

Introduction à l'administration des bases de données

Cours 4*

3 avril 2019

Note globale : DS Promo (2/3) + DS de groupe (1/3)
Isabelle Gonçalves - isabelle.goncalves@univ-lyon1.fr

* contient des extraits des cours de Guillaume Cabanac, Fouad Dahak, Yacine Bouhabel, Amrouche Karima, Richard Grin, Adnene Belfodil, Anes Bendimerad, Alain Pillot, Mooneswar Ramburrun + de la documentation Oracle

- Une vue est une **table virtuelle** construite à partir d'une ou plusieurs tables
 - mécanisme permettant de consulter des données issues d'une ou plusieurs tables de la base
 - en les restituant sous une forme différente de leur implémentation physique
 - l'utilisateur qui interroge une vue a l'illusion d'accéder à une table contenant réellement des données
- Une vue est une fenêtre dynamique sur les données
 - La définition de la requête est stockée mais pas les données
 - Le résultat est calculé dynamiquement

- Gérer la confidentialité des données
 - on peut prévoir des vues différentes par utilisateur
 - on peut restreindre les données accessibles des tables à la vue
- Préparer des requêtes
 - extraire une partie des données de certaines tables et faire des jointures seulement sur cette partie
 - optimiser une requête complexe car la vue se manipule exactement comme une table

Les vue : syntaxe

- Création d'une vue

```
CREATE VIEW nomVue  
[(aliascolonne1, aliascolonne2, ...)]  
AS ordre de sélection des données ;
```

- Exemple

```
CREATE VIEW uneVue  
AS select * FROM  
Commande, Employe  
WHERE Commande.numemp = Employe.numemp  
AND fonctionemp='Vendeur' ;
```

- Suppression

```
DROP VIEW nomVue ;
```

Un schéma est un ensemble d'objets, qui sont les structures logiques renvoyant directement aux données de la base. Ces structures peuvent être des tables, des vues, des séquences, des procédures stockées, ...

Sous Oracle, à chaque utilisateur de la base de données est associé un schéma de même nom. Par défaut, chaque utilisateur crée tous les objets dans le schéma correspondant et peut y accéder

On peut accéder, si on a les privilèges nécessaires, aux objets d'un autre utilisateur avec `schéma.nomObjet` (ex : `scott.emp` pour la table `emp` de l'utilisateur `scott`)

Utilisateur : création et suppression

Création d'un utilisateur

```
CREATE USER nomUtilisateur  
DEFAULT TABLESPACE nomTableSpace  
IDENTIFIED BY motDePasse  
QUOTA {UNLIMITED | tailleEnOctet } ON nomTableSpace ;
```

Création d'un utilisateur

```
DROP USER nomUtilisateur [CASCADE ]
```

Privilèges : définition, octroi et révocation

Les privilèges consistent en l'ensemble des opérations autorisées pour chaque utilisateur. Ils permettent de protéger les données en accordant (ou retirant) des droits différents à chaque utilisateur ou groupe d'utilisateurs.

Chaque utilisateur dispose des privilèges qui lui sont octroyés avec la commande **GRANT**. Par ailleurs, un utilisateur peut se voir révoquer des privilèges en utilisant la commande **REVOKE**.

GRANT : Octroi de privilèges

```
GRANT {ALL PRIVILEGES | priv1, ... privN } TO nomUtilisateur ;
```

REVOKE : Révocation de privilèges

```
REVOKE {ALL PRIVILEGES | priv1, ... privN } FROM nomUtilisateur ;
```

Privilèges : exemples

- - Exécuté en étant connecté en tant que SYSTEM
- - octroie les droits de connexion et de création de tables et de vues
`GRANT CREATE SESSION, CREATE TABLE, CREATE VIEW TO`
bertrand ;
- - Exécuté en étant connecté en tant que martin (propriétaire) ou SYSTEM
- - autorise bertrand à interroger la table bertrand.movie
`GRANT SELECT ON martin.movie TO` bertrand ;
- - autorise bertrand à ajouter des lignes dans martin.catégorie
`GRANT INSERT ON martin.categorie TO` bertrand ;

