

This project creates and manages a Movies Database (MoviesDB) using MySQL. It involves storing and organizing data about movies, genres, directors, actors, and reviews. The project implements SQL concepts like table creation, data insertion, joins, aggregate functions, views, subqueries, and stored procedures to efficiently manage and retrieve movie-related data.

The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' panel shows a tree view of databases, with 'moviesdb' selected. The main editor window shows the SQL script for creating the database and tables. The script includes comments and SQL commands for creating the database, and three tables: 'genres', 'directors', and 'movies'. The 'Output' panel at the bottom shows the execution results, indicating that the database was created successfully.

```
-- Create Database
1
2 CREATE DATABASE MoviesDB;
3
4 USE MoviesDB;
5
6 -- Create Tables
7
8 CREATE TABLE genres (
9     genre_id INT PRIMARY KEY AUTO_INCREMENT,
10    genre_name VARCHAR(100) NOT NULL
11 );
12
13 CREATE TABLE directors (
14     director_id INT PRIMARY KEY AUTO_INCREMENT,
15     director_name VARCHAR(255) NOT NULL,
16     birth_year INT
17 );
18
19 CREATE TABLE movies (
20     movie_id INT PRIMARY KEY AUTO_INCREMENT,
21     title VARCHAR(255) NOT NULL,
22     release_year INT NOT NULL,
23     genre_id INT,
24     director_id INT
```

A red arrow points from the line `CREATE DATABASE MoviesDB;` to the text: A database named MoviesDB is created

Output

#	Time	Action	Message	Duration / Fetch
1	07:13:57	CREATE DATABASE MoviesDB	1 row(s) affected	0.203 sec



- foreign_key
- game
- job
- moviesdb
 - tables
 - Views
 - Stored Procedure
 - Functions
- sakila
- school
- studentdb
- stuinfo
- sys
- world

