



Présentation de la formation

Site : <http://alphorm.com>
Blog : <http://alphorm.com/blog>
Forum : <http://alphorm.com/forum>

Nouredine DRISSI
Expert consultant bases de données
Contact : ndform@valneo-xi.fr

PostgreSQL, administration

alphorm.com™©



Le formateur

- **Nouredine DRISSI**
 - Spécialisé dans la base de données
 - 15 années d'expérience sur tous les SGBDR du marché (SQL Server, Oracle, PostgreSQL, Mysql, DB2)
 - Issue de l'environnement bancaire



ndform@valneo-xi.fr

PostgreSQL, administration

alphorm.com™©

Connaissances préalables

- Connaissances de base du système d'exploitation Linux ou Windows
- Expérience professionnelle de SQL
- Expérience professionnelle des bases de données relationnelles
- Expérience en matière de conception de bases de données

A propos de ce cours

- **Objectifs**
 - Comprendre et maîtriser l'architecture de PostgreSQL;
 - Administrer au quotidien PostgreSQL;
 - Sauvegarder et restaurer des bases de données ou des instances PostgreSQL;
 - Gérer la sécurité;
 - Analyser et comprendre PostgreSQL;
 - Comprendre et optimiser l'architecture de PostgreSQL.

Plan du cours

- **Module 1:** Présentation de PostgreSQL
- **Module 2:** Installation de PostgreSQL
- **Module 3:** Gestion des instances sous PostgreSQL
- **Module 4:** Les outils d'administration sous PostgreSQL
- **Module 5:** La gestion des transactions
- **Module 6:** Les logs PostgreSQL
- **Module 7:** Le cache de données PostgreSQL
- **Module 8:** Gestion de la sécurité
- **Module 9:** Sauvegarde et restauration
- **Module 10:** Administrer au quotidien PostgreSQL



PostgreSQL, administration

alphorm.com™©

Qu'est-ce qu'un SGBDR ?

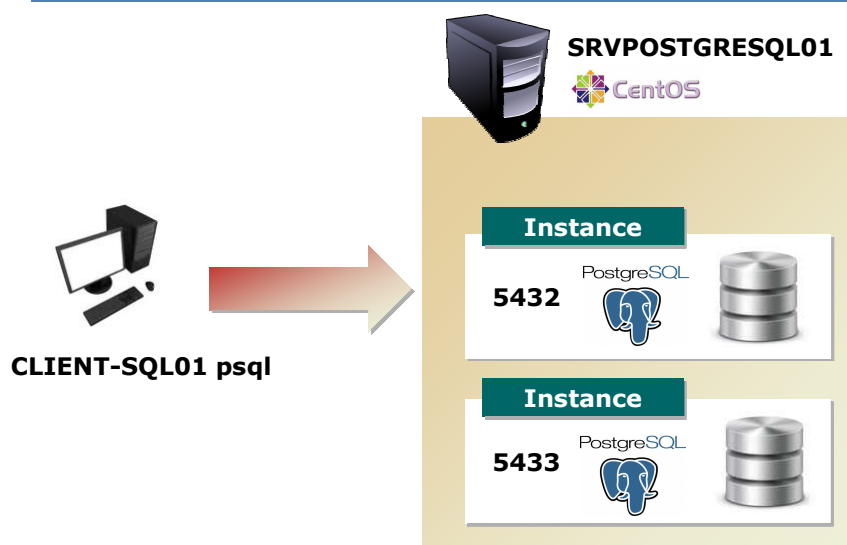
- PostgreSQL est un Système de Gestion de Base de Données Relationnelle (SGBDR)

PostgreSQL est chargé de :
 - stocker les données,
 - vérifier les contraintes d'intégrité définies,
 - garantir la cohérence des données qu'il stocke, même en cas de panne (arrêt brutal) du système,
 - assurer les relations entre les données définies par les utilisateurs.

PostgreSQL, administration

alphorm.com™©

Environnement du cours

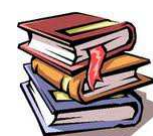


PostgreSQL, administration

alphorm.com™©

Introduction à la base Gescom

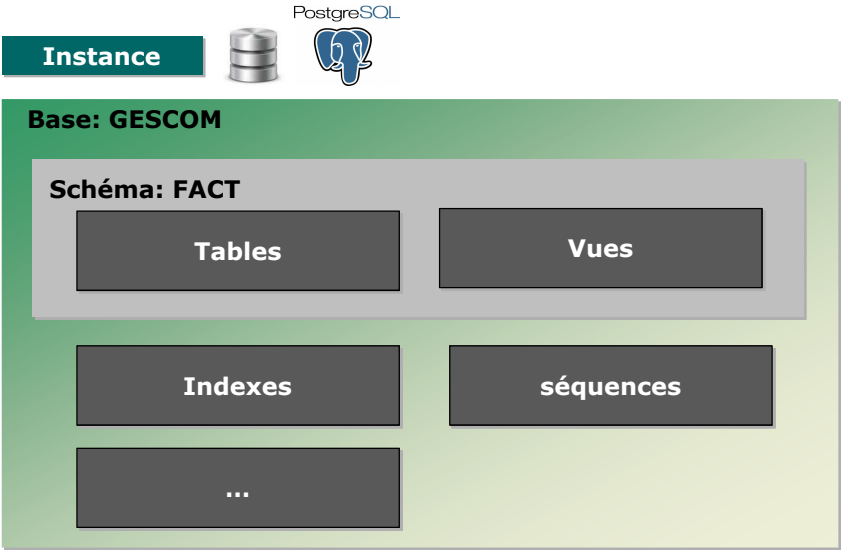
- Dans les ateliers pratique de ce cours, nous allons travailler avec une base de données modèle que l'on nommera GESCOM.
- Au fil des ateliers pratique, nous allons ajouter des fonctionnalités à la base de données GESCOM.
- Cette base de données servira de fil conducteur pour l'ensemble des démonstration (sauvegarde, transfert, fichiers data, réplication, optimisation, etc.)



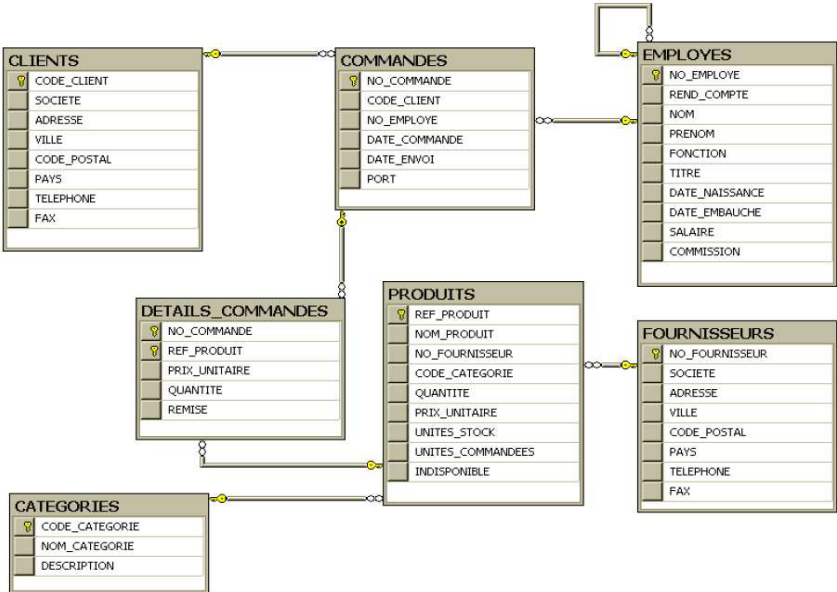
PostgreSQL, administration

alphorm.com™©

Introduction à la base Gescom



Introduction au modèle de données





GO

PostgreSQL, administration

alphorm.com™©



Alphorm.com



Présentation de PostgreSQL

Site : <http://alphorm.com>
Blog : <http://alphorm.com/blog>
Forum : <http://alphorm.com/forum>

