Assignment

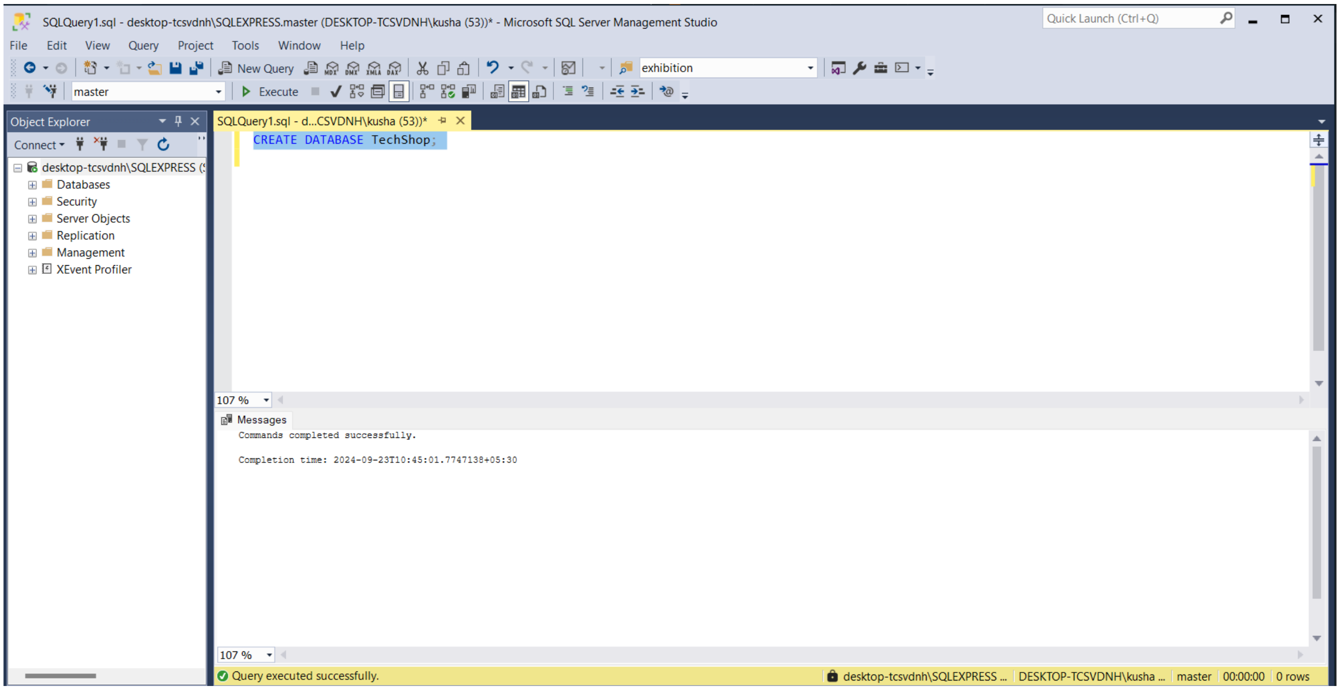
Tech Shop

Submitted by : Sameer Pal

Task:1. Database Design:

1. **Create the database named "TechShop"**

```
Create Database TechShop;
USE TechShop;
```



SQLQuery1.sql - desktop-tcsvdnh\SQLEXPRESS.master (DESKTOP-TCSVDNH\kusha (53))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
exhibition
master
Execute
Object Explorer
Connect
desktop-tcsvdnh\SQLEXPRESS
Databases
Security
Server Objects
Replication
Management
XEvent Profiler
SQLQuery1.sql - d...CSVDNH\kusha (53))*
107 %
Messages

```
CREATE DATABASE TechShop;
```

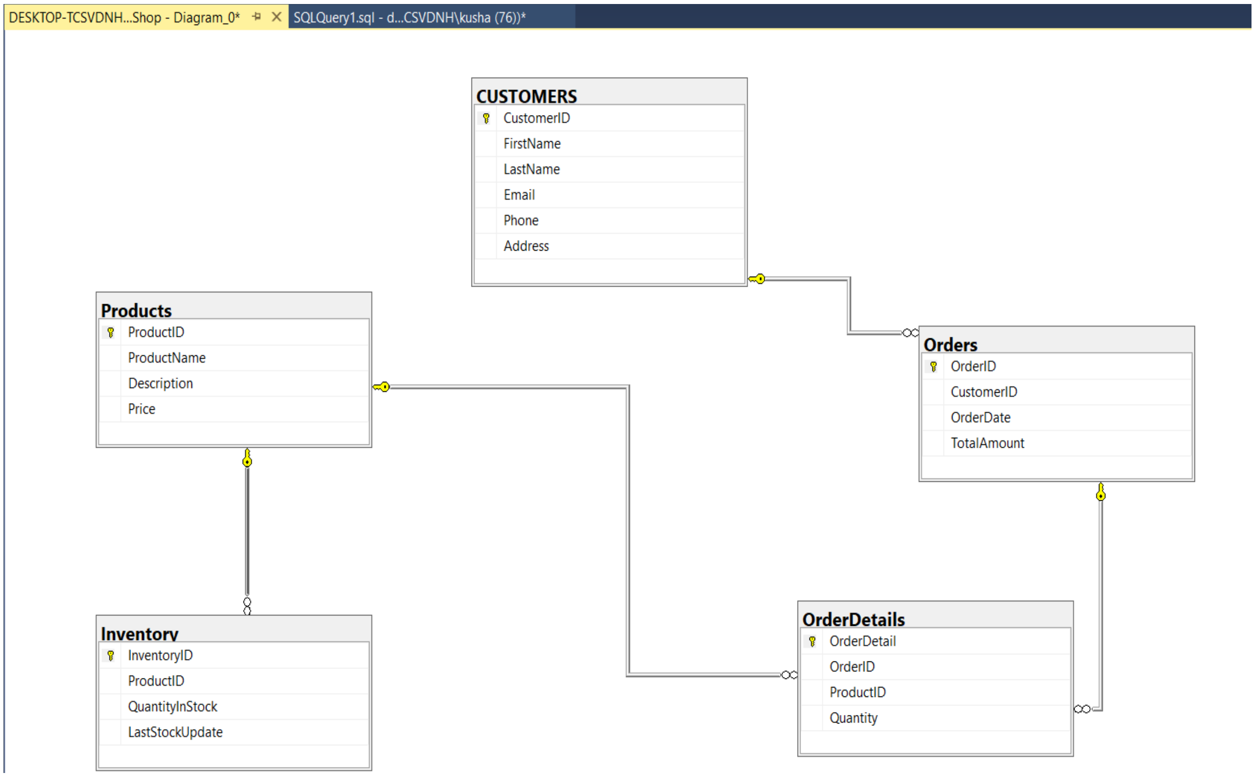
```
Commands completed successfully.

Completion time: 2024-09-23T10:45:01.7747138+05:30
```

Query executed successfully. | desktop-tcsvdnh\SQLEXPRESS ... | DESKTOP-TCSVDNH\kusha ... | master | 00:00:00 | 0 rows

2. Define the schema for the Customers, Products, Orders, OrderDetails and Inventory tables based on the provided schema.

```
/*1. Customers:
• CustomerID (Primary Key)
• FirstName
• LastName
• Email
• Phone
• Address*/


CREATE TABLE CUSTOMERS(
CustomerID int Identity Primary Key,
FirstName varchar(30),
LastName varchar(30),
Email varchar(30),
Phone varchar(30),
Address varchar(40)
);


/*. Products:
• ProductID (Primary Key)
• ProductName
• Description
• Price
*/
create table Products(
ProductID int Identity Primary Key,
ProductName varchar(30),
Description varchar(80),
Price int
);


/*
Orders:
• OrderID (Primary Key)
• CustomerID (Foreign Key referencing Customers)
• OrderDate
• TotalAmount*/

create table Orders(
OrderID int Identity Primary Key,
CustomerID int Foreign key references Customers(CustomerID),
OrderDate date,
TotalAmount int
);


/*OrderDetails:
• OrderDetailID (Primary Key)
• OrderID (Foreign Key referencing Orders)
• ProductID (Foreign Key referencing Products)
• Quantity*/

create table OrderDetails(
OrderDetail int Identity Primary key,
OrderID int foreign key references Orders(OrderID),
ProductID int foreign key references Products(ProductID),
Quantity int
);


/*Inventory
• InventoryID (Primary Key)
• ProductID (Foreign Key referencing Products)
• QuantityInStock
• LastStockUpdate*/

create table Inventory(
InventoryID int Identity Primary key,
ProductID int foreign key references Products(ProductID),
QuantityInStock int,
LastStockUpdate date
);
```

2. Create an ERD (Entity Relationship Diagram) for the database.


DESKTOP-TCSVDNH...Shop - Diagram_0*
SQLQuery1.sql - d...CSVDNH\kusha (76))*

CUSTOMERS
CustomerID
FirstName
LastName
Email
Phone
Address

Products
ProductID
ProductName
Description
Price

Orders
OrderID
CustomerID
OrderDate
TotalAmount

Inventory
InventoryID
ProductID
QuantityInStock
LastStockUpdate

OrderDetails
OrderDetail
OrderID
ProductID
Quantity

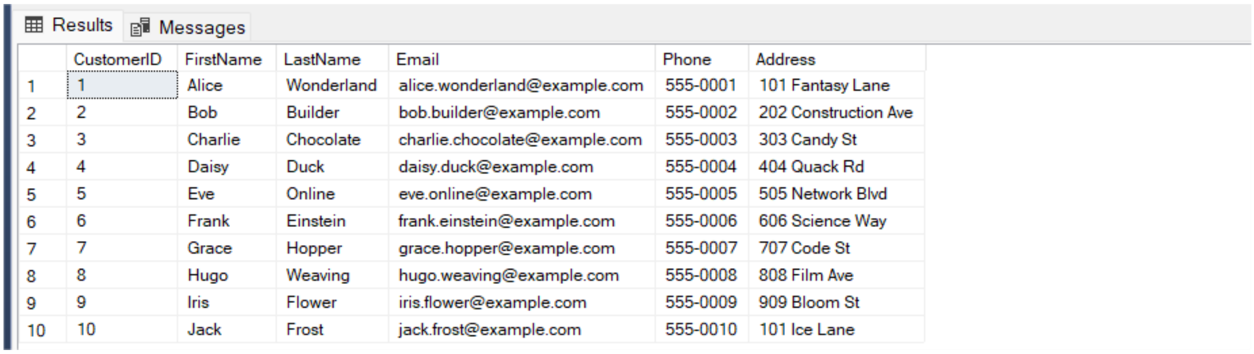
4. Create appropriate Primary Key and Foreign Key constraints for referential integrity.

Primary and Foreign Keys are Being Inserted While The Tables are being Created.

