

Computer Science (083) _ 8th Week Assignment with Notes

Chapter Name: - More on MySQL Class: -12th

MySQL aggregate functions

There are various aggregate functions that can be used with **SELECT** statement. A list of commonly used aggregate functions are given below.

Aggregate Function	Description
count()	returns the count of expression.
sum()	returns the total summed value of expression.
avg()	returns the average value of expression.
min()	returns the minimum value of expression.
max()	returns the maximum value of expression.

MySQL Count () Function

MySQL **count()** function is used to returns the count of an expression. It allows us to count all rows or only some rows of the table that matches a specified condition. It is a type of aggregate function whose return type is **BIGINT**. This function returns 0 if it does not find any matching rows.

We can use the count function in three forms, which are explained below:

- **Count (*)**
- **Count (expression)**
- **Count (distinct)**

Let us discuss each in detail.

COUNT(*) Function: This function uses the **SELECT** statement to returns the count of rows in a result set. The result set contains all Non-Null, Null, and duplicates rows.

COUNT(expression) Function: This function returns the result set without containing Null rows as the result of an expression.

COUNT(distinct expression) Function: This function returns the count of distinct rows without containing NULL values as the result of the expression.

Syntax :

The following are the syntax of the **COUNT()** function:

1. **SELECT COUNT (aggregate_expression)**
2. **FROM table_name**
3. **[WHERE conditions];**

Parameter explanation

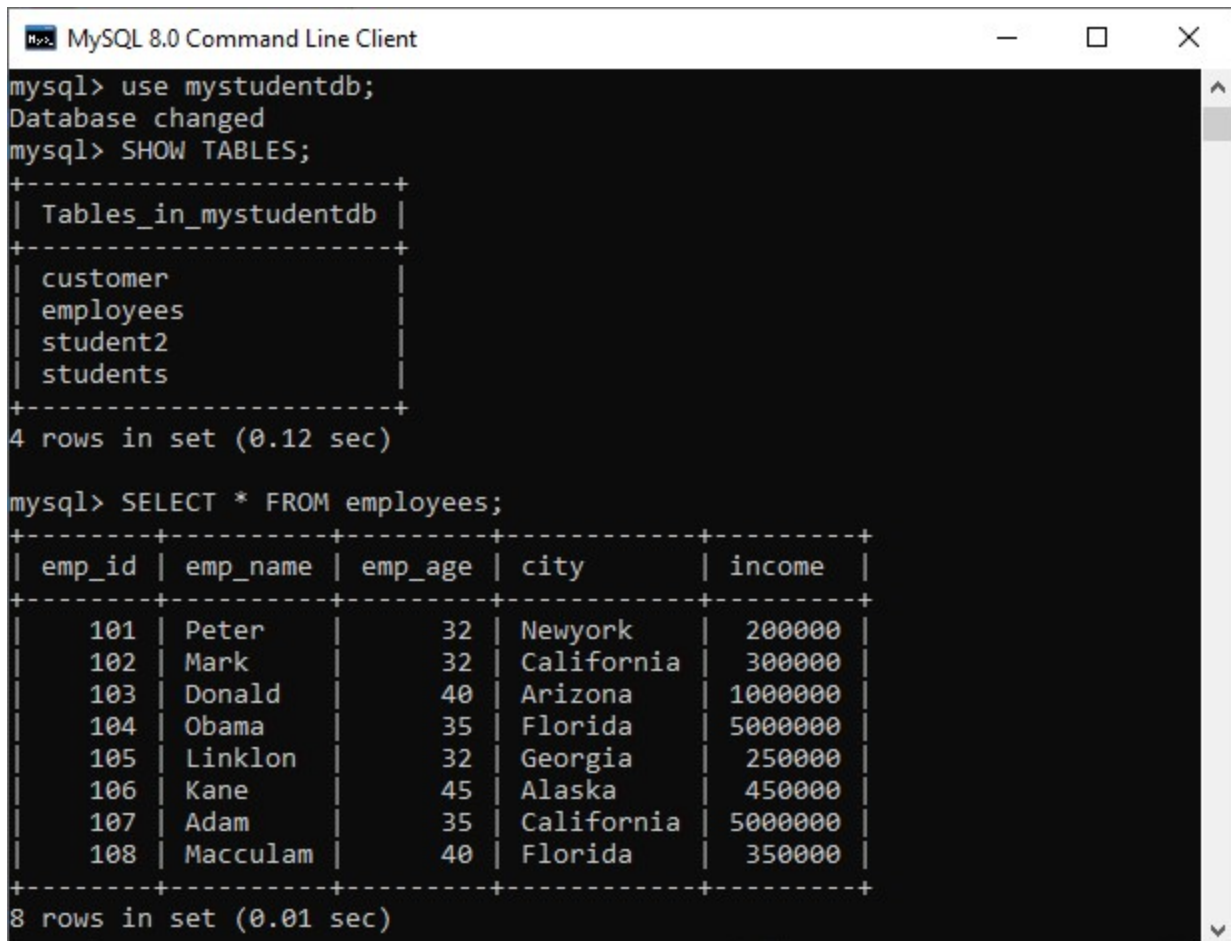
aggregate_expression: It specifies the column or expression whose **NON-NULL** values will be counted.

