ASSIGNMENT

ELECTRONIC GADGET SHOP

NAME: S. NIKHIL SAI

ASSIGNMENT: ELECTRONIC AGDGET SHOP

Task 1: Database Design

1. Create the database named "TechShop".

Ans.

```
Create database techshop;
```

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

Ans.

```
 /* Creating Tables*/
/*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(300),
Phone varchar(15),
Address varchar(100));
```

/*. Products:
• ProductID (Primary Key)
• ProductName
• Description
• Price */

```
create table Products(
ProductID int primary key,
ProductName varchar(30),
Description varchar(30),
Price decimal(10,2),
);
```

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

```
create table Orders(
OrderID int primary key,
CustomerID int,
OrderDate date,
TotalAmount decimal(10,2),
FOREIGN KEY(CustomerID) REFERENCES Customers(CustomerID)
ON DELETE CASCADE,
);
```

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

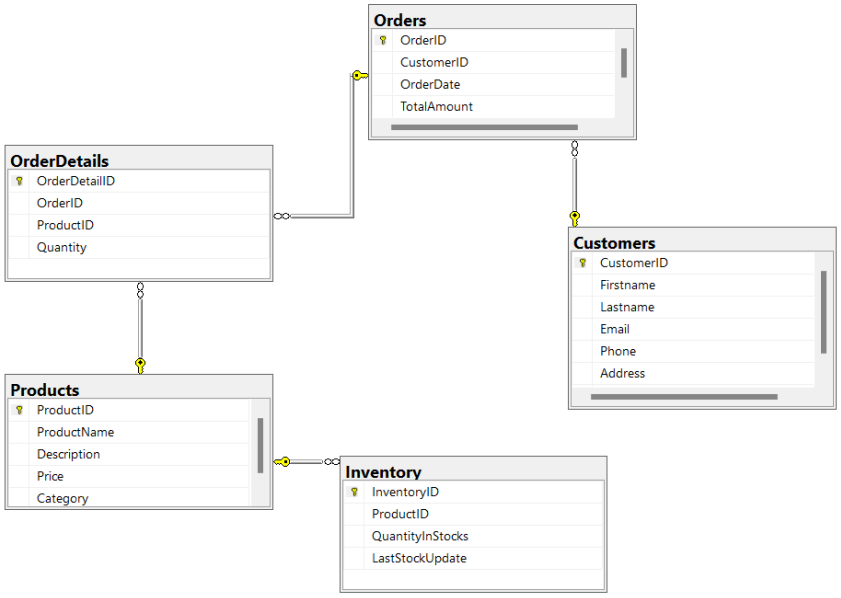
```
create table OrderDetails(
OrderDetailID int primary key,
OrderID int,
ProductID int,
Quantity int,
FOREIGN KEY(OrderID) REFERENCES Orders(OrderID) ON
DELETE CASCADE,
FOREIGN KEY(ProductID) REFERENCES Products(ProductID) ON
DELETE CASCADE,
);

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

create table Inventory(
InventoryID int primary key,
ProductID int,
QuantityInStocks int,
LastStockUpdate date,
FOREIGN KEY(ProductID) REFERENCES Products(ProductID) ON
DELETE CASCADE,
);
```

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

Ans.



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

Ans.

CustomerID int identity primary key,

ProductID int primary key,

OrderID int primary key,

FOREIGN KEY(CustomerID) REFERENCES Customers(CustomerID) ON DELETE CASCADE,

OrderDetailID int primary key,

FOREIGN KEY(OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE,

FOREIGN KEY(ProductID) REFERENCES Products(ProductID) ON DELETE CASCADE,

InventoryID int primary key,

FOREIGN KEY(ProductID) REFERENCES Products(ProductID) ON DELETE CASCADE,

5. Insert at least 10 sample records into each of the following tables.
   a. Customers
   b. Products
   c. Orders
   d. OrderDetails
   e. Inventory

Ans.