Nouredine DRISSI
Expert consultant bases de données
Contact : ndform@valneo-xi.fr

PostgreSQL, administration

alphorm.com™©

Présentation de PostgreSQL

- Présentation
 - Système de gestion de bases de données
 - Fondé sur POSTGRES
 - Supporte une grande partie du standard SQL
 - Licence libérale
 - Programme écrit à l'université de Californie à Berkeley
 - Base de données Open Source de référence
 - Modèle client/serveur

PostgreSQL, administration

alphorm.com™©

Présentation de PostgreSQL

- Les caractéristiques
 - SGBDR
 - Supporte le langage SQL
 - Open source
 - Respect de la norme ACID
 - Fonctionnalités
 - respect de la norme SQL92, requêtes complexes ;
 - clés étrangères ;
 - plusieurs langages procéduraux, triggers, procédures et fonctions ;
 - Déclencheurs, vues ;

PostgreSQL, administration

alphorm.com™©

Présentation de PostgreSQL

- Les versions les plus importantes
 - 1995: V 1.0 le code devient modifiable (évolution de la licence)
 - 2005: V 8.0 natif Windows, savepoints, PITR, tablespaces, prêt pour l'entreprise
 - 2010: V 9.2 réplication, scalabilité

Présentation de PostgreSQL

- La communauté PostgreSQL
 - Le site pgFoundry.org
 - Héberge de nombreux projets développés par des équipes indépendantes
 - connecteurs pour les différents langages ;
 - langages procéduraux ;
 - outils d'aide à l'administration ;
 - logiciels pour la haute disponibilité (réplication, gestion des connexions, etc.).

Présentation de PostgreSQL

- Les atouts
 - Un projet de grande ampleur
 - Un SGBD complet
 - Une communauté réactive et internationale
 - Un large panel de projets complémentaires
 - Une solution stable, ouverte et performante
 - Licence BSD
 - Coût nul
 - Code source disponible
 - Aucune contrainte de redistribution

Présentation de PostgreSQL

- PostgreSQL VS les autres SGBDR

	PostgreSQL	Mysql	Oracle	MSSQL
Tablespace		NON		
Partitionnement	Emulé	INNODB		
Vues matérialisées	Emulé	NON		
Norme ACID		INNODB		
Intégrité référentielles		INNODB		
Langage Procédural	Plusieurs			
Curseurs		Limité		

Présentation de PostgreSQL

- Conclusion
 - SGBDR performant
 - Montée en puissance et en charge progressive
 - Gestion des gros volumes de données
 - Gestion des performances
 - Tuning matériel
 - Tuning logiciel
 - Optimiseur de requêtes

PostgreSQL, administration

alphorm.com™©



Prochain module :
Installation de PostgreSQL



PostgreSQL, administration

alphorm.com™©



Alphorm
com



Installation de PostgreSQL

Windows

Site : <http://alphorm.com>
Blog : <http://alphorm.com/blog>
Forum : <http://alphorm.com/forum>

Nouredine DRISSI
Expert consultant bases de données
Contact : ndform@valneo-xi.fr

PostgreSQL, administration

alphorm.com™©



PostgreSQL, administration

alphorm.com™©

Installation de PostgreSQL

- Les étapes d'installation
 - Téléchargement des sources à partir du site <http://www.postgresql.org>
 - Installation du logiciel PostgreSQL
 - Automatisation du démarrage du service PostgreSQL

PostgreSQL, administration

alphorm.com™©



Prochain module :

Gestion des instances sous
PostgreSQL



PostgreSQL, administration

alphorm.com™©



Installation de PostgreSQL

Site : <http://alphorm.com>
Blog : <http://alphorm.com/blog>
Forum : <http://alphorm.com/forum>

Nouredine DRISSI
Expert consultant bases de données
Contact : ndform@valneo-xi.fr

PostgreSQL, administration

alphorm.com™©



Installation de PostgreSQL

- Les étapes d'installation
 - Téléchargement des sources à partir du site <http://www.postgresql.org>
 - Préparation du système d'exploitation
 - Installation du logiciel PostgreSQL
 - Initialisation des fichiers de l'instance
 - Configuration
 - Installation et configuration des scripts de démarrage sous Unix ou services sous Windows

PostgreSQL, administration

alphorm.com™©

Installation de PostgreSQL

- Les types d'installation
 - Installation à partir des paquetages « rpm » (Linux)
 - Les paquetages nécessaires
 - PostgreSQL-server*
 - PostgreSQL*
 - PostgreSQL-lib*
 - Installation à partir des binaires (Linux)
 - Pré-requis
 - make, GCC, gzip, tar
 - Installation à partir des exécutable (WINDOWS)

PostgreSQL, administration

alphorm.com™©



Prochain module :

Gestion des instances sous
PostgreSQL



PostgreSQL, administration

alphorm.com™©



Gestion des instances sous PostgreSQL

Notion d'instance

Site : <http://alphorm.com>
Blog : <http://alphorm.com/blog>
Forum : <http://alphorm.com/forum>

Nouredine DRISSI
Expert consultant bases de données
Contact : ndform@valneo-xi.fr

PostgreSQL, administration

alphorm.com™©



Gestion d'une instance PostgreSQL

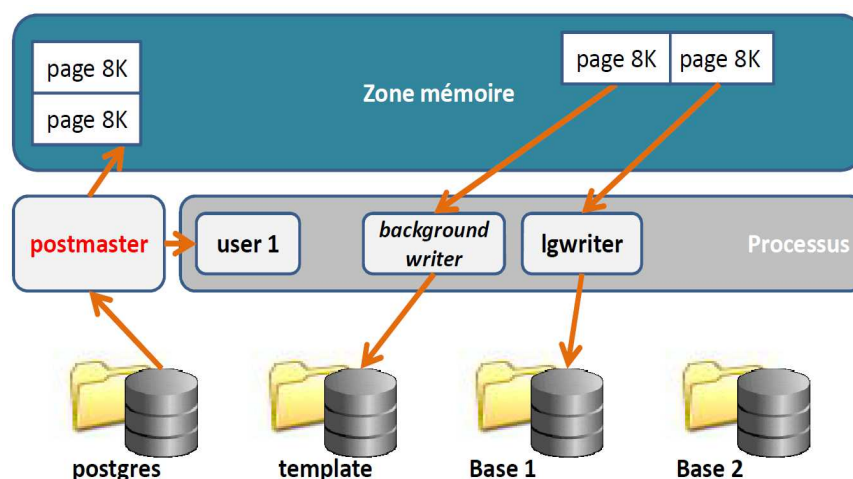
- Une instance PostgreSQL c'est quoi ?
 - Un cache mémoire
 - Des processus
 - Une arborescence physiques (FS, répertoires)
 - Un ensemble de bases de données

