# Gestion des usagers

Pour ce laboratoire, imaginons que pour une nouvelle application, un client veut s’assurer d’avoir une gestion des usagers saines et sécuritaire. Votre travail consiste donc à créer les usagers et leur donner les bons privilèges.

**IMPORTANT**: Afin de pouvoir faire les vérifications (voir étape 7), il est impératif d’utiliser les bons noms des tables, des usagers etc.

**IMPORTANT 2** : Vous devriez conserver une copie de votre script dans un fichier sql (ex : lab-usager.sql)

### Étape 0 : Création des tablespaces

Avec SYS, commencez par créer les deux tablespaces nécessaires à l’application :

CREATE TABLESPACE ACME\_APP DATAFILE 'C:\oraclexe\app\oracle\oradata\XE\acapp01.dbf' SIZE 50M;

CREATE TABLESPACE ACME\_ADMIN DATAFILE 'C:\oraclexe\app\oracle\oradata\XE\acadmin01.dbf' SIZE 50M;

### Étape 1 : CRÉATION DU PROFILE ACME\_PROFIL

Le profil ACME\_PROFIL est très peu restrictif, la seule restriction particulière est qu’un usager ne peut avoir plus de 2 sessions actives en même temps.

Puisque les profils pourraient ne pas être activés dans votre base de données, écrire l’énoncé SQL permettant de les activer :

### Étape 2 : ACME\_CREATEUR

Cette étape consiste à créer un usager qui sera le propriétaire des tables de l’application. Il ne sera pas utilisé directement par le logiciel, mais servira de schéma pour les tables.

Il doit avoir un quota de 30M sur les tablespaces ACME\_APP et ACME\_ADMIN. Son mot de passe est : AAAaaa222. Il doit avoir le profil ACME\_PROFIL. Le tablespace par défaut est ACME\_APP et le tablespace temporaire est TEMP.

Script de création de l’usager ACME\_CREATEUR, et assignation des privilèges nécessaires à la création des tables (voir étape 3). Sous Oracle 11g, cet usager doit également avoir le privilège « CREATE SEQUENCE » et « CREATE TRIGGER ».

### Étape 3 : Ajout des tables

Pour cette étape, connectez-vous avec ACME\_CREATEUR et créez les tables suivantes. Si ça ne fonctionne pas, vérifier qu’il possède les privilèges nécessaires!

CREATE TABLE CATEGORIES(

 ID NUMBER(7) GENERATED BY DEFAULT ON NULL AS IDENTITY,

 NOM VARCHAR2(50) NOT NULL,

 CONSTRAINT PK\_CAT PRIMARY KEY (ID)

) TABLESPACE ACME\_APP;

CREATE TABLE PRODUITS(

 ID NUMBER(8) GENERATED BY DEFAULT ON NULL AS IDENTITY,

 ID\_CATEGORIE NUMBER(8),

 NOM VARCHAR2(150),

 CONSTRAINT PK\_PROD PRIMARY KEY (ID),

 CONSTRAINT FK\_PROD\_CAT FOREIGN KEY (ID\_CATEGORIE) REFERENCES CATEGORIES(ID)

) TABLESPACE ACME\_APP;

REM -- ADMIN tablespace

CREATE TABLE CLIENTS(

 ID NUMBER(8) GENERATED BY DEFAULT ON NULL AS IDENTITY,

 NOM VARCHAR2(50) NOT NULL,

 CONSTRAINT PK\_CLIENTS PRIMARY KEY (ID)

) TABLESPACE ACME\_ADMIN;

CREATE TABLE BACKUPLOGS(

 ID NUMBER(8) GENERATED BY DEFAULT ON NULL AS IDENTITY,

 LOG\_DATE DATE NOT NULL,

 CONSTRAINT PK\_LOGS PRIMARY KEY (ID)

) TABLESPACE ACME\_ADMIN;

À ce point-ci, l’usager ACME\_CREATEUR devrait être désactivé. Écrivez et exécutez l’énoncé SQL qui permet de retirer le droit de connexion :

### Étape 4 : Création des synonymes

Pour éviter d’écrire systématiquement le nom du schéma avant chaque table lors de requêtes SQL, il faut créer des synonymes.

Écrire et exécutez les énoncées SQL permettant de faire des synonymes publics pour les tables créées à l’étape précédente. Pour ce projet, le nom d’un synonyme devrait correspondre au nom de la table. Ceci simplifiera les requêtes : SELECT \* FROM ~~ACME\_CREATEUR.~~CLIENTS;

### Étape 5 : Création du rôle ACME\_ROLE

Avant de créer l’usager qui sera effectivement utilisé par l’application, il faut préparer un rôle. Ce rôle permettra de :

1. Se connecter à Oracle
2. D’insérer, supprimer, modifier ou lire les données des tables de l’application.
3. Il ne doit pas pouvoir supprimer la table ou modifier sa structure!

### Étape 6 : Création de l’usager ACME\_USR

En y ajoutant le rôle ACME\_ROLE, créer l’usager et vérifier qu’il fonctionne bien. Le client pourra donc utiliser cet usager pour son application. Son mot de passe est AAAaaa333 et il doit avoir le profil ACME\_PROFIL. Le tablespace par défaut est ACME\_APP et le tablespace temporaire est TEMP.

### Étape 7 : Script de vérification

Afin de vérifier vos énoncés, exécutez le programme Java que je vous ai fournis (« VerifACME »). Celui-ci peut s’exécuter à partir d’Eclipse.



Il ne vérifie pas tous vos énoncés, mais une bonne partie.

### Étape 8 : Ajouter un test : INSERT/DELETE

(s’il y a du temps)

En modifiant le code de l’application (MngApplication.java et TestDAO.java), ajouter un autre test qui permet de vérifier que l’usager est capable d’insérer et de supprimer dans la table CATEGORIES.

Faites l’insertion d’une ligne avec l’ID 1, par exemple, puis supprimer là.

Cet exercice permet, entre autre, de vous familiariser avec la structure des projets à venir.