Cinema Management System

Team members:
Ghala Marzouq Alsaedi - 445007536
Jumana Khalid Alsaedi - 445007196
Leena Amin Alghamdi-44510682
Layan saleh alQurashi -44511056
Bayan Yousef Alkhayri - 445006095
Shahad Hassan Alkhuzai - 44510698

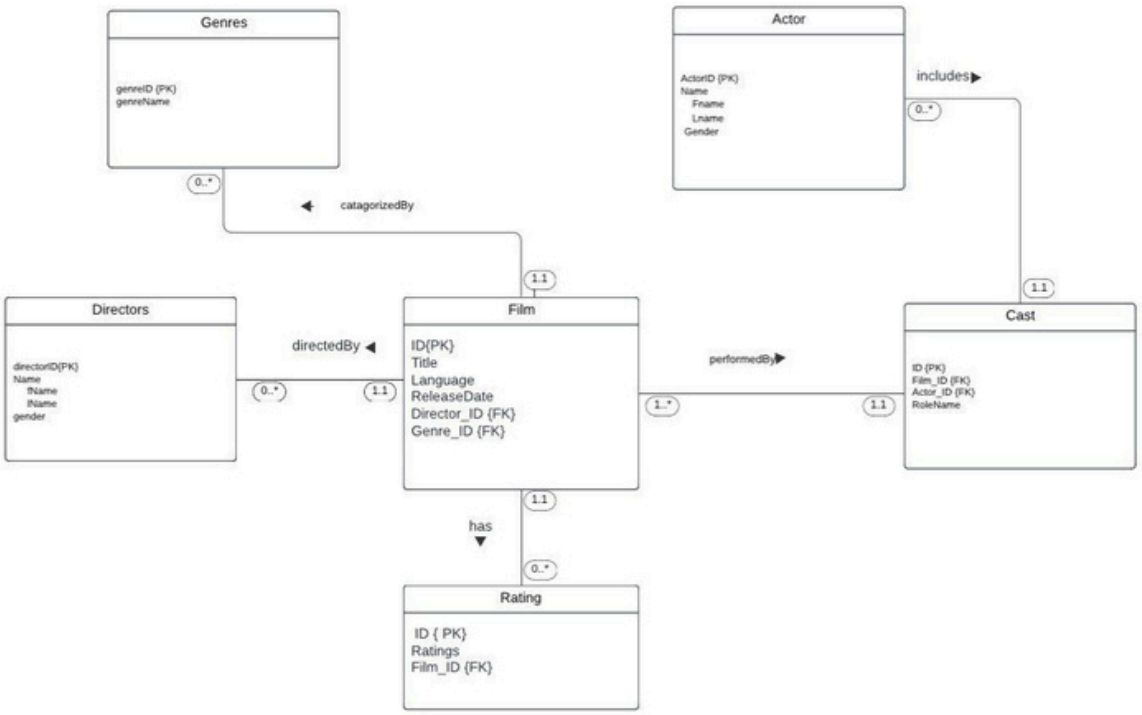
Section Number: 4
Dr. Affaf Almehmadi

Business Rules:

