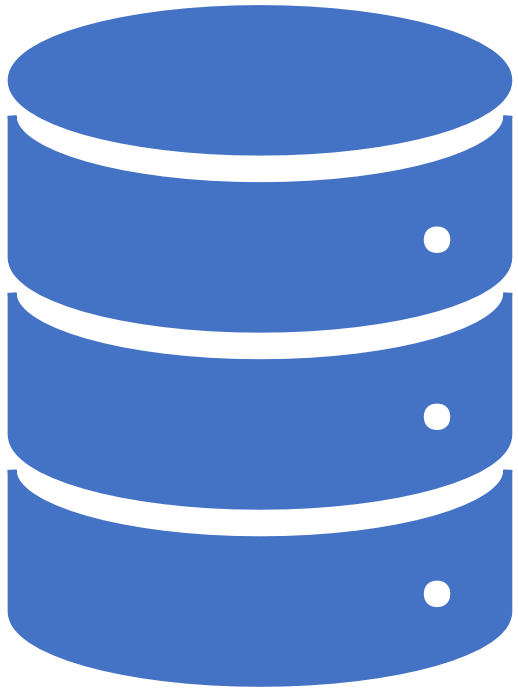


DATABASE for HFS

Malgorzata Olesiewicz

1975361

Business Space

HFS provides housing for 10,101 student.

HFS' amenities include:

- Two residential dining venues—Local Point and Center Table—with over 898,000 transactions last year
- Two student community centers—Area 01 and The MILL—and Fitness Center West, with over 99,000 visits from residents last year
- District Market grocery on West Campus, with 5,000 transactions daily

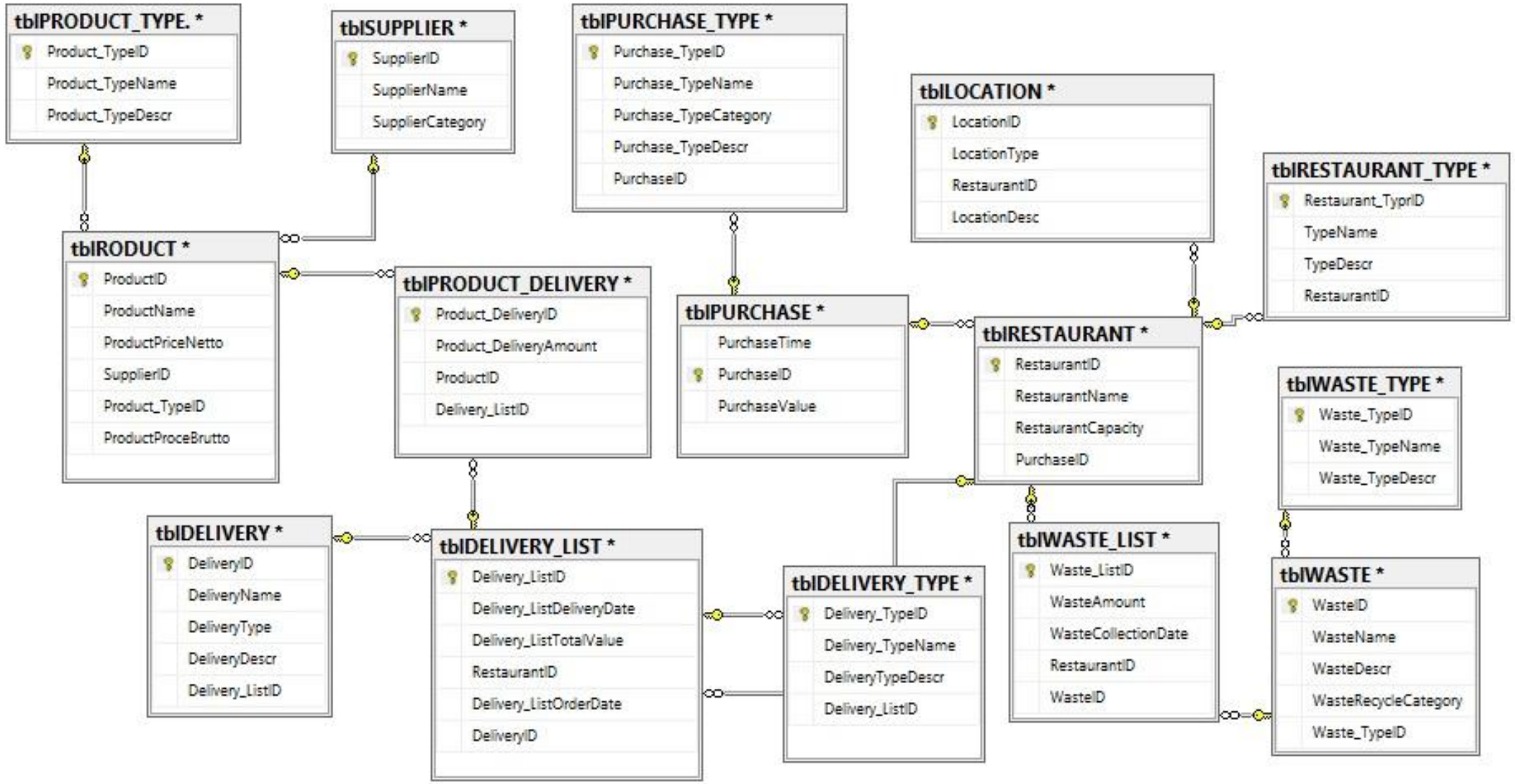
Objectives:

