MOVIE RENTAL ANALYSIS REPORT

Medhavi Singh

(8963517)

Business Analytics

SQL and Data Analytics

INFO8076

Oct 16th 2024

# GOAL OF THE PROJECT

This project aims to provide insights into popular movies, rental trends, and consumer behavior through analyzing movie rental data. Through the analysis of film, actor, director, and customer rental data, we seek to address the following important questions:

* Which genres do people rent the most?
* Which performers and films are the most well-liked by viewers?
* What is the average frequency of movie rentals, and who are the most regular renters?

In addition, we wish to comprehend the rental system's total effectiveness and offer the company insightful information to aid in decision-making.

# DATA

A fictitious movie rental database including details on films, performers, directors, clients, and their rental histories makes up the data used for this project. The tables Movies, Actors, Directors, Customers, Rentals, MovieActors, and MovieDirectors were manually populated with data using SQL INSERT statements.

* For each of the important properties (e.g., CustomerID, MovieID, RentalDate, etc.), there were no entries that were incomplete or missing.
* Referential integrity was preserved by keeping relationships between tables, such as foreign key constraints.
* Every rental date and return date was legitimate and reasonable to show appropriate rental conduct.

Since the data was pre-structured and verified before entry, little cleanup was necessary.

## Tables Description

Below is a description of the relational schema, illustrating the relationships, primary keys, foreign keys, and how the tables interact.

**Tables in the Schema**

1. **Movies** (MovieID, Title, Genre, ReleaseYear, Rating, RuntimeMinutes)
   * **Primary Key**: MovieID
   * **Foreign Keys**: None
   * Stores movie information including title, genre, release year, and rating.
2. **Actors** (ActorID, FirstName, LastName, BirthYear)
   * **Primary Key**: ActorID
   * **Foreign Keys**: None
   * Stores actor details including their name and birth year.
3. **Directors** (DirectorID, FirstName, LastName, BirthYear)
   * **Primary Key**: DirectorID
   * **Foreign Keys**: None
   * Stores director information including their name and birth year.
4. **Customers** (CustomerID, FirstName, LastName, Email, Phone, Address)
   * **Primary Key**: CustomerID
   * **Foreign Keys**: None
   * Stores customer data including name, contact information, and address.
5. **Rentals** (RentalID, CustomerID, MovieID, RentalDate, ReturnDate)
   * **Primary Key**: RentalID
   * **Foreign Keys**:
     + CustomerID references Customers(CustomerID)
     + MovieID references Movies(MovieID)
   * This fact table records rental transactions, linking customers to the movies they have rented.
6. **MovieActors** (MovieActorID, MovieID, ActorID)
   * **Primary Key**: MovieActorID
   * **Foreign Keys**:
     + MovieID references Movies(MovieID)
     + ActorID references Actors(ActorID)
   * This fact table links actors to the movies they have appeared in.
7. **MovieDirectors** (MovieDirectorID, MovieID, DirectorID)
   * **Primary Key**: MovieDirectorID
   * **Foreign Keys**:
     + MovieID references Movies(MovieID)
     + DirectorID references Directors(DirectorID)
   * This fact table links directors to the movies they have done.

## The ERD

A screenshot of a computer

Description automatically generated

# ANALYSIS

## Database Creation

Dimension Tables

**Table 1**

--dimension table

create table Movies (

MovieID int primary key,

Title varchar(255),

Genre varchar(100),

ReleaseYear int,

Rating float,

RuntimeMinutes int );

**Table 2**

--dimension table

create table Actors (

ActorID int primary key,

FirstName varchar(50),

LastName varchar(50),

BirthYear int

);

**Table 3**

--dimension table

create table Directors (

DirectorID int primary key,

FirstName varchar(50),

LastName varchar(50),

BirthYear int

);

**Table 4**

--dimension table

create table Customers (

CustomerID int primary key,

FirstName varchar(50),

LastName varchar(50),

Email varchar(100),

Phone varchar(20),

Address varchar(200)

);

Fact Tables

**Table 5**

--fact table

create table Rentals (

RentalID int primary key,

CustomerID int,

MovieID int,

RentalDate date,

ReturnDate date,

foreign key (CustomerID) references Customers(CustomerID),

foreign key (MovieID) references Movies(MovieID)

);

**Table 6**

--fact table

Create table MovieActors (

MovieActorID int primary key,

MovieID int,

ActorID int,

foreign key (MovieID) references Movies(MovieID),

foreign key (ActorID) references Actors(ActorID)

);

**Table 7**

--fact table

create table MovieDirectors (

MovieDirectorID int primary key,

MovieID int,

DirectorID int,

foreign key (MovieID) references Movies(MovieID),

foreign key (DirectorID) references Directors(DirectorID));

## Populating the tables

--Populating the table

**Table 1 (Movies)**

INSERT INTO Movies (MovieID, Title, Genre, ReleaseYear, Rating, RuntimeMinutes)

VALUES

(1, 'Inception', 'Sci-Fi', 2010, 8.8, 148),

(2, 'The Dark Knight', 'Action', 2008, 9.0, 152),

(3, 'Interstellar', 'Sci-Fi', 2014, 8.6, 169),

(4, 'Avatar', 'Fantasy', 2009, 7.8, 162),

(5, 'Titanic', 'Romance', 1997, 7.8, 195),

(6, 'The Matrix', 'Sci-Fi', 1999, 8.7, 136),

(7, 'Gladiator', 'Action', 2000, 8.5, 155),

(8, 'Joker', 'Drama', 2019, 8.4, 122),

(9, 'The Lord of the Rings: The Fellowship of the Ring', 'Fantasy', 2001, 8.8, 178),

