Jean-François Berdjugin, Jean-François Remm 2005-2006
IUT 1, Département SRC, l'Isle d'Abeau

Installation du SGBD relationnel Postgres

Préambule

Un Système de Gestion de Bases de Données est un ensemble de services permettant de gérer des bases de données (accès aux données, manipulations des données, cohérence, persistance, fiabilité partage, ...). Une Base de Données est un ensemble de données organisées suivant un modèle et mémorisées sur un support permanent. Le modèle relationnel permet de décrire une base de données comme étant un ensemble de relations.

Véhicule

No	***Type***	***Annee***
1	R5	1980
2	2CV	1981
3	R4	1982
4	609	2000

Véhicile est la table (une relation). No, Type, Annee sont des attributs. (2,2CV,1981) est un n-uplet (tuple).

Installation sous Windows

Récuperer sur ftp://192.168.107.200 archive de postgress et la décompacter.

Installer le .msi (Microsoft Service Installer) et suivre l'assistant :

- l'instalation comme service (un service est processus qui peut-être lancé au démarrage et qui s'exécute en arrière plan).
- Account Name : Nom de l'utilisateur Windows sous lequel le SGBD s'éxecutera.

Le cluster (catalog cluster) est une collection de bases de données accéssible via une instance d'un serveur.

- Le SGBD sera accèssible sur une adresse IP et un port donné. Un SGBD possède sa propre gestion des utilisateurs ainsi postgres est-il l'administrateur du SGBD mais n'est pas un utilisateur Windows.
- Vous choisirez comme Langue le français et comme encodage le Latin1

La commande "*sc query*" vous permet de connaître le nom du service sous lequel s'éxecute postgress. Les commande "*sc stop nom*" et "*sc start nom*" vous permettrons d'arrêter ou de relancer le service si vous avez à modifier ses fichiers de configuration.

Rem: Vous pouvez aussi utiliser "*net start nom*" et "*net stop nom*".

Administration du SGBD

Nous allons dans un premier temp utiliser **pgAdmin** un client graphique qui va nous permettre d'administrer notre SGBD.

Première connexion

Cet outil vous permet de vous connecter sur un serveur (Adresse IP + port) avec un login est un mot de passe.

Connectez vous sur votre serveur avec le login postgres.

Vous devez obtenir :

- Bases de données : les bases de données gérées par le sgbd
- TableSpaces : des unités logiques de stockage qui permettent à l'administrateur de choisir où seront stockés les objets des bases de données.
- Roles groupes : Postgres gére les droits d'accès en utilisant le concept de rôle. Un rôle peut être vu soit comme un utilisateur de la base de données, soit comme un groupe d'utilisateurs de la base de données, suivant la façon dont le rôle est configuré. Les rôles peuvent posséder des objets de la base de données (par exemple des tables) et peuvent affecter des droits sur ces objets à d'autres rôles pour contrôler qui a accès à ces objets. De plus, il est possible de donner l'appartenance d'un rôle à un autre rôle, l'autorisant du coup à utiliser les droits affectés au rôle dont il est membre.
- Roles de connexion : role utilisé pour les connexions sur le serveur.

Création d'un rôle

Créer un nouvel utilisateur (rôle de connexion) nomé user et pouvant créer des bases de données.

Création d'une base

Créer une base nommée base1 dont user est le propriétaire.

Dans cette **base** vous devez trouver :

- Conversions : permet de définir les règles de conversions entre les différents types de données.
- Langages : permet d'associer un langage procèdural à la base de données. Ce langage peut-être utilisé lors d'événements (trigger).

- **Schéma** : ce que nous allons réélement manipuler et que nous allons détailler ci après.
- Réplication : permer à une base d'être un réplicat (une copie) d'une autre.

Le schéma d'une base consiste en un ensemble de vues contenant des informations sur les objets définis dans la base de données nous y trouvons :

- Agrégats : qui permet de définir des fonctions d'agrégat. Des fonctions qui calculent une seule valeur résultant d'un ensemble de valeurs en entrée, comme la moyenne par exemple.
- Conversions : identiques aux précédantes mais dont la portée s'applique à la base seule
- Domaines : un domaine est un ensemble de valeurs.
- Fonctions/ FonctionsDéclencheur / Procédure :permettent de définir des fonctions et des procédures dans un langage cible.
- Opérateur / Classes d'opérateur : permettent de définir des opérateurs de comparaison
- Séquences : Des suites de nombres utilisées comme compteur
- **Tables** : Les relations et ce que vous allez utiliser.
- Types : Permet de définir de nouveaux types.
- Vues : Une vue est le résultat de l'éxectution d'une requête qui peut utilisé comme une table.

