**Case Study: Online Retail Store**

1. **Write SQL statements to create the tables as described in the database schema.**

**Create database retailstore**

create database retailstore

use retailstore

**Customers Table:**

create table customer(

custid INT Primary Key,

firstname VARCHAR(100),

Lastname VARCHAR(100),

email VARCHAR(100),

phone varchar(20),

regdate DATE

)

**Products Table:**

create table product(

prodid INT Primary Key,

productame VARCHAR(100),

category VARCHAR(100),

Price DECIMAL,

StockQuantity INT

)

**Orders Table:**

create table orders(

orderid INT Primary Key,

orderdate DATE,

custid INT,

totalamount DECIMAL,

foreign key (custid) references customer(custid)

)

**OrderDetails Table:**

create table orderdetails(

detail\_id INT Primary Key,

orderid INT,

prodid INT,

quantity INT,

price DECIMAL,

foreign key (orderid) references orders(orderid),

foreign key (prodid) references product(prodid)

)

1. **Insert Data**

**Customers**

insert into customer(custid,firstname,lastname,email,phone,regdate)

values(1,'John', 'Doe', 'john.doe@example.com', '123-456-7890', '2023-01-15'),

(2,'Jane', 'Smith', 'jane.smith@example.com', '234-567-8901', '2023-02-20'),

(3,'Alice', 'Johnson', 'alice.johnson@example.com', '345-678-9012', '2023-03-10'),

(4,'Bob', 'Brown', 'bob.brown@example.com', '456-789-0123', '2023-04-05'),

(5,'Charlie', 'Davis', 'charlie.davis@example.com', '567-890-1234', '2023-05-12'),

(6,'David', 'Wilson', 'david.wilson@example.com', '678-901-2345', '2023-06-15'),

(7,'Emma', 'Thomas', 'emma.thomas@example.com', '789-012-3456', '2023-07-01'),

(8,'Fiona', 'Garcia', 'fiona.garcia@example.com', '890-123-4567', '2023-07-10'),

(9,'George', 'Martinez', 'george.martinez@example.com', '901-234-5678', '2023-07-20'),

(10,'Hannah', 'Rodriguez', 'hannah.rodriguez@example.com', '012-345-6789', '2023-07-25')

**Products**

insert into product(prodid,productame,category,price,StockQuantity)

values(20,'Laptop', 'Electronics', 999.99, 50),

(21,'Smartphone', 'Electronics', 499.99, 100),

(22,'Tablet', 'Electronics', 299.99, 75),

(23,'Headphones', 'Accessories', 49.99, 200),

(24,'Charger', 'Accessories', 19.99, 300),

(25,'Keyboard', 'Accessories', 29.99, 150),

(26,'Mouse', 'Accessories', 19.99, 250),

(27,'Monitor', 'Electronics', 199.99, 30),

(28,'Printer', 'Electronics', 149.99, 20),

(29,'USB Cable', 'Accessories', 9.99, 400)

**Orders**

insert into orders(orderid,orderdate,custid,totalamount)

values (43,'2023-06-01', 1, 1049.98),

(31,'2023-06-05', 2, 549.98),

(32,'2023-06-10', 3, 999.99),

(33,'2023-06-15', 4, 69.98),

(34,'2023-06-20', 5, 519.98),

(35,'2023-06-25', 6, 229.98),

(36,'2023-07-02', 7, 1199.97),

(37,'2023-07-12', 8, 49.98),

(38,'2023-07-18', 9, 349.98),

(39,'2023-07-22', 10, 39.98);

**Order details**

insert into orderdetails(detail\_id,orderid,prodid,quantity,price)

values (100,31, 21, 1, 999.99),

(102,31, 24, 1, 49.99),

(103,32, 22, 1, 499.99),

(104,32,25, 1, 49.99),

(105,43, 21, 1, 999.99),

(106,34, 24, 1,49.99),

(107,34, 25, 1, 19.99),

(108,35, 22, 1, 499.99),

(109,35, 25, 1, 19.99),

(110,36, 23, 1, 199.99),

(111,36, 25, 1, 29.99),

(112,37, 21, 1, 999.99),

(113,37, 23, 1, 199.99),

(114,34, 21,4, 49.99),

(115,38, 27, 1, 19.99),

(116,38, 28, 1, 29.99),

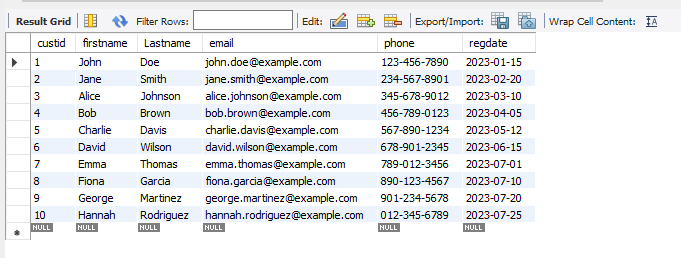
(117,39, 24, 2,149.99),

(118,39, 29, 1, 49.99),

(119,30, 20, 4, 9.99)

