TECHSHOP ASSIGNMENT

- VIKAS REDDY GORANTLA

Task 1 – Database Design:

1. Create the database named "TechShop".

A) Create database TechShopNew;

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

A) create table Customers(

CustomerID int IDENTITY PRIMARY KEY,

FirstName varchar(20),

LastName varchar(20),

Email varchar(300),

Phone varchar(15),

Address varchar(100)

);

create table Products(

ProductID int PRIMARY KEY,

ProductName varchar(20),

Description varchar(100),

Price DECIMAL(10,2),

Category VARCHAR(50)

);

create table Orders(

OrderID int PRIMARY KEY,

CustomerID INT,

OrderDate DATE,

TotalAmount Decimal(10,2),

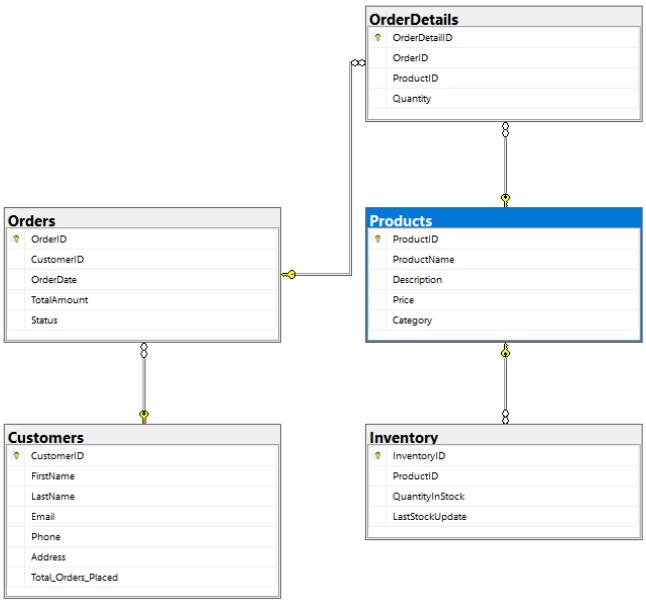
FOREIGN KEY(CustomerID) REFERENCES Customers(CustomerID) ON DELETE CASCADE

);

```
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
);


create table Inventory(
InventoryID int PRIMARY KEY,
ProductID INT,
QuantityInStock INT,
LastStockUpdate DATE,
FOREIGN KEY(ProductID) REFERENCES Products(ProductID) ON DELETE CASCADE
);
```

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



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

A) For CustomerID, ProductID, OrderID, OrderDetailsID, InventoryID -- Primary Key Codes

