ASSIGNMENT

TechShop, an electronic gadgets shop

1) Create the database named "TechShop".
➜ create database techshop;

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

Customer Table:
➜ create table customers
➜ (CustomerID int primary key,
➜ FirstName text,
➜ LastName varchar(50) not null,
➜ Email varchar(50) unique,
➜ Phone bigint,
➜ Address varchar(60));

	Field	Type	Null	Key
▶	CustomerID	int	NO	PRI
	FirstName	text	YES	
	LastName	varchar(50)	NO	
	Email	varchar(50)	YES	UNI
	phone	bigint	YES	
	Address	varchar(60)	YES	
	NoOfOrders	mediumtext	YES	

Product Table:
➜ create table Products
➜ (ProductID int primary key,
➜ ProductName text not null,
➜ Description varchar(255),
➜ Price decimal(10,2) not null);

Field	Type	Null	Key
ProductID	int	NO	PRI
ProductName	text	NO	
description	varchar(255)	YES	
price	decimal(10,2)	YES	

Orders Table:
➜ create table Orders
➜ (OrderID int primary key,
➜ CustomerID int not null,
➜ OrderDate date,
➜ TotalAmount decimal(10,2),
➜ foreign key (CustomerID) references customers(CustomerID));

Field	Type	Null	Key
OrderID	int	NO	PRI
CustomerID	int	NO	MUL
OrderDate	date	YES	
totalamount	decimal(10,2)	YES	
status	varchar(20)	YES	

Orderdetails Table:
➜ create table OrderDetails
➜ (OrderDetailId int primary key,
➜ OrderID int not null,

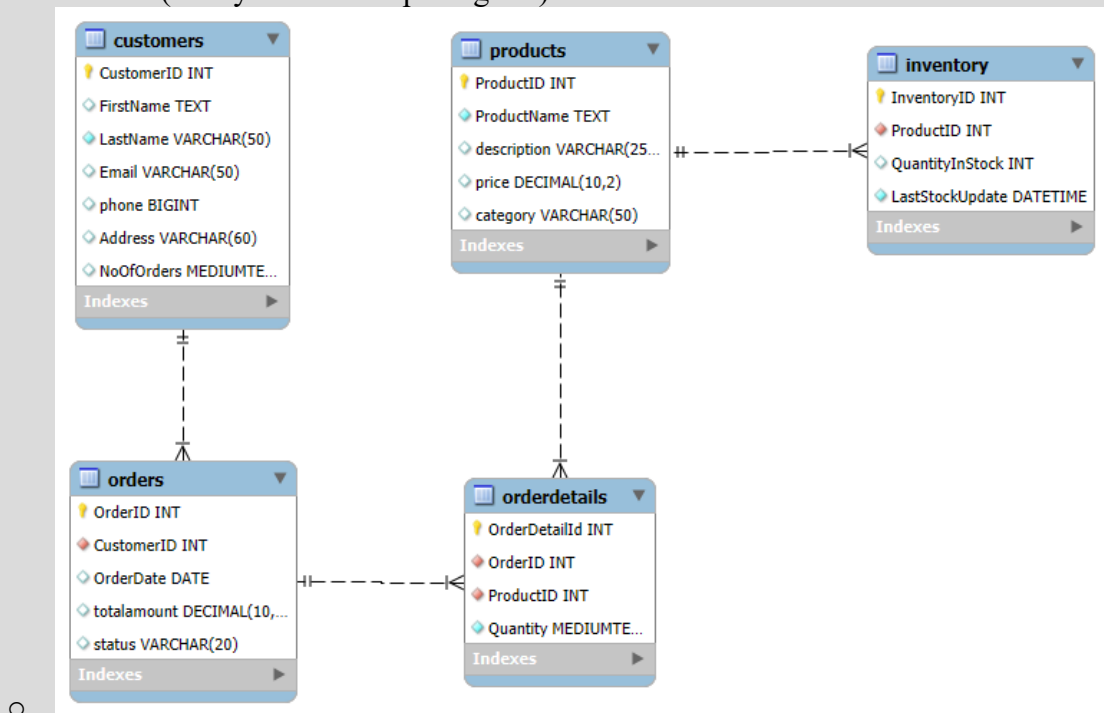| Field	Type	Null	Key
OrderDetailId	int	NO	PRI
OrderID	int	NO	MUL
ProductID	int	NO	MUL
Quantity	mediumtext	NO	

