1.1 Create Movie info database
SQL:

```
CREATE DATABASE Movie_Info;
```

1.2 Create director table
SQL:

```
CREATE TABLE Director (
    Person_Id int PRIMARY KEY,
    Director_Name varchar(200) NOT NULL,
    Birth_Year int NOT NULL,
    No_Of_Films int NOT NULL,
    No_Of_Awards int NOT NULL
);
```

1.3 Data Entry
SQL:

```
INSERT INTO Director (
    Person_ID,
    Director_Name,
    Birth_Year,
    No_Of_Films,
    No_Of_Awards
) VALUES (
    1,
    'Zahir Raihan',
    1935,
    5,
    5
);

INSERT INTO Director (Person_Id, Director_Name, Birth_Year, No_Of_Films, No_Of_Awards)
VALUES
(6, 'Goutam Ghose', 1950, 12, 8),
(7, 'Aparna Sen', 1945, 12, 7),
(8, 'Kaushik Ganguly', 1968, 23, 30);
```

1.4 Create Movie table
SQL:

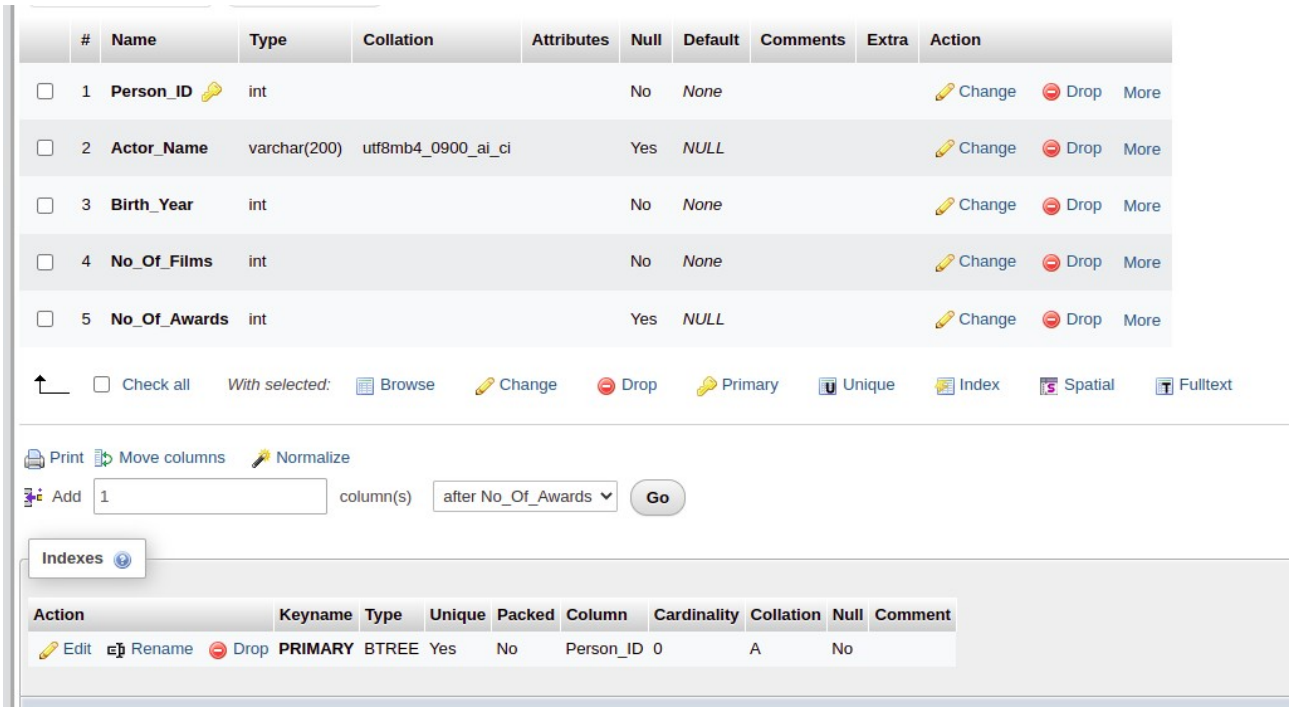
```
CREATE TABLE Movie (
    Movie_Id int PRIMARY KEY,
    Movie_Name varchar(200) NOT NULL,
    Genra varchar(100) NOT NULL,
    Year int NOT NULL,
    IMDB_Rating float,
    FOREIGN KEY (Director_Id) REFERENCES Director (Person_Id)
);
```

