Workshop 12

# Nouveautés de PostgreSQL 12


12

Dalibo & Contributors

https://dalibo.com/formations

# Nouveautés de PostgreSQL 12

Workshop 12

TITRE : Nouveautés de PostgreSQL 12
SOUS-TITRE : Workshop 12

REVISION: 12

# Table des Matières

# 1 NOUVEAUTÉS DE POSTGRESQL 12


Photographie de Ikiwaner[1] , licence GNU FREE Documentation Licence[2] , obtenue sur wikimedia.org[3] .

**Participez à ce workshop !**

Pour des précisions, compléments, liens, exemples, et autres corrections et suggestions, soumettez vos *Pull Requests* dans notre dépôt :

https://github.com/dalibo/workshops/tree/master/fr

Licence : PostgreSQL[4]

Ce workshop sera maintenu encore plusieurs mois après la sortie de la version 12.

---

[1] https://commons.wikimedia.org/wiki/User:Ikiwaner
[2] https://en.wikipedia.org/wiki/fr:Licence_de_documentation_libre_GNU
[3] https://fr.wikipedia.org/wiki/Fichier:Etosha_elefant.jpg
[4] https://github.com/dalibo/workshops/blob/master/LICENSE.md

## 1.1 LA V12

- Développement depuis le 30 juin 2018
- sortie le 3 octobre 2019
- v12.1 sortie le 14/11/2019

---

## 1.2 LES NOUVEAUTÉS

- Développement SQL
- Partitionnement
- Réplication
- Monitoring
- Administration
- Performances : index
- Incompatibilités
- Fonctionnalités futures
- Ateliers

---

## 1.3 DÉVELOPPEMENT / SQL

- Colonnes générées par une expression `IMMUTABLE`
- Colonne `OID` supprimée
- `COMMIT` ou `ROLLBACK AND CHAIN`
- `COPY FROM WHERE`

## 1.4 PARTITIONNEMENT

- Clés étrangères
- Fonctions d'information :
  - `pg_partition_root` renvoie la partition mère d'une partition,
  - `pg_partition_ancestors` renvoie la partition mère ainsi que la partition concernée
  - `pg_partition_tree` renvoie tout l'arbre de la partition sous forme de tuples
- Commande `\dP`

---

## 1.5 RÉPLICATION

- Nouveauté des `postgresql.conf`et `recovery.conf`
- 2 fichiers *trigger*
- Paramètres modifiables à chaud
- Fonction `pg_promote()`
- Copie de slot de réplication

---

## 1.6 MONITORING

- Échantillon des requêtes dans les logs
- Vues de progression pour `CREATE INDEX`, `CLUSTER`, `VACUUM`
- Listing :
  - des fichiers dans les répertoires `status` des archives des *wals*
  - des fichiers temporaires
- `pg_stat_replication` : timestamp du dernier message reçu du secondaire

---

## 1.7 ADMINISTRATION

- Nouveautés sur le `VACUUM`
  - `VACUUM (TRUNCATE on)` : *lock* et libération de l'espace de fin de table
  - `VACUUM (SKIP_LOCKED ON)`
  - `VACUUM (INDEX_CLEANUP OFF)` : favoriser les `VACUUM FREEZE`
  - Nouvelles options de `vacuumdb`
- Recyclage des *WALs*
  - `wal_recycle`
  - `wal_init_zero`
- Outils : `pg_upgrade`, `pg_ctl`, `pg_checksums`
- Paramètres de `postgresql.conf`

---

### 1.7.1 ENVIRONNEMENT CLIENT

- Formatage *CSV* en sortie de `psql`
- `EXPLAIN (SETTINGS)`

---

### 1.7.2 OUTILS

- `pg_upgrade --clone`: clonage à l'aide de *reflink*
- Rotation des logs avec `pg_ctl logrotate`
- `pg_verify_checksums` devient `pg_checksums`
- `pg_checksums --enable | --disable`

### 1.7.3 PARAMÈTRES DE CONFIGURATION

- Nouveaux paramètres
- disparition du fichier `recovery.conf`
- valeur par défaut modifiée

---

## 1.8 PERFORMANCES

- `REINDEX CONCURRENTLY`
- `CREATE STATISTICS` pour les distributions non-uniformes
- paramètre `plan_cache_mode`
- fonctions d'appui : pour améliorer les estimations de coût des fonctions
- *JIT* par défaut
- Optimisation *CTE* : `MATERIALIZED` / `NOT MATERIALIZED`
- Meilleures performances sur le partitionnement

---

### 1.8.1 REINDEX CONCURRENTLY

- `REINDEX CONCURRENTLY`

---

### 1.8.2 CREATE STATISTICS MCV

- Nouveau type `MCV` pour la commande `CREATE STATISTICS`
- MCV signifie *Most Common Values*
- collecte les valeurs les plus communes pour un ensemble de colonnes

