

PHP and Relational Databases



Reza Salehi

CLOUD CONSULTANT

@zaalion linkedin.com/in/rezasalehi2008



Overview



MySQL, MariaDB and MS SQL Server

Connecting to MySQL in PHP

Connecting to MariaDB in PHP

Connecting to MS SQL Server in PHP

Other relational databases

Demo:

- Connecting to MySQL, MariaDB and MS SQL in PHP



PHP and Relational Databases



Working with Relational Databases in PHP

Directly via PHP

Using corresponding databases extensions

Using PDO

Data access layer which uses a unified API

Other libraries such as Doctrine ORM

Database abstraction layer



PDO vs. Database Extensions



PDO supports 12+ different databases



If switching between relational databases, it makes sense to use PDO



Both approaches are object oriented



Both offer protection against SQL injection (prepared statements)



Relational Databases to Use in This Course





The world's most popular open source database

[Contact MySQL](#) | [Login](#) | [Register](#)[MySQL.COM](#)[DOWNLOADS](#)[DOCUMENTATION](#)[DEVELOPER ZONE](#)[MySQL Server](#)[MySQL Enterprise](#)[Workbench](#)[InnoDB Cluster](#)[MySQL NDB Cluster](#)[Connectors](#)[More](#)

version 5.7 ▾

 Search this Manual [Documentation Home](#)

MySQL Installation Guide

- [Preface and Legal Notices](#)
- [Installing and Upgrading MySQL](#)
- [General Installation Guidance](#)

- [Installing MySQL on Unix/Linux Using Generic Binaries](#)
- [Installing MySQL from Source](#)
- [Installing MySQL on Microsoft Windows](#)
- [Installing MySQL on macOS](#)
- [Installing MySQL on Linux](#)
- [Installing MySQL on Solaris](#)
- [Postinstallation Setup and Testing](#)
- [Upgrading MySQL](#)
- [Downgrading MySQL](#)
- [Environment Variables](#)
- [Perl Installation Notes](#)

MySQL Installation Guide

Abstract

This is the MySQL Installation Guide from the MySQL 5.7 Reference Manual.

For legal information, see the [Legal Notices](#).

For help with using MySQL, please visit the [MySQL Forums](#), where you can discuss your issues with other MySQL users.

Document generated on: 2020-08-15 (revision: 67020)

Table of Contents

- [Preface and Legal Notices](#)
- [1 Installing and Upgrading MySQL](#)
- [2 General Installation Guidance](#)
- [3 Installing MySQL on Unix/Linux Using Generic Binaries](#)
- [4 Installing MySQL from Source](#)
- [5 Installing MySQL on Microsoft Windows](#)
- [6 Installing MySQL on macOS](#)
- [7 Installing MySQL on Linux](#)



Get Started for Free

Test Drive MariaDB Server Today.

MariaDB Server, with its continual open source innovation, is a modern relational database that business can count on – with features you can't find in MySQL or Postgres. In fact, MariaDB Server 10.5 is the leading open source database. Its list of features include:

- Columnar analytics at scale
- Temporal tables (system-versioned and application-time)
- Instant schema changes: ADD COLUMN, DROP COLUMN and MODIFY COLUMN
- PL/SQL and JSON support
- Versatility through purpose-built storage engines
- Data obfuscation and full/partial data masking

Test out the community version of MariaDB Server today – and when you're ready for production, transition to the new MariaDB Enterprise Server, a hardened version engineered specifically for production deployments.

Download MariaDB Server 10.5

First Name: *

Last Name: *

Job Title: *

Business Email: *

Company Name: *

Country: *

Business Phone:

Try SQL Server on-premises or in the cloud



SQL Server 2019 on-premises

Build intelligent, mission-critical applications using a scalable, hybrid data platform for demanding workloads. Get started with a 180-day free trial of SQL Server 2019 on Windows.

[Download free trial ↓](#)

SQL Server 2019 on Azure

Get started on Azure SQL, the family of SQL cloud databases that provide flexible options for application migration, modernization, and development. Built on the same SQL Server engine, easily rehost onto SQL Server 2019 on Azure Virtual Machines using preconfigured images on Linux and Windows, or modernize on the latest SQL Server version with fully-managed Azure SQL Database or Azure SQL Managed Instance. Take advantage of unique built-in security and manageability to automate tasks like patching and backups, and save with Azure Hybrid Benefit by reusing your existing on-premises licenses.

