



# Posgres, sqlite comparaison

# Sqlite vu par sqlite

- SQLite is an in-process library that implements a [self-contained](#), [serverless](#), [zero-configuration](#), [transactional](#) SQL database engine.
- SQLite is a C-language library that implements a [small](#), [fast](#), [self-contained](#), [high-reliability](#), [full-featured](#), SQL database engine. SQLite is the [most used](#) database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day

# PostgreSQL vu par PostgreSQL

- PostgreSQL is a powerful, open source object-relational database system that uses and extends the SQL language combined with many features that safely store and scale the most complicated data workloads. The origins of PostgreSQL date back to 1986 as part of the [POSTGRES](#) project at the University of California at Berkeley and has more than 30 years of active development on the core platform.
- PostgreSQL has earned a strong reputation for its proven architecture, reliability, data integrity, robust feature set, extensibility, and the dedication of the open source community behind the software to consistently deliver performant and innovative solutions. PostgreSQL runs on [all major operating systems](#), has been [ACID](#)-compliant since 2001, and has powerful add-ons such as the popular [PostGIS](#) geospatial database extender. It is no surprise that PostgreSQL has become the open source relational database of choice for many people and organisations.
- [Getting started](#) with using PostgreSQL has never been easier - pick a project you want to build, and let PostgreSQL safely and robustly store your data.

## Ce qui semble être consensuel

- SQLite: ultra léger à installer, administrer et très peu gourmand en ressource. Open source.
- PostgreSQL: le SGBD open source le plus avancé au monde, un des plus proche de la norme SQL



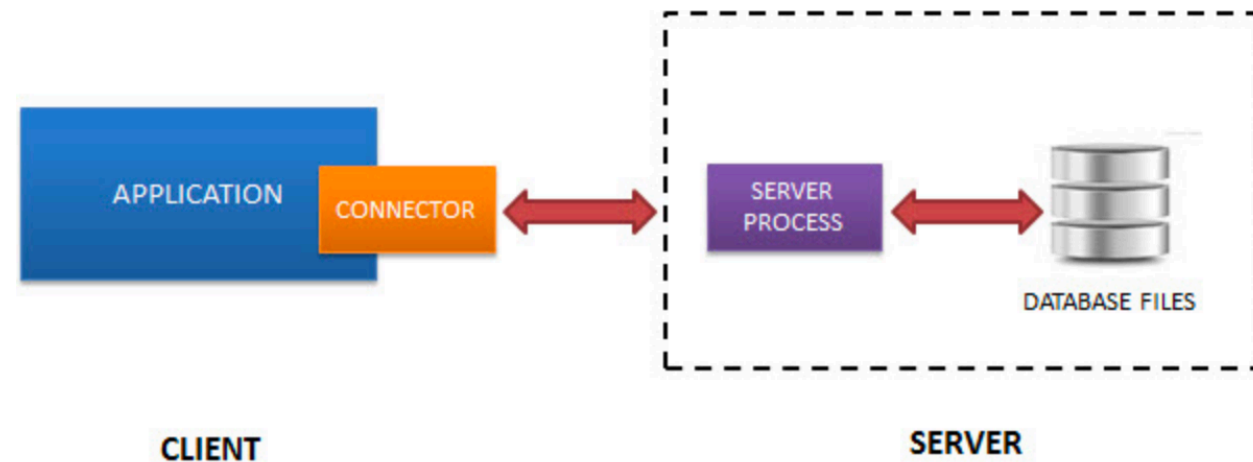
# Les différences principales

# Embarqué/client-serveur

- Sqlite



- Postgresql



# Les types de données

- Sqlite implémente seulement 5 types de données : BLOB, NULL, INTEGER, TEXT, REAL.
- PostgreSQL implémente à peu près tout

## L'encombrement (hors données)

- La bibliothèque SQLite nécessite moins de 500kb
- PostgreSQL est beaucoup plus gourmand 50 à 100 M.



# Portabilité

- SQLite enregistre la base de donnée comme un simple fichier, facile à copier ou déplacer, même sur des plateformes différentes.
- Pour déplacer une BD PostgreSQL il faut l'exporter comme un fichier puis l'uploader depuis un serveur.

# Vitesse

- SQLite est extrêmement rapide
- Pour des simples opérations de lecture, PostgreSQL apparaît comme beaucoup moins performant. Pour des opérations complexes, PostgreSQL s'en tire extrêmement bien.

## Sécurité et authentification

- SQLite ne fournit aucun système d'authentification, le fichier de base de donnée peut être lu/ maj par tout le monde
- Beaucoup de sécurité sont incluse dans PostgreSQL

# SQLite est recommandé pour



des applications qui doivent être portables et ont peu de perspective d'extension future (souvent le cas des applications sur mobile et pour les objets connectés)



pour des applications qui ont besoin de lire/écrire directement sur le disque



les sites web ayant un trafic raisonnable (la plupart en fait)



parfois aussi en phase de test pour ne pas ajouter la complexité de la gestion d'un serveur

# SQLite n'est pas recommandé



Pour de grosses bases de données : sqlite recommande de ne pas dépasser 1T



S'il y a beaucoup d'accès en écriture concurrents sur la base de données, car il ne peut y avoir qu'une seule écriture à la fois pour toute la base de données



Si les données sont séparées de l'application par un réseau , il est préférable de choisir une architecture client serveur qui sera beaucoup plus efficace pour l'accès au données

# PostgreSQL est recommandé

Quand l'intégrité des données est critique

Quand on a besoin d'ajouter ses propres procédures

Quand on a besoin de complexité: PostgreSQL est capable de plans de requête utilisant plusieurs CPU. IL comporte un excellent optimisateur de requêtes. Il gère la concurrence. Il permet l'utilisation de nombreux types de données.

# PostgreSQL n'est pas recommandé



SI LA RAPIDITÉ DES ACCÈS EN LECTURE  
EST CRITIQUE



PAS NÉCESSAIRE D'UTILISER UN  
MARTEAU PIQUEUR POUR ÉCRASER UNE  
MOUCHE

Mais le monde  
du relationnel ne  
s'arrête pas là

