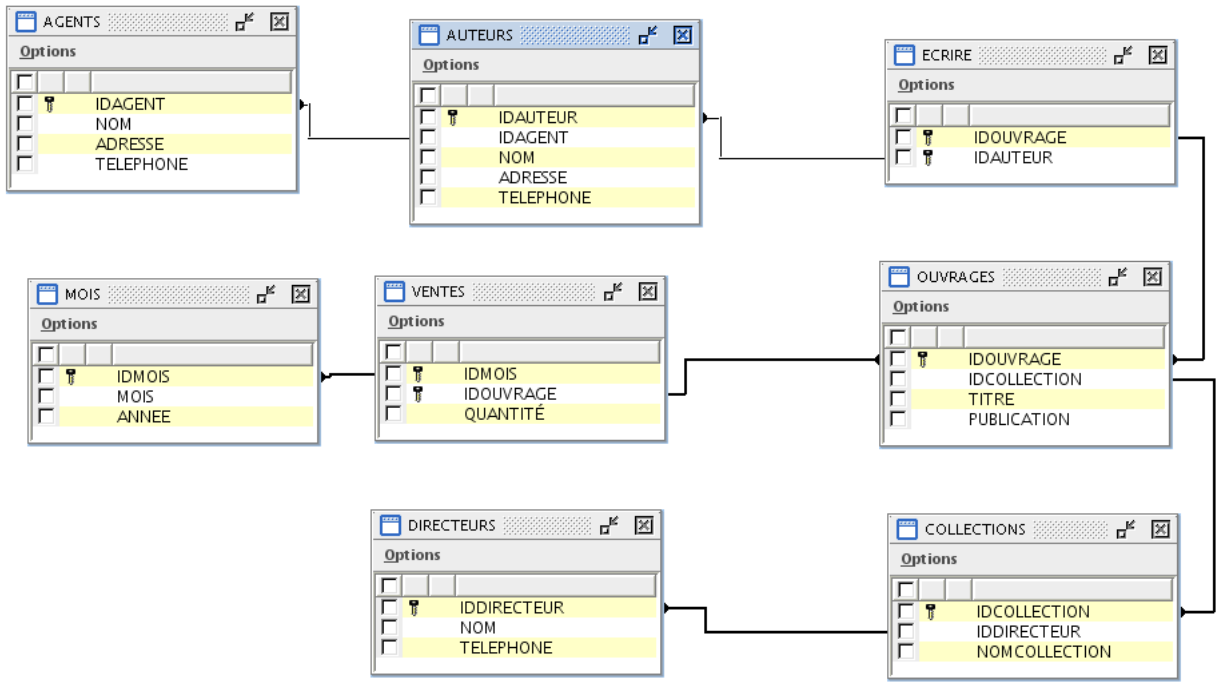
Exercice de SGBD
Protection des données : maison d'édition (Solution)

Jacques THOORENS

22 avril 2010



Avant de commencer

Le schéma *Edition* doit contenir les tables ci-dessus et leurs données. Il faut donc créer un pseudo-administrateur au moyen de la commande suivante :

```
GRANT CONNECT, RESOURCE, UNLIMITED TABLESPACE TO edition
IDENTIFIED BY codd;
GRANT CREATE SEQUENCE TO edition;
GRANT CREATE TRIGGER TO edition;
```

L'utilisateur *edition* sera chargé de donner des droits aux utilisateurs et aux rôles créés par l'administrateur (ou son délégué). Notons qu'il faut donner le droit supplémentaire si on veut créer des vues.

```
GRANT CREATE VIEW TO edition;
```

Créer utilisateurs et rôles

Je reprends ici la création des groupes et des utilisateurs (le mot de passe est arbitraire : mdp). Ces objets doivent être créés par quelqu'un disposant des droits de créer utilisateurs et rôles. Ce sera l'administrateur ou *Edition* pourvu de ces droits. Voici les commandes pour lui donner ces droits :

```
GRANT CREATE USER TO edition;
GRANT CREATE ROLE TO edition;
```