table_name: It specifies the tables from where you want to retrieve records. There must be at least one table listed in the **FROM** clause.

WHERE conditions: It is optional. It specifies the conditions that must be fulfilled for the records to be selected.

MySQL count() function example:

Consider a table named "employees" that contains the following data.



```
mysql> use mystudentdb;
Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_mystudentdb |
+-----+
| customer                |
| employees                |
| student2                |
| students                |
+-----+
4 rows in set (0.12 sec)

mysql> SELECT * FROM employees;
+-----+-----+-----+-----+-----+
| emp_id | emp_name | emp_age | city       | income |
+-----+-----+-----+-----+-----+
| 101    | Peter   | 32      | Newyork    | 200000 |
| 102    | Mark    | 32      | California | 300000 |
| 103    | Donald  | 40      | Arizona    | 1000000 |
| 104    | Obama   | 35      | Florida    | 5000000 |
| 105    | Linklon | 32      | Georgia    | 250000 |
| 106    | Kane    | 45      | Alaska     | 450000 |
| 107    | Adam    | 35      | California | 5000000 |
| 108    | Macculam | 40      | Florida    | 350000 |
+-----+-----+-----+-----+-----+
8 rows in set (0.01 sec)
```

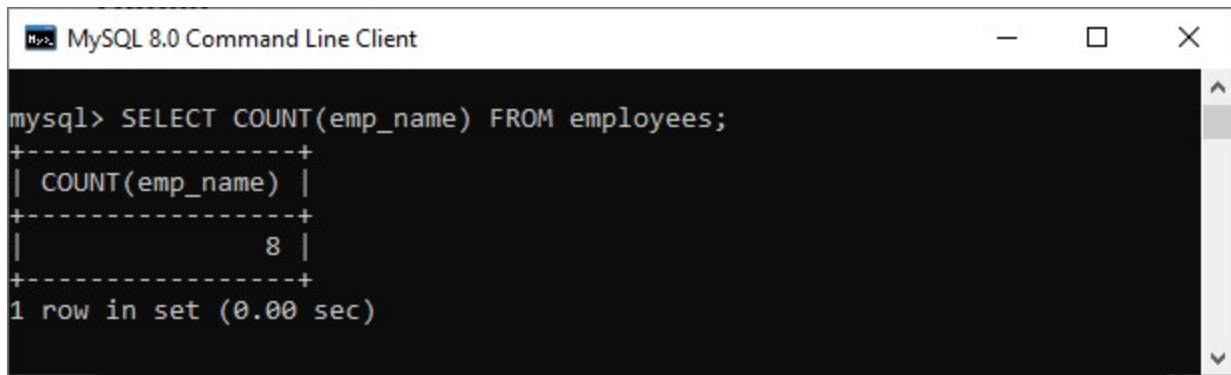
Let us understand how count() functions work in MySQL.

Example1

Execute the following query that uses the COUNT(expression) function to calculates the total number of employees name available in the table:

1. `mysql> SELECT COUNT(emp_name) FROM employees;`

Output:



