

Paris Baguette Café

Business Analysis Report



Project By: Suchita Negi

MSBA Winter 2020

Database Management Systems – SQL

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Business Scenario Description.....	3
1.2 Swim Lane Diagram	3
1.3 Business Analysis Questions	4

2. LIST OF TABLES AND TABLE CREATION SCRIPT

2.1 Data Model	5
2.2 Table Creation Script	6

3. SQL QUERIES

3.1 Query without Joins	10
3.2 Query with Joins	13
3.3 VIEWS.....	18

4. ADVANCED SQL

4.1 Date / Case / Subquery	19
4.2 Trigger	21
4.3 Stored Procedure	22

5. TABLEAU REPORT

5.1 Tableau Dashboard	24
5.2 Tableau Report with Filter	25

6. CONCLUSION	25
---------------------	----

1. INTRODUCTION

Born from a love of bread and a passion for quality, Paris Baguette is an international, franchise, fast-casual bakery founded in 1988 specializing in French-inspired goods. In addition to chef-inspired cakes, pastries, and signature coffee and tea, PB offer a unique experience to their customers.

With the growing competition, Paris Baguette must balance and monitor the sales/revenue to keep its market position and further expand. This project has built a mock database for Paris Baguette Café based on the customer transaction to retrieve data to answer business related queries. With the help of this database Paris Baguette can make informative business decisions and map effective business strategies.

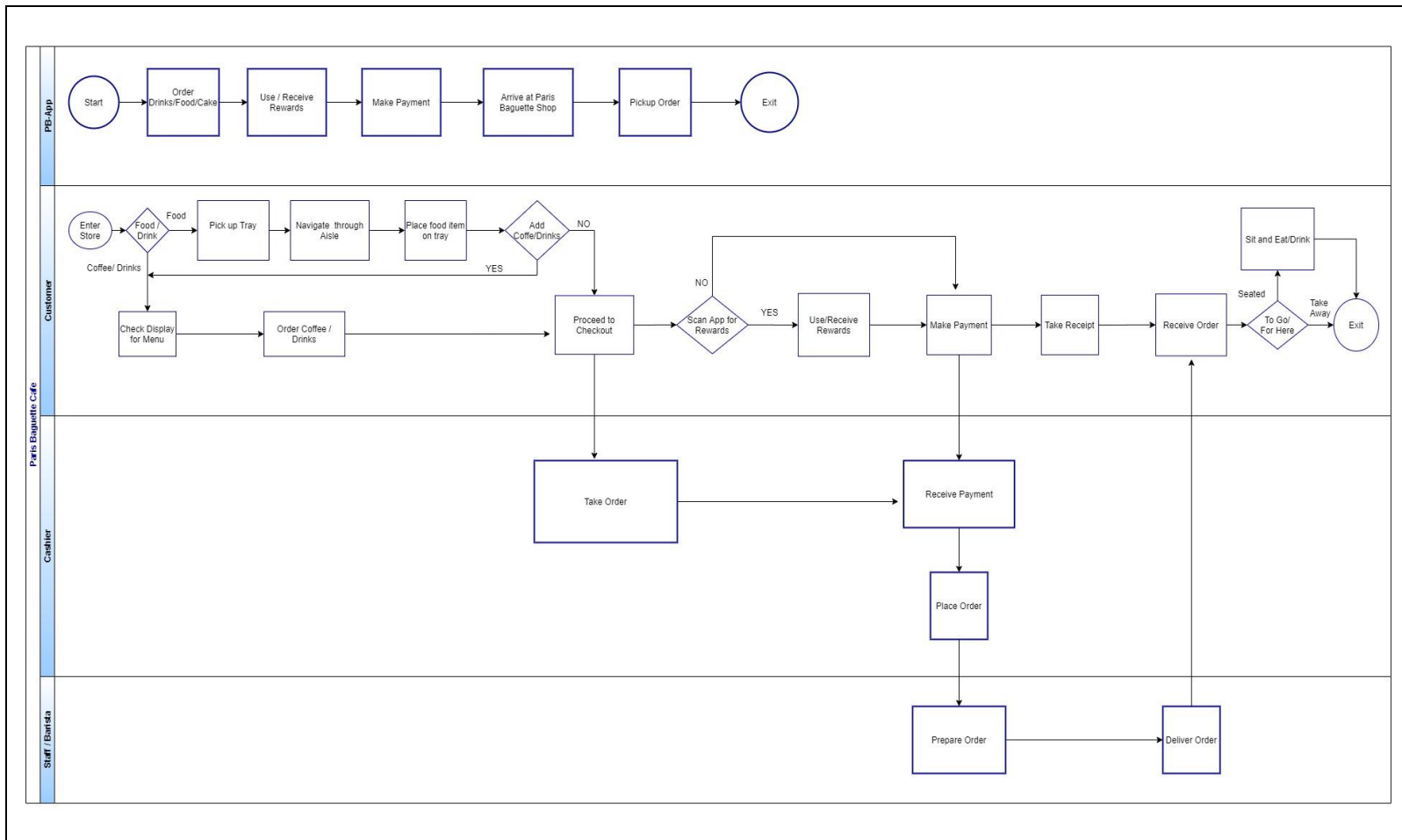
1.1 Business Scenario:

John Doe walked into store at 1:45 pm on a Saturday and got in the food aisle. He picked up a tray, while navigating through different food options, he selected one pastry and one puff and put the food on the tray. Joined the queue and explored the coffee/drinks menu on the display.

At the checkout counter he placed an order for coffee and scanned his mobile app to earn rewards. Payment was made for both coffee and food items by credit card. Received the payment receipt and waited for the coffee.

The Cashier asked for the name and placed the coffee order to the barista after payment. The barista prepared fresh coffee and called for the name after the order was ready. John received order and sat down to consume coffee and food, after eating he disposed the cups in trash and walked off to his car in parking lot at 2:30 pm.

