

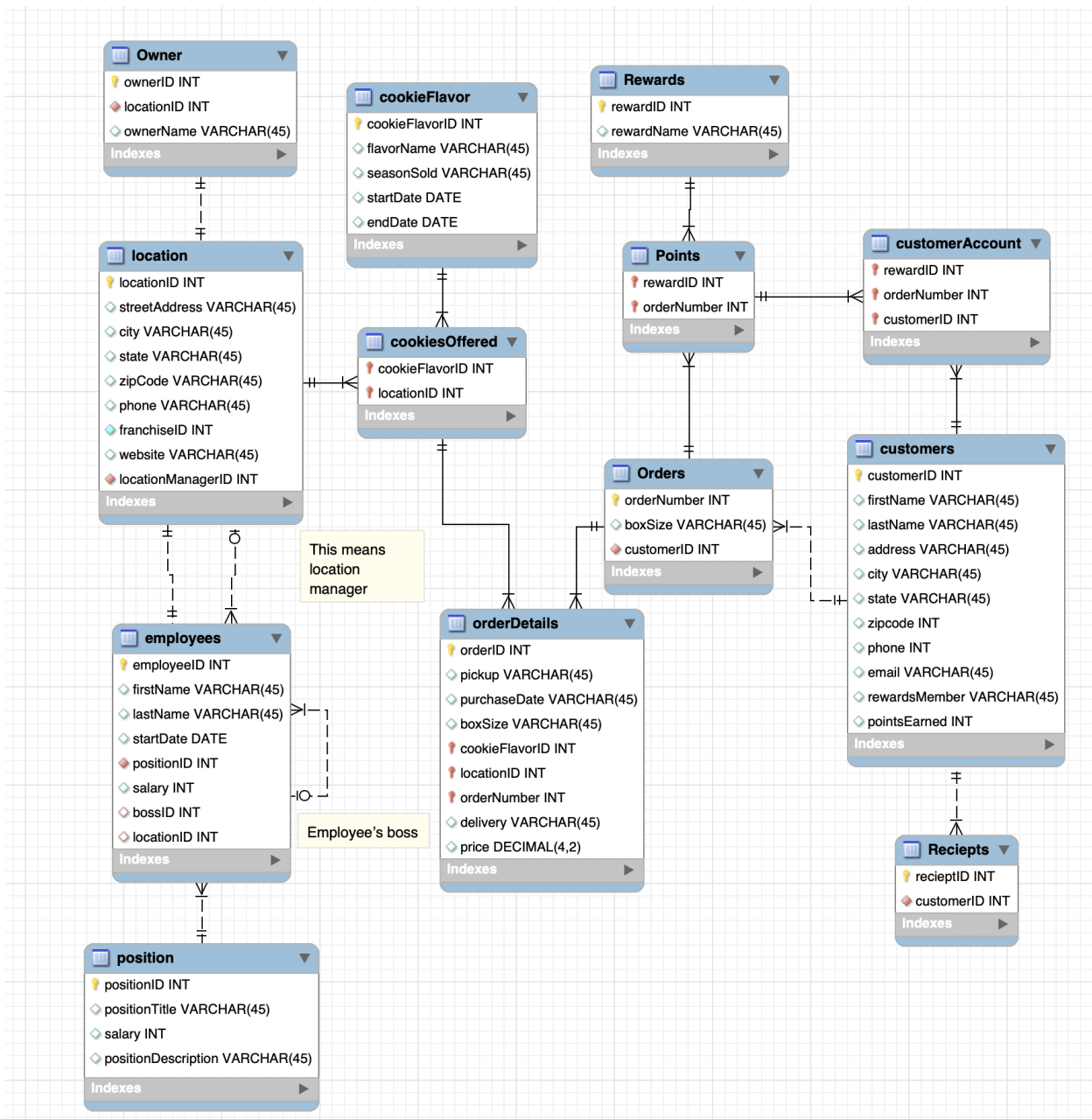
4/7/2023

MIST 4610  
Professor Skousen  
Project 1 Group 2

Caroline Schmidt, Vincent Claussen, Sharvani Mangroo, Matt Faby, and Rebekah  
Thompson

## Data Model:

The problem that we are modeling is how the company Crumbl Cookie could implement a rewards system. The reward system has points which a person can earn through ordering cookies if the person has already signed up for a free reward membership (shown in the entity customers with the attribute title “rewardMember”). They can then earn points to their account on each order of cookies via the “Rewards” table.



