

# POSTGRESQL

**TOUR D'HORIZON DES FONCTIONNALITÉS,  
NOUVEAUTÉS DE LA VERSION 10**

# AUTEUR

- Adrien Nayrat
- Consultant PostgreSQL chez [Dalibo](#)
  - email : [adrien.nayrat@dalibo.com](mailto:adrien.nayrat@dalibo.com)
  - twitter : [@Adrien\\_nayrat](#)
  - blog : <https://blog.anayrat.info/>
- Licence : Creative Common BY-NC-SA
- Extraits de formations Dalibo

# AU MENU

- Histoire
- Idées reçues sur PostgreSQL
- Fonctionnalités
- Nouveautés de la version 10
- Futur

# HISTOIRE

- 1985 : Michael Stonebraker recode Ingres
- post « ingres » => post~~ingres~~ => postgres
- postgres => PostgreSQL
- Licence PostgreSQL
  - Droit de utiliser, copier, modifier, distribuer sans coût de licence

# PRINCIPES FONDATEURS

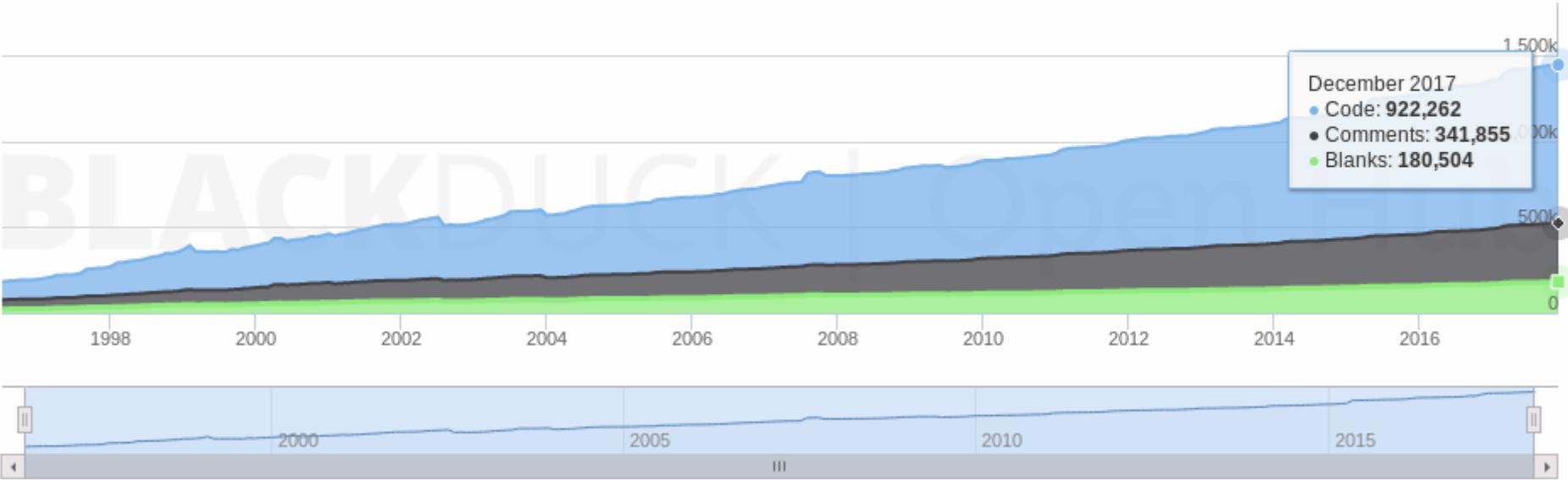
- Sécurité des données (ACID)
- Respect des normes (ISO SQL)
- Fonctionnalités
- Performances
- Extensibilité
- Simplicité du code

- 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
- 2016 : v9.6 -> parallélisation
- 2017 : v10 -> réplication logique

# PROGRESSION DU CODE

Code, Comments and Blank Lines

Zoom 1yr 3yr 5yr 10yr All



# VERSIONS

- Versions obsolètes : 9.2 et antérieures
- Versions actuelles : de 9.3 à 10
- Version en cours de développement : 11

# NUMÉROTATION

- Avant la version 10
  - X.Y : version majeure (8.4, 9.6)
  - X.Y.Z : version mineure (9.6.4)
- Après la version 10
  - X : version majeure (10, 11)
  - X.Y : version mineure (10.1)

# AU MENU

- Histoire
- **Idées reçues sur PostgreSQL**
- Fonctionnalités
- Nouveautés de la version 10
- Futur

# QUELQUES IDÉES REÇUES

- Lent
- Lourd à installer
- `apt install postgresql-10`
- Compliqué
- Opensource == Pas de support

# AU MENU

- Histoire
- Idées reçues sur PostgreSQL
- **Fonctionnalités**
- Nouveautés de la version 10
- Futur

# FONCTIONNALITÉS

- ACID / MVCC
- SQL
- Performances
- Réplication
- Sauvegarde/Restauration
- Sécurité
- Extensibilité

# ACID / MVCC

- **Atomicité** (Atomic)
- **Cohérence** (Consistent)
- **Isolation** (Isolated)
- **Durabilité** (Durable)
- **MultiVersion Concurrency Control**
  - Garantit ACID
  - Gestion de la concurrence

# SQL

- Opérations DML courantes :
  - SELECT
  - INSERT
  - UPDATE
  - DELETE

# SQL

- Opérations DDL :
  - CREATE/ALTER/DROP/...
  - DDL respecte ACID : transactionnel, durable...