1.2 Swim Lane Diagram:



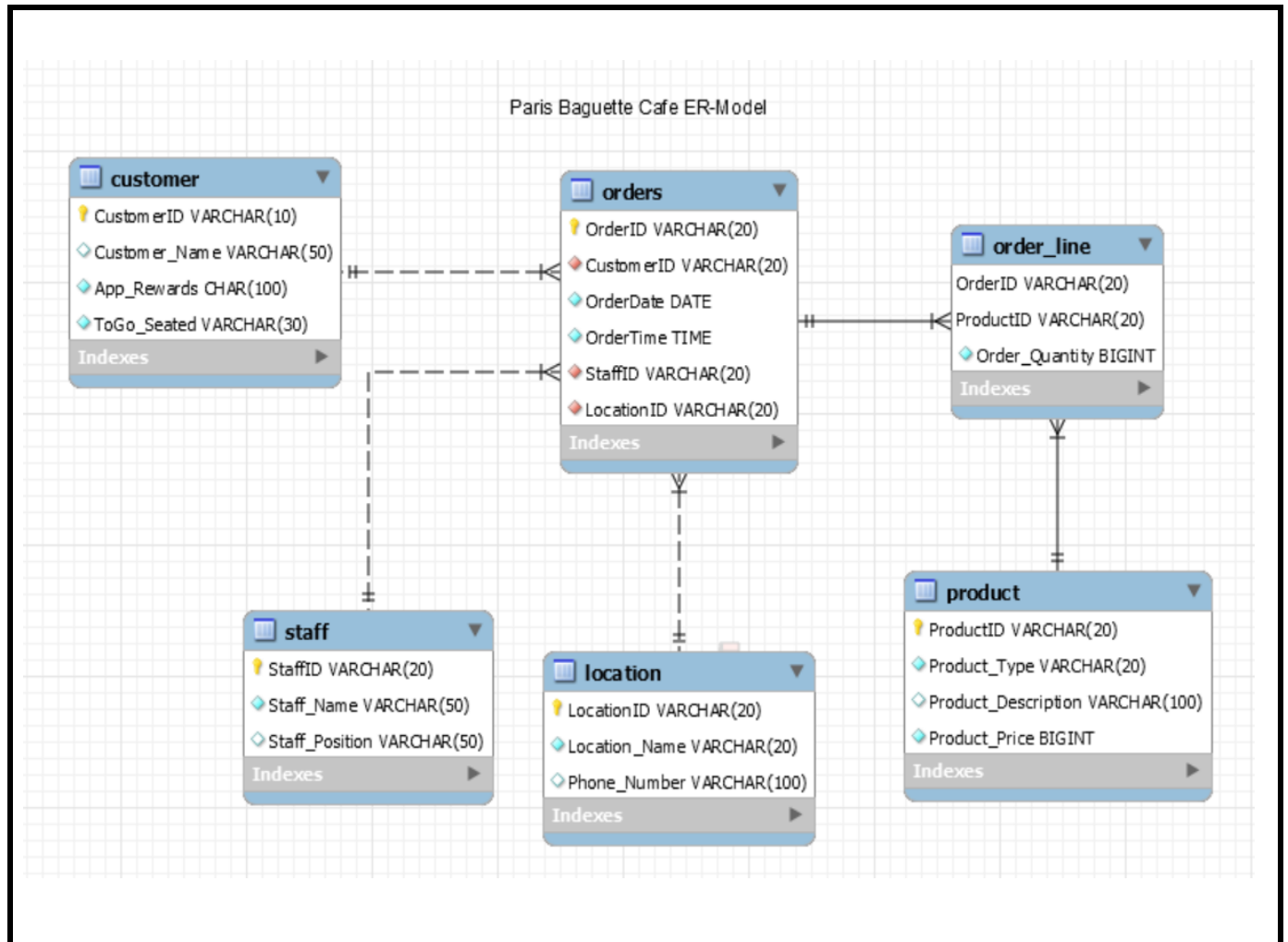
1.3 Business Questions:

1. What are the popular selling items in the cafe?
2. Which month is most profitable?
3. Which product brings the most revenue to the business?
4. What are most customer preferences, take away or Dine In?
5. What are the sales for different days of week (weekdays/weekends)?
6. How many transactions are done using App?
7. What are the average sales per transaction/day?
8. What are the sales numbers for different category (Drinks/Food/Cake)?
9. How much on an average do customer spend?
10. Which location is the most profitable?

