MySQLTUTORIAL

MySQL DEFAULT

Summary: in this tutorial, you'll learn about MySQL DEFAULT constraint and how to use it effectively.

Introduction to the MySQL DEFAULT constraint

MySQL `DEFAULT` constraint allows you to specify a default value for a column. Here's the syntax of the `DEFAULT` constraint:

```
column_name data_type DEFAULT default_value;
```

In this syntax, you specify the `DEFAULT` keyword followed by the default value for the column. The type of the default value matches the data type of the column.

The `default_value` must be a literal constant, e.g., a number or a string. It cannot be a function or an expression. However, MySQL allows you to set the current date and time (`CURRENT_TIMESTAMP`) to the `TIMESTAMP` (https://www.mysqltutorial.org/mysql-timestamp.aspx) and `DATETIME` (https://www.mysqltutorial.org/mysql-datetime/) columns.

When you define a column without the `NOT NULL` (https://www.mysqltutorial.org/mysql-not-null-constraint/) constraint, the column will implicitly take `NULL` as the default value.

If a column has a `DEFAULT` constraint and the `INSERT` (https://www.mysqltutorial.org/mysql-insert-statement.aspx) or `UPDATE` (https://www.mysqltutorial.org/mysql-update-data.aspx) statement doesn't provide the value for that column, MySQL will use the default value specified in the `DEFAULT` constraint.

Typically, you set the `DEFAULT` constraints for columns when you create the table (https://www.mysqltutorial.org/mysql-create-table/) . MySQL also allows you to add default constraints to the columns of existing tables. If you don't want to use default values for columns, you can remove the default constraints.

MySQL DEFAULT constraint example

The following example creates a new table named `cart_items` with four columns `item_id`, `name`, `quantity`, and `sales_tax`:

```
CREATE TABLE cart_items
(
    item_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    quantity INT NOT NULL,
    price DEC(5,2) NOT NULL,
    sales_tax DEC(5,2) NOT NULL DEFAULT 0.1,
    CHECK(quantity > 0),
    CHECK(sales_tax >= 0)
);
```

The `sales_tax` column has a default value 0.1 (10%). The following statement shows the `cart_items` table:

```
DESC cart_items;
```

Output:

```
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| item_id   | int          | NO   | PRI | NULL    | auto_increment |
| name      | varchar(255) | NO   |     | NULL    |                |
| quantity  | int          | NO   |     | NULL    |                |
| price     | decimal(5,2) | NO   |     | NULL    |                |
| sales_tax | decimal(5,2) | NO   |     | 0.10    |                |
+-----------+--------------+------+-----+---------+----------------+
5 rows in set (0.01 sec)
```

The following `INSERT` statement adds a new item to the `cart_items` table:

```
INSERT INTO cart_items(name, quantity, price)
VALUES('Keyboard', 1, 50);
```

In this example, the `INSERT` statement doesn't provide a value for the `sales_tax` column. The `sales_tax` column useS the default value specified in the `DEFAULT` constraint:

```
SELECT * FROM cart_items;
```

Output:

```
+---------+----------+----------+-------+-----------+
| item_id | name     | quantity | price | sales_tax |
+---------+----------+----------+-------+-----------+
|       1 | Keyboard |        1 | 50.00 |      0.10 |
+---------+----------+----------+-------+-----------+
1 row in set (0.00 sec)
```

Also, you can explicitly use the `DEFAULT` keyword when you insert a new row into the `cart_items` table:

```
INSERT INTO cart_items(name, quantity, price, sales_tax)
VALUES('Battery',4, 0.25 , DEFAULT);
```

In this case, the `sales_tax` column takes the default value:

```
SELECT * FROM cart_items;
```

Output:

```
+---------+----------+----------+-------+-----------+
| item_id | name     | quantity | price | sales_tax |
+---------+----------+----------+-------+-----------+
|       1 | Keyboard |        1 | 50.00 |      0.10 |
|       2 | Battery  |        4 |  0.25 |      0.10 |
+---------+----------+----------+-------+-----------+
2 rows in set (0.01 sec)
```

Adding a DEFAULT constraint to a column

To add a default constraint to a column of an existing table, you use the `ALTER TABLE` (https://www.mysqltutorial.org/mysql-alter-table.aspx) statement:

```
ALTER TABLE table_name
ALTER COLUMN column_name SET DEFAULT default_value;
```

The following example adds a `DEFAULT` constraint to the `quantity` column of the `cart_itesm` table:

```
ALTER TABLE cart_items
ALTER COLUMN quantity SET DEFAULT 1;
```

If you describe the `cart_items` table, you'll see the changes:

```
DESC cart_items;
```

Output:

```
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| item_id   | int          | NO   | PRI | NULL    | auto_increment |
| name      | varchar(255) | NO   |     | NULL    |                |
| quantity  | int          | NO   |     | 1       |                |
| price     | decimal(5,2) | NO   |     | NULL    |                |
| sales_tax | decimal(5,2) | NO   |     | 0.10    |                |
+-----------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

The following statement inserts a new row into the `cart_items` table without specifying a value for the `quantity` column:

```
INSERT INTO cart_items(name, price, sales_tax)
VALUES('Maintenance services',25.99, 0)
```

The value of the `quantity` column will default to 1:

```
SELECT * FROM cart_items;
```

Output:

```
+---------+----------------------+----------+-------+-----------+
| item_id | name                 | quantity | price | sales_tax |
+---------+----------------------+----------+-------+-----------+
|       1 | Keyboard             |        1 | 50.00 |      0.10 |
|       2 | Battery              |        4 |  0.25 |      0.10 |
|       3 | Maintenance services |        1 | 25.99 |      0.00 |
```

```
+---------+----------------------+----------+-------+-----------+
3 rows in set (0.00 sec)
```

Removing a DEFAULT constraint from a column

To remove a `DEFAULT` constraint from a column, you use the `ALTER TABLE` statement:

```
ALTER TABLE table_name
ALTER column_name DROP DEFAULT;
```

The following example removes the `DEFAULT` constraint from the `quantity` column of the `cart_items` table:

```
ALTER TABLE cart_items
ALTER COLUMN quantity DROP DEFAULT;
```

And here's the new `cart_items` structure:

```
DESC cart_items;
```

Output:

```
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| item_id   | int          | NO   | PRI | NULL    | auto_increment |
| name      | varchar(255) | NO   |     | NULL    |                |
| quantity  | int          | NO   |     | NULL    |                |
| price     | decimal(5,2) | NO   |     | NULL    |                |
| sales_tax | decimal(5,2) | NO   |     | 0.10    |                |
+-----------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Summary

- MySQL `DEFAULT` constraints set default values for columns.
- Use `DEFAULT default_value` to set a default constraint to a column.

- Use `ALTER TABLE ... ALTER COLUMN ... SET DEFAULT` to add a `DEFAULT` constraint to a column of an existing table.
- Use `ALTER TABLE ... ALTER COLUMN ... DROP DEFAULT` to drop a `DEFAULT` constraint from a column of an existing table.