Schema Design

```
• Movies: movie_id, title, director_id, genre_id
   • Directors: director_id, name
   • Actors: actor_id, name
   • Genres: genre_id, genre_name
   • Cast: movie_id, actor_id, role
   • Ratings: movie_id, rating
********
                        -- Movies table
CREATE TABLE Movies (
  movie_id INT PRIMARY KEY,
  title VARCHAR(255) NOT NULL,
  director_id INT,
  genre id INT,
  FOREIGN KEY (director id) REFERENCES Directors(director id),
  FOREIGN KEY (genre_id) REFERENCES Genres(genre_id)
);
-- Directors table
CREATE TABLE Directors (
  director_id INT PRIMARY KEY,
  name VARCHAR(255) NOT NULL
);
-- Actors table
CREATE TABLE Actors (
  actor id INT PRIMARY KEY,
  name VARCHAR(255) NOT NULL
);
-- Genres table
CREATE TABLE Genres (
  genre_id INT PRIMARY KEY,
  genre_name VARCHAR(100) NOT NULL
);
```

```
-- Cast table
CREATE TABLE Cast (
  movie id INT,
  actor id INT,
  role VARCHAR(255),
  PRIMARY KEY (movie_id, actor_id),
  FOREIGN KEY (movie id) REFERENCES Movies(movie id),
  FOREIGN KEY (actor_id) REFERENCES Actors(actor_id)
);
-- Ratings table
CREATE TABLE Ratings (
  movie id INT,
  rating DECIMAL(3, 1) CHECK (rating >= 0 AND rating <= 10),
  PRIMARY KEY (movie id),
  FOREIGN KEY (movie_id) REFERENCES Movies(movie_id)
);
*******
                      Insertion Part
-- Insert data into Directors table
INSERT INTO Directors (director_id, name) VALUES
(1, 'Christopher Nolan'),
(2, 'Steven Spielberg'),
(3, 'Quentin Tarantino');
-- Insert data into Genres table
INSERT INTO Genres (genre id, genre name) VALUES
(1, 'Action'),
(2, 'Drama'),
(3, 'Sci-Fi');
-- Insert data into Movies table
INSERT INTO Movies (movie_id, title, director_id, genre_id) VALUES
(1, 'Inception', 1, 3),
(2, 'Jurassic Park', 2, 1),
(3, 'Pulp Fiction', 3, 2);
-- Insert data into Actors table
INSERT INTO Actors (actor_id, name) VALUES
(1, 'Leonardo DiCaprio'),
(2, 'Sam Neill'),
(3, 'John Travolta');
```

```
-- Insert data into Cast table
INSERT INTO Cast (movie_id, actor_id, role) VALUES
(1, 1, 'Dom Cobb'),
(2, 2, 'Dr. Alan Grant'),
(3, 3, 'Vincent Vega');
-- Insert data into Ratings table
INSERT INTO Ratings (movie_id, rating) VALUES
(1, 8.8),
(2, 8.1),
```

(3, 8.9);

Available Tables

Actors

actor_id	name
1	Leonardo DiCaprio
2	Sam Neill
3	John Travolta

Cast

movie_id	actor_id	role
1	1	Dom Cobb
2	2	Dr. Alan Grant
3	3	Vincent Vega

Directors

director_id	name
1	Christopher Nolan
2	Steven Spielberg
3	Quentin Tarantino

Genres

genre_id	genre_name
1	Action
2	Drama
3	Sci-Fi

Movies

movie_id	title	director_id	genre_id
1	Inception	1	3
2	Jurassic Park	2	1
3	Pulp Fiction	3	2

Ratings

movie_id	rating
1	8.8
2	8.1
3	8.9

1. List all actors along with the titles of movies in which they appeared.

SELECT a.name AS actor_name, m.title AS movie_title FROM Actors a JOIN Cast c ON a.actor_id = c.actor_id JOIN Movies m ON c.movie id = m.movie id;

utput	
actor_name	movie_title
Leonardo DiCaprio	Inception
Sam Neill	Jurassic Park
John Travolta	Pulp Fiction

2.List all movies and their directors, including movies that don't have an assigned director.

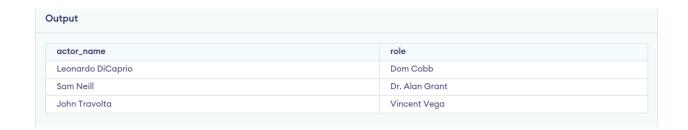
SELECT m.title AS movie_title, d.name AS director_name FROM Movies m LEFT JOIN Directors d ON m.director id = d.director id;

Dutput		
movie_title	director_name	
Inception	Christopher Nolan	
Jurassic Park	Steven Spielberg	
Pulp Fiction	Quentin Tarantino	

3.List all actors and the roles they played, including actors who haven't been cast in any movie.

SELECT a.name AS actor_name, COALESCE(c.role, 'Not Cast') AS role FROM Actors a

LEFT JOIN Cast c ON a.actor_id = c.actor_id;



4. List all movies and the genres, including genres without any movies and movies without an assigned genre.

SELECT m.title AS movie_title, g.genre_name FROM Movies m LEFT JOIN Genres g ON m.genre id = g.genre id;

tput		
movie_title	genre_name	
Inception	Sci-Fi	
Jurassic Park	Action	
Pulp Fiction	Drama	

5.List all pairs of movies directed by the same director.

```
SELECT m1.title AS movie_1, m2.title AS movie_2, d.name AS director_name FROM Movies m1

JOIN Movies m2 ON m1.director_id = m2.director_id

JOIN Directors d ON m1.director_id = d.director_id

WHERE m1.movie_id < m2.movie_id;
```

Subquery in WHERE clause(any one below)

1. Find the titles of movies that received a rating higher than the average rating of all movies.

```
SELECT title
FROM Movies
WHERE movie_id IN (
    SELECT movie_id
    FROM Ratings
    WHERE rating > (
        SELECT AVG(rating) FROM Ratings
    )
);
```

2. Find the names of all directors who have directed more than 5 movies.

```
SELECT name
FROM Directors
WHERE director_id IN (
    SELECT director_id
    FROM Movies
    GROUP BY director_id
    HAVING COUNT(movie_id) > 5
);
```