

# PostgreSql



# Sommaire

- Origine du projet
- Équipe principale, contributeurs et sponsors
- Principes fondateurs
- Fonctionnalités
- Versions
- La communauté
- Outils tiers

# Les origines

- 1970 : développement de Ingres
- 1985 : développement de Postgres (post-ingres)
- 1995 : ajout du langage SQL Postgres95
- 1996 : libération du code

# Qu'est-ce que PostgreSQL ?

- Serveur de bases de données libre (MIT/BSD)
- Respectueux des normes SQL
- Respectueux des données
- Avec une excellente portabilité
- Et une grande communauté
  - réactive et internationale

# Historique rapide

- 1996 : v1.0
- 1997 : v6.0 puis 6.1, 6.2, ...
- 1998 : v7.0 puis 7.1, 7.2, ...
- 2005 : v8.0
- 2005 : v8.1
- 2006 : v8.2
- 2008 : v8.3
- 2009 : v8.4
- 2010 : v9.0

# PostgreSQL Core Team

- Tom Lane (développeur principal)
- Bruce Momjian (leader du groupe)
- Marc G. Fournier (administrateur)
- Dave Page
- Josh Berkus (promotion)
- Peter Eisentraut

# Contributeurs

- Différents types de contribution
  - Codage du moteur, codage d'outils externes, documentation, administration des serveurs, aide aux utilisateurs, traducteurs, préparation de conférences, promotion du logiciel, etc.
- Nombreux contributeurs
  - Environ 20 contributeurs de long terme, réguliers
  - Environ 200 contributeurs à court terme
- Nombreuses sociétés impliquées

# Sponsors

- **Sun Microsystems**
- **NTT** (streaming replication)
- **Fujitsu**
- **RedHat** (Tom Lane)
- **Skype** (projet Skytools)
- **EnterpriseDB** (Bruce Momjian, Dave Page, Heikki Linnakangas, ~~Simon Riggs~~, ~~Greg Stark~~, Robert Haas)
- **2<sup>nd</sup> Quadrant**
- **Dalibo**

# Références

- Météo France (base 3.5 Tera Octets)
- CNAF (1 milliard requêtes/jour)
- Airbus
- Et plein d'autres
  - Yahoo, MySpace, OpenStreetMap, Sony Online, BASF, reddit.com, Skype, Sun xVM, Evergreen, MusicBrainz, International Space Station, Caixa Bank, NTT, Wisconsin Court Systems, etc

# Principes fondateurs

- Sécurité des données
- Respect des normes SQL ANSI
- Fonctionnalités
- Performances
- Simplicité du code

# Caractéristiques

- Libre de tout droit (licence BSD)
- Robustesse prouvée sur plusieurs années
- Conçu pour une administration minimale
- Simplicité des outils
- Portabilité du serveur
- Extensibilité
- Plusieurs alternatives pour la haute-disponibilité et la réplication
- Support excellent, tant de la communauté que de la part d'entreprises spécialisées

# Fonctionnalités : coeur

- Standard SQL
- Respect complet d'ACID
  - Atomicité / Cohérence / Isolation / Durabilité
- Utilisation de MVCC
  - MultiVersion Concurrency Control
  - Gestion des transactions et de la sauvegarde à chaud
  - Gestion de versions de ligne dans chaque table
  - Pas de "redo log" à la Oracle

# Fonctionnalités : développement

- Au niveau SGBD
  - Nombreux langages pour les procédures stockées : C, SQL, PL/pgsql, PL/perl, PL/python, PL/php, PL/ruby, etc.
  - Extensibilité des objets : types, fonctions, opérateurs
- En externe
  - Interfaces natives : ODBC, JDBC, C, PHP, Perl, .NET, etc.
  - API ouverte

