UCL Computer Science
Alternative Assessment coursework

Paper details

Academic year:	2020/21
Module title:	Database and Information Management Systems
Module code:	COMP0022
Exam period:	Main summer examination period
Duration:	24 hours
Deliveries for which intended:	A7P (taught postgraduate, level 7)
Cohorts for which intended:	2020/21; 2019/20

Instructions

This coursework is open-book and open-note. You may consult slides and other materials. You may NOT consult with other students or other students' work, or share your work with other students.

There are THREE questions in total. Answer all questions.

A maximum of 100 marks is available. The marks available for each part of each question are indicated in square brackets.

Submit your answers as a single PDF file. Any handwritten answers should be scanned and compiled according the guidance provided by the UCL Examinations Office.

Answer ALL questions.

1) [Database design] Suppose you are creating a site for local book-sharing: users sign up and list books they are willing to lend or give away.

List of features:

- Users can specify books as being loans or giveaways. Loans can (but do not necessarily) come with dates for return. Users can add loan deadlines after the fact and also issue nudges if the loan is late.
- Loans remain in the system while they are active. Giveaways become inactive once transferred (although the new owner can choose to relist the book under whatever settings they like).
- Book listings by default have standard information about the book (title, author, description/synopsis, edition (if applicable), etc.). In addition, the lister can choose to add their own custom description/commentary.
- Users can browse listings posted by users that live nearby (in the same town, county, etc.).
- Users can friend other users to get notifications of new books that user is offering.
- Users can rate other users for reliability.

(a) Propose a relational schema (a set of tables and attributes) to support the function of this site. Justify your proposed schema by including intermediate work; using normalization forms to argue why the tables are easy to update while maintaining consistency; explaining your choices and assumptions; etc. Suggested length of explanation: 200-500 words.

[25 marks]

(b) Write two SQL queries (using the schema you gave) that would support the function of this site.

[8 marks]
[Total for Question 1: 33 marks]

2) [Transactions]

(a) Consider a simplified version of the Shop database used in class exercises:

Customers (custID, firstname, familyname, town, state)
Items (itemID, description, unitcost, stocklevel)
LineItems (orderID, itemID, quantity, despatched)
Orders (orderID, custID, date)

Give an example of two simultaneous/interleaved transactions that could put this database into an inconsistent state and explain how an inconsistent state results.

[12 marks]

(b) Consider the following locking scheme:

Transaction #1:	Transaction #2:
Lock_write(x) Read(x) Lock_read(y) Read(y) Write(x) Unlock_write(x) Unlock_read(y)	Lock_write(y) Read(y) Write(y) Lock_write(x) Read(x) Write(x) Unlock_write(x) Unlock_write(y)

(i) Does the above locking scheme guarantee serializability, or is it possible for a non-serializable schedule to be created when running Transaction #1 and Transaction #2 simultaneously? Justify your answer.

(ii) Is it possible for a deadlock situation to arise? Justify your answer.

[10 marks]

(c) Give an example of how concurrency can trade off with storage space, and give an example of how concurrency can trade off with consistency.

[12 marks]
[Total for Question 2: 34 marks]

3) [Distributed databases] Consider the following schema for a database that assists the function of a group of grocery stores:

Item (itemCode, name, desc)
Store (storeNo, address, city, manager)
Inventory (storeNo, itemCode, quantity, price)
Transactions (storeNo, itemCode, date, price, quantity)
Personnel (staffNo, fName, lName, storeNo, position, startDate)
CustomItems (storeNo, itemCode, name, desc)

"Item" contains a list of item codes associated with certain items, which are the same for all stores. In addition, individual stores can add custom items not contained in "Item" as long as it doesn't conflict with existing shared item codes. Each store tracks its own inventory. Item prices can change from day to day, from store to store, and even per transaction (due to coupons, etc.). "Transactions" keeps a summary of the quantity of items sold per store and at what price.

(a) The stores would like to distribute their database to multiple sites managed by individual stores. Propose a fragmentation / replication / distribution scheme for this database and justify your choices, giving at least one consideration that influenced your design.

[11 marks]

(b) Say that you have one site that is in charge of analyzing trends in transactions in stores over time. This site makes use of the Transactions table, but that table is too large to feasibly store on a single computer. Propose a fragmentation / replication / distribution scheme to support this site's functions and justify your choices, giving at least one consideration that influenced your design.

[11 marks]

(c) Under the schemes proposed above, briefly describe what work would be involved in performing the following operations:

(i) Adding new products to the standard catalogue used by all stores.

(ii) Recording a new set of purchases by a customer.

[11 marks]
[Total for Question 3: 33 marks]

END OF PAPER