ASSIGNMENT

TECH SHOP-ELECTRONIC GADGETS SHOP

NAME: AKANSHA SINGH (P103)

TOPIC: TECH-SHOP , ELECTRONIC GADGET SHOP

TASK-1:

1.Create the database named "TechShop":

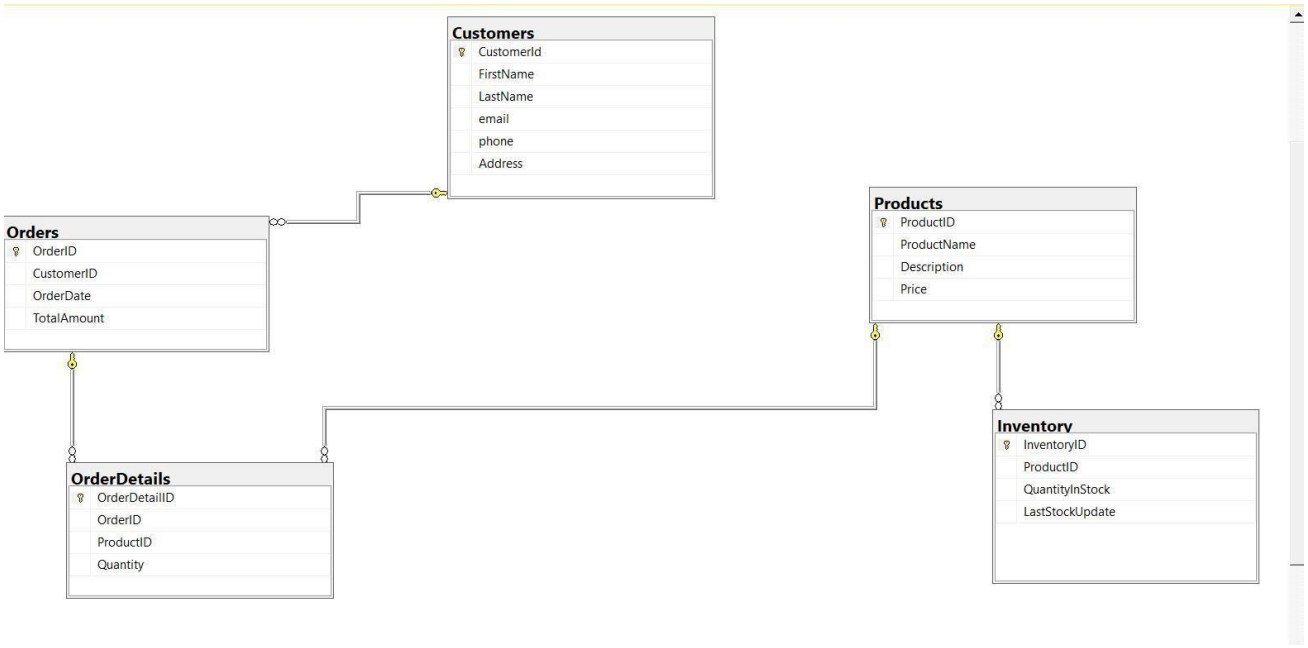
CREATE DATABASE TechShop; Use database
TechShop;

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

```
/* Creating Tables*/

/*1. Customers:
• CustomerID (Primary Key)
• FirstName
• LastName
• Email
• Phone  • Address*/   create table Customers(
CustomerId int Identity constraint C_PK Primary Key,
FirstName varchar(45) not null,
LastName varchar(40), email varchar(65)
not null, phone varchar(20),
Address varchar(80) not null);


/*. Products:
• ProductID (Primary Key)
• ProductName
• Description
• Price
*/
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
ProductName VARCHAR(100),
    Description VARCHAR(MAX),
    Price DECIMAL(10, 2)
);

/* Orders:
• OrderID (Primary Key)
```

```
• CustomerID (Foreign Key referencing Customers)
• OrderDate
• TotalAmount*/


CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    OrderDate DATE,
    TotalAmount DECIMAL(10, 5)
);

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


      CREATE TABLE OrderDetails (
    OrderDetailID INT PRIMARY KEY,
    OrderID INT,
    ProductID INT,
    Quantity INT,
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)  );


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


      CREATE TABLE Inventory (
    InventoryID INT PRIMARY KEY,
    ProductID INT,
    QuantityInStock INT,
    LastStockUpdate DATE,
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)  );
```

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



4. Create appropriate Primary Key and Foreign Key constraints for referential integrity All primary and foreign keys are inserted while creating the table.