# Fonctionnalités : sécurité

- Sécurisé par défaut
- Fichier pg\_hba.conf
- Filtrage IP
- Authentification
  - Interne : mots de passe chiffrés MD5
  - Externe : identd, LDAP, Kerberos, GSSAPI/SSPI, RADIUS
- Chiffrement de la connexion
  - Support natif de SSL et des certificats

# Fonctionnalités : SQL - 1

- Excellent support du SQL ANSI
  - SQL/92, SQL/99, SQL:2003 et SQL:2008
- Objets SQL:
  - Tables, vues, règles, séquences, triggers
- Opérations SQL
  - jointures, sous-requêtes, requêtes CTE, requêtes Window, etc.
- Contraintes
  - clés primaires, clés étrangères
  - CHECK, NOT NULL, UNIQUE, EXCLUDE

# Fonctionnalités : SQL - 2

- Triggers
  - Quand : AFTER, BEFORE
  - Sur : INSERT, UPDATE, DELETE, COPY, TRUNCATE
  - FOR EACH STATEMENT, FOR EACH ROW
  - Conditionnel (par colonne, ou clause WHEN)
  - Tout langage utilisée par les fonctions
- Règles
- Curseurs
- Héritage

# Fonctionnalités : avancées - 1

- Index
  - Btree, Hash, GiST, GIN
  - Complet, partiel, fonctionnel

# Fonctionnalités : avancées - 2

- Tablespace (apparu avec PostgreSQL 8.0)
  - Avant les tablespaces, il fallait passer par des liens symboliques, SGBD arrêté
  - Maintenant, permet de déplacer les objets physiques alors que le SGBD est actif
  - Peut contenir tous les objets physiques : bases, tables, index
  - Amélioration des performances en répartissant les entrées/sorties disque
  - Meilleure flexibilité lorsqu'un disque arrive à saturation

# Fonctionnalités : extensibilités

- Création de types de données et
  - de leurs fonctions
  - de leurs opérateurs
  - de leurs règles
  - de leurs agrégats
- Sans avoir à coder une ligne de C

# Fonctionnalités : XLOG

- Journaux de transactions
- Technologie WAL : Write Ahead Log
  - Les modifications sont d'abord enregistrées dans les journaux de transaction
  - Puis dans les fichiers de données
- Limite les écritures sur disque
- Assure la cohérence des données

# Fonctionnalités : PITR

- Point In Time Recovery
- Avant PITR
  - Sauvegarde pg\_dump
  - Généralement un cron journalier
  - D'où une perte possible de 24h d'activité
- Depuis PITR (apparu avec PostgreSQL 8.0)
  - Sauvegarde de base (les fichiers)
  - Puis sauvegarde de chaque journal de transactions
  - Permet une restauration complète ou jusqu'à une certaine heure

# Fonctionnalités : Warm Standby

- Esclave (non utilisable)
- Restauration en continu
- Réplication sur un serveur complet
- Mise en place simple et efficace
- Deux inconvénients majeurs
  - Esclave non disponible en lecture seule
  - Mise à jour de l'esclave journal par journal

# Fonctionnalités : HS + SR

- HS = Hot Standby
  - Esclave en lecture seule
- SR = Streaming Replication
  - Réplication en flux
  - Mise en place simple et rapide
  - Quelques inconvénients
    - Pas de switchover
    - Failover facile, mais sans récupération des autres esclaves

# Sauvegarde / restauration

- Outils de base
  - pg\_dump, pg\_dumpall
  - pg\_restore
- Sauvegarde à chaud et cohérente
- Sauvegarde des fichiers
  - Possible si PostgreSQL arrêté
- Sauvegarde PITR
  - Sauvegarde des fichiers
  - Et archivage des journaux de transactions le temps de la sauvegarde

# Monitoring

- Traces très complètes
  - Et facilement configurables
- Tables systèmes statistiques
  - Activité du système
  - Informations sur les objets
    - Tables, index, séquences, procédures stockées
  - Informations sur les verrous
  - Informations sur les journaux de transactions
- Utilisation d'outils externes
  - munin, zabbix, nagios