2. LIST OF TABLES AND TABLE CREATION SCRIPT

2.1 Data Model

Data Model shows tables connected: Reverse Engineering

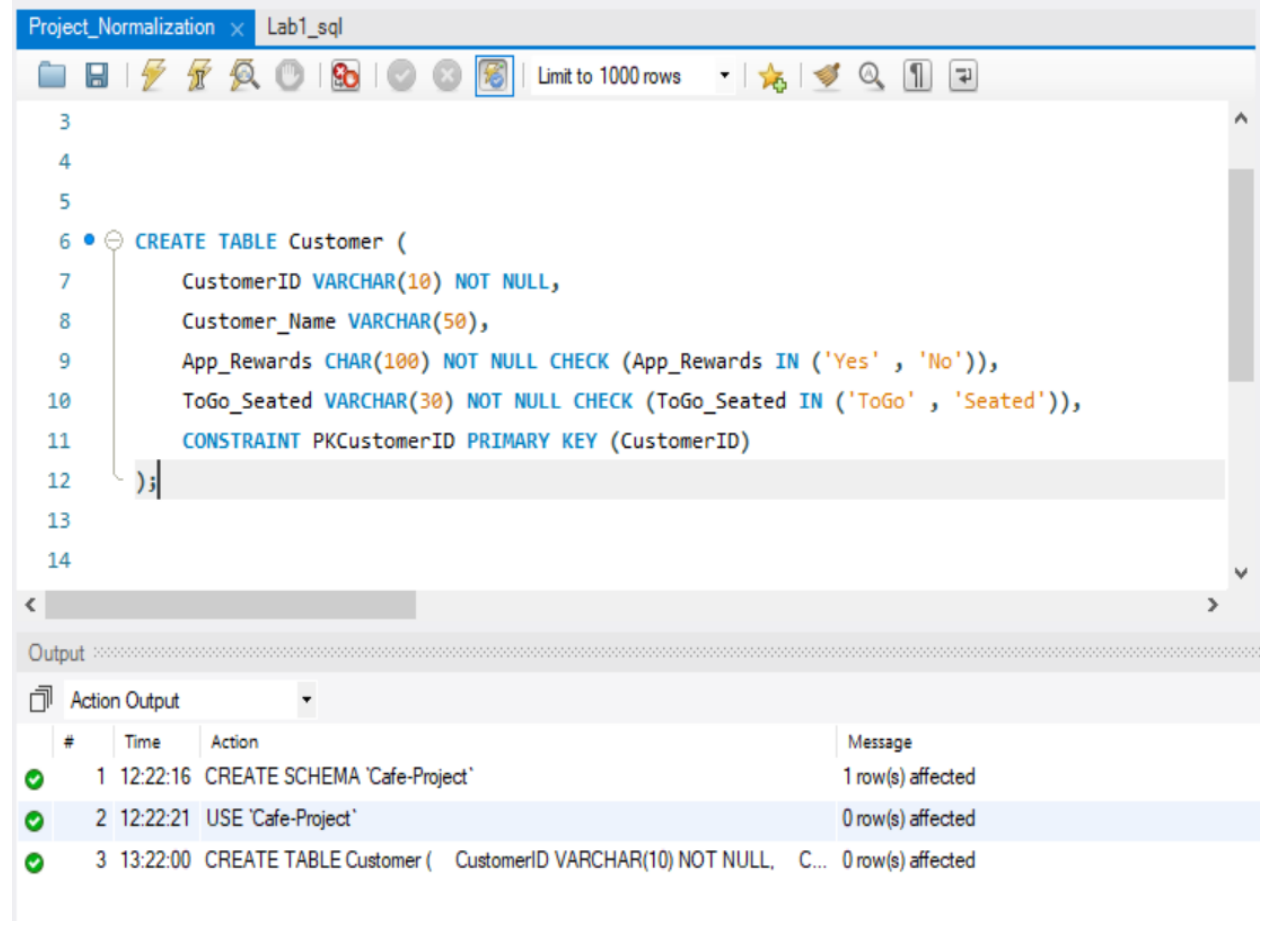


2.2 Table Creation Script

SQL scripts to create tables with Primary and foreign keys. The tables are normalized to 3NF .

- Create Customer Table

The Customer table has details of each customer of PB Cafe such as Name, whether customer uses App or prefer to Go or dine in Options.



The screenshot displays a SQL IDE window titled 'Project_Normalization' with a sub-tab 'Lab1_sql'. The main editor area contains the following SQL script:

```
3
4
5
6 CREATE TABLE Customer (
7     CustomerID VARCHAR(10) NOT NULL,
8     Customer_Name VARCHAR(50),
9     App_Rewards CHAR(100) NOT NULL CHECK (App_Rewards IN ('Yes' , 'No')),
10    ToGo_Seated VARCHAR(30) NOT NULL CHECK (ToGo_Seated IN ('ToGo' , 'Seated')),
11    CONSTRAINT PKCustomerID PRIMARY KEY (CustomerID)
12 );
13
14
```

Below the editor is an 'Output' pane with a dropdown menu set to 'Action Output'. It displays a log of three actions:

#	Time	Action	Message
✓ 1	12:22:16	CREATE SCHEMA 'Cafe-Project'	1 row(s) affected
✓ 2	12:22:21	USE 'Cafe-Project'	0 row(s) affected
✓ 3	13:22:00	CREATE TABLE Customer (CustomerID VARCHAR(10) NOT NULL, C...	0 row(s) affected

- Create Orders Table

The Orders table contains details of every order made at the Cafe such as CustomerID , OrderDate Time, StaffID and location information.

The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the SQL script for creating the Orders table. The bottom pane shows the output of the execution.

```

16 CREATE TABLE Orders (
17     OrderID VARCHAR(20) NOT NULL,
18     CustomerID VARCHAR(20) NOT NULL,
19     OrderDate DATE NOT NULL,
20     OrderTime TIME NOT NULL,
21     StaffID VARCHAR(20) NOT NULL,
22     LocationID VARCHAR(20) NOT NULL,
23     CONSTRAINT PKOrderID PRIMARY KEY (OrderID),
24     CONSTRAINT FK1 FOREIGN KEY (CustomerID)
25     REFERENCES Customer (CustomerID),
26     CONSTRAINT FK2 FOREIGN KEY (StaffID)
27     REFERENCES Staff (StaffID),
28     CONSTRAINT FK3 FOREIGN KEY (LocationID)
29     REFERENCES Location (LocationID)
30 )
  
```

#	Time	Action	Message
1	22:27:13	CREATE TABLE Orders (OrderID VARCHAR(20) NOT NULL, CustomerID...	0 row(s) affected

- Create Product Table

The Product table has details about each Product in Paris Baguette inventory. It contains product type, product description, and product price available at PB cafe .

The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the SQL script for creating the Product table. The bottom pane shows the output of the execution, including a summary of previous actions.

```

25
26
27
28 CREATE TABLE Product (
29     ProductID VARCHAR(20) NOT NULL,
30     Product_Type VARCHAR(20) NOT NULL,
31     Product_Description VARCHAR(100),
32     Product_Price BIGINT NOT NULL,
33     CONSTRAINT PKProductID PRIMARY KEY (ProductID)
34 )
35
36
  
```

#	Time	Action	Message
1	12:22:16	CREATE SCHEMA 'Cafe-Project'	1 row(s) affected
2	12:22:21	USE 'Cafe-Project'	0 row(s) affected
3	13:22:00	CREATE TABLE Customer (CustomerID VARCHAR(10) NOT NULL, C...	0 row(s) affected
4	13:23:47	CREATE TABLE Orders (OrderID VARCHAR(20) NOT NULL, Custom...	0 row(s) affected
5	13:28:15	CREATE TABLE Product (ProductID VARCHAR(20) NOT NULL, Prod...	0 row(s) affected

- Create Location Table

The Location table contains details about the location name , Phone number of each Paris Baguette Cafe .

The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the SQL script for creating the Location table:

```

35
36 CREATE TABLE Location (
37     LocationID VARCHAR(20) NOT NULL,
38     Location_Name VARCHAR(20) NOT NULL,
39     Phone_Number VARCHAR(100),
40     CONSTRAINT PKLocationID PRIMARY KEY (LocationID)
41 );
42
43
44
45
46

```

The bottom pane shows the Output window with the Action Output tab selected. It displays a log of database actions:

#	Time	Action	Message
1	12:22:16	CREATE SCHEMA 'Cafe-Project'	1 row(s) affected
2	12:22:21	USE 'Cafe-Project'	0 row(s) affected
3	13:22:00	CREATE TABLE Customer (CustomerID VARCHAR(10) NOT NULL, C...	0 row(s) affected
4	13:23:47	CREATE TABLE Orders (OrderID VARCHAR(20) NOT NULL, Custom...	0 row(s) affected
5	13:28:15	CREATE TABLE Product (ProductID VARCHAR(20) NOT NULL, Prod...	0 row(s) affected
6	13:36:37	CREATE TABLE Location (LocationID VARCHAR(20) NOT NULL, Lo...	0 row(s) affected

- Create Staff Table

The Staff table has details of each employee of PB Cafe such as Name, Position, etc.