Customer Table:

INSERT INTO Customers (Firstname, Lastname, Email, Phone, Address)

VALUES

('Alice', 'Smith', 'alice.smith@example.com', '1234567890', '123 Maple Street, New York, USA'),

('Bob', 'Johnson', 'bob.johnson@example.com', '2345678901', '456 Oak Avenue, Los Angeles, USA'),

('Carol', 'Williams', 'carol.williams@example.com', '3456789012', '789 Pine Road, Chicago, USA'),

('David', 'Brown', 'david.brown@example.com', '4567890123', '101 Elm Street, Houston, USA'),

('Eve', 'Jones', 'eve.jones@example.com', '5678901234', '202 Cedar Lane, Phoenix, USA'),

('Frank', 'Garcia', 'frank.garcia@example.com', '6789012345', '303 Birch Boulevard, Philadelphia, USA'),

('Grace', 'Martinez', 'grace.martinez@example.com', '7890123456', '404 Spruce Drive, San Antonio, USA'),

('Hank', 'Davis', 'hank.davis@example.com', '8901234567', '505 Aspen Circle, Dallas, USA'),

('Ivy', 'Miller', 'ivy.miller@example.com', '9012345678', '606 Willow Street, San Jose, USA'),

('Jack', 'Wilson', 'jack.wilson@example.com', '0123456789', '707 Redwood Avenue, Austin, USA');

	CustomerID	Firstname	Lastname	Email	Phone	Address
1	1	Alice	Smith	alice.smith@example.com	1234567890	123 Maple Street, New York, USA
2	2	Bob	Johnson	bob.johnson@example.com	2345678901	456 Oak Avenue, Los Angeles, USA
3	3	Carol	Williams	update@gmail.com	9456208136	243 canada
4	4	David	Brown	david.brown@example.com	4567890123	101 Elm Street, Houston, USA
5	5	Eve	Jones	eve.jones@example.com	5678901234	202 Cedar Lane, Phoenix, USA
6	6	Frank	Garcia	frank.garcia@example.com	6789012345	303 Birch Boulevard, Philadelphia, USA
7	7	Grace	Martinez	grace.martinez@example.com	7890123456	404 Spruce Drive, San Antonio, USA
8	8	Hank	Davis	hank.davis@example.com	8901234567	505 Aspen Circle, Dallas, USA
9	9	Ivy	Miller	ivy.miller@example.com	9012345678	606 Willow Street, San Jose, USA
10	10	Jack	Wilson	jack.wilson@example.com	0123456789	707 Redwood Avenue, Austin, USA
11	11	Nikhil	Sai	nikhil@gmail.com	NULL	Tirupati, India

Products Table:

INSERT INTO Products (ProductID, ProductName, Description, Price)

VALUES

(1001, 'Laptop', '15-inch display', 799.99),

(1002, 'Smartphone', '64GB storage', 599.49),

(1003, 'Headphones', 'Noise-canceling', 199.99),

(1004, 'Smartwatch', 'Water-resistant', 249.89),

(1005, 'Tablet', '10-inch screen', 349.99),

(1006, 'Keyboard', 'Mechanical', 99.99),

(1007, 'Mouse', 'Wireless', 29.99),

(1008, 'Monitor', '4K resolution', 299.99),

(1009, 'Printer', 'Laser', 159.99),

(1010, 'External Hard Drive', '1TB storage', 79.99);

	ProductID	ProductName	Description	Price
1	1001	Laptop	15-inch display	879.99
2	1002	Smartphone	64GB storage	659.44
3	1003	Headphones	Noise-canceling	219.99
4	1004	Smartwatch	Water-resistant	274.88
5	1005	Tablet	10-inch screen	384.99
6	1006	Keyboard	Mechanical	109.99
7	1007	Mouse	Wireless	32.99
8	1008	Monitor	4K resolution	329.99
9	1009	Printer	Laser	175.99
10	1010	External Hard Drive	1TB storage	87.99
11	1011	Earpods	50hrs durability	750.00

Orders Table:

```
INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)
VALUES
    (2001, 1, '2024-09-01', 150.75),
    (2002, 2, '2024-09-03', 299.99),
    (2003, 3, '2024-09-05', 89.49),
    (2004, 4, '2024-09-07', 450.00),
    (2005, 5, '2024-09-10', 210.30),
    (2006, 6, '2024-09-12', 320.00),
    (2007, 7, '2024-09-14', 499.99),
    (2008, 8, '2024-09-16', 120.20),
    (2009, 9, '2024-09-18', 75.00),
    (2010, 10, '2024-09-20', 60.99);
```

