

Pourquoi utiliser PostgreSQL pour la SAE2.04 / SAE2.01 ?

1. Types de données avancés

PostgreSQL supporte des types complexes absents de SQLite, comme :

- UUID : identifiants uniques, utiles pour des points de mesure uniques.
- JSON / JSONB : parfait pour stocker directement des données complexes de l'API Hub'eau.
- ARRAY : pour stocker des tableaux de valeurs.
- TIMESTAMP WITH TIME ZONE : utile pour les dates précises.

2. Contraintes d'intégrité robustes

PostgreSQL applique vraiment les contraintes :

- NOT NULL, UNIQUE, CHECK, FOREIGN KEY

Contrairement à SQLite, qui peut les ignorer. Cela assure des relations fiables entre les tables (ex : point de mesure relié à une commune).

3. Fonctions SQL avancées

Exemples utiles :

- ILIKE : pour des recherches insensibles à la casse
- STRING_AGG, ARRAY_AGG : pour agréger du texte ou des listes

Exemple : `SELECT COUNT(*) FROM stations WHERE nom ILIKE '%eau%';`

4. Transactions fiables

PostgreSQL gère très bien les transactions :

`BEGIN; ... COMMIT; ... ROLLBACK;`

Parfait pour sécuriser l'importation de données API sans corrompre la base.

5. Gestion des rôles utilisateurs

PostgreSQL permet d'avoir plusieurs utilisateurs avec des droits différents. SQLite : accès

total et unique.

6. Support complet dans DBeaver

PostgreSQL fonctionne parfaitement dans DBeaver : visualisation des tables, relations, requêtes, auto-complétion, export...

Tableau comparatif clair

Fonctionnalité	PostgreSQL	SQLite
Types JSON / ARRAY	Oui	Non
Contraintes FK / CHECK	Oui	Limité
Transactions	Solides	Basiques
Fonctions avancées	Oui	Non
Utilisateurs / rôles	Oui	Non
Intégration DBeaver	Complète	Moyenne