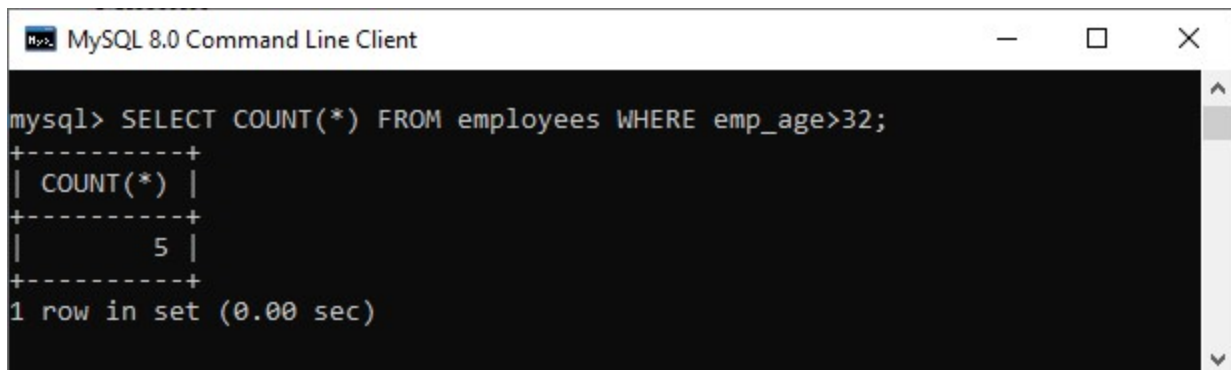
```
mysql> SELECT COUNT(emp_name) FROM employees;
+-----+
| COUNT(emp_name) |
+-----+
|          8      |
+-----+
1 row in set (0.00 sec)
```

Example2

Execute the following statement that returns all rows from the employee table and WHERE clause specifies the rows whose value in the column emp_age is greater than 32:

1. mysql> **SELECT** COUNT(*) **FROM** employees **WHERE** emp_age>32;

Output:



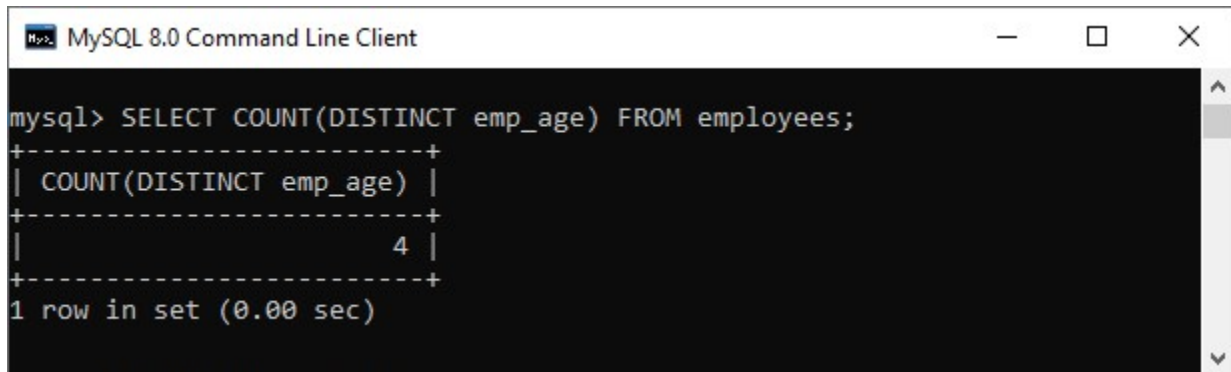
```
mysql> SELECT COUNT(*) FROM employees WHERE emp_age>32;
+-----+
| COUNT(*) |
+-----+
|          5 |
+-----+
1 row in set (0.00 sec)
```

Example3

This statement uses the COUNT(distinct expression) function that counts the Non-Null and distinct rows in the column emp_age:

1. mysql> **SELECT** COUNT(**DISTINCT** emp_age) **FROM** employees;

Output:



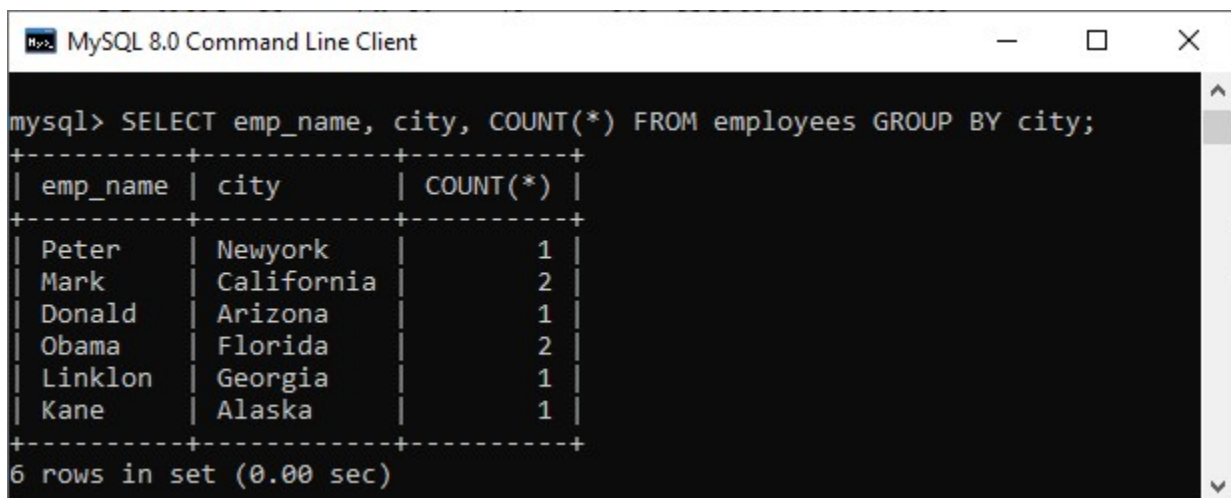
```
mysql> SELECT COUNT(DISTINCT emp_age) FROM employees;
+-----+
| COUNT(DISTINCT emp_age) |
+-----+
| 4 |
+-----+
1 row in set (0.00 sec)
```