	OrderID	CustomerID	OrderDate	TotalAmount
1	2001	1	2024-09-01	17599.80
2	2002	2	2024-09-03	11869.92
3	2003	3	2024-09-05	7699.65
4	2004	4	2024-09-07	3023.68
5	2006	6	2024-09-06	NULL
6	2007	7	2024-09-14	1715.48
7	2008	8	2024-09-16	14849.55
8	2009	9	2024-09-18	3519.80
9	2010	10	2024-09-20	2639.70

Order Details Table:

```
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID, Quantity)
VALUES
    (3001, 2001, 1001, 20),
    (3002, 2002, 1002, 18),
    (3003, 2003, 1003, 35),
    (3004, 2004, 1004, 11),
    (3005, 2005, 1005, 25),
    (3006, 2006, 1006, 40),
    (3007, 2007, 1007, 52),
    (3008, 2008, 1008, 45),
    (3009, 2009, 1009, 20),
    (3010, 2010, 1010, 30);
```

	OrderDetailID	OrderID	ProductID	Quantity
1	3001	2001	1001	20
2	3002	2002	1002	18
3	3003	2003	1003	35
4	3004	2004	1004	11
5	3007	2007	1007	52
6	3008	2008	1008	45
7	3009	2009	1009	20
8	3010	2010	1010	30

Inventory Table:

```
INSERT INTO Inventory (InventoryID, ProductID, QuantityInStocks,
LastStockUpdate)
VALUES
    (4001, 1001, 50, '2024-09-01'),
    (4002, 1002, 30, '2024-09-02'),
    (4003, 1003, 20, '2024-09-03'),
    (4004, 1004, 15, '2024-09-04'),
    (4005, 1005, 25, '2024-09-05'),
    (4006, 1006, 40, '2024-09-06'),
    (4007, 1007, 60, '2024-09-07'),
    (4008, 1008, 10, '2024-09-08'),
    (4009, 1009, 35, '2024-09-09'),
    (4010, 1010, 45, '2024-09-10');
```

	InventoryID	ProductID	QuantityInStocks	LastStockUpdate
1	4001	1001	50	2024-09-01
2	4002	1002	30	2024-09-02
3	4003	1003	20	2024-09-03
4	4004	1004	15	2024-09-04
5	4005	1005	25	2024-09-05
6	4006	1006	40	2024-09-06
7	4007	1007	60	2024-09-07
8	4008	1008	10	2024-09-08
9	4009	1009	35	2024-09-09
10	4010	1010	45	2024-09-10

Task 2: Select, Where, Between, AND , LIKE:

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

Ans.

```
select FirstName as Names, Email  from Customers;
```

OUTPUT:

	Names	Email
1	Alice	alice.smith@example.com
2	Bob	bob.johnson@example.com
3	Carol	update@gmail.com
4	David	david.brown@example.com
5	Eve	eve.jones@example.com
6	Frank	frank.garcia@example.com
7	Grace	grace.martinez@example.com
8	Hank	hank.davis@example.com
9	Ivy	ivy.miller@example.com
10	Jack	jack.wilson@example.com

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

Ans.

```
select Orders.OrderId, Orders.OrderDate, Customers.FirstName
from Orders
join Customers
on Customers.CustomerID=Orders.CustomerID;
```

OUTPUT:

	OrderId	OrderDate	FirstName
1	2001	2024-09-01	Alice
2	2002	2024-09-03	Bob
3	2003	2024-09-05	Carol
4	2004	2024-09-07	David
5	2006	2024-09-06	Frank
6	2007	2024-09-14	Grace
7	2008	2024-09-16	Hank
8	2009	2024-09-18	Ivy
9	2010	2024-09-20	Jack

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

Ans.

```
Insert into Customers values('Nikhil', 'Sai','nikhil@gmail.com',Null ,'Tirupati, India');
```