1.5 Insert movie data
SQL:

2.1 Create Actor Table

SQL:

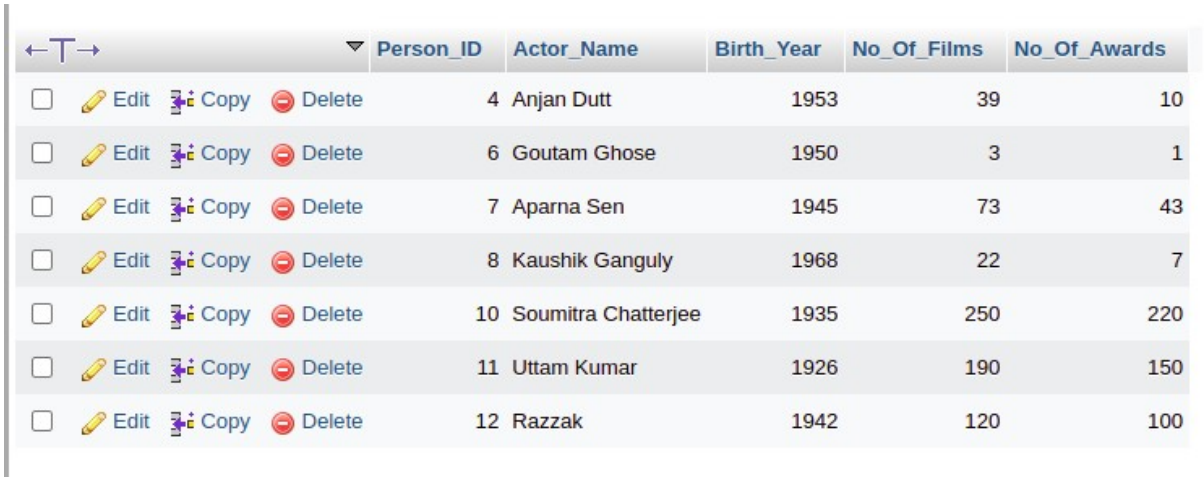
```
CREATE TABLE Actor(
    Person_ID int PRIMARY KEY,
    Actor_Name varchar(200),
    Birth_Year int NOT NULL,
    No_Of_Films int NOT NULL,
    No_Of__Awards int
);
```



2.1 Insert Data to Actor Table

SQL:

```
INSERT INTO Actor (
    Person_ID,
    Actor_Name,
    Birth_Year,
    No_Of_Films,
    No_Of_Awards
)
VALUES
    (11, 'Uttam Kumar', 1926, 190, 150),
    (12, 'Razzak', 1942, 120, 100),
    (7, 'Aparna Sen', 1945, 73, 43),
    (6, 'Goutam Ghose', 1950, 3, 1),
    (8, 'Kaushik Ganguly', 1968, 22, 7),
    (10, 'Soumitra Chatterjee', 1935, 250, 220),
    (4, 'Anjan Dutt', 1953, 39, 10);
```



	Person_ID	Actor_Name	Birth_Year	No_Of_Films	No_Of_Awards
Edit Copy Delete	4	Anjan Dutt	1953	39	10
Edit Copy Delete	6	Goutam Ghose	1950	3	1
Edit Copy Delete	7	Aparna Sen	1945	73	43
Edit Copy Delete	8	Kaushik Ganguly	1968	22	7
Edit Copy Delete	10	Soumitra Chatterjee	1935	250	220
Edit Copy Delete	11	Uttam Kumar	1926	190	150
Edit Copy Delete	12	Razzak	1942	120	100

Question 2.1.1 : Find the name, genre, IMDb Rating and release year of all the movies.

