MySQL - Select Query

Now that we have learned how to create tables in MySQL and insert values into it in the previous tutorials, the next step is to check whether the values are recorded in this table or not. To do this, one must use the SELECT statement to retrieve and view the records in that specific table."

MySQL Select Statement

The MySQL SELECT command is used to fetch data from the MySQL database in the form of a result table. These result tables are called result-sets.

Note – We can use this command at 'mysql>' prompt as well as in any script like PHP, Node.js, Java, python, etc.

Syntax

Here is generic SQL syntax of SELECT command to fetch data from the MySQL table –

```
SELECT field1, field2,...fieldN
FROM table_name1, table_name2...
[WHERE Clause]
[OFFSET M ][LIMIT N]
```

- You can use one or more tables separated by comma to include various conditions using a WHERE clause, but the WHERE clause is an optional part of the SELECT command.
- We can fetch one or more fields in a single SELECT command.
- We can specify star (*) in place of fields. In this case, SELECT will return all the fields.
- We can specify any condition using the WHERE clause.
- We can specify an offset using **OFFSET** from where SELECT will start returning records. By default, the offset starts at zero.
- We can limit the number of returns using the **LIMIT** attribute.

Fetching Data Using SELECT from Command Prompt

This will use SQL SELECT command to fetch data from an MySQL table.

Example

First of all, let us create a table named **CUSTOMERS** using the following query –

```
CREATE TABLE CUSTOMERS (
  ID INT NOT NULL,
  NAME VARCHAR(20) NOT NULL,
  AGE INT NOT NULL,
  ADDRESS CHAR (25),
  SALARY DECIMAL (18, 2),
  PRIMARY KEY (ID)
);
```

```
INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY) VALUES
(1, 'Ramesh', 32, 'Ahmedabad', 2000.00 ),
(2, 'Khilan', 25, 'Delhi', 1500.00 ),
(3, 'Kaushik', 23, 'Kota', 2000.00 ),
(4, 'Chaitali', 25, 'Mumbai', 6500.00 ),
(5, 'Hardik', 27, 'Bhopal', 8500.00 ),
(6, 'Komal', 22, 'Hyderabad', 4500.00 ),
(7, 'Muffy', 24, 'Indore', 10000.00 );
```

```
SELECT * from CUSTOMERS;
```

```
SELECT ID, NAME, ADDRESS FROM CUSTOMERS;
```

Computing using SELECT in Command Prompt

The SELECT statement is not only used to fetch data from tables but can also be used to get the results of mathematical computations in a tabular format. In these cases, you don't have to mention a specific database table in the SELECT statement.

Following is the syntax to do so –

```
SELECT [math_computation];
```

Example

In the following example, let us solve a mathematical computation using the SELECT statement –

```
SELECT 46475*453;
```

Aliasing a Column in SELECT Statement

MySQL database provides a method to alias column names into a more understandable and relative name when being displayed. This is done using the 'AS' keyword. This keyword is used in the SELECT statement as well.

Following is the syntax to do so –

```
SELECT column_name AS alias_name FROM table_name;
```

Example

In the example below, we are retrieving the **ID** column from the previously created CUSTOMERS table. We aliased the ID column as **"Identity_Document"** –

```
SELECT ID AS Identity_Document FROM CUSTOMERS;
```

Select Query into MySQL Database Using a Client Program

```
select_query = "SELECT COLUMN1, COLUMN2,.. FROM TABLE_NAME";
cursorObj.execute(select_query);
```

```
import mysql.connector
import datetime
#establishing the connection
connection = mysql.connector.connect(
    host='localhost',
    user='root',
    password='password',
    database='tut'
)
table_name = 'tutorials_tbl'
#Creating a cursor object
cursorObj = connection.cursor()
select_query = f"SELECT tutorial_id, tutorial_title, tutorial_author, submission_date FROM {table_name}"
cursorObj.execute(select_query)
result = cursorObj.fetchall()
print("Tutorial Table Data:")
for row in result:
    print(row)
cursorObj.close()
connection.close()
```