*All full PDFs of ERD/RDM/Data Dictionary are attached as full files

Austin Dickieson, Aran O'Brien

Project Proposal - Annie's

Introduction:

Annie's Sandwiches is a Vietnamese sandwich restaurant on Park Avenue in Santa Clara University right outside of Santa Clara University. It is popular among the university students because it is family owned and operated, a cheaper alternative to nearby sandwich restaurants such as Ike's, and within walking distance for most students. With this project we will create a database that looks at the relationships between the customers and orders, ingredients in the orders and how the employees relate to these entities.

Our database will show relationships between entities in the business such as which employees made sandwiches, which sandwiches, smoothies, and drinks customers order, and what ingredients go into an order. It will also show how many orders employees make and how many orders a customer has purchased.

The reason this database would help the organization is that Annie's has a current loyalty card and discount system that is likely difficult to keep track of when accounting for cost and accuracy. An updated customer loyalty system with a database would be a stronger replacement. It would also allow for a better management of the resources such as ingredients and appliances and how the employees use these resources as well as what menu items need these resources. Finally, a developed database would allow for Annie's to see popular sandwiches and better understand the busy hours.

Entities/Table Outline:

Regular Entities:

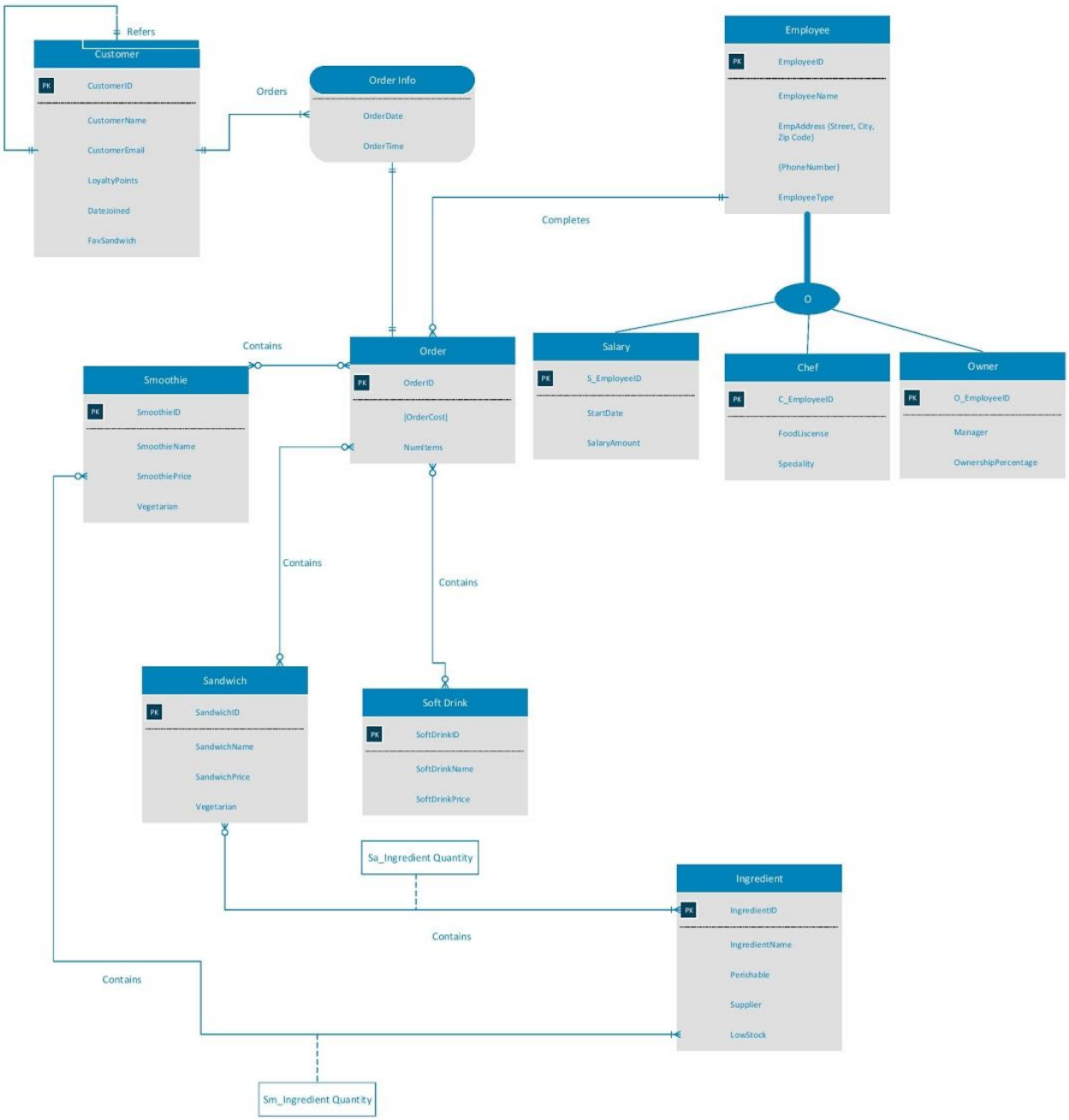
- CUSTOMERS - <u>Customer ID</u>, CustomerName, CustomerEmail, LoyaltyPoints, DateJoined, FavSandwich
- EMPLOYEE - EmployeeID, EmployeeName, EmpStreet, EmpCity, EmpZipCode, EmployeeType
    - OWNER - O_EmployeeID, Manager, Ownership%
    - CHEF - C_EmployeeID, FoodLicense, Speciality
    - SALARY - S_EmployeeID, StartDate, SalaryAmount
- ORDER - <u>OrderID</u>, CustomerID, EmployeeID, OrderCost, NumItems
- SANDWICH - <u>Sandwich ID</u>, SandwichName, SandwichPrice, Vegetarian
- INGREDIENT - <u>IngredientID</u>, IngredientName, Perishable, Supplier, LowStock
- SMOOTHIE - <u>Smoothie ID</u>, SmoothieName, SmoothiePrice, Vegetarian
- SOFT DRINK- <u>SoftDrinkID,</u> SoftDrinkName, SoftDrinkPrice

