mySQLFanclub

Manmitha Neelam
Evan Doran
Jimmy Wu
Jason Golubski

Table of Contents

Description of the Problem

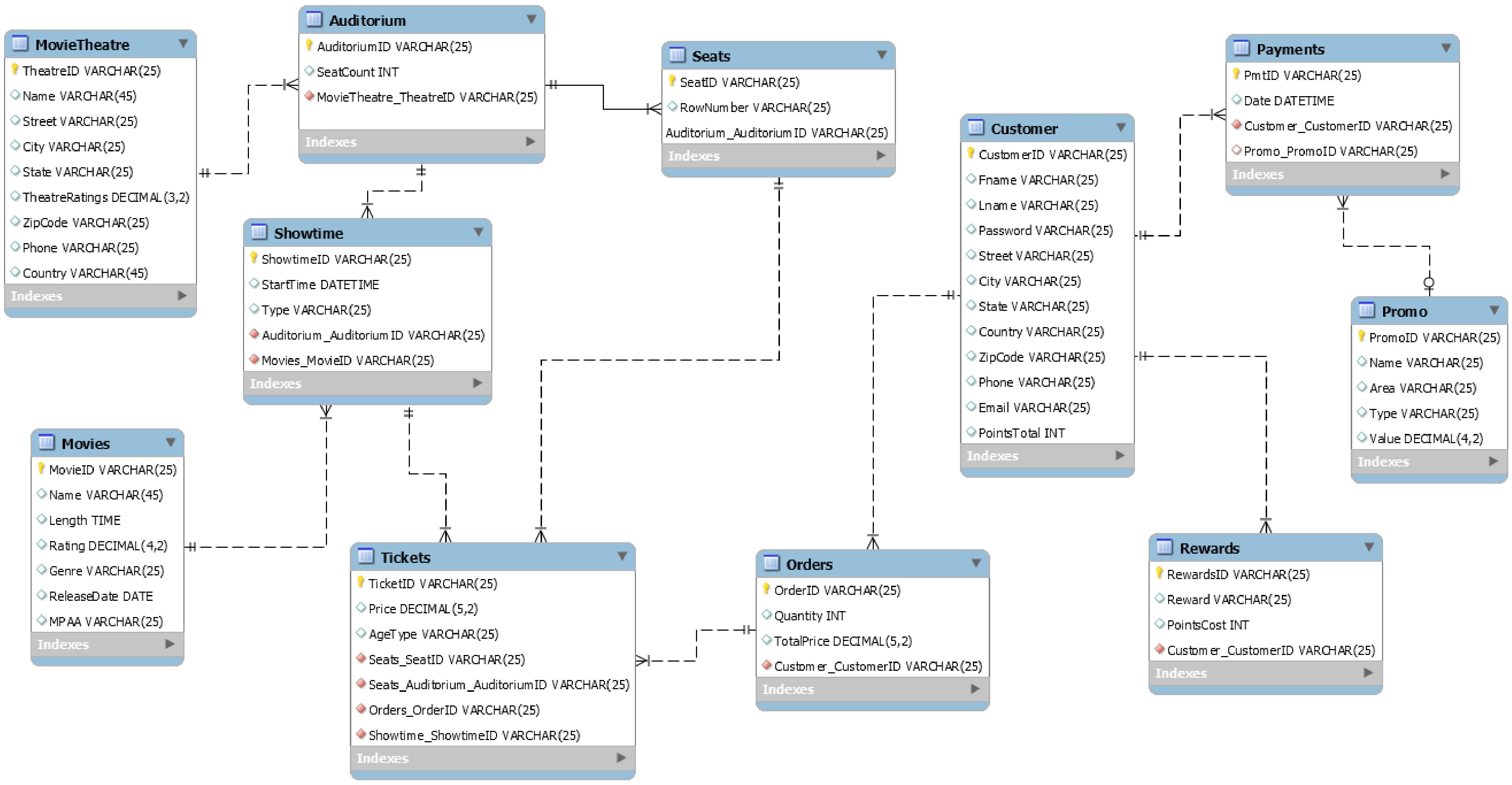
This database was created in response to the massive amounts of data that flows through a movie theater reservation system for Regal Cinemas, commonly known as the Regal Theatres. The transaction information was recorded through customer and ticket information; seats were recorded through auditorium and seating information, information about the movie was documented through the movie itself, theaters, and showtime information, transactions were recorded through payments and order information, discount and freebies were recorded through promotion and reward information.

