

Business Intelligence

Final Project Part 2 - MS SQL

Movies- Database Assignment

BY GROUP 10

Moganaviniith Rathinavel (200531668)

Aswin Narayanan (200568974)

Nanditha Babu (200573747)

Srivas Sathyaranayanan (200544678)

Raveen Kumar (200574798)

1.Create a database “Movies”.

```
CREATE DATABASE Movies;
```

2.Create a “mov” schema under “Movies”.

```
CREATE SCHEMA mov;
```

3.Create a table call “Movie_Director “under mov schema with the following specifications-

```
CREATE TABLE mov.Movie_Director (
    Director_ID INT IDENTITY(1,1) NOT NULL PRIMARY KEY CLUSTERED,
    Director_First_Name VARCHAR(255),
    Director_Last_Name VARCHAR(255),
    Director_Age_in_Years INT,
    Director_Gender VARCHAR(255), -- Assuming 'Director_Gender' is the correct column name
    and 'Date of Joining' was mistakenly described
    CreatedOn DATE NOT NULL DEFAULT GETDATE()
);
```

```
SET IDENTITY_INSERT mov.Movie_Director ON;
```

```
INSERT INTO mov.Movie_Director (Director_ID, Director_First_Name, Director_Last_Name,
Director_Age_in_Years, Director_Gender, CreatedOn)
```

```
VALUES
```

```
(100, 'Kevin', 'Smith', 52, 'Male', DEFAULT),
(110, 'Miguel', 'Arteta', 41, 'Male', DEFAULT),
(120, 'Mark', 'Johnson', 58, 'Male', DEFAULT),
(130, 'Tom', 'Vaughan', 37, 'Male', DEFAULT),
(140, 'Francis', 'Lawrence', 52, 'Male', DEFAULT),
(150, 'Adrienne', 'Shelly', 40, 'Female', DEFAULT),
(160, 'David', 'Slade', 53, 'Male', DEFAULT),
(170, 'Mark', 'Palansky', 53, 'Male', DEFAULT),
(180, 'Jeff', 'Lowell', 49, 'Male', DEFAULT),
(190, 'Simon', 'Curtis', 37, 'Male', DEFAULT),
(200, 'Marc', 'Lawrence', 95, 'Male', DEFAULT),
(210, 'Anand', 'Tucker', 59, 'Male', DEFAULT),
(220, 'Judd', 'Apatow', 55, 'Male', DEFAULT),
```

```

(230, 'Cary', 'Fukunaga', 45, 'Male', DEFAULT),
(240, 'Mark', 'Helfrich', 49, 'Male', DEFAULT),
(250, 'Nanette', 'Burstein', 52, 'Female', DEFAULT),
(260, 'James', 'McAvoy', 44, 'Male', DEFAULT),
(270, 'Mark', 'Waters', 58, 'Male', DEFAULT),
(280, 'Seth', 'Gordon', 46, 'Male', DEFAULT),
(290, 'Alex', 'Kendrick', 52, 'Male', DEFAULT),
(300, 'Kevin', 'Lima', 60, 'Male', DEFAULT),
(310, 'Lasse', 'Hallström', 76, 'Male', DEFAULT),
(320, 'Ewan', 'McGregor', 52, 'Male', DEFAULT),
(330, 'Rajkumar', 'Hirani', 60, 'Male', DEFAULT),
(340, 'Ashutosh', 'Gowariker', 59, 'Male', DEFAULT),
(350, 'Karan', 'Johar', 50, 'Male', DEFAULT),
(360, 'S.S', 'Rajamouli', 49, 'Male', DEFAULT),
(370, 'Sukumar', NULL, 53, 'Male', DEFAULT),
(380, 'Aditya', 'Chopra', 51, 'Male', DEFAULT),
(390, 'Umesh', 'Shukla', 52, 'Male', DEFAULT);

```

```
SET IDENTITY_INSERT mov.Movie_Director OFF;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer displays the database structure of 'MOGANAVINITH\SQLEXPRESS'. In the center, a query window titled 'SQLQuery2.sdl - MO...initmogan (53)' contains the following content:

```

7. Write the following Query based on the above datasets.

-- a. List all the Movies information from the Movies table.
SELECT * FROM mov.Movies;

-- b. List all the Director information from the Director table;
SELECT * FROM mov.Movie_Director;

```

The results pane shows the output of the first query, listing 39 rows of movie data:

Director_ID	Director_First_Name	Director_Last_Name	Director_Age_in_Years	Director_Gender	CreatedOn
4	Tom	Vaughan	52	Male	2024-03-20
5	Felicity	Lawrence	37	Male	2024-03-20
6	Adele	Shelly	40	Female	2024-03-20
7	Edie	Shade	53	Male	2024-03-20
8	Mark	Paltrow	53	Male	2024-03-20
9	Jeff	Lowell	49	Male	2024-03-20
10	Simon	Caine	37	Male	2024-03-20
11	Marc	Lawrence	96	Male	2024-03-20
12	Adam	Tucker	59	Male	2024-03-20
13	Judd	Apatow	55	Male	2024-03-20
14	Cary	Fukunaga	45	Male	2024-03-20
15	Mark	Helfrich	49	Male	2024-03-20
16	Ben	Burstein	52	Female	2024-03-20
17	James	McAvoy	44	Male	2024-03-20
18	Mark	Waters	58	Male	2024-03-20
19	Seth	Gordon	46	Male	2024-03-20
20	Alex	Kendrick	52	Male	2024-03-20
21	Kevin	Lima	60	Male	2024-03-20
22	Lasse	Hallström	76	Male	2024-03-20
23	Ewan	McGregor	52	Male	2024-03-20
24	Rajkumar	Hirani	60	Male	2024-03-20
25	Ashutosh	Gowariker	59	Male	2024-03-20
26	Karan	Johar	50	Male	2024-03-20
27	S.S	Rajamouli	49	Male	2024-03-20
28	Sukumar	Nambi	53	Male	2024-03-20
29	Aditya	Chopra	51	Male	2024-03-20
30	Umesh	Shukla	42	Male	2024-03-20

The status bar at the bottom indicates 'Query executed successfully.' and shows connection details: Connection name: MOGANAVINITH\SQLEXPRESS (1...), Moganaviniith\mogan (53) | master | 06:00:00 | 30 rows.

-- 4. Create a Movies table under mov schema with the following specifications-

```

CREATE TABLE mov.Movies (
    Movie_ID INT IDENTITY(1000,1) NOT NULL PRIMARY KEY CLUSTERED,
    Movie_Name VARCHAR(255),
    Movie_Released_Year INT,
    Movie_Lead_Studio VARCHAR(255),
    Movie_Language VARCHAR(255),
    Movie_Category VARCHAR(255),
    Movie_Duration_in_Min INT,
    Movie_Worldwide_Earning_in_$M DECIMAL(15,2),
    Movie_Type VARCHAR(255) NOT NULL CHECK (Movie_Type IN ('Hollywood', 'Bollywood')),
    Director_ID INT,

```

```

CreatedOn DATE NOT NULL DEFAULT GETDATE(),
FOREIGN KEY (Director_ID) REFERENCES mov.Movie_Director(Director_ID)
);

SET IDENTITY_INSERT mov.Movies ON;
INSERT INTO mov.Movies (Movie_ID, Movie_Name, Movie_Released_Year, Movie_Lead_Studio,
Movie_Language, Movie_Category, Movie_Duration_in_Min, Movie_Worldwide_Earning_in_$M,
Movie_Type, Director_ID) VALUES
(1000, 'Zack and Miri Make a Porno', 2008, 'The Weinstein Company', 'English', 'Romance',
101, 41.94, 'Hollywood', 100),
(1001, 'Youth in Revolt', 2010, 'The Weinstein Company', 'English', 'Comedy', 90, 19.62,
'Hollywood', 110),
(1002, 'When in Rome', 2010, 'Disney', 'English', 'Comedy', 91, 43.04, 'Hollywood', 120),
(1003, 'What Happens in Vegas', 2008, 'Fox', 'English', 'Comedy', 99, 219.37,
'Hollywood', 130),
(1004, 'Water For Elephants', 2011, '20th Century Fox', 'English', 'Drama', 120, 117.09,
'Hollywood', 140),
(1005, 'Waitress', 2007, 'Independent', 'English', 'Romance', 108, 22.18, 'Hollywood',
150),
(1006, 'Twilight', 2008, 'Summit', 'English', 'Romance', 122, 376.66, 'Hollywood', 160),
(1007, 'Penelope', 2008, 'Summit', 'English', 'Comedy', 144, 20.74, 'Hollywood', 170),
(1008, 'Over Her Dead Body', 2008, 'New Line', 'English', 'Comedy', 95, 20.71,
'Hollywood', 180),
(1009, 'My Week with Marilyn', 2011, 'The Weinstein Company', 'English', 'Drama', 99,
8.26, 'Hollywood', 190),
(1010, 'Music and Lyrics', 2007, 'Warner Bros.', 'English', 'Romance', 104, 145.9,
'Hollywood', 200),
(1011, 'Leap Year', 2010, 'Universal', 'English', 'Comedy', 100, 32.59, 'Hollywood',
210),
(1012, 'Knocked Up', 2007, 'Universal', 'English', 'Comedy', 129, 219, 'Hollywood', 220),
(1013, 'Jane Eyre', 2011, 'Universal', 'English', 'Romance', 120, 30.15, 'Hollywood',
230),
(1014, 'Good Luck Chuck', 2007, 'Lionsgate', 'English', 'Comedy', 101, 59.19,
'Hollywood', 240),
(1015, 'Going the Distance', 2010, 'Warner Bros.', 'English', 'Comedy', 103, 42.05,
'Hollywood', 250),
(1016, 'Gnomeo and Juliet', 2011, 'Disney', 'English', 'Animation', 84, 193.97,
'Hollywood', 260),
(1017, 'Ghosts of Girlfriends Past', 2009, 'Warner Bros.', 'English', 'Comedy', 100,
102.22, 'Hollywood', 270),
(1018, 'Four Christmases', 2008, 'Warner Bros.', 'English', 'Comedy', 88, 161.83,
'Hollywood', 280),
(1019, 'Fireproof', 2008, 'Independent', 'English', 'Drama', 122, 33.47, 'Hollywood',
290),
(1020, 'Enchanted', 2007, 'Disney', 'English', 'Comedy', 107, 340.49, 'Hollywood', 300),
(1021, 'Dear John', 2010, 'Sony', 'English', 'Drama', 108, 114.97, 'Hollywood', 310),
(1022, 'Beginners', 2011, 'Independent', 'English', 'Comedy', 105, 14.31, 'Hollywood',
320),
(1023, '3 Idiots', 2009, 'Vinod Chopra Films', 'Hindi', 'Comedy', 171, 4000, 'Bollywood',
330),
(1024, 'Lagaan', 2001, 'Aamir Khan Productions', 'Hindi', 'Romance', 224, 659,
'Bollywood', 340),
(1025, 'My Name Is Khan', 2010, 'Dharma Productions', 'Hindi', 'Drama', 165, 48.77,
'Bollywood', 350),
(1026, 'Baahubali', 2015, 'Arka Media Works', 'Telugu', 'Thriller', 159, 6500,
'Bollywood', 360),
(1027, 'Dilwale Dulhania Le Jayenge', 1995, 'Yash Chopra', 'Hindi', 'Romance', 189, 2000,
'Bollywood', 380),

```

