



PostgreSQL, un éléphant rencontre des éléPHPants

Table des matières

PostgreSQL.....	4
1 Licence.....	5
2 Auteur.....	6
3 Ma société: Dalibo.....	7
4 Au menu.....	8
5 Développement de PostgreSQL.....	9
5.1 PostgreSQL.....	9
5.2 Un peu d'histoire.....	10
5.3 Sa communauté.....	10
5.4 Sponsors.....	11
5.5 Développement.....	11
5.6 Listes.....	12
5.7 Principes fondateurs.....	12
5.8 Versions.....	13
5.9 Pour se lancer.....	13
6 Fonctionnalités de PostgreSQL.....	14
6.1 Mythe: "PostgreSQL est lent et compliqué".....	14
6.2 Fonctionnalités internes.....	15
6.3 Fonctionnalités utilisateurs.....	15
6.4 Fonctionnalités administrateurs.....	16
6.5 Fonctionnalités développeurs.....	16
6.6 Types de données standards.....	17
6.7 Types de données non standards.....	17
6.8 Les extensions.....	18
6.9 Les outils.....	18
6.10 Focus: la réplication.....	19
6.11 Focus: la réplication (2).....	19
6.12 Focus: la recherche plein texte.....	20
6.13 Focus: le type range.....	20
6.14 Focus: le type JSON.....	21
6.15 Focus: le type hstore.....	21
6.16 Projets satellites: PostGIS.....	22
7 Pourquoi utiliser PostgreSQL ?.....	23
8 Références.....	24
8.1 L'État.....	24
8.2 Instagram.....	25
8.3 leboncoin.fr.....	25
8.4 Le moteur d'Orange.....	25
8.5 Météo France.....	26

PostgreSQL



1 Licence



- Licence Creative Common BY-NC-SA
- 3 contraintes de partage :
 - Citer la source (dalibo)
 - Pas d'utilisation commerciale
 - Partager sous licence BY-NC-SA

Cette formation (diapositives, manuels et travaux pratiques) est sous licence **CC-BY-NC-SA**.

Vous êtes libre de redistribuer et/ou modifier cette création selon les conditions suivantes :

- Paternité
- Pas d'utilisation commerciale
- Partage des conditions initiales à l'identique

Vous devez citer le nom de l'auteur original de la manière indiquée par l'auteur de l'œuvre ou le titulaire des droits qui vous confère cette autorisation (mais pas d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent votre utilisation de l'œuvre).

Vous n'avez pas le droit d'utiliser cette création à des fins commerciales.

Si vous modifiez, transformez ou adaptez cette création, vous n'avez le droit de distribuer la création qui en résulte que sous un contrat identique à celui-ci.

Ceci est un résumé explicatif du [Code Juridique](#). La version intégrale du contrat est disponible ici : <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/legalcode>

2 Auteur

- Thomas Reiss
- Travail
 - Expert PostgreSQL chez Dalibo
 - DBA depuis 11 ans
 - email: thomas.reiss@dalibo.com



- Communauté
 - Administrateur de la plate-forme postgresql.fr
 - Vice-président de l'association PostgreSQLFr
 - co-translation “Bases de données PostgreSQL, gestion des performances”
 - twitter: @frostsct1
 - blog: <http://blog.frosties.org/>

Traduction avec Guillaume Lelarge du livre de Greg Smith “PostgreSQL High Performance”. Sorti aux éditions Pearson ([Bases de données PostgreSQL, gestion des performances](#), ISBN 978-2-7440-2483-2).

3 Ma société: Dalibo

- Spécialiste français de PostgreSQL
- Sponsor platinum du projet PostgreSQL
 - <http://www.postgresql.org/about/sponsors/>
- Sites web
 - <http://www.dalibo.com>
 - <http://www.dalibo.org>
- Soutient la communauté :
 - pgBadger
 - Ora2Pg
 - phpPgAdmin
 - pgAdminIII
 - check_postgres
 - traduction de la documentation officielle



4 Au menu



- Développement de PostgreSQL
- Fonctionnalités
- Retours d'expérience

5 Développement de PostgreSQL



- PostgreSQL
- Historique rapide
- Versions
- Communauté
- Sponsors

5.1 PostgreSQL



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

PostgreSQL est un serveur de bases de données libre. Il est disponible sous une licence libre proche de celles de BSD et MIT. C'est l'une des moins restrictives dans le monde informatique. En effet, cette licence garantit:

- l'accès au code source du logiciel ;
- le droit de modifier le code.