# Quelques versions

- Version majeure
  - Sur deux nombres (ex. 8.1, 8.4)
  - Contient des nouvelles fonctionnalités
- Version mineure
  - Sur trois chiffres (8.2.10, 9.0.1)
  - Ne contient que:
    - Des corrections de bug
    - Des corrections de faille de sécurité

# Version 7.4

- Fin 2003
- Plus de corruptions de données
- Amélioration importante des performances (par exemple pour l'opérateur IN et la clause GROUP BY)
- VACUUM plus efficace
- Apparition de deux modules contrib : tsearch2 et autovacuum
- **N'est plus maintenue depuis le 4 octobre !**

# Version 8.0

- Début 2005
- Disponible en natif pour Windows
  - 2000, XP, 2003, Vista, 2008, 7
- Fonctionnalités entreprise
  - Tablespaces, Savepoints, PITR
- Amélioration des performances
  - CHECKPOINT et VACUUM
- **N'est plus maintenue depuis le 4 octobre !**

# Version 8.1

- Fin 2005
- Nouvelles fonctionnalités
  - Rôles
  - Paramètres OUT et INOUT pour les fonctions
  - Two-Phase Commit
  - Intégration de l'autovacuum
- Meilleures performances sur les SMP
- Meilleur partitionnement de tables
- **N'a plus qu'une mise à jour !**

# Version 8.3

- Début 2008 (EOL février 2013)
- Pour les performances
  - HOT, commit asynchrone, etc.
- Pour les utilisateurs
  - Recherche plein texte, XML, nouveaux types (enum, UUID)
- Pour les administrateurs
  - Journalisation CSV, nouvelle authentification (GSSAPI/SSPI)

# Version 9.0 - 1

- Fonctionnalités majeures :
  - Hot Standby
  - Streaming Replication
  - Mise à jour de versions (pg\_upgrade)

# Mise à jour de PostgreSQL

- Mise à jour mineure
  - Mise à jour des binaires, et redémarrage de PostgreSQL
- Mise à jour majeure
  - Changement des tables systèmes
  - 3 solutions
    - pg\_dump, mise à jour, pg\_restore
    - Slony
    - pg\_upgrade

# La communauté : les serveurs

- Site officiel
  - <http://www.postgresql.org>
  - <http://wiki.postgresql.org>
- Association francophone
  - <http://www.postgresql.fr>
  - <http://docs.postgresql.fr>
  - <http://forums.postgresql.fr>
- Autres
  - <http://wiki.postgresql.org/>
  - <http://pgfoundry.org>

# La communauté : les listes

- Annonces
  - pgsql-announce
- Anglophones
  - pgsql-general, pgsql-admin, pgsql-sql, pgsql-perfs, etc.
- Francophone
  - pgsql-fr-generale
- Spécifiques développement et contribution
  - pgsql-hackers, pgsql-patches, pgsql-doc, pgsql-www

# La communauté : les forums

- Forums web
  - <http://forums.postgresql.fr>
  - <http://postgresql.developpez.com>
- Forums IRC (sur Freenode)
  - Anglophone : #postgresql
  - Francophone : #postgresqlfr

# Projets : pgAdmin

The screenshot displays the pgAdmin III interface. The 'Object browser' on the left shows a tree view of databases, with 'mnogo\_int' selected under the 'postgres' database. The 'Properties' pane on the right shows the following details for the 'mnogo\_int' database:

Property	Value
Name	mnogo_int
OID	29197156
Owner	postgres
ACL	
Tablespace	pg_default
Encoding	SQL_ASCII
Default schema	public
Allow connections?	Yes
Connected?	Yes
System database?	No
Comment	Mnogo search for internal websites

The 'SQL pane' at the bottom shows the following SQL commands:

```
-- Database: mnogo_int
-- DROP DATABASE mnogo_int;

CREATE DATABASE mnogo_int
  WITH OWNER = postgres
       ENCODING = 'SQL_ASCII'
       TABLESPACE = pg_default;
COMMENT ON DATABASE mnogo_int IS 'Mnogo search for internal websites'
```

At the bottom of the window, the status bar indicates 'Retrieving Schema details... Done.' and '0.78 secs'.

