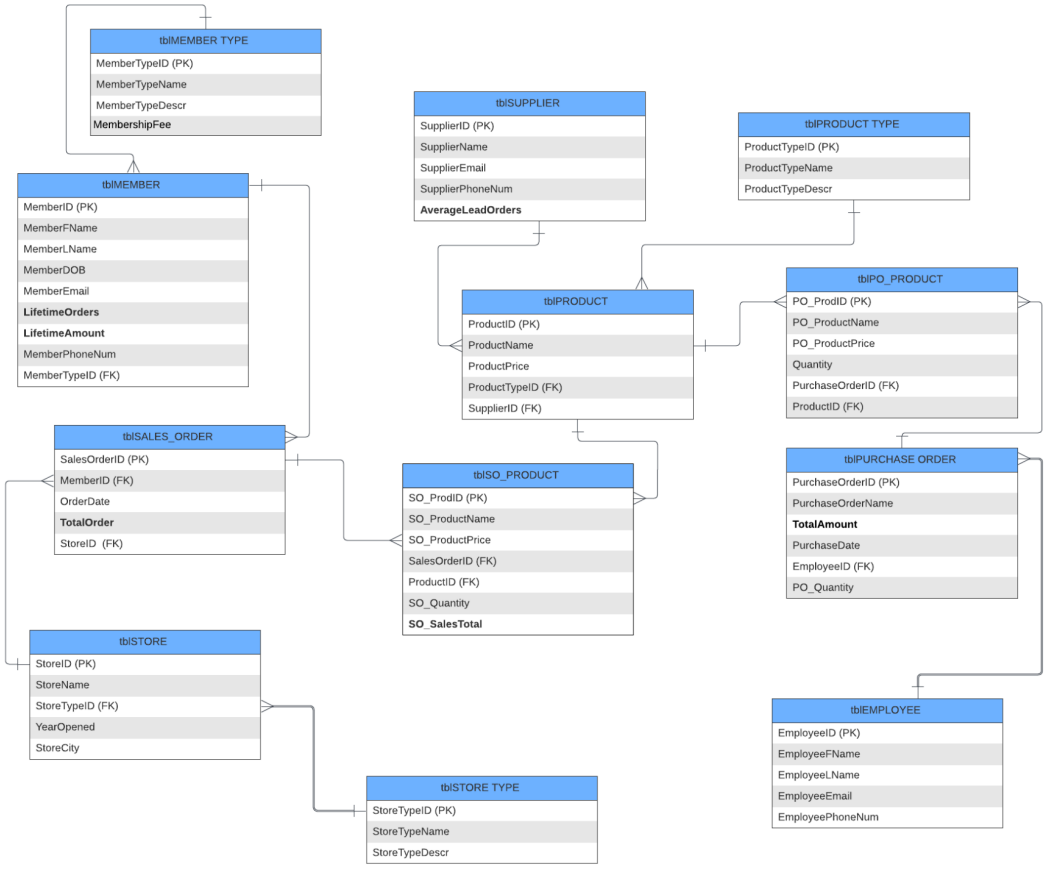
COSTCO ERD:



Creating Tables

Amy:

```
CREATE TABLE tblMEMBER_TYPE
(MemberTypeID INT IDENTITY(1,1)primary key,
MemberTypeName varchar(50)not null,
MemberTypeDescr varchar(80)not null,
MembershipFee NUMERIC(10,2) not null)
GO
—
CREATE TABLE tblSTORE_TYPE
(StoreTypeID INT IDENTITY(1,1)primary key,
StoreTypeName varchar(50)not null,
StoreTypeDescr varchar(50)not null)
GO
```

—

```
CREATE TABLE tblMEMBER
(MemberID INT IDENTITY(1,1)primary key,
MemberFname varchar(50)not null,
MemberLname varchar(50)not null,
MemberEmail varchar(80)not null,
MemberDOB DATE not null,
LifetimeOrders NUMERIC(10,2),
LifetimeAmount NUMERIC (10,2),
MemberPhoneNum varchar(20) not null,
MemberTypeID INT null)
GO
--- foreign key
ALTER TABLE tblMEMBER
ADD CONSTRAINT FK_tblMEMBER_MemberTypeID
FOREIGN KEY (MemberTypeID)
REFERENCES tblMEMBER_TYPE(MemberTypeID)
GO
```

Ahana:

```
--CREATE TABLES
CREATE TABLE tblEMPLOYEE
(EmployeeID INTEGER IDENTITY(1,1) primary key,
EmployeeFname varchar(30) not null,
EmployeeLname varchar(30) not null,
EmployeeEmail varchar(60) not null,
EmployeePhoneNum varchar(20) not null)
GO

GO
CREATE TABLE tblSTORE
(StoreID INTEGER IDENTITY(1,1) primary key,
StoreName varchar(30) not null,
StoreTypeID INT not null,
YearOpened Int not null,
StoreCity varchar(80) not null)
```

```
GO

GO
CREATE TABLE tblSTORE_TYPE
(StoreTypeID INTEGER IDENTITY(1,1) primary key,
StoreTypeName varchar(30) not null,
StoreTypeDescr varchar(500) not null)
GO



GO
CREATE TABLE tblSALES_ORDER
(SalesOrderID INTEGER IDENTITY(1,1) primary key,
MemberID INT not null,
OrderDate DATE not null,
TotalPrice Numeric(8,2) not null -- quantity * price
)
GO
-- ADD FOREIGN KEYS
ALTER TABLE tblSALES_ORDER
ADD CONSTRAINT FK_tblMEMBER_MemberID
FOREIGN KEY (MemberID)
REFERENCES tblMEMBER (MemberID)
GO

ALTER TABLE tblSALES_ORDER
ADD CONSTRAINT FK_tblSTORE_StoreID
FOREIGN KEY (StoreID)
REFERENCES tblSTORE (StoreID)
GO

ALTER TABLE tblSTORE
ADD CONSTRAINT FK_tblSTORE_TYPE_StoreTypeID
FOREIGN KEY (StoreTypeID)
REFERENCES tblSTORE_TYPE (StoreTypeID)
GO
```

Priscilla:

```
-- create table for supplier, product and so_product
CREATE TABLE tblSUPPLIER
(SupplierID INTEGER IDENTITY (1,1) primary key,
SupplierName varchar(50) not null,
SupplierEmail varchar(50) not null,
SupplierPhoneNum varchar(50) not null,
AverageLeadOrders varchar(50) not null)
GO

CREATE TABLE tblPRODUCT
(ProductID INTEGER IDENTITY (1,1) primary key,
ProductName varchar(50) not null,
ProductPrice Numeric(7,2),
ProductTypeID INT not null,
SupplierID INT not null)
GO

CREATE TABLE tblSO_PRODUCT
(SO_ProdID INTEGER IDENTITY (1,1) primary key,
SO_ProductName varchar(50) not null,
SO_ProductPrice Numeric(7,2),
SalesOrderID INT not null,
ProductID INT not null,
SO_Quantity INT not null)
GO

-- add foreign key to the tables

ALTER TABLE tblPRODUCT
ADD CONSTRAINT FK_tblPRODUCT_ProductTypeID
FOREIGN KEY (ProductTypeID)
REFERENCES tblPRODUCT_TYPE (ProductTypeID)
GO
```

```
ALTER TABLE tblPRODUCT
ADD CONSTRAINT FK_tblPRODUCT_SupplierID
FOREIGN KEY (SupplierID)
REFERENCES tblSUPPLIER (SupplierID)
GO

ALTER TABLE tblSO_PRODUCT
ADD CONSTRAINT FK_tblSO_PRODUCT_SalesOrderID
FOREIGN KEY (SalesOrderID)
REFERENCES tblSALES_ORDER (SalesOrderID)
GO

ALTER TABLE tblSO_PRODUCT
ADD CONSTRAINT FK_tblSO_PRODUCT_ProductID
FOREIGN KEY (ProductID)
REFERENCES tblPRODUCT (ProductID)
GO
```