Création des objets :

```
-- Les directeurs de collection
CREATE ROLE Collection;
GRANT CONNECT TO Michel IDENTIFIED BY mdp;
GRANT CONNECT TO Janine IDENTIFIED BY mdp;
GRANT CONNECT TO Jerome IDENTIFIED BY mdp;
GRANT CONNECT TO Pierre IDENTIFIED BY mdp;
GRANT CONNECT TO Patrick IDENTIFIED BY mdp;
GRANT Collection TO Michel,Janine,Jerome,Pierre,Norbert;

-- Le directeur
CREATE ROLE Direction;
GRANT CONNECT TO Robert IDENTIFIED BY mdp;
GRANT Direction TO Robert;

-- Le service de presse
CREATE ROLE Presse;
GRANT CONNECT TO Priscilla IDENTIFIED BY mdp;
GRANT CONNECT TO Vanessa IDENTIFIED BY mdp;
GRANT Presse TO Priscilla,Vanessa;

-- Le service des ventes
CREATE ROLE Vente;
GRANT CONNECT TO Marie IDENTIFIED BY mdp;
GRANT CONNECT TO Gerard IDENTIFIED BY mdp;
GRANT Vente TO Marie, Gerard;

-- Les agents littéraires (Solution provisoire)
CREATE ROLE Agent;
GRANT CONNECT TO Paul IDENTIFIED BY mdp;
GRANT CONNECT TO Sabine IDENTIFIED BY mdp;
...
GRANT Agent TO Paul,Sabine,...;
```

Le dernier groupe pose un problème dans la mesure où ils sont des utilisateurs étrangers qui vont sans doute se connecter par le biais d'Internet. Pour éviter de devoir gérer des utilisateurs par nature changeants, on pourrait employer une autre solution :

```
-- Les agents littéraires (Solution plus réaliste)
GRANT CONNECT TO Agents IDENTIFIED BY mdp;
ALTER TABLE Maison.Agent
ADD InterNetPass VARCHAR2(20);
```

On ajoute ici un champ à la table des Agents afin de mémoriser le mot de passe qu'ils recevront (le gestionnaire du site pourra leur envoyer leur mot de passe ou trouver un moyen fiable de le modifier en ligne). Les différents agents partageront un même utilisateur et leur identification propre se fera par programmation sur le site (en vérifiant le mot de passe stocké dans la table Agent)[1].

```
ALTER TABLE Agents ADD InternetPass VARCHAR(20) NULL;
```

Le groupe presse

C'est probablement le groupe qui pose le moins de problème. Les membres de ce service sont intéressés par toutes les données concernant les publications. On pourrait donc leur donner des droits de lecture sur les tables *Auteur*, *Ecrire*, *Ouvrage*, *Collection* et *Directeurs*. Ce serait plus simple de créer une vue reprenant déjà les données complètes.

[1] On peut envisager aussi d'autres utilisateurs Internet qui consulteront simplement les publications (avec probablement les mêmes droits que le groupe Presse (mais ça n'a pas été demandé dans l'énoncé).

```
CREATE VIEW  Publications AS
SELECT O.idOuvrage, O.Titre, O.Publication AS DatePublication,
NomCollection,
D.Nom AS NomDirecteur,
A.Nom AS NomAuteur, E.idAuteur
FROM Ouvrages O
INNER JOIN Collections USING(idCollection)
INNER JOIN Directeurs D USING(idDirecteur)
INNER JOIN Ecrire E ON O.idOuvrage=E.idOuvrage
INNER JOIN Auteurs A ON A.idAuteur=E.idAuteur;
```

Il reste à donner le droit aux gens du service de presse de lancer cette requête.

```
GRANT SELECT ON Publications TO Presse;
```

La requête provoque l'apparition de plusieurs lignes pour un même ouvrage quand il y a plusieurs auteurs. Les interfaces devront s'en accommoder (quitte à faire disparaître les données répétées).