➔ ProductID int not null,
➔ Quantity long not null,
➔ foreign key(OrderID) references Orders(OrderID),
➔ foreign key(ProductID) references Products (ProductID));

Inventory Table:
➔ create table Inventory
➔ (InventoryID int primary key,
➔ ProductID int not null,
➔ QuantityInStock int,
➔ LastStockUpdate datetime not null,
➔ foreign key (ProductID) references Products(ProductID));

Field	Type	Null	Key	Default
InventoryID	int	NO	PRI	NULL
ProductID	int	NO	MUL	NULL
QuantityInStock	int	YES		NULL
LastStockUpdate	datetime	NO		NULL

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

○

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

Customer Table:
➔ CustomerID int primary key

	Field	Type	Null	Key
▶	CustomerID	int	NO	PRI

Product Table:
➔ ProductID int primary key,

Field	Type	Null	Key
ProductID	int	NO	PRI

Orders Table:

Field	Type	Null	Key
OrderID	int	NO	PRI
CustomerID	int	NO	MUL

➔ (OrderID int primary key,

➔ foreign key (CustomerID) references customers(CustomerID));

Orderdetails Table:

Field	Type	Null	Key
OrderDetailId	int	NO	PRI
OrderID	int	NO	MUL
ProductID	int	NO	MUL

➔ (OrderDetailId int primary key,

➔ foreign key(OrderID) references Orders(OrderID),

➔ foreign key(ProductID) references Products (ProductID));

Inventory Table:

Field	Type	Null	Key	Default
InventoryID	int	NO	PRI	NULL
ProductID	int	NO	MUL	NULL

➔ (InventoryID int primary key,

➔ foreign key (ProductID) references Products(ProductID));

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

a. Customers

insert into customers values

(1, 'Fathima', 'Zahira', 'Fathima.Zah@gmail.com', '1029384756', '123 Amityville St'),

(2, 'Manoj', 'Mani', 'ahmahmed@gmail.com', '9876543210', '108 Bu Danig Ave'),

(3, 'Sungu', 'Saitama', 'onesaitama@gmail.com', '9218239531', '456 Fubuki Ave'),

(4, 'Luffy', 'Krishnan', 'angelfruit@gmail.com', '9182305812', '789 Nami St'),

(5, 'Sungjinwoo', 'Singh', 'jinwoo@gmail.com', '9471823912', '321 Aura Farm'),

(6, 'Ichigo', 'Kumar', 'kurosaki.ichigo@example.com', '9184728248', '123 Bankai Ave'),

(7, 'Isagi', 'Kahn', 'clown@gmail.com', '9432109876', '1 Strika St'),

(8, 'Uzumaki', 'Nair', 'boruto.nair@gmail.com', '9321098765', '106 Sasuke Blvd'),

(9, 'Gojo', 'Reddy', 'gojo.suguru@gmail.com', '9420987654', '753 Shibuya Ave'),

(10, 'Mohammed', 'Aizen', 'Aizen.watashi@gmail.com', '9105576543', '852 Yokoso Rd');