OUTPUT:

	CustomerID	Firstname	Lastname	Email	Phone	Address
1	1	Alice	Smith	alice.smith@example.com	1234567890	123 Maple Street, New York, USA
2	2	Bob	Johnson	bob.johnson@example.com	2345678901	456 Oak Avenue, Los Angeles, USA
3	3	Carol	Williams	update@gmail.com	9456208136	243 canada
4	4	David	Brown	david.brown@example.com	4567890123	101 Elm Street, Houston, USA
5	5	Eve	Jones	eve.jones@example.com	5678901234	202 Cedar Lane, Phoenix, USA
6	6	Frank	Garcia	frank.garcia@example.com	6789012345	303 Birch Boulevard, Philadelphia, USA
7	7	Grace	Martinez	grace.martinez@example.com	7890123456	404 Spruce Drive, San Antonio, USA
8	8	Hank	Davis	hank.davis@example.com	8901234567	505 Aspen Circle, Dallas, USA
9	9	Ivy	Miller	ivy.miller@example.com	9012345678	606 Willow Street, San Jose, USA
10	10	Jack	Wilson	jack.wilson@example.com	0123456789	707 Redwood Avenue, Austin, USA
11	11	Nikhil	Sai	nikhil@gmail.com	NULL	Tirupati, India

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

Ans.

```
update Products
set Price=Price +(Price*0.1)
```

OUTPUT:

	ProductID	ProductName	Description	Price
1	1001	Laptop	15-inch display	879.99
2	1002	Smartphone	64GB storage	659.44
3	1003	Headphones	Noise-canceling	219.99
4	1004	Smartwatch	Water-resistant	274.88
5	1005	Tablet	10-inch screen	384.99
6	1006	Keyboard	Mechanical	109.99
7	1007	Mouse	Wireless	32.99
8	1008	Monitor	4K resolution	329.99
9	1009	Printer	Laser	175.99
10	1010	External Hard Drive	1TB storage	87.99
11	1011	Earpods	50hrs durability	750.00

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.

Ans.

```
declare @orderid int =2006;
delete from Orders where orderid=@orderid;
delete from OrderDetails where orderid=@orderid;
```

OUTPUT:

	OrderID	CustomerID	OrderDate	TotalAmount	Status
1	2001	1	2024-09-01	17599.80	Shipped
2	2002	2	2024-09-03	11869.92	Pending
3	2003	3	2024-09-05	7699.65	Pending
4	2004	4	2024-09-07	3023.68	Pending
5	2006	6	2024-09-06	NULL	Pending
6	2007	7	2024-09-14	1715.48	Pending
7	2008	8	2024-09-16	14849.55	Pending
8	2009	9	2024-09-18	3519.80	Pending
9	2010	10	2024-09-20	2639.70	Pending

	OrderDetailID	OrderID	ProductID	Quantity
1	3001	2001	1001	20
2	3002	2002	1002	18
3	3003	2003	1003	35
4	3004	2004	1004	11
5	3007	2007	1007	52
6	3008	2008	1008	45
7	3009	2009	1009	20
8	3010	2010	1010	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.

Ans.

```
insert into Orders values(2006,6,'2024-09-06',489.98);
```

OUTPUT:

	OrderID	CustomerID	OrderDate	TotalAmount
1	2001	1	2024-09-01	17599.80
2	2002	2	2024-09-03	11869.92
3	2003	3	2024-09-05	7699.65
4	2004	4	2024-09-07	3023.68
5	2006	6	2024-09-06	NULL
6	2007	7	2024-09-14	1715.48
7	2008	8	2024-09-16	14849.55
8	2009	9	2024-09-18	3519.80
9	2010	10	2024-09-20	2639.70

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.

