MySQL Select

Summary: in this tutorial, you'll learn how to use the MySQL `SELECT` statement without referencing any table.

Typically, you use a `SELECT (https://www.mysqltutorial.org/basic-mysql-tutorial-aspx/mysql-select/)` statement to select data from a table in the database:

```
SELECT select_list
FROM table_name;
```

MySQL doesn't require the `FROM` clause. It means that you can have a `SELECT` statement without the `FROM` clause like this:

```
SELECT select_list;
```

Here's a trivial example:

```
SELECT 1 + 1;
```

Output:

```
+-------+
| 1 + 1 |
+-------+
|     2 |
+-------+
1 row in set (0.00 sec)
```

MySQL has many built-in functions like string, date, and Math functions. And you can use the `SELECT` statement to execute these functions.

The following example returns the current date and time of the MySQL server:

```
SELECT NOW();
```

Output:

```
+---------------------+
| NOW()               |
+---------------------+
| 2021-07-26 08:08:02 |
+---------------------+
1 row in set (0.00 sec)
```

The `NOW()` function returns the current date & time of the server on which MySQL runs. The `NOW()` function doesn't have any parameters. To call it, you place the parentheses () after the function name.

If a function has parameters, you need to pass arguments into it. For example, concatenate strings into one string, you can use the `CONCAT()` function:

```
SELECT CONCAT('John',' ','Doe');
```

Output:

```
+--------------------------+
| CONCAT('John',' ','Doe') |
+--------------------------+
| John Doe                 |
+--------------------------+
1 row in set (0.00 sec)
```

The CONCAT() function accepts one or more strings and concatenates them into a single string.

The dual table

Sometimes, you still need to use the `FROM` clause but you don't want to reference any actual table. In this case, you can use the dual table in the `FROM` clause:

```
SELECT select_list
FROM dual;
```

The dual table is a dummy table, not an actual table.

The dual is also necessary in case you need other clauses of the `SELECT` statement. Without the `FROM` clause, the `SELECT` statement would not be valid.

A quick introduction to the column alias

By default, MySQL uses the expression specified in the `SELECT` clause as the column name of the result set. To change a column name of the result set, you can use a column alias:

```
SELECT expression AS column_alias;
```

To assign an alias to a column, you place the AS keyword after the expression followed by a column alias. The `AS` keyword is optional, so you can skip it like this:

```
SELECT expression column_alias;
```

For example:

```
SELECT CONCAT('John',' ','Doe') AS name;
```

Output:

```
+----------+
| name     |
+----------+
| John Doe |
+----------+
1 row in set (0.00 sec)
```

If the column alias contains spaces, you need to place it inside quotes like this:

```
SELECT CONCAT('Jane',' ','Doe') AS 'Full name';
```

Output:

```
+-----------+
| Full name |
+-----------+
| John Doe  |
+-----------+
1 row in set (0.00 sec)
```

Summary

- MySQL `SELECT` statement doesn’t require the `FROM` clause
- Use the `dual` table if you want to use the `FROM` clause but don’t want to reference a table.
- Assign an alias to a column to make it more readable.