Annabelle:

```
-- Create Table for tblPRODUCT_TYPE
CREATE TABLE tblPRODUCT_TYPE
(ProductTypeID INT IDENTITY(1,1) PRIMARY KEY,
ProductTypeName VARCHAR(50) NOT NULL,
ProductTypeDescr VARCHAR(150) NULL)
GO

-- Create Table for tblPO_PRODUCT
CREATE TABLE tblPO_PRODUCT
(PO_ProdID INT IDENTITY(1,1) PRIMARY KEY,
PO_ProductName VARCHAR(50) NOT NULL,
PO_ProductPrice NUMERIC (10,2) NOT NULL,
Quantity INT NOT NULL
)
```

```
GO

-- Create Table for tblPURCHASE_ORDER
CREATE TABLE tblPURCHASE_ORDER
(PurchaseOrderID INT IDENTITY(1,1) PRIMARY KEY,
PurchaseOrderName VARCHAR(150) NOT NULl,
TotalAmount NUMERIC (10,2) NOT NULL, --total price
PurchaseDate DATE NOT NULL
)
GO

ALTER TABLE tblPO_PRODUCT
ADD CONSTRAINT FK_PO_PRODUCT_PurchaseOrderID
FOREIGN KEY (PurchaseOrderID)
REFERENCES tblPURCHASE_ORDER(PurchaseOrderID)
GO

ALTER TABLE tblPO_PRODUCT
ADD CONSTRAINT FK_PO_PRODUCT_ProductID
FOREIGN KEY (ProductID)
REFERENCES tblPRODUCT(ProductID)
GO

ALTER TABLE tblPURCHASE_ORDER
ADD CONSTRAINT FK_PURCHASE_ORDER_EmployeeeID
FOREIGN KEY (EmployeeID)
REFERENCES tblEMPLOYEE(EmployeeID)
GO
```

Populating Tables

Amy:

```
INSERT INTO tblPURCHASE_ORDER
(PurchaseOrderName,TotalAmount,PurchaseDate,EmployeeID,
PO_Quantity,PO_ProductPrice)
```

```
VALUES ('Coco-Cola', 1200,'April 9, 2021', (SELECT EmployeeID FROM
tblEMPLOYEE WHERE EmployeeLname = 'Thierman'),1237,300.00),
('Casper Select 12" Hybrid Medium-Frim Mattress', 1500, 'January 1,
2010', (SELECT EmployeeID FROM tblEMPLOYEE WHERE EmployeeLname =
'Aragao'),1519,450.00),
('Kirkland Signature Daily Multi, 500 Tablets',700, 'May 6, 2013',
(SELECT EmployeeID FROM tblEMPLOYEE WHERE EmployeeLname =
'Fevold'),769,236.00),
('Rasberries', 3400, 'October 27, 2017', (SELECT EmployeeID FROM
tblEMPLOYEE WHERE EmployeeLname = 'Lerer'),3467,532.00),
('Cheddar Chips', 5700, 'December 3, 2011', (SELECT EmployeeID FROM
tblEMPLOYEE WHERE EmployeeLname = 'Zoss'),5799,189.00)
GO
INSERT INTO tblMEMBER_TYPE
(MemberTypeName,MemberTypeDescr,MembershipFee)
VALUES ('Executive','pay more and annual
reward','120.00'),('GoldStar','pay less and no annual reward','80.00')
GO
INSERT INTO tblMEMBER (MemberFname, MemberLname,MemberEmail,
MemberDOB, MemberPhoneNum)
SELECT TOP 2000 CustomerFname, CustomerLname, Email, DateOfBirth,
PhoneNum
FROM peeps.dbo.tblCUSTOMER
GO
```

Ahana:

```
-- POPULATE EMPLOYEE TABLE USING STORED PROC
EXEC aroyINSERT_EMPLOYEESfromPeepsDB 200
SELECT * FROM tblEMPLOYEE

-- POPULATE STORE_TYPE
```

```
EXEC aroyINSERT_STORE_TYPE
@StoreTypeName2 = 'Business Center',
@StoreTypeDescr2 = 'For businesses wholesale shopping.'
select * from tblSTORE_TYPE

-- POPULATE STORE TABLE USING A STORED PROC
EXEC aroyINSERT_STORE
@StoreName = 'Costco Wholesale' ,
@YearOpened = 1999 ,
@StoreCity = 'Lynwood',
@ST_Name =  'Business Center'
SELECT * FROM tblSTORE


-- CALLS STORED PROC TO POPULATE SALES ORDER TABLE
DECLARE @GetDate Date = GETDATE()
EXEC aroyINSERT_SALESORDER
@Memberemail = 'Karima.Butterworth303@bpdelawarematerials.com' ,
@OrderDate = @GetDate ,
@TotalPrice = 0.0 ,
@ST_Name =  'Wholesale'
select * from tblSALES_ORDER




-- insert data into employee table from peeps
ALTER PROCEDURE aroyINSERT_EMPLOYEESfromPeepsDB
@Recordnum INT
AS
    BEGIN TRANSACTION A1
BEGIN
    DELETE tblEMPLOYEE
    INSERT INTO tblEMPLOYEE (EmployeeFname, EmployeeLname,
EmployeeEmail, EmployeePhoneNum)
```

```
    SELECT TOP (@Recordnum) CustomerFname, CustomerLname, Email,
PhoneNum
    FROM Peeps.dbo.tblcustomer
END
    COMMIT TRANSACTION A1

CREATE PROCEDURE aroyINSERT_STORE_TYPE
@StoreTypeName2 varchar(30) ,
@StoreTypeDescr2 varchar(500)

AS
    BEGIN TRANSACTION A1
BEGIN

    INSERT INTO tblSTORE_TYPE (StoreTypeName, StoreTypeDescr)
    VALUES (@StoreTypeName2, @StoreTypeDescr2 )
END
    COMMIT TRANSACTION A1
```

