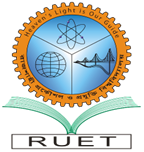
***“Heaven’s Light is Our Guide”***



**Department of Computer Science & Engineering**

**RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOG**

**Lab Report**

|  |
| --- |
| **Submitted By:**  **Name: Khandoker Sefayet Alam**  **Roll:2003121**  **Department: Computer Science & Engineering**  **Section-C**    **Course code: CSE 3102**  **Course name: Database Management System** |

|  |
| --- |
| **Submitted To:**  **Barshon Sen** |

TABLE Creation:

CREATE TABLE `movie\_characters` (

  `Character \_ID` int(11) NOT NULL,

  `Character\_Name` varchar(100) NOT NULL,

  `Age` int(11) NOT NULL

)

INSERT INTO `movie\_characters` (`Character \_ID`, `Character\_Name`, `Age`) VALUES

(1, 'Apu', 7),

(2, 'Durga', 10),

(3, 'Harihar', 42),

(4, 'Sarbajaya', 35),

(5, 'Ramesh', 28),

(6, 'Hemnalini', 25),

(7, 'Nalinaksha', 30),

(8, 'Kamala', 20),

(9, 'Shikha', 21),

(10, 'Apratim', 26),

(11, 'Feluda', 25),

(12, 'Maganlal Meghraj', 35),

(13, 'Jatayu', 45),

(14, 'Topshe', 18),

(15, 'Ruku', 6),

(16, 'Faruk', 22),

(17, 'Bithi', 19),

(18, 'Sathi', 24),

(19, 'Lalon', 0),

(20, 'Kaluah', 25),

(21, 'Siraj Saain', 45),

(22, 'Komli', 25),

(23, 'Subir Banerjee', 0),

(24, 'Ashima', 24),

(25, 'Rashmoni', 0),

(26, 'Somalata', 24),

(27, 'Chaitali', 18),

(28, 'Byomkesh Bakshi', 26),

(29, 'Ajit', 27),

(30, 'Malati', 16),

(31, 'Debkumar', 50),

(32, 'Doctor Anukul', 45),

(33, 'Prabhat', 27),

(34, 'Satyabati', 22),

(35, 'Anadi Babu', 55),

(36, 'PK', 0),

(37, 'Jaggu', 26),

(38, 'Sarfaraz', 29),

(39, 'Tapasvi Maharaj', 50);

ALTER TABLE `movie\_characters`

  ADD PRIMARY KEY (`Character \_ID`);

CREATE TABLE `movie\_character\_relationship` (

  `Character\_ID` int(11) NOT NULL,

  `Movie\_ID` int(11) NOT NULL

)

INSERT INTO `movie\_character\_relationship` (`Character\_ID`, `Movie\_ID`) VALUES

(1, 1),

(2, 1),

(3, 1),

(4, 1),

(5, 2),

(6, 2),

(7, 2),

(8, 2),

(9, 3),

(10, 3),

(11, 4),

(12, 4),

(13, 4),

(14, 4),

(15, 4),

(16, 5),

(17, 5),

(18, 5),

(19, 6),

(20, 6),

(21, 6),

(22, 6),

(23, 7),

(24, 7),

(25, 8),

(26, 8),

(27, 8),

(28, 9),

(28, 10),

(29, 9),

(29, 10),

(30, 9),

(31, 9),

(32, 9),

(33, 10),

(34, 9),

(34, 10),

(35, 10),

(36, 11),

(37, 11),

(38, 11),

(39, 11);

ALTER TABLE `movie\_character\_relationship`

  ADD KEY `Movie\_ID` (`Movie\_ID`),

  ADD KEY `Character\_ID` (`Character\_ID`);