CustomerID int IDENTITY PRIMARY KEY,

ProductID int PRIMARY KEY,

OrderID int PRIMARY KEY,

OrderDetailsID int PRIMARY KEY,

InventoryID int PRIMARY KEY,

For CustomerID, ProductID, OrderID. -- Foreign Key Codes

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

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

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

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

a. Customers b. Products c. Orders d. OrderDetails

A)

```
INSERT INTO Customers VALUES('Rahul', 'Sharma', 'rahulsharma21@example.com', '9875732190', '123 MG Road, Delhi, India');

INSERT INTO Customers VALUES('Emily', 'Johnson', 'emilyjohnson90@example.com', '8751239876', '456 Maple St, Banglore, India');

INSERT INTO Customers VALUES('Amit', 'Patel', 'amitpatel88@example.com', '9123456789', '789 Nehru St, Mumbai, India');

INSERT INTO Customers VALUES('Jayanth', 'Sukhla', 'jayanthsukhla11@example.com', '9759876543', '101 Royal Avenue, Ahmedabad, India');

INSERT INTO Customers VALUES('Priya', 'Reddy', 'priyareddy22@example.com', '9876543219', '202 Rajaji Nagar, Hyderabad, India');

INSERT INTO Customers VALUES('John', 'Martin', 'johnmartin43@example.com', '8790650431', '303 Pine Street, Kochi, India');

INSERT INTO Customers VALUES('Anjali', 'Mehta', 'anjalimehta54@example.com', '9123456780', '404 Church St, Bangalore, India');

INSERT INTO Customers VALUES('Mahendra', 'Chowdary', 'mahendra65@example.com', '8908654321', '505 Elm St, Vijayawada, India');

INSERT INTO Customers VALUES('Suresh', 'Kumar', 'sureshkumar77@example.com', '9876543211', '606 Gandhi Nagar, Chennai, India');

INSERT INTO Customers VALUES('Sophia', 'Martin', 'sophiamartin34@example.com', '9120721001', '707 Sunset Blvd, Amristar, India');
```

```
INSERT INTO Products VALUES(101, 'Laptop', 'High-performance laptop with 16GB RAM', 39999.00,'Electronics');

INSERT INTO Products VALUES(102, 'Smartphone', '5G smartphone with 128GB storage', 27599.00,'Electronics');

INSERT INTO Products VALUES(103, 'Headphones', 'Wireless noise-canceling headphones', 249.99,'Electronics');

INSERT INTO Products VALUES(104, 'Tablet', '10-inch tablet with 64GB storage', 2599.00,'Electronics');

INSERT INTO Products VALUES(105, 'SmartTV', '4K Smart TV', 29999.99,'Accessories');

INSERT INTO Products VALUES(106, 'Coffee Maker', '5 Rated Automatic Coffee Machine', 1699.99,'Accessories');

INSERT INTO Products VALUES(107, 'Refrigerator', 'Ultimate Refrigerator for Modern Living', 25999.99,'Accessories');

INSERT INTO Products VALUES(108, 'Microwave Oven', 'Cooks Everything Perfectly In Minutes', 9999.99,'Appliances');

INSERT INTO Products VALUES(109, 'Blender', 'Powerful and Smooth', 799.99,'Appliances');

INSERT INTO Products VALUES(110, 'Vaccum Cleaner', 'Keep Your Home Clean With Less Effort', 17599.50,'Appliances');


INSERT INTO Orders VALUES(201, 1, '2024-05-01', 9999.99); -- bought 1 oven

INSERT INTO Orders VALUES(202, 2, '2024-05-12', 249.99); -- 1 headphone

INSERT INTO Orders VALUES(203, 3, '2024-06-03', 799.00); -- 1 blender

INSERT INTO Orders VALUES(204, 4, '2024-06-24', 55198.00); -- 2 smartphones

INSERT INTO Orders VALUES(205, 5, '2024-06-30', 39999.00); -- 1 laptop

INSERT INTO Orders VALUES(206, 6, '2024-07-12', 1699.99); -- 1 coffee maker

INSERT INTO Orders VALUES(207, 7, '2024-07-17', 25999.99); -- 1 Refrigerator,

INSERT INTO Orders VALUES(208, 8, '2024-07-21', 749.97); -- 3 headphones

INSERT INTO Orders VALUES(209, 9, '2024-08-09', 2599.99); -- 1 tablet

INSERT INTO Orders VALUES(210, 10, '2024-08-10', 29999.99); -- 1 tv

INSERT INTO Orders VALUES(211, 2, '2024-08-19', 17599.99); -- 1 vaccumclaner

INSERT INTO Orders VALUES(212, 4, '2024-08-24', 27599.00); -- 1 smartphone

INSERT INTO Orders VALUES(213, 6, '2024-09-20', 1599.98); -- 2 blenders
```

```
INSERT INTO OrderDetails VALUES(301, 201, 108, 1);
INSERT INTO OrderDetails VALUES(302, 202, 103, 1);
INSERT INTO OrderDetails VALUES(303, 203, 109, 1);
INSERT INTO OrderDetails VALUES(304, 204, 102, 2);
INSERT INTO OrderDetails VALUES(305, 205, 101, 1);
INSERT INTO OrderDetails VALUES(306, 206, 106, 1);
INSERT INTO OrderDetails VALUES(307, 207, 107, 1);
INSERT INTO OrderDetails VALUES(308, 208, 103, 3);
INSERT INTO OrderDetails VALUES(309, 209, 104, 1);
INSERT INTO OrderDetails VALUES(310, 210, 105, 1);
INSERT INTO OrderDetails VALUES(311, 211, 105, 1);
INSERT INTO OrderDetails VALUES(312, 212, 110, 1);
INSERT INTO OrderDetails VALUES(313, 213, 109, 2);
```

```
INSERT INTO Inventory VALUES(401, 101, 50, '2023-08-10');
INSERT INTO Inventory VALUES(402, 102, 200, '2023-08-16');
INSERT INTO Inventory VALUES(403, 103, 150, '2023-08-25');
INSERT INTO Inventory VALUES(404, 104, 100, '2023-08-28');
INSERT INTO Inventory VALUES(405, 105, 80, '2023-08-30');
INSERT INTO Inventory VALUES(406, 106, 60, '2023-09-02');
INSERT INTO Inventory VALUES(407, 107, 250, '2023-09-15');
INSERT INTO Inventory VALUES(408, 108, 75, '2023-09-17');
INSERT INTO Inventory VALUES(409, 109, 120, '2023-09-25');
INSERT INTO Inventory VALUES(410, 110, 300, '2023-09-30');
```

