IUT2 Université Grenoble Alpes
Département INFO

SQL dans les langages de programmation

TD Transactions

Il s'agit ici de gérer le système de réservation de places dans une salle de spectacles qui propose des places pour des concerts à différentes dates.
La base de données de ce système de réservation comporte (dans une version très simplifiée !!!) les deux tables suivantes :

RESERVATION(<u>numresa</u>, dateresa, nom, prénom, titreconcert, dateconcert, nbpers)

Une réservation comporte un numéro de réservation, une date de réservation, est effectuée par une personne dont on conserve le nom et le prénom, concerne un concert identifié par son titre et sa date, et est faite pour un certain nombre de personnes.

PLACE(<u>numplace, titreconcert, dateconcert</u>, #numresa).

numresa référence RESERVATION(numresa)

Une place a un numéro, concerne un concert donné à une date donnée, et est éventuellement associée à une réservation.
Lorsqu'une personne effectue une réservation pour plusieurs personnes, une place différente est attribuée à chaque personne. Ainsi, pour une réservation de 2 personnes au concert de Shaka Ponk, 2 places sont associées à la réservation.

1) Donner les requêtes SQL qui permettent de :
   a. Créer la réservation n° 57890 effectuée par Mr Joachim Akli pour 2 personnes pour le concert de Shaka Ponk du 27 novembre 2024.
   b. Modifier les places n° 36 et 37 disponibles pour ce concert en les associant à la réservation n° 57890.

2) Donner la requête SQL permettant de savoir les places restantes pour le concert de Shaka Ponk du 27/11/2024.
   On suppose que cette requête renvoie une seule place numéro 100000

3) Mr Akli Joachim et Mme Cortese Célia exécutent cette requête en même temps. quels problèmes peuvent se produire dans les cas suivants ?

1er cas :

Mr Akli	Mme Cortese
Requête question 2	
	Requête question 2

2ème cas :

Mr Akli	Mr Cortese
Requête question 2	
	Requête question 2
INSERT INTO RESERVATION VALUES (57891, now(), ‘Akli’, ‘Joachim’, ‘Shaka Ponk’, ’27/11/2024’, 1);	
UPDATE PLACE SET numresa = 57891 WHERE numplace = 100000 ;	
	Requête question 2

3ème cas :

Mr Akli	Mr Cortese
Requête question 2	
	Requête question 2
INSERT INTO RESERVATION VALUES (57891, now(), ‘Akli’, ‘Joachim’, ‘Shaka Ponk’, ’27/11/2024’, 1);	
UPDATE PLACE SET numresa = 57891 WHERE numplace = 100000 ;	
	INSERT INTO RESERVATION VALUES (57892, now(), ‘Cortese, ‘Paul’, ‘Shaka Ponk’, ’27/11/2024’, 1);
	UPDATE PLACE SET numresa = 57892 WHERE numplace = 100000 ;

4ème cas :

Mr Akli	Mr Cortese
Requête question 2	
INSERT INTO RESERVATION VALUES (57891, now(), ‘Akli’, ‘Joachim’, ‘Shaka Ponk’, ’27/11/2024’, 1);	
Begin	
UPDATE PLACE SET numresa = 57891 WHERE numplace = 100000 ;	
	Requête question 2
ROLLBACK;	

Pour résoudre ces problèmes, on va placer ces instructions dans des transactions. Ceci permettra au système d'assurer les propriétés ACID, et en particulier de poser des verrous sur les données lorsque l'utilisateur souhaitera accéder à des données.

4) Ecrire la transaction correspondante à la question 1. Quels sont les avantages de placer ces instructions dans une transaction ?

5) Pour assurer la propriété d'Isolation et éviter les problèmes de concurrence, le SGBD gère un système de verrous : verrouiller les données avant chaque lecture et les déverrouiller après chaque écriture. Les exécutions suivantes conformes à cette décision résolvent-elles les problèmes ?

