

The ENUM and SET types

Overview

ENUM (enumeration) and SET can be considered character data types. Both data types can be used if the number of values that a column may contain is restricted.

We could let MySQL check whether the value entered is correct with the help of the data types ENUM and SET. For columns with these two data types, the set of permitted values has been defined.

The main difference between the ENUM and SET types is that an ENUM column can store exactly one value, but SET column can store zero, one, or up to 64 different values. An ENUM column can consist of only one member in a set of values, while the SET column may consist of any, or all, members in a set.

Objectives

After completing this lesson, you should be able to do the following:

- Create tables using the ENUM data type.
- Create tables using the SET data type.

The ENUM Data Type

When defining a column with an ENUM data type, the list of permitted values is given. You can define the set of acceptable value for an ENUM column when you create a table.

An ENUM column can have up to **65535** acceptable values. To specify a value for a ENUM column, you code a single text string. If the string contains an acceptable value, that value is stored in the column. Otherwise, the column is assigned an empty string.

If you do not specify a value for an ENUM column when you insert a row, MySQL assigns a default value that depends on whether the column allows null values. If column allows null values, MySQL assigns a null value to the column. If it does not allow null values, MySQL assigns the first value in the set of acceptable values to the column.

Example 1: Create a table color sample and insert rows.

```
mysql> DROP DATABASE IF EXISTS examples;
mysql> CREATE DATABASE examples;
mysql> USE examples;
mysql> CREATE TABLE color_sample
    (color_id INT NOT NULL AUTO_INCREMENT,
    color_number INT NOT NULL DEFAULT 0,
    color_name enum('red', 'green', 'blue', 'orange'),
    CONSTRAINT color_sample_pk PRIMARY KEY (color_id) )
    ENGINE = INNODB;
```

Behind the term ENUM, all legal values are specified between brackets.

Insert three rows.

```
INSERT INTO color_sample (color_number, color_name)
VALUES (999, 'red');
INSERT INTO color_sample (color_number, color_name)
VALUES (998, 'green');
INSERT INTO color_sample (color_number, color_name)
VALUES (808, 'yellow');
2 rows affected
ERROR 1265 (01000): Data truncated for column 'color_name'
```

2.1 Update with an ENUM

The ENUM definition controls the ability to update.

Example 2: Change the value of color_name for the row with color_id of 1.

```
mysql> UPDATE color_sample
      SET color_name = 'orange'
      WHERE color_id = 1;
```

This one fails:

```
mysql> UPDATE color_sample
      SET color_name = 'brown'
      WHERE color_id = 2;
ERROR 1265 (01000):Data truncated for column 'color_name' at row 1
```

Example 3: We can enter nulls into that attribute.

```
mysql> INSERT into color_sample (color_number, color_name)
      VALUES (5, null);
```

2.2 Enum and Storage

Example 4: Display the data in the table sorted by the color_name.

```
mysql> SELECT *
      FROM color_sample
      ORDER BY color_name;
```

color_id	color_number	color_name
3	5	NULL
2	998	green
1	999	orange

3 rows in set (0.00 sec)

This did a sort on the color_name column, but the rows are not in alphabetic order. This is due to the actual storage used for these values. The row order corresponds to the order of the literals in the enumeration.

Example 5: Showing numeric values associated with the enum.

```
mysql> SELECT color_id, color_number * 2, color_name
      FROM color_sample;
```

color_id	color_number * 2	color_name
1	1998	orange
2	1996	green
3	10	NULL

3 rows in set (0.00 sec)

2.3 Sorting

Example 6: To get the rows sorted in alphabetic order we can use a cast.

```
mysql> SELECT *
      FROM color_sample
      ORDER BY cast(color_name as char);
```

color_id	color_number	color_name
1	999	orange
2	998	green
3	5	NULL

3 rows in set (0.00 sec)

Example 7: We can filter on the numeric value and on the name value.

```
mysql> SELECT color_id, color_name
      FROM color_sample
      WHERE color_name = 'green';
```

color_id	color_name
2	green

1 row in set (0.00 sec)

1. The SET data type

The SET data type is similar to the ENUM type in that it allows for defining a type with multiple values. However, the difference is that a column with the SET data type can contain more than one value from the list. Normally, we store in a specific row only one value per column.

To specify values for a SET column, you code a single string with the values separated by commas. Each acceptable value is stored in the column, and any other values are ignored. When you store values in a SET column, MySQL stores the values using the order specified in the column definition, and it does not store duplicate values. You can use a SET column when you want to more than one value. When storing multiple values in a SET column, the order of the values does not matter.

Example 8: Create a table with a SET type.

```
mysql> CREATE TABLE classes( class_id int
      , class_name SET ('CS10A', 'CS11A', 'CS15A', 'CS150A', 'CS11B'))
      Engine = INNODB;
```

Example 9: Insert rows.

```
mysql> INSERT into classes
      VALUES (1, null);
```

Insert with a single element for class_name.

```
mysql> INSERT into classes
      VALUES (2, 'CS10A');
```

Use a comma between the individual elements.

```
mysql> INSERT into classes
      VALUES (3, 'CS11B,CS15A');
```

CS 31A | The ENUM and SET types

Insert with two elements for class_name;

```
mysql> INSERT into classes  
VALUES (4, 'CS11A,CS150A');
```

Invalid entry—since that item is not on the allowed list.

```
mysql> INSERT into classes  
VALUES (5, 'CS160A');  
ERROR 1265 (01000): Data truncated for column ' class_name ' at row 1
```

Example 10: What do we have in the table?

```
mysql> SELECT *  
FROM classes;  
+-----+-----+  
| class_id | class_name |  
+-----+-----+  
| 1 | NULL |  
| 2 | CS10A |  
| 3 | CS15A,CS11B |  
| 4 | CS11A,CS150A |  
+-----+-----+  
4 rows in set (0.01 sec)
```