Priscilla:

```
-- populate look up table

INSERT INTO tblSUPPLIER (SupplierName, SupplierEmail,
SupplierPhoneNum, AverageLeadOrders)
VALUES ('PepsiCo', 'pepsicomksg@pepsico.com', '18004332652', '256'),
('Kraft Heinz Company', 'usprivacy@kraftheinz.com', '18556341984',
'678'),
('Ezaki Glico Food Company', 'info@glicousa.com', '81664778352',
'567'), ('Haribo', 'info@haribo.com', '8662148160', '879'),
('Frito-Lay', 'info@fritolay.com', '18003524477', '889')
```

```
INSERT INTO tblSO_PRODUCT (SO_ProductName, SO_ProductPrice,
SalesOrderID, ProductID, SO_Quantity)
VALUES ('Chocolate Pocky', '28.48', (SELECT MAX(SalesOrderID) FROM
tblSALES_ORDER WHERE OrderDate = '2018-11-11'), (SELECT ProductID FROM
tblPRODUCT WHERE ProductName = 'Chocolate Pocky'), '8'),
('Cheddar Chips', '23.96', (SELECT MAX(SalesOrderID) FROM
tblSALES_ORDER WHERE OrderDate = '2017-01-01'), (SELECT ProductID FROM
tblPRODUCT WHERE ProductName = 'Cheddar Chips'), '4'),
('Ketchup', '7.68', (SELECT MAX(SalesOrderID) FROM tblSALES_ORDER
WHERE OrderDate = '2019-03-02'), (SELECT ProductID FROM tblPRODUCT
WHERE ProductName = 'Ketchup'), '3'),
('Gummy Bears', '15.60', (SELECT MAX(SalesOrderID) FROM tblSALES_ORDER
WHERE OrderDate = '2022-02-02'), (SELECT ProductID FROM tblPRODUCT
WHERE ProductName = 'Gummy Bears'), '10'),
('Coca-Cola', '51.38', (SELECT MAX(SalesOrderID) FROM tblSALES_ORDER
WHERE OrderDate = '2023-07-13'), (SELECT ProductID FROM tblPRODUCT
WHERE ProductName = 'Coca-Cola'),'14')


INSERT INTO tblSALES_ORDER (MemberID, OrderDate, TotalPrice, StoreID)
VALUES((SELECT MemberID FROM tblMEMBER WHERE MemberLname = 'Katin'),
'2022-02-02', '51.38', (SELECT StoreID FROM tblSTORE WHERE StoreCity =
'Seattle'))
```

Annabelle:

```
/*
Populate look-up table tblPRODUCT_TYPE
*/
```

```
INSERT INTO tblPRODUCT_TYPE (ProductTypeName, ProductTypeDescr)
VALUES ('Produce', 'Farm - produced crops, including fruits and
vegetables'),
('Eggs','Produced from poultry.'), ('Furniture','Large objects that
are used in a room'),
('Health & Personal','Helps well-being'), ('Pet', 'Animal and pet
care'),
('Alcohol', 'Beer, Wine, and Liquor')

INSERT INTO tblPRODUCT_TYPE (ProductTypeName, ProductTypeDescr)
VALUES ('Snacks', 'A light meal'), ('Condiment', 'Enhance flavor'),
('Sweets', 'Confectionary'),
('Soda', 'Bubbly drink')

INSERT INTO tblPRODUCT_TYPE (ProductTypeName, ProductTypeDescr)
VALUES ('Vacation', 'Leisure and Recreation')

INSERT INTO tblSUPPLIER (SupplierName, SupplierEmail,
SupplierPhoneNum, AverageLeadOrders)
VALUES ('Organic Farm', 'rasp@orgfarm.com', '9769491814', '1054'),
('Kirkland', 'kirkland@costco.com', '4958338549', '456'),
('Casper', 'amanda@caspersleep.com', '5155622777', '897')

/*
Populate look-up table tblPRODUCT
*/
INSERT INTO tblPRODUCT (ProductName, ProductPrice, ProductTypeID,
SupplierID)
VALUES
('Raspberries, 12oz', 7.99, (SELECT ProductTypeID FROM
tblPRODUCT_TYPE WHERE ProductTypeName = 'Produce'),
(SELECT SupplierID FROM tblSUPPLIER WHERE SupplierName = 'Organic
Farm')),
('Kirkland Signature Organic Free Range Eggs', 8.79,
```

```
(SELECT ProductTypeID FROM tblPRODUCT_TYPE WHERE ProductTypeName =
'Eggs'),
(SELECT SupplierID FROM tblSUPPLIER WHERE SupplierName =
'Kirkland')),
('Casper Select 12" Hybrid Medium-Frim Mattress', 799.99,
(SELECT ProductTypeID FROM tblPRODUCT_TYPE WHERE ProductTypeName =
'Furniture'),
(SELECT SupplierID FROM tblSUPPLIER WHERE SupplierName = 'Casper')),
('Kirkland Signature Daily Multi, 500 Tablets', 19.99,
(SELECT ProductTypeID FROM tblPRODUCT_TYPE WHERE ProductTypeName =
'Health & Personal'),
(SELECT SupplierID FROM tblSUPPLIER WHERE SupplierName =
'Kirkland')),
('Kirkland Signature Adult Dog Food', 49.99,
(SELECT ProductTypeID FROM tblPRODUCT_TYPE WHERE ProductTypeName =
'Pet'),
(SELECT SupplierID FROM tblSUPPLIER WHERE SupplierName =
'Kirkland')),
('Kirkland Signature American Vodka', 14.00,
(SELECT ProductTypeID FROM tblPRODUCT_TYPE WHERE ProductTypeName =
'Alcohol'),
(SELECT SupplierID FROM tblSUPPLIER WHERE SupplierName =
'Kirkland')),
('Chocolate Pocky', 28.48,
(SELECT ProductTypeID FROM tblPRODUCT_TYPE WHERE ProductTypeName =
'Snacks'),
(SELECT SupplierID FROM tblSUPPLIER WHERE SupplierName = 'Ezaki Glico
Food Company')),
('Ketchup', 7.68,
(SELECT ProductTypeID FROM tblPRODUCT_TYPE WHERE ProductTypeName =
'Condiment'),
(SELECT SupplierID FROM tblSUPPLIER WHERE SupplierName = 'Kraft Heinz
Company')),
('Gummy Bears', 15.60,
```

```
(SELECT ProductTypeID FROM tblPRODUCT_TYPE WHERE ProductTypeName =
'Sweets'),
(SELECT SupplierID FROM tblSUPPLIER WHERE SupplierName = 'Haribo')),
('Coca-Cola', 51.38,
(SELECT ProductTypeID FROM tblPRODUCT_TYPE WHERE ProductTypeName =
'Soda'),
(SELECT SupplierID FROM tblSUPPLIER WHERE SupplierName = 'PepsiCo')),

INSERT INTO tblPRODUCT (ProductName, ProductPrice, ProductTypeID,
SupplierID)
VALUES ('Cheddar Chips', 23.96,
(SELECT ProductTypeID FROM tblPRODUCT_TYPE WHERE ProductTypeName =
'Snacks'),
(SELECT SupplierID FROM tblSUPPLIER WHERE SupplierName =
'Frito-Lay'))

/*
Populate look-up table tblPO_PRODUCT
*/
INSERT INTO tblPO_PRODUCT (PO_ProductName, PO_ProductPrice, Quantity,
PurchaseOrderID, ProductID)
VALUES ('Raspberries, 12oz', 786.00, 157, (SELECT PurchaseOrderID
FROM tblPURCHASE_ORDER WHERE PurchaseDate = '2017-10-27'),
(SELECT ProductID FROM tblPRODUCT WHERE ProductName = 'Raspberries,
12oz')),
('Coca-Cola', 850.00, 1678,(SELECT PurchaseOrderID FROM
tblPURCHASE_ORDER WHERE PurchaseDate = '2021-04-09'),
(SELECT ProductID FROM tblPRODUCT WHERE ProductName = 'Coca-Cola')),
('Casper Select 12" Hybrid Medium-Frim Mattress', 898.00, 15,(SELECT
PurchaseOrderID FROM tblPURCHASE_ORDER WHERE PurchaseDate =
'2010-01-01'),
(SELECT ProductID FROM tblPRODUCT WHERE ProductName = 'Casper Select
12" Hybrid Medium-Frim Mattress')),
```

```
('Kirkland Signature Daily Multi, 500 Tablets', 555.00, 87,(SELECT
PurchaseOrderID FROM tblPURCHASE_ORDER WHERE PurchaseDate =
'2013-05-06'),
(SELECT ProductID FROM tblPRODUCT WHERE ProductName = 'Kirkland
Signature Daily Multi, 500 Tablets')),
('Cheddar Chips', 3550.00, 400,(SELECT PurchaseOrderID FROM
tblPURCHASE_ORDER WHERE PurchaseDate = '2011-12-03'),
(SELECT ProductID FROM tblPRODUCT WHERE ProductName = 'Cheddar
Chips'))
```