MySQL Count() Function with GROUP BY Clause

We can also use the count() function with the GROUP BY clause that returns the count of the element in each group. For example, the following statement returns the number of employee in each city:

1. `mysql> SELECT emp_name, city, COUNT(*) FROM employees GROUP BY city;`

After the successful execution, we will get the result as below:



```
mysql> SELECT emp_name, city, COUNT(*) FROM employees GROUP BY city;
+-----+-----+-----+
| emp_name | city      | COUNT(*) |
+-----+-----+-----+
| Peter    | Newyork   | 1 |
| Mark     | California | 2 |
| Donald   | Arizona   | 1 |
| Obama    | Florida   | 2 |
| Linklon  | Georgia   | 1 |
| Kane     | Alaska    | 1 |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

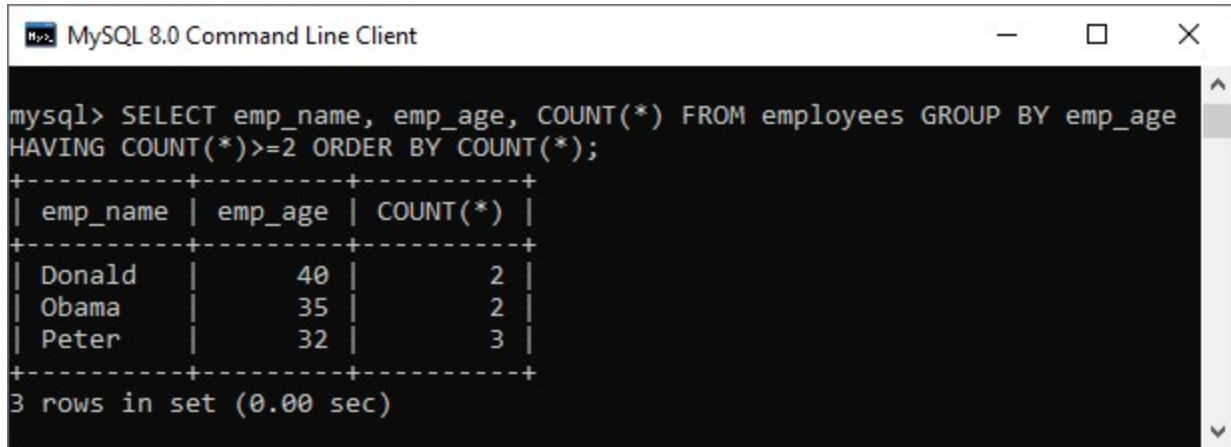
MySQL Count() Function with HAVING and ORDER BY Clause

Let us see another clause that uses ORDER BY and Having clause with the count() function. Execute the following statement that gives the employee name who has at least two age same and sorts them based on the count result:

1. `mysql> SELECT emp_name, emp_age, COUNT(*) FROM employees`
2. `GROUP BY emp_age`
3. `HAVING COUNT(*)>=2`

4. **ORDER BY** COUNT(*);

This statement will give the output as below:



```
mysql> SELECT emp_name, emp_age, COUNT(*) FROM employees GROUP BY emp_age
HAVING COUNT(*)>=2 ORDER BY COUNT(*);
+-----+-----+-----+
| emp_name | emp_age | COUNT(*) |
+-----+-----+-----+
| Donald  | 40      | 2        |
| Obama   | 35      | 2        |
| Peter   | 32      | 3        |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

MySQL sum() function

The MySQL sum() function is used to return the total summed value of an expression. It returns **NULL** if the result set does not have any rows. It is one of the kinds of aggregate functions in MySQL.

