NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

CZ2007: INTRODUCTION TO DATABASES

SQL Submissions

SS2 Group 4:

Lim Wi Teow *(U1921765G)*

Taneja Parthasarthi *(U1722927B)*

Chong Zhe Ming *(U1920757K)*

Soh Qian Yi *(U1922306C)*

Jacob Law Zhen *(U1922430D)*

Zhu Weiji *(U1922876G)*

Poh Kai Kiat *(U1922819C)*

Submission Date: 25/9/2020

Table Creation SQL

```
-- TABLE 1
CREATE TABLE Shops
(
   ShopName VARCHAR (200) NOT NULL,
   PRIMARY KEY (ShopName),
);

-- TABLE 2
CREATE TABLE Users
(
   UserId INT NOT NULL,
   UserName VARCHAR(200) NOT NULL,
   PRIMARY KEY (UserId),
);

-- TABLE 3
CREATE TABLE Orders
(
   OrderId INT NOT NULL,
   UserId INT,
   ShippingAddress VARCHAR (200),
   TotalShippingCost FLOAT,
   OrderDateTime datetime,
   PRIMARY KEY (OrderId),
   FOREIGN KEY (UserId) REFERENCES Users(UserId)
   ON DELETE CASCADE
   ON UPDATE CASCADE,
);

-- TABLE 7
CREATE TABLE Employees
(
   EID INT NOT NULL,
   EmployeeName VARCHAR (200),
   Salary INT CHECK(Salary > 0),
   PRIMARY KEY (EID),
);

-- TABLE 8
CREATE TABLE Products
(
   PID INT NOT NULL IDENTITY(1,1),
   ProductName VARCHAR (200),
```

```
    Maker VARCHAR (200),
    Category VARCHAR (200),
    PRIMARY KEY (PID)
);

-- TABLE 4
CREATE TABLE Complaints
(
    ComplaintID INT NOT NULL IDENTITY(1,1),
    UserID INT,
    EID INT,
    HandledDateTime datetime DEFAULT NULL,
    ComplaintText varchar(1000) NOT NULL,
    ComplaintStatus varchar(20) NOT NULL,
    CHECK (ComplaintStatus IN('pending', 'being handled', 'addressed')),
    FilledDateTime datetime,
    PRIMARY KEY (ComplaintID),
    FOREIGN KEY (UserId) REFERENCES USERS (UserId)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    FOREIGN KEY(EID) REFERENCES Employees(EID)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
);

-- TABLE 5
CREATE TABLE ComplaintsOnShops
(
    ComplaintID INT,
    ShopName VARCHAR (200),
    PRIMARY KEY (ComplaintID),
    FOREIGN KEY (ComplaintID) REFERENCES Complaints(ComplaintID)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    FOREIGN KEY (ShopName) REFERENCES Shops(ShopName)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
);

-- -- TABLE  6
CREATE TABLE ComplaintsOnOrders
(
    ComplaintID INT,
    OrderId INT,
```

```
    PRIMARY KEY (ComplaintID),
    FOREIGN KEY (ComplaintID) REFERENCES Complaints(ComplaintID),
    FOREIGN KEY (OrderId) REFERENCES Orders(OrderId)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
);


-- TABLE 9
CREATE TABLE ProductsInOrder
(
    PID INT,
    OrderId INT,
    ShopName VARCHAR (200) NOT NULL,
    OrderStatus VARCHAR (50) NOT NULL CHECK (OrderStatus IN('being processed',
'shipped', 'delivered')) DEFAULT ('being processed'),
    DeliveryDate DATETIME DEFAULT NULL,
    Price FLOAT NOT NULL CHECK (Price > 0.00),
    Quantity INT NOT NULL CHECK (Quantity > 0),
    PRIMARY KEY
    (PID, OrderId),
    FOREIGN KEY
    (ShopName) REFERENCES Shops
    (ShopName)
    ON
    DELETE CASCADE
    ON
    UPDATE CASCADE,
    FOREIGN KEY (PID) REFERENCES Products (PID)
    ON
    DELETE CASCADE
    ON
    UPDATE CASCADE,
    FOREIGN KEY (OrderId) REFERENCES Orders (OrderId)
    ON
    DELETE CASCADE
    ON
    UPDATE CASCADE,
);

-- TABLE 10
CREATE TABLE ProductsInShops
(
    ShopName VARCHAR (200),
```

```
    PID INT,
    Price FLOAT CHECK (Price >= 0.00),
    Quantity INT CHECK (Quantity >= 0),
    PRIMARY KEY (ShopName, PID),
    FOREIGN KEY (ShopName) REFERENCES Shops (ShopName)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    FOREIGN KEY (PID) REFERENCES Products (PID)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

-- TABLE 12
CREATE TABLE Feedback
(
    UserId INT,
    PID INT,
    FeedbackDateTime DATETIME NOT NULL,
    Rating INT NOT NULL CHECK(Rating <= 5 AND Rating >= 1),
    Comment VARCHAR (100),
    PRIMARY KEY (UserId, PID, FeedbackDateTime),
    FOREIGN KEY (UserId) REFERENCES Users(UserId)
ON DELETE CASCADE
ON UPDATE CASCADE,
    FOREIGN KEY (PID) REFERENCES Products (PID
ON DELETE CASCADE
ON UPDATE CASCADE,
);
```