The customer is given a unique id, and his/her first name, last name, password, location, email, and point systems are also stored in the database. A customer can have multiple payments and movie ticket orders; however, a particular instance of payment belongs to one customer as well as orders. There is a one-to-one relationship between customers and the point table that will keep track of all the points they have accumulated to redeem on their next purchases if they meet promo requirement. The reward table is also associated with the promotion through freebies such as drink or popcorn upgrades. Points accumulated can be used to redeem these items. This table is identified with a unique sequential ID number, reward name, points it required, and the customers who have applied these rewards to their orders. Points can be applied to various forms of rewards, but the rewards can only go through the point system only once meaning no double counting.

Payment of the orders is classified by id as the primary key and contains the date and time the payment was made, the tip amount of the payment, and how the payment was made. These payments are also associated with promo codes in which one promo code can have many payments, but only one promo code can be used on a payment. The promo code is identified by an id and has a name, an area classified as discounting from the subtotal or the convenience fee, type categorized as a percent or an amount, and a total value of the amount as a decimal after the promo is applied. The values can be used to discount from the total payment of the order. An order contains an id, which is its primary key, the quantity (number of tickets purchased within an order), and the total price of the order. A particular order also has only one customer. On the other hand, customers can have many orders.

Orders are related to the ticket information that has a unique TicketID, price of the ticket, age of the customer, seat information, auditorium information, order information, and showtime information. This is a crucial piece of data that tie most of our relationships together. Movies contain a unique ID, its name, duration of the movie, the rating out of 10, genre, released date, and MPAA. It is related to the movie theater information which contains the theater's ID, name of the theater, its street, city, state, rating, zip code, phone, and country. A single movie can be screamed at multiple showtimes at different times, or it can be screened simultaneously in a different auditorium. A single movie theater can also feature multiple movies at a time. Inside a movie theater, there are multiple auditoriums. They are uniquely identified by its ID, count of seats, and the theater in relation to the auditorium stored. The auditorium can also have multiple seats with each auditorium with a variety of seating options ranging from 10-30 enumerated with row number. Because of this one to many relationships with seats, seat table has a unique ID, row number, and the auditorium number it is referencing

Data Model


MovieTheatre
TheatreID VARCHAR(25)
Name VARCHAR(45)
Street VARCHAR(25)
City VARCHAR(25)
State VARCHAR(25)
TheatreRatings DECIMAL(3,2)
ZipCode VARCHAR(25)
Phone VARCHAR(25)
Country VARCHAR(45)
Indexes
Auditorium
AuditoriumID VARCHAR(25)
SeatCount INT
MovieTheatre_TheatreID VARCHAR(25)
Indexes
Seats
SeatID VARCHAR(25)
RowNumber VARCHAR(25)
Auditorium_Auditorium ID VARCHAR(25)
Indexes
Payments
PmtID VARCHAR(25)
Date DATETIME
Customer_CustomerID VARCHAR(25)
Promo_PromoID VARCHAR(25)
Indexes
Showtime
ShowtimeID VARCHAR(25)
StartTime DATETIME
Type VARCHAR(25)
Auditorium_Auditorium ID VARCHAR(25)
Movies_MovieID VARCHAR(25)
Indexes
Customer
CustomerID VARCHAR(25)
Fname VARCHAR(25)
Lname VARCHAR(25)
Password VARCHAR(25)
Street VARCHAR(25)
City VARCHAR(25)
State VARCHAR(25)
Country VARCHAR(25)
ZipCode VARCHAR(25)
Phone VARCHAR(25)
Email VARCHAR(25)
PointsTotal INT
Indexes
Promo
PromoID VARCHAR(25)
Name VARCHAR(25)
Area VARCHAR(25)
Type VARCHAR(25)
Value DECIMAL(4,2)
Indexes
Movies
MovieID VARCHAR(25)
Name VARCHAR(45)
Length TIME
Rating DECIMAL(4,2)
Genre VARCHAR(25)
ReleaseDate DATE
MPAA VARCHAR(25)
Indexes
Tickets
TicketID VARCHAR(25)
Price DECIMAL(5,2)
AgeType VARCHAR(25)
Seats_SeatID VARCHAR(25)
Seats_Auditorium_AuditoriumID VARCHAR(25)
Orders_OrderID VARCHAR(25)
Showtime_ShowtimeID VARCHAR(25)
Indexes
Orders
OrderID VARCHAR(25)
Quantity INT
TotalPrice DECIMAL(5,2)
Customer_CustomerID VARCHAR(25)
Indexes
Rewards
RewardsID VARCHAR(25)
Reward VARCHAR(25)
PointsCost INT
Customer_CustomerID VARCHAR(25)
Indexes

