CHALMERS UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering

Examination in Databases, TDA357/DIT620

Thursday 28 April 2011, 08:30-12:30

Examiner:	Graham Kemp (telephone 772 5411, room 6475 EDIT) The examiner will visit the exam room at 09:30 and 11:30.
Results:	Will be published by mid-May at the latest.
Exam review:	see course web page for time and place http://www.cse.chalmers.se/edu/year/2010/course/TDA357/HT2010/
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 the page number and question number on every page.

Question 1. Consider the following domain description.

12 p

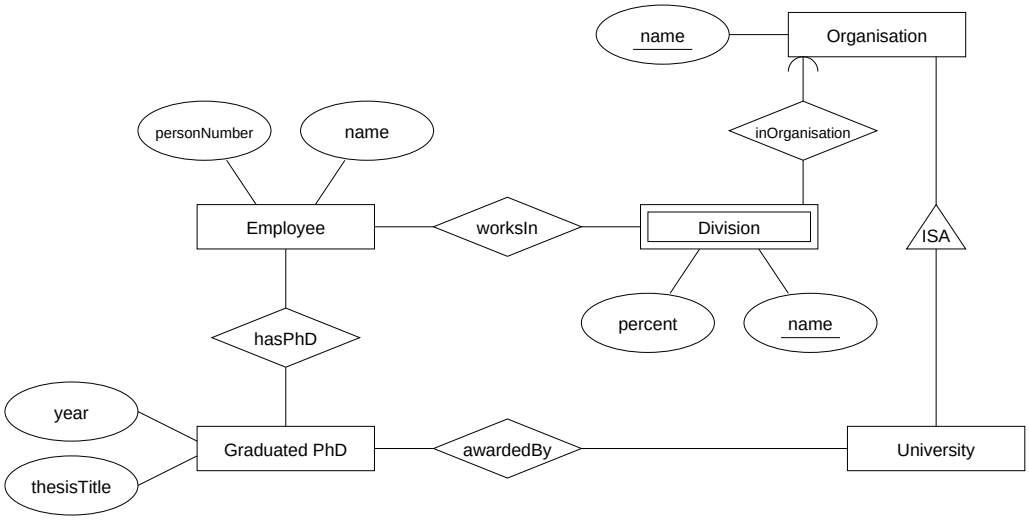
A database will contain information about the employees of several organisations. Organisation names are unique. Each organisation consists of one or more divisions. Within an organisation, the division names are unique. However, different organisations can have divisions with the same name (e.g. "Sales"). Some organisations (universities) differ from other organisations by being allowed to award PhD degrees.

Each employee is identified by a unique personal number. Employee names are not guaranteed to be unique. Each employee can work at more than one division. Therefore, in addition to recording which employees work at each division, we also want to record what percentage of the employee's time is at the different divisions (e.g. employee "e21" could work 75% of their time in the "Sales" division and 25% of their time in the "Personnel" division of an organisation).

We want to record additional information about employees who have a PhD degree. For these employees we want to record the year when the degree was awarded, the title of their thesis, and the university that awarded their PhD.

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.

(Note: there could be other ways to correct the diagram, depending on the approach that you take. So don't be too concerned if your solution requires more than 5 changes to the E-R diagram.)

(6p)

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

(6p)

Question 2. 12 p

a) Suppose we have relation $R(A, B, C, D)$ and functional dependencies $C \rightarrow A$, $D \rightarrow C$.

i) By considering the closures of all subsets of attributes, find **all** non-trivial FDs, superkeys and keys. State which FDs violate BCNF.
(5p)

ii) Decompose relation R to BCNF.
Show each step in the normalisation process, and at each step indicate which functional dependency is being used.
(3p)

b) Suppose relation R_1 is:

A	**B**
a1	b1
a2	b1
a2	b2
a3	b2

and suppose relation R_2 is:

B	**C**	**D**
b1	c1	d1
b2	c2	d1
b2	c3	d2

i) Write out the contents of relation R_3, where $R_3 = R_1 \bowtie R_2$.

ii) State two multivalued dependencies (also referred to as MVDs, or independencies, or INDs) that hold for relation R_3.

(4p)

Question 3. A database used by a course teacher to record students' completed labs has the following relations:
10 p

Students(studentId, name, programme)
GroupMembers(groupNo, studentId)
LabsPassed(group, labNo, tutor)

Student identifiers (*studentId*) are unique, but student names might not be unique. Attribute *programme* is the name of he student's study programme. Students can do their lab work in groups, identified by the attribute *groupNo*. A student cannot be a member of more than one group. There are four labs in the course, so attribute *labNo* has an integer value between 1 and 4, inclusive, and on completing the course a group will have four rows (one for each lab) in relation *LabsPassed*. Attribute *tutor* is the name of the tutor who approved a particular group's solution for a particular lab.

