ECM2419

(with Answers)

UNIVERSITY OF EXETER

COLLEGE OF ENGINEERING, MATHEMATICS AND PHYSICAL SCIENCES

COMPUTER SCIENCE

Examination, January 2022

Database Theory and Design

Module Leader: Dr Zeliang Wang

Duration: TWO HOURS

Answer ALL questions.

The marks for this module are calculated from 60% of the percentage mark for this paper plus 40% of the percentage mark for associated coursework.

This is a CLOSED BOOK examination.

Question 1

(a) A large organisation has adopted a database system rather than a file based system. Briefly explain five advantages of such decision.

(10 marks)

It is common to summarize the advantages of adopting the database approach rather than a file based approach as follows:**[2]** each.

- Data is readily shared between applications, thus eliminating duplication and the problems of maintaining consistency between duplicated values.
- New requests, or one-of-a-kind requests, can more easily be implemented, because the logical interface with the DBMS is simpler than a set of physical interfaces.
- The applications programs are independent of the stored data. If the storage format changes, there is no need to alter the applications program, since they communicate with the DBMS in logical rather than physical terms.
- It can be argued that a single database management system for an integrated database allows for better management of data, since it is effectively in one place under the control of one set of people namely those who implement the database.
- The integration, or sharing of data between applications, puts sophisticated programming within reach of all users of the database.

Marks may also be awarded for other advantages not listed here.

(b) Consider the following relations that form part of a database, where the primary keys are underlined. Note that the relation *Employee* contains employee details; the relation *Department* contains department details and the attribute *mgrEmpID* identifies the employee who is the manager of the department. There is only one manager for each department; the relation *Project* contains details of the projects in each department, and no two departments can run the same project; the relation *WorksOn* contains details of the hours worked by employees on each project.

Employee (empID, fName, lName, address, DOB, sex, position, deptNo)
Department (deptNo, deptName, mgrEmpID)
Project (projNo, projName, deptNo)

WorksOn (empID, projNo, hoursWorked)

Write the following queries in relational algebra.

(i) Find all the details of employees who are male.

(2 marks)

$\sigma_{\text{sex='M'}}(\text{Employee})$

(ii) List the names and addresses of all employees who are Managers.

(4 marks)

$\pi_{\text{fName,lName,address}}(\sigma_{\text{position='Manager'}}(\text{Employee}))$

correct SELECT **[2]**, correct PROJECT **[2]**

(iii) Produce a list of the names and addresses of all employees who work for the 'HR' department.

(5 marks)

$\pi_{\text{fName,lName,address}}(\sigma_{\text{deptName='HR'}}(\text{Department} \bowtie \text{Employee}))$

correct SELECT **[3]**, correct PROJECT **[2]**

(iv) Produce a list of the names of all employees who work on the 'STF' project.

(5 marks)

$\pi_{\text{fName,lName}}(\sigma_{\text{projName='STF'}}(\text{Project} \bowtie \text{WorksOn} \bowtie \text{Employee}))$

correct SELECT **[3]**, correct PROJECT **[2]**

(c) Provide two reasons that Strict Two-Phase Locking (2PL) is popularly used in many database systems.

(4 marks)

Strict 2PL is popular for many reasons. One reason is that it ensures only 'safe' interleaving of transactions so that transactions are recoverable, avoid cascading aborts, etc **[2]**. Another reason is that strict 2PL is very

simple and easy to implement. The lock manager only needs to provide a lookup for exclusive locks and an atomic locking mechanism (such as with a semaphore) **[2]**.

Marks may also be awarded for other reasons not mentioned here.

(d) If the number of *read-only transactions* is increased, what would happen to the database system throughput?

(3 marks)

If the number of read-only transaction is increased, the database system throughput will also increase **[1]** since read-only transactions require only shared locks **[1]**. So we are able to have more concurrency and execute more transactions in a given time **[1]**.

(e) What are the two types of database updating schemes in log-based recovery? Explain the difference between them.

