



# **Présentation de PostgreSQL**

## Table des matières

Présentation de PostgreSQL.....	3
1 Licence.....	4
2 Auteur.....	4
3 Au menu.....	5
4 Présentation générale.....	5
4.1 ACID.....	5
4.2 MVCC.....	6
4.3 SQL.....	6
5 Historique.....	6
5.1 L'histoire du projet.....	7
5.2 Une communauté naît.....	7
5.3 Date de version.....	7
5.4 Numéro de version.....	8
6 Communauté.....	8
6.1 Core Team.....	8
6.2 Committers.....	9
6.3 Contributeurs.....	9
6.4 Utilisateurs.....	9
6.5 Sponsors.....	10
7 Points forts.....	10
7.1 Robustesse.....	10
7.2 Intégration au système.....	10
7.3 Réplication.....	11
7.4 Extensibilité.....	11
7.5 Extensions.....	11
7.6 DDL transactionnel.....	11
7.7 Types de données.....	12
7.8 Langages de procédures stockées.....	12
7.9 Fonctionnalités avancées.....	13
7.10 Ecosystème.....	13
7.11 Documentation.....	13
7.12 Choix dans le prestataire.....	14
8 Limitations.....	14
8.1 Parallélisation.....	14
8.2 Partitionnement.....	15
8.3 Vue matérialisée.....	15
8.4 Embarqué.....	15
8.5 Supervision.....	16
8.6 Manque de DBA expérimentés.....	16
9 Le futur.....	16
10 Conclusion.....	17
11 Des questions ?.....	17



# Présentation de PostgreSQL

- Photo disponible sur <http://www.flickr.com/photos/crazygeorge/5578522008/>
- Prise par Andy Withers
- Licence CC BY-NC-ND 2.0

# 1 Licence



- Creative Common BY-NC-SA
- Vous êtes libre de
  - partager
  - adapter
- Selon les conditions suivantes
  - attribution
  - pas d'utilisation commerciale
  - partage dans les mêmes conditions

# 2 Auteur



- Guillaume Lelarge
- Travail
  - Directeur technique de Dalibo
  - email: [guillaume.lelarge@dalibo.com](mailto:guillaume.lelarge@dalibo.com)
- Communauté
  - contributeur majeur du projet
  - trésorier de PostgreSQL Europe
  - pgAdmin, la traduction de la documentation, l'organisation de [pgconf.eu](http://pgconf.eu)
  - email: [guillaume@lelarge.info](mailto:guillaume@lelarge.info)
  - twitter: [@g\\_lelarge](https://twitter.com/g_lelarge)

## 3 Au menu



- Présentation générale
- Historique
- Communauté
- Points forts
- Limitations
- Futur

## 4 Présentation générale



- Moteur de bases de données
  - ACID
  - MVCC
  - SQL
- Développé par une communauté
- Soutenu par de nombreuses entreprises

### 4.1 ACID



- Les quatre propriétés des transactions
  - Atomic
  - Consistent
  - Isolated
  - Durable
- Respect d'ACID
  - y compris pour les modifications du schéma

## 4.2 MVCC



- MultiVersion Concurrency Control
  - gestion des accès concurrents aux données
- Utilisé par différents SGBD
  - mais implémentation différente
- Minimise les verrous
- ... et automatise leur gestion

## 4.3 SQL



- Respect très fort du standard SQL
  - deux représentants de PostgreSQL dans le comité
- Implémente des extensions à la norme
  - mais propose toujours la norme pour les fonctionnalités qui y sont décrites
  - et précise les extensions dans sa documentation

## 5 Historique



- du projet
- des communautés
- des versions

## 5.1 L'histoire du projet



- 1970, Ingres est développé à Berkeley
- 1985, Postgres succède à Ingres
- 1995, ajout du langage SQL
- 1996
  - Postgres devient PostgreSQL
  - création du PostgreSQL Global Development Group
  - choix d'une licence BSD/MIT