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

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

```
select FirstName, LastName, Email from Customers;
```

100 %

Results | Messages

	FirstName	LastName	Email
1	Rahul	Sharma	rahulsharma21@example.com
2	Emily	Johnson	emilyjohnson90@example.com
3	Amit	Patel	amitpatel88@example.com
4	Jayanth	Sukhla	jayanthsukhla11@example.com
5	Priya	Reddy	reddy28@gamil.com
6	John	Martin	johnmartin43@example.com
7	Anjali	Mehta	anjalimehta54@example.com
8	Mahendra	Chowdary	mahendra65@example.com
9	Suresh	Kumar	sureshkumar77@example.com
10	Sophia	Martin	sophiamartin34@example.com
11	Vikas Reddy	Gorantla	gorantla12@gmail.com
12	Vikas Reddy	Gorantla	gorantla12@gmail.com
13	Vikas Reddy	Gorantla	gorantla12@gmail.com

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

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

100 %

Results | Messages

	OrderId	OrderDate	FirstName	LastName
1	201	2024-05-01	Rahul	Sharma
2	202	2024-05-12	Emily	Johnson
3	203	2024-06-03	Amit	Patel
4	204	2024-06-24	Jayanth	Sukhla
5	205	2024-06-30	Priya	Reddy
6	206	2024-07-12	John	Martin
7	207	2024-07-17	Anjali	Mehta
8	208	2024-07-21	Mahendra	Chowdary
9	210	2024-08-10	Sophia	Martin
10	211	2024-08-19	Emily	Johnson
11	213	2024-09-20	John	Martin
12	214	2024-09-22	Priya	Reddy

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('Vikas Reddy', 'Gorantla', 'gorantla12@gmail.com', 'Near CheckPost, Kurnool, India');
select * from Customers;
```

100 %

Results Messages

	CustomerID	FirstName	LastName	Email	Phone	Address	Total_Orders_Placed
1	1	Rahul	Sharma	rahulsharma21@example.com	9875732190	123 MG Road, Delhi, India	1
2	2	Emily	Johnson	emilyjohnson90@example.com	8751239876	456 Maple St, Banglore, India	2
3	3	Amit	Patel	amitpatel88@example.com	9123456789	789 Nehru St, Mumbai, India	1
4	4	Jayanth	Sukhla	jayanthsukhla11@example.com	9759876543	101 Royal Avenue, Ahmedabad, India	1
5	5	Priya	Reddy	reddy28@gamil.com	9876543219	Gandhi Street, Mumbai, India	2
6	6	John	Martin	johnmartin43@example.com	8790650431	303 Pine Street, Kochi, India	2
7	7	Anjali	Mehta	anjalimehta54@example.com	9123456780	404 Church St, Bangalore, India	1
8	8	Mahendra	Chowdary	mahendra65@example.com	8908654321	505 Elm St, Vijayawada, India	1
9	9	Suresh	Kumar	sureshkumar77@example.com	9876543211	606 Gandhi Nagar, Chennai, India	0
10	10	Sophia	Martin	sophiamartin34@example.com	9120721001	707 Sunset Blvd, Amristar, India	1
11	11	Vikas Reddy	Gorantla	gorantla12@gmail.com	NULL	Near CheckPost, Kurnool, India	0

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 + (Price * 10)/100
where Category='Electronics';
select* from Products;
```