Ans.

```
declare @customerid int =3;
update Customers
set Email = 'update@gmail.com',
    Phone =  '9456208136',
    Address = '243 canada'
    where CustomerID= @customerid;
select * from Customers
```

esults | Messages

CustomerID	Firstname	Lastname	Email	Phone	Address
1	Alice	Smith	alice.smith@example.com	1234567890	123 Maple Street, New York, USA
2	Bob	Johnson	bob.johnson@example.com	2345678901	456 Oak Avenue, Los Angeles, USA
3	Carol	Williams	update@gmail.com	9456208136	243 canada
4	David	Brown	david.brown@example.com	4567890123	101 Elm Street, Houston, USA
5	Eve	Jones	eve.jones@example.com	5678901234	202 Cedar Lane, Phoenix, USA
6	Frank	Garcia	frank.garcia@example.com	6789012345	303 Birch Boulevard, Philadelphia, USA
7	Grace	Martinez	grace.martinez@example.com	7890123456	404 Spruce Drive, San Antonio, USA
8	Hank	Davis	hank.davis@example.com	8901234567	505 Aspen Circle, Dallas, USA
9	Ivy	Miller	ivy.miller@example.com	9012345678	606 Willow Street, San Jose, USA
10	Jack	Wilson	jack.wilson@example.com	0123456789	707 Redwood Avenue, Austin, USA
11	Nikhil	Sai	nikhil@gmail.com	NULL	Tirupati, India

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.

Ans.

```
update Orders
set TotalAmount = (
    select sum(Products.Price * OrderDetails.Quantity)
     from Products, OrderDetails
     where OrderDetails.ProductID = Products.ProductID
      and OrderDetails.OrderID = Orders.OrderID);
```

OUTPUT:

	OrderID	CustomerID	OrderDate	TotalAmount
1	2001	1	2024-09-01	17599.80
2	2002	2	2024-09-03	11869.92
3	2003	3	2024-09-05	7699.65
4	2004	4	2024-09-07	3023.68
5	2006	6	2024-09-06	NULL
6	2007	7	2024-09-14	1715.48
7	2008	8	2024-09-16	14849.55
8	2009	9	2024-09-18	3519.80
9	2010	10	2024-09-20	2639.70

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.

Ans.

```
declare @customerID INT = 5;

delete from OrderDetails where OrderID IN (
     select OrderID from Orders
     where CustomerID = @customerID
);
delete from Orders
where CustomerID = @customerID;
```

OUTPUT:

OrderID	CustomerID	OrderDate	TotalAmount
2001	1	2024-09-01	17599.80
2002	2	2024-09-03	11869.92
2003	3	2024-09-05	7699.65
2004	4	2024-09-07	3023.68
2006	6	2024-09-06	NULL
2007	7	2024-09-14	1715.48
2008	8	2024-09-16	14849.55
2009	9	2024-09-18	3519.80
2010	10	2024-09-20	2639.70

OrderDetailID	OrderID	ProductID	Quantity
3001	2001	1001	20
3002	2002	1002	18
3003	2003	1003	35
3004	2004	1004	11
3007	2007	1007	52
3008	2008	1008	45
3009	2009	1009	20
3010	2010	1010	30

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.

Ans.

```
insert into Products values(1011, 'Earpods', '50hrs durability', 750);
```

OUTPUT:

	ProductID	ProductName	Description	Price
1	1001	Laptop	15-inch display	879.99
2	1002	Smartphone	64GB storage	659.44
3	1003	Headphones	Noise-canceling	219.99
4	1004	Smartwatch	Water-resistant	274.88
5	1005	Tablet	10-inch screen	384.99
6	1006	Keyboard	Mechanical	109.99
7	1007	Mouse	Wireless	32.99
8	1008	Monitor	4K resolution	329.99
9	1009	Printer	Laser	175.99
10	1010	External Hard Drive	1TB storage	87.99
11	1011	Earpods	50hrs durability	750.00

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.

Ans.

```
update Orders
set Status = 'Pending'
declare @orderID int = 2001;
declare @newstatus varchar(20) ='Shipped';

update Orders
set Status = @newstatus
where OrderID = @orderID;
select * from orders
```

OUTPUT:

OrderID	CustomerID	OrderDate	TotalAmount	Status
2001	1	2024-09-01	17599.80	Shipped
2002	2	2024-09-03	11869.92	Pending
2003	3	2024-09-05	7699.65	Pending
2004	4	2024-09-07	3023.68	Pending
2006	6	2024-09-06	NULL	Pending
2007	7	2024-09-14	1715.48	Pending
2008	8	2024-09-16	14849.55	Pending
2009	9	2024-09-18	3519.80	Pending
2010	10	2024-09-20	2639.70	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.