The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the SQL script for creating the Staff table:

```

42
43 CREATE TABLE Staff (
44     StaffID VARCHAR(20) NOT NULL,
45     Staff_Name VARCHAR(50) NOT NULL,
46     Staff_Position VARCHAR(50),
47     CONSTRAINT PKStaffID PRIMARY KEY (StaffID)
48 );
49
50
51
52
53

```

The bottom pane shows the Output window with the Action Output tab selected. It displays a log of database actions:

#	Time	Action	Message
1	12:22:16	CREATE SCHEMA 'Cafe-Project'	1 row(s) affected
2	12:22:21	USE 'Cafe-Project'	0 row(s) affected
3	13:22:00	CREATE TABLE Customer (CustomerID VARCHAR(10) NOT NULL, C...	0 row(s) affected
4	13:23:47	CREATE TABLE Orders (OrderID VARCHAR(20) NOT NULL, Custom...	0 row(s) affected
5	13:28:15	CREATE TABLE Product (ProductID VARCHAR(20) NOT NULL, Prod...	0 row(s) affected
6	13:36:37	CREATE TABLE Location (LocationID VARCHAR(20) NOT NULL, Lo...	0 row(s) affected
7	13:50:18	CREATE TABLE Staff (StaffID VARCHAR(20) NOT NULL, Staff_Nam...	0 row(s) affected

- Create Order Line Table

Orderline table contains one entry per product for every order. It contains the order quantity and productID and OrderID for each order.

The screenshot shows a SQL IDE window with the following tabs: Project_Normalization, Lab1_sql, and customer. The main editor displays the SQL code for creating the Order_Line table. The code is as follows:

```

55     REFERENCES Location (LocationID)
56 );
57
58 CREATE TABLE Order_Line (
59     OrderID VARCHAR(20) NOT NULL,
60     ProductID VARCHAR(20) NOT NULL,
61     Order_Quantity BIGINT NOT NULL,
62     CONSTRAINT PKOrderLineID PRIMARY KEY (OrderID , ProductID),
63     CONSTRAINT FK4 FOREIGN KEY (OrderID)
64         REFERENCES Orders (OrderID),
65     CONSTRAINT FK5 FOREIGN KEY (ProductID)
66         REFERENCES Product (ProductID)
67 );
68
69

```

Below the editor is an "Output" window showing a log of database actions. The log is titled "Action Output" and contains the following entries:

#	Time	Action	Message
3	23:46:09	CREATE TABLE Product (ProductID VARCHAR(20) NOT NULL, Pr...	0 row(s) affected
4	23:46:16	CREATE TABLE Location (LocationID VARCHAR(20) NOT NULL, L...	0 row(s) affected
5	23:46:21	CREATE TABLE Staff (StaffID VARCHAR(20) NOT NULL, Staff_Na...	0 row(s) affected
6	23:46:59	CREATE TABLE Orders (OrderID VARCHAR(20) NOT NULL, Custo...	0 row(s) affected
7	23:47:33	CREATE TABLE Order_Line (OrderID VARCHAR(20) NOT NULL, Pr...	0 row(s) affected

3. SQL QUERIES:

3.1 Query without Joins

Using Project Schema:

```
2 • USE `cafe-project`;  
3  
4  
5  
6
```

Output




Action Output

#	Time	Action	Message
✓ 1	00:23:39	USE `cafe-project`	0 row(s) affected

1. Who are the customers that uses Paris Baguette App?

```
5 • SELECT  
6      *  
7 FROM  
8      customer  
9 WHERE  
10      App_Rewards = 'Yes';  
11
```

Result Grid

Filter Rows: Edit:   

	CustomerID	Customer_Name	App_Rewards	ToGo_Seated
▶	1	Elsa	Yes	ToGo
	10	Olaf	Yes	ToGo
	12	Rob	Yes	Seated
	13	Heather	Yes	ToGo
	16	Jeff	Yes	ToGo
	17	Justin	Yes	Seated
	2	Flynn	Yes	Seated
	20	Jim	Yes	Seated
	4	Ariel	Yes	Seated
	8	Daisy	Yes	ToGo
	9	Gaston	Yes	ToGo
✱	NULL	NULL	NULL	NULL

2. Number of customers using the App.

```
--
12  -- 2. Display the count of customers using the App
13  •  SELECT
14      COUNT(customerID) AS 'Total Customer with App'
15  FROM
16      customer
17  WHERE
18      App_Rewards = 'Yes';
19
20
21
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	Total Customer with App
▶	11

3. What are most customer preferences, take away or Dine In?

```
22
23  -- 3. What are most customer preferences, take away or Dine In?
24  •  SELECT
25      ToGo_Seated, COUNT(CustomerID) AS 'Number of Customers'
26  FROM
27      Customer
28  GROUP BY ToGo_Seated
29  ORDER BY 2;
30
31
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	ToGo_Seated	Number of Customers
▶	Seated	10
	ToGo	11

4. What are the number of available food items in each product category?

The screenshot shows a SQL query in a database tool. The query is as follows:

```
29 -- Number of items in each product category.
30
31 • SELECT
32     Product_Type, COUNT(ProductID) AS 'Number of Products'
33 FROM
34     Product
35 GROUP BY Product_Type;
36
37 -- details of product for category cake
```

The results are displayed in a table with two columns: Product_Type and Number of Products.

Product_Type	Number of Products
Coffee	5
Drink	5
Food	5
Cake	5

The bottom section shows the output of the query, indicating that 4 row(s) were returned.

5. Details of product for category 'cake' with decreasing order of price.

The screenshot shows a SQL query in a database tool. The query is as follows:

```
37 -- Details of product for category 'cake' with decreasing order of price.
38
39 • SELECT
40     *
41 FROM
42     product
43 WHERE
44     product_type = 'Cake'
45 ORDER BY Product_Price DESC;
```

The results are displayed in a table with four columns: ProductID, Product_Type, Product_Description, and Product_Price.

