

Université Ibn khaldoun Tiaret
Faculté des Mathématiques et Informatique
Département Informatique
Master Génie Logiciel

16/12/2018

Rapport TP5

Architecture et administration des
SGBD (transaction & journalisation)

Réalisée par :

ZEGAI Houari

Enseignant :

M.BENATHMANE Lalia

1. Objective du TP:

Voir comment utilisé les transactions et la journalisation.

Transaction :

✚ Définition :

Les transactions sont un concept fondamental de tous les systèmes de bases de données. Une transaction assemble plusieurs étapes en une seule opération tout-ou-rien. Les états intermédiaires entre les étapes ne sont pas visibles par les transactions concurrentes. De plus, si un échec survient qui empêche le succès de la transaction, alors aucune des étapes n'affecte la base de données.

✚ Les Propriétés des transactions :

Comme nous avons vu dans le cours, les transactions ont ces propriétés :

ACID (**A**tomicté, **C**ohérence, **I**solation, **D**urabilité).

✚ Comment contrôler les transactions :

Pour contrôler les transactions on utilise ces commandes :

BEGIN TRANSACTION : pour lancer la transaction.

COMMIT : pour effectuer tous les requêtes (ou bien utilise la commande END TRANSACTION).

ROLLBACK : pour annuler toutes les opérations.

✚ Exemple d'utilisation des transactions :

Si l'on considère, par exemple, la base de données qui contient la table « Employee » qui contient les données suivantes :

	id integer	name character varying(150)	salary real
1	1	Houari	12000
2	2	Mohammed	20000
3	3	Omar	3000

Scénario 1 : Commençons maintenant une transaction et supprimons les enregistrements de la table ayant salary < 4000 et nous utilisons enfin la commande ROLLBACK pour annuler toutes les requêtes :

```
SQL Shell (psql)

tp5_sgbd=# SELECT * FROM employee;
 id |  name  | salary 
-----+-----+-----
  1 | Houari |  12000
  2 | Mohammed | 20000
  3 | Omar  |   3000
(3 rows)

tp5_sgbd=# BEGIN;
BEGIN
tp5_sgbd=# DELETE FROM employee WHERE salary < 4000;
DELETE 1
tp5_sgbd=# ROLLBACK;
ROLLBACK
tp5_sgbd=# SELECT * FROM employee;
 id |  name  | salary 
-----+-----+-----
  1 | Houari |  12000
  2 | Mohammed | 20000
  3 | Omar  |   3000
(3 rows)

tp5_sgbd=#
```

Interprétation des résultats : L'instruction ROLLBACK annule toutes les opérations précédentes (aucun changement).

Scénario 1 : Commençons maintenant une transaction et supprimons les enregistrements de la table ayant `salary < 4000` et nous utilisons enfin la commande COMMIT pour effectuer toutes les requêtes :

```
SQL Shell (psql)

tp5_sgbd=# SELECT * FROM employee;
 id |  name  | salary 
----+-----+-----
  1 | Houari | 12000
  2 | Mohammed | 20000
  3 | Omar   |  3000
(3 rows)

tp5_sgbd=# BEGIN;
BEGIN
tp5_sgbd=# DELETE FROM employee WHERE salary < 4000;
DELETE 1
tp5_sgbd=# COMMIT;
COMMIT
tp5_sgbd=# SELECT * FROM employee;
 id |  name  | salary 
----+-----+-----
  1 | Houari | 12000
  2 | Mohammed | 20000
(2 rows)

tp5_sgbd=#
```

Interprétation des résultats : L'enregistrement Omar sera supprimer car l'instruction COMMIT effectuer toutes requêtes.

Journalisation :

Il est très facile d'activer la journalisation dans PostgreSQL en modifiant quelques paramètres de configuration, puis en redémarrant le serveur. Bien que ces paramètres puissent être modifiés «en mémoire», activant ainsi la journalisation temporaire pour cette session cliente uniquement, nous verrons comment configurer postgresql pour créer de manière permanente des fichiers journaux itératifs pour toutes les sessions et connexions.

