MySQL DATE Data Type

Summary: in this tutorial, we will introduce you to the MySQL `DATE` data type and show you some useful date functions to handle the date data effectively.

Introduction to MySQL DATE data type

MySQL `DATE` is one of the five temporal data types used for managing date values. MySQL uses `yyyy-mm-dd` format for storing a date value. This format is fixed and it is not possible to change it.

For example, you may prefer to use `mm-dd-yyyy` format but you can't. Instead, you follow the standard date format and use the `DATE_FORMAT` function to format the date the way you want.

MySQL uses three bytes to store a `DATE` value. The `DATE` values range from `1000-01-01` to `9999-12-31`.

If you want to store a date value that is out of this range, you need to use a non-temporal data type like an integer e.g., three columns, and each column for the year, month, and day.

Also, you need to create stored functions to simulate the built-in date functions provided by MySQL, which is not recommended.

MySQL Date values with two-digit years

MySQL stores the year of the date value using four digits. In case you use two-digit year values, MySQL still accepts them with the following rules:

- Year values in the range 00-69 are converted to 2000-2069.
- Year values in the range 70-99 are converted to 1970 – 1999.

However, a date value with two digits is ambiguous therefore you should avoid using it.

Let's take a look at the following example.

First, create a table named people with birth date column with `DATE` data type.

```
CREATE TABLE people (
    id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    birth_date DATE NOT NULL
);
```

Next, insert a row into the `people` table.

```
INSERT INTO people(first_name,last_name,birth_date)
VALUES('John','Doe','1990-09-01');
```

Then, query the data from the `people` table.

```
SELECT
    first_name,
    last_name,
    birth_date
FROM
    people;
```

After that, use the two-digit year format to insert data into the `people` table.

```
INSERT INTO people(first_name,last_name,birth_date)
VALUES('Jack','Daniel','01-09-01'),
      ('Lily','Bush','80-09-01');
```

In the first row, we used 01 (range 00-69) as the year, so MySQL converted it to 2001. In the second row, we used 80 (range 70-99) as the year, MySQL converted it to 1980.

Finally, retrieve data from the `people` table to check whether data was converted based on the conversion rules.

```
SELECT
    first_name,
    last_name,
    birth_date
FROM
    people;
```

MySQL Date Functions

MySQL provides many useful date functions that allow you to manipulate dates effectively.

To get the current date and time, you use `NOW()` function.

```
SELECT NOW();
```

```
+---------------------+
| NOW()               |
+---------------------+
| 2017-05-13 07:59:38 |
+---------------------+
1 row in set (0.02 sec)
```

To get only the date part of a `DATETIME` value, you use the `DATE()` function.

```
SELECT DATE(NOW());
```

```
+-------------+
| DATE(NOW()) |
+-------------+
| 2015-07-13  |
+-------------+
1 row in set (0.01 sec)
```

To get the current system date, you use `CURDATE()` function as follows:

```
SELECT CURDATE();
```

```
+------------+
| CURDATE()  |
+------------+
| 2015-07-13 |
+------------+
1 row in set (0.02 sec)
```

To format a date value, you use `DATE_FORMAT` function. The following statement formats the date as `mm/dd/yyyy` using the date format pattern `%m/%d/%Y` :

```
SELECT DATE_FORMAT(CURDATE(), '%m/%d/%Y') today;
```

```
+------------+
| today      |
+------------+
| 07/13/2015 |
+------------+
1 row in set (0.02 sec)
```

To calculate the number of days between two date values, you use the `DATEDIFF` function as follows:

```
SELECT DATEDIFF('2015-11-04','2014-11-04') days;
```

```
+------+
| days |
+------+
|  365 |
+------+
1 row in set (0.02 sec)
```

To add a number of days, weeks, months, years, etc., to a date value, you use the `DATE_ADD` function:

```
SELECT
    '2015-01-01' start,
    DATE_ADD('2015-01-01', INTERVAL 1 DAY) 'one day later',
    DATE_ADD('2015-01-01', INTERVAL 1 WEEK) 'one week later',
    DATE_ADD('2015-01-01', INTERVAL 1 MONTH) 'one month later',
    DATE_ADD('2015-01-01', INTERVAL 1 YEAR) 'one year later';
```

Similarly, you can subtract an interval from a date using the `DATE_SUB` function:

```
SELECT
    '2015-01-01' start,
    DATE_SUB('2015-01-01', INTERVAL 1 DAY) 'one day before',
    DATE_SUB('2015-01-01', INTERVAL 1 WEEK) 'one week before',
    DATE_SUB('2015-01-01', INTERVAL 1 MONTH) 'one month before',
    DATE_SUB('2015-01-01', INTERVAL 1 YEAR) 'one year before';
```