100 %

Results Messages

	ProductID	ProductName	Description	Price	Category
1	101	Laptop	High-performance laptop with 16GB RAM	58562.54	Electronics
2	102	Smartphone	5G smartphone with 128GB storage	40407.70	Electronics
3	103	Headphones	Wireless noise-canceling headphones	366.01	Electronics
4	104	Tablet	10-inch tablet with 64GB storage	3805.20	Electronics
5	105	SmartTV	4K Smart TV	29999.99	Accessories
6	106	Coffee Maker	5 Rated Automatic Coffee Machine	1699.99	Accessories
7	107	Refrigerator	Ultimate Refrigerator for Modern Living	25999.99	Accessories
8	108	Microwave Oven	Cooks Everything Perfectly In Minutes	9999.99	Appliances
9	109	Blender	Powerful and Smooth	799.99	Appliances
10	110	Vaccum Cleaner	Keep Your Home Clean With Less Effort	17599.50	Appliances
11	111	Gaming CoolPad	A Wired High Performance lightening Coolpad	531.74	Electronics

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 = 212;
delete from Orders where OrderID=@orderID;
delete from OrderDetails where OrderDetailID=@orderID;
select * from Orders;
```

100 %

Results | Messages

	OrderID	CustomerID	OrderDate	TotalAmount	Status
1	201	1	2024-05-01	9999.99	Pending
2	202	2	2024-05-12	274.99	Pending
3	203	3	2024-06-03	799.99	Pending
4	204	4	2024-06-24	60717.80	Pending
5	205	5	2024-06-30	43998.90	Pending
6	206	6	2024-07-12	1699.99	Pending
7	207	7	2024-07-17	25999.99	Pending
8	208	8	2024-07-21	824.97	Pending
9	210	10	2024-08-10	29999.99	Pending
10	211	2	2024-08-19	29999.99	Pending
11	213	6	2024-09-20	1599.98	Pending
12	214	5	2024-09-22	NULL	Pending

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(214, 5, '2024-09-22', 249.99);
```

100 %

Results | Messages

	OrderID	CustomerID	OrderDate	TotalAmount	Status
1	201	1	2024-05-01	9999.99	Pending
2	202	2	2024-05-12	274.99	Pending
3	203	3	2024-06-03	799.99	Pending
4	204	4	2024-06-24	60717.80	Pending
5	205	5	2024-06-30	43998.90	Pending
6	206	6	2024-07-12	1699.99	Pending
7	207	7	2024-07-17	25999.99	Pending
8	208	8	2024-07-21	824.97	Pending
9	210	10	2024-08-10	29999.99	Pending
10	211	2	2024-08-19	29999.99	Pending
11	213	6	2024-09-20	1599.98	Pending
12	214	5	2024-09-22	NULL	Pending

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 = 5;
declare @email varchar(30) = 'reddy28@gamil.com'
declare @address varchar(100) = 'Gandhi Street, Mumbai, India'
update Customers
set Email=@email, Address=@address
where CustomerID=@customerID;
```

100 %

Results | Messages

	CustomerID	FirstName	LastName	Email	Phone	Address	Total_Orders_Placed
1	1	Rahul	Sharma	rahulsharma21@example.com	9875732190	123 MG Road, Delhi, India	1
2	2	Emily	Johnson	emilyjohnson90@example.com	8751239876	456 Maple St, Banglore, India	2
3	3	Amit	Patel	amitpatel88@example.com	9123456789	789 Nehru St, Mumbai, India	1
4	4	Jayanth	Sukhla	jayanthsukhla11@example.com	9759876543	101 Royal Avenue, Ahmedabad, India	1
5	5	Priya	Reddy	reddy28@gamil.com	9876543219	Gandhi Street, Mumbai, India	2
6	6	John	Martin	johnmartin43@example.com	8790650431	303 Pine Street, Kochi, India	2
7	7	Anjali	Mehta	anjalimehta54@example.com	9123456780	404 Church St, Bangalore, India	1
8	8	Mahendra	Chowdary	mahendra65@example.com	8908654321	505 Elm St, Vijayawada, India	1
9	9	Suresh	Kumar	sureshkumar77@example.com	9876543211	606 Gandhi Nagar, Chennai, India	0
10	10	Sophia	Martin	sophiamartin34@example.com	9120721001	707 Sunset Blvd, Amristar, India	1
11	11	Vikas Reddy	Gorantla	gorantla12@gmail.com	NULL	Near CheckPost, Kurnool, India	0

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 SUM(OD.Quantity * P.Price)
    FROM OrderDetails OD
    JOIN Products P ON OD.ProductID = P.ProductID
    WHERE OD.OrderID = Orders.OrderID
);
select * from Orders;
```