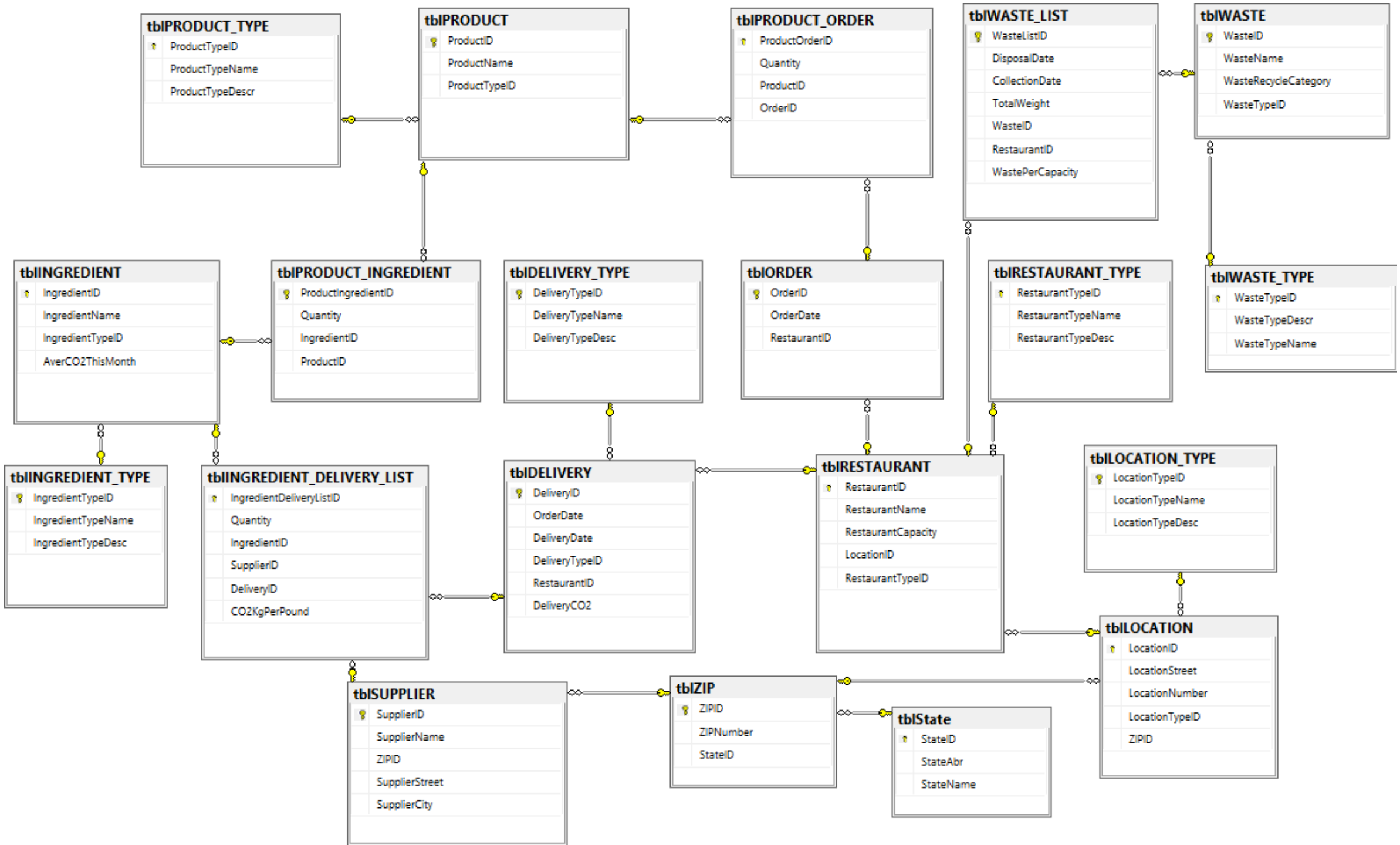
- Reduce the waste among the food facilities through ordering appropriate quantities
- Reduce the carbon footprint of the facilities
- Inform the customer about the carbon footprint of the product they consume

