TechShop, an electronic gadgets shop

1. Create the database named "TechShop"

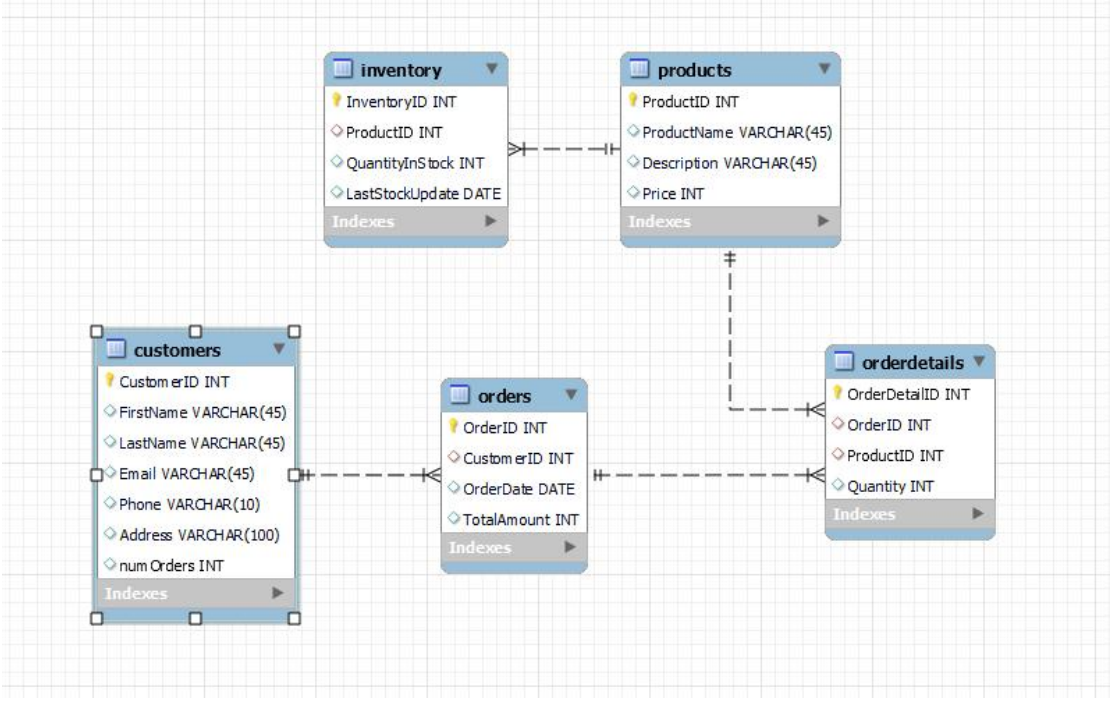
```
create database TechShop;
use TechShop;
```

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

```
CREATE TABLE Customers(
  CustomerID INT,
  FirstName VARCHAR(45),
  LastName VARCHAR(45),
  Email VARCHAR(45),
  Phone VARCHAR(10),
  Address VARCHAR(100),
  PRIMARY KEY (CustomerID));

CREATE TABLE Products(
  ProductID INT,
  ProductName VARCHAR(45),
  Description VARCHAR(45),
  Price INT,
  PRIMARY KEY (ProductID));
```

```
CREATE TABLE Orders(
  OrderID INT,
  CustomerID INT,
  OrderDate DATE,
  TotalAmount INT,
  PRIMARY KEY ( OrderID));

CREATE TABLE OrderDetails(
  OrderDetailID INT,
  OrderID INT,
  ProductID INT,
  Quantity INT,
  PRIMARY KEY ( OrderDetailID));

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

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



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