Ans.

```
alter table Customers add Total_orders_placed int;

update Customers
SET Total_orders_placed = (
     SELECT count(Orders.CustomerID)
     from Orders
     where Orders.CustomerID=Customers.CustomerID);
```

OUTPUT:

CustomerID	Firstname	Lastname	Email	Phone	Address	Total_orders_placed
1	Alice	Smith	alice.smith@example.com	1234567890	123 Maple Street, New York, USA	1
2	Bob	Johnson	bob.johnson@example.com	2345678901	456 Oak Avenue, Los Angeles, USA	1
3	Carol	Williams	update@gmail.com	9456208136	243 canada	1
4	David	Brown	david.brown@example.com	4567890123	101 Elm Street, Houston, USA	1
5	Eve	Jones	eve.jones@example.com	5678901234	202 Cedar Lane, Phoenix, USA	0
6	Frank	Garcia	frank.garcia@example.com	6789012345	303 Birch Boulevard, Philadelphia, USA	1
7	Grace	Martinez	grace.martinez@example.com	7890123456	404 Spruce Drive, San Antonio, USA	1
8	Hank	Davis	hank.davis@example.com	8901234567	505 Aspen Circle, Dallas, USA	1
9	Ivy	Miller	ivy.miller@example.com	9012345678	606 Willow Street, San Jose, USA	1
10	Jack	Wilson	jack.wilson@example.com	0123456789	707 Redwood Avenue, Austin, USA	1
11	Nikhil	Sai	nikhil@gmail.com	NULL	Tirupati, India	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.

Ans.

```
select  OrderId, Customers.Firstname
from Customers join Orders on Customers.CustomerID=Orders.CustomerID
```

OUTPUT:

	OrderId	Firstname
1	2001	Alice
2	2002	Bob
3	2003	Carol
4	2004	David
5	2006	Frank
6	2007	Grace
7	2008	Hank
8	2009	Ivy
9	2010	Jack

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

Ans.

```
select
    P.ProductName,
    sum(OD.Quantity * P.Price) as TotalRevenue
from
    Products P
join
    OrderDetails OD on P.ProductID = OD.ProductID
join
    Orders O on OD.OrderID = O.OrderID
group by
    P.ProductName;
```

esults | Messages

ProductName	TotalRevenue
External Hard Drive	2639.70
Headphones	7699.65
Laptop	17599.80
Monitor	14849.55
Mouse	1715.48
Printer	3519.80
Smartphone	11869.92
Smartwatch	3023.68

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

Ans.

```
select
    Firstname,
    Lastname,
    Email,
    Phone,
    Address
from
    Customers
join
    Orders on Customers.CustomerID = Orders.CustomerID
group by Firstname, Lastname, Email, Phone, Address;
```

esults Messages

Firstname	Lastname	Email	Phone	Address
Alice	Smith	alice.smith@example.com	1234567890	123 Maple Street, New York, USA
Bob	Johnson	bob.johnson@example.com	2345678901	456 Oak Avenue, Los Angeles, USA
Carol	Williams	update@gmail.com	9456208136	243 canada
David	Brown	david.brown@example.com	4567890123	101 Elm Street, Houston, USA
Frank	Garcia	frank.garcia@example.com	6789012345	303 Birch Boulevard, Philadelphia, USA
Grace	Martinez	grace.martinez@example.com	7890123456	404 Spruce Drive, San Antonio, USA
Hank	Davis	hank.davis@example.com	8901234567	505 Aspen Circle, Dallas, USA
Ivy	Miller	ivy.miller@example.com	9012345678	606 Willow Street, San Jose, USA
Jack	Wilson	jack.wilson@example.com	0123456789	707 Redwood Avenue, Austin, USA

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.