ProductID	Product_Type	Product_Description	Product_Price
4005	Cake	Blueberry Yogurt Cake	42
4004	Cake	Cappuccino Cake	40
4003	Cake	Green Tea Chiffon Cake	39
4001	Cake	Red Velvet Cake	38
4002	Cake	Mocha Butter Cake	35
NULL	NULL	NULL	NULL

The bottom section shows the output of the query, indicating that 5 row(s) were returned.

3.2 Query with Joins

1. What is the number of orders taken by each staff in the Café?

```
62  -- Queries with JOINS.
63
64  -- 1. Number of Orders Taken by each staff in Cafe
65  •  SELECT
66      s.staffID, staff_name, COUNT(orderID) AS 'number_of_Orders'
67  FROM
68      staff s
69      INNER JOIN
70      orders o ON s.staffID = o.staffID
71  GROUP BY s.staffID
72  ORDER BY number_of_Orders DESC;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
staffID	staff_name	number_of_Orders	
A001	Jake	3	
C002	Samantha	3	
A002	Nora	2	
A003	Riya	2	
B001	Carol	2	
B002	Sam	2	
B003	Neil	2	
B004	Jane	2	
C003	Mike	1	

2. What products Customer orders and in what quantity?

```
73
74  -- Display what products Customer orders and in what quantity.
75  •  SELECT
76      c.customer_name,
77      p.product_type,
78      p.product_description,
79      ol.order_quantity
80  FROM
81      customer c
82      INNER JOIN orders o ON c.customerID = o.customerID
83      INNER JOIN order_line ol ON o.orderID = ol.orderID
84      INNER JOIN product p ON ol.productID = p.productID;
85
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
customer_name	product_type	product_description	order_quantity
Elsa	Coffee	Cappucino	2
Elsa	Coffee	Caramel Latte	2
Elsa	Cake	Mocha Butter Cake	1
Olaf	Food	Chocolate Cream Bread	1
Olaf	Cake	Green Tea Chiffon Cake	1

3. Number of orders taken at different staff position

```
91  | - 3. Display different staff position with max number of orders
92  •  SELECT
93      s.Staff_Position, COUNT(o.orderID) AS 'Number_od_Orders'
94  FROM
95      staff s
96      INNER JOIN
97      orders o ON s.staffID = o.staffID
98  GROUP BY Staff_Position
99  ORDER BY 2;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Staff_Position	Number_od_Orders		
▶	Barista	7		
	Cashier	12		

4. What is the Average spending per customer?

```
103  -- 4. Average spending per customers
104  •  SELECT
105      avg(ol.order_quantity * p.product_price) as 'Average_Spending'
106  FROM
107      orders o
108      INNER JOIN
109      order_line ol ON o.orderID = ol.orderID
110      INNER JOIN
111      product p ON ol.productid = p.productid;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Average_Spending			
▶	14.5814			

5. What is the average spending's of customer at each Cafe location?

```
114      -- 5. Average spending by customer in each store location
115      •      SELECT
116              l.location_name,
117              AVG(ol.order_quantity * p.product_price) AS 'Average_Spending'
118      FROM
119              location l
120              LEFT JOIN
121              orders o ON l.locationID = o.locationID
122              INNER JOIN
123              order_line ol ON o.orderID = ol.orderID
124              INNER JOIN
125              product p ON ol.productid = p.productid
126      GROUP BY l.Location_Name
127      ORDER BY 'Average_Spending';
```

	location_name	Average_Spending
▶	Milpitas	18.2000
	San Jose	16.1053
	Santa Clara	5.3333

6. What is the Sales Number for different product category?

```
130      -- 6. Sales number for different category
131      •      SELECT
132              p.product_type, COUNT(o.orderID) AS 'Number_of_Orders'
133      FROM
134              product p
135              INNER JOIN
136              order_line ol ON p.productID = ol.productID
137              INNER JOIN
138              orders o ON ol.orderID = o.orderID
139      GROUP BY p.product_type
140      ORDER BY Number_of_Orders DESC;
```

	product_type	Number_of_Orders
▶	Food	13
	Coffee	12
	Cake	9
	Drink	9

7. Number of orders for each item in product list.

```

143  -- 7. Number of orders for each item in product list
144  •  SELECT
145      p.product_type,
146      p.product_description,
147      COUNT(o.orderID) AS 'Number_of_Orders'
148  FROM
149      product p LEFT JOIN order_line ol ON p.productID = ol.productID
150      INNER JOIN orders o ON ol.orderID = o.orderID
151  GROUP BY p.product_description
152  ORDER BY product_type , Number_of_Orders DESC;
153
154

```

Result Grid

	product_type	product_description	Number_of_Orders
▶	Cake	Green Tea Chiffon Cake	4
	Cake	Mocha Butter Cake	3
	Cake	Red Velvet Cake	2
	Coffee	Cappucino	3
	Coffee	Caramel Latte	3
	Coffee	Cafe Mocha	2
	Coffee	Caramel Machiato	2
	Coffee	Espresso	2
	Drink	Kiwi Smoothie	4
	Drink	Strawberry Smoothie	3
	Drink	Bubble Tea	2
	Food	Chocolate Cream Bread	4
	Food	Cheese Whirl Pastry	3
	Food	Hash Brown Bread	2
	Food	Soft Cheesecake	2
	Food	Chestnut Bread	2

8. What is the order count for different days and at different location?

```

155  -- 8. For different days what is the order count at different location
156  •  SELECT
157      location_name,
158      orderDate,
159      COUNT(orderID) AS 'Number_of_orders'
160  FROM
161      orders o
162      INNER JOIN
163      location l ON o.locationID = l.locationID
164  GROUP BY location_name , orderDate;
165
166

```

Result Grid

	location_name	orderDate	Number_of_orders
▶	Milpitas	2020-02-15	2
	Milpitas	2020-03-03	1
	Milpitas	2020-03-05	2
	Milpitas	2020-01-09	1
	Milpitas	2020-02-16	1
	San Jose	2020-02-18	1
	San Jose	2020-03-02	1
	San Jose	2020-03-04	1
	San Jose	2020-03-05	2
	San Jose	2020-02-15	2
	San Jose	2020-02-17	1
	Santa Clara	2020-03-04	1
	Santa Clara	2020-02-16	2
	Santa Clara	2020-02-18	1