```
ALTER TABLE Orders ADD FOREIGN KEY(CustomerID) REFERENCES Customers(CustomerID);
ALTER TABLE OrderDetails ADD FOREIGN KEY(OrderID) REFERENCES Orders(OrderID);
ALTER TABLE OrderDetails ADD FOREIGN KEY(ProductID) REFERENCES Products(ProductID);
ALTER TABLE Inventory ADD FOREIGN KEY(ProductID) REFERENCES Products(ProductID);
```

4. 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 VALUES
    (1, "James", "Smith", "JamesSmith@gmail.com", "1234567890", "Chennai"),
    (2, "Christopher", "Anderson", "ChristopherAnderson@gmail.com", "1234567891", "Noida"),
    (3, "Ronald", "Clark", "RonaldClark@gmail.com", "1234567892", "Pune"),
    (4, "Mary", "Wright", "MaryWright@gmail.com", "1234567893", "Bangalore"),
    (5, "Lisa", "Mitchell", "LisaMitchell@gmail.com", "1234567894", "Hydrabad"),
    (6, "Michelle", "Johnson", "MichelleJohnson@gmail.com", "1234567895", "Delhi"),
    (7, "John", "Thomas", "JohnThomas@gmail.com", "1234567896", "Kochi"),
    (8, "Daniel", "Lopez", "DanielRodriguez@gmail.com","1234567897", "Chennai"),
    (9, "Anthony", "Smith", "AnthonyLopez@gmail.com", "1234567898", "Bangalore"),
    (10, "Patricia", "Perez", "PatriciaPerez@gmail.com", "1234567899", "Noida");
```

```
INSERT INTO Products VALUES
    (1, "Phone", "electronic", 10000),
    (2, "Laptop", "electronic", 20000),
    (3, "PC", "electronic", 30000),
    (4, "Earphone", "Accessories", 2000),
    (5, "SmartWatch", "electronic", 4000),
    (6, "PowerBank", "electronic", 1000),
    (7, "Speaker", "Accessories", 2000),
    (8, "RGBlights", "Accessories", 1000),
    (9, "Cable", "Accessories", 500products),
    (10, "GPU", "electronic", 35000);
```

```
INSERT INTO Orders VALUES
    (1, 10, "2024-1-11", 10000),
    (2, 9, "2024-1-12", 20000),
    (3, 8, "2024-1-13", 30000),
    (4, 7, "2024-1-14", 2000),
    (5, 6, "2024-1-15", 4000),
    (6, 5, "2024-1-16", 1000),
    (7, 4, "2024-1-17", 2000),
    (8, 3, "2024-1-18", 1000),
    (9, 2, "2024-1-19", 500),
    (10, 1, "2024-1-20", 35000);
INSERT INTO OrderDetails VALUES
    (1, 1, 1, 1),
    (2, 2, 2, 1),
    (3, 3, 3, 1),
    (4, 4, 4, 1),
    (5, 5, 5, 1),
    (6, 6, 6, 1),
    (7, 7, 7, 1),
    (8, 8, 8, 1),
    (9, 9, 9, 1),
    (10, 10, 10, 1);
```

```
INSERT INTO Inventory VALUES
    (1, 1, 10, "2024-1-10"),
    (2, 2, 10, "2024-1-10"),
    (3, 3, 10, "2024-1-10"),
    (4, 4, 10, "2024-1-10"),
    (5, 5, 10, "2024-1-10"),
    (6, 6, 10, "2024-1-10"),
    (7, 7, 10, "2024-1-10"),
    (8, 8, 10, "2024-1-10"),
    (9, 9, 10, "2024-1-10"),
    (10, 10, 10, "2024-1-10");
```

CustomerID	FirstName	LastName	Email	Phone	Address
1	James	Smith	mark@gmail.com	1234567890	pondichery
2	Christopher	Anderson	ChristopherAnderson@gmail.com	1234567891	Noida
3	Ronald	Clark	RonaldClark@gmail.com	1234567892	Pune
4	Mary	Wright	MaryWright@gmail.com	1234567893	Bangalore
5	Lisa	Mitchell	LisaMitchell@gmail.com	1234567894	Hydrabad
6	Michelle	Johnson	MichelleJohnson@gmail.com	1234567895	Delhi
7	John	Thomas	JohnThomas@gmail.com	1234567896	Kochi
8	Daniel	Lopez	DanielRodriguez@gmail.com	1234567897	Chennai
9	Anthony	Smith	AnthonyLopez@gmail.com	1234567898	Bangalore
10	Patricia	Perez	PatriciaPerez@gmail.com	1234567899	Noida

