The SQL query cache

What the optional SQL query cache of Exponential does for a database administrator - modes and storage, what is never cached, how writes invalidate results, why writes made outside Exponential require a clear, how to...

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

This page is for database administrators who want to know whether the SQL query cache helps their site, and what it means for writes they make themselves. The cache keeps the rows a `SELECT` returned and answers the same statement again until a write to one of its tables makes the result stale. It is off by default. It works with the MySQL, MariaDB, PostgreSQL, SQLite and Oracle drivers; the MongoDB driver does not use it. Source: The SQL query cache and `settings/querycache.ini`.

Contents

1. The settings
2. What is never cached
3. How writes invalidate
4. Writes outside Exponential
5. Switching it on
6. What is normal, what to look at
7. Checklist

The settings

`settings/querycache.ini [QueryCacheSettings]`, overridden in `settings/override/querycache.ini.append.php`:

Key	Default	Meaning
`Mode`	`off`	`off`: nothing kept. `request`: an identical SELECT is not run twice in one request. `shared`: results kept across requests as well
`Storage`	`auto`	Where `shared` keeps results: `auto` (APCu where the process has it, else files), `apcu`, or `file` (`var/<site>/cache/querycache/entries/`, shared by every web server that mounts `var/`)
`ApcuReserve`	`25`	Percent of APCu that must stay free; below it, results are kept for the request only
`MaxAge`	`300`	Longest life of a result in seconds, the bound for a write the cache cannot see
`MaxRows`	`5000`	Results with more rows are not kept
`ExcludeTables[]`	`ezsession`, `ezpending_actions`, `ezcollab_notification_rule`	Tables whose SELECTs are never cached

`[DebugSettings] Report=enabled` adds the cache's counters to the debug output.

What is never cached

- reads inside a transaction;
- statements whose tables cannot all be read from the text: a subquery, hint or `PARTITION` in `FROM`, a parenthesised join, a comment, a quote left unpaired or inside a quoted name, Oracle's `q''` and PostgreSQL's `E''` and dollar-quoted strings, anything that is not a plain `SELECT`. Each `SELECT` of `UNION`, `MINUS`, `INTERSECT` and `EXCEPT` is read. Where a backslash escapes depends on the driver: in MySQL's literals it does, in PostgreSQL's (with `standard_conforming_strings` on, the default; the driver reads the setting from the connection), SQLite's and Oracle's it does not;
- statements whose result is not a function of their tables: `NOW()`, `RAND()`, `UUID()`, `LAST_INSERT_ID()`, locks (`FOR UPDATE`), variables, `SQL_NO_CACHE`; PostgreSQL's `clock_timestamp()`, `txid_current()`, `pg_*()` functions and `FOR NO KEY UPDATE`; SQLite's `changes()`, `last_insert_rowid()` and `datetime('now')`;
- reads of temporary tables;
- results with more than `MaxRows` rows, and reads of `ExcludeTables`.

On Oracle also statements that read a sequence, the clock, `SYS_GUID`, `SYS_CONTEXT`, `DBMS_RANDOM` or the catalog.

How writes invalidate

`INSERT`, `REPLACE`, `UPDATE`, `DELETE`, `TRUNCATE`, `CREATE/ALTER/DROP TABLE`, index changes and `RENAME TABLE` mark the tables they name as written. A write whose tables cannot be read marks everything stale. Inside a transaction the tables are marked at the outermost `COMMIT`; on `ROLLBACK` nothing is marked. Other processes (scripts, the admin host, Velocity) see the change through a shared state file in `var/<site>/cache/querycache/`. The file is written under a lock on `state.lock`, which must be writable by every user the site runs as (FPM, Velocity, cron); on NFS it needs NFSv4 or `lockd`. A write that cannot take the lock makes every result stale, and **Setup > System information** says so in red. PostgreSQL's `TRUNCATE ... CASCADE` and `DROP ... CASCADE` reach tables the statement does not name, so they make everything stale too. Deleting the cache directory is safe: results kept under the state file that was deleted are never answered again.

Writes outside Exponential

The cache cannot see a write made with a SQL client, a `mysql < dump.sql`, a restore or another program. Results stay valid for up to `MaxAge` seconds. After every such write, and after every restore, clear it:

Shell

```
php bin/php/ezcache.php --clear-id=querycache --allow-root-user
```

or **Setup > Caches**. `--clear-all` and the tags `content` and `ini` include it.

Switching it on

1. Check APCu: **Setup > System information**, PHP section, or `php -m | grep apcu`.
2. Set `Mode=request` in `settings/override/querycache.ini.append.php`; clear the INI cache, reload PHP-FPM, restart Velocity. Browse for a day.
3. Set `Mode=shared` the same way. Watch the hit rate in **Setup > System information**, box **Database queries**.

To switch it off, set `Mode=off` and do the same; nothing needs deleting. The buttons in **Setup > Caches** and **Setup > System information** (clear the cache, reset its counters) need the `setup/managecache` policy.

What is normal, what to look at

- A hit rate of 30 to 40 percent right after a clear or a restart, rising as pages are visited.
- The SQL profiler counts only statements that reached the database: a warm shared-cache page shows 0 (Finding slow queries).

Symptom	Cause
A page shows old data after a change made outside Exponential	clear the query cache, as above
The hit rate stays low	a table the pages read is written on every request; System information lists the tables written last; exclude it or fix the writer
Many files in `entries/`	the `cache_cleanup` part does not run
APCu full, other caches evicted	keep `ApcuReserve` above 0, lower `MaxRows` or `MaxAge`, or raise `apc.shm_size`
`state.ser` not writable	`var/<site>/cache/querycache/` must be writable by every user the site runs as
"Writes could not take the state lock" in System information	make `state.lock` writable for every user the site runs as (on NFS: NFSv4 or `lockd`), then clear the query cache

When in doubt, set `Mode=off` and compare.

Checklist

☐ Every runbook that writes to the database outside Exponential ends with a query cache clear.

☐ `cache_cleanup` scheduled when `Storage` is `file` or `auto` without APCu.

☐ Hit rate and APCu memory watched after switching to `shared`.

Related pages

- Performance and scaling
- Caches
- Caching layers (for hosting companies)
- 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.