(10, 'The Godfather', 'Crime', 1972, 9.2, 175);

select \* from Movies;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| "movieid" | "title" | "genre" | "releaseyear" | "rating" | "runtimeminutes" |
| 1 | "Inception" | "Sci-Fi" | 2010 | 8.8 | 148 |
| 2 | "The Dark Knight" | "Action" | 2008 | 9 | 152 |
| 3 | "Interstellar" | "Sci-Fi" | 2014 | 8.6 | 169 |
| 4 | "Avatar" | "Fantasy" | 2009 | 7.8 | 162 |
| 5 | "Titanic" | "Romance" | 1997 | 7.8 | 195 |
| 6 | "The Matrix" | "Sci-Fi" | 1999 | 8.7 | 136 |
| 7 | "Gladiator" | "Action" | 2000 | 8.5 | 155 |
| 8 | "Joker" | "Drama" | 2019 | 8.4 | 122 |
| 9 | "The Lord of the Rings: The Fellowship of the Ring" | "Fantasy" | 2001 | 8.8 | 178 |
| 10 | "The Godfather" | "Crime" | 1972 | 9.2 | 175 |

**Table 2 (Actors)**

INSERT INTO Actors (ActorID, FirstName, LastName, BirthYear)

VALUES

(1, 'Leonardo', 'DiCaprio', 1974),

(2, 'Christian', 'Bale', 1974),

(3, 'Matthew', 'McConaughey', 1969),

(4, 'Sam', 'Worthington', 1976),

(5, 'Kate', 'Winslet', 1975),

(6, 'Keanu', 'Reeves', 1964),

(7, 'Russell', 'Crowe', 1964),

(8, 'Joaquin', 'Phoenix', 1974),

(9, 'Elijah', 'Wood', 1981),

(10, 'Al', 'Pacino', 1940);

Select \* from Actors;

|  |  |  |  |
| --- | --- | --- | --- |
| "actorid" | "firstname" | "lastname" | "birthyear" |
| 1 | "Leonardo" | "DiCaprio" | 1974 |
| 2 | "Christian" | "Bale" | 1974 |
| 3 | "Matthew" | "McConaughey" | 1969 |
| 4 | "Sam" | "Worthington" | 1976 |
| 5 | "Kate" | "Winslet" | 1975 |
| 6 | "Keanu" | "Reeves" | 1964 |
| 7 | "Russell" | "Crowe" | 1964 |
| 8 | "Joaquin" | "Phoenix" | 1974 |
| 9 | "Elijah" | "Wood" | 1981 |
| 10 | "Al" | "Pacino" | 1940 |

**Table 3 (Directors)**

INSERT INTO Directors (DirectorID, FirstName, LastName, BirthYear)

VALUES

(1, 'Christopher', 'Nolan', 1970),

(2, 'James', 'Cameron', 1954),

(3, 'Lana', 'Wachowski', 1965),

(4, 'Lilly', 'Wachowski', 1967),

(5, 'Ridley', 'Scott', 1937),

(6, 'Todd', 'Phillips', 1970),

(7, 'Peter', 'Jackson', 1961),

(8, 'Francis Ford', 'Coppola', 1939);

Select \* from Directors;

|  |  |  |  |
| --- | --- | --- | --- |
| "directorid" | "firstname" | "lastname" | "birthyear" |
| 1 | "Christopher" | "Nolan" | 1970 |
| 2 | "James" | "Cameron" | 1954 |
| 3 | "Lana" | "Wachowski" | 1965 |
| 4 | "Lilly" | "Wachowski" | 1967 |
| 5 | "Ridley" | "Scott" | 1937 |
| 6 | "Todd" | "Phillips" | 1970 |
| 7 | "Peter" | "Jackson" | 1961 |
| 8 | "Francis Ford" | "Coppola" | 1939 |

**Table 4 (Customers)**

INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone, Address)

VALUES

(1, 'John', 'Doe', 'john.doe@example.com', '123-456-7890', '123 Elm St'),

(2, 'Jane', 'Smith', 'jane.smith@example.com', '555-987-6543', '456 Oak St'),

(3, 'Michael', 'Johnson', 'michael.johnson@example.com', '234-567-8901', '789 Maple St'),

(4, 'Emily', 'Davis', 'emily.davis@example.com', '345-678-9012', '101 Pine St'),

(5, 'Chris', 'Brown', 'chris.brown@example.com', '456-789-0123', '202 Cedar St'),

(6, 'Olivia', 'Wilson', 'olivia.wilson@example.com', '567-890-1234', '303 Birch St'),

(7, 'James', 'Taylor', 'james.taylor@example.com', '678-901-2345', '404 Fir St'),

(8, 'Sophia', 'Moore', 'sophia.moore@example.com', '789-012-3456', '505 Spruce St'),

(9, 'David', 'Anderson', 'david.anderson@example.com', '890-123-4567', '606 Poplar St'),

(10, 'Emma', 'Thomas', 'emma.thomas@example.com', '901-234-5678', '707 Redwood St');

select \* from Customers;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| "customerid" | "firstname" | "lastname" | "email" | "phone" | "address" |
| 1 | "John" | "Doe" | "john.doe@example.com" | "123-456-7890" | "123 Elm St" |
| 2 | "Jane" | "Smith" | "jane.smith@example.com" | "555-987-6543" | "456 Oak St" |
| 3 | "Michael" | "Johnson" | "michael.johnson@example.com" | "234-567-8901" | "789 Maple St" |
| 4 | "Emily" | "Davis" | "emily.davis@example.com" | "345-678-9012" | "101 Pine St" |
| 5 | "Chris" | "Brown" | "chris.brown@example.com" | "456-789-0123" | "202 Cedar St" |
| 6 | "Olivia" | "Wilson" | "olivia.wilson@example.com" | "567-890-1234" | "303 Birch St" |
| 7 | "James" | "Taylor" | "james.taylor@example.com" | "678-901-2345" | "404 Fir St" |
| 8 | "Sophia" | "Moore" | "sophia.moore@example.com" | "789-012-3456" | "505 Spruce St" |
| 9 | "David" | "Anderson" | "david.anderson@example.com" | "890-123-4567" | "606 Poplar St" |
| 10 | "Emma" | "Thomas" | "emma.thomas@example.com" | "901-234-5678" | "707 Redwood St" |

