

# Etat de l'art

## PostgreSQL, un système open source

### Présentation de PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle (SGBDR<sup>1</sup>). Le projet est initié en 1986 par Michael Stonebraker et Andrew Yu à l'Université de Californie à Berkley.

L'une des forces majeure de ce système est d'être OpenSource, ce qui signifie qu'il est développé et maintenu par la communauté en plus des développements apportés par la société mère PostgreSQL.

PostgreSQL tient sa réputation de sa fiabilité, sa robustesse et sa richesse fonctionnelle que je détaillerais juste après.

### Les principes de base

Comme dit précédemment, PostgreSQL est un SGBDR<sup>1</sup>. Il utilise le langage SQL<sup>2</sup> pour chercher ou manipuler les données stockées. Le système met à disposition une série de fonctions pour permettre ces interactions, à savoir:

- Les transactions: un ensemble d'une ou de plusieurs opérations regroupées en une seule opération atomique.
- Les vues: table virtuelle qui sélectionne et affiche des données à partir d'une ou plusieurs tables réelles.
- Les contraintes d'intégrité: règles qui garantissent la validité et la cohérence des données dans une base de données.
- Les procédures stockées: programme écrit en SQL qui est stocké dans une base de données et peut être exécuté à la demande.
- Les triggers: procédure stockée qui est automatiquement exécutée en réponse à un événement spécifique sur une table.
- Les fonctions utilisateurs: procédure stockée qui renvoie une valeur et peut être utilisée dans une requête SQL comme une fonction intégrée.

PostgreSQL a également l'avantage d'être multiplateforme. Il peut ainsi fonctionner sur des environnements variés avec des systèmes d'exploitations différents, comme par exemple Windows, Linux, Mac,... L'une des forces de ce système de gestion de base de données réside dans sa capacité à gérer des volumes importants de données allant jusqu'à plusieurs Teraoctets. Cette gestion passe par différents points clefs, à savoir:

- L'indexation
- Le partitionnement
- La gestion du cache
- Des notions de concurrence et d'isolation
- De la réplication et du sharding<sup>3</sup>.

### Les avantages de PostgreSQL

PostgreSQL est un SGBDR<sup>1</sup> très populaire pour plusieurs raisons:

- Il est open source, ce qui signifie qu'il est gratuit et que son code source est disponible pour tous. Cela permet à la communauté de développeurs de contribuer à son amélioration et de créer des extensions pour ajouter des fonctionnalités supplémentaires.

---

<sup>1</sup>Système de gestion de base de données relationnelle

<sup>2</sup>Structured Query Language

<sup>3</sup>Alié à la réplication il permet de répartir la charge sur plusieurs instances d'un même serveur.

- Il est très fiable et robuste, ce qui en fait un choix idéal pour les applications critiques et les environnements de production.
- Comme vu précédemment il est très performance, grâce à son moteur de stockage et son optimiseur de requêtes. Il est capable de gérer de gros volumes de données et de supporter des charges de travail élevées.
- Le système au complet est très flexible, grâce à son architecture modulaire et à son support des extensions. Il peut s'adapter à de nombreux types d'applications et de besoins, notamment pour des applications géographiques avec des besoins plus complets.
- De par sa nature open source, il est compatible avec de nombreux langages de programmation, tels que Python, Java, C++, Ruby, PHP, etc.

Il est également important de noter que PostgreSQL tient sa popularité, au delà de ses performances et fonctionnalités déjà complètes, de par sa capacité à gérer des types de données bien plus complexes. Il propose la gestion de modèles de données complexes tel que des données géographiques et des données attributaires, mais permet surtout de gérer les relations entre ces données.

Cette gestion de données complexe permet une ouverture sur d'autre système, notamment QGIS, un système d'informations géographiques, et ainsi d'étendre les fonctionnalités proposées par ce système.

En type de fichiers volumineux, on peut par exemple citer les fichiers MAJICS, RPG, référentiels vecteurs, ...

### **Les inconvénients de PostgreSQL**

PostgreSQL présente également quelques inconvénients qu'il faut prendre en compte:

- Il peut être plus complexe à installer et à configurer que d'autres SGBDR<sup>1</sup>, tels que MySQL ou SQLite.
- Il peut nécessiter plus de ressources matérielles (mémoire, CPU, espace disque) que d'autres SGBDR<sup>1</sup> pour fonctionner de manière optimale.
- Il peut être moins performant que d'autres SGBDR<sup>1</sup> pour certaines tâches spécifiques, telles que les requêtes de type OLAP (Online Analytical Processing).

“PostgreSQL 12 - Guide de l'administrateur” de Guillaume Lelarge et Stéphane Schildknecht, éditions Eyrolles, 2020.

“PostgreSQL - Maîtrisez les fondamentaux du SGBD open source” de Régis Montoya, éditions ENI, 2019.

“PostgreSQL - Le guide complet de l'administrateur et du développeur” de Joshua D. Drake et Peter Eisentraut, éditions Pearson, 2018.

### **Conclusion**

PostgreSQL est un SGBDR<sup>1</sup> open source très populaire, grâce à sa fiabilité, sa robustesse, sa richesse fonctionnelle et sa flexibilité. Il est utilisé dans de nombreux domaines, tels que la finance, la santé, l'éducation, le gouvernement, etc. Il est également compatible avec de nombreux langages de programmation et de nombreux systèmes d'exploitation. Cependant, il peut être plus complexe à installer et à configurer que d'autres SGBDR<sup>1</sup> et nécessiter plus de ressources matérielles. Malgré ces inconvénients, PostgreSQL reste un choix idéal pour de nombreuses applications critiques et environnements complexes.

## **Des outils pour simplifier les migrations**

### **Le but d'une telle migration**

**Les points clefs**

**Les avantages**

**Les inconvénients**

**En conclusion**

### **Une étape trop coûteuse ?**

Effectuer une migration de base de données n'est pas une tâche anodine. Cela demande du travail en amont, il faut analyser les différents scénarios possibles, estimer un budget pour une telle tâche, réaliser de potentiels développements, s'assurer de la fiabilité avant de mettre tout ça en pratique et enfin la réalisation de l'étape cruciale sur les environnements sensibles.

**Les technologies utilisées**

Pour faire une migration de base de données il existe de nombreuses solutions, certaines plus coûteuses, d'autres plus fiables, encore d'autres des plus spécialisées, ... Il faut donc dans un premier temps trouver différents outils pour comparer les avantages et leurs faiblesses.

**Les ressources disponibles**

**La philosophie de l'entreprise**

**Conclusion**