ProductID	ProductName	Description	Price
1	Phone	electronic	11000
2	Laptop	electronic	22000
3	PC	electronic	33000
4	Earphone	Accessories	2000
5	SmartWatch	electronic	4400
6	PowerBank	elec	100
7	Speaker	Accessories	2000
8	RGBlights	Accessories	1000
9	Cable	Accessories	500
10	GPU	electronic	38500

OrderID	CustomerID	OrderDate	TotalAmount
1	10	2024-01-11	11000
4	7	2024-01-14	2000
5	6	2024-01-15	4400
6	5	2024-01-16	1100
7	4	2024-01-17	2000
8	3	2024-01-18	1000
9	2	2024-01-19	500
10	1	2024-01-20	38500

	OrderDetailID	OrderID	ProductID	Quantity
▶	1	1	1	1
	4	4	4	1
	5	5	5	1
	6	6	6	1
	7	7	7	1
	8	8	8	1
	9	9	9	1
	10	10	10	1

	InventoryID	ProductID	QuantityInStock	LastStockUpdate
▶	1	1	10	2024-01-10
	2	2	10	2024-01-10
	3	3	10	2024-01-10
	4	4	10	2024-01-10
	5	5	10	2024-01-10
	6	6	10	2024-01-10
	7	7	10	2024-01-10
	8	8	10	2024-01-10
	9	9	10	2024-01-10
	10	10	10	2024-01-10

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
▶	James	Smith	mark@gmail.com
	Christopher	Anderson	ChristopherAnderson@gmail.com
	Ronald	Clark	RonaldClark@gmail.com
	Mary	Wright	MaryWright@gmail.com
	Lisa	Mitchell	LisaMitchell@gmail.com
	Michelle	Johnson	MichelleJohnson@gmail.com
	John	Thomas	JohnThomas@gmail.com
	Daniel	Lopez	DanielRodriguez@gmail.com
	Anthony	Smith	AnthonyLopez@gmail.com
	Patricia	Perez	PatriciaPerez@gmail.com
	Mark	Antony	markanto@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, customers where orders.customerid = customers.customerid;
```

orderid	orderdate	firstname	lastname
1	2024-01-11	Patricia	Perez
4	2024-01-14	John	Thomas
5	2024-01-15	Michelle	Johnson
6	2024-01-16	Lisa	Mitchell
7	2024-01-17	Mary	Wright
8	2024-01-18	Ronald	Clark
9	2024-01-19	Christopher	Anderson
10	2024-01-20	James	Smith
11	2024-01-21	Mark	Antony
12	2024-01-21	Mark	Antony

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, "Mark", "Antony", "markanto@gmail.com", "1234567880", "Kolkatta");
select * from customers;
```

CustomerID	FirstName	LastName	Email	Phone	Address
1	James	Smith	mark@gmail.com	1234567890	pondichery
2	Christopher	Anderson	ChristopherAnderson@gmail.com	1234567891	Noida
3	Ronald	Clark	RonaldClark@gmail.com	1234567892	Pune
4	Mary	Wright	MaryWright@gmail.com	1234567893	Bangalore
5	Lisa	Mitchell	LisaMitchell@gmail.com	1234567894	Hydrabad
6	Michelle	Johnson	MichelleJohnson@gmail.com	1234567895	Delhi
7	John	Thomas	JohnThomas@gmail.com	1234567896	Kochi
8	Daniel	Lopez	DanielRodriguez@gmail.com	1234567897	Chennai
9	Anthony	Smith	AnthonyLopez@gmail.com	1234567898	Bangalore
10	Patricia	Perez	PatriciaPerez@gmail.com	1234567899	Noida
11	Mark	Antony	markanto@gmail.com	1234567000	Kolkatta

