THIS PAPER IS NOT TO BE REMOVED FROM THE EXAMINATION HALLS

UNIVERSITY OF LONDON **CO2209 ZA**

BSc Examination

COMPUTING AND INFORMATION SYSTEMS, CREATIVE COMPUTING and COMBINED DEGREE SCHEME

Database Systems

Date and Time: Monday 8 May 2017: 10.00 – 13.00

Duration: 3 hours

There are **FIVE** questions on this paper. Candidates should answer **FOUR** questions. All questions carry equal marks and full marks can be obtained for complete answers to **FOUR** questions. The marks for each part of a question are indicated at the end of the part in [.] brackets.

Only your first **FOUR** answers, in the order that they appear in your answer book, will be marked.

There are 100 marks available on this paper.

A handheld calculator may be used when answering questions on this paper but it must not be pre-programmed or able to display graphics text or algebraic equations. The make and type of machine must be stated clearly on the front cover of the answer book.

Question 1

A. Under what circumstances, and why, might a database system want to undo changes that it has just made to the stored data?

[2 marks]

B. Why is it necessary to deny read access to the tables of a database which are in the process of being updated (in the middle of a transaction)?

[2 marks]

C. Describe the "data access protocol" (using shared and exclusive locks) in the context of the problem of multiple users wanting simultaneous ("concurrent") access to the same data.

[4 marks]

D. Illustrate how the data access protocol can be applied in order to solve a concurrent access problem (either the lost update, uncommitted dependency or inconsistent analysis problem).

[4 marks]

E. What is meant by saying that a set of transactions whose execution operations are interleaved with each other, is "serialisable"? How can this be guaranteed for a set of transactions? What is the "two-phase locking protocol"?

[4 marks]

F. Most database management systems have one or more files which make up a "data dictionary" or "system catalogue". What do these terms refer to? Describe two sorts of data which might be found in a system catalogue, and briefly indicate to what uses each sort of data might be put by the database system.

[5 marks]

G. What is the difference between 'database security' and 'database integrity'? What are some ways of maintaining each?

[4 marks]

TOTAL = 25 marks

Question 2

The database for a small budget airline needs to record information about the airline's Aircraft, the regular Flights they make, and the Passengers on board a Flight that takes place on a certain date, called a Trip. Passengers make Bookings for Trips. A Trip can exist before any bookings have been made for it.

Flights are identified by Flight Numbers and the days of the week on which the flight will operate. Each Flight has a Departure City and a Destination City, a Take-off Time and an Arrival Time, and a First-Class-Ticket-Price and an Economy-Class-Ticket-Price. A Flight can be recorded in the system before any specific Trips have taken place.

A Passenger is given a unique Customer-Number, has a Family-Name and First-Name, and an (optional) Credit-Card Number. For a Passenger to be recorded, he must have booked at least one Trip.

A Passenger can make a booking on one or more Trips, each for a specific date. (He can book multiple Trips.) (A given Flight on a particular Date is called a Trip.) Each Trip will have many Passengers, and one Pilot. A Passenger's booking of a Trip will be for a given class of ticket. A given Flight is in one direction between two cities. (Return journeys have different Flight Numbers.)

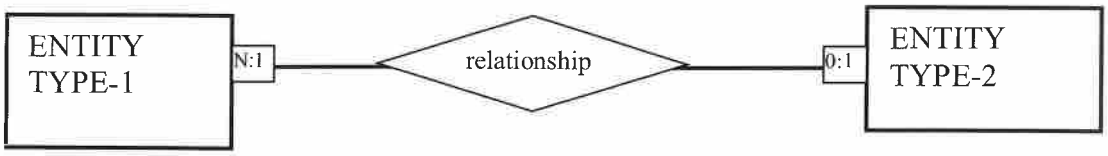
For example, Flight 034 takes place between London and Edinburgh, takes off at 6.30 am on Monday through Friday, arriving at 7.20 am, and is always made by an Airbus 320. (The airline owns several Airbus 320 Aircraft, and any one of them may be used on this Flight on a particular date.)