Trigger Queries for Complaints

```
-- Since both ComplaintsOnShops and ComplaintsOnOrders are both ensuring referential
-- integrity onto the same value Complaints.ComplaintID, SQL Server does not allow us to
-- CASCADE CHANGES ON UPDATE AND ON DELETE on both at once
-- Hence we have created a trigger to update one of the tables(ComplaintsOnOrders) on change


CREATE TRIGGER trigger_complaints
ON Complaints
AFTER DELETE
AS
BEGIN
   DELETE FROM ComplaintsOnOrders WHERE
   ComplaintsOnOrders.ComplaintID IN
   (
      SELECT ComplaintID
   FROM Complaints
   );
END
```

Populating the tables manually

```
--  Shops (ShopName)
INSERT INTO SS2G4.dbo.Shops
VALUES('Shop 1');
INSERT INTO SS2G4.dbo.Shops
VALUES('Shop 2');
INSERT INTO SS2G4.dbo.Shops
VALUES('Shop 3');
INSERT INTO SS2G4.dbo.Shops
VALUES('Shop 4');
INSERT INTO SS2G4.dbo.Shops
VALUES('Shop 5');
INSERT INTO SS2G4.dbo.Shops
VALUES('Shop 6');
INSERT INTO SS2G4.dbo.Shops
VALUES('Shop 7');
INSERT INTO SS2G4.dbo.Shops
VALUES('Shop 8');
INSERT INTO SS2G4.dbo.Shops
VALUES('Shop 9');
INSERT INTO SS2G4.dbo.Shops
VALUES('Shop 10');

-- Users (UserID, UserName)
INSERT INTO SS2G4.dbo.Users
VALUES('1', 'Jane Doe');
INSERT INTO SS2G4.dbo.Users
VALUES('2', 'Sally Doe');
INSERT INTO SS2G4.dbo.Users
VALUES('3', 'John Doe');
INSERT INTO SS2G4.dbo.Users
VALUES('4', 'Alan');-- zheming
INSERT INTO SS2G4.dbo.Users
VALUES('5', 'John');
INSERT INTO SS2G4.dbo.Users
VALUES('6', 'Sammy');
INSERT INTO SS2G4.dbo.Users
VALUES('7', 'Tan Ah Kao');
INSERT INTO SS2G4.dbo.Users
VALUES('8', 'Nina Gan');
INSERT INTO SS2G4.dbo.Users
VALUES('9', 'Zhe Ming');
INSERT INTO SS2G4.dbo.Users
VALUES('10', 'Jonny');
```

```
INSERT INTO SS2G4.dbo.Users
VALUES('11', 'Cathy');
INSERT INTO SS2G4.dbo.Users
VALUES('12', 'Tan Liang Ming');
INSERT INTO SS2G4.dbo.Users
VALUES('13', 'Christy Chan');
INSERT INTO SS2G4.dbo.Users
VALUES('14', 'Annabelle');
INSERT INTO SS2G4.dbo.Users
VALUES('15', 'Kaitlyn');
INSERT INTO SS2G4.dbo.Users
VALUES('16', 'Chan De Xing');
INSERT INTO SS2G4.dbo.Users
VALUES('17', 'Lao Ji');
INSERT INTO SS2G4.dbo.Users
VALUES('18', 'Lim Ah Kao');
INSERT INTO SS2G4.dbo.Users
VALUES('19', 'Nnan Gan');
INSERT INTO SS2G4.dbo.Users
VALUES('20', 'Tan Ah Beng');


-- Orders (OID, UID, Shipping Addr, Shipping Cost, OrderDateTime)
INSERT INTO SS2G4.dbo.Orders
VALUES('1', '1', 'Jurong Street 21', '5.00', '2020-08-23 15:30:47');
INSERT INTO SS2G4.dbo.Orders
VALUES('2', '1', 'Jurong Street 21', '8.00', '2020-08-23 16:20:52');
INSERT INTO SS2G4.dbo.Orders
VALUES('3', '2', 'Boon Lay Street 21', '12', '2020-06-18 17:50:20');
INSERT INTO SS2G4.dbo.Orders
VALUES('4', '3', 'Yishun Street 11', '7', '2020-09-18 13:12:23');
INSERT INTO SS2G4.dbo.Orders
VALUES('5', '3', 'Yishun Street 11', '5', '2020-06-10 14:25:52');
INSERT INTO SS2G4.dbo.Orders
VALUES('6', '2', 'Boon Lay Street 21', '1', '2020-07-08 08:30:32');
INSERT INTO SS2G4.dbo.Orders
VALUES('7', '4', '628 Veerasamy Road', '2', '2020-09-11 16:20:21');
INSERT INTO SS2G4.dbo.Orders
VALUES('8', '5', '25 Paya Lebar Road', '1', '2020-06-28 09:05:05');
INSERT INTO SS2G4.dbo.Orders
VALUES('9', '6', 'Bishan Street 91', '1', '2020-09-08 11:20:52');

INSERT INTO SS2G4.dbo.Orders
VALUES('10', '7', '810 Woodlands Street 7', '1.5', '2020-06-23 15:30:47');
```

```
INSERT INTO SS2G4.dbo.Orders
VALUES('11', '8', '811 Woodlands Street 8', '2.5', '2020-06-24 16:20:52');

INSERT INTO SS2G4.dbo.Orders
VALUES('12', '9', '812 Woodlands Street 9', '3.5', '2020-07-18 17:50:20');
INSERT INTO SS2G4.dbo.Orders
VALUES('13', '10', '813 Woodlands Street 10', '4', '2020-07-18 13:12:23');
INSERT INTO SS2G4.dbo.Orders
VALUES('14', '11', '814 Woodlands Street 11', '1', '2020-07-23 14:25:52');

INSERT INTO SS2G4.dbo.Orders
VALUES('15', '12', '815 Woodlands Street 12', '2', '2020-08-24 08:30:32');
INSERT INTO SS2G4.dbo.Orders
VALUES('16', '13', '816 Woodlands Street 13', '1', '2020-08-24 16:20:21');
INSERT INTO SS2G4.dbo.Orders
VALUES('17', '14', '817Woodlands Street 14', '1', '2020-08-22 09:05:05');
INSERT INTO SS2G4.dbo.Orders
VALUES('18', '15', '818 Woodlands Street 15818', '1', '2020-08-21 11:20:52');

INSERT INTO SS2G4.dbo.Orders
VALUES('19', '16', '819 Woodlands Street 16', '1', '2020-10-13 11:20:52');

INSERT INTO SS2G4.dbo.Orders
VALUES('20', '17', '820 Woodlands Street 17', '1', '2020-10-15 11:20:52');

INSERT INTO SS2G4.dbo.Orders
VALUES('21', '18', 'Bishan Street 18', '1', '2020-10-11 11:20:52');
INSERT INTO SS2G4.dbo.Orders
VALUES('22', '19', 'Bishan Street 19', '1', '2020-10-12 11:20:52');


INSERT INTO SS2G4.dbo.Orders
VALUES('23', '15', '818 Woodlands Street 15818 ', '1', '2020-08-11 13:20:52');
INSERT INTO SS2G4.dbo.Orders
VALUES('24', '16', '819 Woodlands Street 16', '1', '2020-08-13 14:20:52');