Movie Theatre Word Dictionary

Table: **Auditorium**

Column Name	Description	Data Type	Size	Format	Key?
AuditoriumID	Unique integer digit used to identify auditorium	Varchar	25	1	PK
SeatCount	Number of seats in the auditorium	INT	11	30	
MovieTheatre_TheatreID	Reference to the unique VARCHAR assigned to the individual Movie Theatre in the Movie Theatre table	Varchar	25	2	FK: Referencing MovieTheatre Table

Table: **Customer**

Column Name	Description	Data Type	Size	Format	Key?
CustomerID	Integer digit to identify customer	Varchar	25	2	PK
Fname	Customer’s first name	Varchar	25	Jerry	
Lname	Customer’s last name	Varchar	25	Springer	
Password	Customer’s password	Varchar	25	123456	
Street	Customer’s street address	Varchar	25	12 Abcd St	
City	City that the customer lives in	Varchar	25	Athens	
State	State that the customer lives in	Varchar	25	GA	
Country	Country that the customer lives in	Varchar	25	United States	
Phone	Customer’s phone number	Varchar	25	1234567890	
Email	Customer’s email address	Varchar	25	jerryspringer@gmail.com	
ZipCode	Customer’s zip code for place of residence	Varchar	25	12345	
PointsTotal	The total number of reward points associated with customer	INT	11	1000	

Table: **Movies**

Column Name	Description	Data Type	Size	Format	Key?
MovieID	unique integer digit assigned to identify individual Movies	Varchar	25	12	PK
Name	Movie name	Varchar	45	SQL: A Srini Story	
Length	Movie runtime	Time	10	2:15:00	
Rating	Movie quality on a number scale (1-10)	Decimal	4,2	6.30	
Genre	Movie genre	Varchar	25	Fantasy	
ReleaseDate	Movie release date	Date	10	2019-01-25	
MPAA	Movie advisory rating based on content	Varchar	25	PG-13	

Table: **MovieTheatre**

Column Name	Description	Data Type	Size	Format	Key?
TheatreID	Unique integer digit assigned to identify individual theatres	Varchar	25	1	PK
Name	Theatre's Name	Varchar	25	Small Mo-Town Regal	
Street	Theatre's Street	Varchar	25	242 Hull Street	
City	Theatre's city	Varchar	25	Athens	
State	Theatre's state	Varchar	25	GA	
TheatreRating	Quality rating of Theatre (1-5)	Decimal	3,2	4.45	
ZipCode	Theatre's Zipcode	Varchar	25	30604	
Phone	Theatre's phone number	Varchar	25	1234567890	
Country	Theatre's country	Varchar	45	United States	

Table: **Orders**

Column Name	Description	Data Type	Size	Format	Key?
OrderID	Unique integer digit assigned to identify individual Orders	Varchar	25	1	PK
Quantity	Number of Tickets within the Order	Int	11	5	
Total Price	Total calculated price of the order	Decimal	5,2	45.00	
Customer_CustomerID	ID of Customer associated with the Order	Varchar	25	9	FK: Referencing Customer Table

Table: **Payments**

Column Name	Description	Data Type	Size	Format	Key?
PmtID	unique integer digit assigned to identify individual Payments	Varchar	25	12	PK
Date	Date and time a payment was made	Datetime		YYYY-MM-DD HH:MM:SS	
Customer_CustomerID	ID of Customer associated with this payment	Varchar	25	9	FK: Referencing Customer Table
Promo_PromoID	ID of promo code associated with this payment	Varchar	25	3	FK: Referencing Promo Table