**Level # 1**

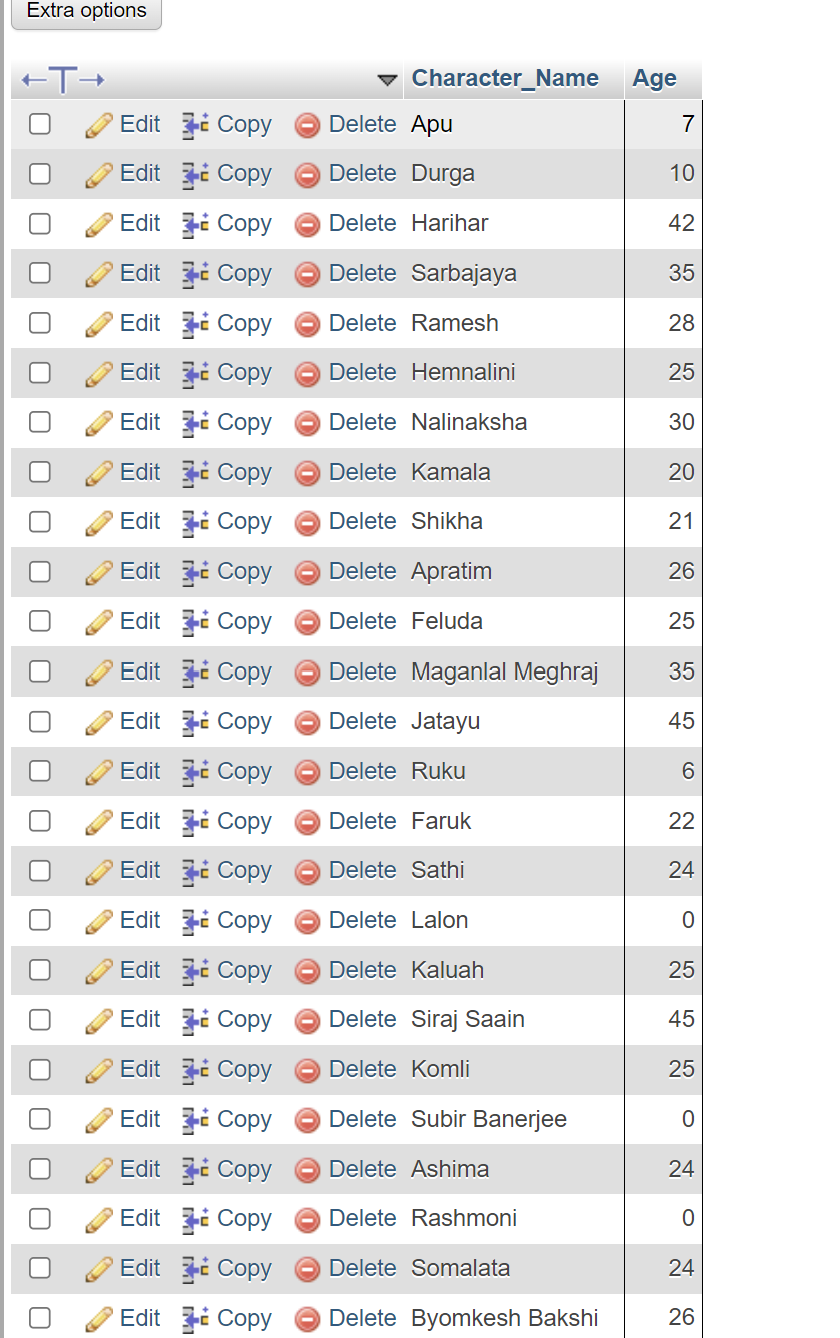
❖ Query 1.1: Find the name and age of all the movie-characters who are not

teenagers.