5. Insert at least 10 sample records into each of the following tables.

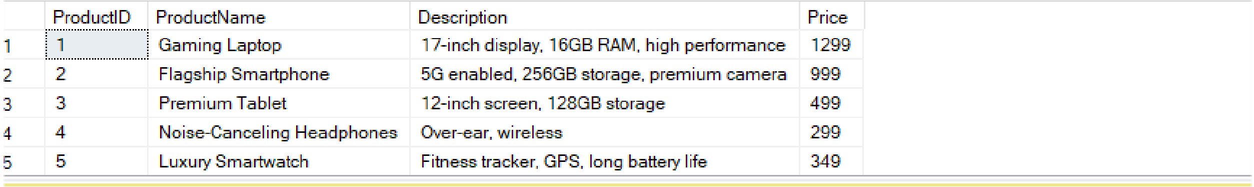
- **Customers**
- **Products**
- **Orders**
- **OrderDetail**
- **Inventory**

```sql
INSERT INTO Customers (FirstName, LastName, Email, Phone, Address) VALUES
('John', 'Doe', 'john.doe@example.com', '555-1234', '123 Elm St'),
('Jane', 'Smith', 'jane.smith@example.com', '555-5678', '456 Oak St'),
('Alice', 'Johnson', 'alice.j@example.com', '555-8765', '789 Pine St'),
('Bob', 'Brown', 'bob.brown@example.com', '555-4321', '101 Maple St'),
('Charlie', 'Davis', 'charlie.davis@example.com', '555-9876', '102 Cedar St'),
('David', 'Wilson', 'david.wilson@example.com', '555-1122', '222 Birch St'),
('Eva', 'Taylor', 'eva.taylor@example.com', '555-3344', '333 Walnut St'),
('Frank', 'Thomas', 'frank.thomas@example.com', '555-5566', '444 Chestnut St'),
('Grace', 'Moore', 'grace.moore@example.com', '555-7788', '555 Spruce St'),
('Harry', 'Anderson', 'harry.anderson@example.com', '555-9911', '666 Aspen St'),
('Ivy', 'Clark', 'ivy.clark@example.com', '555-2233', '777 Willow St'),
('Jack', 'Martin', 'jack.martin@example.com', '555-4455', '888 Redwood St'),
('Kelly', 'Lee', 'kelly.lee@example.com', '555-6677', '999 Cedar St'),
('Liam', 'Harris', 'liam.harris@example.com', '555-8899', '123 Maplewood St'),
('Mia', 'Walker', 'mia.walker@example.com', '555-1212', '456 Oakwood St'),
('Noah', 'Young', 'noah.young@example.com', '555-3434', '789 Pinewood St'),
('Olivia', 'King', 'olivia.king@example.com', '555-5656', '101 Birchwood St'),
('Paul', 'Scott', 'paul.scott@example.com', '555-7878', '202 Elmwood St'),
('Quinn', 'Adams', 'quinn.adams@example.com', '555-9090', '303 Cedarwood St'),
('Rachel', 'Baker', 'rachel.baker@example.com', '555-0101', '404 Maple St');
```



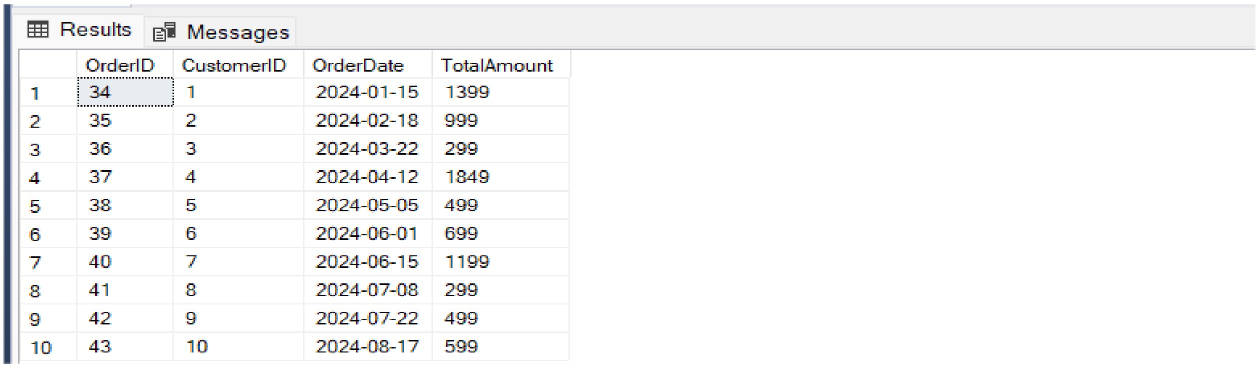
	CustomerID	FirstName	LastName	Email	Phone	Address
1	1	Alice	Wonderland	alice.wonderland@example.com	555-0001	101 Fantasy Lane
2	2	Bob	Builder	bob.builder@example.com	555-0002	202 Construction Ave
3	3	Charlie	Chocolate	charlie.chocolate@example.com	555-0003	303 Candy St
4	4	Daisy	Duck	daisy.duck@example.com	555-0004	404 Quack Rd
5	5	Eve	Online	eve.online@example.com	555-0005	505 Network Blvd
6	6	Frank	Einstein	frank.einstein@example.com	555-0006	606 Science Way
7	7	Grace	Hopper	grace.hopper@example.com	555-0007	707 Code St
8	8	Hugo	Weaving	hugo.weaving@example.com	555-0008	808 Film Ave
9	9	Iris	Flower	iris.flower@example.com	555-0009	909 Bloom St
10	10	Jack	Frost	jack.frost@example.com	555-0010	101 Ice Lane

```sql
INSERT INTO Products (ProductName, Description, Price) VALUES
('Laptop', '15-inch display, 8GB RAM', 799.99),
('Smartphone', '5G enabled, 64GB storage', 599.99),
('Tablet', '10-inch screen, 32GB storage', 299.99),
('Headphones', 'Noise-canceling, over-ear', 199.99),
('Smartwatch', 'Fitness tracker, heart-rate monitor', 149.99),
('Gaming Console', 'Next-gen gaming console', 499.99),
('Wireless Mouse', 'Bluetooth mouse, ergonomic design', 29.99),
('Mechanical Keyboard', 'RGB backlit, Cherry MX switches', 99.99),
('Monitor', '27-inch 4K display', 399.99),
('External SSD', '1TB, USB-C, Portable', 149.99),
('Portable Charger', '10,000mAh, fast charging', 49.99),
('Smart Speaker', 'Voice-controlled, built-in assistant', 129.99),
('Bluetooth Earbuds', 'Wireless, noise-canceling', 89.99),
('VR Headset', 'Virtual reality, high resolution', 599.99),
('Smart Home Hub', 'Control smart devices, voice assistant', 99.99),
('Drone', '4K camera, GPS navigation', 799.99),
('Fitness Band', 'Activity tracker, heart-rate monitor', 59.99),
('Camera', 'DSLR, 24MP, 18-55mm lens', 899.99),
('Smart Thermostat', 'WiFi-enabled, energy saving', 199.99),
('E-reader', '6-inch display, 8GB storage', 129.99);
```



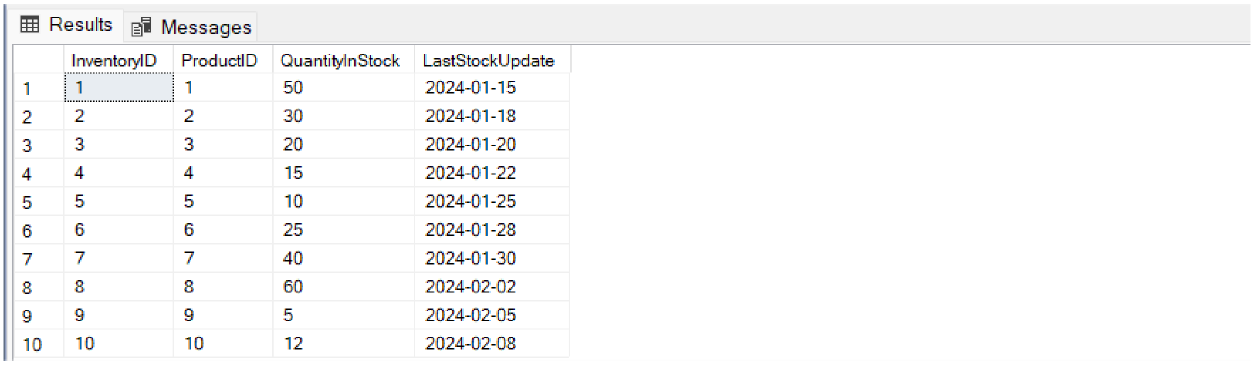
	ProductID	ProductName	Description	Price
1	1	Gaming Laptop	17-inch display, 16GB RAM, high performance	1299
2	2	Flagship Smartphone	5G enabled, 256GB storage, premium camera	999
3	3	Premium Tablet	12-inch screen, 128GB storage	499
4	4	Noise-Canceling Headphones	Over-ear, wireless	299
5	5	Luxury Smartwatch	Fitness tracker, GPS, long battery life	349

```
INSERT INTO Orders (CustomerID, OrderDate, TotalAmount) VALUES
(1, '2023-01-15', 1099.98),
(2, '2023-02-18', 749.99),
(3, '2023-03-22', 199.99),
(4, '2023-04-12', 1249.98),
(5, '2023-05-05', 299.99),
(6, '2023-06-01', 449.99),
(7, '2023-06-15', 599.98),
(8, '2023-07-08', 199.99),
(9, '2023-07-22', 349.99),
(10, '2023-08-17', 399.99),
(11, '2023-08-25', 749.98),
(12, '2023-09-02', 599.98),
(13, '2023-09-12', 499.99),
(14, '2023-09-20', 899.99),
(15, '2023-09-28', 1099.98),
(16, '2023-10-05', 299.99),
(17, '2023-10-12', 399.99),
(18, '2023-10-22', 999.99),
(19, '2023-11-01', 1299.98),
(20, '2023-11-10', 799.99);
```



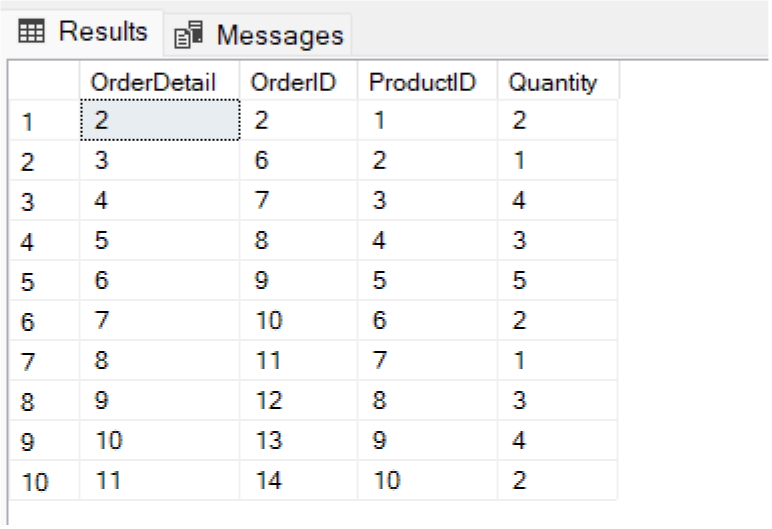
	OrderID	CustomerID	OrderDate	TotalAmount
1	34	1	2024-01-15	1399
2	35	2	2024-02-18	999
3	36	3	2024-03-22	299
4	37	4	2024-04-12	1849
5	38	5	2024-05-05	499
6	39	6	2024-06-01	699
7	40	7	2024-06-15	1199
8	41	8	2024-07-08	299
9	42	9	2024-07-22	499
10	43	10	2024-08-17	599

```
INSERT INTO Inventory (ProductID, QuantityInStock, LastStockUpdate) VALUES
(1, 50, '2024-01-15'),
(2, 30, '2024-01-18'),
(3, 20, '2024-01-20'),
(4, 15, '2024-01-22'),
(5, 10, '2024-01-25'),
(6, 25, '2024-01-28'),
(7, 40, '2024-01-30'),
(8, 60, '2024-02-02'),
(9, 5, '2024-02-05'),
(10, 12, '2024-02-08');
```



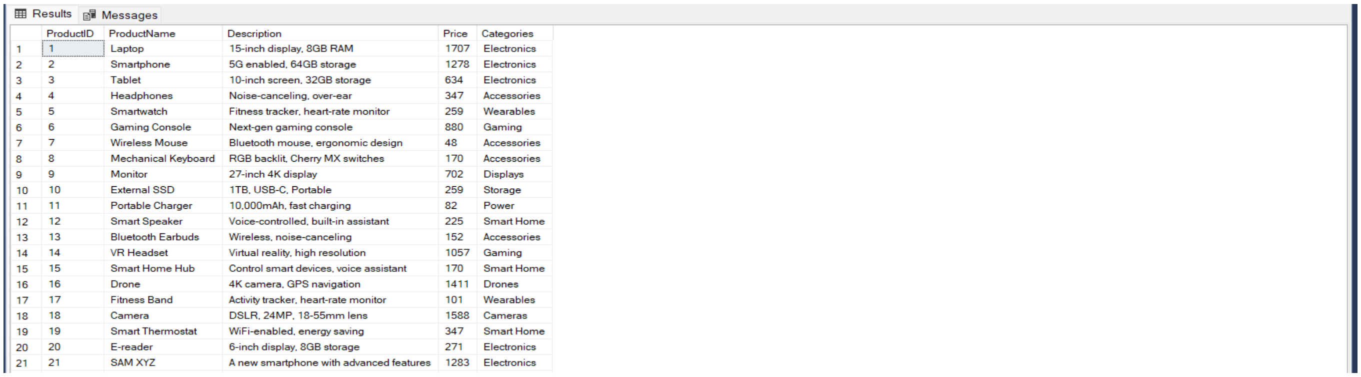
	InventoryID	ProductID	QuantityInStock	LastStockUpdate
1	1	1	50	2024-01-15
2	2	2	30	2024-01-18
3	3	3	20	2024-01-20
4	4	4	15	2024-01-22
5	5	5	10	2024-01-25
6	6	6	25	2024-01-28
7	7	7	40	2024-01-30
8	8	8	60	2024-02-02
9	9	9	5	2024-02-05
10	10	10	12	2024-02-08

```
INSERT INTO OrderDetails (OrderID, ProductID, Quantity) VALUES
(2, 1, 2),
(6, 2, 1),
(7, 3, 4),
(8, 4, 3),
(9, 5, 5),
(10, 6, 2),
(11, 7, 1),
(12, 8, 3),
(13, 9, 4),
(14, 10, 2);
```