SQL:

SELECT Movie_Name, Genre, IMDB_Rating, Year FROM Movie;

	Movie_Name	Genre	IMDB_Rating	Year
Edit Copy Delete	Pather Panchali	Drama	8.5	2011
Edit Copy Delete	Noukadubi	Drama	7.6	2011
Edit Copy Delete	Abohomaan	Drama	7.3	2009
Edit Copy Delete	Joi Baba Felunath	Thriller	8	1979
Edit Copy Delete	Jibon Theke Neya	Drama	9.4	1970
Edit Copy Delete	Moner Manush	Bioghaphy	8	2010
Edit Copy Delete	Apur Panchali	Bioghaphy	8.2	2013
Edit Copy Delete	Goynar Baksho	Comedy	7.1	2013
Edit Copy Delete	Byomkesh O Agnibaan	Thriller	7.4	2017
Edit Copy Delete	Byomkesh Bakshi	Thriller	7.4	2010
Edit Copy Delete	PK	Fiction	8.2	2014

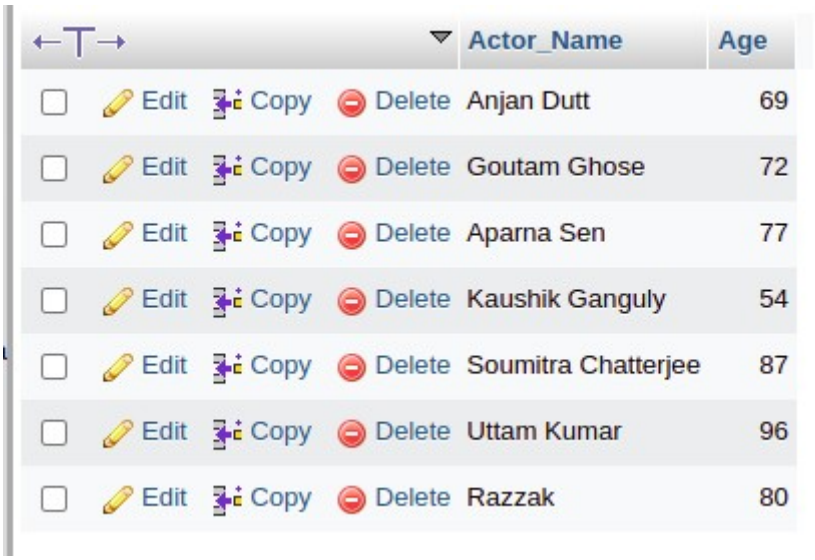
Question 2.1.2: Display all the distinct movie-genres
SQL:

SELECT DISTINCT Genre FROM Movie;

Edit Copy Delete	Drama
Edit Copy Delete	Thriller
Edit Copy Delete	Bioghaphy
Edit Copy Delete	Comedy
Edit Copy Delete	Fiction

Question 2.1.3: Find the name and the current age of all the actors
SQL:

SELECT Actor_Name, 2022 - Birth_Year as Age from Actor;



	Actor_Name	Age
Edit Copy Delete	Anjan Dutt	69
Edit Copy Delete	Goutam Ghose	72
Edit Copy Delete	Aparna Sen	77
Edit Copy Delete	Kaushik Ganguly	54
Edit Copy Delete	Soumitra Chatterjee	87
Edit Copy Delete	Uttam Kumar	96
Edit Copy Delete	Razzak	80

Question 2.1.4: Find the name, IMDb rating and the name of director of all the movies.

SQL:
SELECT Movie_Name, IMDB_Rating, Director_Name from Movie LEFT JOIN Director ON Movie.Director_Id = Director.Person_Id;

Movie_Name	IMDB_Rating	Director_Name
Pather Panchali	8.5	Saytajit Ray
Noukadubi	7.6	Rituparno Ghosh
Abohomaan	7.3	Rituparno Ghosh
Joi Baba Felunath	8	Saytajit Ray
Jibon Theke Neya	9.4	Zahir Raihan
Moner Manush	8	Goutam Ghose
Apur Panchali	8.2	Kaushik Ganguly
Goynar Baksho	7.1	Aparna Sen
Byomkesh O Agnibaan	7.4	Anjan Dutt
Byomkesh Bakshi	7.4	Anjan Dutt
PK	8.2	Rajkumar Hirani

Question 2.1.5: Find the name, IMDb rating, release year and the director name of all the movies of drama genre whose IMDb rating is greater than or equal to 8.0.

