa...	4.300874	/tcNniniS4rfqrLH0oORikJfnlwY.jpg	2017-10-27	0	130	Released	NULL	Call Me by Your Name
0	284053	tt3501632	en	Thor: Ragnarok	Thor is imprisoned on the other side of the univ...	57.283628	/avy7IR8UMlIlyE2BPCl4plW4Csc.jpg	2017-10-25	0	0	Post Production	NULL	Thor: Ragnarok
0	271404	tt1724970	en	Beyond Skyline	A tough-as-nails detective embarks on a relentle...	6.416249	/ktZvjFmwCLrMXtr1dl6o5PhhHS4.jpg	2017-10-20	0	0	Post Production	NULL	Beyond Skyline
0	432607	tt6333058	en	Dina	Dina, an outspoken and eccentric 49-year-old in...	2.936666	/g1aaNxraRncJbLrNCp36aoQ62Oz.jpg	2017-10-20	0	101	Post Production	NULL	Dina
0	439502	tt5816564	en	Resurrecting Hassan	Traces the lives of the Hartings, a blind Montrea...	1.319057	/6Slio4SyYi6aOyd6k5IwkoVHAFp.jpg	2017-09-22	0	100	Released	NULL	Resurrecting Hassan
0	374471	tt4446472	en	Porto	Jake and Mati are two outsiders in the northerly...	2.152189	/2jsNrAJx3z1uzrw4Jg1XlvOl7Cv.jpg	2017-09-14	0	75	Released	NULL	Porto
0	300665	tt2620590	en	Leatherface	A young nurse is kidnapped by a group of violen...	9.742082	/qdRq1EygZK2LIYM6v29ai9uBOhE.jpg	2017-09-14	0	90	Released	The Origin Story of The Texas Chain Saw Mass...	Leatherface
0	428493	tt5635086	en	God's Own Country	A Yorkshire farmer develops a relationship with...	2.595488	/uKShYyGsleXAZ6e6gK5jF6Wa2tW.jpg	2017-09-01	0	105	Released	NULL	God's Own Country
0	426264	tt6193424	en	The Trip to Spain	Steve Coogan and Rob Brydon embark on a ro...	1.223145	/74RSdBGWdtbdCIQ2NGJeTTyO7fq.jpg	2017-08-31	0	115	Released	NULL	The Trip to Spain

Fig – 6.2.9: SQL query

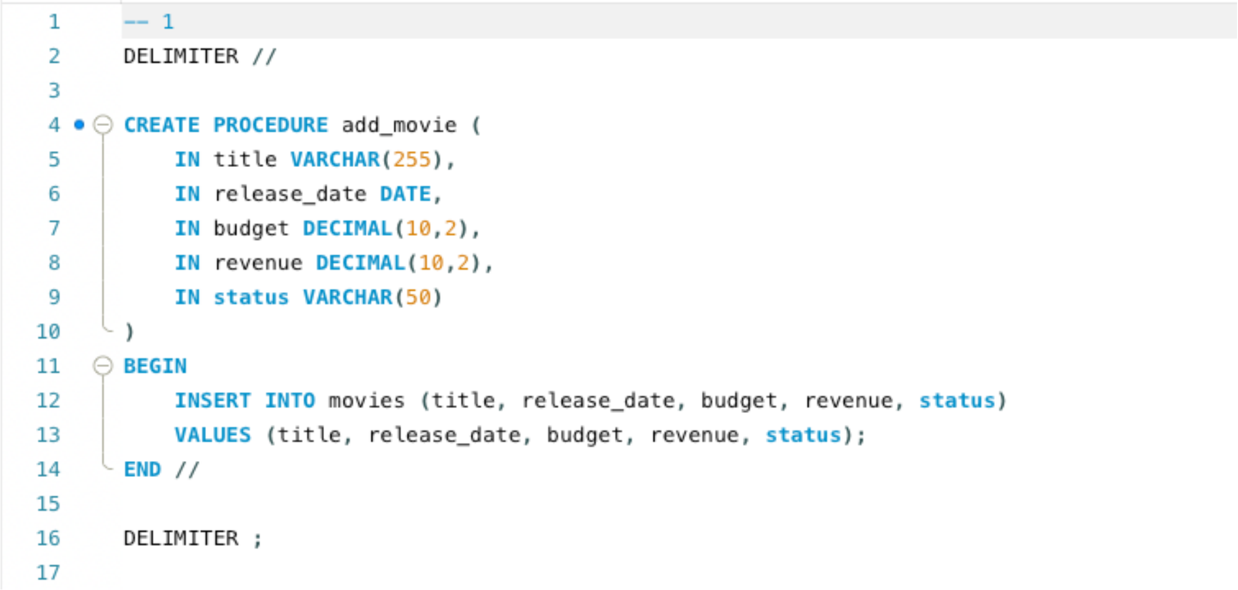


```sql
SELECT title,
       CASE
           WHEN revenue > 1000000 THEN 'Blockbuster'
           ELSE 'Average'
       END AS movie_type
FROM movies;
```

title	movie_type
Ariel	Average
Shadows in Paradise	Average
Four Rooms	Blockbuster
Judgment Night	Blockbuster
Star Wars	Blockbuster
Finding Nemo	Blockbuster
Forrest Gump	Blockbuster
American Beauty	Blockbuster
Citizen Kane	Blockbuster
Dancer in the Dark	Blockbuster
The Dark	Average
The Fifth Element	Blockbuster
Metropolis	Average
My Life Without Me	Blockbuster
The Endless Summer	Average
Pirates of the Carib...	Blockbuster
Kill Bill: Vol. 1	Blockbuster
Jarhead	Blockbuster
Walk on Water	Average
9 Songs	Blockbuster
Apocalypse Now	Blockbuster
Magnetic Rose	Average
Unforgiven	Blockbuster
The Simpsons Movie	Blockbuster