5. Insert at least 10 sample records into each of the following tables. a. Customers b. Products

c. Orders

d. OrderDetails

e. Inventory

```
INSERT INTO Customers (FirstName, LastName, Email, Phone, Address)
VALUES
    ('John', 'Doe', 'john.doe@example.com', '123-456-7890', '123 Main St, Anytown,
USA'),
    ('Jane', 'Smith', 'jane.smith@example.com', '456-789-0123', '456 Elm St, Anycity,
USA'),
    ('Michael', 'Johnson', 'michael.johnson@example.com', '789-012-3456', '789 Oak St,
Anystate, USA'),
    ('Emily', 'Brown', 'emily.brown@example.com', '321-654-9870', '321 Maple St,
Anyvillage, USA'),
    ('Daniel', 'Martinez', 'daniel.martinez@example.com', '654-987-0123', '654 Cedar St,
Anysuburb, USA'),
    ('Sarah', 'Wilson', 'sarah.wilson@example.com', '987-012-3456', '987 Pine St,
Anyhamlet, USA'),
    ('David', 'Taylor', 'david.taylor@example.com', '111-222-3333', '111 Oak St,
Anycity, USA'),
    ('Olivia', 'Anderson', 'olivia.anderson@example.com', '444-555-6666', '444 Elm St,
Anystate, USA'),
    ('James', 'Hernandez', 'james.hernandez@example.com', '777-888-9999', '777 Maple St,
Anytown, USA'),
    ('Emma', 'Garcia', 'emma.garcia@example.com', '000-111-2222', '000 Cedar St,
Anyvillage, USA'),
```

```
    ('Ava', 'Lopez', 'ava.lopez@example.com', '333-444-5555', '333 Pine St, Anysuburb,
USA'),
    ('Alexander', 'Martinez', 'alexander.martinez@example.com', '666-777-8888', '666 Oak
St, Anyhamlet, USA'),
    ('Sophia', 'Gonzalez', 'sophia.gonzalez@example.com', '999-000-1111', '999 Elm St,
Anytown, USA'),
    ('Mia', 'Perez', 'mia.perez@example.com', '222-333-4444', '222 Maple St, Anycity,
USA'),
    ('Logan', 'Rodriguez', 'logan.rodriguez@example.com', '555-666-7777', '555 Cedar St,
Anystate, USA');



       -- Insert 15 sample records into Products table
INSERT INTO Products (ProductID, ProductName, Description, Price)
VALUES
    (1, 'Smartphone', 'Smartphone with high-resolution camera', 599.99),
    (2, 'Laptop', 'Thin and lightweight laptop with SSD storage', 999.99),
(3, 'Headphones', 'Wireless noise-canceling headphones', 199.99),
    (4, 'Smart Watch', 'Fitness tracker with heart rate monitor', 149.99),
    (5, 'Tablet', '10-inch tablet with retina display', 399.99),
    (6, 'Digital Camera', 'Mirrorless digital camera with 4K video recording', 799.99),
    (7, 'Gaming Console', 'Next-gen gaming console with VR support', 499.99),
    (8, 'Bluetooth Speaker', 'Portable Bluetooth speaker with long battery life',
79.99),
    (9, 'External Hard Drive', '1TB external hard drive with USB 3.0', 69.99),
    (10, 'Wireless Router', 'Dual-band wireless router for high-speed internet',
129.99),
    (11, 'Fitness Tracker', 'Waterproof fitness tracker with GPS', 129.99),
    (12, 'Smart Home Hub', 'Voice-controlled smart home hub', 149.99),
    (13, 'Wireless Earbuds', 'True wireless earbuds with touch controls', 129.99),
    (14, 'Monitor', '27-inch 4K monitor with IPS display', 399.99),
    (15, 'Printer', 'All-in-one printer with wireless connectivity', 199.99);
       INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount) VALUES
    (1, 1, '2024-02-01', 249.99),
    (2, 3, '2024-02-03', 799.99),
(3, 5, '2024-02-05', 149.99),     (4,
2, '2024-02-07', 399.99),
    (5, 4, '2024-02-10', 999.99),
(6, 6, '2024-02-12', 79.99),
    (7, 8, '2024-02-15', 129.99),
    (8, 10, '2024-02-18', 499.99),
(9, 12, '2024-02-20', 129.99),
(10, 14, '2024-02-22', 69.99),
    (11, 7, '2024-02-25', 149.99),     (12,
9, '2024-02-28', 129.99),
    (13, 11, '2024-03-01', 399.99),
    (14, 13, '2024-03-03', 199.99),
    (15, 15, '2024-03-05', 999.99);
       INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
VALUES
       (16, 1, '2024-03-08', 329.99),
    (17, 3, '2024-03-10', 699.99),
    (18, 5, '2024-03-12', 199.99),
    (19, 2, '2024-03-15', 449.99),
    (20, 4, '2024-03-18', 1299.99),
    (21, 6, '2024-03-20', 89.99),
```

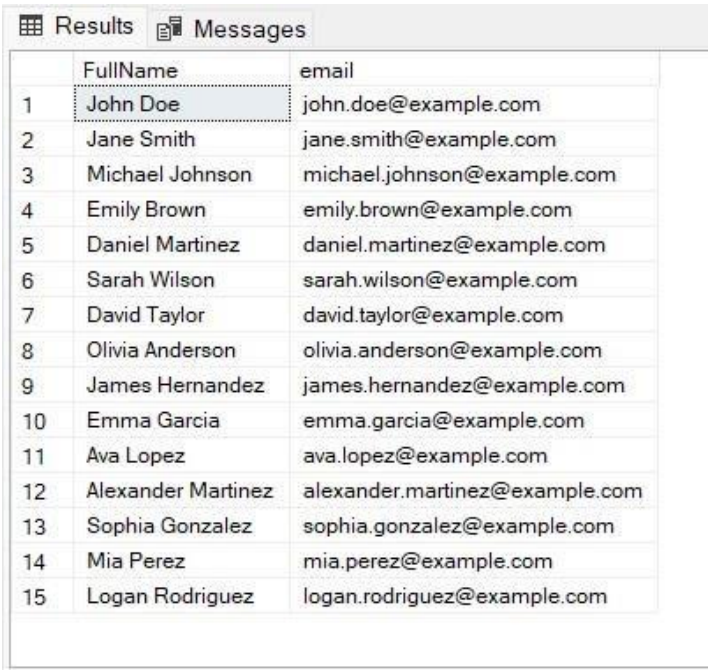
```
    (22, 8, '2024-03-22', 149.99),
    (23, 10, '2024-03-25', 599.99),
    (24, 12, '2024-03-28', 179.99),
    (25, 14, '2024-03-30', 79.99),
    (26, 7, '2024-04-01', 199.99),
    (27, 9, '2024-04-03', 169.99),
    (28, 11, '2024-04-05', 499.99),     (29,
13, '2024-04-08', 249.99),
    (30, 15, '2024-04-10', 1199.99),
    (31, 1, '2024-04-12', 359.99),     (32,
3, '2024-04-15', 799.99),
    (33, 5, '2024-04-18', 249.99),     (34,
2, '2024-04-20', 499.99),
    (35, 4, '2024-04-22', 1399.99);

-- Insert 35 sample records into OrderDetails table
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity) VALUES
(1, 1, 3, 2), (2,
2, 7, 3),
(3, 3, 2, 1),
(4, 4, 10, 5),
(5, 5, 14, 2), (6,
6, 5, 3),
(7, 7, 15, 4),
(8, 8, 8, 1), (9,
9, 4, 3),
(10, 10, 1, 2),
(11, 11, 13, 1),
(12, 12, 9, 2), (13,
13, 6, 3),
(14, 14, 11, 4),
(15, 15, 15, 1),
(16, 16, 3, 2), (17,
17, 7, 3),
(18, 18, 2, 1),
(19, 19, 10, 5),
(20, 20, 14, 2),  (21,
21, 5, 3),
(22, 22, 15, 4),
(23, 23, 8, 1),
(24, 24, 4, 3), (25,
25, 1, 2),
(26, 26, 13, 1),
(27, 27, 9, 2), (28,
28, 6, 3),
(29, 29, 11, 4),
(30, 30, 15, 1),
(31, 31, 3, 2), (32,
32, 7, 3),
(33, 33, 2, 1),
(34, 34, 10, 5),
(35, 35, 14, 2);

       -- Insert 15 sample records into Inventory table
INSERT INTO Inventory (InventoryID, ProductID, QuantityInStock, LastStockUpdate)  VALUES
```

```
    (1, 1, 50, '2024-02-01'),
    (2, 2, 30, '2024-02-01'),
    (3, 3, 80, '2024-02-01'),
(4, 4, 20, '2024-02-01'),     (5,
5, 60, '2024-02-01'),
    (6, 6, 40, '2024-02-01'),
(7, 7, 25, '2024-02-01'),     (8,
8, 70, '2024-02-01'),
    (9, 9, 55, '2024-02-01'),
    (10, 10, 45, '2024-02-01'),
    (11, 11, 35, '2024-02-01'),
(12, 12, 65, '2024-02-01'),     (13,
13, 75, '2024-02-01'),
    (14, 14, 15, '2024-02-01'),
    (15, 15, 10, '2024-02-01');


       --displaying the data

       select * from customers;
select * from products;   select *
from orders;         select * from
orderdetails;        select * from
inventory;
```