# SQL

- CTE (2009) :
  - factoriser des sous-requêtes
  - améliorer la lisibilité d'une requête

```
WITH resultat AS (  
    /* requête complexe */  
)  
SELECT *  
    FROM resultat  
WHERE nb < 5;
```

# SQL

- CTE RECURSIVE:

```
WITH RECURSIVE suite AS (  
  SELECT 1 AS valeur  
  UNION ALL  
  SELECT valeur + 1  
     FROM suite  
     WHERE valeur < 10  
)  
SELECT * FROM suite;
```

# SQL

- Vue matérialisées
- Contraintes différables
- Clause LATERAL
- Analyse de données :
  - Fonctions de fenêtrage
  - Fonctions OLAP GROUPING SETS, CUBE, ROLLUP ...

# SQL

- Types
  - Standards SQL : int, ARRAY, varchar, timestamp, interval...
  - Spécifiques à PostgreSQL : réseau, géométrie, JSONB...
  - Utilisateurs : enum, composite...
- Validation des données
- Opérateurs spécifiques
- Indexable!

# SQL

## Full Text Search

- Recherche plein texte
- Requêtes "google like"
- Indexable
- Un seul langage : SQL

# SQL

## Foreign Data Wrapper

- Norme SQL/MED 2008
- Accès à des données distantes
  - Postgres
  - Oracle
  - Twitter
  - ...
- Il existe une pléthore de FDW

# PERFORMANCES

- Méthodes d'accès
  - 6 types d'index (7 avec les bloom filter)
  - B-Tree
  - GIN
  - Hash
  - GiST
  - SP-GIST
  - BRIN
  - Bloom
    - Extensible depuis la 9.6!
- Index Only Scan

# PERFORMANCES

- Algorithmes de jointure
  - Nested Loop (boucle imbriquée)
  - Hash Join (hachage de la table interne)
  - Merge Join (tri-fusion)
  - Parallélisation possible (Depuis 9.6)

# PERFORMANCES

- Parallélisation (9.6+)
  - Seqscan
  - Aggregats
  - Jointure
  - Index Scan
  - ...

# RÉPLICATION

- Physique
  - Réplique les modifications de blocs
  - Toute l'instance
  - Asynchrone / synchrone
  - Multi-secondaire
  - Cascade
  - Secondaire en lecture seule

# RÉPLICATION

- Logique
  - Décodage logique des journaux de transaction
  - Possibilité de sélectionner les tables
  - Asynchrone / synchrone
  - Multi-secondaire
  - Cascade
  - Secondaire accepte les écritures
  - Depuis la 10 (infrastructure depuis la 9.4)

# SAUVEGARDE/RESTAURATION

- Physique
  - A chaud
  - Intégralité de l'instance
  - Permet le *Point In Time Recovery*

# SAUVEGARDE/RESTAURATION

- Logique
  - A chaud
  - Flexible
  - Parallélisable
  - Dump cohérent malgré les modifications durant la sauvegarde

# SÉCURITÉ

- Droits sur les objets
- Droits de connexion
- Row Level Security
- SSL

# EXTENSIBILITÉ

- Extensions
  - Postgis
- Index
- Full Text Search
- Types

# AU MENU

- Histoire
- Idées reçues sur PostgreSQL
- Fonctionnalités
- **Nouveautés de la version 10**
- Futur

# NOUVEAUTÉS DE LA VERSION 10

- Changements importants
- Partitionnement
- Réplication logique
- Performances
- Sécurité

# CHANGEMENTS IMPORTANTS

- `xlog => wal`
- Numérotation de version

# PARTITIONNEMENT

- Gestion native
  - Performances
  - Administration facilitée
- Partitionnement par :
  - Listes
  - Intervalles

# RÉPLICATION LOGIQUE

- Infrastructure en 9.4
- Choix des tables à répliquer
- PUBLICATION / SUBSCRIPTION

# PERFORMANCES

- Gains importants sur les fonctions de hashage
- Optimisation des tris
- Optimisations agrégats
- Parallélisme étendu à d'autres noeuds
- Statistiques multicolonne
- `postgres_fdw` : Join et aggregate pushdown

# SÉCURITÉ

- Authentification *SCRAM-SHA-256*
- Rôles d'administration

# FUTUR V11?

- Partitionnement par hashage (commité)
- Parallel Hash Join (commité)
- Parallel Append (commité)
- Mise à jour clé partitionnement (commité)
- Infrastructure JIT (WIP)
- JSON (Norme SQL 2016) (WIP)
- tris parallélisés (WIP)
- Partition pruning (WIP)
- Statistiques étendues ... étendues (WIP)
- Covering Btree (WIP)

**QUESTIONS?**