9. Display customer name visiting Santa Clara location.

```
167 -- 9. Customer details visiting Santa Clara Paris Baguette Location
168 • SELECT
169     customer_name
170 FROM
171     customer c
172     INNER JOIN
173     orders o ON c.customerID = o.customerID
174     INNER JOIN
175     location l ON o.locationID = l.locationID
176 WHERE
177     l.location_name = 'Santa Clara';
178
```

<	Result Grid	Filter Rows:	Export:	Wrap Cell Content:
	customer_name			
▶	Cicilia			
	Minnie			
	Kristoff			
	Gaston			

10. What are the total sales for each month?

```
181 -- 10. What is the total sales for each month
182 • SELECT
183     MONTHNAME(o.orderDate) AS 'MonthOfOrder',
184     COUNT(o.orderID) AS '# of Orders',
185     SUM(p.product_price * ol.order_quantity) AS 'Total Sales'
186 FROM
187     orders o
188     INNER JOIN
189     order_line ol ON o.orderID = ol.orderID
190     INNER JOIN
191     product p ON ol.productID = p.productID
192 GROUP BY 1
193 ORDER BY 3;
194
```

<	Result Grid	Filter Rows:	Export:	Wrap Cell Content:
	MonthOfOrder	# of Orders	Total Sales	
▶	January	1	3	
	February	20	251	
	March	22	373	

3.3 VIEWS

1. View that list all staff members and append their work location

```
177 • CREATE VIEW Satff_Location AS
178     SELECT
179         s.staffID, s.staff_name, l.location_name
180     FROM
181         staff s
182     LEFT JOIN
183         orders o ON s.staffID = o.StaffID
184     INNER JOIN
185         location l ON o.locationID = l.locationID
186     ORDER BY staffID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	staffID	staff_name	location_name
▶	A001	Jake	Milpitas
	A002	Nora	Milpitas
	A003	Riya	Milpitas
	B001	Carol	San Jose
	B002	Sam	San Jose
	B003	Neil	San Jose
	B004	Jane	San Jose
	C002	Samantha	Santa Clara

2. Create view that displays number of TOGO orders at each location

```
194 -- Create view that displays number of TOGO orders at each location
195 • CREATE VIEW ToGo_Location AS
196     SELECT
197         l.location_name, COUNT(o.orderID) AS 'To_GO_Orders'
198     FROM
199         customer c
200     INNER JOIN
201         orders o ON c.customerID = o.customerID
202     INNER JOIN
203         location l ON o.locationID = l.locationID
204     WHERE
205         c.ToGo_Seated = 'ToGo'
206     GROUP BY location_name;
```

Limit to 1000 rows | | |

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

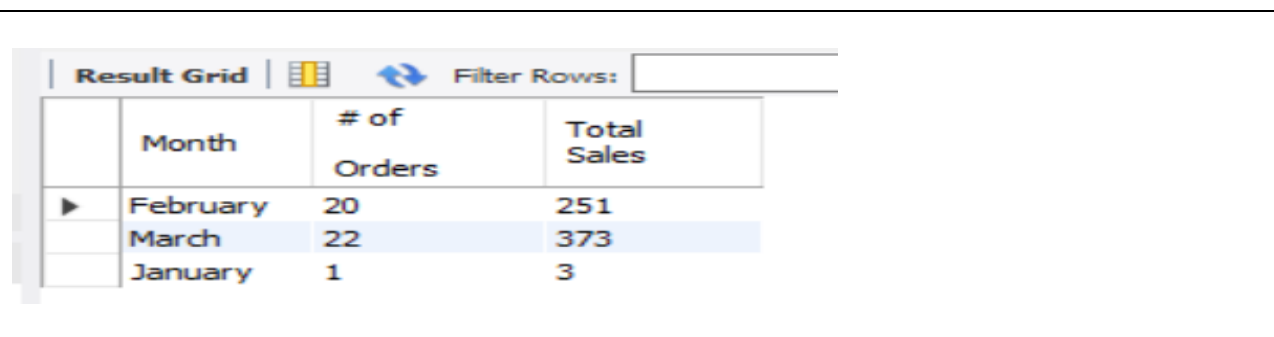
	location_name	To_GO_Orders
▶	Milpitas	2
	San Jose	2
	Santa Clara	2

4. ADVANCED SQL

4.1 Date / Case / Subquery:

-- 1. Total Sales every month (Date Function)

```
SELECT MONTHNAME(o.orderdate) AS 'Month',COUNT(o.orderid) AS '# of Orders',  
SUM(p.Product_price * ol.Order_quantity) AS 'Total Sales'  
FROM orders o INNER JOIN  
order_line ol ON o.orderID = ol.orderID INNER JOIN  
product p ON ol.productID = p.productID  
GROUP BY 1  
ORDER BY 'Total Sales';
```



The screenshot shows a 'Result Grid' window with a 'Filter Rows' input field. The grid contains the following data:

	Month	# of Orders	Total Sales
▶	February	20	251
	March	22	373
	January	1	3



-- 2. High and low Order for each location (Case)

```
SELECT l.location_Name,COUNT(o.OrderID) AS 'NumberofOrders',  
CASE  
    WHEN COUNT(o.OrderID) >= 5 THEN 'High Order Location'  
    WHEN COUNT(o.OrderID) < 5 THEN 'Low Order Location'  
END AS Category  
FROM Orders o INNER JOIN location l ON l.locationID = o.locationID  
GROUP BY location_name  
ORDER BY NumberofOrders DESC;
```

	location_Name	NumberOfOrders	Category
▶	San Jose	8	High Order Location
	Milpitas	7	High Order Location
	Santa Clara	4	Low Order Location

-- 3. Total orders day wise (Date Function)

```
SELECT DAYNAME(OrderDate) AS dayname, OrderDate, COUNT(orderID) AS 'TotalOrders'
FROM orders
GROUP BY dayname
ORDER BY OrderDate;
```

Result Grid   Filter Rows: <input type="text"/>			
	dayname	OrderDate	TotalOrders
▶	Saturday	2020-02-15	4
	Sunday	2020-02-16	3
	Tuesday	2020-02-18	3
	Monday	2020-03-02	2
	Wednesday	2020-03-04	2
	Thursday	2020-03-05	5

-- 4. Most expensive item in product list (SubQuery)