The data model begins with a location that has many relationships. First, the location can have one owner and an owner can have one location. Next, the location can have many employees which includes the location manager as well as the employee's boss.

Continuing with employees, an employee can only have one position, but one position can be held by many employees. An example of this would be employee 1 and employee 2 can hold position 1. Moving forward, a location can have many cookies offered and many cookie flavors, with an associative entity of cookies offered.

Moving to the next entity, cookiesOffered can belong to many orderDetails, however, orderDetails can only belong to one cookiesOffered.

Next, orderDetails can only be connected to one Order, however, Orders can be connected to many orderDetails. Orders can also be connected to Rewards through the common entity of Points.

Following, Points can be connected to customers through the common entity of customerAccounts. This means that Points can belong to many customers through the customerAccount entity. Customers can also have many receipts but one receipt can only belong to one customer.

Finally, Customers can have many orders, but only one order can belong to one customer.

## Data Dictionary:

Table: **Owner**

Column Name	Description	Data Type	Size	Format
ownerID	PK for owner table	int		
locationID	Connection to location table	int		
ownerName	Provides owner name	varchar	45	

Table: **location**

Column Name	Description	Data Type	Size	Format
locationID	PK	int		
streetAddress	Address of the store	varchar	45	196 Alps Road Suite 5
city	City of the store	varchar	45	Athens
state	State of the store	varchar	45	GA
zipCode	Zip code of the store	varchar	45	30606
phone	Phone number for the store	varchar	45	7066780981
franchiseID	Connection to franchise	int		
website	Website to the specific location	varchar	45	crumble.com/locationID
locationManagerID	Manager ID for that location	int		

Table: **cookieFlavor**

Column Name	Description	Data Type	Size	Format
cookieFlavorID	Primary key, identification number for flavors	int		
flavorName	Name of cookie flavor	varchar	45	
seasonSold	For seasonal flavor options	varchar	45	

startDate	Identify start date for flavor	date		yyyy-mm-dd
endDate	Identify end date for flavor	date		yyyy-mm-dd

Table: **employees**

Column Name	Description	Data Type	Size	Format
employeeID	PK for employees table	int		
firstName	First name of employee	varchar	45	
lastName	Last name of employee	varchar	45	
startDate	Start date of employee	date		yyyy-mm-dd
positionID	Connection from position table	int		
bossID	Boss ID of the employee	int		
salary	How much employee is paid	int		
locationID	FK from location table	int		

Table: **position**

Column Name	Description	Data Type	Size	Format
positionID	PK for position table	int		
positionTitle	Name of the job	varchar	45	
salary	Salary of the job	int		
positionDescription	Describes position	varchar	45	

Table: **Orders**

Column Name	Description	Data Type	Size	Format
orderNumber	PK for orders table	int		
boxSize	Size of box	varchar	45	4 pack

customerID	Connection from customer table	int		
------------	--------------------------------	-----	--	--

Table: **customers**

Column Name	Description	Data Type	Size	Format
customerID	PK for customers table	int		
firstName	First name of customer	varchar	45	
lastName	Last name of customer	varchar	45	
address	Address of customer	varchar	45	
city	City of customer	varchar	45	Athens
state	State of customer	varchar	45	GA
zipcode	Zip Code for customer	int		30606
phone	Phone number of customer	int		7066780981
email	Email of customer	varchar	45	
rewardsMember	Indicates if they are a rewards member	varchar	45	
pointsEarned	How many points that customer has	int		

Table: **orderDetails**

Column Name	Description	Data Type	Size	Format
orderID	PK for order details table	int		
pickup	Shows if it is a pickup order	varchar	45	
purchaseDate	Date of purchase	varchar	45	yyyy-mm-dd
boxsize	Box size of the order	varchar	45	4 pack
delivery	Shows if order is for delivery	varchar	45	
price	Price of order	decimal	4,2	10.99
cookieFlavorID	FK to connect to cookieFlavor table	int		

locationID	FK to connect to location table	int		
orderNumber	FK to connect to order table	int		

