# The University of Nottingham

SCHOOL OF COMPUTER SCIENCE
AND INFORMATION TECHNOLOGY

A LEVEL 1 MODULE, SPRING SEMESTER 2006-2007

**DATABASE SYSTEMS**

Time allowed TWO hours

---

*Candidates must NOT start writing their answers until told to do so*

***Answer THREE out of FIVE questions***

*Marks available for sections of questions are shown in brackets in the right-hand margin.*

*No calculators are permitted in this examination.*

*Dictionaries are not allowed with one exception. Those whose first language is not English may use a dictionary to translate between that language and English provided that neither language is the subject of this examination.*

*No electronic devices capable of storing and retrieving text, including electronic dictionaries, may be used.*

***DO NOT turn examination paper over until instructed to do so***

**Question 1**

(a) Given the relations `Patient`, `Treatment` and `Doctor` below:

| Patient | |
|---|---|
| pID | pName |
| 111 | Tom |
| 222 | John |
| 333 | Sue |
| 444 | Anne |

| Treatment | | |
|---|---|---|
| pID | dID | Date |
| 111 | 1 | 19 − 01 − 07 |
| 111 | 1 | 12 − 03 − 07 |
| 111 | 2 | 12 − 03 − 07 |
| 222 | 1 | 26 − 01 − 07 |
| 333 | 2 | 5 − 03 − 07 |

| Doctor | |
|---|---|
| dID | dName |
| 1 | Jones |
| 2 | Smith |

give the results of the following relational algebra expressions:

(i) $\pi_{\texttt{pID}}(\texttt{Patient})$ (1)

(ii) $\pi_{\texttt{pID}}(\texttt{Patient}) \cup \pi_{\texttt{pID}}(\texttt{Treatment})$ (1)

(iii) $\pi_{\texttt{pID}}(\texttt{Patient}) - \pi_{\texttt{pID}}(\texttt{Treatment})$ (1)

(iv) $\pi_{\texttt{pID}}(\texttt{Patient}) \cap \pi_{\texttt{pID}}(\texttt{Treatment})$ (1)

(v) $\sigma_{(\texttt{pName=Tom})\texttt{OR}(\texttt{pName=Sue})}(\texttt{Patient})$ (1)

(vi) $\sigma_{(\texttt{pName=Tom})\texttt{AND}(\texttt{dID=2})}(\texttt{Patient} \times \texttt{Treatment})$ (5)

(b) Write a relational algebra expression which computes the dates of Tom's appointments with doctor Jones. (5)

(c) Translate the following SQL query in relational algebra: (5)

```
SELECT pName, dName
FROM Patient, Treatment, Doctor
WHERE Patient.pID = Treatment.pID AND
      Treatment.dID=Doctor.dID
```

(d) Which of the two queries below is more computationally expensive and why:

(i) $\sigma_{\texttt{Date=19-01-07}}(\texttt{Patient} \times \texttt{Treatment})$

(ii) $\texttt{Patient} \times \sigma_{\texttt{Date=19-01-07}}(\texttt{Treatment})$

(5)

**Question 2**

(a) Imagine that you need to design a database for an on-line wine shop. The shop sells bottles of wine. Each bottle has a price and a volume (for example, 0.75 l or 0.375 l). Each wine has a name, type (red, white or rose), year of production, grape variety, and a wine-maker. Each wine-maker comes from a particular region and country (for example, Bordeaux, France). The same wine-maker can make several different kinds of wine, and there can be several wine-makers in the same region. The same wine name can be used by several wine-makers, but the (wine name, wine-maker) combination is unique. You can also assume that wine-maker's name is unique.