[Choose the best Azure SQL service for your needs >](#)

Or, download a free specialized edition



Developer

SQL Server 2019 Developer is a full-featured free edition, licensed for use as a development and test database in a non-production environment.



Express

SQL Server 2019 Express is a free edition of SQL Server, ideal for development and production for desktop, web, and small server applications.



Contact Sales

Search Q

My account

Portal

Reza

Overview

Products

Documentation

Pricing

Training

Marketplace

Partners

Support

Blog

More

Free account

[Home](#) / [Products](#) / [Azure SQL Database](#)

Azure SQL Database

Build apps that scale with the pace of your business with managed and intelligent SQL in the cloud

[Start for free >](#)

Azure SQL family

Product overview

Features

Security

Pricing

Customer stories

Documentation

Updates

FAQs

Build your next app on fully managed Azure SQL Database

Part of the [Azure SQL family](#), Azure SQL Database is the intelligent, scalable, relational database service built for the cloud. It's evergreen and always up to date, with AI-powered and automated features that optimise performance and durability for you. Serverless compute and Hyperscale storage options automatically scale resources on demand, so you can focus on building new applications without worrying about storage size or resource management.

[Chat with sales](#)

PHP supports other
relational database as well!



PHP and MySQL



PHP and MySQL

PHP 5 and later

MySQLi extension ("i" stands for improved), or PDO

Earlier PHP versions

MySQL extension (deprecated in 2012)



Steps to Use MySQL in PHP

Install MySQLi extension

Configuring PHP to load the MySQLi extension

Open a MySQL connection

Using “new mysqli()” command

Perform CRUD operations

Using OOP, or procedural approach



PHP: Installation - Manual

<https://www.php.net/manual/en/mysqli.installation.php>

php Downloads Documentation Get Involved Help Search

PHP 8.0.0beta2 Released!

PHP Manual > Function Reference > Database Extensions > Vendor Specific Database Extensions > MySQL > MySQLi > Installing/Configuring

« Requirements Runtime Configuration »

Change language: English Edit Report a Bug

Installation

The *mysqli* extension was introduced with PHP version 5.0.0. The MySQL Native Driver was included in PHP version 5.3.0.

Installation on Linux

The common Unix distributions include binary versions of PHP that can be installed. Although these binary versions are typically built with support for the MySQL extensions, the extension libraries themselves may need to be installed using an additional package. Check the package manager that comes with your chosen distribution for availability.

For example, on Ubuntu the *php5-mysql* package installs the *ext/mysql*, *ext mysqli*, and *pdo_mysql* PHP extensions. On CentOS, the *php-mysql* package also installs these three PHP extensions.

Alternatively, you can compile this extension yourself. Building PHP from source allows you to specify the MySQL extensions you want to use, as well as your choice of client library for each extension.

The MySQL Native Driver is the recommended client library option, as it results in improved performance and gives access to features not available when using the MySQL Client Library. Refer to [What is PHP's MySQL Native Driver?](#) for a brief overview of the advantages of MySQL Native Driver.

The */path/to/mysql_config* represents the location of the *mysql_config* program that comes with MySQL Server.

mysqli compile time support matrix

PHP Version	Default	Configure Options: myslnd	Configure Options: <i>libmysqlclient</i>	Changelog
5.4.x and above	myslnd	--with-mysqli	--with-mysqli=/path/to/mysql_config	myslnd is the default
5.3.x	libmvsalclient	--with-mysqli=myslnd	--with-mysqli=/path/to/mysql_config	mvsalnd is supported

```
<?php  
  
$servername = "localhost";  
  
$username = "username";  
  
$password = "password";  
  
// Open MySQL connection  
  
$conn = new mysqli($servername, $username, $password);  
  
// Check connection  
  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}  
  
echo "Successfully connected to MySQL!";  
  
?>
```

◀ MySQL Server name/IP

◀ Database credentials

◀ Open a MySQL connection (Object Oriented)

◀ Exit if failed, print the connection error



```
<?php

$rvname = "localhost";
$username = "username";
$password = "password";

// Open MySQL connection
$conn = mysqli_connect($rvname, $username, $password);

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

echo "Successfully connected to MySQL!";

?>
```