PostgreSQL, administration

alphorm.com™©

Gestion d'une instance PostgreSQL

- Représentation schématique d'une instance



PostgreSQL, administration

alphorm.com™©

Gestion d'une instance PostgreSQL

- Les processus PostgreSQL
 - Le processus postmaster
 - Le Processus de gestion des journaux applicatifs (optionnel)
 - Le processus de collecte des statistiques (optionnel)
 - Le processus d'écriture en tâche de fond
 - Le processus d'écriture des journaux de transactions
 - Le Processus de maintenance des tables (optionnel)
 - Le processus d'archivage (optionnel)
 - Les processus de communication client/serveur

PostgreSQL, administration

alphorm.com™©

Gestion d'une instance PostgreSQL

- Le processus **postmaster**
 - Père de tous les autres processus
 - Ecoute toutes les connexions entrantes
 - Ecoute par la socket ou le port TCP/IP (par défaut 5432)
 - Lance les différents services nécessaires à PostgreSQL
 - le processus de gestion des journaux applicatifs (si activé) ;
 - le processus de collecte des statistiques (si activé) ;
 - le « autovacuum launcher » (si activé) ;
 - le processus d'écriture en tâche de fond ;
 - le processus d'écriture des journaux de transactions

PostgreSQL, administration

alphorm.com™©

Gestion d'une instance PostgreSQL

- Le processus **bgwriter**
 - Ecriture des blocs modifiés en mémoire cache dans les fichiers de données
 - Processus d'arrière plan
 - Déclenché par un checkpoint
 - **checkpoint_timeout** indique la durée maximale sans CHECKPOINT
 - **checkpoint_segments** indique le nombre maximum de journaux de transactions utilisés sans CHECKPOINT

PostgreSQL, administration

alphorm.com™©

Gestion d'une instance PostgreSQL

- Le « wal writer process »
 - Apparue en 8.3
 - Enregistre les modifications dans les journaux de transactions WAL
 - Paramètres d'optimisation
 - wal_buffer
 - fsync
 - ...

PostgreSQL, administration

alphorm.com™©

Gestion d'une instance PostgreSQL

- Le processus de collecte des statistiques
 - Activé par défaut en 8.3 (track_activities, track_counts)
 - Récupère les informations des processus postgres
 - Statistiques sur
 - le nombre de lignes lues, insérées, modifiées et supprimées
 - le nombre de blocs disque lus ou écrits
 - Informations récupérées via un port UDP configuré en mode non bloquant

PostgreSQL, administration

alphorm.com™©

Gestion d'une instance PostgreSQL

- Le processus de maintenance des tables
 - Procède au nettoyage des tables
 - Depuis la version 8.3
 - autovacuum launcher
 - autovacuum worker
 - Exécuté au démarrage du serveur
 - Traces disponibles pour ce processus

PostgreSQL, administration

alphorm.com™©

Gestion d'une instance PostgreSQL

- Le processus d'archivage
 - Désactivé par défaut (archive_mode)
 - Gère l'archivage des journaux de transactions
 - Pas de consommation mémoire importante
 - Traces disponibles pour ce processus

PostgreSQL, administration

alphorm.com™©

Gestion d'une instance PostgreSQL

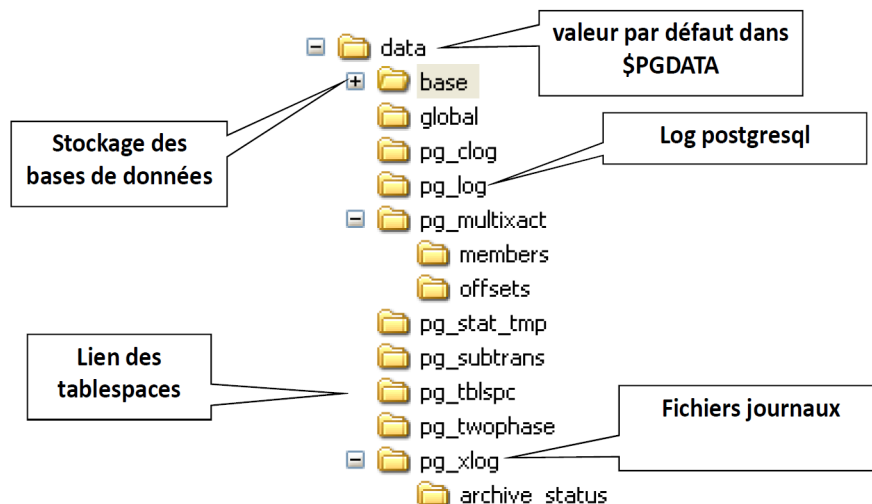
- Le processus des journaux applicatifs
 - Disponible depuis la version 8.0
 - Activé par défaut
 - Apparaît sous le nom de « **logger process** »
 - Exécuté au lancement du serveur PostgreSQL

Gestion d'une instance PostgreSQL

- Les processus de communication client/serveur
 - Ce sont les plus nombreux (limitation avec max_connections)
 - Gère la communication entre le client et le serveur
 - Démarré par le processus « **postmaster** » après authentification de l'utilisateur
 - Traite les requêtes des clients
 - Utilise la mémoire partagée et la mémoire dédiée

Gestion d'une instance PostgreSQL

■ Arborescence physique d'une instance



PostgreSQL, administration

alphorm.com™©

Gestion d'une instance PostgreSQL

■ Fonction de chaque répertoire

- **base** répertoire associé à l'espace de tables pg_default
- **base/pgsql_tmp** fichiers temporaires pour certaines opérations
- **global** tables système (espace de tables pg_global)
- **pg_clog** données relatives au statut de validation des transactions
- **pg_log** journaux d'activité
- **pg_multixact** données relatives au statut des transactions multiples
- **pg_subtrans** données relatives au statut des transactions imbriquées
- **pg_tblspc** contient un lien symbolique pour chaque espace de tables
- **pg_twophase** données pour les transactions préparées
- **pg_xlog** journaux binaires (WAL)

PostgreSQL, administration

alphorm.com™©

Gestion d'une instance PostgreSQL

- La racine de l'instance **\$PGDATA**
 - Contient tous les répertoires de l'instance
 - Contient les fichiers de configuration de l'instance
 - **PG_VERSION** version de PostgreSQL associée à l'instance
 - **pg_hba.conf** configuration de l'authentification
 - **pg_ident.conf** configuration de l'authentification ident
 - **postgresql.conf** configuration de l'instance
 - **postmaster.opts** options de démarrage de postmaster
 - **postmaster.pid** PID du processus postmaster

PostgreSQL, administration

alphorm.com™©

Gestion d'une instance PostgreSQL

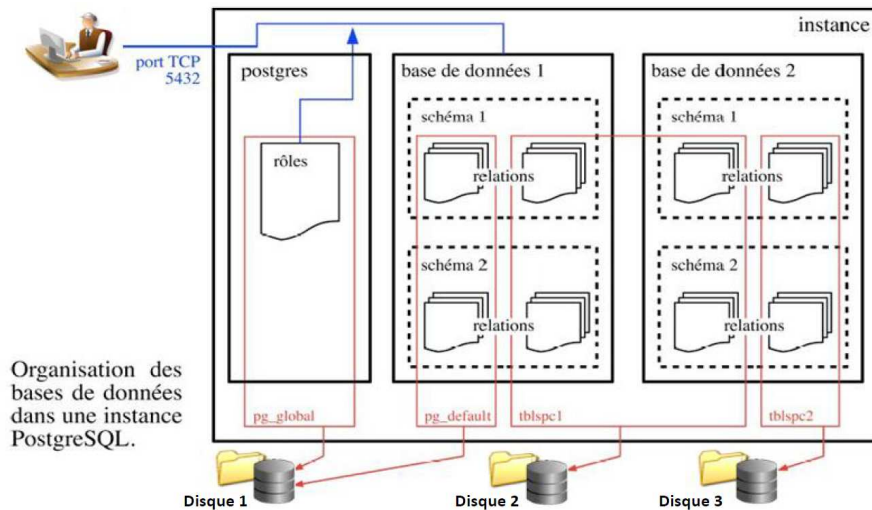
- Configuration de l'instance
 - Se fait dans le fichier « **postgresql.conf** »
 - A la racine de chaque instance \$PGDATA
 - Toujours faire une sauvegarde du fichier avant modification