Stored Procedures

Amy:

```
CREATE PROCEDURE inserttblsalesorder
@MembrEmail varchar(80),
@StorName varchar(50),
@ODate DATE,
@TotalPrice INT
AS
DECLARE @MembrID INT
SET @MembrID = (SELECT MemberID FROM tblMEMBER WHERE MemberEmail =
@MembrEmail)
DECLARE @StorID INT
SET @StorID = (SELECT StoreID FROM tblSTORE WHERE StoreName =
@StorName)
BEGIN TRANSACTION ao6
INSERT INTO tblSALES_ORDER (MemberID,OrderDate, TotalPrice, StoreID)
VALUES (@MembrID,@ODate,@TotalPrice,@StorID)
IF @@Error <> 0
BEGIN
PRINT 'stop rollback'
ROLLBACK TRANSACTION ao6
END
COMMIT TRANSACTION ao6
GO
```

```
—-
CREATE PROCEDURE inserttblmember
@MemTName varchar(50),
@MFname varchar(50),
@MLname varchar(50),
@MDOB DATE,
@MEmail varchar(50),
@LifeOrders INT,
@LifeAmount INT,
@MemPhone varchar(20)
AS
DECLARE @MemTypeID INT
SET @MemTypeID = (SELECT MemberTypeID FROM tblMEMBER_TYPE WHERE
MemberTypeName = @MemTName)
BEGIN TRANSACTION ao62
INSERT INTO tblMEMBER
(MemberFname,MemberLname,MemberDOB,MemberEmail,LifetimeOrders,Lifetime
Amount,MemberPhoneNum,MemberTypeID)
VALUES
(@MFname,@MLname,@MDOB,@MEmail,@LifeOrders,@LifeAmount,@MemPhone,@MemT
ypeID)
IF @@Error <> 0
BEGIN
PRINT 'stop rollback'
ROLLBACK TRANSACTION ao62
END
COMMIT TRANSACTION ao62
GO
```

Ahana:

```
-- insert store data stored proc
CREATE PROCEDURE aroyINSERT_STORE
@StoreName varchar(30) ,
@YearOpened Int ,
```

```
@StoreCity varchar(80),
@ST_Name varchar(30)


AS

DECLARE @ST_ID INT
SET @ST_ID = (SELECT StoreTypeID FROM tblSTORE_TYPE WHERE
StoreTypeName = @ST_Name)

    BEGIN TRANSACTION A1
BEGIN

    INSERT INTO tblSTORE (StoreName, StoreTypeID, YearOpened,
StoreCity)
    VALUES (@StoreName,@ST_ID, @YearOpened, @StoreCity )
END
    COMMIT TRANSACTION A1




-- Insert Sales Order stored proc
CREATE PROCEDURE aroyINSERT_SALESORDER
@Memberemail varchar(80) ,
@OrderDate Date ,
@TotalPrice numeric(8,2),
@ST_Name varchar(30)


AS
--get store type ID
DECLARE @ST_ID INT
SET @ST_ID = (SELECT StoreTypeID FROM tblSTORE_TYPE WHERE
StoreTypeName = @ST_Name)
--Get memberID
DECLARE @mem_ID INT
```

```
SET @mem_ID = (SELECT MemberID FROM tblMember WHERE MemberEmail =
@Memberemail)
    BEGIN TRANSACTION A1
BEGIN

    INSERT INTO tblSALES_ORDER (MemberID, OrderDate, TotalPrice,
StoreID)
    VALUES (@mem_ID,@OrderDate, @TotalPrice, @ST_ID )
END
    COMMIT TRANSACTION A1
```

Priscilla:

```
-- populate tblProduct using stored procedure for tblPRODUCT &
tblSO_PRODUCT


CREATE PROCEDURE pnINSERT_tblProd
@ProdName varchar(50),
@ProdPrice varchar(50),
@PT_Name varchar(50),
@Sup_Name varchar(50)
AS
DECLARE @PT_ID INT
SET @PT_ID = (SELECT ProductTypeID
FROM tblPRODUCT_TYPE
WHERE ProductTypeName = @PT_Name)
DECLARE @Sup_ID INT
SET @Sup_ID = (SELECT SupplierID
FROM tblSUPPLIER
WHERE SupplierName = @Sup_Name)
BEGIN TRANSACTION P1
INSERT INTO tblPRODUCT(ProductName, ProductPrice, ProductTypeID,
SupplierID)
VALUES(@ProdName, @ProdPrice, @PT_ID, @Sup_ID)
IF @@ERROR <> 0
BEGIN
PRINT 'Stop, drop, and rollback'
ROLLBACK TRANSACTION P1
```

```
END
COMMIT TRANSACTION P1
GO

CREATE PROCEDURE pnINSERT_SO_Prod
@So_ProdName varchar(50),
@So_Price Numeric (7,2),
@S_O_tl varchar(10),
@ProddyName varchar(50),
@So_Q varchar(10)
AS
DECLARE @SO_ID INT
SET @SO_ID = (SELECT SalesOrderID
FROM tblSALES_ORDER
WHERE TotalPrice = @S_O_tl)
DECLARE @Prod_ID INT
SET @Prod_ID = (SELECT ProductID
FROM tblPRODUCT
WHERE ProductName = @ProddyName)
BEGIN TRANSACTION P1
INSERT INTO tblSO_PRODUCT(SO_ProductName, SO_ProductPrice,
SalesOrderID, ProductID, SO_Quantity)
VALUES (@So_ProdName, @So_Price, @SO_ID, @Prod_ID, @So_Q)
IF @@ERROR <> 0
BEGIN
PRINT 'Stop, drop, and rollback'
ROLLBACK TRANSACTION P1
END
COMMIT TRANSACTION P1
GO
```

