CSI2132/CSI2532 Final Examination 2013

Professors: Herna Viktor and Fadi Malek

Date: April 20th, 2013

Duration: 3 hours

Total: 70

Marking Guidelines

Instructions:

1. This is a closed book examination.
2. No calculators or any other electronic devices are allowed.
3. You are allowed to bring along one letter-size "cheat sheet", printed or written on both sides.
4. Answer the questions in the space provided on the examination paper.

Name	
Student Number	

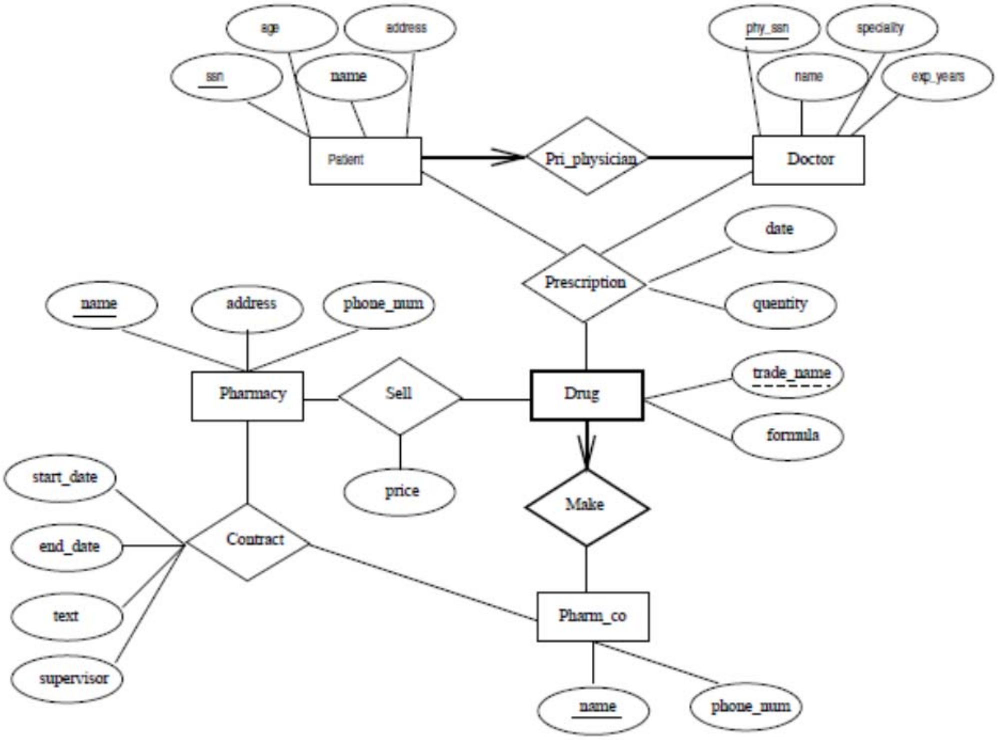
Good luck! Bonne Chance!

This examination contains 5 questions.

Question 1: EER Diagrams		12
Question 2: Relational algebra and Relational calculus		10
Question 3: Relational model and SQL		15
Question 4: Normalization		16
Question 5: Storage and Indexing		17
TOTAL		**70**

Question 1: EER Diagrams [12 marks]

Consider the following EER diagram of the database of the BetterDrugs chain of pharmacies. This database captures the information about the prescription medicine that doctors prescribe to patients that are then sold by the pharmacies. The drugs are manufacturer by pharmaceutical companies that supply them to the pharmacies.