```
(1028, 'Oh My God', 2012, '', 'Hindi', 'Comedy', 165, 1200, 'Bollywood', 390),
(1029, 'Pushpa', 2021, 'Mythri Movie Makers', 'Telugu', 'Drama', 179, 3730, 'Bollywood',
370);
SET IDENTITY_INSERT mov.Movies OFF;
```

The screenshot shows the SQL Server Management Studio interface. The left pane displays the Object Explorer with the connection to 'MOGANAVINITHI\SQLExpress'. The right pane shows the results of a query named 'SQLQuery2.sql' which lists all movies from the 'Movies' table. The results grid contains 30 rows of movie details, such as 'Movie_ID', 'Movie_Name', 'Movie_Released_Year', 'Movie_Lead_Studio', 'Movie_Language', 'Movie_Category', 'Movie_Duration_in_Min', 'Movie_Worldwide_Earnings_in_SM', and 'Movie_Type'. The results are grouped by 'Movie_Type' (e.g., Hollywood, Romance, Comedy). The bottom status bar indicates the connection is 'MOGANAVINITHI\SQLExpress (1...' and the current time is '2024-03-20 2:24:31 PM'.

Movie_ID	Movie_Name	Movie_Released_Year	Movie_Lead_Studio	Movie_Language	Movie_Category	Movie_Duration_in_Min	Movie_Worldwide_Earnings_in_SM	Movie_Type
1	1001 - The Artist Makes a Porno	2008	The Weinstein Company	English	Romantic	101	21.94	Hollywood
2	1001 - Youth in Revolt	2010	The Weinstein Company	English	Comedy	90	19.62	Hollywood
3	1002 - When in Rome	2010	Disney	English	Comedy	91	43.04	Hollywood
4	1003 - What Happens in Vegas	2008	Fox	English	Comedy	99	219.37	Hollywood
5	1004 - Water For Elephants	2011	20th Century Fox	English	Drama	120	117.09	Hollywood
6	1005 - Water for Elephants	2011	Independent	English	Romance	128	20.18	Hollywood
7	1006 - Twilight	2008	Burnett	English	Romance	122	370.68	Hollywood
8	1007 - Penelope	2008	Summit	English	Comedy	144	20.74	Hollywood
9	1008 - Over the Hedge	2006	New Line	English	Comedy	95	26.71	Hollywood
10	1009 - My Week with Marilyn	2011	The Weinstein Company	English	Drama	125	3.25	Hollywood
11	1010 - Music and Lyrics	2007	Warner Bros.	English	Romance	104	145.00	Hollywood
12	1011 - Leap Year	2010	Universal	English	Comedy	100	32.69	Hollywood
13	1012 - Knocked Up	2007	Universal	English	Comedy	129	219.09	Hollywood
14	1013 - Dear John	2009	Universal	English	Romance	130	30.15	Hollywood
15	1014 - Good Luck Chuck	2007	Lionsgate	English	Comedy	101	58.19	Hollywood
16	1015 - Going the Distance	2010	Warner Bros.	English	Comedy	103	42.05	Hollywood
17	1016 - Brothers and Juliet	2009	Universal	English	Romantic	84	160.25	Hollywood
18	1017 - The Diving Bell and the Butterfly	2007	Warner Bros.	English	Comedy	100	102.25	Hollywood
19	1018 - Four Christmases	2008	Warner Bros.	English	Comedy	88	161.83	Hollywood
20	1019 - Fireproof	2008	Independent	English	Drama	122	33.47	Hollywood
21	1020 - Enchanted	2007	Disney	English	Comedy	107	340.49	Hollywood
22	1021 - Dear John	2010	Bonni	English	Drama	108	114.97	Hollywood
23	1022 - Desperates	2011	Independent	English	Comedy	105	14.31	Hollywood
24	1023 - 3 Idiots	2009	Vinod Chopra Films	Hindi	Comedy	171	4000.00	Bollywood
25	1024 - Lageen	2001	Aamir Khan Productions	Hindi	Romance	224	650.00	Bollywood
26	1025 - My Name Is Khan	2010	Dharma Productions	Hindi	Drama	165	48.77	Bollywood
27	1026 - Black	2009	Akshay Kumar	Telugu	Thriller	150	650.00	Bollywood
28	1027 - Dilwale Dulhania Le Jayen	1995	Yash Chopra	Hindi	Romance	109	2000.00	Bollywood
29	1028 - Oh My God	2012	Hindi	Comedy	165	1200.00	Bollywood	

5.Create a Movie_Actor table under mov schema with the following specifications

```
CREATE TABLE mov.Movie_Actor (
    Actor_ID INT IDENTITY(10,1) NOT NULL PRIMARY KEY CLUSTERED,
    Actor_First_Name VARCHAR(255),
    Actor_Last_Name VARCHAR(255),
    Actor_Age_in_Years INT,
    Actor_Location VARCHAR(255),
    Movie_ID INT,
    CreatedOn DATE NOT NULL DEFAULT GETDATE(),
    CONSTRAINT fk_Movie_ID FOREIGN KEY (Movie_ID) REFERENCES mov.Movies(Movie_ID)
);
SET IDENTITY_INSERT mov.Movie_Actor ON;
INSERT INTO mov.Movie_Actor (Actor_ID, Actor_First_Name, Actor_Last_Name,
Actor_Age_in_Years, Actor_Location, Movie_ID) VALUES
(10, 'Seth', 'Rogen', 53, 'Los Angeles', 1000),
(11, 'Michael', 'Cera', 49, 'New York', 1001),
(12, 'Josh', 'Duhamel', 37, 'North Dakota', 1002),
(13, 'Jason', 'Sudeikis', 60, 'Kansas', 1003),
(14, 'Robert', 'Pattinson', 45, 'Los Angeles', 1004),
(15, 'Nathan', 'Fillion', 55, 'Canada', 1005),
(16, 'Robert', 'Pattinson', 45, 'Los Angeles', 1006),
(17, 'James', 'McAvoy', 49, 'Scotland', 1007),
(18, 'Paul', 'Rudd', 52, 'New York', 1008),
(19, 'Kenneth', 'Branagh', 44, 'Northern Ireland', 1009),
(20, 'Hugh', 'Grant', 58, 'London', 1010),
(21, 'Matthew', 'Goode', 46, 'England', 1011),
(22, 'Judd', 'Apatow', 58, 'Los Angeles', 1012),
(23, 'Michael', 'Fassbender', 46, 'Germany', 1013),
```

```

(24, 'Dane', 'Cook', 52, 'United States', 1014),
(25, 'Jason', 'Sudeikis', 60, 'Kansas', 1015),
(26, 'Kelly', 'Asbury', 76, 'United States', 1016),
(27, 'Matthew', 'McConaughey', 52, 'Los Angeles', 1017),
(28, 'Vince', 'Vaughn', 60, 'Minnesota', 1018),
(29, 'Kirk', 'Cameron', 59, 'United States', 1019),
(30, 'James', 'Marsden', 50, 'Columbia', 1020),
(31, 'Channing', 'Tatum', 58, 'Alabama', 1021),
(32, 'Mike', 'Mills', 37, 'New York', 1022),
(33, 'Aamir', 'Khan', 52, 'India', 1023),
(34, 'Aamir', 'Khan', 52, 'India', 1024),
(35, 'Shah Rukh', 'Khan', 53, 'India', 1025),
(36, 'Prabhas', NULL, 53, 'India', 1026),
(37, 'Allu', 'Arjun', 49, 'India', 1027),
(38, 'Shah Rukh', 'Khan', 53, 'India', 1028),
(39, 'Akshay', 'Kumar', 50, 'India', 1029);

```

```
SET IDENTITY_INSERT mov.Movie_Actor OFF;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center, there is a results grid displaying data from the Movie_Actor table. The columns are Actor_ID, Actor_First_Name, Actor_Last_Name, Actor_Age_in_Years, Actor_Location, Movie_ID, and CreatedOn. The data consists of 30 rows of actor information. To the left of the results grid is the Object Explorer, which shows the database structure. To the right is the Solution Explorer, showing a project named 'SQL Server Scripts' with a file named 'SQLQuery2.sql'. Below the results grid is a properties pane for the current connection.

Actor_ID	Actor_First_Name	Actor_Last_Name	Actor_Age_in_Years	Actor_Location	Movie_ID	CreatedOn
1	Seth	Rogen	53	Los Angeles	1000	2024-03-20
2	Michael	Cera	49	New York	1001	2024-03-20
3	Josh	Duhameh	37	North Dakota	1002	2024-03-20
4	Jason	Sudeikis	60	Kansas	1003	2024-03-20
5	Robert	Pattinson	45	Los Angeles	1004	2024-03-20
6	Nelson	Fifer	55	Canada	1005	2024-03-20
7	Robert	Pattinson	45	Los Angeles	1006	2024-03-20
8	James	McAvoy	49	Scotland	1007	2024-03-20
9	Paul	Rudd	52	New York	1008	2024-03-20
10	Kenneth	Brenagh	44	Northern Ireland	1009	2024-03-20
11	Hugh	Grant	58	London	1010	2024-03-20
12	Matthew	Goode	46	England	1011	2024-03-20
13	Jude	Apatoe	58	Los Angeles	1012	2024-03-20
14	Michael	Fassbender	46	Germany	1013	2024-03-20
15	Dane	Cook	52	United States	1014	2024-03-20
16	Jason	Sudeikis	60	Kansas	1015	2024-03-20
17	Kelly	Asbury	76	United States	1016	2024-03-20
18	Matthew	McConaughey	52	Los Angeles	1017	2024-03-20
19	Vince	Vaughn	60	Minnesota	1018	2024-03-20
20	Kirk	Cameron	59	United States	1019	2024-03-20
21	James	Marsden	50	Columbia	1020	2024-03-20
22	Channing	Tatum	58	Alabama	1021	2024-03-20
23	Mike	Mills	37	New York	1022	2024-03-20
24	Aamir	Khan	52	India	1023	2024-03-20
25	Aamir	Khan	52	India	1024	2024-03-20
26	Shah Rukh	Khan	53	India	1025	2024-03-20
27	Prabhas	NULL	53	India	1026	2024-03-20
28	Allu	Arjun	49	India	1027	2024-03-20
29	Shah Rukh	Khan	53	India	1028	2024-03-20
30	Akshay	Kumar	50	India	1029	2024-03-20

6.Create a Movie_Rating table under mov schema with the following specifications

```

CREATE TABLE mov.Movie_Rating (
    Movie_Rating_ID INT IDENTITY(1,1) NOT NULL PRIMARY KEY CLUSTERED,
    Rating_Audience_Score VARCHAR(255),
    Rating_Rotten_Tomatoes VARCHAR(255),
    Movie_ID INT,
    CreatedOn DATE NOT NULL DEFAULT GETDATE(),
    CONSTRAINT fk_MovieRating_Movie_ID FOREIGN KEY (Movie_ID) REFERENCES
    mov.Movies(Movie_ID));
    INSERT INTO mov.Movie_Rating (Rating_Audience_Score, Rating_Rotten_Tomatoes, Movie_ID)
    VALUES
    (70, 64, 1000),

```

```

(52, 68, 1001),
(44, 15, 1002),
(72, 28, 1003),
(72, 60, 1004),
(67, 89, 1005),
(82, 49, 1006),
(74, 52, 1007),
(47, 15, 1008),
(84, 83, 1009),
(70, 63, 1010),
(49, 21, 1011),
(83, 91, 1012),
(77, 85, 1013),
(61, 3, 1014),
(56, 53, 1015),
(52, 56, 1016),
(47, 27, 1017),
(52, 26, 1018),
(51, 40, 1019),
(80, 93, 1020),
(66, 29, 1021),
(80, 84, 1022),
(95, 100, 1023),
(81, 95, 1024),
(79, 83, 1025),
(80, 90, 1026),
(76, 82, 1027),
(85, 100, 1028),
(81, 74, 1029);

```

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The main area displays a query results grid for the 'mov.Movie_Rating' table. The grid has columns: Movie_Rating_ID, Rating_Audience_Score, Rating_Rotten_Tomatoes, Movie_ID, and CreatedOn. The data consists of 30 rows. Above the grid, two SQL queries are visible in the Query pane:

```

--e. List all the movies released in the year "2010".
SELECT * FROM mov.Movies WHERE Movie_Released_Year = 2010;

```

The bottom status bar indicates the query was executed successfully.

7. Write the following Query based on the above datasets.

a. List all the Movies information from the Movies table.

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with the database 'MOGANAVINITH\SQLEXPRESS' selected. The right pane shows the results of a query against the 'mov.Movies' table. The results grid contains 30 rows of movie information, including columns like Movie_ID, Movie_Name, Movie_Released_Year, and Movie_Type. The Properties pane on the right shows connection details for the current session.