100 %

Results | Messages

	OrderID	CustomerID	OrderDate	TotalAmount	Status
1	201	1	2024-05-01	9999.99	Pending
2	202	2	2024-05-12	366.01	Pending
3	203	3	2024-06-03	799.99	Pending
4	204	4	2024-06-24	80815.40	Pending
5	205	5	2024-06-30	58562.54	Pending
6	206	6	2024-07-12	1699.99	Pending
7	207	7	2024-07-17	25999.99	Pending
8	208	8	2024-07-21	1098.03	Pending
9	210	10	2024-08-10	29999.99	Pending
10	211	2	2024-08-19	29999.99	Pending
11	213	6	2024-09-20	1599.98	Pending
12	214	5	2024-09-22	NULL	Pending

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 @customerID INT = 9;
DELETE FROM OrderDetails
WHERE OrderID IN (SELECT OrderID FROM Orders WHERE CustomerID = @CustomerID);
DELETE FROM Orders WHERE CustomerID = @CustomerID;
```

100 %

Results | Messages

	OrderID	CustomerID	OrderDate	TotalAmount	Status
1	201	1	2024-05-01	9999.99	Pending
2	202	2	2024-05-12	366.01	Pending
3	203	3	2024-06-03	799.99	Pending
4	204	4	2024-06-24	80815.40	Pending
5	205	5	2024-06-30	58562.54	Pending
6	206	6	2024-07-12	1699.99	Pending
7	207	7	2024-07-17	25999.99	Pending
8	208	8	2024-07-21	1098.03	Pending
9	210	10	2024-08-10	29999.99	Pending
10	211	2	2024-08-19	29999.99	Pending
11	213	6	2024-09-20	1599.98	Pending
12	214	5	2024-09-22	NULL	Pending

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(111, 'Gaming CoolPad', 'A Wired High Performance lightening Coolpad', 399.50,'Electronics');
```

100 %

Results | Messages

	ProductID	ProductName	Description	Price	Category
1	101	Laptop	High-performance laptop with 16GB RAM	58562.54	Electronics
2	102	Smartphone	5G smartphone with 128GB storage	40407.70	Electronics
3	103	Headphones	Wireless noise-canceling headphones	366.01	Electronics
4	104	Tablet	10-inch tablet with 64GB storage	3805.20	Electronics
5	105	SmartTV	4K Smart TV	29999.99	Accessories
6	106	Coffee Maker	5 Rated Automatic Coffee Machine	1699.99	Accessories
7	107	Refrigerator	Ultimate Refrigerator for Modern Living	25999.99	Accessories
8	108	Microwave Oven	Cooks Everything Perfectly In Minutes	9999.99	Appliances
9	109	Blender	Powerful and Smooth	799.99	Appliances
10	110	Vaccum Cleaner	Keep Your Home Clean With Less Effort	17599.50	Appliances
11	111	Gaming CoolPad	A Wired High Performance lightening Coolpad	531.74	Electronics

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(30);  --added column Status
update Orders  -- updating staus of order_id 201 to pending
set Status = 'Pending'
where OrderID in (201,202,203,204,205,206,207,208,210,211,213,214);
declare @orderID int = 201;
declare @newstatus varchar(20) ='Shipped';
update Orders  -- updating staus of order_id 201 to shipped
set Status = @newstatus
where OrderID = @orderID;
```

100 %

Results | Messages

	OrderID	CustomerID	OrderDate	TotalAmount	Status
1	201	1	2024-05-01	9999.99	Shipped
2	202	2	2024-05-12	366.01	Pending
3	203	3	2024-06-03	799.99	Pending
4	204	4	2024-06-24	80815.40	Pending
5	205	5	2024-06-30	58562.54	Pending
6	206	6	2024-07-12	1699.99	Pending
7	207	7	2024-07-17	25999.99	Pending
8	208	8	2024-07-21	1098.03	Pending
9	210	10	2024-08-10	29999.99	Pending
10	211	2	2024-08-19	29999.99	Pending
11	213	6	2024-09-20	1599.98	Pending
12	214	5	2024-09-22	NULL	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 Total_Orders_Placed int;
select * from Customers;
update Customers
SET Total_Orders_Placed = (
    SELECT count(*)
    from Orders
    where Orders.CustomerID=Customers.CustomerID);
```