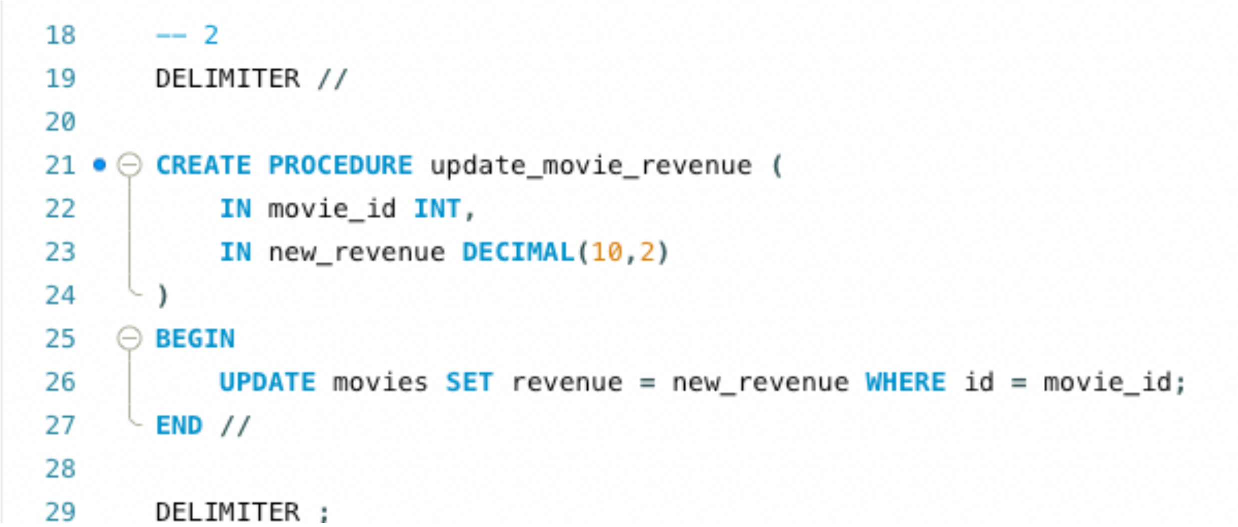
Fig – 6.2.10: SQL query

- Triggers: Automatically update related tables when a new movie is added or modified.
- Stored Procedures: Encapsulate complex business logic for efficient data processing.



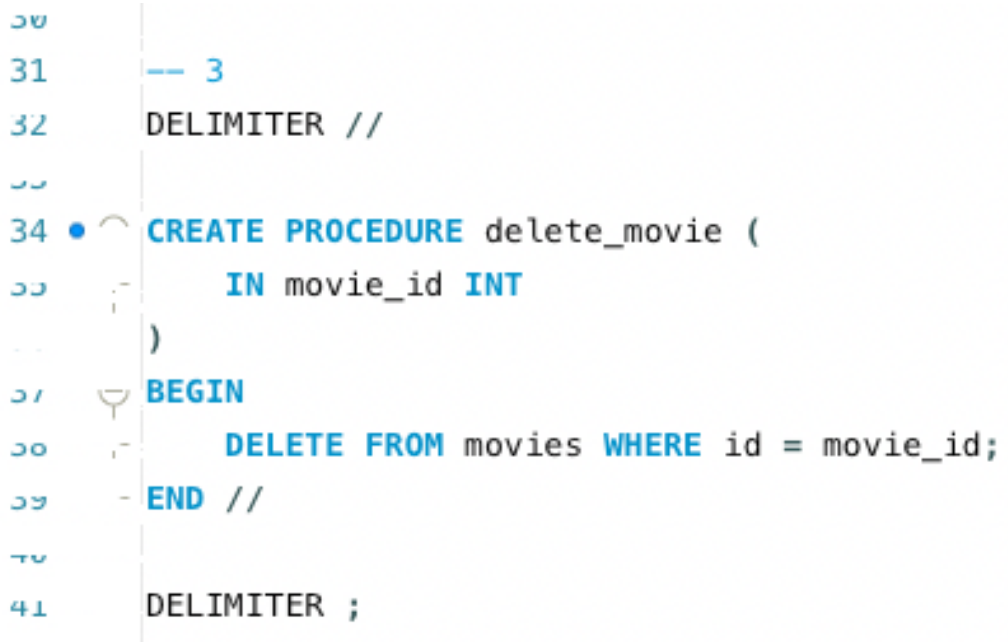
```
-- 1
DELIMITER //

CREATE PROCEDURE add_movie (
    IN title VARCHAR(255),
    IN release_date DATE,
    IN budget DECIMAL(10,2),
    IN revenue DECIMAL(10,2),
    IN status VARCHAR(50)
)
BEGIN
    INSERT INTO movies (title, release_date, budget, revenue, status)
    VALUES (title, release_date, budget, revenue, status);
END //

DELIMITER ;
```