INSERT INTO SS2G4.dbo.Orders
VALUES('25', '17', '820 Woodlands Street 17', '1', '2020-08-15 15:20:52');
INSERT INTO SS2G4.dbo.Orders
VALUES('26', '18', 'Bishan Street 18', '1', '2020-08-11 16:20:52');
INSERT INTO SS2G4.dbo.Orders
VALUES('27', '19', 'Bishan Street 19', '1', '2020-08-13 17:20:52');
```

```
-- Employees (EID, EmployeeName, Salary)
INSERT INTO SS2G4.dbo.Employees
VALUES('1', 'Shawn Lim', '2500');
INSERT INTO SS2G4.dbo.Employees
VALUES('2', 'Alfred Walter', '3500');
INSERT INTO SS2G4.dbo.Employees
VALUES('3', 'Kurt Stracke', '3500');
INSERT INTO SS2G4.dbo.Employees
VALUES('4', 'Darius Tan', '3500');
INSERT INTO SS2G4.dbo.Employees
VALUES('5', 'Sean Goh', '2500');
INSERT INTO SS2G4.dbo.Employees
VALUES('6', 'Joshua', '2500');
INSERT INTO SS2G4.dbo.Employees
VALUES('7', 'Kenina', '4500');
INSERT INTO SS2G4.dbo.Employees
VALUES('8', 'Dominic Tan', '2500');
INSERT INTO SS2G4.dbo.Employees
VALUES('9', 'Sean Lim', '2800');
INSERT INTO SS2G4.dbo.Employees
VALUES('10', 'Lionel Peh', '3900');
INSERT INTO SS2G4.dbo.Employees
VALUES('11', 'Howen Goh', '3200');
INSERT INTO SS2G4.dbo.Employees
VALUES('12', 'Melvin Sim', '3800');


-- Products (PID, PName, Maker, Category)
INSERT INTO SS2G4.dbo.Products
VALUES('IPhone 11 Pro Casing', 'Apple', 'Accessories');
INSERT INTO SS2G4.dbo.Products
VALUES('iPhone Xs', 'Apple', 'Phone');
INSERT INTO SS2G4.dbo.Products
VALUES('iPhone Xs Max', 'Apple', 'Phone');
INSERT INTO SS2G4.dbo.Products
VALUES('iPhone X', 'Apple', 'Phone');
INSERT INTO SS2G4.dbo.Products
VALUES('iPhone 12', 'Apple', 'Phone');

INSERT INTO SS2G4.dbo.Products
VALUES('iPhone 12 Pro', 'Apple', 'Phone');
INSERT INTO SS2G4.dbo.Products
VALUES('iPhone 12 Pro Max', 'Apple', 'Phone');
INSERT INTO SS2G4.dbo.Products
```

```
VALUES('iPhone SE', 'Apple', 'Phone');
INSERT INTO SS2G4.dbo.Products
VALUES('iPhone Mini', 'Apple', 'Phone');
INSERT INTO SS2G4.dbo.Products
VALUES('Phone 11 Pro', 'Apple', 'Phone');

INSERT INTO SS2G4.dbo.Products
VALUES('Samsung Galaxy S20', 'Samsung', 'Phone');
INSERT INTO SS2G4.dbo.Products
VALUES('Samsung Note 7', 'Samsung', 'Phone');
INSERT INTO SS2G4.dbo.Products
VALUES('Samsung Note 8', 'Samsung', 'Phone');
INSERT INTO SS2G4.dbo.Products
VALUES('Samsung Note 9', 'Samsung', 'Phone');
INSERT INTO SS2G4.dbo.Products
VALUES('Samsung Note 10', 'Samsung', 'Phone');

INSERT INTO SS2G4.dbo.Products
VALUES('Samsung Galaxy A9', 'Samsung', 'Phone');
INSERT INTO SS2G4.dbo.Products
VALUES('Samsung Galaxy A10', 'Samsung', 'Phone');
INSERT INTO SS2G4.dbo.Products
VALUES('Samsung Galaxy s10', 'Samsung', 'Phone');
INSERT INTO SS2G4.dbo.Products
VALUES('Xiaomi Handcam', 'Xiaomi', 'Camera');
INSERT INTO SS2G4.dbo.Products
VALUES('Creative Desktop Speaker', 'Creative', 'Speaker');

-- Complaints
INSERT INTO SS2G4.dbo.Complaints
VALUES('1', '2', '2020-09-10 14:30:00', 'Scratches found on product', 'addressed', '2020-09-05
14:23:00');
INSERT INTO SS2G4.dbo.Complaints
VALUES('1', '1', '2020-08-29 14:30:00', 'Product images not found on the website !', 'addressed',
'2020-08-28 15:30:00');
INSERT INTO SS2G4.dbo.Complaints
VALUES('1', '4', '2020-07-10 14:25:00', 'Scratches found on product', 'being handled',
'2020-07-09 14:50:00');
INSERT INTO SS2G4.dbo.Complaints
VALUES('1', '3', '2020-09-26 14:30:00', 'Product images not found on the website !', 'being
handled', '2020-09-24 15:30:00');
INSERT INTO SS2G4.dbo.Complaints
VALUES('4', '2', '2020-08-28 14:30:00', 'Sound not clear', 'addressed', '2020-08-24 14:30:00');
INSERT INTO SS2G4.dbo.Complaints
```

```
VALUES('4', '1', '2020-10-04 14:30:00', 'Product images not found on the website !', 'being handled', '2020-10-01 15:30:00');
INSERT INTO SS2G4.dbo.Complaints
VALUES('2', '2', '2020-06-29 14:30:00', 'Screen cracked!!', 'addressed', '2020-06-28 13:30:00');
INSERT INTO SS2G4.dbo.Complaints
VALUES('6', '3', '2020-09-25 14:40:00', 'Product is faulty!', 'addressed', '2020-09-24 12:30:00');
INSERT INTO SS2G4.dbo.Complaints
VALUES('5', null, null, 'Scratched Case', 'pending', '2020-07-07 19:30:00');

-- ComplaintsOnShops (ComplaintID, ShopName)
INSERT INTO SS2G4.dbo.ComplaintsOnShops
VALUES('2', 'Shop 4');
INSERT INTO SS2G4.dbo.ComplaintsOnShops
VALUES('4', 'Shop 4');
INSERT INTO SS2G4.dbo.ComplaintsOnShops
VALUES('6', 'Shop 6');

-- ComplaintsOnOrders (ComplaintID, OrderId)
INSERT INTO SS2G4.dbo.ComplaintsOnOrders
VALUES('1', '1');
INSERT INTO SS2G4.dbo.ComplaintsOnOrders
VALUES('3', '2');
INSERT INTO SS2G4.dbo.ComplaintsOnOrders
VALUES('5', '7');
INSERT INTO SS2G4.dbo.ComplaintsOnOrders
VALUES('7', '3');
INSERT INTO SS2G4.dbo.ComplaintsOnOrders
VALUES('8', '9');
INSERT INTO SS2G4.dbo.ComplaintsOnOrders
VALUES('9', '8');

-- ProductsInOrder
