ECM2419

(with Answers)

UNIVERSITY OF EXETER

COLLEGE OF ENGINEERING, MATHEMATICS AND PHYSICAL SCIENCES

COMPUTER SCIENCE

Examination, January 2021

Database Theory and Design

Module Leader: Dr Alberto Moraglio

Duration: TWO HOURS + 30 MINUTES UPLOAD TIME

Answer ALL questions.

The marks for this module are calculated from 70% of the percentage mark for this paper plus 30% of the percentage mark for associated coursework.

This is an OPEN BOOK examination.

Question 1

(a) There are a number of important functions provided by the *database management system*. Describe *four* of them.

(8 marks)

Any 4 of the following functions are correct. **[2]** each.

- *Data storage management*. Besides storage of data, it is also involved with a database's efficiency in relation to storage and access speed. Users do not need to know how data is stored or manipulated.
- *Data access and application programming interfaces*. SQL is the most common query language supported by the majority of DBMS vendors. It allows users to define the database, usually through a data definition language (DDL). It also allows users to insert, update, delete and retrieve data from the database, usually througha data manipulation language (DML).
- *A system catalog*. A system catalog stored the description of data items and is accessible to users.
- *Data communication interfaces*. Users' requests for database access are transmitted to DBMS in the form of communication messages.
- *Integrity services*. It ensures that both the data in the database and changes to the data follow certain rules.
- *Transaction management*. It ensures that all the updates corresponding to a given transaction are made or none of them is made.
- *Concurrency control services*. It ensures that the database is updated correctly when multiple users are updating the database concurrently.
- *Backup and recovery services*. It ensures that the database could be recovered in the event that the database is damaged in an way.
- *Authorization/security management*. It protects the database against unauthorized access, either accidental or intentional.

(b) What are the two main *access control methods* of database system? Briefly explain each method.

(8 marks)

The Discretionary Access Control (DAC) **[1]** and the Mandatory Access Control (MAC) **[1]**.

In discretionary access control each user is given appropriate access rights on specific database objects **[1]**. Typically users obtain certain privileges when they create an object **[1]** and can pass some of all privileges to other users at their discretion **[1]**.

In mandatory access control, each database object is assigned a certain classification level (e.g., top secret) **[1]**, and each user is given a certain clearance level **[1]**. The classification levels form a strict ordering and a subject requires the necessary clearance to read or write database objects **[1]**.

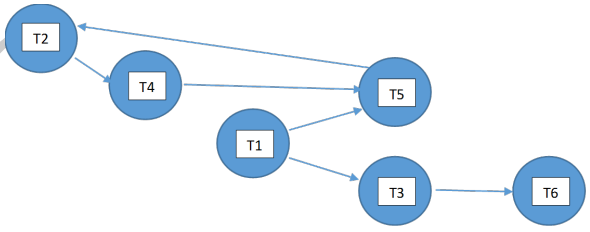
(c) What is *deadlock*? Produce a *wait-for graph* for the following transaction scenario and determine whether deadlock exists.

Transaction	**Data items locked by transaction**	**Data items transaction is waiting for**
T_1	z_1	z_2, z_3
T_2	z_4, z_5	z_6
T_3	z_3	z_7
T_4	z_6	z_2
T_5	z_2, z_8	z_4
T_6	z_7, z_9	

(9 marks)

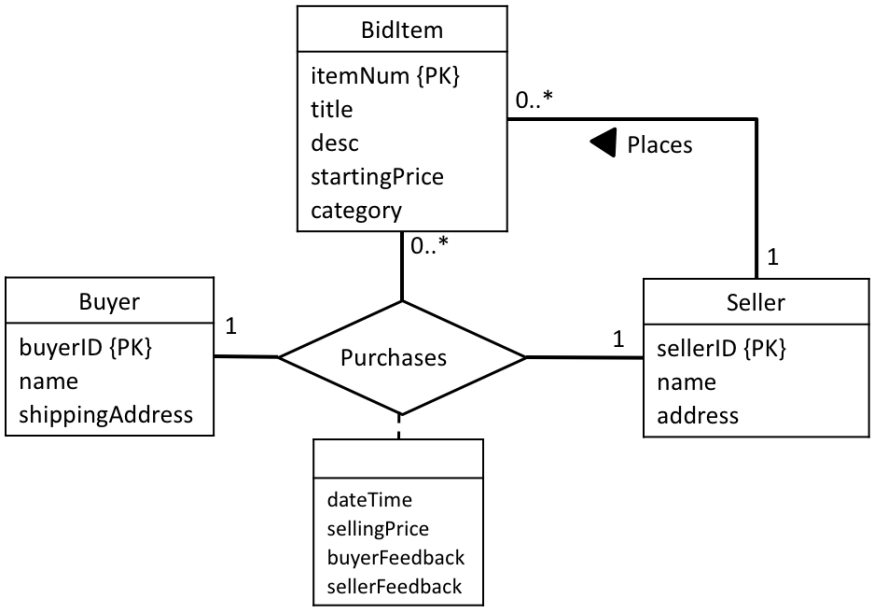
Deadlock is an impasse that may result when two (or more) transactions are each waiting for locks to be released that are held by the other **[2]**.