**Table 5 (Rentals)**

INSERT INTO Rentals (RentalID, CustomerID, MovieID, RentalDate, ReturnDate)

VALUES

(1, 1, 1, '2024-10-01', '2024-10-05'),

(2, 2, 2, '2024-10-02', '2024-10-06'),

(3, 3, 3, '2024-10-03', '2024-10-07'),

(4, 4, 4, '2024-10-04', '2024-10-08'),

(5, 5, 5, '2024-10-05', '2024-10-09'),

(6, 6, 6, '2024-10-06', '2024-10-10'),

(7, 7, 7, '2024-10-07', '2024-10-11'),

(8, 8, 8, '2024-10-08', '2024-10-12'),

(9, 9, 9, '2024-10-09', '2024-10-13'),

(10, 10, 10, '2024-10-10', '2024-10-14'),

(11, 1, 2, '2024-10-11', '2024-10-15'),

(12, 2, 3, '2024-10-12', '2024-10-16'),

(13, 3, 4, '2024-10-13', '2024-10-17'),

(14, 4, 5, '2024-10-14', '2024-10-18'),

(15, 5, 6, '2024-10-15', '2024-10-19'),

(16, 6, 7, '2024-10-16', '2024-10-20'),

(17, 7, 8, '2024-10-17', '2024-10-21'),

(18, 8, 9, '2024-10-18', '2024-10-22'),

(19, 9, 10, '2024-10-19', '2024-10-23'),

(20, 10, 1, '2024-10-20', '2024-10-24'),

(21, 1, 3, '2024-10-21', '2024-10-25'),

(22, 2, 4, '2024-10-22', '2024-10-26'),

(23, 3, 5, '2024-10-23', '2024-10-27'),

(24, 4, 6, '2024-10-24', '2024-10-28'),

(25, 5, 7, '2024-10-25', '2024-10-29'),

(26, 6, 8, '2024-10-26', '2024-10-30'),

(27, 7, 9, '2024-10-27', '2024-10-31'),

(28, 8, 10, '2024-10-28', '2024-11-01'),

(29, 9, 1, '2024-10-29', '2024-11-02'),

(30, 10, 2, '2024-10-30', '2024-11-03');

Select \* from Rentals;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| "rentalid" | "customerid" | "movieid" | "rentaldate" | "returndate" |
| 1 | 1 | 1 | "2024-10-01" | "2024-10-05" |
| 2 | 2 | 2 | "2024-10-02" | "2024-10-06" |
| 3 | 3 | 3 | "2024-10-03" | "2024-10-07" |
| 4 | 4 | 4 | "2024-10-04" | "2024-10-08" |
| 5 | 5 | 5 | "2024-10-05" | "2024-10-09" |
| 6 | 6 | 6 | "2024-10-06" | "2024-10-10" |
| 7 | 7 | 7 | "2024-10-07" | "2024-10-11" |
| 8 | 8 | 8 | "2024-10-08" | "2024-10-12" |
| 9 | 9 | 9 | "2024-10-09" | "2024-10-13" |
| 10 | 10 | 10 | "2024-10-10" | "2024-10-14" |
| 11 | 1 | 2 | "2024-10-11" | "2024-10-15" |
| 12 | 2 | 3 | "2024-10-12" | "2024-10-16" |
| 13 | 3 | 4 | "2024-10-13" | "2024-10-17" |
| 14 | 4 | 5 | "2024-10-14" | "2024-10-18" |
| 15 | 5 | 6 | "2024-10-15" | "2024-10-19" |
| 16 | 6 | 7 | "2024-10-16" | "2024-10-20" |
| 17 | 7 | 8 | "2024-10-17" | "2024-10-21" |
| 18 | 8 | 9 | "2024-10-18" | "2024-10-22" |
| 19 | 9 | 10 | "2024-10-19" | "2024-10-23" |
| 20 | 10 | 1 | "2024-10-20" | "2024-10-24" |
| 21 | 1 | 3 | "2024-10-21" | "2024-10-25" |
| 22 | 2 | 4 | "2024-10-22" | "2024-10-26" |
| 23 | 3 | 5 | "2024-10-23" | "2024-10-27" |
| 24 | 4 | 6 | "2024-10-24" | "2024-10-28" |
| 25 | 5 | 7 | "2024-10-25" | "2024-10-29" |
| 26 | 6 | 8 | "2024-10-26" | "2024-10-30" |
| 27 | 7 | 9 | "2024-10-27" | "2024-10-31" |
| 28 | 8 | 10 | "2024-10-28" | "2024-11-01" |
| 29 | 9 | 1 | "2024-10-29" | "2024-11-02" |
| 30 | 10 | 2 | "2024-10-30" | "2024-11-03" |

**Table 6 (MovieActors)**

INSERT INTO MovieActors (MovieActorID, MovieID, ActorID)