	OrderDetail	OrderID	ProductID	Quantity
1	2	2	1	2
2	3	6	2	1
3	4	7	3	4
4	5	8	4	3
5	6	9	5	5
6	7	10	6	2
7	8	11	7	1
8	9	12	8	3
9	10	13	9	4
10	11	14	10	2

```
/* Adding Categories Attribute to Products Table*/

ALTER TABLE Products
ADD Categories VARCHAR(255);
UPDATE Products
SET Categories = CASE
    WHEN ProductName IN ('Laptop', 'Tablet', 'Smartphone') THEN 'Electronics'
    WHEN ProductName IN ('Headphones', 'Wireless Mouse', 'Mechanical Keyboard',
'Bluetooth Earbuds') THEN 'Accessories'
    WHEN ProductName = 'Monitor' THEN 'Displays'
    WHEN ProductName = 'External SSD' THEN 'Storage'
    WHEN ProductName = 'Portable Charger' THEN 'Power'
    WHEN ProductName = 'Smart Speaker' THEN 'Smart Home'
    WHEN ProductName = 'Smartwatch' THEN 'Wearables'
    WHEN ProductName = 'VR Headset' THEN 'Gaming'
    WHEN ProductName = 'Gaming Console' THEN 'Gaming'
    WHEN ProductName = 'Smart Home Hub' THEN 'Smart Home'
    WHEN ProductName = 'Drone' THEN 'Drones'
    WHEN ProductName = 'Fitness Band' THEN 'Wearables'
    WHEN ProductName = 'Camera' THEN 'Cameras'
    WHEN ProductName = 'Smart Thermostat' THEN 'Smart Home'
    WHEN ProductName = 'E-reader' THEN 'Electronics'
    ELSE 'Other'
END;
```

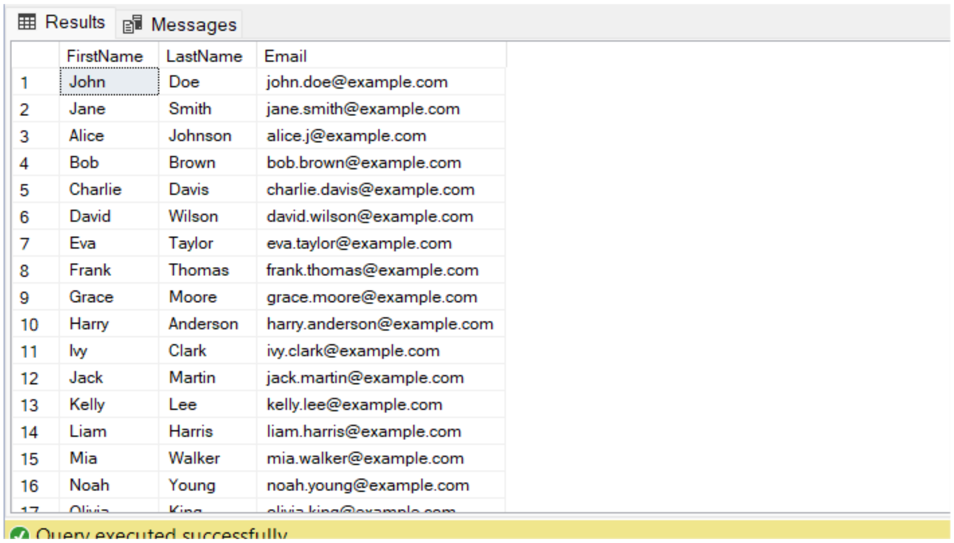


	ProductID	ProductName	Description	Price	Categories
1	1	Laptop	15-inch display, 8GB RAM	1707	Electronics
2	2	Smartphone	5G enabled, 64GB storage	1278	Electronics
3	3	Tablet	10-inch screen, 32GB storage	634	Electronics
4	4	Headphones	Noise-canceling, over-ear	347	Accessories
5	5	Smartwatch	Fitness tracker, heart-rate monitor	259	Wearables
6	6	Gaming Console	Next-gen gaming console	880	Gaming
7	7	Wireless Mouse	Bluetooth mouse, ergonomic design	48	Accessories
8	8	Mechanical Keyboard	RGB backlit, Cherry MX switches	170	Accessories
9	9	Monitor	27-inch 4K display	702	Displays
10	10	External SSD	1TB, USB-C, Portable	259	Storage
11	11	Portable Charger	10,000mAh, fast charging	82	Power
12	12	Smart Speaker	Voice-controlled, built-in assistant	225	Smart Home
13	13	Bluetooth Earbuds	Wireless, noise-canceling	152	Accessories
14	14	VR Headset	Virtual reality, high resolution	1057	Gaming
15	15	Smart Home Hub	Control smart devices, voice assistant	170	Smart Home
16	16	Drone	4K camera, GPS navigation	1411	Drones
17	17	Fitness Band	Activity tracker, heart-rate monitor	101	Wearables
18	18	Camera	DSLR, 24MP, 18-55mm lens	1588	Cameras
19	19	Smart Thermostat	WiFi-enabled, energy saving	347	Smart Home
20	20	E-reader	6-inch display, 8GB storage	271	Electronics
21	21	SAM XYZ	A new smartphone with advanced features	1283	Electronics

Tasks 2

1. **/ *Write an SQL query to retrieve the names and emails of all customers. */**

```
SELECT FirstName, LastName, Email
FROM  customers;
```

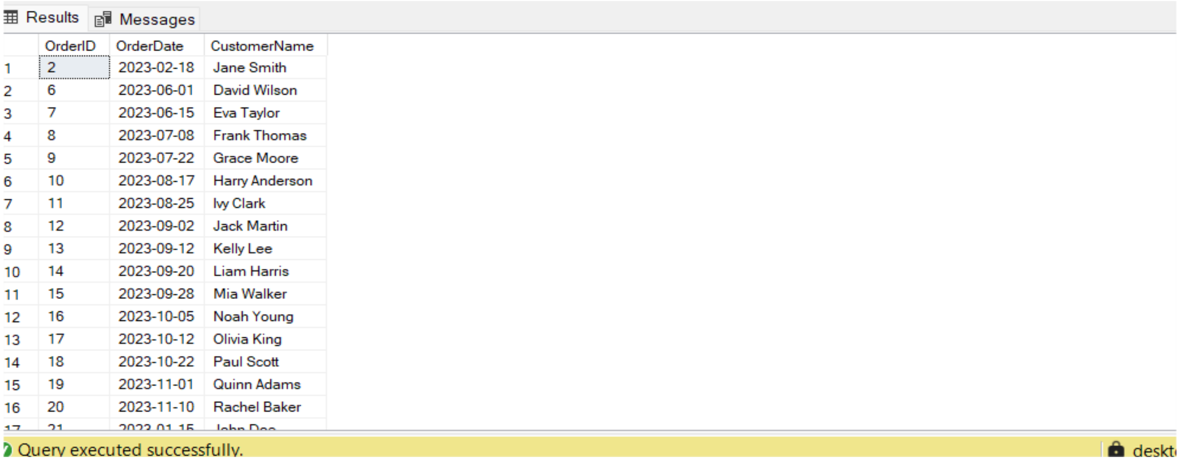


	FirstName	LastName	Email
1	John	Doe	john.doe@example.com
2	Jane	Smith	jane.smith@example.com
3	Alice	Johnson	alice.j@example.com
4	Bob	Brown	bob.brown@example.com
5	Charlie	Davis	charlie.davis@example.com
6	David	Wilson	david.wilson@example.com
7	Eva	Taylor	eva.taylor@example.com
8	Frank	Thomas	frank.thomas@example.com
9	Grace	Moore	grace.moore@example.com
10	Harry	Anderson	harry.anderson@example.com
11	Ivy	Clark	ivy.clark@example.com
12	Jack	Martin	jack.martin@example.com
13	Kelly	Lee	kelly.lee@example.com
14	Liam	Harris	liam.harris@example.com
15	Mia	Walker	mia.walker@example.com
16	Noah	Young	noah.young@example.com
17	Olivia	King	olivia.king@example.com

Query executed successfully.

2. **/* Write an SQL query to list all orders with their order dates and corresponding customer names. */**

```
SELECT OrderID,
 OrderDate,
 FirstName + ' ' + LastName AS CustomerName
 FROM
 Orders, Customers
  where
 Orders.CustomerID = Customers.CustomerID;
```

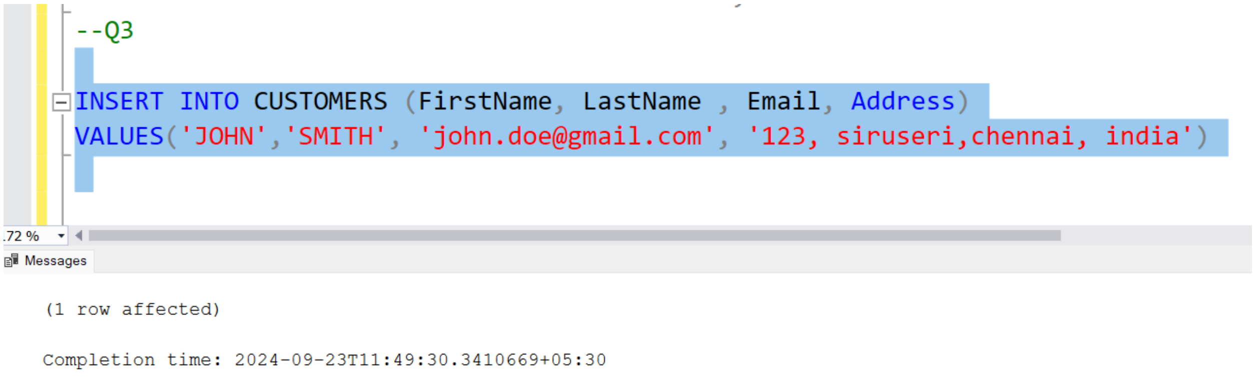


	OrderID	OrderDate	CustomerName
1	2	2023-02-18	Jane Smith
2	6	2023-06-01	David Wilson
3	7	2023-06-15	Eva Taylor
4	8	2023-07-08	Frank Thomas
5	9	2023-07-22	Grace Moore
6	10	2023-08-17	Harry Anderson
7	11	2023-08-25	Ivy Clark
8	12	2023-09-02	Jack Martin
9	13	2023-09-12	Kelly Lee
10	14	2023-09-20	Liam Harris
11	15	2023-09-28	Mia Walker
12	16	2023-10-05	Noah Young
13	17	2023-10-12	Olivia King
14	18	2023-10-22	Paul Scott
15	19	2023-11-01	Quinn Adams
16	20	2023-11-10	Rachel Baker
17	21	2023-01-15	John Doe

Query executed successfully.