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

```
update products set price = price*1.1
where products.description = "electronic";
select * from products;
```

	ProductID	ProductName	Description	Price
▶	1	Phone	electronic	11000
	2	Laptop	electronic	22000
	3	PC	electronic	33000
	4	Earphone	Accessories	2000
	5	SmartWatch	electronic	4400
	6	PowerBank	electronic	1100
	7	Speaker	Accessories	2000
	8	RGBlights	Accessories	1000
	9	Cable	Accessories	500
	10	GPU	electronic	38500
	11	Music player	electronic	700

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.

```
delimiter @@
create procedure del(in val1 int)
    begin
        delete from orderdetails where orderdetails.orderid = val1;
        delete from orders where orders.orderid = val1;
    end@@
delimiter ;
call del(3);
```

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(11, 11, "2024-1-21", 33000);
insert into orderdetails values(11, 11, 3, 1);
```

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.

```
delimiter @@
create procedure up(in val0 int, val1 varchar(45), val2 varchar(100))
    begin
        update customers set email = val1 where customerid = val0;
        update customers set address = val2 where customerid = val0;
    end@@
delimiter ;
call up(1, "mark@gmail.com", "pondichery");
```

	CustomerID	FirstName	LastName	Email	Phone	Address
▶	1	James	Smith	mark@gmail.com	1234567890	pondichery

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 from orderdetails where orders.orderid = orderdetails.orderid)*
(select price from products where
(select productid from orderdetails where
orders.orderid = orderdetails.orderid) = products.productid);

select * from orders;
```

	OrderID	CustomerID	OrderDate	TotalAmount
▶	1	10	2024-01-11	11000
	4	7	2024-01-14	2000
	5	6	2024-01-15	4400
	6	5	2024-01-16	1100
	7	4	2024-01-17	2000
	8	3	2024-01-18	1000
	9	2	2024-01-19	500
	10	1	2024-01-20	38500
	11	11	2024-01-21	33000
	12	11	2024-01-21	33000

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.

```
delimiter @@
create procedure delful(in val1 int)
    begin
        delete from orderdetails where
        orderdetails.orderid =
        (select orderid from orders where customerid = val1);
        delete from orders where customerid = val1;
    end@@
delimiter ;
call delful(9);
```

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, "Music player", "electronic", 700);
select * from products;
```

ProductID	ProductName	Description	Price
1	Phone	electronic	11000
2	Laptop	electronic	22000
3	PC	electronic	33000
4	Earphone	Accessories	2000
5	SmartWatch	electronic	4400
6	PowerBank	electronic	1100
7	Speaker	Speaker s	2000
8	RGBlights	Accessories	1000
9	Cable	Accessories	500
10	GPU	electronic	38500
11	Music player	electronic	700

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.

```
delimiter @@
create procedure stat(in val1 int)
    begin
        select if(datediff((SELECT CURRENT_DATE) ,
        (select orderdate from orders where orderid = val1)) >= 3,
        "shipped", "pending") as status;
    end@@
delimiter ;
call stat(5);
```

status
shipped

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 numOrders int;
update customers set numOrders =
(select count(orderid) from orders where customers.customerid = orders.customerid);
select * from customers;
```

CustomerID	FirstName	LastName	Email	Phone	Address	numOrders
1	James	Smith	mark@gmail.com	1234567890	pondichery	1
2	Christopher	Anderson	ChristopherAnderson@gmail.com	1234567891	Noida	1
3	Ronald	Clark	RonaldClark@gmail.com	1234567892	Pune	1
4	Mary	Wright	MaryWright@gmail.com	1234567893	Bangalore	1
5	Lisa	Mitchell	LisaMitchell@gmail.com	1234567894	Hydrabad	1
6	Michelle	Johnson	MichelleJohnson@gmail.com	1234567895	Delhi	1
7	John	Thomas	JohnThomas@gmail.com	1234567896	Kochi	1
8	Daniel	Lopez	DanielRodriguez@gmail.com	1234567897	Chennai	0
9	Anthony	Smith	AnthonyLopez@gmail.com	1234567898	Bangalore	0
10	Patricia	Perez	PatriciaPerez@gmail.com	1234567899	Noida	1
11	Mark	Antony	markanto@gmail.com	1234567000	Kolkatta	2

