1.Movie Table

* Primary Key: MovieID (Unique identifier for each movie)
* Columns: Title, ReleaseDate, Director, Description, GenreID (Foreign Key), ArtistID (Foreign Key)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| movieId | Tittle | Director | GernreID | Artishid | Release date |
| 1 | 8 Mile | Curtis Hanson | 11 | 100 | 22-09-2024 |
| 2 | The Adventures of Priscilla, Queen of the Desert | Stephan Elliott | 12 | 101 | 23-12-2023 |
| 3 | The Pianist | Roman Polanski | 13 | 100 | 23-11-2024 |
| 4 | Pitch Perfect | Jason Moore | 14 | 105 | 22-12-2023 |
| 5 | Elvis | MArk | 16 | 106 | 12-09-2023 |
| 6 | On the Come Up | On the Come Up | 17 | 107 | 09-5-2023 |

CREATE TABLE Movie (

MovieID INT PRIMARY KEY,

Title VARCHAR(255) NOT NULL,

ReleaseDate DATE,

Director VARCHAR(255),

GenreID INT,

ArtistID INT,

CONSTRAINT FK\_Genre FOREIGN KEY (GenreID) REFERENCES Genre(GenreID),

CONSTRAINT FK\_Artist FOREIGN KEY (ArtistID) REFERENCES Artist(ArtistID)

);

2.Genre

* Primary Key: GenreID (Unique identifier for each genre)
* Columns: GenreName

CREATE TABLE Grenre(

GenreID INT PRIMARYKEY,

Genre Name VARCHAR,

);

|  |  |
| --- | --- |
| Genre ID | Genre Name |
| 11 | Drama |
| 12 | Music |
| 13 | Love |
| 14 | Drama |
| 15 | Adventure |
| 16 | Thrill |
| 17 | comedy |

3.user table

CREATE TABLE User(

UserID INT PRIMARYKEY,

User Name VARCHAR,

Email VARCHAR,

);

* Primary Key: UserID (Unique identifier for each user)
* Columns: Username, Email

|  |  |  |  |
| --- | --- | --- | --- |
| UserId | username | Email |  |
| 1000 | Rose | Rose@gmail.com |
| 2000 | Lemon | Lemon@gmail.com |
| 3000 | Lilly | lilly@gmail.com |
| 4000 | Jack | jack@gmail.com |
| 5000 | Sam | Sam@gmail.com |
| 6000 | Tom | Tom@gmail.com |
| 7000 | naven | naven@gmail.com |

4. Review Table:

* Primary Key: ReviewID (Unique identifier for each review)
* Columns: MovieID (Foreign Key), UserID (Foreign Key), Rating, Comment, DatePosted
* CREATE TABLE Review (
* Review Id INT PRIMARY KEY,
* Ratings INT,
* MovieID INT,
* UserID INT,
* Comment VARCHAR,
* CONSTRAINT FK\_Genre FOREIGN KEY (UserID) REFERENCES Genre(GenreID),
* CONSTRAINT FK\_Artist FOREIGN KEY (MovieID) REFERENCES Artist(ArtistID)
* );

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Review Id | Movie Id | userId | Ratings | Comment |
| 22 | 1 | 1000 | 5star | verygood |
| 33 | 2 | 2000 | 3star | Good |
| 44 | 3 | 3000 | 5star | Excellent |
| 55 | 4 | 4000 | 2star | Poor |
| 66 | 5 | 5000 | 3star | Nice |
| 77 | 6 | 6000 | 6star | Marvellous |
| 88 | 7 | 7000 | 5star | excellent |

5. Artists Table:

CREATE TABLE Artist(

ArtistID INT PRIMARYKEY,

Artist Name VARCHAR,

BirthDate Date,

Biography VARCHAR,

);

* Primary Key: ArtistID (Unique identifier for each artist)
* Columns: ArtistName, BirthDate, Biography

|  |  |  |  |
| --- | --- | --- | --- |
| Artist id | ArtistName | BirthDate | Biography |
| 101 | Barath | 12-071975 | Doctor |
| 102 | Jack | 24-09-1977 | Doctor |
| 103 | Manu | 12-03-1978 | Dancer |
| 100 | Adev | 12-09-1998 | Musician |
| 105 | Vijay | 10-09-1976 | Comedian |
| 107 | Soorya | 22-10-1994 | Stund king |
| 106 | Ajith | 23-05-1992 | comedian |

INSERT INTO movies VALUES (7, "neverland2", "pan", 11, "105","4-08-2024");

4. INSERT INTO movies VALUES (8, "spiderman", "martin", 12, "103","3-06-2024");

Relationship

* The Movie Table has two foreign keys: GenreID and ArtistID. The GenreID links to the Genre Table, indicating the genre of the movie, and the ArtistID links to the Artists Table, indicating the director or primary artist associated with the movie.
* The Genre Table has no direct relationships with other tables, but it provides a list of genres that can be associated with movies.
* The Users Table has no direct relationships with other tables in this design, but it represents the users who can write reviews for movies.
* The Review Table has two foreign keys: MovieID and UserID. The MovieID links to the Movie Table, indicating the movie being reviewed, and the UserID links to the Users Table, indicating the user who wrote the review.
* The Artists Table has no direct relationships with other tables in this design, but it provides information about artists associated with movies in the Movie Table.