Group 27
COS221 Practical Assignment 5

Task 4

Relational Schema:

Genre(GenreID, Description, CatalogID)
Catalog(CatalogID, Title, Director, ReleaseDate)
Movies(Duration, CatalogID)
Shows(Seasons, Episodes, CatalogID)
User(UserID, Password, Email)
Actors(ActorID, Name(FName, LName), CatalogID)
Accounts(AccountID, PaymentDetails(AccountHolder, AccountNumber, CVV), SubscriptionType, SubscriptionStartDate, SubscriptionEndDate, UserID)
User-Views(UserID, CatalogID)

Primary keys:
- Genre Table: GenreID
- Catalog Table: CatalogID
- User Table: UserID
- Actors Table: ActorID
- Accounts Table: AccountID

Secondary keys:
- User Table: Email can be used as a secondary key

Foreign keys:
- Genre Table: CatalogID
- Movies Table: CatalogID
- Shows Table: CatalogID
- Actors Table: CatalogID
- Accounts Table: UserID
- User-Views Table: UserID, CatalogID

Constraints:
- Primary keys cannot be NULL

Genre

Attribute	Datatype
GenreID (PK)	INTEGER
Description	VARCHAR(100)
CatalogID (FK)	INTEGER

Catalog

Attribute	Datatype
CatalogID (PK)	INTEGER
Title	VARCHAR(255)
Director	VARCHAR(100)

ReleaseDate	YEAR

Movies

Attribute	Datatype
Duration	TIME
CatalogID (FK)	INTEGER

Shows

Attribute	Datatype
Seasons	INTEGER
Episodes	INTEGER
CatalogID (FK)	INTEGER

User

Attribute	Datatype
UserID (PK)	INTEGER
Password	VARCHAR(100)
Email	VARCHAR(255)

Actors

Attribute	Datatype
ActorID (PK)	INTEGER
FName	VARCHAR(150)
LName	VARCHAR(150)
CatalogID (FK)	INTEGER

Accounts

Attribute	Datatype
AccountID (PK)	INTEGER
AccountHolder	VARCHAR(150)
AccountNumber	VARCHAR(255)
CVV	VARCHAR(4)
SubscriptionType	VARCHAR(50)
SubscriptionStartDate	DATE
SubscriptionEndDate	DATE
UserID (FK)	INTEGER

User-Views

Attribute	Datatype
UserID (FK)	INTEGER
CatalogID (FK)	INTEGER

Relational Database Schema:

Genre Table

GenreID	Description	CatalogID
1	Action	1
2	Comedy	2

3	Drama	3

Catalog Table

CatalogID	Title	Director	ReleaseDate
1	The Matrix	Lana Wachowski	1999-03-31
2	Inception	Christopher Nolan	2010-07-16
3	The Shawshank Redemption	Frank Darabont	1994-09-23

Movies Table

Duration	CatalogID
02:16:00	1
02:28:00	2
02:22:00	3

Shows Table

Seasons	Episodes	CatalogID
5	122	5
2	10	10
1	25	23

User Table

UserID	Password	Email
1	Password123	User1@example.com
2	Abc123	User2@gmail.com
3	Pass456	User3@yahoo.com

Actors Table

ActorID	FName	LName	CatalogID
1	Keanu	Reeves	1
2	Leonardo	DiCaprio	2
3	Tim	Robbins	3

Accounts Table

Account ID	AccountHolder	AccountNumber	CVV	Subscription Type	SubscriptionStartDate	SubscriptionEndDate	UserID
1	John Doe	123456789	123	Premium	2024-01-01	2024-12-31	1
2	Jane Doe	987654321	456	Basic	2024-02-01	2024-12-31	2
3	Mary Doe	111122223	789	Premium	2024-03-01	2024-12-31	3

User-Views Table

UserID	CatalogID
1	4
2	5
3	6

SQL statements:

CREATE TABLE Genre (

```
    GenreID INTEGER AUTO_INCREMENT PRIMARY KEY,
    Description VARCHAR(100),
    CatalogID INTEGER,
    FOREIGN KEY (CatalogID) REFERENCES Catalog(CatalogID)
);

CREATE TABLE Catalog (
    CatalogID INTEGER PRIMARY KEY,
    Title VARCHAR(255),
    Director VARCHAR(100),
    ReleaseDate YEAR
);

CREATE TABLE Movies (
    Duration TIME,
    CatalogID INTEGER,
    FOREIGN KEY (CatalogID) REFERENCES Catalog(CatalogID)
);

CREATE TABLE Shows (
    Seasons INTEGER,
    Episodes INTEGER,
    CatalogID INTEGER,
    FOREIGN KEY (CatalogID) REFERENCES Catalog(CatalogID)
);

CREATE TABLE User (
    UserID INTEGER AUTO_INCREMENT PRIMARY KEY,
    Password VARCHAR(100),
    Email VARCHAR(255)
);

CREATE TABLE Actors (
    ActorID INTEGER AUTO_INCREMENT PRIMARY KEY,
    FName VARCHAR(150),
    LName VARCHAR(150),
    CatalogID INTEGER,
    FOREIGN KEY (CatalogID) REFERENCES Catalog(CatalogID)
);

CREATE TABLE Accounts (
    AccountID INTEGER AUTO_INCREMENT PRIMARY KEY,
    AccountHolder VARCHAR(150),
    AccountNumber VARCHAR(255),
    CVV VARCHAR(4),
    SubscriptionType VARCHAR(50),
    SubscriptionStartDate DATE,
    SubscriptionEndDate DATE,
    UserID INTEGER,
    FOREIGN KEY (UserID) REFERENCES User(UserID)
```

```
);

CREATE TABLE User_Views (
    UserID INTEGER,
    CatalogID INTEGER,
    FOREIGN KEY (UserID) REFERENCES User(UserID),
    FOREIGN KEY (CatalogID) REFERENCES Catalog(CatalogID)
);
```