CHALMERS UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering

Examination in Databases, TDA357/DIT620

Friday 9 May 2014, 13:00-17:00

Examiner:	Graham Kemp (telephone 772 54 11, room 6475 EDIT) The examiner will visit the exam room at 14:00 and 16:00.
Results:	Will be published by the end of May at the latest.
Exam review:	See course web page for time and place: http://www.cse.chalmers.se/edu/year/2013/course/TDA357/HT2013/
Grades:	Grades for Chalmers students (TDA357) are normally determined as follows: ≥ 48 for grade 5; ≥ 36 for grade 4; ≥ 24 for grade 3. Grades for GU students (DIT620) are normally determined as follows: ≥ 42 for grade VG; ≥ 24 for grade G.
Help material:	One A4 sheet with hand-written notes. You may write on both sides of that sheet. If you bring a sheet, it must be handed in with your answers to the exam questions. English language dictionaries are allowed.

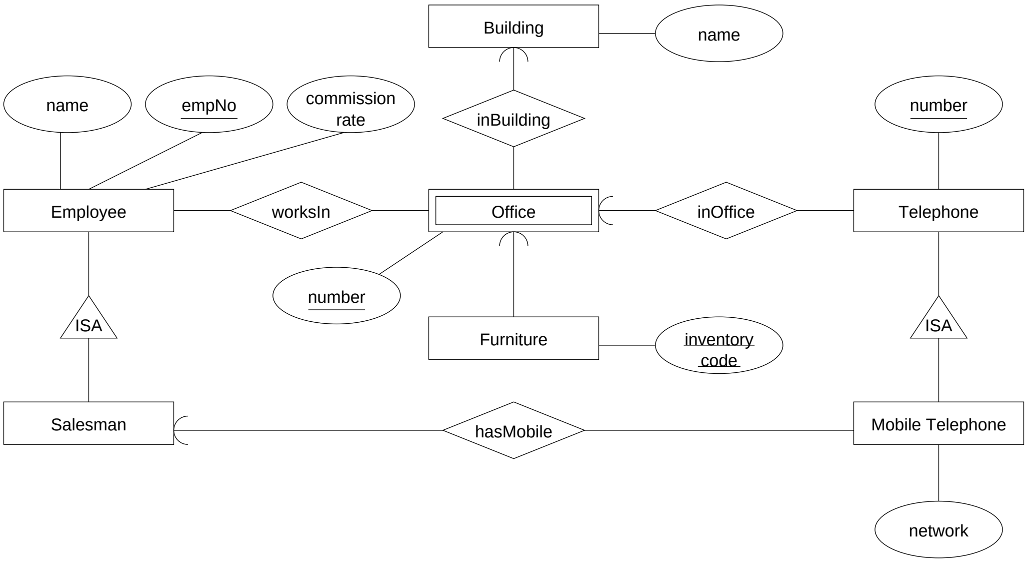
Specific instructions:

- Please answer in English where possible. You may clarify your answers in Swedish if you are not confident you have expressed yourself correctly in English.
- Begin the answer to each question on a new page.
- Write clearly; unreadable = wrong!
- Fewer points are given for unnecessarily complicated solutions.
- Indicate clearly if you make any assumptions that are not given in the question.
- Write the page number and question number on every page.

Question 1. Consider the following domain description.
10 p

> A company wants to use a database system to manage information about its employees, buildings and offices. Each employee is identified within the company by an employee number. Employee names are not guaranteed to be unique. The company has several buildings on its site and each building has a unique name. The offices within each building are identified by an office number, however some offices in different buildings happen to have the same office number. There can be several telephones and items of furniture in each office. Telephones are identified by their number, and items of furniture have unique inventory codes. Each office is shared by several employees, and some employees (who have more than one role in the company) will work in several different offices during a normal working week. Some salesman have one mobile telephone that is provided by the company, but not all salesmen have a company mobile telephone. Each mobile telephone is connected to a network. Different mobile telephones can be connected to a different network. Unlike other employees, the salesmen have a commission rate.

The following Entity-Relationship (E-R) diagram attempts to model this domain. The diagram has several errors.



a) List 5 errors in the E-R diagram.
Draw a corrected E-R diagram. (5p)

b) Translate the corrected E-R diagram into a set of relations, clearly marking references and keys. (5p)

Question 2.
12 p

Suppose we have relation $R(A, B, C, D)$ and functional dependencies $AB \to D$, $BC \to A$, $BC \to D$, $D \to B$.

a) By considering the closures of all subsets of attributes, find **all** non-trivial FDs, all superkeys and all keys.
(4p)

b) Considering the FDs given above, and those found in your answer to part (a), state which FDs violate BCNF.
(2p)

c) i) Decompose relation R to BCNF. Show each step in the normalisation process, and at each step indicate which functional dependency is being used.
ii) Does the decomposition in your answer to part (i) give *dependency preservation*? Explain your answer.
(4p)

d) For each of the four FDs given at the top of this question, state whether it violates third normal form (3NF) and give a reason to motivate your answer.
(2p)

Question 3.
10 p

An estate agent uses a relational database to manage information about properties for sale, and occasions when prospective buyers can view properties. This database has the following relations:

$Properties(\underline{ref}, address, area, guidePrice)$
$Viewing(\underline{property}, \underline{viewingNumber}, year, month, day, hour)$
$Viewers(\underline{property}, \underline{viewing}, \underline{person})$

Each property is identified by a unique reference number. The address, living area (in m^2) and guide price (in SEK) of each property for sale are stored in the database.