Code:

SELECT Character\_Name,Age FROM movie\_characters WHERE (Age>=20 OR Age<=12)

Table:



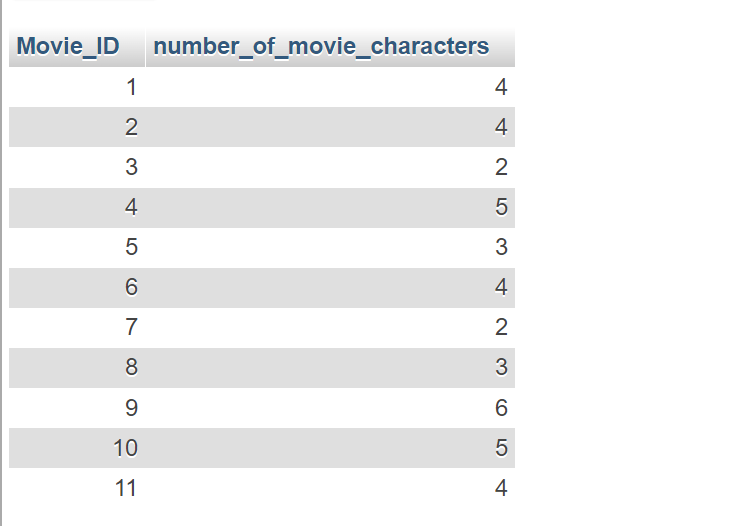
❖ Query 1.2: Find the movie ID of all the movies and the number of

movie-characters in each movie.

Code:

SELECT Movie\_ID, COUNT(Character\_ID) As number\_of\_movie\_characters FROM movie\_character\_relationship GROUP BY Movie\_ID

Table:



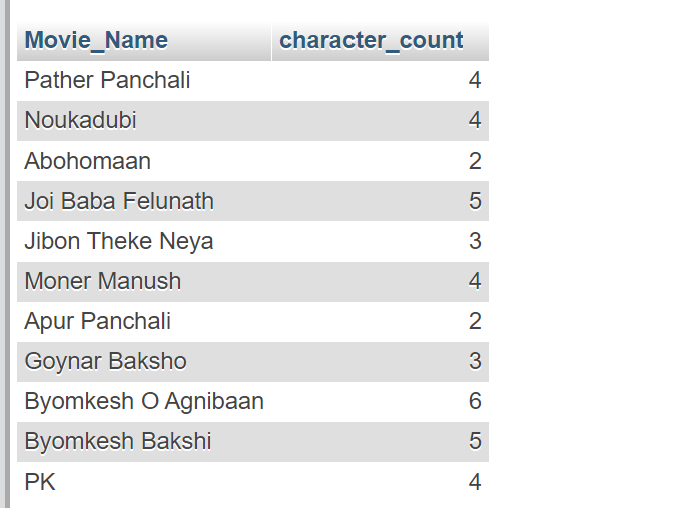
❖ Query 1.3: Find the movie name of all the movies and the number of

movie-characters in each movie.

Code:

SELECT movie.Movie\_Name,COUNT(movie\_character\_relationship.Character\_ID) As character\_count FROM movie LEFT JOIN movie\_character\_relationship ON movie.Movie\_ID=movie\_character\_relationship.Movie\_ID GROUP BY movie.Movie\_ID

Table:



❖ Query 1.4: Find the name, year of release, IMDB rating and the name of the

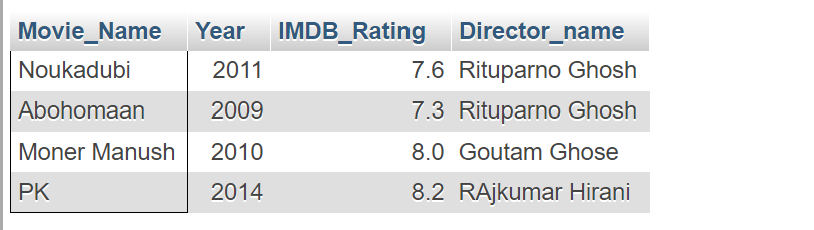
director of all the movies that contain at most 12 characters (including

spaces) in their name.