VALUES

(1, 1, 1), -- Leonardo DiCaprio in Inception

(2, 2, 2), -- Christian Bale in The Dark Knight

(3, 3, 3), -- Matthew McConaughey in Interstellar

(4, 4, 4), -- Sam Worthington in Avatar

(5, 5, 1), -- Leonardo DiCaprio in Titanic

(6, 5, 5), -- Kate Winslet in Titanic

(7, 6, 6), -- Keanu Reeves in The Matrix

(8, 7, 7), -- Russell Crowe in Gladiator

(9, 8, 8), -- Joaquin Phoenix in Joker

(10, 9, 9), -- Elijah Wood in LOTR: Fellowship of the Ring

(11, 10, 10), -- Al Pacino in The Godfather

(12, 1, 1), -- Leonardo DiCaprio in Inception

(13, 2, 2), -- Christian Bale in The Dark Knight

(14, 3, 3), -- Matthew McConaughey in Interstellar

(15, 4, 4), -- Sam Worthington in Avatar

(16, 5, 1), -- Leonardo DiCaprio in Titanic

(17, 5, 5), -- Kate Winslet in Titanic

(18, 6, 6), -- Keanu Reeves in The Matrix

(19, 7, 7), -- Russell Crowe in Gladiator

(20, 8, 8), -- Joaquin Phoenix in Joker

(21, 9, 9), -- Elijah Wood in LOTR: Fellowship of the Ring

(22, 10, 10), -- Al Pacino in The Godfather

(23, 1, 1), -- Leonardo DiCaprio in Inception

(24, 2, 2), -- Christian Bale in The Dark Knight

(25, 3, 3), -- Matthew McConaughey in Interstellar

(26, 4, 4), -- Sam Worthington in Avatar

(27, 5, 1), -- Leonardo DiCaprio in Titanic

(28, 5, 5), -- Kate Winslet in Titanic

(29, 6, 6), -- Keanu Reeves in The Matrix

(30, 7, 7); -- Russell Crowe in Gladiator

Select \* from MovieActors;

|  |  |  |
| --- | --- | --- |
| "movieactorid" | "movieid" | "actorid" |
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 3 | 3 |
| 4 | 4 | 4 |
| 5 | 5 | 1 |
| 6 | 5 | 5 |
| 7 | 6 | 6 |
| 8 | 7 | 7 |
| 9 | 8 | 8 |
| 10 | 9 | 9 |
| 11 | 10 | 10 |
| 12 | 1 | 1 |
| 13 | 2 | 2 |
| 14 | 3 | 3 |
| 15 | 4 | 4 |
| 16 | 5 | 1 |
| 17 | 5 | 5 |
| 18 | 6 | 6 |
| 19 | 7 | 7 |
| 20 | 8 | 8 |
| 21 | 9 | 9 |
| 22 | 10 | 10 |
| 23 | 1 | 1 |
| 24 | 2 | 2 |
| 25 | 3 | 3 |
| 26 | 4 | 4 |
| 27 | 5 | 1 |
| 28 | 5 | 5 |
| 29 | 6 | 6 |
| 30 | 7 | 7 |

**Table 7 (MovieDirectors)**

INSERT INTO MovieDirectors (MovieDirectorID, MovieID, DirectorID)

VALUES

(1, 1, 1), -- Christopher Nolan directed Inception

(2, 2, 1), -- Christopher Nolan directed The Dark Knight

(3, 3, 1), -- Christopher Nolan directed Interstellar

(4, 4, 2), -- James Cameron directed Avatar

(5, 5, 2), -- James Cameron directed Titanic

(6, 6, 3), -- Wachowski directed The Matrix

(7, 6, 4), -- Wachowski directed The Matrix

(8, 7, 5), -- Ridley Scott directed Gladiator

(9, 8, 6), -- Todd Phillips directed Joker

(10, 9, 7), -- Peter Jackson directed LOTR: Fellowship of the Ring

(11, 10, 8), -- Francis Ford Coppola directed The Godfather

(12, 1, 1), -- Christopher Nolan directed Inception

(13, 2, 1), -- Christopher Nolan directed The Dark Knight

(14, 3, 1), -- Christopher Nolan directed Interstellar

(15, 4, 2), -- James Cameron directed Avatar

(16, 5, 2), -- James Cameron directed Titanic

(17, 6, 3), -- Wachowski directed The Matrix

(18, 6, 4), -- Wachowski directed The Matrix

(19, 7, 5), -- Ridley Scott directed Gladiator

(20, 8, 6), -- Todd Phillips directed Joker

(21, 9, 7), -- Peter Jackson directed LOTR: Fellowship of the Ring

(22, 10, 8), -- Francis Ford Coppola directed The Godfather

(23, 1, 1), -- Christopher Nolan directed Inception

(24, 2, 1), -- Christopher Nolan directed The Dark Knight

(25, 3, 1), -- Christopher Nolan directed Interstellar

(26, 4, 2), -- James Cameron directed Avatar

(27, 5, 2), -- James Cameron directed Titanic

(28, 6, 3), -- Wachowski directed The Matrix

(29, 6, 4), -- Wachowski directed The Matrix

(30, 7, 5); -- Ridley Scott directed Gladiator

Select \* from MovieDirectors;

|  |  |  |
| --- | --- | --- |
| "moviedirectorid" | "movieid" | "directorid" |
| 1 | 1 | 1 |
| 2 | 2 | 1 |
| 3 | 3 | 1 |
| 4 | 4 | 2 |
| 5 | 5 | 2 |
| 6 | 6 | 3 |
| 7 | 6 | 4 |
| 8 | 7 | 5 |
| 9 | 8 | 6 |
| 10 | 9 | 7 |
| 11 | 10 | 8 |
| 12 | 1 | 1 |
| 13 | 2 | 1 |
| 14 | 3 | 1 |
| 15 | 4 | 2 |
| 16 | 5 | 2 |
| 17 | 6 | 3 |
| 18 | 6 | 4 |
| 19 | 7 | 5 |
| 20 | 8 | 6 |
| 21 | 9 | 7 |
| 22 | 10 | 8 |
| 23 | 1 | 1 |
| 24 | 2 | 1 |
| 25 | 3 | 1 |
| 26 | 4 | 2 |
| 27 | 5 | 2 |
| 28 | 6 | 3 |
| 29 | 6 | 4 |
| 30 | 7 | 5 |

