

Home » MySQL Basics » MySQL Primary Key

# MySQL Primary Key

Discover more

 [MySQL learning resources](#)

 [Sample MySQL databases](#)

 [Online MySQL courses](#)

 [database](#)

 [Programming language IDEs](#)

 [Python database programming](#)

... 

**Summary:** in this tutorial, you will learn how to use the **MySQL primary key** constraint to create the primary key for a table.

## Introduction to the MySQL primary key

In MySQL, a primary key is a column or a set of columns that uniquely identifies each row in the table. A primary key column must contain unique values.

If the primary key consists of multiple columns, the combination of values in these columns must be unique. Additionally, a primary key column cannot contain NULL.

A table can have either zero or one primary key, but not more than one.

## Defining a single-column primary key

Typically, you define a primary key for a table when you [create the table](#). Here's the syntax for defining the primary key that consists of one column:

```
CREATE TABLE table_name(
    column1 datatype PRIMARY KEY,
    column2 datatype,
    ...)
```

```
);
```

In this syntax, you define the `PRIMARY KEY` constraint as a *column constraint*.

Additionally, you can put the `PRIMARY KEY` at the end of the column list:

```
CREATE TABLE table_name(  
    column1 datatype,  
    column2 datatype,  
    ...,  
    PRIMARY KEY(column1)  
);
```

In this syntax, you define the `PRIMARY KEY` constraint as a *table constraint*.

## Defining a multi-column primary key

If the primary key consists of two or more columns, you need to use a table constraint to define the primary key:

```
CREATE TABLE table_name(  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ...,  
    PRIMARY KEY(column1, column2)  
);
```

In this syntax, you list the primary key columns inside parentheses, separated by commas, followed by the `PRIMARY KEY` keywords.

## Adding a primary key to an existing table

If an existing table does not have a primary key, you can add a primary key to the table using the `ALTER TABLE ... ADD PRIMARY KEY` statement:

```
ALTER TABLE table_name  
ADD PRIMARY KEY(column1, column2, ...);
```

## Removing a primary key

In practice, you'll rarely remove a primary key. However, if you want to do so, you can use the

```
ALTER TABLE ... DROP PRIMARY KEY statement:
```

```
ALTER TABLE table_name  
DROP PRIMARY KEY;
```

## MySQL PRIMARY KEY examples

We'll explore some examples of defining primary keys.

### 1) Defining a single-column primary key example

The following example [creates a table](#) called `products`, which has the `id` column as the primary key:

```
CREATE TABLE products(  
    id INT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL  
);
```

When you [insert data](#) into the `products` table, you need to ensure the uniqueness of values in the `id` column. For example:

```
INSERT INTO products (id, name)  
VALUES  
    (1, 'Laptop'),  
    (2, 'Smartphone'),  
    (3, 'Wireless Headphones');
```

If you attempt to insert a duplicate value into the primary key column, you'll get an error. For example:

```
INSERT INTO products (id, name)  
VALUES  
    (1, 'Bluetooth Speaker');
```

Error:

```
ERROR 1062 (23000): Duplicate entry '1' for key 'products.PRIMARY'
```

The output indicates that MySQL found a duplicate entry 1 for the primary key of the `products` table.

Keeping track of primary key values manually can be challenging. To simplify this process, MySQL provides the [AUTO\\_INCREMENT](#) attribute, which automatically assigns a unique value to the primary key each time you insert a new record.

## 2) Defining a single-column primary key with AUTO\_INCREMENT attribute example

The following statements re-create the `products` table with the primary key that uses the `AUTO_INCREMENT` attribute:

```
DROP TABLE products;

CREATE TABLE products(
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL
);
```

Now, you can insert new rows into the `products` table without having to provide the values for the primary key column. For example:

```
INSERT INTO products (name)
VALUES
    ('Laptop'),
    ('Smartphone'),
    ('Wireless Headphones');
```

MySQL automatically generates sequential integer values for the `id` column when a new row is inserted.

Here's the contents of the `products` table:

```
SELECT * FROM products;
```

Output:

```
+----+-----+
| id | name           |
+----+-----+
| 1  | Laptop         |
| 2  | Smartphone     |
| 3  | Wireless Headphones |
+----+-----+
3 rows in set (0.00 sec)
```

### 3) Defining a multi-column primary key example

We'll create a new table called `customers` :

```
CREATE TABLE customers(
    id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(255) NOT NULL,
    last_name VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL
);
```

Suppose each customer has some favorite products and each product is favored by some customers.

To model this relationship, you need to create a table called `favorites` :

```
CREATE TABLE favorites(
    customer_id INT,
    product_id INT,
    favorite_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY(customer_id, product_id)
);
```

The `favorites` table has a primary that consists of two columns `customer_id` and `product_id`.

Note that in the [foreign key tutorial](#), you'll learn how to define a foreign key for the `customer_id` column that references the `id` column of the `customers` table and a foreign key for the `product_id` column that references the `id` column of the `products` table.

## 4) Adding a primary key to a table example

The following statement creates a table called `tags` without a primary key:

```
CREATE TABLE tags(
    id INT,
    name VARCHAR(25) NOT NULL
);
```

To make the `id` column the primary key, you use the `ALTER TABLE ... ADD PRIMARY KEY` statement:

```
ALTER TABLE tags
ADD PRIMARY KEY(id);
```

## 5) Removing the primary key from a table

The following statement removes the primary key from the `tags` table:

```
ALTER TABLE tags
DROP PRIMARY KEY;
```

## Summary

- A primary key is a unique identifier for a row in a table.
- Use the `PRIMARY KEY` constraint to define a primary key for a table.

Was this tutorial helpful?



ADVERTISEMENTS

PREVIOUSLY

[How to Use the MySQL Generated Columns](#)

UP NEXT

[MySQL Foreign Key](#)

ADVERTISEMENTS

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

ADVERTISEMENTS

## **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

ADVERTISEMENTS

### **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](#)

ADVERTISEMENTS

### **ABOUT MYSQL TUTORIAL**

#### **WEBSITE**

MySQLTutorial.org helps you master MySQL quickly, easily, and with enjoyment. Our

### **LATEST TUTORIALS**

[MySQL Port](#)

[MySQL Commands](#)

[innodb\\_dedicated\\_server:](#)

### **SITE LINKS**

[Donation](#) 

[Contact Us](#)

[About](#)

tutorials make learning MySQL a breeze.

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

[More About Us](#)

[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](#)

[Privacy Policy](#)

**OTHERS**

[MySQL Cheat Sheet](#)

[MySQL Resources](#)

[MySQL Books](#)

Copyright © 2008 - Present by [www.mysqltutorial.org](http://www.mysqltutorial.org). All Rights Reserved.