University of Glasgow

May 2015

DEGREE of MSc

Or

DEGREES of MSc, MSci, MEng, BEng, BSc, MA and MA (Social Sciences)

Or

DEGREES OF MSci, MEng, BEng, BSc, MA and MA (Social Sciences)

CS 1Q

Exam Rubric: Answer All Questions

INSTRUCTIONS TO INVIGILATORS

Please collect all exam question papers and exam answer scripts and retain for school to collect. Candidates must not remove exam question papers.

Section A 25 Marks

1(a) In the context of interactive systems design, explain the following concepts and methods:

i. Containment hierarchy

This is the hierarchy of screen areas in a GUI—usually rectangles—that describes the nesting of windows, widgets and other GUI components [1]. It is used in the management of windows, e.g. for finding which component to send a mouse click event to [1].

[2]

ii. Field studies

[2]

Field studies are a form of evaluation done in uncontrolled environments such as workplaces and homes[1]. Data is collected by notes, audio/video recording and/or system logs, and then analysed to find categories and examples useful for understanding and design. [1]

iii. The serif hypothesis

[2]

This is the claim that typefaces with the small tails and tags on letters, such as Times, are easier to read than non-serif fonts such as Helvetica [1]. Experiments have often had conflicting results, as it seems people find it easiest to read whatever style of typeface they are most used to [1].

iv. Distributed cognition

[2]

A theoretical approach to psychology and design, in which cognition is treated as something involving many people and objects, i.e. distributed across them all in one wider system [1]. The key role of external representations helps us understand design features/requirements such as representation of complex data in visualisations [1].

v. Low fidelity prototype

[2]

A prototype is a model or rough version of a system design, used to refine requirements and other design features [1]. A low fidelity prototype uses materials unlike those of the finished digital product, e.g. paper or cardboard [1].

1(b) Event-driven programming is a better paradigm for interactive systems design than command-driven programming. Why?

[4]

The most essential advantage is the flexibility of response of event-driven programming, in that an event can be

fired (and the system can respond) whenever a user action or sensor input happens [1]. In contrast, command-driven programming delays such a response until any previous command's response has finished [1]. Event-driven systems can therefore react quickly to the many widgets of GUIs, such as icons, menus and scrollbars [1]. This makes event-driven systems feel much more interactive [1].

1(c) Trainline.com is a web site that gives train timetable information and sells train tickets. It has a new web site prototype, ready for A/B testing. It is intended to better serve the train travellers' needs and interests, and to increase ticket revenues. Explain what A/B testing is, and describe how you could use it to achieve Trainline's aims.

[11]

In A/B testing, traffic to versions of an application, such as a web site, is randomly split among users [1]. Usually one ('A') is the current or control version, and a second version ('B') implements some new design idea [1]. A/B testing shows causal relationships in a real world setting, but may not explain exactly why a given version is better or worse [1].

Usually, B will be introduced to a small proportion (e.g. 1% of users). Then, if initial analysis shows that it is working satisfactorily, then the proportion can gradually be increased up to 50%. We would do the same here, ramping up from a small proportion to half of the user population. [1]

Metrics based on logged use, such as product sales and users clicking through to chosen pages, are then used to assess the strengths and weaknesses of each version [1]. If B is much better than A then it gives a strong argument for shifting all users to the new version [1].

Here we choose just one overall evaluation criterion mentioned in the spec: ticket revenues [1].

Usually, B will be introduced to a small proportion (e.g. 1% of users). Then, if initial analysis shows that it is working satisfactorily then the proportion can gradually be increased up to 50%. We would do the same here, ramping up from a small proportion to half of the user population if the revenue from B is significantly greater than A. [1]

In parallel, data from different subgroups of users given the same version should be tested (i.e. A/A testing, or B/B testing) [1] to see if random variation within groups is leading to statistically significant variation in the main metric (or metrics) and to see if the sample sizes are large enough [1].

Care must be taken in randomisation, e.g. each user repeatedly returning to the site must be given the same version each time (via cookies, for example) [1].

Section B 25 Marks

2(a) Provide short descriptions of any three of the following terms. Use examples and/or diagrams to illustrate your answers.

(i) The union set operator (provide an example of usage and a Venn diagram).
Combines two sets, taking items from both sets [1]. Example would show that items from both sets should be kept [1]. Correct Venn diagram [1].

(ii) The difference between conceptual and implementation models for a database.
The conceptual model for a database is a description of the real-world entities and their relationships [1]. The implementation model is the conceptual scheme: what relations (tables) will be created, primary & foreign keys [1]. The example could show an E/R diagram and resulting DB relational schema [1].

(iii) Weak entities in an ER diagram.
Weak entities do not have sufficient attributes to be uniquely identified by themselves [1]. They are represented in an ER diagram with a double lined box, with the partial key dotted underlined [1]. An example would be a building and a room. [1]

(iv) A many-to-many relationship in an ER diagram.
A relationship where both entity types can participate more than once in the relationship [1]. An example of the notation in the ER diagram should be given [1]. An example could be students and supervisors. [1]

(v) The degree and cardinality of a relation.
The degree of a relation is the number of attribute [1]; the cardinality is the number of rows [1]. An example should show a table and give the counts. [1].

