

2013



# Mise en œuvre d'une application de gestion de contrôle d'accès

## Analyse Transactionnelle



Groupe 3

Nicolas Jérémy  
Vaschalde Rudy  
Michel Rémi  
Quinard Cyril



## Sommaire

I - Identification des transactions .....	3
A. Les transactions liées à l'administrateur .....	3
a. Gestion des bâtiments.....	3
b. Gestion des groupes de bâtiments.....	4
c. Gestion des jours fériés .....	5
B. Les transactions liées au superviseur .....	5
a. Gestion des personnes et des badges .....	5
b. Gestion des plages d'accès d'un groupe de personnes.....	7
c. Gestion des plages d'accès d'un groupe de bâtiment.....	8
d. Gestions des réservations de salles.....	8
C. Les transactions liées à l'utilisateur .....	9
a. Passage d'un point d'accès.....	9

## I - Identification des transactions

### A. Les transactions liées à l'administrateur

#### a. Gestion des bâtiments

- Ajout d'un bâtiment: **READ COMMITTED**
  - Insérer le bâtiment
- Suppression d'un bâtiment: **SERIALIZABLE**
  - Supprimer toutes les réservations du bâtiment
  - Supprimer toutes les salles du bâtiment
  - Supprimer le bâtiment

Lors de la suppression d'un bâtiment, un trigger est lancé pour supprimer les salles de ce bâtiment ainsi que les réservations correspondantes. Ces opérations sont critiques et nécessitent d'être réalisées en mode SERIALIZABLE (voir exemple ci-dessous).

Exemple en mode READ COMMITTED (**posant problème**):

Transaction 1	Transaction 2
Suppression de toutes les réservations pour les salles du bâtiment	
	Insertion d'une réservation sur une des salles du bâtiment
	Validation
Suppression des salles du bâtiment → problème car T2 a rajouté une référence dans réservation après que T1 les ait supprimées	
Suppression du bâtiment	
Validation	

- Modification : **READ COMMITTED**

(Réaffectation du bâtiment ou changement d'adresse)

- Modifier groupe de bâtiment (ou adresse) dans Batiment

## b. Gestion des groupes de bâtiments

- Ajout: **READ COMMITTED - non DEFERRABLE**
  - Ajouter groupe\_bat
  - Ajouter meta-groupe

Lorsqu'un superviseur souhaite ajouter un groupe de bâtiment, nous allons, au niveau applicatif, réaliser une transaction qui va ajouter le groupe bâtiment dans la table GROUPE\_BAT ainsi que son meta-groupe correspondant dans la table GROUPE. Ces opérations ne présentent pas de réel problème de concurrence, c'est pourquoi nous avons décidé de réaliser cette transaction en mode READ COMMITTED. La cohérence étant toujours garantie au cours de l'exécution de la transaction, il n'est pas nécessaire d'évaluer les contraintes en mode différé.

Transaction T1	Transaction T2
Ajout d'un groupe de bâtiment	
	Ajout d'un groupe de bâtiment
	Ajout du meta-groupe
	Validation
Ajout du meta-groupe	
Validation	

- Suppression: **SERIALIZABLE**
  - Supprimer les réservations
  - Supprimer les autorisations correspondant au groupe de bâtiment
  - Mettre les bâtiments du groupe de bâtiment à NULL
  - Supprimer le meta-groupe
  - Supprimer le groupe\_bat

Transaction 1	Transaction 2
	Suppression du groupe de bâtiment n°2
Modification du bâtiment vers le groupe de bâtiment n°2	
Validation	
	Validation

Au niveau applicatif, forcer le superviseur à réaffecter les bâtiments dont le groupe de bâtiment vient de passer à NULL (Cas d'utilisation Modification d'un bâtiment)

### c. Gestion des jours fériés

- Ajout d'un jour férié : **READ COMMITTED**
  - Insérer nouveau jour férié
- Modification d'un jour férié : **READ COMMITTED**
  - Update jour férié (exemple : jour de l'ascension)
- Suppression : **READ COMMITTED**
  - Supprimer jour férié

## B. Les transactions liées au superviseur

### a. Gestion des personnes et des badges

- Ajouter une personne: **READ COMMITTED**
  - Placer l'évaluation des contraintes en mode DEFERRABLE (Respect de la contrainte: Une personne doit avoir au moins 1 badge actif)
  - Insérer la nouvelle personne
  - Insérer un nouveau badge
  - Lier la nouvelle personne et le nouveau badge

*Scénario*: 2 transactions concurrentes ajoutent une personne avec le même et le même prénom.

