**Assignment - 1**

1. \*\*Identify the primary keys and foreign keys in maven movies db. Discuss the differences:\*\*

**Primary Key:** A primary key is a column or a set of columns in a table that uniquely identifies each record in the table. It must have a unique value for each row, and it cannot contain NULL values. There can be only one primary key in a table.

**Foreign Key:** A foreign key is a column or a set of columns in a table that refers to the primary key of another table. It establishes a link between the data in two tables, creating a relationship between them. The foreign key constraint ensures referential integrity, meaning that values in the foreign key column must match values in the referenced primary key column. Let's take an example to illustrate these concepts:

-- Movies table

CREATE TABLE Movies (

MovieID INT PRIMARY KEY,

Title VARCHAR(255),

ReleaseYear INT

);

-- Actors table

CREATE TABLE Actors (

ActorID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50)

);

-- Roles table with foreign keys

CREATE TABLE Roles (

RoleID INT PRIMARY KEY,

MovieID INT,

ActorID INT,

CharacterName VARCHAR(100),

FOREIGN KEY (MovieID) REFERENCES Movies(MovieID),

FOREIGN KEY (ActorID) REFERENCES Actors(ActorID)

);

In this example:

Movies.MovieID is the primary key of the "Movies" table.

Actors.ActorID is the primary key of the "Actors" table.

Roles.RoleID is the primary key of the "Roles" table. Roles.MovieID is a foreign key that references the primary key Movies.MovieID.

Roles.ActorID is a foreign key that references the primary key Actors.ActorID.

The foreign keys (MovieID and ActorID in the "Roles" table) establish relationships between the "Roles" table and the "Movies" and "Actors" tables, respectively.

2. \*\*List all details of actors:\*\*

**SELECT \* FROM actors;**

3. \*\*List all customer information from DB:\*\*

**SELECT \* FROM customer;**

4. \*\*List different countries:\*\*

**SELECT DISTINCT country FROM countries;**

5. \*\*Display all active customers:\*\*

**SELECT \* FROM customer WHERE active = 1;**

6. \*\*List of all rental IDs for customer with ID 1:\*\*

**SELECT rental\_id FROM rentals WHERE customer\_id = 1;**

7. \*\*Display all the films whose rental duration is greater than 5:\*\*

**SELECT \* FROM films WHERE rental\_duration > 5;**

8. \*\*List the total number of films whose replacement cost is greater than $15 and less than $20:\*\*

**SELECT COUNT(\*) FROM films WHERE replacement\_cost > 15 AND replacement\_cost < 20;**

9. \*\*Display the count of unique first names of actors:\*\*

**SELECT COUNT(DISTINCT first\_name) FROM actors;**

10. \*\*Display the first 10 records from the customer table:\*\*

**SELECT \* FROM customer LIMIT 10;**

11. \*\*Display the first 3 records from the customer table whose first name starts with ‘b’:\*\*

**SELECT \* FROM customer WHERE first\_name LIKE 'b%' LIMIT 3;**

12. \*\*Display the names of the first 5 movies which are rated as ‘G’:\*\*

**SELECT title FROM films WHERE rating = 'G' LIMIT 5;**

13. \*\*Find all customers whose first name starts with "a":\*\*

**SELECT \* FROM customer WHERE first\_name LIKE 'a%';**

14. \*\*Find all customers whose first name ends with "a":\*\*

**SELECT \* FROM customer WHERE first\_name LIKE '%a';**

15. \*\*Display the list of first 4 cities which start and end with ‘a’:\*\*

**SELECT city FROM cities WHERE city LIKE 'a%a' LIMIT 4;**

16. \*\*Find all customers whose first name has "NI" in any position. Which SQL function do we need to use:\*\*

**SELECT \* FROM customer WHERE first\_name LIKE '%NI%';**

17. \*\*Find all customers whose first name has "r" in the second position:\*\*

**SELECT \* FROM customer WHERE SUBSTRING (first\_name, 2, 1) = 'r';**

18. \*\*Find all customers whose first name starts with "a" and is at least 5 characters in length:\*\*

**SELECT \* FROM customer WHERE first\_name LIKE 'a%' AND LENGTH(first\_name) >= 5;**

19. \*\*Find all customers whose first name starts with "a" and ends with "o":\*\*

**SELECT \* FROM customer WHERE first\_name LIKE 'a%o';**

20. \*\*Get the films with pg and pg-13 rating using IN operator:\*\*

**SELECT \* FROM films WHERE rating IN ('PG', 'PG-13');**

21. \*\*Get the films with length between 50 to 100 using BETWEEN operator:\*\*

**SELECT \* FROM films WHERE length BETWEEN 50 AND 100;**

22. \*\*Get the top 50 actors using LIMIT operator:\*\*

**SELECT \* FROM actors LIMIT 50;**

23. \*\*Get the distinct film IDs from the inventory table:\*\*

**SELECT DISTINCT film\_id FROM inventory;**