Table: **Promo**

Column Name	Description	Data Type	Size	Format	Key?
PromoID	Unique integer digit assigned to identify individual promotions	Varchar	25	1	PK
Name	Name of Promotion	Varchar	25	NOCONV	
Area	The area of the transaction that the Promo affects	Varchar	25	Subtotal	
Type	Whether the promotion is a percentage off or an amount off	Varchar	25	Amount	
Value	The actual percentage or amount off	Decimal	4,2	2.00	

Table: **Rewards**

Column Name	Description	Data Type	Size	Format	Key?
RewardsID	Unique integer digit assigned to identify individual rewards redeemed	Varchar	25	2	PK
Reward	Name of the Reward in the account	Varchar	25	Free Drink Upgrade	
PointsCost	Price of the Reward	INT	11	1500	
Customer_CustomerID	ID of the Customer associated with the Rewards Account	Varchar	25	3	FK: Referencing Customer Table

Table: **Seats**

Column Name	Description	Data Type	Size	Format	Key?
SeatID	unique integer digit assigned to identify individual Seats	Varchar	25	1	PK
RowNumber	Row number of the seat	Varchar	25	9	
Auditorium_AuditoriumID	VARCHAR of the Auditorium associated with the Seat	Varchar	25	10	FK: Referencing Auditorium Table

Table: **Showtime**

Column Name	Description	Data Type	Size	Format	Key?
ShowtimeID	unique integer digit assigned to identify individual Showtimes	Varchar	25	2	PK
StartTime	The time that the movie starts	Datetime		YYYY-MM-DD HH:MM:SS	
Type	Type of movie screening	Varchar	25	IMAX	
Auditorium_AuditoriumID	VARCHAR of the Auditorium associated with the Showtime	Varchar	25	12	FK: Referencing Auditorium Table
Movies_MovieID	VARCHAR of the Movie associated with the Showtime	Varchar	25	8	FK- Referencing Movies Table

Table: **Tickets**

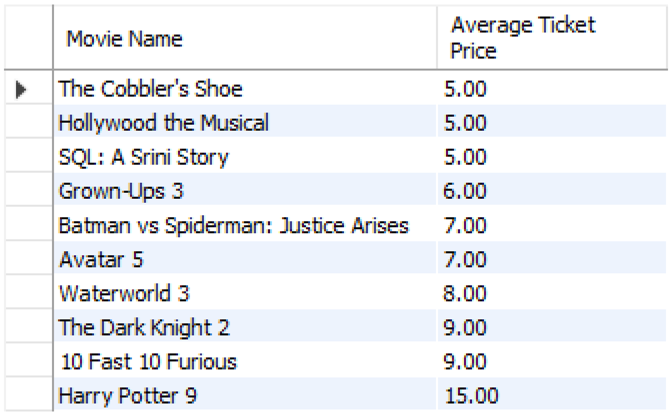
Column Name	Description	Data Type	Size	Format	Key?
TicketID	unique integer digit assigned to identify individual Tickets	Varchar	25	11	PK
Price	The Ticket's price	Varchar	5,2	9.00	
AgeType	Customer Classification of Ticket	Varchar	25	Military	
Seats_SeatID	ID of the Seat associated with the Ticket	Varchar	25	56	FK: Referencing Seats Table
Seats_Auditorium_AuditoriumID	ID of the Auditorium associated with the Ticket	Varchar	25	11	FK: Referencing Auditorium Table
Orders_OrderID	ID of the Order associated with the Ticket	Varchar	25	9	FK: Referencing Orders Table
Showtime_ShowtimeID	ID of the Showtime associated with the Ticket	Varchar	25	5	FK: Referencing Showtime Table

Format, Queries, & Justifications

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Multiple Table Join/Left/Right/Inner	X	X	X	X	X	X	X	X	X	X
Subquery					X		X			
Correlated Subquery			X							
GROUP BY	X			X		X	X	X		X
HAVING				X		X		X		
ORDER BY	X			X	X	X	X	X	X	
IN or NOT IN							X			
Built in function or a calculated field	X	X	X	X		X	X	X	X	X
REGEXP		X								
EXISTS or NOT EXISTS					X					

1. CALL TP_Q01;
This query calculates the average ticket price of the showtimes associated with a particular movie and orders by ascending value. This is useful as it allows the customer to determine what movies are the cheapest to attend and lets the seller know which movies are likely nearing the end of their theatrical run.

