-- DVD RENTAL SYSTEM

--

--Create the Database

create database DVD;

--Use the DVD Database

use DVD;

--Create the actor table

CREATE TABLE actor (

actor\_id INT NOT NULL ,

first\_name VARCHAR(255) NOT NULL,

last\_name VARCHAR(255) NOT NULL,

PRIMARY KEY (actor\_id)

);

--Entries in actor table

INSERT INTO actor (actor\_id,first\_name, last\_name)

VALUES

(1,'John', 'Doe'),

(2,'Jane', 'Doe'),

(3,'Bill', 'Smith'),

(4,'Susan', 'Smith'),

(5,'Tom', 'Jones'),

(6,'Mary', 'Jones'),

(7,'Peter', 'Green'),

(8,'Sarah', 'Green'),

(9,'David', 'Brown'),

(10,'Emily', 'Brown');

-- Create the film table

CREATE TABLE film (

film\_id INT NOT NULL ,

title VARCHAR(255) NOT NULL,

release\_year NVARCHAR(4) NOT NULL,

length INT NOT NULL,

rating VARCHAR(255) NOT NULL,

PRIMARY KEY (film\_id)

);

--Entries in film table

INSERT INTO film (film\_id,title, release\_year, length, rating)

VALUES

(1,'The Shawshank Redemption', 1994, 142, 'PG-13'),

(2,'The Godfather', 1972, 175, 'R'),

(3,'The Dark Knight', 2008, 152, 'PG-13'),

(4,'The Lord of the Rings: The Return of the King', 2003, 201, 'PG-13'),

(5,'12 Angry Men', 1957, 120, 'Not Rated'),

(6,'The Godfather Part II', 1974, 202, 'R'),

(7,'Schindlers List', 1993, 195, 'R'),

(8,'The Silence of the Lambs', 1991, 118, 'R'),

(9,'Casablanca', 1942, 102, 'No Rating'),

(10,'Batman', 1994, 102, 'No Rating');

--Create the game table

CREATE TABLE game (

game\_id INT NOT NULL ,

game\_title VARCHAR(255) NOT NULL,

release\_date DATE NOT NULL,

genre VARCHAR(255) NOT NULL,

rating VARCHAR(10) NOT NULL,

price INT NOT NULL,

PRIMARY KEY (game\_id)

);

--Entries in game table

INSERT INTO game (game\_id, game\_title, release\_date, genre, rating, price)

VALUES

(1,'Super Mario Odyssey', '2017-10-27', 'Platformer', 'E', 59.99),

(2,'The Legend of Zelda: Breath of the Wild', '2017-03-03', 'Action-adventure', 'T', 59.99),

(3,'Pokémon Sword', '2019-11-15', 'Role-playing', 'E', 59.99),

(4,'Grand Theft Auto V', '2013-09-17', 'Action-adventure', 'M', 59.99),

(5,'Red Dead Redemption 2', '2018-10-26', 'Action-adventure', 'M', 59.99),

(6,'Minecraft', '2009-05-13', 'Sandbox', 'E', 19.99),

(7,'Fortnite', '2017-09-26', 'Battle royale', 'T', 70.00),

(8,'FIFA 23', '2023-09-27', 'Sports', 'E', 59.99),

(9,'Madden NFL 23', '2023-09-27', 'Sports', 'E', 59.99),

(10,'Pubg', '2022-04-27', 'Battle royale', 'T', 66.99);

--Create the software table

CREATE TABLE software (

software\_id INT NOT NULL,

software\_title NVARCHAR(255) NOT NULL,

release\_date DATE NOT NULL,

genre VARCHAR(255) NOT NULL,

rating VARCHAR(10) NOT NULL,

price INT NOT NULL,

PRIMARY KEY (software\_id)

);

--Entries in software table

INSERT INTO software (software\_id, software\_title, release\_date, genre, rating, price)

VALUES

(1,'Microsoft Office 2023', '2023-09-27', 'Productivity', 'E', 199.99),

(2,'Adobe Photoshop CC', '2018-10-18', 'Graphics', 'M', 99.99),

(3,'Logic Pro X', '2013-10-22', 'Audio','M', 199.99),

(4,'Visual Studio 2023', '2023-09-27', 'Programming','E', 199.99),

(5,'Autodesk Maya', '2023-03-08', '3D modeling', 'M', 299.99),

(6,'Adobe After Effects CC', '2018-10-18', 'Video editing', 'M', 99.99),

(7,'Premiere Pro CC', '2018-10-18', 'Video editing', 'M', 99.99),

(8,'Final Cut Pro X', '2017-10-30', 'Video editing', 'M', 299.99),

(9,'Audacity', '2022-06-15', 'Audio editing', 'E', 500.00),

(10,'Blender', '2023-03-03', '3D modeling', 'E', 650.00);