◀ MySQL Server name/IP

◀ Database credentials

◀ Open a MySQL connection (Procedural)

◀ Exit if failed, print the connection error



MySQL Operations

Create database

Insert data

Update data

Create table

Read data

Delete data



```
$sql = "CREATE DATABASE myDB";  
  
if ($conn->query($sql) === TRUE) {  
    echo "Successfully created the database!";  
}  
else {  
    echo "Error creating database: " . $conn->error;  
}
```

Create a MySQL Database **MySQLi Object-oriented**



```
$sql = "CREATE DATABASE myDB";  
  
if (mysqli_query($conn, $sql) === TRUE) {  
    echo "Successfully created the database!";  
} else {  
    echo "Error creating database: " . $conn->error;  
}
```

Create a MySQL Database **MySQLi procedural**



```
$sql = "CREATE TABLE Person (  
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
firstname VARCHAR(30) NOT NULL,  
lastname VARCHAR(30) NOT NULL,  
email VARCHAR(50))";
```

Create a MySQL Table



```
$sql = "INSERT INTO Person (firstname, lastname, email)  
VALUES ('John', 'Smith', 'john@example.com')"
```

Insert Data into MySQL



```
// object-oriented
```

```
$conn->multi_query($sql)
```

```
// procedural
```

```
mysqli_multi_query($conn, $sql)
```

Insert Multiple Rows at Once



```
$stmt = $conn->prepare("INSERT INTO MyGuests (firstname, lastname, email)
VALUES (?, ?, ?)");
$stmt->bind_param("sss", $firstname, $lastname, $email);

$firstname = "John";
$lastname = "Smith";
$email = "john@example.com";
$stmt->execute();
```

Insert Data using Prepared Statements



Use Prepared Statements to
avoid SQL injection attacks!



```
$sql = "SELECT firstname, lastname, email FROM person
```

Select Data from MySQL



```
$sql = "UPDATE person SET firstname='Reza' WHERE Id=301
```

Update MySQL Data



```
$sql = "DELETE FROM person WHERE Id=301
```

Delete Data from MySQL



Remember to close the
MySQL connection after
done with it!



```
// object-oriented
```

```
$conn->close()
```

```
// procedural
```

```
mysqli_close($conn)
```

Closing MySQL Connection



PHP and MariaDB



“The PHP connectors for MySQL generally work with MariaDB as well.”

mariadb.com/kb/en/php



PHP and Microsoft SQL Server



SQLSRV extension



Azure **SQL Database**



Microsoft **SQL Server**



PHP Extensions for MS SQL Server

Mssql extension

Removed in PHP 7.0.0!

SQLSRV extension

Supported by Microsoft



PHP: Introduction - Manual x + https://www.php.net/manual/en/intro.mssql.php

php Downloads Documentation Get Involved Help Search

PHP 8.0.0beta2 Released!

PHP Manual > Function Reference > Database Extensions > Vendor Specific Database Extensions > Mssql « Mssql Installing/Configuring »

Change language: English Edit Report a Bug

Introduction

Warning This feature was *REMOVED* in PHP 7.0.

Alternatives to this feature include:

- [PDO_SQLSRV](#)
- [PDO_ODBC](#)
- [SQLSRV](#)
- [Unified ODBC API](#)

These functions allow you to access MS SQL Server database.

This extension is not available anymore on Windows with PHP 5.3 or later.

SQLSRV, an alternative extension for MS SQL connectivity is available from Microsoft: » <http://msdn.microsoft.com/en-us/sqlserver/ff657782.aspx>

User Contributed Notes 3 notes

[+ add a note](#)

▲ 4 ▼ rjaehnrich at gmail dot com 4 years ago

to use MSSQL-connections on Linux with PHP7 you can use PDO with PDO_DBLIB.

Install driver using this command:
sudo apt-get install php7.0-sybase

Mssql

» [Introduction](#)

[Installing/Configuring](#)

[Predefined Constants](#)

[Mssql Functions](#)

PHP: Installation - Manual

<https://www.php.net/manual/en/sqlsrv.installation.php>

php Downloads Documentation Get Involved Help Search

PHP 8.0.0beta2 Released!