## 5.2 Une communauté naît



- 2000, communauté japonaise
- 2004, communauté francophone
- 2007, communauté italienne
- 2008, PostgreSQL Europe et PostgreSQL US

## 5.3 Date de version



- 1996, v1.0 → première version publiée
- 2003, v7.4 → première version réellement stable
- 2005, v8.0 → arrivée sur Windows
- 2008, v8.3 → performance et fonctionnalités
- 2010, v9.0 → réplication intégrée

## 5.4 Numéro de version



- 2 types de versions
  - mineure ou corrective (tous les 3/4 mois)
  - majeure (tous les ans)
- 5 versions stables
  - 8.4, 9.0, 9.1, 9.2
  - ne pas utiliser les versions précédentes !
- 1 version en cours de développement
  - la future 9.3

## 6 Communauté



- Core-team
- Committers
- Contributeurs
- Utilisateurs
- Sponsors

### 6.1 Core Team



- 6 personnes
  - de différentes sociétés
  - avec des intérêts différents (développement, promotion, norme SQL)
- En charge
  - de décider de la date de sortie des versions
  - d'être le contact principal pour les problèmes de sécurité
- Et rien d'autre...



## 6.2 Committers



- 16 personnes
- Certains sont restreints sur certaines parties du code

## 6.3 Contributeurs



- Développeurs
- Testeurs
- Traducteurs
- Évangélistes
- 80 identifiés par la communauté
  - de l'ordre de 300 personnes en tout

## 6.4 Utilisateurs



- Impossible à dénombrer
- Service public et sociétés privées
- À l'international
  - Afiliat, Apple, Cisco, Fujitsu, Greenpeace, Heroku,
  - Instagram, Mozilla, NTT, Skype, VMWare,
  - Wisconsin Court System
- En France
  - CNAF, Eyrolles, LeBonCoin.fr, Leroy Merlin
  - Meteo France, Orange, Overblog
  - Différents ministères et conseils généraux

## 6.5 Sponsors



- Sponsors Platinum
  - 2ndQuadrant
  - Command Prompt
  - Dalibo
  - EnterpriseDB
  - Red Hat
- Mais aussi
  - Heroku, HP, NEC, NTT, Overblog, Redpill Linpro, Skype, etc.

## 7 Points forts



- Robustesse
- Fonctionnalités
- Extensibilité
- Éco-système
- Documentation

### 7.1 Robustesse



- SGBD très robuste
- Survit aux crashes machine grâce aux journaux de transactions
- Configuration par défaut ciblant la résistance aux crashes

### 7.2 Intégration au système



- Forte adhésion au système d'exploitation
- Facilite son utilisation sur du virtualisé
- Améliorations apportées au système d'exploitation
  - gros intérêt pour PostgreSQL

## 7.3 Réplication



- Maître/esclaves
- Asynchrone... ou synchrone
- Cascade
- S'appuie sur les journaux de transactions
  - pierre angulaire de la robustesse de PostgreSQL

## 7.4 Extensibilité



- Objets utilisateurs
  - types utilisateurs, fonctions, opérateurs, agrégats
- Bibliothèques attachées
  - pour fournir de nouvelles fonctions
- Processus attaché
  - pour du traitement au plus proche du moteur

## 7.5 Extensions



- Scripts SQL
  - adminpack, pg\_buffercache
- Bibliothèques partagées
  - pg\_stat\_statements, auth\_delay, auto\_explain, intarray
- Programmes
  - pgbench, pg\_archivecleanup, pg\_standby

## 7.6 DDL transactionnel



- Possible d'intégrer le DDL dans une transaction
- Respect d'ACID même dans ce cas
- Permet de changer le schéma de manière atomique
  - facilite les montées de version applicatives

## 7.7 Types de données



- Gestion intégrée des tableaux de données
- Type range
  - intervalle de données (entier, date)
  - tests
- contrainte d'exclusion
- Type XML
  - validation, et fonctions de conversion et de recherche
- Type JSON
  - validation, et fonctions de conversion
- Évidemment indexable