Associative Attributes: EMPLOYEE_PHONE_NUMBER, SMOOTHIE_ORDER, SANDWICH_ORDER, SOFTDRINK_ORDER, SANDWICH_INGREDIENT, SMOOTHIE_INGREDIENT

Relationship Entity: ORDER_INFO

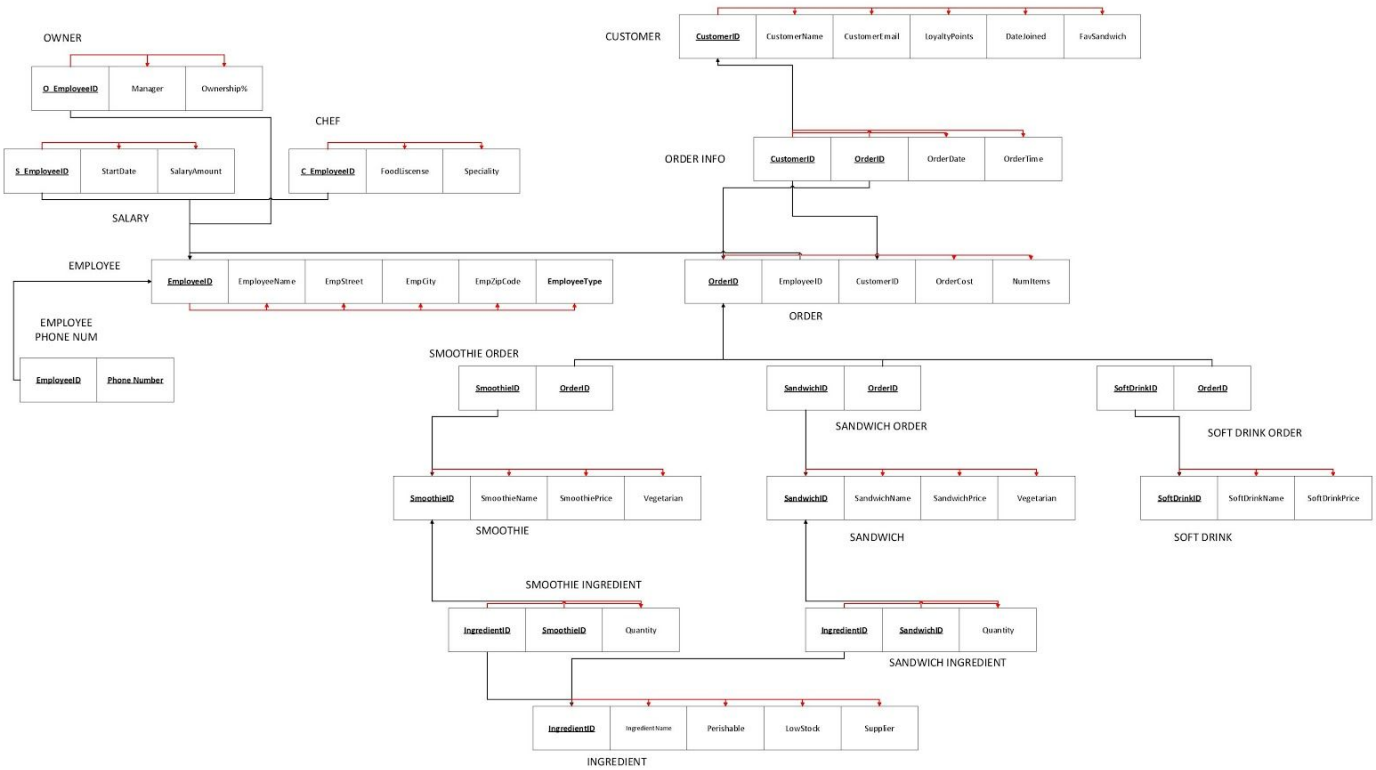
*All full PDFs of ERD/RDM/Data Dictionary are attached as full files

