1. Math Functions:

- Math functions are used to perform mathematical operations on numeric data. They are commonly used for calculations, such as addition, subtraction, multiplication, division, and more.

  - `SUM()`: Calculates the sum of a set of values.

  - `AVG()`: Calculates the average of a set of values.

  - `MIN()`: Finds the minimum value in a set of values.

  - `MAX()`: Finds the maximum value in a set of values.

  - `COUNT()`: Counts the number of rows or non-null values.

  - `ABS()`: Returns the absolute value of a number.

  - `ROUND()`: Rounds a number to a specified number of decimal places.

  - `SQRT()`: Calculates the square root of a number.

2. String Functions:

- String functions are used for manipulating and working with text data. They allow you to perform operations on strings, such as concatenation, conversion, formatting, and extraction.

  - `CONCAT()`: Combines two or more strings together.

  - `LENGTH()` or `LEN()`: Returns the length of a string.

  - `UPPER()` or `UCASE()`: Converts a string to uppercase.

  - `LOWER()` or `LCASE()`: Converts a string to lowercase.

  - `SUBSTRING()` or `SUBSTR()`: Extracts a portion of a string.

  - `LEFT()`: Returns a specified number of characters from the beginning of a string.

  - `RIGHT()`: Returns a specified number of characters from the end of a string.

  - `TRIM()`: Removes leading and trailing spaces from a string.

  - `REPLACE()`: Replaces occurrences of a substring with another substring in a string.

3. Aggregate Functions:

- Aggregate functions are used to perform operations on sets of values, typically used with the `GROUP BY` clause to group data based on one or more columns.

  - `SUM()`: Calculates the sum of a set of values within a group.

  - `AVG()`: Calculates the average of a set of values within a group.

  - `MIN()`: Finds the minimum value within a group.

  - `MAX()`: Finds the maximum value within a group.

  - `COUNT()`: Counts the number of rows within a group.

  - `GROUP_CONCAT()`: Concatenates values within a group into a single string.

  - `FIRST()` or `LAST()`: Returns the first or last value within a group.

These functions are an essential part of SQL and are used for a wide range of data manipulation tasks, making it possible to retrieve, calculate, and transform data in a database. The choice of function depends on the specific operation you need to perform.

1. Counting the Number of Rows:

```
   SELECT COUNT(*) FROM employee;
```

   - Explanation: This query counts the total number of rows (employees) in the "employee" table and returns that count as a single value.

2. Calculating the Average Salary for All Employees:

```
   SELECT AVG(salary) FROM employee;
```

   - Explanation: This query calculates and returns the average salary of all employees in the "employee" table.

3. Calculating the Average Salary for Managers:

```
   SELECT AVG(salary) FROM employee WHERE job_desc = 'MANAGER';
```

   - Explanation: This query calculates and returns the average salary of employees with the job description 'MANAGER.'

4. Calculating the Total Salary for Analysts:

```
   SELECT SUM(salary) FROM employee WHERE job_desc = 'ANALYST';
```

   - Explanation: This query calculates and returns the total salary of employees with the job description 'ANALYST.'

5. Selecting Employee(s) with the Highest Salary:

```
   SELECT * FROM employee WHERE salary = (SELECT MAX(salary) FROM employee);
```

   - Explanation: This query retrieves the employee(s) with the highest salary. It uses a subquery to find the maximum salary in the "employee" table and then selects the employee(s) with that maximum salary.

6. Finding the Minimum Salary:

```
   SELECT MIN(salary) FROM employee;
```

- Explanation: This query calculates and returns the minimum salary among all employees in the "employee" table.

7. Converting Employee Names to Uppercase:

```
SELECT UCASE(ename), salary FROM employee;
```

- Explanation: This query selects the employee names in uppercase (uppercase function) along with their salaries from the "employee" table.

8. Finding the Character Length of Employee Names:

```
SELECT ename, CHAR_LENGTH(ename) FROM employee;
```

- Explanation: This query retrieves the employee names and calculates the character length of each name, displaying the result alongside the names.

9. Concatenating "Rs." Prefix to Salaries:

```
SELECT ename, CONCAT("Rs.", salary) FROM employee;
```

- Explanation: This query combines the "Rs." prefix with each employee's salary and displays the result alongside the employee names.

10. Formatting Salaries with "Rs." Prefix and Comma Separators:

```
SELECT ename, CONCAT("Rs.", FORMAT(salary, 0)) FROM employee;
```

- Explanation: This query formats salaries with a "Rs." prefix and includes commas as thousand separators, displaying the result alongside the employee names.

11. Extracting the First 4 Characters of Job Descriptions:

```
SELECT ename, LEFT(job_desc, 4) FROM employee;
```

- Explanation: This query retrieves employee names and extracts the first 4 characters of their job descriptions (using the LEFT function), displaying the result alongside the names.