



# DexterityDB Plugin Technical Overview

## Introduction

The DexterityDB plugin is a high-performance database storage engine for MySQL and MariaDB databases. It is designed for OLAP applications on very large datasets. It excels in read-intensive analytical environments (i.e. as part of an ETL process), but performs well at writes as well, making it a good solution for most non-transactional use cases. DexterityDB stores data in custom-built, hybrid data structures that are tuned for modern day data needs.

## Plugin Architecture

Running MySQL or MariaDB with the DexterityDB plugin doesn't change the database architecture. DexterityDB replaces InnoDB (MySQL & MariaDB's default storage engine) at the storage engine layer in order to optimize how data is stored and speed up queries. MySQL and MariaDB have full support for this type of integration so no custom builds or versions are required. This change is mostly behind the scenes and users will probably not even notice a difference between the operation of the engines apart from the massive speed benefits that DexterityDB provides. There are, however, subtle differences which may impact peripheral infrastructure. Highlighted characteristics are as follows:

## Supported Data Types

The DexterityDB storage engine supports most of the data types in MySQL/MariaDB. The only limitation is lack of support for "blob" and "text" types ("tinytext," "mediumtext," and "longtext"). These are in development. The data types supported by the plugin include, but are not limited to, the ones in the table below:

Integer	Float	String	Date/Time	Boolean	Decimal

tiny	float	varchar	timestamp	bit	decimal
short	double	var_string	date		newdecimal
int24			time		
long			datetime		
longlong			year		
			newdate		
			Timestamp2		
			datetime2		
			time2		

## Transactions

Currently, DexterityDB does not support all of the transactional features that are present in InnoDB, so it is not recommended to switch to DexterityDB if your table is mostly used in a write-heavy, transactional use case. However, DexterityDB works well to support transactional use cases when used as a read-only table to obtain information in executing transactions.

## High Availability and Replication

DexterityDB is working with the Galera team to become the second storage engine to be officially supported by Galera Cluster! But for the time being, DexterityDB already works with binlog replication in MySQL/MariaDB Master-Slave configurations and the Spider Storage Engine.

## Install the DexterityDB MySQL/MariaDB Plugin

DexterityDB acts like a normal MySQL/MariaDB plugin and is installed in the same way as many other popular storage engines. Currently DexterityDB is compatible with MySQL 5.6 and above, and MariaDB 10.1 and above.

1. Since DexterityDB is a custom storage engine, it is not included with default MySQL/MariaDB installations. Therefore, the dynamically linked library file (.dll on Windows) or shared object file (.so on Linux) must be downloaded and dropped into the plugin folder of the database instance:
  - a. For default installations of MySQL 8.0, this folder can be found at:
    - i. Windows: C:\Program Files\MySQL\MySQL Server 8.0\lib\plugin
    - ii. Ubuntu: /usr/lib/mysql/plugin
  - b. For default installations of MariaDB 10.3, this folder can be found at:
    - i. Windows: C:\Program Files\MariaDB 10.3\lib\plugin
    - ii. Ubuntu: /usr/lib/mariadb/plugin
2. Once that has been done, the DexterityDB storage engine can be installed by running the following SQL query in MySQL Workbench or any other tool that can be used to run SQL queries on the server.

```
INSTALL PLUGIN dexteritydb SONAME 'ha_dexteritydb.dll';
```

3. From this point on, DexterityDB can be treated like any other storage engine that is bundled with MySQL/MariaDB and usage is identical. In order to use DexterityDB as the default storage engine for all newly created tables, simply run:

```
SET default_storage_engine = DEXTERITYDB;
```

To create a new table without setting it as default, you can run:

```
CREATE TABLE my_table (id INT) ENGINE = DEXTERITYDB;
```

To alter a current table, you can run:

```
ALTER TABLE my_table ENGINE = DEXTERITYDB;
```

## Find Out More

To learn more, contact us at [contact@dexteritydb.com](mailto:contact@dexteritydb.com)