Table: **cookiesOffered**

Column Name	Description	Data Type	Size	Format
cookieFlavorID	FK to the cookieFlavor table, composite key	int		
locationID	FK to the location table, composite key	int		

Table: **Rewards**

Column Name	Description	Data Type	Size	Format
rewardID	PK for rewards table	int		
rewardName	Name of reward	varchar	45	

Table: **Points**

Column Name	Description	Data Type	Size	Format
rewardID	FK to connect to rewards table, composite key	int		
orderNumber	FK to connect to orders table, composite key	int		

Table: **customerAccount**

Column Name	Description	Data Type	Size	Format
rewardID	FK for reward table, composite key	int		
orderNumber	FK for orders table, composite key	int		
customerID	FK for customers table, composite key	int		

## Queries:

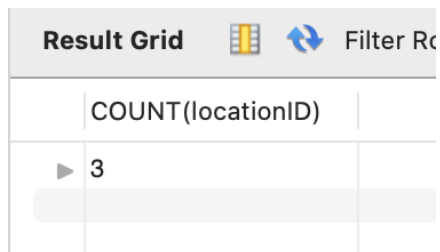
### Query #1:

Count the number of stores in Georgia.

### Query:

```
SELECT COUNT(locationID) FROM location
WHERE state REGEXP 'GA';
```

### Result:



The screenshot shows a 'Result Grid' interface. At the top, there are icons for a grid, a refresh, and a filter. Below the header, there is a table with one column labeled 'COUNT(locationID)' and one row containing the value '3'.

COUNT(locationID)
3

### Justification:

This information might be valuable to corporate because if they have too many stores in Georgia it could indicate that they are oversaturated there. If this is the case, it may be wise to look outside the state of Georgia for new opportunities.

### Query #2:

What percentage of orders include the flavor Sugar?

### Query:

```
SELECT CONCAT(COUNT(*)/(SELECT COUNT(*) FROM orderDetails)*100, '%') AS
percOfSugar FROM cookieFlavor
JOIN cookiesOffered ON cookiesOffered.cookieFlavorID =
cookieFlavor.cookieFlavorID
JOIN orderDetails ON orderDetails.cookieFlavorID = cookiesOffered.cookieFlavorID
WHERE orderDetails.cookieFlavorID = 2
GROUP BY orderDetails.cookieFlavorID;
```



Result Grid   Filter Rows:

percOfSugar	
▶ 16.6667%	

**Result:**

**Justification:**

This information would be valuable to the research and development teams at Crumbl Cookie because if they are creating new flavors for the next cycle of cookie flavors they would want them to be new and different. Having the flavor contain sugar may seem reused and too similar to old flavors. Almost 20% of their flavors contain the flavor sugar. Crumbl rotates their selection of 6 cookies a week and if one of those every week will contain the flavor sugar then it may indicate it is time to increase the diversity of flavors and reduce that flavor in the rotation.

**Query #3:**

List the owner names for the stores in Georgia and Texas.

**Query:**

```
SELECT ownerName FROM Owner
JOIN location ON location.locationID = Owner.locationID
WHERE state IN ('GA', 'TX');
```

Result Grid   Fi

ownerName	
▶ Rebekah	
▶ Matt	
▶ Vincent	
▶ Sharvani	

**Result:**

**Justification:**

Corporate would be interested in this information because if store owners are preferring to open new stores in these two states they might want to research why. If those regions are booming economies they may want to look for more owners in that area or inversely recommend those owners move to newer territories.

**Query #4:**

List the location ID for all of the owners except Matt.

**Query:**

```
SELECT DISTINCT locationID FROM Owner
```

WHERE NOT EXISTS (SELECT \* FROM location WHERE ownerName = 'Matt');

locationID
92
197
167
22
156

**Result:**

**Justification:**

Corporate may have done some research and found that Matt has a large percentage of the store locations in the region. They want to find out how many other owners are in the area. This is valuable information because if hypothetically Matt's stores are performing differently from the other owner they would want to know why to increase the performance of the others.