## Creating Views

**View 1: CustomerRentalHistory**

CREATE VIEW CustomerRentalHistory AS

SELECT C.FirstName,

C.LastName,

M.Title,

R.RentalDate,

R.ReturnDate

FROM Customers C

JOIN Rentals R ON C.CustomerID = R.CustomerID

JOIN Movies M ON R.MovieID = M.MovieID;

This view helps analyze customer rental patterns, showing what movies were rented and when. It reveals customer preferences and behaviors, such as frequent renters or late returns.

Select from CustomerRentalHistory;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| "firstname" | "lastname" | "title" | "rentaldate" | "returndate" |
| "John" | "Doe" | "Inception" | "2024-10-01" | "2024-10-05" |
| "Jane" | "Smith" | "The Dark Knight" | "2024-10-02" | "2024-10-06" |
| "Michael" | "Johnson" | "Interstellar" | "2024-10-03" | "2024-10-07" |
| "Emily" | "Davis" | "Avatar" | "2024-10-04" | "2024-10-08" |
| "Chris" | "Brown" | "Titanic" | "2024-10-05" | "2024-10-09" |
| "Olivia" | "Wilson" | "The Matrix" | "2024-10-06" | "2024-10-10" |
| "James" | "Taylor" | "Gladiator" | "2024-10-07" | "2024-10-11" |
| "Sophia" | "Moore" | "Joker" | "2024-10-08" | "2024-10-12" |
| "David" | "Anderson" | "The Lord of the Rings: The Fellowship of the Ring" | "2024-10-09" | "2024-10-13" |
| "Emma" | "Thomas" | "The Godfather" | "2024-10-10" | "2024-10-14" |
| "John" | "Doe" | "The Dark Knight" | "2024-10-11" | "2024-10-15" |
| "Jane" | "Smith" | "Interstellar" | "2024-10-12" | "2024-10-16" |
| "Michael" | "Johnson" | "Avatar" | "2024-10-13" | "2024-10-17" |
| "Emily" | "Davis" | "Titanic" | "2024-10-14" | "2024-10-18" |
| "Chris" | "Brown" | "The Matrix" | "2024-10-15" | "2024-10-19" |
| "Olivia" | "Wilson" | "Gladiator" | "2024-10-16" | "2024-10-20" |
| "James" | "Taylor" | "Joker" | "2024-10-17" | "2024-10-21" |
| "Sophia" | "Moore" | "The Lord of the Rings: The Fellowship of the Ring" | "2024-10-18" | "2024-10-22" |
| "David" | "Anderson" | "The Godfather" | "2024-10-19" | "2024-10-23" |
| "Emma" | "Thomas" | "Inception" | "2024-10-20" | "2024-10-24" |
| "John" | "Doe" | "Interstellar" | "2024-10-21" | "2024-10-25" |
| "Jane" | "Smith" | "Avatar" | "2024-10-22" | "2024-10-26" |
| "Michael" | "Johnson" | "Titanic" | "2024-10-23" | "2024-10-27" |
| "Emily" | "Davis" | "The Matrix" | "2024-10-24" | "2024-10-28" |
| "Chris" | "Brown" | "Gladiator" | "2024-10-25" | "2024-10-29" |
| "Olivia" | "Wilson" | "Joker" | "2024-10-26" | "2024-10-30" |
| "James" | "Taylor" | "The Lord of the Rings: The Fellowship of the Ring" | "2024-10-27" | "2024-10-31" |
| "Sophia" | "Moore" | "The Godfather" | "2024-10-28" | "2024-11-01" |
| "David" | "Anderson" | "Inception" | "2024-10-29" | "2024-11-02" |
| "Emma" | "Thomas" | "The Dark Knight" | "2024-10-30" | "2024-11-03" |

**View 2: ActorMovies**

CREATE VIEW ActorMovies AS

SELECT A.FirstName,

A.LastName,

M.Title,

M.ReleaseYear

FROM Actors A

JOIN MovieActors MA ON A.ActorID = MA.ActorID

JOIN Movies M ON MA.MovieID = M.MovieID;

This view provides a list of movies that actors have appeared in, along with release years. It allows analysis of actor popularity, trends, and actor-based recommendations.