PostgreSQL, administration

alphorm.com™©

Gestion d'une instance PostgreSQL

- Représentation de l'architecture logique et physique



PostgreSQL, administration

alphorm.com™©

Gestion d'une instance PostgreSQL

- Configuration de l'instance
 - Se fait dans le fichier « **postgresql.conf** »
 - A la racine de chaque instance \$PGDATA
 - Toujours faire une sauvegarde du fichier avant modification

PostgreSQL, administration

alphorm.com™©



Prochain module :

Les outils d'administration sous
PostgreSQL



PostgreSQL, administration

alphorm.com™©



Alphorm
_com



Gestion des instances sous
PostgreSQL

Les tablespaces

Site : <http://alphorm.com>
Blog : <http://alphorm.com/blog>
Forum : <http://alphorm.com/forum>

Nouredine DRISSI
Expert consultant bases de données
Contact : ndform@valneo-xi.fr

PostgreSQL, administration

alphorm.com™©

Gestion d'une instance PostgreSQL

- Les espaces de stockage ou tablespace
 - Répertoire d'un système de fichier, dans lequel PostgreSQL écrit les fichiers des tables et des indexes
 - Utilisation
 - Extension de l'espace de stockage existant
 - Répartition des objets à fort accès sur des disques différents
 - Amélioration des performances pour les I/O

PostgreSQL, administration

alphorm.com™©

Gestion d'une instance PostgreSQL

- Caractéristiques des espaces de tables
 - N'est pas spécifique à une base de données
 - Outil pour l'administrateur afin d'optimiser le stockage sur disque
 - Gestion fine des permissions sur les espaces de tables
 - Maîtrise de la répartition des fichiers DATA, en fonction des bases de données et des rôles utilisés

PostgreSQL, administration

alphorm.com™©

Gestion d'une instance PostgreSQL

- Mise en oeuvre des espaces de table
 - Au niveau de l'OS
 - Création du nouveau répertoire de stockage
 - Modification du propriétaire du répertoire
 - Attribution des privilèges sur le répertoire
 - Au niveau de PostgreSQL
 - Création du tablespace avec la commande DDL
 - Vérification de la création du tablespace

PostgreSQL, administration

alphorm.com™©

Gestion d'une instance PostgreSQL

- Création des espaces de table
 - Syntaxe

```
CREATE TABLESPACE <nom> LOCATION <dir>
```

- Exemple

```
# CREATE TABLESPACE app LOCATION '/u101/pgsql/data'
```

PostgreSQL, administration

alphorm.com™©

Gestion d'une instance PostgreSQL

- Suppression des espaces de table
 - Le tablespace doit être vide (pas de table ni d'index)
 - Suppression du tablespace via la commande DDL
 - Le répertoire de stockage peut être supprimé physiquement avec les commandes de l'OS
 - Syntaxe

```
DROP TABLESPACE [ IF EXISTS] <nom>
```

Gestion d'une instance PostgreSQL

- Modifier des espaces de table
 - Renommer le tablespace
 - Changer le propriétaire
 - Syntaxe

```
ALTER TABLESPACE nomtblspc RENAME nomtblspc2;  
ALTER TABLESPACE nomtblspc OWNER nomROLE;
```



Gestion des instances sous PostgreSQL

Initialisation d'une instance

Site : <http://alphorm.com>
Blog : <http://alphorm.com/blog>
Forum : <http://alphorm.com/forum>

Nouredine DRISSI
Expert consultant bases de données
Contact : ndform@valneo-xi.fr

PostgreSQL, administration

alphorm.com™©



Initialisation d'une instance

- Configuration des ressources du noyau
 - PostgreSQL utilise les ressources IPC de l'OS pour la gestion de la mémoire vive
 - La taille maximale d'un segment de mémoire partagée et positionné par défaut à 32 Mo
 - Le nombre de pages est positionné par défaut à 2097152 pages
 - Nécessité de modifier les paramètres par défaut
 - **shmmax** taille maximale d'un segment de mémoire partagée
 - **shmall** nombre maximale de pages de mémoire partagée (OS)

PostgreSQL, administration

alphorm.com™©

Initialisation d'une instance

▪ initdb

- Syntaxe

```
initdb -E <codage> --locale=<locale> -D <dir>
```

- Exemple

```
$ initdb -E UTF8 -D $PGDATA
```

Initialisation d'une instance

▪ initdb

- Syntaxe

```
initdb -E <codage> --locale=<locale> -D <dir>
```

- Exemple

```
$ initdb -E UTF8 -D $PGDATA
```

Initialisation d'une instance

- Création d'une instance
 - Utilisation du programme **initdb**
 - Les options
 - **-D <dir>** indique le répertoire de stockage des données
 - **-U <utilisateur>** indique le nom du super user
 - **-E <encodage>** indique le jeux de caractères sélectionné
 - **-locale=<locale>** initialise les paramètres locaux

PostgreSQL, administration

alphorm.com™©

Initialisation d'une instance

- Démarrage et arrêt d'une instance
 - Utilisation du programme **pg_ctl**
 - **-D <dir>** indique le répertoire de l'instance
 - **-l <journal de trace>** fichier de log de démarrage de l'instance
 - **start, stop, restart** et **reload** permettent de démarrer, d'arrêter, de redémarrer et de recharger le serveur

PostgreSQL, administration

alphorm.com™©

Initialisation d'une instance

- **pg_ctl**

- Syntaxe

```
pg_ctl -D $PGDATA start | stop | restart -l <journal trace>
```

- Exemple

```
pg_ctl -D /var/lib/pgsql/data stop
```

```
pg_ctl -D /var/lib/pgsql/data start
```

Initialisation d'une instance

- **pg_ctl**

- Syntaxe

```
pg_ctl -D $PGDATA start | stop | restart -l <journal trace>
```

- Exemple

```
pg_ctl -D /var/lib/pgsql/data start
```

Initialisation d'une instance

▪ Démarrage et arrêt d'une instance

- Utilisation du script de démarrage (SysV) Linux

/etc/init.d/postgresql stop | start | restart | reload

- Windows



Initialisation d'une instance

▪ Suppression d'une instance

- Linux

- Arrêtez le processus postmaster PostgreSQL
- Supprimer tous les fichiers et répertoires à la racine de l'instance (\$PGDATA)

- Windows

- Arrêtez le service PostgreSQL
- Supprimer tous les fichiers et répertoire à la racine de l'instance (\$PGDATA)

Initialisation d'une instance

- Suppression d'une instance
 - Linux
 - Arrêtez le processus postmaster PostgreSQL
 - Supprimer tous les fichiers et répertoires à la racine de l'instance (\$PGDATA)
 - Windows
 - Arrêtez le service PostgreSQL
 - Supprimer tous les fichiers et répertoire à la racine de l'instance (\$PGDATA)