SELECT Name as 'Movie Name', ROUND(AVG(Price),2) AS 'Average Ticket Price'
FROM Movies
JOIN Tickets ON Movies.MovieID = Tickets.TicketID
GROUP BY Name
ORDER BY AVG(Price) ASC;

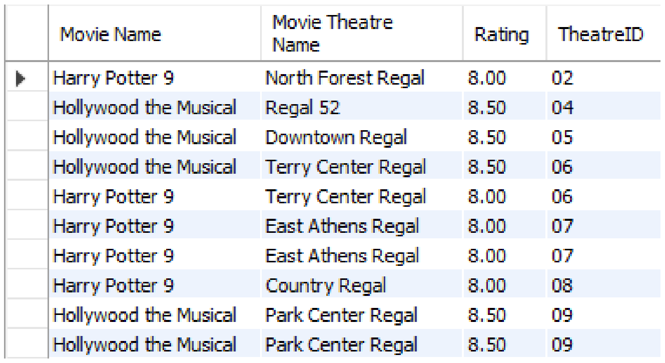


Movie Name	Average Ticket Price
The Cobbler's Shoe	5.00
Hollywood the Musical	5.00
SQL: A Srini Story	5.00
Grown-Ups 3	6.00
Batman vs Spiderman: Justice Arises	7.00
Avatar 5	7.00
Waterworld 3	8.00
The Dark Knight 2	9.00
10 Fast 10 Furious	9.00
Harry Potter 9	15.00

2. CALL TP_Q02;

This query returns any the name of the movies that have above an 8 rating and the showtimes available for that movie. This is very useful for the customer as it allows them to find the best rated movies and for the seller it allows them to track the performance of highly rated movies at the box office and adjust supply for demand.

SELECT Movies.Name AS 'Movie Name', MovieTheatre.Name AS 'Movie Theatre Name', Rating, TheatreID
FROM Movies
JOIN Showtime on MovieID = Movies_MovieID
JOIN Auditorium on Auditorium_AuditoriumID = AuditoriumID
JOIN MovieTheatre on MovieTheatre_TheatreID = TheatreID
WHERE Rating regexp '^8.';

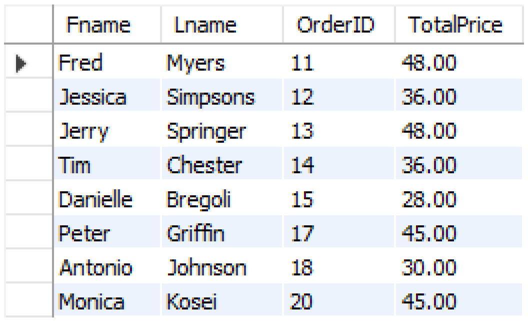


Movie Name	Movie Theatre Name	Rating	TheatreID
Harry Potter 9	North Forest Regal	8.00	02
Hollywood the Musical	Regal 52	8.50	04
Hollywood the Musical	Downtown Regal	8.50	05
Hollywood the Musical	Terry Center Regal	8.50	06
Harry Potter 9	Terry Center Regal	8.00	06
Harry Potter 9	East Athens Regal	8.00	07
Harry Potter 9	East Athens Regal	8.00	07
Harry Potter 9	Country Regal	8.00	08
Hollywood the Musical	Park Center Regal	8.50	09
Hollywood the Musical	Park Center Regal	8.50	09

3. CALL TP_Q03;

This query returns the customers who placed orders that are greater than average, the orderID of the order and the Total Price of the Order. This is useful for determining what is the typical price range customers are willing to pay for the group experience of a movie and how many people typically come in groups. This information can be used for advertising or promotions.

