CHALMERS UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering

Examination in Databases, TDA357/DIT620

Tuesday 18 December 2007, 14:00-18:00, Building M

Examiner:	Graham Kemp (telephone 772 5411, room 6475 EDIT) The examiner will visit the exam room at 15:00 and 17:00.
Results:	Will be published by the middle of January at the latest.
Exam review:	Friday 21 December 2007 (see course web page for time and place).
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. That sheet must be handed in with your answers to the exam questions. English language dictionaries are allowed.

Specific instructions:

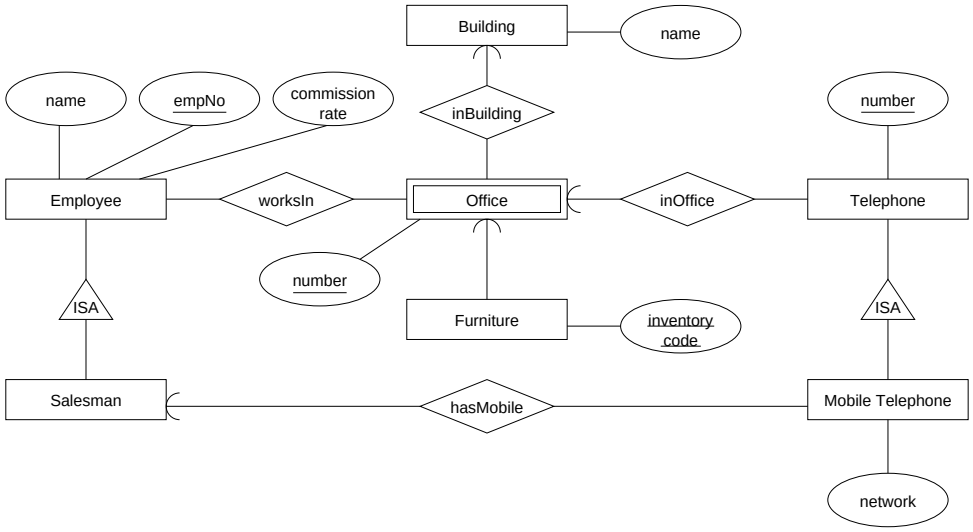
- Please answer in English where possible. You may clarify your answers in Swedish if you are not confident you have expressed yourself correctly in English.
- Begin the answer to each question on a new page.
- Write clearly; unreadable = wrong!
- Fewer points are given for unnecessarily complicated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

Question 1. 12 p

a) Consider the following domain description.

Each employee is identified within a 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. 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.



i) List 5 errors in the E-R diagram.
Draw a corrected E-R diagram. (6p)

ii) Translate the corrected E-R diagram into a set of relations, clearly marking references and keys. (3p)

b) Each protein has a name and a UniProt identifier. A UniProt identifier is a string that uniquely identifies a protein. Enzymes are a class of proteins that catalyse chemical reactions. Each enzyme has an "EC number" that describes the reaction that the enzyme catalyses. Proteins that are not enzymes do not have EC numbers.

i) Draw an E-R diagram that correctly models this domain.
Translate this E-R diagram into a set of relations, clearly marking references and keys. You should use the "E-R approach" for translating the ISA relationship. (2p)

ii) Name one other approach for translating an ISA relationship to relations.
Show the relations that are produced when using this approach. (1p)

Question 2. 12 p

a) Suppose we have relation $R(A, B, C, D)$ and functional dependencies $A \to B$, $B \to C$, $B \to D$.
By considering the closures of all subsets of attributes, find **all** non-trivial FDs, superkeys and keys. (4p)

b) Information about student, courses, exam marks and tutorials is stored in the relation:
$S(studentNo, name, course, grade, year, period, teacher, examDate, mark, day, hour, tutor)$.
Each course is held in one or more periods each year. Only one tutorial takes place at one time for each course in each period. The grade awarded to a student is determined by their mark in the exam.