100 %

Results | Messages

	CustomerID	FirstName	LastName	Email	Phone	Address	Total_Orders_Placed
1	1	Rahul	Sharma	rahulsharma21@example.com	9875732190	123 MG Road, Delhi, India	1
2	2	Emily	Johnson	emilyjohnson90@example.com	8751239876	456 Maple St, Banglore, India	2
3	3	Amit	Patel	amitpatel88@example.com	9123456789	789 Nehru St, Mumbai, India	1
4	4	Jayanth	Sukhla	jayanthsukhla11@example.com	9759876543	101 Royal Avenue, Ahmedabad, India	1
5	5	Priya	Reddy	reddy28@gamil.com	9876543219	Gandhi Street, Mumbai, India	2
6	6	John	Martin	johnmartin43@example.com	8790650431	303 Pine Street, Kochi, India	2
7	7	Anjali	Mehta	anjalimehta54@example.com	9123456780	404 Church St, Bangalore, India	1
8	8	Mahendra	Chowdary	mahendra65@example.com	8908654321	505 Elm St, Vijayawada, India	1
9	9	Suresh	Kumar	sureshkumar77@example.com	9876543211	606 Gandhi Nagar, Chennai, India	0
10	10	Sophia	Martin	sophiamartin34@example.com	9120721001	707 Sunset Blvd, Amristar, India	1
11	11	Vikas Reddy	Gorantla	gorantla12@gmail.com	NULL	Near CheckPost, Kurnool, 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.

```
SELECT O.OrderID,O.OrderDate,O.TotalAmount,C.FirstName,C.LastName,C.Email,C.Phone,C.Address
FROM Orders O
JOIN Customers C ON O.CustomerID = C.CustomerID;
```

100 %

Results Messages

	OrderID	OrderDate	TotalAmount	FirstName	LastName	Email	Phone	Address
1	201	2024-05-01	9999.99	Rahul	Sharma	rahulsharma21@example.com	9875732190	123 MG Road, Delhi, India
2	202	2024-05-12	366.01	Emily	Johnson	emilyjohnson90@example.com	8751239876	456 Maple St, Banglore, India
3	203	2024-06-03	799.99	Amit	Patel	amitpatel88@example.com	9123456789	789 Nehru St, Mumbai, India
4	204	2024-06-24	80815.40	Jayanth	Sukhla	jayanthsukhla11@example.com	9759876543	101 Royal Avenue, Ahmedabad, India
5	205	2024-06-30	58562.54	Priya	Reddy	reddy28@gamil.com	9876543219	Gandhi Street, Mumbai, India
6	206	2024-07-12	1699.99	John	Martin	johnmartin43@example.com	8790650431	303 Pine Street, Kochi, India
7	207	2024-07-17	25999.99	Anjali	Mehta	anjalimehta54@example.com	9123456780	404 Church St, Bangalore, India
8	208	2024-07-21	1098.03	Mahendra	Chowdary	mahendra65@example.com	8908654321	505 Elm St, Vijayawada, India
9	210	2024-08-10	29999.99	Sophia	Martin	sophiamartin34@example.com	9120721001	707 Sunset Blvd, Amristar, India
10	211	2024-08-19	29999.99	Emily	Johnson	emilyjohnson90@example.com	8751239876	456 Maple St, Banglore, India
11	213	2024-09-20	1599.98	John	Martin	johnmartin43@example.com	8790650431	303 Pine Street, Kochi, India
12	214	2024-09-22	NULL	Priya	Reddy	reddy28@gamil.com	9876543219	Gandhi Street, Mumbai, India