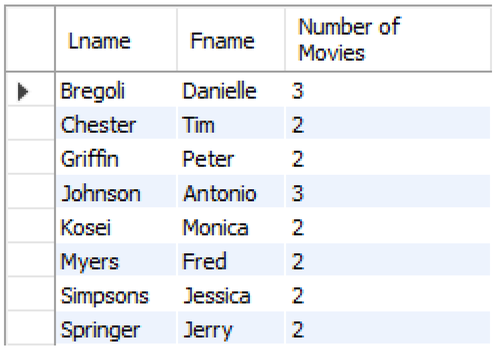
SELECT Fname, Lname, OrderID, TotalPrice
FROM Customer
JOIN Orders ON CustomerID = Orders.Customer_CustomerID
WHERE TotalPrice > (SELECT AVG(TotalPrice)
FROM Orders
WHERE CustomerID = Orders.Customer_CustomerID);



Fname	Lname	OrderID	TotalPrice
Fred	Myers	11	48.00
Jessica	Simpsons	12	36.00
Jerry	Springer	13	48.00
Tim	Chester	14	36.00
Danielle	Bregoli	15	28.00
Peter	Griffin	17	45.00
Antonio	Johnson	18	30.00
Monica	Kosei	20	45.00

4. CALL TP_Q04;
This query returns any customer that has gone to more than 1 movie in the same day. This is useful as it allows the company to know who are the most frequent customers who should be targeted for retention and advertisements.

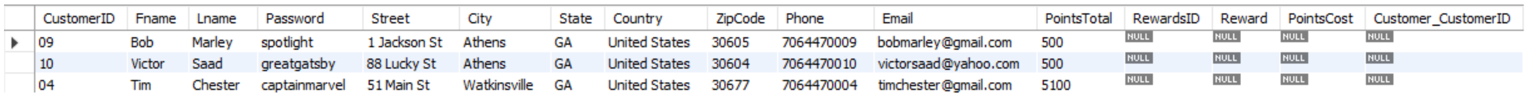
```
SELECT Lname, Fname, COUNT(OrderID) AS 'Number of Movies'
FROM Customer
JOIN Orders ON CustomerID = Orders.Customer_CustomerID
GROUP BY CustomerID
HAVING COUNT(OrderID) > 1
ORDER BY Lname;
```



Lname	Fname	Number of Movies
Bregoli	Danielle	3
Chester	Tim	2
Griffin	Peter	2
Johnson	Antonio	3
Kosei	Monica	2
Myers	Fred	2
Simpsons	Jessica	2
Springer	Jerry	2

5. CALL TP_Q05;
This query shows the Customers with points who have not gotten or used their points for rewards. Ordering by points allows us to examine any associated correlation between the number of current points in their account and the lack of a reward in their current account. We can also target these specific customers to encourage them to use their rewards or remind them about the points in their account to use their rewards.

```
SELECT *
FROM Customer
LEFT JOIN Rewards ON Customer.CustomerID = Rewards.Customer_CustomerID
WHERE NOT EXISTS (SELECT *
                  FROM Customer
                  WHERE CustomerID = Customer_CustomerID)
ORDER BY PointsTotal;
```

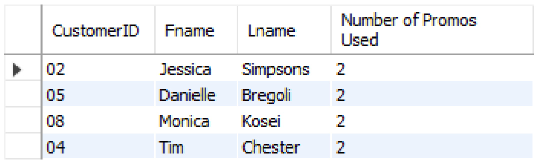


CustomerID	Fname	Lname	Password	Street	City	State	Country	ZipCode	Phone	Email	PointsTotal	RewardsID	Reward	PointsCost	Customer_CustomerID
09	Bob	Marley	spotlight	1 Jackson St	Athens	GA	United States	30605	7064470009	bobmarley@gmail.com	500	NULL	NULL	NULL	NULL
10	Victor	Saad	greatgatsby	88 Lucky St	Athens	GA	United States	30604	7064470010	victorsaad@yahoo.com	500	NULL	NULL	NULL	NULL
04	Tim	Chester	captainmarvel	51 Main St	Watkinsville	GA	United States	30677	7064470004	timchester@gmail.com	5100	NULL	NULL	NULL	NULL