3. /* Write an SQL query to insert a new customer record into the "Customers" table. Include customer information such as name, email, and address. */

```
INSERT INTO CUSTOMERS (FirstName, LastName , Email, Address)
VALUES('JOHN','SMITH', 'john.doe@gmail.com', '123, siruseri,chennai, india')
```

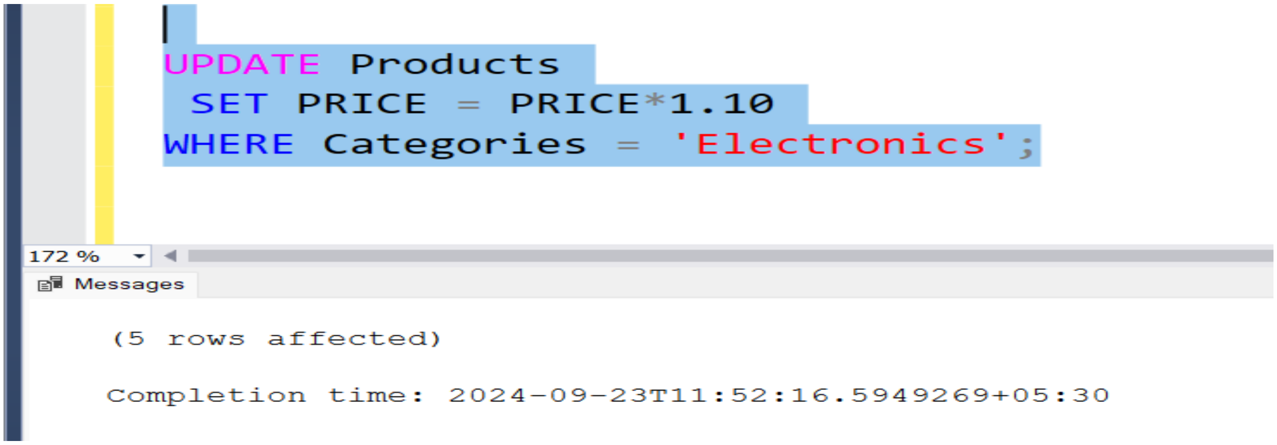


```
--Q3

INSERT INTO CUSTOMERS (FirstName, LastName , Email, Address)
VALUES('JOHN','SMITH', 'john.doe@gmail.com', '123, siruseri,chennai, india')
```

```
(1 row affected)

Completion time: 2024-09-23T11:49:30.3410669+05:30
```

4. /* Write an SQL query to update the prices of all electronic gadgets in the "Products" table by increasing them by 10%. */

```
UPDATE Products
 SET PRICE = PRICE*1.10
WHERE Categories = 'Electronics';
```

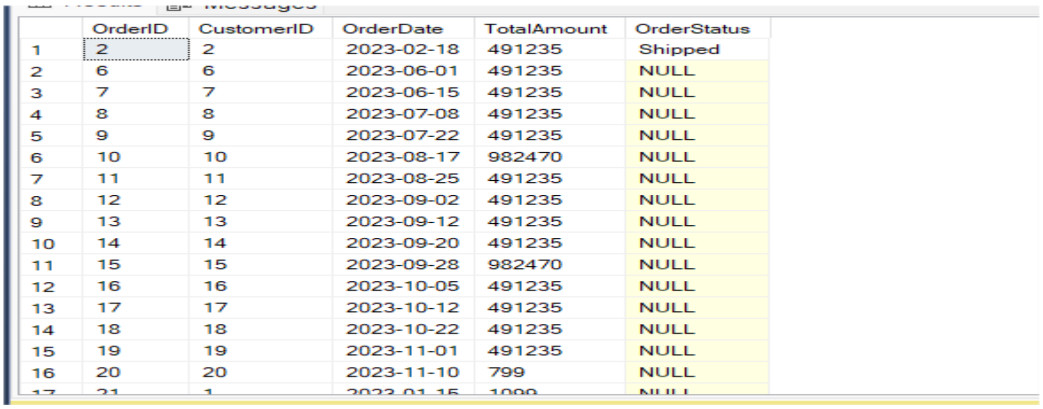


```
UPDATE Products
 SET PRICE = PRICE*1.10
WHERE Categories = 'Electronics';
```

```
(5 rows affected)

Completion time: 2024-09-23T11:52:16.5949269+05:30
```

/* 5. Write an SQL query to delete a specific order and its associated order details from the "Orders" and "OrderDetails" tables. Allow users to input the order ID as a parameter. */

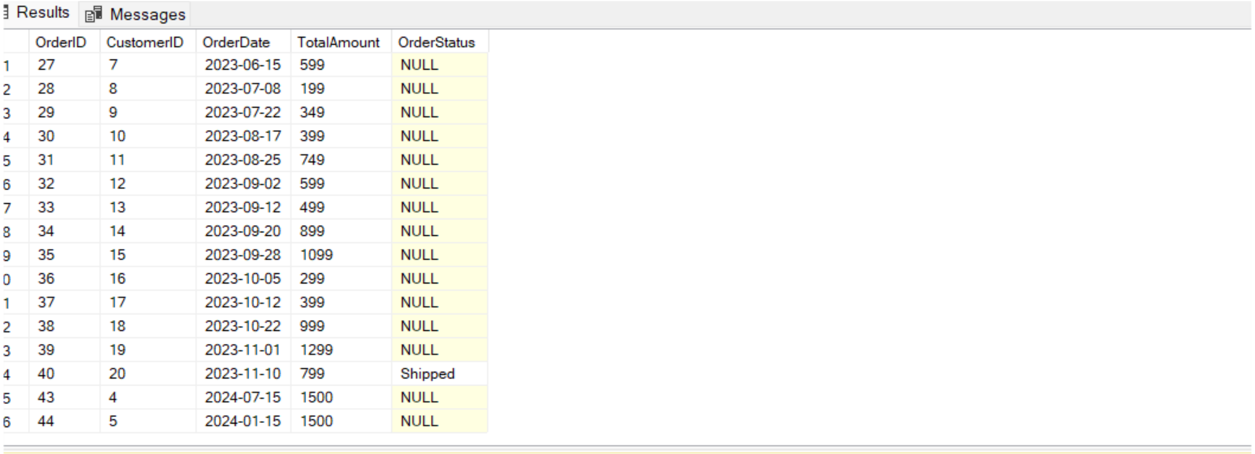
```
DECLARE  @orderID  int =  1;

         DELETE FROM OrderDetails
         WHERE ORDERiD = @orderID;

         delete from Orders
         where orderID =@orderID;

select * from Orders;
```



	OrderID	CustomerID	OrderDate	TotalAmount	OrderStatus
1	2	2	2023-02-18	491235	Shipped
2	6	6	2023-06-01	491235	NULL
3	7	7	2023-06-15	491235	NULL
4	8	8	2023-07-08	491235	NULL
5	9	9	2023-07-22	491235	NULL
6	10	10	2023-08-17	982470	NULL
7	11	11	2023-08-25	491235	NULL
8	12	12	2023-09-02	491235	NULL
9	13	13	2023-09-12	491235	NULL
10	14	14	2023-09-20	491235	NULL
11	15	15	2023-09-28	982470	NULL
12	16	16	2023-10-05	491235	NULL
13	17	17	2023-10-12	491235	NULL
14	18	18	2023-10-22	491235	NULL
15	19	19	2023-11-01	491235	NULL
16	20	20	2023-11-10	799	NULL
17	21	1	2023-01-15	1099	NULL

/* 6. Insert a new order into the "Orders" table. */

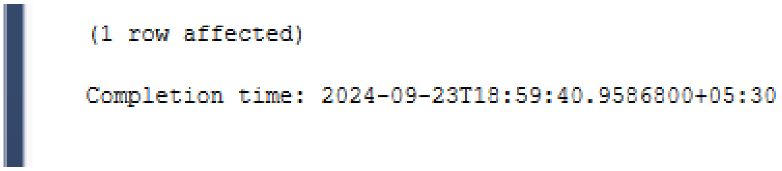
```
INSERT INTO Orders (CustomerID, OrderDate, TotalAmount)
VALUES (5, '2024-01-15', 1500);
```



	OrderID	CustomerID	OrderDate	TotalAmount	OrderStatus
1	27	7	2023-06-15	599	NULL
2	28	8	2023-07-08	199	NULL
3	29	9	2023-07-22	349	NULL
4	30	10	2023-08-17	399	NULL
5	31	11	2023-08-25	749	NULL
6	32	12	2023-09-02	599	NULL
7	33	13	2023-09-12	499	NULL
8	34	14	2023-09-20	899	NULL
9	35	15	2023-09-28	1099	NULL
0	36	16	2023-10-05	299	NULL
1	37	17	2023-10-12	399	NULL
2	38	18	2023-10-22	999	NULL
3	39	19	2023-11-01	1299	NULL
4	40	20	2023-11-10	799	Shipped
5	43	4	2024-07-15	1500	NULL
6	44	5	2024-01-15	1500	NULL

/* 7. Update the contact information of a specific customer in the "Customers" table. */

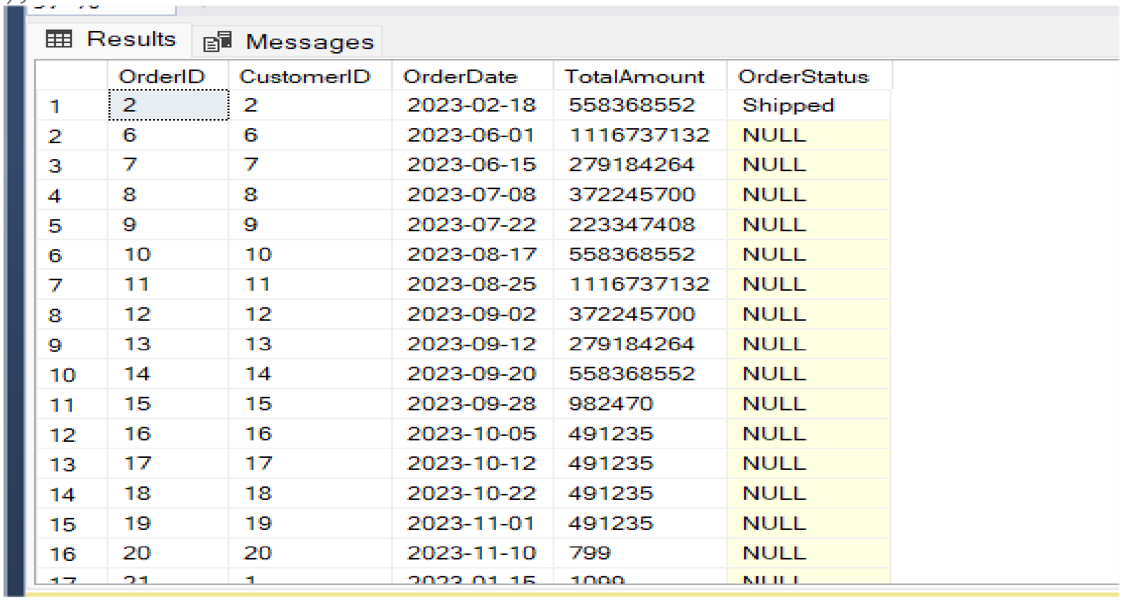
```sql
DECLARE @CustomerID INT;
DECLARE @NewPhone NVARCHAR(20);
DECLARE @NewEmail NVARCHAR(255);
DECLARE @NewAddress NVARCHAR(255);
SET @CustomerID = 21;
SET @NewEmail = 'ACVZ@GMAIL.COM'
SET @NewAddress = '1DQW, WALL STREET, NEW YORK, U.S.A'
SET @NewPhone = '123-456-8490';
UPDATE Customers
SET Email = @NewEmail,  Address = @NewAddress,  Phone = @NewPhone
WHERE CustomerID = @CustomerID;
```



```
(1 row affected)

Completion time: 2024-09-23T18:59:40.9586800+05:30
```

/* 8. Recalculate and update the total cost of each order in the "Orders" table. */

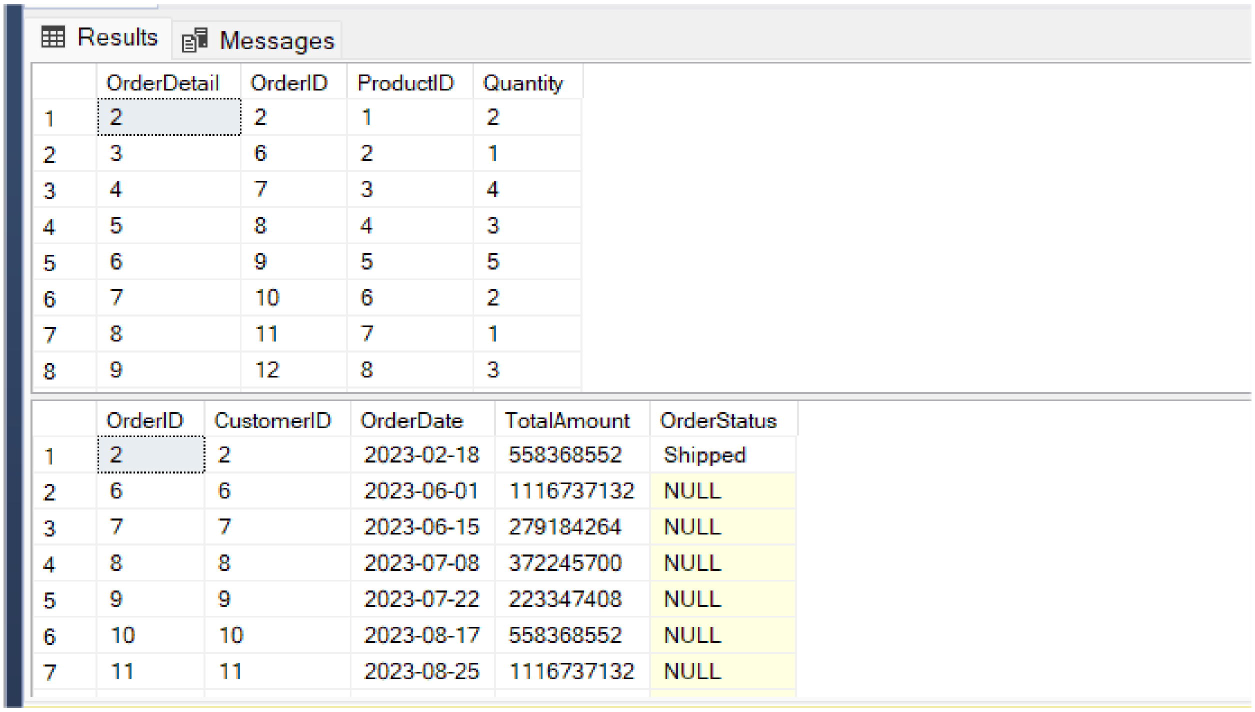
```sql
select * from Orders

UPDATE Orders
SET TotalAmount = (
SELECT SUM(o.TotalAmount * od.Quantity)
FROM OrderDetails od  , Orders o
WHERE od.OrderID = Orders.OrderId
)
WHERE EXISTS (
SELECT 1
FROM OrderDetails od
WHERE od.OrderID = Orders.OrderID

);
```