Les agents

On a vu qu'ils partageaient tous un même utilisateur/internaute. Il n'est donc pas possible, au niveau d'Oracle, de donner des droits pour des utilisateurs définis ailleurs. Le travail devra se faire au niveau du programme gérant le site. Pour obtenir les ventes, on travaillera avec une vue également et on donnera une autorisation de lecture sur cette vue :

```
CREATE VIEW ResumeVentes AS
SELECT Auteur.Nom ANom, Titre,Quantite, Mois.ANNEE, Mois.MOIS,
Agent.Nom AgNom
FROM Ouvrage
INNER JOIN Ecrire USING (idOuvrage)
INNER JOIN Auteur USING (idAUteur)
INNER JOIN Agent USING (idAgent)
INNER JOIN Ventes Using (idOuvrage)
INNER JOIN MOIS Using(idMois)
ORDER BY Auteur.Nom,Titre,Mois.ANNEE, Mois.MOIS;

GRANT SELECT ON ResumeVentes TO Agents;
```

Il suffira d'utiliser une requête comprenant le nom de l'agent dans le programme du site. Par exemple, en PHP :

```
$Agent= recupererAgent();
$SQL = "SELECT * FROM ResumeVentes WHERE AgNom = '$Agent'";
$Result = mysql_query($SQL);
// exploiter le curseur...
```

Si on veut utiliser des utilisateurs d'Oracle, on pourrait envisager la version suivante de la vue[2] :

```
CREATE VIEW ResumeVentes2 AS
SELECT Auteur.Nom ANom,Titre,Quantite, Mois.ANNEE, Mois.MOIS, Agent.Nom AgNom
FROM Ouvrage
INNER JOIN Ecrire USING (idOuvrage)
INNER JOIN Auteur USING (idAUteur)
INNER JOIN Agent USING (idAgent)
INNER JOIN Ventes Using (idOuvrage)
INNER JOIN MOIS Using(idMois)
WHERE Agent.nom = USER
WITH CHECK OPTION;
```

[2]Il semble que la clause `ORDER BY` ne soit pas compatible avec `WITH CHECK OPTION`. Certains prétendent même que `ORDER BY` ne peut pas figurer dans les vues, ce qui est manifestement faux

Cette requête ne fonctionnera que si le champ *Nom* de la table Agent est identique au nom d'utilisateur Oracle. Avec des noms comme 'Jean Deflandre', ce ne sera pas simple. On pourrait ajouter un champ supplémentaire dans la table :

```
ALTER TABLE Maison.Agent
ADD InternetName VARCHAR2(20);
```

Dans cette hypothèse, le champ *InternetPass* ne sert plus à rien. Notons que quelle que soit la technique utilisée, il faudra faire attention à la casse (le champ *InternetName* devrait ne contenir que des majuscules).

Le service des ventes.

Ce service va uniquement modifier les tables Mois et Ventes :

```
GRANT INSERT, UPDATE, SELECT, DELETE ON Ventes to Vente;
GRANT INSERT, UPDATE, SELECT, DELETE ON Mois to Vente;
```

On pourrait aussi leur donner des droits en lecture sur la table ouvrage. Mais il serait plus confortable qu'ils aient une vision des auteurs et des collections (au cas où des titres se ressembleraient). Notons que pour faire la jointure avec la table vente, on doit disposer du champ `Ouvrage.idOuvrage`. Ce sont les droits dont disposent le service de presse avec la vue *Publications*. Le plus simple est de leur donner directement les mêmes droits :

```
GRANT Presse TO Vente;
```

Malheureusement, cette commande suppose qu'on dispose du droit `CREATE ROLE` (ce qui est une des options retenues au départ). On obtiendra le même effet en assignant un droit en lecture sur la vue *Publications* :

```
GRANT SELECT ON Publications TO Vente;
```