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

b) Suppose that no tutor is allowed to pass more than 100 labs during the course. Write an assertion that checks this.
(2p)

c) No group can have more than 3 members.
Write a trigger that checks whether a new member is being added to a group in relation *GroupMembers* and, if there are already 3 members in that group, then add that student to a new group that contains only that student (e.g. if the highest group number used so far is 20, then this student should be added to a new group with group number 21).
(4p)

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

Students(studentId, name, programme)
GroupMembers(groupNo, studentId)
LabsPassed(group, labNo, tutor)

a) Write a relational algebra expression that finds the number of labs approved by each tutor. The result should be sorted by tutors' names.
(2p)

b) Write a relational algebra expression that finds the group numbers of all groups who have *not* passed all four labs.
(3p)

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

8 p

Students(studentId, name, programme)
GroupMembers(groupNo, studentId)
LabsPassed(group, labNo, tutor)

a) Write an SQL query that finds the number of labs approved by each tutor. The result should be sorted by tutors' names.
(2p)

b) Create a view $V(studentId, name, numLabsPassed)$ which contains the total number of labs that have been passed by the student with $studentId$ and $name$.
(This view should only contain information about students who have passed at least one lab, and there should be one row for each such student.)
(3p)

c) Write an SQL query that finds the group numbers of all groups who have *not* passed all four labs.
(3p)

Question 6. In database transaction processing, what is a *dirty read*?

2 p

Give one advantage of allowing dirty reads.
(2p)

Question 7. Suppose we have relation $PassedCourses(courseId, studentId, grade)$, and that this relation is stored in 20 disc blocks.

5 p

Each course has records stored in 7 disc blocks, on average. Similarly, on average, each student has records stored in 4 disc blocks.

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

- task 1: inserting a new row;
- task 2: finding the $courseId$ and $grade$ for a given $studentId$;
- task 3: finding the $studentId$ and $grade$ for a given $courseId$.

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

a) there are no indexes
(1p)

b) there is an index on $courseId$ (assume that this index fits into a single disc block)
(1p)

c) there is an index on $(studentId, grade)$
(assume that this index fits into a single disc block)
(1p)

d) both of the indexes mentioned in (ii) and (iii) are used
(1p)

e) there is an index on $(grade, studentId)$ and also an index on $(grade, courseId)$
(assume that each of these indexes fits into a single disc block)
(1p)

Question 8. Consider the following piece of XML:

6 p

```
<?xml version="1.0" standalone="yes" ?>
<Courses>
 <Course code="TDA357" name="Databases">
  <Student studentId="s003">
   <Grade>5</Grade>
  </Student>
  <Student studentId="s004">
   <Grade>3</Grade>
  </Student>
  <Student studentId="s001">
   <Grade>3</Grade>
  </Student>
 </Course>
 <Course code="TIN092" name="Algorithms">
  <Student studentId="s001">
   <Grade>5</Grade>
  </Student>
  <Student studentId="s002">
   <Grade>4</Grade>
  </Student>
 </Course>
 <Course code="TDA506" name="Structural Bioinformatics" />
</Courses>
```

a) Write an XPath expression that finds the grade obtained by student "s001" in course "TIN092", i.e. your XPath expression should find the element `<Grade>5</Grade>`
(2p)

b) Write an XQuery expression that finds student identifiers and course names where grade 5 has been obtained. The result of this query should look as follows:

```
<GradeFive studentId="s003" name="Databases" />
<GradeFive studentId="s001" name="Algorithms" />
```

(2p)

c) The flexibility of XML enables us to represent the same information in different ways using XML.
Write a piece of XML that is compatible with the DTD that is given below, and contains the same information that is in the XML at the top of this question.

```
<!DOCTYPE Question8 [

<!ELEMENT Question8 (Courses, Grades) >

<!ELEMENT Courses (Course*) >
 <!ELEMENT Course EMPTY >
  <!ATTLIST Course
    code ID    #REQUIRED
    name CDATA #REQUIRED >

<!ELEMENT Grades (Grade*) >
 <!ELEMENT Grade (#PCDATA) >
  <!ATTLIST Grade
    course    IDREF #REQUIRED
    studentId CDATA #REQUIRED >

]>
```

(2p)