CHALMERS UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering

Examination in Databases, TDA357/DIT620

Tuesday 17 December 2013, 14:00-18:00

Examiner:	Graham Kemp (telephone 772 54 11, 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:	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.
12 p

A local authority manages several nurseries which provide daycare for children. They want to use a database to record information about their nurseries. Each nursery is identified by its name. Each nursery is organised into several sections, each with around 15 children. The sections within each nursery have unique names, but there might be sections with the same name in different nurseries. The local authority employs several teachers and each teacher is assigned to one of the sections. Each teacher's name and person number should be stored in the database. The name and person number of each child should also be stored. Initially, an application is made for a nursery place for a child. The application contains information about the child, the child's starting date at nursery, and a list of nursery choices (e.g. choice 1 is nursery "A", choice 2 is nursery "B", etc.). After an application is processed, the child is allocated an available place in one of the sections of one of the nurseries. Information about the application and the child's placement should be stored in the database. For each child, the person number, name and telephone number of each parent should be stored in the database.

a) Draw an E-R diagram that correctly models this domain.
(6p)

b) Translate this E-R diagram into a set of relations, clearly marking all references and keys. If any attributes can contain null values, state which ones.
(6p)

Question 2. Suppose we have relation $R(A, B, C, D)$ and functional dependencies $A \to B$, $B \to C$, $A \to C$, $BD \to A$.
11 p

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),
   i) state which FDs violate BCNF.
   ii) state which FDs violate third normal form (3NF).
(2p)

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

d) Suppose we are told that relation $S(A, B, C, D)$ contains the following tuples and possibly some others:

A	**B**	**C**	**D**
a1	b1	c1	d1
a1	b1	c1	d2
a1	b2	c2	d1
a2	b1	c1	d1
a2	b3	c3	d3

Suppose we are also told that multivalued dependency $a \twoheadrightarrow b, c$ holds for relation S. Which other tuple(s) must be present in relation S?
(2p)

Question 3. 9 p

A multi-national company uses a relational database to manage information about its offices in different cities, and its employees. This database has the following relations:

Offices(city, supplement)
Departments(city, dname, departmentHead)
Employees(empId, name, salary, dept, city)

The company has one office in each city, and several departments can be located at each office.
Attribute *supplement* is the monthly salary supplement that each employee working at that office receives (e.g. employees at the London office might receive a supplement of 1000 SEK per month to cover higher living costs in London). The default city supplement is 0 SEK.
Attribute *dname* describes the department's function (e.g. 'sales' or 'personnel').
Attribute *departmentHead* is the employee identifier of the head of the department.
Employee identifiers (*empId*) are unique.
Attribute *salary* is an employee's basic monthly salary.
The total monthly salary for an employee can be calculated by adding the city supplement to the employee's basic monthly salary.

a) Suggest references for these relations.
Write SQL statements that create these relations with constraints in a DBMS.
(3p)

b) A department head must be an employee who works at that department.
Write an assertion that checks this.
(2p)

c) Write a stored procedure called `Merge` that merges two departments into one. This procedure should have four input parameters: `city1`, `dept1`, `city2`, `dept2`. All employees from the first department should be reassigned to the second department. Who becomes the head of the merged department depends on which department was larger before merging: if the first department was larger then the head of that department becomes the head of the merged department, otherwise the head of the second department becomes the head of the merged department.
(4p)

Question 4. 6 p

Assume the same relations as in Question 3:

Offices(city, supplement)
Departments(city, dname, departmentHead)
Employees(empId, name, salary, dept, city)

a) Write a relational algebra expression that finds the employee identifier, name and total monthly salary of all employees (recall that the total monthly salary for an employee can be calculated by adding the city supplement to the employee's basic monthly salary). The results should be sorted by employee name.
(3p)

b) Write a relational algebra expression that finds the names of cities where there is a sales department and, for each of these departments, the average basic salary of the employees in that department.
(3p)

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

10 p

Offices(city, supplement)
Departments(city, dname, departmentHead)
Employees(empId, name, salary, dept, city)

a) Write an SQL query that finds the employee identifier, name and total monthly salary of each employee (recall that the total monthly salary for an employee can be calculated by adding the city supplement to the employee's basic monthly salary). The results should be sorted by employee name.
(3p)

b) Write SQL queries that find the names of departments in the London office where there is no department with the same name in the Paris office. Do this in two ways:
i) using `IN`
ii) using a set operation
(4p)

c) Create a view *SalaryBill(city, amount)* where *city* is a city name, and *amount* is the total monthly salary bill for all employees who work for the company in that city (taking into account their basic salary and any city supplement).
(3p)

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

5 p

Offices(city, supplement)
Departments(city, dname, departmentHead)
Employees(empId, name, salary, dept, city)

a) Suppose transaction T1 counts the number of employees who are employed in the London office, prints the result, then reads some data from the Departments relation, then counts and prints the number of employees who are employed in the London office again.
Suppose transaction T2 deletes one of the London employees from the Employees relation, and transaction T3 inserts a new London employees into the Employees relation.
Explain why the result of running transaction T1 with isolation level `REPEATABLE READ` could be different to running it with isolation level `SERIALIZABLE`. You should mention transaction T2 or T3 in your answer.
(2p)

b) Suppose relation Employees is stored in 30 disc blocks, and that each city has employee records stored (on average) in 5 disc blocks.
Suppose that two kinds of task are performed on this relation:
- task 1: inserting a new row;
- task 2: finding the employee identifiers for a given *city*.