Annie's Sandwiches ERD – Austin Dickieson, Aran O'Brian



*All full PDFs of ERD/RDM/Data Dictionary are attached as full files

Annie's Sandwiches RDM – Austin Dickieson, Aran O'Brian



*All full PDFs of ERD/RDM/Data Dictionary are attached as full files

Annie's Sandwhiches Data Dictionary

Austin Dickieson, Aran O'Brian

CUSTOMER

Name	Data Type	Constraints	Key
CustomerID	bigint	>0, NOT NULL	PK
CustomerName	nvarchar(50)	NOT NULL	
CustomerEmail	nvarchar(50)		
Loyalty Points	int		
Date Joined	date	NOT NULL	
FavSandwich	nvarchar(20)		

EMPLOYEE

Name	Data Type	Constraints	Key
EmployeeID	bigint	>0, NOT NULL	PK
EmployeeName	nvarchar(50)	NOT NULL	
EmpStreet	nvarchar(50)		
EmpCity	nvarchar(20)		
EmpZipCode	bigint		
EmployeeType	char	NOT NULL	

EMPLOYEE PHONE NUMBER

Name	Data Type	Constraints	Key
Phone_Number	bigint	NOT NULL	PK
EmployeeID	bigint	>0, NOT NULL	FK

OWNER

Name	Data Type	Constraints	Key
O_EmployeeID	bigint	NOT NULL	PK/FK
Manager	bool	NOT NULL	
Ownership%	int	>0 AND <=100	

CHEF

Name	Data Type	Constraints	Key
C_EmployeeID	bigint	NOT NULL	PK/FK
FoodLiscense	bool	NOT NULL	
Speciality	nvarchar(20)		

SALARY

Name	Data Type	Constraints	Key
S_EmployeeID	bigint	NOT NULL	PK/FK
StartDate	date	NOT NULL	
SalaryAmount	bigint	NOT NULL	

*All full PDFs of ERD/RDM/Data Dictionary are attached as full files

ORDER

Name	Data Type	Constraints	Key
OrderID	bigint	>0, NOT NULL	PK
CustomerID	bigint	>0, NOT NULL	FK
EmployeeID	bigint	>0, NOT NULL	FK
OrderCost	float	NOT NULL	
NumItems	int	>0, NOT NULL	

ORDER INFO

Name	Data Type	Constraints	Key
CustomerID	bigint	>0, NOT NULL	PK/FK
OrderID	bigint	>0, NOT NULL	PK/FK
OrderDate	date	NOT NULL	
OrderTime	time	NOT NULL	