Movie_ID	Movie_Name	Movie_Released_Year	Movie_Lead_Studio	Movie_Language	Movie_Category	Movie_Duration_in_Min	Movie_Worldwide_Earning_in_SM	Movie_Type
1	Zack and Miri Make a Porno	2008	The Weinstein Company	English	Romance	101	41.94	Hollywood
2	Youth in Revolt	2010	The Weinstein Company	English	Comedy	90	19.62	Hollywood
3	When in Rome	2010	Disney	English	Comedy	91	43.04	Hollywood
4	What Happens in Vegas	2008	Fox	English	Comedy	99	219.37	Hollywood
5	Water For Elephants	2011	20th Century Fox	English	Drama	120	117.09	Hollywood
6	Waitress	2007	Independent	English	Romance	108	22.18	Hollywood
7	Twilight	2008	Summit	English	Romance	122	376.66	Hollywood
8	Penelope	2008	Summit	English	Comedy	144	20.74	Hollywood
9	Over Her Dead Body	2008	New Line	English	Comedy	95	20.71	Hollywood
10	My Week with Marilyn	2011	The Weinstein Company	English	Drama	99	8.26	Hollywood
11	Music and Lyrics	2007	Warner Bros.	English	Romance	104	145.90	Hollywood
12	Leap Year	2010	Universal	English	Comedy	105	32.59	Hollywood
13	Kung Fu Panda	2007	Universal	English	Comedy	129	210.00	Hollywood
14	Jane Eyre	2011	Universal	English	Romance	120	30.15	Hollywood
15	Good Luck Chuck	2007	Lonegate	English	Comedy	101	68.18	Hollywood
16	Going the Distance	2010	Warner Bros.	English	Comedy	103	42.06	Hollywood
17	Gnomeo and Juliet	2011	Disney	English	Animation	84	193.97	Hollywood
18	Ghosts of Girlfriends Past	2009	Warner Bros.	English	Comedy	100	102.22	Hollywood
19	Four Christmases	2008	Warner Bros.	English	Comedy	88	161.83	Hollywood
20	Fireproof	2008	Independent	English	Drama	122	33.47	Hollywood
21	Enchanted	2007	Disney	English	Comedy	107	340.49	Hollywood
22	Dear John	2010	Sony	English	Drama	108	114.97	Hollywood
23	Beginners	2011	Independent	English	Comedy	105	14.31	Hollywood
24	3 Idiots	2009	Vinod Chopra Films	Hindi	Comedy	171	4000.00	Bollywood
25	Legaae	2007	Aamir Khan Productions	Hindi	Romance	224	600.00	Bollywood
26	My Name Is Khan	2010	Chhotta Productions	Hindi	Drama	160	48.77	Bollywood
27	Bodyroll	2015	Aka Media Works	Telugu	Thriller	159	4500.00	Bollywood
28	Dilwale Dulhania Le Jayenge	1995	Yash Chopra	Hindi	Romance	109	2000.00	Bollywood
29	Oh My God	2012		English	Comedy	165	1200.00	Bollywood

b. List all the Director information from the Director table.

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "SQL Server Scripts3 - Microsoft SQL Server Management Studio". The menu bar includes File, Edit, View, Query, Project, Tools, Window, Help. The toolbar has icons for New Query, Execute, Save, Undo, Redo, Copy, Paste, Find, Replace, and others. The Object Explorer on the left shows "Registered Servers" with "Database Engine" expanded, showing "Local Server Groups" and "Central Management Servers". The Solution Explorer on the right shows a solution named "MOGANAVINITH\SQLSERVEXPRESS (1 project)" with files like "Connections", "Queries", "Miscellaneous", and "Miscellaneous Files". The main area displays a query window titled "SQLQuery2.sql - MOGANAVINITH\mogan (53) - x" containing two queries:

```
USE [master]
GO
-- a. List all the movie information from the movies table.
SELECT * FROM mov.Movies;
-- b. List all the Director information from the Director table.
SELECT * FROM mov.Movie_Director;
```

The results grid shows the output of the second query, listing 30 rows of director information:

Director_ID	Director_First_Name	Director_Last_Name	Director_Age_in_Years	Director_Gender	CreatedOn
100	Kevin	Smith	41	Male	2024-03-20
3	Miguel	Arteta	52	Male	2024-03-20
3	Mark	Johnson	58	Male	2024-03-20
120	Tom	Vaughan	37	Male	2024-03-20
5	Francis	Lawrence	52	Male	2024-03-20
6	Adrienne	Shelly	40	Female	2024-03-20
7	Dawn	Slade	53	Male	2024-03-20
8	Mark	Palensky	53	Male	2024-03-20
9	Jeff	Lowell	49	Male	2024-03-20
10	Simon	Curtis	37	Male	2024-03-20
11	Mark	Lorraine	59	Male	2024-03-20
12	Travis	Tucker	59	Male	2024-03-20
13	Anand	Judd	55	Male	2024-03-20
14	Cary	Apatow	45	Male	2024-03-20
14	Mark	Fukunaga	45	Male	2024-03-20
15	David	Heldrich	49	Male	2024-03-20
16	Nicole	Burnett	52	Female	2024-03-20
17	James	McAvoy	44	Male	2024-03-20
18	Mark	Waters	58	Male	2024-03-20
19	Seth	Gordon	46	Male	2024-03-20
20	Alex	Kendrick	52	Male	2024-03-20
21	Kevin	Lima	60	Male	2024-03-20
22	Lasse	Hillestrom	76	Male	2024-03-20
23	Ewan	McGregor	52	Male	2024-03-20
24	Rajkumar	Hirani	60	Male	2024-03-20
25	Shahrukh	Gawariker	59	Male	2024-03-20
26	Karan	Johar	50	Male	2024-03-20
27	S.S	Rajamouli	49	Male	2024-03-20
28	Sukumar	NULL	53	Male	2024-03-20
29	Aditya	Chopra	91	Male	2024-03-20
30	Umesh	Shukla	52	Male	2024-03-20

The status bar at the bottom indicates "Query executed successfully." and "MOGANAVINITH\SQLSERVEXPRESS (1... Moganavinith\mogan (53) master 00:00:00 30 rows". The properties pane on the right shows connection details for "MOGANAVINITH\SQLSERVEXPRESS".

c. List all the Actor information from the Actor table.

The screenshot shows the Microsoft SQL Server Management Studio interface. The main window displays a query results grid for a SELECT statement from the Movie_Director table. The results grid has columns: Actor_ID, Actor_First_Name, Actor_Last_Name, Actor_Age_in_Years, Actor_Location, Movie_ID, and Column1. The data includes rows for actors like Michael, Jason, and Robert. Below the results grid, a status bar indicates "Query executed successfully." To the left is the Object Explorer showing database structures, and to the right is the Solution Explorer displaying the project structure.

Actor_ID	Actor_First_Name	Actor_Last_Name	Actor_Age_in_Years	Actor_Location	Movie_ID	Column1
1	Uma	Tan	49	Hong Kong	1001	2024-03-20
2	Michael	Gere	49	New York	1002	2024-03-20
3	Josh	Duhamek	37	North Dakota	1003	2024-03-20
4	Jason	Sudeikis	60	Kansas	1004	2024-03-20
5	Robert	Pattinson	45	Los Angeles	1005	2024-03-20
6	Nathan	Fillion	55	Canada	1006	2024-03-20
7	Robert	Pattinson	45	Los Angeles	1007	2024-03-20
8	James	McAvoy	49	Scotland	1008	2024-03-20
9	Paul	Rudd	52	New York	1009	2024-03-20
10	Kenneth	Branagh	44	Northern Ireland	1010	2024-03-20
11	Hugh	Grant	58	London	1011	2024-03-20
12	Matthew	Goode	46	England	1012	2024-03-20
13	Jude	Apatow	58	Los Angeles	1013	2024-03-20
14	Michael	Fassbender	46	Germany	1014	2024-03-20
15	Dane	Cook	52	United States	1015	2024-03-20
16	Jason	Sudeikis	60	Kansas	1016	2024-03-20
17	Kelly	Ashbury	76	United States	1017	2024-03-20
18	Matthew	McConaughey	62	Los Angeles	1018	2024-03-20
19	Vince	Vaughn	60	Minnesota	1019	2024-03-20
20	Kirk	Cameron	59	United States	1020	2024-03-20
21	James	Marsden	50	Columbia	1021	2024-03-20
22	Channing	Tatum	58	Alabama	1022	2024-03-20
23	Mike	Mills	37	New York	1023	2024-03-20
24	Aamir	Khan	52	India	1024	2024-03-20
25	Aamir	Khan	52	India	1025	2024-03-20
26	Shah Rukh	Khan	53	India	1026	2024-03-20
27	Prabhas	NULL	53	India	1027	2024-03-20
28	Allu	Ajay	49	India	1028	2024-03-20
29	Shah Rukh	Khan	53	India	1029	2024-03-20
30	Akshay	Kumar	50	India	1030	2024-03-20

d. List all the Rating information from the Rating table.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```
--e. List all the movie released in year "2010".
SELECT * FROM mov.Movies WHERE Movie_Released_Year = 2010;
```

Movie_ID	Movie_Name	Movie_Released_Year	Movie_Lead_Studio	Movie_Language	Movie_Category	Movie_Duration_in_Min	Movie_Worldwide_Earning_in_\$M	Movie_Type	Director_ID
1001	When in Rome	2010	The Weinstein Company	English	Comedy	90	19.62	Hollywood	110
1002	When in Rome	2010	Disney	English	Comedy	91	43.04	Hollywood	120
1011	Leap Year	2010	Universal	English	Comedy	100	32.59	Hollywood	210
1015	Going the Distance	2010	Warner Bros.	English	Comedy	103	42.05	Hollywood	250
1021	Dear John	2010	Sony	English	Drama	108	114.97	Hollywood	310
1025	My Name is Khan	2010	Dharma Productions	Hindi	Drama	165	40.77	Bollywood	350

Query executed successfully.

e. List all the movie released in year “2010”.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```
--e. List all the movies released in year "2010".
SELECT * FROM mov.Movies WHERE Movie_Released_Year = 2010;
```

```
--f. List all the movies released by "The Weinstein Company" studio.
SELECT * FROM mov.Movies WHERE Movie_Lead_Studio = 'The Weinstein Company';
```

```
--g. List all the movies released in "English".
SELECT * FROM mov.Movies WHERE Movie_Language = 'English';
```

```
--h. List all the movies whose name starts with 'G'.
SELECT * FROM mov.Movies WHERE Movie_Name LIKE 'G%';
```

```
--i. Display all the movies under "Comedy" category.
SELECT * FROM mov.Movies WHERE Movie_Category = 'Comedy';
```

```
--j. Display all the movies where movie type is "Hollywood".
SELECT * FROM mov.Movies WHERE Movie_Type = 'Hollywood';
```

Movie_ID	Movie_Name	Movie_Released_Year	Movie_Lead_Studio	Movie_Language	Movie_Category	Movie_Duration_in_Min	Movie_Worldwide_Earning_in_\$M	Movie_Type	Director_ID
1001	When in Rome	2010	The Weinstein Company	English	Comedy	90	19.62	Hollywood	110
1002	When in Rome	2010	Disney	English	Comedy	91	43.04	Hollywood	120
1011	Leap Year	2010	Universal	English	Comedy	100	32.59	Hollywood	210
1015	Going the Distance	2010	Warner Bros.	English	Comedy	103	42.05	Hollywood	250
1021	Dear John	2010	Sony	English	Drama	108	114.97	Hollywood	310
1025	My Name is Khan	2010	Dharma Productions	Hindi	Drama	165	40.77	Bollywood	350

Query executed successfully.

f. List all the movie released by “The Weinstein Company” studio.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```
--e. List all the movies released in the year "2010".
SELECT * FROM mov.Movies WHERE Movie_Released_Year = 2010;

--f. List all the movies released by "The Weinstein Company" studio.
SELECT * FROM mov.Movies WHERE Movie_Lead_Studio = 'The Weinstein Company';

--g. List all the movies released in "English".
SELECT * FROM mov.Movies WHERE Movie_Language = 'English';

--h. List all the movies whose name starts with 'G'.
SELECT * FROM mov.Movies WHERE Movie_Name LIKE 'G%';

--i. Display all the movies under "Comedy" category.
SELECT * FROM mov.Movies WHERE Movie_Category = 'Comedy';

--j. Display all the movies where movie type is "Hollywood".
SELECT * FROM mov.Movies WHERE Movie_Type = 'Hollywood';
```

Movie_ID	Movie_Name	Movie_Released_Year	Movie_Lead_Studio	Movie_Language	Movie_Category	Movie_Duration_in_Min	Movie_Worldwide_Earning_in_SM	Movie_Type	Director
1000	Zack and Miri Make a Porno	2008	The Weinstein Company	English	Romance	101	41.94	Hollywood	100
1001	Youth in Revolt	2010	The Weinstein Company	English	Comedy	90	19.62	Hollywood	110
1009	My Week with Marilyn	2011	The Weinstein Company	English	Drama	99	8.26	Hollywood	190

Query executed successfully.

g. List all the movie released in “English”.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```
--e. List all the movies released in the year "2010".
SELECT * FROM mov.Movies WHERE Movie_Released_Year = 2010;

--f. List all the movies released by "The Weinstein Company" studio.
SELECT * FROM mov.Movies WHERE Movie_Lead_Studio = 'The Weinstein Company';

--g. List all the movies released in "English".
SELECT * FROM mov.Movies WHERE Movie_Language = 'English';

--h. List all the movies whose name starts with 'G'.
SELECT * FROM mov.Movies WHERE Movie_Name LIKE 'G%';

--i. Display all the movies under "Comedy" category.
SELECT * FROM mov.Movies WHERE Movie_Category = 'Comedy';

--j. Display all the movies where movie type is "Hollywood".
SELECT * FROM mov.Movies WHERE Movie_Type = 'Hollywood';
```

