Exploitation d'une base de données

Département Informatique

IUT2 de Grenoble

BUT1 - Ressource 2.06

Introduction à l'exploration et l'administration des BD

1. Révisions : BD relationnelle et SQL simple
2. SQL pour explorer et analyser
3. SQL augmenté : *triggers*
4. Administration de 1er niveau

4. Administration de 1er niveau

4.1. Introduction à l'administration d'une BD

- Administrer : créer et maintenir
- Authentification : à ne pas confondre avec les droits

4.2. Droits

- Rôle
- Droits sur une base
- Droits sur les tables et vues qu'elle contient

4.3. Vue et droits

4. Administration de 1er niveau

4.1. Introduction à l'administration d'une BD

- Administrer : créer et maintenir
- Authentification : à ne pas confondre avec les droits

4.2. Droits

- Rôle
- Droits sur une base
- Droits sur les tables et vues qu'elle contient

4.3. Vue et droits

Administrer une base de données

Faire en sorte qu'elle puisse être utilisée dans de bonnes conditions

La mise en place initiale

- installer et configurer le programme serveur `[postgres]`
- créer un rôle gestionnaire des bases et rôles `[postgres]`
- créer la base et les utilisateurs `[gestionnaire]`
- définir les données `[propriétaire]`

Assurer une utilisation performante et sûre

- performante : exécution rapide des requêtes
- sûre

Administrer une base de données

Faire en sorte qu'elle puisse être utilisée dans de bonnes conditions

La mise en place initiale

- installer et configurer le programme serveur `[postgres]`
- créer un rôle gestionnaire des bases et rôles `[postgres]`
- créer la base et les utilisateurs `[gestionnaire]`
- définir les données `[propriétaire]`

Assurer une utilisation performante et sûre

- performante : exécution rapide des requêtes
- sûre

Installer et configurer le programme serveur

⚠ À ne pas confondre avec

- le programme client **`psql`**
- la machine hôte sur laquelle s'exécute le programme serveur (**`-h`**)

Programme serveur **`postgres`**

- **`-D datadir`** : répertoire contenant les données
- **`-c config_file`** : fichier de configuration (**`postgresql.conf`**)
  - Paramètres physiques de la machine hôte
  - Paramètres liés aux actions d'admin/maintenance
  - Paramètres d'utilisation par les clients
- 1 processus père et des fils (1 par connexion), exécutés par le user unix **`postgres`**

Administrer une base de données

Faire en sorte qu'elle puisse être utilisée dans de bonnes conditions

La mise en place initiale

- installer et configurer le programme serveur `[postgres]`
- créer un rôle gestionnaire des bases et rôles `[postgres]`
- créer la base et les utilisateurs `[gestionnaire]`
- définir les données `[propriétaire]`

Assurer une utilisation performante et sûre

- performante : exécution rapide des requêtes
- sûre

Sécurité d'un Système d'Information

Assurer la cohérence des informations (intégrité)