SQL:

SELECT Movie_Name, IMDB_Rating, Year, Director_Name FROM Movie, Director WHERE Movie.Director_Id = Director.Person_Id AND Movie.Genre = 'Drama' AND Movie.IMDB_Rating >= 8.0;

Movie_Name	IMDB_Rating	Year	Director_Name
Pather Panchali	8.5	2011	Saytajit Ray
Jibon Theke Neya	9.4	1970	Zahir Raihan

Question 2.1.6: Find the name and genre of all the movies that contain an ‘A’ or a ‘J’ as the 1st character of its name.

SQL:

SELECT Movie_Name, Genre FROM Movie WHERE Movie_Name LIKE 'A%' OR Movie_Name LIKE 'J%';

←T→	Movie_Name	Genre
Edit Copy Delete	Abohomaan	Drama
Edit Copy Delete	Joi Baba Felunath	Thriller
Edit Copy Delete	Jibon Theke Neya	Drama
Edit Copy Delete	Apur Panchali	Bioghaphy

Question 2.1.7: Find the name and the number of awards of those directors whose name contains an ‘a’ as its 2nd character.

SQL:
SELECT Director_Name, No_Of_Awards FROM Director WHERE Director_Name LIKE '_a%';

Question 2.1.8: Find the name, genre and release year of all the movies that contain the term ‘Byomkesh’ in anywhere of its name.

SQL:
SELECT Movie_Name, Genre, Year FROM Movie WHERE Movie_Name LIKE '%Byomkesh%';

Question 2.1.9: Find the name and IMDb rating of those movies whose name consists of only 2 characters.
SQL:
SELECT Movie_Name, IMDB_Rating FROM Movie WHERE Movie_Name LIKE '__';

Question 2.1.10: Find the name, genre and IMDb rating of all the movies that contain at least 3 characters in its name.
SQL:
SELECT Movie_Name, Genre, IMDB_Rating FROM Movie WHERE Movie_Name LIKE '___%';

Question 2.1.final: Find the name, IMDb rating and the name of the directors of all the moves that contain at most 10 characters in its name.

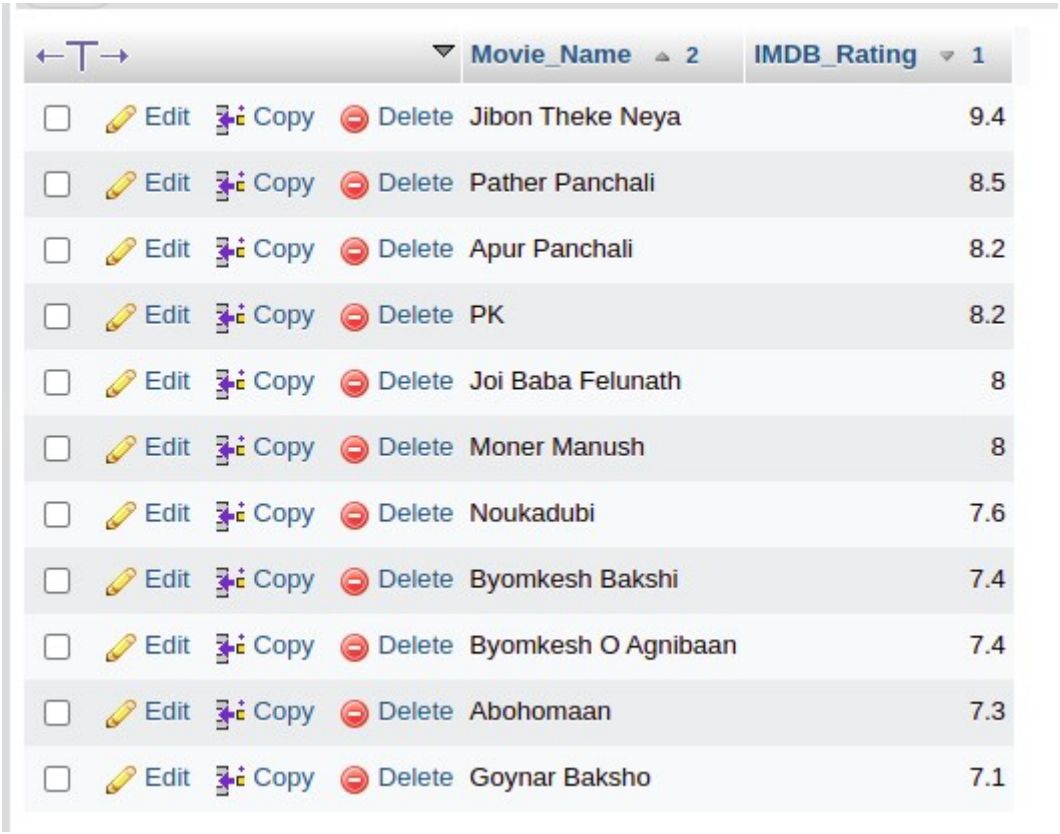
SQL:
SELECT Movie_Name, Genre, IMDB_Rating FROM Movie WHERE Movie_Name NOT LIKE '__________%';

