SAN DIEGO STATE UNIVERSITY

Integrated Inventory Management & Online Customer Order Processing System

MIS 686: Enterprise Data Management

Semester Project

San Diego State University

May 7, 2018

a) Team Member Names

- Vidushi Mankotia
- Pradnya Malegaonkar
- Swathika Rameshchandran
- Preethi Narayanan
- Monish Thakore

b) Problem-owner Names

Director

c) Organization Name

WS Bakers

d) Project Proposal

The project proposes to increase the revenue of WS Bakers by 18% and reduce the manual data entry errors by 15% by creating the database to track, maintain and analyze the inventory.

e) Problem Statement

WS Bakers is a bakery shop which is currently expanding its operations by opening 5 branches and its inventory is currently managed manually, which is not very efficient as it will lead to stockout and overstock situations which can harm the business adversely. WS Bakers have no data for sales projections which will lead to a situation of overstocks which will increase the carrying cost and spoilage which will result in fewer profits. It can also lead to a situation of

stockouts and theft which will result in fewer sales, fewer profits, and lower customer satisfaction and loyalty. Another problem which WS Bakers is facing is that they are currently having manual inventory management which requires more time and labor which will result in more costs and thereby fewer profits. Therefore, WS Bakers needs a database system to manage the inventory of their stores and warehouse.

f) Business Needs

- **BN1.** Expansion of a business to five new branches and absence of a database will increase the need of manual data processing and which will result in high hiring and training cost leading to high investment cost.
- **BN2.** Lower production quality is caused due to high material waste and high manufacturing time, which is turning into degraded customer satisfaction, customer loyalty and high customer churn rate which results into lower profits and market share.
- **BN3.** Lack of historical data has led to inefficient stock management which prevents WS Bakers from analyzing stockouts and overstocks beforehand. Lack of real time data analytics leads to failures in predicting market demand failing to scale their inventory up or down on time.
- **BN4.** Increase in shoplifting caused less tracking of inventory will lead to inventory losses and increase in cost and lower profits.
- **BN5.** Increase in stock outs can lead to lower sales which in turn will reduce customer satisfaction.
- **BN6.** Inaccurate inventory counts can lead to overstocking or under stocking which will result in higher spoilage and lower profits.

g) Business Requirements

- **BR1. Ref. BN1,3,5,6.** The customer orders the product from the bakery in order to fulfil their needs.
- **BR2. Ref. BN2,3,5,6.** The salesperson processes an order received from the customer in order to add up to sales.
- **BR3. Ref. BN2,4,5,6.** The product manager places product order to warehouse in order to avoid stock-outs in the local branch.
- **BR4. Ref. BN1,2,3.** The production manager places the ingredient order to the supplier to ensure that products are being manufactured without any interruption.
- **BR5. Ref. BN2.** The quality control tech returns defective ingredients to the suppliers to improve the quality and taste of the product.
- **BR6. Ref. BN5,6.** The accountant clerk audits the local branch inventory to track and identify shoplifting
- **BR7. Ref. BN5,6.** The accountant admin audits the warehouse inventory to maintain a balance between the actual stock and the stock value in the database, in order to identify the missing products and ingredients.
- **BR8. Ref. BN2,4.** The shipping clerk ships the products to the local branch from the warehouse to ensure product availability at the local branch.

h) Business Values

i. Tangible:

- **BV1. Ref. BN5.** Better management of stock shall help in reducing overstocks and stockouts by 20%

- **BV2. Ref. BN1,6.** Increase the revenue by 20% and inflation of stock market share by 10%
- **BV3. Ref. BN3.** Reduce the manual errors by 15%
- **BV4. Ref. BN3**. Reduced employee hiring and training costs by 15%

ii. Intangible:

- **BV5. Ref. BN1,2**. Increase customer loyalty and customer satisfaction leading to improved customer retention rate.

i) Constraints

- Annual maintenance cost of hardware must not go greater than $10,000
- Must preserve PII (Personal Identifiable Information) data as per state and federal compliance regulations
- Maintenance downtime of the server should not be greater than 30 minutes
- New migrated system must be up and running before 30th October 2018

j) System Requirements Specified in Use Cases

SR1. Ref BR1: While placing an order by the Customer, the Customer records,

CustomerName

CustomerAddress,

CustomerContact

System automatically generates the CustomerID and stores the entry in the Customer table in the database.

