MySQL DECIMAL Data Type

Summary: in this tutorial, you will learn about MySQL `DECIMAL` data type and how to use it to store exact numeric values in the databases.

Introduction to MySQL DECIMAL data type

The MySQL `DECIMAL` data type allows you to store exact numeric values in the database. In practice, you often use the `DECIMAL` data type for columns that preserve exact precision e.g., monetary data in financial systems.

To define a column whose data type is `DECIMA` L, you use the following syntax:

```
column_name DECIMAL(P,D);
```

In this syntax:

- `P` is the precision that represents the number of significant digits. The range of `P` is 1 to 65.
- `D` is the scale that represents the number of digits after the decimal point. The range of `D` is 0 and 30. MySQL requires that `D` is less than or equal to (<=) `P`.

The `DECIMAL(P,D)` means that the column can store up to `P` digits with `D` decimals. The

actual range of the decimal column depends on the precision and scale.

Besides the `DECIMAL` keyword, you can also use `DEC`, `FIXED`, or `NUMERIC` because they are synonyms for `DECIMAL`.

The following example defines the `amount` column with `DECIMAL` data type.

```
amount DECIMAL(6,2);
```

In this example, the amount column can store 6 digits with 2 decimal places; therefore, the range of the amount column is from 9999.99 to -9999.99.

MySQL allows you to use the following syntax:

```
column_name DECIMAL(P);
```

This is equivalent to:

```
column_name DECIMAL(P,0);
```

In this case, the column contains no fractional part or decimal point.

In addition, you can even use the following syntax:

```
column_name DECIMAL;
```

The default value of P is 10 and D is 0, which is equivalent to the following:

```
column_name DECIMAL(10,0);
```

MySQL DECIMAL storage

MySQL assigns the storage for integer and fractional parts separately. MySQL uses a binary format to store the `DECIMAL` values. It packs 9 digits into 4 bytes.

For each part, it takes 4 bytes to store each multiple of 9 digits. The storage required for leftover digits is illustrated in the following table:

Leftover Digits	Bytes
0	0
1–2	1
3–4	2
5–6	3
7–9	4

For example, `DECIMAL(19,9)` has 9 digits for the fractional part and 19-9 = 10 digits for the integer part. The fractional part requires 4 bytes. The integer part requires 4 bytes for the first 9 digits, for 1 leftover digit, it requires 1 more byte. In total, the `DECIMAL(19,9)` column requires 9 bytes.

MySQL DECIMAL data type and monetary data

We often use the `DECIMAL` data type for monetary data such as prices, salary, account balances, and so on. If you design a database that handles the monetary data, the following syntax should be fine.

```
amount DECIMAL(19,2);
```

However, if you want to comply with Generally Accepted Accounting Principles (GAAP) rules, the monetary column must have at least 4 decimal places to make sure that the rounding value does not exceed $0.01. In this case, you should define the column with 4 decimal places as follows:

```
amount DECIMAL(19,4);
```

MySQL DECIMAL data type example

First, create a new table named `materials` :

```
CREATE TABLE materials (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
    description VARCHAR(255) NOT NULL,
    cost DECIMAL(19,4) NOT NULL
);
```

The `materials` table has three columns:

- `id` is the auto-increment primary key column with the `INT` data type.
- `description` represents the material's description with the `VARCHAR` data type.
- `cost` represents the cost of the material, which has the `DECIMAL(19,4)`.

The `cost` column can store up to 19 digits with 4 decimal places.

Second, insert data into the `materials` table.

```
INSERT INTO materials(description, cost)
VALUES
  ('Bicycle', 500.34),
  ('Seat', 10.23),
  ('Break', 5.21);
```

Third, query data from the `materials` table.

```
SELECT
    *
FROM
    materials;
```

Output:

```
+----+-------------+----------+
| id | description | cost     |
+----+-------------+----------+
|  1 | Bicycle     | 500.3400 |
|  2 | Seat        |  10.2300 |
|  3 | Break       |   5.2100 |
+----+-------------+----------+
3 rows in set (0.00 sec)
```

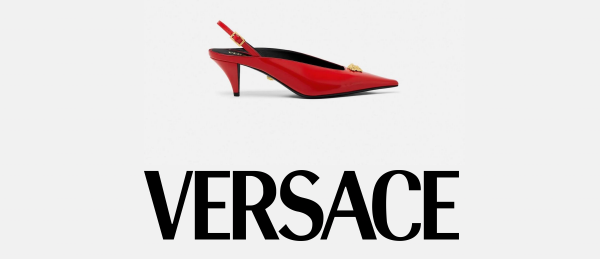
Summary

- Use MySQL `DECIMAL` data type to store exact numeric values such as financial data in the database.
- Use `column_name DECIMAL (P, D)` to define a column with the `DECIMAL` data type that has up to `P` digits and `D` decimal places.
- The `DECIMAL(P)` is equivalent to `DECIMAL(P,0)` and `DECIMAL` is equivalent to `DECIMAL(P, 0)` .

Was this tutorial helpful?

PREVIOUSLY
MySQL BOOLEAN Data Type

UP NEXT
MySQL DATETIME Data Type



Search ...

MANAGING DATABASES

MANAGING 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

SITE LINKS

Donation ❤️

Contact Us

About

Privacy Policy

OTHERS

MySQL Cheat Sheet

MySQL Resources

MySQL Books

Pool Instances for Improved Concurrency in MySQL

innodb_buffer_pool_size: Configure InnoDB Buffer Pool Size

MySQL InnoDB Architecture

How to Kill a Process in MySQL