SMOOTHIE ORDER

Name	Data Type	Constraints	Key
SmoothieID	bigint	>0, NOT NULL	PK/FK
OrderID	bigint	>0, NOT NULL	PK/FK

SANDWICH ORDER

Name	Data Type	Constraints	Key
SandwichID	bigint	>0, NOT NULL	PK/FK
OrderID	bigint	>0, NOT NULL	PK/FK

SOFTDRINK ORDER

Name	Data Type	Constraints	Key
SoftDrinkID	bigint	>0, NOT NULL	PK/FK
OrderID	bigint	>0, NOT NULL	PK/FK

SMOOTHIE

Name	Data Type	Constraints	Key
SmoothieID	bigint	>0, NOT NULL	PK
SmoothieName	nvarchar(10)	NOT NULL	
SmoothiePrice	float	NOT NULL	
Vegetarian	bool	NOT NULL	

SANDWICH

Name	Data Type	Constraints	Key
SandwichID	bigint	>0, NOT NULL	PK
SandwichName	nvarchar(10)	NOT NULL	
SandwichPrice	float	NOT NULL	

*All full PDFs of ERD/RDM/Data Dictionary are attached as full files

Vegetarian	bool	NOT NULL	

SOFT DRINK

Name	Data Type	Constraints	Key
SoftDrinkID	bigint	>0, NOT NULL	PK
SofDrinkName	nvarchar(10)	NOT NULL	
SoftDrinkPrice	float	NOT NULL	

INGREDIENT

Name	Data Type	Constraints	Key
IngredientID	bigint	>0, NOT NULL	PK
ingredientName	nvarchar(20)	NOT NULL	
Perishable	bool	NOT NULL	
LowStock	bool	NOT NULL	
Supplier	nvarchar(30)		

SMOOTHIE INGREDIENT

SmoothieID	bigint	>0, NOT NULL	PK/FK
IngredientID	bigint	>0, NOT NULL	PK/FK
Quantity	int	NOT NULL	

SANDWICH INGREDIENT

SandwichID	bigint	>0, NOT NULL	PK/FK
IngredientID	bigint	>0, NOT NULL	PK/FK
Quantity	int	NOT NULL	

*Descriptions and Example values are in attached Data Dictionary File

*All full PDFs of ERD/RDM/Data Dictionary are attached as full files

SQL - TABLE CREATION AND VIEWS (INSERTIONS IN SQL DATABASE FILE)

```
--AUSTIN DICKIESON, ARAN O'BRIEN
--ANNIES SANDWICHES SQL DATABASE

--CREATE ALL TABLES
CREATE TABLE Customer
        (CustomerID         BIGINT    NOT NULL,
                        CustomerName        NVARCHAR(50)     NOT NULL,
                CustomerEmail       NVARCHAR(50),
         LoyaltyPoints        Int,
        DateJoined           Date            Not Null,
        FavSandwich         NVARCHAR(20),
CONSTRAINT Customer_PK PRIMARY KEY (CustomerID));

CREATE TABLE Employee
        (EmployeeID         BIGINT     NOT NULL,
                        EmployeeName        NVARCHAR(50)     NOT NULL,
                EmpStreet       NVARCHAR(50),
                        EmpCity        NVARCHAR(20),
         EmpZipCode        BIGINT,
                        EmployeeType     CHAR              Not Null,
CONSTRAINT Employee_PK PRIMARY KEY (EmployeeID));

CREATE TABLE Employee_Phone_Number
        (Phone_Number          BIGINT     NOT NULL,
                        EmployeeID          BIGINT     NOT NULL,
CONSTRAINT Employee_Phone_Number_PK PRIMARY KEY (Phone_Number),
CONSTRAINT Employee_Phone_Number_FK1 FOREIGN KEY (EmployeeID) REFERENCES
Employee(EmployeeID));
```

