MySQL Storage Engine Audit & 8.2.0 → 8.4.8+ Upgrade Impact

Scope: VisoAdmin (`mvpold_db`) and KaaryamAPI (`mvpold_db_one`) — two Laravel codebases sharing the same local MySQL server (WAMP).
Current version (confirmed live): MySQL 8.2.0. **Target:** 8.4.8+.
Trigger: discovered while adding the Trade License Verification vendor-API to KaaryamAPI — a `DB::transaction()` rollback silently failed to undo test writes.
Date: 2026-08-24

TL;DR

For a local-only 8.2.0 → 8.4.8+ bump, the app **code** in both repos is fine — nothing found that the version jump itself breaks. But the current server's **my.ini has two non-default settings baked in**, and if the new 8.4.8 install doesn't carry them over (or isn't given equivalent replacements), it will break real things:

1. **root's auth plugin is `mysql_native_password`** — a plugin MySQL 8.4 turns **off by default**. Both apps connect as `DB_USERNAME=root`. Left as-is, **neither app will be able to connect to a fresh 8.4.8 server.**
2. **`sql_mode` is completely empty** on this server (no `STRICT_TRANS_TABLES`, no `ONLY_FULL_GROUP_BY`, nothing) — a fresh MySQL install defaults to a strict mode set. If 8.4.8 is installed with its own defaults instead of this server's current (blank) `sql_mode`, queries that currently only work *because* strict mode is off will start throwing errors.

Both are **server/ops config items**, not code changes — but they need to be handled deliberately during the upgrade, not left to a fresh install's defaults.

1. Blocker candidates specific to reaching 8.4.8+

1.1 mysql_native_password — MySQL 8.4 disables it by default

```
mysql> SELECT user, host, plugin FROM mysql.user;
user              host        plugin
mysql.infoschema  localhost   caching_sha2_password
mysql.session     localhost   caching_sha2_password
mysql.sys         localhost   caching_sha2_password
root              localhost   mysql_native_password   ← both apps connect as this user
```

Both `.env` files use `DB_USERNAME=root` with this exact account. Starting in MySQL 8.4, the `mysql_native_password` plugin is **disabled by default** (already deprecated since 8.0.34; 8.4 turns it off unless explicitly re-enabled). If the `root` account isn't converted before cutover, the connection will fail outright for both apps — this is the single biggest risk in this upgrade.

Fix — do one of:

- Convert the account ahead of the upgrade: `ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password BY '<password>';` (the PHP driver already fully supports this — `mysqlnd 8.2.13` handles `caching_sha2_password` natively, confirmed below).
- Or, if `mysql_native_password` must be kept, explicitly re-enable it on the 8.4.8 install via `--mysql-native-password=ON` in the new `my.ini`.

Converting the account is the cleaner fix and costs nothing on a local dev box.

1.2 sql_mode is blank — a fresh install won't be

```
mysql> SHOW VARIABLES LIKE 'sql_mode';
```

```
sql_mode   (empty)
```

Both apps' `config/database.php` set `'strict' => false`, so Laravel never overrides this at the session level — actual behavior is entirely whatever the server's global `sql_mode` is. Right now that's **nothing at all**: no `STRICT_TRANS_TABLES`, no `ONLY_FULL_GROUP_BY`, no `NO_ZERO_DATE`. A stock MySQL 8.x install defaults `sql_mode` to `ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION`.

Concretely, this codebase already contains queries that rely on the *absence* of `ONLY_FULL_GROUP_BY` — e.g. `VendorServicesInfo::getServices()` in KaaryamAPI (`app/Model/VendorServicesInfo.php`) does `GROUP BY vendor_services_locations.vendor_services_id` while selecting `uuid`, `name`, `vendor_services.description`, `images` — non-aggregated columns not covered by MySQL's functional-dependency detection in every case. This runs fine today because `sql_mode` is empty; it may not survive a fresh 8.4.8 default.

Fix: carry the current `my.ini`'s `sql_mode=` line (or an explicit empty override) into the new install's config, rather than accepting 8.4.8's out-of-the-box default. Longer-term, tightening `sql_mode` back to the MySQL default and fixing the handful of queries that depend on its absence would be the more correct fix — but that's a separate, larger cleanup from "make the version upgrade not break things."

1.3 Everything else checked — no issues

Area	Status
Charset/collation in `config/database.php`	OK Explicit `utf8mb4` / `utf8mb4_unicode_ci` in both apps, overriding the server's default (`utf8mb4_0900_ai_ci`) on every connection
PHP driver (pdo_mysql/mysqlnd)	OK `mysqlnd 8.2.13` — fully speaks `caching_sha2_password` and the 8.4 wire protocol
`Schema::table(...)->change()` / `renameColumn()`	OK VisoAdmin: none, doctrine/dbal not installed. KaaryamAPI: 3 already-applied migrations (won't re-run); dbal pinned at 2.13.0 — only a risk for future alter migrations
Deprecated `GROUP BY ... ASC/DESC`, `FULLTEXT` on MyISAM	OK None found in either repo
`CHANGE MASTER TO` / `SHOW SLAVE STATUS` (renamed to REPLICA terminology in 8.4)	OK Not used anywhere in app/database/config/routes in either repo
`mysql_upgrade` binary (removed in 8.4)	OK Not referenced in either repo
Reserved-word collisions added across 8.0.x	OK Already on 8.2 — no new exposure moving to 8.4.8

