UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2001

BEng Honours Degree in Computing Part III
MEng Honours Degree in Information Systems Engineering Part IV
BSc Honours Degree in Mathematics and Computer Science Part III
MSci Honours Degree in Mathematics and Computer Science Part III
MSc in Advanced Computing
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute
This paper is also taken for the relevant examinations for the Associateship of the Royal College of Science

PAPER C312=I4.4

ADVANCED DATABASES

Wednesday 9 May 2001, 10:00
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators not required

1a In the context of distributed databases explain what is meant by a *minterm predicate* and formally define what is meant by a *fragment.*

The process of fragmentation should satisfy the three conditions of *completeness*, *reconstruction* and *disjointness.* Explain what is meant by these.

b The following three global relations have been defined:

STAFF (StaffNo, StaffName, Dept)

STUDENT (StudentNo, StudentName, Address, TutorNo)

COURSE (StaffNo, CourseNo, StudentNo, StartDate)

The attribute TutorNo gives the member of staff (identified by StaffNo) acting as Tutor to the given student. A student would have no more than one Tutor.

The relation COURSE gives details of the courses given by members of staff and the students attending these courses.

The relation STAFF has a horizontal fragmentation based on StaffNo.

The relation STUDENT has a horizontal fragmentation based on StudentNo.

Two main applications are run using these relations – one requires details about staff and the courses they give, the other requires details about students and the courses they attend.

Discuss the fragmentation process that would be best suited to the relation COURSE and show how the three conditions given in part a would be satisfied.

c The following relation is maintained by an organisation:

EMPLOYEE (EmpNo, Dept, Salary)

The relation is fragmented on Salary, with Salary ≤ 100,000 assigned to the first site, 100,000 < Salary ≤ 200,000 assigned to the second site, and so on. No Salary can ever exceed 1,000,000.

The fragment with Salary ≤ 100,000 is frequently accessed and is therefore replicated at every site. No other fragments are replicated.

The following operation is to be carried out:

Give all employees a 10 percent increase in Salary, except that for those currently earning ≤ 100,000 the increase should not take their Salary above 100,000.

Describe the processes involved (eg locks obtained, operations performed) if this update operation is to be satisfactorily completed, and indicate two approaches to ensuring that the overall integrity of the information is maintained.

The three parts carry, respectively, 25%, 30%, and 45% of the marks.

2a Describe the Basic Timestamp Ordering scheme for scheduling concurrent transactions and discuss how it can be implemented.

b Tranactions T1, T2, T3 are to be run concurrently. The following table gives details of the proposed interleaving of read/write operations and the time when each such operation is to be scheduled.

Time	T1	T2	T3
1	Read (E)		
2		Read (B)	
3		Read (C)	
4			Read (D)
5			Write (E)
6		Write (E)	
7	Write (E)		

Determine whether the operations can be executed in this order if concurrency is to be controlled using

i) two-phase locking;
ii) timestamp ordering.

c Explain why the timestamp ordering scheme can be considered to be too restrictive (i.e. it could abort transactions unnecessarily). Adapt the scheme to avoid such unnecessary aborts.

d Explain, with the aid of an example, how the Basic Timestamp Ordering scheme may produce a serialisable history which is not a strict history.

Discuss how it could be ensured that all histories produced by the Basic Timestamp Ordering scheme are also strict.

The four parts carry, respectively, 25%, 20%, 25% and 30% of the marks.

3a In the *algebra of qualified relations* one of the rules that applies is:

$$[R : q_R]\ \mathbf{JN}_F\ [S : q_S] \Rightarrow [R\ \mathbf{JN}_F\ S : q_R \text{ AND } q_S \text{ AND } F]$$

Prove this rule stating clearly any further rules used in the proof. (Note that join (JN) can be derived from selection and Cartesian product).

b A small Building Society has 20 branch offices in London (LON), 10 in Manchester (MCR) and 10 in Newcastle (NCL). Each branch office is identified by a unique two character branch number (B#).