	OrderID	CustomerID	OrderDate	TotalAmount	OrderStatus
1	2	2	2023-02-18	558368552	Shipped
2	6	6	2023-06-01	1116737132	NULL
3	7	7	2023-06-15	279184264	NULL
4	8	8	2023-07-08	372245700	NULL
5	9	9	2023-07-22	223347408	NULL
6	10	10	2023-08-17	558368552	NULL
7	11	11	2023-08-25	1116737132	NULL
8	12	12	2023-09-02	372245700	NULL
9	13	13	2023-09-12	279184264	NULL
10	14	14	2023-09-20	558368552	NULL
11	15	15	2023-09-28	982470	NULL
12	16	16	2023-10-05	491235	NULL
13	17	17	2023-10-12	491235	NULL
14	18	18	2023-10-22	491235	NULL
15	19	19	2023-11-01	491235	NULL
16	20	20	2023-11-10	799	NULL
17	21	1	2023-01-15	1000	NULL

/* 9. Delete all orders and their associated order details for a specific customer. */

```
DECLARE @CustomerID INT = 4;

DELETE FROm OrderDetails
WHERE OrderID IN (
SELECT OrderID FROM Orders WHERE CustomerID = @CustomerID
);
DELETE FROM Orders
WHERE CustomerID = @CustomerID;
select * from OrderDetails
Select  * from  Orders
```

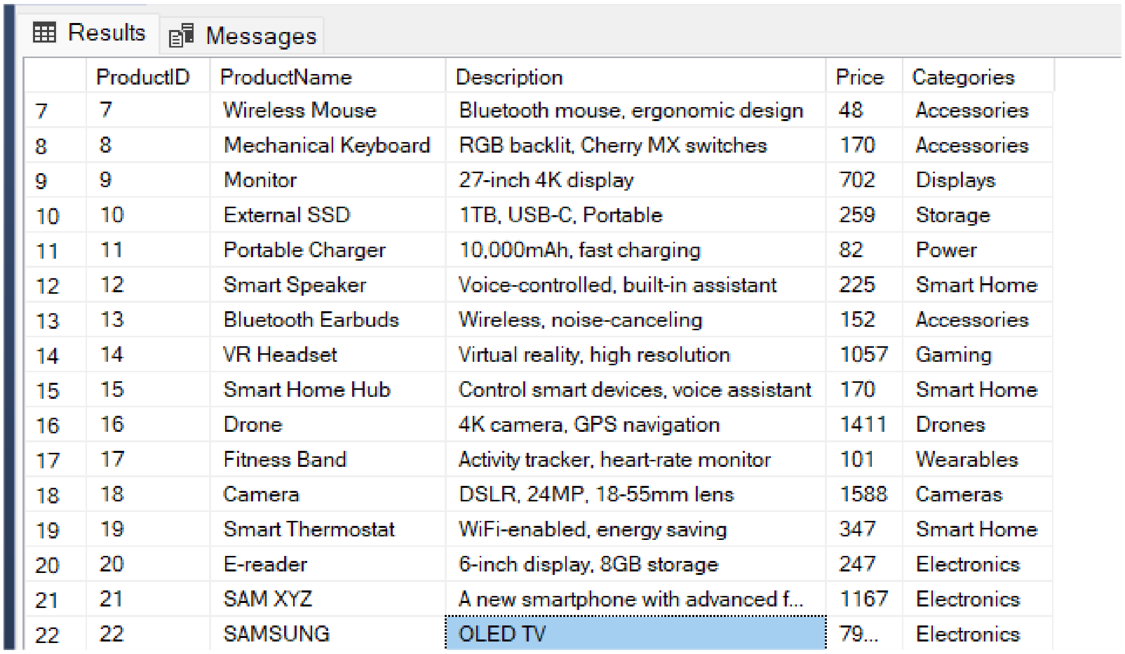


	OrderDetail	OrderID	ProductID	Quantity
1	2	2	1	2
2	3	6	2	1
3	4	7	3	4
4	5	8	4	3
5	6	9	5	5
6	7	10	6	2
7	8	11	7	1
8	9	12	8	3

	OrderID	CustomerID	OrderDate	TotalAmount	OrderStatus
1	2	2	2023-02-18	558368552	Shipped
2	6	6	2023-06-01	1116737132	NULL
3	7	7	2023-06-15	279184264	NULL
4	8	8	2023-07-08	372245700	NULL
5	9	9	2023-07-22	223347408	NULL
6	10	10	2023-08-17	558368552	NULL
7	11	11	2023-08-25	1116737132	NULL

/* 10. Insert a new electronic gadget product into the "Products" table. */

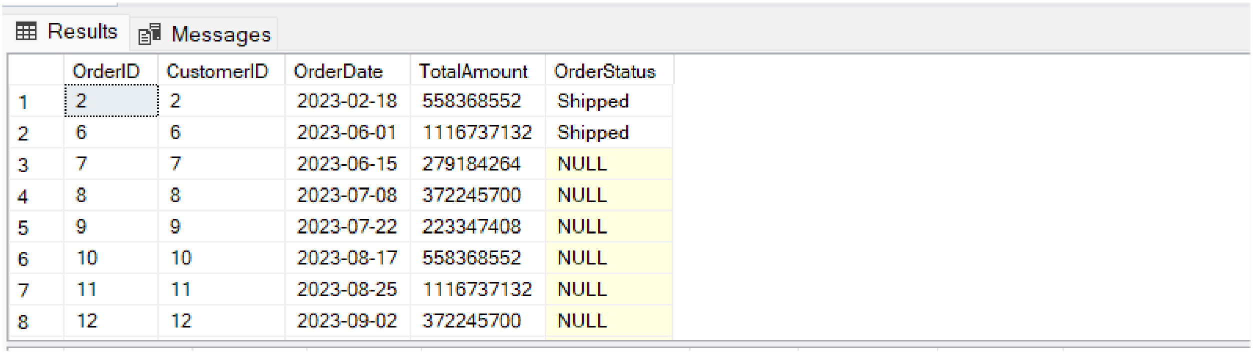
```
SELECT * FROM Products
INSERT INTO Products ( ProductName, Description, Price,Categories)
VALUES ('SAMSUNG', 'OLED TV', 79099.99,'Electronics');
```



	ProductID	ProductName	Description	Price	Categories
7	7	Wireless Mouse	Bluetooth mouse, ergonomic design	48	Accessories
8	8	Mechanical Keyboard	RGB backlit, Cherry MX switches	170	Accessories
9	9	Monitor	27-inch 4K display	702	Displays
10	10	External SSD	1TB, USB-C, Portable	259	Storage
11	11	Portable Charger	10,000mAh, fast charging	82	Power
12	12	Smart Speaker	Voice-controlled, built-in assistant	225	Smart Home
13	13	Bluetooth Earbuds	Wireless, noise-canceling	152	Accessories
14	14	VR Headset	Virtual reality, high resolution	1057	Gaming
15	15	Smart Home Hub	Control smart devices, voice assistant	170	Smart Home
16	16	Drone	4K camera, GPS navigation	1411	Drones
17	17	Fitness Band	Activity tracker, heart-rate monitor	101	Wearables
18	18	Camera	DSLR, 24MP, 18-55mm lens	1588	Cameras
19	19	Smart Thermostat	WiFi-enabled, energy saving	347	Smart Home
20	20	E-reader	6-inch display, 8GB storage	247	Electronics
21	21	SAM XYZ	A new smartphone with advanced f...	1167	Electronics
22	22	SAMSUNG	OLED TV	79...	Electronics

/* 11. Update the status of a specific order in the "Orders" table. */

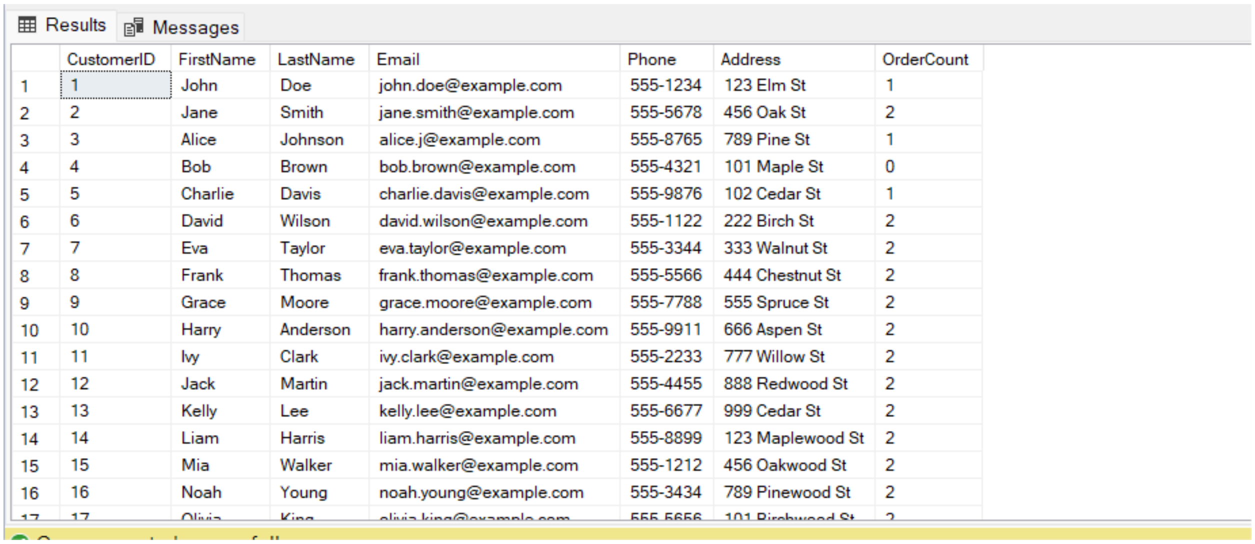
```
SELECT * FROM Orders

ALTER TABLE Orders
ADD OrderStatus NVARCHAR(50);
DECLARE @OrderID INT= 6;
DECLARE @NewStatus NVARCHAR(50) = 'Shipped';
UPDATE Orders
SET OrderStatus = @NewStatus
WHERE OrderID = @OrderID;
```



	OrderID	CustomerID	OrderDate	TotalAmount	OrderStatus
1	2	2	2023-02-18	558368552	Shipped
2	6	6	2023-06-01	1116737132	Shipped
3	7	7	2023-06-15	279184264	NULL
4	8	8	2023-07-08	372245700	NULL
5	9	9	2023-07-22	223347408	NULL
6	10	10	2023-08-17	558368552	NULL
7	11	11	2023-08-25	1116737132	NULL
8	12	12	2023-09-02	372245700	NULL

/* 12. Calculate and update the number of orders placed by each customer in the "Customers" table. */

```
ALTER TABLE Customers
ADD OrderCount INT;
SELECT * FROM CUSTOMERS
UPDATE Customers
SET OrderCount = (
SELECT COUNT(*)
FROM Orders
WHERE Orders.CustomerID = Customers.CustomerID
);
```