Syntax :

Following are the syntax of sum() function in MySQL:

1. **SELECT** SUM(aggregate_expression)
2. **FROM** tables
3. [**WHERE** conditions];

Parameter Explanation

aggregate_expression: It specifies the column or expression that we are going to calculate the sum.

table_name: It specifies the tables from where we want to retrieve records. There must be at least one table listed in the FROM clause.

WHERE conditions: It is optional. It specifies the conditions that must be fulfilled for the records to be selected.

MySQL sum() function example

Consider our database has a table named **employees**, having the following data. Now, we are going to understand this function with various examples:

```
MySQL 8.0 Command Line Client
mysql> SELECT * FROM employees;
+-----+-----+-----+-----+-----+
| emp_id | emp_name | occupation | working_date | working_hours |
+-----+-----+-----+-----+-----+
| 1      | Joseph   | Business   | 2020-04-10    | 10             |
| 2      | Stephen  | Doctor     | 2020-04-10    | 15             |
| 3      | Mark     | Engineer   | 2020-04-10    | 12             |
| 4      | Peter    | Teacher    | 2020-04-10    | 9              |
| 1      | Joseph   | Business   | 2020-04-12    | 10             |
| 2      | Stephen  | Doctor     | 2020-04-12    | 15             |
| 4      | Peter    | Teacher    | 2020-04-12    | 9              |
| 3      | Mark     | Engineer   | 2020-04-12    | 12             |
| 1      | Joseph   | Business   | 2020-04-14    | 10             |
| 4      | Peter    | Teacher    | 2020-04-14    | 9              |
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

1. Basic Example

Execute the following query that calculates the total number of working hours of all employees in the table:

1. `mysql> SELECT SUM(working_hours) AS "Total working hours" FROM employees;`

Output:

We will get the result as below:

```
MySQL 8.0 Command Line Client
mysql> SELECT SUM(working_hours) AS "Total working hours" FROM employees;
+-----+
| Total working hours |
+-----+
| 111                 |
+-----+
1 row in set (0.00 sec)
```

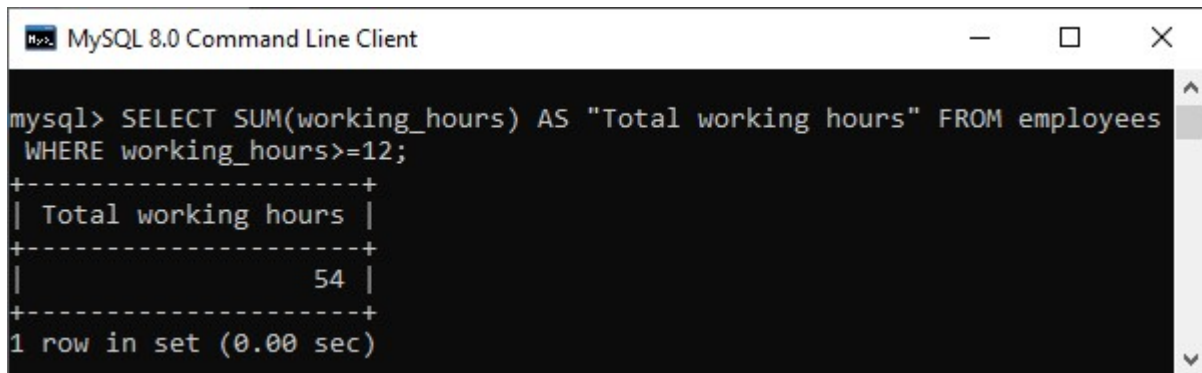
2. MySQL sum() function with WHERE clause

This example is used to return the result based on the condition specified in the WHERE clause. Execute the following query to calculate the total working hours of employees whose **working_hours** **>= 12**.

1. **mysql> SELECT SUM(working_hours) AS "Total working hours" FROM employees WHERE working_hours>=12;**

Output:

This statement will give the output as below:



```
mysql> SELECT SUM(working_hours) AS "Total working hours" FROM employees
WHERE working_hours>=12;
+-----+
| Total working hours |
+-----+
|                54 |
+-----+
1 row in set (0.00 sec)
```

3. MySQL sum() function with GROUP BY clause

We can also use the SUM() function with the GROUP BY clause to return the total summed value for each group. For example, this statement calculates the total working hours of each employee by using the SUM() function with the GROUP BY clause, as shown in the following query:

1. **mysql> SELECT emp_id, emp_name, occupation, SUM(working_hours) AS "Total working hours" FROM employees GROUP BY occupation;**

Output:

Here, we can see that the total working hours of each employee calculates by grouping them based on their occupation.