CustomerID	FirstName	LastName	Email	phone	Address
1	Fathima	Zahira	Fathima.Zah@gmail.com	1029384756	123 Amityville St
2	Manoj	Mani	ahmahmed@gmail.com	9876543210	108 Bu Danig Ave
3	Sungu	Saitama	onesaitama@gmail.com	9218239531	456 Fubuki Ave
4	Luffy	Krishnan	angelfruit@gmail.com	9182305812	789 Nami St
5	Sungjinwoo	Singh	jinwoo@gmail.com	9471823912	321 Aura Farm
6	Ichigo	Kumar	kurosaki.ichigo@example.com	9184728248	123 Bankai Ave
7	Isagi	Kahn	clown@gmail.com	9432109876	1 Strika St
8	Uzumaki	Nair	boruto.nair@gmail.com	9321098765	106 Sasuke Blvd
9	Gojo	Reddy	gojo.suguru@gmail.com	9420987654	753 Shibuya Ave
10	Mohammed	Aizen	Aizen.watashi@gmail.com	9105576543	852 Yokoso Rd

b. Products

insert into products values
('Laptop', 'High-performance laptop', 5500.00),
('Laser Printer', 'Lightweight and lighting fast Printer', 1100.00),
('Smart Gloves', 'Highly durable and innovative Smart Gloves', 880.00),
('Stretchable Smartphone', 'A flexible, unbreakable smartphone', 1925.00),
('Shadow SSD', 'A high-speed 1TB SSD with rapid data transfer', 825.00),
('Hollow VR Headset', 'A VR headset with immersive audio and visuals', 2530.00),
('Tactical Smart Watch', 'Advanced sports analyzing watch', 1700.00),
('Rasengan Drone', 'High-speed drone with rotor blades', 3200.00),
('Infinity Projector', 'Projector with limitless focus and crystal-clear visuals', 1980.00),
('Illusionary Smart Glasses', 'Smart glasses with holographic displays', 2200.00);

ProductID	ProductName	description	price
1	Laptop	High-performance laptop	5500.00
2	Laser Printer	Lightweight and lighting fast Printer	1100.00
3	Smart Gloves	Highly durable and innovative Smart Gloves	880.00
4	Stretchable Smartphone	A flexible, unbreakable smartphone	1925.00
5	Shadow SSD	A high-speed 1TB SSD with rapid data transfer	825.00
6	Hollow VR Headset	A VR headset with immersive audio and visuals	2530.00
7	Tactical Smart Watch	Advanced sports analyzing watch	1700.00
8	Rasengan Drone	High-speed drone with rotor blades	3200.00
9	Infinity Prjector	Projector with limitless focus and crystal-clear vi...	1980.00
10	Illusionary Smart Glasses	Smart glasses with holographic displays	2200.00

c. Orders

insert into orders values
(1, 1, '2024-12-30', 5500.00),
(2, 2, '2024-11-29', 2000.00),
(3, 3, '2025-03-13', 88000.00),
(4, 4, '2025-03-11', 38500.00),
(5, 5, '2025-02-05', 11550.00),
(6, 6, '2025-01-28', 154330.00),
(7, 7, '2025-03-25', 180200.00),
(8, 8, '2025-03-15', 26400.00),
(9, 9, '2025-03-19', 3960.00),
(10, 10, '2025-04-30', 22000.00);

OrderID	CustomerID	OrderDate	TotalAmount
1	1	2024-12-30	5500.00
2	2	2024-11-29	2000.00
3	3	2025-03-13	88000.00
4	4	2025-03-11	38500.00
5	5	2025-02-05	11550.00
6	6	2025-01-28	154330.00
7	7	2025-03-25	180200.00
8	8	2025-03-15	26400.00
9	9	2025-03-19	3960.00
10	10	2025-04-30	22000.00

d. OrderDetails

insert into OrderDetails values
(101,1, 1, 1),
(102,2, 2, 75),
(103,3, 3, 100),

(104,4, 3, 20),
(105,5, 5, 14),
(106,6, 6, 61),
(107,7, 7, 106),
(108,8, 5, 32),
(109,9, 9, 2),
(110,10, 10, 10);