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 *city* (assume that this index fits into a single disc block).

Suppose that 90% of operations performed on this relation are task 1 (inserting new rows) and 10% are task 2 (finding the employee identifiers for given cities).
iii) Discuss whether it would be better to have an index on *city*, or to have no indexes.
(3p)

Question 7. Consider the following piece of XML:

7 p

```
<A a1="A1">
  <B b1="B1" b2="15">
    <C c1="red">first</C>
  </B>
  <B b1="B2">
    <C c1="green">second</C>
  </B>
  <B b1="B3" b2="25">
    <C c1="blue">third</C>
  </B>
  <B b1="B4" b2="35">
    <C c1="red">fourth</C>
  </B>
</A>
```

a) Give *two* reasons why the XML shown above is not compatible with the following DTD. State how the DTD should be modified so that it is compatible with the XML shown above.

```
<!DOCTYPE A [
<!ELEMENT A (B) >
<!ELEMENT B (C) >
<!ELEMENT C (#PCDATA) >
<!ATTLIST A
  a1 CDATA #REQUIRED >
<!ATTLIST B
  b1 CDATA #REQUIRED
  b2 CDATA #REQUIRED >
<!ATTLIST C
  c1 CDATA #REQUIRED >
]>
```

(2p)

b) What are the results of the applying the following path expressions:

i) `/A/B[C/@c1="red"]`

ii) `//B[attribute::b2>20]/C`

(2p)

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

```
<Result>
  <C c1="blue">
    <B b1="B3"/>
  </C>
  <C c1="green">
    <B b1="B2"/>
  </C>
  <C c1="red">
    <B b1="B1"/>
  </C>
  <C c1="red">
    <B b1="B4"/>
  </C>
</Result>
```

(3p)

CHALMERS UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering

Examination in Databases, TDA357/DIT620

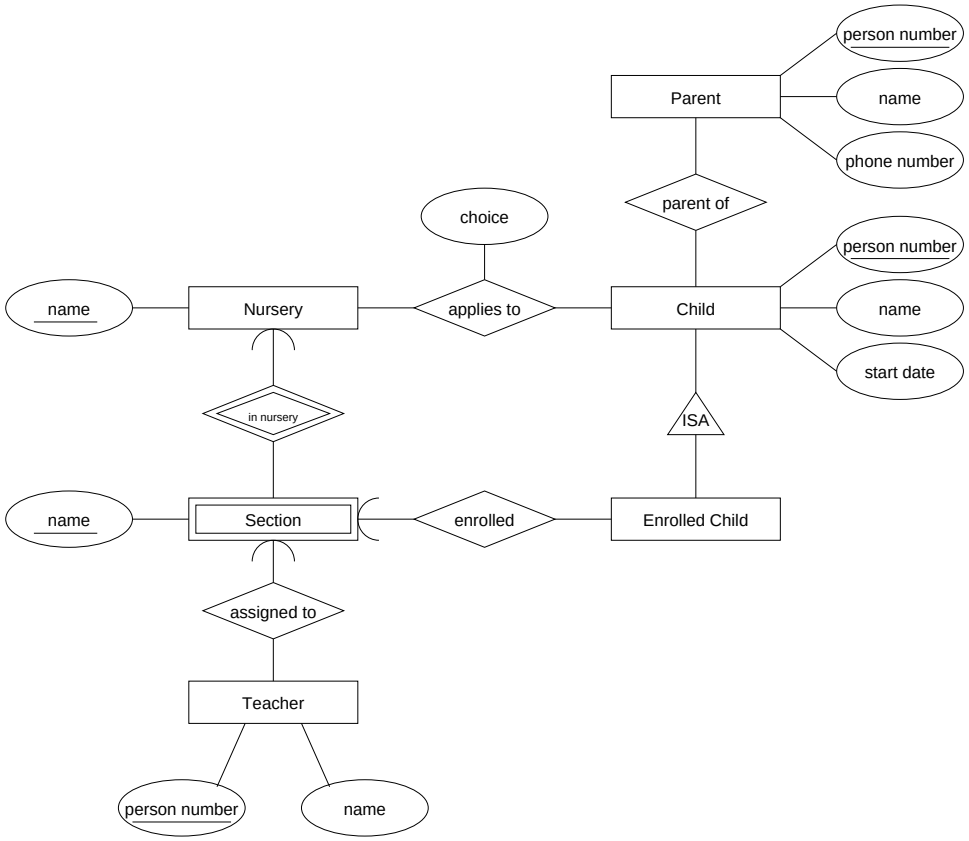
Tuesday 17 December 2013, 14:00-18:00

Solutions

Updated 2014-11-19

Question 1. 12 p

a) (Here is one suggestion. Several other designs are also accepted. For example, modelling 'Teacher', 'Parent' and 'Child' as subclasses of 'Person'.)



