University of Glasgow

May 2009
3.00 pm – 4.30 pm
(Duration: 2 hours)

EXAMINATION FOR MSc & POSTGRADUATE DIPLOMA IN INFORMATION TECHNOLOGY

&

MRES IN BIOINFORMATICS

INFORMATION SYSTEMS AND DATABASES

SOLUTIONS

Answer Question 1 in Section A and Two Questions from Section B

This examination paper is worth a total of 70 marks

You must not leave the examination room within the first hour or the last half-hour of the examination. (for exams of 2 hours duration)

You must not leave the examination room within the first half hour or the last fifteen minutes of the examination. (for exams of less than 2 hours duration)

Section A

1. Take **two** of the following eight topics and write a short description with examples:

(a) The use of **styles** in **word processing**.

Style as set of formatting decision that can be shared between paragraphs so that you can change the look of several parts of the document at the same time.

They can also used for structure as well – e.g. heading styles are used to create a table of contents.

(b) **Cell formatting** in a spreadsheet.

Use for giving clarity to the spreadsheet – options include font, colour, lines, etc.

Also for distinguishing different numerical types.

(c) **Relative and absolute addressing** in a spreadsheet formula.

An absolute address refers to one specific cell, while a relative one describes a style relative to the position of the formula.

Cell references are relative by default. You must insert $'s for absolute.

Thus in X8, the formula "=Y7" means copy the cell one down and one to the right, while "=Y7" means Y7 wherever the formula is moved to.

(d) **Colour**

Describe main kinds of colour formatting RGB, CYMK, YUV and the reasons for two main kinds – display and printing.

(e) **Haptic** user interfaces and their use**.**

A haptic interface uses touch to communicate device the computer, usually by pressing a force feedback device, which allows the computer to detect pressure or movement and also to provide various degrees of resistance.

It is used for applications which need to simulate physical manipulation but are either too dangerous, costly, small or remote.

(f) **MIME types**

A type structure for multimedia data.

There are two parts to the description: the media type (image, video, etc.) and the specific format (jpg, avi, etc.)

Used by internet applications to describe the kind of documents is being sent.

(g) **Core attributes** in XHTML.

name, id, class, style, lang and title. These are attributes available for all elements.

They are used for formatting and as identifiers for an object thus permitting access to the information they contain.

(h) **Lookup functions** in spreadsheets**.**

Use of VLOOKUP / HLOOKUP to retrieve values.

They have one parameter which is a lookup table and others to identify the value to be looked up and where to find the related value.

[5 marks each]

Section B

2. **(a)** Describe how **fan traps** and **chasm traps** cause modeling problems in ER diagrams.

[4]

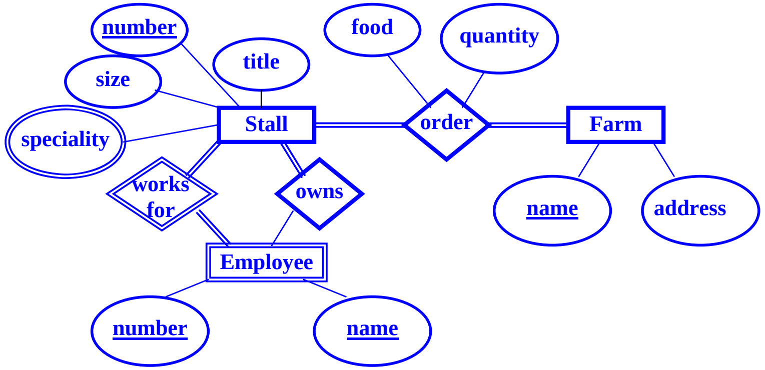
Fan traps occur when one entity types stands in 1-N relationship with two other entity types, which themselves should be related. From each of these, it is possible to get back to the first entity but not to the other related – a separate relationship is required. [2]

Chasm traps occur when entity type A is related to many entities of type B and each B is related to many entities of type C. You should be able to find the A which is related to any C, unless there is partial participation between B and C in which case there might be C's with no related B and therefore no way of finding the related A. Again a separate relationship is required. [2]

(b) Create an ER diagram from the following specification:

A food market is made up of a set of numbered stalls, each with a title, a size and a set of specialities (vegetables, meat, etc.). The stalls buy food from farms, which have a name and an address by making orders each of which is for a particular quantity of a certain foodstuff. Each stall has several employees, one of whom is identified as the stall holder and each of whom is numbered using a numbering system which is the same for each stall - i.e. each stall has employee 1, employee 2, etc. The name and age of each employee is recorded.