*All full PDFs of ERD/RDM/Data Dictionary are attached as full files

```
CREATE TABLE Owner_
        (O_EmployeeID       BIGINT    NOT NULL,
                        Manager          BIT          NOT NULL,
                        Ownership_Percentage Int,
CONSTRAINT Owner_PK PRIMARY KEY (O_EmployeeID),
CONSTRAINT Owner_FK1 FOREIGN KEY (O_EmployeeID) REFERENCES
Employee(EmployeeID));

CREATE TABLE Chef
        (C_EmployeeID       BIGINT    NOT NULL,
                        FoodLicense          BIT          NOT NULL,
                        Specialty          nvarchar(20),
CONSTRAINT Chef_PK PRIMARY KEY (C_EmployeeID),
CONSTRAINT Chef_FK1 FOREIGN KEY (C_EmployeeID) REFERENCES
Employee(EmployeeID));

CREATE TABLE Salary
        (S_EmployeeID       BIGINT    NOT NULL,
                        StartDate          Date          NOT NULL,
                        SalaryAmount       BigInt    Not Null,
CONSTRAINT Salary_PK PRIMARY KEY (S_EmployeeID),
CONSTRAINT Salary_FK1 FOREIGN KEY (S_EmployeeID) REFERENCES
Employee(EmployeeID));

CREATE TABLE Order_
        (OrderID       BIGINT    NOT NULL,
                        CustomerID      BIGINT   NOT NULL,
                        EmployeeID       BigInt    Not Null,
CONSTRAINT Order_PK PRIMARY KEY (OrderID),
```

```
CONSTRAINT Order_FK1 FOREIGN KEY (CustomerID) REFERENCES
Customer(CustomerID),

Constraint Order_FK2 Foreign Key (EmployeeID) References Employee(EmployeeID));


CREATE TABLE Order_Info

        (OrderID        BIGINT    NOT NULL,

                        CustomerID      BIGINT    NOT NULL,

                        OrderDate       Date    Not Null,

                        OrderTime       Time     Not Null,

CONSTRAINT Order_Info_FK1 FOREIGN KEY (CustomerID) REFERENCES
Customer(CustomerID),

CONSTRAINT Order_Info_FK2 FOREIGN KEY (OrderID) REFERENCES Order_(OrderID));


CREATE TABLE Ingredient

        (IngredientID       BIGINT    NOT NULL,

                        IngredientName     nvarchar(10)    NOT NULL,

                        Perishable          BIT          Not Null,

                        LowStock          BIT        Not Null,

CONSTRAINT Ingredient_PK PRIMARY KEY (IngredientID));


CREATE TABLE Smoothie

        (SmoothieID       BIGINT   NOT NULL,

                        SmoothieName     nvarchar(10)    NOT NULL,

                        SmoothiePrice    float          Not Null,

                        Vegetarian       BIT        Not Null,

CONSTRAINT Smoothie_PK PRIMARY KEY (SmoothieID));


CREATE TABLE Sandwich

        (SandwichID       BIGINT  NOT NULL,
```

```
                    SandwichName     nvarchar(20)    NOT NULL,
                    SandwichPrice    float           Not Null,
                    Vegetarian       BIT             Not Null,
CONSTRAINT Sandwich_PK PRIMARY KEY (SandwichID));

CREATE TABLE SoftDrink
        (SoftDrinkID       BIGINT   NOT NULL,
                    SoftDrinkName     nvarchar(10)    NOT NULL,
                    SoftDrinkPrice    float           Not Null,
CONSTRAINT SoftDrink_PK PRIMARY KEY (SoftDrinkID));

CREATE TABLE Smoothie_Order
        (SmoothieID       BIGINT    NOT NULL,
                    OrderID     BIGINT    NOT NULL,
CONSTRAINT Smoothie_Order_FK1 FOREIGN KEY (SmoothieID) REFERENCES
Smoothie(SmoothieID),
CONSTRAINT Smoothie_Order_FK2 FOREIGN KEY (OrderID) REFERENCES
Order_(OrderID));
CREATE TABLE Sandwich_Order
        (SandwichID       BIGINT    NOT NULL,
                    OrderID     BIGINT    NOT NULL,
CONSTRAINT Sandwich_Order_FK1 FOREIGN KEY (SandwichID) REFERENCES
Sandwich(SandwichID),
CONSTRAINT Sandwich_Order_FK2 FOREIGN KEY (OrderID) REFERENCES
Order_(OrderID));

CREATE TABLE SoftDrink_Order
        (SoftDrinkID       BIGINT    NOT NULL,
                    OrderID     BIGINT   NOT NULL,
CONSTRAINT SoftDrink_Order_FK1 FOREIGN KEY (SoftDrinkID) REFERENCES
SoftDrink(SoftDrinkID),
```

```
CONSTRAINT SoftDrink_Order_FK2 FOREIGN KEY (OrderID) REFERENCES
Order_(OrderID));


CREATE TABLE Smoothie_Ingredient

        (SmoothieID       BIGINT    NOT NULL,

                          IngredientID     BIGINT    NOT NULL,

                          Quantity         int          Not Null,

CONSTRAINT Smoothie_Ingredient_FK1 FOREIGN KEY (SmoothieID) REFERENCES
Smoothie(SmoothieID),

CONSTRAINT Smoothie_Ingredient_FK2 FOREIGN KEY (IngredientID) REFERENCES
Ingredient(IngredientID));


CREATE TABLE Sandwich_Ingredient

        (SandwichID       BIGINT   NOT NULL,

                          IngredientID     BIGINT    NOT NULL,

                          Quantity         int          Not Null,

CONSTRAINT Sandwich_Ingredient_FK1 FOREIGN KEY (SandwichID) REFERENCES
Sandwich(SandwichID),

CONSTRAINT Sandwich_Ingredient_FK2 FOREIGN KEY (IngredientID) REFERENCES
Ingredient(IngredientID));


--INSERTED VALUES INTO ALL TABLES


--VIEWS
```