Select \* from ActorMovies

|  |  |  |  |
| --- | --- | --- | --- |
| "firstname" | "lastname" | "title" | "releaseyear" |
| "Leonardo" | "DiCaprio" | "Inception" | 2010 |
| "Christian" | "Bale" | "The Dark Knight" | 2008 |
| "Matthew" | "McConaughey" | "Interstellar" | 2014 |
| "Sam" | "Worthington" | "Avatar" | 2009 |
| "Leonardo" | "DiCaprio" | "Titanic" | 1997 |
| "Kate" | "Winslet" | "Titanic" | 1997 |
| "Keanu" | "Reeves" | "The Matrix" | 1999 |
| "Russell" | "Crowe" | "Gladiator" | 2000 |
| "Joaquin" | "Phoenix" | "Joker" | 2019 |
| "Elijah" | "Wood" | "The Lord of the Rings: The Fellowship of the Ring" | 2001 |
| "Al" | "Pacino" | "The Godfather" | 1972 |
| "Leonardo" | "DiCaprio" | "Inception" | 2010 |
| "Christian" | "Bale" | "The Dark Knight" | 2008 |
| "Matthew" | "McConaughey" | "Interstellar" | 2014 |
| "Sam" | "Worthington" | "Avatar" | 2009 |
| "Leonardo" | "DiCaprio" | "Titanic" | 1997 |
| "Kate" | "Winslet" | "Titanic" | 1997 |
| "Keanu" | "Reeves" | "The Matrix" | 1999 |
| "Russell" | "Crowe" | "Gladiator" | 2000 |
| "Joaquin" | "Phoenix" | "Joker" | 2019 |
| "Elijah" | "Wood" | "The Lord of the Rings: The Fellowship of the Ring" | 2001 |
| "Al" | "Pacino" | "The Godfather" | 1972 |
| "Leonardo" | "DiCaprio" | "Inception" | 2010 |
| "Christian" | "Bale" | "The Dark Knight" | 2008 |
| "Matthew" | "McConaughey" | "Interstellar" | 2014 |
| "Sam" | "Worthington" | "Avatar" | 2009 |
| "Leonardo" | "DiCaprio" | "Titanic" | 1997 |
| "Kate" | "Winslet" | "Titanic" | 1997 |
| "Keanu" | "Reeves" | "The Matrix" | 1999 |
| "Russell" | "Crowe" | "Gladiator" | 2000 |

**View 3: MovieDirectorsList**

CREATE VIEW MovieDirectorsList AS

SELECT

M.Title,

D.FirstName AS DirectorFirstName,

D.LastName AS DirectorLastName

FROM Movies M

JOIN MovieDirectors MD ON M.MovieID = MD.MovieID

JOIN Directors D ON MD.DirectorID = D.DirectorID;

This view links movies with their directors, showing the creative influence behind each film. It helps identify popular directors and analyze customer preferences by director.

select \* from MovieDirectorsList;

|  |  |  |
| --- | --- | --- |
| "title" | "directorfirstname" | "directorlastname" |
| "Inception" | "Christopher" | "Nolan" |
| "The Dark Knight" | "Christopher" | "Nolan" |
| "Interstellar" | "Christopher" | "Nolan" |
| "Avatar" | "James" | "Cameron" |
| "Titanic" | "James" | "Cameron" |
| "The Matrix" | "Lana" | "Wachowski" |
| "The Matrix" | "Lilly" | "Wachowski" |
| "Gladiator" | "Ridley" | "Scott" |
| "Joker" | "Todd" | "Phillips" |
| "The Lord of the Rings: The Fellowship of the Ring" | "Peter" | "Jackson" |
| "The Godfather" | "Francis Ford" | "Coppola" |
| "Inception" | "Christopher" | "Nolan" |
| "The Dark Knight" | "Christopher" | "Nolan" |
| "Interstellar" | "Christopher" | "Nolan" |
| "Avatar" | "James" | "Cameron" |
| "Titanic" | "James" | "Cameron" |
| "The Matrix" | "Lana" | "Wachowski" |
| "The Matrix" | "Lilly" | "Wachowski" |
| "Gladiator" | "Ridley" | "Scott" |
| "Joker" | "Todd" | "Phillips" |
| "The Lord of the Rings: The Fellowship of the Ring" | "Peter" | "Jackson" |
| "The Godfather" | "Francis Ford" | "Coppola" |
| "Inception" | "Christopher" | "Nolan" |
| "The Dark Knight" | "Christopher" | "Nolan" |
| "Interstellar" | "Christopher" | "Nolan" |
| "Avatar" | "James" | "Cameron" |
| "Titanic" | "James" | "Cameron" |
| "The Matrix" | "Lana" | "Wachowski" |
| "The Matrix" | "Lilly" | "Wachowski" |
| "Gladiator" | "Ridley" | "Scott" |

## Functions Applied

1. **Join Query**

select C.FirstName, C.LastName, M.Title, R.RentalDate, R.ReturnDate

from Rentals R

join Customers C on R.CustomerID = C.CustomerID

join Movies M on R.MovieID = M.MovieID;