Movie_ID	Movie_Name	Movie_Released_Year	Movie_Lead_Studio	Movie_Language	Movie_Category	Movie_Duration_in_Min	Movie_Worldwide_Earning_in_SM	Movie_Type	Director
1000	Zack and Miri Make a Porno	2008	The Weinstein Company	English	Romance	101	41.94	Hollywood	100
1001	Youth in Revolt	2010	The Weinstein Company	English	Comedy	90	19.62	Hollywood	110
1002	When in Rome	2010	Disney	English	Comedy	91	10.34	Hollywood	120
1003	Water for Elephants	2011	20th Century Fox	English	Drama	120	117.09	Hollywood	130
1005	Water for Elephants	2011	20th Century Fox	English	Drama	120	117.09	Hollywood	130
1006	Twilight	2008	Summit	English	Romance	122	376.66	Hollywood	140
1007	Penelope	2008	Summit	English	Comedy	144	20.74	Hollywood	150
1008	Over Her Dead Body	2008	New Line	English	Comedy	95	20.71	Hollywood	160
1009	My Week with Marilyn	2011	The Weinstein Company	English	Drama	99	8.26	Hollywood	170
1010	Miranda and Lorraine	2007	Warner Bros	Romanian	104	145.93	Hollywood	180	

Query executed successfully.

h. List all the movie whose name starts with ‘G’.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```
--e. List all the movies released in the year "2010".
SELECT * FROM mov.Movies WHERE Movie_Released_Year = 2010;

--f. List all the movies released by "The Weinstein Company" studio.
SELECT * FROM mov.Movies WHERE Movie_Lead_Studio = 'The Weinstein Company';

--g. List all the movies released in "English".
SELECT * FROM mov.Movies WHERE Movie_Language = 'English';

--h. List all the movies whose name starts with 'G'.
SELECT * FROM mov.Movies WHERE Movie_Name LIKE 'G%';

--i. Display all the movies under "Comedy" category.
SELECT * FROM mov.Movies WHERE Movie_Category = 'Comedy';

--j. Display all the movies where movie type is "Hollywood".
SELECT * FROM mov.Movies WHERE Movie_Type = 'Hollywood';
```

Query executed successfully.

Movie_ID	Movie_Name	Movie_Released_Year	Movie_Lead_Studio	Movie_Language	Movie_Category	Movie_Duration_in_Min	Movie_Worldwide_Earning_in_SM	Movie_Type	Director_ID
1014	Good Luck Chuck	2007	Lionsgate	English	Comedy	101	59.19	Hollywood	240
1015	Going the Distance	2010	Warner Bros.	English	Comedy	103	42.05	Hollywood	250
1016	Gnomeo and Juliet	2011	Disney	English	Animation	84	193.87	Hollywood	260
1017	Ghosts of Girlfriends Past	2009	Warner Bros.	English	Comedy	100	102.22	Hollywood	270

i. Display all the movie under “Comedy” category.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```
--g. List all the movies released in English .
SELECT * FROM mov.Movies WHERE Movie_Language = 'English';

--h. List all the movies whose name starts with 'G'.
SELECT * FROM mov.Movies WHERE Movie_Name LIKE 'G%';

--i. Display all the movies under "Comedy" category.
SELECT * FROM mov.Movies WHERE Movie_Category = 'Comedy';

--j. Display all the movies where movie type is "Hollywood".
SELECT * FROM mov.Movies WHERE Movie_Type = 'Hollywood';

--k. Display all the "Female" directors.
SELECT * FROM mov.Movie_Director WHERE Director_Gender = 'Female';

--l. Display all the directors whose Age is more than 45 years.
SELECT * FROM mov.Movie_Director WHERE Director_Age_in_Years > 45;

--m. Display all the Actors from "Los Angeles".
SELECT * FROM mov.Movie_Actor WHERE Actor_Location = 'Los Angeles';
```

Query executed successfully.

Movie_ID	Movie_Name	Movie_Released_Year	Movie_Lead_Studio	Movie_Language	Movie_Category	Movie_Duration_in_Min	Movie_Worldwide_Earning_in_SM	Movie_Type	Director_ID	CreatedOn
1001	Love Happens	2010	The Weinstein Company	English	Comedy	98	10.00	Hollywood	110	2024-03-20
1002	When in Rome	2010	Disney	English	Comedy	91	43.04	Hollywood	120	2024-03-20
1003	What Happens in Vegas	2008	Fox	English	Comedy	99	219.37	Hollywood	130	2024-03-20
1007	Penelope	2008	Summit	English	Comedy	144	20.74	Hollywood	170	2024-03-20
1008	Over Her Dead Body	2008	New Line	English	Comedy	95	20.71	Hollywood	180	2024-03-20
1011	Leap Year	2010	Universal	English	Comedy	100	32.59	Hollywood	210	2024-03-20
1012	Knocked Up	2007	Universal	English	Comedy	129	219.00	Hollywood	220	2024-03-20
1014	Good Luck Chuck	2007	Lionsgate	English	Comedy	101	59.19	Hollywood	240	2024-03-20
1015	Going the Distance	2010	Warner Bros.	English	Comedy	103	42.05	Hollywood	250	2024-03-20
1017	Ghosts of Girlfriends Past	2009	Warner Bros.	English	Comedy	100	102.22	Hollywood	270	2024-03-20
1018	Four Christmases	2008	Warner Bros.	English	Comedy	88	161.83	Hollywood	280	2024-03-20

j. Display all the movie where movie type is “Hollywood”.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```
--i. Display all the movies under "Comedy" category.
SELECT * FROM mov.Movies WHERE Movie_Category = 'Comedy';

--j. Display all the movies where movie type is "Hollywood".
SELECT * FROM mov.Movies WHERE Movie_Type = 'Hollywood';

--k. Display all the "Female" directors.
SELECT * FROM mov.Movie_Director WHERE Director_Gender = 'Female';

--l. Display all the directors whose Age is more than 45 years.
SELECT * FROM mov.Movie_Director WHERE Director_Age_in_Years > 45;

--m. Display all the Actors from "Los Angeles".
SELECT * FROM mov.Movie_Actor WHERE Actor_Location = 'Los Angeles';
--n. Display all the Actors whose Age is less than 50 years.
SELECT * FROM mov.Movie_Actor WHERE Actor_Age_in_Years < 50;
```

Results

Movie_ID	Movie_Name	Movie_Released_Year	Movie_Lead_Studio	Movie_Language	Movie_Category	Movie_Duration_in_Min	Movie_Worldwide_Earning_in_SM	Movie_Type	Director_ID	CreatedOn
1000	Zack and Miri Make a Porno	2008	The Weinstein Company	English	Romance	101	41.94	Hollywood	100	2024-03-2
1001	Youth in Revolt	2010	The Weinstein Company	English	Comedy	90	19.63	Hollywood	110	2024-03-2
1002	When in Rome	2010	Disney	English	Comedy	91	43.04	Hollywood	120	2024-03-2
1003	What Happens in Vegas	2008	Fox	English	Comedy	99	219.37	Hollywood	130	2024-03-2
1004	Water For Elephants	2011	20th Century Fox	English	Drama	120	117.09	Hollywood	140	2024-03-2
1005	Wal-Mart	2007	Independent	English	Romance	108	22.18	Hollywood	150	2024-03-2
1006	Twilight	2008	Summit	English	Romance	122	376.68	Hollywood	160	2024-03-2
1007	Penelope	2008	Summit	English	Comedy	144	20.74	Hollywood	170	2024-03-2
1008	Over Her Dead Body	2008	New Line	English	Comedy	95	20.71	Hollywood	180	2024-03-2
1009	My Week with Marilyn	2011	The Weinstein Company	English	Drama	99	8.26	Hollywood	190	2024-03-2

Query executed successfully.

k. Display all the “Female” directors.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```
--i. Display all the movies under "Comedy" category.
SELECT * FROM mov.Movies WHERE Movie_Category = 'Comedy';

--j. Display all the movies where movie type is "Hollywood".
SELECT * FROM mov.Movies WHERE Movie_Type = 'Hollywood';

--k. Display all the "Female" directors.
SELECT * FROM mov.Movie_Director WHERE Director_Gender = 'Female';

--l. Display all the directors whose Age is more than 45 years.
SELECT * FROM mov.Movie_Director WHERE Director_Age_in_Years > 45;

--m. Display all the Actors from "Los Angeles".
SELECT * FROM mov.Movie_Actor WHERE Actor_Location = 'Los Angeles';
--n. Display all the Actors whose Age is less than 50 years.
SELECT * FROM mov.Movie_Actor WHERE Actor_Age_in_Years < 50;
```

Results

Director_ID	Director_First_Name	Director_Last_Name	Director_Age_in_Years	Director_Gender	CreatedOn
1	Adrienne	Belly	40	Female	2024-03-20
2	Nanette	Burstein	52	Female	2024-03-20

Query executed successfully.

l. Display all the director whose Age is more than 45 years.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```
--i. List all the movies whose name starts with "G".
SELECT * FROM mov.Movies WHERE Movie_Name LIKE 'G%';

--i. Display all the movies under "Comedy" category.
SELECT * FROM mov.Movies WHERE Movie_Category = 'Comedy';

--j. Display all the movies where movie type is "Hollywood".
SELECT * FROM mov.Movies WHERE Movie_Type = 'Hollywood';

--k. Display all the "Female" directors.
SELECT * FROM mov.Movie_Director WHERE Director_Gender = 'Female';

--l. Display all the directors whose Age is more than 45 years.
SELECT * FROM mov.Movie_Director WHERE Director_Age_in_Years > 45;

--m. Display all the Actors from "Los Angeles".
SELECT * FROM mov.Movie_Actor WHERE Actor_Location = 'Los Angeles';
--n. Display all the Actors whose Age is less than 50 years.
SELECT * FROM mov.Movie_Actor WHERE Actor_Age_in_Years < 50;
```

Results

Director_ID	Actor_First_Name	Director_Last_Name	Director_Age_in_Years	Director_Gender	CreatedOn
1	John	Smith	62	Male	2024-03-20
2	Mark	Johson	58	Male	2024-03-20
3	Francis	Lawrence	52	Male	2024-03-20
4	David	Blade	53	Male	2024-03-20
5	Mark	Palensky	53	Male	2024-03-20
6	Jeff	Lowell	49	Male	2024-03-20
7	Marc	Lawrence	95	Male	2024-03-20
8	Anand	Tucker	59	Male	2024-03-20
9	Judd	Apatow	55	Male	2024-03-20
10	Mark	Helfrich	49	Male	2024-03-20
11	Nanette	Burstein	52	Female	2024-03-20

Query executed successfully.

m. Display all the Actors from “Los Angeles”.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```
--j. Display all the movies where movie type is "Hollywood".
SELECT * FROM mov.Movies WHERE Movie_Type = 'Hollywood';

--k. Display all the "Female" directors.
SELECT * FROM mov.Movie_Director WHERE Director_Gender = 'Female';

--l. Display all the directors whose Age is more than 45 years.
SELECT * FROM mov.Movie_Director WHERE Director_Age_in_Years > 45;

--m. Display all the Actors from "Los Angeles".
SELECT * FROM mov.Movie_Actor WHERE Actor_Location = 'Los Angeles';

--n. Display all the Actors whose Age is less than 50 years.
SELECT * FROM mov.Movie_Actor WHERE Actor_Age_in_Years < 50;

--o. Display all the Actors whose name starts from "J".
SELECT * FROM mov.Movie_Actor WHERE Actor_First_Name LIKE 'J%';

--p. Display all the Actors who are from "Los Angeles" or "New York".
SELECT * FROM mov.Movie_Actor WHERE Actor_Location IN ('Los Angeles', 'New York');
```

Results

Actor_ID	Actor_First_Name	Actor_Last_Name	Actor_Age_in_Years	Actor_Location	Movie_ID	CreatedOn
1	John	Smith	62	Los Angeles	1001	2024-03-20
2	Robert	Pattinson	45	Los Angeles	1004	2024-03-20
3	Robert	Pattinson	45	Los Angeles	1006	2024-03-20
4	Judd	Apatow	58	Los Angeles	1012	2024-03-20
5	Matthew	McConaughey	52	Los Angeles	1017	2024-03-20

Query executed successfully.

n. Display all the Actor whose Age is less than 50 years.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```
--j. Display all the movies where movie_type is "Hollywood".
SELECT * FROM mov.Movies WHERE Movie_Type = 'Hollywood';

