

ASSIGNMENT

TechShop, Electronic Gadgets Shop

Name – Yash Pandey

Assignment - TechShop, Electronic Gadgets Shop

TASK - 1 :

1. Create the database named "TechShop".

```
create database TechShop;  
use TechShop;
```

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)  
);
```

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

```

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

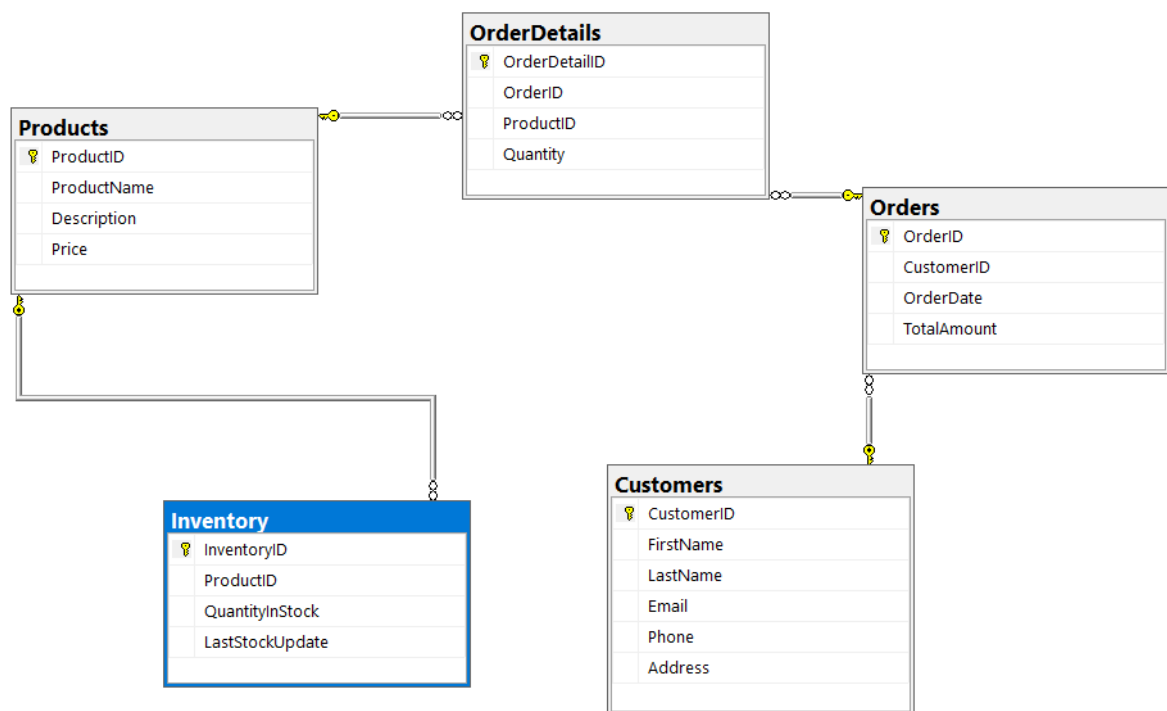
/*
3. Orders:
• OrderID (Primary Key)
• CustomerID (Foreign Key referencing Customers)
• OrderDate
• TotalAmount
*/
create table Orders(
OrderID int Identity Primary Key,
CustomerID int Foreign key references Customers(CustomerID),
OrderDate date,
TotalAmount int
);

/*
4. OrderDetails:
• OrderDetailID (Primary Key)
• OrderID (Foreign Key referencing Orders)
• ProductID (Foreign Key referencing Products)
• Quantity
*/
create table OrderDetails(
OrderDetailID int Identity Primary key,
OrderID int foreign key references Orders(OrderID),
ProductID int foreign key references Products(ProductID),
Quantity int
);