# Projets : phpPgAdmin

phpPgAdmin

PostgreSQL 8.1.8 lancé sur 127.0.0.1:5432 -- Vous êtes connecté avec le profil « guillaume » -- 1 Apr 2007, 17:10 [SQL](#) | [Rechercher](#) | [Déconnexion](#)

phpPgAdmin : Serveur local :

Bases de données?

Rôles?

Tablespaces?

Exporter

Rapports

Base de données	Propriétaire	Codage	Tablespace	Taille	Actions			Commentaire
amarok	amarok	UTF8	pg_default	16 Mo	<a href="#">Supprimer</a>	<a href="#">Droits</a>	<a href="#">Modifier</a>	
archiveopteryx	aoxsuper	UTF8	pg_default	4985 Ko	<a href="#">Supprimer</a>	<a href="#">Droits</a>	<a href="#">Modifier</a>	
dbmail	dbmail	UTF8	pg_default	4113 Ko	<a href="#">Supprimer</a>	<a href="#">Droits</a>	<a href="#">Modifier</a>	
depot_development	guillaume	UTF8	pg_default	3593 Ko	<a href="#">Supprimer</a>	<a href="#">Droits</a>	<a href="#">Modifier</a>	
depot_production	guillaume	UTF8	pg_default	3480 Ko	<a href="#">Supprimer</a>	<a href="#">Droits</a>	<a href="#">Modifier</a>	
depot_test	guillaume	UTF8	pg_default	3480 Ko	<a href="#">Supprimer</a>	<a href="#">Droits</a>	<a href="#">Modifier</a>	
guillaume	guillaume	LATIN1	pg_default	3657 Ko	<a href="#">Supprimer</a>	<a href="#">Droits</a>	<a href="#">Modifier</a>	
neo	guillaume	UTF8	pg_default	7569 Ko	<a href="#">Supprimer</a>	<a href="#">Droits</a>	<a href="#">Modifier</a>	
pagila	guillaume	UTF8	pg_default	11 Mo	<a href="#">Supprimer</a>	<a href="#">Droits</a>	<a href="#">Modifier</a>	
phppgadmin	postgres	UTF8	pg_default	3552 Ko	<a href="#">Supprimer</a>	<a href="#">Droits</a>	<a href="#">Modifier</a>	
postgres	postgres	UTF8	pg_default	3537 Ko	<a href="#">Supprimer</a>	<a href="#">Droits</a>	<a href="#">Modifier</a>	

[Créer une base de données](#)

- Serveurs
  - Serveur local
    - amarok
      - Schémas
        - public
          - Tables
          - Vues
          - Séquences
          - Fonctions
          - Domaines
  - archiveopteryx
  - dbmail
  - depot\_development
  - depot\_production
  - depot\_test
  - guillaume
  - neo
  - pagila
  - phppgadmin
  - postgres

# Projets : Slony

- Réplication 1 maître/plusieurs esclaves
- Utilise un système de noeuds supportant la cascade
- Versions 1.2.21 et 2.0.5
- <http://slony.info/>

# Projets : pgPool, pgBouncer

- Pooler de connexions
- pgPool fait aussi de l'équilibrage de charges, de la HA, de la réplication
- pgPool bien connu, très apprécié, mais un peu fourre-tout
- PgBouncer tout jeune, commence à être apprécié

# Projets : PostGIS

- Module spatial de PostgreSQL
- Version 1.5.1
- <http://www.postgis.org/>

# Conclusion

- Projet de grande ampleur
- Licence BSD
- Robuste, souple, extensible, performant
- Communauté réactive et internationale
- Nombreux projets complémentaires