Fig – 6.2.11: SQL Stored Procedure



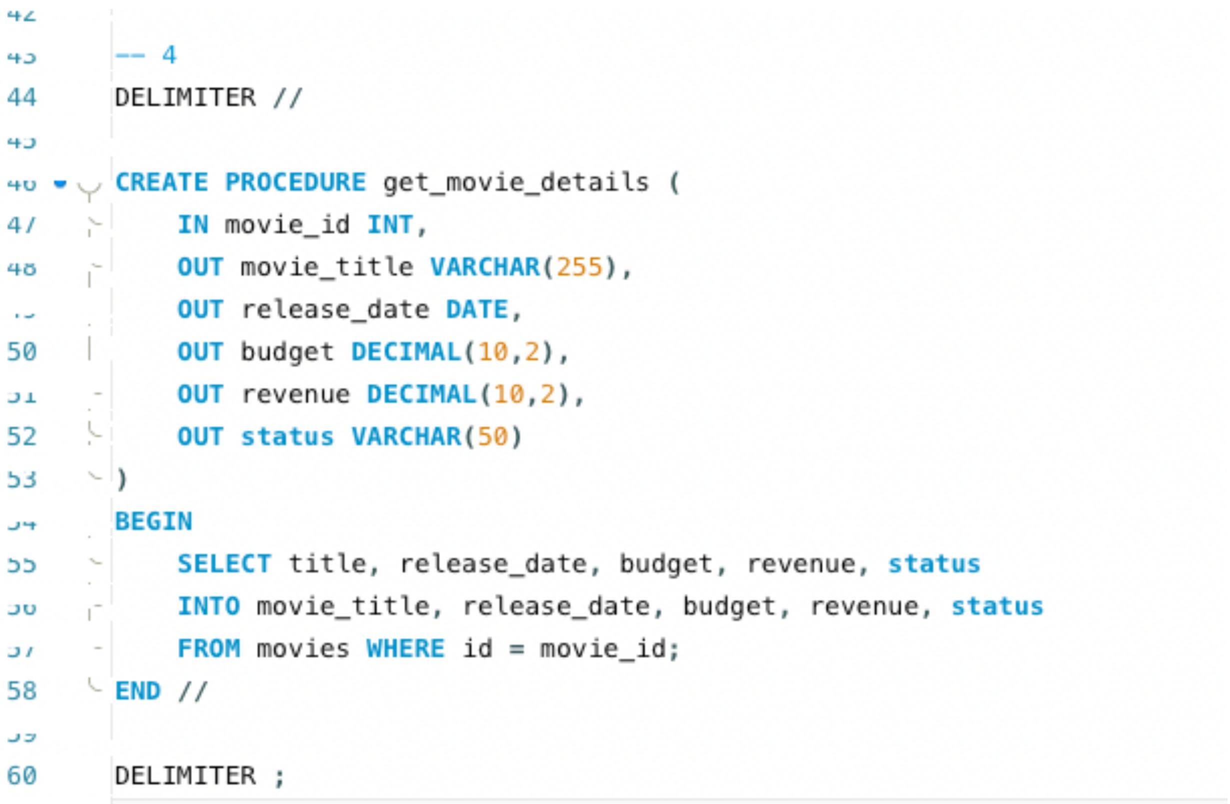
```
-- 2
DELIMITER //

CREATE PROCEDURE update_movie_revenue (
    IN movie_id INT,
    IN new_revenue DECIMAL(10,2)
)
BEGIN
    UPDATE movies SET revenue = new_revenue WHERE id = movie_id;
END //

DELIMITER ;
```

Fig – 6.2.12: SQL Stored Procedure



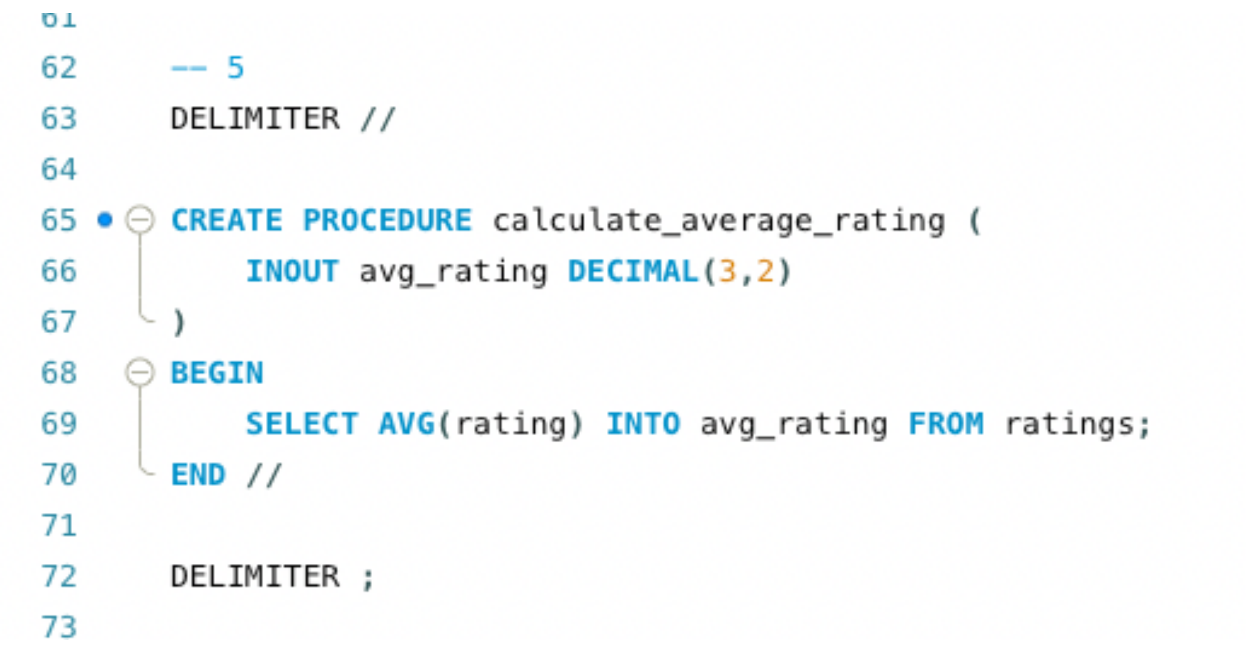
```
-- 3
DELIMITER //

CREATE PROCEDURE delete_movie (
    IN movie_id INT
)
BEGIN
    DELETE FROM movies WHERE id = movie_id;
END //

DELIMITER ;
```