6. CALL TP_Q06;

This query shows us the customers that used more than one promo code. These customers may be more likely to use more than one promo code again and would allow us to segment these customers are those who are likely to use different promo codes when buying a ticket. Further data, research, and deals would allow us to explore if there is a correlation between if having these promo codes would make them more likely keep buying tickets with Regal.

```
SELECT Customer.CustomerID, Customer.Fname, Customer.Lname,
COUNT(Customer.CustomerID) AS 'Number of Promos Used'
FROM Customer
JOIN Payments ON Payments.Customer_CustomerID = Customer.CustomerID
JOIN Promo ON Promo.PromoID = Payments.Promo_PromoID
WHERE Promo.PromoID != 0
GROUP BY CustomerID
HAVING COUNT(customerID) > 1
ORDER BY COUNT(Customer.CustomerID);
```

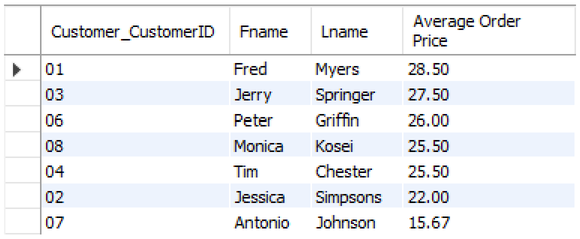


CustomerID	Fname	Lname	Number of Promos Used
02	Jessica	Simpsons	2
05	Danielle	Bregoli	2
08	Monica	Kosei	2
04	Tim	Chester	2

7. CALL TP_Q07;

This query calculates the average price of tickets that customers paid for who have gone to multiple movies to show us the highest revenue-generating customers. This would allow us to offer special promotions and deals to them to retain these customers and the business they provide Regal.

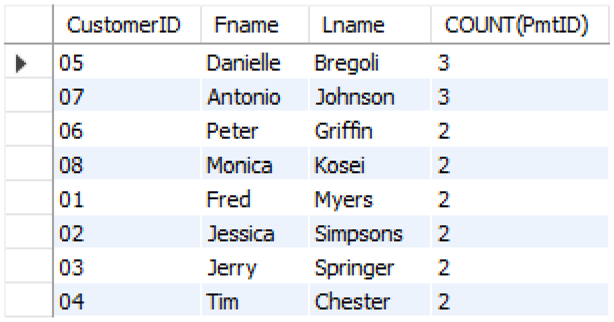
```
SELECT Customer_CustomerID, Customer.Fname, Customer.Lname,
ROUND(AVG(Orders.TotalPrice),2) AS 'Average Order Price'
FROM Customer
JOIN Orders ON Customer.CustomerID = Orders.Customer_CustomerID
WHERE CustomerID IN (SELECT CustomerID
                                FROM Customer
                                JOIN Orders ON Orders.Customer_CustomerID =
Customer.CustomerID
                                WHERE Quantity > 2)
GROUP BY CustomerID
ORDER BY AVG(TotalPrice) DESC;
```



Customer_CustomerID	Fname	Lname	Average Order Price
01	Fred	Myers	28.50
03	Jerry	Springer	27.50
06	Peter	Griffin	26.00
08	Monica	Kosei	25.50
04	Tim	Chester	25.50
02	Jessica	Simpsons	22.00
07	Antonio	Johnson	15.67

8. CALL TP_Q08;
This query finds the recurring customers ordered by the number of orders they have purchased in descending order. This would allow us to, again, offer special deals and promotions to recurring customers to keep them satisfied.

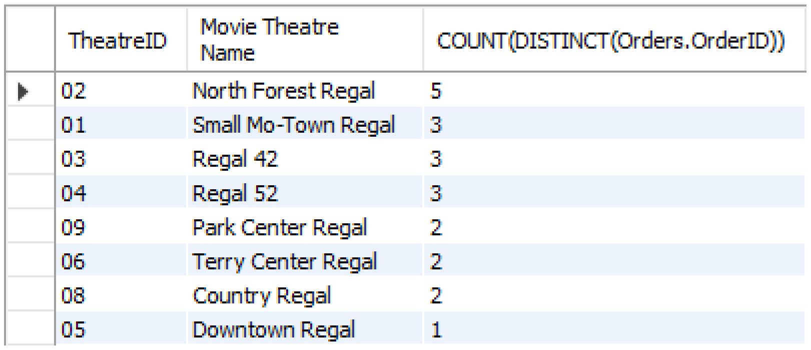
SELECT CustomerID, Fname, Lname, COUNT(PmtID)
FROM Customer
JOIN Payments ON Customer.CustomerID = Payments.Customer_CustomerID
GROUP BY CustomerID
HAVING COUNT(PmtID) > 1
ORDER BY COUNT(PmtID) DESC;