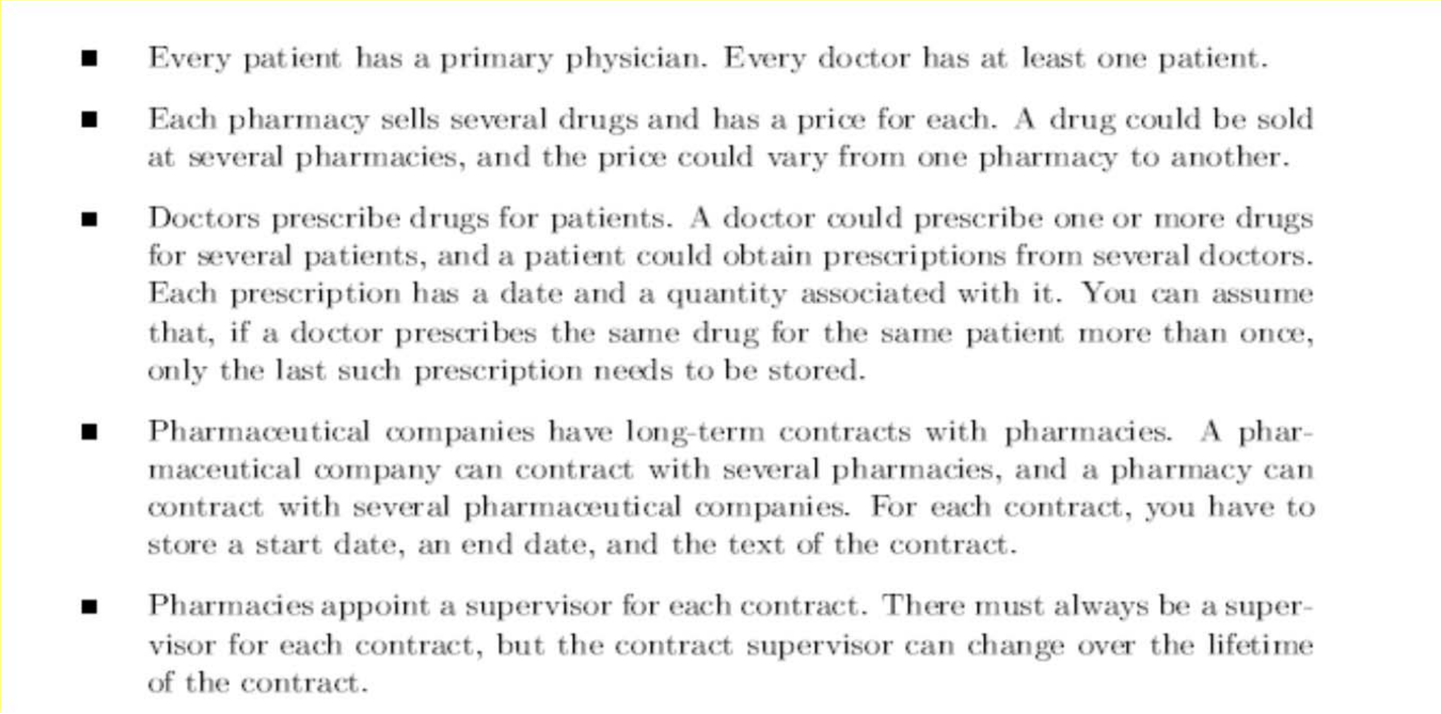
The CEO of BetterDrugs asks you to review this EER diagram design with her.

a. In the current design, would a doctor be able to prescribe the same drug for the same patient more than once? Motivate your answer. [2]

Answer: No, he is not able to do so; we can only store the last prescription for a drug. We have to create a new entity call Prescription-Date and make Prescription a 4-way relationship that involves the additional entity.

b. Write the *exact* business rules that this EER diagram capture. That is, for every relationship, you are asked to identify the minimum and maximum times that the two entities participate in the relationship and to write these rules in the form of full sentences. [10]

Answer:



- Every patient has a primary physician. Every doctor has at least one patient.
- Each pharmacy sells several drugs and has a price for each. A drug could be sold at several pharmacies, and the price could vary from one pharmacy to another.
- Doctors prescribe drugs for patients. A doctor could prescribe one or more drugs for several patients, and a patient could obtain prescriptions from several doctors. Each prescription has a date and a quantity associated with it. You can assume that, if a doctor prescribes the same drug for the same patient more than once, only the last such prescription needs to be stored.
- Pharmaceutical companies have long-term contracts with pharmacies. A pharmaceutical company can contract with several pharmacies, and a pharmacy can contract with several pharmaceutical companies. For each contract, you have to store a start date, an end date, and the text of the contract.
- Pharmacies appoint a supervisor for each contract. There must always be a supervisor for each contract, but the contract supervisor can change over the lifetime of the contract.