--VIEWS 1: This view is an example of displaying every menu item with the ingredients required, perfect for training new hires and staying true to recipes

```
--EVERY SANDWICH AND ITS INGREDIENT LIST

CREATE VIEW Sandwich_Recipies AS

SELECT Sandwich.SandwichID,Sandwich.SandwichName,

Sandwich_Ingredient.IngredientID,Sandwich_Ingredient.Quantity

FROM Sandwich INNER JOIN Sandwich_Ingredient
```

ON Sandwich.SandwichID = Sandwich_Ingredient.SandwichID

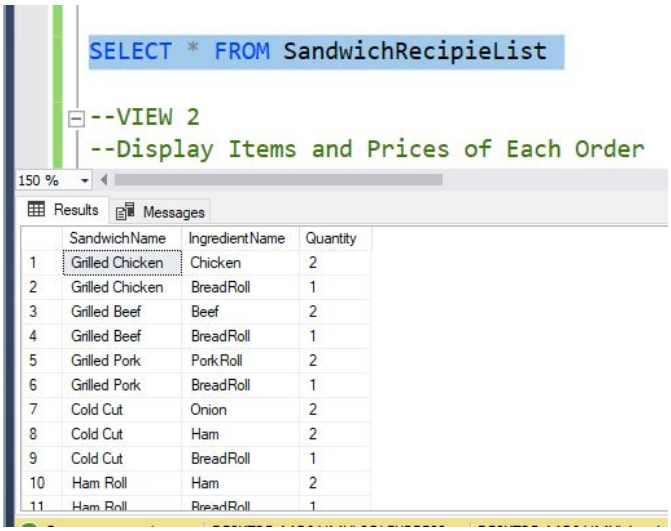
CREATE VIEW SandwichRecipieList AS

SELECT SandwichName, Ingredient.IngredientName, Quantity

FROM Sandwich_Recipies INNER JOIN Ingredient

ON Sandwich_Recipies.IngredientID=Ingredient.IngredientID

SELECT * FROM SandwichRecipieList



	SandwichName	IngredientName	Quantity
1	Grilled Chicken	Chicken	2
2	Grilled Chicken	BreadRoll	1
3	Grilled Beef	Beef	2
4	Grilled Beef	BreadRoll	1
5	Grilled Pork	PorkRoll	2
6	Grilled Pork	BreadRoll	1
7	Cold Cut	Onion	2
8	Cold Cut	Ham	2
9	Cold Cut	BreadRoll	1
10	Ham Roll	Ham	2
11	Ham Roll	BreadRoll	1

--VIEW 2: This complex view combines the Order table with the Smoothie,Sandwich, and SoftDrink Order tables to accurately show each Order and how many items are in it. We then join that view with our Sandwich, Softdrink, and Smoothie tables to display items' names and price. Using advanced NULL handling, we then sum the prices to get a total OrderCost

