ANF SQL – 15/05/2017 – 17/05/2017

Langage de Contrôle de Données

Marie-Claude Quidoz (CEFE/CNRS)

17/05/2017

Plan

- Langage de contrôle des données
  - Gestion des transactions
  - Gestion des privilèges sur un objet
- La notion de rôle
  - Utilisateur / groupe d'utilisateur
  - Gestion des groupes d'utilisateur
- Droit sur les objets créés
  - Droit lors de la création d'un schéma
  - Droit lors de la création d'une table

Plan

- Langage de contrôle des données
  - Gestion des transactions
  - Gestion des privilèges sur un objet
- La notion de rôle
  - Utilisateur / groupe d'utilisateur
  - Gestion des groupes d'utilisateur
- Droit sur les objets créés
  - Droit lors de la création d'un schéma
  - Droit lors de la création d'une table

Langage de contrôle de données

- Protéger la base de données de modifications accidentelles ou intentionnelles qui pourraient mettre en danger son intégrité

- Deux niveaux de protection
  - Gestion des transactions
    - COMMIT, ROLLBACK
  - Gestion des privilèges sur un objet
    - GRANT, REVOKE

Gestion des transactions

- Transaction = ensemble de modifications qui doivent être effectuées en totalité pour que la base reste dans un état cohérent
  - Démarre une transaction : START TRANSACTION
  - Valide les modifications : COMMIT
  - Annuler les modifications : ROLLBACK
  - Possibilité de mettre des points de reprise : SAVEPOINT nom_point
- PostgreSQL fonctionne en auto-commit à moins d'ouvrir explicitement une transaction (cad toute modification sans erreur de syntaxe est définitive)

Exemples

```
SELECT count(*) FROM poisson.methode; – 4 enregistrements
START TRANSACTION;
INSERT INTO poisson.methode(nom_methode,description,url)
VALUES ('pêche à la mouche','La pêche à la mouche consiste à pêcher un poisson
avec un leurre nommé mouche de pêche','https://fr.wikipedia.org/wiki/Pêche_à_la_mouche');
SELECT count(*) FROM poisson.methode; – 5 enregistrements
ROLLBACK;
SELECT count(*) FROM poisson.methode; – 4 enregistrements

SELECT count(*) FROM poisson.methode; – 4 enregistrements
START TRANSACTION;
INSERT INTO poisson.methode(nom_methode,description,url)
VALUES ('pêche à la mouche','La pêche à la mouche consiste à pêcher un poisson
avec un leurre nommé mouche de pêche','https://fr.wikipedia.org/wiki/Pêche_à_la_mouche');
SELECT count(*) FROM poisson.methode; – 5 enregistrements
COMMIT;
SELECT count(*) FROM poisson.methode; – 5 enregistrements
```

Gestion des privilèges sur un objet

- Donner aux utilisateurs uniquement les droits qui lui sont nécessaires pour travailler.
- Protéger l'accès aux données afin d'en assurer la confidentialité.
- Par exemple :
  - "stagiaire" a uniquement le droit de visualiser les enregistrements d'une table.
  - "formation" a le droit de modifier la structure de la table et de modifier les enregistrements.

Donner des autorisations : GRANT

- Donner des droits spécifiques sur un objet
- Droits différents : SELECT, INSERT, UPDATE, DELETE, TRUNCATE, REFERENCES, TRIGGER, CREATE, CONNECT, TEMPORARY, EXECUTE et USAGE.
- Objets différents : SCHEMA, TABLE, COLONNE, VUE, FONCTION, SEQUENCE, ROLE, ...
- Les droits applicables à un objet particulier varient selon le type d'objet (table, fonction...).

Syntaxe de GRANT sur une table

GRANT <droits> **ON** <objet>
TO <usagers>

<droits> ::= ALL | SELECT | DELETE |
INSERT [<attribut>] | UPDATE [<attribut>] |
REFERENCE [<attribut>]

<objet> ::= TABLE <relation> *

<usagers> : := PUBLIC | <username> *

Exemple: GRANT SELECT **ON TABLE** poisson.personne **TO** stagiaire

Exemples d'autorisation sur une table

- Permettre de voir une table
  - GRANT SELECT ON poisson.methode TO stagiaire24
- Permettre d'ajouter des enregistrements à une table
  - GRANT INSERT ON poisson.methode TO stagiaire24
- Permettre de modifier des enregistrements à une table
  - GRANT UPDATE ON poisson.methode TO stagiaire24
- Permettre de supprimer des enregistrements à une table
  - GRANT DELETE ON poisson.methode TO stagiaire24
- Possibilité de donner tous les droits en seule commande
  - GRANT SELECT, INSERT, UPDATE, DELETE ON poisson.methode TO stagiaire24

Supprimer des autorisations : REVOKE

- Supprimer des droits spécifiques sur un objet
- Droits différents : SELECT, INSERT, UPDATE, DELETE, TRUNCATE, REFERENCES, TRIGGER, CREATE, CONNECT, TEMPORARY, EXECUTE et USAGE.
- Objets différents : SCHEMA, TABLE, COLONNE, VUE, FONCTION, SEQUENCE, ROLE, ...
- Les droits applicables à un objet particulier varient selon le type d'objet (table, fonction...).

