ASSIGNMENT 2

PART 1

```
show databases;

create database ordersdb;

use ordersdb;

create table Customers(

CustomerID int primary key,

Name varchar(100),

Email varchar(100),

City varchar(50),

Signupdate date

);


insert into Customers values

(1, 'Alice Johnson', 'alice@abc.com', 'Mumbai', '2024-01-15'),

(2, 'John Mathew', 'john@djd.com', 'Kochi', '2024-02-10'),

(3, 'Rakesh Sharma', 'rakesh@abc.com', 'Delhi', '2023-12-20'),

(4, 'Sandeep Reddy', 'sandeep@aaa.com', 'Mumbai', '2024-03-05');


create table Orders(

OrderID int primary key,

CustomerID int,

OrderDate date,
```

```
TotalAmount decimal(10,2),
foreign key (CustomerID) references Customers(CustomerID)
);

insert into Orders values
(101, 1, curdate() - interval 10 day, 5500.00),
(102, 2, curdate() - interval 40 day, 2300.00),
(103, 1, '2024-03-01', 1300.00);

create table Products (
   ProductID int primary key,
   ProductName varchar(100),
   Category varchar(50),
   Price decimal(10,2)
);

insert into Products values
(1001, 'Wireless Mouse', 'Electronics', 25.00),
(1002, 'Bluetooth Speaker', 'Electronics', 55.50),
(1003, 'Notebook', 'Stationery', 3.99),
(1004, 'Ball Pen', 'Stationery', 1.49),
(1005, 'Power Bank', 'Electronics', 1200.00);
```

```
create table OrderDetails (
   OrderDetailID int primary key,
   OrderID int,
   ProductID int,
   Quantity int,
   Price decimal(10,2),
   foreign key (OrderID) references Orders(OrderID),
   foreign key (ProductID) references Products(ProductID)
);



insert into OrderDetails values
(1, 101, 1001, 2, 25.00),
(2, 101, 1002, 1, 55.50),
(3, 102, 1003, 10, 3.99),
(4, 103, 1004, 5, 1.49),
(5, 103, 1003, 2, 3.99);
```

PART 2

1. select * from Customers;
2. select * from Orders where OrderDate >= curdate() - interval 30 day;
3. select ProductName, Price from Products;
4. select Category, count(*) as TotalProducts from Products group by Category;

5. select * from Customers where City = 'Mumbai';
6. select * from Orders where TotalAmount > 5000;
7. select * from Customers where SignupDate > '2024-01-01';

8. select o.OrderID, o.OrderDate, o.TotalAmount, c.Name
   from Orders o
   join Customers c on o.CustomerID = c.CustomerID;
9. select od.OrderID, p.ProductName, od.Quantity, od.Price
   from OrderDetails od
   join Products p on od.ProductID = p.ProductID;

10. select *
    from Customers
    where CustomerID not in (
    select CustomerID from Orders
    );

11. select c.CustomerID, c.Name, sum(o.TotalAmount) as TotalSpent
    from Customers c
    join Orders o on c.CustomerID = o.CustomerID
    group by c.CustomerID, c.Name;

12. select p.ProductID, p.ProductName, sum(od.Quantity) as TotalQuantitySold
    from Products p
    join OrderDetails od on p.ProductID = od.ProductID
    group by p.ProductID, p.ProductName
    order by TotalQuantitySold desc
    limit 1;

13. select c.CustomerID, c.Name, avg(o.TotalAmount) as AvgOrderValue
    from Customers c
    join Orders o on c.CustomerID = o.CustomerID
    group by c.CustomerID, c.Name;

14. select p.Category, sum(od.Quantity * od.Price) as TotalSales
    from Products p
    join OrderDetails od on p.ProductID = od.ProductID
    group by p.Category;

15. select c.CustomerID, c.Name
    from Customers c
    join Orders o on c.CustomerID = o.CustomerID
    group by c.CustomerID, c.Name
    having sum(o.TotalAmount) > (
    select avg(total) from (
    select sum(TotalAmount) as total
    from Orders
    group by CustomerID
) as avg_table
    );

16. select * from Products
    where ProductID not in (
    select distinct ProductID from OrderDetails
    );

17. select o.*
    from Orders o
    join (
    select CustomerID, max(OrderDate) as LatestOrder
    from Orders
    group by CustomerID
) latest
    on o.CustomerID = latest.CustomerID and o.OrderDate = latest.LatestOrder;

18. select c.CustomerID, c.Name, sum(o.TotalAmount) as TotalSpent,
    rank() over (order by sum(o.TotalAmount) desc) as SpendingRank
    from Customers c
    join Orders o on c.CustomerID = o.CustomerID
    group by c.CustomerID, c.Name;

19. select c.CustomerID, c.Name, count(o.OrderID) as OrderCount
    from Customers c
    join Orders o on c.CustomerID = o.CustomerID
    group by c.CustomerID, c.Name
    order by OrderCount desc
    limit 3;

20. select p.ProductID, p.ProductName, count(distinct o.CustomerID) as
    UniqueCustomers
    from Products p
    join OrderDetails od on p.ProductID = od.ProductID
    join Orders o on od.OrderID = o.OrderID
    group by p.ProductID, p.ProductName;