

# **NFE 113 – Conception et administration de bases de données**

## **Présentation des TP**

**André Miralles**

## **Sommaire**

TP 15.	Transactions et niveau d'isolation .....	2
--------	--	---

# Travaux pratiques

## TP 15. Transactions et niveau d'isolation

### Objectif :

Le but de ce TP est de s'approprier l'impact du niveau d'isolation de transactions concurrentes.

### A faire :

La vente de billets d'un spectacle conduit à réduire le nombre de places disponibles du nombre de places achetées (nb) d'un spectacle mais aussi d'une part, à accroître de nb le nombre de places réservées par le client et, d'autre part, à réduire son solde bancaire du prix total des places.

Cette opération va être effectuée pour 2 clients, « Philippe » et « Julie », qui souhaitent tous deux assister au Spectacle « Ben hur ». Ces deux achats vont être effectués au cours de 2 transactions concurrentes décrites ci-dessous.

Pour ce faire :

- Créer la base de données TP18\_Spectacles avec le schéma et les données du script fourni ;
- Créer les deux scripts des transactions 1 et 2 décrites dans le tableau ci-dessous ;
- Exécuter ces deux scripts pas à pas en PARALLÈLE pour les trois niveaux d'isolation (READ COMMITTED, REPEATABLE READ, SERIALIZABLE) de PostgreSQL ;

Analyser le contenu des tables à chacune des étapes et expliquer le résultat obtenu. Est-ce qu'il y a une solution ?

Pourquoi le niveau d'isolation READ UNCOMMITTED n'est pas à tester ?

### Fortement conseillé :

Il est fortement conseillé de :

- mettre en commentaire dans les scripts les requêtes de l'autre transaction (cellules vertes du tableau) ;
- mettre aussi des requêtes pour analyser l'évolution du contenu des tables tout le long du processus.

T	Session 1 Transaction 1	Session 2 Transaction 2
	Vente de 5 places « Ben hur » à « Philippe »	Vente de 2 places « Ben hur » à « Julie »
1	BEGIN TRANSACTION...	-- BEGIN TRANSACTION...
2	/* BEGIN TRANSACTION...;	BEGIN TRANSACTION...;
3	UPDATE Spectacle...; • Réduire de 2 nb_places_libres	UPDATE Spectacle...; • Réduire de 2 nb_places_libres
4	UPDATE Client...; • Accroître de 2 nb_places_reservees • Modifier aussi le solde correspondant	UPDATE Client...; • Accroître de 2 nb_places_reservees • Modifier aussi le solde correspondant
5	COMMIT TRANSACTION; */	COMMIT TRANSACTION;
7	UPDATE Spectacle...; • Réduire de 5 nb_places_libres	/* UPDATE Spectacle...; • Réduire de 5 nb_places_libres
8	UPDATE Client...; • Accroître de 5 nb_places_reservees • Modifier aussi le solde correspondant	UPDATE Client...; • Accroître de 5 nb_places_reservees • Modifier aussi le solde correspondant
9	COMMIT TRANSACTION;	COMMIT TRANSACTION; */