List entities, relationships, and attributes. State for each relationship whether it is one-to-one, one-to-many, or many-to-many. (6)

(b) Describe the tables for the wine-shop database (names of tables, column names, types of values for each attribute). (5)

(c) Give the two conditions which must be satisfied for a set of attributes to be a candidate key of a relation. (2)

(d) What are the candidate keys in the tables for the wine shop? (3)

(e) What are the foreign keys in the tables for the wine shop? (3)

(f) Define the terms entity integrity and referential integrity. (2)

(g) Explain, using an example, how sequences in Oracle can be used to generate numerical ID values when adding values to a table. (4)

**Question 3**

This question refers to the following tables. `sID` is the primary key for `Staff`, `pID` is the primary key for `Project`, and {`sID`, `pID`} is the primary key for `Allocation`. `sID` and `pID` in `Allocation` are foreign keys to `sID` in `Staff` and `pID` in `Project` respectively. `Manager` in `Staff` is a foreign key to `sID` in `Staff`.

| Staff | | |
|---|---|---|
| sID | Name | Manager |
| 1 | Alice | |
| 2 | Bill | 1 |
| 3 | Chris | 1 |
| 4 | Dave | 2 |

| Allocation | | |
|---|---|---|
| sID | pID | Time |
| 1 | 1 | 12 |
| 1 | 2 | 10 |
| 2 | 1 | 20 |
| 3 | 1 | 36 |
| 2 | 2 | 20 |
| 4 | 2 | 52 |

| Project | | |
|---|---|---|
| pID | Funder | Budget |
| 1 | EPSRC | 400,000 |
| 2 | BBSRC | 300,000 |
| 3 | EPSRC | 40,000 |

Note the following points:

- `Manager` in `Staff` gives the `sID` of the person who manages the staff member described in each row. If this entry is NULL (blank) then that person has no manager.
- Each row in `Allocation` represents the number of weeks (`Time`) the staff member (`sID`) works on the project with the given `pID`.
- The values given in the tables above are samples only, and your answers should be applicable even if data is altered, added to, or removed from the tables.

Write SQL queries to do the following:

(a) Find a list of names of staff members. (1)

(b) Find a list of funders which fund projects with budgets over 100,000. (2)

(c) Find a list of names of staff members who are allocated to a project funded by EPSRC. (4)

(d) Find a list of staff names and total time in weeks they are committed (across all projects). (5)

(e) Add 1 week to the allocated time of all staff working on the projects funded by EPSRC. (4)

(f) Add a new staff member called Edith with sID 5 who is managed by Bill and works on the project with `pID` 2 for 10 weeks. Note: you cannot use Bill's sID directly, only his name. You need to update both the `Staff` and the `Allocation` tables. (4)

(g) Create a new table called `Deliverable` which should have the following columns: (5)

- `dID` an integer value which is the primary key
- `pID` an integer value which is a foreign key to the `Project` table
- `Description` which is a string up to 250 characters long
- `Pages`, an integer value which is optional.

**Question 4**

(a) Define functional dependency. (2)

(b) Define 2 NF. (2)

(c) Define 3 NF. (2)

(d) Define BCNF. (2)

(e) What are insertion, deletion and update anomalies? Give examples. (3)

(f) Consider a relation Marks with schema (StudentID, StudentName, ModuleID, ModuleName, Mark) which records marks of students for modules which they are taking. StudentID determines StudentName and ModuleID determines ModuleName. The only candidate key in this relation is (StudentID, ModuleID). Is this relation in 2NF? Explain your answer. (5)

(g) Decompose the relation above to BCNF. Explain why the resulting tables are in BCNF. (5)

(h) What is lossless decomposition? Is the decomposition you did in answer to the question above lossless and why? (4)

**Question 5**

(a) Describe the information that is held in a *transaction log* and show how this information can be used to recover from system failures. (5)

(b) Explain the difference between a *serial* and *serialisable* transaction schedule. (2)

(c) What is the difference between a read lock and a write lock? (1)

(d) Describe the *two-phase locking protocol*. (4)

(e) What is a *deadlock*? Give an example of a deadlock involving two transactions which use the two-phase locking protocol. (4)

(f) What is a *wait-for graph* and how can it be used for deadlock detection? Illustrate this on the example from your answer to part (e). (5)

(g) List two ways of deadlock prevention. (4)