[11]



4 marks for thestrong entities, 2 for the weak entity, 2 for the entity attributes and, 1 for the relationships attributes and 1 for the MV attribute.

(c) Compare the use of binary search over ordered data with the use of hash functions to find records efficiently.

[5]

Description of how each work – slides 538-540s. Speed of inserting using hash function if well designed. [2 marks each description and 1 for discussion]

(d) Describe **two problems** which might arise if there was no control over the concurrent access to data by multiple users.

[5]

Lost update, incorrect summary and temporary update described – slides 493-495. [2 marks each and 1 discretion]

(e) Describe the concept of **isolation** to deal with these problems and give situations in which different **isolation levels** would be used.

[5]

These are completely eliminated by serialisable isolation using locks – slide 490. If conflicts are unlikely a less restricted isolation level may be acceptable with example slide 504. [2 for isolation, 3 for context]

3. The following relational database is used within a library. The *Borrows* table contains only the currently active loans. A few of the records in each table are shown.

BOOK

title	author	access#	branch#
Emma	Austen	501	5
Dracula	Stoker	502	5
Justine	Durrell	503	5
Kim	Kipling	504	6

Uses

user	ubranch
701	5
703	6
704	5
704	6

BORROWS

book#	reader#	dueDate
501	702	01-MAY-2009
503	703	12-MAY-2009
504	703	25-APR-2009

BRANCH

bName	address	br#
Hillhead	31 Byres ...	5
Partick	874, Dum..	6

READER

rName	address	ticket#	dateOfBirth
T Black	12 Oak Ave	701	15-MAR-1944
A Gray	5 High St	702	22-JAN-1989
M Jones	1 High St	703	19-JUN-1978

The foreign keys are as follows:

Book.branch# refers to Branch.br#

Borrows.book# refers to Book.access#

Borrows.reader# refers to Reader.ticket#

Uses.user refers to Reader.ticket#

Uses.ubranch refers to Branch.br#

(a) Give **SQL** and **relational algebra** queries to retrieve the following:

(i) the names of all the readers and authors in one query;

[2]

RA (project Author from Book) union (project Rname from Reader) [1]

SQL (select Author from Book) union (select Rname from Reader) [1]

(ii) the address of the reader who has borrowed the book 'Justine';

[4]

RA J = select Book where Title = 'Justine'
Jborrower = join J with Borrows on book#=access#
Jreader = join Jborrower with Reader on reader# = Ticket#
project Address from Jreader [2]

SQL select Address from Reader, Borrows, Book
where reader# = Ticket# and Title = 'Justine'
and book#=access# [2]

(iii) the names of readers who do **not** use branch 5.

[6]

RA branch5users = select Uses where ubranch=5
branch5userNumbers = project user from branch5users
allUserNumbers = project ticket# from Reader
requiredUserNumbers = allUserNumbers - branch5userNumbers
requiredUsers = join Reader with requiredUserNumbers
on user = ticket#
project rname from requiredUsers [3]

SQL select rname from Reader where ticket# not in
(select user from Uses where ubranch=5) [3]

(b) Give **SQL** queries to achieve the following:

(i) Return the branch names and the number of users of each branch.

[3]

select bName, count(*) from Branch, Users where ubranch=branch#
group by bName [3]

(ii) Move the book 'Emma' to branch 6.

[1]

update Book set branch# = 6 where title='Emma' [1]

(iii) Remove branch 6 and all of its users and books.