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 orders.orderid, orders.orderdate, orders.totalamount,
customers.firstname, customers.lastname
from orders join customers on orders.customerid = customers.customerid;
```

orderid	orderdate	totalamount	firstname	lastname
1	2024-01-11	11000	Patricia	Perez
4	2024-01-14	2000	John	Thomas
5	2024-01-15	4400	Michelle	Johnson
6	2024-01-16	1100	Lisa	Mitchell
7	2024-01-17	2000	Mary	Wright
8	2024-01-18	1000	Ronald	Clark
9	2024-01-19	500	Christopher	Anderson
10	2024-01-20	38500	James	Smith
11	2024-01-21	33000	Mark	Antony
12	2024-01-21	33000	Mark	Antony

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 Products.productname, SUM(OrderDetails.quantity * products.price)
AS total_revenue
FROM Products JOIN OrderDetails
ON Products.productid = OrderDetails.productid
WHERE Products.description = "electronic" GROUP BY Products.productname;
```

productname	total_revenue
Phone	11000
PC	66000
SmartWatch	4400
PowerBank	1100
GPU	38500

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, email, phone
from customers join orders
on customers.customerid = orders.orderid;
```

	firstname	lastname	email	phone
▶	Patricia	Perez	PatriciaPerez@gmail.com	1234567899
	Anthony	Smith	AnthonyLopez@gmail.com	1234567898
	Daniel	Lopez	DanielRodriguez@gmail.com	1234567897
	John	Thomas	JohnThomas@gmail.com	1234567896
	Michelle	Johnson	MichelleJohnson@gmail.com	1234567895
	Lisa	Mitchell	LisaMitchell@gmail.com	1234567894
	Mary	Wright	MaryWright@gmail.com	1234567893
	James	Smith	mark@gmail.com	1234567890
	Mark	Antony	markanto@gmail.com	1234567000

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 products.productname, sum(OrderDetails.quantity) AS total_ordered
FROM Products JOIN OrderDetails
ON Products.productid = OrderDetails.productid
WHERE Products.description = "electronic" GROUP BY Products.productname
order by total_ordered desc limit 1;
```

	productname	total_ordered
▶	PC	2

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

```
select productname, description as categories
from products having description = "electronic";
```

	productname	categories
▶	Phone	electronic
	Laptop	electronic
	PC	electronic
	SmartWatch	electronic
	PowerBank	electronic
	GPU	electronic
	Music player	electronic

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 customers.firstname, avg(orders.totalamount) as avg_order_value
from customers join orders
on customers.customerid = orders.customerid group by customers.firstname;
```

	firstname	avg_order_value
▶	Patricia	11000.0000
	John	2000.0000
	Michelle	4400.0000
	Lisa	1100.0000
	Mary	2000.0000
	Ronald	1000.0000
	Christopher	500.0000
	James	38500.0000
	Mark	33000.0000

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 customers.firstname, customers.lastname, orders.totalamount
from customers join orders
on customers.customerid = orders.customerid
where totalamount =(select max(totalamount) from orders);
```

	firstname	lastname	totalamount
▶	James	Smith	38500

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

```
select products.productname, count(orderdetails.orderid) as num_of_orders
FROM Products JOIN OrderDetails
ON Products.productid = OrderDetails.productid
WHERE Products.description = "electronic" GROUP BY Products.productname;
```