PHP Manual > Function Reference > Database Extensions > Vendor Specific Database Extensions > SQLSRV > Installing/Configuring

« Requirements Runtime Configuration »

Change language: English Edit Report a Bug

Installation

The SQLSRV extension is enabled by adding appropriate DLL file to your PHP extension directory and the corresponding entry to the `php.ini` file. The SQLSRV download comes 8 driver files, four of which are for PDO support. If you are running non-thread-safe PHP (PHP 5.3), use the `php_sqlsrv_53_nts.dll` file. (You should use a non-thread-safe version if you are using IIS as your web server). If you are running thread-safe PHP, use the `php_sqlsrv_53_ts.dll` file. Similarly for PHP 5.4, use the `php_sqlsrv_54_nts.dll` or `php_sqlsrv_54_ts.dll` depending on whether your PHP installation is non-thread-safe or thread-safe.

The most recent version of the driver is available for download here: » [SQLSRV 4.0 download](#). If you need support for PHP 5.2 and/or PHP compiled with VC6, use the 2.0 release of the driver: » [SQLSRV 2.0 download](#).

For more information about SQLSRV requirements, see » [SQLSRV System Requirements](#).

The SQLSRV extension is only compatible with PHP 5 running on Windows. Since version 4.0 the SQLSRV extension is compatible only with PHP 7.0 running on Linux or Windows.

User Contributed Notes 3 notes

[+ add a note](#)

▲ 1 ▼ grant at seljax dot com 2 years ago

Running PHP 7.2 on IIS 10, I had to use the IIS Web Platform Components wizard to install the Microsoft PHP drivers version 5.2 x64 to get this command to work. Manually installing 4.0 drivers did not work.

▲ -3 ▼ simon 6 years ago

SQLSRV doesn't use `mssql_connect` - it is its own extension, therefore to connect you'd use `sqlsrv_connect`:

Installing/Configuring

Requirements

» **Installation**

Runtime Configuration

Resource Types

SQLSRV Operations

Insert data

Update data

Transactions

Select data

Delete data

...



```
<?php

$servername = "localhost, 8080";                                ◀ MS SQL Server name/IP, port

$connectionInfo = array( "Database"=>"dbName",
                        "UID"=>"userName",
                        "PWD"=>"password");

$conn = sqlsrv_connect( $serverName, $connectionInfo);           ◀ Database credentials

if( $conn ) {
    echo "Successfully connected to MS SQL!";
} else {
    echo "Connection failed!";
    die( print_r( sqlsrv_errors(), true));
}

?>
```

◀ Open a MS SQL connection

◀ Exit if failed, print the
connection error



```
$sql = "INSERT INTO person (firstname, lastname) VALUES (?, ?)" ;  
$params = array("Reza", "Salehi");  
$stmt = sqlsrv_query ($conn, $sql, $params);  
if( $stmt === false ) {  
    die( print_r (sqlsrv_errors(), true));  
}
```

Insert Data into MS SQL



```
if (sqlsrv_begin_transaction($conn) === false)
{
    die( print_r( sqlsrv_errors(), true ) );
}

$stmt1 = sqlsrv_query( $conn, $sql1, $params1
);
$stmt2 = sqlsrv_query( $conn, $sql2, $params2
);

if( $stmt1 && $stmt2 ) {
    sqlsrv_commit( $conn );
} else {
    sqlsrv_rollback( $conn );
}
```

◀ Begin the transaction

◀ Run command #1

◀ Run command #2

◀ Commit if both successful

◀ Rollback if not



PHP and Other Relational Databases



Other Relational Databases

Oracle

OCI8 extension

DB2

ibm_db2 extension

MS Access

ODBC

PostgreSQL

pgsql

SQLite

SQLite3 extension



PHP: Installation - Manual

<https://www.php.net/manual/en/oci8.installation.php>

php Downloads Documentation Get Involved Help Search

PHP 8.0.0beta2 Released!

PHP Manual > Function Reference > Database Extensions > Vendor Specific Database Extensions > OCI8 > Installing/Configuring

« Requirements Testing »

Change language: English Edit Report a Bug

Installation

Configuring PHP with OCI8

Review the previous [Requirements](#) section before configuring OCI8.

To enable the OCI8 extension, configure PHP with the option `--with-oci8`.

Before starting the web server, OCI8 typically requires several Oracle environment variables (see below) to locate libraries, point to configuration files, and set some basic properties such as the character set used by Oracle libraries. The variables must be set *before* any PHP process starts.