Question 2: Relational algebra and relational calculus [10 marks]

Consider the following partial relational schema concerning the *National Ballet of Canada (NBC)* and the ballet dancers that work there. This schema contains four relations. The Dancer relation stores in the personal about the Dancers, including their current Age and their Rating, which is a number between 1 and 5. We also store their monthly salary. The Dance relation keeps the information about which Ballets a Dancer danced in, together with the Season (the calendar year) and the Role (e.g. a dancer danced Romeo in Romeo and Juliet). Note that a Dancer may dance in many Ballets, and Ballet includes many Dancers. We record the total duration of a Ballet. There are many Performances of a specific Ballet and a Performance consists of only one Ballet. For each Performance, we keep the total cost incurred by the NBC, which includes expenses such as the rental of the venue, travel costs, hotel stays, etc. We also record the city in which this performance took place, together with the venue (such as the National Arts Center or Scotiabank Place.)

```
Dancer(Did, StageName, Age, Rating, Salary)
Dance(Did, Balletid, Season, RoleName)
Ballet(BalletID, Title, Duration)
Performance(PerformanceID, Cost, Venue, City, BalletID)
```

a. Provide the relational algebra expression to find the stage names and ages of all Dancers who are older than 21 and have never danced in the ballet with title “Carmen”. [4]

Answer: SN stands for stagename

$$\rho\,(\text{Carmen},\ \pi_{SN,\,Age}\ (\text{Dancer} \bowtie_{DID} \text{Dance} \bowtie_{BID} (\sigma_{Title='Carmen'}\ \text{Ballet})))$$

$$\rho\ (\text{All Dancers},\ \pi_{SN,\,Age}\ \text{Dancer})$$

$$\text{Result} \longleftarrow \text{All Dancers} - \text{Carmen}$$

b. Provide the relational algebra to find the stage names and ratings of the Dancers who have danced in all the 2013 season’s performances of the Ballet with title “The Nutcracker”. [4]

Answer:

$$\rho\ (\text{NC Perform},\ \pi_{PerfID}\ (\sigma_{title='Nutcracker'}\ \text{Ballet} \bowtie_{BalletID} \text{Performance}))$$

$$\rho\ (\text{Dance Perform},\ \pi_{DID,\,PerfID}\ (\text{Dancer} \bowtie_{DID} \text{Dance} \bowtie_{BalletID} \text{Ballet} \bowtie_{BalletID} \text{Performance}))$$

$$\rho\ (\text{Dancer IDs},\ \text{Dance Perform} / \text{NC Perform})$$

$$\text{Result} \longleftarrow \pi_{sname,\,rating}\ (\text{Dancer} \bowtie_{DID} \text{Dancer IDs})$$

c. Explain, by means of your own example, what an unsafe query is and explain why it is important to disallow such queries. [2]

Answer:

An unsafe query in relational calculus is a query that has an infinite number of results. For example

{D | ˥(D € (Dancers)}

The query is for all things that are not in Dancers which is of course everything else. Clearly there is an infinite number of answers, and this query is unsafe. It is important to disallow unsafe queries because we want to be able to get back to users with a list of all answers after a finite amount of time.

Question 3: The Relational Model and SQL programming [15 marks]

Reconsider the following partial relational schema concerning the National Ballet of Canada (NBC) and the ballet dancers that work there.

```
Dancer(Did, StageName, Age, Rating, Salary)
Dance(Did, Balletid, Season, RoleName)
Ballet(BalletID, Title, Duration)
Performance(PerformanceID, Cost, Venue, City, BalletID)
```

a. Explain, by means of your own example against the NBC database, how referential integrity is enforced in a relational database. [3]

Answer:

We need to use foreign keys.

Give your own example of using a foreign key to enforce referential integrity.

