UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1998

BEng Honours Degree in Computing Part III
MEng Honours Degrees in Computing Part IV
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 Degree 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
Diploma of Membership of Imperial College
Associateship of the Royal College of Science

PAPER 3.11 / I4.4

ADVANCED DATABASES
Friday, May 8th 1998, 10.00 - 12.00

Answer THREE questions

For admin. only: paper contains 4 questions

1 a Distinguish between a *homogeneous* and a *heterogeneous* distributed database system. Discuss some of the problems associated with obtaining a unified view of the data under a heterogeneous system.

b Explain what is meant by a *minterm predicate.*

Discuss the conditions under which a set of predicates would be considered to be:

i) minimal

ii) complete

and explain why it is desirable for predicates to satisfy both of these properties.

c Distinguish between *location*, *replication* and *fragmentation transparency.*

The following two relations have been defined:

E_1 (Emp#, Ename, Salary, Site)

E_2 (Emp#, Proj#, Job, Start-date).

E_1 gives the Employee number, name, salary and site (place of work) for each employee in the organisation.

E_2 gives for each employee the project number(s) on which he is working and, for each such project, his job title and start-date on the project.

Relations E_1 and E_2 are updated on a regular basis.

i) Describe a possible hybrid fragmentation process that could be applied to relation E_1 .

ii) Discuss the problems associated with implementing fragmentation transparency for relation E_1, after hybrid fragmentation has been applied to it.

iii) Give an example of a query based on relations E_1 and E_2 for which the use of *derived fragmentation* is necessary and show how the *semi-join* operator may be used to obtain this information.

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

2 This question relates to the two phase commit protocol (2PC) and the three phase commit protocol (3PC).

a Under 2PC it is a normal requirement for any local decision to be recorded in the local log.

i) In addition to the local decision, indicate any further information that is likely to be recorded in the log.

ii) Explain why a COMMIT vote/decision *must* first be recorded in the local log before being communicated to the coordinator/participants, whilst recording an ABORT vote/ decision can be delayed until after the information has been communicated to the coordinator/participants.

iii) It will be necessary, every so often, to delete log records so as to reclaim space taken up by outdated information.

Discuss the factors that need to be considered before the log records of a transaction can be safely deleted.

b In a commit protocol, the *uncertainty period* for a site is the period between the moment the site votes COMMIT and the moment it has received sufficient information to know what the final decision is. During this period, the site is said to be *uncertain.*

A protocol is said to satisfy the *non-blocking property* if, whenever any operational site is uncertain, then no site (whether operational or failed) can have decided to COMMIT.

i) Discuss the circumstances under which 2PC does not satisfy the non-blocking property given above.

ii) In 2PC and 3PC a site failure is discovered whenever a site fails to receive an expected message within a given time interval (a 'timeout' has occurred).

In 3PC a timeout could occur at one of the following stages:

- 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
- a participant is waiting for the final COMMIT decision.

For each of these stages, discuss the actions that need to be taken in the event of a timeout occurring and explain clearly how the non-blocking property will be satisfied.

Turn Over

3 a i) Distinguish between *conflict equivalent* and *view equivalent* histories.

Using the concept of conflict equivalence and assuming that there is no replication of data, explain what is meant by a *serialisable* history and a *serialisation graph.*

Prove that if two histories are conflict equivalent, then their serialisation graphs are identical. Give an example to show that the converse of this statement is not necessarily true.

ii) By choosing an appropriate example show that a view serialisable history may not necessarily be conflict serialisable.

A *blind write* is defined as a write on some data item x by a transaction that did not previously read x.

Assuming that histories do not include blind writes show, by choosing suitable examples, that a history will only be view serialisable if it is conflict serialisable.

b i) Describe a *centralised* approach to deadlock detection in a distributed environment.

ii) Describe, in outline, a *fully distributed* approach to deadlock detection and discuss any advantages this approach may have over the centralised approach.

iii) Consider a distributed deadlock detection scheme in which the sites are organised in a hierarchy. Each site checks for deadlocks local to the site and for global deadlocks that involve its descendent sites in the hierarchy.

Compare the relative merits of this scheme with those of the centralised scheme and the fully distributed scheme.

The two parts carry, respectively, approximately 55% and 45% of the marks.

4a i) Define the relational algebra *semi-join* operator in terms of the *project* and *join* operators.

ii) In the *algebra of qualified relations* one of the rules that applies is:

$$[R : q_R]\ \mathbf{SJ}_F\ [S : q_s] \Rightarrow [R\mathbf{SJ}_F S\ :\ q_R \text{ and } q_s \text{ and } F]$$

where SJ is the semi-join operator.

Give a proof of the above rule explaining any further rules used in the proof.

b The *equi-join* $R\mathbf{JN}_{A=B}S$ between two relations R and S where A is an attribute of R and B is an attribute of S can be expressed as:

$$R\mathbf{JN}_{A=B}S = S\mathbf{JN}_{A=B}(R\mathbf{SJ}_{A=B}\mathbf{PJ}_B\ S)$$

where JN, SJ, PJ are the operators *join*, *semi-join* and *project* respectively.

Analyse under which conditions it will be cheaper (in terms of transmission costs) to execute the join using an approach based on the right-hand side of the equality rather than executing the join directly. Illustrate your answer by reference to the following relations:

R	A	C	D
	1	"	"
	2	"	"
	3	"	"
	4	"	"

S	B	E	F
	2	"	"
	3	"	"
	3	"	"
	4	"	"
	5	"	"

The following additional information is available:

Size (A) = Size (B) = 4 bytes
Size (C) = 6 bytes
Size (D) = 6 bytes
Size (E) = 6 bytes
Size (F) = 15 bytes

where 'Size' indicates the width of the relevant field.

The two parts carry, respectively approximately 40% and 60% of the marks.

End of Paper