--k. Display all the "Female" directors.
SELECT * FROM mov.Movie_Director WHERE Director_Gender = 'Female';

--l. Display all the directors whose Age is more than 45 years.
SELECT * FROM mov.Movie_Director WHERE Director_Age_in_Years > 45;

--m. Display all the Actors from "Los Angeles".
SELECT * FROM mov.Movie_Actor WHERE Actor_Location = 'Los Angeles';

--n. Display all the Actors whose Age is less than 50 years.
SELECT * FROM mov.Movie_Actor WHERE Actor_Age_in_Years < 50;

--o. Display all the Actors whose name starts from "J".
SELECT * FROM mov.Movie_Actor WHERE Actor_First_Name LIKE 'J%';

--p. Display all the Actors who are from "Los Angeles" or "New York".
SELECT * FROM mov.Movie_Actor WHERE Actor_Location IN ('Los Angeles', 'New York');
```

Actor_ID	Actor_First_Name	Actor_Last_Name	Actor_Age_in_Years	Actor_Location	Movie_ID	CreatedOn
1	Levi	Moseley	49	New York	1001	2024-03-20
2	Josh	Duhameel	37	North Dakota	1002	2024-03-20
3	Robert	Pattinson	45	Los Angeles	1004	2024-03-20
4	Robert	Pattinson	45	Los Angeles	1006	2024-03-20
5	James	McAvoy	49	Scotland	1007	2024-03-20
6	Kenneth	Branagh	44	Northern Ireland	1009	2024-03-20
7	Matthew	Goode	46	England	1011	2024-03-20
8	Michael	Fassbender	46	Germany	1013	2024-03-20
9	Mike	Mills	37	New York	1022	2024-03-20
10	Allu	Arun	49	India	1027	2024-03-20

Query executed successfully.

o. Display all the Actor whose name is starts from “J”.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```
--l. Display all the Directors whose Age is more than 45 years.
SELECT * FROM mov.Movie_Director WHERE Director_Age_in_Years > 45;

--m. Display all the Actors from "Los Angeles".
SELECT * FROM mov.Movie_Actor WHERE Actor_Location = 'Los Angeles';

--n. Display all the Actors whose Age is less than 50 years.
SELECT * FROM mov.Movie_Actor WHERE Actor_Age_in_Years < 50;

--o. Display all the Actors whose name starts from "J".
SELECT * FROM mov.Movie_Actor WHERE Actor_First_Name LIKE 'J%';

--p. Display all the Actors who are from "Los Angeles" or "New York".
SELECT * FROM mov.Movie_Actor WHERE Actor_Location IN ('Los Angeles', 'New York');

--q. List Director_FullName, Director_Age_in_Years, Director_Gender from Director
SELECT Director_First_Name + ' ' + Director_Last_Name AS Director_FullName, Director_Age_in_Years, Director_Gender FROM mov.Movie_Director;

--r. List Director_FullName, Director_Age_in_Years, Director_Gender from Director whose Age is less than 45 years.
SELECT Director_First_Name + ' ' + Director_Last_Name AS Director_FullName, Director_Age_in_Years, Director_Gender FROM mov.Movie_Director WHERE Director_Age_in_Years < 45;
```

Actor_ID	Actor_First_Name	Actor_Last_Name	Actor_Age_in_Years	Actor_Location	Movie_ID	CreatedOn
1	Levi	Moseley	49	Hart Dakota	1001	2024-03-20
2	Jason	Sudeikis	60	Kansas	1003	2024-03-20
3	James	McAvoy	49	Scotland	1007	2024-03-20
4	Judd	Apatow	58	Los Angeles	1012	2024-03-20
5	Jason	Sudeikis	60	Kansas	1015	2024-03-20
6	James	Marsden	50	Columbia	1020	2024-03-20

Query executed successfully.

p. Display all the Actor who is from “Los Angeles” or “New York”.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```

--n. Display all the Actors whose Age is less than 50 years.
SELECT * FROM mov.Movie_Actor WHERE Actor_Age_in_Years < 50;

--o. Display all the Actors whose name starts from "J".
SELECT * FROM mov.Movie_Actor WHERE Actor_First_Name LIKE 'J%';

--p. Display all the Actors who are from "Los Angeles" or "New York".
SELECT * FROM mov.Movie_Actor WHERE Actor_Location IN ('Los Angeles', 'New York');

--q. List Director_FullName, Director_Age_in_Years, Director_Gender from Director
SELECT Director_First_Name + ' ' + Director_Last_Name AS Director_FullName, Director_Age_in_Years, Director_Gender FROM mov.Movie_Director;

--r. List Director_FullName, Director_Age_in_Years, Director_Gender from Director whose Age is less than 45 years.
SELECT Director_First_Name + ' ' + Director_Last_Name AS Director_FullName, Director_Age_in_Years, Director_Gender FROM mov.Movie_Director WHERE Director_Age_in_Years < 45;

-- 8. Write the following Query based on the above datasets.
SELECT m.Movie_Name, a.Actor_First_Name, a.Actor_Last_Name
FROM mov.Movie m
JOIN mov.Movie_Actor a ON m.Movie_ID = a.Movie_ID
JOIN mov.Movie_Director d ON m.Movie_ID = d.Movie_ID
WHERE a.Actor_Location IN ('Los Angeles', 'New York') AND d.Director_Age_in_Years < 45;

```

Query executed successfully.

Actor_ID	Actor_First_Name	Actor_Last_Name	Actor_Age_in_Years	Actor_Location	Movie_ID	CreatedOn
10	Seth	Rogen	63	Los Angeles	1000	2024-03-20
11	Michael	Cera	49	New York	1001	2024-03-20
14	Robert	Pattinson	45	Los Angeles	1004	2024-03-20
16	Robert	Pattinson	45	Los Angeles	1006	2024-03-20
18	Paul	Rudd	52	New York	1008	2024-03-20
22	Judd	Apatow	58	Los Angeles	1012	2024-03-20
27	Matthew	McConaughey	52	Los Angeles	1017	2024-03-20
32	Mike	Mills	37	New York	1022	2024-03-20

q. List Director_FullName, Director_Age_in_Years, Director_Gender from Director

[Director_FullName = Director_First_Name + " " + Director_Last_Name]

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```

--n. Display all the Actors whose Age is less than 50 years.
SELECT * FROM mov.Movie_Actor WHERE Actor_Age_in_Years < 50;

--o. Display all the Actors whose name starts from "J".
SELECT * FROM mov.Movie_Actor WHERE Actor_First_Name LIKE 'J%';

--p. Display all the Actors who are from "Los Angeles" or "New York".
SELECT * FROM mov.Movie_ACTOR WHERE Actor_Location IN ('Los Angeles', 'New York');

--q. List Director_FullName, Director_Age_in_Years, Director_Gender from Director
SELECT Director_First_Name + ' ' + Director_Last_Name AS Director_FullName, Director_Age_in_Years, Director_Gender FROM mov.Movie_Director;

--r. List Director_FullName, Director_Age_in_Years, Director_Gender from Director whose Age is less than 45 years.
SELECT Director_First_Name + ' ' + Director_Last_Name AS Director_FullName, Director_Age_in_Years, Director_Gender FROM mov.Movie_Director WHERE Director_Age_in_Years < 45;

```

Query executed successfully.

Director_FullName	Director_Age_in_Years	Director_Gender
Kevin Smith	52	Male
Miguel Arteta	41	Male
Mark Johnson	58	Male
Tom Vaughan	37	Male
Francis Lawrence	52	Male
Adrienne Shelly	40	Female
David O'Russell	63	Male
Mark Palansky	53	Male
Jeff Lowell	49	Male
Simon Cutts	37	Male
Marc Lawrence	95	Male
Anand Tucker	59	Male
Judd Apatow	55	Male
Cary Fukunaga	45	Male
Mark Helrich	49	Male
Nanette Burstein	52	Female
James McAvoy	44	Male
Mark Waters	58	Male

r. List Director_FullName, Director_Age_in_Years, Director_Gender from Director whose Age is less than 45 years. [Director_FullName = Director_First_Name + " " + Director_Last_Name]

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```

--1. Display all the actors whose name starts with 'J' .
SELECT * FROM mov.Movie_Actor WHERE Actor_First_Name LIKE 'J%';

--2. Display all the Actors who are from "Los Angeles" or "New York".
SELECT * FROM mov.Movie_Actor WHERE Actor_Location IN ('Los Angeles', 'New York');

--3. List Director_FullName, Director_Age_in_Years, Director_Gender from Director
SELECT Director_First_Name + ' ' + Director_Last_Name AS Director_FullName, Director_Age_in_Years, Director_Gender FROM mov.Director;

--4. List Director_FullName, Director_Age_in_Years, Director_Gender from Director whose Age is less than 45 years.
SELECT Director_First_Name + ' ' + Director_Last_Name AS Director_FullName, Director_Age_in_Years, Director_Gender FROM mov.Director WHERE Director_Age_in_Years < 45;

-- 8. Write the following Query based on the above datasets.
SELECT m.Movie_Name, a.Actor_First_Name, a.Actor_Last_Name
FROM mov.Movies m
      JOIN Movie_Name a ON m.Movie_ID = a.Movie_ID;
  
```

Result Messages

	Director_FullName	Director_Age_in_Years	Director_Gender
1	Miguel Arteta	41	Male
2	Tom Vaughan	37	Male
3	Adrienne Shelly	40	Female
4	Simon Cutts	37	Male
5	James McAvoy	44	Male

Query executed successfully.

8. Write the following Query based on the above datasets.

a. Display all the Movies and their Actors information based on the relationship.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```

--1. List Director_FullName, Director_Age_in_Years, Director_Gender from Director
SELECT Director_First_Name + ' ' + Director_Last_Name AS Director_FullName, Director_Age_in_Years, Director_Gender FROM mov.Movie_Director;

--2. List Director_FullName, Director_Age_in_Years, Director_Gender from Director whose Age is less than 45 years.
SELECT Director_First_Name + ' ' + Director_Last_Name AS Director_FullName, Director_Age_in_Years, Director_Gender FROM mov.Movie_Director WHERE Director_Age_in_Years < 45;

-- 8. Write the following Query based on the above datasets.

-- a. Display all the Movies and their Actors information based on the relationship.
SELECT m.Movie_Name, a.Actor_First_Name, a.Actor_Last_Name
FROM mov.Movies m
      JOIN Movie_Actor a ON m.Movie_ID = a.Movie_ID;

-- b. Display the Movies name and their Ratings.
SELECT m.Movie_Name, r.Rating_Audience_Score, r.Rating_Rotten_Tomatoes
FROM mov.Movies m
      JOIN Rating r ON m.Movie_ID = r.Movie_ID;
  
```

Result Messages

	Movie_Name	Actor_First_Name	Actor_Last_Name
1	Zack and Miri Make a Porno	Seth	Rogen
2	Youth in Revolt	Michael	Cera
3	When in Rome	Josh	Duhamel
4	What Happens in Vegas	Jason	Sudeikis
5	Water For Elephants	Robert	Pattinson
6	Waitress	Nathan	Fillion
7	Twilight	Robert	Pattinson
8	Percy Jackson	James	McAvoy
9	Over Her Dead Body	Paul	Rudd
10	My Week with Marilyn	Kenneth	Brenagh
11	Music and Lyrics	Hugh	Graig
12	Leap Year	Matthew	Goode
13	Knocked Up	Judd	Apatow
14	Jane Eyre	Michael	Fassbender
15	Good Luck Chuck	Dane	Cook
16	Going the Distance	Jason	Sudeikis
17	Gnomeo and Juliet	Kelly	Ashley
18	Ghosts of Girlfriends Past	Mathew	McConaughey

Query executed successfully.

b. Display the Movies name and their Ratings.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```

-- 8. Write the following Query based on the above datasets.