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('1', '1', 'Shop 1', 'delivered', '2020-08-25 15:30:47', '50.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('7', '2', 'Shop 4', 'delivered', '2020-08-26 16:20:52', '2000.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('8', '3', 'Shop 5', 'delivered', '2020-08-21 16:18:32', '1410.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('9', '4', 'Shop 5', 'delivered', '2020-09-22 17:20:42', '1880.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('10', '5', 'Shop 10', 'delivered', '2020-06-15 18:20:42', '1950.00', '1');

INSERT INTO SS2G4.dbo.ProductsInOrder
```

```
VALUES('11', '6', 'Shop 5', 'delivered', '2020-07-11 17:20:36', '1580.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('12', '7', 'Shop 6', 'delivered', '2020-09-24 16:18:52', '1650.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('13', '8', 'Shop 6', 'delivered', '2020-07-01 17:20:52', '1750.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('14', '9', 'Shop 8', 'delivered', '2020-09-15 12:28:02', '1800.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('2', '10', 'Shop 1', 'delivered', '2020-06-28 16:05:22', '1500.00', '1');

INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('2', '11', 'Shop 1', 'delivered', '2020-06-28 15:10:22', '1500.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('2', '12', 'Shop 1', 'delivered', '2020-07-25 13:29:41', '1500.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('2', '13', 'Shop 2', 'delivered', '2020-07-22 13:10:42', '1550.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('2', '14', 'Shop 3', 'delivered', '2020-07-24 15:21:27', '1508.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('2', '15', 'Shop 1', 'delivered', '2020-08-28 15:03:05', '1500.00', '1');

INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('2', '16', 'Shop 2', 'delivered', '2020-08-29 15:25:30', '1550.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('2', '17', 'Shop 7', 'delivered', '2020-08-28 14:20:00', '1550.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('2', '18', 'Shop 7', 'delivered', '2020-08-29 16:20:00', '1550.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('3', '19', 'Shop 1', 'shipped', null, '1400.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('4', '20', 'Shop 1', 'being processed', null, '1300.00', '1');


INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('5', '21', 'Shop 2', 'delivered', '2020-07-25 16:20:52', '1750.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('6', '22', 'Shop 3', 'delivered', '2020-07-22 16:30:00', '1908.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('7', '23', 'Shop 4', 'delivered', '2020-08-19 15:30:47', '2000.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('7', '24', 'Shop 4', 'delivered', '2020-08-22 16:20:52', '2000.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('7', '25', 'Shop 4', 'delivered', '2020-08-20 16:18:32', '2000.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
```

```
VALUES('7', '26', 'Shop 4', 'delivered', '2020-08-20 17:20:42', '2000.00', '1');
INSERT INTO SS2G4.dbo.ProductsInOrder
VALUES('7', '27', 'Shop 4', 'delivered', '2020-08-22 18:20:42', '2000.00', '1');



-- ProductsInShops (ShopName,PID, Price,Quantity)
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 1', '1', '50', '500');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 1', '2', '1500', '150');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 1', '3', '1400', '500');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 1', '4', '1300', '150');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 1', '5', '1700', '500');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 1', '6', '1900', '150');

INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 2', '1', '55', '500');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 2', '2', '1550', '150');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 2', '3', '1450', '500');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 2', '4', '1350', '150');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 2', '5', '1750', '500');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 2', '6', '1950', '150');

INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 3', '1', '58', '500');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 3', '2', '1508', '150');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 3', '3', '1408', '500');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 3', '4', '1308', '150');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 3', '5', '1708', '500');
```

```
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 3', '6', '1908', '150');

INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 4', '7', '2000', '500');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 4', '8', '1400', '150');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 4', '9', '1850', '500');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 4', '10', '1950', '150');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 4', '11', '1500', '500');

INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 5', '7', '2010', '150');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 5', '8', '1410', '500');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 5', '9', '1880', '150');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 5', '10', '1980', '500');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 5', '11', '1580', '150');


INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 6', '12', '1650', '500');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 6', '13', '1750', '150');


INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 7', '1', '55', '500');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 7', '2', '1550', '150');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 7', '13', '1750', '500');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 7', '14', '1850', '150');


INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 8', '14', '1800', '500');
```

```
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 8', '15', '1900', '150');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 8', '16', '1420', '500');


INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 9', '17', '1320', '500');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 9', '18', '1650', '150');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 9', '19', '300', '500');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 9', '20', '250', '150');
-- xiaomi handcam


INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 10', '10', '1950', '500');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 10', '12', '1600', '150');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 10', '15', '1900', '500');
INSERT INTO SS2G4.dbo.ProductsInShops
VALUES('Shop 10', '16', '50', '150');



-- Feedback (UID, PID, FeedbackDateTime, Rating,Comment)
INSERT INTO SS2G4.dbo.Feedback
VALUES('1', '1', '2020-08-26 17:50:30', '5', 'Delivery is on time');
INSERT INTO SS2G4.dbo.Feedback
VALUES('1', '7', '2020-08-27 13:50:20', '5', 'Great Product');
INSERT INTO SS2G4.dbo.Feedback
VALUES('2', '8', '2020-08-22 17:12:25', '5', 'Delivery is on time');
INSERT INTO SS2G4.dbo.Feedback
VALUES('3', '9', '2020-09-23 14:23:20', '5', 'Friendly seller');
INSERT INTO SS2G4.dbo.Feedback
VALUES('3', '10', '2020-06-16 12:30:40', '5', 'Delivery is on time');
INSERT INTO SS2G4.dbo.Feedback
VALUES('2', '11', '2020-07-12 16:40:30', '5', 'Great Product');

INSERT INTO SS2G4.dbo.Feedback
VALUES('4', '12', '2020-09-25 17:13:22', '5', 'Delivery is on time');
```

```
INSERT INTO SS2G4.dbo.Feedback
VALUES('5', '13', '2020-07-02 18:50:50', '5', 'Friendly Seller');
INSERT INTO SS2G4.dbo.Feedback
VALUES('6', '14', '2020-09-16 12:50:10', '5', 'Delivery is on time');
INSERT INTO SS2G4.dbo.Feedback
VALUES('7', '2', '2020-08-02 18:55:24', '5', 'Friendly seller');
INSERT INTO SS2G4.dbo.Feedback
VALUES('8', '2', '2020-08-05 13:50:21', '5', 'Delivery is on time');

INSERT INTO SS2G4.dbo.Feedback
VALUES('9', '2', '2020-08-01 15:40:40', '5', 'Fast delivery');
INSERT INTO SS2G4.dbo.Feedback
VALUES('10', '2', '2020-08-02 16:52:20', '5', 'Seller is friendly');
INSERT INTO SS2G4.dbo.Feedback
VALUES('11', '2', '2020-08-01 17:31:52', '5', 'Delivery is on time');
INSERT INTO SS2G4.dbo.Feedback
VALUES('12', '2', '2020-08-29 18:21:25', '3', 'Seller is friendly');
INSERT INTO SS2G4.dbo.Feedback
VALUES('13', '2', '2020-08-30 19:10:20', '3', 'Seller is rude');

INSERT INTO SS2G4.dbo.Feedback
VALUES('14', '2', '2020-08-29 12:54:24', '5', 'Delivery is on time');
INSERT INTO SS2G4.dbo.Feedback
VALUES('15', '2', '2020-08-30 13:44:53', '5', 'Delivery is on time');

INSERT INTO SS2G4.dbo.Feedback
VALUES('18', '5', '2020-07-26 16:42:24', '3', 'Delivery is slightly late');
INSERT INTO SS2G4.dbo.Feedback
VALUES('19', '6', '2020-07-23 18:25:20', '3', 'Delivery is slightly late');

INSERT INTO SS2G4.dbo.Feedback
VALUES('15', '7', '2020-08-25 17:13:22', '5', 'Delivery is on time');
INSERT INTO SS2G4.dbo.Feedback
VALUES('16', '7', '2020-08-23 18:50:50', '5', 'Friendly Seller');
INSERT INTO SS2G4.dbo.Feedback
VALUES('17', '7', '2020-08-26 12:50:10', '5', 'Delivery is on time');
INSERT INTO SS2G4.dbo.Feedback
VALUES('18', '7', '2020-08-27 18:55:24', '5', 'Friendly seller');
INSERT INTO SS2G4.dbo.Feedback
VALUES('19', '7', '2020-08-28 13:50:21', '5', 'Delivery is on time');
```