```
SELECT ProductID, Product_Type, Product_description, Product_Price
FROM Product
WHERE Product_Price = (SELECT MAX(Product_Price) FROM Product);
```

	ProductID	Product_Type	Product_description	Product_Price
▶	4005	Cake	Blueberry Yogurt Cake	42

4.2 Trigger

-- Insert Trigger –

This trigger will be initiated every time a new staff is added .

--Create table for Insert

```
CREATE TABLE StaffInsert (  
    staffID VARCHAR(30) NOT NULL,  
    staff_Name VARCHAR(30),  
    staff_position VARCHAR(30)  
);
```

--Create Trigger

```
CREATE TRIGGER StaffInsertTrigger  
AFTER INSERT ON staff FOR EACH ROW  
    INSERT INTO staffInsert SELECT staffID, staff_Name, staff_position FROM staff;
```

--Insert new staff member

```
INSERT INTO staff(staffID, staff_Name, staff_position) VALUES  
( 'A005','David','Barista');  
  
SELECT * FROM staffInsert;
```

#	Time	Action	Message
1	19:23:55	CREATE TRIGGER StaffInsertTrigger AFTER INSERT ON staff FOR EA...	0 row(s) affected
2	19:24:00	INSERT INTO staff(staffID, staff_Name, staff_position) VALUES ('A005','David...	1 row(s) affected

staffID	staff_Name	staff_position
A001	Jake	Cashier
A002	Nora	Barista
A003	Riya	Cashier
A005	David	Barista
B001	Carol	Cashier
B002	Sam	Barista
B003	Neil	Cashier
B004	Jane	Barista
C002	Samantha	Cashier
C003	Mike	Barista

#	Time	Action	Message
2	19:24:00	INSERT INTO staff(staffID, staff_Name, staff_position) VALUES ('A005','David...	1 row(s) affected
3	19:24:07	SELECT * FROM staffInsert LIMIT 0, 1000	10 row(s) returned

4.3 Stored Procedure

-- Stored Procedure

This procedure will take the location / Staff as input and it will return the total sale made by each location or by each staff.

DELIMITER //

CREATE PROCEDURE TotalSales(opt char(1))

BEGIN

if (opt = 'L') Then

SELECT l.location_name ,sum(ol.order_quantity * p.product_price) as 'total_sales' FROM
product p inner join order_line ol on ol.productID = p.productID

inner join orders o on ol.orderID = o.orderID inner join location l on o.locationId = l.locationId

group by l.location_name order by total_sales desc;

elseif (opt = 'S') Then

```
SELECT s.staff_name ,sum(ol.order_quantity * p.product_price) as 'total_sales' FROM product
p inner join order_line ol on ol.productID = p.productID
```

```
inner join orders o on ol.orderID = o.orderID inner join staff s on o.staffId = s.staffId
```