SR2. Ref BR1: While Customer placing an order, the Order records ,

COrderID,

CustomerID,

EmployeeID,

COrderDate,

PaymentType,

PaymentStatus,

Amount

System automatically generates the COrderID and stores the entry in the C_Order table in the database.

SR3. Ref BR3: While adding products to the Products table, the Product table records,

ProductID,

CategoryCode,

ProductName,

ProductPrice,

QuantityAvailable

System automatically generates the ProductID and stores the entry in the Product table in the database.

SR4. Ref BR3: While Product Manager placing an order to the supplier , the Supplier table records,

SupplierName

SupplierAddress,

SupplierContact

System automatically generates the SupplierID and stores the entry in the Supplier table in the database.

SR5. Ref BR3: While adding the different category products , the Category table records,

CategoryCode,

CategoryName

System automatically creates CategoryID and stores the entry in the Category table in the database.

SR6. Ref BR6: While adding the different branch inventory in the database , the Branch Inventory table records,

BranchID,

ProductID,

Quantity

System automatically displays the total quantity of products and stores the new entry in the Branch Inventory table in the database

k) Data Requirements Analysis

Use Case 1

A. IDENTITY SECTION

Project: WS Bakers Database

ID: UC 1-1

Use Case Name: Customer Places the order

Priority: ☑ Critical
☐ Important
☐ Nice-to-have

Actor(s): Customer, Clearing House

Trigger Event: Customer has a desire to eat

Trigger Type: ☑ External
☐ Internal

Description:

In Place-the-order aspect, Customer visits WS Bakers' website and chooses the food product from the menu displayed by the system to place an order. The order is updated in the customer order table and the sales person is notified.

B. PROCEDURAL SECTION

Preconditions:

1. The customer logged in.
2. The system must display the current menu.

Normal Course:

Procedure NC 1: Place an order	
Major Steps for This Process	**Inputs and outputs for each step**
1. System displays the menu	> Menu
2. System prompts the login page to the customer	> Login page
3.Customer logs in to the system	< Credentials
4. Customer adds food items to the cart	< Cart
5. Customer makes the payment	< Type of payment <Amount < Credit/Debit card Information Authorization status > Receipt
6. Use case ends	

Alternate Course:

Decision Branch AC 1:(From NC step 1.5) Customer selects cash on delivery option	
Major Steps	**Inputs and outputs for each step**
Customer offers cash	< Amount
System updates records of cash-on-hand	> Receipt
System provides receipt to customer	
Use case ends.	

Exceptions:

Exception E 1: (From NC step 1.5) Credit Card Declined

Major Steps	Inputs and outputs for each step
System notifies customer that card was declined	> Card Declined Message
Return to NC Step 5.	> Order Slip

Post Conditions:

1. Customer has a receipt
2. Cash drawer balances
3. Salesperson has an order slip

C. DATA SECTION

Summary of Inputs

Inputs	Sources
< Credentials	System
< Type of payment < Amount < Credit/Debit card Information < Authorization status	System, Clearing House

Summary of Outputs

Outputs	Destinations
> Login page	Customer
> Cart	Customer
> Receipt	Customer
> Card Declined Message	Customer
>Order Slip	Sales Person

Use Case 2

A. IDENTITY SECTION

Project: WS Bakers Database

ID: UC 1-2

Use Case Name: Salesperson Processes the order

Priority: ☑ Critical
☐ Important
☐ Nice-to-have

Actor(s): Salesperson

Trigger Event: Customer has placed an order

Trigger Type: ☑ External
☐ Internal

Description:

In process-the-order aspect, Salesperson checks the availability of the product. If the product is available, salesperson processes the order received from the customer in the system. If it is not available, then Salesperson reports to the product manager. The completed order detailed are updated in the customer order table and the customer receives the products.