définition et vérification des contraintes

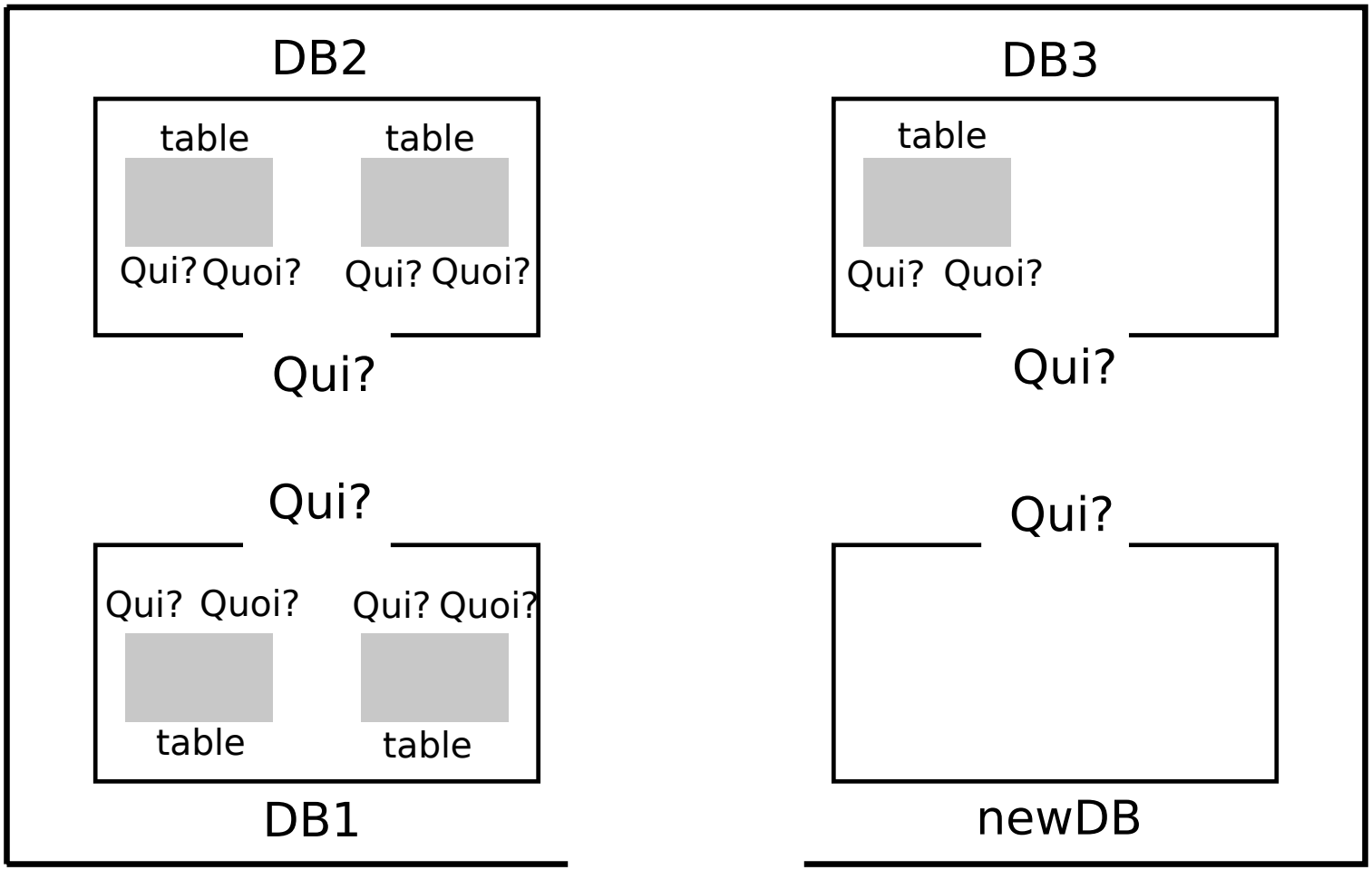
Etre robuste face aux pannes

assurer un service pérenne

Assurer la confidentialité des données

- qui peut passer ? (authentification)
- que peut-il faire une fois à l'intérieur ? (droits)

Authentification : un droit de passage



Authentification

d'où? comment? (mdp, passe-droit....)

qui? 1ère base

Authentification sous postgreSQL 1/2

- contrôlée par fichier de configuration **pg_hba.conf**
- **qui** peut se connecter **à quelles tables d'où** et **comment**

```
TYPE    DATABASE    USER        ADDRESS          METHOD

# Unix domain socket connections (local)
local   all         postgres                     peer

# IPv4 local connections
host    all         all         127.0.0.1/32     reject

# IPv4 non-local connections
host    all         superman    0.0.0.0/0        scram-sha-256
host    ma_base     +invites    192.168.141.0/24 scram-sha-256
```