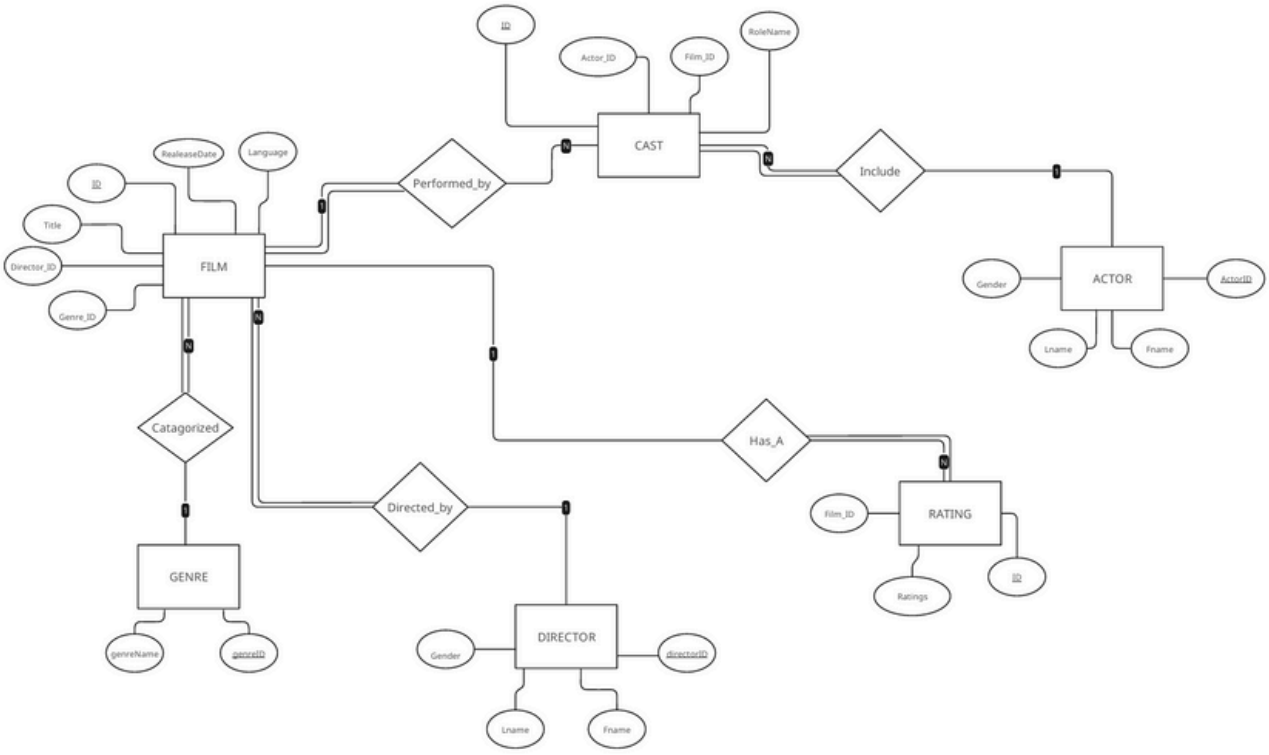
- Each film has a unique **film ID, a title, a language, a release date,** and **categorized by one genre**.
- Each film is **directed by** one director, and each director may direct zero or more films.
- Each film **performed by** at least one actor in its cast, and each actor may act in zero or more films.
- Each film can have zero or more ratings, and each rating belongs to one film.
- Each **director** has a unique **director ID, a first name, a last name, and a gender**
- Each **genre** has a **unique genre ID and a genre name**.
- Each **actor** has a **unique actor ID, a first name, a last name, and a gender**
- Each **cast** entry has a **unique ID**, specifies **one film** and **one actor**, and includes the actor's **role name**.
- Each **rating** entry has a **unique ID** and specifies a **Ratings** given to a particular film.
- Each film must be associated with one genre, and a genre can belong to zero or more films
- Each film must have exactly one director.

ER Model:

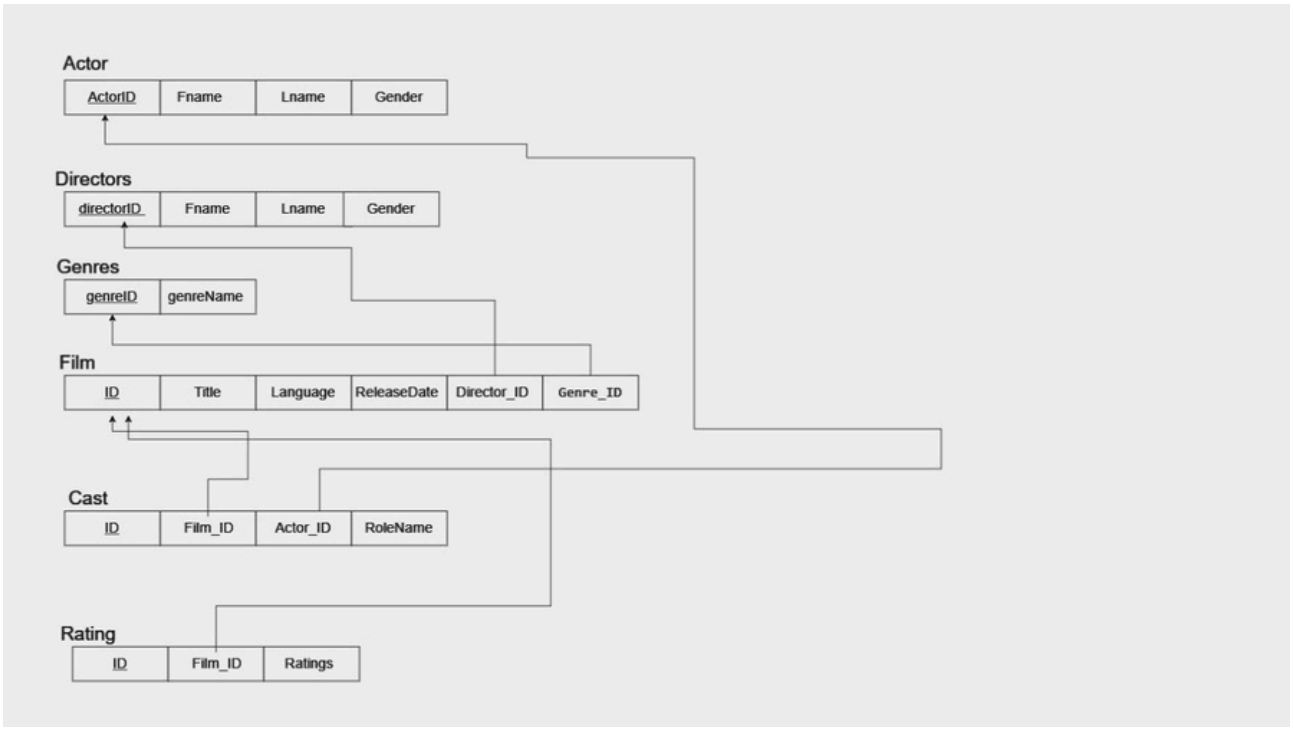
1- UML Diagram



2- Chen's Model:



3-Relational Schema:



Screenshots of SQL command and their result: Tables declarations:

```sql
CREATE SCHEMA IF NOT EXISTS MoviesDB;
USE MoviesDB;

CREATE TABLE IF NOT EXISTS Actor (
    ActorID INT(5) NOT NULL,
    Fname VARCHAR(50) NOT NULL,
    Lname VARCHAR(50) NOT NULL,
    gender VARCHAR (1)CHECK (gender IN ('M','F')),
    CONSTRAINT PK_ActorID PRIMARY KEY (ActorID)
);

CREATE TABLE IF NOT EXISTS Directors (
directorID INT(5) NOT NULL,
fName VARCHAR(50),
lName VARCHAR(50),
gender VARCHAR (1)CHECK (gender IN ('M','F')),
CONSTRAINT PK_directorID PRIMARY KEY (directorID)
);

CREATE TABLE IF NOT EXISTS Genres(
genreID INT(2) NOT NULL,
genreName VARCHAR(50) NOT NULL,
CONSTRAINT PK_genreID PRIMARY KEY (genreID)
);

CREATE TABLE IF NOT EXISTS MoviesDB.Film (
ID INT,
Title VARCHAR(100) NOT NULL,
Language VARCHAR(50) NOT NULL,
ReleaseDate DATE NOT NULL,
Director_ID INT NOT NULL,
Genre_ID INT NOT NULL,
CONSTRAINT Film_PK PRIMARY KEY (ID),
FOREIGN KEY (Director_ID) REFERENCES Directors(directorID) ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (Genre_ID) REFERENCES Genres(genreID)  ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE IF NOT EXISTS Cast (
    ID INT PRIMARY KEY,
    Film_ID INT NULL,
    Actor_ID INT,
    RoleName VARCHAR(20) NOT NULL,
    FOREIGN KEY (Film_ID) REFERENCES Film(ID) ON DELETE SET NULL ON UPDATE CASCADE,
    FOREIGN KEY (Actor_ID) REFERENCES Actor(ActorID)ON DELETE SET NULL ON UPDATE CASCADE,
    CHECK (RoleName IN ('Main actor', 'Supporting actor'))
);
);

CREATE TABLE IF NOT EXISTS Rating (
    ID INT(10) PRIMARY KEY,
    Film_ID INT,
    Ratings DECIMAL(3,1),
    FOREIGN KEY (Film_ID) REFERENCES Film(ID) ON DELETE CASCADE ON UPDATE CASCADE
);
```

Tables inserts:

```
55      INSERT INTO Directors (directorID, fName, lName, gender)
56      VALUES
57      (99, 'Christopher', 'Nolan', 'M'),  -- Interstellar
58      (102, 'Alexander', 'Payne', 'M'),   -- The Holdovers
59      (103, 'David', 'Fincher', 'M'),     -- Gone Girl
60      (104, 'Chris', 'Columbus', 'M'),    -- Home Alone
61      (105, 'Richard', 'Kelly', 'M'),     -- Donnie Darko
62      (106, 'Mel', 'Gibson', 'M'),       -- Hacksaw Ridge
63      (107,'Denis', 'Villeneuve', 'M'), -- Dune
64      (108, 'Charlotte', 'Wells', 'F'),  -- Aftersun
65      (109, 'Hang-jun', 'Jang', 'M'),    -- Forgotten
66      (110, 'Guy', 'Ritchie', 'M'),      -- Sherlock Holmes
67      (111, 'Ridley', 'Scott', 'M');     -- Black Hawk Down

68
69      INSERT INTO Genres (genreID, genreName ) VALUES
70          (1,'Science Fiction '),
71          (2,'Drama'),
72          (3,'Thriller'),
73          (4,'Comedy'),
74          (5,'Mystery'),
75          (6,'War');
76
```

```
    INSERT INTO film (ID, Title, Language, ReleaseDate, Director_ID, Genre_ID) -- Insert
VALUES
(01, 'Interstellar', 'English', '2014-10-26', 99, 1),
(02, 'The Holdovers', 'English', '2023-10-27', 102, 2),
(03, 'Gone Girl', 'English', '2014-10-3', 103, 3),
(04, 'Home Alone', 'English', '1990-11-10', 104, 4),
(05, 'Donnie Darko', 'English', '2001-10-26', 105, 5),
(06, 'Hacksaw Ridge', 'English', '2016-10-16', 106, 6),
(07, 'Dune', 'English', '2021-9-3', 107, 1),
(08, 'Aftersun', 'English', '2022-10-21', 108, 2),
(09, 'Forgotten', 'Korean', '2017-11-29', 109, 3),
(10, 'Sherlock Holmes', 'English', '2009-12-25', 110, 5),
(11, 'Black Hawk Down', 'English', '2002-1-18', 111, 6);
```