For example, in Performance:

```
FOREIGN KEY BalletID REFERENCE Ballet ON DELETE RESTRICT
                                      ON UPDATE CASCADE
```

b. Consider the following query: “Find the names of the Dancers with a higher salary that all Dancers that are younger than 30.” The following query attempts to obtain the answer to this question. Determine whether this query will produce correct results and motivate your answer. [3]

```
SELECT D.Stage-name
FROM Dancer D
WHERE D.salary > ANY (SELECT D2.salary
                      FROM Dancer D2
                      WHERE D2.age < 30)
```

Answer: This query is not correct. It returns the names of Dancers with a higher income than at least one dancer with age < 30. So, it will not necessarily give the correct answer. In particular, if all the dancers are at least 30 years old, it will return the empty set. This is because of the use of ANY, rather than NOT EXISTS.

c. Write the SQL statement to find the stage names of the Dancers whose salaries are less that the Cost of the cheapest performance that took place in Ottawa. [3]

Answer:

```
SELECT DISTINCT D.stagename
FROM Dancer D
WHERE D.salary <  (SELECT MIN(P.cost)
                   FROM Performance P
                   WHERE P.city = “Ottawa”)
```

d. Write the SQL statement to identify, for each Ballet with duration longer than 3 hours, the name of the Ballet and the average Age of all Dancers who danced in this Ballet. [6]

Answer:

```
SELECT Temp.title, Temp.AvgAGE
FROM  (SELECT B.bid, B.title as title, Avg(D.Age) as AvgAGE
      FROM Dancer D, Dance A, Ballet B
      WHERE D.did = A.did
      AND A.bid = B.bid
      AND B.duration > 3   // 3 hours or 180 minutes
      GROUP BY B.bid, B.title) AS TEMP
```

Question 4: Normalization [16 marks]

Consider the following relation concerning the costumes that Ballet Dancers wear during Performances:

`Costume(CostumeID, Year, Price, Supplier, Color)`

This relation may be abbreviated as Costume(C, Y, P, S, R) and we may assume that the following set of functional dependencies hold: C → S, CY → P and S → R.

(a) This relation may contain redundant data and is thus susceptible to i) insertion, ii) deletion and iii) update anomalies. Explain, by means of your own examples, what these three anomalies refer to. [3]

Answer:

See p607 of textbook. You need to explain what insertion, deletion and update anomalies are and give your own examples.

(b) Identify the candidate key for this relation, showing the steps you followed using Armstrong's Axioms. [4]

Answer: CY is a candidate key

Since CY → P and C → S, by augmentation CY → S

Since S → R, by transitivity C → S, S → R, which gives C → R. By augmentation CY → R

(c) Determine the highest normal from in which this relation is and motivate your answer by referring to INF, 2NF, 3NF and BCNF. [4]

Answer: The highest normal form in 1NF. It is not even in 2NF, due to partial dependencies.

Not, 3NF since transitive dependencies; not BCNF because not every determinant is a candidate key.

(d) Explain what de-normalization is and give an example where this may be needed. [2]

Answer: De-normalization means that we "go back" to an earlier NF, e.g. from BCNF to 3NF. We usually do this for performance reasons. Say we have a query that is very slow, ad that JOINS two tables. Then, by having some redundancy, we may get better results. A typical example is postal codes. This is also often used in data warehousing.

(e) Suppose that Costume is decomposed into C1(C, Y, P) and C2(C, S, R). Determine whether this decomposition is lossless and motivate your answer. [3]

Answer: See p.619 of textbook.

Yes, this decomposition is lossless since we will get the original relation back when we join over C. (Students could also have computed the attribute closure.)

Question 5: Storage and Indexing [17 marks]

Consider, for the last time, the following partial relational schema concerning the National Ballet of Canada (NBC) and the ballet dancers that work there.

```
Dancer(Did, StageName, Age, Rating, Salary)
Dance(Did, Balletid, Season, RoleName)
Ballet(BalletID, Title, Duration)
Performance(PerformanceID, Cost, Venue, City, BalletID)
```

Suppose that the Dancer relation contains the following data.

Did	StageName	Age	Rating	Salary
100	Joe	20	1	100,000
101	John	35	6	30,000
102	Ann	18	3	80,000
103	Sue	28	2	90,000
104	Sue	32	1	100,000
105	Suzy	22	8	20,000
106	Annie	19	1	100,000
107	Maple	31	6	30,000
108	Louis	27	7	25,000
109	Eric	25	2	90,000
110	Joe	22	6	30,000

a. You decide to create a search key (index) on the Age attribute of the Dancer relation. Explain the three alternative ways in which the data entries in the index may constructed. [3]

Answer: See p.276 of text book
Alternative 1 is the actual record
Alternative 2 is <k, rid> pair(s)
Alternative 3 is a <k, list of rid> pair

b. Suppose that the Performance relation has been created using variable length fields for the City and Venue attributes. Discuss two issues that may arise when attempting to modify a tuple/record. [2]