Authentification sous postgreSQL 2/2

Méthodes

- `peer` : en local et user(bd) = user(OS)
- `ident` : non local et user(bd) = user(OS)
- `reject` : refusé et lignes suivantes ignorées
- `scram-sha-256` : mot de passe stocké avec hachage
- `pam` : utilise le Pluggable Authentication Module de l'OS
- `trust` : OK sans autre condition

Configuration dans `postgresql.conf`

- `listen_adresses = '*'`
- `password_encryption = scram-sha-256`

Administrer une base de données

Faire en sorte qu'elle puisse être utilisée dans de bonnes conditions

La mise en place initiale

- installer et configurer le programme serveur `[postgres]`
- créer un rôle gestionnaire des bases et rôles `[postgres]`
- créer la base et les utilisateurs `[gestionnaire]`
- définir les données `[propriétaire]`

Assurer une utilisation performante et sûre

- performante : exécution rapide des requêtes
- sûre

Utilisateur

```
psql -h host -U user ma_base
```

```
CREATE USER nom_user [WITH option [...]];
                option :
                         superuser | nosuperuser
                         createrole | nocreaterole
                         createdb | nocreatedb
                         password 'xxx'
```

```
DROP USER nom_user ;
```

```
ALTER USER nom_user WITH option ;
```

⚠ préférer \password pour changer son mot de passe

Administrer une base de données

Faire en sorte qu'elle puisse être utilisée dans de bonnes conditions

La mise en place initiale

- installer et configurer le programme serveur `[postgres]`
- créer un rôle gestionnaire des bases et rôles `[postgres]`
- créer la base et les utilisateurs `[gestionnaire]`
- définir les données `[propriétaire]`

Assurer une utilisation performante et sûre

- performante : exécution rapide des requêtes
- sûre

Pour créer une nouvelle base

Bases définies par défaut

- **`template0`** : modèle de sauvegarde
- **`template1`** : modèle recopié pour créer une base
- **`postgres`** : pour une connexion par défaut

Utilisateur **`with createdb`**

```
CREATE DATABASE nom_base [WITH option [...]];
```

L'utilisateur est le propriétaire de la base.

Pour créer une nouvelle base pour un autre, il faut être **`superuser`**

```
CREATE DATABASE nom_base WITH OWNER=nom_user ;
```

Administrer une base de données

Faire en sorte qu'elle puisse être utilisée dans de bonnes conditions

La mise en place initiale

- installer et configurer le programme serveur `[postgres]`
- créer un rôle gestionnaire des bases et rôles `[postgres]`
- créer la base et les utilisateurs `[gestionnaire]`
- définir les données `[propriétaire]`

Assurer une utilisation performante et sûre

- performante : exécution rapide des requêtes
- sûre

Les droits du propriétaire d'une base

- gérer les droits de connexion sur la base (voir plus loin).

- s'il est aussi **createdb**, il peut la renommer :
  ALTER DATABASE *nom_base* **RENAME** *autre_nom* ;

- s'il est **superuser**, il peut en changer le propriétaire :
  ALTER DATABASE *nom_base* **OWNER TO** *nom_user* ;

Quel propriétaire pour une base ?

Deux situations

- pour une base dont le propriétaire ne gère que les connexions : exécuter par **superuser**
  CREATE DATABASE *nom_base* **WITH OWNER=***nom_user* **;**
- sinon, un *gestionnaire* sera créateur et propriétaire des bases.

Gestionnaire des bases et rôles

- dispose des privilèges **createdb** et **createrole**
- mais n'est pas **superuser** pour éviter les actions fâcheuses

Bilan : un exemple de procédure d'installation

Installation du programme

```
apt install postgresql
```

Vérification du serveur en fonctionnement

```
pg_lsclusters
```

Mise à jour des fichiers de configuration

```
nano postgresql.conf
nano pg_hba.conf
service postgresql restart
```

Création du gestionnaire des bases et rôles

```
su - postgres
psql
create user gege with createdb createrole password 'gege';
\c postgres gege
\password
\q
```