Task 2:

Select, Where, Between, AND, LIKE:

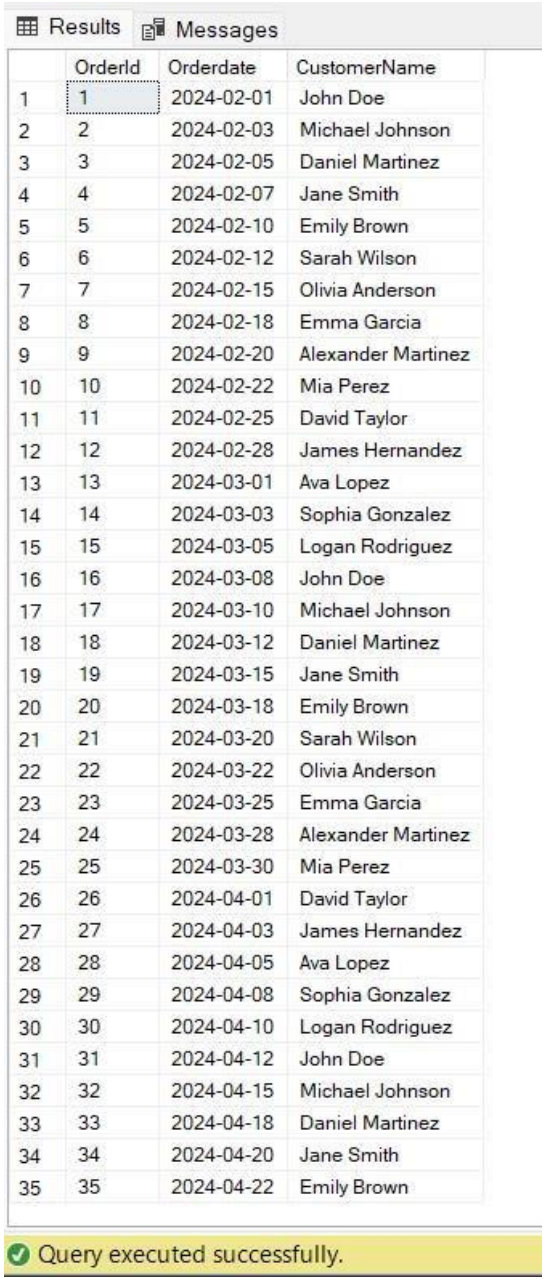
```
/*1.Write an SQL query to retrieve the names and emails of all customers.
*/   select concat(FirstName,' ',LastName) as FullName ,email from
Customers;  OUTPUT:
```



Results | Messages

	FullName	email
1	John Doe	john.doe@example.com
2	Jane Smith	jane.smith@example.com
3	Michael Johnson	michael.johnson@example.com
4	Emily Brown	emily.brown@example.com
5	Daniel Martinez	daniel.martinez@example.com
6	Sarah Wilson	sarah.wilson@example.com
7	David Taylor	david.taylor@example.com
8	Olivia Anderson	olivia.anderson@example.com
9	James Hernandez	james.hernandez@example.com
10	Emma Garcia	emma.garcia@example.com
11	Ava Lopez	ava.lopez@example.com
12	Alexander Martinez	alexander.martinez@example.com
13	Sophia Gonzalez	sophia.gonzalez@example.com
14	Mia Perez	mia.perez@example.com
15	Logan Rodriguez	logan.rodriguez@example.com

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

select OrderId,Orderdate,CONCAT(FirstName,' ',LastName) as CustomerName from
Customers,Orders where orders.CustomerID=customers.customerID;    OUTPUT:
```



Results | Messages

	OrderId	Orderdate	CustomerName
1	1	2024-02-01	John Doe
2	2	2024-02-03	Michael Johnson
3	3	2024-02-05	Daniel Martinez
4	4	2024-02-07	Jane Smith
5	5	2024-02-10	Emily Brown
6	6	2024-02-12	Sarah Wilson
7	7	2024-02-15	Olivia Anderson
8	8	2024-02-18	Emma Garcia
9	9	2024-02-20	Alexander Martinez
10	10	2024-02-22	Mia Perez
11	11	2024-02-25	David Taylor
12	12	2024-02-28	James Hernandez
13	13	2024-03-01	Ava Lopez
14	14	2024-03-03	Sophia Gonzalez
15	15	2024-03-05	Logan Rodriguez
16	16	2024-03-08	John Doe
17	17	2024-03-10	Michael Johnson
18	18	2024-03-12	Daniel Martinez
19	19	2024-03-15	Jane Smith
20	20	2024-03-18	Emily Brown
21	21	2024-03-20	Sarah Wilson
22	22	2024-03-22	Olivia Anderson
23	23	2024-03-25	Emma Garcia
24	24	2024-03-28	Alexander Martinez
25	25	2024-03-30	Mia Perez
26	26	2024-04-01	David Taylor
27	27	2024-04-03	James Hernandez
28	28	2024-04-05	Ava Lopez
29	29	2024-04-08	Sophia Gonzalez
30	30	2024-04-10	Logan Rodriguez
31	31	2024-04-12	John Doe
32	32	2024-04-15	Michael Johnson
33	33	2024-04-18	Daniel Martinez
34	34	2024-04-20	Jane Smith
35	35	2024-04-22	Emily Brown

Query executed successfully.

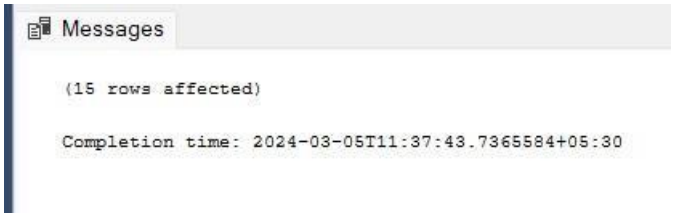
```sql
/*3.Write an SQL query to insert a new customer record into the "Customers" table.
Include customer information such as name, email, and address*/

insert into Customers (FirstName,LastName,email,Address)
values('Usha','Kiranmai','ushakiranmai2002@example.com', '123 Main Street'); select *
from Customers;
```

OUTPUT:

10	10	Emma	Garcia	emma.garcia@example.com	000-111-2222	000 Cedar St, Anyvillage, USA
11	11	Ava	Lopez	ava.lopez@example.com	333-444-5555	333 Pine St, Anysuburb, USA
12	12	Alexander	Martinez	alexander.martinez@example.com	666-777-8888	666 Oak St, Anyhamlet, USA
13	13	Sophia	Gonzalez	sophia.gonzalez@example.com	999-000-1111	999 Elm St, Anytown, USA
14	14	Mia	Perez	mia.perez@example.com	222-333-4444	222 Maple St, Anycity, USA
15	15	Logan	Rodriguez	logan.rodriguez@example.com	555-666-7777	555 Cedar St, Anystate, USA
16	16	Usha	Kiranmai	ushakiranmai2002@example.com	NULL	123 Main Street