Fig – 6.2.13: SQL Stored Procedure



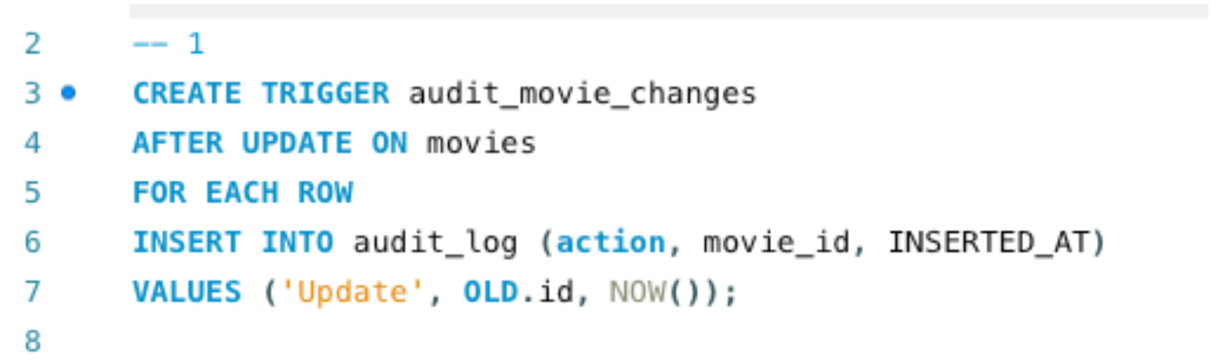
```
-- 4
DELIMITER //

CREATE PROCEDURE get_movie_details (
    IN movie_id INT,
    OUT movie_title VARCHAR(255),
    OUT release_date DATE,
    OUT budget DECIMAL(10,2),
    OUT revenue DECIMAL(10,2),
    OUT status VARCHAR(50)
)
BEGIN
    SELECT title, release_date, budget, revenue, status
    INTO movie_title, release_date, budget, revenue, status
    FROM movies WHERE id = movie_id;
END //

DELIMITER ;
```

Fig – 6.2.14: SQL Stored Procedure



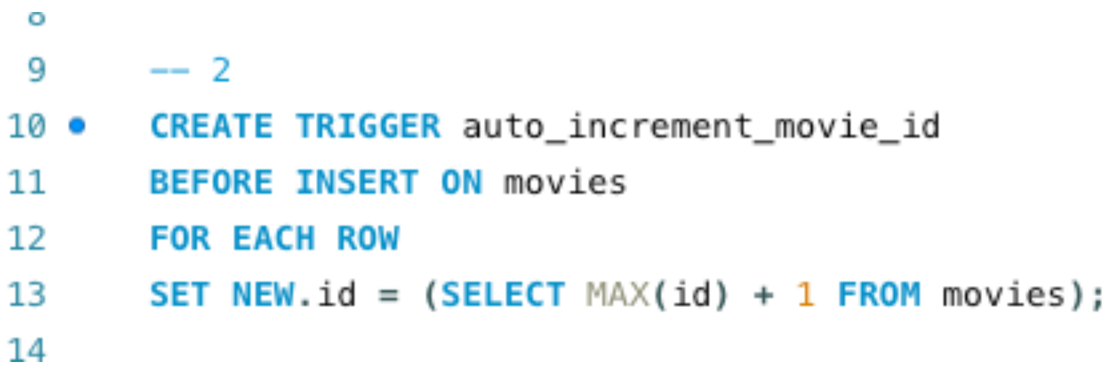
```
-- 5
DELIMITER //

CREATE PROCEDURE calculate_average_rating (
    INOUT avg_rating DECIMAL(3,2)
)
BEGIN
    SELECT AVG(rating) INTO avg_rating FROM ratings;
END //

DELIMITER ;
```

Fig – 6.2.15: SQL Stored Procedure



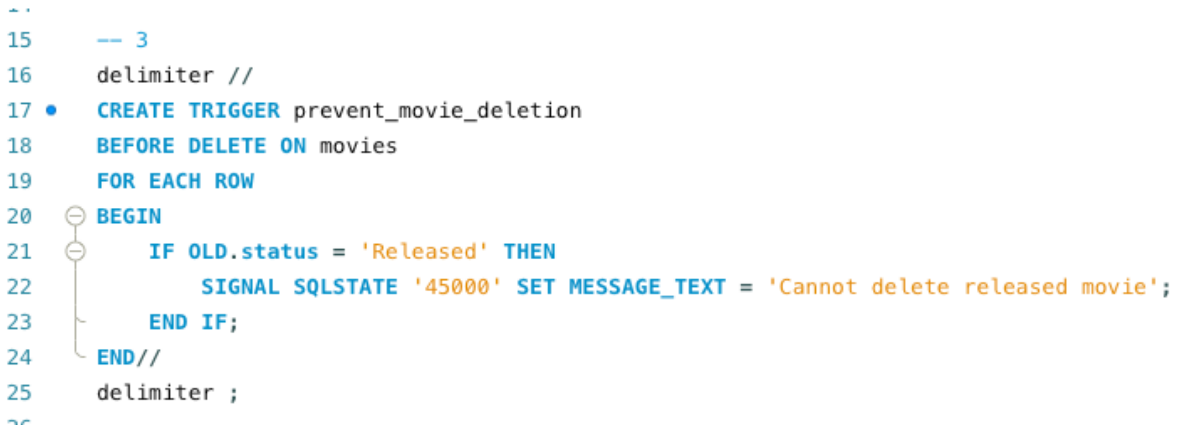
```
-- 1
CREATE TRIGGER audit_movie_changes
AFTER UPDATE ON movies
FOR EACH ROW
INSERT INTO audit_log (action, movie_id, INSERTED_AT)
VALUES ('Update', OLD.id, NOW());
```