1.4 Deeper verification pass (server config file + live schema)

The items above were checked from the codebase side. This pass checked the server itself:

- **my.ini (`D:\wamp64\bin\mysql\mysql8.2.0\my.ini`) reviewed line by line** for now-removed/deprecated 8.4 startup variables. None found. Specifically ruled out `default_authentication_plugin` (removed in 8.4, would abort startup if present) — it isn't set there, meaning root's `mysql_native_password` (§1.1) is an **account-level** setting, not a config-file one, confirming `ALTER USER` is the correct fix. The file already uses modern replication naming (`skip_replica_start`, not the old `skip_slave_start`).
- Confirmed `sql_mode=""` and `default_storage_engine=MYISAM` are **explicit, deliberate lines** in that file — matches §1.2 and §2 exactly.
- **Column-level charset audit**: every column in both `mvpold_db` and `mvpold_db_one` is `utf8mb4` — queried `information_schema.COLUMNS` directly, zero legacy `utf8` / `utf8mb3` columns.
- **Views / triggers / stored procedures / events**: none exist in either database.
- **Reserved-word collision check**: cross-referenced actual table and column names in both databases against the words MySQL 8 added as reserved (`RANK`, `GROUPS`, `LAG`, `LEAD`, `WINDOW`, `RECURSIVE`, `ROW`, `ROWS`, `SYSTEM`, etc.) — zero real collisions. One table is named `events`; `EVENT` / `EVENTS` were never reserved words, so that's not a risk.

2. Storage engine audit (separate from the version upgrade, same layer)

Root cause

Both codebases' `config/database.php` sets:

```
'strict' => false,
'engine' => null,
```

`'engine' => null` means Laravel never appends `ENGINE=...` to `CREATE TABLE`, so every table falls back to the server's `default_storage_engine`. That variable is set to **MyISAM** at the server level (`my.ini`) — non-standard; stock MySQL has defaulted to InnoDB since 5.5.

Confirmed scope — worse than the initial spot-check suggested

Database	InnoDB tables	MyISAM tables	% MyISAM
`mvpold_db` (VisoAdmin)	21	**25**	**54%**
`mvpold_db_one` (KaaryamAPI)	26	**18**	**41%**

MyISAM tables in mvpold_db (VisoAdmin):
`categories, client_rm_requests, event_packages, event_services_quote_discount, event_shortlisted_vendor, failed_jobs, issuing_authorities, jobs, license_audit_log, migrations, notifications, occasion, package_services, preferences_tmp_data, rm_request_details, role_permissions, roles, service_verifications, social_media_link, trade_license_submissions, user_online, user_profile_completion, vendor_packages, viso_user_activity_log, whatsapp_templates`

MyISAM tables in mvpold_db_one (KaaryamAPI):
`categories, client_rm_requests, event_packages, event_services_quote_discount, event_shortlisted_vendor, issuing_authorities, license_audit_log, notifications, occasion, package_services, preferences_tmp_data, rm_request_details, role_permissions, roles, service_verifications, social_media_link, trade_license_submissions, vendor_packages`

> Note: `trade_license_submissions` / `service_verifications` / `license_audit_log` in `mvpold_db_one` currently hold 1 leftover row each — test data from the smoke test run while building the TLV vendor API, left behind specifically *because* MyISAM doesn't support transactions (the test's `DB::rollBack()` was a no-op). Still there, untouched, pending confirmation before deleting.

Why this matters, independent of the version upgrade

- **Foreign keys are not enforced** — every `$table->foreign(...)` in these migrations is accepted syntactically but silently dropped by MySQL on a MyISAM table.
- **Transactions are a no-op** — confirmed directly via the leftover test rows above.
- **Table-level locking only**, not crash-safe.

Recommended action items

1. Set `'engine' => 'InnoDB'` in `config/database.php` in both repos.
2. `ALTER TABLE ... ENGINE=InnoDB` the tables listed above.
3. Set `default-storage-engine=InnoDB` explicitly in the new 8.4.8 `my.ini` — don't just inherit whatever the old server had.
4. Decide whether to delete the 3 leftover smoke-test rows in `mvpold_db_one` — left in place until confirmed.

3. Upgrade checklist (local dev, 8.2.0 → 8.4.8+)

- ☐ Convert `root@localhost` to `caching_sha2_password` **before** cutover (§1.1)

☐ Carry forward the current blank `sql_mode`, or explicitly set it in the new `my.ini` (§1.2) — or budget time to fix the strict-mode-dependent queries first

☐ Decide the storage-engine fix now or separately (§2) — not required for the version bump to succeed, but this is a natural time to do it

☐ After upgrading, re-run `SELECT VERSION();`, `SHOW VARIABLES LIKE 'sql_mode'`, `SHOW VARIABLES LIKE 'default_storage_engine'` to confirm the new server matches what's expected before pointing either app at it