B. PROCEDURAL SECTION

Preconditions:

1. The salesperson must be logged in to the system to process the order.
2. The order must be placed by the customer.

Normal Course:

Procedure NC 1: Salesperson processes the order	
Major Steps for This Process	**Inputs and outputs for each step**
1. Salesperson checks the availability of the product in the system	> Available Product List
2. The order is handled by the salesperson if the product is available.	
3. Use case ends	

Alternate Course:

Decision Branch AC 1:(From NC step 1.2) Product is not available	
Major Steps	**Inputs and outputs for each step**
Salesperson reports to the Product Manager if the product is not available.	>Stockout Notice
2. Use case ends.	

Exceptions:

Exception E 1: (From NC step 1.2) Wrong product ID is entered by the salesperson	
Major Steps	**Inputs and outputs for each step**
System prompts an error of invalid entry	> Invalid Data Notification

Post Conditions:

1. Product availability is recorded in the database

C. DATA SECTION

Summary of Inputs

Inputs	Sources
< Order Details	System

Summary of Outputs

Outputs	Destinations
> Available Product List	System

Use Case 3

A. IDENTITY SECTION

Project: WS Bakers Database

ID: UC 1-3

Use Case Name: Production manager placing the ingredient order to the supplier

Priority: ☑ Critical
☐ Important
☐ Nice-to-have

Actor(s): Salesperson

Trigger Event: Salesperson informs the production manager about the non-availability of a product.

Trigger Type: ☑ External
☐ Internal

Description:

In place-the-order aspect, Production manager places the ingredient order to the Supplier in the case of non-availability of product. Supplier receives the order. Supplier sends the ordered ingredients to the warehouse after which it is received by the local branch. The order details are updated in the supplier order table.

B. PROCEDURAL SECTION

Preconditions:

1. The ingredient is not available in the local branch.
2. Salesperson informs supplier about lack of a particular ingredient.

Normal Course:

Procedure NC 1: Production manager places the ingredient order to the supplier	
Major Steps for This Process	**Inputs and outputs for each step**
1. Production manager receives the ingredient order from the Salesperson	< Order Details
2. Production manager places the ingredient order to the supplier	> Ingredient list
3. The supplier receives the order and processes it	
4. Supplier send the ingredient order to the warehouse	<Ingredient list updated
5. The order is received by the local branch through the shipping clerk.	>System updated
6.Use Case ends	

Alternate Course:

Decision Branch AC 1:(From NC step 1.5)The order is received by some other local branch	
Major Steps	**Inputs and outputs for each step**
1.Return to NC2	< Available Ingredient List
2. Follow all the steps and Use case ends.	

Exceptions:

Exception E 1: (From NC step 1.2) Wrong product ID is entered by the production manager	
Major Steps	**Inputs and outputs for each step**
System prompts an error of invalid entry	> Invalid Data Notification

Post Conditions:

1. Product availability of the branch and warehouse is recorded in the database

C. DATA SECTION

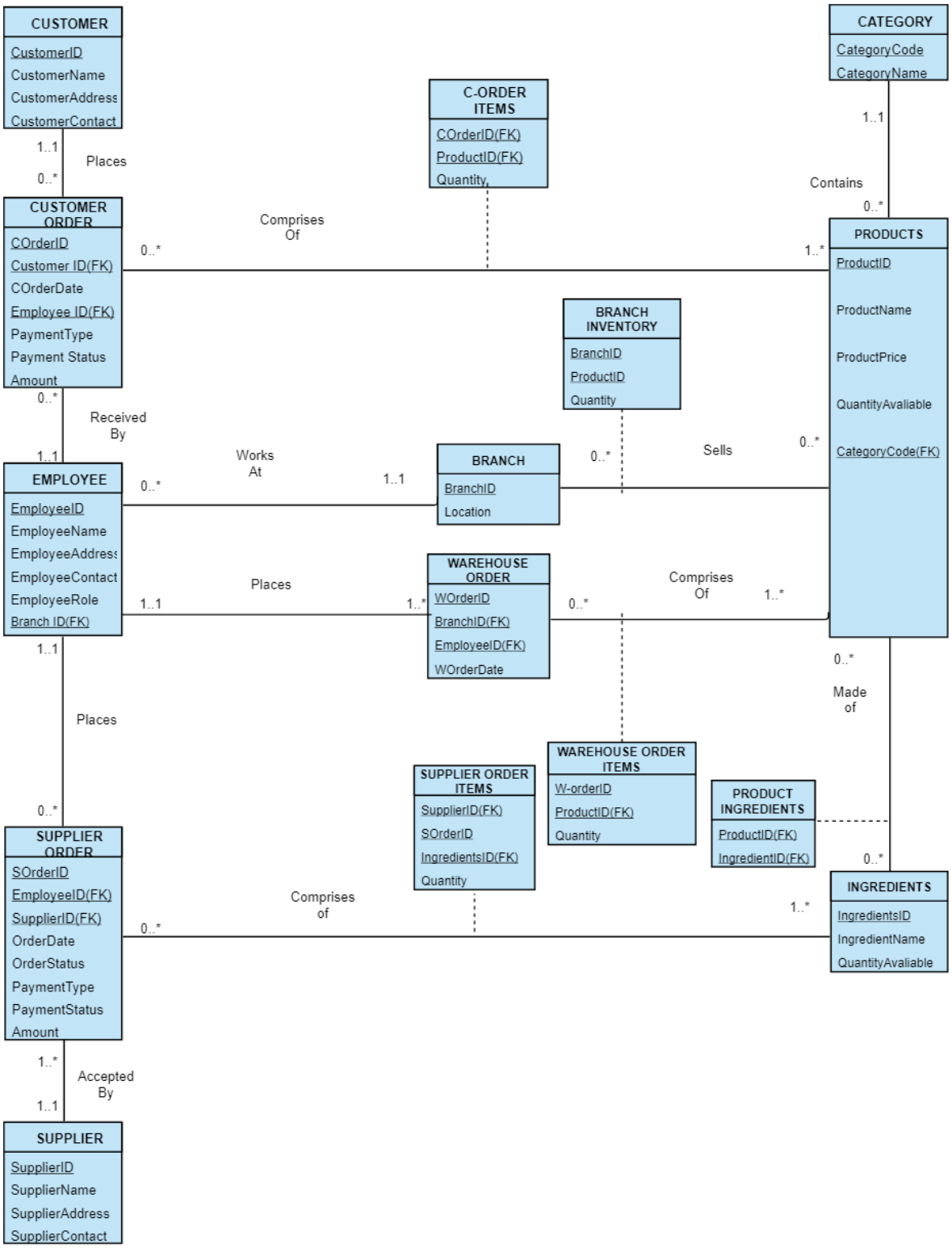
Summary of Inputs

Inputs	**Sources**
> Order Details	System
>Ingredient List	System

Summary of Outputs

Outputs	**Destinations**
> Available Product List	System

l) Data model


CUSTOMER
CustomerID
CustomerName
CustomerAddress
CustomerContact
1..1
Places
0..*
CUSTOMER ORDER
COrderID
Customer ID(FK)
COrderDate
Employee ID(FK)
PaymentType
Payment Status
Amount
0..*
Received By
1..1
EMPLOYEE
EmployeeID
EmployeeName
EmployeeAddress
EmployeeContact
EmployeeRole
Branch ID(FK)
1..1
Places
0..*
SUPPLIER ORDER
SOrderID
EmployeeID(FK)
SupplierID(FK)
OrderDate
OrderStatus
PaymentType
PaymentStatus
Amount
1..*
Accepted By
1..1
SUPPLIER
SupplierID
SupplierName
SupplierAddress
SupplierContact
Comprises Of
0..*
C-ORDER ITEMS
COrderID(FK)
ProductID(FK)
Quantity
Works At
0..*
1..1
BRANCH
BranchID
Location
Places
1..1
1..*
WAREHOUSE ORDER
WOrderID
BranchID(FK)
EmployeeID(FK)
WOrderDate
Comprises of
0..*
SUPPLIER ORDER ITEMS
SupplierID(FK)
SOrderID
IngredientsID(FK)
Quantity
BRANCH INVENTORY
BranchID
ProductID
Quantity
0..*
Sells
0..*
WAREHOUSE ORDER ITEMS
W-orderID
ProductID(FK)
Quantity
Comprises Of
0..*
1..*
CATEGORY
CategoryCode
CategoryName
1..1
Contains
0..*
1..*
PRODUCTS
ProductID
ProductName
ProductPrice
QuantityAvaliable
CategoryCode(FK)
0..*
Made of
PRODUCT INGREDIENTS
ProductID(FK)
IngredientID(FK)
0..*
1..*
INGREDIENTS
IngredientsID
IngredientName
QuantityAvaliable

