Code for every supported database

Writing PHP and SQL that works on all six databases Exponential supports - SQLite 3 (the default), PostgreSQL, MongoDB, MariaDB, Oracle and MySQL - the driver classes, the portable helpers of eZDBInterface, schema files…

Source file: `doc/eng-us/help/php-developers/code-for-every-supported-database.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/php-developers/code-for-every-supported-database.md

This page is for PHP developers whose code runs SQL. Exponential 6.0.x supports six databases, and an extension that ships for the community may meet any of them. The support statement with versions and settings is Database support; this page is about writing code that works on all of them.

Contents

1. The six databases and their drivers
2. Rules for portable SQL
3. The portable helpers
4. Schema: .dba files, not DDL
5. MongoDB is different
6. Checking which database runs
7. Testing across engines

The six databases and their drivers

In the project's order of preference:

Database	`DatabaseImplementation`	Driver class	PHP extension
1. SQLite 3 (the installer default)	`sqlite3`	`eZSQLite3DB`	`sqlite3`
2. PostgreSQL	`ezpostgresql`	`eZPostgreSQLDB`	`pgsql`
3. MongoDB	`mongodb`	`expMongoDB`	`mongodb` and the library `mongodb/mongodb`
4. MariaDB	`ezmysqli`	`eZMySQLiDB`	`mysqli`
5. Oracle	`ezoracle`	`eZOracleDB` (extension `ezoracle`)	`oci8`
6. MySQL	`ezmysqli`	`eZMySQLiDB`	`mysqli`

Microsoft SQL Server is possible with additional work (Database support).

Rules for portable SQL

Do	Avoid
lower case, unquoted table and column names (`acme_report_item`)	reserved words, mixed case, quoting with backticks
`arrayQuery( $sql, array( 'limit' => 20, 'offset' => 40 ) )`	`LIMIT 40, 20` in the SQL
standard joins and `WHERE` clauses	`UPDATE ... JOIN` (SQLite needs 3.33 and a rewrite; the driver rewrites the kernel's own)
integers for timestamps (`time()`)	engine-specific date types and functions
the helpers below for strings and bits	`CONCAT()`, `SUBSTRING()`, & written by hand
short transactions	long-running writes (SQLite allows one writer at a time)
`utf8mb4` on MariaDB and MySQL	`utf8` (three bytes), which loses characters outside the Basic Multilingual Plane

The portable helpers

`eZDBInterface` offers functions that each driver writes in its own dialect:

Helper	Gives
`$db->concatString( array( 'a', "'-'", 'b' ) )`	string concatenation
`$db->subString( 'title', 1, 10 )`	a substring expression
`$db->md5( 'title' )`	an MD5 expression
`$db->bitAnd( 'flags', 4 )`	a bitwise AND expression
`$db->generateSQLINStatement( $ids, 'id', false, true, 'int' )`	an `IN` list with cast values
`$db->escapeString( $value )`	the driver's string escaping
`$db->databaseServerVersion()`	the server version, when you must branch

Branching on the driver's class is the last resort; prefer a helper, or move the query into `eZPersistentObject`.

Schema: .dba files, not DDL

Do not ship `CREATE TABLE` statements for one engine. Describe your tables once in a `.dba` schema file in the extension's `share/` and let the schema handler (`lib/ezdbschema/`) generate the DDL for each engine; the kernel's own schema lives the same way, with per-engine SQL in `kernel/sql/`. Update scripts per engine live in `update/database/sqlite/`, `postgresql/` and `mysql/` (Database schema and .dba files).

MongoDB is different

The MongoDB adapter `expMongoDB` (since 6.0.14) never runs real SQL. It translates the kernel's SQL-shaped calls into `find`, `aggregate`, insert, upsert and delete operations:

- Hand-written SQL that it cannot translate returns an **empty result** and logs a warning; search the logs for `MONGO TODO`. Extensions with SQL of their own may need a MongoDB path.
- The adapter needs PHP 8.2 or later; the rest of the kernel runs on 8.0.
- The SQL query cache does not apply.
- There are no SQL update files; the upgrade check compares collections.

Code that uses `eZPersistentObject` and the content API works without changes far more often than hand-written SQL (MongoDB as the database, MongoDB porting reference).

Checking which database runs

PHP

```php
$db = eZDB::instance();
$name = $db->databaseName();    // 'sqlite', 'postgresql', 'mongo', 'mysql' or 'oracle'
```

MariaDB and MySQL both answer `mysql`, because they share the driver. On the command line:

Shell

```
php bin/php/console exp:ini get site.ini/DatabaseSettings/DatabaseImplementation
```

In the administration, **Setup > System information** has a **Database** card (Database support).

Testing across engines

Engine	How to test your code
SQLite	the default install; your development machine
PostgreSQL, MySQL	`tests/runtests.php --dsn ... --db-per-test`, as CI does (Database and live tests)
MariaDB	covered by the MySQL driver; test on MariaDB if your SQL is unusual
MongoDB	an installation with `--db=mongodb`; read the logs for `MONGO TODO`
Oracle	an installation with the `ezoracle` extension

At the least, test on SQLite and on one server database before you release.

Related pages

- SQL with eZDB, safely
- Databases and SQL standard
- Chapter 9: Databases
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.

Become an Exponential Partner

Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.

Partners: start here | How to apply

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.