How?

- By trucking transactions and deliveries of the supplies to all the venues
- By trucking the amount of waste and type of waste in each of the facilities
- By trucking the carbon footprint for each product

First Draft of ERD




tblPRODUCT_TYPE
ProductTypeID
ProductTypeName
ProductTypeDescr
tblPRODUCT
ProductID
ProductName
ProductTypeID
tblPRODUCT_ORDER
ProductOrderID
Quantity
ProductID
OrderID
tblWASTE_LIST
WasteListID
DisposalDate
CollectionDate
TotalWeight
WasteID
RestaurantID
WastePerCapacity
tblWASTE
WasteID
WasteName
WasteRecycleCategory
WasteTypeID
tblINGREDIENT
IngredientID
IngredientName
IngredientTypeID
AverCO2ThisMonth
tblPRODUCT_INGREDIENT
ProductIngredientID
Quantity
IngredientID
ProductID
tblDELIVERY_TYPE
DeliveryTypeID
DeliveryTypeName
DeliveryTypeDesc
tblORDER
OrderID
OrderDate
RestaurantID
tblRESTAURANT_TYPE
RestaurantTypeID
RestaurantTypeName
RestaurantTypeDesc
tblWASTE_TYPE
WasteTypeID
WasteTypeDescr
WasteTypeName
tblINGREDIENT_TYPE
IngredientTypeID
IngredientTypeName
IngredientTypeDesc
tblINGREDIENT_DELIVERY_LIST
IngredientDeliveryListID
Quantity
IngredientID
SupplierID
DeliveryID
CO2KgPerPound
tblDELIVERY
DeliveryID
OrderDate
DeliveryDate
DeliveryTypeID
RestaurantID
DeliveryCO2
tblRESTAURANT
RestaurantID
RestaurantName
RestaurantCapacity
LocationID
RestaurantTypeID
tblLOCATION_TYPE
LocationTypeID
LocationTypeName
LocationTypeDesc
tblLOCATION
LocationID
LocationStreet
LocationNumber
LocationTypeID
ZIPID
tblSUPPLIER
SupplierID
SupplierName
ZIPID
SupplierStreet
SupplierCity
tblZIP
ZIPID
ZIPNumber
StateID
tblState
StateID
StateAbr
StateName

Stored Procedures: Add supplier with new ZIP

```
CREATE PROCEDURE addnewsupplierWithNewZIP
@SupName varchar(50),
@SupStreet varchar(50),
@SupCity varchar(50),
@ZIPNumber INT,
@StateAbr varchar(4)

AS
BEGIN TRANSACTION T1
DECLARE @ZIPID INT

INSERT INTO tblZIP (ZIPNumber, StateID)
VALUES (@ZIPNumber, (SELECT StateID FROM tblState WHERE Stateabr =
@StateAbr))

SET @ZIPID = (SELECT SCOPE_IDENTITY())

INSERT INTO tblSupplier (SupplierName,ZIPID, SupplierStreet, SupplierCity )
VALUES (@SupName, @ZIPID, @SupStreet, @SupCity)

COMMIT TRANSACTION T1
```

