Reviewing extension SQL for every engine

A database administrator's checklist for reviewing the SQL of an Exponential extension before it goes onto a site that does not run MySQL - backticks, table aliases, driver helpers, MySQL session statements, empty…

Source file: `doc/eng-us/help/database-administrators/portable-sql-for-extensions.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/database-administrators/portable-sql-for-extensions.md

This page is for database administrators who must approve an extension before it is installed on a site running PostgreSQL, SQLite, Oracle or MongoDB. Exponential's own SQL runs on every supported engine; extension code and hand-written SQL often do not, because most extensions were written against MySQL. This page lists what to look for and where. Source: 9.8 Cross-engine notes.

Contents

1. Where an extension's SQL lives
2. What to search for
3. Driver helpers
4. Semantics that differ
5. MongoDB
6. Testing
7. Checklist

Where an extension's SQL lives

- `share/db_schema.dba` and `share/db_data.dba`: the engine-neutral schema, the best case; the upgrade check reads it.
- `sql/<engine>/*.sql`: DDL per engine. An extension with only `sql/mysql/` has no tables for other engines yet.
- PHP: `eZPersistentObject` definitions (portable) and hand-written statements in `query()` and `arrayQuery()` calls (the ones to read).
- Update files of the extension's own releases.

What to search for

Shell

```
grep -rn '`' extension/<name> --include=*.php | grep -i 'select\|insert\|update\|delete'     # backticks
grep -rniE 'FROM [a-z_]+ AS [a-z_]+' extension/<name> --include=*.php                        # AS before a table alias
grep -rniE 'SET (NAMES|FOREIGN_KEY_CHECKS|SQL_MODE|AUTOCOMMIT)' extension/<name>              # MySQL session statements
grep -rniE 'LIMIT [0-9]+ *, *[0-9]+|IFNULL|GROUP_CONCAT|ON DUPLICATE KEY|REPLACE INTO' extension/<name>   # MySQL dialect
```

Finding	Problem	Fix
Backticks around names	PostgreSQL and Oracle reject them; the MongoDB adapter's parser does not expect them	name tables and columns plainly
`AS` before a table alias	Oracle rejects it (`ORA-00933`)	write the alias without `AS`; `AS` before a column alias is fine everywhere
`SET NAMES`, `SET FOREIGN_KEY_CHECKS` and similar	SQLite ignores them (the driver accepts and drops them); PostgreSQL understands only `SET NAMES`; Oracle has no such statement	leave them out of code meant for every engine
MySQL-only functions and syntax	fail elsewhere	use the driver's helpers or the `arrayQuery()` offset and limit parameters

Driver helpers

The drivers implement what differs between engines; code should call them instead of writing the SQL:

Helper	For
`$db->subString()`, `$db->concatString()`	substring and concatenation
`$db->md5()`	MD5 (PostgreSQL through `pgcrypto`, Oracle through `STANDARD_HASH`, SQLite through a registered function)
`$db->bitAnd()`, `$db->bitOr()`	bitwise operations (Oracle `BITAND`)
`$db->escapeString()`	quoting values
`$db->lastSerialID()`	the id of the last insert (PostgreSQL sequences, Oracle sequences)
`arrayQuery( $sql, array( 'offset' => ..., 'limit' => ... ) )`	paging
`$db->databaseName()`	`mysql`, `postgresql`, `sqlite`, `mongo` or `oracle`, when a branch is unavoidable

Semantics that differ

- **Empty strings**: Oracle returns NULL for `''` unless `OracleEmptyStringForNull=enabled`.
- **Case**: MySQL's default collations ignore case; PostgreSQL and Oracle compare with case; SQLite compares `=` with case and ignores ASCII case in `LIKE`. Compare `lower(...)` on both

sides where case must not matter (Character sets and collations).

- **Failure versus empty result**: a failed statement and an empty result are not reported alike on every engine. To ask whether a table exists, ask the catalog (the driver's `eZTableList()`), not a `SELECT` that may fail.
- **Index names** are database-wide on SQLite and PostgreSQL; the schema handlers create a clashing name as `<table>__<name>`. Keep names within the lint limits (The .dba schema format).
- **Integers**: column ranges differ (`$db::MAX_INT`); `$db->clampInteger()` keeps a value in range.
- **Error codes** differ (Database error codes and messages).

MongoDB

The adapter never runs SQL; it translates what it can and returns an **empty result** for the rest, logging `MONGO TODO`. An extension with hand-written SQL needs either SQL the adapter understands or a MongoDB path of its own. Click through every feature of the extension on a MongoDB staging copy and grep the logs (MongoDB: limits and operations).

Testing

1. Install the extension on a staging copy of the target engine.
2. Run its schema through `ezsqlinsertschema.php` or its installer; run the upgrade check.
3. Use every feature; watch `var/log/error.log` and the debug output with `SQLOutput=enabled` for your address.
4. Run the extension's own tests if it has them.

Checklist

- ☐ The extension has `share/db_schema.dba` or SQL files for the target engine.
- ☐ No backticks, no `AS` before table aliases, no MySQL-only session statements.
- ☐ Driver helpers used for strings, hashes, bits, paging and ids.
- ☐ Case-insensitive lookups use `lower(...)`.
- ☐ Tested on staging on the target engine without new errors.

Related pages

- Extension development
- Choosing an engine
- Database support
- 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.