This query combines data from multiple tables to show the full rental history for each customer, including the movies they rented and the corresponding rental and return dates. This is useful for understanding rental patterns, customer behavior, and overall engagement with the movie collection. For example, it can reveal which customers rent frequently or if certain movies are rented more during specific periods.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| "firstname" | "lastname" | "title" | "rentaldate" | "returndate" |
| "John" | "Doe" | "Inception" | "2024-10-01" | "2024-10-05" |
| "Jane" | "Smith" | "The Dark Knight" | "2024-10-02" | "2024-10-06" |
| "Michael" | "Johnson" | "Interstellar" | "2024-10-03" | "2024-10-07" |
| "Emily" | "Davis" | "Avatar" | "2024-10-04" | "2024-10-08" |
| "Chris" | "Brown" | "Titanic" | "2024-10-05" | "2024-10-09" |
| "Olivia" | "Wilson" | "The Matrix" | "2024-10-06" | "2024-10-10" |
| "James" | "Taylor" | "Gladiator" | "2024-10-07" | "2024-10-11" |
| "Sophia" | "Moore" | "Joker" | "2024-10-08" | "2024-10-12" |
| "David" | "Anderson" | "The Lord of the Rings: The Fellowship of the Ring" | "2024-10-09" | "2024-10-13" |
| "Emma" | "Thomas" | "The Godfather" | "2024-10-10" | "2024-10-14" |
| "John" | "Doe" | "The Dark Knight" | "2024-10-11" | "2024-10-15" |
| "Jane" | "Smith" | "Interstellar" | "2024-10-12" | "2024-10-16" |
| "Michael" | "Johnson" | "Avatar" | "2024-10-13" | "2024-10-17" |
| "Emily" | "Davis" | "Titanic" | "2024-10-14" | "2024-10-18" |
| "Chris" | "Brown" | "The Matrix" | "2024-10-15" | "2024-10-19" |
| "Olivia" | "Wilson" | "Gladiator" | "2024-10-16" | "2024-10-20" |
| "James" | "Taylor" | "Joker" | "2024-10-17" | "2024-10-21" |
| "Sophia" | "Moore" | "The Lord of the Rings: The Fellowship of the Ring" | "2024-10-18" | "2024-10-22" |
| "David" | "Anderson" | "The Godfather" | "2024-10-19" | "2024-10-23" |
| "Emma" | "Thomas" | "Inception" | "2024-10-20" | "2024-10-24" |
| "John" | "Doe" | "Interstellar" | "2024-10-21" | "2024-10-25" |
| "Jane" | "Smith" | "Avatar" | "2024-10-22" | "2024-10-26" |
| "Michael" | "Johnson" | "Titanic" | "2024-10-23" | "2024-10-27" |
| "Emily" | "Davis" | "The Matrix" | "2024-10-24" | "2024-10-28" |
| "Chris" | "Brown" | "Gladiator" | "2024-10-25" | "2024-10-29" |
| "Olivia" | "Wilson" | "Joker" | "2024-10-26" | "2024-10-30" |
| "James" | "Taylor" | "The Lord of the Rings: The Fellowship of the Ring" | "2024-10-27" | "2024-10-31" |
| "Sophia" | "Moore" | "The Godfather" | "2024-10-28" | "2024-11-01" |
| "David" | "Anderson" | "Inception" | "2024-10-29" | "2024-11-02" |
| "Emma" | "Thomas" | "The Dark Knight" | "2024-10-30" | "2024-11-03" |

1. **Where Filter (Sci-Fi Genre)**

select \* from Movies where Genre = 'Sci-Fi';

Filtering movies by the Sci-Fi genre helps identify all films that belong to this category. This is valuable for assessing the performance and popularity of Sci-Fi movies. By understanding which genres are most rented or highest-rated, you can tailor movie recommendations or identify gaps in genre representation within your catalog.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| "movieid" | "title" | "genre" | "releaseyear" | "rating" | "runtimeminutes" |
| 1 | "Inception" | "Sci-Fi" | 2010 | 8.8 | 148 |
| 3 | "Interstellar" | "Sci-Fi" | 2014 | 8.6 | 169 |
| 6 | "The Matrix" | "Sci-Fi" | 1999 | 8.7 | 136 |

1. **Listing Movies Rented by 'John Doe'**

select M.Title

from Movies M

where M.MovieID in (

select R.MovieID from Rentals R

join Customers C on R.CustomerID = C.CustomerID

where C.FirstName = 'John' and C.LastName = 'Doe'

);

This query pinpoints the exact movies rented by John Doe, giving insights into his personal movie preferences and viewing habits. Such customer-specific data can be used for targeted marketing, such as suggesting similar films or offering promotions based on his tastes, increasing customer satisfaction and retention.

|  |
| --- |
| **"title"** |
| "Interstellar" |
| "Inception" |
| "The Dark Knight" |

1. **Counting Total Rentals Per Customer**

select C.FirstName, C.LastName, count(R.RentalID) as TotalRentals

from Customers C

join Rentals R on C.CustomerID = R.CustomerID

group by C.FirstName, C.LastName;

This query counts how many movies each customer has rented, highlighting the most active customers. It helps identify loyal customers or those who frequently rent, making them ideal candidates for loyalty programs or special offers. Conversely, it can also reveal customers who rent infrequently, presenting opportunities for re-engagement.

|  |  |  |
| --- | --- | --- |
| "firstname" | "lastname" | "totalrentals" |
| "Sophia" | "Moore" | 3 |
| "James" | "Taylor" | 3 |
| "John" | "Doe" | 3 |
| "Chris" | "Brown" | 3 |
| "Olivia" | "Wilson" | 3 |
| "Emma" | "Thomas" | 3 |
| "David" | "Anderson" | 3 |
| "Jane" | "Smith" | 3 |
| "Emily" | "Davis" | 3 |
| "Michael" | "Johnson" | 3 |

1. **Movies with More Than One Rental**

with RentalCount as (

select MovieID, count(\*) as RentalsCount

from Rentals

group by MovieID

)

select M.Title, RC.RentalsCount

from Movies M

join RentalCount RC on M.MovieID = RC.MovieID

where RC.RentalsCount > 1;

Identifying movies rented multiple times helps determine which titles are the most popular or frequently re-watched. This is useful for stock management (for physical rentals) or for understanding trends in movie preferences. It can also guide future content acquisitions or highlight movies that could be featured in promotions.

|  |  |
| --- | --- |
| "title" | "rentalscount" |
| "Inception" | 3 |
| "The Dark Knight" | 3 |
| "Interstellar" | 3 |
| "Avatar" | 3 |
| "Titanic" | 3 |
| "The Matrix" | 3 |
| "Gladiator" | 3 |
| "Joker" | 3 |
| "The Lord of the Rings: The Fellowship of the Ring" | 3 |
| "The Godfather" | 3 |

1. **Range Movies Based on Rating**

select M.Title, M.Rating,

(select count(\*)

from Movies

where Rating > M.Rating) + 1 as rank

from Movies M;

Ranking movies by rating allows for a comparison of films based on viewer feedback or critical reviews. This insight is valuable for curating lists of top-rated movies for new customers or promoting the highest-rated films. It helps in understanding the perceived quality of the movie catalog and in identifying top performers.