Les directeurs de collection

Les directeurs de collection ont besoin de droits étendus sur les tables AUTEURS, AGENT, ECRIRE et OUVRAGES :

```
GRANT INSERT, UPDATE, SELECT, DELETE ON Auteur TO Collection;
GRANT INSERT, UPDATE, SELECT, DELETE ON Agent TO Collection;
GRANT INSERT, UPDATE, SELECT, DELETE ON Ecrire TO Collection;
GRANT INSERT, UPDATE, SELECT, DELETE ON Ouvrage TO Collection;
```

Pour le reste, on leur donnera la possibilité de lire les autres tables

```
GRANT SELECT ON Collection TO Collection;
GRANT SELECT ON Ventes TO Collection;
GRANT SELECT ON Mois TO Collection;
GRANT SELECT ON Directeur TO Collection;
```

Nous reviendrons sur des droits plus fins dans la remarque finale.

La direction générale

C'est le seul groupe qui doit pouvoir accéder en écriture aux collections et aux directeurs :

```
GRANT INSERT, UPDATE, SELECT, DELETE ON Collection TO Direction;
GRANT INSERT, UPDATE, SELECT, DELETE ON Directeur TO Direction;
```

Pour la suite, il suffira de leur donner les mêmes droits que le directeur de collection et les gens du services vente. Ils auront ainsi des droits de regard et de modification sur toute la base de données.

```
GRANT Vente TO Direction;
GRANT Collection TO Direction;
```

Remarque finale

Il reste un point problématique : les directeurs ont le droit de modifier les ouvrages de leurs collègues. Cela pose problème de le leur interdire. Effectivement, ils doivent pouvoir manipuler les auteurs et leur attribuer des livres. Les vues qu'il faudrait écrire pour leur interdire de toucher aux ouvrages (et aux relations écrire) des collections qu'ils ne dirigent pas poseraient plusieurs problèmes :

- étant multi-table, elles ne permettraient pas l'écriture partout
- la clause `WITH CHECK OPTION` doit concerner des données contenues dans la table où se produit l'insertion.

En conséquence, on peut supposer que les différents directeurs de collection ne passent pas leurs journées à se saboter mutuellement. Si on verse dans une telle paranoïa, on pourrait ajouter dans les tables *Ouvrage* et *Ecrire* un champ *Propriétaire* qui permettrait de vérifier qu'on modifie une ligne dont on est propriétaire. On utiliserait un mécanisme semblable à la restriction vue dans la deuxième méthode de gestion des agents (un champ d'identification serait ajouté aux directeurs pour servir à marquer leurs ajouts dans les tables concernées). On créerait alors deux vues *Parano_Ouvrage* et *Parano_Ecrire* sur le modèle suivant :

```
ALTER TABLE Ouvrage
ADD PROPRIETAIRE VARCHAR(15);

CREATE VIEW Parano_Ouvrage AS
SELECT IDOUVRAGE, DCOLLECTION, TITRE, PUBLICATION, PROPRIETAIRE
FROM Ouvrage
WHERE Proprietaire = USER
WITH CHECK OPTION;
```

Pour que cette solution marche, il faudrait encore :

- remplir les champs nouvellement créés avec des données correspondant aux différents directeurs
- remplacer les droits sur les tables *Ouvrage* et *Ecrire* par leurs vues paranoïaques équivalentes
- créer un trigger sur chaque table qui remplirait le champ *Propriétaire* lors de l'insertion

```
CREATE OR REPLACE
TRIGGER ProprietaireOuvrage
BEFORE INSERT ON OUVRAGES
FOR EACH ROW
BEGIN
  :new.proprietaire:= user;
END;
```

Cela n'empêcherait pas un utilisateur de créer un ouvrage dans la collection d'un autre. En conclusion, je proposerais une solution alternative : le directeur de la maison d'édition pourrait informer ses directeurs de collection que toute création ou modification malintentionnée d'une donnée appartenant aux autres directeurs serait immanquablement sanctionnée par une privation de la participation au banquet de fin d'année. Avec une telle menace, aucune erreur ne se produira.