Annabelle:

```
/*
2 Stored Procedures
Populating table with stored procedure
```

```
-- build a stored procedure to populate tblPO_PRODUCT
*/
-- Insert PO Product
sp_help tblPO_PRODUCT
CREATE PROCEDURE arhINSERT_PO_PRODUCT
@POProdN VARCHAR (50),
@POProdP NUMERIC (10,2),
@POQty INT,
@TotAmount NUMERIC(10,2),
@ProdN VARCHAR(50)
AS

DECLARE @PO_ID INT
SET @PO_ID = (SELECT PurchaseOrderID
FROM tblPURCHASE_ORDER
WHERE TotalAmount = @TotAmount)
DECLARE @PD_ID INT
SET @PD_ID = (SELECT ProductID
FROM tblPRODUCT
WHERE ProductName = @ProdN)
BEGIN TRANSACTION arh2
BEGIN
INSERT INTO tblPO_PRODUCT(PO_ProductName, PO_ProductPrice, Quantity,
PurchaseOrderID,
ProductID)
VALUES(@POProdN, @POProdP, @POQty, @TotAmount, @ProdN)
END
COMMIT TRANSACTION arh2

/*
Populating table with stored procedure
-- build a stored procedure to populate tblPURCHASE_ORDER
*/
-- Insert Purchase Order
CREATE PROCEDURE arhINSERT_PurchaseOrder
```

```
@EmpEmail VARCHAR(60),
@PurchDate DATE,
@TotAmount NUMERIC(10,2),
@PT_Name VARCHAR(50)
AS


DECLARE @PT_ID INT
SET @PT_ID = (SELECT ProductTypeID
FROM tblPRODUCT_TYPE
WHERE ProductTypeName = @PT_Name)

DECLARE @EmpID INT
SET @EmpID = (SELECT EmployeeID
FROM tblEMPLOYEE
WHERE EmployeeEmail = @EmpEmail)
BEGIN TRANSACTION arh1
BEGIN
INSERT INTO tblPURCHASE_ORDER(PurchaseOrderName, TotalAmount,
PurchaseDate, EmployeeID)
VALUES(@PT_Name, @TotAmount, @PurchDate, @EmpID)
END
COMMIT TRANSACTION arh1
```

Business Rules

Amy:

```
/*
enforce the following business rule with these conditions no member
under the age of 21 can purchase alcohol
Purpose behind this is because under federal regulations, many of our
stores can not sell alcohol to underage US citizens (underage is
defined us under 21 years of age)*/
CREATE FUNCTION noalcforteens()
RETURNS INT
```

```
AS
BEGIN


DECLARE @RET INTEGER = 0
IF EXISTS (SELECT *
           FROM tblMEMBER M
                JOIN tblSALES_ORDER SO ON M.MemberID = SO.MemberID
                JOIN tblSO_PRODUCT SP ON SO.SalesOrderID =
SP.SalesOrderID
                JOIN tblPRODUCT P ON SP.ProductID = P.ProductID
                JOIN tblPRODUCT_TYPE PT ON P.ProductTypeID =
PT.ProductTypeID
           WHERE PT.ProductTypeName = 'Alcohol'
           AND M.MemberDOB > (DATEADD(Year,-21,GETDATE())))
SET @RET = 1
RETURN @RET
END
GO
ALTER TABLE tblMEMBER
ADD CONSTRAINT CK_NoYounger21_Alc
CHECK (dbo.noalcforteens() = 0)
__
/*
business rule- restrict the following business rule with these
conditions
no GoldStar members can buy a vacation from Costco Travel
im making this to show that members who purchase an executive
membership have exclusive benefits that goldstar members do not have
*/
CREATE FUNCTION olderppltravel()
RETURNS INT
AS
BEGIN
```

```
DECLARE @RET INTEGER = 0
IF EXISTS (SELECT *
           FROM tblMEMBER M
               JOIN tblMEMBER_TYPE MT ON M.MemberTypeID =
MT.MemberTypeID
               JOIN tblSALES_ORDER SO ON M.MemberID = SO.MemberID
               JOIN tblSO_PRODUCT SP ON SO.SalesOrderID =
SP.SalesOrderID
               JOIN tblPRODUCT P ON SP.ProductID = P.ProductID
               JOIN tblPRODUCT_TYPE PT ON P.ProductTypeID =
PT.ProductTypeID
           WHERE PT.ProductTypeName = 'Vacation'
           AND MT.MemberTypeName = 'GoldStar')
SET @RET = 1
RETURN @RET
END
GO
ALTER TABLE tblMEMBER_TYPE
ADD CONSTRAINT CK_NoTravel_GS
CHECK (dbo.olderppltravel() = 0)
```

Ahana:

```
--Employee need be older than 18
GO

CREATE FUNCTION aroy_No18_INFO()
RETURNS INT

AS
BEGIN

DECLARE @RET INT = 0

IF
EXISTS (SELECT *
```

```
FROM tblEMPLOYEE
WHERE AGE < 18)
SET @RET = 1
RETURN @RET
END
GO

ALTER TABLE tblEMPLOYEE
ADD CONSTRAINT CK_NoYounger18_Info
CHECK (dbo.aroy_No18_INFO() = 0)
```

```
--The store type need to be either Business center or Wholesale

GO

CREATE FUNCTION aroy_BC_Or_WS_INFO()
RETURNS INT

AS
BEGIN

DECLARE @RET INT = 0

IF
EXISTS (SELECT *
FROM tblSTORE_TYPE
WHERE StoreTypeName != 'Wholesale' AND StoreTypeName != 'Business
Center')
SET @RET = 1
RETURN @RET
END
GO

ALTER TABLE tblSTORE_TYPE
```

```
ADD CONSTRAINT CK_aroy_BC_Or_WS_INFO
CHECK (dbo.aroy_BC_Or_WS_INFO()= 0)









-- No city can have more than one store

create FUNCTION aroy_NoDup_City_INFO(@StoreCity varchar(80))
RETURNS INT

AS
BEGIN

DECLARE @RET INT = 0

IF
EXISTS (SELECT *
FROM tblSTORE
WHERE StoreCity = @StoreCity)
SET @RET = 0
RETURN @RET
END
GO

ALTER TABLE tblSTORE
ADD CONSTRAINT CK_aroy_NoDup_City_Info
--DROP CONSTRAINT CK_aroy_NoDup_City_Info
CHECK (dbo.aroy_NoDup_City_INFO(StoreCity) = 0)
```