A customer, identified by a unique six character customer number (C#), would maintain an account at one of the branches but could pay in amounts to this account at any one of the branches. A customer may be assumed to have just one account with the Society.

The Building Society has set up three relations:

BRANCH – giving details of individual branch offices, with attributes B#, City, Manager. (Size of tuple = 15 bytes.)

CUSTOMER – giving details of individual customers, with attributes C#, Name, Address, B# (the branch at which the account in maintained). (Size of tuple = 30 bytes.)

TRANSACTION – with attributes C #, B #, Amount, indicating the amount paid into his account by a given customer and the branch at which it was paid in. (Size of tuple = 16 bytes.)

The following query has been formulated and is to be run against these relations:

Find the customer number and customer name for each customer who has paid in an amount at a branch in MCR.

i) Express the given query in relational algebra and draw the corresponding operator tree.

ii) The information recorded by the Building Society is to be distributed over 5 sites:

Site 1 - TRANSACTION details relating to customers maintaining accounts in LON (approx. 30,000 tuples; 10,000 of these relate to amounts paid in at branches in MCR).

Site 2 - TRANSACTION details relating to customers maintaining accounts in MCR or NCL (approx. 25,000 tuples; 8,000 of these relate to amounts paid in at branches in MCR).

Site 3 - BRANCH details for LON branch offices and CUSTOMER details for those maintaining accounts in LON (approx. 9500 CUSTOMER tuples).

Site 4 - BRANCH details for MCR branch offices and CUSTOMER details for those maintaining accounts in MCR (approx. 6500 CUSTOMER tuples).

Site 5 - BRANCH details for NCL branch offices and CUSTOMER details for those maintaining accounts in NCL (approx. 5000 CUSTOMER tuples).

Explain what is meant by a *canonical* expression and incorporate the relevant canonical expressions in the operator tree produced in part (i).

Use equivalence transformations to optimise the tree, identifying and explaining any transformations used.

iii) The information to be retrieved by the Society is required at Site 3. Suggest *two* possible strategies for the retrieval of this information and compare the estimated delay involved in retrieving the information.

[Assume that customer names are no longer than 10 characters.]

The four parts a,b(i),b(ii),b(iii) carry, respectively, 25%, 10%, 25% and 40% of the marks.

4a Explain what is meant by *transaction atomicity.*

Discuss how atomicity can be achieved in a distributed environment using the two-phase commit (2PC) protocol with linear coordinator. Give details of the sequence of messages sent by the coordinator and the participants. Where the overall decision is to abort the transaction, participating sites should be informed of this with a minimum of delay.

b Explain what is meant by a *blocking protocol* and discuss the circumstances under which blocking will occur in the 2PC protocol.

In the three-phase commit protocol, four of the places at which a process (the coordinator or a participant) could be waiting for a message are:

- a participant is waiting for a PREPARE message
- the coordinator is waiting for a vote (COMMIT/ABORT)
- a participant is waiting for a PRE-COMMIT or ABORT message
- the coordinator is waiting for acknowledgement following the PRE-COMMIT or ABORT message.

Should a process fail to receive a message within a given time interval, a *timeout* is said to have occurred, and appropriate action will need to be taken by that process.

For each of the places mentioned above, discuss the actions that need to be taken by the relevant process in the event of a timeout occurring.

c Explain what is meant by a *history* in the context of concurrency control.

A history H is said to be *recoverable* if, whenever transaction T_i reads from transaction T_j in history H and $C_i \in H$, then $C_j < C_i$.

Give similar formal definitions for

i) a *cascadeless* history

ii) a *strict* history

Prove that recoverability (RC), cascadelessness (CD), and strictness (ST) are increasingly restrictive properties.

i.e. $ST \subset CD \subset RC$.

The three parts carry, respectively, 20%, 40% and 40% of the marks