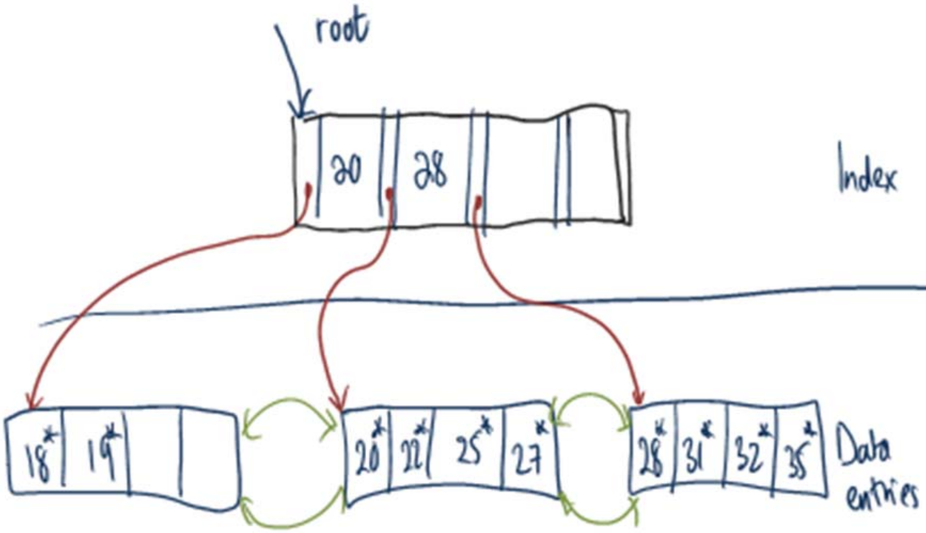
Answer: See p.332 of text book

- **The overhead in terms of managing this type of data structure may be prohibitive, especially if we have a large number of inserts and deletes.**
- **Difficult to modify; say the venue changes from "NAC" to "Scotiabank Place". In this case, your records will not fit in the space that has been allocated and you need to move it.**

c. You are required to construct a B+ tree with order 2, using the Age attribute as search key. Show the resulting tree after inserting the records in the order that that they appear in the table above. That is, you **<u>should not</u>** use the bulk loading algorithm. [5]

Answer:

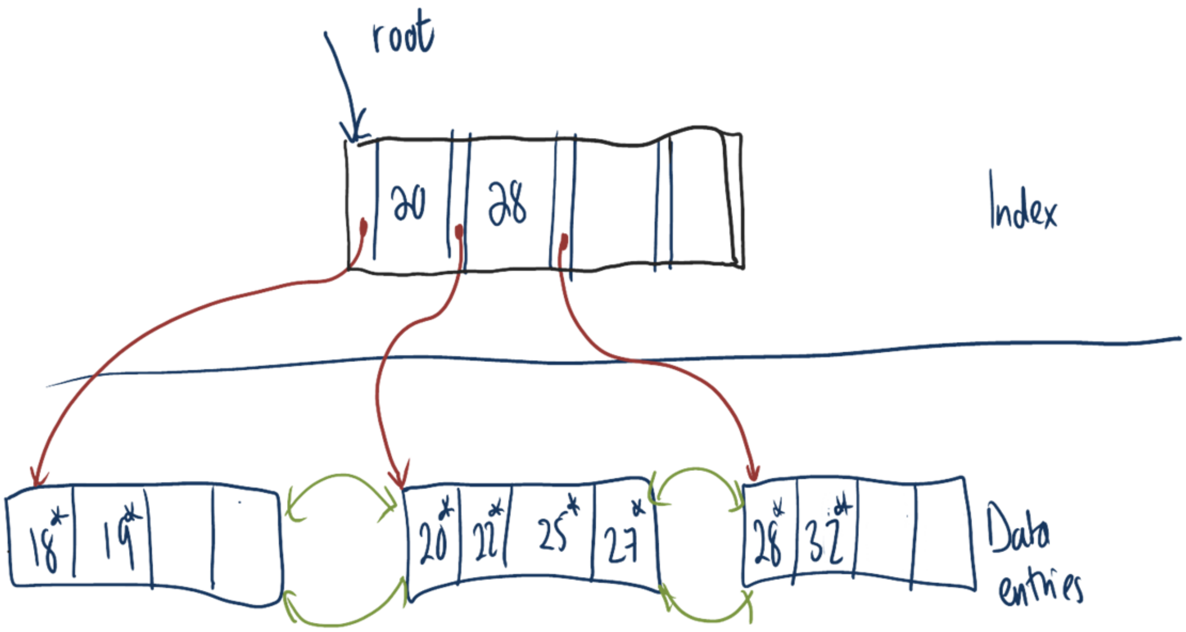
Since the order of the tree is 2, the number of entries in the B+ tree is 4 and we split when it is full.



