CREATE TABLE Employee (

EmployeeID int NOT NULL IDENTITY(1,1),

EmployeeName varchar(255) NOT NULL,

Salary float NOT NULL,

PRIMARY KEY (EmployeeID)

);

CREATE TABLE Users (

UserID int NOT NULL IDENTITY(1,1),

UserName varchar(255) NOT NULL,

PRIMARY KEY (UserID)

);

CREATE TABLE Shops (

ShopName varchar(255) NOT NULL,

PRIMARY KEY (ShopName)

);

CREATE TABLE Products (

ProductName varchar(255) NOT NULL,

Maker varchar(255) NOT NULL,

Category varchar(255) NOT NULL,

PRIMARY KEY (ProductName)

);

CREATE TABLE Orders (

OrderID int NOT NULL IDENTITY(1,1),

Shipping\_address varchar(255) NOT NULL,

Date\_time datetime NOT NULL,

UserID int NOT NULL,

PRIMARY KEY (OrderID),

FOREIGN KEY (UserID) REFERENCES Users(UserID)

ON UPDATE CASCADE

ON DELETE CASCADE,

);

CREATE TABLE Products\_in\_Orders (

ProductName varchar(255) NOT NULL,

Price float NOT NULL,

Quantity int NOT NULL,

Delivery\_date datetime,

DeliveryStatus varchar(255) NOT NULL DEFAULT 'being processed',

ShopName varchar(255) NOT NULL,

OrderID int NOT NULL,

PRIMARY KEY (ProductName, OrderID, ShopName),

FOREIGN KEY (ProductName) REFERENCES Products(ProductName)

ON UPDATE CASCADE

ON DELETE CASCADE,

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)

ON UPDATE CASCADE

ON DELETE CASCADE,

FOREIGN KEY (ShopName) REFERENCES Shops(ShopName)

ON UPDATE CASCADE

ON DELETE CASCADE

);

CREATE TABLE Feedback (

UserID int NOT NULL,

Rating float NOT NULL,

Date\_time datetime NOT NULL,

FeedbackComments varchar(255),

ProductName varchar(255) NOT NULL,

ShopName varchar(255) NOT NULL,

OrderID int NOT NULL,

PRIMARY KEY (UserID, ProductName),

FOREIGN KEY (UserID) REFERENCES Users(UserID)

ON UPDATE CASCADE

ON DELETE CASCADE,

FOREIGN KEY (ProductName, OrderID, ShopName) REFERENCES Products\_in\_Orders(ProductName, OrderID, ShopName)

ON UPDATE NO ACTION

ON DELETE NO ACTION

);

CREATE TABLE Complaint (

ComplaintID int NOT NULL IDENTITY(1,1),

ComplaintText varchar(255) NOT NULL,

ComplaintStatus varchar(255) NOT NULL DEFAULT 'pending',

Filled\_date\_time datetime NOT NULL,

Handled\_date\_time datetime,

EmployeeID int,

UserID int NOT NULL,

PRIMARY KEY (ComplaintID),

FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)

ON UPDATE CASCADE

ON DELETE CASCADE,

FOREIGN KEY (UserID) REFERENCES Users(UserID)

ON UPDATE CASCADE

ON DELETE CASCADE

);

CREATE TABLE Complaints\_on\_Shops (

ComplaintID int NOT NULL,

ShopName varchar(255) NOT NULL,

PRIMARY KEY (ComplaintID),

FOREIGN KEY (ComplaintID) REFERENCES Complaint(ComplaintID)

ON UPDATE CASCADE

ON DELETE CASCADE,

FOREIGN KEY (ShopName) REFERENCES Shops(ShopName)

ON UPDATE CASCADE

ON DELETE CASCADE

);

CREATE TABLE Complaints\_on\_Orders (

ComplaintID int NOT NULL,

OrderID int NOT NULL,

PRIMARY KEY (ComplaintID),

FOREIGN KEY (ComplaintID) REFERENCES Complaint(ComplaintID)

ON UPDATE NO ACTION

ON DELETE NO ACTION,

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)

ON UPDATE NO ACTION

ON DELETE NO ACTION

);

CREATE TABLE Products\_in\_Shops (

ProductName varchar(255) NOT NULL,

Price float NOT NULL,

Quantity int NOT NULL,

ShopName varchar(255) NOT NULL,

PRIMARY KEY (ProductName, ShopName),

FOREIGN KEY (ProductName) REFERENCES Products(ProductName)

ON UPDATE CASCADE

ON DELETE CASCADE,

);

CREATE TABLE Price\_History (

ProductName varchar(255) NOT NULL,

Price float NOT NULL,

sDate date NOT NULL,

eDate date,

ShopName varchar(255) NOT NULL,

PRIMARY KEY (ProductName, ShopName, sDate),

FOREIGN KEY (ProductName, ShopName) REFERENCES Products\_in\_Shops(ProductName, ShopName)