[5] marks for the wait-for-graph:



There is a deadlock as the wait-for-graph has a cycle (T2, T4, T5) **[2]**.

(d) Consider an online auction database system with the Entity-Relationship (ER) diagram below.



(i) Transform the above ER diagram into a *relational model* (with primary keys and foreign keys specified).

(8 marks)

BidItem (itemNum, title, desc, startingPrice, category, sellerID)

Primary Key: itemNum **[1]**

Foreign Key: sellerId references Seller(sellerID) **[1]**

Purchases (itemNum, sellerId, buyerId, dateTime, sellingPrice, buyerFeedback, sellerFeedback) **[1]**

Primary Key: itemNum **[1]**

[2] if the following 3 foreign keys are listed:
Foreign Key: itemNum references BidItem(itemNum)

Foreign Key: sellerId references Seller(sellerID)

Foreign Key: buyerId references Buyer(buyerID)

Buyer (buyerId, name, shippingAddress)

Primary Key: buyerID **[1]**

Seller (sellerId, name, address)

Primary Key: sellerId **[1]**

(ii) Write the following queries in *relational algebra*:

A) Find all the items whose starting prices are higher than 100 in the 'Furniture' category. List their titles and starting prices.

(3 marks)

$$\pi_{\text{title,startingPrice}}(\sigma_{\text{startingPrice}>100 \wedge \text{category}=\text{'Furniture'}}(\text{BidItem}))$$

B) List all the buyers' feedback for the items which are classified as 'Furniture' and sold by a seller with ID 'KP0105'.

(4 marks)

$$\pi_{\text{buyerFeedback}}(\sigma_{\text{sellerId}=\text{'KP0105'} \wedge \text{category}=\text{'Furniture'}}(\text{BidItem} \bowtie \text{Purchases}))$$

(Total 40 marks)

Question 2

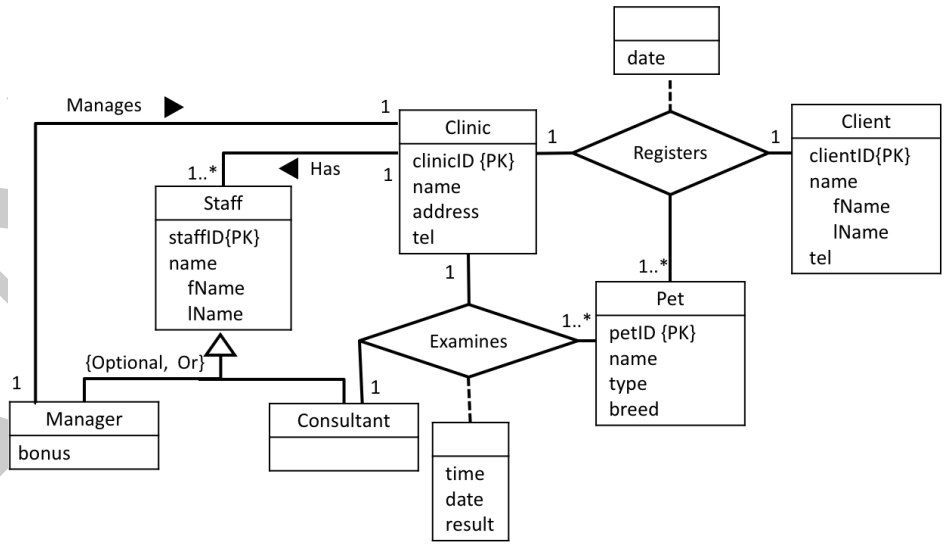
(a) A company runs a number of veterinary clinics for dogs. A clinic (identified by clinic ID, with some basic information such as name, address and telephone number) has one manager, a few consultants and some other staff. Each manager manages only one clinic. Different from normal staff members, a manger has an annual bonus.

When a dog owner contacts a clinic for the first time, his/her dog needs to be registered with the clinic. The registration records both the dog and its owner's basic information, together with the registration date. An owner can own one or more dogs, but each dog has only one owner. A dog can only register with one clinic. Once registered, it can come to any clinic for examination.

When the dog arrives at the clinic, it is examined by a member of the consulting staff. The examination's result is then recorded in the system in detail, together with examination's time and date.

Draw an *entity-relationship diagram* (using UML notation) with *specialisation/generalisation* concepts for the pet 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)



Marks:

- 6 entities with their attributes **[6]**
- 4 primary keys in the 4 entities **[2]**
- Relationship 'Registers', with multiplicity, attributes **[3]**
- Relationship 'Examines', with multiplicity, and attributes **[3]**
- Relationship 'Has', with multiplicity **[1]**
- Relationship 'Manages', with multiplicity **[1]**
- Superclass/subclass with {Optional, Or} **[2]**

(b) A company runs a minicab service. The `Job` table below shows some details of client bookings for minicabs. Assume that a minicab driver is assigned to a single minicab, but a minicab can be assigned to more than one driver at different times. It happens that one client might book more than one minicab at a given time.

date	24/09/18	29/09/18	30/09/18	30/09/18	01/10/18
time	10:00	10:00	19:10	12:30	08:30
driverID	D01	D01	D02	D03	D02
driverName	Joe	Joe	John	Neo	John
taxiID	T1	T1	T2	T3	T2
taxiMake	Ford	Ford	Skoda	Ford	Skoda
clientID	C1	C1	C1	C2	C4
clientName	Anne	Anne	Anne	Tom	David
pickupLoc	EX4 5AY	EX4 5AY	EX4 2ES	EX2 9BR	EX1 2NP

(i) Identify 2 *candidate keys* for the table.

