CS–A1150 Databases

Exam Feb 22nd, 2019 **No calculators or extra material allowed**

Students whose mother tongue is not Finnish may use a dictionary, if it does not contain any markings. Those students may also obtain both Finnish and English/Swedish exam sheet, if they want.

1. a) (8 p) Construct an E/R diagram for a chain of shops and its credit customers (i.e. customers who can pay their purchases using a special credit card and pay their bills according to certain rules) based on the following information. Use the notation used in the course text book and underline the key attributes.

   Each customer has a unique ID, a name and an address. A customer can have at most one charge account (i.e. the account in which the purchases and payments are registered). However, the same charge account can have several owners. For example, a couple can have a common charge account, but each of them is registered as a separate customer. The information about charge account includes, in addition to its owners, its unique ID, balance, interest rate, credit limit and minimum monthly payment.

   The chain of shops consists of several shops. Information on each shop includes a unique ID, a name and an address. If the customer visits a certain shop often enough, he / she is registered as a regular customer of this shop. The same customer can be a regular customer of several shops, but he / she has only one charge account, which he / she can use in all shops of this chain.

   Each purchase made using the charge account is registered by a unique ID. However, these IDs are only unique in a certain shop, not in the whole chain. The information on the purchase also includes the customer, the shop, the date and the value of the purchase. The system also registers all payments the customer has made to her / his charge account. The information on a payment includes the date, the sum, the unique ID (index number) and the charge account.

   b) (2 p) Convert the E/R diagram from part (a) into relations. Write the relation schemas and underline the names of the key attributes.

2. Consider the following database schema, which contains information about the employees of an airline company. The relation `Aircraft` contains information of the different types of the aircraft the company has. The attribute `cruisingrange` tells the cruisingrange in kilometers and the attribute `seats` the number of passenger seats in the airplane. The relation `Employee` contains both pilots and other kinds of employees as well. Each pilot is certified for at least one aircraft and only pilots are certified to fly. The value of attribute `salary` is the monthly salary in euros. The relation `Bonus` contains information about the yearly bonuses paid to the employees. The attribute `amount` is the amount of an individual bonus in euros. Each employee may obtain at most one bonus during a certain year.

   ```
   Aircraft(id, aname, cruisingrange, seats)
   Employee(ssNo, name, yearofbirth, salary)
   Certified(employeeSsNo, aircraftId)
   Bonus(employeeSsNo, year, amount)
   ```

   You may assume that the attributes of the tuples do not have NULL values.

   Write the following SQL queries:

   a) (2 p) The ssNos and names of those employees who have received a bonus of over 3000 euros in Year 2017.

   b) (2 p) The ssNos and names of those pilots who are certified to fly at least one aircraft having over 250 passenger seats.

   c) (2 p) The ssNos and names of the employees who are not pilots, but whose salary is over 5500 euros.

   d) (2 p) We want to find the pilots (not other employees) whose total bonus from various years is over 25000 euros. The sum is calculated separately for each pilot. For each pilot fullfilling the conditions, the query must produce ssNo, name and the maximum bonus (not the sum) this pilot has received.

   Write the following queries as expressions of the relational algebra:

   e) (2 p) The ssNos and names of those **pilots** who have received a bonus of over 3000 euros in Year 2017.

   f) (2 p) The ssNos and names of those pilots who are certified to fly at least two different aircraft types with over 200 passenger seats.

TURN THE PAPER, PLEASE!

3. Consider a relation R with schema $R(A, B, C, D, E)$ and functional dependencies $A \rightarrow B$, $B \rightarrow C\ D$, and $E \rightarrow B$.

   a) (1 p) Explain why this relation is not in Boyce-Codd normal form (BCNF).

   b) (6 p) Decompose the relation using the BCNF decomposition algorithm taught in this course and in the text book. Give a short justification for each new relation. Continue the decomposition until the final relations are in BCNF. Explain why the final relations are in BCNF.

4. a) (2 p) What does it mean that the transactions are serializable? Give an example of a problem which can arise, if the transactions are not serializable.

   b) (2 p) What does it mean that the transactions must have the *consistency* property? Give a short example.

   c) (3 p) What are the differences between the following isolations levels of the transactions: READ COMMITTED, REPEATABLE READ, and SERIALIZABLE? Why is the serializability not always required?

5. a) The existence of an index on an attribute usually speeds up queries containing conditions for this attribute. Why is it not sensible to build indexes on every attribute of every relation?

   b) Assume that very many tuples of a relation typically have the same value for a certain attribute, like year of birth in a relation containing students of a university. Is it profitable to build an index on this attribute (justify your answer)?

   c) Consider the database given in Problem 2. Assume that the relation `Employee` is stored over 180 disk pages (i.e. the relation requires 180 disk pages) and on the average, eight employees of the company have the same year of birth. The employees have been stored in the table in random order. Consider the queries which search for all employees having a certain year of birth. How much can the query be sped up, if the relation `Employee` has an index on attribute `yearofbirth`? A rough estimate is enough, but justify your answer. You do not have to consider any other operations except the described query.