```
/*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*1.1;
```

OUTPUT:




```
/*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.*/
CREATE PROCEDURE DeleteOrderByOrderID
        @OrderID INT
AS
BEGIN
        DELETE FROM OrderDetails
        WHERE OrderID = @OrderID
DELETE FROM Orders
        WHERE OrderID = @OrderID
END;
EXEC DeleteOrderByOrderID @OrderID = 15;
```

OUTPUT:

```
(1 row affected)

(1 row affected)

Completion time: 2024-03-05T11:41:54.9704112+05:30
```

```
/*6.Write an SQL query to insert a new order into the "Orders" table. Include the
customer ID,
order date, and any other necessary information.*/

insert into Orders values(36,11,'2024-03-24', 1111.24000); insert
into orderdetails values(36,36,5,3);
 select * from
orders;
```

OUTPUT:

26	27	9	2024-04-03	169.99000
27	28	11	2024-04-05	499.99000
28	29	13	2024-04-08	249.99000
29	30	15	2024-04-10	1199.99000
30	31	1	2024-04-12	359.99000
31	32	3	2024-04-15	799.99000
32	33	5	2024-04-18	249.99000
33	34	2	2024-04-20	499.99000
34	35	4	2024-04-22	1399.99000
35	36	11	2024-03-24	1111.24000

```
/*7. Write an SQL query to update the contact information (e.g., email and address) of
a specific
customer in the "Customers" table. Allow users to input the customer ID and new
contact information.*/
```

```
CREATE PROCEDURE UpdateInfo
@CustomerID INT, @NewEmail VARCHAR(50),
@NewAddress VARCHAR(100)
AS
BEGIN
	UPDATE Customers
	SET Email = @NewEmail
	WHERE CustomerID = @CustomerID
UPDATE Customers
	SET Address = @NewAddress
	WHERE CustomerID = @CustomerID
END;
EXEC UpdateInfo
	@CustomerID = 16,
	@NewEmail = 'ushakiranmai2002@gmail.com',
	@NewAddress = 'Vijayawada, AP';
```

OUTPUT:

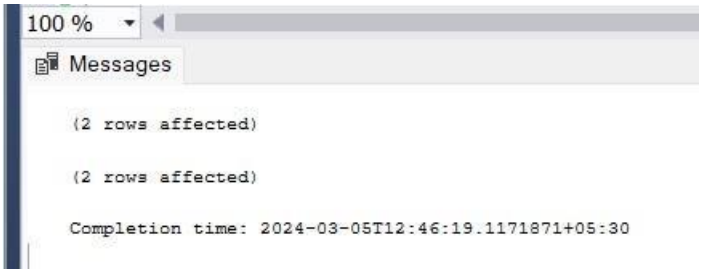

```
/*8. Write an SQL query to recalculate and update the total cost of each order in the
"Orders"
table based on the prices and quantities in the "OrderDetails" table.*/

UPDATE Orders
SET TotalAmount = (
    SELECT (Quantity * Products.Price)
    FROM OrderDetails
```

```
    JOIN Products ON OrderDetails.ProductID = Products.ProductID
WHERE OrderDetails.OrderID = Orders.OrderID); select * from
orders; OUTPUT:
```

Results | Messages

	OrderID	CustomerID	OrderDate	TotalAmount
1	1	1	2024-02-01	439.98000
2	2	3	2024-02-03	1649.97000
3	3	5	2024-02-05	1099.99000
4	4	2	2024-02-07	714.95000
5	5	4	2024-02-10	879.98000
6	6	6	2024-02-12	1319.97000
7	7	8	2024-02-15	879.96000
8	8	10	2024-02-18	87.99000
9	9	12	2024-02-20	494.97000
10	10	14	2024-02-22	1319.98000
11	11	7	2024-02-25	142.99000
12	12	9	2024-02-28	153.98000
13	13	11	2024-03-01	2639.97000
14	14	13	2024-03-03	571.96000
15	16	1	2024-03-08	439.98000
16	17	3	2024-03-10	1649.97000
17	18	5	2024-03-12	1099.99000
18	19	2	2024-03-15	714.95000
19	20	4	2024-03-18	879.98000
20	21	6	2024-03-20	1319.97000
21	22	8	2024-03-22	879.96000
22	23	10	2024-03-25	87.99000
23	24	12	2024-03-28	494.97000
24	25	14	2024-03-30	1319.98000
25	26	7	2024-04-01	142.99000
26	27	9	2024-04-03	153.98000
27	28	11	2024-04-05	2639.97000
28	29	13	2024-04-08	571.96000
29	30	15	2024-04-10	219.99000
30	31	1	2024-04-12	439.98000
31	32	3	2024-04-15	1649.97000
32	33	5	2024-04-18	1099.99000
33	34	2	2024-04-20	714.95000
34	35	4	2024-04-22	879.98000
35	36	11	2024-03-24	1319.97000

```
/*9. Write an SQL query to delete all orders and their associated order details for a
specific
customer from the "Orders" and "OrderDetails" tables. Allow users to input the
customer ID as a parameter.*/

CREATE PROCEDURE DeleteOrders
@CusID INT
AS
BEGIN
        DELETE FROM OrderDetails
        WHERE OrderID IN(
                SELECT OrderID FROM Orders
                WHERE CustomerID = @CusID);
        DELETE FROM Orders
        WHERE CustomerID = @CusID
END;
EXEC DeleteOrders @CusID = 14;   OUTPUT:
```




```
/*10. Write an SQL query to insert a new electronic gadget product into the "Products"
table,
including product name, category, price, and any other relevant details.*/

insert into Products values(16,'Keyboard','Gamming Keyboard of 7 colors,wired,Circle
Brand',1500.50); OUTPUT:
```

```
(1 row affected)
```

```
/*11. Write an SQL query to update the status of a specific order in the "Orders" table
(e.g., from
"Pending" to "Shipped"). Allow users to input the order ID and the new status.*/
ALTER TABLE Orders
ADD Status VARCHAR(15);

UPDATE Orders
SET Status='Pending';