Procedure to enter new Ingredient Delivery List Item

```
CREATE PROCEDURE newIngredientDelivery
@Quan numeric (6,2),
@InName varchar(50),
@SupName varchar(50),
@ResName varchar (50),
@DelOrderDate Date,
@DeDeDate Date,
        @DeType varchar(50),
          @CO2PerPound numeric(6,2)
            AS
            BEGIN TRANSACTION T2
             DECLARE @IngID INT
             DECLARE @ResID INT
             DECLARE @SupID INT
             DECLARE @DelID INT
          SET @ResID = (SELECT RestaurantID FROM tblRestaurant
       WHERE RestaurantName =  @ResName)
      SET @DelID = (SELECT DeliveryID FROM tblDelivery WHERE OrderDate = @DelOrderDate AND
          DeliveryDate = @DeDeDate AND RestaurantId = @ResID)
SET @IngID = (SELECT IngredientID FROM tblIngredient WHERE IngredientName = @InName)
SET @SupID = (SELECT SupplierID FROM tblSupplier WHERE SupplierName  = @SupName)
INSERT INTO tblINGREDIENT_DELIVERY_LIST (Quantity, IngredientID, SupplierID, DeliveryID,
CO2KgPerPound)
VALUES (@Quan, @IngID, @SupID, @DelID, @CO2PerPound)
COMMIT TRANSACTION T2
```

Calculate the average CO2 footprint for every ingredient this month

```
CREATE FUNCTION AverCO2(@PK INT)
RETURNS Numeric (6,2)
AS
BEGIN
DECLARE @Ref Numeric = (SELECT AVG(IDL.CO2KgPerPound)
                        FROM tblProduct P
                        JOIN tblProduct_Order OD ON OD.ProductID = P.ProductID
                        JOIN tblProduct_Ingredient  PIT ON PIT.ProductID = P. ProductID
                         JOIN tblIngredient I ON I.IngredientID = PIT.IngredientID
                         JOIN tblIngredient_Delivery_List IDL on IDL.IngredientID = I.IngredientID
                         JOIN tblDelivery D ON D.DeliveryID = IDL.DeliveryID

WHERE IDL.IngredientID  = @PK
AND YEAR(GetDate())=YEAR(D.DeliveryDate)
AND MONTH(GetDate())=MONTH(D.DeliveryDate))

RETURN @Ref
END
GO

ALTER TABLE tblIngredient
ADD AverCO2ThisMonth AS (dbo.TotalCO2(IngredientID))

GO
```

Relative amount of waste (waste per capacity)

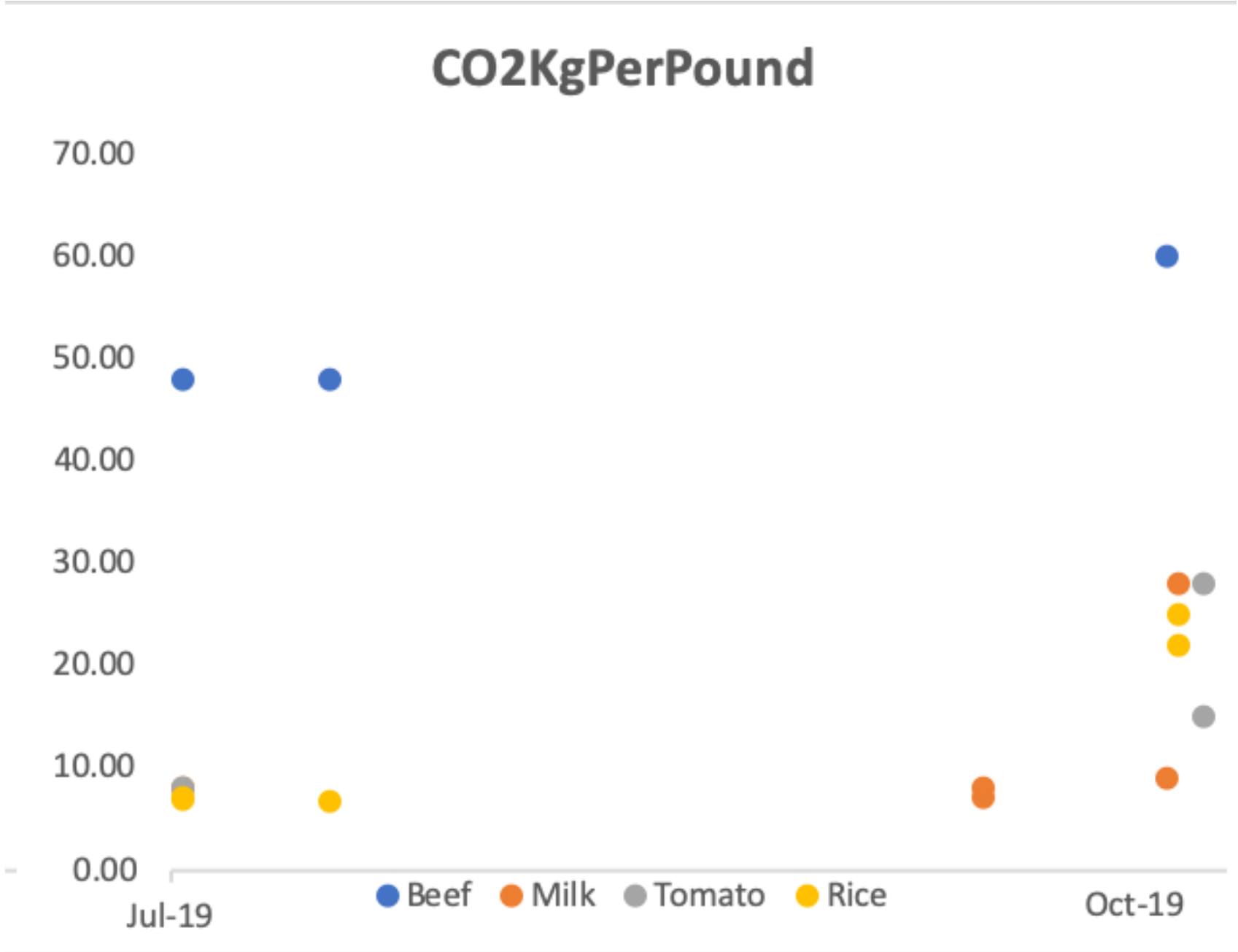
```
CREATE Function wastePerCapacity(@PK INT)
RETURNS Numeric(4,2)
AS
BEGIN
DECLARE @REF Numeric = (SELECT WL.TotalWeight/R.RestaurantCapacity
        FROM tblWaste_List WL
        JOIN tblRestaurant R On R.RestaurantID = WL.RestaurantID
        WHERE WL.WasteListID = @PK)

RETURN @REF
END
GO

ALTER TABLE tblWaste_List
ADD WastePerCapacity AS (dbo.wastePerCapacity(WasteListID))

GO
```