Populating Tables with randomised values

Populating values for *Products*

```
-- SQL Fill Product TABLE

Declare @Pname int
Declare @foo int
Declare @MakerName as varchar(10)
Declare @CatName as varchar(10)

Set @Pname  = 11
Declare @MakerList table (name varchar(15))

SELECT * FROM Products;

While @Pname <100
Begin
	Select @foo = Round((4*Rand()+1),0);
	Select @MakerName = CHOOSE(@foo, 'Apple', 'XiaoMi', 'Samsung', 'Amazon',
'UGreen');
	Select @CatName = CHOOSE(@foo, 'Phone', 'Speaker', 'Phone', 'Accessories',
'Accessories');

	INSERT INTO Products VALUES('Product Name ' + CAST(@Pname as nvarchar(10)),
                                                    @MakerName,
                                                    @CatName);
	Set @Pname = @Pname + 1
End
SELECT * FROM Products;
```

Populating values for *ProductsInShops*

```
-- SQL Fill ProductsInShops
Declare @Sname varchar(50)
Declare @ProductID int
Declare @Price int
Declare @Quantity int

-- Insert our own insert queries here
Set @ProductID = 21

While @ProductID <= 99
Begin
```

```
        set @Sname = (SELECT TOP 1 ShopName FROM Shops
        ORDER BY NEWID());
        set @Price = Round((5*Rand()+1),1) * 100;
        set @Quantity = Round((5*Rand()+1),1) * 100;

        INSERT INTO ProductsInShops VALUES(@Sname,

@ProductID,

@Price,

@Quantity);
        set @ProductID = @ProductID + 1

End
SELECT * FROM ProductsInShops;
```

Populating values for *Orders*

```
-- Fill Orders table
Declare @OID int
Declare @UIDlist table (userid int)
Declare @UID int
Declare @AddressList table (name varchar(50))
Declare @Address varchar(50)
Declare @ShippingCost int
Declare @OrderDateTimeList table (orderdatetime datetime)
Declare @OrderDateTime datetime

Insert into @UIDlist
SELECT DISTINCT Userid FROM Users;
Insert into @AddressList
SELECT DISTINCT ShippingAddress FROM Orders;
Insert Into @OrderDateTimeList
SELECT DISTINCT OrderDateTime FROM ORders;

-- ADD Own queries from docs
-- start from 23
set @OID = 23
WHILE @OID <= 50
Begin
        set @UID = (SELECT TOP 1 userid FROM @UIDlist
                ORDER BY NEWID());
```

```
        set @Address = (SELECT TOP 1 name FROM @AddressList
                ORDER BY NEWID());
        set @OrderDateTime = (SELECT TOP 1 orderdatetime FROM @OrderDateTimeList
                ORDER BY NEWID());
        set @ShippingCost = Round((9*Rand()+1),1) * 10;
        set @OID = @OID + 1
        INSERT INTO SS2G4.dbo.Orders VALUES(@OID,
                                                                @UID,
                                                                @Address,
                                                                @ShippingCost,
                                                                @OrderDateTime);

END

GO
SELECT * FROM Orders;
```

Populating values for *ProductsInOrders*

```
-- Fill ProductsInOrders
SELECT * FROM ProductsInOrder;

--Fill Productsinorders
declare @PIDList table (pid int)
declare @PID int
declare @OrderId int
declare @ShopNameList table(name varchar(10))
declare @ShopName varchar(50)
declare @OrderStatus varchar(50)
declare @DeliveryDateList table(deliverydate datetime)
declare @DeliveryDate datetime
declare @PriceList table(price int)
declare @Price int
declare @Quantity int
declare @foo int

Insert into @PIDList
SELECT DISTINCT PID FROM Products;
Insert into @ShopNameList
SELECT DISTINCT ShopName From Shops;
Insert into @DeliveryDateList
SELECT DISTINCT DeliveryDate From ProductsInOrder;

set @OrderId = 31
```

```
WHILE @OrderId <= 151
BEGIN
		SELECT TOP 1 @Price = Price, @ShopName = ShopName, @PID = PID
		FROM ProductsInShops ORDER BY NEWID();

		Select @foo = Round((2*Rand()+1),0);
		Select @OrderStatus = CHOOSE(@foo, 'being processed', 'shipped',
'delivered');

		set @DeliveryDate= (SELECT TOP 1 deliverydate FROM @DeliveryDateList
		ORDER BY NEWID());

		set @Quantity = Round((5*Rand()+1),0);

	INSERT INTO ProductsInOrder VALUES(@PID,
							@OrderId,
							@ShopName,
							@OrderStatus,
							@DeliveryDate,
							@Price,
							@Quantity);

set @OrderId = @OrderId + 1
END

SELECT * FROM ProductsInOrder;
```