Priscilla:

```
Businesses Rules
1. No Member from Renton Costco over the age of 35 can buy products
from Chocolate Pocky
*/

CREATE FUNCTION pnNoMem35()
Returns INT
AS
BEGIN

DECLARE @RET INT = 0
IF EXISTS (SELECT *
FROM tblMEMBER M
JOIN tblSALES_ORDER SO ON M.MemberID = SO.MemberID
JOIN tblSO_PRODUCT SP ON SO.SalesOrderID = SP.SalesOrderID
JOIN tblPRODUCT P ON SP.ProductID = P.ProductID
JOIN tblSTORE S ON SO.StoreID = S.StoreID
WHERE S.StoreCity = 'Renton'
AND M.MemberDOB < DATEADD(YEAR, -35, GETDATE())
AND P.ProductName = 'Chocolate Pocky')
SET @RET = 1
RETURN @RET
END
GO
ALTER TABLE tblSALES_ORDER
ADD CONSTRAINT CK_NoRentonLikePocky
```

```
CHECK (dbo.pnNoMem35() = 0)

/*
2. Supplier name should start with F with sales quantity 10
*/

CREATE FUNCTION pnSupF10()
Returns INT
AS
BEGIN

DECLARE @RET INT = 0
IF EXISTS (SELECT *
FROM tblSUPPLIER SR
JOIN tblPRODUCT P ON SR.SupplierID = P.SupplierID
JOIN tblSO_PRODUCT SP ON P.ProductID = SP.ProductID
WHERE SR.SupplierName = '%F%'
AND SP.SO_Quantity = 10)
SET @RET = 1
RETURN @RET
END
GO
ALTER TABLE tblSO_PRODUCT
ADD CONSTRAINT CK_NoSup10
CHECK (dbo.pnSupF10() = 0)
```

Annabelle:

```
/*
2 Business Rules
1) Write the SQL to create a business rule that restricts
the following condition:
- No member who's first name starts with 'L'
- who is younger than 21 may purchase Alcohol.
-- CRAB DIES
*/
```

```sql
CREATE FUNCTION arh_BusinessRule1()
RETURNS INT
AS
BEGIN
DECLARE @RET INT = 0
IF
EXISTS (SELECT *
FROM tblMEMBER M
JOIN tblSALES_ORDER SO ON M.MemberID = SO.MemberID
JOIN tblSO_PRODUCT SP ON SO.SalesOrderID = SP.SalesOrderID
JOIN tblPRODUCT P ON SP.ProductID = P.ProductID
JOIN tblPRODUCT_TYPE PT ON P.ProductTypeID = PT.ProductTypeID
WHERE M.MemberFname LIKE 'L%'
AND M.MemberDOB > DATEADD(YEAR, -21, GETDATE())
AND PT.ProductTypeName = 'Alcohol')
SET @RET = 1
RETURN @RET
END
GO

ALTER TABLE tblSALES_ORDER
ADD CONSTRAINT CK_NoYounger21_Alch
CHECK (dbo.arh_BusinessRule1() = 0)

/*
2) Write the SQL to create a business rule that restricts
the following condition:
- An executive member type who is older than 18
- may purchase furniture
-- CRAB DIES
*/

CREATE FUNCTION arh_BusinessRule2()
RETURNS INT
AS
```

```
BEGIN
DECLARE @RET INT = 0
IF
EXISTS (SELECT *
FROM tblMEMBER M
JOIN tblMEMBER_TYPE MT ON M.MemberTypeID = MT.MemberTypeID
JOIN tblSALES_ORDER SO ON M.MemberID = SO.MemberID
JOIN tblSO_PRODUCT SP ON SO.SalesOrderID = SP.SalesOrderID
JOIN tblPRODUCT P ON SP.ProductID = P.ProductID
JOIN tblPRODUCT_TYPE PT ON P.ProductTypeID = PT.ProductTypeID
WHERE MT.MemberTypeName = 'Executive'
AND M.MemberDOB < DATEADD(YEAR, -18, GETDATE())
AND PT.ProductTypeName = 'Furniture')
SET @RET = 1
RETURN @RET
END
GO

ALTER TABLE tblMEMBER_TYPE
ADD CONSTRAINT CK_NoOlder18_Furn
CHECK (dbo.arh_BusinessRule2() = 0)
```

Computed Columns

Amy:

```
/* Write the SQL to determine the Lifetime Amount
  an Executive member spent in 2019
Now, write the SQL to create a computed column that returns
the TotAmt for previous 5 years.
*/

SELECT M.MemberID, M.MemberFname, M.MemberLname, SUM(M.LifetimeAmount)
AS TotalAmt
FROM tblMEMBER M
JOIN tblMEMBER_TYPE MT ON M.MemberTypeID = MT.MemberTypeID
```

```
JOIN tblSALES_ORDER SO ON M.MemberID = SO.MemberID
WHERE SO.OrderDate LIKE '2019'
AND MT.MemberTypeName = 'Executive'
GROUP BY M.MemberID, M.MemberFname, M.MemberLname
HAVING SUM(M.LifetimeAmount) <= 1000

CREATE FUNCTION totalamtexecmembers(@PK INT)
RETURNS NUMERIC(10,2)
AS
BEGIN
DECLARE @RET NUMERIC(10,2) =  (
SELECT SUM(LifetimeAmount)
FROM tblMEMBER M
JOIN tblMEMBER_TYPE MT ON M.MemberTypeID = MT.MemberTypeID
JOIN tblSALES_ORDER SO ON M.MemberID = SO.MemberID
WHERE SO.OrderDate > DateAdd(Year, -5, GetDate())
AND M.MemberID = @PK)
RETURN @RET
END
GO
ALTER TABLE tblMEMBER
ADD TotalAmt AS (dbo.totalamtexecmembers(MemberID))
__
Write the SQL to determine the Lifetime Orders
  A GoldStar member spent in 2017
Now, write the SQL to create a computed column that returns
the TotAmt for previous 3 years.
SELECT M.MemberID, M.MemberFname, M.MemberLname, SUM(M.LifetimeOrders)
AS TotalOrdrs
FROM tblMEMBER M
JOIN tblMEMBER_TYPE MT ON M.MemberTypeID = MT.MemberTypeID
JOIN tblSALES_ORDER SO ON M.MemberID = SO.MemberID
WHERE SO.OrderDate LIKE '2017'
AND MT.MemberTypeName = 'GoldStar'
GROUP BY M.MemberID, M.MemberFname, M.MemberLname
```

```
HAVING SUM(M.LifetimeOrders) <= 1000

CREATE FUNCTION totalordrsgdstrmembers(@PK INT)
RETURNS NUMERIC(10,2)
AS
BEGIN
DECLARE @RET NUMERIC(10,2) =  (
SELECT SUM(LifetimeOrders)
FROM tblMEMBER M
JOIN tblMEMBER_TYPE MT ON M.MemberTypeID = MT.MemberTypeID
JOIN tblSALES_ORDER SO ON M.MemberID = SO.MemberID
WHERE SO.OrderDate > DateAdd(Year, -3, GetDate())
AND M.MemberID = @PK)
RETURN @RET
END
GO
ALTER TABLE tblMEMBER
ADD TotalOrdrs AS (dbo.totalordrsgdstrmembers(MemberID))
```