/*
5. 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
);

```

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
( 'Rahul', 'Sharma', 'rahul.sharma@gmail.com', '9876543210', 'Delhi, India'),
( 'Anjali', 'Verma', 'anjali.verma@yahoo.com', '9898989898', 'Mumbai, India'),
```

```
( 'Rakesh', 'Mehta', 'rakesh.mehta@gmail.com', '9876501234', 'Bangalore, India'),
( 'Priya', 'Singh', 'priya.singh@hotmail.com', '9823456789', 'Chennai, India'),
( 'Rajesh', 'Gupta', 'rajesh.gupta@outlook.com', '9801234567', 'Kolkata, India'),
( 'Suresh', 'Patel', 'suresh.patel@gmail.com', '9834567890', 'Ahmedabad, India'),
( 'Vikas', 'Chopra', 'vikas.chopra@yahoo.com', '9845671234', 'Hyderabad, India'),
( 'Neha', 'Jain', 'neha.jain@gmail.com', '9887654321', 'Pune, India'),
( 'Amit', 'Kumar', 'amit.kumar@hotmail.com', '9798989898', 'Jaipur, India'),
( 'Sneha', 'Rao', 'sneha.rao@gmail.com', '9878765432', 'Goa, India');
```

```
INSERT INTO Products ( ProductName, Description, Price) VALUES
( 'Laptop', 'Dell Inspiron 15 5000', 55000),
( 'Smartphone', 'Samsung Galaxy M31', 17000),
( 'Headphones', 'Sony WH-1000XM4', 25000),
( 'Washing Machine', 'LG 7 kg Fully-Automatic', 30000),
( 'Refrigerator', 'Samsung 253L Frost Free', 27000),
( 'Air Conditioner', 'Daikin 1.5 Ton Inverter', 45000),
( 'Microwave Oven', 'IFB 23L Convection', 12000),
( 'Television', 'Sony 55 inch 4K LED', 80000),
( 'Tablet', 'Apple iPad 10.2 inch', 30000),
( 'Smartwatch', 'Apple Watch Series 6', 40000);
```

```
INSERT INTO Orders (CustomerID, OrderDate, TotalAmount) VALUES
(1, '2024-09-01', 55000),
(2, '2024-09-05', 17000),
(3, '2024-09-08', 25000),
(4, '2024-09-10', 80000),
(5, '2024-09-12', 45000),
(6, '2024-09-15', 12000),
(7, '2024-09-18', 30000),
(8, '2024-09-20', 27000),
(9, '2024-09-22', 30000),
(10, '2024-09-25', 40000);
```

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

```
INSERT INTO Inventory ( ProductID, QuantityInStock, LastStockUpdate)
VALUES
(1, 50, '2024-08-30');
```

```
(2, 200, '2024-08-30'),  
(3, 150, '2024-08-30'),  
(4, 80, '2024-08-30'),  
(5, 120, '2024-08-30'),  
(6, 60, '2024-08-30'),  
(7, 90, '2024-08-30'),  
(8, 30, '2024-08-30'),  
(9, 100, '2024-08-30'),  
(10, 75, '2024-08-30');
```

TASK 2 :

/*

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

```
select (FirstName + ' ' + LastName) as Name , Email from Customers;
```

	Name	Email
1	Rahul Sharma	rahul.sharma@gmail.com
2	Anjali Vema	anjali.vema@yahoo.com
3	Rakesh Mehta	rakesh.mehta@gmail.com
4	Priya Singh	priya.singh@hotmail.com
5	Rajesh Gupta	rajesh.gupta@outlook.com
6	Suresh Patel	suresh.patel@gmail.com
7	Vikas Chopra	vikas.chopra@yahoo.com
8	Neha Jain	neha.jain@gmail.com
9	Amit Kumar	amit.kumar@hotmail.com
10	Sneha Rao	sneha.rao@gmail.com

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

*/

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

	OrderID	OrderDate	CustomerName
1	1	2024-09-01	Rahul Sharma
2	2	2024-09-05	Anjali Vema
3	3	2024-09-08	Rakesh Mehta
4	4	2024-09-10	Priya Singh
5	5	2024-09-12	Rajesh Gupta
6	6	2024-09-15	Suresh Patel
7	7	2024-09-18	Vikas Chopra
8	8	2024-09-20	Neha Jain
9	9	2024-09-22	Amit Kumar
10	10	2024-09-25	Sneha Rao

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('Yash', 'Pandey', 'yyashp381@gmail.com', 'Bareilly, India');
```

(1 row affected)

Completion time: 2024-09-24T11:58:52.8640059+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;
```

(10 rows affected)

Completion time: 2024-09-24T11:59:51.4504330+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 = 4;
```

```
delete from OrderDetails where OrderID = @OrderID;
delete from Orders where OrderID=@OrderID;
```

(1 row affected)

(1 row affected)

Completion time: 2024-09-24T12:01:07.4660704+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(1 , '2024-09-30', 5000);
```

(1 row affected)

Completion time: 2024-09-24T12:02:16.3523550+05:30

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.

```
*/  
declare @CustomerID int = 2;  
declare @new_email varchar(30) = 'abc@gmail.com';  
declare @new_address varchar(40) = ' Delhi, India';
```

```
update Customers  
set Email = @new_email , Address = @new_address  
where CustomerID = @CustomerID;
```

(1 row affected)

Completion time: 2024-09-24T12:03:17.0726116+05:30

```
/*
```

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;
```


	OrderID	CustomerID	OrderDate	TotalAmount
1	1	1	2024-09-01	60500
2	2	2	2024-09-05	37400
3	3	3	2024-09-08	55000
4	5	5	2024-09-12	99000
5	6	6	2024-09-15	13200
6	7	7	2024-09-18	66000
7	8	8	2024-09-20	29700
8	9	9	2024-09-22	66000
9	10	10	2024-09-25	44000
10	11	1	2024-09-30	NULL

/*

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.

*/