m) Logical Database Design

TBLCUSTOMER (CustomerID,

CustomerName not null,

CustomerAddress, CustomerContact)

TBLBRANCH (BranchID,

BranchLocation not null)

TBLEMPLOYEE (EmployeeID,

Branch ID not null,

EmployeeName not null,

EmployeeAddress, EmployeeContact, EmployeeRole)

foreign key BranchID references TBLBRANCH

TBLC_ORDER (COrderID,

CustomerID not null,

EmployeeID not null,

COrderDate, PaymentType, PaymentStatus,

Amount not null)

foreign key CustomerID references TBLCUSTOMER

foreign key EmployeeID references TBLEMPLOYEE

TBLCATEGORY (CategoryCode,

CategoryName not null)

TBLPRODUCTS (ProductID,

CategoryCode not null,

ProductName not null,

ProductPrice not null,

QuantityAvailable)

foreign key CategoryCode references TBLCATEGORY

TBLINGREDIENTS (IngredientID,

IngredientName not null,

QuantityAvailable)

TBLBRANCHINVENTORY (BranchID, ProductID, Quantity)

foreign key BranchID references TBLBRANCH not null

foreign key ProductID references TBLPRODUCTS not null

TBLW_ORDER (WOrderID,

EmployeeID not null,

WOrderData not null)

foreign key EmployeeID references TBLEMPLOYEE

TBLW_ORDERITEMS (WOrderID, ProductID, Quantity)

foreign key WOrderID references TBLW_ORDER not null

foreign key ProductID references TBLPRODUCTS not null

TBLSUPPLIER (SupplierID,

SupplierName not null,

SupplierAddress,

SupplierContact not null)

TBLS_ORDER (SOrderID,

EmployeeID not null,

SupplierID not null,

OrderDate not null,

OrderStatus not null,

PaymentType not null,

PaymentStatus not null,

AmountBilled)

foreign key SupplierID references TBLSUPPLIER

foreign key EmployeeID references TBLEMPLOYEE

TBLS_ORDERITEMS (SOrderID, IngredientID, Quantity)

foreign key SOrderID references TBLS_ORDER not null

foreign key IngredientID references TBLINGREDIENT not null

TBLPRODUCTINGREDIENTS (ProductID, IngredientID)

foreign key ProductID references TBLPRODUCTS not null

foreign key IngredientID references TBLINGREDIENT not null

TBLC_ORDERITEMS (COrderID, ProductID, Quantity)