Exécution 1 :

Mr Akli	Mr Cortese
BEGIN;	
	BEGIN ;
Verrouiller(PLACE)	
Requête question 2	
Verrouiller(RESERVATION)	
	Verrouiller(PLACE)
INSERT INTO RESERVATION VALUES (57891, now(), 'Akli', 'Joachim', 'Shaka Ponk', '27/11/2024', 1);	
Libérer(RESERVATION)	
UPDATE PLACE SET numresa = 57891 WHERE numplace = 100000 ;	
Libérer(PLACE)	
COMMIT ;	
	Requête question 2
	Libérer (PLACE)
	COMMIT ;

Exécution 2 :

Mr Akli	Mr Cortese
BEGIN ;	
	BEGIN ;
Verrouiller(PLACE)	
Requête question 2	
	Verrouiller(PLACE)
Verrouiller(RESERVATION)	
INSERT INTO RESERVATION VALUES (57891, now(), ‘Akli’, ‘Joachim’, ‘Shaka Ponk, ’27/11/2024’, 1);	
Libérer(RESERVATION) ;	
UPDATE PLACE SET numresa = 57891 WHERE numplace = 100000 ;	
Libérer(PLACE);	
ROLLBACK ;	
	Requête question 2
	Libérer (PLACE)
	COMMIT ;

Exécution 3 :

Mr Akli	Mr Cortese
BEGIN ;	
	BEGIN ;
	Verrouiller(PLACE) ;
	Requête question 2
	Libérer(PLACE) ;
Verrouiller(RESERVATION) ;	
INSERT INTO RESERVATION VALUES (57891, now(), ‘Akli’, ‘Joachim’, ‘Shaka Ponk’, ’27/11/2024’, 1);	
Libérer(RESERVATION)	
Verrouiller(PLACE)	
Requête question 2	

	Verrouiller(PLACE) ;
UPDATE PLACE SET numresa = 57891 WHERE numplace = 100000 ;	
Libérer(PLACE) ;	
COMMIT;	Requête question 2
	Libérer (PLACE) ;
	COMMIT ;

6) Dans la question précédente, que remarquez-vous dans les exécutions qui ne posent plus de problème ?

7) Quel problème peut se produire dans le cas suivant ?

Mr Akli	Mr Cortese
	BEGIN ;
BEGIN ;	
	Verrouiller(PLACE) ;
	Requête question 2
Verrouiller(RESERVATION) ;	
UPDATE RESERVATION SET nbpers = 3 WHERE numresa = 57891;	
	Verrouiller(RESERVATION) ;
	En attente
Verrouiller(PLACE) ;	
En attente	

8) Donner une exécution respectant le protocole de verrouillage à 2 phases sans interblocage pour la question 7.

9) Donner une exécution respectant le protocole de verrouillage à 2 phases avec des verrous sans interblocage des 2 transactions ci-dessous. Votre exécution doit proposer un entrelacement des instructions des 2 transactions.

BEGIN ; SELECT sum(nbpers) FROM RESERVATION ; SELECT count(numplace) FROM PLACE WHERE numplace is not NULL; COMMIT;	BEGIN ; SELECT * FROM PLACE WHERE numplace is NULL; UPDATE PLACE SET numresa = 2187 WHERE numplace = 111 AND titreconcert = 'Shaka Ponk' and dateconcert = '27/11/2024'; COMMIT;

10) Mr. Akli souhaite maintenant modifier sa réservation en ajoutant une 3ème personne. En supposant qu'il restait encore une place, la transaction permettant cette modification est donnée ci-dessous.

BEGIN ;
UPDATE RESERVATION SET nbpers = 3 WHERE numresa = 57890 ;
UPDATE PLACE SET numresa = 57890 WHERE numplace = 56 AND titreconcert = 'Shaka Ponk' AND dateconcert = '27/11/2024';
COMMIT;

Malheureusement, une panne logicielle se produit. Que doit faire le SGBD dans les cas suivants pour assurer la propriété de Durabilité :

- 1er cas : si un point de contrôle a été fait entre les 2 UPDATE et que la panne se produit avant le COMMIT ?
- 2ème cas : si un point de contrôle a été fait entre les 2 UPDATE et que la panne se produit après le COMMIT ?
- 3ème cas : si un point de contrôle a été fait avant le BEGIN et que la panne se produit avant le COMMIT ?
- 4ème cas : si un point de contrôle a été fait avant le BEGIN et que la panne se produit après le COMMIT ?