Limit to 1000 rows

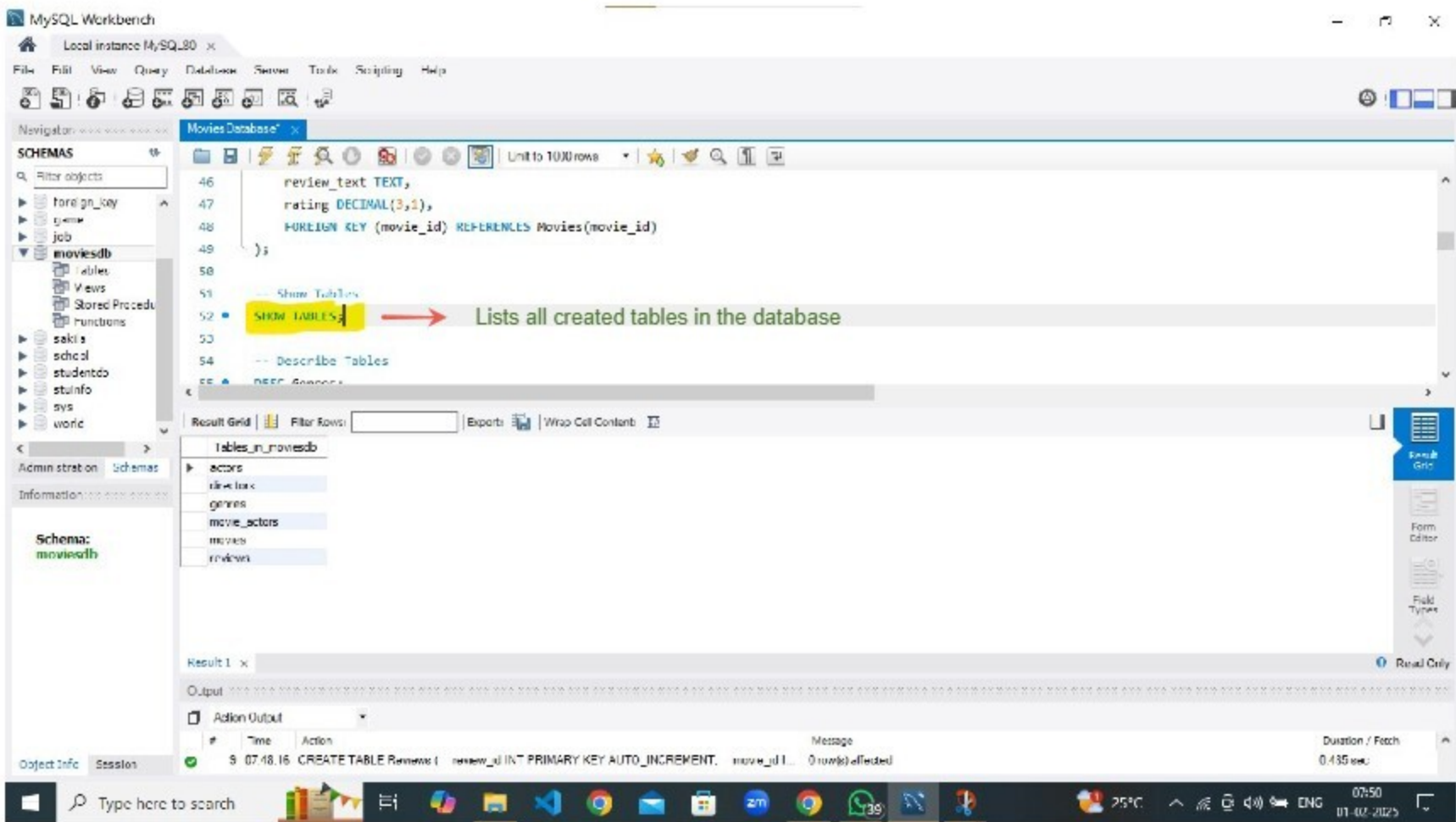
```
1 -- Create Database
2 CREATE DATABASE MoviesDB;
3 USE MoviesDB;
4
5 -- Create Tables
6 CREATE TABLE Genres (
7     genre_id INT PRIMARY KEY AUTO_INCREMENT,
8     genre_name VARCHAR(100) NOT NULL
9 );
10
11 CREATE TABLE Directors (
12     director_id INT PRIMARY KEY AUTO_INCREMENT,
13     director_name VARCHAR(255) NOT NULL,
14     birth_year INT
15 );
16
17 CREATE TABLE Movies (
18     movie_id INT PRIMARY KEY AUTO_INCREMENT,
19     title VARCHAR(255) NOT NULL,
20     release_year INT NOT NULL,
21     genre_id INT,
22     director_id INT;
```

Ensures that all subsequent operations are performed within this database

Output

Action Output

#	Time	Action	Message	Duration / Fetch
2	07:43:52	USE MoviesDB	0 row(s) affected	0.000 sec



MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: www.mysql.com

SCHEMAS

Filter objects

- foreign_key
- genre
- job
- moviesdb
 - Tables
 - Views
 - Stored Procedure
 - Functions
- sakila
- school
- studentdb
- stulinfo
- sys
- world

Adminstration Schemas

Information: www.mysql.com

Schema: moviesdb

49);

50

51 -- Show tables

52 SHOW TABLES;

53

54 Describe Tables

55 DESC Genres; Displays the structure of each table (columns and data types)

56 DESC Directors;

57 DESC Movies;

58 DESC Actors;

59 DESC Movie_Actors;

60 DESC Reviews;

61

Result Grid | Filter Rows: | Exports | Wrap Cell Contents

Field	Type	Null	Key	Default	Extra
genre_id	int	NO	PK	0000	auto_increment
genre_name	varchar(100)	NO		0000	

Result 3 x Result 4 Result 5 Result 6 Result 7 Result 8

Output

Action Output

#	Time	Action	Message	Duration / Feeds
✓	6 07.53.23	DESC Movie_Actors	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

25°C 07:54 01-02-2025

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: xxxxxxxx Movies Database x

SCHEMAS

Filter objects

- foreign_key
- game
- job
- moviesdb
 - tables
 - Views
 - Stored Procedure
 - Functions
- skills
- school
- studentdo
- stuinfo
- sys
- world