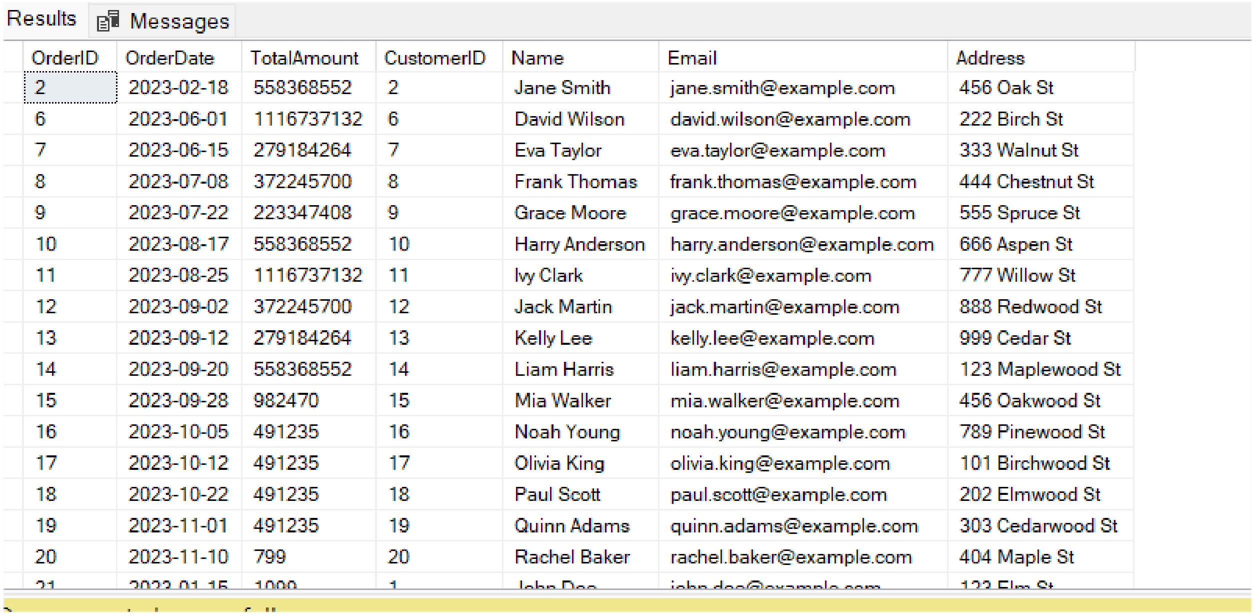


	CustomerID	FirstName	LastName	Email	Phone	Address	OrderCount
1	1	John	Doe	john.doe@example.com	555-1234	123 Elm St	1
2	2	Jane	Smith	jane.smith@example.com	555-5678	456 Oak St	2
3	3	Alice	Johnson	alice.j@example.com	555-8765	789 Pine St	1
4	4	Bob	Brown	bob.brown@example.com	555-4321	101 Maple St	0
5	5	Charlie	Davis	charlie.davis@example.com	555-9876	102 Cedar St	1
6	6	David	Wilson	david.wilson@example.com	555-1122	222 Birch St	2
7	7	Eva	Taylor	eva.taylor@example.com	555-3344	333 Walnut St	2
8	8	Frank	Thomas	frank.thomas@example.com	555-5566	444 Chestnut St	2
9	9	Grace	Moore	grace.moore@example.com	555-7788	555 Spruce St	2
10	10	Harry	Anderson	harry.anderson@example.com	555-9911	666 Aspen St	2
11	11	Ivy	Clark	ivy.clark@example.com	555-2233	777 Willow St	2
12	12	Jack	Martin	jack.martin@example.com	555-4455	888 Redwood St	2
13	13	Kelly	Lee	kelly.lee@example.com	555-6677	999 Cedar St	2
14	14	Liam	Harris	liam.harris@example.com	555-8899	123 Maplewood St	2
15	15	Mia	Walker	mia.walker@example.com	555-1212	456 Oakwood St	2
16	16	Noah	Young	noah.young@example.com	555-3434	789 Pinewood St	2
17	17	Olivia	King	olivia.king@example.com	555-5656	101 Birchwood St	2

Tasks 3

/* 1. Retrieve a list of all orders along with customer information for each order. */

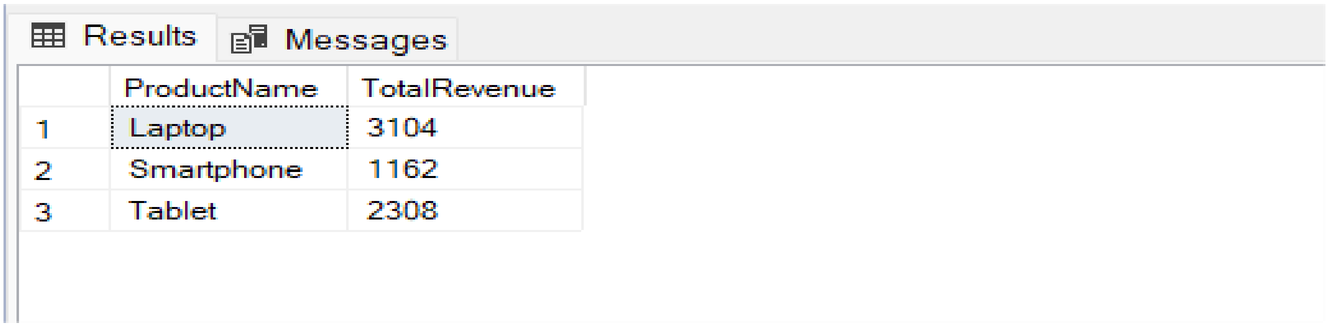
```
SELECT
    o.OrderID,
    o.OrderDate,
    o.TotalAmount,
    c.CustomerID,
    c. FirstName + ' ' + LastName as Name,
    c.Email,
    c.Address
FROM
    Orders o
JOIN
    Customers c ON o.CustomerID = c.CustomerID
```



OrderID	OrderDate	TotalAmount	CustomerID	Name	Email	Address
2	2023-02-18	558368552	2	Jane Smith	jane.smith@example.com	456 Oak St
6	2023-06-01	1116737132	6	David Wilson	david.wilson@example.com	222 Birch St
7	2023-06-15	279184264	7	Eva Taylor	eva.taylor@example.com	333 Walnut St
8	2023-07-08	372245700	8	Frank Thomas	frank.thomas@example.com	444 Chestnut St
9	2023-07-22	223347408	9	Grace Moore	grace.moore@example.com	555 Spruce St
10	2023-08-17	558368552	10	Harry Anderson	harry.anderson@example.com	666 Aspen St
11	2023-08-25	1116737132	11	Ivy Clark	ivy.clark@example.com	777 Willow St
12	2023-09-02	372245700	12	Jack Martin	jack.martin@example.com	888 Redwood St
13	2023-09-12	279184264	13	Kelly Lee	kelly.lee@example.com	999 Cedar St
14	2023-09-20	558368552	14	Liam Harris	liam.harris@example.com	123 Maplewood St
15	2023-09-28	982470	15	Mia Walker	mia.walker@example.com	456 Oakwood St
16	2023-10-05	491235	16	Noah Young	noah.young@example.com	789 Pinewood St
17	2023-10-12	491235	17	Olivia King	olivia.king@example.com	101 Birchwood St
18	2023-10-22	491235	18	Paul Scott	paul.scott@example.com	202 Elmwood St
19	2023-11-01	491235	19	Quinn Adams	quinn.adams@example.com	303 Cedarwood St
20	2023-11-10	799	20	Rachel Baker	rachel.baker@example.com	404 Maple St
21	2023-01-15	1000	1	John Doe	john.doe@example.com	123 Elm St

/* 2. Find the total revenue generated by each electronic gadget product. */

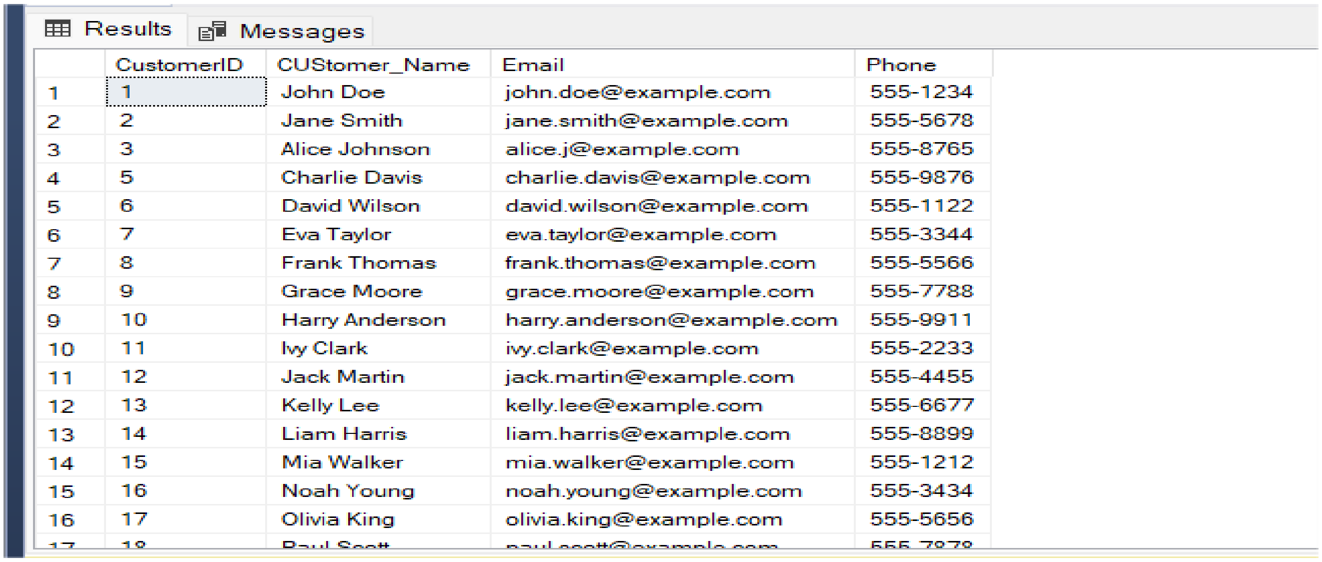
```sql
SELECT p.ProductName,
       SUM(od.Quantity * p.Price) AS TotalRevenue
FROM Products p
JOIN OrderDetails od ON p.ProductID = od.ProductID
WHERE p.Categories = 'Electronics'
GROUP BY p.ProductName;
```



	ProductName	TotalRevenue
1	Laptop	3104
2	Smartphone	1162
3	Tablet	2308

/* 3. List all customers who have made at least one purchase. */

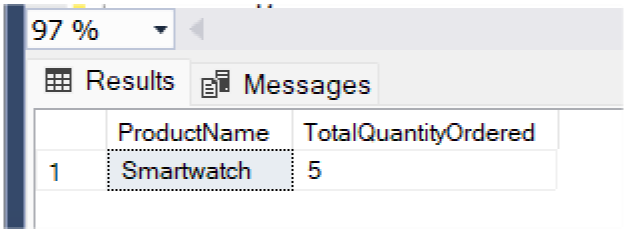
```sql
SELECT
c.CustomerID,
c.FirstName + ' ' + LastName   'CUStomer_Name',
c.Email,
c.Phone
FROM
CUSTOMERS as c
Join Orders o ON c.CustomerID = o.CustomerID
Group By
c.CustomerID,  c.FirstName + ' ' + LastName , c.Email, c.Phone;
```



	CustomerID	CUStomer_Name	Email	Phone
1	1	John Doe	john.doe@example.com	555-1234
2	2	Jane Smith	jane.smith@example.com	555-5678
3	3	Alice Johnson	alice.j@example.com	555-8765
4	5	Charlie Davis	charlie.davis@example.com	555-9876
5	6	David Wilson	david.wilson@example.com	555-1122
6	7	Eva Taylor	eva.taylor@example.com	555-3344
7	8	Frank Thomas	frank.thomas@example.com	555-5566
8	9	Grace Moore	grace.moore@example.com	555-7788
9	10	Harry Anderson	harry.anderson@example.com	555-9911
10	11	Ivy Clark	ivy.clark@example.com	555-2233
11	12	Jack Martin	jack.martin@example.com	555-4455
12	13	Kelly Lee	kelly.lee@example.com	555-6677
13	14	Liam Harris	liam.harris@example.com	555-8899
14	15	Mia Walker	mia.walker@example.com	555-1212
15	16	Noah Young	noah.young@example.com	555-3434
16	17	Olivia King	olivia.king@example.com	555-5656
17	18	Paul Scott	paul.scott@example.com	555-7878

/* 4. Find the most popular electronic gadget with the highest total quantity ordered. */