DECLARE @OID INT = 1
UPDATE Orders
SET  Status='Shipped'
WHERE OrderID = @OID;
select * from orders;
OUTPUT:
```

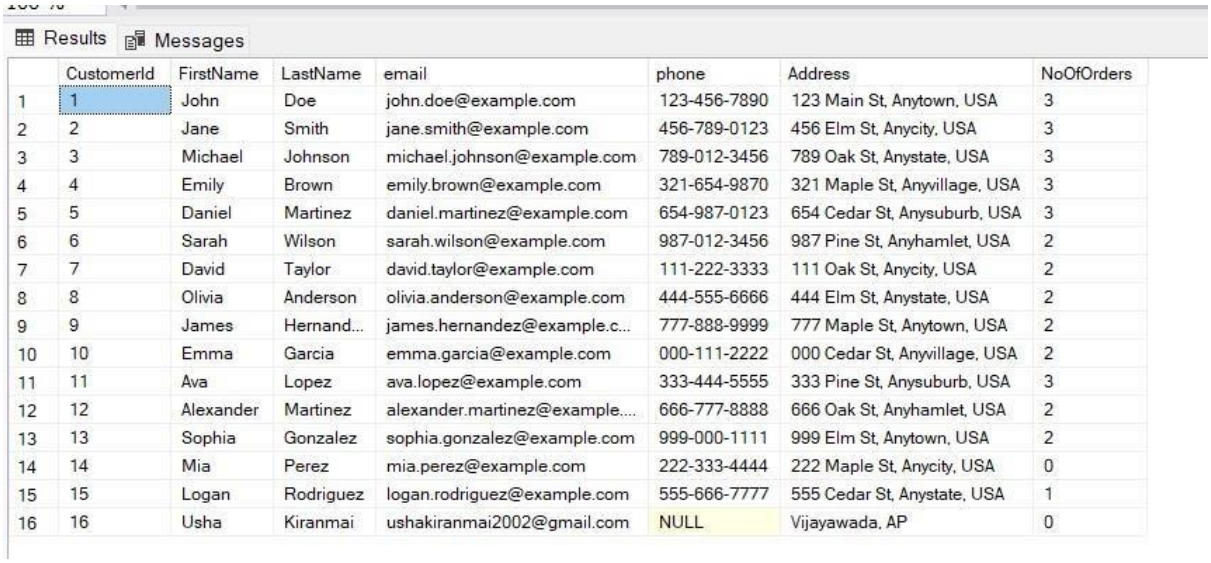
Results Messages

	OrderID	CustomerID	OrderDate	TotalAmount	Status
1	1	1	2024-02-01	439.98000	Shipped
2	2	3	2024-02-03	1649.97000	Pending
3	3	5	2024-02-05	1099.99000	Pending
4	4	2	2024-02-07	714.95000	Pending
5	5	4	2024-02-10	879.98000	Pending
6	6	6	2024-02-12	1319.97000	Pending
7	7	8	2024-02-15	879.96000	Pending
8	8	10	2024-02-18	87.99000	Pending
9	9	12	2024-02-20	494.97000	Pending
10	11	7	2024-02-25	142.99000	Pending

```
/*12. Write an SQL query to calculate and update the number of orders placed by each
customer
in the "Customers" table based on the data in the "Orders" table.*/
alter table
```

```
Customers add NoOfOrders int; update Customers set
NoOfOrders=( select count(ORderID) from Orders  where
ORders.CustomerID=Customers.CustomerId); select *
from Customers;
```

OUTPUT:



Results | Messages

	CustomerId	FirstName	LastName	email	phone	Address	NoOfOrders
1	1	John	Doe	john.doe@example.com	123-456-7890	123 Main St, Anytown, USA	3
2	2	Jane	Smith	jane.smith@example.com	456-789-0123	456 Elm St, Anycity, USA	3
3	3	Michael	Johnson	michael.johnson@example.com	789-012-3456	789 Oak St, Anystate, USA	3
4	4	Emily	Brown	emily.brown@example.com	321-654-9870	321 Maple St, Anyvillage, USA	3
5	5	Daniel	Martinez	daniel.martinez@example.com	654-987-0123	654 Cedar St, Anysuburb, USA	3
6	6	Sarah	Wilson	sarah.wilson@example.com	987-012-3456	987 Pine St, Anyhamlet, USA	2
7	7	David	Taylor	david.taylor@example.com	111-222-3333	111 Oak St, Anycity, USA	2
8	8	Olivia	Anderson	olivia.anderson@example.com	444-555-6666	444 Elm St, Anystate, USA	2
9	9	James	Hernand...	james.hernandez@example.c...	777-888-9999	777 Maple St, Anytown, USA	2
10	10	Emma	Garcia	emma.garcia@example.com	000-111-2222	000 Cedar St, Anyvillage, USA	2
11	11	Ava	Lopez	ava.lopez@example.com	333-444-5555	333 Pine St, Anysuburb, USA	3
12	12	Alexander	Martinez	alexander.martinez@example....	666-777-8888	666 Oak St, Anyhamlet, USA	2
13	13	Sophia	Gonzalez	sophia.gonzalez@example.com	999-000-1111	999 Elm St, Anytown, USA	2
14	14	Mia	Perez	mia.perez@example.com	222-333-4444	222 Maple St, Anycity, USA	0
15	15	Logan	Rodriguez	logan.rodriguez@example.com	555-666-7777	555 Cedar St, Anystate, USA	1
16	16	Usha	Kiranmai	ushakiranmai2002@gmail.com	NULL	Vijayawada, AP	0

Task 3. Aggregate functions, Having, Order By, GroupBy and Joins:

```
/*1. Write an SQL query to retrieve a list of all orders along with customer
information
(e.g., customer name) for each
order.*/   select
o.OrderId,
          o.Orderdate,          concat(c.firstName,'
',c.lastName) as CustomerName,          c.email
 from Orders o,Customers c  where
o.CustomerID=c.CustomerId;
```

OUTPUT:

100 %

Results | Messages

	OrderId	Orderdate	CustomerName	email
1	1	2024-02-01	John Doe	john.doe@example.com
2	2	2024-02-03	Michael Johnson	michael.johnson@example.com
3	3	2024-02-05	Daniel Martinez	daniel.martinez@example.com
4	4	2024-02-07	Jane Smith	jane.smith@example.com
5	5	2024-02-10	Emily Brown	emily.brown@example.com
6	6	2024-02-12	Sarah Wilson	sarah.wilson@example.com
7	7	2024-02-15	Olivia Anderson	olivia.anderson@example.com
8	8	2024-02-18	Emma Garcia	emma.garcia@example.com
9	9	2024-02-20	Alexander Martinez	alexander.martinez@example.com
10	11	2024-02-25	David Taylor	david.taylor@example.com
11	12	2024-02-28	James Hernandez	james.hernandez@example.com
12	13	2024-03-01	Ava Lopez	ava.lopez@example.com
13	14	2024-03-03	Sophia Gonzalez	sophia.gonzalez@example.com
14	16	2024-03-08	John Doe	john.doe@example.com
15	17	2024-03-10	Michael Johnson	michael.johnson@example.com
16	18	2024-03-12	Daniel Martinez	daniel.martinez@example.com
17	19	2024-03-15	Jane Smith	jane.smith@example.com
18	20	2024-03-18	Emily Brown	emily.brown@example.com
19	21	2024-03-20	Sarah Wilson	sarah.wilson@example.com
20	22	2024-03-22	Olivia Anderson	olivia.anderson@example.com
21	23	2024-03-25	Emma Garcia	emma.garcia@example.com
22	24	2024-03-28	Alexander Martinez	alexander.martinez@example.com
23	26	2024-04-01	David Taylor	david.taylor@example.com
24	27	2024-04-03	James Hernandez	james.hernandez@example.com
25	28	2024-04-05	Ava Lopez	ava.lopez@example.com
26	29	2024-04-08	Sophia Gonzalez	sophia.gonzalez@example.com
27	30	2024-04-10	Logan Rodriguez	logan.rodriguez@example.com
28	31	2024-04-12	John Doe	john.doe@example.com
29	32	2024-04-15	Michael Johnson	michael.johnson@example.com
30	33	2024-04-18	Daniel Martinez	daniel.martinez@example.com
31	34	2024-04-20	Jane Smith	jane.smith@example.com
32	35	2024-04-22	Emily Brown	emily.brown@example.com
33	36	2024-03-24	Ava Lopez	ava.lopez@example.com

Query executed successfully.

