Série 2

La série d'exercice ci-dessous n'est qu'une suite de proposition de manipulations et ne doivent pas nécessairement être suivie à la lettre. Il est plus important de comprendre les manipulations effectuées.

Exercice 1

Ci –dessous quelques exemples illustrant le comportement du système avec la gestion des transactions.

Faire les points 1.1 et 1.2 sans l'ordinateur dans un premier temps. Les refaire seul à l'aide de son ordinateur afin de vérifier ses résultats.

1.1. Opérations liées :

0	Valeurs ?
1	UPDATE camps SET prix = prix + 2 ;
2	Valeurs ?
3	UPDATE camps SET prix = prix + 2 ;
4	Valeurs ?
5	ROLLBACK ;
6	Valeurs ?

0	Valeurs ?
1	UPDATE camps SET prix = prix + 2 ;
2	Valeurs ?
3	UPDATE camps SET prix = prix + 2 ;
4	Valeurs ?
5	COMMIT ;
6	Valeurs ?

0	Valeurs ?
1	UPDATE camps SET prix = prix + 2 ;
2	Valeurs ?
3	UPDATE camps SET prix = prix + 2 ;
4	Valeurs ?
5	COMMIT ;
6	Valeurs ?
7	UPDATE camps SET prix = prix + 2 ;
8	Valeurs ?
9	ROLLBACK ;
10	Valeurs ?

1.2. Schéma

Testez la création d'objets dans une transaction (LDD) à l'aide de l'instruction CREATE SCHEMA.

Exercice 2

Conseil : faire ces exercices sans l'ordinateur dans un premier temps. Le refaire seul avec n connexions si nécessaire (2 fois SQL*Plus par exemple). Enfin, faire cette série de manipulation avec un collègue.

1. Verrous :

1.1 cas A :

	Connexion 1	Connexion 2
1	`UPDATE camps` `SET prix = prix + 2;`	
2		`UPDATE camps` `SET prix = prix - 2;`
3	`ROLLBACK ;`	
4		`ROLLBACK ;`

1.2 cas B:

	Connexion 1	Connexion 2

1	`UPDATE camps`
	`SET prix = prix + 2`
	`WHERE numero <5;`	

2		`UPDATE camps`
		`SET prix = prix – 2`
		`WHERE numero >=5 ;`

3	`ROLLBACK ;`	
4		`ROLLBACK ;`

1.3 cas C:

	Connection 1	Connection 2
1	`UPDATE camps` `SET prix = prix + 2;`	

2		`UPDATE classes`
		`SET obligation = 'O'`
		`WHERE niveau < 3 ;`

3	`UPDATE classes`
	`SET obligation = 'O'`
	`WHERE niveau < 3 ;`	

4		`UPDATE camps` `SET prix = prix + 2;`
5	`ROLLBACK ;`	
6		`ROLLBACK ;`

1.4 cas D:

	Connection 1	Connection 2
1	`SELECT prix FROM camps ;`	
2		`UPDATE camps` `SET prix = prix + 2;`
3	`SELECT prix FROM camps` `FOR UPDATE;`	
4		`ROLLBACK ;`
5	`ROLLBACK ;`	

Vous pouvez modifier la commande 3 en testant les clauses **WAIT** et **NOWAIT**.(Consulter la doc en ligne)

2. Isolation des transactions

2.1 cas A :

	Connection 1	Connection 2
1	**UPDATE camps** **SET prix = prix + 2;**	
2	valeurs ?	valeurs ?
3	**COMMIT ;**	
4	valeurs ?	valeurs ?
5		**UPDATE camps** **SET prix = prix – 2;**
6	valeurs ?	valeurs ?
7		**ROLLBACK ;**
8	valeurs ?	valeurs ?

2.2 Cas B :

	Connection 1	Connection 2
1	**SET TRANSACTION READ ONLY ;**	
2	valeurs ?	valeurs ?
3		**UPDATE camps** **SET prix = prix – 2;**
4	valeurs ?	valeurs ?
5		**COMMIT ;**
6	valeurs ?	valeurs ?
7	**COMMIT ;**	
8	valeurs ?	valeurs ?

Avant de « commiter », la transaction 1 peut-elle modifier la table camp ?

2.3 *Cas D*:

A évaluer en modifiant le niveau d'isolation des transactions

	Connection 1	Connection 2
1	**SET TRANSACTION ISOLATION LEVEL …;**	
2	valeurs ?	Valeurs ?
3		**UPDATE camps SET prix = prix – 2;**
4	valeurs ?	Valeurs ?
5		**COMMIT ;**
6	valeurs ?	Valeurs ?
7	**COMMIT ;**	
8	valeurs ?	Valeurs ?

3 Verrous explicites

Tester la compatibilité de différents types de verrous Oracle en utilisant les verrous explicites de l'instruction **LOCK TABLE IN … MODE ;**

4 Contraintes et des transactions

Modifier la contrainte de clé étrangère de inscrit sur camp, afin qu'il soit possible de la différer en fin de transaction. Tester une violation temporaire de contrainte (interne à la transaction). Peut-on lire des valeurs erronées depuis une autre session?