Fig – 6.2.16: SQL Trigger



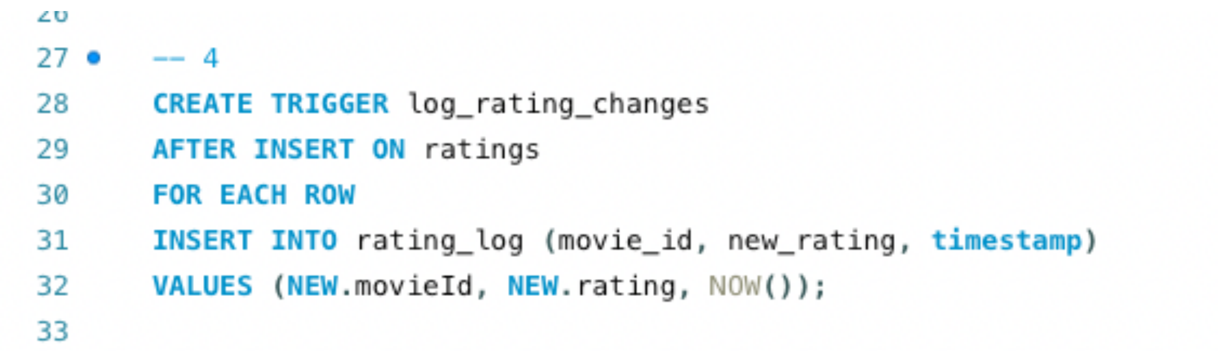
```
-- 2
CREATE TRIGGER auto_increment_movie_id
BEFORE INSERT ON movies
FOR EACH ROW
SET NEW.id = (SELECT MAX(id) + 1 FROM movies);
```

Fig – 6.2.17: SQL Trigger



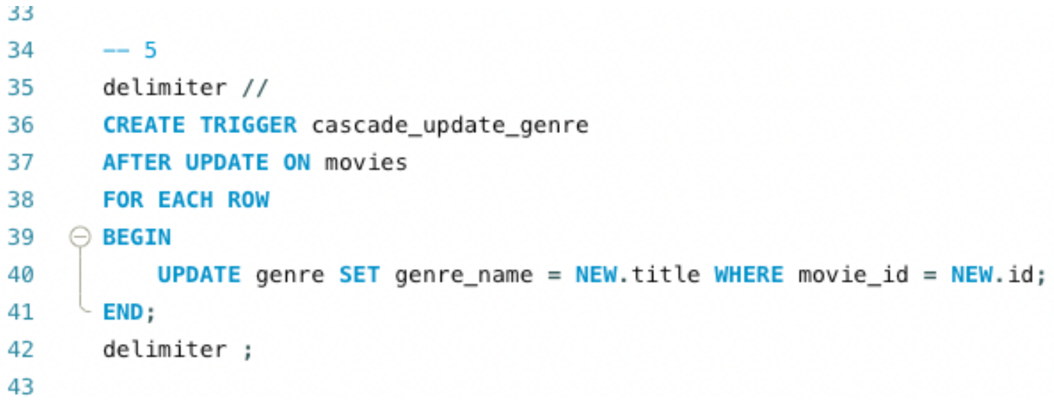
```
-- 3
delimiter //
CREATE TRIGGER prevent_movie_deletion
BEFORE DELETE ON movies
FOR EACH ROW
BEGIN
    IF OLD.status = 'Released' THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Cannot delete released movie';
    END IF;
END//
delimiter ;
```

Fig – 6.2.18: SQL Trigger



```
-- 4
CREATE TRIGGER log_rating_changes
AFTER INSERT ON ratings
FOR EACH ROW
INSERT INTO rating_log (movie_id, new_rating, timestamp)
VALUES (NEW.movieId, NEW.rating, NOW());
```

Fig – 6.2.19: SQL Trigger



```
-- 5
delimiter //
CREATE TRIGGER cascade_update_genre
AFTER UPDATE ON movies
FOR EACH ROW
BEGIN
    UPDATE genre SET genre_name = NEW.title WHERE movie_id = NEW.id;
END;
delimiter ;
```

Fig – 6.2.20: SQL Trigger

The database interactions for each component will be the execution of SQL queries, triggers, and stored procedures to retrieve data, insert data, update data, and delete data in the conformity to the requirements of the database interface. Privilege usage will depend solely on user designers as to his position and responsibilities, which guarantees information safety and authenticity.

7. Implementation

7.1. Database Migration to AWS:

Step by Step Guide for Database migration to AWS is s following:

1. Create an AWS account and browse to the AWS Management Console.

2. Set up an RDS (Relational Database Service) instance, specifying the database engine, instance size, storage, and security settings.

3. Prepare the TMDB dataset by obtaining the files and preparing them for cleanliness and analytical appropriateness.

This is processed through python and the specific codes are uploaded to GitHub.