(7 marks)

The two types of database updating schemes are deferred update **[1]** and immediate update **[1]**. Using deferred updates, writes are done initially to the log only **[1]** and the log records are used to perform actual updates to the database **[1]**. If the system fails, it examines the log to determine which transactions it needs to redo, but there is no need to undo any writes **[1]**. Using immediate updates, an update may be made to the database itself any time after a log record is written **[1]**. The log can be used to undo and redo transactions in the event of failure **[1]**.

(Total 40 marks)

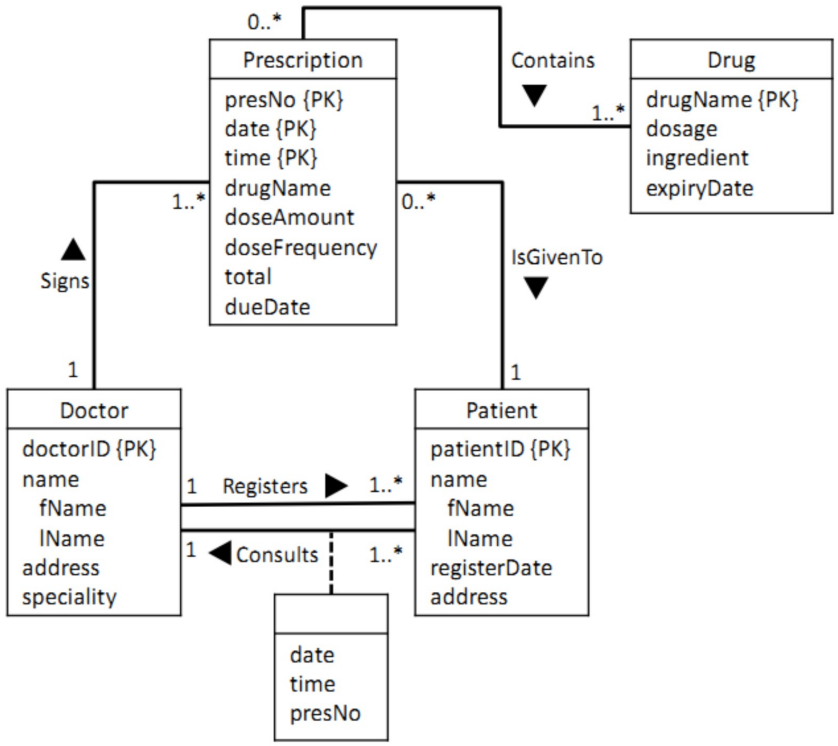
Question 2

(a) A GP practice in Exeter has a number of doctors, with each of whom a number of patients are registered. The practice wishes to record on a computer details of drugs prescribed for its patients, and also the dates and times of consultations between the patients and particular doctors in the practice. Prescriptions, which may be for one or more drugs, are given either in consultations, or as a repeat of prescriptions previously given and signed in a consultation. All repeat prescriptions, which are dated, must be signed by a doctor in the practice, not necessarily the same one who signed the original prescription. Prescriptions give not only the name of the drug, but also the amount and frequency of the dose, the total quantity to be supplied, and the date, if any, when a repeat prescription would be due.

Draw an *entity-relationship diagram* (using UML notation) for the patient management system as described above. Identify all entities, relationships, attributes (you may add some basic information for each entity), primary keys and multiplicity constraints.

(18 marks)

Example solution of ER diagram:



Marks:

- Correctly identify entities with their corresponding attributes **[4]**
- Correctly identify primary keys in the entities **[4]**
- Relationship 'Registers', with multiplicity **[2]**
- Relationship 'Consults', with multiplicity and attributes **[2]**
- Relationship 'Contains', with multiplicity **[2]**
- Relationship 'Signs', with multiplicity **[2]**
- Relationship 'IsGivenTo', with multiplicity **[2]**

Marks may also be awarded for other reasonable answers.

(b) The following table shows some sample data about mortgage application appointments from a bank. Each mortgage applicant is given an appointment at a specific date and time with a mortgage advisor at a particular branch of the bank. Assume that for each day of appointments, a mortgage advisor is allocated to a specific branch for that day. Each applicant is registered at

only one branch, and may have more than one appointment on a given day.

date	time	staffNo	staffName	applicantNo	applicantName	branchNo
19/06/18	9:00	S022	Paul	A030	Josh	B11
19/06/18	11:00	S022	Paul	A035	Julian	B11
19/06/18	9:00	S028	John	A039	Fabian	B05
21/06/18	13:00	S028	John	A039	Fabian	B05
21/06/18	15:30	S010	Emma	A035	Julian	B11
22/06/18	17:00	S010	Emma	A050	Liz	B03

(i) The above table is susceptible to update anomalies. Provide an example of deletion anomaly.