```
group by s.staff_name order by total_sales desc;
```

end if;

END //

Call TotalSales('L');

Call TotalSales('S');

Call TotalSales('L');

5	19:33:02	CREATE PROCEDURE TotalSales(opt char(1)) BEGIN if (opt = 'L') Then SEL...	0 row(s) affected
6	19:33:09	Call TotalSales('L'); Call TotalSales('S');	3 row(s) returned

	location_name	total_sales
▶	San Jose	306
	Milpitas	273
	Santa Clara	48

Call TotalSales('S');

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	staff_name	total_sales
	Jake	177
	Sam	135
	Riya	90
	Neil	81
	Jane	54
	Carol	36
	Samantha	30
	Mike	18
	Nora	6

Result 14 ×

Output

Action Output

#	Time	Action	Message
10	19:35:51	CREATE PROCEDURE TotalSales(opt char(1)) BEGIN if (opt = 'L') Then SEL...	0 row(s) affected
11	19:35:55	Call TotalSales('S');	9 row(s) returned

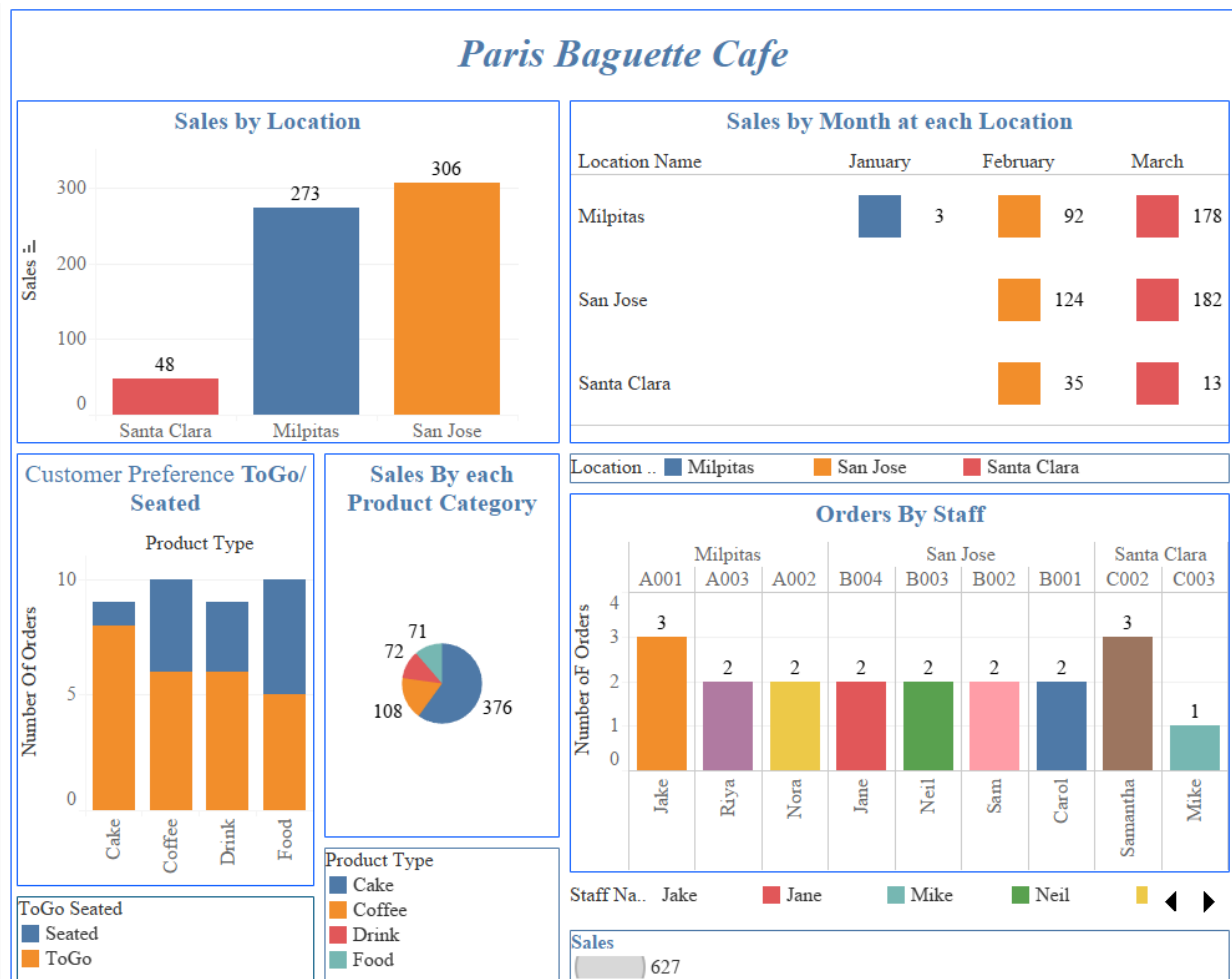
5. TABLEAU REPORT

5.1 Tableau Dashboard

The tableau dashboard created below covers the different attributes which impacts the sales.

The dashboard has the following graphs -

- Sales by each Location
- Sales by each Month at each Location
- Number of Orders for Customer Preference(Togo/Seated)
- Sales by each product Category
- Orders completed by each staff.



Location ..

■ Milpitas ■ San Jose ■ Santa Clara

Product Type

■ Cake ■ Coffee ■ Drink ■ Food

Customer Preference

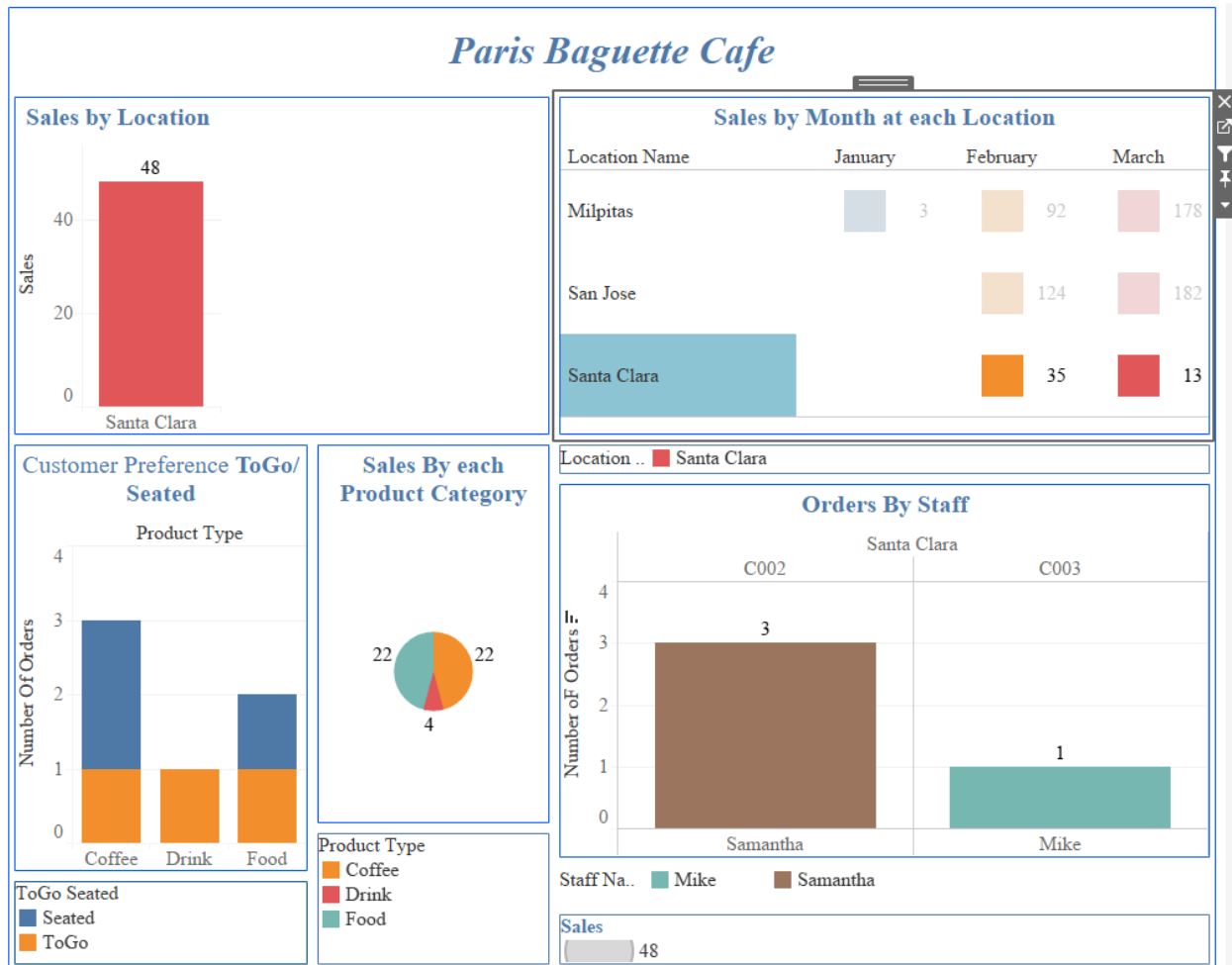
■ ToGo Seated ■ Seated ■ ToGo

Staff Na.. Jake Jane Mike Neil Sam Carol Samantha Mike

Sales 627

5.2 Tableau Report with Filter

Filter For Santa Clara Outlet: This report shows all details in dashboard for Santa Clara location.



6. CONCLUSION

This project helps us analyze various processes of Cafe management for Paris Baguette and answered few business questions with the help of SQL and Tableau.

Few Observations:

We observe Paris Baguette café have the most sales at San Jose Outlet and Santa Clara sales is the least, this information can help the café management to focus more on the sales at Santa Clara outlet.

We notice that customer prefer the takeaway option more than dining in so the management can improve their dine in experience and promote Togo offers to further increase the sales.

The dashboard created on Tableau will help the management make informed decisions on how to make the café sales more efficient and productive.