SELECT Director_First_Name + ' ' + Director_Last_Name AS Director_FullName, Director_Age_in_Years, Director_Gender FROM mov.Movie_Director
-- a. Display all the Movies and their Actors information based on the relationship.
SELECT m.Movie_Name, a.Actor_First_Name, a.Actor_Last_Name
FROM mov.Movies m
JOIN mov.Movie_Actor a ON m.Movie_ID = a.Movie_ID;

-- b. Display the Movies name and their Ratings.
SELECT m.Movie_Name, r.Rating_Audience_Score, r.Rating_Rotten_Tomatoes
FROM mov.Movies m
JOIN mov.Movie_Rating r ON m.Movie_ID = r.Movie_ID;

-- c. Display all the Movies, Actors, and Directors information based on the relationship.

```

Results

Movie_Name	Rating_Audience_Score	Rating_Rotten_Tomatoes
Zack and Miri Make a Porno	70	64
Youth in Revolt	52	60
In Time	44	15
What Happens in Vegas	72	28
Water For Elephants	72	60
Waitress	67	89
Twilight	82	49
Penelope	74	52
Over Her Dead Body	47	15
My Week with Marilyn	84	83
Music and Lyrics	70	63
Leap Year	49	21
Knocked Up	83	91
Jane Eyre	77	85
Good Luck Chuck	61	3
Going the Distance	56	53
Gnomeo and Juliet	52	56
Ghosts of Girlfriends Past	47	27

Query executed successfully.

c. Display all the Movies, Actors, and Directors information based on the relationship.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```

--c. Display all the Movies, Actors, and Directors information based on the relationship.
SELECT m.Movie_Name, a.Actor_First_Name, a.Actor_Last_Name, d.Director_First_Name, d.Director_Last_Name
FROM mov.Movies m
JOIN mov.Movie_Actor a ON m.Movie_ID = a.Movie_ID
JOIN mov.Movie_Director d ON m.Director_ID = d.Director_ID;

```

Results

Movie_Name	Actor_First_Name	Actor_Last_Name	Director_First_Name	Director_Last_Name
Zack and Miri Make a Porno	Seth	Rogen	Kevin	Smith
Youth in Revolt	Michael	Cera	Miguel	Arteaga
In Time	Josh	Duhamel	Mark	Johnson
What Happens in Vegas	Jason	Sudeikis	Tom	Vaughn
Water For Elephants	Robert	Pattinson	Francis	Lawrence
Waitress	Nathan	Fillion	Adrienne	Shelly
Twilight	Robert	Pattinson	David	Slade
Penelope	James	McAvoy	Mark	Palansky
Over Her Dead Body	Paul	Rudd	Jeff	Lowell
My Week with Marilyn	Kenneth	Branagh	Simon	Curtis
Music and Lyrics	Hugh	Grant	Marc	Lawrence
Leap Year	Matthew	Goode	Anand	Tucker
Knocked Up	Judd	Apatow	Judd	Apatow
Good Luck Chuck	Michael	Furillo	Colin	Feldman
Going the Distance	Dane	Cook	Mark	Hilfiker
Gnomeo and Juliet	Kelly	Sudeikis	Nanette	Burstein
Ghosts of Girlfriends Past	Matthew	McConaughey	Mark	Walley
Four Christmases	Vince	Vaughn	Seth	Gordon
Fireproof	Kirk	Cameron	Alex	Kendrick
Enchanted	James	Marsden	Kevin	Lima
Dear John	Channing	Tatum	Lasse	Hallstrom
Beginners	Mike	Mills	Ewan	McGregor
3 Idiots	Aamir	Khan	Rajkumar	Hirani
Legends	Aamir	Khan	Ashutosh	Gowariker
Moammar Is Khan	Shah Rukh	Khan	Karan	Johar
Bashirbuli	Pubhas	NULL	S.S	Rajemouli
Divwaa Duhariya Le Jayen	Allu	Arujan	Aditya	Chopra
Oh My佛	Shah Rukh	Khan	Umesh	Shakya

Query executed successfully.

d. Display all the Movies, Actors, Directors, and Movie Rating information based on the relationship.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```
--d. Display all the Movies, Actors, Directors, and Movie Rating information based on the relationship.
--SELECT m.Movie_Name, a.Actor_First_Name, a.Actor_Last_Name, d.Director_First_Name, d.Director_Last_Name, r.Rating_Audience_Score, r.Rating_Rotten_Tomatoes
FROM mov_Movies m
JOIN mov_Movie_Actor a ON m.Movie_ID = a.Movie_ID
JOIN mov_Movie_Director d ON m.Director_ID = d.Director_ID
JOIN mov_Movie_Rating r ON m.Movie_ID = r.Movie_ID
```

Results

Movie_Name	Actor_First_Name	Actor_Last_Name	Director_First_Name	Director_Last_Name	Rating_Audience_Score	Rating_Rotten_Tomatoes
Zack and Miri Make a Porno	Seth	Rogen	Kevin	Smith	70	64
Youth in Revolt	Michael	Cera	Miguel	Arata	52	68
When in Rome	Josh	Duhame	Mark	Johnson	44	15
What Happens in Vegas	Jason	Sudeikis	Tori	Vaughn	72	28
Water For Elephants	Robert	Pattinson	Francis	Lawrence	72	60
Witless	Nathan	Fillion	Adrienne	Shelly	67	89
Twink	Robert	Pattinson	David	Slade	62	69
Penelope	Jamie	McAvoy	Mark	Parasky	74	52
One for the Dead	Paul	Rudd	Jeff	Lovell	47	15
My Week with Marilyn	Kenneth	Branagh	Simon	Curtis	84	83
Music and Lyrics	Hugh	Grant	Marc	Lawrence	70	63
Leap Year	Matthew	Goode	Anand	Tucker	49	21
Knocked Up	Judd	Apatow	Judd	Apatow	83	91
Jane Eyre	Michael	Fassbender	Cary	Fukunaga	77	85
Good Luck Chuck	Dave	Cook	Mark	Helfrich	61	3
Going the Distance	Jason	Sudeikis	Nanette	Burstein	56	53
Gromos and Juliet	Kathy	Asbury	James	McAvoy	52	56
Ghosts of Girlfriends Past	Mathew	McConaughey	Mark	Waters	47	27
Four Christmases	Vince	Vaughn	Seth	Gordon	52	26
20th Century Women	Kirk	Cameron	Alec	Kendrick	51	40
Ernest	James	Moroni	Kevin	Lima	80	93
Dear John	Channing	Tatum	Lasse	Hallström	66	29
Beginners	Mike	Miles	Ewan	McGinley	80	84
3 Idiots	Aamir	Khan	Rajkumar	Hirani	95	100
Lagaan	Aamir	Khan	Ashutosh	Gowariker	81	95
My Name Is Khan	Shah Rukh	Khan	Karan	Johar	79	83
Baahubali	Prabhas	NULL	S.S	Rajamouli	80	90
Dilwale Duluha Le Jayenge	Allu	Arun	Aditya	Chopra	76	82
Oh My God!	Shah Rukh	Khan	Umesh	Shukla	86	100

Query executed successfully.

MOGANAVINITH\SQLEXPRESS (1...) Moganavinith\mogan (53) master | 00:00:00 | 30 rows

LN 264 Col 1 Ch 1 INS

Aggregate Status

Connection

Connection Details

Name

Current connection parameters

Aggregate Status

Connection

Connection Details

Name

Current connection parameters

e. Display all the Movies, Actors, Directors, and Movie Rating information whose Rating_Audience_Score is more than 80% based on the relationship.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```
--e. Display all the Movies, Actors, Directors, and Movie Rating information whose Rating_Audience_Score is more than 80% based on the relationship.
--SELECT m.Movie_Name, a.Actor_First_Name, a.Actor_Last_Name, d.Director_First_Name, d.Director_Last_Name, r.Rating_Audience_Score, r.Rating_Rotten_Tomatoes
FROM mov_Movies m
JOIN mov_Movie_Actor a ON m.Movie_ID = a.Movie_ID
JOIN mov_Movie_Director d ON m.Director_ID = d.Director_ID
JOIN mov_Movie_Rating r ON m.Movie_ID = r.Movie_ID
```

Results

Movie_Name	Actor_First_Name	Actor_Last_Name	Director_First_Name	Director_Last_Name	Rating_Audience_Score	Rating_Rotten_Tomatoes
Twilight	Robert	Pattinson	David	Slade	82	49
My Week with Marilyn	Kenneth	Branagh	Simon	Curtis	84	83
Knocked Up	Judd	Apatow	Judd	Apatow	83	91
3 Idiots	Aamir	Khan	Rajkumar	Hirani	95	100
Lagaan	Aamir	Khan	Ashutosh	Gowariker	81	95
Oh My God!	Shah Rukh	Khan	Umesh	Shukla	85	100
Pushpa	Aishay	Kumar	Sukumar	NULL	81	74

Query executed successfully.

MOGANAVINITH\SQLEXPRESS (1...) Moganavinith\mogan (53) master | 00:00:00 | 7 rows

LN 279 Col 1 Ch 1 INS

Aggregate Status

Connection

Connection Details

Name

Current connection parameters

f. Display all the Movies information whose Rating_Rotten_Tomatoes is more than 90%.

```

--f. Display all the Movies information whose Rating_Rotten_Tomatoes is more than 80.
SELECT m.Movie_Name, m.Movie_Released_Year, m.Movie_Lead_Studio, m.Movie_Language, m.Movie_Category, m.Movie_Duration_in_Min, m.Movie_Worldwide_Earning_in_SM, m.Movie_Type
FROM mov_Movies m
JOIN mov_Movie_Rating r ON m.Movie_ID = r.Movie_ID
WHERE r.Rating_Audience_Score > 80;

```

Query executed successfully.

9. Write the following Query based on the above datasets.

a. Create new table MovieCopy and copy all records of Movie table.

```

SELECT *
INTO mov.MovieCopy
FROM mov.Movies;

```

b. Create a new table DirectorCopy and copy only the schema of director table.

```

SELECT *
INTO mov.DirectorCopy
FROM mov.Movie_Director
WHERE 1=0; -- No data is copied, only the schema

```

c. Create new table ActorCopy and copy all records of Actor table.

```

SELECT *
INTO mov.ActorCopy
FROM mov.Movie_Actor;

```

d. Create new table RatingCopy and copy all records of Rating table.

```

SELECT *
INTO mov.RatingCopy
FROM mov.Movie_Rating;

```

e. Create new table RatingCopies and copy only the schema from Rating table.

```

SELECT *
INTO mov.RatingCopies
FROM mov.Movie_Rating
WHERE 1=0; -- No data is copied, only the schema

```

10. Write the following Query based on the above datasets.

a. Delete all the records from the RatingCopy table.
`DELETE FROM mov.RatingCopy;`

b. Delete all the movies from MovieCopy whose released in year “2010”.
`DELETE FROM mov.MovieCopy
WHERE Movie_Released_Year = 2010;`

c. Delete all the movies from MovieCopy where language is 'Hindi'.
`DELETE FROM mov.MovieCopy
WHERE Movie_Language = 'Hindi';`

d. Delete all the movies from MovieCopy where Rating_Audience_Score is less than 80%.
`DELETE m
FROM mov.MovieCopy m
JOIN mov.RatingCopy r ON m.Movie_ID = r.Movie_ID
WHERE r.Rating_Audience_Score < 80;`

e. Delete all the movies from MovieCopy where Rating_Rotten_Tomatoes is less than 70%.
`DELETE m
FROM mov.MovieCopy m
JOIN mov.RatingCopy r ON m.Movie_ID = r.Movie_ID
WHERE r.Rating_Rotten_Tomatoes < 70;`

12. Write the following Query based on the above datasets.

a. Create a view to display all the movie information.

`CREATE VIEW mov.View_AllMovies AS
SELECT *
FROM mov.Movies;`

b. Create a view to display all the movies and their rating information.

`CREATE VIEW mov.View_MoviesWithRatings AS
SELECT m.Movie_Name, m.Movie_Released_Year, m.Movie_Lead_Studio, r.Rating_Audience_Score,
r.Rating_Rotten_Tomatoes
FROM mov.Movies m
JOIN mov.Movie_Rating r ON m.Movie_ID = r.Movie_ID;`

c. Create a view to display all the movies and their actor information.

`CREATE VIEW mov.View_MoviesWithActors AS
SELECT m.Movie_Name, a.Actor_First_Name, a.Actor_Last_Name, a.Actor_Age_in_Years,
a.Actor_Location
FROM mov.Movies m
JOIN mov.Movie_Actor a ON m.Movie_ID = a.Movie_ID;`

d. Create a view to display all the movies, rating, actor along with director information.

`CREATE VIEW mov.View_MoviesRatingsActorsDirectors AS
SELECT m.Movie_Name, r.Rating_Audience_Score, r.Rating_Rotten_Tomatoes,
a.Actor_First_Name, a.Actor_Last_Name, d.Director_First_Name, d.Director_Last_Name
FROM mov.Movies m
JOIN mov.Movie_Rating r ON m.Movie_ID = r.Movie_ID
JOIN mov.Movie_Actor a ON m.Movie_ID = a.Movie_ID
JOIN mov.Movie_Director d ON m.Director_ID = d.Director_ID;`

e. Create a view to display all the information based on the result set returned by the specified query.