Ans.

```
select top 1 ProductName , Quantity from Products
join OrderDetails on OrderDetails.ProductID=Products.ProductID
order by Quantity desc;
```

sults Messages

ProductName	Quantity
Mouse	52

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

Ans.

```
alter table Products add Category varchar(40);
update Products
set Category='Electronic Gadgets';

select ProductName, Category from Products
where Category like 'Electronic%';
```

esults | Messages

ProductName	Category
Laptop	Electronic Gadgets
Smartphone	Electronic Gadgets
Headphones	Electronic Gadgets
Smartwatch	Electronic Gadgets
Tablet	Electronic Gadgets
Keyboard	Electronic Gadgets
Mouse	Electronic Gadgets
Monitor	Electronic Gadgets
Printer	Electronic Gadgets
External Hard Drive	Electronic Gadgets
Earpods	Electronic Gadgets

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

Ans.

```
select
c.Firstname,
c.Lastname,
avg(o.TotalAmount) as Average_Amount
from Customers
join Orders o on c.CustomerID = o.CustomerID
group by  c.FirstName, c.LastName;
```

esults | Messages

Firstname	Lastname	Average_Amount
David	Brown	3023.680000
Hank	Davis	14849.550000
Frank	Garcia	NULL
Bob	Johnson	11869.920000
Grace	Martinez	1715.480000
Ivy	Miller	3519.800000
Alice	Smith	17599.800000
Carol	Williams	7699.650000
Jack	Wilson	2639.700000

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

Ans.

```
select Top 1
    O.OrderID,
    C.CustomerID,
    C.Firstname,
    C.Lastname,
    C.Email,
    O.TotalAmount as Total_Revenue
from Customers C
join Orders O on O.CustomerID = C.CustomerID
order by O.TotalAmount desc;
```

esults | Messages

OrderID	CustomerID	Firstname	Lastname	Email	Total_Revenue
2001	1	Alice	Smith	alice.smith@example.com	17599.80

8. Write an SQL query to list electronic gadgets and the number of times each product has been ordered.

Ans.

```
select P.ProductName, count(OD.ProductID) as Times_Ordered
from Products P
join OrderDetails OD on OD.ProductID=P.ProductID
group by P.ProductName
```

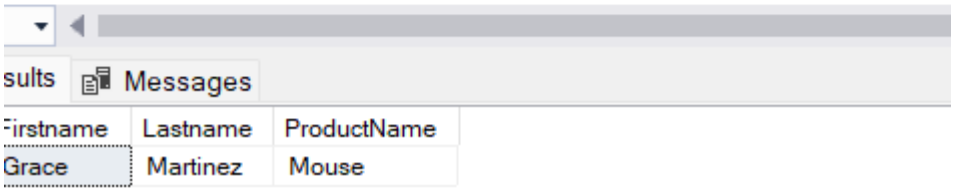
esults | Messages

ProductName	Times_Ordered
External Hard Drive	1
Headphones	1
Laptop	1
Monitor	1
Mouse	1
Printer	1
Smartphone	1
Smartwatch	1

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.

Ans.

```
declare @productname varchar(20) = 'Mouse';
select
C.Firstname,
C.Lastname,
P.ProductName
From Customers C
join Orders O on C.CustomerID = O.CustomerID
join OrderDetails OD on O.OrderID = OD.OrderID
join Products P on OD.ProductID = P.ProductID
where P.ProductName = @productname;
```



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.

Ans.

```
declare @StartDate date = '2024-09-1'
declare @EndDate date = '2024-09-15'
select sum(TotalAmount) as TotalRevenue
from Orders
where OrderDate >= @StartDate and OrderDate <= @EndDate;
```

Task 4: Subquery and its type:

1. Write an SQL query to find out which customers have not placed any orders.

Ans.

```
select
    CustomerID,
    Firstname,
    Lastname
from
    Customers
where
    CustomerID NOT IN (select CustomerID from Orders);
```

Messages

CustomerID	Firstname	Lastname
5	Eve	Jones
11	Nikhil	Sai

2. Write an SQL query to find the total number of products available for sale.

Ans.

```
select
(select count(QuantityInStocks) from Inventory)
as Available_For_Sale;
```

Messages

Available_For_Sale
10

3. Write an SQL query to calculate the total revenue generated by TechShop.

Ans.

```
select
(select sum(TotalAmount) from Orders)
as Total_Revenue_Generated;
```

esults | Messages

Total_Revenue_Generated
62917.58

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.