i) Identify all functional dependencies that you expect to hold. (2p)

ii) Decompose this relation into Boyce-Codd normal form (BCNF). (2p)

c) Consider the following relation that holds information about chains of book shops:
$B(branch, chain, title, price)$

branch	**chain**	**title**	**price**
Göteborg	Books-R-Us	Genome	100
Göteborg	Books-R-Us	Collapse	150
Stockholm	Books-R-Us	Genome	100
Stockholm	Books-R-Us	Collapse	150
Malmö	ABC-Books	Genome	120
Malmö	ABC-Books	Bioinformatics	200
Stockholm	ABC-Books	Genome	120
Stockholm	ABC-Books	Bioinformatics	200

i) What multi-valued dependencies hold for this relation?
If a tuple with values (Stockholm,ABC-Books,Collapse,150) is added to this relation, what other tuple(s) would have to be added to this relation for these multi-valued dependencies to continue to hold? (2p)

ii) Decompose this relation into 4th normal form.
Show the contents of the resulting relations. (2p)

Question 3. Consider the following relations that contain tram timetable information:

4 p

$Lines(lineNo, stopNumber, stopName)$

$Journeys(line, direction, stopNumber, hour, minute)$
$line \rightarrow Lines.line$
$stopNumber \rightarrow Lines.stopNumber$

On each line there are two directions of travel: generally from north to south ("NtoS") and generally from south to north ("StoN").

Relation *Lines* contains a list of the stops on that line; the *stopNumber* is the stop's position on the line when the stops are ordered in the direction "NtoS". Thus, for example, Brunnsparken could have different stop numbers on different lines.

a) Write a relational algebra expression that computes the line numbers of all lines that pass through both Chalmers and Kungsportsplatsen? (2p)

b) Write a relational algebra expression that finds how many trams pass through Brunnsparken each hour. The result relation should have one column that contains the *hour* and a second column that contains the total number of trams that pass through Brunnsparken during that hour. The tuples in the result relation should be sorted by *hour*. (2p)

Question 4. Consider the following relations that contain tram timetable information:

8 p

$Trams(type, capacity)$

$Lines(lineNo, stopNumber, stopName)$

$Journeys(line, direction, tramType, stopNumber, hour, minute)$
$tramType \rightarrow Trams.type$
$line \rightarrow Lines.line$
$stopNumber \rightarrow Lines.stopNumber$

Capacity is the number of passenger places available on that type of tram.

On each line there are two directions of travel: generally from north to south ("NtoS") and generally from south to north ("StoN").

Relation *Lines* contains a list of the stops on that line; the *stopNumber* is the stop's position on the line when the stops are ordered in the direction "NtoS". Thus, for example, Brunnsparken could have different stop numbers on different lines.

a) Write an SQL query that finds the line numbers of all lines that pass through both Chalmers and Kungsportsplatsen. (2p)

b) Write two SQL queries that find all lines that do not stop at Chalmers.
   i) One query must use EXISTS. (1p)
   ii) The other must not use EXISTS. (1p)

c) Write an SQL query that finds the hour(s) during which most trams pass through Brunnsparken. (2p)

d) Create a view $RouteCapacities(line, direction, totalCapacity)$ which has two tuples for each line (one for each direction), and gives the total capacity for that route during the day (this query will need to find all journeys for that route, and the capacities of the trams for these journeys). (2p)

Question 5. 8 p

The following relation schemas describe relations that will hold information about a company's employees, and the departments in which they work.

$Departments(\underline{deptNo}, dname, location)$
$Employees(\underline{empNo}, eName, salary, manager, dept)$
$dept \rightarrow Departments.deptNo$
$manager \rightarrow Employees.empNo$

Every employee will have a manager who works in the same department, except for the head of the department, who will have NULL in their *manager* field. No employee can have a salary below 20000, and when no salary is specified for a new employee a default salary of 20000 should be assigned.