```
MySQL 8.0 Command Line Client

mysql> SELECT emp_id, emp_name, occupation, SUM(working_hours) AS "Total working hours"
FROM employees GROUP BY occupation;

+-----+-----+-----+-----+
| emp_id | emp_name | occupation | Total working hours |
+-----+-----+-----+-----+
| 1      | Joseph  | Business  | 30                  |
| 2      | Stephen | Doctor    | 30                  |
| 3      | Mark    | Engineer  | 24                  |
| 4      | Peter   | Teacher   | 27                  |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

4. MySQL sum() function with HAVING clause

The HAVING clause is used to **filter** the group with the sum() function in MySQL. Execute the following statement that calculates the working hours of all employees, grouping them based on their occupation and returns the result whose Total_working_hours>24.

1. **mysql> SELECT emp_id, emp_name, occupation,**
2. **SUM(working_hours) Total_working_hours**
3. **FROM employees**
4. **GROUP BY occupation**
5. **HAVING SUM(working_hours)>24;**

Output:

```
MySQL 8.0 Command Line Client

mysql> SELECT emp_id, emp_name, occupation, SUM(working_hours) Total_working_hours FROM
employees GROUP BY occupation HAVING SUM(working_hours) > 24;

+-----+-----+-----+-----+
| emp_id | emp_name | occupation | Total_working_hours |
+-----+-----+-----+-----+
| 1      | Joseph  | Business  | 30                  |
| 2      | Stephen | Doctor    | 30                  |
| 4      | Peter   | Teacher   | 27                  |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

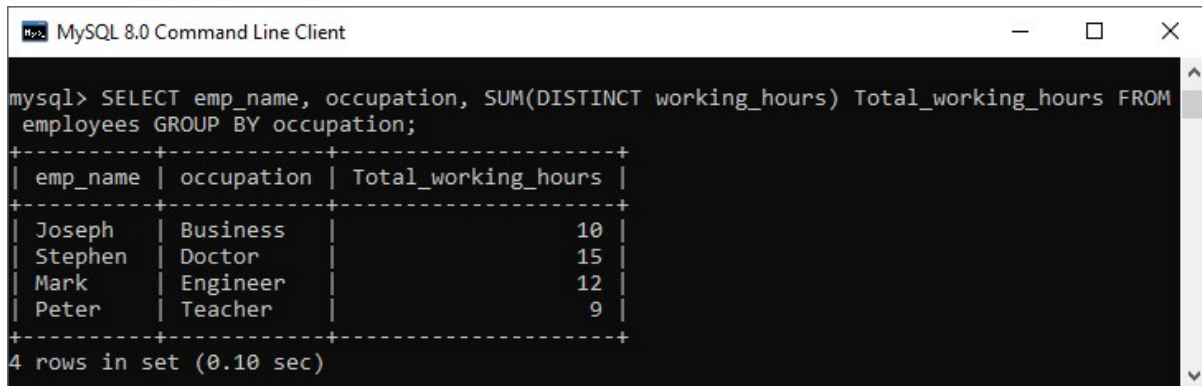
5. MySQL sum() function with DISTINCT clause

MySQL uses the DISTINCT keyword to remove the **duplicate** rows from the column name. This clause can also be used with sum() function to return the total summed value of a Unique number of records present in the table.

Execute the following query that removes the duplicate records in the working_hours column of the employee table and then calculates the sum:

1. **mysql> SELECT emp_name, occupation,**
2. **SUM(DISTINCT working_hours) Total_working_hours**
3. **FROM employees**
4. **GROUP BY occupation;**

Output:



```
mysql> SELECT emp_name, occupation, SUM(DISTINCT working_hours) Total_working_hours FROM
employees GROUP BY occupation;
+-----+-----+-----+
| emp_name | occupation | Total_working_hours |
+-----+-----+-----+
| Joseph   | Business   | 10                  |
| Stephen  | Doctor     | 15                  |
| Mark     | Engineer   | 12                  |
| Peter    | Teacher    | 9                   |
+-----+-----+-----+
4 rows in set (0.10 sec)
```

Answer the following questions:

1. What are aggregate functions in MySQL?
2. Which clause is used with an aggregate functions?
3. How do you write aggregate function in SQL?
4. Why are aggregate functions called so?
5. Is sum an aggregate function in SQL?
6. What is the syntax of Count() function and sum() function ?