OrderDetailId	OrderID	ProductID	Quantity
101	1	1	1
102	2	2	75
103	3	3	100
104	4	3	20
105	5	5	14
106	6	6	61
107	7	7	106
108	8	5	32
109	9	9	2
110	10	10	10

e. Inventory
insert into Inventory values
(201,1, 100, '2025-03-16 15:10:39'),
(202,2, 150, '2025-03-16 15:28:18'),
(203,3, 200, '2025-03-16 15:28:18'),
(204,4, 25, '2025-03-16 15:28:18'),
(205,5, 100, '2025-03-16 15:28:18'),
(206,6, 100, '2025-03-16 15:28:18'),
(207,7, 160, '2025-03-16 15:28:18'),
(208,8, 50, '2025-03-16 15:28:18'),
(209,9, 75, '2025-03-16 15:28:18'),
(210, 10, 100, '2025-03-16 15:28:18');

InventoryID	ProductID	QuantityInStock	LastStockUpdate
201	1	100	2025-03-16 15:10:39
202	2	150	2025-03-16 15:28:18
203	3	200	2025-03-16 15:28:18
204	4	25	2025-03-16 15:28:18
205	5	100	2025-03-16 15:28:18
206	6	100	2025-03-16 15:28:18
207	7	160	2025-03-16 15:28:18
208	8	50	2025-03-16 15:28:18
209	9	75	2025-03-16 15:28:18
210	10	100	2025-03-16 15:28:18

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;

FirstName	LastName	Email
Fathima	Zahira	Fathima.Zah@gmail.com
Manoj	Mani	ahmahmed@gmail.com
Sungu	Saitama	onesaitama@gmail.com
Luffy	Krishnan	angelfruit@gmail.com
Sungjinwoo	Singh	jinwoo@gmail.com
Ichigo	Kumar	kurosaki.ichigo@example.com
Isagi	Kahn	clown@gmail.com
Uzumaki	Nair	boruto.nair@gmail.com
Gojo	Reddy	gojo.suguru@gmail.com
Mohammed	Aizen	Aizen.watashi@gmail.com

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

➜ select o.OrderID, o.OrderDate, c.FirstName, c.LastName
   From Orders o
   join Customers c on o.CustomerID = c.CustomerID;

OrderID	OrderDate	FirstName	LastName
1	2024-12-30	Fathima	Zahira
3	2025-03-13	Sungu	Saitama
4	2025-03-11	Luffy	Krishnan
5	2025-02-05	Sungjinwoo	Singh
6	2025-01-28	Ichigo	Kumar
7	2025-03-25	Isagi	Kahn
8	2025-03-15	Uzumaki	Nair
9	2025-03-19	Gojo	Reddy
10	2025-04-30	Mohammed	Aizen

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 values
   (11, 'Musfi', 'Begum', 'musfirah@gmail.com', 9182471829, '123 One Tower St');

11	Musfi	Begum	musfirah@gmail.com	9182471829	123 One Tower St

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

price
6050.00
1210.00
968.00
2117.50
907.50
2783.00
1870.00
3520.00
2178.00
2420.00
3190.00

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.

➔ delete from OrderDetails
   where orderid = 2;
   delete from orders
   where orderid =2;

OrderDetailId	OrderID	ProductID	Quantity
101	1	1	1
103	3	3	100
104	4	4	20
105	5	5	14
106	6	6	61
107	7	7	106
108	8	5	32
109	9	9	2
110	10	10	10

OrderID	CustomerID	OrderDate	totalamount
1	1	2024-12-30	5500.00
3	3	2025-03-13	88000.00
4	4	2025-03-11	38500.00
5	5	2025-02-05	11550.00
6	6	2025-01-28	154330.00
7	7	2025-03-25	180200.00
8	8	2025-03-15	26400.00
9	9	2025-03-19	3960.00
10	10	2025-04-30	22000.00

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
   (2, 2, '2025-04-15', 4500.00);

OrderID	CustomerID	OrderDate	totalamount
2	2	2025-04-15	4500.00

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.