De plus, elle n'impose:

- aucune restriction d'utilisation ;
- aucune viralité de sa licence ;
- aucune contrainte de redistribution des modifications éventuelles ;
- aucune contrainte de publicité pour PostgreSQL et/ou sa communauté.

Ses caractéristiques les plus importantes au niveau technique sont son respect de la norme SQL et des données des utilisateurs. Tout est fait, lors du développement de PostgreSQL, pour respecter le plus possible ses deux caractéristiques.

Une autre caractéristique essentielle est sa portabilité. PostgreSQL fonctionne sur un

grand nombre d'architectures processeur (x86, x86_64, IA64, PowerPC, PowerPC 64, S/390, S/390x, Sparc, Sparc64, Alpha, ARM, MIPS, MIPSEL, M68K et PA-RISC) et sur plusieurs systèmes d'exploitation (Linux, Windows 2000 et ultérieurs, toutes les distributions de la famille BSD, Mac OS X, AIX, HP/UX, IRIX, Tru64 Unix, UnixWare, Solaris).

Sa caractéristique humaine la plus flagrante est sa communauté. Elle est importante et répartie dans de nombreux pays, sur tous les continents. Elle est très réactive. Il est très fréquent qu'une question reçoive une réponse dans l'heure qui suit. Les bugs sont pris très au sérieux et là-aussi les correctifs sont rapidement disponibles après la découverte du bug.

5.2 Un peu d'histoire



- 1970 : Ingres est développé à Berkeley
- 1985 : Postgres succède à Ingres
- 1995 : Ajout du langage SQL
- 1996 : Postgres devient PostgreSQL
- 1996 : Création du PostgreSQL Global Development Group

5.3 Sa communauté



- Core Team
 - 6 personnes
 - décident de la date de sortie des versions
 - rien de plus...
- Contributeurs majeurs
 - 38 personnes
- Contributeurs
 - 40 personnes
 - internationaux
 - nombreuses sociétés différentes
- Contributeurs sur une version
 - plus d'une centaine

5.4 Sponsors



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

5.5 Développement



- Code source écrit en C
 - accessible sur un dépôt git
 - <http://git.postgresql.org>
- Committers sont peu nombreux
 - 16 committers
 - excellente connaissance du code
 - méritocratie
 - <http://wiki.postgresql.org/wiki/Committers>
- Contributeurs plus nombreux
 - proposent des patchs
 - discutent des patchs sur les listes

5.6 Listes



- Plusieurs listes de diffusion
- pgsql-hackers
 - discussion sur les fonctionnalités, le code
- pgsql-committers
 - archivage des messages de commit
- pgsql-bugs
 - référencement et discussion des bugs

5.7 Principes fondateurs



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

5.8 Versions



- 2 types de versions
 - mineure ou corrective
 - majeure
- 4 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
- Politique de versions
 - <http://www.postgresql.org/support/versioning/>

Le projet PostgreSQL fait preuve d'une très grande retenue quant aux modifications des anciennes versions. Toute version stable impactée par un bug sera corrigée. Par contre, aucun ajout de fonctionnalité ne se fera sur les anciennes branches stables. Autrement dit, une version mineure ne contiendra que des corrections de bug et de faille de sécurité.

5.9 Pour se lancer



- Site principal
 - <http://www.postgresql.org/>
- Site de la communauté francophone
 - <http://www.postgresql.fr/>
- Forum francophone
 - <http://forum.postgresql.fr/>
- Documentation française officielle
 - <http://docs.postgresql.fr/>

6 Fonctionnalités de PostgreSQL



- Mythe
 - PostgreSQL est lent et compliqué !
- Fonctionnalités
 - internes
 - utilisateurs
 - administrateurs
 - développeurs
- Types de données
- Focus sur certaines fonctionnalités
- Projets externes

6.1 Mythe: "PostgreSQL est lent et compliqué"



- PostgreSQL a la réputation d'être difficile à utiliser !
 - pas de version Windows native avant 2005
 - pas de base de données pour travailler
 - gestion difficile des accès
 - configuration de départ faible
- PostgreSQL a la réputation d'être lent !
 - versions < 8.3 peu efficaces
 - performances améliorées de versions en versions
- PostgreSQL ne sait pas faire de réplication !
 - réplication interne depuis la 9.0
 - possible avant avec des logiciels externes

6.2 Fonctionnalités internes



- ACID
 - PostgreSQL complètement transactionnel
 - respecte ACID entièrement
- MVCC
 - utilisé par la majorité des SGBD
 - permet de lire et écrire sans se bloquer
- Journaux de transactions
 - toute modification y est enregistrée
 - permet de retrouver les données après un crash
 - permet la sauvegarde physique moteur allumé
 - permet la réplication