✚ L'emplacement du fichier de configuration :

La première chose que nous devons savoir où se trouve le fichier de configuration « postgresql.conf », la méthode la plus simple pour trouver l'emplacement consiste à exécuter la commande **SHOW config_file** :

```
SQL Shell (psql)

tp5_sgbd=# SHOW config_file;
               config_file
-----
C:/Program Files/PostgreSQL/9.6/data/postgresql.conf
(1 row)

tp5_sgbd=#
```

Maintenant, ouvrez ce fichier avec votre éditeur de texte préféré et nous pouvons commencer à modifier les paramètres.

✚ Configuration de PostgreSQL pour générer une sortie de journal :

Lorsque le fichier postgresql.conf est ouvert, faites défiler la page RAPPORTS ET ERREURS D'ERREURS et plusieurs options de configuration seront probablement commentées. Les plus critiques de ces paramètres sont `log_destination` et `logging_collector`. Vous trouverez ci-dessous les paramètres recommandés, mais n'hésitez pas à les modifier pour répondre à vos besoins :

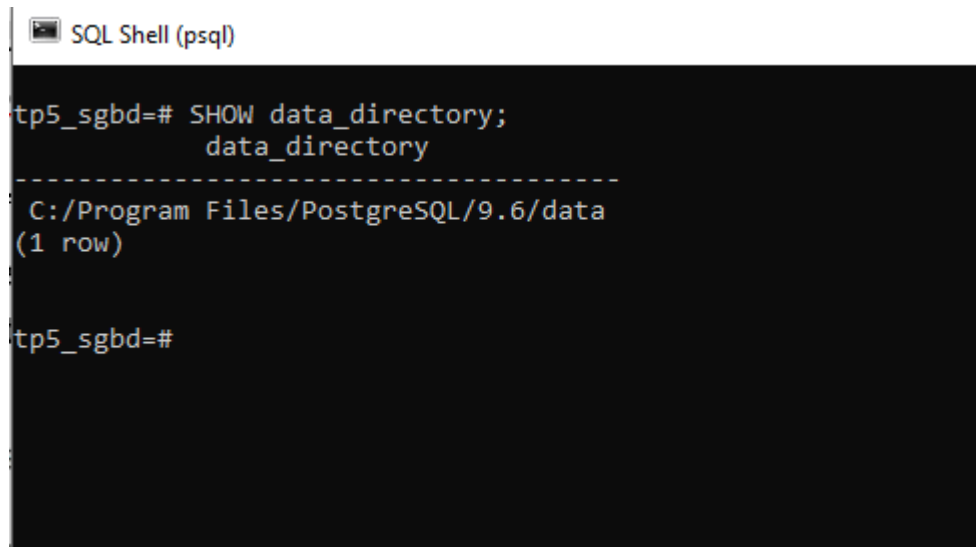
- ✚ `logging_collector` : on (Activer la capture de stderr et csvlog dans les fichiers journaux).
- ✚ `log_destination` : `csvlog` (où se connecter).
- ✚ `log_filename` : `postgresql-%Y-%m-%d_%H%M%S.log` (modèle de nom de fichier journal).
- ✚ `log_directory` : `pg_log` (répertoire dans lequel les fichiers journaux sont écrits).

Nous demandons à postgresql de générer des journaux au format CSV et de les sortir dans le répertoire `pg_log` . Nous avons également annulé la mise en commentaire du paramètre `log_filename` pour produire un nom approprié, y compris des horodatages pour les fichiers journaux.

La dernière étape consiste à redémarrer le service PostgreSQL afin que ces paramètres, en particulier `logging_collector`, prennent effet.

Vérification de la génération de journaux

Nous avons d'abord besoin du chemin du répertoire de données pour votre installation Postgres, et la récupération de ce chemin relève d'une autre instruction SHOW simple déclaration :



```
SQL Shell (psql)

tp5_sgbd=# SHOW data_directory;
          data_directory
-----
C:/Program Files/PostgreSQL/9.6/data
(1 row)

tp5_sgbd=#
```

Une fois le système redémarré, la journalisation doit commencer immédiatement. Pour vous assurer que c'est bien le cas, accédez au répertoire data / pg_log de votre installation Postgres. Il vous suffit donc de naviguer vers ce répertoire en ajoutant / pg_log à la fin pour accéder au répertoire de journalisation.

Vous devriez maintenant voir qu'un fichier journal a été créé après le redémarrage du service précédent.

Là nous l'avons Les fichiers journaux générés automatiquement sont activés avec PostgreSQL en ne modifiant que quelques paramètres de configuration.