Ahana:

```
-- UDF THAT CREATES A COMPUTED COL IN EMPLOYEE TABLE
-- COMPUTES AGES OF EMPLOYEES
GO
CREATE FUNCTION aroy_COMPUTE_AGE(@EmployeeDOB Date)

RETURNS INT
AS
BEGIN
DECLARE @AGE INT =  (

DATEDIFF(YEAR, @EmployeeDOB, GetDate()) )

RETURN @AGE
END
GO
ALTER TABLE tblEMPLOYEE
```

```
ADD AGE AS (dbo.aroy_COMPUTE_AGE(EmployeeDOB))

SELECT * FROM tblEMPLOYEE

-- UDF THAT CREATES A COMPUTED COL IN EMPLOYEE TABLE
-- COMPUTES THE FULL NAME OF THE EMPLOYEES



GO
CREATE FUNCTION aroy_COMPUTE_Employee_Full_Name(@EmployeeFname
varchar(30), @EmployeeLname varchar(30))

RETURNS VARCHAR(70)
AS
BEGIN
DECLARE @FULL_NAME VARCHAR(70) =  (
@EmployeeFname + ' ' + @EmployeeLname
 )

RETURN @FULL_NAME
END
GO
ALTER TABLE tblEMPLOYEE
ADD EmployeeFullName AS
(dbo.aroy_COMPUTE_Employee_Full_Name(EmployeeFname, EmployeeLname))
```

Priscilla:

```
Computed columns
```

```sql
/*
1. What is the min average orders does costco wholesale get from their
suppliers?
*/
SELECT * FROM tblMEMBER

SELECT TOP 1 S.StoreID, S.StoreName, S.StoreCity,
MIN(SR.AverageLeadOrders)
FROM tblStore S
JOIN tblSALES_ORDER SO ON S.StoreID = SO.StoreID
JOIN tblSO_PRODUCT SP ON SO.SalesOrderID = SP.SalesOrderID
JOIN tblPRODUCT P ON SP.ProductID = P.ProductID
JOIN tblSUPPLIER SR ON P.SupplierID = SR.SupplierID
WHERE S.StoreName = 'Costco Wholesale'
GROUP BY S.StoreID, S.StoreName, S.StoreCity


/*
2. Write a SQL query to determine which members have a total of at
least 1000 products purchased
and have a exective membership in Redmond
*/

SELECT M.MemberID, M.MemberFName, M.MemberLname, MT.MemberTypeName,
S.StoreCity, SUM(PO.PO_Quantity) AS SumQuant
FROM tblMEMBER_TYPE MT
JOIN tblMEMBER M ON MT.MemberTypeID = M.MemberTypeID
JOIN tblSALES_ORDER SO ON M.MemberID = SO.MemberID
JOIN tblSTORE S ON SO.StoreID = S.StoreID
JOIN tblSO_PRODUCT SP ON SO.SalesOrderID = SP.SalesOrderID
JOIN tblPRODUCT P ON SP.ProductID = P.ProductID
JOIN tblPO_PRODUCT PP ON P.ProductID = PP.ProductID
JOIN tblPURCHASE_ORDER PO ON PP.PurchaseOrderID = PO.PurchaseOrderID
WHERE MT.MemberTypeName = 'Executive'
AND S.StoreCity = 'Redmond'
```

```
GROUP BY M.MemberID, M.MemberFName, M.MemberLname, MT.MemberTypeName,
S.StoreCity
HAVING SUM(PO.PO_Quantity) >= 1000
```

Annabelle:

```
-- 2 Computed Columns
-- 1) Write the SQL to determine the Lifetime Amount
-- an Executive member during the 1990's
SELECT M.MemberID, SUM(LifetimeAmount) AS TotAmt
FROM tblMEMBER M
JOIN tblMEMBER_TYPE MT ON M.MemberTypeID = MT.MemberTypeID
JOIN tblSALES_ORDER SO ON M.MemberID = SO.MemberID
WHERE MT.MemberTypeName = 'Executive'
AND SO.OrderDate LIKE '199%'
GROUP BY M.MemberID

-- Now, write the SQL to create a computed column that returns
-- the TotAmt for previous 5 years.
GO
CREATE FUNCTION arhTotAmt5Years (@PK INT)
RETURNS NUMERIC(10,2)
AS
BEGIN
DECLARE @RET NUMERIC(10,2) = (
SELECT SUM(LifetimeAmount)
FROM tblMEMBER M
JOIN tblMEMBER_TYPE MT ON M.MemberTypeID = MT.MemberTypeID
JOIN tblSALES_ORDER SO ON M.MemberID = SO.MemberID
WHERE SO.OrderDate > DATEADD(YEAR, -5, GETDATE())
AND M.MemberID = @PK)
RETURN @RET
END
GO

ALTER TABLE tblMEMBER
```

```
ADD arhCalc1 AS (dbo.arhTotAmt5Years(MemberID))

-- 2) Write the SQL to determine the LifetimeAmount a goldstar member
-- spend in purchasing Alcohol between 2012-2022

SELECT M.MemberID, SUM(LifetimeAmount) AS TotAmount,
YEAR(SO.OrderDate) AS Year
FROM tblMEMBER M
JOIN tblMEMBER_TYPE MT ON M.MemberTypeID = MT.MemberTypeID
JOIN tblSALES_ORDER SO ON M.MemberID = SO.MemberID
JOIN tblSO_PRODUCT SP ON SO.SalesOrderID = SP.SalesOrderID
JOIN tblPRODUCT P ON SP.ProductID = P.ProductID
JOIN tblPRODUCT_TYPE PT ON P.ProductTypeID = PT.ProductTypeID
WHERE MT.MemberTypeName = 'GoldStar'
AND PT.ProductTypeName = 'Alcohol'
AND YEAR(SO.OrderDate) BETWEEN 2012 AND 2022
GROUP BY M.MemberID, YEAR(SO.OrderDate)

-- now, write the SQL to create a computed column that returns the
-- TotAmount for previos 10 years
-- CRAB
CREATE FUNCTION arhTotAmount_10_Years (@PK INT)
RETURNS NUMERIC (10,2)
AS
BEGIN
DECLARE @RET NUMERIC(10,2)

SELECT @RET = SUM(LifetimeAmount)
FROM tblMEMBER M
JOIN tblMEMBER_TYPE MT ON M.MemberTypeID = MT.MemberTypeID
JOIN tblSALES_ORDER SO ON M.MemberID = SO.MemberID
JOIN tblSO_PRODUCT SP ON SO.SalesOrderID = SP.SalesOrderID
JOIN tblPRODUCT P ON SP.ProductID = P.ProductID
JOIN tblPRODUCT_TYPE PT ON P.ProductTypeID = PT.ProductTypeID
WHERE M.MemberID = @PK
```

```
AND MT.MemberTypeName = 'GoldStar'
AND PT.ProductTypeName = 'Alcohol'
AND YEAR(SO.OrderDate) BETWEEN YEAR(GETDATE()) - 10 AND
YEAR(GETDATE())

RETURN @RET
END
GO

ALTER TABLE tblMEMBER
ADD arhCalc2 AS (dbo.arhTotAmount_10_Years(MemberID))
```