Tables inserts:

```
55      INSERT INTO Directors (directorID, fName, lName, gender)
56      VALUES
57      (99, 'Christopher', 'Nolan', 'M'),  -- Interstellar
58      (102, 'Alexander', 'Payne', 'M'),   -- The Holdovers
59      (103, 'David', 'Fincher', 'M'),     -- Gone Girl
60      (104, 'Chris', 'Columbus', 'M'),    -- Home Alone
61      (105, 'Richard', 'Kelly', 'M'),     -- Donnie Darko
62      (106, 'Mel', 'Gibson', 'M'),       -- Hacksaw Ridge
63      (107,'Denis', 'Villeneuve', 'M'), -- Dune
64      (108, 'Charlotte', 'Wells', 'F'),  -- Aftersun
65      (109, 'Hang-jun', 'Jang', 'M'),    -- Forgotten
66      (110, 'Guy', 'Ritchie', 'M'),      -- Sherlock Holmes
67      (111, 'Ridley', 'Scott', 'M');     -- Black Hawk Down

68
69      INSERT INTO Genres (genreID, genreName ) VALUES
70          (1,'Science Fiction '),
71          (2,'Drama'),
72          (3,'Thriller'),
73          (4,'Comedy'),
74          (5,'Mystery'),
75          (6,'War');
76
```

```
    INSERT INTO film (ID, Title, Language, ReleaseDate, Director_ID, Genre_ID) -- Insert
VALUES
(01, 'Interstellar', 'English', '2014-10-26', 99, 1),
(02, 'The Holdovers', 'English', '2023-10-27', 102, 2),
(03, 'Gone Girl', 'English', '2014-10-3', 103, 3),
(04, 'Home Alone', 'English', '1990-11-10', 104, 4),
(05, 'Donnie Darko', 'English', '2001-10-26', 105, 5),
(06, 'Hacksaw Ridge', 'English', '2016-10-16', 106, 6),
(07, 'Dune', 'English', '2021-9-3', 107, 1),
(08, 'Aftersun', 'English', '2022-10-21', 108, 2),
(09, 'Forgotten', 'Korean', '2017-11-29', 109, 3),
(10, 'Sherlock Holmes', 'English', '2009-12-25', 110, 5),
(11, 'Black Hawk Down', 'English', '2002-1-18', 111, 6);
```

Tables inserts:

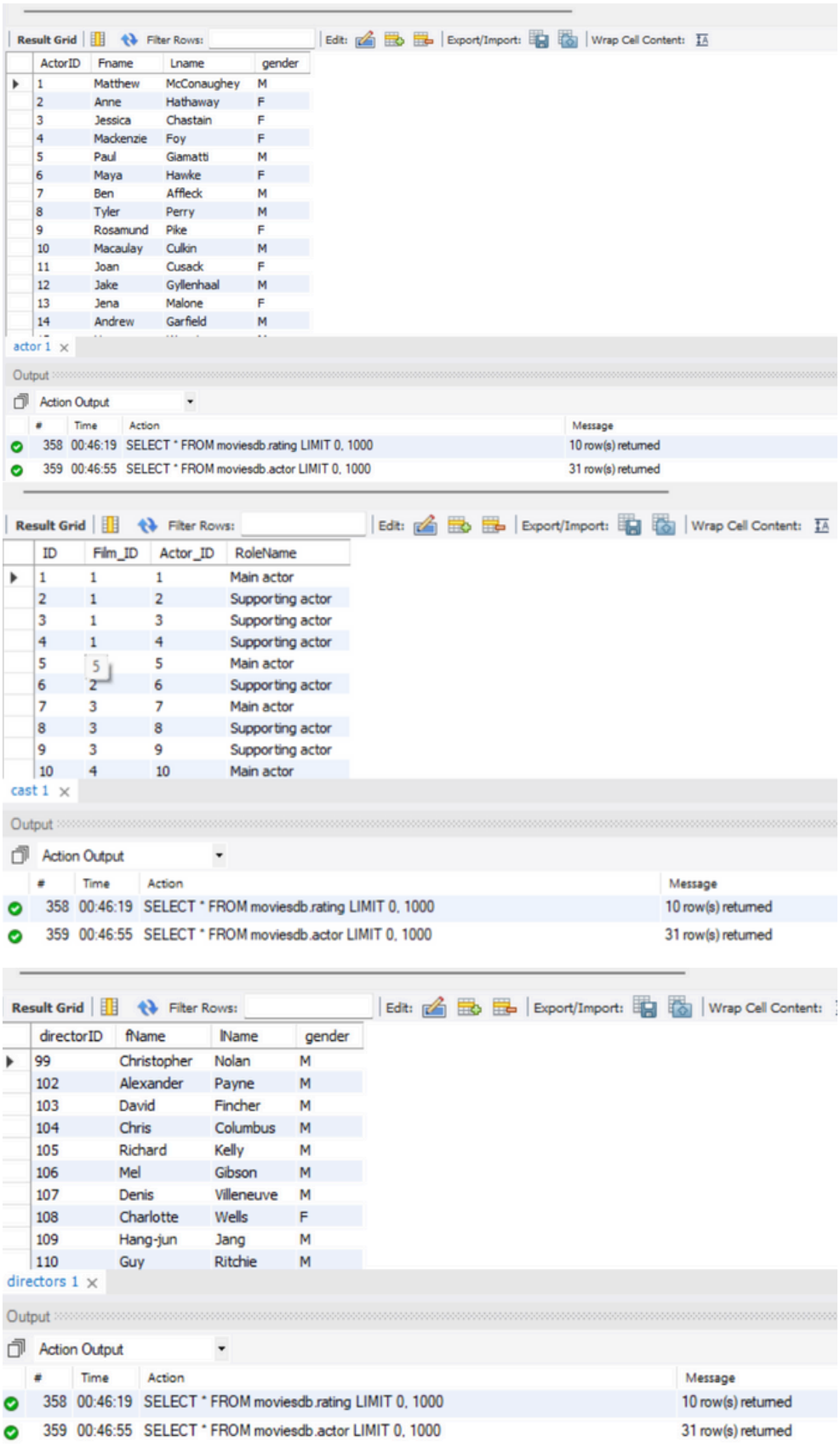
```
INSERT INTO Actor(ActorID,Fname,Lname,gender ) VALUES
(1, 'Matthew', 'McConaughey', 'M'),  -- Interstellar
(2, 'Anne', 'Hathaway', 'F'),
(3, 'Jessica', 'Chastain', 'F'),
(4, 'Mackenzie', 'Foy', 'F'),
(5, 'Paul', 'Giamatti', 'M'),       -- The Holdovers
(6, 'Maya', 'Hawke', 'F'),
(7, 'Ben', 'Affleck', 'M'),         -- Gone Girl
(8, 'Tyler', 'Perry', 'M'),
(9, 'Rosamund', 'Pike', 'F'),
(10, 'Macaulay', 'Culkin', 'M'),    -- Home Alone
(11, 'Joan', 'Cusack', 'F'),
(12, 'Jake', 'Gyllenhaal', 'M'),    -- Donnie Darko
(13, 'Jena', 'Malone', 'F'),
(14, 'Andrew', 'Garfield', 'M'),    -- Hacksaw Ridge
(15, 'Hugo', 'Weaving', 'M'),
(16, 'Vera', 'Farmiga', 'F'),
(17, 'Timothée', 'Chalamet', 'M'),  -- Dune
(18, 'Zendaya', 'Coleman', 'F'),
(19, 'Oscar', 'Isaac', 'M'),
(20, 'Paul', 'Mescal', 'M'),        -- Aftersun
(21, 'Tessa', 'Thompson', 'F'),
(22, 'Colin', 'Farrell', 'M'),
(23, 'Kang', 'Ha-neul', 'M'),       -- Forgotten
(24, 'Jeon', 'Hye-jin', 'F'),
(25, 'Lee', 'Joon', 'M'),
(26, 'Robert', 'Downey Jr.', 'M'),  -- Sherlock Holmes
```

```
INSERT INTO Cast (ID, Film_ID, Actor_ID, RoleName)
VALUES
    -- Interstellar (Film_ID = 1)
    (1, 1, 1, 'Main actor'),           -- Matthew McConaughey
    (2, 1, 2, 'Supporting actor'),     -- Anne Hathaway
    (3, 1, 3, 'Supporting actor'),     -- Jessica Chastain
    (4, 1, 4, 'Supporting actor'),     -- Mackenzie Foy

    -- The Holdovers (Film_ID = 2)
    (5, 2, 5, 'Main actor'),           -- Paul Giamatti
    (6, 2, 6, 'Supporting actor'),     -- Maya Hawke

    -- Gone Girl (Film_ID = 3)
    (7, 3, 7, 'Main actor'),           -- Ben Affleck
    (8, 3, 8, 'Supporting actor'),     -- Tyler Perry
    (9, 3, 9, 'Supporting actor'),     -- Rosamund Pike

    -- Home Alone (Film_ID = 4)
    (10, 4, 10, 'Main actor'),         -- Macaulay Culkin
    (11, 4, 11, 'Supporting actor'),   -- Joan Cusack

    -- Donnie Darko (Film_ID = 5)
    (12, 5, 12, 'Main actor'),         -- Jake Gyllenhaal
    (13, 5, 13, 'Supporting actor'),   -- Jena Malone

    -- Hacksaw Ridge (Film_ID = 6)
    (14, 6, 14, 'Main actor'),         -- Andrew Garfield
```