a) Write SQL statements that create the relations with constraints as tables in a DBMS. Motivate the update and delete policies that you choose for attributes *dept* and *manager*. (3p)

b) Write some SQL statements that insert tuples into relations *Departments* and *Employees*. Insert at least two tuples into each, including one head of department, and their immediate employee (who should be given the default salary).
Show an SQL insert statement that will generate an error and state the reason for the error. (2p)

c) It is the company's policy that no employee can earn more than their manager. Write a trigger that checks for updates to an employee's salary that would violate this policy, and in this case set the employee's salary to be the same as their manager's salary. (3p)

Question 6. 4 p

Suppose we have relation $Accounts(accNo, custNo, balance)$, and that this relation contains the following tuples:

accNo	custNo	balance
A001	C001	10000
A002	C002	20000

Suppose that customer "C001" withdraws 1000 from a cash dispenser and that customer "C002" transfers 2000 to customer "C001" (from account "A002" to account "A001"). These transactions (let's call them T1 and T2) can be summarised by the following steps:

T1:

$Step\ T1_A:\ get\ balance\ of\ "A001"\ into\ X$
$Step\ T1_B:\ set\ balance\ of\ "A001"\ to\ X - 1000$

T2:

$Step\ T2_A:\ get\ balance\ of\ "A002"\ into\ Y$
$Step\ T2_B:\ set\ balance\ of\ "A002"\ to\ Y - 2000$
$Step\ T2_C:\ get\ balance\ of\ "A001"\ into\ Z$
$Step\ T2_D:\ set\ balance\ of\ "A001"\ to\ Z + 2000$

a) What are the possible outcomes if neither T1 nor T2 is run as a serializable transaction? Explain your answer. (2p)

b) What are the possible outcomes if T1 is run as a serializable transaction, but T2 is not? Explain your answer. (2p)

Question 7. 4 p

Suppose we have relation $Accounts(accNo, custNo, balance)$, and that this relation is stored in 30 disc blocks.

Each customer can have more than one account and, on average, each customer has accounts stored in two disc blocks.

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

- task 1: inserting a new row;
- task 2: finding account information for a given customer.

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

   i) there are no indexes (1p)

   ii) there is an index on $custNo$ (assume that this index fits into a single disc block). (1p)

b) Suppose that 90% of the operations performed on this relation are inserting new rows (task 1), and that the remaining 10% are finding account information for given customers (task 2).

   In this case, is it better to have an index on $custNo$, or is it better to have no indexes on this relation?
   Show the calculations that support your answer. (2p)

Question 8. Consider the following piece of XML which has a "relational" structure.
8 p

```
<?xml version="1.0" standalone="no">

<Schedules>
 <Courses>
  <Course code="TDA357" name="Databases" />
  <Course code="TIN090" name="Algorithms" />
 </Courses>
 <GivenCourses>
  <GivenCourse course="TDA357" period="2" teacher="Broberg" />
  <GivenCourse course="TDA357" period="4" teacher="Heldal" />
  <GivenCourse course="TIN090" period="1" teacher="Dubhashi" />
 </GivenCourses>
 <Lectures>
  <Lecture course="TDA357 period="2" day="Monday" hour="10" room="HA4" />
  <Lecture course="TDA357 period="2" day="Thursday" hour="10" room="HB1" />
  <Lecture course="TDA357 period="4" day="Friday" hour="10" room="VR" />
  <Lecture course="TIN090 period="1" day="Monday" hour="13" room="VR" />
 </Lectures>
</Schedules>
```

a) Write a Document Type Definition (DTD) for this XML example. (2p)

b) Write XPath expressions that find:
   - all course elements;
   - all lectures that take place on a Monday. (2p)

c) The flexibility of XML enables us to nest elements in a more natural way than in the example shown at the top of this question. Write a piece of XML that contains the same information as in the example shown above, but which uses more natural tags and nesting, and avoids duplication of course codes and period numbers. (2p)

d) Based on the XML in your answer to part (c), write an XQuery expression that gives the course name and teacher of given courses that have a lecture on a Monday, i.e.

```
<Result>Databases: Broberg</Result>
<Result>Algorithms: Dubhashi</Result>
```

(2p)