➔ update customers
  set email = 'sherry@gmail.com', address= ' 108 Jeju isle Ave'
  where customerid=2;

CustomerID	FirstName	LastName	Email	phone	Address
2	Manoj	Manoj	sherry@gmail.com	9876543210	108 Jeju isle Ave

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 o
  join (select od.OrderID, sum(p.Price * od.Quantity) as TotalAmount
  from OrderDetails od
  join Products p on od.ProductID = p.ProductID
  group by od.OrderID) as OrderTotals on o.OrderID = OrderTotals.OrderID
  set o.TotalAmount = OrderTotals.TotalAmount;

OrderID	CustomerID	OrderDate	totalamount
1	1	2024-12-30	6050.00
2	2	2025-04-15	4500.00
3	3	2	2025-03-13	96800.00
4	4	2025-03-11	42350.00
5	5	2025-02-05	12705.00
6	6	2025-01-28	169763.00
7	7	2025-03-25	198220.00
8	8	2025-03-15	29040.00
9	9	2025-03-19	4356.00
10	10	2025-04-30	24200.00

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.

➔ delete from OrderDetails
  where OrderID in (
  select OrderID
  from Orders
  where CustomerID = 2);

delete from orders
where customerID = 2;

OrderDetailId	OrderID	ProductID	Quantity
101	1	1	1
103	3	3	100
104	4	4	20
105	5	5	14
106	6	6	61
107	7	7	106
108	8	5	32
109	9	9	2
110	10	10	10

OrderID	CustomerID	OrderDate	totalamount
1	1	2024-12-30	6050.00
3	3	2025-03-13	96800.00
4	4	2025-03-11	42350.00
5	5	2025-02-05	12705.00
6	6	2025-01-28	169763.00
7	7	2025-03-25	198220.00
8	8	2025-03-15	29040.00
9	9	2025-03-19	4356.00
10	10	2025-04-30	24200.00

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
   (11,'Tablet', 'Advanced M4 chip integrated Tablet', 3190.00);

11	Tablet	Advanced M4 chip integrated Tablet	3190.00

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.

➔ alter table orders
  add status varchar(20);

update orders
set status = 'Pending';

update Orders
set status = 'Shipped'
where orderid=1;

OrderID	CustomerID	OrderDate	totalamount	status
1	1	2024-12-30	6050.00	Shipped
3	3	2025-03-13	96800.00	Pending
4	4	2025-03-11	42350.00	Pending
5	5	2025-02-05	12705.00	Pending
6	6	2025-01-28	169763.00	Pending
7	7	2025-03-25	198220.00	Pending
8	8	2025-03-15	29040.00	Pending
9	9	2025-03-19	4356.00	Pending
10	10	2025-04-30	24200.00	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 long;

  update Customers c
  set NoOfOrders =(
  select COUNT(*)
  from Orders o
  where o.CustomerID = c.CustomerID);

NoOfOrders
1
0
1
1
1
1
1
1
1
1
1
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 orderid, totalamount, customers.*
    from orders
    join customers on orders.customerid = customers.customerid;

orderid	totalamount	CustomerID	FirstName	LastName	Email	phone	Address	NoOfOrders
1	6050.00	1	Fathima	Zahira	Fathima.Zah@gmail.com	1029384756	123 Amityville St	1
3	96800.00	3	Sungu	Saitama	onesaitama@gmail.com	9218239531	456 Fubuki Ave	1
4	42350.00	4	Luffy	Krishnan	angelfruit@gmail.com	9182305812	789 Nami St	1
5	12705.00	5	Sungjinwoo	Singh	jinwoo@gmail.com	9471823912	321 Aura Farm	1
6	169763.00	6	Ichigo	Kumar	kurosaki.ichigo@example.com	9184728248	123 Bankai Ave	1
7	198220.00	7	Isagi	Kahn	clown@gmail.com	9432109876	1 Strika St	1
8	29040.00	8	Uzumaki	Nair	boruto.nair@gmail.com	9321098765	106 Sasuke Blvd	1
9	4356.00	9	Gojo	Reddy	gojo.suguru@gmail.com	9420987654	753 Shibuya Ave	1
10	24200.00	10	Mohammed	Aizen	Aizen.watashi@gmail.com	9105576543	852 Yokoso Rd	1