ON UPDATE CASCADE

ON DELETE CASCADE

);

CREATE TABLE Shops\_Products\_PriceVar (

ShopName varchar(255) NOT NULL,

ProductName varchar(255) NOT NULL,

sDate date NOT NULL,

eDate date,

Price\_variation float NOT NULL,

PRIMARY KEY (ProductName, ShopName),

FOREIGN KEY (ProductName, ShopName) REFERENCES Products\_in\_Shops(ProductName, ShopName)

ON UPDATE CASCADE

ON DELETE CASCADE,

FOREIGN KEY (ShopName) REFERENCES Shops(ShopName)

ON UPDATE CASCADE

ON DELETE CASCADE

);

##################################################################

ALTER TABLE Employee

ADD CHECK (Salary>0);

ALTER TABLE Feedback

ADD CHECK (Rating BETWEEN 1 AND 5);

ALTER TABLE Products\_in\_Orders

ADD CHECK (Price >= 0 and Quantity >= 0);

ALTER TABLE Products\_in\_Shops

ADD CHECK (Price >= 0 and Quantity >= 0);

###################################################################

CREATE TRIGGER CheckRefundEligible

ON Products\_in\_Orders

AFTER UPDATE

AS

BEGIN

IF EXISTS(SELECT x.Delivery\_date FROM inserted x

WHERE x.DeliveryStatus = 'returned' AND DATEDIFF(DAY, x.Delivery\_date, GETDATE()) > 30

GROUP BY x.Delivery\_date

HAVING COUNT(\*) > 0)

ROLLBACK TRANSACTION

END

######################################################################

Q1

SELECT AVG(price) as 'Average\_Price'

FROM Price\_History

WHERE ProductName = 'iPhone XS'

AND (eDate BETWEEN '2020-08-01' AND '2020-08-31')

OR (sDate BETWEEN '2020-08-01' AND '2020-08-31')

OR (sDate <= '2020-08-31' AND eDate is null)

#######################################################################

Q2

SELECT ProductName, AVG(Rating) as 'Average Rating'

FROM Feedback as F1

WHERE F1.ProductName IN (SELECT F2.ProductName

FROM Feedback as F2

WHERE F2.Rating = 5

AND F2.Date\_time BETWEEN '2020-08-01 00:00:00' AND '2020-08-31 23:59:59'

GROUP BY F2.ProductName

HAVING COUNT(F2.Rating) >= 100)

GROUP BY F1.ProductName

ORDER BY [Average Rating] DESC;

#######################################################################

Q3

SELECT Products\_in\_Orders.ProductName, AVG(DATEDIFF(day, Orders.Date\_time, Products\_in\_Orders.Delivery\_date)) as 'Avg Delivery Time'

FROM Products\_in\_Orders, Orders

WHERE Products\_in\_Orders.OrderID = Orders.OrderID

AND Orders.Date\_time >= '2020-06-01 00:00:00'

AND Products\_in\_Orders.DeliveryStatus = 'delivered'

GROUP BY Products\_in\_Orders.ProductName;

#######################################################################

Q4

SELECT C.EmployeeID, AVG(DATEDIFF(MINUTE, C.Filled\_date\_time, C.Handled\_date\_time)) as 'Average\_Latency'

INTO C2

FROM Complaint as C

GROUP BY C.EmployeeID

SELECT E.EmployeeName, C2.Average\_Latency

FROM C2 as C2, Employee as E

WHERE C2.Average\_Latency IN(SELECT MIN(Average\_Latency) FROM C2)

AND E.EmployeeID = C2.EmployeeID

DROP TABLE C2;

#######################################################################

Q5

SELECT P.ProductName

INTO S1

FROM Products as P

WHERE P.Maker = 'Samsung';

SELECT PS.ProductName, COUNT(PS.ShopName) as NumOfShops

FROM Products\_in\_Shops as PS

WHERE PS.ProductName IN (SELECT \*

FROM S1)

GROUP BY PS.ProductName

DROP TABLE S1;

#######################################################################

Q6

SELECT PO.ShopName, SUM(PO.Price \* PO.Quantity) as Rev

INTO RevTable

FROM Products\_in\_Orders as PO, Orders as O

WHERE Date\_time BETWEEN '2020-08-01' AND '2020-08-31'

GROUP BY PO.ShopName;

SELECT ShopName, Rev

FROM RevTable

WHERE Rev IN (SELECT MAX(Rev)

FROM RevTable)

DROP TABLE RevTable

#######################################################################

Q7

SELECT UserID, COUNT(UserID) as 'Num\_Complaints'

INTO User\_Complaints

FROM Complaint

GROUP BY UserID;