[2]

delete Book where branch# = 6
delete Branch where branch# = 6 // order important [2]

(c) Describe the use of **null values** in a database including reasons why they might be needed and anomalies which might arise when querying a table with null values using the database above as an example.

[5]

All cells of a table must be filled. Null values are used where no other suitable value is available - e.g. not known and not relevant. [2]

Anomalies occur because of the semantics of queries involving null - e.g.

select rname from Reader where dateOfBirth <1970 and dateOfBirth>=170 omits reader 502 when it might reasonably have been expected to return everybody. [3]

(d) Give one **constraint** for the database that can be asserted when creating the tables and another constraint which could only be asserted in a program managing the database.

[3]

Many examples possible - non-null names for instance [1 each, 1 discretion]

Complex constraint example - readers can only borrow books from branches for which they are registered as users.

(e) Describe the way in which **distributed databases** commit their data.

[4]

Description of two phase commit – slide 512. [2 for idea of distributed commit, 2 for the sequence of events]

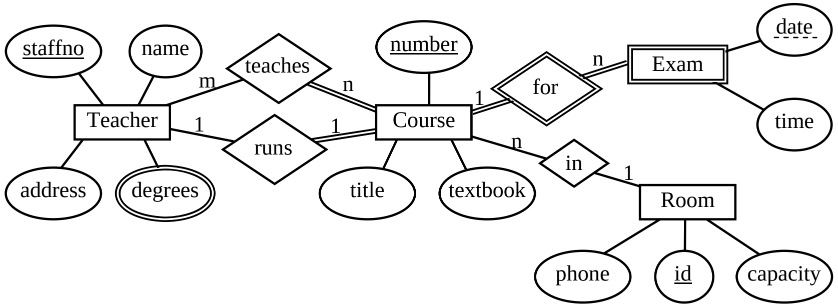
4. (a) To what extent can a **spreadsheet program** be said to provide **database functions**?

[4]

Spreadsheet structure is like a table and provides sorting and limited ways of selecting data and returning summary information. [4]

(b) Given the following ER diagram, produce a **relational schema**, indicating the primary and foreign keys:

[10]



Teacher(StaffNo, name, address)
Degrees(teacher references Teacher(staffNo), degree)
Teaches(teacher references Teacher(staffNo),
course references Course(number))
Course(number, title, textbook, inroom references Room(id))
Exam(course references Course(number), date, time)
Room(id, phone, capacity)
[1 each for Teacher and Room, 2 each for Course, Teaches, Exam, Degree]

(c) A schema for a database holding information books, publishers and authors is given below:

Book(ISBN, title, price, pubid)

Author(auID, aname, nationality, gender)

Publisher(pubID, pname, phone, address)

WrittenBy(ISBN, auID)

Suggest the **functional dependencies** that are implied by this schema.

[2]

ISBN -> title, price, pubID, pname, phone, address
pubID -> pname, phone, address
auID -> aname, nationality, gender [1 for most, 2 for all]

(d) Show the two **tables** that would be created by (i) the inner join of Book and Publisher and (ii) the inner join of WrittenBy and Author. State which **normal form** each of these is in, justifying your answer with a reason and giving some update anomalies that might occur if these were used as the base tables.

[6]

BookPublisher(ISBN, title, price, pubid, pname, phone, address) [1]

In 2NF as only has one key column, but has transitive dependencies. [1]

WrittenByAuthor(ISBN, auID, aname, nationality, gender) [1]

In 1NF, since there are columns dependent on part of the key. [1]

Updating the tables could cause multiple inconsistent versions of author or publisher data to be added and could cause publisher data to be lost when a book is deleted. [2]

(e) The publishers want to have the ability to assign more than one price to a book for different types of sale. What **changes** need to be made to this **schema**?

[2]

Remove price from Book and add table Price(ISBN, TypeOfSale, Price) [2]

(f) Using pseudo-code, show how a web site program makes use of data submitted through **XHTML forms**.

[6]

Form controls are named using ID or name attribute. [2]

Value gets sent with HTTP request [2]

Server side program gets access to the values entered into the form by using the name and can then use them as data in the program in any way it needs to. [2]