foreign key ProductID references TBLPRODUCTS not null

foreign key COrderID references TBLC_ORDER not null

n) Data Dictionary

Data Dictionary

/* This script demonstrates viewing metadata for a database */

-- Get a list of all databases on your SQL Server

```
USE master
GO
SELECT * FROM sysdatabases;
GO
-- Get a list of all tables in your database

USE WS_Inventory;
GO

SELECT * From information_schema.tables;
GO

--Get a list of all columns and their properties from your database

SELECT * FROM information_schema.columns;
GO

-- Get a list of all columns and their properties for a specific table in your database

SELECT * FROM information_schema.columns
WHERE Table_Name = 'tblCustomer';
GO
```

```
SELECT * FROM information_schema.columns
WHERE Table_Name = 'tblBranch';
GO

SELECT * FROM information_schema.columns
WHERE Table_Name = 'tblSuppliers';
GO

SELECT * FROM information_schema.columns
WHERE Table_Name = 'tblEmployee';
GO

SELECT * FROM information_schema.columns
WHERE Table_Name = 'tblC_Order';
GO

SELECT * FROM information_schema.columns
WHERE Table_Name = 'tblCategory';
GO

SELECT * FROM information_schema.columns
WHERE Table_Name = 'tblProducts';
GO

SELECT * FROM information_schema.columns
WHERE Table_Name = 'tblIngredients';
GO
```

```
SELECT * FROM information_schema.columns
WHERE Table_Name = 'tblBranchIngredients';
GO

SELECT * FROM information_schema.columns
WHERE Table_Name = 'tblWarehouse';
GO
```

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
WS_Inventory	dbo	TblCustomer	BASE TABLE
WS_Inventory	dbo	TblEmployee	BASE TABLE
WS_Inventory	dbo	TblSupplier	BASE TABLE
WS_Inventory	dbo	TblBranch	BASE TABLE
WS_Inventory	dbo	TblProducts	BASE TABLE
WS_Inventory	dbo	TblIngredients	BASE TABLE
WS_Inventory	dbo	TblCategory	BASE TABLE
WS_Inventory	dbo	TblW_Order	BASE TABLE
WS_Inventory	dbo	TblS_Order	BASE TABLE
WS_Inventory	dbo	TblC_Order	BASE TABLE
WS_Inventory	dbo	TblBranchInventory	BASE TABLE
WS_Inventory	dbo	TblW_OrderItems	BASE TABLE
WS_Inventory	dbo	TblC_OrderItems	BASE TABLE

WS_Inventory	dbo	TblS_OrderItems	BASE TABLE
WS_Inventory	dbo	TblWarehouse	BASE TABLE

TABLE_NAME	COLUMN_NAME	ORDINAL_POSITION	IS_NULLABLE	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH
TblCustomer	CustomerID	1	NO	varchar	20
TblCustomer	CustomerName	2	NO	varchar	64
TblCustomer	CustomerAddress	3	YES	varchar	128
TblCustomer	CustomerContact	4	YES	varchar	32
TblBranch	BranchID	1	NO	varchar	20
TblBranch	BranchLocation	2	NO	varchar	64
TblEmployee	EmployeeID	1	NO	varchar	20
TblEmployee	BranchID	2	YES	varchar	20
TblEmployee	EmployeeName	3	NO	varchar	64
TblEmployee	EmployeeAddress	4	YES	varchar	128
TblEmployee	EmployeeContact	5	YES	varchar	32
TblEmployee	EmployeeRole	6	YES	char	20

TblC_Order	COrderID	1	NO	varchar	20
TblC_Order	CustomerID	2	NO	varchar	20
TblC_Order	EmployeeID	3	NO	varchar	20
TblC_Order	COrderDate	4	YES	date	NULL
TblC_Order	PaymentType	5	YES	varchar	10
TblC_Order	PaymentStatus	6	YES	varchar	10
TblC_Order	Amount	7	NO	float	NULL
TblCategory	CategoryCode	1	NO	varchar	20
TblCategory	CategoryName	2	NO	varchar	64
TblProducts	ProductID	1	NO	varchar	20
TblProducts	CategoryCode	2	NO	varchar	20
TblProducts	ProductName	3	NO	varchar	128
TblProducts	ProductPrice	4	NO	int	NULL
TblProducts	QuantityAvailable	5	YES	int	NULL
TblIngredients	IngredientID	1	NO	varchar	20
TblIngredients	Ingredient Name	2	NO	varchar	60
TblIngredient	QuantityAv	3	YES	int	NULL