Syntaxe de REVOKE sur une table

REVOKE <droits> **ON** <objet>
FROM <usagers>

<droits> ::= ALL | SELECT | DELETE |
INSERT [<attribut>] | UPDATE [<attribut>] |
REFERENCE [<attribut>]

<objet> ::= TABLE <relation> *

<usagers> : := PUBLIC | <username> *

Exemple: REVOKE SELECT **ON TABLE** poisson.personne **FROM** stagiaire

Exemples de révocation d'autorisation sur une table

- Empêcher de voir une table
  - REVOKE SELECT ON poisson.methode FROM stagiaire24
- Empêcher d'ajouter des enregistrements à une table
  - REVOKE INSERT ON poisson.methode FROM stagiaire24
- Empêcher de modifier des enregistrements à une table
  - REVOKE UPDATE ON poisson.methode FROM stagiaire24
- Empêcher de supprimer des enregistrements à une table
  - REVOKE DELETE ON poisson.methode FROM stagiaire24
- Possibilité de révoquer tous les droits en seule commande
  - REVOKE SELECT, INSERT, UPDATE, DELETE ON poisson.methode FROM stagiaire24

ALL (objets / privilèges)

- Donne / révoque des droits sur toutes les objets d'un même type sur un ou plusieurs schémas. Disponibles actuellement pour les tables, les séquences et fonctions.
  - GRANT SELECT ON ALL TABLES IN SCHEMA poisson TO stagiaire
  - REVOKE SELECT ON ALL TABLES IN SCHEMA poisson FROM stagiaire
- Donne / révoque tous les droits sur un objet
  - GRANT ALL PRIVILEGES ON poisson.methode TO stagiaire
  - REVOKE ALL PRIVILEGES ON poisson.methode FROM stagiaire
- Possibilité de combiner les deux
  - GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA poisson TO stagiaire
  - REVOKE ALL PRIVILEGES ON ALL TABLES IN SCHEMA poisson FROM stagiaire

PUBLIC (utilisateur)

- Indique que les droits sont donnés à tous les rôles, y compris ceux créés ultérieurement

- GRANT SELECT ON ALL TABLES IN SCHEMA poisson TO public
- REVOKE SELECT ON ALL TABLES IN SCHEMA poisson FROM public

- GRANT ALL PRIVILEGES ON poisson.methode TO public
- REVOKE ALL PRIVILEGES ON poisson.methode FROM public

- GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA poisson TO public
- REVOKE ALL PRIVILEGES ON ALL TABLES IN SCHEMA poisson FROM public

GRANT / REVOKE WITH GRANT OPTION

- Indique que celui qui reçoit le droit peut le transmettre aussi. Cette option ne peut pas être donnée à PUBLIC
  - GRANT SELECT ON ALL TABLES IN SCHEMA poisson TO stagiaire WITH GRANT OPTION
  - GRANT ALL PRIVILEGES ON poisson.methode TO stagiaire WITH GRANT OPTION

- Il est possible de révoquer
  - Uniquement l'option de transmission
    - REVOKE GRANT OPTION FOR SELECT ON ALL TABLES IN SCHEMA poisson FROM stagiaire
  - Le droit et l'option de transmission
    - REVOKE SELECT ON ALL TABLES IN SCHEMA poisson FROM stagiaire

Plan

- Langage de contrôle des données
  - Gestion des transactions
  - Gestion des privilèges sur un objet
- La notion de rôle
  - Utilisateur / groupe d'utilisateur
  - Gestion des groupes d'utilisateur
- Droit sur les objets créés
  - Droit lors de la création d'un schéma
  - Droit lors de la création d'une table

ROLES

- Concept des "utilisateurs" ou des "groupes"
  - Utilisateur : droit de se connecter
  - Groupe : ensemble d'utilisateurs -> affecter des droits
  - Utilisateur : stagiaire1 – Groupe : lecteur_savoie
- Conceptuellement les rôles de la base sont totalement séparés des utilisateurs du système d'exploitation (possibilité cependant de les maintenir en correspondance)
- Globaux à toutes les bases de données de l'instance

Créer un utilisateur

- CREATE USER stagiaire25 ENCRYPTED PASSWORD 'formation'
- CREATE ROLE stagiaire25 LOGIN ENCRYPTED PASSWORD 'formation'
  - Créé un rôle de connexion "stagiaire25"
  - Sans les droits de créer une base de données
  - Sans les droits de créer des rôles
  - Sans le droit super administrateur
- DROP USER stagiaire25
  - Supprime le rôle de connexion "stagiaire25"

Superutilisateur : postgres

- Un rôle est prédéfini à l'installation. Par habitude, ce rôle sera nommé "postgres". Il a le statut d'un superutilisateur qui a tous les droits.

- Sa première mission devrait être de créer un utilisateur qui a le droit de créer des bases de données et des rôles.