`CREATE VIEW mov.View_MoviesDirectors AS`

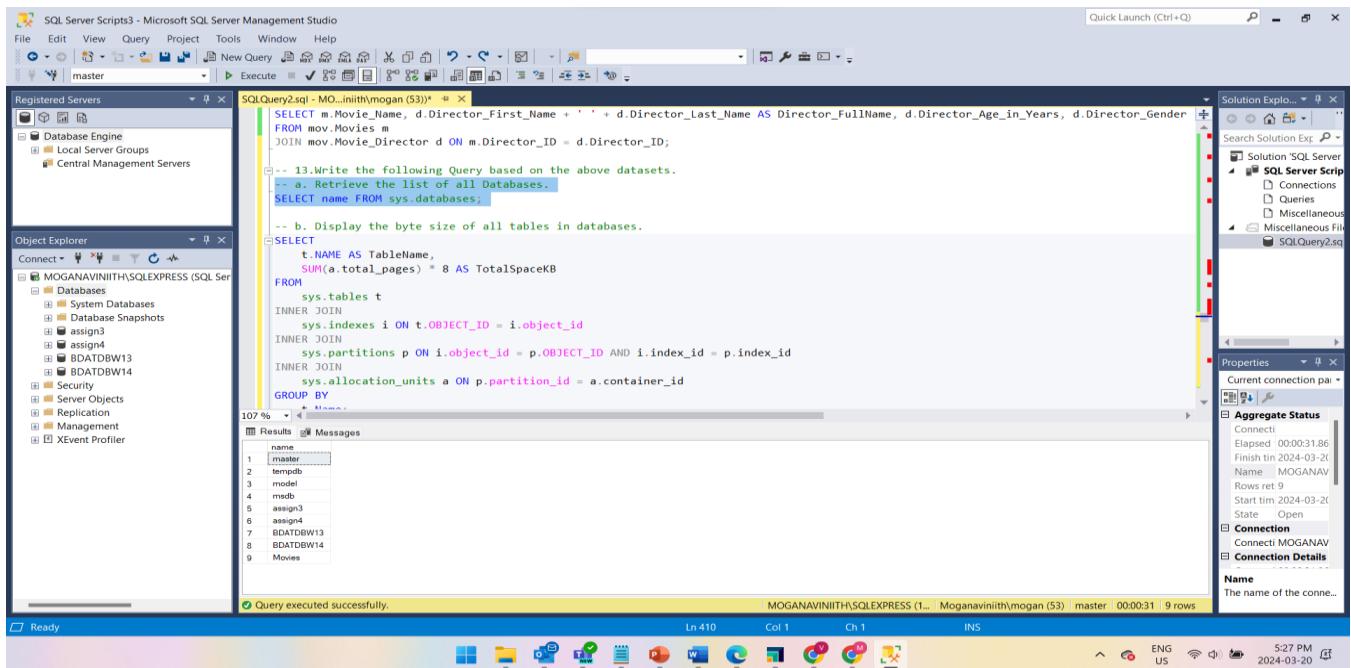
```

SELECT m.Movie_Name, d.Director_First_Name + ' ' + d.Director_Last_Name AS Director_FullName, d.Director_Age_in_Years, d.Director_Gender
FROM mov.Movies m
JOIN mov.Movie_Director d ON m.Director_ID = d.Director_ID;

```

13. Write the following Query based on the above datasets.

a. Retrieve the list of all Databases.



The screenshot shows the Microsoft SQL Server Management Studio interface. The query window contains the following code:

```

-- a. Retrieve the list of all Databases.
SELECT name FROM sys.databases;

```

The results pane displays a table with one column 'name' containing the names of the databases:

name
1 master
2 tempdb
3 model
4 msdb
5 assign3
6 assign4
7 BDATDBW13
8 BDATDBW14
9 Movies

The status bar at the bottom indicates "Query executed successfully." and "MOGANAVINITH\SQLEXPRESS (1... Moganavinith\mogan (53) | master | 00:00:31 | 9 rows".

b. Display the byte size of all tables in databases.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```

SELECT name FROM sys.databases;

-- b. Display the byte size of all tables in databases.
SELECT
    t.NAME AS TableName,
    SUM(a.total_pages) * 8 AS TotalSpaceKB
FROM
    sys.tables t
INNER JOIN
    sys.indexes i ON t.OBJECT_ID = i.object_id
INNER JOIN
    sys.partitions p ON i.object_id = p.OBJECT_ID AND i.index_id = p.index_id
INNER JOIN
    sys.allocation_units a ON p.partition_id = a.container_id
GROUP BY
    t.Name;

--c. List of tables with number of records.
SELECT
    t.NAME AS TableName,
    SUM(p.rows) AS RowCounts
FROM
    sys.tables t
INNER JOIN
    sys.partitions p ON t.object_id = p.object_id
GROUP BY
    t.Name;

```

Query executed successfully.

TableName	TotalSpaceKB
ActorCopy	16
Appointments	16
AugAgeByPetType	16
AugVisitLength	16
Customers	32
DirectorCopy	16
Movie_Actor	16
Movie_Director	16
Movie_Rating	16
MovieCopy	16
Movies	16

TableName	RowCounts
ActorCopy	30
Appointments	23
AugAgeByPetType	3
AugVisitLength	6
Customers	10
DirectorCopy	0
Movie_Actor	30
Movie_Director	30
Movie_Rating	30
MovieCopy	20
Movies	30
MSReplication_options	3
PassedAwayInfo	14
PetOwnerInfo	10
Pets	22
PetsWithoutAppoint...	2

c. List of tables with number of records.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```

SELECT
    t.NAME AS TableName,
    SUM(p.rows) AS RowCounts
FROM
    sys.tables t
INNER JOIN
    sys.partitions p ON t.object_id = p.object_id
WHERE
    p.index_id IN (0, 1)
GROUP BY
    t.Name;

```

Query executed successfully.

TableName	RowCounts
ActorCopy	30
Appointments	23
AugAgeByPetType	3
AugVisitLength	6
Customers	10
DirectorCopy	0
Movie_Actor	30
Movie_Director	30
Movie_Rating	30
MovieCopy	20
Movies	30
MSReplication_options	3
PassedAwayInfo	14
PetOwnerInfo	10
Pets	22
PetsWithoutAppoint...	2

d. List of Primary Key and Foreign Key for Whole Database.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```

-- For Primary Keys:
SELECT
    t.name AS Table_Name,
    i.name AS PK_Name
FROM
    sys.tables t
INNER JOIN
    sys.indexes i ON t.object_id = i.object_id
WHERE
    i.is_primary_key = 1;

-- For Foreign Keys:
SELECT
    f.name AS FK_Name,
    t.name AS Table_Name,
    c.name AS Column_Name
FROM
    sys.foreign_keys AS f
INNER JOIN
    sys.foreign_key_columns AS fc ON f.object_id = fc.constraint_object_id
INNER JOIN
    sys.tables t ON f.parent_object_id = t.object_id
INNER JOIN
    sys.columns c ON fc.parent_object_id = c.object_id AND fc.parent_column_id = c.column_id;

```

Query executed successfully.

MOGANAVINITH\SQLEXPRESS (1...) Moganavinith\mogan (53) master 00:00:00 9 rows

Ready Ln 450 Col 26 Ch 26 INS

ENG US 5:42 PM 2024-03-20

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```

-- For Primary Keys:
SELECT
    t.name AS Table_Name,
    i.name AS PK_Name
FROM
    sys.tables t
INNER JOIN
    sys.indexes i ON t.object_id = i.object_id
WHERE
    i.is_primary_key = 1;

-- For Foreign Keys:
SELECT
    f.name AS FK_Name,
    t.name AS Table_Name,
    c.name AS Column_Name
FROM
    sys.foreign_keys AS f
INNER JOIN
    sys.foreign_key_columns AS fc ON f.object_id = fc.constraint_object_id
INNER JOIN
    sys.tables t ON f.parent_object_id = t.object_id
INNER JOIN
    sys.columns c ON fc.parent_object_id = c.object_id AND fc.parent_column_id = c.column_id;

```

Query executed successfully.

MOGANAVINITH\SQLEXPRESS (1...) Moganavinith\mogan (53) master 00:00:00 7 rows

Ready Ln 452 Col 1 Ch 1 INS

ENG US 5:43 PM 2024-03-20

e. Get all Nullable columns of a table.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```

--e. Get all Nullable columns of a table.

SELECT
    COLUMN_NAME
FROM
    INFORMATION_SCHEMA.COLUMNS
WHERE
    TABLE_NAME = 'YourTableName' AND IS_NULLABLE = 'YES';

--f. Get All tables that do not have a primary key.

SELECT
    t.name AS TableName
FROM
    sys.tables AS t
WHERE
    NOT EXISTS(SELECT * FROM sys.indexes AS i WHERE t.object_id = i.object_id AND i.is_primary_key = 1);

```

Query executed successfully.

Results of Messages

COLUMN_NAME

MOGANAVINITH\SQLEXPRESS (1... Moganavinith\mogan (53) master 00:01:34 0 rows

LN 466 Col 1 Ch 1 INS

Ready

ENGLISH US 5:39 PM 2024-03-20

f. Get All table that do not have primary key.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```

--f. Get All tables that do not have a primary key.

SELECT
    t.name AS TableName
FROM
    sys.tables AS t
WHERE
    NOT EXISTS(SELECT * FROM sys.indexes AS i WHERE t.object_id = i.object_id AND i.is_primary_key = 1);

--g. Get All tables that do not have an identity column.

SELECT
    t.name AS TableName
FROM
    sys.tables AS t

```

Query executed successfully.

Results of Messages

TableName
1 spt_fallback_db
2 spt_fallback_dev
3 spt_fallback_uog
4 ArgvListLength
5 ArgvListOffset
6 TaskInCntrByCity
7 PutOwnerInfo
8 ParseAssemblyInfo
9 ParseWhosAppointments
10 spt_monitor
11 MovieCopy
12 DirectorCopy
13 ActorCopy
14 RangCopy
15 RangCopies
16 M\$replication_options

MOGANAVINITH\SQLEXPRESS (1... Moganavinith\mogan (53) master 00:00:00 16 rows

LN 475 Col 1 Ch 1 INS

Ready

ENGLISH US 5:41 PM 2024-03-20

g. Get All table that do not have identity column.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```

SELECT
    t.name AS TableName
FROM
    sys.tables AS t
WHERE
    NOT EXISTS(SELECT * FROM sys.indexes AS i WHERE t.object_id = i.object_id AND i.is_primary_key = 1);
--g. Get All tables that do not have an identity column.

SELECT
    t.name AS TableName
FROM
    sys.tables AS t
WHERE
    NOT EXISTS(SELECT * FROM sys.columns AS c WHERE t.object_id = c.object_id AND c.is_identity = 1);
--h. Get First Date of Current Month.

```

Result Messages

TableName
spt_fallback_db
spt_fallback_dev
spt_fallback_usr
Customers
Pets
Appointments
AgeVisitLength
AgeAgeByPetType
TotalVisitsByCity
PetOwnerInfo
PassedAwayInfo
PetsWithoutAppointments
Customers
Pets
spt_monitor
MSpecification_Options

Query executed successfully.

h. Get First Date of Current Month.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```

SELECT
    t.name AS TableName
FROM
    sys.tables AS t
WHERE
    NOT EXISTS(SELECT * FROM sys.columns AS c WHERE t.object_id = c.object_id AND c.is_identity = 1);
--h. Get First Date of Current Month.
SELECT
    DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()), 0) AS FirstDay_CurrentMonth;
--i. Get Last date of Current month.
SELECT
    DATEADD(DAY, -1, DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()) + 1, 0)) AS LastDay_CurrentMonth;
--j. Get first date of next month.
SELECT
    DATEADD(MONTH, 1, 0) AS FirstDay_NextMonth;

```

Result Messages

FirstDay_CurrentMonth
2024-03-01 00:00:00.000

Query executed successfully.

i. Get Last date of Current month.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```
--h. Get First Date of Current Month.
SELECT
    DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()), 0) AS FirstDay_CurrentMonth;

--i. Get Last date of Current month.
SELECT
    DATEADD(DAY, -1, DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()) + 1, 0)) AS LastDay_CurrentMonth;

--j. Get first date of next month.
SELECT
    DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()) + 1, 0) AS FirstDay_NextMonth;

--k. Get Last date of next month.
SELECT
    DATEADD(DAY, -1, DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()) + 2, 0)) AS LastDay_NextMonth;
```

Results

LastDay_CurrentMonth
2024-03-31 00:00:00.000

Query executed successfully.

MOGANAVINITH\SQLEXPRESS (1... Moganavinith\mogan (53) master 00:00:00 1 rows

Ready Ln 496 Col 1 Ch 1 INS

5:46 PM 2024-03-20

j. Get first date of next month.

SQL Server Scripts3 - Microsoft SQL Server Management Studio