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

```
SELECT P.ProductName,SUM(OD.Quantity * P.Price) AS TotalRevenue
FROM OrderDetails OD
JOIN Products P ON OD.ProductID = P.ProductID
JOIN Orders O ON OD.OrderID = O.OrderID
WHERE P.Category = 'Electronics'
GROUP BY P.ProductName;
```

100 %

Results Messages

	ProductName	TotalRevenue
1	Headphones	1464.04
2	Laptop	58562.54
3	Smartphone	80815.40

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

```
select distinct FirstName , LastName, Email, Phone, Address from Customers
join Orders On Customers.CustomerID=Orders.CustomerID;
```

100 %

Results | Messages

	FirstName	LastName	Email	Phone	Address
1	Amit	Patel	amitpatel88@example.com	9123456789	789 Nehru St, Mumbai, India
2	Anjali	Mehta	anjalimehta54@example.com	9123456780	404 Church St, Bangalore, India
3	Emily	Johnson	emilyjohnson90@example.com	8751239876	456 Maple St, Banglore, India
4	Jayanth	Sukhla	jayanthsukhla11@example.com	9759876543	101 Royal Avenue, Ahmedabad, India
5	John	Martin	johnmartin43@example.com	8790650431	303 Pine Street, Kochi, India
6	Mahendra	Chowdary	mahendra65@example.com	8908654321	505 Elm St, Vijayawada, India
7	Priya	Reddy	reddy28@gamil.com	9876543219	Gandhi Street, Mumbai, India
8	Rahul	Sharma	rahulsharma21@example.com	9875732190	123 MG Road, Delhi, India
9	Sophia	Martin	sophiamartin34@example.com	9120721001	707 Sunset Blvd, Amristar, India

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 P.ProductName,SUM(OD.Quantity) AS TotalQuantityOrdered
FROM OrderDetails OD
JOIN Products P ON OD.ProductID = P.ProductID
JOIN Orders O ON OD.OrderID = O.OrderID
WHERE P.Category = 'Electronics'
GROUP BY P.ProductName
ORDER BY TotalQuantityOrdered DESC;
```

100 %

Results | Messages

	ProductName	TotalQuantityOrdered
1	Headphones	4
2	Smartphone	2
3	Laptop	1

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

```
select ProductName, Category from Products;
```

100 %

Results | Messages

	ProductName	Category
1	Laptop	Electronics
2	Smartphone	Electronics
3	Headphones	Electronics
4	Tablet	Electronics
5	SmartTV	Accessories
6	Coffee Maker	Accessories
7	Refrigerator	Accessories
8	Microwave Oven	Appliances
9	Blender	Appliances
10	Vaccum Cleaner	Appliances
11	Gaming CoolPad	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,AVG(O.TotalAmount) AS AverageOrderValue
FROM Customers C
JOIN Orders O ON C.CustomerID = O.CustomerID
GROUP BY C.FirstName, C.LastName;
```

100 %

Results | Messages

	FirstName	LastName	AverageOrderValue
1	Mahendra	Chowdary	1098.030000
2	Emily	Johnson	15183.000000
3	John	Martin	1649.985000
4	Sophia	Martin	29999.990000
5	Anjali	Mehta	25999.990000
6	Amit	Patel	799.990000
7	Priya	Reddy	58562.540000
8	Rahul	Sharma	9999.990000
9	Jayanth	Sukhla	80815.400000

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 O.OrderID,C.FirstName,C.LastName,O.TotalAmount AS TotalRevenue
FROM Orders O
JOIN Customers C ON O.CustomerID = C.CustomerID
ORDER BY O.TotalAmount DESC
```

100 %

Results | Messages

	OrderID	FirstName	LastName	TotalRevenue
1	204	Jayanth	Sukhla	80815.40
2	205	Priya	Reddy	58562.54
3	210	Sophia	Martin	29999.99
4	211	Emily	Johnson	29999.99
5	207	Anjali	Mehta	25999.99
6	201	Rahul	Sharma	9999.99
7	206	John	Martin	1699.99
8	213	John	Martin	1599.98
9	208	Mahendra	Chowdary	1098.03
10	203	Amit	Patel	799.99
11	202	Emily	Johnson	366.01
12	214	Priya	Reddy	NULL

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