6.3 Fonctionnalités utilisateurs



- Standard SQL
 - objets: tables, vues, règles, séquences, triggers
 - opérations: jointures, sous-requêtes, requêtes CTE, requêtes Window, etc.
- Standard SQL/MED
 - tables distantes et ses connecteurs
- Va au-delà
 - différents types et algorithmes d'index
 - contrainte d'exclusion
 - nombreux types supplémentaires (inet, géométrique, range)

6.4 Fonctionnalités administrateurs



- Configuration
 - globale
 - par base, utilisateur, ou couple base/utilisateur
- Sécurité
 - en accès aux bases
 - en accès aux objets et aux données
- Réplication
 - asynchrone/synchrone
 - en cascade
- Plusieurs solutions
 - de sauvegarde
 - de réplication
 - de supervision

6.5 Fonctionnalités développeurs



- Grande extensibilité
 - types utilisateurs
 - procédures stockées
 - nombreux langages
 - nombreux types d'index

6.6 Types de données standards



- varchar, char
- integer, smallint, bigint
- numeric, decimal
- boolean
- bit, bit varying
- float, real
- date, time, timestamp, interval
- xml

6.7 Types de données non standards



- text
- bytea
- array
- enum
- serial, smallserial, bigserial
- cidr, inet, macaddr
- uuid
- json
- hstore
- range

6.8 Les extensions



- Pour les développeurs
 - débbugger pl/pgsql, profiler
- Pour l'optimisation
 - pg_stat_statements, auto_explain
- Pour des fonctionnalités supplémentaires
 - PostGIS, sepgsql
- Pour des types de données spécialisés
 - hstore, citext
- Pour accéder à des objets distants
 - oracle_fdw, mysql_fdw, redis_fdw, twitter_fdw

6.9 Les outils



- Administration
 - pgAdminIII
 - phpPgAdmin
- Supervision
 - check_postgres
- Migration
 - Ora2Pg
 - MySQLMigrator 😊

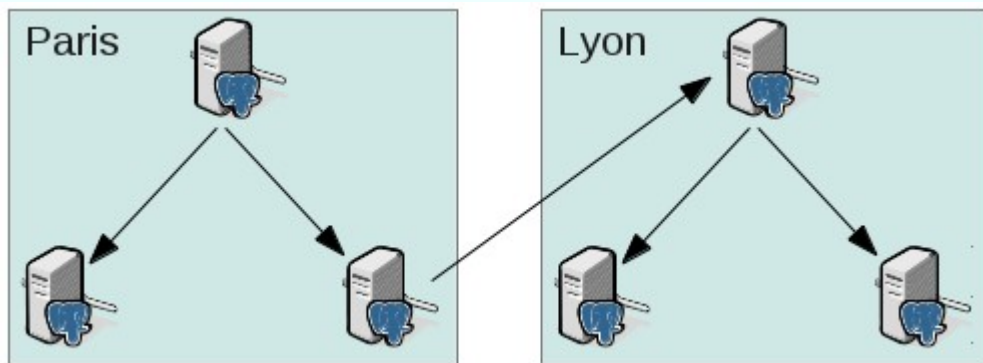
6.10 Focus: la réplication

- Plusieurs moyens de faire de la réplication
- Fonctionnalité interne depuis PG 9.0
 - asynchrone
 - synchrone depuis la 9.1
 - en cascade depuis la 9.2
- Solutions externes :
 - Slony
 - Londiste
 - Bucardo
- Pas encore de réplication multi-maître
 - sauf avec Bucardo



6.11 Focus: la réplication (2)

- Réplication en cascade depuis la 9.2 :



6.12 Focus: la recherche plein texte



- Intégré au moteur en 8.3
- Basé sur un dictionnaire (stemmer)
- Un document est analysé et devient une liste de jetons
- Chaque jeton fait partie d'une classe
 - nombre, mot, mot composé, adresse mail, URL, etc.
- Le dictionnaire identifie les jetons
 - les termes courants et mots de liaisons sont ignorés
 - les autres sont normalisés et transformés en lexèmes
- Des index permettent de faire des recherches sur ces lexèmes
- Pour aller plus loin
 - http://www.dalibo.org/glmf111_recherche_plein_texte_avec_postgresql