Adminstration Schemas

Information: xxxxxxxx

Schema: moviesdb

```
49 );
50
51 -- Show tables
52 SHOW TABLES;
53
54 -- Describe Tables
55 DESC genres;
56 DESC Directors;
57 DESC Movies;
58 DESC Actors;
59 DESC Movie_Actors;
60 DESC Reviews;
61
```

Displays the structure of each table (columns and data types)

Field	Type	Null	Key	Default	Extra
director_id	int	NO	PK	NULL	auto_increment
director_name	varchar(255)	NO		NULL	
birth_year	int	YES		NULL	

Result Grid | Filter Rows: | Exports | Wrap Cell Contents

Result 0 | Result 4 x | Result 5 | Result 6 | Result 7 | Result 8

Output

Adion Output

#	Time	Action	Message	Duration / Fetch
6	07:53:23	DESC Movie_Actors	2 rows returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

25°C

07:55 01-02-2025

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigation: www.mysql.com

SCHEMAS

Filter objects

- foreign_key
- game
- job
- moviesdb**
 - tables
 - Views
 - Stored Procedure
 - Functions
- sales
- school
- studentdb
- stuinfo
- sys
- world

Adminstration Schemas

Information: www.mysql.com

Schema: moviesdb

Unit to 1000 rows

49
50
51 -- Show tables
52 • SHOW TABLES;
53
54 -- Describe Tables
55 • DESC Genres;
56 • DESC Directors;
57 • **DESC Movies;**
58 • DESC Actors;
59 • DESC Movie_Actors;
60 • DESC Reviews;
61

Displays the structure of each table (columns and data types)

Result Grid | Filter Rows: | Exports: | Wrap Cell Contents: |

Field	Type	Null	Key	Default	Extra
movie_id	int	NO	PK	1234	auto_increment
title	varchar(255)	NO		1234	
release_year	int	NO		1234	
genre_id	int	YES	MUL	1234	
director_id	int	YES	MUL	1234	
rating	decimal(3,1)	YES		1234	

Result 1 Result 4 Result 5 x Result 6 Result 7 Result 8

Output

✓ Action Output

#	Time	Action	Message	Duration / Fetch
6	07:53.23	DESC Movie_Actors	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

25°C 07:58 07-02-2025

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigation Movies Database x

SCHEMAS

Filter objects

- fore_ph_key
- game
- job
- moviesdb**
 - Tables
 - Views
 - Stored Procedure
 - Functions
- sakila
- school
- studenthm
- stinfo
- sys
- world

Administration Schemas

Information

Schema: moviesdb

Limit to 1000 rows

```
49 );
50
51 -- Show Tables
52 SHOW TABLES;
53
54 -- Describe Tables
55 DESC Genres;
56 DESC Directors;
57 DESC Movies;
58 DESC Actors;
59 DESC Movie_Actors;
60 DESC Reviews;
61
```

Displays the structure of each table (columns and data types)

Field	Type	Null	Key	Default	Extra
actor_id	int	NO	PRY	NULL	auto_increment
actor_name	varchar(255)	NO		NULL	
birth_year	int	YES		NULL	

Result 3 Result 4 Result 5 Result 6 x Result 7 Result 8

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓	16 07:53:23	DESC Movie_Actors	2 rows returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

25°C

07:59 01.02.2025

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigation: www.mysql.com

SCHEMAS

Filter objects

- foreign_key
- game
- job
- moviesdb**
 - tables
 - Views
 - Stored Procedure
 - Functions
- sakila
- school
- studentdb
- stunfo
- sys
- world

Adminstration Schemas

Information: www.mysql.com

Schema: **moviesdb**

Unit to 1000 rows

```
-- Show tables
SHOW TABLES;

-- Describe Tables
DESC Genres;
DESC Directors;
DESC Movies;
DESC Actors;
DESC Movie_Actors;
DESC Reviews;
```

Displays the structure of each table (columns and data types)

Field	Type	Null	Key	Default	Extra
movie_id	int	NO	PK	NULL	
actor_id	int	NO	PK	NULL	