```
declare @Customer_ID int = 1;
DELETE FROM OrderDetails
WHERE OrderID IN (
    SELECT OrderID
    FROM Orders
    WHERE CustomerID = @Customer_ID
);
```

```
delete from Orders
where CustomerID = @Customer_ID;
```

(1 row affected)

(2 rows affected)

Completion time: 2024-09-24T12:06:06.1797631+05:30

/*

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

*/

```
insert into Products values('Mobile Phone' , 'iPhone 15' , 75000);
```

(1 row affected)

Completion time: 2024-09-24T12:07:08.5453906+05:30

```
/*
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(20);

declare @Order_ID int = 2;
update Orders
set Status = 'Shipped'
where OrderID = @Order_ID;

select * from Orders;
```

	OrderID	CustomerID	OrderDate	TotalAmount	Status
1	2	2	2024-09-05	37400	Shipped
2	3	3	2024-09-08	55000	NULL
3	5	5	2024-09-12	99000	NULL
4	6	6	2024-09-15	13200	NULL
5	7	7	2024-09-18	66000	NULL
6	8	8	2024-09-20	29700	NULL
7	9	9	2024-09-22	66000	NULL
8	10	10	2024-09-25	44000	NULL

```
/*
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 Order_Placed int;

UPDATE Customers
SET Order_Placed = (
    SELECT COUNT(*)
    FROM Orders
    WHERE Orders.CustomerID = Customers.CustomerID
);
select * from Customers;
```

	CustomerID	FirstName	LastName	Email	Phone	Address	Order_Placed
1	1	Rahul	Shama	rahul.shama@gmail.com	9876543210	Delhi, India	0
2	2	Anjali	Vema	abc@gmail.com	9898989898	Delhi, India	1
3	3	Rakesh	Mehta	rakesh.mehta@gmail.com	9876501234	Bangalore, India	1
4	4	Priya	Singh	priya.singh@hotmail.com	9823456789	Chennai, India	0
5	5	Rajesh	Gupta	rajesh.gupta@outlook.com	9801234567	Kolkata, India	1
6	6	Suresh	Patel	suresh.patel@gmail.com	9834567890	Ahmedabad, India	1
7	7	Vikas	Chopra	vikas.chopra@yahoo.com	9845671234	Hyderabad, India	1
8	8	Neha	Jain	neha.jain@gmail.com	9887654321	Pune, India	1
9	9	Amit	Kumar	amit.kumar@hotmail.com	9798989898	Jaipur, India	1
10	10	Sneha	Rao	sneha.rao@gmail.com	9878765432	Goa, India	1
11	11	Yash	Pandey	yyashp381@gmail.com	NULL	Bareilly, India	0

TASK 3

/*

1. Write an SQL query to retrieve a list of all orders along with customer information (e.g., customer name) for each order.

*/

```
select ( FirstName + ' ' + LastName ) as Name, Email, Phone, OrderDate
from Customers join Orders
on Orders.CustomerID = Customers.CustomerID;
```

	Name	Email	Phone	OrderDate
1	Anjali Verma	abc@gmail.com	9898989898	2024-09-05
2	Rakesh Mehta	rakesh.mehta@gmail.com	9876501234	2024-09-08
3	Rajesh Gupta	rajesh.gupta@outlook.com	9801234567	2024-09-12
4	Suresh Patel	suresh.patel@gmail.com	9834567890	2024-09-15
5	Vikas Chopra	vikas.chopra@yahoo.com	9845671234	2024-09-18
6	Neha Jain	neha.jain@gmail.com	9887654321	2024-09-20
7	Amit Kumar	amit.kumar@hotmail.com	9798989898	2024-09-22
8	Sneha Rao	sneha.rao@gmail.com	9878765432	2024-09-25

/*

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 ProductName, SUM(Price) as Total_Revenue
from Products
group by(ProductName);
```

	ProductName	Total_Revenue
1	Air Conditioner	49500
2	Headphones	27500
3	Laptop	60500
4	Microwave Oven	13200
5	Mobile Phone	75000
6	Refrigerator	29700
7	Smartphone	18700
8	Smartwatch	44000
9	Tablet	33000
10	Television	88000
11	Washing Machine	33000

```
/*
```

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

```
*/
```

```
select ( FirstName + ' ' + LastName ) as Name, Email, Phone
from Customers join Orders
on Orders.CustomerID = Customers.CustomerID;
```