```
/*2. Write an SQL query to find the total revenue generated by each electronic gadget
product.
Include the product name and the total revenue.*/

select p.ProductName, sum(p.price*o.Quantity) as TotalRevenue from Products p left
join OrderDetails o
on o.ProductID=p.ProductID group by productName order by TotalRevenue desc;
```

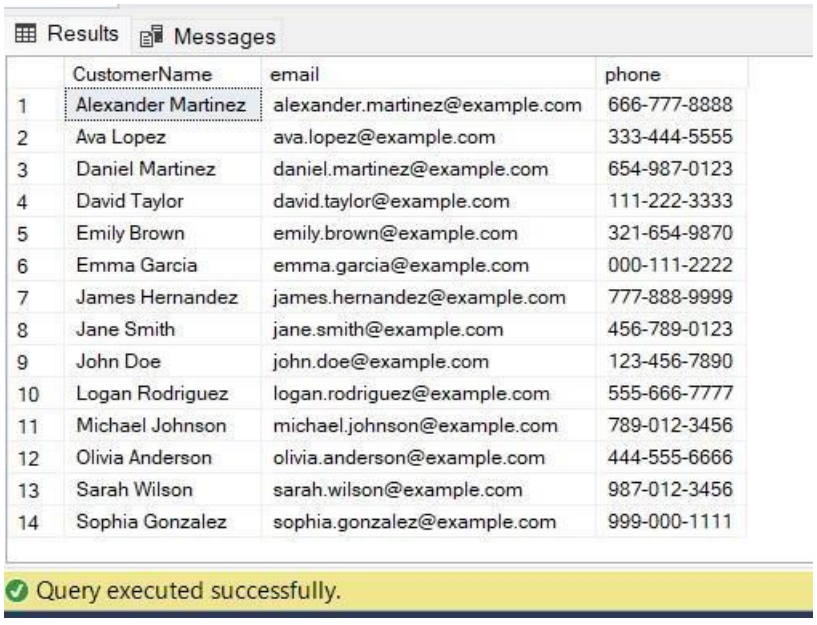
OUTPUT:

Results | Messages

	ProductName	TotalRevenue
1	Digital Camera	5279.94
2	Gaming Console	4949.91
3	Tablet	3959.91
4	Laptop	3299.97
5	Monitor	2639.94
6	Wireless Router	2144.85
7	Printer	1979.91
8	Headphones	1319.94
9	Fitness Tracker	1143.92
10	Smart Watch	989.94
11	External Hard Drive	307.96
12	Wireless Earbuds	285.98
13	Bluetooth Speaker	175.98
14	Smart Home Hub	NULL
15	Smartphone	NULL
16	Keyboard	NULL

```
/*3.Write an SQL query to list all customers who have made at least one purchase.
Include their
names and contact information*/

select concat(FirstName,' ',LastName)as CustomerName,email,phone from
Customers,Orders  where Customers.CustomerId=orders.CustomerID group
by FirstName,
        lastname,
email,          phone;
```

OUTPUT:



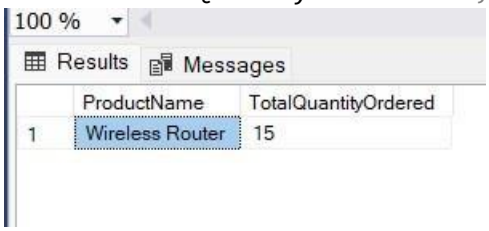
Results | Messages

	CustomerName	email	phone
1	Alexander Martinez	alexander.martinez@example.com	666-777-8888
2	Ava Lopez	ava.lopez@example.com	333-444-5555
3	Daniel Martinez	daniel.martinez@example.com	654-987-0123
4	David Taylor	david.taylor@example.com	111-222-3333
5	Emily Brown	emily.brown@example.com	321-654-9870
6	Emma Garcia	emma.garcia@example.com	000-111-2222
7	James Hernandez	james.hernandez@example.com	777-888-9999
8	Jane Smith	jane.smith@example.com	456-789-0123
9	John Doe	john.doe@example.com	123-456-7890
10	Logan Rodriguez	logan.rodriguez@example.com	555-666-7777
11	Michael Johnson	michael.johnson@example.com	789-012-3456
12	Olivia Anderson	olivia.anderson@example.com	444-555-6666
13	Sarah Wilson	sarah.wilson@example.com	987-012-3456
14	Sophia Gonzalez	sophia.gonzalez@example.com	999-000-1111

Query executed successfully.

```
/*4. Write an SQL query to find the most popular electronic gadget, which is the one
with the highest total quantity ordered. Include the product name and the total
quantity ordered*/

SELECT  top 1 p.ProductName, SUM(od.Quantity) AS TotalQuantityOrdered
FROM OrderDetails od
```

```
INNER JOIN Products p ON od.ProductID = p.ProductID
GROUP BY p.ProductName
ORDER BY TotalQuantityOrdered DESC; OUTPUT:
```




```
/*5.Write an SQL query to retrieve a list of electronic gadgets along with their
corresponding categories.*/