The airline employs Pilots, who have names, and (unique) Employee Numbers. Pilots fly the Aircraft that make Trips. A Pilot's details may exist in the system before he or she has flown on any Trips.

An Aircraft will be a certain Model (such as an Airbus 320), be identified by a unique Plane-Number, have a certain number of First-Class Seats and a certain number of Economy-Class Seats. (These may differ slightly from Aircraft to Aircraft even among Aircraft of the same model.) An Aircraft may be recorded in the system before it has been used for any Trips.

Our database must be able to hold the following information: Passenger P2094, whose name is Sally Languid, and whose credit card number is 4546-9823-1902, booked an Economy-Class Seat on Flight 034 for the 18th of September 2016. On this Trip she was allocated Seat D in Row 30. The Aircraft used for that Trip was Plane-Number 23, an Airbus 330, flown by Captain Hector Tzippas, whose Employee Number is P345. This particular Aircraft has 40 First-Class and 120 Economy Class Seats.

A. Draw an Entity/Relationship diagram to represent this situation. Include all entity types and their relationships but not their attributes. It is not necessary to show a 'SEAT' entity type. Indicate cardinality and participation constraints. Use the following conventions:



This illustrates a situation where 'Entity-type-1' can have a relationship to zero, or one (but no more than one) instance of 'Entity-type-2', and where 'Entity-type-2' must have that relationship with at least one instance of 'Entity-type-1', and can have it with an unlimited number of them.

[10 marks]

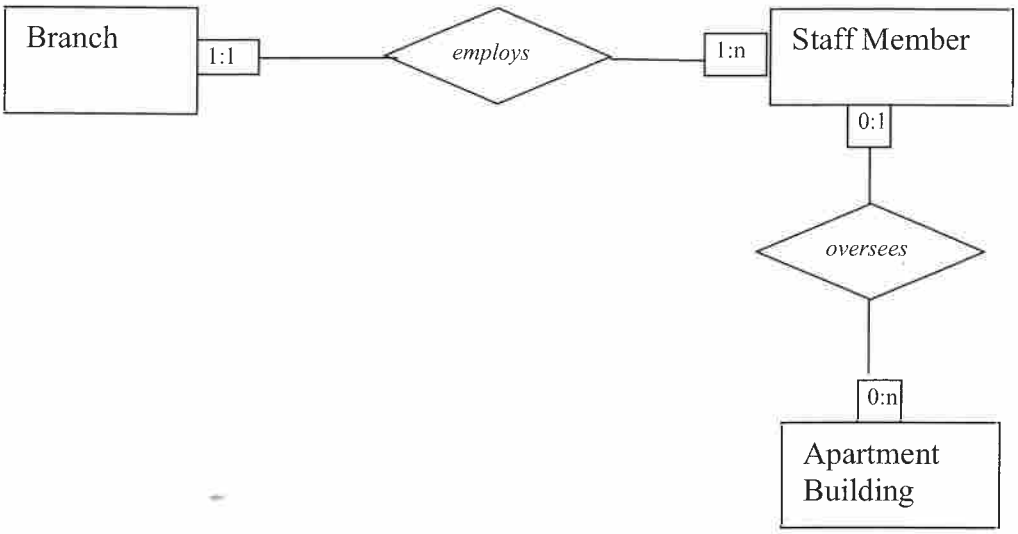
B. Design a normalized relational schema which could hold the information represented by the E/R diagram you drew up for **Part A.** (The previous description gives information about the attributes you will need, which you did not have to include on the E/R diagram.) Populate it with the example data given in **Part A**. Be sure to indicate both Primary and Foreign Keys.

