

## ***154 - TP Transactions et concurrence***

### ***Partie 2***

#### ***Réservons des places de spectacles pour Quentin et Grégoire<sup>1</sup>***

Pour comprendre et tester la concurrence entre deux transactions, créer les tables suivantes :

```
CREATE TABLE Spectacle
  (nomS VARCHAR(15) NOT NULL PRIMARY KEY,
   Nbplaces INTEGER NOT NULL,
   NbplacesLibres INTEGER NOT NULL,
   tarif DECIMAL(10,2) NOT NULL
  );

CREATE TABLE Client
  (nomC VARCHAR(10) NOT NULL PRIMARY KEY,
   Solde integer NOT NULL
  );

CREATE TABLE Reservation
  (nomC VARCHAR(10) NOT NULL references Client,
   nomS VARCHAR(15) NOT NULL references Spectacle,
   NbplacesReservees INT NOT NULL,
   PRIMARY KEY (nomC, nomS),
  );
```

Pour que la base de données reste cohérente, il faut toujours que pour un spectacle donné, on est la propriété suivante :

$$(1) \text{ sum}(\text{NbplacesReservees}) = (\text{NbPlaces} - \text{NbplacesLibres})$$

#### ***Données***

L'état d'origine de la base de données est donné par les requêtes suivantes :

```
INSERT INTO Client VALUES ('Quentin', 50);
INSERT INTO Client VALUES ('Grégoire ', 50);
INSERT INTO Spectacle VALUES ('Happy Potter', 250, 250, 20);
SET SESSION TRANSACTION ISOLATION LEVEL ... ;
```

**Par exemple :** set session transaction isolation level repeatable read;

---

<sup>1</sup> Inspiré des TP de N. Travers, P. Rigaux et S. Gancarski

Entre chaque test de concurrence, la base de données doit avoir un état cohérent.  
Ainsi, il faudra sans doute remettre à zéro la base en supprimant les tuples des 3 tables et en réinsérant les tuples proposés précédemment (ou d'autres)

## ***Transactions***

Pour chaque transaction, la séquence de requêtes n'est pas interchangeable. On cherche à tester pas à pas deux sessions simultanées ; celle de Quentin et celle de Grégoire.

Vous devez donc vérifier vous-même la cohérence, faire un ROLLBACK si besoin est.

Pour vérifier la condition (1), on vous demande de faire deux requêtes SQL distinctes, l'une pour obtenir le nombre de places réservées (partie gauche de l'égalité), l'autre pour extraire le résultat de la partie droite de l'égalité. Ainsi vous verrez si les deux nombres obtenus sont égaux ou pas.

Le but est de réserver des places : Quentin réserve 2 places pour Harry Potter, et Grégoire en réserve 5. On a donc deux exécutions de la procédure de réservation, l'une à dérouler dans la session 1, l'autre dans la session 2.

On se place en mode sérialisable.

1. Commencez par écrire les requêtes effectuées par chaque transaction :

- a) La réservation de Quentin : T1
- b) La réservation de Grégoire : T2

Pensez à vérifier la cohérence (1) à réaliser dans l'une et l'autre session (dans T1 et T2).

2. Effectuez les deux transactions l'une après l'autre (en série). Quel est l'état de la base à la fin?
3. Réinitialiser vos tables.
4. Testez par la suite, une histoire (une exécution, un entrelacement) ou plus d'une, sous forme de suite d'opérations grâce aux transactions que vous venez de produire. Analysez les situations.
5. Choisissez ensuite une des commandes ci-dessous.

```
set session transaction isolation level read committed;  
set session transaction isolation level repeatable read;
```

Trouver pour chaque niveau d'isolation, un enchainement d'opérations (une histoire) dans 2 sessions différentes (deux terminaux), pour mettre en valeur l'erreur/le cas demandé. Exécuter pas à pas les commandes de sélection et de mise à jour et de vérification de cohérence.

## 1. Mode READ COMMITTED : Lecture validée

Une transaction peut lire seulement des données validées (on ne peut donc plus avoir des lectures sales). Il peut cependant y avoir des lectures non reproductibles: la relecture de données dans la même transaction peut montrer un résultat différent.

## 2. Mode REPEATABLE READ : lecture répétable

Ce niveau empêche les lectures sales et non reproductible. Il peut y avoir cependant le problème lié aux lectures fantômes : pour une transaction, si on relance la même requête à deux instants différents, l'ensemble des lignes résultat aux deux instants peut être différent car de nouveaux enregistrements ont pu être ajoutés/supprimés dans/de la base par une autre transaction, ce qui modifie le résultat. Ceci est différent de la lecture non reproductible dans le sens où les enregistrements lus précédemment n'ont pas changé, mais des enregistrements peuvent être introduits/supprimés du résultat de la requête.

## ***Partie 3 - optionnelle***

Développez une petite application à partir de l'énoncé précédent (ou de votre projet de l'an dernier) en choisissant le niveau d'isolation de votre choix.