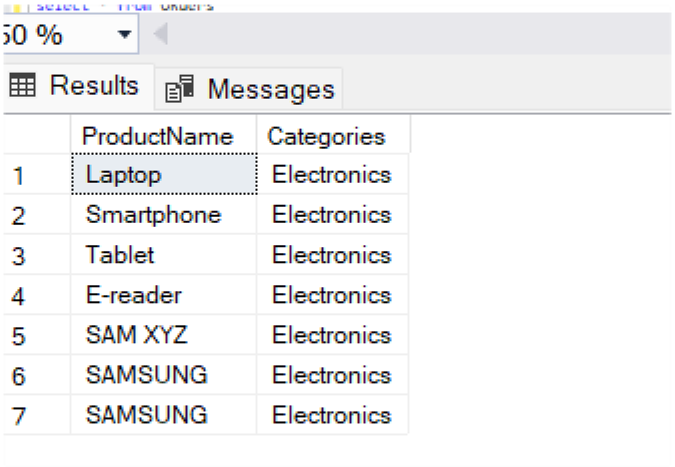
```
SELECT TOP 1 p.ProductName, SUM(od.Quantity) AS TotalQuantityOrdered
FROM Products AS p
JOIN OrderDetails AS od ON p.ProductID = od.ProductID
GROUP BY p.ProductName
ORDER BY SUM(od.Quantity) DESC;
```



	ProductName	TotalQuantityOrdered
1	Smartwatch	5

/* 5. Retrieve a list of electronic gadgets along with their corresponding categories. */

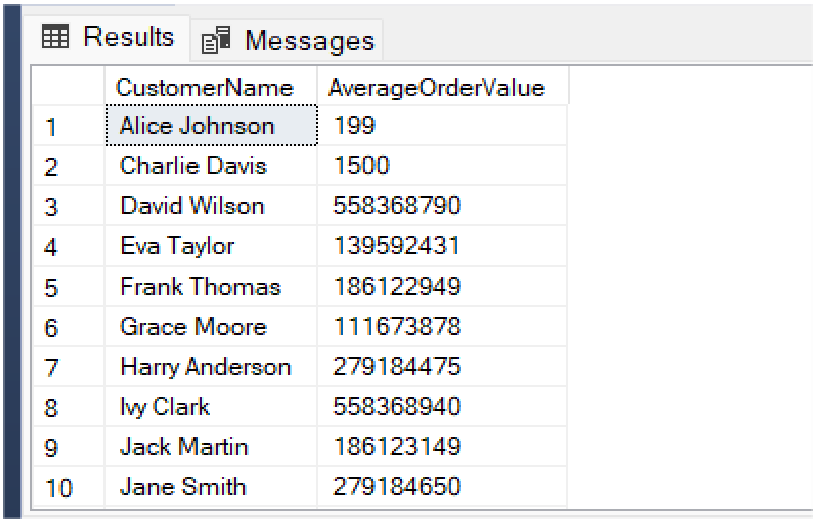
```
SELECT ProductName, Categories
from Products
Where
Categories = 'Electronics'
```



	ProductName	Categories
1	Laptop	Electronics
2	Smartphone	Electronics
3	Tablet	Electronics
4	E-reader	Electronics
5	SAM XYZ	Electronics
6	SAMSUNG	Electronics
7	SAMSUNG	Electronics

/* 6. Calculate the average order value for each customer. */

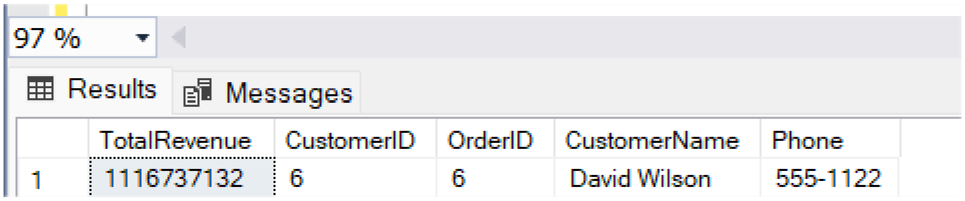
```
Select c.FirstName + ' '+ LastName  'CustomerName',
AVG(o.TotalAmount) as AverageOrderValue
FROM
CUSTOMERS c
JOIN Orders o ON c.CustomerID =o.CustomerID
Group BY
c.FirstName + ' '+ LastName;
```



	CustomerName	AverageOrderValue
1	Alice Johnson	199
2	Charlie Davis	1500
3	David Wilson	558368790
4	Eva Taylor	139592431
5	Frank Thomas	186122949
6	Grace Moore	111673878
7	Harry Anderson	279184475
8	Ivy Clark	558368940
9	Jack Martin	186123149
10	Jane Smith	279184650

/* 7. Find the order with the highest total revenue. */

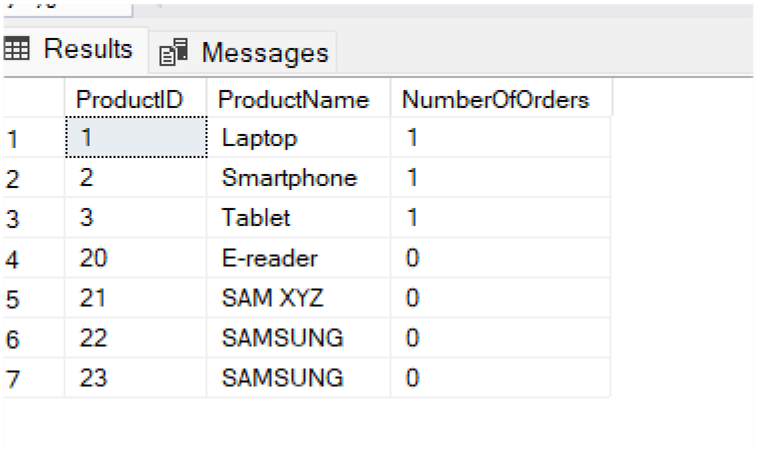
```
select TOP 1
o.TotalAmount AS TotalRevenue,
c.CustomerID,
o.OrderID,
c.FirstName + ' ' + LastName as CustomerName,
c.Phone
from Orders as o
Join CUSTOMERS as c ON c.CustomerID = o.CustomerID
ORDER BY o.TotalAmount DESC;
```



	TotalRevenue	CustomerID	OrderID	CustomerName	Phone
1	1116737132	6	6	David Wilson	555-1122

/* 8. List electronic gadgets and the number of times each product has been ordered. */

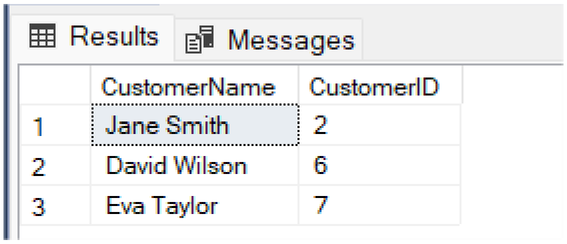
```
Select p.ProductID, p.ProductName, COUNT(od.OrderID) AS NumberOfOrders
From Products p
Left Join  OrderDetails as od ON p.ProductID = od.ProductID
WHERE p.Categories = 'Electronics'
GROUP BY
P.ProductID  , p.ProductName
```



	ProductID	ProductName	NumberOfOrders
1	1	Laptop	1
2	2	Smartphone	1
3	3	Tablet	1
4	20	E-reader	0
5	21	SAM XYZ	0
6	22	SAMSUNG	0
7	23	SAMSUNG	0

/* 9. Find customers who have purchased a specific electronic gadget product. */

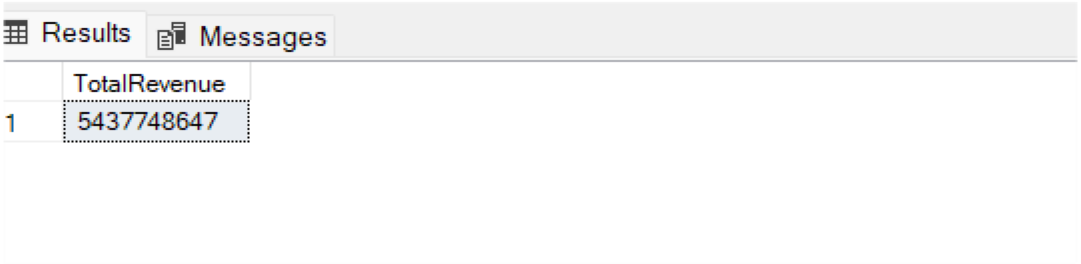
```
DECLARE @NameOfProduct VARCHAR(255) = 'Electronics';
SELECT c.FirstName + ' '+ c.LastName as CustomerName,
c.CustomerID
from CUSTOMERS c

Join Orders as o ON c.CustomerID = o.CustomerID
Join OrderDetails as od ON od.OrderID  = o.OrderID
Join Products as p ON p.ProductID = od.ProductID
where p.Categories = @NameOfProduct;
```



	CustomerName	CustomerID
1	Jane Smith	2
2	David Wilson	6
3	Eva Taylor	7

/* 10. Calculate the total revenue generated by all orders placed within a specific time period. */

```
DECLARE @STARTDATE DATE = '2020-06-01'
DECLARE  @ENDDATE DATE = '2024-09-01'

SELECT SUM(TotalAmount ) AS TotalRevenue
From
Orders
Where
OrderDate Between @STARTDATE and @ENDDATE
```

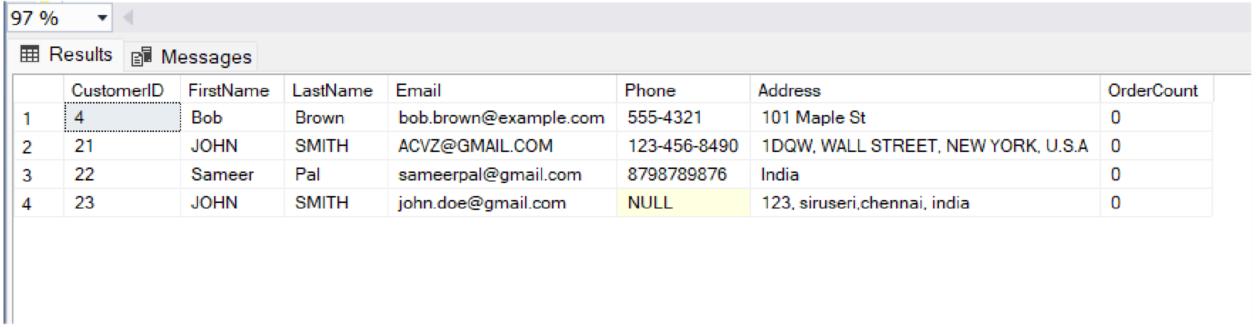


	TotalRevenue
1	5437748647

Tasks 4

/* 1. Find out which customers have not placed any orders. */

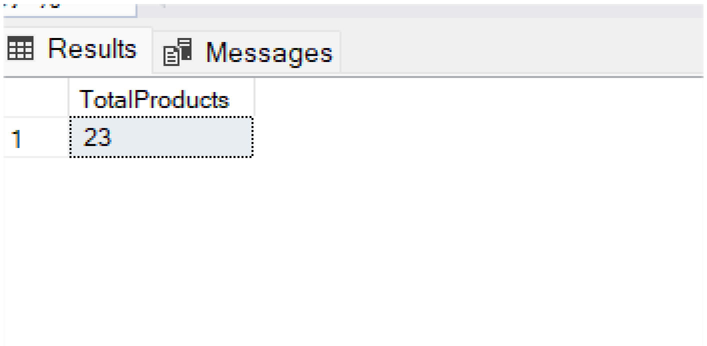
```
Select c.FirstName+' '+c.LastName as NAME
from CUSTOMERS c
WHERE c.CustomerID NOT IN (SELECT o.CustomerID FROM Orders o )
```



	CustomerID	FirstName	LastName	Email	Phone	Address	OrderCount
1	4	Bob	Brown	bob.brown@example.com	555-4321	101 Maple St	0
2	21	JOHN	SMITH	ACVZ@GMAIL.COM	123-456-8490	1DQW, WALL STREET, NEW YORK, U.S.A	0
3	22	Sameer	Pal	sameerpal@gmail.com	8798789876	India	0
4	23	JOHN	SMITH	john.doe@gmail.com	NULL	123, siruseri,chennai, india	0

/* 2. Find the total number of products available for sale. */

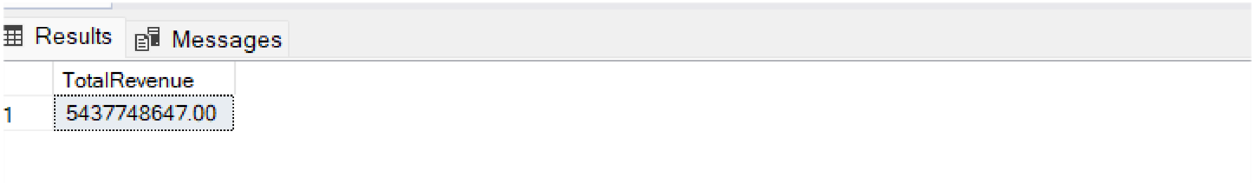
```
SELECT COUNT (*) AS TotalProducts
FROM Products
```