Relation *Viewing* contains information about the viewing occasions. The viewing occasions for a particular property will be numbered (e.g. 1, 2, and so on), and the year, month (1-12), day (1-31) and hour of the viewing are stored.

Unique identifiers of people who have registered for particular viewings are stored in attribute *person* in relation *Viewers*.

a) Suggest references for these relations.
Write SQL statements that create these relations with constraints in a DBMS.
(4p)

b) A person can attend at most one viewing at any particular date and time.
Write an assertion that checks this.
(2p)

c) At most 20 people can view a property on any viewing occasion. If someone tries to register for a viewing that is already full, then they should be registered instead for another viewing occasion for that property that has an available place. If all viewing occasions for that property are full, then a new viewing occasion should be created on the same day as the viewing occasion for that property with the highest *viewingNumber*, but starting one hour later. (You can assume that the number of viewing registrations is fairly small, and so you can ignore the complication of filling a whole day and having to schedule viewings on the next day.)
Write a trigger that implements this.
(4p)

Question 4. Assume the same relations as in Question 3:
6 p

Properties(ref, address, area, guidePrice)
Viewing(property, viewingNumber, year, month, day, hour)
Viewers(property, viewing, person)

a) Write a relational algebra expression that finds the identifiers of people who will view properties on 9 May 2014, together with the property reference numbers and addresses.

(3p)

b) Write a relational algebra expression that finds, for each property with a guide price greater than 5000000, the property's reference number and the total number of people who have registered to view that property (remember that there could be several viewing occasions for each property).

(You may ignore those properties for which no viewings have been scheduled, or for which nobody has registered to attend a viewing.)

(3p)

Question 5. Assume the same relations as in Question 3:
10 p

Properties(ref, address, area, guidePrice)
Viewing(property, viewingNumber, year, month, day, hour)
Viewers(property, viewing, person)

a) Write an SQL query that finds the identifiers of people who will view properties on 9 May 2014, together with the property reference numbers and addresses.

(3p)

b) Write an SQL query that finds people who viewed properties with a guide price greater than 5000000 in both 2012 and 2013.

(3p)

c) Create a view *NumViewers(property, numViewers)* where *property* is a reference number and *numViewers* is the number of people who have registered to view that property (remember that there could be several viewing occasions for each property). Attribute *numViewers* should have the value 0 if nobody has registered to view that property, or if no viewing occasions have been scheduled for that property.

(4p)

Question 6. Assume the same relations as in Question 3:
6 p

Properties(<u>ref</u>, address, area, guidePrice)
Viewing(<u>property</u>, <u>viewingNumber</u>, year, month, day, hour)
Viewers(<u>property</u>, <u>viewing</u>, <u>person</u>)

a) Suppose relation *Viewers(property, viewing, person)* is stored in 20 disc blocks. Each property has records stored in 6 disc blocks, on average. Similarly, on average, each person has records stored in 4 disc blocks.

Suppose that two kinds of task are performed on this relation:

- task 1: inserting a new row;
- task 2: finding the property reference numbers of properties viewed by a given *person*.

For each of these tasks, state how many disc block transfers will be needed if:

i) there are no indexes;

ii) there is an index on *person* (assume that this index fits into a single disc block).

iii) there is an index on *person* and also an index on *property* (assume that the index on *property* also fits into a single disc block).

Suppose that 50% of operations performed on this relation are task 1 (inserting new rows) and 50% are task 2 (finding the property reference numbers for given people).

iv) Discuss whether it would be better to have an index on *person*, indexes on both *person* and *property*, or to have no indexes.

(4p)

b) Suppose transaction T1 inserts a new row into relation *Viewers*.

Suppose transaction T2 counts the number of properties viewed by a particular person, prints the result, then reads some data from the *Properties* relation, then counts and prints the number of properties viewed by the same person.

Give a reason why the two numbers printed when transaction T2 is run could be different to each other. You should mention isolations levels in your answer.

(2p)

Question 7. Consider the following piece of XML:
6 p

```
<Hospital>
  <Patient name="smith" pid="p001">
    <Test pid="p001" time="07:00" temp="36.8" heartRate="75"/>
    <Test pid="p001" time="11:00" temp="36.9" heartRate="77"/>
    <Test pid="p001" time="15:00" temp="36.7" heartRate="74"/>
  </Patient>
  <Patient name="jones" pid="p002">
    <Test pid="p002" time="10:00" temp="36.8" heartRate="66"/>
  </Patient>
  <Patient name="green" pid="p003">
    <Test pid="p003" time="17:00" temp="36.8"/>
    <Test pid="p003" time="18:00" heartRate="60"/>
  </Patient>
</Hospital>
```

a) Write a Document Type Definition (DTD) for the XML that is given above.
(2p)

b) Write an XPath expression that finds Patient elements where the patient has a test with a heart rate value lower than 70.
(1p)

c) Assuming that the XML shown above is in file *hospital.xml*, write an XQuery expression that constructs the following result:

```
<Hospital>
  <Patients>
    <Patient name="smith" pid="p001"/>
    <Patient name="jones" pid="p002"/>
    <Patient name="green" pid="p003"/>
  </Patients>
  <Tests>
    <Test pid="p001" time="07:00" temp="36.8" heartRate="75"/>
    <Test pid="p001" time="11:00" temp="36.9" heartRate="77"/>
    <Test pid="p001" time="15:00" temp="36.7" heartRate="74"/>
    <Test pid="p002" time="10:00" temp="36.8" heartRate="66"/>
    <Test pid="p003" time="17:00" temp="36.8"/>
    <Test pid="p003" time="18:00" heartRate="60"/>
  </Tests>
</Hospital>
```

(3p)