

ADR-002 : SQLite + Drizzle plutôt que Postgres

ADR-002 : SQLite + Drizzle plutôt que Postgres

“ Date : 2026 (initial setup) — Statut : accepté

Contexte

Stockage relationnel pour : users, games, questions, feedbacks. Volume attendu : <10K users, <100K questions sur la durée de vie du projet (usage perso + qq invités potentiels).

Options : Postgres (standard, riche), MySQL/MariaDB, SQLite (local, simple).

Décision

SQLite (via `better-sqlite3`) + **Drizzle ORM**.

Raisons :

- **Self-host minimal** : un fichier `.db` dans un volume Unraid, zéro service supplémentaire à monitorer
- **ACID suffisant** : SQLite est ACID et plus que rapide pour ce volume. WAL mode active par défaut.
- **`better-sqlite3` synchrone** : pas de Promise pour chaque query, code plus simple, perfs excellentes (< 1ms par query)

- **Drizzle ORM** : types end-to-end, migrations propres (`drizzle-kit generate` + `migrate`), syntaxe expressive sans magie
- **Zéro pool / latence réseau** : la DB tourne dans le même process que le backend

Conséquences

Bonnes

- Stack minimaliste : un container app + Qdrant, c'est tout
- Backup trivial : copier le `.db` (en mode WAL : aussi `.db-shm` et `.db-wal`)
- Migrations rapides à appliquer (single-instance, pas de coordination)
- Tests unitaires faciles : `:memory:` ou `tmpdir`

Mauvaises

- **Pas de scaling horizontal** : un seul process écrit. Pour scaler, il faudrait migrer vers Postgres.
- **Pas de réplication** : si le fichier corrompt, restore depuis backup obligatoire.
- **Concurrent writes** : SQLite serialize les writes. Pas un problème sur ce volume, mais à connaître.
- **Pas de full-text search avancé** : SQLite a FTS5 mais on ne l'utilise pas. Le retrieval vectoriel est sur Qdrant.

Alternative envisagée

- **Postgres** : plus robuste mais demande un container supplémentaire, configuration users/perms, backup `pg_dump` à scripter, tuning à apprendre. Overkill pour <100K rows.
- **PlanetScale / Neon serverless** : non self-hostable simplement. Et envoyer les données chez un cloud tiers va à l'encontre du principe self-host du projet.

Migration future possible

Si on doit scaler :

1. Drizzle supporte Postgres natif → migrer le schéma `src/schema.ts` (changer le driver)
2. Migrations SQL traduites manuellement (SQLite et Postgres ont des syntaxes proches mais quelques diffs : `INTEGER PRIMARY KEY AUTOINCREMENT` vs `SERIAL`)

3. Sessions in-memory → table `sessions` dans la même DB
4. Container Postgres + volume + healthcheck dans `docker-compose.yml`

L'effort est borné : Drizzle abstrait l'essentiel. C'est faisable en 1-2 jours quand le besoin se présente.

Revision #1

Created 2026-05-10 15:19:51 UTC by thymon

Updated 2026-05-10 15:19:51 UTC by thymon