## 2. COUNT

This function counts the number of rows in a given column or expression.

*Syntax*

```
COUNT({ * [DISTINCT | ALL] expr})
```

- Returns the number of rows in the query.
- If you specify argument *expr*, this function returns rows where *expr* is not null. You can count either all rows, or only distinct values of *expr*.
- If you specify the asterisk (*), this function returns all rows, including duplicates and nulls.

*Argument type* : Numeric  *Return value* : Numeric

Example 8.2 *Count number of records in table empl.*

*Solution.*

```
mysql> SELECT COUNT(*) "Total"
       FROM empl ;
```

```
+----------+
| Total    |
+----------+
|   14     |
+----------+
1 row in set (0.00 sec)
```

Example 8.3 *Count number of jobs in table empl.*

*Solution.*

```
mysql> SELECT COUNT(job) "Job Count"
       FROM empl ;
```

```
+-------------+
| Job Count   |
+-------------+
|     14      |
+-------------+
1 row in set (0.01 sec)
```

Example 8.4 *How many distinct jobs are listed in table empl ?*

*Solution.*

```
mysql> SELECT COUNT(DISTINCT job) "Distinct Jobs"
       FROM empl ;
```

```
+----------------+
| Distinct Jobs  |
+----------------+
|       15       |
+----------------+
1 row in set (0.04 sec)
```

## 3. MAX

This function returns the maximum value from a given column or expression.

*Syntax*

```
MAX([DISTINCT|ALL] expr)
```

- Returns maximum value of argument *expr*.

*Argument type* : Numeric  *Return value* : Numeric

Example 8.5 *Display maximum salary from table empl.*

*Solution.*

```
mysql> SELECT MAX(sal) "Maximum Salary"
       FROM empl ;
```

```
+----------------+
| Maximum Salary |
+----------------+
|    5000.00     |
+----------------+
1 row in set (0.01 sec)
```

Scanned with CamScanner

### 4. MIN

This function returns the minimum value from a given column or expression.

*Syntax*

```
MIN([DISTINCT | ALL] expr)
```

▲ Returns minimum value of *expr*.

*Argument type* : Numeric  *Return value* : Numeric

Example 8.6 *Display the joining date of seniormost employee.*

*Solution.*

```
mysql> SELECT MIN(hiredate) "Minimum Hire Date"
       FROM empl ;
```

```
+--------------------+
| Minimum Hire Date  |
+--------------------+
|     1990-12-18     |
+--------------------+
1 row in set (0.06 sec)
```

### 5. SUM

This function returns the sum of values in given column or expression.

*Syntax*

```
SUM([DISTINCT | ALL] n)
```

▲ Returns sum of values of *n*.

*Argument type* : Numeric  *Return value* : Numeric

Example 8.7 *Display total salary of all employees listed in table empl.*

*Solution.*

```
mysql>  SELECT SUM(sal) "Total Salary"
        FROM empl ;
```

```
+---------------+
|  Total Salary |
+---------------+
|    29035.00   |
+---------------+
1 row in set (0.01 sec)
```

Some more examples of group functions are being given below :

**NOTE**

These functions are called *aggregate functions* because they operate on aggregates of tuples. The result of an aggregate function is a single value.

Examples :

1. To calculate the total gross for employees of grade 'E2', the command is :

```
SELECT SUM(gross) FROM employee
WHERE grade = 'E2' ;
```

2. To display the average gross of employees with grades 'E1' or 'E2', the command used is :

```
SELECT AVG(gross) FROM employee
WHERE (grade = 'E1' OR grade = 'E2') ;
```

3. To count the number of employees in *employee* table, the *SQL* command is :

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

4. To count the number of cities, the different members belong to, you use the following command :

```
SELECT COUNT(DISTINCT city)  FROM members ;
```

Scanned with CamScanner

Here the DISTINCT keyword ensures that multiple entries of the same *city* are ignored. The * is the only argument that includes NULLs when it is used only with COUNT, functions other than COUNT disregard NULLs in any case.

If you want to count the entries including repeats, the keyword ALL is used. The following command will COUNT the number of non NULL *city* fields in the *members* table :

```
SELECT COUNT(ALL city)
FROM members ;
```

In general, GROUP functions

- Return a single value for a set of rows.
- Can be applied to any numeric values, and some Text types and DATE values.

## 8.4 Types of SQL Functions

Now that you have learnt different types of functions, let us talk about their broad categories. SQL supports many and many functions. All these functions can be generally categorized into following *two* types :

- Single Row (or Scalar) functions.
- Multiple Row (or Group or Aggregate) functions.

(*i*) ***Single Row functions*** work with a single row at a time. A single row function returns a result for every row of a queried table.

(*ii*) ***Multiple Row or Group functions*** work with data of multiple rows at a time and return aggregated value.

Examples of multiple row functions are the group functions that you have learnt in previous section *i.e.*, sum( ), count( ), max( ), min( ), Avg( ) etc.

The difference between these two types of functions is in the number of rows they act upon. A **single row function** works with the data of a single row at a time and returns a single result for each row queried upon ; a **multiple row function** works with a group of rows and returns a single result for that group.

Scanned with CamScanner