--Display Items and Prices of Each Order

--First view connects all items in an order

CREATE VIEW FullOrders AS SELECT Order_.OrderID, Smoothie_Order.SmoothieID, Sandwich_Order.SandwichID, SoftDrink_Order.SoftDrinkID

FROM Order_

LEFT JOIN Smoothie_Order

ON Order_.OrderID=Smoothie_Order.OrderID

LEFT JOIN Sandwich_Order

ON Order_.OrderID=Sandwich_Order.OrderID

```
LEFT JOIN SoftDrink_Order
        ON Order_.OrderID=SoftDrink_Order.OrderID

--Then join order with prices and names of items in order
CREATE VIEW OrdersWithPrices AS SELECT FullOrders.OrderID,
Smoothie.SmoothieName, Smoothie.SmoothiePrice,
Sandwich.SandwichName, Sandwich.SandwichPrice,
SoftDrink.SoftDrinkName, SoftDrink.SoftDrinkPrice
FROM FullOrders
LEFT JOIN Smoothie
        ON FullOrders.SmoothieID = Smoothie.SmoothieID
LEFT JOIN Sandwich
        ON FullOrders.SandwichID = Sandwich.SandwichID
LEFT JOIN SoftDrink
        ON FullOrders.SoftDrinkID = SoftDrink.SoftDrinkID

--Replace NULLS with 0s to sum prices in order correct
SELECT OrderID,OrdersWithPrices.SmoothiePrice,
OrdersWithPrices.SandwichPrice,OrdersWithPrices.SoftDrinkPrice,
COALESCE(OrdersWithPrices.SmoothiePrice,0)+
COALESCE(OrdersWithPrices.SandwichPrice,0)+
COALESCE(OrdersWithPrices.SoftDrinkPrice,0)
AS OrderTotal
FROM OrdersWithPrices
```

*All full PDFs of ERD/RDM/Data Dictionary are attached as full files

```
--Replace NULLS with 0s to sum prices in order correct
SELECT OrderID,OrdersWithPrices.SmoothiePrice,
OrdersWithPrices.SandwichPrice,OrdersWithPrices.SoftDrinkPrice,
COALESCE(OrdersWithPrices.SmoothiePrice,0)+
COALESCE(OrdersWithPrices.SandwichPrice,0)+
COALESCE(OrdersWithPrices.SoftDrinkPrice,0)
AS OrderTotal
FROM OrdersWithPrices
```

150 %

Results | Messages

	OrderID	SmoothiePrice	SandwichPrice	SoftDrinkPrice	OrderTotal
1	0	5.25	NULL	NULL	5.25
2	1	5.5	NULL	NULL	5.5
3	2	5.25	NULL	NULL	5.25
4	3	5.25	NULL	NULL	5.25
5	4	6	NULL	3.25	9.25
6	5	5.25	NULL	2.5	7.75
7	6	5.5	NULL	2.5	8
8	6	5.5	NULL	3.25	8.75
9	7	5.25	NULL	3.25	8.5
10	12345	5.25	NULL	NULL	5.25
11	12345	5.25	NULL	NULL	5.25

--VIEW 3: Display amount of orders per customer to see most loyal. This view will help develop a loyalty program and help employees get to know their most loyal customers!

```
CREATE VIEW LoyalCustomer AS

SELECT Order_.CustomerID, Customer.CustomerName,

COUNT(OrderID) AS NumOfOrders

From Order_

INNER JOIN Customer ON Order_.CustomerID=Customer.CustomerID

GROUP BY Order_.CustomerID, Customer.CustomerName


--Display customers who have ordered more than once

SELECT * FROM LoyalCustomer WHERE NumOfOrders > 1
```

```
SELECT * FROM LoyalCustomer WHERE NumOfOrders > 1
```

150 %

Results | Messages

	CustomerID	CustomerName	NumOfOrders
1	12345	Guillem Badenes	2
2	22345	Jack Stoecker	2
3	23345	Martin Aasen	2
4	23445	Terrell Smith	2
5	23456	Emil Christensen	2
6	33456	Liam Anderson	2
7	34456	Tennyson Stevens	2
8	34566	Dayo Thomas	2