Hand-out

Advanced database technology

March 13, 2006

1 ADBT exam, June 2005: Transaction processing (20%)

In this problem we consider the following two transactions, consisting of 1 and 7 SQL statements, respectively.

Transaction 1:

```
1. UPDATE Seats SET reserved=1 WHERE seatID=42 AND reserved=0;
```

Transaction 2:

```
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);
```

Suppose these transactions are run in a system using two-phase locking, with the exception that for transactions running at isolation level `READ COMMITTED` all read locks are released after every statement, and the necessary read locks are obtained immediately before each statement. Exclusive locks follow two-phase locking for all transactions. The system is designed to minimize the duration of locks.

a) State, for each of the relations accessed by Transaction 2, when shared and exclusive locks are obtained and released during the execution, if it is run at isolation level `SERIALIZABLE`.

b) Suppose Transaction 2 runs at isolation level `READ COMMITTED`. Argue that it is possible that `Seats` changes, because Transaction 1 runs, between statement 3 and statement 6.

Next we consider concurrency control using timestamps as described in GUW 18.8.

c) Explain what happens if the scheduler attempts to run Transaction 1 between statements 3 and 6 of Transaction 2:

- If the timestamp of Transaction 1 is smallest.
- If the timestamp of Transaction 2 is smallest.