Complex Queries

Amy:

```
/*
multiple join statements - Determine the goldstar members with L in
their first name that bought cheddar chips from the store costco
wholesale business in Issaquah
*/


SELECT M.MemberID,M.MemberFname, M.MemberLname
FROM tblMEMBER_TYPE MT
   JOIN tblMEMBER M ON MT.MemberTypeID = M.MemberTypeID
   JOIN tblSALES_ORDER SO ON M.MemberID = SO.MemberID
   JOIN tblSTORE S ON SO.StoreID = S.StoreID
   JOIN tblSO_PRODUCT SOP ON SO.SalesOrderID = SOP.SalesOrderID
WHERE MT.MemberTypeName = 'GoldStar'
AND M.MemberFname LIKE '%L%'
AND SOP.SO_ProductName = 'Cheddar Chips'
AND S.StoreCity = 'Issaquah'
AND S.StoreName = 'Costco Wholesale'
GROUP BY M.MemberID,M.MemberFname, M.MemberLname
—
```

```
/*
aggregate functions - how many members purchased spent at least $50 on
items from a Bellevue Store in 2019
*/

SELECT M.MemberID,M.MemberFname, M.MemberLname, SUM(SOP.SO_SalesTotal)
AS Amt
FROM tblMEMBER_TYPE MT
   JOIN tblMEMBER M ON MT.MemberTypeID = M.MemberTypeID
   JOIN tblSALES_ORDER SO ON M.MemberID = SO.MemberID
   JOIN tblSTORE S ON SO.StoreID = S.StoreID
   JOIN tblSO_PRODUCT SOP ON SO.SalesOrderID = SOP.SalesOrderID
WHERE SO.OrderDate LIKE '2019'
AND S.StoreCity = 'Bellevue'
GROUP BY M.MemberID,M.MemberFname, M.MemberLname
HAVING SUM(SOP.SO_SalesTotal) >= 50
```

Ahana:

```
--This procdure takes an order date as an input and lists the products
that sold that day that have a quanity sold
-- is greater than 5.This SP list the store name, product name, and
sum of quantity in desending order
EXEC aroy_Get_prodSold_byDate  @Orderdate = '2023-07-13'
GO
ALter PROCEDURE aroy_Get_prodSold_byDate
@Orderdate Date

AS
BEGIN
BEGIN Try
    Select P.ProductName, S.StoreName, SO.OrderDate, SUM(SO_Quantity)
AS 'NumSold'
    From tblSO_PRODUCT as SOP
    Join tblSALES_ORDER as SO on SOP.SalesOrderID = SO.SalesOrderID
    Join tblPRODUCT as P on SOP.ProductID = P.ProductID
```

```
    Join tblSTORE as S on SO.StoreID = S.StoreID
    Where SO.OrderDate = @Orderdate
    Group By P.ProductName, S.StoreName, SO.OrderDate
    Having SUM(SO_Quantity) > 5
    Order By SUM(SO_Quantity) DESC
END try
Begin CATCH
    SELECT
    ERROR_NUMBER() AS ErrorNumber
    ,ERROR_SEVERITY() AS ErrorSeverity
    ,ERROR_STATE() AS ErrorState
    ,ERROR_PROCEDURE() AS ErrorProcedure
    ,ERROR_LINE() AS ErrorLine
    ,ERROR_MESSAGE() AS ErrorMessage;
END CATCH
END
GO


-- This procedure takes the member email as a parameter and finds and
returns the total sum of sales order
-- for that member in the last one month

DECLARE @member_order_total_1 NUMERIC(10,2)
EXEC aroy_Get_Member_Order_Total @Memberemail =
'Karima.Butterworth303@bpdelawarematerials.com' ,
@member_order_total = @Member_Order_Total_1 OUTPUT
```

```
GO
ALTER PROCEDURE aroy_Get_Member_Order_Total
@Memberemail VARCHAR(80),
@Member_Order_Total NUMERIC(10,2) OUTPUT

AS
BEGIN
BEGIN TRY
SET @Member_Order_Total = (
SELECT  SUM(SO_SalesTotal) --member order total
```

```
FROM tblSALES_ORDER AS  SO
JOIN tblSO_PRODUCT AS SOP ON SO.SalesOrderID = SOP.SalesOrderID
JOIN tblMEMBER AS M ON SO.MemberID = M.MemberID
WHERE M.MemberEmail = @Memberemail AND SO.OrderDate < = GETDATE() AND
SO.OrderDate > DATEADD(MONTH, -1, GETDATE())
GROUP BY SO.MemberID , SO.OrderDate
)
END TRY
BEGIN CATCH
SET @Member_Order_Total = -1
END CATCH
END
```

Priscilla: combined with computed columns
Annabelle:

```
/*
2 Complex Queries
- Top Command with ORDER BY
1) Write the SQL to determine which Executive member is the youngest
*/
CREATE VIEW OldestExecMember
```

```
AS
SELECT TOP 1 M.MemberID, M.MemberDOB
FROM tblMEMBER M
JOIN tblMEMBER_TYPE MT ON M.MemberTypeID = MT.MemberTypeID
WHERE MT.MemberTypeName = 'Executive'
ORDER BY M.MemberDOB DESC

SELECT *
FROM OldestExecMember

/*
2) Write the SQL to determine the LifetimeAmount a goldstar member
spend in purchasing Alcohol between 2012-2022 that also
spent a minimum of $200 on Health & Personal in Costco wholesale
located
in Redmond.
*/
CREATE VIEW GoldStarMemSpend
AS
SELECT A.MemberID, A.TotAmount, A.Year, B.TotalSpent
FROM
(SELECT M.MemberID, SUM(LifetimeAmount) AS TotAmount,
YEAR(SO.OrderDate) AS Year
FROM tblMEMBER M
JOIN tblMEMBER_TYPE MT ON M.MemberTypeID = MT.MemberTypeID
JOIN tblSALES_ORDER SO ON M.MemberID = SO.MemberID
JOIN tblSO_PRODUCT SP ON SO.SalesOrderID = SP.SalesOrderID
JOIN tblPRODUCT P ON SP.ProductID = P.ProductID
JOIN tblPRODUCT_TYPE PT ON P.ProductTypeID = PT.ProductTypeID
WHERE MT.MemberTypeName = 'GoldStar'
AND PT.ProductTypeName = 'Alcohol'
AND YEAR(SO.OrderDate) BETWEEN 2012 AND 2022
GROUP BY M.MemberID, YEAR(SO.OrderDate)) A
JOIN
(SELECT M.MemberID, SUM(LifetimeAmount) AS TotalSpent
```

```
FROM tblMEMBER M
JOIN tblMEMBER_TYPE MT ON M.MemberTypeID = MT.MemberTypeID
JOIN tblSALES_ORDER SO ON M.MemberID = SO.MemberID
JOIN tblSO_PRODUCT SP ON SO.SalesOrderID = SP.SalesOrderID
JOIN tblPRODUCT P ON SP.ProductID = P.ProductID
JOIN tblPRODUCT_TYPE PT ON P.ProductTypeID = PT.ProductTypeID
JOIN tblSTORE S ON SO.StoreID = S.StoreID
JOIN tblSTORE_TYPE ST ON S.StoreTypeID = ST.StoreTypeID
WHERE MT.MemberTypeName = 'GoldStar'
AND PT.ProductTypeName = 'Health & Personal'
AND S.StoreCity = 'Redmond'
AND ST.StoreTypeName = 'Costco Wholesale'
GROUP BY M.MemberID
HAVING SUM(LifetimeAmount) >= 200) B
ON A.MemberID = B.MemberID;
```