Lab 3:

Problem 1.1: Display all the movies according to the IMDb rating (in descending order); if multiple movies have same IMDb rating then rearrange those movies alphabetically according to the movie-name.

SQL:
SELECT Movie_Name, IMDB_Rating FROM Movie ORDER BY IMDB_Rating DESC, Movie_Name ASC

Output:



	Movie_Name ▲ 2	IMDB_Rating ▼ 1
Edit Copy Delete	Jibon Theke Neya	9.4
Edit Copy Delete	Pather Panchali	8.5
Edit Copy Delete	Apur Panchali	8.2
Edit Copy Delete	PK	8.2
Edit Copy Delete	Joi Baba Felunath	8
Edit Copy Delete	Moner Manush	8
Edit Copy Delete	Noukadubi	7.6
Edit Copy Delete	Byomkesh Bakshi	7.4
Edit Copy Delete	Byomkesh O Agnibaan	7.4
Edit Copy Delete	Abohomaan	7.3
Edit Copy Delete	Goynar Baksho	7.1

Question 1.2: Find the total number of movies in the DB.

SQL:
SELECT COUNT(Movie_Id) AS Total_Movie FROM Movie

Question 1.3: Find how many movie-genres are there in the DB.

SQL:
SELECT COUNT(DISTINCT Genre) AS TotalGenre FROM Movie;

Question 1.4: Find the average IMDb rating of all the movies of drama genre.

SQL:
SELECT AVG(IMDB_Rating) FROM Movie WHERE Genre = 'Drama'

Question 1.5: Find total number of movies in each genre.

SQL:
SELECT Genre, COUNT(Movie_Id) AS NumberOfMovies FROM Movie GROUP BY Genre;

Question 2.1: Find the name and the current age of all the actors and directors.

SQL:
(SELECT Actor_Name as Name, 2022 - Birth_Year AS Age FROM Actor) UNION (SELECT Director_Name as Name, 2022 - Birth_Year AS Age FROM Director);

Question 2.2: Find the name, birth year , number of awards as an actor and the number of awards as a director of those actors who directed at least one movie of those directors who acted in at least one movie.

SQL:
SELECT Director.Director_Name AS Name, Director.Birth_Year, Director.No_Of_Awards from Director, Actor WHERE Director.Person_ID = Actor.Person_ID;

Question 2.3: Find the name, genre, IMDb rating and the name of the director of all the movies directed by Satyajit Ray, Zahir Raihan, Riporno Ghosh and Kaushik Ganguly.

SQL:
SELECT Movie.Movie_Name, Movie.Genre, Movie.IMDB_Rating, Director.Director_Name from Movie NATURAL JOIN Director WHERE Director.Person_ID = Movie.Director_Id AND Director_Name in ('Saytajit Ray', 'Zahir Raihan', 'Rituparno Ghosh', 'Kaushik Ganguly');

Question 2.4: Find the number of movies directed by a director in average.