d. Suppose that Suzy, Maple and John resign and that their data are deleted from the NCB database. Show the resultant B+ tree once they are removed from the database. [3]

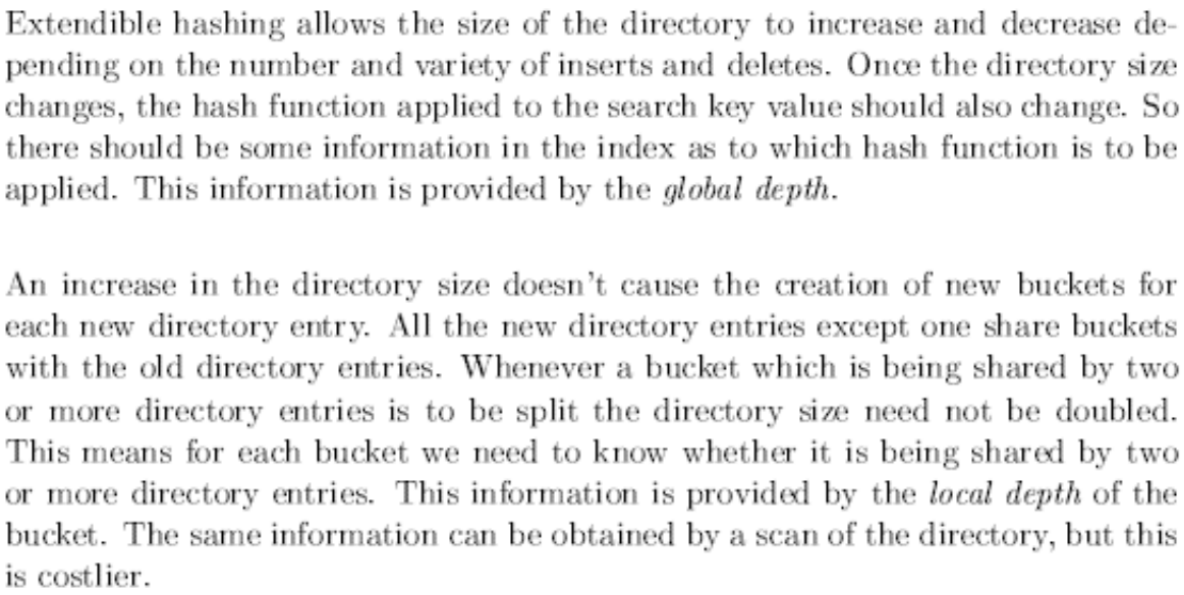
Answer:

Note that we do not delete the node for search key 22. Rather, we need to update the data entry by removing the link to the record identifier for Suzy.



e. Explain i) *how* the Extendible Hashing algorithm makes use of global and local depths and ii) *why* this algorithm is considered to be a computationally efficient method. [4]

Answer: See p.377 of textbook.



Extendible hashing allows the size of the directory to increase and decrease depending on the number and variety of inserts and deletes. Once the directory size changes, the hash function applied to the search key value should also change. So there should be some information in the index as to which hash function is to be applied. This information is provided by the *global depth*.

An increase in the directory size doesn't cause the creation of new buckets for each new directory entry. All the new directory entries except one share buckets with the old directory entries. Whenever a bucket which is being shared by two or more directory entries is to be split the directory size need not be doubled. This means for each bucket we need to know whether it is being shared by two or more directory entries. This information is provided by the *local depth* of the bucket. The same information can be obtained by a scan of the directory, but this is costlier.