**Query #5:**

How many flavors include the word chocolate?

**Query:**

SELECT COUNT(flavorName) FROM cookieFlavor  
WHERE flavorName REGEXP 'chocolate';

COUNT(flavorNam...
1

**Result:**

**Justification:**

By finding out how many cookie flavors contain the word chocolate, the research and development teams for the flavors can see if they need to diversify the flavors. Having diversified flavors is important because it is part of the appeal of their rotating selection each week. If many of the cookie flavors, which isn't the case here, contain the word chocolate it may be redundant and boring for customers.

**Query #6:**

List the cookie flavor, and price for cookies whose price is greater than the average price of cookies.

**Query:**

SELECT flavorName, price FROM cookieFlavor  
JOIN cookiesOffered ON cookieFlavor.cookieFlavorID =  
cookiesOffered.cookieFlavorID  
JOIN orderDetails ON cookiesOffered.locationID = orderDetails.locationID

WHERE price > (SELECT AVG(price) FROM orderDetails);

flavorName	price
Apple Pie	35.68
Birthday Cake	20.98
Brownie Batter	20.98

**Result:**

**Justification:**

This information might be valuable to corporate because if certain cookies are priced higher than the average price of the cookie that might defer some customers from buying them. Some price research might want to be done to see if at this price point Crumbl can reasonably sell these cookies.

**Query #7:**

Count the number of stores per state and list the state(s) that have more than 1 location.

**Query:**

SELECT COUNT(locationID), state FROM location  
GROUP BY state  
HAVING COUNT(locationID) > 1;

**Result:**

COUNT(locationl...	state
3	GA

a.

**Justification:**

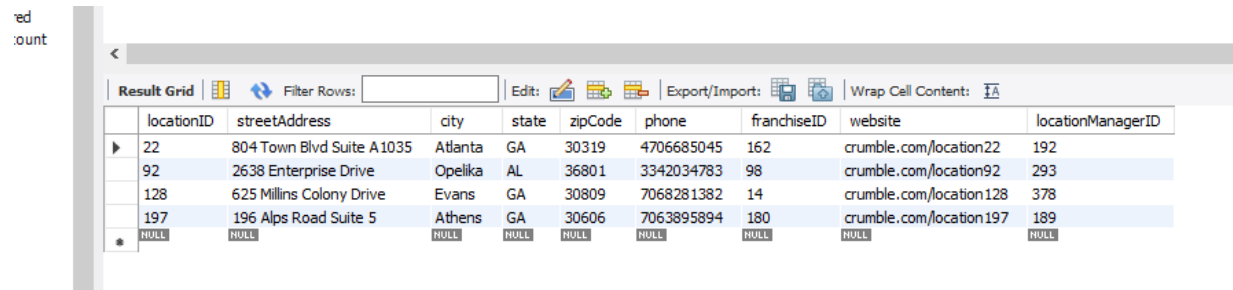
Crumble corporate is interested in expanding the amount of stores they have in the nation. They want to know how many stores have more than one store so that they can focus on opening in areas that have fewer stores. This will target newer regional markets that they can focus on.

**Query #8:**

List the number of stores that are specifically in the states of GA, AL.

**Query:**

```
SELECT * FROM location
WHERE state in ('GA', 'AL');
```

**Result:**

The screenshot shows a database query result grid with the following columns: locationID, streetAddress, city, state, zipCode, phone, franchiseID, website, and locationManagerID. The results are filtered to show only locations in Georgia (GA) and Alabama (AL). The data is as follows:

locationID	streetAddress	city	state	zipCode	phone	franchiseID	website	locationManagerID
22	804 Town Blvd Suite A1035	Atlanta	GA	30319	4706685045	162	crumble.com/location22	192
92	2638 Enterprise Drive	Opelika	AL	36801	3342034783	98	crumble.com/location92	293
128	625 Millins Colony Drive	Evans	GA	30809	7068281382	14	crumble.com/location128	378
197	196 Alps Road Suite 5	Athens	GA	30606	7063895894	180	crumble.com/location197	189

