1.SELECT * FROM Customers;

2.SELECT *
FROM Employees
ORDER BY salary DESC
LIMIT 5;

3.Where and having clause is used to filter the data, where clause is used to filter single row level of data.

```
SELECT customer_id, COUNT(*) AS order_count
FROM orders
GROUP BY customer_id
HAVING order_count > 3;
```

having clause is used to print data with contain aggregate function.

```
SELECT customer_id, SUM(order_total) AS total_orders
FROM orders
GROUP BY customer_id
HAVING total_orders > 1000;
```

4.SELECT AVG(salary) AS average_salary
FROM EmployeeDetails;

5.SELECT customer_id, SUM(order_amount) AS total_order_amount
FROM Orders
GROUP BY customer_id;

6.SELECT Name, Age
FROM Employees;

7.SELECT MONTH(Date) AS Month, YEAR(Date) AS Year, SUM(Amount) AS TotalSales
FROM Sales
GROUP BY YEAR(Date), MONTH(Date)
ORDER BY Year, Month;

8.Inner join.
Inner join will be return the record that having pair on both sides.

```
SELECT Employees.Name, Departments.DepartmentName
FROM Employees
INNER JOIN Departments
ON Employees.DepartmentID = Departments.ID;
```

Left join.
If we need all record from left table no matter if they have pair in the right table

```
SELECT Employees.Name, Departments.DepartmentName
FROM Employees
LEFT JOIN Departments
ON Employees.DepartmentID = Departments.ID;
```

9.

```
SELECT DISTINCT Salary AS SecondHighestSalary
FROM Employees
WHERE Salary = (
   SELECT MAX(Salary)
   FROM Employees
   WHERE Salary < (
      SELECT MAX(Salary)
      FROM Employees
   )
);
```

10.

```
ALTER TABLE Customers
ADD City VARCHAR(50);
```

11.

```
SELECT Customers.Name
FROM Customers
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID
WHERE Orders.CustomerID IS NULL;
```

12.The COUNT() function is used to count the number of rows or records that match a specified condition or column.

The SUM() function is used to calculate the sum of numeric values in a specific column.

The AVG() function is used to calculate the average (mean) value of numeric data in a specific column.

13.

```
SELECT Category, COUNT(Category) AS Frequency
FROM Products
GROUP BY Category
ORDER BY Frequency DESC
LIMIT 3;
```

14.

```
UPDATE Inventory
SET Quantity = Quantity * 2
```

```
WHERE Category = 'Electronics';
```

15.The UNION operator is used to combine and return distinct rows from the result sets of multiple SELECT queries.

```
SELECT column1, column2, ...
FROM table1
UNION
SELECT column1, column2, ...
FROM table2;
```

The UNION ALL operator is used to combine and return all rows from the result sets of multiple SELECT queries.

```
SELECT column1, column2, ...
FROM table1
UNION ALL
SELECT column1, column2, ...
FROM table2;
```

16.UPDATE Contacts
SET Email = REPLACE(Email, '@old.com', '@new.com')
WHERE Email LIKE '%@old.com';

17.SELECT CustomerName
FROM Orders
GROUP BY CustomerName
HAVING COUNT(*) >= 2;

18.A primary key is a column (or a set of columns) in a database table that uniquely identifies each row or record within that table. It ensures that every value in the primary key column(s) is unique across all rows in the table.

A foreign key is a column (or a set of columns) in one table that establishes a link between the data in that table and the data in another table. It defines a relationship between the tables, indicating that the values in the foreign key column(s) in one table correspond to the values in the primary key column(s) of another table.

19.DELETE FROM Products

```
WHERE Category = 'Books';

20.SELECT CustomerID, COUNT(*) AS TotalOrders
FROM Orders
GROUP BY CustomerID;
```