Tables inserts:

```
INSERT INTO Rating (ID, Film_ID, Ratings) VALUES
(1, 01, 9.5),
(2, 02, 8.0),
(3, 03, 9.0),
(4, 04, 8.5),
(5, 05, 9.0),
(6, 06, 8.5),
(7, 07, 8.5),
(8, 08, 8.5),
(9, 09, 8.5),
(10, 10, 8.5),
(11, 11, 8.0);
```

Tables inserts:



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

ActorID	Fname	Lname	gender
1	Matthew	McConaughey	M
2	Anne	Hathaway	F
3	Jessica	Chastain	F
4	Mackenzie	Foy	F
5	Paul	Giamatti	M
6	Maya	Hawke	F
7	Ben	Affleck	M
8	Tyler	Perry	M
9	Rosamund	Pike	F
10	Macaulay	Culkin	M
11	Joan	Cusack	F
12	Jake	Gyllenhaal	M
13	Jena	Malone	F
14	Andrew	Garfield	M

actor 1

Output

Action Output

#	Time	Action	Message
358	00:46:19	SELECT * FROM moviesdb.rating LIMIT 0, 1000	10 row(s) returned
359	00:46:55	SELECT * FROM moviesdb.actor LIMIT 0, 1000	31 row(s) returned

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

ID	Film_ID	Actor_ID	RoleName
1	1	1	Main actor
2	1	2	Supporting actor
3	1	3	Supporting actor
4	1	4	Supporting actor
5	5	5	Main actor
6	2	6	Supporting actor
7	3	7	Main actor
8	3	8	Supporting actor
9	3	9	Supporting actor
10	4	10	Main actor

cast 1

Output

Action Output

#	Time	Action	Message
358	00:46:19	SELECT * FROM moviesdb.rating LIMIT 0, 1000	10 row(s) returned
359	00:46:55	SELECT * FROM moviesdb.actor LIMIT 0, 1000	31 row(s) returned

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

directorID	fName	lName	gender
99	Christopher	Nolan	M
102	Alexander	Payne	M
103	David	Fincher	M
104	Chris	Columbus	M
105	Richard	Kelly	M
106	Mel	Gibson	M
107	Denis	Villeneuve	M
108	Charlotte	Wells	F
109	Hang-jun	Jang	M
110	Guy	Ritchie	M

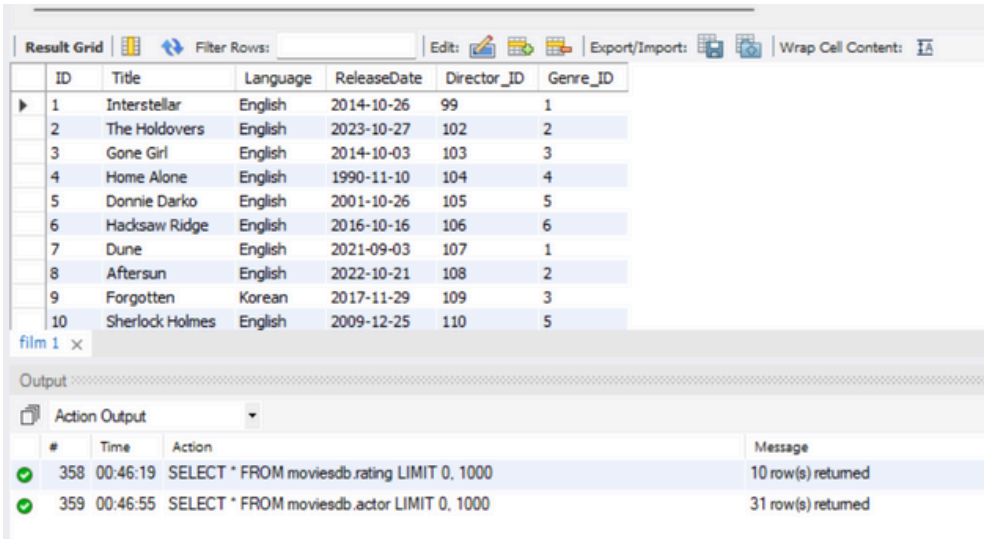
directors 1

Output

Action Output

#	Time	Action	Message
358	00:46:19	SELECT * FROM moviesdb.rating LIMIT 0, 1000	10 row(s) returned
359	00:46:55	SELECT * FROM moviesdb.actor LIMIT 0, 1000	31 row(s) returned

Tables inserts:



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

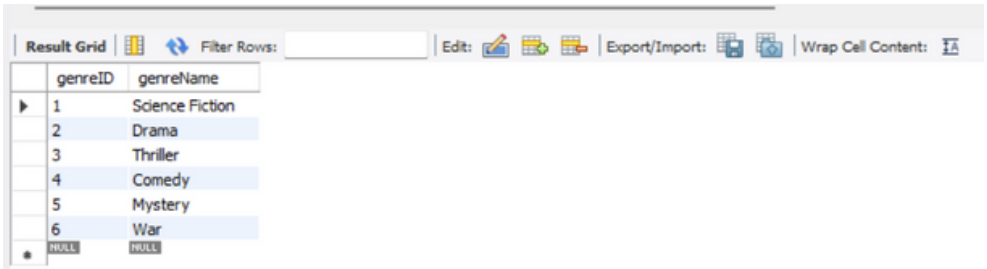
ID	Title	Language	ReleaseDate	Director_ID	Genre_ID
1	Interstellar	English	2014-10-26	99	1
2	The Holdovers	English	2023-10-27	102	2
3	Gone Girl	English	2014-10-03	103	3
4	Home Alone	English	1990-11-10	104	4
5	Donnie Darko	English	2001-10-26	105	5
6	Hacksaw Ridge	English	2016-10-16	106	6
7	Dune	English	2021-09-03	107	1
8	Aftersun	English	2022-10-21	108	2
9	Forgotten	Korean	2017-11-29	109	3
10	Sherlock Holmes	English	2009-12-25	110	5

film 1 ×

Output

Action Output

#	Time	Action	Message
358	00:46:19	SELECT * FROM moviesdb.rating LIMIT 0, 1000	10 row(s) returned
359	00:46:55	SELECT * FROM moviesdb.actor LIMIT 0, 1000	31 row(s) returned



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

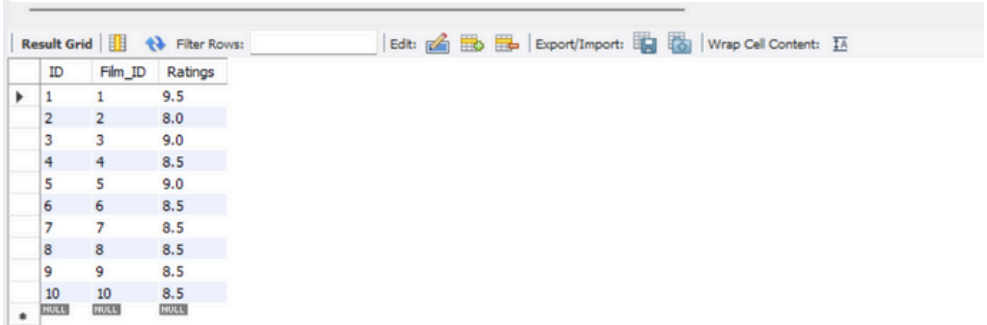
genreID	genreName
1	Science Fiction
2	Drama
3	Thriller
4	Comedy
5	Mystery
6	War
NULL	NULL

genres 1 ×

Output

Action Output

#	Time	Action	Message
358	00:46:19	SELECT * FROM moviesdb.rating LIMIT 0, 1000	10 row(s) returned
359	00:46:55	SELECT * FROM moviesdb.actor LIMIT 0, 1000	31 row(s) returned



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

ID	Film_ID	Ratings
1	1	9.5
2	2	8.0
3	3	9.0
4	4	8.5
5	5	9.0
6	6	8.5
7	7	8.5
8	8	8.5
9	9	8.5
10	10	8.5
NULL	NULL	NULL

rating 1 ×

Output

Action Output

#	Time	Action	Message
358	00:46:19	SELECT * FROM moviesdb.rating LIMIT 0, 1000	10 row(s) returned
359	00:46:55	SELECT * FROM moviesdb.actor LIMIT 0, 1000	31 row(s) returned

statements:

```
199 •   SELECT genreName, COUNT(*) AS FilmCount
200     FROM Film
201     INNER JOIN Genres ON Film.Genre_ID = Genres.genreID
202     GROUP BY genreName
203     ORDER BY FilmCount DESC;
204
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

genreName	FilmCount
Science Fiction	2
Drama	2
Thriller	2
Mystery	2
War	2
Comedy	1

```
208
209 •   SELECT Film_ID, COUNT(Actor_ID) AS ActorCount
210     FROM Cast
211     GROUP BY Film_ID
212     HAVING COUNT(Actor_ID) > 2;
213
```

Result Grid | Filter Rows: | Edit: | Export/Import:

directorID	fName	lName	gender
104	Chris	Columbus	M
103	David	Fincher	M
106	Mel	Gibson	M
109	Hang-jun	Jang	M
105	Richard	Kelly	M
99	Christopher	Nolan	M
102	Alexander	Payne	M
110	Guy	Ritchie	M
111	Ridley	Scott	M
107	Denis	Villeneuve	M

statements:

```
209 •   SELECT Film_ID, COUNT(Actor_ID) AS ActorCount
210     FROM Cast
211     GROUP BY Film_ID
212     HAVING COUNT(Actor_ID) > 2;
213
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Film_ID	ActorCount
1	4
3	3
6	3
7	3
8	3
9	3
10	3
11	3

```
DELETE FROM Film -- Delet
WHERE Title = 'Black Hawk Down';
```

Result Grid | Filter Rows: | Edit: | Export/Impor

ID	Title	Language	ReleaseDate	Director_ID	Genre_ID
1	Interstellar	English	2014-10-26	99	1
2	The Holdovers	English	2023-10-27	102	2
3	Gone Girl	English	2014-10-03	103	3
4	Home Alone	English	1990-11-10	104	4
5	Donnie Darko	English	2001-10-26	105	5
6	Hacksaw Ridge	English	2016-10-16	106	6
7	Dune	English	2021-09-03	107	1
8	Aftersun	English	2022-10-21	108	2
9	Forgotten	Korean	2017-11-29	109	3
10	Sherlock Holmes	English	2009-12-25	110	5

statements:

```
220 •  SELECT Film.Title, Directors.fName, Directors.lName -- Inner join
221    FROM Film
222    INNER JOIN Directors
223    ON Film.Director_ID = Directors.directorID;
224
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Title	fName	lName
Interstellar	Christopher	Nolan
The Holdovers	Alexander	Payne
Gone Girl	David	Fincher
Home Alone	Chris	David us
Donnie Darko	Richard	Kelly
Hacksaw Ridge	Mel	Gibson
Dune	Denis	Villeneuve
Aftersun	Charlotte	Wells
Forgotten	Hang-jun	Jang
Sherlock Holmes	Guy	Ritchie

```
25 •  SELECT Fname, Lname
26    FROM Actor
27    WHERE ActorID = (SELECT MAX(ActorID) FROM Actor);
28
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

Fname	Lname
Ewan	McGregor

```
233 •  UPDATE Genres
234    SET genreName = 'Sci-Fi'
235    WHERE genreID = 1;
236
```

Result Grid | Filter Rows:

genreID	genreName
1	Sci-Fi
2	Drama
3	Thriller
4	Comedy
5	Mystery
6	War
NULL	NULL

statements:

```
209 •   SELECT Film_ID, COUNT(Actor_ID) AS ActorCount
210     FROM Cast
211     GROUP BY Film_ID
212     HAVING COUNT(Actor_ID) > 2;
213
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Film_ID	ActorCount
1	4
3	3
6	3
7	3
8	3
9	3
10	3
11	3

```
DELETE FROM Film -- Delet
WHERE Title = 'Black Hawk Down';
```

Result Grid | Filter Rows: | Edit: | Export/Impor

ID	Title	Language	ReleaseDate	Director_ID	Genre_ID
1	Interstellar	English	2014-10-26	99	1
2	The Holdovers	English	2023-10-27	102	2
3	Gone Girl	English	2014-10-03	103	3
4	Home Alone	English	1990-11-10	104	4
5	Donnie Darko	English	2001-10-26	105	5
6	Hacksaw Ridge	English	2016-10-16	106	6
7	Dune	English	2021-09-03	107	1
8	Aftersun	English	2022-10-21	108	2
9	Forgotten	Korean	2017-11-29	109	3
10	Sherlock Holmes	English	2009-12-25	110	5

statements:

```
237 •   UPDATE Actor
238     SET Lname = 'David'
239     WHERE ActorID = 1;
```

Result Grid | Filter Rows: | Edit:

ActorID	Fname	Lname	gender
1	Matthew	David	M
2	Anne	Hathaway	F
3	Jessica	Chastain	F
4	Mackenzie	Foy	F
5	Paul	Giamatti	M
6	Maya	Hawke	F
7	Ben	Affleck	M
8	Tyler	Perry	M
9	Rosamund	Pike	F
10	Macaulay	Culkin	M

```
192 •   SELECT *
193     FROM Rating
194     WHERE Ratings > 8;
195
196 •   DELETE FROM Rating
197     WHERE Film_ID = 11;
198
```

Result Grid | Filter Rows: | Edit:

ID	Film_ID	Ratings
1	1	9.5
3	3	9.0
4	4	8.5
5	5	9.0
6	6	8.5
7	7	8.5
8	8	8.5
9	9	8.5
10	10	8.5
NULL	NULL	NULL

Tasks Table:

Name	Name
Ghala Alsaedi	• Business Rules • Directors table • Order by statement • Update statement • UML diagram
Jumana Alsaedi	• Business Rules • Genres table • Group by statement • Update statement • UML diagram
Leena Alghamdi	• Business Rules • Rating Table • Where Statement • Delete Statement • Relational Schema
Layan alQurashi	• Business Rules • Actor table • SUBQUERIES statement • Update statement • Relational Schema
Bayan Alkhayri	• Business Rules • Film table • JOIN statement • DELETE statement • Chen model
Shahad Alkhuzai	• Business Rules • Cast table • HAVING statement • DELETE statement • Chen model