**Justification:**

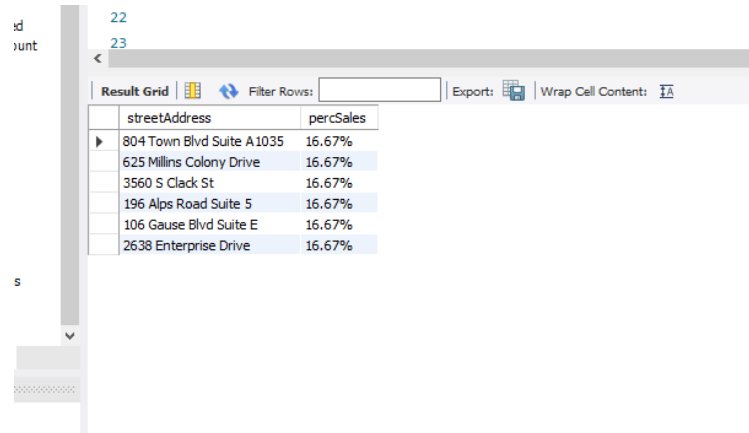
Crumbl corporate is interested in finding out which stores they have in Georgia and Alabama. This information could be valuable in a scenario where for some reason these stores have to shut down and there won't be incoming revenue from these stores the next couple weeks.

**Query #9:**

Write a query to list the different locations and what percentage of cookie sales come from each location.

**Query:**

```
SELECT streetAddress, CONCAT(ROUND((COUNT(orderID)/(SELECT COUNT(orderId)
FROM orderDetails))*100,2),'%') AS percSales FROM orderDetails
JOIN cookiesOffered on cookiesOffered.cookieFlavorID = orderDetails.cookieFlavorID
JOIN location on cookiesOffered.locationID = location.locationID
GROUP BY streetAddress;
```

**Result:**

The screenshot shows a database query result grid with the following columns: streetAddress and percSales. The results are filtered to show only locations in Georgia and Alabama. The data is as follows:

streetAddress	percSales
804 Town Blvd Suite A1035	16.67%
625 Millins Colony Drive	16.67%
3560 S Clack St	16.67%
196 Alps Road Suite 5	16.67%
106 Gause Blvd Suite E	16.67%
2638 Enterprise Drive	16.67%

**Justification:**

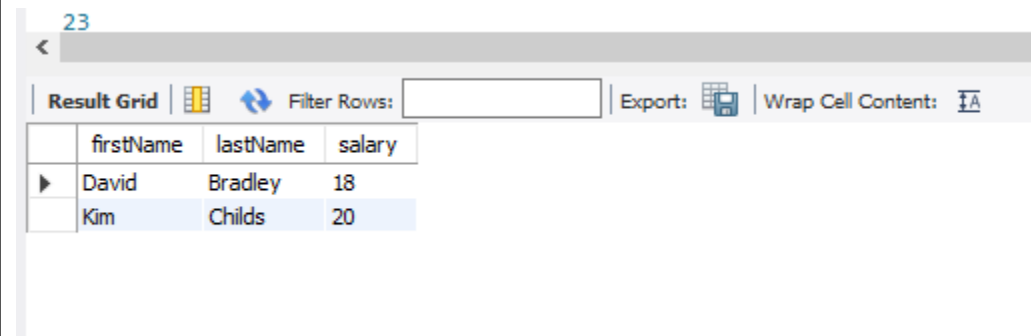
This query could be valuable to Crumbl Cookies because if they have some kind of reward program with their owners that rewards the top 50 earners in the nation, they would need to know who those people are. Additionally, they can see the top earners and research why they are performing better than the other stores and translate that to the others.

**Query #10:**

Write a query showing the employees at locations that have salaries higher than the average of all employees associated with that position. Order by employee ID ascending.