	TotalProducts
1	23

/* 3. Calculate the total revenue generated by TechShop. */

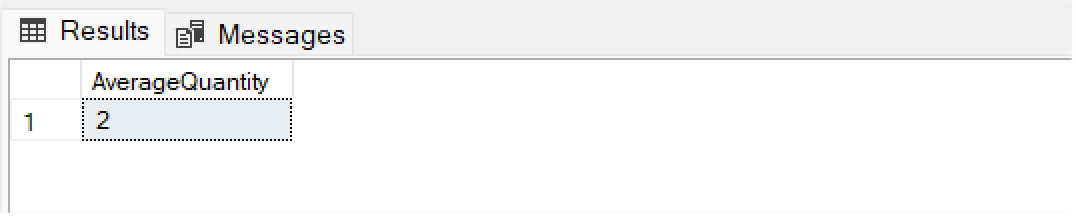
```
SELECT SUM(TotalAmount) AS TotalRevenue
FROM Orders;
```



	TotalRevenue
1	5437748647.00

/* 4. Calculate the average quantity ordered for products in a specific category. */

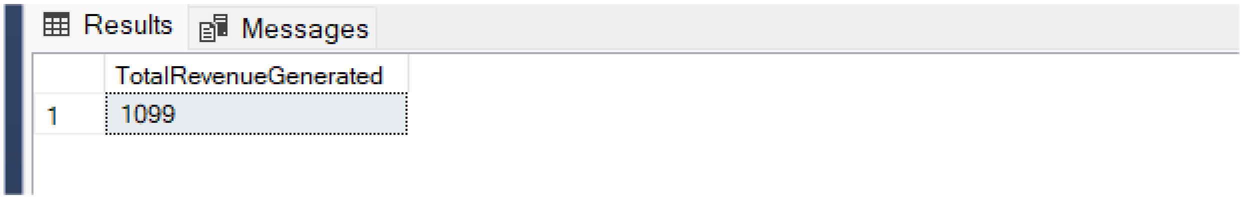
```
DECLARE @CategoryName VARCHAR(255) = 'Electronics';

SELECT AVG(Quantity) AS AverageQuantity
FROM OrderDetails
WHERE ProductID IN (
    SELECT ProductID
    FROM Products
    WHERE Categories = @CategoryName
);
```



	AverageQuantity
1	2

/* 5. Calculate the total revenue generated by a specific customer. */

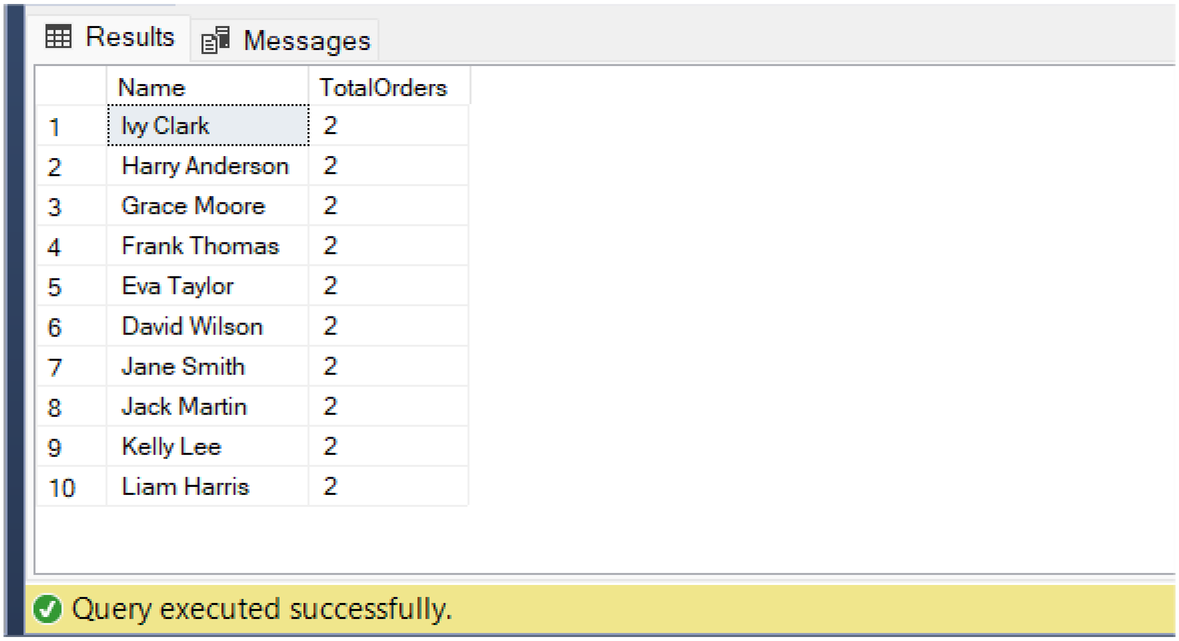
```
DECLARE @Customer_ID int=1;
Select Sum(TotalAmount) as TotalRevenueGenerated from Orders
where CustomerID  =  (Select CustomerID = @Customer_ID)
```



	TotalRevenueGenerated
1	1099

/* 6. Find the customers who have placed the most orders. */

```
SELECT top 10
    c.FirstName + ' ' + c.LastName AS Name,
    (
        SELECT COUNT(o.OrderID)
FROM Orders o
WHERE o.CustomerID = c.CustomerID
) AS TotalOrders
FROM
CUSTOMERS c
ORDER BY
TotalOrders DESC;
```

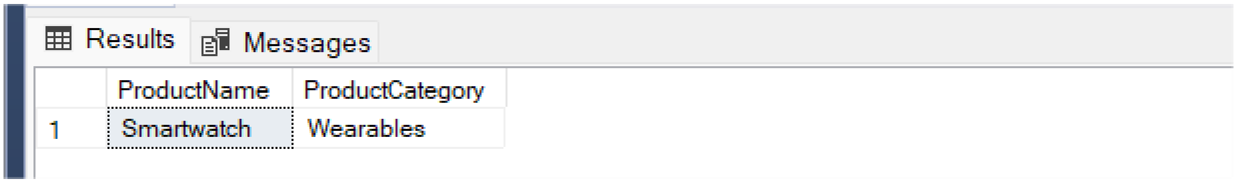


	Name	TotalOrders
1	Ivy Clark	2
2	Harry Anderson	2
3	Grace Moore	2
4	Frank Thomas	2
5	Eva Taylor	2
6	David Wilson	2
7	Jane Smith	2
8	Jack Martin	2
9	Kelly Lee	2
10	Liam Harris	2

Query executed successfully.

/* 7. Find the most popular product category with the highest total quantity ordered. */

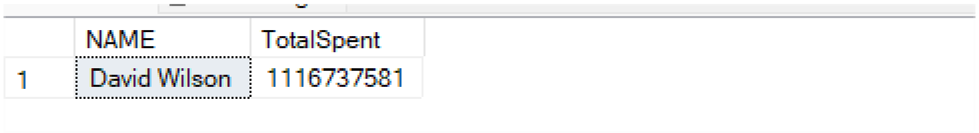
```
Select p.ProductName, p.Categories as ProductCategory
from Products as p
Where p.ProductID = (Select Top  1 od.ProductID
from  OrderDetails od
Group By od.ProductID
Order By SUM(od.Quantity) desc)
```



	ProductName	ProductCategory
1	Smartwatch	Wearables

/* 8. Find the customer who has spent the most money on electronic gadgets. */

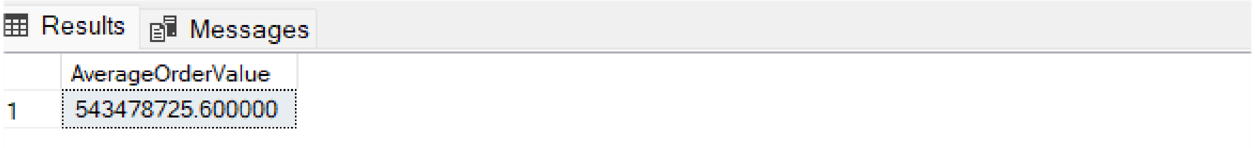
```
SELECT c.FirstName+' '+ c.LastName as NAME,
SUM(o.TotalAmount) AS TotalSpent
from CUSTOMERS c
Join Orders o ON o.CustomerID = c.CustomerID
Where o.CustomerID IN
( SELECT TOP 1 o.CustomerID
from Orders o
where o.OrderID in (
Select od.OrderID
from OrderDetails od
Where od.ProductID In (
SELECT p.ProductId
from Products p
Where p.Categories = 'Electronics'
)
)
GROUP BY o.CustomerID
ORder By SUM(o.TotalAmount) DESC
)
GROUP BY c.FirstName + ' '+ c.LastName;
```



	NAME	TotalSpent
1	David Wilson	1116737581

/* 9. Calculate the average order value for all customers. */

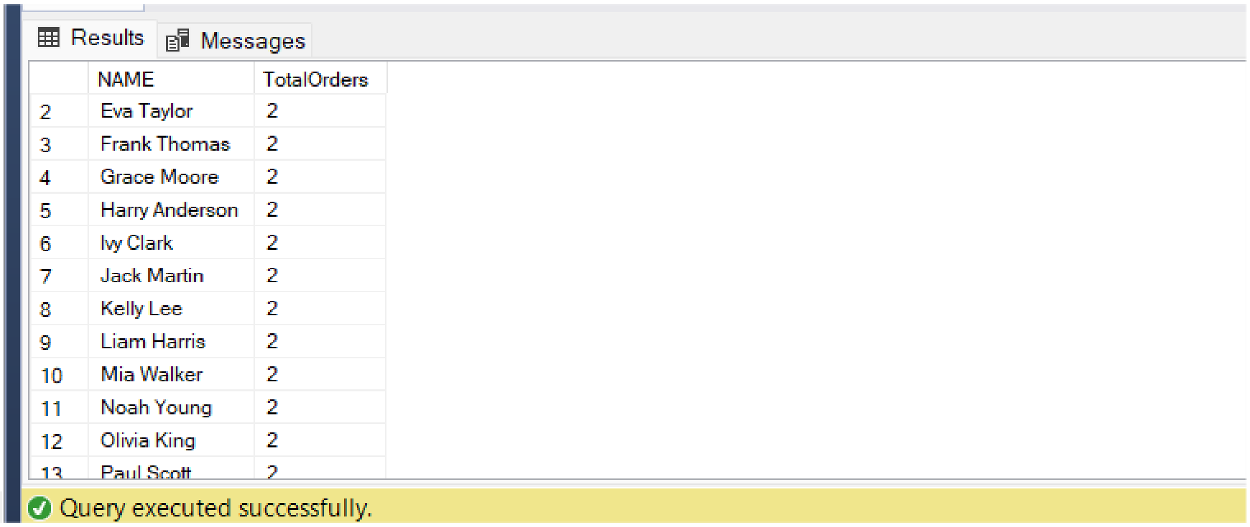
```
SELECT AVG(TotalRevenue/OrderCount) AS AverageOrderValue

From (
Select  o.CustomerID,
SUM(o.TotalAmount) AS TotalRevenue,
COUNT(o.OrderID) AS OrderCount
from Orders o
JOIN OrderDetails od ON od.OrderID = o.OrderID
Group By o.CustomerID
) AS OrderSummary
WHERE OrderCount > 0;
```



	AverageOrderValue
1	543478725.600000

/* 10. Find the total number of orders placed by each customer and list their names along with the order count. */

```sql
select c.FirstName + ' ' + c.LastName as NAME,

(SELECT COUNT(o.OrderID)
FROM Orders o
WHERE o.CustomerID  = c.CustomerID)

AS TotalOrders
From  CUSTOMERS c

Group By c.CustomerID, c.FirstName,c.LastName

Order BY TotalOrders DESC;
```



	NAME	TotalOrders
2	Eva Taylor	2
3	Frank Thomas	2
4	Grace Moore	2
5	Harry Anderson	2
6	Ivy Clark	2
7	Jack Martin	2
8	Kelly Lee	2
9	Liam Harris	2
10	Mia Walker	2
11	Noah Young	2
12	Olivia King	2
13	Paul Scott	2

Query executed successfully.