-- Create the film\_actor table

CREATE TABLE film\_actor (

film\_id INT NOT NULL,

actor\_id INT NOT NULL,

PRIMARY KEY (film\_id, actor\_id),

FOREIGN KEY (film\_id) REFERENCES film (film\_id),

FOREIGN KEY (actor\_id) REFERENCES actor (actor\_id)

);

--Entries in film\_actor table

INSERT INTO film\_actor (film\_id, actor\_id)

VALUES

(1, 1),

(1, 2),

(2, 3),

(2, 4),

(3, 5),

(3, 6),

(4, 7),

(4, 8),

(5, 9),

(5, 10);

-- Create the F\_category table

CREATE TABLE F\_category (

F\_category\_id INT NOT NULL ,

name VARCHAR(255) NOT NULL,

PRIMARY KEY (F\_category\_id)

);

--Entries in F\_catagory table

INSERT INTO F\_category (F\_category\_id,name)

VALUES

(1,'Action'),

(2,'Adventure'),

(3,'Comedy'),

(4,'Drama'),

(5,'Horror'),

(6,'Mystery'),

(7,'Romance'),

(8,'Sci-Fi'),

(9,'Thriller'),

(10,'Western');

-- Create the film\_category table

CREATE TABLE film\_category (

film\_id INT NOT NULL,

F\_category\_id INT NOT NULL,

PRIMARY KEY (film\_id, F\_category\_id),

FOREIGN KEY (film\_id) REFERENCES film (film\_id),

FOREIGN KEY (F\_category\_id) REFERENCES F\_category (F\_category\_id)

);

--Entries in film\_category table

INSERT INTO film\_category (film\_id, F\_category\_id)

VALUES

(1, 1),

(1, 4),

(2, 1),

(2, 5),

(3, 1),

(3, 4),

(4, 2),

(4, 5),

(5, 3),

(5, 6);

-- Create the G\_category table

CREATE TABLE G\_category (

G\_category\_id INT NOT NULL ,

name VARCHAR(255) NOT NULL,

PRIMARY KEY (G\_category\_id)

);

--Entries in G\_catagory table

INSERT INTO G\_category (G\_category\_id,name)

VALUES

(1,'Platformer'),

(2,'Action-adventure'),

(3,'Role-playing'),

(4,'Sandbox'),

(5,'Battle royale'),

(6,'Sports');

-- Create the game\_category table

CREATE TABLE game\_category (

game\_id INT NOT NULL,

G\_category\_id INT NOT NULL,

PRIMARY KEY (game\_id, G\_category\_id),

FOREIGN KEY (game\_id) REFERENCES game (game\_id),

FOREIGN KEY (G\_category\_id) REFERENCES G\_category (G\_category\_id)

);

--Entries in game\_category table

INSERT INTO game\_category (game\_id, G\_category\_id)

VALUES

(1, 1),

(1, 4),

(2, 1),

(2, 5),

(3, 1),

(3, 4),

(4, 2),

(4, 5),

(5, 3),

(5, 6);

-- Create the S\_category table

CREATE TABLE S\_category (

S\_category\_id INT NOT NULL ,

name VARCHAR(255) NOT NULL,

PRIMARY KEY (S\_category\_id)

);

--Entries in S\_catagory table

INSERT INTO S\_category (S\_category\_id,name)

VALUES

(1,'Productivity'),

(2,'Graphics'),

(3,'Audio'),

(4,'Programming'),

(5,'3D modeling'),

(6,'Video editing'),

(7,'Audio editing');

-- Create the software\_category table

CREATE TABLE software\_category (

software\_id INT NOT NULL,

S\_category\_id INT NOT NULL,

PRIMARY KEY (software\_id, S\_category\_id),

FOREIGN KEY (software\_id) REFERENCES software(software\_id),

FOREIGN KEY (S\_category\_id) REFERENCES S\_category (S\_category\_id)

);

--Entries in software\_category table

INSERT INTO software\_category (software\_id, S\_category\_id)

VALUES

(1, 1),

(1, 4),

(2, 1),

(2, 5),

(3, 1),

(3, 4),

(4, 2),

(4, 5),

(5, 3),

(5, 6);

-- Create the staff table

CREATE TABLE staff (

staff\_id INT NOT NULL,

first\_name VARCHAR(255) NOT NULL,

last\_name VARCHAR(255) NOT NULL,

job\_title VARCHAR(255) NOT NULL,

PRIMARY KEY (staff\_id)

);