Ans.

```
declare @categoryname varchar(50)='Electronic Gadgets';
select
P.Category,
(select avg(Quantity) from OrderDetails) as Avg_Quantity
from Products P
join OrderDetails OD on P.ProductId=OD.ProductID
where P.Category = @categoryname
group by P.Category;
```

esults | Messages

Category	Avg_Quantity
Electronic Gadgets	28

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.

Ans.

```
declare @customerid int = 4;
select
CustomerID,
(select sum(TotalAmount) from Orders where CustomerID= @customerid) as TotalRevenue
from Orders
where CustomerID= @customerid
group by CustomerID
```

esults | Messages

CustomerID	TotalRevenue
4	3023.68

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.

Ans.

```
select
FirstName,
LastName,
TotalOrders
from (select Customers.FirstName, Customers.LastName, count(Orders.OrderID) as TotalOrders from Custome
      JOIN Orders on Orders.CustomerID = Customers.CustomerID
      group by Customers.FirstName, Customers.LastName) as CustomerOrders
      order by TotalOrders desc;
```

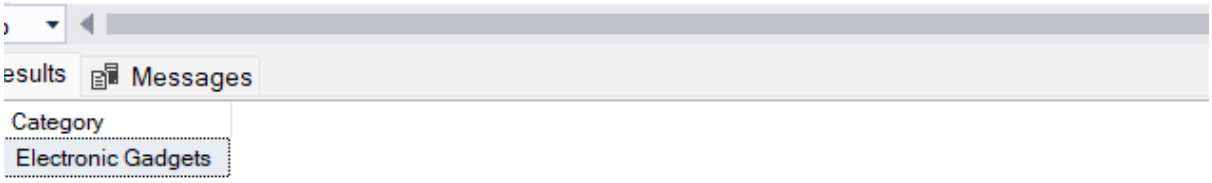
esults | Messages

FirstName	LastName	TotalOrders
David	Brown	1
Hank	Davis	1
Frank	Garcia	1
Bob	Johnson	1
Grace	Martinez	1
Ivy	Miller	1
Alice	Smith	1
Carol	Williams	1
Jack	Wilson	1

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.

Ans.

```
select Category
from (select max(quantity) as MaxQuantity, Products.Category
      from OrderDetails
      join Products on Products.ProductID=OrderDetails.ProductID
      group by Products.Category) AS ProductsOrders;
```



Category
Electronic Gadgets

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.

Ans.

```
select
C.Firstname,
C.Lastname,
O.TotalAmount
from Customers C
join Orders O on C.CustomerID=O.CustomerID
where O.TotalAmount = (select max(TotalAmount) from Orders)
```

Firstname	Lastname	TotalAmount
Alice	Smith	17599.80

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

Ans.

```
select avg(TotalRevenue) as AverageOrderValue
from (select C.CustomerID, sum(O.TotalAmount) as TotalRevenue, count(O.OrderID) as NumberOfOrders from Customers C
      join Orders O on C.CustomerID = O.CustomerID
      group by C.CustomerID) AS RevenueData
where NumberOfOrders > 0;
```

Messages

AverageOrderValue
7864.697500

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.

Ans.

```
select
    FirstName,
    LastName,
    TotalSpent,
    OrderCount
from(select sum(Orders.TotalAmount) as TotalSpent,
            Customers.FirstName,
            Customers.LastName,
            count(Orders.OrderID) as OrderCount from Orders
            join Customers on Orders.CustomerID = Customers.CustomerID
            group by Customers.FirstName, Customers.LastName) as CustomerSpending;
```

Messages

FirstName	LastName	TotalSpent	OrderCount
David	Brown	3023.68	1
Hank	Davis	14849.55	1
Frank	Garcia	NULL	1
Bob	Johnson	11869.92	1
Grace	Martinez	1715.48	1
Ivy	Miller	3519.80	1
Alice	Smith	17599.80	1
Carol	Williams	7699.65	1
Jack	Wilson	2639.70	1