Pour aller plus loin :

http://www.dalibo.org/glmf111_recherche_plein_texte_avec_postgresql

6.13 Focus: le type range



- Permet d'exprimer un intervalle mathématique de valeurs
- Supporte plusieurs types de données :
 - date, entier, texte, point, etc.
- Extensible à d'autres types de données
- Indexable
- ```
SELECT '(“2012-01-01”, “2012-02-01”]'::daterange @> '2012-01-15'::date;
```
- Pour aller plus loin :
  - [http://wiki.postgresql.org/wiki/What%27s\\_new\\_in\\_PostgreSQL\\_9.2#Range\\_Types](http://wiki.postgresql.org/wiki/What%27s_new_in_PostgreSQL_9.2#Range_Types)

## 6.14 Focus: le type JSON



- Permet la validation du type
- Fonctions
  - `array_to_json`
  - `row_to_json`
- Permet de récupérer le résultat de requêtes au format JSON
  - directement interprétable par de nombreux langages (comme JavaScript)
- Pour aller plus loin :
  - [http://wiki.postgresql.org/wiki/What%27s\\_new\\_in\\_PostgreSQL\\_9.2#JSON\\_datatype](http://wiki.postgresql.org/wiki/What%27s_new_in_PostgreSQL_9.2#JSON_datatype)

json est un format particulier pouvant stocker des données. Comme le type xml introduit en 8.4, la validité de la valeur est vérifiée à chaque insertion ou modification. Le plus intéressant réside certainement dans le fait qu'il est possible de récupérer le résultat d'une requête ou le contenu d'une table dans un champ textuel codé suivant le format JSON. De nombreux langages (généralement ciblant le web) comprennent directement le format json. Cela facilite ainsi la discussion entre un serveur PostgreSQL et un langage comme JavaScript.

## 6.15 Focus: le type hstore



- Stockage de type clé→valeur
- Fonctionnalité NoSQL dans une base SQL
- Opérateurs particuliers
- Indexable
- Exemple
  - [https://wiki.postgresql.org/wiki/Audit\\_trigger\\_91plus](https://wiki.postgresql.org/wiki/Audit_trigger_91plus)

## 6.16 Projets satellites: PostGIS



- Cartouche spatiale de PostgreSQL
- Version actuelle: 2.0
- Standard SQL/MM
- Manipule des données géométriques et géographiques
- 2D et 3D

## 7 Pourquoi utiliser PostgreSQL ?



- Beaucoup de transactions simultanées
- Besoin d'intégrité des données
- Sécurité
- Nombreux utilisateurs simultanés
- Exécution de requêtes complexes
- ... ou simplement parce qu'il vous plait
- ... et ne pas dépendre d'Oracle

## 8 Références



- Qui utilise PostgreSQL ?
  - L'État
  - Instagram
  - leboncoin.fr
  - La CNAF
  - Le moteur d'Orange
  - Météo France

### 8.1 L'État



- Circulaire Ayrault pour le logiciel libre
  - “Il faut favoriser la migration des bases propriétaires vers des bases libres, en particulier PostgreSQL”
  - [http://circulaire.legifrance.gouv.fr/pdf/2012/09/cir\\_35837.pdf](http://circulaire.legifrance.gouv.fr/pdf/2012/09/cir_35837.pdf)
- PostgreSQL et PostGIS à l'IGN :
  - <http://www.postgresql.fr/temoignages:ign>
- CG de Loire-Atlantique
  - <http://www.postgresql.fr/temoignages:cg44>



## 8.2 Instagram



- Instagram utilise PostgreSQL depuis le début
- plusieurs solutions NoSQL ont été évaluées
- PostgreSQL s'est imposé
- <http://instagram-engineering.tumblr.com/>
- <http://instagram-engineering.tumblr.com/post/13649370142/what-powers-instagram-hundreds-of-instances-dozens-of>

## 8.3 leboncoin.fr



- Site de petites annonces, pour particuliers et professionnels le plus visité en France
- Historiquement sur MySQL
- problèmes de montées en charge et performances
- Passe à PostgreSQL 8.1 en 2008
- [http://www.postgresql.fr/temoignages:le\\_bon\\_coin](http://www.postgresql.fr/temoignages:le_bon_coin)

## 8.4 Le moteur d'Orange



- Moteur de recherche “Le moteur”:
  - <http://www.lemoteur.fr/>
- 5 milliards de tuples sont ainsi répartis sur
- 160 machines qui abritent 800 serveurs PostgreSQL,
- Pour une volumétrie totale de 24 téra-octets
- [http://www.postgresql.fr/temoignages:moteur\\_orange](http://www.postgresql.fr/temoignages:moteur_orange)

## 8.5 Météo France



- Bases de données techniques et scientifiques
- Bases temps-réel
- Fortes volumétries (plusieurs Tera-Octets)
- [http://www.postgresql.fr/temoignages:meteo\\_france](http://www.postgresql.fr/temoignages:meteo_france)