|  |  |  |
| --- | --- | --- |
| "title" | "rating" | "rank" |
| "Inception" | 8.8 | 3 |
| "The Dark Knight" | 9 | 2 |
| "Interstellar" | 8.6 | 6 |
| "Avatar" | 7.8 | 9 |
| "Titanic" | 7.8 | 9 |
| "The Matrix" | 8.7 | 5 |
| "Gladiator" | 8.5 | 7 |
| "Joker" | 8.4 | 8 |
| "The Lord of the Rings: The Fellowship of the Ring" | 8.8 | 3 |
| "The Godfather" | 9.2 | 1 |

1. **Genres with More Than Two Movies**

select Genre, count(\*) as MovieCount

from Movies

group by Genre

having count(\*) > 2;

This query identifies genres with significant representation in the database (more than two movies). It helps analyze genre diversity and determine whether any genres are underrepresented. This insight can guide future decisions on expanding the movie catalog to balance genre offerings, based on customer demand and rental history.

|  |  |
| --- | --- |
| "genre" | "moviecount" |
| "Sci-Fi" | 3 |

1. **Customer with No Rental**

select \*

from Customers

where CustomerID not in (select distinct CustomerID from Rentals);

This query identifies customers who have not rented any movies. These customers may represent lost opportunities, and targeted marketing strategies such as email campaigns, discounts, or personalized movie recommendations could be used to re-engage them. Understanding why these customers are inactive can lead to improvements in customer retention strategies.

Empty as all the customers in the data rented at least one movie.

1. **Average Rating of Movies**

SELECT AVG(Rating) AS AverageRating

FROM Movies;

The average rating of movies helps assess the overall quality of the movie collection. A high average rating suggests that most movies are well-regarded, while a low rating could indicate a need to improve the selection of films. This insight helps in maintaining the quality of content offered to customers and ensuring customer satisfaction.

|  |
| --- |
| "averagerating" |
| 8.559999999999999 |

1. **Most frequently rented genre by each customer**

WITH GenreCount AS (

SELECT C.CustomerID, C.FirstName, C.LastName, M.Genre, COUNT(M.MovieID) AS GenreRentalCount

FROM Rentals R

JOIN Customers C ON R.CustomerID = C.CustomerID

JOIN Movies M ON R.MovieID = M.MovieID

GROUP BY C.CustomerID, C.FirstName, C.LastName, M.Genre

),

MaxGenreCount AS (

SELECT CustomerID, MAX(GenreRentalCount) AS MaxCount

FROM GenreCount

GROUP BY CustomerID

)

SELECT GC.FirstName, GC.LastName, GC.Genre, GC.GenreRentalCount

FROM GenreCount GC

JOIN MaxGenreCount MC ON GC.CustomerID = MC.CustomerID AND GC.GenreRentalCount = MC.MaxCount;

This query shows the most rented genre for each customer, revealing their preferred types of movies. This can be used to enhance customer experience by providing tailored recommendations or personalized promotions based on their favorite genre. It's a powerful way to increase customer engagement and satisfaction by aligning your offerings with their specific tastes.

|  |  |  |  |
| --- | --- | --- | --- |
| "firstname" | "lastname" | "genre" | "genrerentalcount" |
| "David" | "Anderson" | "Sci-Fi" | 1 |
| "Emma" | "Thomas" | "Crime" | 1 |
| "Chris" | "Brown" | "Romance" | 1 |
| "Chris" | "Brown" | "Action" | 1 |
| "James" | "Taylor" | "Drama" | 1 |
| "James" | "Taylor" | "Fantasy" | 1 |
| "Olivia" | "Wilson" | "Sci-Fi" | 1 |
| "David" | "Anderson" | "Crime" | 1 |
| "Sophia" | "Moore" | "Crime" | 1 |
| "James" | "Taylor" | "Action" | 1 |
| "Chris" | "Brown" | "Sci-Fi" | 1 |
| "Emma" | "Thomas" | "Sci-Fi" | 1 |
| "Michael" | "Johnson" | "Sci-Fi" | 1 |
| "Emma" | "Thomas" | "Action" | 1 |
| "David" | "Anderson" | "Fantasy" | 1 |
| "John" | "Doe" | "Sci-Fi" | 2 |
| "Emily" | "Davis" | "Fantasy" | 1 |
| "Emily" | "Davis" | "Romance" | 1 |
| "Jane" | "Smith" | "Action" | 1 |
| "Jane" | "Smith" | "Sci-Fi" | 1 |
| "Sophia" | "Moore" | "Drama" | 1 |
| "Michael" | "Johnson" | "Fantasy" | 1 |
| "Michael" | "Johnson" | "Romance" | 1 |
| "Olivia" | "Wilson" | "Action" | 1 |
| "Sophia" | "Moore" | "Fantasy" | 1 |
| "Olivia" | "Wilson" | "Drama" | 1 |
| "Emily" | "Davis" | "Sci-Fi" | 1 |
| "Jane" | "Smith" | "Fantasy" | 1 |

# CONCLUSION

The analysis of the movie rental database revealed several key insights that can be used to improve business operations and customer engagement. Firstly, popular genres such as Sci-Fi and Action tend to dominate rentals, with movies like "Inception" and "The Dark Knight" being frequently rented, reflecting high customer demand. Customer behavior analysis identified both loyal and inactive customers, enabling targeted re-engagement strategies.

Additionally, the director and actor-based views highlighted the impact of well-known filmmakers and actors on rental popularity, providing a basis for personalized movie recommendations and marketing. Overall, this project demonstrates how data-driven insights can enhance the effectiveness of movie rental businesses by aligning offerings with customer preferences.