SELECT O.UserID, PO.OrderID, ProductName, Price

INTO Users\_Bought

FROM Orders as O, Products\_in\_Orders as PO

WHERE O.OrderID = PO.OrderID

AND PO.OrderID IN (SELECT OrderID

FROM Orders O

WHERE O.UserID IN (SELECT UserID

FROM User\_Complaints as UC

WHERE UC.Num\_Complaints IN (SELECT MAX(Num\_Complaints)

FROM User\_Complaints)));

SELECT UserID, MAX(Price) as Price

INTO Most\_Ex

FROM Users\_Bought

GROUP BY UserID;

SELECT Users\_Bought.UserID, Users\_Bought.ProductName, Users\_Bought.Price

FROM Users\_Bought, Most\_Ex

WHERE Most\_Ex.UserID = Users\_Bought.UserID

AND Most\_Ex.Price = Users\_Bought.Price;

DROP TABLE User\_Complaints;

DROP TABLE Users\_Bought;

DROP TABLE Most\_Ex;

#####################################################################

Q8

SELECT O.UserID, PO.ProductName

INTO User\_Bought

FROM Orders as O, Products\_in\_Orders as PO

WHERE O.OrderID = PO.OrderID

SELECT UB.UserID, P.ProductName

INTO User\_Not\_Bought

FROM User\_Bought as UB, Products as P

WHERE P.ProductName IN (

SELECT ProductName

FROM User\_Bought

WHERE UB.ProductName <> P.ProductName)

EXCEPT

SELECT \*

FROM User\_Bought

SELECT \*

FROM User\_Not\_Bought

SELECT TOP 5 ProductName, COUNT(ProductName) AS 'Num\_Product'

INTO Aug\_Top

FROM Products\_in\_Orders as PO, Orders as O

WHERE PO.OrderID = O.OrderID

AND O.date\_time BETWEEN '2020-08-01 00:00:00' AND '2020-08-31 23:59:59'

GROUP BY ProductName

SELECT\*

FROM Aug\_Top;

SELECT UserID, User\_Not\_Bought.ProductName

FROM Aug\_Top, User\_Not\_Bought

WHERE Aug\_Top.ProductName = User\_Not\_Bought.ProductName

DROP TABLE User\_Bought;

DROP TABLE User\_Not\_Bought;

DROP TABLE Aug\_Top;

####################################################################

Q9

SELECT PO.ProductName, DATEADD(MONTH, DATEDIFF(MONTH, 0, O.Date\_time), 0) as 'Year\_Month', SUM(Quantity) as 'Total\_Sold'

INTO Monthly\_Sold

FROM Orders as O

JOIN Products\_in\_Orders as PO ON O.OrderID = PO.OrderID

GROUP BY DATEADD(MONTH, DATEDIFF(MONTH, 0, O.Date\_time), 0), PO.ProductName;

SELECT Monthly\_Sold.ProductName, FORMAT(Year\_Month, 'yyyy-MM') as 'Increasing\_3\_Month'

FROM Monthly\_Sold

WHERE Monthly\_Sold.ProductName IN (SELECT MS2.ProductName

FROM Monthly\_Sold as MS2

WHERE Monthly\_Sold.ProductName = MS2.ProductName

AND DATEDIFF(MONTH, Monthly\_Sold.Year\_Month, MS2.Year\_Month) = 1

AND Monthly\_Sold.Year\_Month < MS2.Year\_Month

AND Monthly\_Sold.Total\_Sold < MS2.Total\_Sold

AND MS2.ProductName IN (SELECT MS3.ProductName

FROM Monthly\_Sold as MS3

WHERE MS2.ProductName = MS3.ProductName

AND DATEDIFF(MONTH, MS2.Year\_Month, MS3.Year\_Month) = 1

AND MS2.Year\_Month < MS3.Year\_Month

AND MS2.Total\_Sold < MS3.Total\_Sold))

DROP TABLE Monthly\_Sold;

########################################################################

A1

SELECT E.EmployeeName, COUNT(C.EmployeeID) as 'Num\_Of\_Handled\_Complaints'

FROM Employee as E, Complaint as C

WHERE C.ComplaintStatus = 'addressed'

AND E.EmployeeID = C.EmployeeID

GROUP BY E.EmployeeName

SELECT TOP 5 PO.ProductsName, COUNT(PO.ProductsName) as 'Num\_Purchased'

FROM Products\_in\_Orders as PO, Orders as O

WHERE PO.OrderID = O.OrderID

AND O.Date\_time BETWEEN '2020-08-01 00:00:00' AND '2020-08-31 23:59:59'

AND O.UserID = SOME (SELECT UserID FROM Users)

GROUP BY PO.ProductsName

ORDER BY ‘Num\_Purchased’ DESC;