(2 marks)

(date, time, driverID), (date, time, taxiID)

(ii) Explain the reason why the above relation is in the *1st Normal Form*. Transform it into relations in the *2nd Normal Form* by using relational schemas (with primary keys underlined) rather than detailed tables.

(4 marks)

Because the above relation has no multi-value attributes, but has a partial dependency if identifying the primary key as (date, time, driverID), which is:

`driverID` → `driverName`, `taxiID`, `taxiMake` **[2]**

The 2nd Normal Form:

`Job`(`date`, `time`, `driverID`, `clientID`, `clientName`, `pickupLoc`)

`Driver`(`driverID`, `driverName`, `taxiID`, `taxiMake`) **[2]**

(iii) What process should be done to transfer relations in the *2nd Normal Form* into the *3rd Normal Form*? Transform the above relations into relations in the *3rd Normal Form* by using relational schemas (with primary keys underlined) rather than detailed tables.

(4 marks)

Removing the transitive dependencies. **[1]**

The transitive dependencies are:

`date`, `time`, `driverID` → `clientID` → `clientName`

`driverID` → `taxiID` → `taxiMake` **[1]**

The 3rd Normal Form:

`Job`(`date`, `time`, `driverID`, `clientID`, `pickupLoc`)

Driver(<u>driverID</u>, driverName, taxiID)

Client(<u>clientID</u>, clientName)

Taxi(<u>taxiID</u>,taxiMake) **[2]**

(iv) Which TWO of the following are the advantages of *normalisation* in a database?

- A. Removing potential insertion, modification, deletion anomalies
- B. Removing redundancy
- C. Improving data security
- D. Speeding up data access

(2 marks)

A and B (Give marks only if both answers are correct)

(Total 30 marks)

Question 3

Consider the following database with airline flight information, where the primary keys are underlined. Note that the employees relation describes pilots and other kinds of employees as well; the aircraft relation describes types of planes, and not specific instances of planes; the certified relation distinguishes pilots from other employees: every pilot is certified for some aircraft, and only pilots, not other employees, are certified to fly.

flights(flNo, origin, destination, distance, depart-time, arrive-time, price)
aircraft(airID, air-name, cruising-range)
certified(empID, airID)
employees(empID, emp-name, salary)

Give an expression in SQL for each of the following queries.

(a) For each pilot certified for at least *four* aircraft, find the *empID* and the maximum *cruising-range* of the aircraft for which she or he is certified.

```
SELECT C.empID, MAX (A.cruising-range)
FROM Certified C, Aircraft A
WHERE C.airID = A.airID
GROUP BY C.empID
HAVING COUNT (*) > 3
```

correct SELECT **[1]**, correct FROM **[1]**, correct WHERE **[1]**, correct GROUP BY **[1]**, correct HAVING **[2]**

(6 marks)

(b) Retrieve the names of pilots whose *salary* is less than the price of the cheapest route from London to Barcelona.

```
SELECT DISTINCT E.emp-name
FROM employees E
WHERE E.salary < ( SELECT MIN (F.price)
FROM flights F
WHERE F.origin = 'London' AND F.destination = 'Barcelona' )
```

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

sub-query **[3]**

(6 marks)

(c) Find the names of pilots certified for Airbus aircraft.

```
SELECT DISTINCT E.emp-name
FROM employees E, certified C, aircraft A
WHERE E.empID = C.empID AND
C.airID = A.airID AND
A.air-name LIKE 'Airbus%'
```

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

Alternative correct syntax using the keyword 'JOIN' will be accepted as a correct answer.

(6 marks)

(d) Retrieve the *airID* of all aircraft whose cruising range allows them to be used on at least a route from London to New York.

```
SELECT A.airID
FROM aircraft A
WHERE A.cruising-range > ( SELECT MIN (F.distance)
FROM flights F
WHERE F.origin = 'London' AND F.destination = 'New York' )
```

correct SELECT **[1]**, correct FROM **[1]**, correct WHERE **[1]**, correct sub-query **[3]**

(6 marks)

(e) Find the names of employees who are certified to fly aircraft with cruising range longer than 1000 kilometres.

```
SELECT E.emp-name
FROM employees E, certified C, aircraft A
WHERE C.airID = A.airID AND E.empID = C.empID
GROUP BY E.empID, E.emp-name
```

HAVING (A.cruising-range > 1000)

correct SELECT **[1]**, correct FROM **[1]**, correct WHERE **[1]**, correct GROUP BY **[1]**, correct HAVING **[2]**

(6 marks)

(Total 30 marks)