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 * quantity) as TotalRevenue
    from orderdetails
    join products on orderdetails.productid = products.productid
    group by productname;

productname	TotalRevenue
Laptop	6050
Smart Gloves	96800
Stretchable Smartphone	42350
Shadow SSD	41745
Hollow VR Headset	169763
Tactical Smart Watch	198220
Infinity Prjector	4356
Illusionary Smart Glasses	24200

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

➡ select distinct c.customerid, concat(c.firstname,' ', c.lastname) as fullname, c.email, c.phone

from customers c
join orders o on c.customerid = o.customerid;

customerid	fullname	email	phone
1	Fathima Zahira	Fathima.Zah@gmail.com	1029384756
3	Sungu Saitama	onesaitama@gmail.com	9218239531
4	Luffy Krishnan	angelfruit@gmail.com	9182305812
5	Sungjinwoo Singh	jinwoo@gmail.com	9471823912
6	Ichigo Kumar	kurosaki.ichigo@example.com	9184728248
7	Isagi Kahn	clown@gmail.com	9432109876
8	Uzumaki Nair	boruto.nair@gmail.com	9321098765
9	Gojo Reddy	gojo.suguru@gmail.com	9420987654
10	Mohammed Aizen	Aizen.watashi@gmail.com	9105576543

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(q.quantity) as TotalQuantity
from orderdetails q
join products p on q.productid = p.productid
group by p.productname
order by totalquantity desc
limit 1;

productname	TotalQuantity
Tactical Smart Watch	106

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

➜ select productname, category
    from products;

productname	category
Laptop	Touch Devices
Laser Printer	Printer
Smart Gloves	Wearable Technology
Stretchable Smartphone	Touch Devices
Shadow SSD	Storage Devices
Hollow VR Headset	Wearable Technology
Tactical Smart Watch	Wearable Technology
Rasengan Drone	Drones
Infinity Prjector	Projector
Illusionary Smart Glasses	Wearable Technology
Tablet	Touch Devices

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.customerid, concat(c.firstName,' ', c.lastName)as FullName, avg(o.TotalAmount) as AverageOrderValue
    from customers c
    join orders o on c.customerid = o.customerid
    group by c.customerID, c.firstname, c.lastname;

customerid	FullName	AverageOrderValue
1	Fathima Zahira	6050.000000
3	Sungu Saitama	96800.000000
4	Luffy Krishnan	42350.000000
5	Sungjinwoo Singh	12705.000000
6	jigo Kumar	169763.000000
7	Isagi Kahn	198220.000000
8	Uzumaki Nair	29040.000000
9	Gojo Reddy	4356.000000
10	Mohammed Aizen	24200.000000

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 totalrevenue desc

limit 1;

orderid	firstname	lastname	totalrevenue
7	Isagi	Kahn	198220.00

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

➜ select productname, count(orderdetailid) as NoOfOrders
     from products
     left join orderdetails on products.productid = orderdetails.productid
     group by productname;

productname	NoOfOrders
Laptop	1
Laser Printer	0
Smart Gloves	1
Stretchable Smartphone	1
Shadow SSD	2
Hollow VR Headset	1
Tactical Smart Watch	1
Rasengan Drone	0
Infinity Prjector	1
Illusionary Smart Glasses	1
Tablet	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.

➜ select c.CustomerID, c.FirstName, c.LastName, c.Email, c.Phone
     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 = 'Laptop';

CustomerID	FirstName	LastName	Email	Phone
1	Fathima	Zahira	Fathima.Zah@gmail.com	1029384756

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.