The PHP binary must link with the same, or more recent, major version of Oracle libraries as it was configured with. For example, if you build OCI8 with Oracle 11.2 libraries, then PHP should also be deployed and run with Oracle 11.2 libraries. PHP applications can connect to other versions of Oracle Database, since Oracle has client-server cross-version compatibility.

Installing OCI8 as a Shared Extension

The configuration `shared` option builds OCI8 as a shared library that can be dynamically loaded into PHP. Building a shared extension allows OCI8 to be upgraded easily without impacting the rest of PHP.

Configure OCI8 using one of the following configure options.

- If using the free » [Oracle Instant Client](#) libraries, then do:

Installing/Configuring

- Requirements
- » **Installation**
- Testing
- Runtime Configuration

PHP: Installation - Manual

<https://www.php.net/manual/en/ibm-db2.installation.php>

php Downloads Documentation Get Involved Help Search

PHP 8.0.0beta2 Released!

PHP Manual > Function Reference > Database Extensions > Vendor Specific Database Extensions > IBM DB2 > Installing/Configuring

« Requirements Runtime Configuration »

Change language: English Edit Report a Bug

Installation

To build the ibm_db2 extension, the DB2 application development header files and libraries must be installed on your system. DB2 does not install these by default, so you may have to return to your DB2 installer and add this option. The header files are included with the DB2 Application Development Client freely available for download from the IBM DB2 Universal Database » [support site](#).

If you add the DB2 application development header files and libraries to a Linux or Unix operating system on which DB2 was already installed, you must issue the command `db2iupd -e` to update the symbolic links to the header files and libraries in your DB2 instances.

ibm_db2 is a » [PECL](#) extension, so follow the instructions in [Installation of PECL extensions](#) to install the ibm_db2 extension for PHP. Issue the `configure` command to point to the location of your DB2 header files and libraries as follows:

```
bash$ ./configure --with-IBM_DB2=/path/to/DB2
```

The `configure` command defaults to `/opt/IBM/db2/V8.1`.

Note: Note for IIS users

If you are using the ibm_db2 driver with Microsoft Internet Information Server (IIS) you may have to do the following:

- Install DB2 with extended operating system security.
- Add the PHP binary path to the `PATH` system environment variable (default `C:\php\`).
- Create another system environment variable equal to the path where the `PHP.INI` file is located (eg: `PHPRC = C:\php\`).
- Add the `IUSR_COMPUTERNAME` to the `DB2USERS` group.

PHP: Introduction - Manual x + https://www.php.net/manual/en/intro.uodbc.php

php.net/manual/en/intro.uodbc.php

← → C 🔒 Search

php Downloads Documentation Get Involved Help

PHP 8.0.0beta2 Released!

PHP Manual > Function Reference > Database Extensions > Abstraction Layers > ODBC

Change language: English Edit Report a Bug

Introduction

In addition to normal ODBC support, the Unified ODBC functions in PHP allow you to access several databases that have borrowed the semantics of the ODBC API to implement their own API. Instead of maintaining multiple database drivers that were all nearly identical, these drivers have been unified into a single set of ODBC functions.

The following databases are supported by the Unified ODBC functions: » [Adabas D](#), » [IBM DB2](#), » [iODBC](#), » [Solid](#), and » [Sybase SQL Anywhere](#).

Note:
With the exception of iODBC, there is no ODBC involved when connecting to the above databases. The functions that you use to speak natively to them just happen to share the same names and syntax as the ODBC functions. However, building PHP with iODBC support enables you to use any ODBC-compliant drivers with your PHP applications. More information on iODBC, is available at » [www.iodbc.org](#) with the alternative unixODBC available at » [www.unixodbc.org](#).

User Contributed Notes

There are no user contributed notes for this page.

« ODBC Installing/Configuring »

ODBC

» [Introduction](#)

[Installing/Configuring](#)

[Predefined Constants](#)

[ODBC Functions](#)

PHP: Installation - Manual

<https://www.php.net/manual/en/pgsql.installation.php>

php Downloads Documentation Get Involved Help Search

PHP 8.0.0beta2 Released!

