# AGGREGATE FUNCTIONS

**Aggregate Functions**

aggregate function is used to perform calculations on multiple values and return the result in a single value like the average of all values, the sum of all values, and maximum & minimum value among certain groups of values.

There are various aggregation functions supported in MySQL. They are:

| Sl. No | Aggregate Function | Descriptions |
|---|---|---|
| 1 | COUNT() | It returns the number of rows, including rows with NULL values in a group. |
| 2 | SUM() | It returns the total summed values (Non-NULL) in a set. |
| 3 | AVERAGE() | It returns the average value of an expression. |
| 4 | MIN() | It returns the minimum (lowest) value in a set. |
| 5 | MAX() | It returns the maximum (highest) value in a set. |
| 6 | GROUP_CONCAT() | It returns a concatenated string. |
| 7 | FIRST() | It returns the first value of an expression. |
| 8 | LAST() | It returns the last value of an expression. |

Let us take an exmple table called **EMPLOYEE** having columns **name**, **occupation**, **working**_date and **working**_hours. Let us create the table by using the following statement:

```
CREATE TABLE employee(
    name varchar(45) NOT NULL,
    occupation varchar(35) NOT NULL,
    working_date date,
    working_hours varchar(10)
    );
```

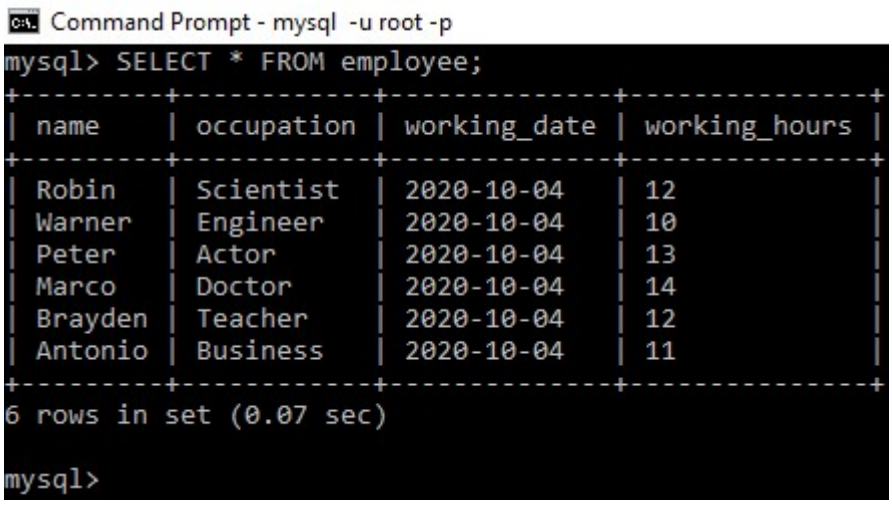
Now, let us insert following records into this table EMPLOYEE by using following statements:

```
INSERT INTO employee VALUES
('Robin', 'Scientist', '2020-10-04', 12),
('Warner', 'Engineer', '2020-10-04', 10),
('Peter', 'Actor', '2020-10-04', 13),
('Marco', 'Doctor', '2020-10-04', 14),
('Brayden', 'Teacher', '2020-10-04', 12),
('Antonio', 'Business', '2020-10-04', 11);
```

Now, we can see the recods of the table EMPLOYEE by using SELECT * statement as below:

Statement: **SELECT * FROM employee;**

**Output:**




Now, let us use the aggregate functions on this table EMPLOYEE.

**1) Use of COUNT() Function**

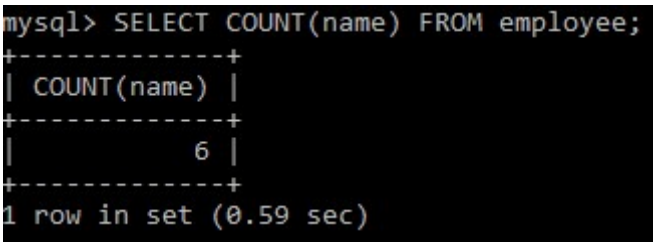
MySQL count() function returns **the total number of values** in the expression. This function produces all rows or only some rows of the table based on a specified condition, and its return type is BIGINT (**size 8 byte, maximum value can be assigned upto $2^{63}$-1**). It returns zero if it does not find any matching rows. It can work with both **numeric** and **non-numeric** data types.

**Example:** Suppose we want to get the total number of employees in the employee table, we need to use the count() function as shown in the following query:

**Statement**:

**mysql> SELECT COUNT(name) FROM employee;**

**Output:**




**i.e. total number of employee in the EMPLOYEE table is 6**

**2) Use of SUM() Function**

The MySQL **sum**() function **returns the total summed (non-NULL)** value of an expression. It returns **NULL** if the result set does not have any rows. It works with **numeric** data type only.

