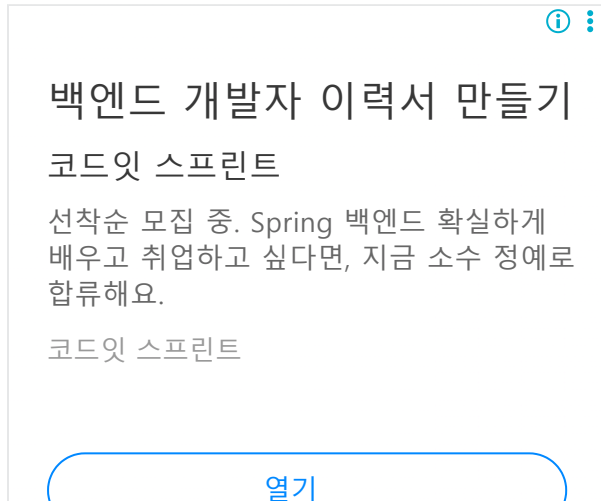


# MySQL BINARY Data Type



**Summary:** in this tutorial, you will learn about MySQL `BINARY` data type and how to use it to store fixed-length binary data.

## Introduction to MySQL BINARY data type

The `BINARY` data type is used to store fixed-length binary data. For example, you can use `BINARY` data type for columns that store hashes and checksums such as `SHA-256` because these values have a fixed length.

To declare a column that uses the `BINARY` data type, you specify the maximum length of binary data it can hold:

```
column_name BINARY(size);
```

In this syntax, the `size` specifies the number of bytes that the column name can store.

### Right-padding with 0x00 (zero bytes)

[When you insert a binary value](#) whose length is less than the specified length for the `BINARY` column, MySQL will automatically pad the value with zero bytes (0x00) on the right side to reach the defined `size` length.

## No trailing byte removal for retrievals

When you retrieve a value from the `BINARY` column, MySQL does not remove any trailing zero bytes that were padded during insertion.

In other words, if you inserted a binary value and it was right-padded with zero bytes, those zero bytes will be present when you retrieve the data.

## All bytes are significant in comparisons

When comparing `BINARY` value in the `WHERE` clause, `ORDER` clause, or `DISTINCT`, MySQL considers all bytes.

This means that even trailing zero bytes can affect the comparison results, and MySQL will not consider two `BINARY` values are equal unless all of their bytes match.

## 0x00 and space differ in comparisons

MySQL treats the zero bytes (0x00) and the space character (0x20) differently in comparisons.

If you have a `BINARY` column with values that contain zero bytes and space characters, MySQL will not consider these values to be equal.

Additionally, MySQL places null bytes before space characters in sorting operations (e.g., `ORDER BY`).

## MySQL BINARY data type example

We'll take an example of using the `BINARY` data type to store `SHA-256` hashes.

First, [create a table](#) to store the `SHA-256` hashes:

```
CREATE TABLE binary_demo(  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  data BINARY(32) -- 32 bytes for SHA-256  
);
```

The `binary_demo` has two columns:

- id: An [auto-incremented primary key](#) column.

- data: A `BINARY` column with a fixed size of 32 bytes to store `SHA-256` hashes.

Second, insert a `SHA-256` hash into the table:

```
INSERT INTO binary_demo(data)
VALUES (UNHEX(SHA2('Hello', 256)));
```

The `SHA2('Hello', 256)` computes the `SHA-256` hash of the string 'Hello'.

The `UNHEX()` function converts the hexadecimal representation of the `SHA-256` hash into binary data before inserting it into the `BINARY` column.

Third, retrieve data from the `BINARY` column and convert the data back to its hexadecimal using the `HEX()` function:

```
SELECT HEX(data)
FROM binary_demo WHERE id = 1;
```

Output:

```
+-----+-----+
| id | data |
+-----+-----+
| 1 | 0x185F8DB32271FE25F561A6FC938B2E264306EC304EDA518007D1764826381969 |
+-----+-----+
1 row in set (0.00 sec)
```

## Summary

- Use `BINARY` data type to store fixed-length binary data such as hashes or `UUID`.

Was this tutorial helpful?



Discover more

[⚡ MySQL consulting service](#)

[⚡ MySQL training course](#)

[⚡ Server monitoring tools](#)

[⚡ database](#)

[⚡ Database](#)

[⚡ Programming language IDEs](#)

[⚡ PHP PDO library](#)

[⚡ Backup and recovery software](#)

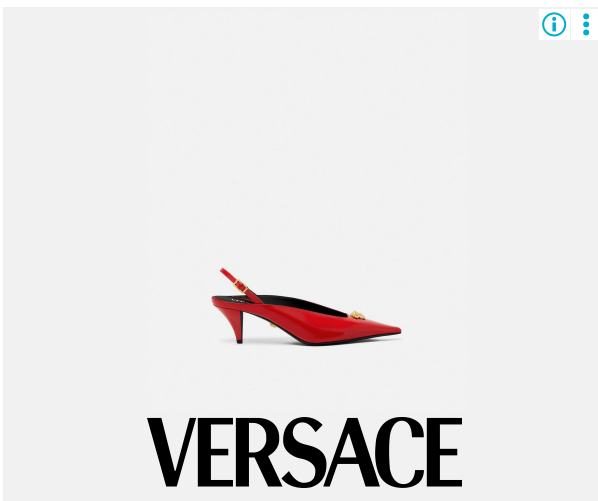
PREVIOUSLY

[MySQL TEXT Data Type](#)

UP NEXT

[MySQL VARBINARY Data Type](#)

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 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](#)

## SITE LINKS

[Donation](#) ❤️

[Contact Us](#)

[About](#)

[Privacy Policy](#)

## OTHERS

[MySQL Cheat Sheet](#)

[MySQL Resources](#)

[MySQL Books](#)

MySQL InnoDB Architecture

How to Kill a Process in MySQL

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