	Name	Email	Phone
1	Anjali Vema	abc@gmail.com	9898989898
2	Rakesh Mehta	rakesh.mehta@gmail.com	9876501234
3	Rajesh Gupta	rajesh.gupta@outlook.com	9801234567
4	Suresh Patel	suresh.patel@gmail.com	9834567890
5	Vikas Chopra	vikas.chopra@yahoo.com	9845671234
6	Neha Jain	neha.jain@gmail.com	9887654321
7	Amit Kumar	amit.kumar@hotmail.com	9798989898
8	Sneha Rao	sneha.rao@gmail.com	9878765432

```
/*
```

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
ProductName, max(Quantity)
from Products join OrderDetails
on Products.ProductID = OrderDetails.ProductID
group by (ProductName);
```

	ProductName	(No column name)
1	Air Conditioner	2

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

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

```

	ProductName	Categories
1	Laptop	Electronics
2	Smartphone	Electronics
3	Tablet	Electronics

```

/*
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 (c.FirstName+' '+c.LastName) as 'CustomerName',
avg(o.TotalAmount) as TotalAvgValue
from Customers c
join Orders o on c.CustomerID = o.CustomerID
group by (c.FirstName+' '+c.LastName);

```

	CustomerName	TotalAvgValue
1	Amit Kumar	66000
2	Anjali Verna	37400
3	Neha Jain	29700
4	Rajesh Gupta	99000
5	Rakesh Mehta	55000
6	Sneha Rao	44000
7	Suresh Patel	13200
8	Vikas Chopra	66000

```

/*
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 top 1
o.TotalAmount as HighestRenue,
c.FirstName+' '+c.LastName as 'Name',
c.Email, c.Phone, c.Address
from Orders o join Customers c
on o.CustomerID = c.CustomerID
order by o.TotalAmount desc;

```

	HighestRenue	Name	Email	Phone	Address
1	99000	Rajesh Gupta	rajesh.gupta@outlook.com	9801234567	Kolkata India

Click to select the whole column

```

/*
8. Write an SQL query to 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	0
2	2	Smartphone	1
3	9	Tablet	1

Click to select the whole column

```

/*
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.
*/

```

```

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	Anjali Verma	2
2	Vikas Chopra	7

/*

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.

*/

```
DECLARE @STARTDATE DATE = '2020-06-01';
```

```
DECLARE @ENDDATE DATE = '2024-09-22';
```

```
Select SUM(TotalAmount) as TotalRevenue
```

```
From
```

```
Orders
```

```
Where
```

```
OrderDate Between @STARTDATE and @ENDDATE
```

	TotalRevenue
1	366300

TASK 4 :

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

```
SELECT FirstName, LastName
FROM Customers
WHERE CustomerID NOT IN (
    SELECT CustomerID FROM Orders
);
```

	FirstName	LastName
1	Yash	Pandey

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

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

	TotalProducts
1	10

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

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

	TotalRevenue
1	361000

/*

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 @inputCategory varchar(20) = 'Electronics';
```

```
SELECT AVG(od.Quantity) AS AverageQuantity
FROM OrderDetails od
JOIN Products p ON od.ProductID = p.ProductID
WHERE p.Categories = @inputCategory;
```

	AverageQuantity
1	2

/*

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 = 1;
SELECT SUM(o.TotalAmount) AS TotalCustomerRevenue
FROM Orders o
WHERE o.CustomerID = @Customerid;
```

	TotalCustomerRevenue
1	NULL

/*

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.

*/

```
SELECT top 1
c.FirstName+' '+ c.LastName, COUNT(o.OrderID) AS OrderCount
FROM Customers c
JOIN Orders o ON c.CustomerID = o.CustomerID
GROUP BY c.FirstName+' '+c.LastName
ORDER BY OrderCount DESC;
```

	(No column name)	OrderCount
1	Anjali Verna	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.

*/

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

	Categories	TotalQuantity
1	Other	6

/*

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 1
c.FirstName+' '+ c.LastName Name, SUM(o.TotalAmount) AS TotalSpent
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.Categories = 'Electronics'
GROUP BY c.FirstName+' '+ c.LastName
ORDER BY TotalSpent DESC;
```

	Name	TotalSpent
1	Vikas Chopra	66000

/*

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

*/

```
SELECT AVG(o.TotalAmount) AS AverageOrderValue
FROM Orders o;
```

	AverageOrderValue
1	51287

/*

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 c.FirstName+' '+ c.LastName Name, COUNT(o.OrderID) AS OrderCount
FROM Customers c
LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
GROUP BY c.FirstName+' '+ c.LastName;
```

	Name	OrderCount
1	Amit Kumar	1
2	Anjali Verma	1
3	Neha Jain	1
4	Priya Singh	0
5	Rahul Sharma	0
6	Rajesh Gupta	1
7	Rakesh Mehta	1
8	Sneha Rao	1
9	Suresh Patel	1
10	Vikas Chopra	1
11	Yash Pandey	0