If you want to get the day, month, quarter, and year of a date value, you can use the corresponding function `DAY` , `MONTH` , `QUARTER` , and `YEAR` as follows:

```
SELECT DAY('2000-12-31') day,
       MONTH('2000-12-31') month,
       QUARTER('2000-12-31') quarter,
       YEAR('2000-12-31') year;
```

```
+------+-------+---------+------+
| day  | month | quarter | year |
+------+-------+---------+------+
|   31 |    12 |       4 | 2000 |
+------+-------+---------+------+
1 row in set (0.00 sec)
```

To get the week's information, you use the week-related functions. For example, `WEEK` function returns the week number, `WEEKDAY` function returns the weekday index, and `WEEKOFYEAR` function returns the calendar week.

```
SELECT
    WEEKDAY('2000-12-31') weekday,
    WEEK('2000-12-31') week,
    WEEKOFYEAR('2000-12-31') weekofyear;
```

```
+---------+------+------------+
| weekday | week | weekofyear |
+---------+------+------------+
|       6 |   53 |         52 |
+---------+------+------------+
1 row in set (0.04 sec)
```

The week function returns the week number with the zero-based index if you don't pass the second argument or if you pass 0. If you pass 1, it will return the week number with 1-indexed.

```
SELECT
    WEEKDAY('2000-12-31') weekday,
    WEEK('2000-12-31',1) week,
    WEEKOFYEAR('2000-12-31') weekofyear;
```

```
+---------+------+------------+
| weekday | week | weekofyear |
+---------+------+------------+
|       6 |   52 |         52 |
+---------+------+------------+
1 row in set (0.00 sec)
```

In this tutorial, you have learned about the MySQL `DATE` data type and how to use some useful date functions to manipulate date values.

Was this tutorial helpful?

PREVIOUSLY
MySQL TIMESTAMP Data Type

UP NEXT
MySQL TIME Data Type

Search ...

GETTING STARTED

What Is MySQL?

Install MySQL Database Server

Connect to MySQL Server

Download MySQL Sample Database

Load Sample Database

QUERYING DATA

SELECT FROM

SELECT

ORDER BY

WHERE

SELECT DISTINCT

AND

OR

IN

NOT IN

BETWEEN

LIKE

LIMIT

IS NULL

Table & Column Aliases

Joins

INNER JOIN

LEFT JOIN

RIGHT JOIN

Self Join

CROSS JOIN

GROUP BY

HAVING

HAVING COUNT

ROLLUP

Subquery

Derived Tables

EXISTS

EXCEPT

INTERSECT

MANAGING DATABASES

Select a Database

Create Databases

Drop Databases

MANAGING TABLES

Create Tables

AUTO_INCREMENT

Rename Tables

Add Columns

Drop Columns

Drop Tables

Temporary Tables

Generated Columns

MYSQL CONSTRAINTS

Primary Key

Foreign Key

Disable Foreign Key Checks

UNIQUE Constraint

NOT NULL Constraint

DEFAULT Constraint

CHECK Constraint

INSERT DATA

Insert Into

Insert Multiple Rows

INSERT INTO SELECT

Insert On Duplicate Key Update

INSERT IGNORE

Insert DateTimes

Insert Dates

UPDATE DATA

UPDATE

UPDATE JOIN

DELETE DATA

DELETE JOIN

ON DELETE CASCADE

TRUNCATE TABLE

MYSQL TRANSACTIONS

Table Locking

MYSQL DATA TYPES

BIT

INT

BOOLEAN

DECIMAL

DATETIME

TIMESTAMP

DATE

TIME

CHAR

VARCHAR

TEXT

BINARY

VARBINARY

ENUM

BLOB

MYSQL GLOBALIZATION

MySQL Character Sets

MySQL Collation

MYSQL IMPORT & EXPORT

Import a CSV File Into a Table

Export a Table to a CSV File

ABOUT MYSQL TUTORIAL WEBSITE

MySQLTutorial.org helps you master MySQL quickly, easily, and with enjoyment. Our tutorials make learning MySQL a breeze.

All MySQL tutorials are clear, practical and easy-to-follow. More About Us

LATEST TUTORIALS

MySQL Port

MySQL Commands

innodb_dedicated_server: Configure InnoDB Dedicated Server

innodb_flush_method: Configure InnoDB Flush Method

innodb_log_buffer_size: Configure InnoDB Log Buffer Size

innodb_buffer_pool_chunk_size: Configure Buffer Pool Chunk Size

innodb_buffer_pool_instances: Configuring Multiple Buffer Pool Instances for Improved Concurrency in MySQL

innodb_buffer_pool_size: Configure InnoDB Buffer Pool Size

MySQL InnoDB Architecture

SITE LINKS

Donation ❤️

Contact Us

About

Privacy Policy

OTHERS

MySQL Cheat Sheet

MySQL Resources

MySQL Books

How to Kill a Process in MySQL