(2 marks)

An example of a deletion anomaly is if we delete the details of the advisor called 'John', we also lose the appointment details of the applicant called 'Fabian'.

(ii) Identify a *candidate key* for the table.

(2 marks)

(date, time, staffNo)

(iii) Transform the table into relations in the *2nd Normal Form* and then *3rd Normal Form* by using relational schemas (with primary keys underlined) rather than detailed tables.

(8 marks)

The 2nd Normal Form: **[4]**

Appointment(date, time, staffNo, applicantNo, applicantName)

Location(staffNo, date, branchNo)

Advisor(staffNo, staffName)

The 3rd Normal Form: **[4]**

Appointment(date, time, staffNo, applicantNo)

Location(staffNo, date, branchNo)

Advisor(staffNo, staffName)

Applicant(applicantNo, applicantName)

(Total 30 marks)

Question 3

Given the following database in which the primary keys are underlined. The relation *Catalog* lists the prices charged for *Parts* by the *Suppliers*. Write expressions in SQL for each of the following queries.

Suppliers (supplierID, supplierName, address)
Parts (partID, partName, color)
Catalog (supplierID, partID, cost)

(a) List the names of suppliers who supply a black part that costs less than 200.

```
SELECT DISTINCT S.supplierName
FROM Suppliers S, Catalog C, Parts P
WHERE S.supplierID = C.supplierID AND C.partID = P.partID AND
P.color = 'Black' AND C.cost < 200
```

correct SELECT **[1]**, correct FROM **[1]**, correct WHERE **[4]**

(6 marks)

(b) Retrieve the names of parts supplied by *Smart Widgets* and no one else.

```
SELECT P.partName
FROM Parts P, Catalog C, Suppliers S
WHERE P.partID = C.partID AND C.supplierID = S.supplierID
AND S.supplierName = 'Smart Widgets'
AND NOT EXISTS ( SELECT *
FROM Catalog C1, Suppliers S1
WHERE P.partID = C1.partID AND C1.supplierID = S1.supplierID
AND
S1.supplierName <> 'Smart Widgets')
```

correct SELECT **[1]**, correct FROM **[1]**, correct WHERE **[2]**, correct sub-query **[2]**

(6 marks)

(c) For each part, retrieve the name of the supplier who charges the most for that part.

```
SELECT P.partID, S.supplierName
FROM Parts P, Suppliers S, Catalog C
WHERE C.partID = P.partID
AND C.supplierID = S.supplierID
AND C.cost = (SELECT MAX (C1.cost)
FROM Catalog C1
WHERE C1.partID = P.partID)
```

correct SELECT **[1]**, correct FROM **[2]**, correct WHERE **[3]**

(6 marks)

(d) Retrieve the IDs of suppliers who supply only black parts.

```
SELECT DISTINCT S.supplierID
FROM Suppliers S
WHERE S.supplierID NOT IN ( SELECT C1.supplierID
FROM Parts P, Catalog C1
WHERE C1.partID = P.partID AND P.color <> 'Black' )
```

correct SELECT **[1]**, correct FROM **[1]**, correct WHERE **[2]**, correct sub-query **[2]**

(6 marks)

(e) Retrieve the IDs of suppliers who supply at least a blue part and at least a yellow part.

```
SELECT DISTINCT C.supplierID
FROM Catalog C, Parts P
WHERE C.partID = P.partID AND P.color = 'Blue'
INTERSECT
SELECT DISTINCT C1.supplierID
FROM Catalog C1, Parts P1
WHERE C1.partID = P1.partID AND P1.color = 'Yellow'
```

correct SELECT **[1]**, correct FROM **[1]**, correct WHERE **[2]**, correct INTERSECT **[2]**

(6 marks)

(Total 30 marks)