	productname	num_of_orders
▶	Phone	1
	PC	2
	SmartWatch	1
	PowerBank	1
	GPU	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.

```
delimiter @@
create procedure find(in val1 varchar(45))
    begin
        select firstname from orders inner join customers
        on orders.customerid = customers.customerid inner join orderdetails
        on orders.orderid = orderdetails.orderid inner join products
        on orderdetails.productid = products.productid
        where products.productname = val1;
    end@@
delimiter ;
call find("pc");
```

	firstname
▶	Mark
	Mark

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.

```
delimiter @@
create procedure findrev(in val1 date, val2 date)
    begin
        select sum(totalamount) as revenue_gen
        from orders
        where orderdate between val1 and val2;
    end@@
delimiter ;
call findrev("2024-1-17", "2024-1-19");
```

	revenue_gen
▶	3500

Task 4. Subquery and its type:

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

```
select customers.firstname from customers
where customers.customerid not in
(select orders.customerid from orders);
```

	firstname
▶	Daniel
	Anthony

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

```sql
select products.productname, subquery.quantityinstock
from products, (select productid, quantityinstock from inventory)
as subquery
where products.productid = subquery.productid;
```

	productname	quantityinstock
▶	Phone	10
	Laptop	10
	PC	10
	Earphone	10
	SmartWatch	10
	PowerBank	10
	Speaker	10
	RGBlights	10
	Cable	10
	GPU	10

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

```sql
select sum(subquery.totalamount) as total_revenue_gen
from (select totalamount from orders) as subquery;
```

	total_revenue_gen
▶	126500

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.

```sql
delimiter @@
create procedure findcatavg(in val1 varchar(45))
   begin
       select description, avg(orderdetails.quantity) as avg
       from products, orderdetails
       where products.productid = orderdetails.productid
       and products.description = val1 group by products.description;
   end@@
delimiter ;
call findcatavg("electronic");
```

	description	avg
▶	electronic	1.0000

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.

```
delimiter @@
create procedure findrevcust(in val1 int)
   begin
       select customers.firstname, sum(totalamount)
       from customers, orders
       where customers.customerid = 7 and orders.customerid = 7;
   end@@
delimiter ;
call findrevcust(7);
```

	firstname	sum(totalamount)
▶	John	2000

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 firstname, count from (select customers.firstname, count(orders.orderid) as count
from customers ,orders
where customers.customerid = orders.customerid group by customers.firstname) as subquery
where count = (select max(count) from (select count(orderid) as count
from customers ,orders
where customers.customerid = orders.customerid group by customers.firstname)as maxcount);
```

	firstname	count
▶	Mark	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 description as category, count
from (select description, count(orderdetails.quantity) as count
from products, orderdetails where products.productid = orderdetails.productid
group by products.description) as subquery where count = (select max(count)
from (select count(orderdetails.quantity) as count from products, orderdetails
where products.productid = orderdetails.productid group by products.description) as couquery);
```

	category	count
▶	electronic	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 customers.firstname, total
from customers, (select customerid, sum(totalamount) as total
from (select customerid, totalamount from orders
where orderid in (select orderid from orderdetails
where productid in(select productid from products
where description = "electronic"))) as subquery group by subquery.customerid) as squery1
where customers.customerid = squery1.customerid and total = (select max(total)
from (select customerid, sum(totalamount) as total
from (select customerid, totalamount from orders
where orderid in (select orderid from orderdetails
where productid in(select productid from products
where description = "electronic"))) as subquery group by subquery.customerid) as squery1);
```

	firstname	total
▶	Mark	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(subquery.totalamount) as avg_order_val
from (select totalamount from orders) as subquery;
```

	avg_order_val
▶	12650.0000

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.

```
update customers set numOrders = (select count(orderid)
from orders where customers.customerid = orders.customerid);

select customers.firstname, numorders from customers;
```

	firstname	numorders
▶	James	1
	Christopher	1
	Ronald	1
	Mary	1
	Lisa	1
	Michelle	1
	John	1
	Daniel	0
	Anthony	0
	Patricia	1
	Mark	2