Don't allow Restaurant Type 'Italian' order more than 20 pounds of rice per weekly order

```
CREATE FUNCTION NoRiceForItalians
RETURNS INT
AS
BEGIN

DECLARE @Ret INT = 0
IF EXISTS (SELECT R.RestaurantID
        FROM tblRestaurant_Type RT
          JOIN tblRestaurant R ON RT.RestaurantTypeID = R.RestaurantTypeID
            JOIN tblDelivery D ON D.ResturantID=R.RestaurantID
              JOIN tblDelivery_Type DT ON DT.DeliveryTypeID = D.DeliveryTypeID
                JOIN tblIngredient_Delivery_List IDL ON IDL.DeliveryID = D.DeliveryID
        WHERE RT.RestaurantTypeName LIKE '%Italian%'
        AND DT.DeliveryTypeName = 'Weekly'
        AND I.IngredientName = 'Rice'
        AND IDL.Quantity>20)

        BEGIN
          SET @Ret = 1
        END
RETURN @Ret
END
```

Don't let Restaurant Roma order products o utside of Washington State

```
CREATE FUNCTION NoProductOusideOfWA
RETURNS INT
AS
BEGIN

DECLARE @Ret INT = 0
IF EXISTS (SELECT R.RestaurantID
        FROM tblRestaurant_Type RT
        JOIN tblRestaurant R ON RT.RestaurantTypeID = R.RestaurantTypeID
        JOIN tblDelivery D ON D.ResturantID = R.RestaurantID
        JOIN tblLocation L ON L.LocationID = R.LocationID
        JOIN tblZIP Z ON Z.ZIPID = L.ZIPID
        JOIN tblState S ON S.StateID = Z.StateID
        WHERE RT.RestaurantName = 'Roma'
        AND S.StateAbr != 'WA')

        BEGIN
           SET @Ret = 1
        END
RETURN @Ret
END
```


CO2KgPerPound
70.00
60.00
50.00
40.00
30.00
20.00
10.00
0.00
Jul-19
Oct-19
Beef
Milk
Tomato
Rice

Lessons learned

- Choose only essential data to the problem you try to tackle
- Write query or two before you insert data
- Think about what data are feasible to obtain