PostgreSQL, administration

alphorm.com™©



Alphorm.com



Gestion des instances sous PostgreSQL

Sécurisation d'une instance

Site : <http://alphorm.com>
Blog : <http://alphorm.com/blog>
Forum : <http://alphorm.com/forum>

Nouredine DRISSI
Expert consultant bases de données
Contact : ndform@valneo-xi.fr

PostgreSQL, administration

alphorm.com™©

Sécurisation d'une instance

- Gérée par
 - Le fichier « **pg_hba.conf** »
 - Configure la manière d'authentifier les connexions
 - Par défaut autorise toutes les connexions locales
 - Nécessaire de modifier ce fichier après l'installation
 - Les objets rôles
 - Pour la connexion et la sécurité au niveau des objets

PostgreSQL, administration

alphorm.com™©

Sécurisation d'une instance

- Configuration de « **pg_hba.conf** »
 - Format d'une ligne d'authentification
 - *Type | Base | User | Adresse | Méthode | Option*
 - Colonne « Type »
 - **local** : connexion par une socket du domaine Unix (la colonne adresse est vide) ;
 - **host** : connexion chiffrée ou non par une socket TCP/IP ;
 - **hostssl** : connexion chiffrée par une socket TCP/IP ;
 - **hostnoss**l : connexion non chiffrée par une socket TCP/IP.

PostgreSQL, administration

alphorm.com™©

Sécurisation d'une instance

- Configuration de « **pg_hba.conf** »
 - Format d'une ligne d'authentification
 - *Type | Base | User | Adresse | Méthode | Option*
 - Colonne « Base »
 - all ;
 - **samerole, sameuser** ;
 - une liste de ces éléments séparés par une virgule.

PostgreSQL, administration

alphorm.com™©

Sécurisation d'une instance

- Configuration de « **pg_hba.conf** »
 - Format d'une ligne d'authentification
 - *Type | Base | User | Adresse | Méthode | Option*
 - Colonne « User »
 - all ;
 - le nom d'un groupe (rôle) précédé du caractère « + » ;
 - une liste de ces éléments séparés par une virgule.

PostgreSQL, administration

alphorm.com™©

Sécurisation d'une instance

- Configuration de « **pg_hba.conf** »
 - Format d'une ligne d'authentification
 - *Type | Base | User | Adresse | Méthode | Option*
 - Colonne « Adresse »
 - IP d'origine de la connexion au format CIDR
 - Doit être vide lorsque le type est « local »

PostgreSQL, administration

alphorm.com™©

Sécurisation d'une instance

- Configuration de « **pg_hba.conf** »
 - Format d'une ligne d'authentification
 - *Type | Base | User | Adresse | Méthode | Option*
 - Colonne « Méthode »
 - **trust** : autoriser sans même vérifier le mot de passe ;
 - **reject** : rejeter la connexion ;
 - **md5** : vérifier le mot de passe fourni (ne plus utiliser crypt ou password) ;
 - **ident** : utiliser le protocole IDENT pour vérifier l'utilisateur ;
 - **krb5**, **pam** ou **ldap** : authentifications spécifiques.

PostgreSQL, administration

alphorm.com™©

Sécurisation d'une instance

- Configuration de « **pg_hba.conf** »
 - Format d'une ligne d'authentification
 - *Type | Base | User | Adresse | Méthode | [Option]*
 - Colonne « Méthode »
 - **trust** : autoriser sans même vérifier le mot de passe ;
 - **reject** : rejeter la connexion ;
 - **md5** : vérifier le mot de passe fourni (ne plus utiliser crypt ou password) ;
 - **ident** : utiliser le protocole IDENT pour vérifier l'utilisateur ;
 - **krb5, pam** ou **ldap** : authentifications spécifiques.

Sécurisation d'une instance

- Les sessions
 - Connexion des clients via une socket UNIX
 - Port d'écoute par défaut: 5432
 - Quatre informations nécessaires pour ouvrir une session
 - Le nom/IP de la machine qui héberge l'instance
 - Le nom de la base
 - Le nom d'un rôle
 - Un mot de passe (si configuré dans « **pg_hba.conf** »)



Prochain module :

Les outils d'administration sous
PostgreSQL



PostgreSQL, administration

alphorm.com™©



Alphorm
_com



Les outils d'administration
sous PostgreSQL

Site : <http://alphorm.com>
Blog : <http://alphorm.com/blog>
Forum : <http://alphorm.com/forum>

Nouredine DRISSI
Expert consultant bases de données
Contact : ndform@valneo-xi.fr

PostgreSQL, administration

alphorm.com™©

Les outils d'administration

- Mode ligne de commande
 - createdb, dropdb
 - createuser, dropuser
 - reindexdb, vacuumdb
 - pg_dump, pg_dumpall, pg_restore
 - initdb, clusterdb, postmaster, pg_ctl
 - **psql**, Droplang, pg_config, pg_resetxlog, pg_controldata, createlang, ipcclean

Les outils d'administration

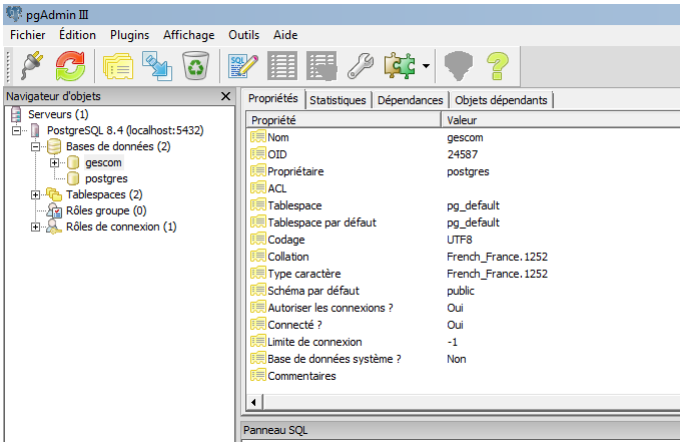
- Mode ligne de commande
 - L'interpréteur de commande « **psql** »
 - Les principaux paramètres
 - -h <machine>
 - -p <port>
 - -d <nom base de données>
 - -U <utilisateur>

 Les outils d'administration

- Mode ligne de commande
 - L'interpréteur de commande « **psql** »
 - Les variables utilisées par « psql » (.bash_profile)
 - \$PGDATA
 - \$PGHOST
 - \$PGDATABASE
 - \$PGPORT
 - \$PGUSER

 Les outils d'administration

- Graphique
 - pgAdmin III



Les outils d'administration

- Graphique
 - phpPgAdmin



Prochain module :

La gestion des transactions sous
PostgreSQL





La gestion des transactions sous PostgreSQL

Site : <http://alphorm.com>
Blog : <http://alphorm.com/blog>
Forum : <http://alphorm.com/forum>

Nouredine DRISSI
Expert consultant bases de données
Contact : ndform@valneo-xi.fr

PostgreSQL, administration

alphorm.com™©



La gestion des transactions

- Transactions
 - Conforme à la norme ACID
 - **Atomicité** toutes les instructions de la transaction sont validées ou aucune ne l'est
 - **Cohérence** chaque transaction validée garantit un état cohérent de la base de données
 - **Isolation** les modifications d'une transaction sont invisibles aux autres transactions
 - **Durabilité** si la transaction est validée, les modifications engendrées sont pérennes

PostgreSQL, administration

alphorm.com™©

La gestion des transactions

- Transactions
 - Par défaut
 - **autocommit**
 - Une transaction
 - Démarrage d'une transaction avec **BEGIN**
 - **COMMIT** permet de valider une transaction
 - **ROLLBACK** permet d'annuler une transaction