(vi) A foreign key in the context of a relational database.
A foreign key is an attribute of a relation (field) which references the primary key of some other relation [1]. This is required in order to uniquely identify a row in another table [1]. OR It is a column that is used to establish and enforce a link between two tables [1]. An example could be Staff.DeptNo references Department.Id.

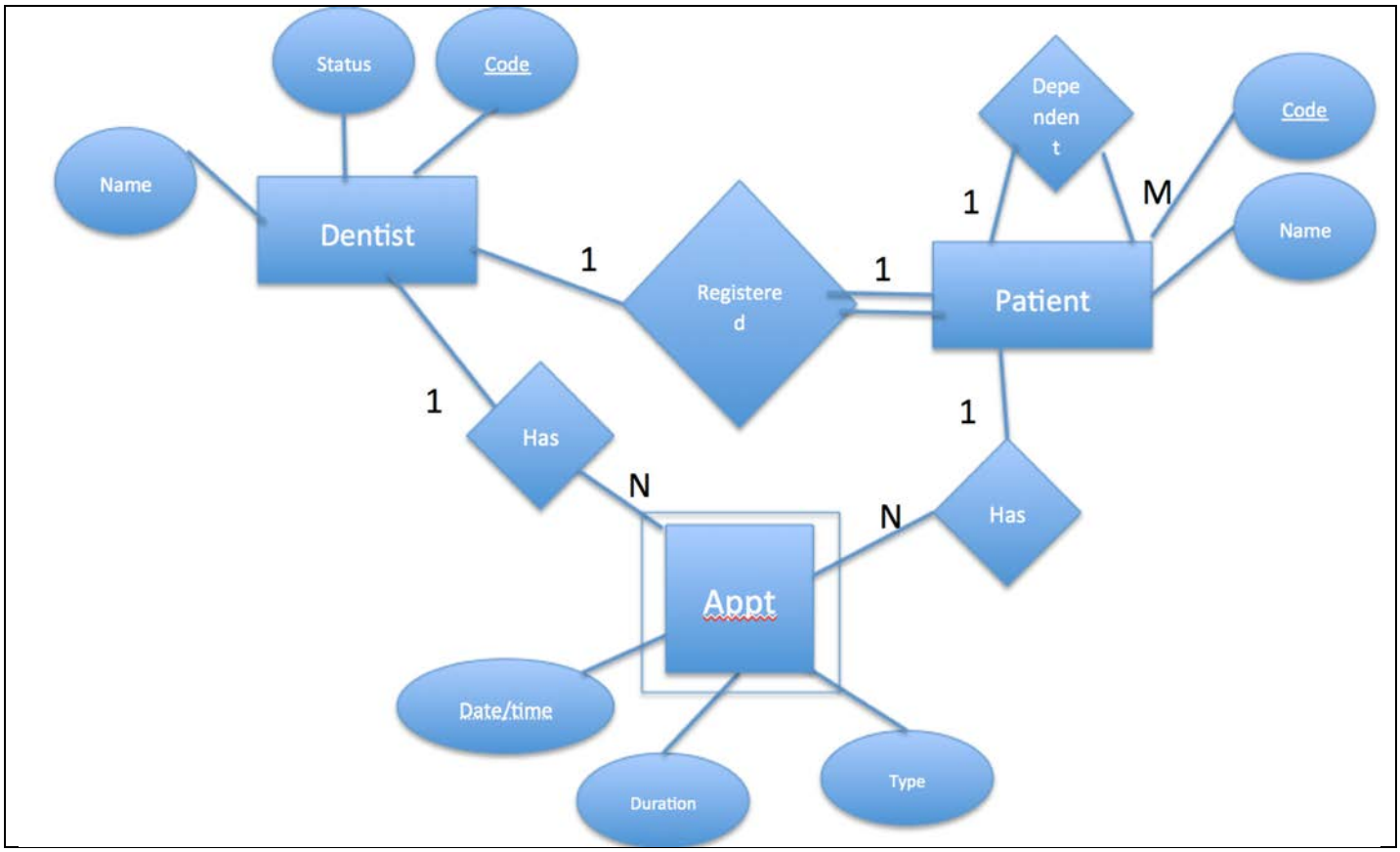
(vii) The projection and selection operations in relational algebra.
The projection operator selects attributes from a relation to return as a new relation, identified using the PI symbol [1]. The selection operator, identified by the SIGMA symbol, identifies rows to be returned in a new relation, identified using the conditions. [1] The provided example should show clearly the symbols, input relations and what rows and attributes are identified [1].

[9]

2(b) A dental practice keeps data on its dentists and patients. A patient has a unique code and name, and may be a dependent of another patient. Each dentist also has a unique code, a name and a status (part-time or full-time). Each patient must be registered with a single dentist. An appointment is when a single patient meets a dentist at a given date & time (not necessarily the patient's own registered dentist). Appointments also have durations and appointment types (checkup, operation).

Draw an ER diagram that could be used in the development of a database to support the dental practice.