```
--h. Get First Date of Current Month.
SELECT
    DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()), 0) AS FirstDay_CurrentMonth;

--i. Get Last date of Current month.
SELECT
    DATEADD(DAY, -1, DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()) + 1, 0)) AS LastDay_CurrentMonth;

--j. Get first date of next month.
SELECT
    DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()) + 1, 0) AS FirstDay_NextMonth;

--k. Get Last date of next month.
SELECT
    DATEADD(DAY, -1, DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()) + 2, 0)) AS LastDay_NextMonth;
```

Results

FirstDay_NextMonth
2024-04-01 00:00:00.000

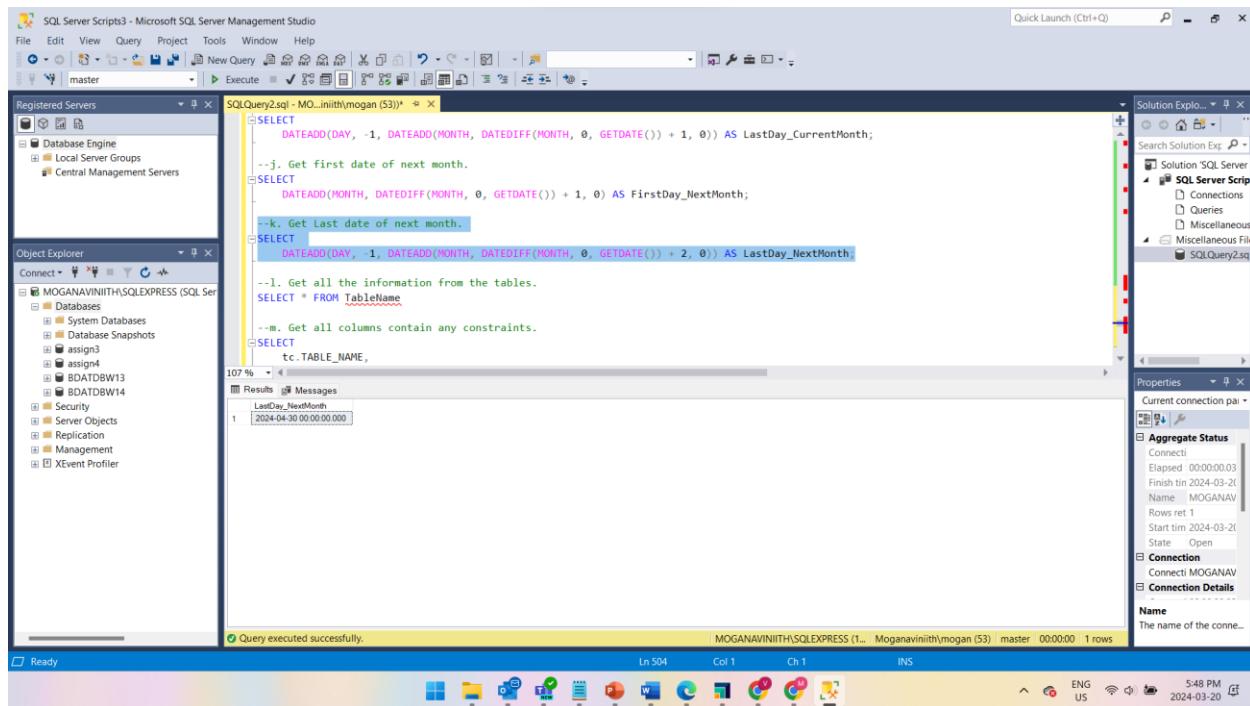
Query executed successfully.

MOGANAVINITH\SQLEXPRESS (1... Moganavinith\mogan (53) master 00:00:00 1 rows

Ready Ln 500 Col 1 Ch 1 INS

5:47 PM 2024-03-20

k. Get Last date of next month.



```

SELECT
    DATEADD(DAY, -1, DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()) + 1, 0)) AS LastDay_CurrentMonth;

--j. Get first date of next month.
SELECT
    DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()) + 1, 0) AS FirstDay_NextMonth;

--k. Get Last date of next month.
SELECT
    DATEADD(DAY, -1, DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()) + 2, 0)) AS LastDay_NextMonth;

--l. Get all the information from the tables.
SELECT * FROM TableName

--m. Get all columns contain any constraints.
SELECT
    tc.TABLE_NAME,
    kcu.COLUMN_NAME,
    tc.CONSTRAINT_TYPE
FROM
    INFORMATION_SCHEMA.TABLE_CONSTRAINTS AS tc
JOIN
    INFORMATION_SCHEMA.KEY_COLUMN_USAGE AS kcu ON tc.CONSTRAINT_NAME = kcu.CONSTRAINT_NAME
ORDER BY
    tc.TABLE_NAME,
    kcu.COLUMN_NAME;

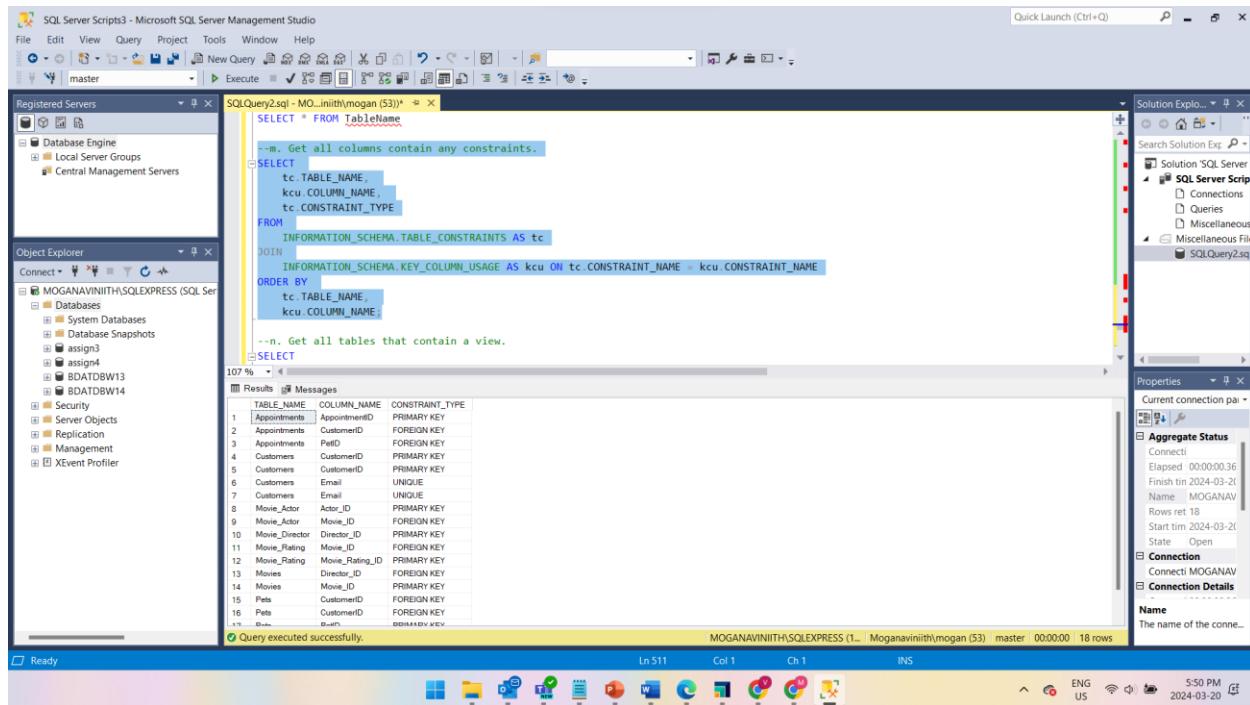
```

Query executed successfully.

MOGANAVINITH\SQLEXPRESS (1...) Moganavinith\mogan (53) master 00:00:00 1 rows

Ready Ln 504 Col 1 Ch 1 INS 5:48 PM 2024-03-20

m. Get all columns contain any constraints.



```

SELECT * FROM TableName

--m. Get all columns contain any constraints.
SELECT
    tc.TABLE_NAME,
    kcu.COLUMN_NAME,
    tc.CONSTRAINT_TYPE
FROM
    INFORMATION_SCHEMA.TABLE_CONSTRAINTS AS tc
JOIN
    INFORMATION_SCHEMA.KEY_COLUMN_USAGE AS kcu ON tc.CONSTRAINT_NAME = kcu.CONSTRAINT_NAME
ORDER BY
    tc.TABLE_NAME,
    kcu.COLUMN_NAME;

--n. Get all tables that contain a view.
SELECT
    TABLE_NAME, COLUMN_NAME, CONSTRAINT_TYPE

```

TABLE_NAME	COLUMN_NAME	CONSTRAINT_TYPE
Appointments	AppointmentID	PRIMARY KEY
Appointments	CustomerID	FOREIGN KEY
Appointments	PetID	FOREIGN KEY
Customers	CustomerID	PRIMARY KEY
Customers	CustomerID	UNIQUE
Customers	Email	UNIQUE
Movie_Actor	Actor_ID	PRIMARY KEY
Movie_Actor	Movie_ID	FOREIGN KEY
Movie_Director	Director_ID	PRIMARY KEY
Movie_Rating	Movie_ID	FOREIGN KEY
Movie_Rating	Movie_Rating_ID	PRIMARY KEY
Movies	Director_ID	FOREIGN KEY
Movies	Movie_ID	PRIMARY KEY
Pets	CustomerID	FOREIGN KEY
Pets	CustomerID	FOREIGN KEY

Query executed successfully.

MOGANAVINITH\SQLEXPRESS (1...) Moganavinith\mogan (53) master 00:00:00 18 rows

Ready Ln 511 Col 1 Ch 1 INS 5:50 PM 2024-03-20

n. Get all tables that contain a view.

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with a connection to 'MOGANAVINITH\SQLSERVER'. The center pane contains a query window with the following T-SQL script:

```
SELECT
    distinct t.name as TableName,
    v.name as ViewName
FROM
    sys.views v
JOIN
    sys.sql_expression_dependencies sed ON sed.referencing_id = v.object_id
JOIN
    sys.tables t ON t.object_id = sed.referenced_id;
```

The results grid below shows the output of the query:

	TableName	ViewName
1	Customers	Customer_FullNames
2	Movie_Actor	View_MovieRatingsActorsDirectors
3	Movie_Actor	View_MoviesWithActors
4	Movie_Director	View_MovieDirectors
5	Movie_Director	View_MovieRatingsActorsDirectors
6	Movie_Rating	View_MovieRatingsActorsDirectors
7	Movie_Rating	View_MovieWithRatings
8	Movies	View_MovieAllCores
9	Movies	View_MovieDirectors
10	Movies	View_MovieRatingsActorsDirectors
11	Movies	View_MoviesWithActors
12	Movies	View_MoviesWithRatings

The status bar at the bottom indicates: 'Query executed successfully.' and 'LN 524 Col 1 Ch 1 INS'.

- o. Get all columns of table that using in views.

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar reads "SQL Server Scripts3 - Microsoft SQL Server Management Studio". The menu bar includes File, Edit, View, Query, Project, Tools, Window, Help. The toolbar has various icons for file operations like New Query, Save, Execute, and Refresh. The left sidebar has "Registered Servers" and "Object Explorer" sections. Object Explorer shows a connection to "MOGANAVINITH\SQLEXPRESS (SQL Server)". The main area displays a T-SQL query in the "Query Editor" window:

```
INNER JOIN sys.tables t ON t.object_id = c.object_id
WHERE sed.referenced_minor_id = c.column_id;

-- o. Get all columns of a table that are used in views.
SELECT v.name as Viewname,
       c.name as ColumnName,
       t.name as TableName
FROM sys.views v
INNER JOIN sys.sql_expression_dependencies sed ON sed.referencing_id = v.object_id
INNER JOIN sys.columns c ON c.object_id = sed.referenced_id
INNER JOIN sys.tables t ON t.object_id = c.object_id
WHERE sed.referenced_minor_id = c.column_id;
```

The status bar at the bottom indicates "Query executed successfully." and "MOGANAVINITH\SQLEXPRESS (1...) Moganavinith@mogan (53) master 00:00:33 0 rows". The bottom right corner shows system icons for battery, signal, and time.

Name of students who has participated in the assignment.

Student name: Moganaviniith Rathinavel (200531668)

Student name: Nanditha Babu (200573747)
--

Student name: Aswin Narayanan (200568974)
--

Student name: Srivas Sathyaranayanan (200544678)

Student name: Raveen Kumar (200574798)