Ce qu'il est important de retenir c'est qu'un SGBD relationnel est un ensemble de services qui permettent de gérer des bases de données et que chaque base de données contient un ensemble de table (les relations).

Création de tables

Créer une table Vehicule avec trois colonnes :

- No de type serial avec comme contrainte d'être une clef primaire.
- Type de type varchar ne pouvant être nulle
- Annee de type varchar

Une fois votre table crée vous devez y trouver l'arborescence suivante :

-Vehicule

-Colonnes

-No

-Type

-Annee

-Contraintes

-Vehicule pkey

-Indexes

-Règles

-Déclencheur

La contrainte clef primaire indique que la valeur du No ne peut être nulle et doit être unique.

Ainsi par exemple la table Vehicule ne pourra contenir les enregistrements suivants :

(2,2CV,1981) et (2,4L,1905).

Les indexes sont des mécanismes utilisés pour améliorer le temps d'accés aux données. Les indexes ralentissent le système lors de l'insertion des données mais augementent les performances lors des recherches.

Les règles et les déclencheurs permettent de restreindre les plages de données lors de certains événements (insertion, mise à jour, suppression).

Créer de même une deuxième table Personne avec quatres colonnes :

- No de type serial avec comme contrainte d'être une clef primaire.
- Nom de type varchar non nulle
- Prenom de type varchar
- Naissance de type Date

Créer enfin une table Possede avec deux colonnes :

- No_Personne de type int
- No_Voiture de type int

No_Personne et No_Voiture doivent former la clef primaire de la relation Possede.

No_Personne doit être une clef étrangère telle que le No_Personne référence la clef primaire de Personne (No). Une clef étrangère est un moyen d'exprimer une contrainte d'intégrité référentielle. Une voiture de peut-être possédée par une personne qui n'existe pas.

De même No_Voiture doit être une clef étrangére telle que No_Voiture référence la clef primaire de Voiture(No).

Peupler la base

Nous allons insérer des tuples dans nos relations, commençons par la table "*Personne*".

La première méthode est graphique : un click droit sur la table Personne vous permet de trouver la propriété "*afficher*" qui est en réalité un formulaire de saisie.

Personne

Peupler personne avec les tuples suivants :

"1";"Durand";"Jean";"1980-10-10"

"11";"Belmondo";"Paul";"1990-10-10"

"2";"Belmondo";"Jean-Paul";"1900-01-01"

"4";"Remm";"Jean-François",""

Essayer de saisir "4";"Berdjugin"; "Jean-François" quel est le problème ?

Essayer de mêm "5";"";"Jean-François" quel est le problème ?

Vehicule

Peupler Véhicule avec mes tuples suivants :

"1";"R4";"1980"

"2";"R5";"1981"

"3";"Super5";"1990"

"4";"Clio";"2000"

Possede

Nous souhaitons maintenant exprimer le le fait que Jean-François Remm (no 4) possède une super 5 (no 3) et une Clio (no 4); que Paul Belmondo (no 11) posséde la R4 (no 1); que Jean-Paul Belmondo (no 2) possède la R4 (no 1) et la R5 (no 2).

La table possède devra posséder les enregistrements suivants :

"2";"1"

"2";"2"

"4";"3"

"4";"4"

"11";"1"

Essayer de saisir "10";"1" ou "11";"5" que se passe-t-il, pourquoi ?

Essayer de saisir "11";"1" que se passe-t-il, pourquoi ?

Structured Query Language

SQL est un :

- Langage de manipulation des données
- Langage de définition des données
- Langage de contrôle des données

La création de la table Possede est par exemple réalisée par la commande SQL suivante (LDD) :

```
CREATE TABLE "Possede"
(
  "No_Personne" int2 NOT NULL,
  "No_Vehicule" int2 NOT NULL,
  CONSTRAINT "Possede_pkey" PRIMARY KEY ("No_Personne", "No_Vehicule"),
  CONSTRAINT "Possede_No_Personne_fkey" FOREIGN KEY ("No_Personne")
     REFERENCES "Personne" ("No") MATCH SIMPLE
     ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT "Possede_No_Vehicule_fkey" FOREIGN KEY ("No_Vehicule")
     REFERENCES "Vehicule" ("No") MATCH SIMPLE
     ON UPDATE NO ACTION ON DELETE NO ACTION
) ;
```

L'insertion d'un enregistrement (tuple) a été réalisé par la commande suivante (LMD) :

```
INSERT INTO Possede (No_Personne, No_Vehicule) VALUES (2,1);
```

Sans y préter attention vous avez aussi défini des prorpiétaires et donnés des droits sur vos tables :

```
ALTER TABLE "Possede" OWNER TO postgres;
```

Maintenant nous souhaterions manipuler éxecuter quelques requêtes SQL :