ALTER TABLE Products
ADD Category VARCHAR(20);
UPDATE Products
SET Category =
    CASE
        WHEN Description LIKE '%smartphone%' THEN 'Electronics'
WHEN Description LIKE '%laptop%' THEN 'Electronics'
        WHEN Description LIKE '%headphones%' THEN 'Electronics'
        WHEN Description LIKE '%smart watch%' THEN 'Electronics'
WHEN Description LIKE '%tablet%' THEN 'Electronics'
        WHEN Description LIKE '%digital camera%' THEN 'Electronics'
        WHEN Description LIKE '%gaming console%' THEN 'Electronics'
        WHEN Description LIKE '%bluetooth speaker%' THEN 'Electronics'
        WHEN Description LIKE '%external hard drive%' THEN 'Electronics'
        WHEN Description LIKE '%wireless router%' THEN 'Electronics'
        WHEN Description LIKE '%fitness tracker%' THEN 'Electronics'
        WHEN Description LIKE '%smart home hub%' THEN 'Electronics'
        WHEN Description LIKE '%wireless earbuds%' THEN 'Electronics'
        WHEN Description LIKE '%monitor%' THEN 'Electronics'
        WHEN Description LIKE '%printer%' THEN 'Electronics'
              WHEN Description LIKE '%scanner%' THEN 'Electronics'
              WHEN Description LIKE '%Keyboard%' THEN 'Electronics'
        ELSE 'Other'
    END;
SELECT ProductName, Category FROM Products OUTPUT:
```

Results Messages

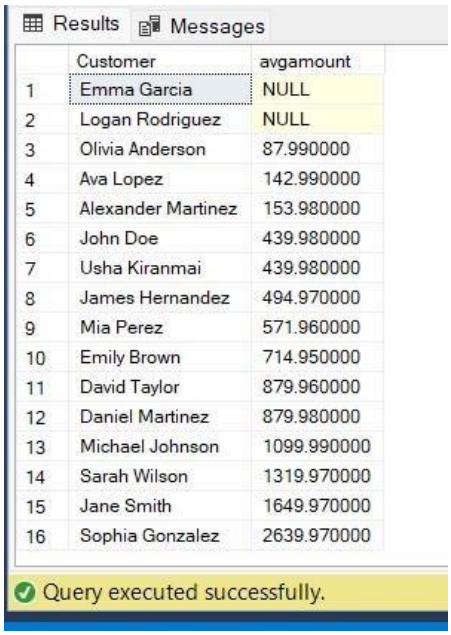
	ProductName	Category
2	Laptop	Electronics
3	Headphones	Electronics
4	Smart Watch	Electronics
5	Tablet	Electronics
6	Digital Camera	Electronics
7	Gaming Console	Electronics
8	Bluetooth Speaker	Electronics
9	External Hard Dri...	Electronics
10	Wireless Router	Electronics
11	Fitness Tracker	Electronics
12	Smart Home Hub	Electronics
13	Wireless Earbuds	Electronics
14	Monitor	Electronics
15	Printer	Electronics
16	Keyboard	Electronics

Query executed successfully.

```
/*6.Write an SQL query to calculate the average order value for each customer. Include
the
customer's name and their average order value.*/
```

```
select concat(FirstName,' ',LastNAme) as Customer,avg(totalAmount)as avgamount from
Customers left join Orders on Customers.CustomerId=Orders.OrderID group by
Orders.CustomerID,Customers.FirstName,Customers.LastName order
by avgamount ;
```

OUTPUT:



Results Messages

	Customer	avgamount
1	Emma Garcia	NULL
2	Logan Rodriguez	NULL
3	Olivia Anderson	87.990000
4	Ava Lopez	142.990000
5	Alexander Martinez	153.980000
6	John Doe	439.980000
7	Usha Kiranmai	439.980000
8	James Hernandez	494.970000
9	Mia Perez	571.960000
10	Emily Brown	714.950000
11	David Taylor	879.960000
12	Daniel Martinez	879.980000
13	Michael Johnson	1099.990000
14	Sarah Wilson	1319.970000
15	Jane Smith	1649.970000
16	Sophia Gonzalez	2639.970000

Query executed successfully.

```
/*7. Write an SQL query to find the order with the highest total revenue. Include the
order ID,
customer information, and the total revenue.*/
```

```
select TotalAmount,concat(FirstName,' ',Lastname) as CustomerName from Orders join Customers on Orders.CustomerID=Customers.CustomerId where totalAmount=(Select max(totalAmount) from Orders); OUTPUT:
```

Results Messages

	TotalAmount	CustomerName
1	2639.97000	Ava Lopez
2	2639.97000	Ava Lopez

```
/*8. Write an SQL query to list electronic gadgets and the number of times each product has been ordered.*/
select products.ProductID,Products.ProductName ,count(Quantity)as NumberOfTimesOrdered
from OrderDetails right join Products
on OrderDetails.ProductID=Products.ProductID group
by products.ProductID,products.ProductName;
```

OUTPUT:

Results Messages

	ProductID	ProductName	NumberOfTimesOrdered
1	1	Smartphone	0
2	2	Laptop	3
3	3	Headphones	3
4	4	Smart Watch	2
5	5	Tablet	3
6	6	Digital Camera	2
7	7	Gaming Console	3
8	8	Bluetooth Speaker	2
9	9	External Hard Drive	2
10	10	Wireless Router	3
11	11	Fitness Tracker	2
12	12	Smart Home Hub	0
13	13	Wireless Earbuds	2
14	14	Monitor	3
15	15	Printer	3

Query executed successfully.

```
/*9. Write an SQL query to find customers who have purchased a specific electronic gadget product.
Allow users to input the product name as a parameter.*/
```

```sql
CREATE PROCEDURE FindCustomer
	@ProductName VARCHAR(20)
AS
BEGIN
	SELECT C.FirstName+' '+C.LastName AS CustomerName,
		P.ProductName
	FROM Customers AS C
	JOIN Orders AS O ON O.CustomerID = C.CustomerID
	JOIN OrderDetails AS OD ON OD.OrderID = O.OrderID
	JOIN Products AS P ON P.ProductID = OD.ProductID
	WHERE P.ProductName = @ProductName
END;
EXEC FindCustomer @ProductName ='Laptop'; OUTPUT:
```

Results Messages

	CustomerName	ProductName
1	Daniel Martinez	Laptop
2	Daniel Martinez	Laptop
3	Daniel Martinez	Laptop

```sql
/*10. Write an SQL query to calculate the total revenue generated by all orders placed
within a
specific time period. Allow users to input the start and end dates as parameters.*/


    create
procedure
CALTOTREV
	@stdate date,
	@enddate date
AS begin    select sum(TotalAmount) as Total from Orders
where OrderDate between @stdate and @enddate; end; exec
CALTOTREV @stdate='2024-02-19',@enddate='2024-02-23'; OUTPUT:
```

Results Messages

	Total
1	494.97000

TASK-4

```sql
/*1. Write an SQL query to find out which customers have not placed any orders.*/
SELECT CONCAT(FIRSTNAME,' ',LASTNAME) AS CUSTOMER FROM CUSTOMERS WHERE CUSTOMERID NOT
IN  (SELECT CUSTOMERID FROM Orders); OUTPUT:
```

Results Messages

	CUSTOMER
1	Mia Perez
2	Usha Kiranmai

```sql
/*2. Write an SQL query to find the total number of products available for sale. */
select
coalesce(ProductName,'TotalProducts')as ProductName , sum(QuantityInStock) as
QuantityAvailable        from        products,Inventory                where
products.productID=Inventory.productID  group
by rollup (ProductName);
(OR)
select sum(QuantityInStock) from inventory;
```

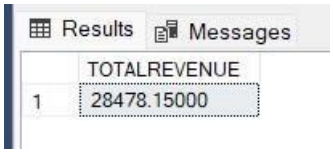
OUTPUT:

Results Messages

	ProductName	QuantityAvailable
1	Bluetooth Speaker	70
2	Digital Camera	40
3	External Hard Drive	55
4	Fitness Tracker	35
5	Gaming Console	25
6	Headphones	80
7	Laptop	30
8	Monitor	15
9	Printer	10
10	Smart Home Hub	65
11	Smart Watch	20
12	Smartphone	50
13	Tablet	60
14	Wireless Earbuds	75
15	Wireless Router	45
16	TotalProducts	675

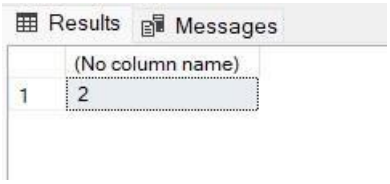
Query executed successfully.