-- The following two queries ensure that the data in the table

-- are valid

Remove Delivery Dates if Product not delivered

```
declare @deliverystatus varchar(50)

declare @oid int
set @oid = 1

WHILE @OID <= 151
BEGIN
	SELECT @deliverystatus = OrderStatus FROM
	ProductsInOrder
```

```
	WHERE ProductsInOrder.OrderId = @oid;

	IF @deliverystatus = 'shipped' OR @deliverystatus = 'being processed'
	BEGIN
		UPDATE ProductsInOrder
		SET DeliveryDate = NULL
		WHERE OrderId = @oid
	END
set @oid = @oid + 1
END
	SELECT * FROM
	ProductsInOrder JOIN Orders ON Orders.OrderId = ProductsInOrder.OrderId
```

Swap Delivery Dates and Order Dates if required

```
declare @deliverydatetime datetime
declare @ordereddatetime datetime
declare @oid int
set @oid = 1

WHILE @OID <= 151
BEGIN
	SELECT @deliverydatetime = DeliveryDate, @ordereddatetime =  OrderDateTime
FROM
	ProductsInOrder JOIN Orders ON Orders.OrderId = ProductsInOrder.OrderId
	WHERE Orders.OrderId = @oid;

	IF @deliverydatetime < @ordereddatetime
	BEGIN
		UPDATE Orders SET OrderDateTime = @deliverydatetime WHERE
OrderId=@oid
		UPDATE ProductsInOrder SET DeliveryDate = @ordereddatetime WHERE
OrderId=@oid;
	END
set @oid = @oid + 1
END
	SELECT * FROM
	ProductsInOrder JOIN Orders ON Orders.OrderId = ProductsInOrder.OrderId
```

Populating values for Ratings (For Query 2)

```
-- FILL Feedback
declare @UID int
declare @PID int
```

```
declare @FeedBackTimeList table (date datetime)
declare @FeedBackTime datetime
declare @rating int
declare @commentlist table (comment varchar(500))
declare @comment varchar(500)
declare @feedbacknum int
declare @foo int
declare @bar int
declare @baz int
declare @x int
declare @y int
declare @datetime2 datetime
declare @datetime3 datetime
-- 26 queries above
Insert into @FeedBackTimeList
SELECT DISTINCT FeedbackDateTime FROM Feedback;
Insert into @commentlist
SELECT DISTINCT Comment FROM Feedback;
set @feedbacknum = 250
SET @FeedBackTime = '2020-08-31 00:00:00.000'
WHILE @feedbacknum <= 300
BEGIN
	set @UID = (SELECT Userid FROM Orders Where OrderId=@feedbacknum)
	set @PID = 2
	Select @foo = Round((11*Rand()+1),0);

	set @FeedBackTime = @FeedBackTime + 0.0001
	Select @bar = Round((1*Rand()+1),0);
	Select @rating = 5;
	set @comment = 'Delivery was great!'
	INSERT INTO SS2G4.dbo.Feedback VALUES(@UID,
                                              @PID,
                                              @FeedBackTime,
                                              @rating,
                                              @comment);


	set @feedbacknum = @feedbacknum + 1
END
```

Populating values for *Feedback*

```
-- FILL Feedback

declare @UID int
declare @PID int
declare @FeedBackTimeList table (date datetime)
declare @FeedBackTime datetime
declare @rating int
declare @commentlist table (comment varchar(500))
declare @comment varchar(500)
declare @feedbacknum int
declare @foo int
declare @bar int
declare @baz int
declare @x int
declare @y int
-- 26 queries above

Insert into @FeedBackTimeList
SELECT DISTINCT FeedbackDateTime FROM Feedback;

Insert into @commentlist
SELECT DISTINCT Comment FROM Feedback;

set @feedbacknum = 26

WHILE @feedbacknum < 145
BEGIN
	set @UID = (SELECT TOP 1 UserId FROM Users ORDER BY NEWID())
	set @PID = (SELECT TOP 1 PID FROM Products ORDER BY NEWID())
	Select @foo = Round((11*Rand()+1),0);
	set @FeedBackTime = CHOOSE (@foo,  '2020-08-05 05:30:47',
                                                '2020-08-12 15:20:47',
                                                '2020-08-13 13:10:47',
                                                '2020-08-14 16:40:47',
                                                '2020-08-16 12:20:47',
                                                '2020-08-17 17:30:47',
                                                '2020-08-18 24:10:47',
                                                '2020-08-21 21:50:47',
                                                '2020-08-23 13:40:47',
                                                '2020-08-24 04:30:47',
                                                '2020-08-25 07:20:47',
                                                '2020-08-27 06:10:47')
```

```
        Select @bar = Round((4*Rand()+1),0);
        Select @rating = CHOOSE(@bar, '1','2','3','4','5');
        IF @rating = 1 OR @rating = 2
                set @comment = 'Delivery was bad'
        IF @rating = 3
                set @comment = 'Delivery was okay'
        IF @rating = 4 OR @rating = 5
                set @comment = 'Delivery was great!'

        INSERT INTO SS2G4.dbo.Feedback VALUES(@UID,
                                                @PID,
                                                @FeedBackTime,
                                                @rating,
                                                @comment);

END
```

