UNIVERSITY OF EDINBURGH

FACULTY OF SCIENCE AND ENGINEERING

MASTER OF SCIENCE IN COMPUTER SCIENCE

DATABASE SYSTEMS

Thursday 19 April 2001

09:30 to 11:00

Examiners:

I. Watson (External)

R.N. Ibbett (Chairman)

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

1. (a) A database system must ensure the *atomicity property* of transactions. Explain what is meant by the atomicity property and how it is enforced in a database system. [*5 marks*]

   (b) Explain the difference between a *serial* and a *conflict-serializable* set of transactions. [*4 marks*]

   (c) Consider the three transactions T_1, T_2, and T_3, and the two schedules S_a and S_b shown below:

T_1: $r_1(x); r_1(z); w_1(x);$
T_2: $r_2(z); r_2(y); w_2(z); w_2(y);$
T_3: $r_3(x); r_3(y); w_3(y);$

S_a: $r_1(x); r_2(z); r_1(z); r_3(x); r_3(y); w_1(x); w_3(y); r_2(y); w_2(z); w_2(y);$
S_b: $r_1(x); r_2(z); r_3(x); r_1(z); r_2(y); r_3(y); w_1(x); w_2(z); w_3(y); w_2(y);$

   (i) List all of the conflicts and draw the serialization graphs for S_a and S_b (show your working clearly). [*12 marks*]

   (ii) State whether S_a and S_b are serializable. If the schedule is serializable, write down the equivalent serial schedule. [*4 marks*]

2. (a) Let R be a relation, and X, Y be arbitrary subsets of the attributes of R. Define what it means for X to be *functionally dependent* on Y. *[4 marks]*

(b) Briefly define the following italicised terms:

(i) A *trivial* functional dependency. *[2 marks]*

(ii) The *closure* of a set of functional dependencies. *[2 marks]*

(iii) A *lossless decomposition* of a relation. *[2 marks]*

(c) Let $R = \{A,\ B,\ C,\ D,\ E,\ F,\ G,\ H,\ I,\ J\}$ be a relational schema. Consider the following set S of functional dependencies over R:

$$S = \{AB \to C,\ A \to DE,\ B \to F,\ F \to GH,\ D \to IJ\}$$

For each of the following, draw the appropriate dependency graphs to justify your solution.

(i) What is the key for R? *[5 marks]*

(ii) Decompose R into second normal form (2NF) relations. *[6 marks]*

(iii) Further decompose R into third normal form (3NF) relations. *[4 marks]*

3. (a) The two-phase locking (2PL) protocol can prevent transactions from interfering with one another.

   (i) Describe the *basic* 2PL protocol. [*3 marks*]

   (ii) What recovery problems might occur if the 2PL mechanism is not *Strict* 2PL. [*2 marks*]

   (iii) Why is *Conservative* 2PL rarely used in practice? [*2 marks*]

(b) In multi-version concurrency control, several versions of each database object may be maintained.

   (i) Explain the three different kinds of locks in two-version concurrency control, stating when each kind of lock may be acquired. [*6 marks*]

   (ii) Why may there be no extra memory overhead in multi-version concurrency control? [*1 mark*]

(c) Consider the following concurrency control scheme, which combines features of both *optimistic* and *conservative* protocols. The first time that the transaction is executed, optimistic concurrency control is used, but if the transaction is aborted and subsequently retried, a conservative concurrency control scheme based on locking is used to control the transaction.

   (i) Describe the three phases of the Optimistic 2PL algorithm, detailing how conflicts are detected and the action that is taken. [*6 marks*]

   (ii) Outline the main advantage of optimistic concurrency control for distributed databases. [*1 mark*]

   (iii) Explain how the scheme described above might have performance advantages in comparison with conventional optimistic concurrency control. [*4 marks*]