Transaction T1	Transaction T2
	Ajout d'une personne ayant pour nom et prénom (nom1, prenom1)
Ajout d'une personne ayant pour nom et prénom (nom1, prenom1)	
	Ajout d'un badge étant en état actif
	Ajout d'une affectation entre personne et badge
Ajout d'un badge étant en état actif	
	Validation
Ajout d'une affectation entre personne et badge	
Validation	

**Remarque** : 2 personnes physiques peuvent avoir les mêmes nom/prénom. La clé porte sur l'identifiant de la personne

- Modifier un badge : **READ COMMITTED**
  - Modifier l'état d'un badge (Avec cohérence de la contrainte suivante : une personne doit toujours avoir un badge actif)
- Supprimer une personne: **SERIALIZABLE**
  - Placer l'évaluation des contraintes en mode DEFERRABLE (Respect de la contrainte: Une personne doit avoir au moins 1 badge actif)
  - Rechercher le badge affecté à la personne supprimée
  - Changer l'état du badge Actif → Inactif
  - Supprimer toutes les références de la personne (Affectation, Membre, Accès)
  - Supprimer la personne dans PERSONNE

*Scénario:* Une transaction tente de supprimer une personne alors qu'une autre transaction tente d'affecter cette personne à un groupe

Exemple en mode **READ COMMITTED** (posant problème):

Transaction T1	Transaction T2
	Sélection des personnes à supprimer dans la table affectation
	Mise à jour de la table badge mettant le badge de la personne en inactif
	Suppression des références de la personne dans les tables affectation, membre et accès
Affectation de la personne à un groupe → problème car T1 affecte une personne à un groupe ayant été supprimé par T2	
VALIDATION	
	Suppression de la personne
	VALIDATION

## b. Gestion des plages d'accès d'un groupe de personnes

- Ajouter une plage horaire: **READ COMMITTED**
  - Insérer une nouvelle plage horaire
- Ajouter une période d'accès: **READ COMMITTED**
  - Insérer une nouvelle plage d'accès
  - lier plage horaire avec groupe
- Modifier une plage horaire: **READ COMMITTED**
  - Modifier une plage horaire
- Modifier une période d'accès: **READ COMMITTED**
  - Modifier une période d'accès
- Supprimer une plage horaire: **SERIALIZABLE**
  - Sélectionner plage horaire à supprimer
  - Supprimer la plage horaire

Exemple en mode READ COMMITTED (**posant problème**):

Transaction T1	Transaction T2
Sélection d'une plage horaire à supprimer	
	Sélection d'une plage horaire à référencer pour l'ajout d'une nouvelle période d'accès
Suppression de la plage horaire sélectionnée par T2	
	Ajout de la nouvelle période d'accès
VALIDATION	
	VALIDATION

- Supprimer une période d'accès: **SERIALIZABLE**
  - Sélectionner période d'accès à supprimer
  - Supprimer la période d'accès

Exemple en mode READ COMMITTED (**ne fonctionne pas**):

Transaction T1	Transaction T2
	Sélection d'une période d'accès à référencer pour l'ajout d'une nouvelle 'plage'
Suppression de la période d'accès sélectionnée par T2	
	Ajout de la nouvelle plage
VALIDATION	
	VALIDATION

### c. Gestion des plages d'accès d'un groupe de bâtiment

Les cas d'utilisations de gestion des plages d'accès des groupes de bâtiment sont similaires à la gestion des plages d'accès des groupes de personnes (Adaptation des tables).

### d. Gestions des réservations de salles

- Ajouter une réservation: **SERIALIZABLE**

Exemple en mode READ COMMITTED (**posant problème**):

Transaction T1	Transaction T2
	Sélection des salles disponibles
Sélection des salles disponibles	
	Le groupe de personne N°1 réserve la salle 101 pour une période X
Le groupe de personne N°2 réserve la salle 101 pour une période X	
	VALIDATION
VALIDATION	

Contrainte de clé primaire ok car les ID\_GROUPEPERS sont différents mais nous avons un problème de cohérence car présence d'une réservation de la même salle sur une même période par 2 groupes différents.



- Modifier une réservation: **SERIALIZABLE**

Exemple en mode READ COMMITTED (posant problème):

Transaction T1	Transaction T2
	Sélection des salles disponibles
Sélection des salles disponibles	
	Le groupe de personnes N°1 modifie la réservation et prend la salle 101 pour une période X
Le groupe de personne N°2 modifie la réservation et prend la salle 101 pour une période X	
	VALIDATION
VALIDATION	

Contrainte de clé primaire ok car les ID\_GROUPEPERS sont différents mais nous avons un problème de cohérence car présence d'une même salle pour une même période pour 2 groupes différents.

**Modification future possible : utilisation d'un "SELECT FOR UPDATE" avec un mode d'isolation en READ COMMITTED. Utilisé pour bloquer la ligne sur laquelle porte la modification.**

- Supprimer une réservation: **READ COMMITTED**
  - Sélectionner la réservation à supprimer
  - Supprimer la réservation

## C. Les transactions liées à l'utilisateur

### a. Passage d'un point d'accès

- Ajouter un passage : **READ COMMITTED**
  - A chaque passage → Insertion dans la table ACCES
  - Si entrée → Insertion dans ENTREE
  - Si sortie → Insertion dans SORTIE
  - Si problème → Insertion dans ALARME

Pas de problème de concurrence car un nouveau numéro d'accès est généré à chaque accès.

- Suppression des passages : **READ COMMITTED**
  - Pour chaque ENTREE/SORTIE → Lister date\_entree/date\_sortie >= 48 h par rapport à la date actuelle (Sélection table entree/sortie)
  - Pour chaque tuple de cette liste → Faire une nouvelle liste avec tuples n'ayant pas générés d'alarmes
  - Pour chaque tuple de cette nouvelle liste → supprimer les occurrences (Delete sur table Entree/Sortie et Acces)