- CREATE ROLE dba LOGIN ENCRYPTED PASSWORD 'formation' CREATEDB CREATEROLE;

Créer un groupe d'utilisateur

- CREATE ROLE lecture_BDD
  - Créé un rôle groupe "lecture_BDD"
  - Sans le droit de se connecter
  - Sans les droits de créer une base de données
  - Sans les droits de créer des rôles
  - Sans le droit super administrateur

- DROP ROLE lecture_BDD
  - Supprime le rôle groupe "lecture_BDD"

Gestion d'un groupe d'utilisateur

- GRANT lecture_BDD TO stagiaire25
  - Ajoute l'utilisateur "stagiaire25" au groupe "lecture_BDD"
- GRANT écriture_BDD TO formateur5
  - Ajoute l'utilisateur "formateur5" au groupe "ecriture_BDD"
- REVOKE lecture_BDD FROM stagiaire25
  - Supprime l'utilisateur "stagiaire25" du groupe "lecture_BDD"
- REVOKE ecriture_BDD FROM formateur5
  - Supprime l'utilisateur "formateur5" du groupe "ecriture_BDD"

Exemples d'autorisation / révocation sur une table

- Permettre à tous les membres du groupe « lecture_BDD » de voir une table
  - GRANT SELECT ON poisson.methode TO lecture_BDD
- Permettre à tous les membres du groupe « ecriture_BDD » de modifier les enregistrements d'une table
  - GRANT SELECT,INSERT,UPDATE,DELETE ON poisson.methode TO ecriture_BDD
- Permettre à tous les membres du groupe « ecriture_BDD » de modifier une table et ses enregistrements
  - GRANT ALL ON poisson.methode TO ecriture_BDD

Supprimer des rôles

- DROP USER stagiaire25 / DROP ROLE lecture_BDD
- Tout objet appartenant à un rôle doit d'abord être supprimé ou réaffecté à d'autres propriétaires et tout droit donné à un rôle doit être révoqué.
- Pour réaffecter une table
  - ALTER TABLE poisson.methode OWNER TO stagiaire24;
- Pour réaffecter tous les objets du rôle à supprimer
  - REASSIGN OWNED BY stagiaire25 TO stagiaire24;

Plan

- Langage de contrôle des données
  - Gestion des transactions
  - Gestion des privilèges sur un objet
- La notion de rôle
  - Utilisateur / groupe d'utilisateur
  - Gestion des groupes d'utilisateur
- Droit sur les objets créés
  - Droit lors de la création d'un schéma
  - Droit lors de la création d'une table

Création d'un objet

- Quand un objet est créé, il se voit affecter un propriétaire. Le propriétaire est normalement le rôle qui a exécuté la requête de création.

- Pour la plupart des objets, l'état initial est que seul le propriétaire (et les superutilisateurs) peuvent faire quelque chose avec cet objet. Pour permettre aux autres rôles de l'utiliser, des droits doivent être donnés.

Changement de propriétaire

- Un objet peut se voir affecter un nouveau propriétaire avec la commande ALTER correspondant à l'objet.

  ALTER TABLE poisson.methode OWNER TO stagiaire24;

- Les superutilisateurs peuvent toujours le faire. Les rôles ordinaires peuvent seulement le faire s'ils sont le propriétaire actuel de l'objet (ou un membre du rôle propriétaire) et un membre du nouveau rôle propriétaire.

Droit sur les schémas

- Si on créé un nouveau schéma, seul le propriétaire (dba) a des droits (et tous les droits)

  CREATE SCHEMA test_schema AUTHORIZATION dba;

- Il faudra immédiatement qu'il donne des droits aux autres utilisateurs (si il existe d'autres utilisateurs !)

  GRANT USAGE ON SCHEMA test_schema TO lecture_BDD;

  GRANT ALL ON SCHEMA test_schema TO ecriture_BDD;

Schéma "public"

- Le schéma public est le schéma créé lors de la création de la base de données.
- Tous les utilisateurs ont tous les accès à ce schéma.

```
CREATE SCHEMA public AUTHORIZATION postgres;
GRANT ALL ON SCHEMA public TO postgres;
GRANT ALL ON SCHEMA public TO public;
```

Droits sur les tables

- Si on créé une nouvelle table, seul le propriétaire (dba) a des droits (et tous les droits)

  CREATE TABLE test_bd (id integer, libelle character varying);

- Il faudra immédiatement qu'il donne des droits aux autres utilisateurs (si il existe d'autres utilisateurs !)

  GRANT SELECT ON TABLE test_bd TO lecture_BDD;

  GRANT ALL ON TABLE test_bd TO ecriture_BDD;

En conclusion

- Faire simple :
  - Des groupes d'utilisateurs : lecture_ et ecriture
  - SELECT pour lecture_ & ALL pour ecriture_
  - Propriétaire : ecriture_
- Pour en savoir plus sur les commandes
  - http://docs.postgresql.fr/9.5/sql-grant.html
  - http://docs.postgresql.fr/9.5/sql-revoke.html
  - http://docs.postgresql.fr/9.5/user-manag.html