PostgreSQL, administration

alphorm.com™©

La gestion des transactions

- Transactions
 - Spécificités
 - Les instructions LDD sont prises en charge par les transactions
 - La fermeture d'une session provoque un rollback implicite

PostgreSQL, administration

alphorm.com™©

La gestion des transactions

- Transactions
 - Isolation
 - Le standard SQL
 - read uncommitted
 - read committed
 - repeatable read
 - serializable

PostgreSQL, administration

alphorm.com™©

La gestion des transactions

- Transactions
 - Isolation
 - Modification du niveau d'isolation
SET SESSION CHARACTERISTICS AS TRANSACTION ISOLATION LEVEL
{ SERIALIZABLE | REPEATABLE READ | READ COMMITTED | READ
UNCOMMITTED }

PostgreSQL, administration

alphorm.com™©

La gestion des transactions

- Les verrous
 - Verrous implicites
 - Verrous explicite
 - Avec l'instruction **lock**
 - En utilisant les clauses **for update** et **for share** de l'instruction select
 - Les verrous ne peuvent être posés que dans le cadre d'une transaction commencée avec *begin*.
 - Les verrous sont libérés lorsque une transaction est terminée
 - La liste des verrous actifs est consultable dans la vue système **pg_locks**



Prochain module :
Les logs PostgreSQL





Gestion de la sécurité sous PostgreSQL

Site : <http://alphorm.com>
Blog : <http://alphorm.com/blog>
Forum : <http://alphorm.com/forum>

Nouredine DRISSI

Expert consultant bases de données

Contact : ndform@valneo-xi.fr

PostgreSQL, administration

alphorm.com™©



Gestion de la sécurité

- Gestion des droits d'accès
 - Utilise la notion de rôle
 - Peut représenter un utilisateur ou un groupe
 - Etre propriétaire d'objets de l'instance (bases de données, tables, etc.)
 - Se voir octroyer des droits d'accès à des objets ou des droit système
 - Etre membre d'un autre rôle
 - Se voir attribuer un mot de passe
 - Non lié au utilisateurs du système d'exploitation

PostgreSQL, administration

alphorm.com™©

Gestion de la sécurité

- Création des rôles

- Syntaxe

```
CREATE ROLE nom_role [ [ WITH ] [ SUPERUSER ] [ CREATEDB ]  
[ CREATEROLE ] [ LOGIN ]  
[ ENCRYPTED PASSWORD 'mot_de_passe' ]  
[ IN ROLE nom_role [, ...] ]  
[ ROLE nom_role [, ...] ] ]
```

PostgreSQL, administration

alphorm.com™©

Gestion de la sécurité

- Suppression des rôles

- Syntaxe

```
DROP ROLE [ IF EXISTS ] nom
```

PostgreSQL, administration

alphorm.com™©

Gestion de la sécurité

- Les rôles groupes
 - Modélise la notion de groupe d'utilisateurs
 - Suffit de créer un rôle sans mot de passe
 - Sans l'attribut LOGIN

PostgreSQL, administration

alphorm.com™©

Gestion de la sécurité

- Les droits
 - Basé sur les rôles et utilise les instructions SQL
 - GRANT et REVOKE
 - Octroyer ou révoquer des droits d'accès sur des objets
 - Ajouter ou retirer des rôles (utilisateurs) à d'autres rôles (groupes)

PostgreSQL, administration

alphorm.com™©

Gestion de la sécurité

- Les droits
 - Les attributs des rôles:
 - LOGIN / NOLOGIN ;
 - SUPERUSER / NOSUPERUSER ;
 - CREATEDB / NOCREATEDB ;
 - CREATEROLE / NOCREATEROLE ;
 - PASSWORD.
 - Associés aux rôles lors de leur création (CREATE ROLE) ou avec l'instruction ALTER ROLE
 - Les rôles qui disposent de l'attribut SUPERUSER contournent le système des privilèges (c'est-à-dire qu'ils ont tous les privilèges)

PostgreSQL, administration

alphorm.com™©

Gestion de la sécurité

- Droits, objets
 - Les privilèges octroyés dépendent du type d'objet
 - tables : SELECT | INSERT | UPDATE | DELETE | REFERENCES | TRIGGER
 - séquences : USAGE | SELECT | UPDATE
 - base de données : CREATE | CONNECT | TEMPORARY
 - fonctions : EXECUTE
 - langage : USAGE
 - schémas : CREATE | USAGE
 - espaces de tables : CREATE

PostgreSQL, administration

alphorm.com™©

Gestion de la sécurité

- Droits, objets
 - ALL représente tous les privilèges et PUBLIC tous les rôles
 - L'option WITH GRANT OPTION autorise le ou les rôles cibles à octroyer ces privilèges

Gestion de la sécurité

- Droits, objets
 - ALL représente tous les privilèges et PUBLIC tous les rôles
 - L'option WITH GRANT OPTION autorise le ou les rôles cibles à octroyer ces privilèges

Gestion de la sécurité

- Droits, objets

- Révocation des droits avec REVOKE

- Syntaxe

```
REVOKE [ GRANT OPTION FOR ]
```

```
{ privilège [...] | ALL } ON [ type_objet ] nom_objet [, ...]
```

```
FROM { nom_role | PUBLIC } [, ...] [ CASCADE | RESTRICT ]
```

Gestion de la sécurité

- Droits, objets

- Bonnes pratiques

- Associer une base de données à un rôle
 - Associer une base de données à une application
 - Un schéma est associé à un rôle utilisateur
 - Les privilèges objets sont donnés à des rôles groupes et non à des rôles LOGIN



Prochain module :

Sauvegarde restauration sous
PostgreSQL



PostgreSQL, administration

alphorm.com™©



Alphorm
_com



Les logs PostgreSQL

Les logs d'activités

Site : <http://alphorm.com>
Blog : <http://alphorm.com/blog>
Forum : <http://alphorm.com/forum>

Nouredine DRISSI
Expert consultant bases de données
Contact : ndform@valneo-xi.fr

PostgreSQL, administration

alphorm.com™©

Gestion des logs

- Les logs d'activités
 - Enregistre les évènements pendant l'exécution du serveur
 - Méthode d'enregistrement
 - fsync, stderr, syslog, eventlog (Windows)
 - Niveau de trace
 - Différentes nature de message
 - Possibilité d'ajuster le niveau de trace

PostgreSQL, administration

alphorm.com™©

Gestion des logs

- Les logs d'activités
 - Ou tracer ?
 - **log_destination (string)** destination des logs (syslog, stderr, etc.)
 - **logging_collector (boolean)** autorise la capture et la redirection des messages envoyés à stderr
 - **log_directory (string)** détermine le répertoire dans lequel les fichiers de trace sont créés
 - **log_filename (string)** indique les noms des journaux applicatifs créés
 - ...

PostgreSQL, administration

alphorm.com™©

Gestion des logs

- Les logs d'activités
 - Quand tracer ?
 - **client_min_messages (enum)** contrôle les niveaux de message envoyés au client
 - **log_min_messages (enum)** contrôle les niveaux de message écrits dans les traces du serveur
 - **log_error_verbosity (enum)** contrôle le niveau de détail des traces de chaque message
 - **log_min_error_statement (enum)** contrôle si l'instruction SQL à l'origine d'une erreur doit être enregistrée dans les traces du serveur
 - ...

