

# TP ORACLE - Concurrence

Imen Sebei

## 1 Préparation du TP

On donne le lien suivant pour aider lors du TP notamment lors de la création des tables:

<http://www.techonthenet.com/oracle/index.htm>

On souhaite créer une base de données afin de gérer ses jeux vidéo. A cet effet vous devez tout d'abord créer une table **games** avec les attributs suivants:

1. **codeG**: un code identifiant de manière unique le jeu (entier de longueur maximale 10)
2. **nomG**: le nom du jeu (chaîne de caractères de longueur maximale 40)
3. **genre**: la famille de jeux auquel il se rattache (chaîne de caractères de longueur maximale 40)
4. **annee**: l'année de sortie du jeu (entier de longueur 4)
5. **codeEd**: code de l'éditeur du jeu (entier de longueur maximale 10)
6. **prix**: le prix payé pour le jeu (nombre avec jusqu'à deux décimales)

On précise que **codeG** est une clé primaire de la relation et que **codeEd** est une clé étrangère (il est donc conseillé de créer d'abord la table **Editeur** avant celle-ci, sinon il faut modifier la table avec la commande ALTER).

Créez également un index sur codeEd et un index sur prix.

On souhaite également conserver des informations sur l'éditeur des jeux. Vous devez alors créer une table **Editeur** avec les attributs suivants:

1. **codeEd**: code de l'éditeur (entier de longueur maximale 10)
2. **nomEd**: nom de l'éditeur (chaîne de caractères de longueur maximale 40)
3. **pays**: pays du siège social de la compagnie (chaîne de caractères de longueur maximale 40)
4. **fondation**: année de fondation de la compagnie (entier de longueur 4)

**codeEd** est la clé primaire de la relation alors qu'on veut également la création d'un index sur le nom des pays.

Exécuter ensuite le script sql de chargement des données: script.sql:

Le TP porte sur la concurrence d'accès à une base de données. Pour cela vous allez créer dans un premier temps, à l'aide de votre éditeur préféré les 5 fichiers suivants:

1. **sel.sql**

```
PROMPT '***** Liste des Jeux *****';  
SELECT * FROM Games  
WHERE nomG='Fifa 2005' OR nomG='Warcraft 3' OR nomG='The Sims';
```

2. **majFifa.sql**

```
PROMPT '***** Augmentation du prix du jeu Fifa2005 de 10 euros *****';  
UPDATE Games SET prix = prix + 10  
WHERE nomG = 'Fifa 2005';
```

3. **majWar3.sql**

```
PROMPT '***** Augmentation du prix du jeu Warcraft 3 de 10 euros *****';  
UPDATE Games SET prix = prix + 10  
WHERE nomG = 'Warcraft 3';
```

4. **majSims.sql**

```
PROMPT '***** Augmentation du prix du jeu The Sims de 10 euros *****';  
UPDATE Games SET prix = prix + 10  
WHERE nomG = 'The Sims';
```

5. **annule.sql**

```
PROMPT '***** Retour a la base initiale *****';  
UPDATE Games SET prix = 54 where codeG=1;  
UPDATE Games SET prix = 50 where codeG=2;  
UPDATE Games SET prix = 15 where codeG=4;
```

Pour simuler l'exécution de programmes concurrents, ouvrez 2 fenêtres terminal, placez vous dans le répertoire contenant les fichiers sql précédemment créés et connectez vous sur chacune de ces consoles à la base oracle avec sqlplus. ORACLE considère chaque session comme un utilisateur. Ainsi même si une personne est connectée deux fois à la base, ORACLE considère qu'il s'agit de 2 personnes différentes. Nous avons donc deux utilisateurs notés 1 et 2 réalisant des opérations de lecture, écriture, validation, annulation, en concurrence sur la base. On considère pour ce TP 3 ressources correspondant aux 3 lignes de la base: les jeux de code 1, 2 et 4. On nommera ces ressources dans les histoires fifa, war3 et sims.

Dans les histoires suivantes les différentes opérations sont indexées par le numéro de l'utilisateur et on désigne par:

- *r* une opération de lecture réalisée sur les 3 ressources simultanément des 3 ressources (executez sel.sql)

- *w* une opération d'écriture sur 1 des ressources (exécutez `ma jFifa . sql` pour la ressource fifa par exemple)
- *c* pour la validation d'une transaction (exécutez la commande `commit ;`)
- *a* pour l'annulation d'une transaction (exécutez la commande `rollback ;`)

Pour chacune des histoires suivantes analysez le résultat obtenu et essayez de comprendre ce qui s'est passé **Ecrivez sur une feuille en une ou 2 phrases le résultat observé et tentez de le justifier**. Ensuite annulez toutes les modifications, après chaque histoire, en lançant la commande `sql` suivante: `annule . sql`.

## 2 Niveau "Read-Committed" (défaut)

Il s'agit du mode par défaut d'ORACLE. Après avoir exécuté et expliqué chacune des histoire, essayez d'expliquer quelle est la stratégie par défaut d'ORACLE. Ce type de verrouillage garantit-t'il la sérialisabilité Pourquoi?

1.  $r_1.r_2.w_1[fifa].r_2.w_1[war3].r_2.a_1.r_2$
2.  $r_1.r_2.w_1[fifa].r_2.c_1.r_2.w_1[fifa].r_2.c_1.r_2$
3.  $r_1.r_2.w_1[fifa].w_2[war3].r_1.r_2.c_1.c_2.r_1.r_2$
4.  $r_1.r_2.w_1[fifa].w_2[war3].r_1.r_2.w_1[sims].w_2[sims].r_1.r_2.c_1.c_2.r_1.r_2$
5.  $r_1.r_2.w_1[fifa].w_2[war3].r_1.r_2.w_2[fifa].w_1[war3].r_1.r_2.a_1.a_2.r_1.r_2$

## 3 Autres modes de verrouillage

### 3.1 Avec verrous en lecture

Comme vous avez pu le constater à la section précédente, le verrouillage par défaut d'ORACLE est différent du verrouillage à deux phases présenté en TD. Transformez le code pour la selection en ajoutant la clause `FOR UPDATE` et reprenez toutes les histoires présentes. Expliquez ce que vous observez pour chacune et pour conclure expliquez quelle est la nouvelle stratégie appliquée. N'oubliez pas après chaque histoire de supprimer vos modification en exécutant `annule . sql`.

### 3.2 Avec le niveau "Serializable"

Fixer le mode de gestion des transactions au niveau "Serializable" en exécutant la commande suivante:

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
```

Reprenez toutes les histoires précédentes. Expliquez ce que vous observez pour chacune et pour conclure expliquez quelle est la nouvelle stratégie appliquée. N'oubliez pas après chaque histoire de supprimer vos modification en exécutant `annule . sql`.