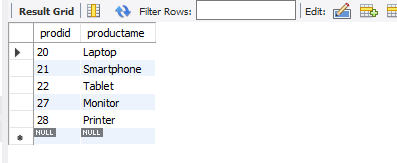
1. **Retrieve all customers who registered in 2023**

select \* from customer where year(regdate)=2023



1. **List all products in the 'Electronics' category**

select prodid,productame from product where category='Electronics'

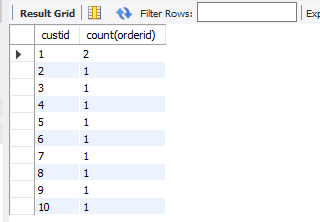


1. **Find the total number of orders placed by each customer.**

select custid,count(orderid)

from orders

group by custid

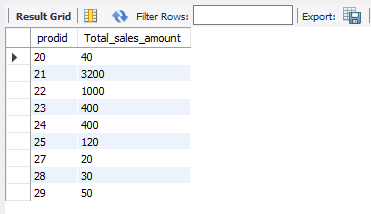
****

1. **Calculate the total sales amount for each product.**

select prodid,sum(price\*quantity) as Total\_sales\_amount

from orderdetails

group by prodid

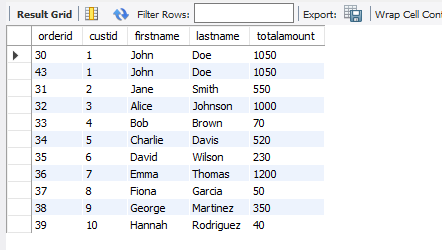


1. **Retrieve the details of all orders, including the customer name and total amount**

select orders.orderid,orders.custid,c.firstname,lastname,totalamount

from orders

inner join customer c on orders.custid=c.custid



1. **List all products that have been ordered along with the quantity ordered for each.**

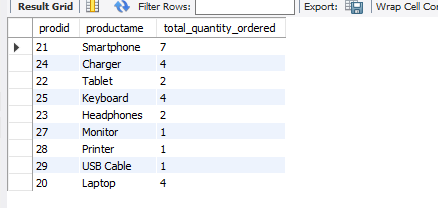
SELECT p.prodid, p.productame, SUM(od.quantity) AS total\_quantity\_ordered

FROM product p

JOIN orderdetails od ON p.prodid = od.prodid

JOIN orders o ON od.orderid = o.orderid

GROUP BY p.prodid



1. **Find the order details for orders placed by 'John Doe'.**

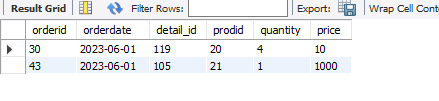
SELECT o.orderid, o.orderdate, od.detail\_id, od.prodid, od.quantity, od.price

FROM customer c

JOIN orders o ON c.custid = o.custid

JOIN orderdetails od ON o.orderid = od.orderid

WHERE c.firstname = 'John' AND c.lastname = 'Doe';



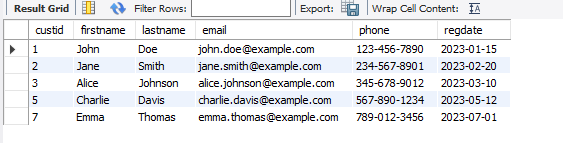
1. **Find customers who have placed an order totaling more than $500.**

SELECT DISTINCT c.custid, c.firstname, c.lastname, c.email, c.phone, c.regdate

FROM customer c

JOIN orders o ON c.custid = o.custid

WHERE o.totalamount > 500;



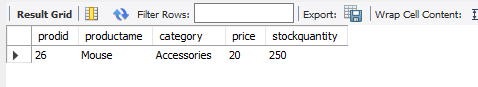
1. **List the products that have never been ordered.**

SELECT p.prodid, p.productame, p.category, p.price, p.stockquantity

FROM product p

LEFT JOIN orderdetails od ON p.prodid = od.prodid

WHERE od.prodid IS NULL;



1. **Retrieve the order history for a specific customer, including order date, product names,**

**and quantities. (Assume customer ID is 2)**

SELECT o.orderdate, p.productame, od.quantity

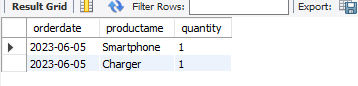
FROM customer c

JOIN orders o ON c.custid = o.custid

JOIN orderdetails od ON o.orderid = od.orderid

JOIN product p ON od.prodid = p.prodid

WHERE c.custid = 2;