➜ select sum(totalamount) as totalrevenue

from orders
where orderdate between '2025-01-01' and '2025-04-02';

totalrevenue
553234.00

Task 4. Subquery and its type:

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

➜ select customerid, firstname, lastname
    from customers
    where customerid not in (
    select distinct customerid
    from orders);

customerid	firstname	lastname
2	Manoj	Mani
11	Musfi	Begum

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

➜ select count(*) as totalproducts
    from products
    where productid in (
    select productid
    from inventory
    where quantityinstock > 0);

totalproducts
11

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

➜ select sum (TotalAmount) as TotalRevenue
    from (
    select TotalAmount
    from Orders)
    as OrderTotals;

TotalRevenue
583484.00

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.

➔ select avg(quantity) as averagequantityordered
   from orderdetails
   where productid in (
   select productid
   from products
   where category = 'wearable technology');

averagequantityordered
69.25

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.

➔ select customerid, fullname, sum(totalamount) as totalrevenue
   from (
   select c.customerid, concat(c.firstname, ' ', c.lastname) as fullname,
   o.totalamount
   from customers c
   join orders o on c.customerid = o.customerid
   where c.customerid = 1
) as customertotals
   group by customerid, fullname;

customerid	fullname	totalrevenue
1	Fathima Zahira	6050.00

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 customerid, fullname, numorders
   from (
   select c.customerid, concat(c.firstname, ' ', c.lastname) as fullname,
   count(o.orderid) as numorders
   from customers c
   join orders o on c.customerid = o.customerid
   group by c.customerid, fullname) as customerorders
   order by numorders desc;

customerid	fullname	numorders
1	Fathima Zahira	1
3	Sungu Saitama	1
4	Luffy Krishnan	1
5	Sungjinwoo Singh	1
6	Ichigo Kumar	1
7	Isagi Kahn	1
8	Uzumaki Nair	1
9	Gojo Reddy	1
10	Mohammed Aizen	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 category, TotalQuantityOrdered
      from (select p.category, sum(o.quantity) as totalquantityordered
      from orderdetails o
      join products p on o.productid=p.productid
      group by p.category
) as CategoryTotal
      order by TotalQuantityOrdered desc
      limit 1;

category	TotalQuantityOrdered
Wearable Technology	277

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 customerid, fullname, TotalSpending
      from (
      select c.customerid, concat(c.FirstName, ' ', c.LastName) as FullName,
      sum(o.totalamount) as totalspending
      from customers c
      join orders o on c.customerid = o.customerid
      group by c.customerid, FullName) as customerspending
      order by TotalSpending desc
      limit 1;

customerid	fullname	TotalSpending
7	Isagi Kahn	198220.00

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

➔ select c.customerid, concat(c.firstname, ' ', c.lastname) as fullname,
       (select sum(o.totalamount) from orders o where o.customerid = c.customerid)
       /
       (select count(o.orderid) from orders o where o.customerid = c.customerid) as
       averageordervalue
       from customers c
       where exists (
       select 1
       from orders o
       where o.customerid = c.customerid);

customerid	fullname	averageordervalue
1	Fathima Zahira	6050.000000
3	Sungu Saitama	96800.000000
4	Luffy Krishnan	42350.000000
5	Sungjinwoo Singh	12705.000000
6	Ichigo Kumar	169763.000000
7	Isagi Kahn	198220.000000
8	Uzumaki Nair	29040.000000
9	Gojo Reddy	4356.000000
10	Mohammed Aizen	24200.000000

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 fullname, ordercount
       from (
       select
       concat(c.firstname,' ', c.lastname) as fullname,
       count(o.orderid) as ordercount
       from customers c
       left join orders o on c.customerid = o.customerid
       group by c.customerid, c.firstname, c.lastname)
       as customer_orders;

fullname	ordercount
Fathima Zahira	1
Manoj Mani	0
Sungu Saitama	1
Luffy Krishnan	1
Sungjinwoo Singh	1
Ichigo Kumar	1
Isagi Kahn	1
Uzumaki Nair	1
Gojo Reddy	1
Mohammed Aizen	1
Musfi Begum	0