```sql
/*3. Write an SQL query to calculate the total revenue generated by TechShop. */ select
sum(TotalAmount) AS TOTALREVENUE from Orders;    OUTPUT:
```



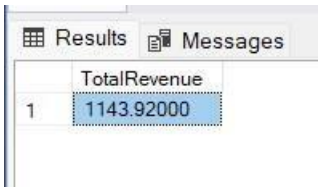

```
/*4. Write an SQL query to calculate the average quantity ordered for products in a
specific category.
Allow users to input the category name as a parameter.*/
DECLARE @CATGORYNAME VARCHAR(20) = 'ELECTRONICS';
SELECT AVG(QUANTITY) FROM ORDERDETAILS WHERE PRODUCTID IN
(SELECT PRODUCTID FROM Products WHERE Category=@CATGORYNAME);

OR

SELECT AVG(QUANTITY) FROM ORDERDETAILS;
```

OUTPUT:




```
/*5. Write an SQL query to calculate the total revenue generated by a specific customer.
Allow users  to input the customer ID as a parameter*/

DECLARE @CustomerID INT = 13
SELECT
	SUM(TotalAmount) TotalRevenue
FROM orders where CustomerID=@CustomerID;
```

OUTPUT:




```
*6. Write an SQL query to find the customers who have placed the most orders. List
their names
and the number of orders they've placed.*/


WITH OrderCount AS(
	SELECT
		C.CustomerID,
		C.FirstName,
```

```
            C.LastName,
            COUNT(O.OrderID) NoOfOrders
        FROM Customers C
        JOIN Orders O ON O.CustomerID = C.CustomerID
        GROUP BY C.CustomerID, C.FirstName, C.LastName )
SELECT FirstName, LastName, NoOfOrders FROM OrderCount
WHERE NoOfOrders = (SELECT MAX(NoOfOrders) FROM OrderCount);
```

OUTPUT:

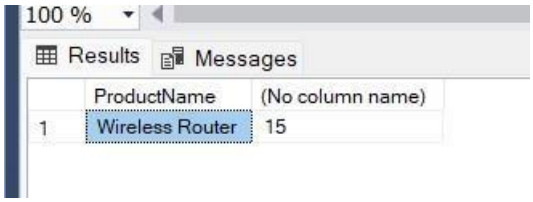
Results | Messages

	FirstName	LastName	NoOfOrders
1	John	Doe	3
2	Jane	Smith	3
3	Michael	Johnson	3
4	Emily	Brown	3
5	Daniel	Martinez	3
6	Ava	Lopez	3

Query executed successfully.

```
/*7. Write an SQL query to find the most popular product category, which is the one
with the highest
total quantity ordered across all orders*/

select top 1 ProductName,sum(Quantity) from orderdetails,products
where products.productid=orderdetails.ProductID group by productName
order by sum(Quantity) desc;
```

OUTPUT:



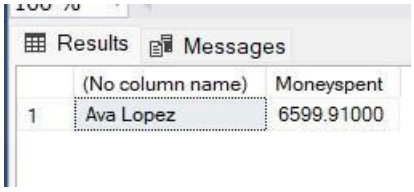
100 %

Results | Messages

	ProductName	(No column name)
1	Wireless Router	15

```
/*8. Write an SQL query to find the customer who has spent the most money (highest
total revenue)
on electronic gadgets. List their name and total spending*/

select top 1concat(c.Firstname,' ',c.lastname),sum(o.TotalAmount) as Moneyspent  from
customers c join orders o on c.CustomerId=o.CustomerId join orderdetails od on
od.orderId=o.orderID join products p on p.ProductID=od.ProductID group by
c.firstname,c.lastname order by Moneyspent desc;
```

OUTPUT:



	(No column name)	Moneyspent
1	Ava Lopez	6599.91000

/*9. Write an SQL query to calculate the average order value (total revenue divided by the number of orders) for all customers.*/

```
WITH OrderValue AS(
SELECT
	O.CustomerID,
	C.FirstName+' '+C.LastName AS CustomerName,
	SUM(O.TotalAmount) AS TotalRevenue,
	COUNT(O.OrderID) AS NoOfOrders
FROM Orders O
 RIGHT JOIN Customers C ON C.CustomerID = O.CustomerID
GROUP BY O.CustomerID, C.FirstName, C.LastName)
SELECT
	OV.CustomerID,
	OV.CustomerName,
	AVG(OV.TotalRevenue/OV.NoOfOrders) AS AvgOrderValue
FROM OrderValue OV
GROUP BY OV.CustomerID, OV.CustomerName;   OUTPUT:
```

	CustomerID	CustomerName	AvgOrderValue
1	12	Alexander Martinez	494.970000
2	11	Ava Lopez	2199.970000
3	5	Daniel Martinez	1099.990000
4	7	David Taylor	142.990000
5	4	Emily Brown	879.980000
6	10	Emma Garcia	87.990000
7	9	James Hernandez	153.980000
8	2	Jane Smith	714.950000
9	1	John Doe	439.980000
10	15	Logan Rodriguez	219.990000
11	NULL	Mia Perez	NULL
12	3	Michael Johnson	1649.970000
13	8	Olivia Anderson	879.960000
14	6	Sarah Wilson	1319.970000
15	13	Sophia Gonzalez	571.960000
16	NULL	Usha Kiranmai	NULL

Query executed successfully.

/*10. Write an SQL query to find the total number of orders placed by each customer and list their
names along with the order count*/

```
select concat(c.firstname,' ',c.lastname) as customer,(COUNT(O.OrderID)) as
totalorders
from customers c left join Orders o on c.CustomerId=o.CustomerID
left join OrderDetails od on od.orderId=o.orderID left join products
p on p.ProductID=od.ProductID group by
c.firstname,c.lastname order by c.firstname,c.lastname ;
```

OUTPUT:

Results Messages

	customer	totalorders
1	Alexander Martinez	2
2	Ava Lopez	3
3	Daniel Martinez	3
4	David Taylor	2
5	Emily Brown	3
6	Emma Garcia	2
7	James Hernandez	2
8	Jane Smith	3
9	John Doe	3
10	Logan Rodriguez	1
11	Mia Perez	0
12	Michael Johnson	3
13	Olivia Anderson	2
14	Sarah Wilson	2
15	Sophia Gonzalez	2
16	Usha Kiranmai	0