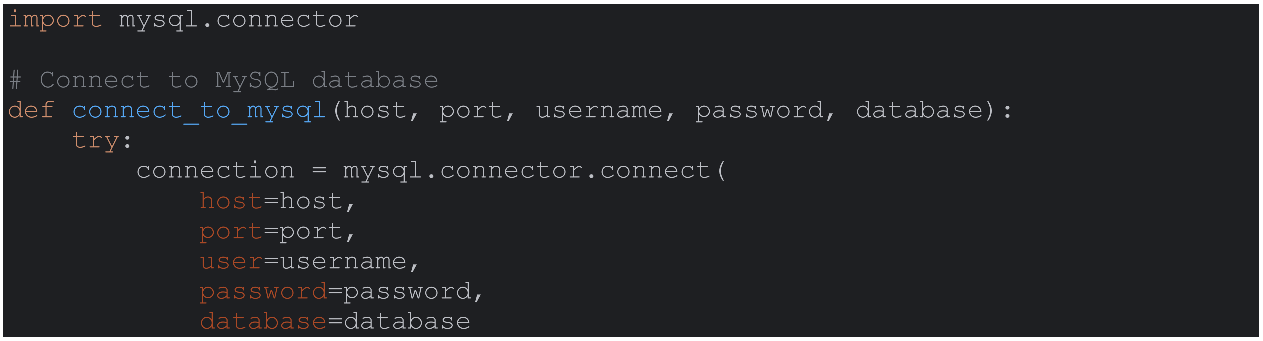
4. Transfer the data from CSV files to MYSQL workbench through python (codes are specifically uploaded to GitHub).

5. Relate every table to the other tables using Primary and Foreign keys

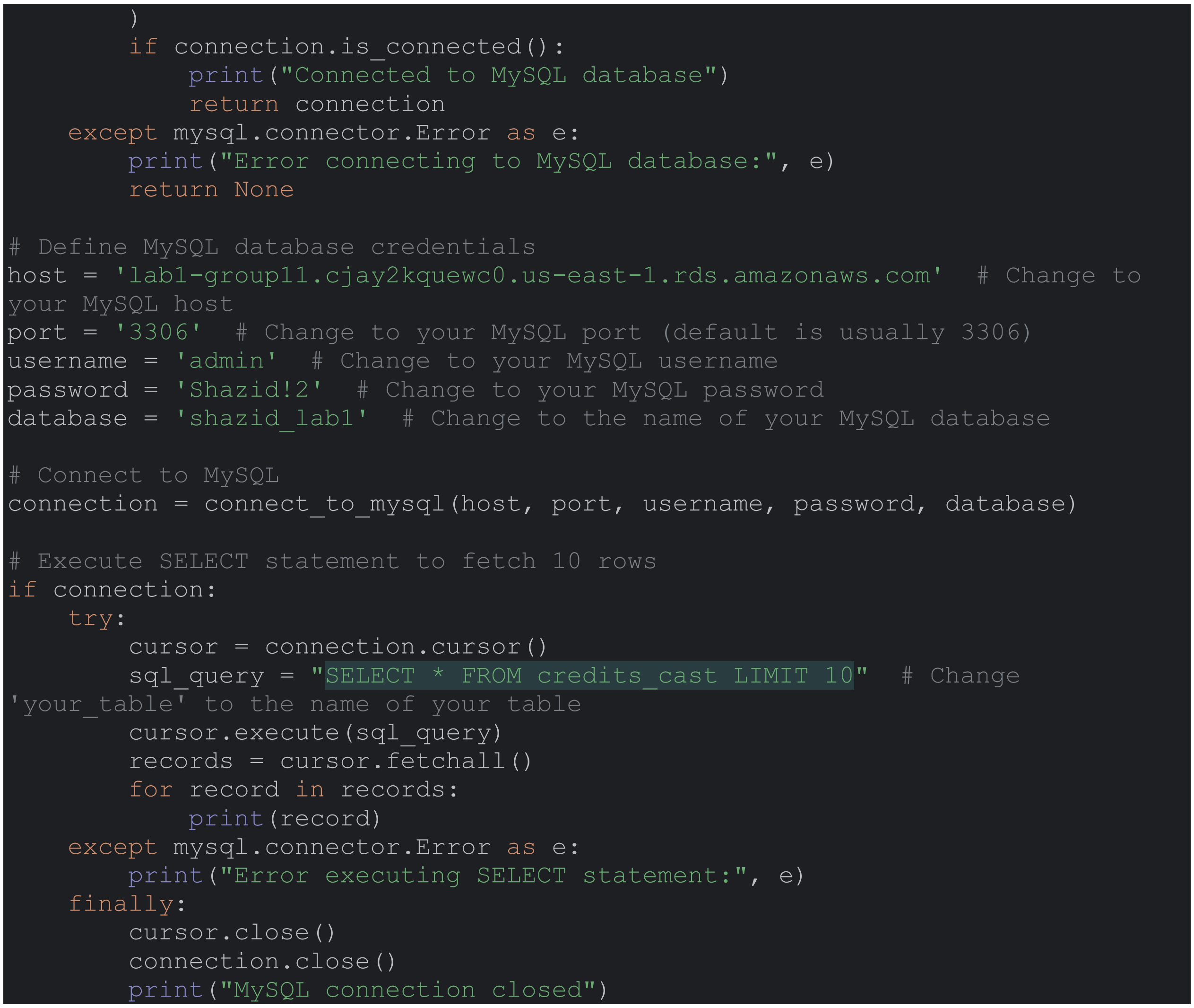
6. Using python, push every data and table from SQL to AWS-RDS instance created in the previous steps.