**Example:** Suppose we want to calculate the total number of working hours of all employees in the table, we need to use the sum() function as shown in the following query:

**Statement:**

**SEELECT SUM(working_hours) AS "Total working hours" FROM employee;**

**Output:**

```
mysql> SELECT SUM(working_hours) AS "Total working hours" FROM employee;
+---------------------+
| Total working hours |
+---------------------+
|                  72 |
+---------------------+
1 row in set (0.26 sec)

mysql>
```

**i.e. total number of working hours of all the employees in the EMPLOYEE table is 72**

**3) Use of AVG() Function**

MySQL AVG() function **calculates the average of the values** specified in the column. Similar to the SUM() function, it also works with numeric data type only.

**Example:** Suppose we want to get the average working hours of all employees in the table, we need to use the AVG() function as shown in the following query:

**Statement:**

**SELECT AVG(working_hours) AS "Average working hours" FROM employee;**

**Output:**

```
mysql>  SELECT AVG(working_hours) AS "Average working hours" FROM employee;
+-----------------------+
| Average working hours |
+-----------------------+
|                    12 |
+-----------------------+
1 row in set (0.06 sec)
```

**i.e. Average number of working hours of all the employees is 12**

**4) Use of MIN() Function**

MySQL MIN() function **returns the minimum (lowest) value** of the specified column. It also works with numeric data type only.

**Example:** Suppose we want to get minimum working hours of an employee available in the table, we need to use the MIN() function as shown in the following query:

**Statement:**

**SELECT MIN(working_hours) AS Minimum_working_hours FROM employee;**

**Output:**

```
mysql> SELECT MIN(working_hours) AS Minimum_working_hours FROM employee;
+-----------------------+
| Minimum_working_hours |
+-----------------------+
| 10                    |
+-----------------------+
1 row in set (0.08 sec)
```

**i.e. the minimum working hours of all the employees is 10**

**5) Use of MAX() Function**

MySQL MAX() function **returns the maximum (highest) value** of the specified column. It also works with numeric data type only.

**Example:** Suppose we want to get maximum working hours of an employee available in the table, we need to use the MAX() function as shown in the following query:

**Statement:**

**SELECT MIN(working_hours) AS Minimum_working_hours FROM employee;**

**Output:**

```
mysql> SELECT MAX(working_hours) AS Minimum_working_hours FROM employee;
+-----------------------+
| Minimum_working_hours |
+-----------------------+
| 14                    |
+-----------------------+
1 row in set (0.00 sec)
```

**i.e. the maximum working hours of all the employees is 10**

**6) Use of FIRST() Function**

SQL FIRST() function **returns the first value of the specified column**. To get the first value of the column, we must have to use the LIMIT clause. It is because FIRST() function only supports in MS Access.

**Example:** Suppose we want to get the first working date of an employee available in the table, we need to use the following query:

**Statement:**

```
SELECT working_date FROM employee LIMIT 1;
```

**Output:**

```
mysql> SELECT working_date FROM employee LIMIT 1;
+--------------+
| working_date |
+--------------+
| 2020-10-04   |
+--------------+
1 row in set (0.06 sec)
```

**7) Use of LAST() Function**

SQL LAST() function **returns the last value of the specified column.** To get the last value of the column, we must have to use the ORDER BY and LIMIT clause. It is because the LAST() function only supports in MS Access.

**Example:** Suppose we want to get the last working hour of an employee available in the table, we need to use the following query:

**Statement:**

```
SELECT working_hours FROM employee ORDER BY name DESC LIMIT 1;
```

**Output:**

```
mysql> SELECT working_hours FROM employee ORDER BY name DESC LIMIT 1;
+---------------+
| working_hours |
+---------------+
| 10            |
+---------------+
1 row in set (0.05 sec)
```

**8) Use of GROUP_CONCAT() Function**

The **GROUP_CONCAT**() function **returns the concatenated string from multiple rows** into a single string. If the group contains at least one non-null value, it always returns a string value. Otherwise, we will get a null value.

Suppose we have another employee table as below:

| emp_id | emp_fname | emp_lname | dept_id | designation |
|---|---|---|---|---|
| 1 | David | Miller | 2 | Engineer |
| 2 | Peter | Watson | 3 | Manager |
| 3 | Mark | Boucher | 1 | Scientist |
| 2 | Peter | Watson | 3 | BDE |
| 1 | David | Miller | 2 | Developer |
| 4 | Adam | Warner | 4 | Receptionist |
| 3 | Mark | Boucher | 1 | Engineer |
| 4 | Adam | Warner | 4 | Clerk |

**Example:** Suppose we want to concatenate the designation of the same **dept**_id on the employee table, we need to use the following query:

**Statement:**

**SELECT emp_id, emp_fname, emp_lname, dept_id, GROUP_CONCAT(designation) as "designation" FROM employee GROUP BY emp_id;**

**Output:**

| emp_id | emp_fname | emp_lname | dept_id | designation |
|---|---|---|---|---|
| 1 | David | Miller | 2 | Engineer,Developer |
| 2 | Peter | Watson | 3 | Manager,BDE |
| 3 | Mark | Boucher | 1 | Scientist,Engineer |
| 4 | Adam | Warner | 4 | Receptionist,Clerk |