Result 1 / x Result 0

Output

☐ Action Output

#	Time	Action	Message	Duration / Fetch
6	07:53:23	DESC Movie_Actors	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

25°C

08:00 01.02.2025



```
49 }
50
51 -- Show tables
52 SHOW TABLES;
53
54 -- Describe Tables
55 DESC Genres;
56 DESC Directors;
57 DESC Movies;
58 DESC Actors;
59 DESC Movie_Actors;
60 DESC Reviews;
61
```

Displays the structure of each table (columns and data types)

Field	Type	Null	Key	Default	Extra
review_id	int	NO	PK	NULL	auto_increment
movie_id	int	YES	MI	NULL	
reviewer_name	varchar(255)	YES		NULL	
review_text	text	YES		NULL	
rating	decimal(3,1)	YES		NULL	

Result 1 Result 4 Result 5 Result 6 Result 7 Result 8 x

Output

Action Output

Time Action Message Duration / Fetch

✓ 07:53:23 DESC Movie_Actors Show() returned 0.000 sec / 0.000 sec

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigation: www.mysql.com

SCHEMAS

Filter objects

- example
- foreign_key
- game
- job
- moviesdb
- tables
- views
- stored procedure
- functions
- sakila
- school
- studentdb
- stinfo
- sys

Unit to 1000 rows

```
88 (1, 'Rahul', 'Epic movie with great visuals!', 8.5),
89 (2, 'Priya', 'Unique concept and amazing execution.', 9.0),
90 (3, 'Arun', 'A powerful film with a strong message.', 9.2),
91 (4, 'Deepa', 'Best action movie in recent times.', 9.5);
92
93 -- Select Values From Tables
94 SELECT * FROM Genres;
95 SELECT * FROM Directors;
96 SELECT * FROM Movies;
97 SELECT * FROM Actors;
98 SELECT * FROM Movie_Actors;
99 SELECT * FROM Reviews;
100
```

Retrieves all movies details.

Information: www.mysql.com

Result Grid

movie_id	title	release_year	genre_id	director_id	rating
1	Pannaen Selven	2022	1	1	8.1
2	Jeejeyan	2005	2	2	8.5
3	Asuran	2010	3	3	8.8
4	Vilam	2022	4	4	9.0

Schema: moviesdb

Genres 10 Directors 11 Movies 12 x Actors 13 Movie_Actors 14 Reviews 15

Output

Action Output

#	Time	Action	Message	Duration / Fetch
20	08.07.13	INSERT INTO Movies (title, release_year, genre_id, director_id, rating) VALUES ('Pannaen Selven',	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0	0.2 9 sec

Object Info Session

Type here to search

08:33 01-02-2025

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: www.mysql.com

SCHEMAS

Filter objects

- example
- foreign_key
- game
- job
- moviesdb
 - Tables
 - Views
 - Stored Procedure
 - Functions
- sakila
- school
- studentdb
- stinfo
- sys

Adminstration Schemas

Information: www.mysql.com

Schema: moviesdb

Result Grid

Filter Rows: [] Edit Export/Import Wrap Cell Contents

actor_id actor_name birth_year

1	Kamal Haasan	1954
2	Vikram	1956
3	Dhanush	1983
4	Sunya	1975

Genre: 10 Directors: 11 Movies: 12 Actors: 13 Movie_Actors: 14 Reviews: 15

Output

Action Output

Time Action Message Duration / Fetch

21 03:07:13 INSERT INTO Actors (actor_name, birth_year) VALUES ('Kamal Haasan', 1954), ('Vikram', 1956), ('Dhanush', 1983), ('Sunya', 1975); 4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0 0.230 sec

Retrieves all actors details.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Window Tools Scripting Help

Navigator: www.mysql.com

SCHEMAS

Filter objects

- example
- foreign_key
- game
- job
- moviesdb
- sakila
- school
- studentdo
- stunfo
- sys

Administration Schemas

Information: www.mysql.com

Schema: **moviesdb**

Result Grid | Filter Rows: | Edit | Export/Import | Wrap Cell Contents

movie_id actor_id

4	1
1	2
2	2
3	3

Genres 10 Directors 11 Movies 12 Actors 13 **Movie_Actors 14** Reviews 15

Output

Action Output

#	Time	Action	Message	Duration / Fetch
29	08:13:02	SELECT * FROM Movie_Actors LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Retrieves all movie-actor relationships.