****

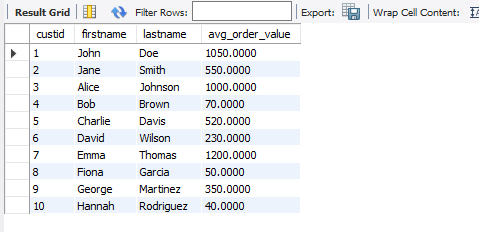
1. **Calculate the average order value for each customer.**

SELECT c.custid, c.firstname, c.lastname, AVG(o.totalamount) AS avg\_order\_value

FROM customer c

JOIN orders o ON c.custid = o.custid

GROUP BY c.custid, c.firstname, c.lastname;



1. **Find the most popular product category based on the number of orders**

SELECT p.category, COUNT(o.orderid) AS num\_orders

FROM product p

JOIN orderdetails od ON p.prodid = od.prodid

JOIN orders o ON od.orderid = o.orderid

GROUP BY p.category

ORDER BY num\_orders DESC

LIMIT 1;



1. **List all customers who have ordered more than one product in a single order.**

SELECT c.custid, c.firstname, c.lastname, COUNT(od.prodid) AS product\_count

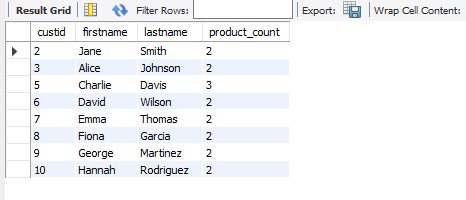
FROM customer c

JOIN orders o ON c.custid = o.custid

JOIN orderdetails od ON o.orderid = od.orderid

GROUP BY c.custid, c.firstname, c.lastname, o.orderid

HAVING COUNT(od.prodid) > 1;



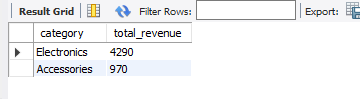
1. **Find the total revenue generated from each product category**

SELECT p.category,SUM(od.quantity \* od.price) AS total\_revenue

FROM product p

JOIN orderdetails od ON p.prodid = od.prodid

GROUP BY p.category;



1. **Retrieve the list of customers along with the total amount they have spent.**

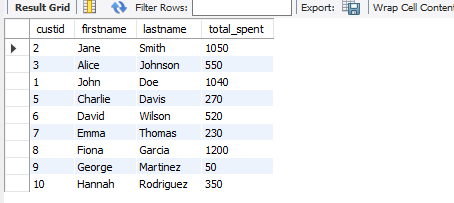
SELECT c.custid,c.firstname,c.lastname,SUM(od.quantity \* od.price) AS total\_spent

FROM customer c

JOIN orders o ON c.custid = o.custid

JOIN orderdetails od ON o.orderid = od.orderid

GROUP BY c.custid, c.firstname, c.lastname;



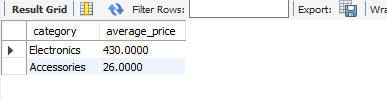
1. **Find the average price of products in each category**

SELECT category,

AVG(price) AS average\_price

FROM product

GROUP BY category;



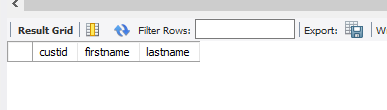
1. **Find all customers who have not placed any orders.**

SELECT c.custid,c.firstname,c.lastname

FROM customer c

LEFT JOIN orders o ON c.custid = o.custid

WHERE o.orderid IS NULL;



1. **List the top 3 products with the highest total sales amount.**

SELECT p.prodid,p.productame,SUM(od.quantity \* od.price) AS total\_sales

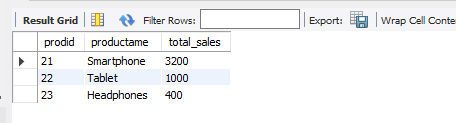
FROM product p

JOIN orderdetails od ON p.prodid = od.prodid

GROUP BY p.prodid, p.productame

ORDER BY total\_sales DESC

LIMIT 3;



1. **Find customers who have placed orders for more than 3 different products**

SELECT c.custid,c.firstname,c.lastname,COUNT(DISTINCT od.prodid) AS product\_count

FROM customer c

JOIN orders o ON c.custid = o.custid

JOIN orderdetails od ON o.orderid = od.orderid

GROUP BY c.custid, c.firstname, c.lastname

HAVING COUNT(DISTINCT od.prodid) > 3;