Gestion des logs

- Les logs d'activités
 - Que tracer ?
 - **log_checkpoints (boolean)** trace les points de vérification dans les journaux applicatifs
 - **log_connections (boolean)** trace chaque tentative de connexion sur le serveur, ainsi que la réussite de l'authentification du client
 - **log_duration (boolean)** indique le temps de chaque requête noté par log_statement
 - **log_statement (boolean)** indique la nature des requête a tracer
 - ...

Gestion des logs

- Les logs d'activités
 - Paramètres du fichier de trace
 - **log_rotation_age (entier)** indique le temps de vie d'un fichier de traces d'activités
 - **log_rotation_size (entier)** indique la taille maximale d'un fichier de traces
 - ...

PostgreSQL, administration

alphorm.com™©



Alphorm.com



La gestion des transactions sous PostgreSQL

Site : <http://alphorm.com>
Blog : <http://alphorm.com/blog>
Forum : <http://alphorm.com/forum>

Nouredine DRISSI
Expert consultant bases de données
Contact : ndform@valneo-xi.fr

PostgreSQL, administration

alphorm.com™©

Gestion des logs

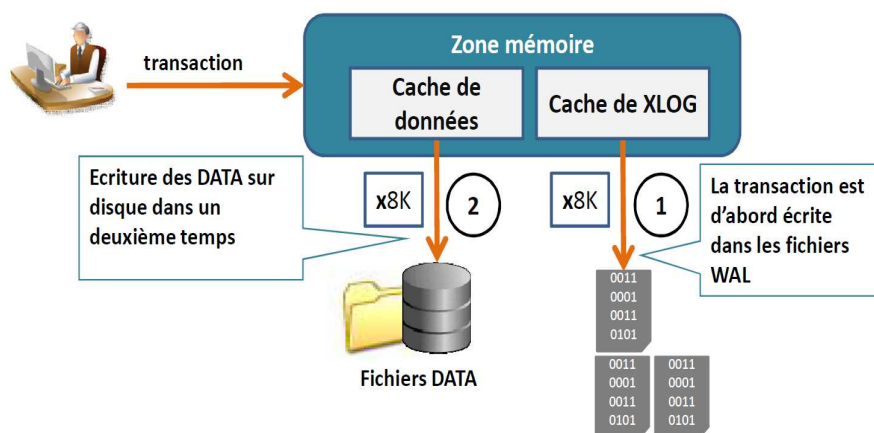
- Les logs binaires ou WAL
 - Mécanisme d'écriture des transactions
 - Chaque transaction est écrite dans un fichier binaire, avant que les données ne soient réellement écrites dans les fichiers de données et d'index
 - Mécanisme qui garantit l'écriture des données
 - Permet d'obtenir de meilleures performances, tout en garantissant l'écriture des données

PostgreSQL, administration

alphorm.com™©

Gestion des logs

- Les logs binaires ou WAL
 - Schéma du mécanisme d'écriture des transactions



PostgreSQL, administration

alphorm.com™©

Gestion des logs

- Les logs binaires ou WAL
 - Caractéristiques
 - Chaque segment WAL a une taille de 16 Mo
 - Chaque segment est divisé en pages de 8 Ko
 - Les fichiers sont nommés suivant un nombre qui est toujours incrémenté et qui commence à 000000010000000000000000
 - Les segments WAL sont stockés dans le répertoire **pg_xlog**
 - Le serveur peut en créer plusieurs si nécessaire

Gestion des logs

- Les logs binaires ou WAL
 - Paramètres de configuration
 - **fsync(boolean)** permet de forcer l'écriture des fichiers de traces binaires
 - **checkpoint_segment(entier)** indique l'écart maximum entre deux points de contrôle
 - **checkpoint_timeout(entier)** indique le temps maximal, en seconde, entre deux points de contrôle
 - **checkpoint_warning(entier)** déclenche l'envoi de messages dans les traces d'activités lorsque les points de contrôles sont trop fréquents
 - **wal_buffer** cache des logs binaires



Prochain module :
Les logs PostgreSQL



PostgreSQL, administration

alphorm.com™©



Alphorm
_com



Le cache de données
PostgreSQL

Site : <http://alphorm.com>
Blog : <http://alphorm.com/blog>
Forum : <http://alphorm.com/forum>

Nouredine DRISSI
Expert consultant bases de données
Contact : ndform@valneo-xi.fr

PostgreSQL, administration

alphorm.com™©

Le cache de données

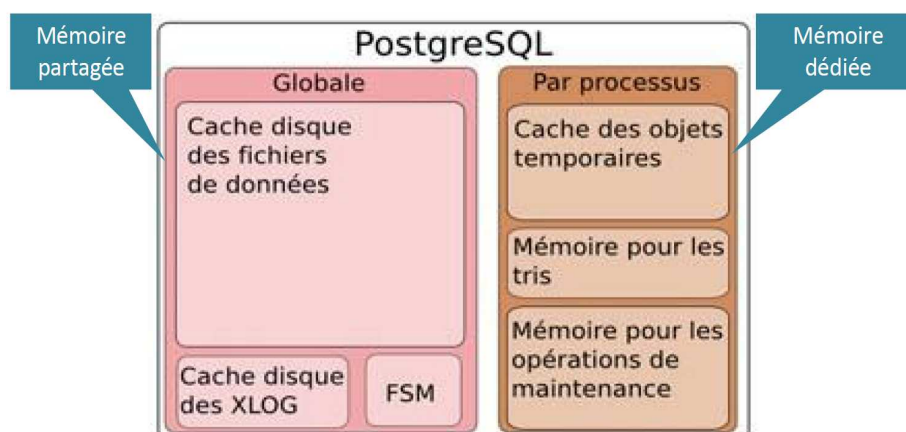
- Le cache PostgreSQL
 - `shared_buffers` zone de travail principale partagée
 - `wal_buffers` cache pour les transactions WAL partagé
 - `temp_buffers` cache pour les tables temporaire locale
 - `work_mem` cache pour les opérations de tris
 - `maintenance_work_mem` cache pour les opérations de maintenance

PostgreSQL, administration

alphorm.com™©

Le cache de données

- Le cache PostgreSQL



PostgreSQL, administration

alphorm.com™©



Prochain module :

Gestion de la sécurité sous
PostgreSQL



PostgreSQL, administration

alphorm.com™©



Alphorm
_com



Sauvegarde et Restauration
sous PostgreSQL

Import Export

Site : <http://alphorm.com>
Blog : <http://alphorm.com/blog>
Forum : <http://alphorm.com/forum>

Nouredine DRISSI
Expert consultant bases de données
Contact : ndform@valneo-xi.fr

PostgreSQL, administration

alphorm.com™©

Sauvegarde et restauration

- Trois stratégies de sauvegardes possibles
 - Export à partir d'une base de données ou une instance active en utilisant le programme **pg_dump** et **pg_dumpall**
 - Sauvegarde des fichiers de l'instance lorsqu'elle est arrêtée (sauvegarde à froid)
 - Sauvegarde à chaud en utilisant des journaux des transactions

Sauvegarde et restauration

- Export avec **pg_dump**
 - Permet de transférer rapidement des données entre bases de données
 - Supporte trois types de format, dont le SQL
 - Utilise les mêmes options de connexion que **psql**
 - Simple d'emploi..