SCHEMAS

Filter objects

- company
- customerinfo
- employee
- example
- foreign_key
- game
- job
- moviesdb
- sakila
- school
- studentdb
- stuinfo
- sys
- world

Information:

Schema: foreign_key

Link to 1000 rows

```
94 • SELECT * FROM Genres;
95 • SELECT * FROM Directors;
96 • SELECT * FROM Movies;
97 • SELECT * FROM Actors;
98 • SELECT * FROM Movie_Actors;
99 • SELECT * FROM Reviews;
100
101 -- Update Queries
102 • UPDATE Movies SET rating = 8.3 WHERE title = 'Ponniyin Selvan';
103 • UPDATE Directors SET birth_year = 1958 WHERE director_name = 'Mani Ratnam';
104
105 -- Joins
```

Retrieves all movie reviews.

Result Grid | Filter Rows: | Edit: | Export (Import): | Wrap Cell Contents: |

review_id	movie_id	reviewer_name	review_text	rating
1	1	Rahul	Epic movie with great visuals!	8.5
2	1	Priya	Unique concept and amazing execution.	9.0
3	3	Arun	A powerful film with a strong message.	9.2
4	4	Deepa	Best action movie in recent times.	9.5

Genres 10 Directors 11 Movies 12 Actors 13 Movie_Actors 14 Reviews 15 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
29	08:13:02	SELECT * FROM Movie_Actors LIMIT 0, 1030	4 row(s) returned	0.030 sec / 0.000 sec
30	08:13:02	SELECT * FROM Reviews LIMIT 0, 1030	4 row(s) returned	0.030 sec / 0.000 sec





Navigator: www.mysql.com

Movies Database x

SCHEMAS

Filter objects

- company
- customerinfo
- employee
- example
- foreign_key
- game
- job
- moviesdb
 - Tables
 - Views
 - Stored Procedure
 - Functions
- sakila
- school
- test

Administration Schemas

Information: www.mysql.com

No object
selected

Limit to 1000 rows

```
109
110 -- Joins
111 SELECT m.title, d.director_name, m.release_year
112 FROM Movies m
113 JOIN Directors d ON m.director_id = d.director_id;
114
115 SELECT m.title, a.actor_name
116 FROM Movies m
117 JOIN Movie_Actors ma ON m.movie_id = ma.movie_id
118 JOIN Actors a ON ma.actor_id = a.actor_id;
119
```



Movies with Directors: Displays movies along with their directors and released year.

Result Grid Filter Name: Report: Wrap Cell Contents: T

title	director name	release year
Ponniyin Selvan	Mani Ratnam	2022
Anniyan	Shankar	2005
Asuran	Vetri Maaran	2019
Vikram	Lokesh Kanagaraj	2022

Result 1 x

Output

Action Output

#	Time	Action	Message
1	14:19:18	SELECT m.title, d.director_name, m.release_year FROM Movies m JOIN Directors d ON m.director_id = d.director_id;	4 rows returned

Duration / Fetch
0.594 sec / 0.000 sec

Object Info Session



MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Window Tools Scripting Help

Navigator: www.mysql.com

SCHEMAS

Filter objects

- company
- customerinfo
- employee
- example
- foreign_key
- game
- job
- moviesdb
 - Tables
 - Views
 - Stored Procedure
 - Functions
- sakila
- school

Adminstration Schemas

Information: no object selected

No object selected

Movies Database

Limit to 1000 rows

```
111 SELECT m.title, d.director_name, m.release_year
112 FROM Movies m
113 JOIN Directors d ON m.director_id = d.director_id;
114
115 SELECT m.title, a.actor_name
116 FROM Movies m
117 JOIN Movie_Actors ma ON m.movie_id = ma.movie_id
118 JOIN Actors a ON ma.actor_id = a.actor_id;
119
120 SELECT r.review_id, m.title AS movie_title, r.reviewer_name, r.review_text, r.rating
121 FROM Reviews r
```