---

### 1.8.3 MÉTHODE DE MISE EN CACHE DES PLANS D'EXÉCUTION

- Transactions **préparées**
- `plan_cache_mode auto`
- Trois modes:
  - `auto`
  - `force_custom_plan`
  - `force_generic_plan`

---

### 1.8.4 FONCTIONS D'APPUI (SUPPORT FUNCTIONS)

- Améliore la visibilité du planificateur sur les fonctions
- possibilité d'associer à une fonction une fonction « de support »
- produit dynamiquement des informations sur:
  - la sélectivité
  - le nombre de ligne produit
  - son coût d'exécution
- La fonction doit être écrite en C

### 1.8.5 JIT PAR DÉFAUT

- JIT (Just-In-time) est maintenant activé par défaut

---

### 1.8.6 MODIFICATION DU COMPORTEMENT PAR DÉFAUT DES REQUÊTES CTE

- Les *CTE* ne sont plus des barrières d'optimisation
- Modification du comportement par défaut des CTE
  - `MATERIALIZED`
  - `NOT MATERIALIZED` (par défaut)
- requêtes non récursives
- requêtes référencées une seule fois
- requêtes n'ayant aucun effet de bord

---

### 1.8.7 PERFORMANCES DU PARTITIONNEMENT

- Performances accrues avec un grand nombre de partitions
- Verrous lors des manipulations de partitions
- support des clés étrangères vers une table partitionnée
- Amélioration du chargement de données

---

## 1.9 INCOMPATIBILITÉS

- Disparition du `recovery.conf`
- `max_wal_senders` n'est plus inclus dans `max_connections`
- Noms des clés étrangères
- Tables `WITH OIDS` n'existent plus
- Types de données supprimés
- Fonctions `to_timestamp` et `to_date`
- Outil `pg_checksums`

### 1.9.1 PG_VERIFY_CHECKSUMS RENOMMÉE EN PG_CHECKUMS

- pg_verify_checksums devient pg_checkums

---

## 1.10 FONCTIONNALITÉS FUTURES

- *Pluggable storage*
  - *HEAP storage*
  - *column storage*
  - *Zed Heap*
  - *blackhole*

---

### 1.10.1 PLUGGABLE STORAGE

#### 1.10.1.1 HEAP storage

- `HEAP storage`
- méthode de stockage par défaut
- seule méthode supportée pour le moment

---

#### 1.10.1.2 Zedstore: Column storage

- méthode orientée colonne
- données compressées
- nom temporaire !

---

#### 1.10.1.3 zHeap

- *UNDO* plutôt que *REDO*
- meilleur contrôle du *bloat*
- réduction de l'amplification des écritures
- réduction de la taille des entêtes
- méthode basée sur les différences

---

#### 1.10.1.4 Méthode d'accès *Blackhole*

- sert de base pour créer une extension *Access Method*
- toute donnée ajoutée est envoyée dans le néant

---

## NOTES

## NOTES

# NOS AUTRES PUBLICATIONS

## FORMATIONS

- **DBA1 : Administration PostgreSQL**
  https://dali.bo/dba1
- **DBA2 : Administration PostgreSQL avancé**
  https://dali.bo/dba2
- **DBA3 : Sauvegardes et réplication avec PostgreSQL**
  https://dali.bo/dba3
- **DEVPG : Développer avec PostgreSQL**
  https://dali.bo/devpg
- **DEVSQLPG : SQL pour PostgreSQL**
  https://dali.bo/devsqlpg
- **PERF1 : PostgreSQL Performances**
  https://dali.bo/perf1
- **PERF2 : Indexation et SQL avancé**
  https://dali.bo/perf2
- **MIGORPG : Migrer d'Oracle vers PostgreSQL**
  https://dali.bo/migorpg

## LIVRES BLANCS

- **Migrer d'Oracle à PostgreSQL**
- **Industrialiser PostgreSQL**
- **Bonnes pratiques de modélisation avec PostgreSQL**
- **Bonnes pratiques de développement avec PostgreSQL**

## TÉLÉCHARGEMENT GRATUIT

Les versions électroniques de nos publications sont disponibles gratuitement sous licence open-source ou sous licence Creative Commons. Contactez-nous à l'adresse contact@dalibo.com pour plus d'information.

## DALIBO, L'EXPERTISE POSTGRESQL

Depuis 2005, DALIBO met à la disposition de ses clients son savoir-faire dans le domaine des bases de données et propose des services de conseil, de formation et de support aux entreprises et aux institutionnels.

En parallèle de son activité commerciale, DALIBO contribue aux développements de la communauté PostgreSQL et participe activement à l'animation de la communauté francophone de PostgreSQL. La société est également à l'origine de nombreux outils libres de supervision, de migration, de sauvegarde et d'optimisation.

Le succès de PostgreSQL démontre que la transparence, l'ouverture et l'auto-gestion sont à la fois une source d'innovation et un gage de pérennité. DALIBO a intégré ces principes dans son ADN en optant pour le statut de SCOP : la société est contrôlée à 100 % par ses salariés, les décisions sont prises collectivement et les bénéfices sont partagés à parts égales.