Sauvegarde et restauration

- Export avec **pg_dump**
 - Trois types de formats
 - Format texte **SQL** (par défaut), les données seront restaurées avec **psql**
 - Format **tar**, les données seront restaurées avec **pg_restore**
 - Format **dump** spécifique, restauration avec **pg_restore**

Sauvegarde et restauration

- Export avec **pg_dumpall**
 - Permet d'exporter toutes les bases de données d'une instance
 - Un seul type de format d'export SQL



Sauvegarde et Restauration sous PostgreSQL

Sauvegarde à froid

Site : <http://alphorm.com>
Blog : <http://alphorm.com/blog>
Forum : <http://alphorm.com/forum>

Nouredine DRISSI
Expert consultant bases de données
Contact : ndform@valneo-xi.fr

PostgreSQL, administration

alphorm.com™©



Sauvegarde et restauration

- Sauvegarde à froid
 - Arrêt de l'instance
 - Sauvegarde de la racine de l'instance \$PGDATA au niveau système
 - Sauvegarde des répertoires des espaces de tables au niveau système

PostgreSQL, administration

alphorm.com™©

Sauvegarde et restauration

- Restauration à partir d'une sauvegarde à froid
 - Arrêt de l'instance
 - Suppression de tous les répertoires et fichiers de l'instance
 - \$PGDATA
 - Espaces de table
 - Restaurer les fichiers au niveau système
 - Démarrer l'instance

PostgreSQL, administration

alphorm.com™©



Alphorm.com



Sauvegarde et Restauration sous PostgreSQL

Sauvegarde à chaud

Site : <http://alphorm.com>
Blog : <http://alphorm.com/blog>
Forum : <http://alphorm.com/forum>

Nouredine DRISSI
Expert consultant bases de données
Contact : ndform@valneo-xi.fr

PostgreSQL, administration

alphorm.com™©

Sauvegarde et restauration

- Sauvegarde à chaud
 - Pré-requis
 - Activer l'archivage des journaux des transactions
 - wal_level = archive
 - archive_mode = on
 - archive_command = 'cp %p /archives/%f'
 - Exécuter régulièrement une sauvegarde complète de l'instance.

PostgreSQL, administration

alphorm.com™©

Sauvegarde et restauration

- Sauvegarde à chaud
 - Mise en oeuvre
 - Exécution de la fonction SQL *pg_start_backup*
 - Sauvegarde des fichiers de l'instances à l'exception du répertoire pg_xlog
 - Exécution de la fonction *pg_stop_backup*
 - Une étiquette est conservée dans le fichier \$PGDATA/backup_label.
 - Un fichier \$PGDATA/pg_xlog/000000010000x.xxxxxx.backup est créé pour indiquer le dernier journal utilisé avant la sauvegarde.

PostgreSQL, administration

alphorm.com™©

Sauvegarde et restauration

- Sauvegarde à chaud
 - Restauration
 - Si disponibles, mettre de côté les journaux du répertoire *\$PGDATA/pg_xlog*
 - Supprimer tous les fichiers et répertoires de l'instance et ses espaces de tables
 - Restaurer la sauvegarde complète en prenant garde à rétablir les permissions
 - Supprimer les fichiers dans *\$PGDATA/pg_xlog* ou recréer ce répertoire
 - S'assurer que le répertoire *\$PGDATA/pg_xlog/archive_status* existe
 - Recopier les journaux éventuellement mis de côté dans *\$PGDATA/pg_xlog*
 - Créer un fichier **recovery.conf** dans *\$PGDATA* à partir de *recovery.conf.sample*
 - Démarrer PostgreSQL en s'assurant que les utilisateurs ne pourront se connecter

Sauvegarde et restauration

- Sauvegarde à chaud
 - Restauration
 - Le fichier **recovery.conf**
 - `restore_command = 'cp /archive/%f %p'`
 - `recovery_target_time = '2008-04-23 11:20:55 EST'`

Sauvegarde et restauration

- Les étapes de notre TP
 - Je met l'instance en mode ARCHIVELOG
 - Je lance une transaction **T1 insert**
 - Je démarre Le mode sauvegarde à chaud sous PostgreSQL *select pg_start_backup()*
 - Je réalise la sauvegarde au niveau du système d'exploitation *tar*
 - J'arrête le mode sauvegarde à chaud *select pg_stop_backup()*
 - Je lance une transaction **T2 insert**
 - Je bascule vers un nouveau fichier journal *select pg_switch_xlog*
 - Je lance une transaction **T3 insert**
 - Je bascule vers un nouveau fichier journal *select pg_switch_xlog*
 - Je lance une transaction **T4 insert**
 - Je bascule vers un nouveau fichier journal *select pg_switch_xlog*



Prochain module :

Administrer au quotidien PostgreSQL





Alphorm
com



Administrer au quotidien PostgreSQL

Les bases de données

Site : <http://alphorm.com>
Blog : <http://alphorm.com/blog>
Forum : <http://alphorm.com/forum>

Nouredine DRISSI
Expert consultant bases de données
Contact : ndform@valneo-xi.fr

PostgreSQL, administration

alphorm.com™©



Administrer au quotidien

- Pour chaque instance il existe les bases
 - postgres
 - template0
 - template1
 - Les relations (tables, indexes, etc.) sont stockées dans les bases de données
 - Pour chaque base, il existe au mois le schéma *public* et les catalogues (*information_schema* et *pg_catalog*)

PostgreSQL, administration

alphorm.com™©



Administrer au quotidien

- Les relations
 - Désigne des objets (tables, indexes, sequences)
 - Les données de chaque relation sont stockées dans un fichier identifié par son OID
 - \$PGDATA/base/oid_bdd/filenode_relation
 - Exception pour la base de données postgres
 - \$PGDATA/global



Administrer au quotidien

- Création des bases de données
 - Syntaxe

```
CREATE DATABASE nom
[ [ WITH ] [ OWNER [=] utilisateur ] [ TEMPLATE [=] modèle ]
[ ENCODING [=] encodage ] [ TABLESPACE [=] espace_de_tables ]
```
 - Le propriétaire par défaut est l'utilisateur connecté
 - La base de données modèle par défaut est *template1*
 - L'encodage par défaut est celui de l'instance
 - L'espace de tables par défaut est *pg_default*
 - Disposer du droit `createdb` pour pouvoir créer une base



Administrer au quotidien

- Suppression des bases de données
 - Syntaxe
`DROP DATABASE [ IF EXISTS] nom`
 - Aucune connexion sur la base de données à supprimer
 - Seul l'administrateur ou le propriétaire peut supprimer une base de données



Administrer au quotidien

- Les schémas
 - Espace de nom logique
 - Permet d'avoir plusieurs objets portant le même nom dans une même base de données
 - Les objets sont accessible en préfixant le nom de l'objet par le nom du schéma « schéma.objet »
 - La variable *search_path* est utilisée pour la recherche dans les schémas



Administrer au quotidien

- Création des schémas

- Syntaxe

`CREATE SCHEMA nom [ AUTHORIZATION utilisateur ]`

- Le propriétaire par défaut est l'utilisateur connecté
 - Pour créer un schéma
 - Être administrateur
 - Avoir le privilège *create* sur la base de données cible



Administrer au quotidien

- Suppression des schémas

- Syntaxe

`DROP SCHEMA [ IF EXISTS ] nom [ CASCADE ]`

- Seul l'administrateur ou le propriétaire peut supprimer un schéma
 - L'option *CASCADE* permet de supprimer un schéma qui possède des objets



Administrer au quotidien

- Suppression des schémas

- Syntaxe

- ```
DROP SCHEMA [IF EXISTS] nom [CASCADE]
```

- Seul l'administrateur ou le propriétaire peu supprimer un schéma
    - L'option CASCADE permet de supprimer un schéma qui possède des objets