CustomerID	Fname	Lname	COUNT(PmtID)
05	Danielle	Bregoli	3
07	Antonio	Johnson	3
06	Peter	Griffin	2
08	Monica	Kosei	2
01	Fred	Myers	2
02	Jessica	Simpsons	2
03	Jerry	Springer	2
04	Tim	Chester	2

9. CALL TP_Q09;
This query shows the movie theatres and the number of orders for each movie theatre in descending order. This is important because it shows us which movies have the greatest number of ticket sales through the platform that customers are ordering them through and where sales could be improved.

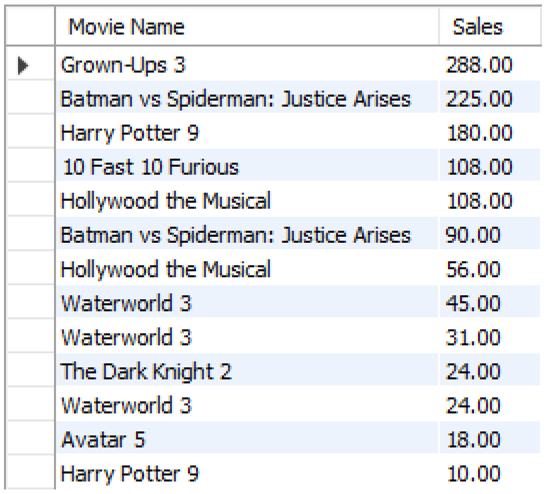
SELECT TheatreID, MovieTheatre.Name as 'Movie Theatre Name', COUNT(DISTINCT(Orders.OrderID))
FROM MovieTheatre
JOIN Auditorium ON MovieTheatre.TheatreID = Auditorium.MovieTheatre_TheatreID
JOIN Seats ON Auditorium.AuditoriumID =Seats.Auditorium_AuditoriumID
JOIN Tickets ON Seats.SeatID = Tickets.Seats_SeatID
JOIN Orders ON Tickets.Orders_OrderID = Orders.OrderID
GROUP BY TheatreID,MovieTheatre.Name
ORDER BY COUNT(DISTINCT(Orders.OrderID)) DESC;



TheatreID	Movie Theatre Name	COUNT(DISTINCT(Orders.OrderID))
02	North Forest Regal	5
01	Small Mo-Town Regal	3
03	Regal 42	3
04	Regal 52	3
09	Park Center Regal	2
06	Terry Center Regal	2
08	Country Regal	2
05	Downtown Regal	1

10. CALL TP_Q10;
This query displays the movie names and total sales generated for each movie allowing us to see the most popular or revenue-generating movies at Regal Cinemas. This is important in determining various financial margins in relation to expense as well as other marketing purposes for Regal to determine which movies to focus their efforts on more vs. viewings they may need to cut back on.

SELECT Movies.Name as 'Movie Name', SUM(TotalPrice) AS 'Sales'
FROM Movies
JOIN Showtime ON Movies.MovieID = Showtime.Movies_MovieID
JOIN Auditorium ON Auditorium.MovieTheatre_TheatreID =
Showtime.Auditorium_AuditoriumID
JOIN Seats ON Auditorium.AuditoriumID =Seats.Auditorium_AuditoriumID
JOIN Tickets ON Seats.SeatID = Tickets.Seats_SeatID
JOIN Orders ON Tickets.Orders_OrderID = Orders.OrderID
GROUP BY Movies.Name,Orders.Quantity
ORDER BY SALES DESC;



Movie Name	Sales
Grown-Ups 3	288.00
Batman vs Spiderman: Justice Arises	225.00
Harry Potter 9	180.00
10 Fast 10 Furious	108.00
Hollywood the Musical	108.00
Batman vs Spiderman: Justice Arises	90.00
Hollywood the Musical	56.00
Waterworld 3	45.00
Waterworld 3	31.00
The Dark Knight 2	24.00
Waterworld 3	24.00
Avatar 5	18.00
Harry Potter 9	10.00