MySQL INT Data Type

Summary: in this tutorial, you will learn about MySQL `INT` data type, and how to use it to store whole numbers in the databases.

Introduction to MySQL INT data type

In MySQL, `INT` stands for the integer that represents the whole numbers. An integer can be written without a fractional component such as 1, 100, 4, -10, and it cannot be 1.2, 5/3, etc. An integer can be zero, positive, and negative.

MySQL supports all standard SQL integer types `INTEGER` or `INT` and `SMALLINT`. Additionally, MySQL provides `TINYINT` `MEDIUMINT`, and `BIGINT` as extensions to the SQL standard.

MySQL `INT` data type can be signed and unsigned. The following table illustrates the characteristics of each integer type including storage in bytes, minimum value, and maximum value.

Type	Storage	Minimum Value	Maximum Value
	(Bytes)	(Signed/Unsigned)	(Signed/Unsigned)
TINYINT	1	-128	127

Type	Storage	Minimum Value	Maximum Value
	(Bytes)	(Signed/Unsigned)	(Signed/Unsigned)
		0	255
SMALLINT	2	-32768	32767
		0	65535
MEDIUMINT	3	-8388608	8388607
		0	16777215
INT	4	-2147483648	2147483647
		0	4294967295
BIGINT	8	-9223372036854775808	9223372036854775807
		0	18446744073709551615

MySQL INT data type examples

Let's look at some examples of using the integer data type.

1) Using INT for a column example

Because integer type represents exact numbers, you usually use it as the primary key of a table. In addition, the `INT` column can have an `AUTO_INCREMENT` attribute.

When you insert a `NULL` value or 0 into the `INT AUTO_INCREMENT` column, the value of the column is set to the next sequence value. Notice that the sequence value starts with 1.

When you insert a value, which is not `NULL` or zero, into the `AUTO_INCREMENT` column, the column accepts the value. In addition, the sequence is reset to the next value of the inserted value.

First, create a new table named `items` with an integer column as the primary key:

```
CREATE TABLE items (
    item_id INT AUTO_INCREMENT PRIMARY KEY,
    item_text VARCHAR(255)
);
```

You can use either `INT` or `INTEGER` in the `CREATE TABLE` statement above because they are interchangeable. Whenever you insert a new row into the `items` table, the value of the `item_id` column is increased by 1.

Next, the following `INSERT` statement inserts three rows into the `items` table.

```
INSERT INTO
    items(item_text)
VALUES
    ('laptop'),
    ('mouse'),
    ('headphone');
```

Then, query data from the `items` table using the following `SELECT` statement:

```
SELECT * FROM items;
```

Output:

```
+---------+-----------+
| item_id | item_text |
+---------+-----------+
|       1 | laptop    |
|       2 | mouse     |
|       3 | headphone |
+---------+-----------+
3 rows in set (0.00 sec)
```

After that, insert a new row whose value of the `item_id` column is specified explicitly.

```
INSERT INTO items(item_id,item_text)
VALUES(10,'Server');
```

Since the current value of the `item_id` column is 10, the sequence is reset to 11. If you insert a new row, the `AUTO_INCREMENT` column will use 11 as the next value.

```
INSERT INTO items(item_text)
VALUES('Router');
```

Finally, query the data of the `items` table again to see the result.

```
SELECT * FROM items;
```

Output:

```
+---------+-----------+
| item_id | item_text |
+---------+-----------+
|       1 | laptop    |
|       2 | mouse     |
|       3 | headphone |
|      10 | Server    |
|      11 | Router    |
+---------+-----------+
5 rows in set (0.00 sec)
```

2) Using INT UNSIGNED example

First, create a table called `classes` that has the column `total_member` with the unsigned integer data type:

```
CREATE TABLE classes (
    class_id INT AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    total_member INT UNSIGNED,
    PRIMARY KEY (class_id)
);
```

Second, insert a new row into the `classes` table:

```
INSERT INTO classes(name, total_member)
```

```
VALUES('Weekend',100);
```

It worked as expected.

Third, attempt to insert a negative value into the `total_member` column:

```
INSERT INTO classes(name, total_member)
VALUES('Fly',-50);
```

MySQL issued the following error:

```
Error Code: 1264. Out of range value for column 'total_member' at row 1
```

> Note that the display width has been deprecated. Additionally, the ZEROFILL attribute has also been deprecated, and the suggested alternatives are to use LPAD for zero-padding numbers or to store the formatted numbers in a CHAR column.

Summary

- `INT` represents the integer type.
- MySQL offers various variants of the `INT` type including `TINYINT`, `SMALLINT`, `MEDIUMINT`, and `BIGINT`.

Was this tutorial helpful? 👍 👎

PREVIOUSLY
MySQL BIT Data Type

UP NEXT
MySQL BOOLEAN 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

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

How to Kill a Process in MySQL

SITE LINKS

Donation ❤️

Contact Us

About

Privacy Policy

OTHERS

MySQL Cheat Sheet

MySQL Resources

MySQL Books