Queries

```
-- Question 1
SELECT AVG(Price) AS AveragePrice
FROM ProductsInOrder, Orders, Products
WHERE ProductsInOrder.OrderId = Orders.OrderId
   AND ProductsInOrder.PID = Products.PID
   AND month(OrderDateTime) = '8'
   AND year(OrderDateTime) = '2020'
GROUP BY Products.PID
HAVING Products.PID = (SELECT PID
FROM Products
WHERE Products.ProductName = 'iPhone Xs' );


-- Question 2
SELECT ProductName, ROUND(AVG(CAST(Rating AS FLOAT)), 2) AS AvgRating
FROM Feedback
   INNER JOIN Products
   ON Feedback.PID = Products.PID
   WHERE ProductName IN (SELECT ProductName
   FROM Feedback
      INNER JOIN Products
      ON Feedback.PID = Products.PID
         AND [FeedbackDateTime] >= '2020-08-01'
         AND [FeedbackDateTime] < '2020-09-01'
   GROUP BY ProductName, Rating
   HAVING Rating = 5 AND COUNT(*) >= 100)
GROUP BY ProductName

-- Question 3
SELECT AVG(DATEDIFF(hour,OrderDateTime,DeliveryDate)) AS AverageInHours
FROM ProductsInOrder, Orders
WHERE month(OrderDateTime)='6'
   AND year(OrderDateTime) ='2020'
   AND Orders.OrderId = ProductsInOrder.OrderId
   AND ProductsInOrder.OrderStatus = 'Delivered';

-- Question 4
SELECT TOP 1
   EmployeeName, AVG(DATEDIFF(HOUR,FilledDateTime,HandledDateTime)) as
SmallestLatencyHours
FROM Complaints, Employees
WHERE Employees.EID = Complaints.EID
GROUP BY Employees.EmployeeName
```

```
ORDER BY SmallestLatencyHours ASC

-- Question 5
Select DISTINCT ProductName
From Products as t1
Where Maker = 'Samsung';

SELECT ProductName, Count (Distinct t2.ShopName) as NumberofShops
FROM Products as t1

   /* Left join on common attribute ProductID of both tables */
   LEFT JOIN ProductsInShops AS t2
   ON t1.PID = t2.PID
WHERE Maker = 'Samsung'
GROUP BY ProductName;

-- Question 6
SELECT TOP 1
   t2.ShopName, SUM(t2.Price*t2.Quantity) AS revenue
FROM Orders as t1

   /* Left join on common attribute OrderID of both tables */
   LEFT JOIN ProductsInOrder AS t2
   ON t1.OrderID = t2.OrderID

/* OrderDateTime should fall under 2020/08 */
WHERE MONTH(t1.OrderDateTime) = 8 AND YEAR(t1.OrderDateTime) = 2020
/* Group by Shop name with aggregate function SUM of all revenue(OrderPrice*OrderQuantity)
by this shop */
GROUP BY ShopName
ORDER BY revenue DESC


-- Question 7
Select TOP 1
   ProductName
From Orders
   Inner Join ProductsInOrder
   On ProductsInOrder.OrderId = Orders.OrderId
   Inner Join Products
   ON Products.PID = ProductsInOrder.PID
Where UserId = (SELECT TOP 1
   UserID
FROM Complaints
```

```
Group by UserID
Order by COUNT(*) DESC)
Order by Price DESC

-- Quesiont 8
Select TOP 5
   ProductName
From Orders
   Inner Join ProductsInOrder
   On ProductsInOrder.OrderId = Orders.OrderId
   Inner Join Products
   ON Products.PID = ProductsInOrder.PID
WHERE MONTH(OrderDateTime) = 8 AND YEAR(OrderDateTime) = 2020
Group by ProductName
Having COUNT(DISTINCT UserId) < (Select Count(*)
From Users)
Order by COUNT(*) DESC

-- Question 9
With
   A1
   as
   (
      SELECT Products.PID, (DATEDIFF(MONTH,'1900-01-01 00:00:00.000',OrderDateTime))
AS Months, Count(*) as ProductsSoldinMonth
      From Orders
         Inner Join ProductsInOrder
         On ProductsInOrder.OrderId = Orders.OrderId
         Inner Join Products
         ON Products.PID = ProductsInOrder.PID
      Group by (DATEDIFF(MONTH,'1900-01-01 00:00:00.000',OrderDateTime)), Products.PID
   )

Select ProductName
From Products
Where PID in (Select t1.pid
From A1 as t0
   Join A1 as t1
   On t0.pid = t1.pid
   Join A1 as t2
   On t1.pid = t2.pid
Where (t1.months - t0.months = 1 AND t2.months - t1.months = 1)
   AND (t0.ProductsSoldinMonth < t1.ProductsSoldinMonth)
   AND (t1.ProductsSoldinMonth < t2.ProductsSoldInMonth))
```

Drop Tables SQL

```
DROP TABLE dbo.Feedback;
DROP TABLE dbo.ProductsInOrder;
DROP TABLE dbo.ProductsInShops;
DROP TABLE dbo.ComplaintsOnOrders;
DROP TABLE dbo.ComplaintsOnShops;
DROP TABLE dbo.Complaints;
DROP TABLE dbo.Products;
DROP TABLE dbo.Employees;
DROP TABLE dbo.Orders;
DROP TABLE dbo.Users;
DROP TABLE dbo.Shops;
DROP TRIGGER trigger_complaints;
```