

# Database Systemer, ITU, Forår 2006

Rasmus Pagh

April 6, 2006

## Exercises on April 6

### 1. Problem 4 from exam in Databasesystemer, June 2005 (15 %)

The relation `Seats(seatID, class, reserved)` is used to handle seat reservations in an airplane. It contains one tuple for each seat, and the attribute `reserved` is 0 or 1 depending on whether a seat is free or booked. 15 minutes before departure, the booking of business class seats is closed, by transferring any free business class seats to “economy plus” customers. The below transaction (written in Oracle SQL) makes the transfer of seats, using the two relations `FreeBusinessSeats(seatID, reserved)` and `Upgrades(seatID)` to store intermediate results.

1. `DELETE FROM FreeBusinessSeats;`
2. `DELETE FROM Upgrades;`
3. `INSERT INTO FreeBusinessSeats (SELECT seatID, reserved FROM Seats  
WHERE class='business' AND reserved=0);`
4. `INSERT INTO Upgrades (SELECT *  
FROM (SELECT seatID FROM Seats  
WHERE class='economy plus' AND reserved=1)  
WHERE rownum<=(SELECT COUNT(*) FROM FreeBusinessSeats));`
5. `UPDATE FreeBusinessSeats SET reserved=1  
WHERE rownum<=(SELECT COUNT(*) FROM Upgrades);`
6. `UPDATE Seats SET reserved=0 WHERE seatID IN (SELECT * FROM Upgrades);`
7. `UPDATE Seats SET reserved=1  
WHERE (seatID,1) IN (SELECT seatID, reserved FROM FreebusinessSeats);`

**Explanation:** The first two lines delete any old intermediate results. Line 3 inserts the free business class seats in the relation `FreeBusinessSeats`. Line 4 chooses reservations from “economy plus” that are to be upgraded. The number of upgrades is kept below the number of free seats by use of the `rownum` variable, which returns the row number of the current row in the relation. Line 5 marks the right number of free seats in `FreeBusinessSeats` as reserved. In line 6 and 7, the information on the new reservations are transferred to the `Seats` relation.

a) Assume that the above transaction runs at SQL isolation level `READ COMMITTED`. Argue that if, at the same time, a reservation is made for a business class seat (i.e. a transaction that changes a value of `reserved` from 0 to 1), there may be a double booking, that is, the number of reserved seats is smaller than the number of passengers.

Because of the above problem, it seems like a good idea to consider a higher isolation level. We consider SQLs `REPEATABLE READ` og `SERIALIZABLE`, as well as “snapshot isolation” described in the article *A Critique of ANSI SQL Isolation Levels*.

b) Consider for each of the three mentioned isolation levels whether a double booking may occur. Argue for your answer.

- In Oracle, create two relations that have foreign key references to each other's primary key. Make the foreign key constraints DEFERRABLE. Now experiment with inserting new tuples in the two relations. Try committing: 1) After having inserted tuples in the two relations that refer to each other. 2) After having inserted several tuples in one relation, that refer to no tuple in the other relation (i.e., in a state where the referential integrity constraint is violated).
- Suppose that user A has an empty relation Primes(p INT). Then users A and B issue the below statements, in this order. The start and end of transactions are indicated by horizontal lines.

A	B
SELECT * FROM Primes;	
INSERT INTO Primes VALUES (2);	SELECT * FROM A.Primes;
SELECT * FROM Primes;	INSERT INTO A.Primes VALUES (3);
DELETE FROM Primes WHERE p=3;	SELECT * FROM A.Primes;
SELECT * FROM Primes;	DELETE FROM Primes WHERE p=2;
COMMIT;	SELECT * FROM A.Primes;
-----	
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 three SQL isolation levels READ COMMITTED, REPEATABLE READ, 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?
- This exercise are only for those who have some acquaintance with Java, and the subject of this exercise is not a part of the syllabus of this course. It is an opportunity to see how Java can be used to manipulate with your database in Oracle.

The exercise is based on a small program that makes it possible to view the tuples in a database and update the tuples in the database. The exercise consist in three things:

- Download the program from <http://www.itu.dk/people/pagh/DBS06/javaprogram.zip>.
- Edit line 17 in the file dbTest.java: replace <your db user name> with your database user name and replace <your db password> with your database password
- Compile dbTest.java (javac dbTest.java)
- Run dbTest. This can be done by typing: `java -cp ".;ojdbc14.jar" dbTest` or running the file `run.bat`
- Run the program and try to manipulate on the existing database through the program
- Implement deletion in file `dbTest.java`
- Implement insertion in file `dbTest.java`