Code:

SELECT movie.Movie\_Name,movie.Year,movie.IMDB\_Rating,director.Director\_name FROM movie LEFT JOIN director ON movie.Director\_ID=director.Person\_Id WHERE LENGTH(movie.Movie\_Name)<=12

Table:



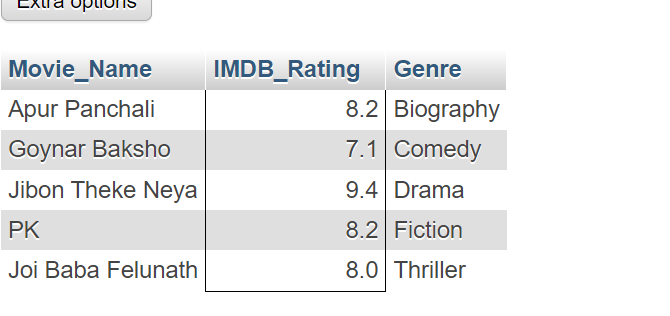
❖ Query 1.5: Display the name and the IMDb rating of the movie with the

highest IMDb rating in each genre.

Code:

SELECT movie.Movie\_Name,movie.IMDB\_Rating,movie.Genre FROM Movie JOIN (SELECT MAX(IMDB\_Rating) AS MAXM,Genre FROM movie GROUP BY Genre) SUB ON movie.Genre=SUB.Genre WHERE movie.IMDB\_Rating=SUB.MAXM

Table:



**Level # 2**

❖ Query 2.1: Find the name, year of release, IMDB rating of all the movies of

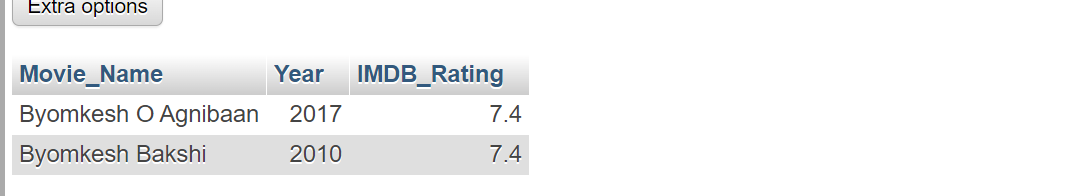
Byomkesh (N.B.**:** A movie of Byomkesh may not contain the term

‘Byomkesh’ in its name).

Code:

SELECT movie.Movie\_Name,movie.Year,movie.IMDB\_Rating FROM (SELECT \* FROM movie\_characters JOIN movie\_character\_relationship USING (Character\_ID) WHERE movie\_characters.Character\_Name LIKE '%Byomkesh%') as sub1 JOIN movie USING (movie\_id);

Table:



❖ Query 2.2: Find the name of all the movies and, the name & the age of the

youngest movie-character in each movie.

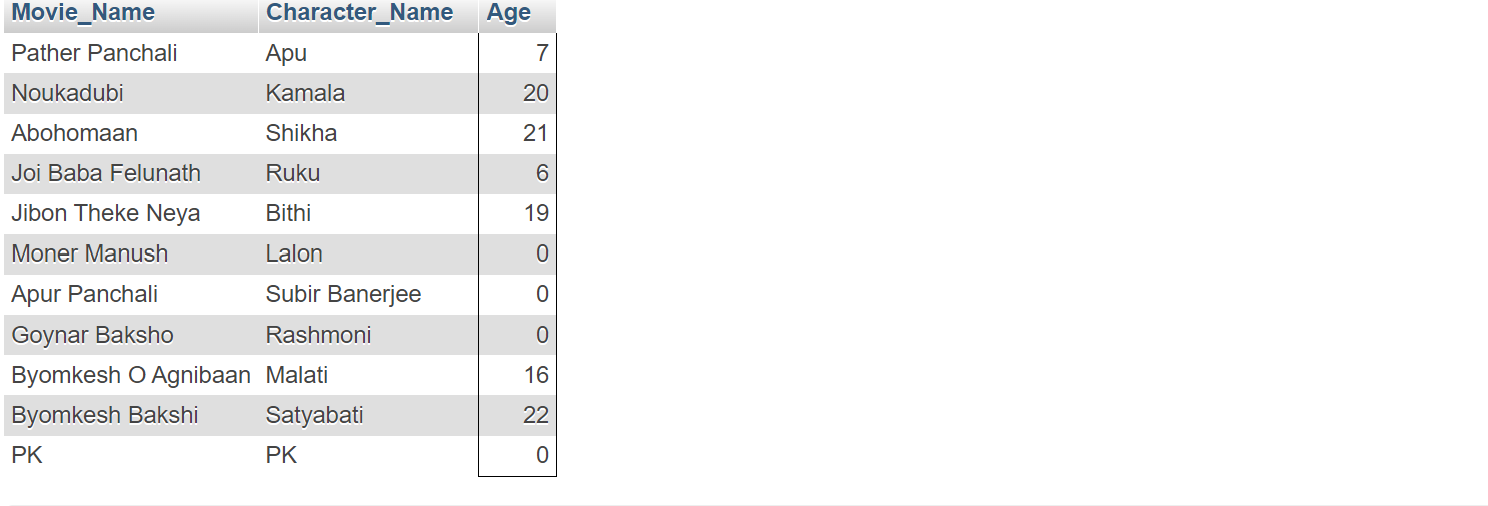
Code:

SELECT Movie.Movie\_Name,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

WHERE movie\_characters.Age = (SELECT MIN(movie\_characters.Age) FROM movie\_character\_relationship JOIN Movie\_characters ON movie\_character\_relationship.Character\_ID = movie\_characters.Character\_ID WHERE movie\_character\_relationship.Movie\_ID = movie.Movie\_ID);

Table:



❖ Query 2.3: Find the name and age of all the characters from the movies of

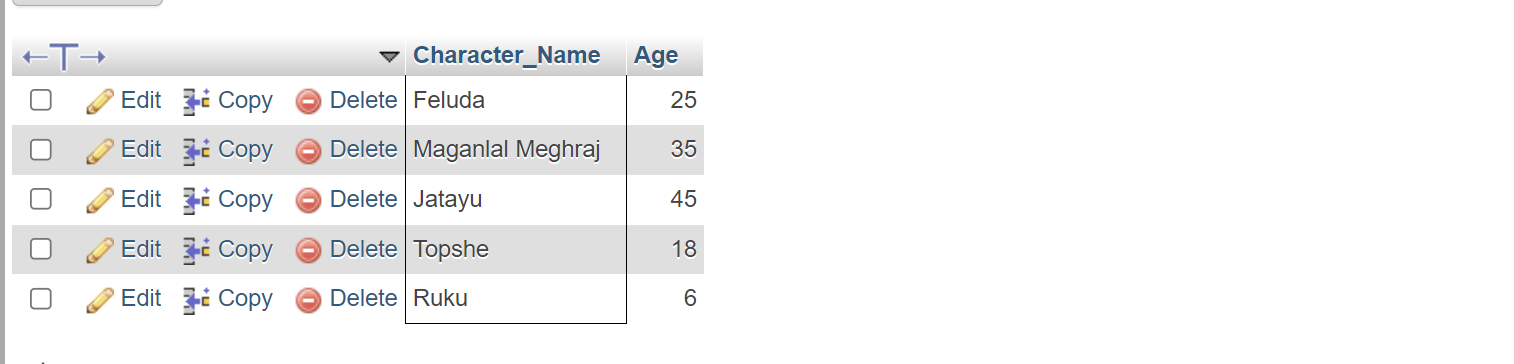
Feluda (N.B.**:** A movie of Feluda may not contain the term ‘Feluda’ in its

name).