SQL:
SELECT AVG(No_Of_Films) AS AverageFilms from Director;

Question 2.5: Find the name of all the directors and the number of movies directed by each director considering there exists no No of Films column in the director table.

SQL:
SELECT Director.Director_Name, COUNT(Movie.Director_Id) AS No_Of_Films from Director NATURAL JOIN Movie WHERE Director.Person_ID = Movie.Director_Id GROUP BY Director.Person_ID;

Lab 4:

Query 1.1: Display the name, IMDb rating and genre of all the movies with an IMDB rating greater than than or equal to 8.0

SQL:

Query 1.2:

SQL:
SELECT Genre, COUNT(Movie_ID) Number_Of_Movies from Movie WHERE IMDB_Rating >= 8.0 GROUP BY Genre;

Query 1.3:

SQL:
SELECT Genre, COUNT(IMDB_Rating) from Movie GROUP BY Genre HAVING AVG(IMDB_Rating) >= 7.5;

Query 1.4:

SQL:
SELECT Director_Name, No_Of_Films, No_Of_Awards from Director WHERE No_Of_Films = (SELECT MAX(No_Of_Films) from Director);

Query 1.5:

SQL:
SELECT Actor_Name, (Year(CURRENT_DATE) - Birth_Year) as Age, No_Of_Films, No_Of_Awards from Actor where Birth_Year = (SELECT MAX(Birth_Year) from Actor);

Label 2:

Query 2.1: Find the name, number of films as a director, number of films as an actor, number of awards as a director and number of awards as an actor of all the directors(If a director never acted in any film then number of films and number of awards as an actor will be null).

SQL:
SELECT D.Director_Name AS Name, D.No_Of_Films AS Director_No_Of_Films, A.No_Of_Films AS Actor_No_Of_Films, D.No_Of_Awards AS Director_No_Of_Awards, A.No_Of_Awards AS Actor_No_Of_Awards FROM Director AS D LEFT OUTER JOIN Actor AS A USING (Person_ID);

Query 2.2: Find the name, number of films as a director, number of films as an actor, number of awards as a director and number of awards as an actor of all the actor and directors.

SQL:
(SELECT D.Director_Name AS Name, D.No_Of_Films AS Director_No_Of_Films, A.No_Of_Films AS Actor_No_Of_Films, D.No_Of_Awards AS Director_No_Of_Awards, A.No_Of_Awards AS Actor_No_Of_Awards FROM Director AS D LEFT OUTER JOIN Actor AS A USING (Person_ID)) UNION (SELECT A.Actor_Name AS Name, D.No_Of_Films AS Director_No_Of_Films, A.No_Of_Films AS Actor_No_Of_Films, D.No_Of_Awards AS Director_No_Of_Awards, A.No_Of_Awards AS Actor_No_Of_Awards FROM Director AS D RIGHT OUTER JOIN Actor AS A USING (Person_ID));

Query 2.3: Find the name, number of films and number of awards of those actors who never directed any film.

SQL:
SELECT Actor_Name, No_Of_Films, No_Of_Awards FROM Actor WHERE Person_ID NOT IN (SELECT Person_ID FROM Director);

Query 2.4: Find the name, IMDb rating and genre of those movies whose IMDb Rating is above average.

SQL:
SELECT Movie_Name, Genre, IMDB_Rating FROM Movie WHERE IMDB_Rating > (SELECT AVG(IMDB_Rating) FROM Movie);

Query 2.5: Among all the movie genres, find the highest average IMDb rating.

SQL:
SELECT MAX(Avg_IMDB_Rating) FROM (SELECT Genre, AVG(IMDB_Rating) AS Avg_IMDB_Rating FROM Movie GROUP BY Genre) AS Avg_Genre ;

Lab: 5

Query 4.1: Find the name and age of all the move characters who are not teenagers.

SQL:
SELECT Character_Name, Age FROM `Movie_Characters` WHERE Age > 19 OR Age < 13

Query 4.2: Find the movie ID of all the movies and the number of movie characters in each movie.

SQL:
SELECT Movie_ID, count(Movie_ID) AS CharacterCount FROM
`Movie_Character_Relationship` GROUP BY Movie_ID;

Query 4.3: Find the movie name of all movies and the number of movie-characters in each movie.