Result Grid

title	actor_name
Vikram	Kareel Hassan
Aniryan	Vikram
Ponniyin Selvan	Vikram
Aarav	Dhanush

Result 2 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	14:19:18	SELECT m.title, d.director_name, m.release_year FROM Movies m JOIN Directors d ON m.director_id = d.director_id;	4 row(s) returned	0.594 sec / 0.000 sec
2	14:22:22	SELECT m.title, a.actor_name FROM Movies m JOIN Movie_Actors ma ON m.movie_id = ma.movie_id JOIN Actors a ON ma.actor_id = a.actor_id;	4 row(s) returned	0.641 sec / 0.000 sec

Object Info Session

Read Only

Breaking news

14:22 03-02-2025

Movies with Actors: Retrieves actors who acted in each movie.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Window Tools Scripting Help

Navigator: www.mysql.com

SCHEMAS

Filter objects

- company
- customerinfo
- employee
- example
- foreignkey
- game
- job
- moviesdb
 - Tables
 - Views
 - Stored Procedure
 - Functions
- sakila
- school

Adminstration Schemas

Information

No object selected

Result Grid

Filter Rows: Export: Wrap Cell Contents

review_id	movie_title	reviewer_name	review_text	rating
1	Avengers: Endgame	Kahul	epic movie with great visuals!	8.5
2	Avengers	Priya	Unique concept and amazing execution.	9.0
3	Avengers	Anur	A powerful film with a strong message.	9.2
4	Avengers	Deepee	Best action movie in recent times.	9.5

Result 4 x

Output

Action Output

Object Info Session

3 14:24:50 SELECT r.review_id, m.title AS movie_title, r.reviewer_name, r.review_text, r.rating FROM Review... 4 row(s) returned

Duration / Fetch 0.13 / sec / 0.016 sec

14:28 03-02-2025

Breaking news

Movie Reviews: Retrieves all reviews along with the movie name.

```
116 FROM Movies m
117 JOIN Movie_Actors ma ON m.movie_id = ma.movie_id
118 JOIN Actors a ON ma.actor_id = a.actor_id;
119
120 SELECT r.review_id, m.title AS movie_title, r.reviewer_name, r.review_text, r.rating
121 FROM Reviews r
122 JOIN Movies m ON r.movie_id = m.movie_id;
123
124 -- Aggregate Functions
125 SELECT m.title, COUNT(r.review_id) AS total_reviews
126 FROM Movies m
127 LEFT JOIN Reviews r ON m.movie_id = r.movie_id
```

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigation: Movies Database x

SCHEMAS

Filter objects

- company
- customersinfo
- employee
- example
- foreignkey
- game
- job
- moviesdb
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sekila
- school

Administration Schemas

Information: No object selected

Result Grid | Filter Rows: | Export: | Wrap Cell Contents: 16

title	total_reviews
Purnima Salyan	1
Anirban	1
Asuran	1
Vikram	1

Result 0 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
4	14:27:33	SELECT r.review_id, m.title AS movie_title, r.reviewer_name, r.review_text, r.rating FROM Review...	4 row(s) returned	0.030 sec / 0.000 sec
5	14:29:50	SELECT m.title, COUNT(r.review_id) AS total_reviews FROM Movies m LEFT JOIN Reviews r O...	4 row(s) returned	0.110 sec / 0.000 sec

Count Reviews Per Movie: Calculates the total number of reviews for each movie.



SCHEMAS

Filter objects

- company
- customerinfo
- employee
- example
- foreignkey
- game
- job
- moviesdb
 - Tables
 - Views
 - Stored Procedure
 - Functions
- sakila
- school
- testdatabase

Administration Schemas

Information schema database

No object selected

Object Info Session

MoviesDatabase x

Link to 1000 rows

```
127 LEFT JOIN Reviews r ON m.movie_id = r.movie_id
128 GROUP BY m.title;
129
130 SELECT g.genre_name, AVG(m.rating) AS avg_rating
131 FROM Movies m
132 JOIN Genres g ON m.genre_id = g.genre_id
133 GROUP BY g.genre_name;
134
135 SELECT MIN(release_year) AS Oldest_Movie, MAX(release_year) AS Newest_Movie
136 FROM Movies;
137
138 -- Views
```

→ Average Rating Per Genre: Computes the average rating of movies in each genre.