[7]



Strong Entities: Dentist, Patient & attributes [1]

Identification of Strong entity Primary Keys [1]

Weak Entities: Appointment & partial key (date/time) [1]. *Student identifying Appointment as a relationship doesn't get the mark, and may also lose a mark for cardinalities below.*

Identification of relationships with cardinalities. [2] ½ for Registered; 1 for Dependent; 1 for the two involved in the Appointment weak entity.

Identification of participations [2]: 1 for total of patient in Registered ("must"), 1 for total of Appt in Dentist & Patient (weak entity).

(c) Assume a relational database with two tables, as follows:

Staff(NI_Num, Firstname, Surname, PartTime, Dept)

Department(DeptNo, Name, Building)

Building(BuildingId, StreetNum, StreetAddress, City)

where Staff.Dept is a foreign key reference to Department.DeptNo and Department.Building is a foreign key reference to Buidling.BuildingId.

(i) Write a relational algebra expression to find the surnames of all staff who work in department 5.[1]

$\Pi_{Surname}\ (\sigma_{Dept = 5}\ (\ Staff\ )\ )$

(ii) Write the SQL query to find the names of the departments based in Glasgow city. [2]

```
SELECT Name
FROM Department, Building
WHERE Department.Building = Building.BuildIngId
AND Building.City = "Glasgow"
```

Projection (1/2 mark); WHERE on City (1/2 mark); FROM & join condition (1 mark)

(iii) Write the SQL query to find how many part time staff work in each department number. [2]

```
SELECT Dept, Count(NI_Num)
FROM Staff
WHERE PartTime = 1
GROUP BY Dept
```

COUNT & Group BY (1 mark); WHERE on PartTime (1/2 mark), project Dept (1/2 mark)

(d) Given the following sets:

A = {1,4,5} B = {a,e}

give the following sets. Assume that $\square$ is the Cartesian product.

(i) | A |

(ii) A $\square$ B

(iii) A $\cup$ B

(iv) {<x,y> | x $\in$ A $\wedge$ y $\in$A $\wedge$ x<y}

[4]

(i) 3

(ii) { <1,a>, <1,e>, <4,a>, <4,e>, <5,a>, <5,e> }

(iii) {1,4,5,a,e}

(iv) { <1,4>, <4,5> }

Section C 25 Marks

1. (a) Convert 1101 1010 to a decimal number, assuming binary representation.

[2]

128 + 64 + 16 + 8 + 2 = 218 by adding the powers of 2 corresponding to the positions where there is a 1 bit in the word. [Problem solving.]

(b) Convert 1101 1010 to a decimal number, assuming two's complement representation.

[3]

Since the leftmost bit is 1, this is a negative number. Negate it to get a nonnegative number. To negate, first invert giving 0010 0101. Then increment, giving 0010 0110. Now this is nonnegative so its binary representation is the same as its two's complement value; this is 32 + 4 + 2 = 38. Since the negation of the original word is 38, the answer is -38. [Problem solving. 1 mark for identifying it as negative; 2 marks for negation.]

(c) Translate the following program fragment into assembly language for Sigma16. You may assume that the variables have been declared with data statements; just write the instructions needed.

```
if x<y
   then  a = a + 3
   else  b = a - b
```

[10]

```
   load   R1,x[R0]    ; R1 = x
   load   R2,y[R0]    ; R2 = y
   cmplt  R3,R1,R2    ; R3 = x<y
   jumpf  R3,else     ; if not (x<y) then goto else
   load   R4,a[R0]    ; R4 = a
   lea    R5,3[R0]    ; R5 = 3
   add    R4,R4,R5    ; R4 = a + 3
   store  R4,a[R0]    ; a = a + 3
   jump   done        ; goto done
else
   load   R4,a[R0]    ; R4 = a
   load   R5,b[R0]    ; R5 = b
   sub    R4,R4,R5    ; R4 = a – b
   store  R4,b[R0]    ; b = a – b
done
```

[Unseen problem. 2 marks for comparison, 3 marks for branching, 5 marks for assignments.]

(d) Suppose `x` is an array containing `n` integers, where `n` is an integer variable in memory. There is another integer variable `const`. Write a loop that adds `const` to every element of the array. You may assume that the variables `n` and `const`, and the array `x` have been declared; just write the necessary instructions.

[10]

```
   add    R1,R0,R0       ; R1 = index = 0
   load   R2,n[R0]       ; R2 = n
   load   R3,const[R0]   ; R3 = const
   lea    R4,1[R0]       ; R4 = 1
loop
   cmplt  R5,R1,R2       ; R5 = (index < n)
   jumpf  R5,done[R0]    ; if not (index < n) then goto done
   load   R5,x[R1]       ; R5 = x[index]
   add    R5,R5,R3       ; R5 = x[index] + const
   store  R5,x[R1]       ; x[index] = x[index] + const
   add    R1,R1,R4       ; index = index + 1
   jump   loop[R0]       ; goto loop
```

[Problem solving, requires understanding of indexed addressing and loops. 3 marks for initialization, 3 marks for loop, 4 marks for array update.]