Privilèges : exemples

- - Exécuté en étant connecté en tant que SYSTEM
- - l'utilisateur bertrand ne pourra plus créer des vues
`REVOKE CREATE VIEW FROM bertrand ;`
- - Exécuté en étant connecté en tant que martin (propriétaire) ou SYSTEM
- - annule l'autorisation d'interroger la table martin.movie
`GRANT SELECT ON martin.movie FROM bertrand ;`
- - annule l'autorisation d'ajouter des lignes dans martin.categorie
`REVOKE INSERT ON martin.categorie FROM bertrand ;`

Rôle : regroupement de privilèges

Définition

Un rôle est un regroupement de privilèges. Une fois créé, il peut être assigné à un utilisateur.

Syntaxe

- - définition d'un rôle

CREATE ROLE nomRole ;

- - octroie des privilèges au rôle nomRole

GRANT {ALL PRIVILEGES | priv1, ... privN } TO nomRole ;

- - octroie du rôle nomRole à l'utilisateur nomUtilisateur

GRANT nomRole TO nomUtilisateur ;

création et attribution d'un rôle

- - Création d'un rôle et octroi de privilèges à ce rôle

```
CREATE ROLE etudiant ;
```

```
GRANT CREATE SESSION, CREATE TABLE, CREATE VIEW TO  
etudiant ;
```

- - création d'un utilisateur et octroi du rôle

```
CREATE USER bertrand ...
```

```
GRANT etudiant TO bertrand ;
```

- - Avantage des rôles : modification répercutée sur tous les utilisateurs

```
GRANT CREATE SEQUENCE TO etudiant ;
```

```
REVOKE CREATE VIEW FROM etudiant ;
```

création d'un rôle par extension d'un rôle existant

- - Un autre avantage des rôles : héritage des privilèges par création hiérarchique des rôles

```
CREATE ROLE enseignant ;
```

```
GRANT etudiant, CREATE PROCEDURE TO enseignant ;
```

```
CREATE USER martin ...
```

```
GRANT enseignant TO martin ;
```

consultation des rôles et privilèges d'un utilisateur

- - consultation des rôles octroyés au compte courant

```
SELECT * FROM user_role_privs;
```

- - consultation des privilèges octroyés au compte courant

```
SELECT * FROM user_tab_privs_recd;
```

Le rôle **dba** (database administrator) rassemble tous les privilèges

Définition

Le méta-schéma du SGBD mémorise des informations sur les objets qu'il gère : utilisateurs, tables, vues, procédures, etc. Il comprend des vues statiques (modifiées dans les tables associées uniquement en cas de changement du dictionnaire des données comme par exemple si on crée une table). Ces vues préfixées «all_», «dba_», «user_» sont interrogeables par les utilisateurs.

exemple : occupation disque des objets du schéma courant

- - Interrogation de la vue user_segments depuis le compte martin

```
SELECT segment_name, tablespace_name, bytes/1024 Ko
FROM user_segments ORDER BY Ko DESC;
```

Méta-schéma : les types de vues

- Vues préfixées par «dba__» : elles listent les informations sur tous les objets de la BD.
- Vues préfixées par «all__» : elles listent les informations sur les objets accessibles par l'utilisateur courant.
- Vues préfixées par «user__» : elles listent les informations sur les objets possédés par l'utilisateur courant.

Méta-schéma : Quelques vues

- dba_users : les utilisateurs de la base de données
- user_segments : l'espace disque occupé par les objets de l'utilisateur courant
- user_tables : les tables créées par l'utilisateur courant
- user_constraints : les contraintes créées sur des tables de l'utilisateur courant
- user_procedures : procédures et fonctions créées par l'utilisateur courant
- user_views : vues créées par l'utilisateur courant
- user_sys_privs : privilèges octroyés à l'utilisateur courant
- user_role_privs : rôle octroyés à l'utilisateur courant
- role_sys_privs : privilèges octroyés au rôles