6. Verify data migration by running SQL queries and looking for problems or difficulties.

7.2. Demonstrating Connectivity to AWS using Python:

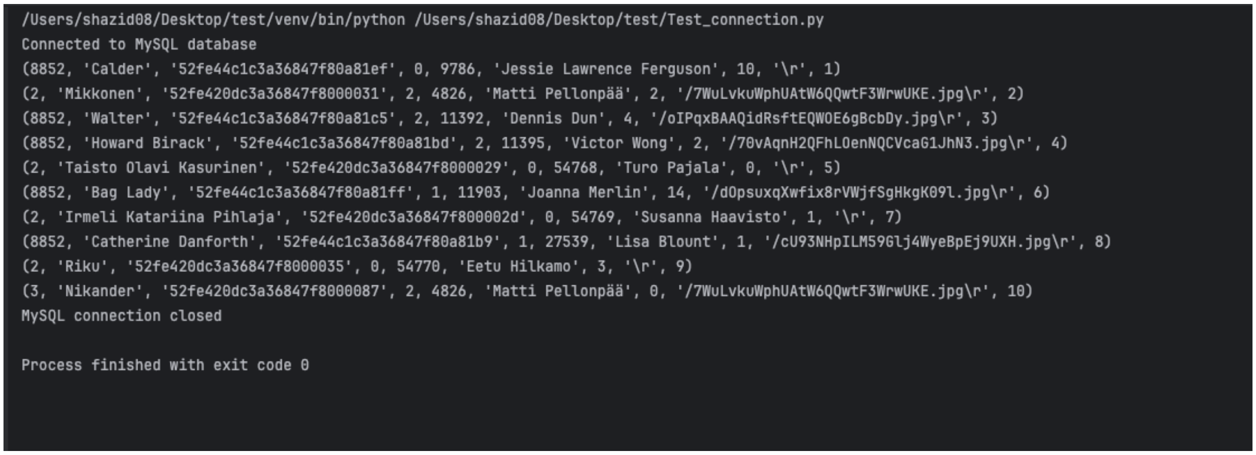


```python
import mysql.connector

# Connect to MySQL database
def connect_to_mysql(host, port, username, password, database):
    try:
        connection = mysql.connector.connect(
            host=host,
            port=port,
            user=username,
            password=password,
            database=database
```



```
        )
        if connection.is_connected():
            print("Connected to MySQL database")
            return connection
    except mysql.connector.Error as e:
        print("Error connecting to MySQL database:", e)
        return None

# Define MySQL database credentials
host = 'lab1-group11.cjay2kquewc0.us-east-1.rds.amazonaws.com'  # Change to
your MySQL host
port = '3306'  # Change to your MySQL port (default is usually 3306)
username = 'admin'  # Change to your MySQL username
password = 'Shazid!2'  # Change to your MySQL password
database = 'shazid_lab1'  # Change to the name of your MySQL database

# Connect to MySQL
connection = connect_to_mysql(host, port, username, password, database)

# Execute SELECT statement to fetch 10 rows
if connection:
    try:
        cursor = connection.cursor()
        sql_query = "SELECT * FROM credits_cast LIMIT 10"  # Change
'your_table' to the name of your table
        cursor.execute(sql_query)
        records = cursor.fetchall()
        for record in records:
            print(record)
    except mysql.connector.Error as e:
        print("Error executing SELECT statement:", e)
    finally:
        cursor.close()
        connection.close()
        print("MySQL connection closed")
```

7.2.1. Result:

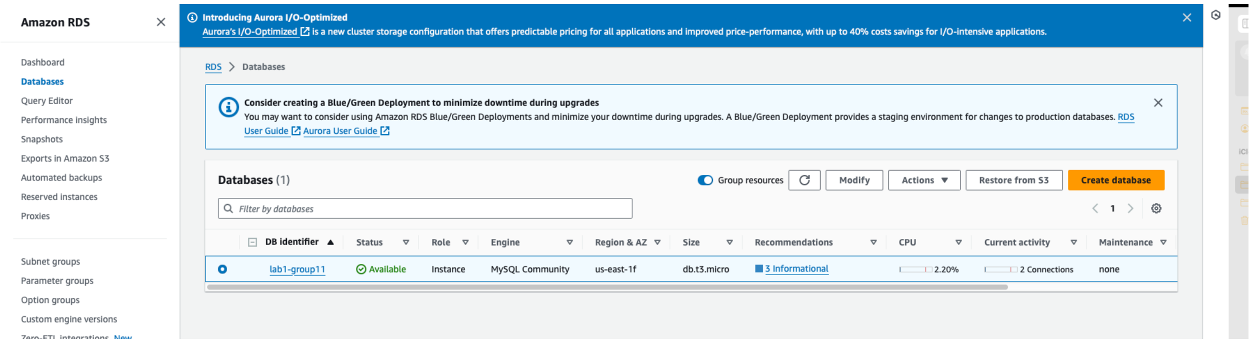


```
/Users/shazid08/Desktop/test/venv/bin/python /Users/shazid08/Desktop/test/Test_connection.py
Connected to MySQL database
(8852, 'Calder', '52fe44c1c3a36847f80a81ef', 0, 9786, 'Jessie Lawrence Ferguson', 10, '\r', 1)
(2, 'Mikkonen', '52fe420dc3a36847f8000031', 2, 4826, 'Matti Pellonpää', 2, '/7WuLvkuWphUAtW6QQwtF3WrwUKE.jpg\r', 2)
(8852, 'Walter', '52fe44c1c3a36847f80a81c5', 2, 11392, 'Dennis Dun', 4, '/oIPqxBAAQidRsftEQWOE6gBcbDy.jpg\r', 3)
(8852, 'Howard Birack', '52fe44c1c3a36847f80a81bd', 2, 11395, 'Victor Wong', 2, '/70vAqnH2QFhLOenNQCVcaG1JhN3.jpg\r', 4)
(2, 'Taisto Olavi Kasurinen', '52fe420dc3a36847f8000029', 0, 54768, 'Turo Pajala', 0, '\r', 5)
(8852, 'Bag Lady', '52fe44c1c3a36847f80a81ff', 1, 11903, 'Joanna Merlin', 14, '/dOpsuxqXwfix8rVWjfSgHkgK09l.jpg\r', 6)
(2, 'Irmeli Katariina Pihlaja', '52fe420dc3a36847f800002d', 0, 54769, 'Susanna Haavisto', 1, '\r', 7)
(8852, 'Catherine Danforth', '52fe44c1c3a36847f80a81b9', 1, 27539, 'Lisa Blount', 1, '/cU93NHpILM59Glj4WyeBpEj9UXH.jpg\r', 8)
(2, 'Riku', '52fe420dc3a36847f8000035', 0, 54770, 'Eetu Hilkamo', 3, '\r', 9)
(3, 'Nikander', '52fe420dc3a36847f8000087', 2, 4826, 'Matti Pellonpää', 0, '/7WuLvkuWphUAtW6QQwtF3WrwUKE.jpg\r', 10)
MySQL connection closed

Process finished with exit code 0
```