s	ailable				
TblBranchInventory	BranchID	1	NO	varchar	20
TblBranchInventory	ProductID	2	NO	varchar	20
TblBranchInventory	Quantity	3	YES	int	NULL
TblW_Order	WOrderID	1	NO	varchar	20
TblW_Order	EmployeeID	2	NO	varchar	20
TblW_Order	WOrderData	3	NO	date	NULL
TblW_OrderItems	WOrderID	1	NO	varchar	20
TblW_OrderItems	ProductID	2	NO	varchar	20
TblW_OrderItems	Quantity	3	YES	int	NULL
TblSupplier	SupplierID	1	NO	varchar	20
TblSupplier	SupplierName	2	NO	varchar	50
TblSupplier	SupplierAddress	3	YES	varchar	128
TblSupplier	SupplierContact	4	NO	varchar	68
TblS_Order	SOrderID	1	NO	varchar	20
TblS_Order	EmployeeID	2	NO	varchar	20
TblS_Order	SupplierID	3	NO	varchar	20
TblS_Order	OrderDate	4	NO	date	NULL

TblS_Order	OrderStatus	5	NO	varchar	20
TblS_Order	PaymentType	6	NO	varchar	30
TblS_Order	PaymentStatus	7	NO	varchar	30
TblS_Order	AmountBilled	8	YES	float	NULL
TblS_OrderItems	SOrderID	1	NO	varchar	20
TblS_OrderItems	IngredientID	2	NO	varchar	20
TblS_OrderItems	Quantity	3	YES	int	NULL
TblProductIngredients	ProductID	1	NO	varchar	20
TblProductIngredients	IngredientID	2	NO	varchar	20
TblC_OrderItems	COrderID	1	NO	varchar	20
TblC_OrderItems	ProductID	2	NO	varchar	20
TblC_OrderItems	Quantity	3	YES	int	NULL

o) Query Dictionary

Query #	Reference	Query Description	SQL
Q1	**SR01**	LIST OF ALL CUSTOMERS WHO HAVE BOUGHT PRODUCTS FOR MORE THAN $100 OF VALUE IN 2017 BUT NOT BOUGHT AT ALL IN 2018	SELECT a.CustomerID, c.CustomerName FROM TblC_Order AS a JOIN TblCustomer AS c ON a.CustomerID = c.CustomerID WHERE a.COrderDate > '2017-01-01' AND a.COrderDate< '2017-12-31' AND a.CustomerID NOT IN(SELECT TblC_Order.CustomerID FROM TblC_Order WHERE TblC_Order.COrderDate > '2018-01-01') GROUP BY a.CustomerID, c.CustomerName HAVING SUM(a.Amount) > =100;
Q2	**SR01**	WHICH CUSTOMERS HAVE PLACED AN ORDER IN MAY 2018?	SELECT o.CustomerID, c.CustomerName FROM TblC_Order AS o JOIN TblCustomer AS c ON o.CustomerID =c.CustomerID WHERE o.COrderDate >= '2018-05-01' AND o.COrderDate <= '2018-05-30';
Q3	**SR03**	PRODUCT MANAGER CHECKS FOR THE PRODUCTS' AVAILABILITY AT DIFFERENT BRANCHES TO PLACE THE WAREHOUSE ORDERS SORTED PRIORITY-WISE	SELECT bi.BranchID, bi.ProductID, p.ProductName, bi.Quantity FROM tblBranchInventory AS bi JOIN tblProducts AS p ON bi.ProductID = p.ProductID WHERE bi.Quantity < 10

			ORDER BY bi.Quantity ASC;
Q4	**SR05,** **SR03**	CHECK FOR THE MOST POPULAR CATEGORY BEING SOLD AMONGST THE CUSTOMERS	SELECT ct.CategoryName, SUM(oi.Quantity) AS TotalQuantitySold FROM TblCategory AS ct JOIN TblProducts AS pr ON pr.CategoryCode = ct.CategoryCode JOIN TblC_OrderItems AS oi ON pr.ProductID = oi.ProductID GROUP BY ct.CategoryName ORDER BY TotalQuantitySold DESC;
Q5	**SR01**	DISPLAY THE CUSTOMERS WHOSE AVERAGE AMOUNT SPENT IS MORE THAN THE TOTAL AVERAGE AMOUNT SPENT	SELECT oc.CustomerID, c.CustomerName, (SELECT ROUND (AVG(ic.Amount), 2) AS avgamt FROM TblC_Order AS ic WHERE ic.CustomerID = oc.CustomerID) AS CustAvgAmt FROM TblC_Order AS oc JOIN TblCustomer AS c ON c.CustomerID = oc.CustomerID WHERE oc.Amount > (SELECT AVG(amount) FROM TblC_Order)