[15 marks]

Total = 25 marks

Question 3

A. Consider a situation where a company is responsible for the maintenance and repair of apartment buildings. The company has a number of Branches, each of which is responsible for maintaining one or more Apartment Buildings in its area. Each Branch employs several maintenance Staff, each of whom is responsible for several Apartment Buildings. The relationships have been modelled this way. A Branch, represented by a BranchNum, employs one or more Staff Members, represented by EmpNums, and a Staff Member must be employed by a single Branch. A Staff Member can oversee several Apartment Buildings, represented by PropNums, but may not yet have been assigned any. An Apartment Building is overseen by only one Staff Member, but may not have been assigned one to oversee it yet. An analyst has modelled this situation with the following Entity-Relationship diagram.



The analyst has proposed the following relational schema to implement this Entity-Relationship model.

EMPLOYED BY
Primary Key: EmpNum
Other Attributes: BranchNum

OVERSEEN BY
Primary Key: PropNum
Other Attributes: EmpNum

The analyst asserts that this model will always allow us to see which Staff Members work for which Branches, which Apartment Buildings are overseen by which Staff Members, and, via a simple join, which Apartment Buildings are maintained by which Branches. (All of these links have been requested by the client for whom the database is being built.)

However, there is a flaw in this Entity-Relationship diagram and the proposed Relational Schema. What is it, and how can we repair the Entity-Relationship diagram and the Relational Schema to fix it? Draw a better Entity-Relationship Diagram and show a correct Relational Schema.

[6 marks]

B. Briefly define each of the following terms, in the context of their use when referring to databases:

1) Determinant
2) Boyce-Codd Normal Form
3) Foreign Key
4) View

[4 marks]

C. When the relational model was first proposed, in the 1970s, most database systems dealt with relatively small numbers of well-structured business records confined to a single computer or an in-house network, accessed in a controlled, predictable manner.

1) What has changed since then, leading to interest in non-relational approaches to data management?

2) What are some of the developments in ways of describing and/or structuring data that attempt to meet these changes? (In other words, how may data that is to be queried be held, other than in relational form?)

A good answer will mention several developments, and include at least one concrete example of a situation involving data for which the relational model is a poor fit.

[15 marks]

TOTAL = 25 marks

Question 4

The following relation holds information about appearances given by stand-up comedians: it holds the dates of their performances, the time of the beginning of their show, where it will be held (the town and a 'venue' in that town – some cities have more than one venue), the capacity of the venue, and how long the performance will last. Some venues are part of a chain, so that the same venue name may appear in more than one town. A particular venue (in a particular town) has only one seating capacity.

Comedians give titles to their performances, and may have more than one title in their repertoire. A particular comedian's performance, as identified by its title, has only one length. The same performance (identified by its unique title) is only given by one comedian – no two comedians give performances with the same title.

A comedian can give a performance more than once in the same town but of course on different dates, possibly at the same venue. A comedian gives, at most, only one performance per day. A comedian also has a 'holiday month' during which he or she cannot be booked.

We will use the database to record where comedians are performing and when. We also want to record which towns have which venues, and the seating capacity of each venue in that town. We also want to record the titles of the routine(s) each comedian can give, and how long they last. (A 'routine' is the same thing as a 'performance', or an 'appearance'.)

PERFORMANCE

PRIMARY KEY: COMEDIAN + DATE

COMEDIAN	DATE	TITLE	HOL-MONTH	TIME	TOWN	VENUE	CAPACITY	LENGTH
Lord Laugh-a-lot	2017-02-17	"Jobs of the Future"	Aug	23.00	Los Diablos	After Hours Spot	125	45 min
Ali Baba	2017-01-07	"Electric Cars"	Jul	22.00	Victoria	After Hours Spot	65	75 min
Ali Baba	2017-01-08	"Thought for the Day"	Jul	22.00	Victoria	After Hours Spot	65	40 min
Lord Laugh-a-lot	2017-02-01	"Jobs of the Future"	Aug	22.30	New Winston	Workers Club	50	45 min
Lord Laugh-a-lot	2017-02-03	"Jobs of the Future"	Aug	23.00	Rudbridge	Elmo's Joint	30	45 min
Ali Baba	2017-02-04	"Thought for the Day"	Jul	23.00	Goford	After Hours Spot	55	40 min
Jerry Jokeman	2017-02-03	"PC Nursery Stories"	Sep	22.00	Rudbridge	Elmo's Joint	30	60 min
Lord Laugh-a-lot	2017-02-20	"Jobs of the Future"	Aug	22.00	New Winston	Chillax Corner	75	45 min
Ali Baba	2017-02-30	"Thought for the Day"	Jul	22.00	Victoria	After Hours Spot	65	40 min

A. Identify the Functional Dependencies in this table.

[10 marks]

B. This table is susceptible to insertion, deletion, and update anomalies. Give an example, based on venues and their seating capacities table, of each kind.