Fig – 7.2.1: Output of the AWS connectivity code in Python

7.3. Screenshots

7.3.1. 1. The AWS Management Console displays the RDS instance data.



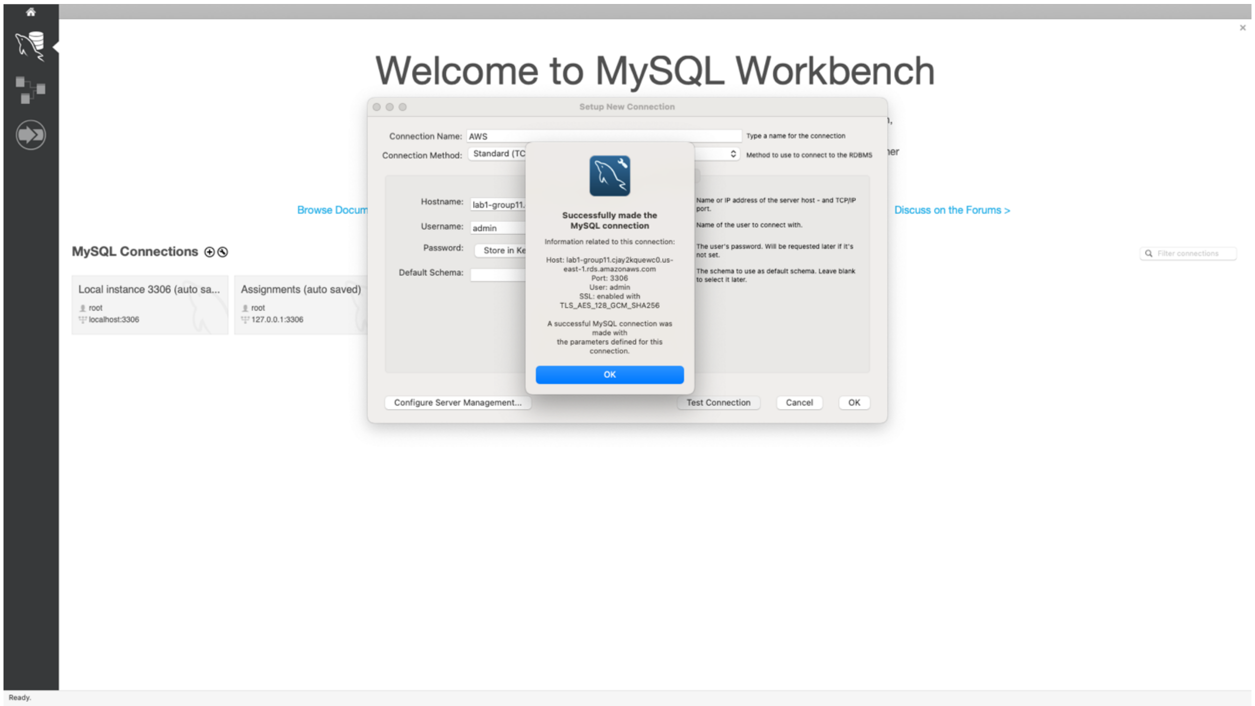
Introducing Aurora I/O-Optimized
Aurora's I/O-Optimized is a new cluster storage configuration that offers predictable pricing for all applications and improved price-performance, with up to 40% costs savings for I/O-intensive applications.

Consider creating a Blue/Green Deployment to minimize downtime during upgrades
You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. RDS User Guide Aurora User Guide

Databases (1)

DB identifier	Status	Role	Engine	Region & AZ	Size	Recommendations	CPU	Current activity	Maintenance
lab1-group11	Available	Instance	MySQL Community	us-east-1f	db.t3.micro	3 Informational	2.20%	2 Connections	none

Fig – 7.3.1: Screenshot of AWS Console



Connection Name: AWS — Type a name for the connection
Connection Method: Standard (TC — Method to use to connect to the RDBMS
Hostname: lab1-group11. — Name or IP address of the server host - and TCP/IP port.
Username: admin — Name of the user to connect with.
Password: Store in Ke — The user's password. Will be requested later if it's not set.
Default Schema: — The schema to use as default schema. Leave blank to select it later.

Successfully made the MySQL connection

Information related to this connection:

Host: lab1-group11.cjay2kquewc0.us-east-1.rds.amazonaws.com
Port: 3306
User: admin
SSL: enabled with TLS_AES_128_GCM_SHA256

A successful MySQL connection was made with the parameters defined for this connection.

OK

Fig – 7.3.2: Screenshot of Test Connection to AWS

8. GitHub Repository link:

https://github.com/Shazid08/Data225-lab1-grp11