			GROUP BY oc.CustomerID, c.CustomerName;
Q6	**SR06,** **SR03**	DISPLAY THE MOST POPULAR PRODUCT SOLD AT EVERY BRANCH WITH THE TOTAL QUANTITIES SOLD	SELECT tab1.BranchID, tab1.TotalQtySold, tab2.ProductName FROM (SELECT BranchID, MAX(TotalQuantitySold) AS TotalQtySold FROM (SELECT b.BranchID, coi.ProductID, t.ProductName, SUM(coi.Quantity) AS TotalQuantitySold FROM TblC_OrderItems AS coi JOIN TblProducts AS t ON t.ProductID = coi.ProductID JOIN TblC_Order AS o ON o.COrderID = coi.COrderID JOIN TblEmployee AS e ON e.EmployeeID = o.EmployeeID JOIN TblBranch AS b ON b.BranchID = e.BranchID GROUP BY coi.ProductID, t.ProductName, b.BranchID) TempT1 GROUP BY BranchID) tab1

			INNER JOIN (SELECT b.BranchID, coi.ProductID, t.ProductName, SUM(coi.Quantity) AS TotalQuantitySold FROM TblC_OrderItems AS coi JOIN TblProducts AS t ON t.ProductID = coi.ProductID JOIN TblC_Order AS o ON o.COrderID = coi.COrderID JOIN TblEmployee AS e ON e.EmployeeID = o.EmployeeID JOIN TblBranch AS b ON b.BranchID = e.BranchID GROUP BY coi.ProductID, t.ProductName, b.BranchID)tab2 ON tab1.BranchID = tab2.BranchID AND tab1.TotalQtySold = tab2.TotalQuantitySold;
Q7	**SR02**	DISPLAY THE CART ITEMS FOR A PARTICULAR ORDER - FOR EXAMPLE ORDER ID = C007	SELECT p.ProductID, p.ProductName, o.Quantity, p.ProductPrice, (o.Quantity*p.ProductPrice) AS TotalPrice FROM TblProducts AS p JOIN TblC_OrderItems AS o ON

			o.ProductID = p.ProductID JOIN TblC_Order AS co ON co.COrderID = o.COrderID WHERE o.COrderID = 'C007';
Q8	**SR03 ,** **SR05**	DISPLAY THE MENU ITEMS BASED ON THE FILTERS APPLIED BY THE CUSTOMERS	SELECT p.ProductName, p.ProductPrice FROM TblProducts AS p WHERE p.CategoryCode = 'C01' AND p.ProductPrice < 20 ORDER BY ProductName ASC;
Q9	**SR03,SR0** **6**	DISPLAY THE QUANTITY OF AVAILABLE PRODUCTS AT A PARTICULAR BRANCH	SELECT bi.BranchID, bi.ProductID, p.ProductName, bi.Quantity FROM TblBranchInventory AS bi JOIN TblProducts AS p ON bi.ProductID = p.ProductID WHERE bi.Quantity > 0;
Q10	**SR04**	IDENTIFY THE SUPPLIER ID FOR THE INGREDIENTS WHICH ARE FALLING BELOW QUANTITY 60	SELECT TblS_Order.SupplierID, TblS_Order.SOrderID, TblS_OrderItems.IngredientID FROM TblS_Order JOIN (SELECT i.IngredientID, TblS_OrderItems.SOrderID FROM TblIngredients AS i INNER JOIN TblS_OrderItems ON i.IngredientID = TblS_OrderItems.IngredientID SWHERE i.QuantityAvailable < 60) AS tab ON tab.SOrderID =

			TblS_Order.SOrderID JOIN TblS_OrderItems ON TblS_OrderItems.IngredientID = tab.IngredientID;

p) SQL

SQL script file is zipped along with this report.