SQL:
SELECT Movie.Movie_Name, CharacterCount FROM (SELECT Movie_ID, count(Movie_ID) AS CharacterCount FROM `Movie_Character_Relationship` GROUP BY Movie_ID) AS RelationTable, Movie WHERE Movie.Movie_Id = RelationTable.Movie_ID ;

Query 4.4: Find the name year of release IMDB rating and the name of the director of all the movies that contain at least 12 characters (including spaces) in their name.

SQL:
SELECT Movie_Name, Year, IMDB_Rating FROM Movie JOIN Director WHERE Movie.Director_Id = Director.Person_ID AND length(Movie_Name) >=12;

Query 4.5: Find the name year of release IMDB rating and the name of the director of all the movies that contain at most 12 characters (including spaces) in their name.

SQL:
SELECT Movie_Name, Year, IMDB_Rating FROM Movie JOIN Director WHERE Movie.Director_Id = Director.Person_ID AND length(Movie_Name) <=12;

Query 2.6: Find the name, year of release, IMDB rating of all the movies of Byomkesh (NB: A movie of Byomkesh may not contain the term Byomkesh in it’s name);

SQL:

```
SELECT Movie_Name, Year, IMDB_Rating FROM Movie_Characters LEFT OUTER JOIN
Movie_Character_Relationship ON Movie_Characters.Character_ID =
Movie_Character_Relationship.Character_ID LEFT OUTER JOIN Movie ON Movie.Movie_Id =
Movie_Character_Relationship.Movie_ID
WHERE Character_Name LIKE '%Byomkesh%';
```

Query 2.7: Find the name and age of all the characters from the movies of Feluda

SQL:

```
SELECT Character_Name, Age from (SELECT Movie.Movie_ID FROM Movie_Characters  JOIN
Movie_Character_Relationship ON Movie_Characters.Character_ID =
Movie_Character_Relationship.Character_ID  JOIN Movie ON Movie.Movie_Id =
Movie_Character_Relationship.Movie_ID
WHERE Character_Name LIKE '%Feluda%') as MovieForCharacter  JOIN
Movie_Character_Relationship ON MovieForCharacter.Movie_ID =
Movie_Character_Relationship.Movie_ID   JOIN Movie_Characters ON
Movie_Characters.Character_ID = Movie_Character_Relationship.Character_ID;
```

Lab 6:

* View:
1. Movie join(x) movie.char
SQL:
CREATE VIEW movie_character_view AS SELECT Movie.Movie_Name, Movie.IMDB_Rating, Movie_Characters.Character_Name, Movie_Characters.Age FROM Movie JOIN Movie_Character_Relationship ON Movie.Movie_Id = Movie_Character_Relationship.Movie_ID JOIN Movie_Characters ON Movie_Character_Relationship.Character_ID = Movie_Characters.Character_ID;

2. Each genre & Highest IMDB rating in each genre
SQL:
CREATE VIEW GenreMaxRatingView AS SELECT Genre, MAX(IMDB_Rating) AS Max_Rating from Movie GROUP BY Genre;

3. From Movie Relation.
SQL:
CREATE VIEW MovieView AS SELECT * from Movie;

Test 1:
SQL:
INSERT INTO movie_character_view (Movie_Name, IMDB_Rating, Character_Name, Age) VALUES ('Hello World', 8.5, 'CharacterName', 41)
=> Gives error: `Can not modify more than one base table through a join view 'Movie_Info.movie_character_view`

Test 2:
SQL:
INSERT INTO GenreMaxRatingView (Genre, Max_Rating) VALUES ('Action', 7.8)
=> Gives error: `The target table GenreMaxRatingView of the INSERT is not insertable-into`

Test 3:
SQL:
INSERT INTO MovieView (Movie_ID, Movie_Name, Genre, Year, IMDB_Rating, Director_ID) VALUES (12, 'Name', 'Action', 2015, 7.6, 4)
=> Values get inserted in main table

Work:
Find the name and age of all the characters from all the movies directed by Satyajit Ray
i) By using sub queries
ii) By using join