## 7.8 Langages de procédures stockées



- Nombreux langages disponibles
  - Perl, Python, PHP, Ruby, etc
- Dernier en date: PL/V8
  - le moteur javascript de Google en langage de procédures stockées
  - s'intègre particulièrement bien avec la gestion des données JSON

## 7.9 Fonctionnalités avancées



- Extension hstore
  - système clé/valeur sur une colonne
- Full Text Search
  - Recherche plein texte
- Extension PostGIS
  - Couche spatiale
  - Certainement la meilleure démonstration de l'extensibilité de PostgreSQL
- Accès aux données distantes
  - fichiers
  - bases de données relationnelles et NoSQL
  - webservice

## 7.10 Ecosystème



- Outils d'analyse
  - pgbadger, explain.depesz.com, debugger et profileur PL/pgsql
- Outils de réplication
  - Slony, Londiste, Bucardo
- Outils de migration
  - ora2pg

## 7.11 Documentation



- Documentation exhaustive
  - ou presque
- Complètement intégré dans le développement
  - une nouvelle fonctionnalité sans documentation n'est pas acceptée

## 7.12 Choix dans le prestataire



- Nombreuses sociétés d'expertise PostgreSQL
- Pas de lien à un seul vendeur majoritaire
- Donc possible de changer de prestataire si l'actuel ne convient pas
  - sans avoir à changer l'infrastructure

## 8 Limitations



- Techniques
  - Parallélisation
  - Partitionnement
  - Vue matérialisée
  - Embarqué
  - Supervision
- Humaines
  - Manque de DBA

### 8.1 Parallélisation



- PostgreSQL est multi-processus
- Chaque processus a une vision spécifique de la base
- Chaque processus exécute une requête complètement
  - le travail ne peut pas être réparti
  - ni à un autre processus, ni à un thread
- Travaux en cours
  - pg\_restore et pg\_dump
  - le reste ?

## 8.2 Partitionnement



- Partitionnement simpliste
- Manque
  - une syntaxe propre
  - un optimiseur de requêtes plus adroit

## 8.3 Vue matérialisée



- Il est possible d'avoir un contournement
  - une vraie table alimentée par des triggers
  - mais mise en place manuelle
  - long à mettre en place et parfois douloureux
- Un début d'implémentation arrive en 9.3

## 8.4 Embarqué



- PostgreSQL n'est pas fait pour l'embarqué
  - multiprocessus
  - gourmand en ressources
- Cela ne changera probablement jamais
- Gros travail de réduction de la consommation électrique en 9.2
  - moins de réveils du CPU inutiles
  - pour du virtualisé
  - mais aussi pour l'embarqué

## 8.5 Supervision



- Outils généralistes (comme Nagios, Zabbix, etc)
  - très fonctionnel mais de configuration complexe
- Outils spécialisés
  - PostgreSQL Enterprise Manager d'EnterpriseDB
  - et c'est tout...

## 8.6 Manque de DBA expérimentés



- Très difficile de trouver un DBA expérimenté
  - en France comme ailleurs
- D'où l'apparition de sociétés de support et de formation

## 9 Le futur



- 9.3 en septembre 2013 ?
  - vues matérialisées
  - vues en écriture
  - triggers sur les événements
  - tables distantes en écriture
  - somme de contrôle dans les fichiers de données
  - sauvegarde parallélisée
  - FDW pour PostgreSQL
- 9.4
  - parallélisation ?



## 10 Conclusion



- PostgreSQL est un système
  - robuste
  - fiable
  - pérenne
- Sa communauté est son plus fort atout
  - elle peut vous aider

## 11 Des questions ?



- Si on manque de temps...
  - email: [guillaume.lelarge@dalibo.com](mailto:guillaume.lelarge@dalibo.com)
  - email: [guillaume@lelarge.info](mailto:guillaume@lelarge.info)
  - twitter: [@g\\_lelarge](https://twitter.com/g_lelarge)
- Slides disponibles sur [http://www.dalibo.org/presentation\\_de\\_postgresql](http://www.dalibo.org/presentation_de_postgresql)