Code:

SELECT Character\_Name,Age from movie\_characters WHERE Character\_ID IN (SELECT Character\_ID FROM movie\_character\_relationship WHERE movie\_character\_relationship.Movie\_ID IN (SELECT movie.Movie\_ID FROM (SELECT \* FROM movie\_characters JOIN movie\_character\_relationship USING (Character\_ID) WHERE movie\_characters.Character\_Name LIKE '%Feluda%') as sub1 JOIN movie USING (movie\_id)));

Table:



❖ Query 2.4: Find the name and age of all the characters from the movies

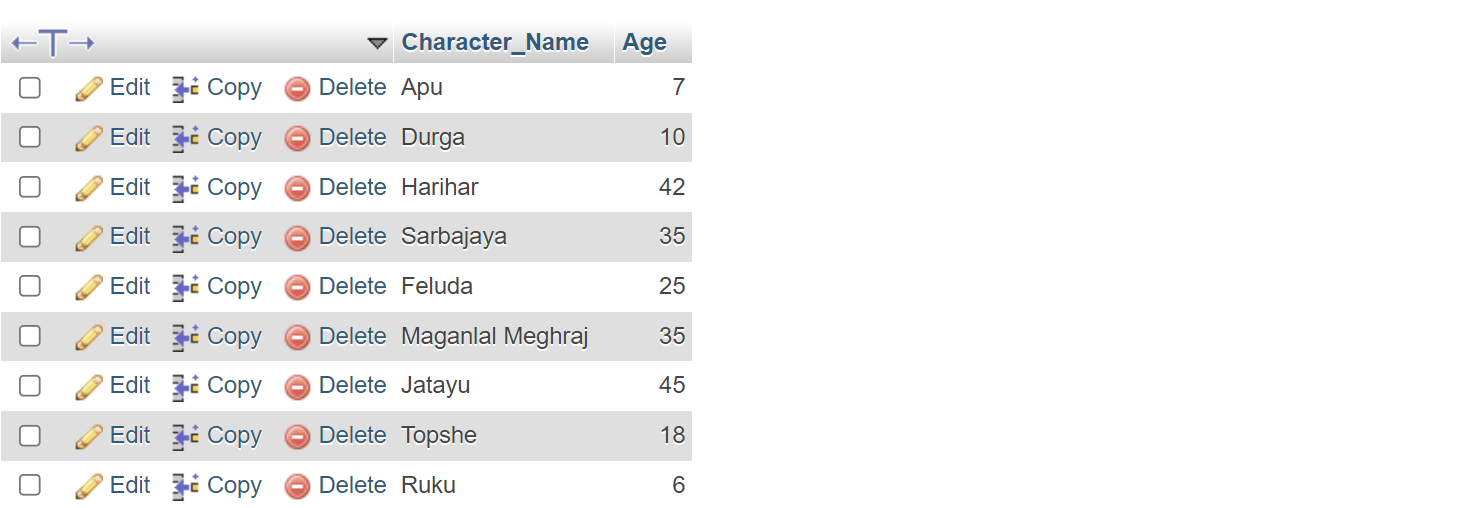
directed by Satyajit Ray.

➵ 2.4.1: Without Join (using Subqueries)

Code:

SELECT Character\_Name,Age from movie\_characters WHERE Character\_ID IN (SELECT Character\_ID FROM movie\_character\_relationship WHERE Movie\_ID IN (SELECT Movie\_ID FROM movie Where director\_id=(SELECT Person\_id FROM director WHERE director\_name Like '%Satyajit Ray%')));

Table:



➵ 2.4.2: With Join

Code:

SELECT Character\_name,Age from movie\_characters join movie\_character\_relationship on movie\_characters.Character\_ID=movie\_character\_relationship.Character\_ID JOIN movie USING (movie\_id) JOIN director ON director.Person\_Id=movie.Director\_ID Where director.Director\_name LIKE '%Satyajit Ray%';

Table:



❖ Query 2.5: Find the movie genre with highest average IMDb rating.

Code:

SELECT Genre,Genre\_avg FROM (SELECT Genre, AVG(IMDB\_Rating) as Genre\_avg FROM movie GROUP BY genre) AS SUB WHERE Genre\_Avg=(SELECT MAX(Genre\_avg) FROM (SELECT AVG(IMDB\_Rating) as Genre\_avg FROM movie GROUP BY genre) As sub2)

Table:

