| PAPER CODE NO. | EXAMINER : **Dr C Ghidini** |
|---|---|
| COMP502 | DEPARTMENT : Computer Science Tel. No. 46787 |

THE UNIVERSITY
*of* LIVERPOOL

# SUMMER 2002 EXAMINATIONS

Master of Science : Year 1
Master of Science : Year 2

## ADVANCED DATABASE MANAGEMENT

**TIME ALLOWED : Two Hours**

---

**INSTRUCTIONS TO CANDIDATES**

Attempt ***all*** questions in Section A
Attempt ***two*** questions from Section B

If you attempt to answer more than the required number of questions (in any section), the marks awarded for the excess questions will be discarded (starting with your lowest mark).

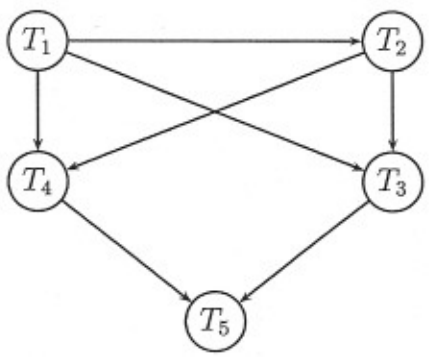
THE UNIVERSITY
*of* LIVERPOOL

# Section A.

**Attempt ALL questions from this section**

A1. describe the four (so-called ACID) properties of transactions. (6 marks)

A2. Consider the following precedence graph




Is the corresponding schedule conflict-serializable? Explain your answer.

If it is conflict-serializable give a corresponding serial schedule. (4 marks)

A3. What is timestamping? (4 marks)

A4. Consider the two following schedules:

**Schedule 1**

| $T_1$ | $T_2$ | $T_3$ |
|---|---|---|
| read(B) | | |
| read(A) | | |
| write(B) | | |
| | read(B) | |
| | read(C) | |
| | write(C) | |
| | | read(C) |
| | | read(A) |
| | | write(A) |

**Schedule 2**

| $T_1$ | $T_2$ | $T_3$ | $T_4$ |
|---|---|---|---|
| read(A) | | | |
| | read(A) | | |
| read(B) | | | |
| | read(B) | | |
| | | read(A) | |
| | | | read(A) |
| write(A) | | | |
| | write(B) | | |

For each of these schedules:

- draw the precedence graph for the schedule;
- determine whether the schedule is conflict-serializable. If it is conflict-serializable, list all the equivalent serial schedules.

(6 marks)

A5. Consider a banking system where A and B are two accounts accessed by transactions $T_1$ and $T_2$. Transaction $T_1$ transfers £50 from account B to account A. Transaction $T_2$ displays the total amount of money in accounts A and B. In order to control the interaction among the concurrent execution of $T_1$ and $T_2$ lock requests have been added to $T_1$ and $T_2$ as follows:

| $T_1$ : | $T_2$ : |
|---|---|
| write_lock(B) | read_lock(A) |
| read(B) | read(A) |
| B:= B - 50 | unlock(A) |
| write(B) | read_lock(B) |
| unlock(B) | read(B) |
| write_lock(A) | unlock(B) |
| read(A) | display(A+B) |
| A:=A+50 | |
| write(A) | |
| unlock(A) | |

Suppose that the values of accounts A and B are £100 and £200, respectively. If these transactions are executed serially, either in the order $T_1, T_2$ or in the order $T_2, T_1$, then transaction $T_2$ will display the value £300.

If these transactions are executed concurrently, are the locking operations enough to ensure that transaction $T_2$ always displays the correct value £300? Justify your answer.

If locking operations are not correct modify $T_1$ and $T_2$ to ensure serializability. (6 marks)

A6. What is a distributed database ? (4 marks)

A7. Consider a distributed database with the following relation:

`EmployeeInfo(employeeID, name, designation, salary).`

where the primary key is underlined.

For privacy and security reasons, this relation is fragmented into two fragments:

$F_1 = \Pi_{\texttt{employeeID,salary}}(\texttt{EmployeeInfo})$

$F_2 = \Pi_{\texttt{name,designation}}(\texttt{EmployeeInfo})$

which are stored at different sites of the distributed database.

Is the fragmentation operation correct? Justify your answer. If the fragmentation operation is not correct, define an alternative fragmentation of the relation `EmployeeInfo` which satisfies the initial requirement of storing data about salary, and data about name and designation in different sites. (6 marks)

A8. Object-relational capabilities in SQL are centralized around the UDT, or user defined type. Describe the two main contexts in which UDTs can be used. Give an example of UDT usage in each of the two contexts. (4 marks)

# Section B.

**Attempt TWO questions from this section. Each question carries 20 marks. Credit will be given for the best two answers only.**

B1. a. Describe what is generally known as "the lost update" problem. Give an example of two concurrent transactions that generate a lost update. (10 marks)

b. Describe how to prevent the lost update problem in the example that you have shown above using the two-phase locking (2PL) protocol. (10 marks)

B2. a. How does the concept of an object in the object-oriented database model differ from the concept of an entity in the entity-relationship model? (10 marks)

b. Describe the use of views as a security mechanism. Illustrate your answer with examples and/or diagrams if needed. (5 marks)

c. Consider a database with the following relation:

`Manages(employee-name, manager-name)`

Assume that we wish to allow a user `manager` the `insert` privilege on the table `Manages`, and allow `manager` to grant this privilege to others. Write an SQL statement that performs this task. (5 marks)

B3. a. Referential integrity is a form of integrity constraint that ensures that all non-null values that appear in a certain column A of a table R must also appear in the column B of a table S. Table S is called the parent table, and table R is called the child table.

When a user attempts to update a row in the parent table, and there are matching rows in the child tables, SQL supports four possible outcomes. Describe the four possible outcomes. (8 marks)

b. Triggers can be used in Oracle to enforce enterprise constraints. They are often called *event–condition–action* rules.

Describe what the event, the condition and the action are. (5 marks)

c. Consider a database with the following relations:

`Manages(employee-name, manager-name)`

`Employee(employee-name, street, city)`

Write an Oracle trigger to carry out the following action: after insert on the table `Employee`, check if the employee name is present in the table `Manages` and insert it if not (with a null manager name). (7 marks)