[6 marks]

C. Identify the partial and transitive dependencies in the original relation.

[2 marks]

D. Change the schema so that the data in this table is in BCNF, specifying the Primary Keys of each new table, and populating two tuples of each of your new relations with data held in the old relations.

[7 marks]

TOTAL = 25 marks

Question 5

A. A metal alloy supply company wants you to implement a database describing its products. Each distinct metal alloy it sells has a unique ALCODE, a NAME, a single THICKNESS, and should -- but doesn't always – have a particular employee who possesses specialist knowledge of that metal alloy and who has been designated the 'go to' person for information about it, identified by an EMPNUM.

It also wants to record which customer, identified by CUSTNUM, has ordered how much of which product, and the date of the order. A customer never places more than one order for the same product on the same date.

Relation Name: **PRODUCT** Primary Key: ALCODE

Attributes:

ALCODE A number, ranging from 0 to 9999

NAME A string of characters, from 4 to 36 characters long. Note that more than one product may have the same NAME.

THICKNESS A real number, ranging from 0.5 to 25.0 – (mm).

EMPNUM A number, ranging from 0 to 999999, of the employee who is a specialist in this alloy. This field may be empty, in which case we put the keyword NULL there.)

ALCODE	NAME	THICKNESS	EMPNUM
8832	Chromium Steel	4.0	400302
5801	Nichrome wire	5.0	440377
7634	Carbon Steel	4.0	440377
4460	Galvanized iron	2.5	210008
4461	Galvanized iron	8.0	210008
6722	Anodized Aluminium	4.0	NULL

Relation Name:	**ORDERS** Primary Key: CUSTNUM + ALCODE + DATE
Attributes:	
CUSTNUM	A whole number, ranging from 0 to 999999
ALCODE	A whole number, ranging from 0 to 9999. This is a Foreign Key which references ALCODE in PRODUCT
DATE	A date.
QTY	A whole number, must be greater than 0 .... Never above 10000

CUSTNUM	ALCODE	DATE	QTY
458876	8832	2012-03-21	60
458876	8832	2016-10-02	60
458876	6722	2016-10-16	40
469855	6722	2016-10-20	70
687744	8832	2015-11-23	40
458876	4460	2015-03-14	50
687744	8832	2015-05-09	40

Write the SQL which can answer the following queries.

1) How many different products are in the PRODUCT table?

[1 mark]

2) List the ALCODEs of all metals which have no employee as a specialist.

[1 mark]

3) List the customer numbers of all customers who have ordered any metal alloy with a thickness > 4.

[2 marks]

4) What is the name (or names) of the metal(s) we stock with the greatest thickness?

[2 marks]

5) List the customer numbers and total quantities of all metals ordered by each customer.

[2 marks]

6) List the customer numbers and total quantities of all metals ordered by each customer from 2016 until now.

[2 marks]

7) List the customer numbers and total quantities of all metals ordered by each customer where the total quantity ordered is greater than 75.

[2 marks]

8) List the customer numbers of customers who have placed orders, but did not place any orders in 2015.

[3 marks]

B. What is meant by the term 'query optimisation'? Give one example of an optimisation that an optimiser might carry out on a complex query, involving selections on and joins of tables held on several different computers in a distributed database.

[5 marks]

C. Discuss briefly the problems involved in implementing a 'family tree' database, consisting of several pairs of ancestors, each with one or more descendants, who in turn have descendants, and so on.

[5 marks]

Total = 25 marks

END OF PAPER