--Entries in staff table

INSERT INTO staff (staff\_id,first\_name, last\_name, job\_title)

VALUES

(1,'John', 'Doe', 'Manager'),

(2,'Jane', 'Doe', 'Sales Associate'),

(3,'Bill', 'Smith', 'Customer Service Representative'),

(4,'Susan', 'Smith', 'Warehouse Clerk'),

(5,'Tom', 'Jones', 'Janitor'),

(6,'Mary', 'Jones', 'Security Guard'),

(7,'Peter', 'Green', 'Assistant Manager'),

(8,'Sarah', 'Green', 'Sales Manager'),

(9,'David', 'Brown', 'Customer Service Manager'),

(10,'Emily', 'Brown', 'Warehouse Manager');

-- Create the store table

CREATE TABLE store (

store\_id INT NOT NULL ,

manager\_staff\_id INT NOT NULL,

address VARCHAR(255) NOT NULL,

PRIMARY KEY (store\_id),

FOREIGN KEY (manager\_staff\_id) REFERENCES staff (staff\_id)

);

--Entries in store table

INSERT INTO store (store\_id, address, manager\_staff\_id)

VALUES

(1, '123 Main Street, Anytown, CA', 1),

(2, '456 Elm Street, Anytown, CA', 2),

(3, '789 Oak Street, Anytown, CA', 3),

(4, '101 Pine Street, Anytown, CA', 4),

(5, '1234 Broadway, Anytown, CA', 5),

(6, '54321 Main Street, Anytown, CA', 6),

(7, '98765 Elm Street, Anytown, CA', 7),

(8, '43210 Oak Street, Anytown, CA', 8),

(9, '76543 Pine Street, Anytown, CA', 9),

(10, '32100 Broadway, Anytown, CA', 10);

-- Create the inventory table

CREATE TABLE inventory (

inventory\_id INT NOT NULL,

film\_id INT NOT NULL,

game\_id INT NOT NULL,

software\_id INT NOT NULL,

store\_id INT NOT NULL,

F\_quantity INT NOT NULL,

G\_quantity INT NOT NULL,

S\_quantity INT NOT NULL,

PRIMARY KEY (inventory\_id),

FOREIGN KEY (film\_id) REFERENCES film (film\_id),

FOREIGN KEY (game\_id) REFERENCES game (game\_id),

FOREIGN KEY (software\_id) REFERENCES software (software\_id),

FOREIGN KEY (store\_id) REFERENCES store (store\_id)

);

--Entries in inventory table

INSERT INTO inventory (inventory\_id,film\_id,game\_id,software\_id, store\_id, F\_quantity,G\_quantity,S\_quantity)

VALUES

(1,1,1,1, 1,5,6,8),

(2,1,2,2, 2,3,4,6),

(3,1,3,3, 3,2,3,5),

(4,2,4,4, 1,4,5,7),

(5,2,5,5, 2,2,3,5),

(6,2,6,6, 3,1,2,5),

(7,3,7,7, 1,3,4,6),

(8,3,8,8, 2,2,3,7),

(9,3,9,9, 3,1,4,7),

(10,4,10,10,1,2,4,2);

-- Create the customer table

CREATE TABLE customer (

customer\_id INT NOT NULL,

first\_name VARCHAR(255) NOT NULL,

last\_name VARCHAR(255) NOT NULL,

email NVARCHAR(255) NOT NULL,

phone\_number NVARCHAR(255) NOT NULL,

address NVARCHAR(255) NOT NULL,

city NVARCHAR(255) NOT NULL,

state NVARCHAR(255) NOT NULL,

zip\_code NVARCHAR(255) NOT NULL,

PRIMARY KEY (customer\_id)

);

--Entries in customer table

INSERT INTO customer (customer\_id, first\_name, last\_name, email, phone\_number, address, city, state, zip\_code)

VALUES

(1, 'John', 'Doe', 'john.doe@email.com', '123-456-7890', '123 Main Street', 'Anytown', 'CA', '91234'),

(2, 'Jane', 'Doe', 'jane.doe@email.com', '555-555-5555', '456 Elm Street', 'Anytown', 'CA', '91234'),

(3, 'Bill', 'Smith', 'bill.smith@email.com', '987-654-3210', '789 Oak Street', 'Anytown', 'CA', '91234'),

(4, 'Susan', 'Smith', 'susan.smith@email.com', '321-098-7654', '101 Pine Street', 'Anytown', 'CA', '91234'),

(5, 'Tom', 'Jones', 'tom.jones@email.com', '210-987-6543', '1234 Broadway', 'Anytown', 'CA', '91234'),