PHP Manual > Function Reference > Database Extensions > Vendor Specific Database Extensions > PostgreSQL > Installing/Configuring

« Requirements Runtime Configuration »

Change language: English Edit Report a Bug

Installation

In order to enable PostgreSQL support, `--with-pgsql[=DIR]` is required when you compile PHP. `DIR` is the PostgreSQL base install directory, defaults to `/usr/local/pgsql`. If shared object module is available, PostgreSQL module may be loaded using [extension](#) directive in `php.ini` or [dl\(\)](#) function.

User Contributed Notes 7 notes

[+ add a note](#)

▲ 9 ▼ Serjik 10 years ago

On a Windows server, configured with Apache, adding the following line to `httpd.conf` to load `libpq.dll` can save you a lot of time :

```
LoadFile "C:/Program Files/PostgreSQL/8.4/bin/libpq.dll"
```

Note that you will have to change your folder accordingly to the installation path and version of PostgreSQL you have installed. Also note that having Apache and PostgreSQL on the same server for production environments is not recommended.

Cheers,
Serjik

▲ 3 ▼ Anonymous 3 years ago

for php 7, "sudo apt install php7.0-pgsql" worked for me

Installing/Configuring

Requirements
» **Installation**
Runtime Configuration
Resource Types

PHP: Installation - Manual

<https://www.php.net/manual/en/sqlite3.installation.php>

php Downloads Documentation Get Involved Help Search

PHP 8.0.0beta2 Released!

PHP Manual > Function Reference > Database Extensions > Vendor Specific Database Extensions > SQLite3 > Installing/Configuring

« Requirements Runtime Configuration »

Change language: English Edit Report a Bug

Installation

The SQLite3 extension is enabled by default as of PHP 5.3.0. It's possible to disable it by using `--without-sqlite3` at compile time.

Windows users must enable `php_sqlite3.dll` in order to use this extension. This DLL is included with Windows distributions of PHP as of PHP 5.3.0.

Note: Additional setup on Windows as of PHP 7.4.0
In order for this extension to work, there are DLL files that must be available to the Windows system PATH. For information on how to do this, see the FAQ entitled "[How do I add my PHP directory to the PATH on Windows](#)". Although copying DLL files from the PHP folder into the Windows system directory also works (because the system directory is by default in the system's PATH), this is not recommended. *This extension requires the following files to be in the PATH: libsqlite3.dll*.

Note:
This extension was briefly a PECL extension but that version is only recommended for experimental use.

User Contributed Notes

[+ add a note](#)

There are no user contributed notes for this page.

Installing/Configuring

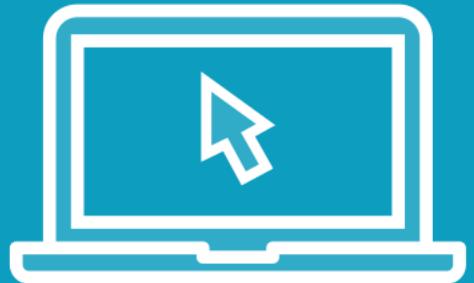
Requirements

» **Installation**

Runtime Configuration

Resource Types

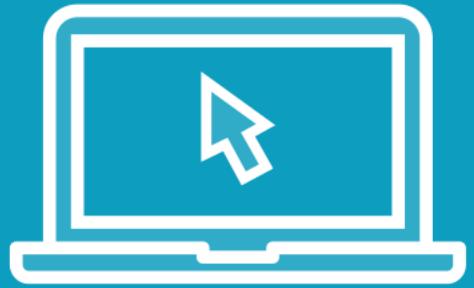
Demo



PHP and MySQL



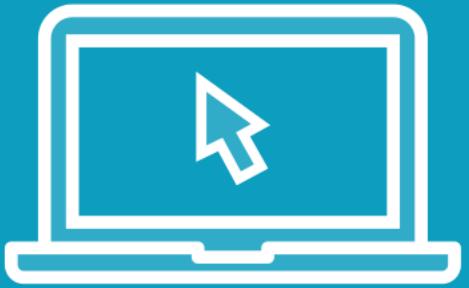
Demo



PHP and MariaDB



Demo



PHP and Microsoft SQL Server



Summary



MySQL, MariaDB and MS SQL

Connecting to MySQL, MariaDB and MS SQL in PHP

Other relational databases

Demo: Connecting to MySQL, MariaDB and MS SQL in PHP