**Query:**

```
SELECT firstName, lastName, employees.salary from employees
JOIN position on position.positionID = employees.positionID
WHERE employees.salary > (SELECT AVG(salary) FROM employees
                           WHERE position.positionID =
employees.positionID)
ORDER BY employeeID ASC;
```

**Result:**

The screenshot shows a database interface with a 'Result Grid' tab selected. The grid displays two rows of data. The first row is for David Bradley with a salary of 18, and the second row is for Kim Childs with a salary of 20. The grid has columns for firstName, lastName, and salary. There are also buttons for 'Filter Rows', 'Export', and 'Wrap Cell Content'.

firstName	lastName	salary
David	Bradley	18
Kim	Childs	20

**Justification:**

A store owner might be interested in this query because if they have a group of employees that are paid different amounts, they might want to see who is getting paid more or less than the average according to their position. If someone is being paid more like David and Kim, it might be the financially responsible thing for the owner to cut their salaries to the average.

### SQL Features Matrix:

	Query 1	Query 2	Query 3	Query 4	Query 5	Query 6	Query 7	Query 8	Query 9	Query 10
<b>Multiple join table</b>		X				X			X	
<b>Subquery</b>		X		X		X			X	
<b>Correlated Subquery</b>										X
<b>GROUP BY</b>		X							X	
<b>GROUP BY with HAVING</b>							X			
<b>ORDER BY</b>										X
<b>IN or NOT IN</b>			X					X		
<b>Built-in Function</b>	X	X			X	X	X			
<b>REGEXP</b>	X				X					
<b>NOT EXISTS</b>				X						

**Database Name:**

ts\_61608\_2

**Database Implementation:**

Procedure 1
Code: create procedure TP_Q1() SELECT COUNT(locationID) FROM location WHERE state REGEXP 'GA';  call TP_Q1();
Procedure 2
Code: create procedure TP_Q2() SELECT CONCAT(COUNT(*)/(SELECT COUNT(*) FROM orderDetails)*100, '%') AS percOfSugar FROM cookieFlavor JOIN cookiesOffered ON cookiesOffered.cookieFlavorID = cookieFlavor.cookieFlavorID JOIN orderDetails ON orderDetails.cookieFlavorID = cookiesOffered.cookieFlavorID WHERE orderDetails.cookieFlavorID = 2 GROUP BY orderDetails.cookieFlavorID;  call TP_Q2;
Procedure 3
Code: create procedure TP_Q3() SELECT ownerName FROM Owner JOIN location ON location.locationID = Owner.locationID WHERE state IN ('GA', 'TX');  call TP_Q3;
Procedure 4

Code:  
create procedure TP\_Q4()  
SELECT DISTINCT locationID FROM Owner  
WHERE NOT EXISTS (SELECT \* FROM location WHERE ownerName = 'Matt');  
  
call TP\_Q4;

#### Procedure 5

Code:  
create procedure TP\_Q5()  
SELECT COUNT(flavorName) FROM cookieFlavor  
WHERE flavorName REGEXP 'chocolate';  
  
call TP\_Q5;

#### Procedure 6

Code:  
create procedure TP\_Q6()  
SELECT flavorName, price FROM cookieFlavor  
JOIN cookiesOffered ON cookieFlavor.cookieFlavorID =  
cookiesOffered.cookieFlavorID  
JOIN orderDetails ON cookiesOffered.locationID = orderDetails.locationID  
WHERE price > (SELECT AVG(price) FROM orderDetails);  
  
call TP\_Q6;

#### Procedure 7

Code:  
create procedure TP\_Q7()  
SELECT COUNT(locationID), state FROM location  
GROUP BY state  
HAVING COUNT(locationID) > 1;  
  
call TP\_Q7;

#### Procedure 8

Code:  
create procedure TP\_Q8()



```
select * from location
where state in ('GA', 'AL');

call TP_Q8;
```

#### Procedure 9

```
Code:
create procedure TP_Q9()
select streetAddress, concat(round((count(orderID)/(select count(orderId) from
orderDetails))*100,2),'%') as percSales from orderDetails
join cookiesOffered on cookiesOffered.cookieFlavorID = orderDetails.cookieFlavorID
join location on cookiesOffered.locationID = location.locationID
group by streetAddress;

call TP_Q9;
```

#### Procedure 10

```
Code:
create procedure TP_Q10()
select firstName, lastName, employees.salary from employees
join position on position.positionID = employees.positionID
where employees.salary > (select avg(salary) from employees
                           where position.positionID =
employees.positionID)
order by employeeID asc;

call TP_Q10;
```