Databasesystemer, Forår 2005

Rasmus Pagh

April 26, 2005

Øvelser d. 28/4

1. **Read committed.** Suppose that user A has created relations `Accounts (id INT, balance INT)` and `NegativeAccounts(id INT, balance INT)`, both initially empty. Then user A and B issue the below statements, in this order.

A	B

	`INSERT INTO A.Accounts VALUES (42,1000);`
	`INSERT INTO A.Accounts VALUES (43,-100);`
	`COMMIT;`

`INSERT INTO NegativeAccounts`
`(SELECT * FROM Accounts`
`WHERE balance<0);`	

	`UPDATE A.Accounts SET balance=balance-2000`
	`WHERE id=42;`
	`COMMIT;`

`DELETE FROM Accounts WHERE balance<0;` `COMMIT;`	

User B creates two accounts, and later makes a withdrawal from one of them. User A tries to move all accounts with negative balance to `NegativeAccounts`. Suppose user A runs at isolation level `READ COMMITTED` (Oracle default). What will be the result of the above concurrent transactions? Test your answer in Oracle (You can use two connections to the database instead of two different users). What alternative solution would you suggest? *(Side remark: Our guest lecturer, Martin Jensen, said that nobody uses other isolation levels that READ COMMITTED. This may be true in systems where performance is crucial, but as the above example shows one must then be very careful with the way transactions are programmed.)*

2. **Serializability.** Suppose that user A has an empty relation `Primes (p INT)`. Then user A and B issue the below statements, in this order.

A	B
`SELECT * FROM Primes;`	
	`SELECT * FROM A.Primes;`
`INSERT INTO Primes VALUES (2);`	
	`INSERT INTO A.Primes VALUES (3);`
`SELECT * FROM Primes;`	
	`SELECT * FROM A.Primes;`
`DELETE FROM Primes WHERE p=3;`	
	`DELETE FROM Primes WHERE p=2;`
`SELECT * FROM Primes;`	
	`SELECT * FROM A.Primes;`
`COMMIT;`	
`SELECT * FROM Primes;`	
	`SELECT * FROM A.Primes;`
	`COMMIT;`
`SELECT * FROM Primes;`	
	`SELECT * FROM A.Primes;`

   - Explain what may be seen by user A and user B for each of the SQL isolation levels `READ COMMITTED` and `SERIALIZABLE`.
   - Try it out in Oracle at isolation levels `READ COMMITTED` and `SERIALIZABLE`. The transactions may be carried out by two different users, or by two different connections to the database.
   - What state would the database be in after each of the possible serial schedules for the transactions?
   - What happens if the transactions run at different isolation levels?

3. **Deadlocks.** Create your own deadlock. You may use the example given at the lecture as inspiration.