(6, 'Mary', 'Jones', 'mary.jones@email.com', '109-876-5432', '54321 Main Street', 'Anytown', 'CA', '91234'),

(7, 'Peter', 'Green', 'peter.green@email.com', '987-654-3210', '98765 Elm Street', 'Anytown', 'CA', '91234'),

(8, 'Sarah', 'Green', 'sarah.green@email.com', '321-098-7654', '43210 Oak Street', 'Anytown', 'CA', '91234'),

(9, 'David', 'Brown', 'david.brown@email.com', '210-987-6543', '76543 Pine Street', 'Anytown', 'CA', '91234'),

(10, 'Emily', 'Brown', 'emily.brown@email.com', '109-876-5432', '32100 Broadway', 'Anytown', 'CA', '91234');

-- Create the rental table

CREATE TABLE rental (

rental\_id INT NOT NULL,

customer\_id INT NOT NULL,

film\_id INT NOT NULL,

game\_id INT NOT NULL,

software\_id INT NOT NULL,

store\_id INT NOT NULL,

rental\_date DATE NOT NULL,

return\_date DATE NOT NULL,

rental\_year NVARCHAR(4) NOT NULL,

PRIMARY KEY (rental\_id),

FOREIGN KEY (customer\_id) REFERENCES customer (customer\_id),

FOREIGN KEY (film\_id) REFERENCES film (film\_id),

FOREIGN KEY (game\_id) REFERENCES game (game\_id),

FOREIGN KEY (software\_id) REFERENCES software (software\_id),

FOREIGN KEY (store\_id) REFERENCES store (store\_id)

);

--Entries in rental table

INSERT INTO rental (rental\_id, customer\_id, film\_id,game\_id,software\_id,store\_id, rental\_date, return\_date,rental\_year)

VALUES

(1, 1, 1,1,1, 1, '01-01-2001','01-03-2001','2001'),

(2, 2, 2,2,2, 2, '01-01-2002','01-03-2002','2002'),

(3, 3, 3,3,3, 3, '01-01-2003','01-03-2003','2003'),

(4, 4, 4,4,4, 4, '01-01-2004','01-03-2004','2004'),

(5, 5, 5,5,5, 5, '01-01-2005','01-03-2005','2005'),

(6, 1, 1,6,6, 1, '01-01-2006','01-03-2006','2006'),

(7, 2, 2,7,7, 2, '01-01-2007','01-03-2007','2007'),

(8, 3, 3,8,8, 3, '01-01-2008','01-03-2008','2008'),

(9, 4, 4,9,9, 4, '01-01-2009','01-03-2009','2009'),

(10, 5, 5,10,10, 5,'01-01-2010','01-03-2010','2010');

-- Create the payment table

CREATE TABLE payment (

payment\_id INT NOT NULL ,

customer\_id INT NOT NULL,

rental\_id INT NOT NULL,

amount INT NOT NULL,

payment\_date NVARCHAR(255) NOT NULL,

PRIMARY KEY (payment\_id),

FOREIGN KEY (customer\_id) REFERENCES customer (customer\_id),

FOREIGN KEY (rental\_id) REFERENCES rental (rental\_id)

);

--Entries in payment table

INSERT INTO payment (payment\_id, customer\_id, rental\_id, amount, payment\_date)

VALUES

(1, 1, 1, 500, '01-01-2001'),

(2, 2, 2, 750, '01-01-2002'),

(3, 3, 3, 1000,'01-01-2003'),

(4, 4, 4, 1250,'01-01-2004'),

(5, 5, 5, 1500,'01-01-2005'),

(6, 1, 6, 500, '01-01-2006'),

(7, 2, 7, 750, '01-01-2007'),

(8, 3, 8, 1000,'01-01-2008'),

(9, 4, 9, 1250,'01-01-2009'),

(10, 5, 10, 1500,'01-01-2010');

--queries to access the data of the DVD rental system Database

-- Get all films

SELECT \* FROM film;

-- Get all customers

SELECT \* FROM customer;

-- Get all rentals

SELECT \* FROM rental;

-- Get all payments

SELECT \* FROM payment;

-- Get all films that are currently rented

SELECT film.\*

FROM film

JOIN rental

ON film.film\_id = rental.film\_id

WHERE rental.return\_date IS NULL;

-- Get all customers who have rented a film

SELECT customer.\*

FROM customer

JOIN rental

ON customer.customer\_id = rental.customer\_id;

-- Get the total amount of money that has been paid in rentals

SELECT SUM(amount) AS total\_amount

FROM payment;

-- Get the average number of days that a film is rented for

SELECT AVG(DATEDIFF(year,return\_date, rental\_date)) AS average\_rental\_length

FROM rental;