Result Grid Filter Rows: Export: Wrap Cell Contents: 16

genre_name	avg_rating
Action	8.30000
Drama	8.50000
Comedy	8.80000
Thriller	9.00000

Result 7 x

Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
6	14:36:36	SELECT g.genre_name, AVG(m.rating) AS avg_rating FROM Movies m JOIN Genres g ON m.genre...	4 row(s) returned	0.231 sec / 0.000 sec



MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Window Tools Scripting Help

Navigation: www.mysql.com

SCHEMAS

Filter objects

- company
- customerinfo
- employee
- example
- foreign_key
- game
- job
- moviesdb
 - Tables
 - Views
 - Stored Procedure
 - Functions
- sakila
- school

Administration Schemas

Information

No object selected

Link to 1000 rows

```
137
138 -- View
139 CREATE VIEW HighRatedMovies AS
140 SELECT title, rating
141 FROM Movies
142 WHERE rating >= 8.0;
143
144 SELECT * FROM HighRatedMovies;
145
146 -- Show View Structure
147 SHOW CREATE VIEW HighRatedMovies;
148
149 -- Subquery
150 SELECT title, rating
151 FROM Movies
152 WHERE rating > (SELECT AVG(rating) FROM Movies);
153
154 -- Stored Procedure
155 DELIMITER //
156 CREATE PROCEDURE GetMovieCountByGenre()
157 BEGIN
158     SELECT g.genre name, COUNT(m.movie id) AS movie count
```

View for High-Rated Movies: A virtual table (HighRatedMovies) stores movies with ratings above 8.0.

Output

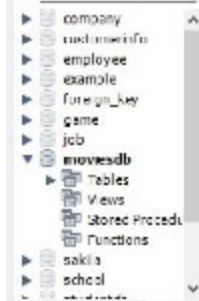
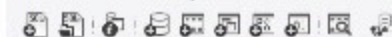
Action Output

#	Time	Action	Message	Duration / Fetch
10	14:42:44	CREATE VIEW HighRatedMovies AS SELECT title, rating FROM Movies WHERE rating >= 8.0	0 row(s) affected	0.520 sec

Object Info Session

Type here to search

14:44 03-02-2025



No object selected



title	rating
Pamihya Soliman	8.5
Anniyan	8.5
Asuran	8.8
Vikram	9.0

#	Time	Action	Message	Duration / Fetch
1	14:45:10	SELECT * FROM HighRatedMovies LIMIT 0 - 1000	4 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: www.mysql.com

SCHEMAS

Filter objects

- company
- customerinfo
- employee
- example
- foreignkey
- game
- job
- moviesdb
 - Tables
 - Views
 - Stored Procedure
 - Functions
- sakila
- school

Adminstration Schemas

Information: www.mysql.com

No object selected

Link to 1000 rows

```
148
149 -- Subquery
150 SELECT title, rating
151 FROM Movies
152 WHERE rating > (SELECT AVG(rating) FROM Movies)
153
154 -- Stored Procedure
155 DELIMITER //
156 CREATE PROCEDURE GetMovieCountByGenre()
157 BEGIN
158     SELECT g.genre name, COUNT(m.movie id) AS movie count
159     FROM Movies m
160     JOIN Genre g ON m.genre id = g.genre id
161     GROUP BY g.genre name
162 END //
163
```

Retrieves movies with a rating higher than the average rating of all movies.

Result Grid

title	rating
Action	8.8
War	9.0

Movies 10 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
17	14.45.26	SELECT title, rating FROM Movies WHERE rating > (SELECT AVG(rating) FROM Movies) LIMIT 2	2 row(s) returned	0.063 sec / 0.000 sec

Object Info Session

Type here to search

27°C 14:49 03-02-2025



SCHEMAS

Filter objects

- company
- customerinfo
- employee
- example
- foreignkey
- game
- job
- moviesdb
 - Tables
 - Views
 - Stored Procedure
 - Functions
- sakila
- school
- test

Administration Schemas

Information

No object selected



```
152 WHERE rating > (SELECT AVG(rating) FROM Movies);
153
154 -- Stored Procedure
155 DELIMITER //
156 CREATE PROCEDURE GetMovieCountByGenre()
157 BEGIN
158     SELECT g.genre_name, COUNT(m.movie_id) AS movie_count
159     FROM Movies m
160     JOIN Genres g ON m.genre_id = g.genre_id
161     GROUP BY g.genre_name;
162 END //
163 DELIMITER ;
164
165 CALL GetMovieCountByGenre();
166
167 -- Show Stored Procedures
168 SHOW PROCEDURE STATUS WHERE DB = 'MoviesDB';
169
170 -- Trigger: Auto-update Movie Rating after inserting a Review
171 DELIMITER //
172 CREATE TRIGGER UpdateMovieRating
173 AFTER INSERT ON Reviews
```

GetMovieCountByGenre(): A stored procedure that returns the total number of movies in each genre.

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	14:51:13	CREATE PROCEDURE GetMovieCountByGenre() BEGIN SELECT g.genre_name, COUNT(m...	0 row(s) affected	0.338 sec



MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Tools Scripting Help

Navigation: www.mysql.com

SCHEMAS

Filter objects

- company
- customerinfo
- employee
- example
- foreignkey
- game
- job
- moviesdb
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sakila
- school
- test

Administration Schemas

Information: www.mysql.com

Link to 1000 rows

```
161 GROUP BY g.genre_name;
162 END //
163 DELIMITER ;
164
165 CALL GetMovieCountByGenre();
166
167 -- Show Stored Procedures
168 SHOW PROCEDURE STATUS WHERE db = 'MoviesDB';
169
170 -- Trigger: Auto-update Movie Rating after inserting a Review
171 DELIMITER //
172 CREATE TRIGGER UpdateMovieRating
173 AFTER INSERT ON Reviews
174 FOR EACH ROW
```

The procedure is called using CALL GetMovieCountByGenre();

No object selected

Result Grid | Filter Rows | Export | Wrap Cell Contents

genre_name	movie_count
Action	1
Drama	1
Comedy	1
Thriller	1

Result 1/ x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
21	11:53:12	CALL GetMovieCountByGenre()	4 row(s) returned	0.218 sec / 0.000 sec

Object Info Session

Type here to search

27°C 14:55 10/12/2025

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Tools Scripting Help

Navigator: Local instance MySQL80

SCHEMAS

Filter objects

- company
- customerinfo
- employee
- example
- foreignkey
- game
- job
- moviesdb
 - Tables
 - Views
 - Stored Procedure
 - Functions
- sakila
- school
- test

Administration Schemas

Information

No object selected

Limit to 1000 rows

```
164
165 • CALL GetMovieCountByGenre();
166
167 -- Show Stored Procedures
168 • SHOW PROCEDURE STATUS WHERE Db = 'MoviesDB';
169
170 -- Trigger: Auto-update Movie Rating after inserting a Review
171 DELIMITER //
172 • CREATE TRIGGER UpdateMovieRating
173 AFTER INSERT ON Reviews
174 FOR EACH ROW
175 BEGIN
176     UPDATE Movies
177     SET rating = (SELECT AVG(r.rating) FROM Reviews r WHERE r.movie_id = NEW.movie_id)
178     WHERE movie_id = NEW.movie_id;
179 END;
180 //
181 DELIMITER ;
182
183 • SHOW TRIGGERS;
184
```

UpdateMovieRating: A trigger that automatically updates the movie rating in the Movies table after a new review is inserted into the Reviews table.

Output

Action Output

#	Time	Action	Message	Duration / Fetch
24	14:58:08	CREATE TRIGGER UpdateMovieRating AFTER INSERT ON Reviews FOR EACH ROW BEGIN;	0 row(s) affected	0.339 sec

Object Info Session

Type here to search

27°C 14:59 03-02-2025