```
SELECT P.ProductName,COUNT(OD.OrderDetailID) AS NumberOfOrders
FROM Products P
LEFT JOIN OrderDetails OD ON P.ProductID = OD.ProductID
GROUP BY P.ProductName;
```

100 %

Results | Messages

	ProductName	NumberOfOrders
1	Blender	2
2	Coffee Maker	1
3	Gaming CoolPad	0
4	Headphones	2
5	Laptop	1
6	Microwave Oven	1
7	Refrigerator	1
8	Smartphone	1
9	SmartTV	2
10	Tablet	0
11	Vaccum Cleaner	0

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 @productname varchar(50) = 'Smartphone';
select FirstName,LastName from Customers
join Orders on Customers.CustomerID = Orders.CustomerID
join OrderDetails on Orderdetails.OrderID = Orders.OrderID
join Products on OrderDetails.ProductID = Products.ProductID
where ProductName=@productname;
```

100 %

Results | Messages

	FirstName	LastName
1	Jayanth	Sukhla

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 = '2024-08-16'
declare @EndDate date = '2024-09-30'
SELECT SUM(TotalAmount) AS TotalRevenue
FROM Orders
WHERE OrderDate BETWEEN @StartDate AND @EndDate;
```

100 %

Results | Messages

	TotalRevenue
1	31599.97

Task 4. Subquery and its type:

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

```
SELECT C.FirstName,C.LastName
FROM Customers C
LEFT JOIN Orders O ON O.CustomerID = C.CustomerID
WHERE O.OrderID IS NULL;
```

100 %

Results | Messages

	FirstName	LastName
1	Suresh	Kumar
2	Vikas Reddy	Gorantla

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

```
select count(*) Total_Products from Products;
```

100 %

Results | Messages

	Total_Products
1	11

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

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

100 %

Results | Messages

	TotalRevenue
1	240941.90

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 @categoryname varchar(50) = 'Electronics';
select avg(Quantity) Average from OrderDetails
join Products on Products.ProductID=OrderDetails.ProductID
where Category in (select Category from Products where Category=@categoryname);
```

100 %

Results | Messages

	Average
1	1

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 C.FirstName,C.LastName,SUM(O.TotalAmount) AS TotalRevenue
FROM Customers C
JOIN Orders O ON C.CustomerID = O.CustomerID
WHERE C.CustomerID = @CustomerID
GROUP BY C.FirstName, C.LastName;
```

100 %

Results | Messages

	FirstName	LastName	TotalRevenue
1	Rahul	Sharma	9999.99

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

100 %

Results | Messages

	FirstName	LastName	NumberOfOrders
1	Emily	Johnson	2
2	John	Martin	2
3	Priya	Reddy	2

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 Category
from (select top 1 max(quantity) MaxQuantity,Products.Category
      from OrderDetails
      join Products on Products.ProductID=OrderDetails.ProductID
      group by Products.Category
      order by Category desc) AS ProductsOrders;
```

100 %

Results | Messages

	Category
1	Electronics

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,SUM(O.TotalAmount) AS TotalSpending
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.Category = 'Electronics'
GROUP BY C.FirstName, C.LastName
ORDER BY TotalSpending DESC;
```

100 %

Results | Messages

	FirstName	LastName	TotalSpending
1	Jayanth	Sukhla	80815.40

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

```
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;
```

100 %

Results | Messages

	AverageOrderValue
1	26771.322222

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,COUNT(O.OrderID) AS NumberOfOrders
FROM Customers C
LEFT JOIN Orders O ON C.CustomerID = O.CustomerID
GROUP BY C.FirstName, C.LastName
ORDER BY NumberOfOrders DESC;
```

100 %

Results | Messages

	FirstName	LastName	NumberOfOrders
1	Emily	Johnson	2
2	John	Martin	2
3	Priya	Reddy	2
4	Rahul	Sharma	1
5	Jayanth	Sukhla	1
6	Sophia	Martin	1
7	Anjali	Mehta	1
8	Amit	Patel	1
9	Mahendra	Chowdary	1
10	Vikas Reddy	Gorantla	0
11	Suresh	Kumar	0