Dans l'invite SQL tapper les requêtes suivantes :

1. select * from "Personne";
2. select "Nom"
   from "Personne";
3. select distinct "Nom"
   from "Personne"
4. select "Nom", "Prenom", "Type"
   from "Personne", "Vehicule", "Possede"

```
where "Personne"."No"="No_Personne" and "Vehicule"."No" = "No_Vehicule";
```

5.
```
select "Nom", "Prenom", "Type"
from "Personne", "Vehicule", "Possede"
where "Personne"."No"="No_Personne" and "Vehicule"."No" = "No_Vehicule" and "Type"='R5';
```

Donner la requête permettant de connaître le nom et le prenom des propriétaires de clio.

PSQL

Postgres est livré aussi bien sous Linux que sous Windows avec un utilitaire en ligne de commande.

Lancer l'utilitaire

Lancer depuis le menu démarer et le sous menu Postgres une invite de commande.

Taper psql --help pour avoir l'aide puis connectez vous en local avec le login user et le pass que vous aviez choisi sur la base base1.

Une fois connécté essayer, par exemple, de lister tous les enregistrements de la table personne.

\q vous permet de quitter.

Utiliser un fichier pour créer une base

Psql peut aussi vous être util pour gréer une base de donnée depuis un fichier texte. Utiliser le fichier texte donné avec l'énoncé pour générer la base de donnée :

1. Créer la base : createdb -U user recette
2. Utiliser le script : psql -U user -f recette.sql recette

Vous pouvez vérifier avec pgAdmin ou psql la création de la base et son peuplement

Se connecter sur le SGBD de son voisin

Essayer de vous connecter sur le serveur de votre voisin, si la connexion est refusée demander à votre voisin d'authoriser votre machine dans le fichier pg_hba.conf et de relancer son serveur.

Instalation sous Linux

Ce qui suit est extrait d'un article de Jacques Grelet sur www.idr.fr.

Instalation

Vous pouvez vérifier si prostgres est installé avec la commande :

rpm -qa | grep postgres

Si ce n'est pas le cas installer le serveur et les outils clients, vous pouvez aussi installer pgaccess une interface graphique.

Comme pour les autres démons, on peut démarrer ou arreter le serveur en étant root par la commande :

/etc/rc.d/init.d/postgresql {start | stop }

ou mieux :

service postgresql {start | stop }

On vérifiera que le programme fonctionne en arrière plan :

ps aux |grep postgres

Le processus suivant appartient à l'utilisateur postgres est en exécution :

postgres 4426 /usr/bin/postmaster -p 5432

Le processus d'installation à créé une arborescence sous /var/lib/pgsql, avec 2 bases initiales qui

s'appellent postgres et template1.

Sous l'utilisateur postgres ou root, éditer le fichier de configuration /var/lib/pgsql/data/postgresql.conf et autoriser les connections par le réseau au serveur :

```
tcpip_socket = true
```

Vérifier également l'initialisation des variables locales comme :

```
timezone = UTC
#
#       Locale settings
#
# (initialized by initdb -- may be changed)
LC_MESSAGES = 'fr_FR.UTF-8'
LC_MONETARY = 'fr_FR.UTF-8'
LC_NUMERIC = 'fr_FR.UTF-8'
LC_TIME = 'fr_FR.UTF-8'
```

Editer également le fichier pg_hba.conf pour modifier ou ajouter/restreindre les autorisations d'accés au travers du réseau en fonction des contraintes liées à la sécurité.

Initialiser PostgreSQL

```
su -l pgsql -c initdb
```

On se consultera les pages man et autres fichiers README, INSTALL car les procédures de démarrage varient quelque peu d'une version/distribution à une autre.

Tests

Tester le fonctionnement du serveur en se connectant avec le client psql.

```
$ su -l postgres
```

Password:

-bash-2.05b$ psql -d template1

Welcome to psql 7.3.4-RH, the PostgreSQL interactive terminal.

Type: \copyright for distribution terms
\h for help with SQL commands
\? for help on internal slash commands
\g or terminate with semicolon to execute query
\q to quit

template1=#

Creation d'une base

Pour créer des bases de données vides ou des utilisateurs on utilisera les scripts fournit avec PostgreSQL :

$ createdb <database>

en étant root

createdb -U pgsql <database>

Il est plus simple de créer une base vide pour l'utilisateur afin qu'il puisse se connecter à postgreSQL directement avec la commande psql.

Il devra ensuite se connecter à sa base avec \c sous psql ou avec des outils graphiques gratuits comme pgadmin3 ou pgaccess.

Pour plus d'informations sur les commandes, utiliser l'option help :

createdb --help

Suppression d'une base : $ dropdb <database>