b) $Nurseries(\underline{name})$

$Sections(\underline{nursery, name})$
$nursery \rightarrow Nurseries.name$

$Teachers(\underline{personNumber}, name, nursery, section)$
$(nursery, section) \rightarrow Sections.(nursery, name)$

$Parents(\underline{personNumber}, name, phoneNumber)$

$Children(\underline{personNumber}, name, startDate)$

$EnrolledChildren(\underline{personNumber}, nursery, section)$
$personNumber \rightarrow Children.personNumber$
$(nursery, section) \rightarrow Sections.(nursery, name)$

$AppliesTo(\underline{child}, \underline{nursery}, choice)$
$child \rightarrow Children.personNumber$
$nursery \rightarrow Nurseries.name$

$ParentOf(\underline{parent}, \underline{child})$
$parent \rightarrow Parents.personNumber$
$child \rightarrow Children.personNumber$

Question 2. a) In addition to the FDs listed in the question, we also have:
11 p

```
AB -> C
AC -> B
AD -> B
AD -> C
BD -> C
ABD -> C
ACD -> B
BCD -> A

Superkeys: AD, BD, ABD, ACD, BCD, ABCD
Keys:      AD, BD
```

b) i) FDs violating BCNF: A->B, B->C, A->C, AB->C, AC->B

ii) FDs violating 3NF: B->C, A->C, AB->C

c) Decompose R on A->B
{A}+ = {ABC}

```
R1(_A,B,C)
R2(_A,_D)
        A -> R1.A
```

Decompose R1 on B->C
{B}+ = {BC}

```
R11(_B,C)
R12(_A,B)
        B->R11.B
```

Update reference for R2: A -> R12.A

d) (a1,b2,c2,d2)
(a2,b1,c1,d3)
(a2,b3,c3,d1)

Question 3. 9 p

a) $Offices(\underline{city}, supplement)$
$Departments(\underline{city}, \underline{dname}, departmentHead)$
$city \rightarrow Offices.city$
$departmentHead \rightarrow Employees.empId$
$Employees(\underline{empId}, name, salary, dept, city)$
$(city, dept) \rightarrow Departments.(city, dname)$

```
CREATE TABLE Offices (
    city        VARCHAR(20) PRIMARY KEY,
    supplement  INT DEFAULT 0,
);

CREATE TABLE Departments (
    city           VARCHAR(20),
    dname          VARCHAR(20),
    departmentHead CHAR(10),
    PRIMARY KEY (city, dname),
    FOREIGN KEY (city) REFERENCES Offices(city)
    FOREIGN KEY (departmentHead) REFERENCES Employees(empId)
        ON DELETE SET NULL
        ON UPDATE CASCADE,
);

CREATE TABLE Employees (
    empId          CHAR(10) PRIMARY KEY,
    name           VARCHAR(30),
    salary         INT,
    dept           VARCHAR(20),
    city           VARCHAR(20),
    FOREIGN KEY (city, dept) REFERENCES Departments(city, dname)
);
```

Several of the solutions we saw used the policy "CASCADE" instead of "SET NULL" for the foreign key constraint on attribute departmentHead in table Departments. Consider what would be the consequences of this.

The solution shown above would give an error if executed, due to the forward references from the Departments table definition to the Employees table, which hasn't been created yet. This complication was ignored when marking the exam. In practice, we could omit this foreign key constraint when creating the Departments table, and then add this constraint after the Employees table has been created. This can be done using the ALTER TABLE statement, e.g.

```
ALTER TABLE Departments ADD CONSTRAINT departmentREFemployee
    FOREIGN KEY (departmentHead) REFERENCES Employees(empId)
    INITIALLY DEFERRED DEFERRABLE;
```

For more information on this, see the section on "Deferring Constraint Checking" on the website for the course textbook:
`http://infolab.stanford.edu/~ullman/fcdb/oracle/or-triggers.html`

b)
```
CREATE ASSERTION HeadOfOwnDept CHECK
   ( NOT EXISTS (
        SELECT departmentHead
        FROM   Departments JOIN Employees ON departmentHead = empId
        WHERE  dname <> dept
               OR Departments.city <> Employees.city ) )
```

c)
```
CREATE PROCEDURE Merge (
    IN city1 VARCHAR(20),
    IN dept1 VARCHAR(20),
    IN city2 VARCHAR(20),
    IN dept2 VARCHAR(20)
)
BEGIN
    IF ( SELECT COUNT(empId)
         FROM   Employees
         WHERE  city = city1 AND dept = dept1 ) >
       ( SELECT COUNT(empId)
         FROM   Employees
         WHERE  city = city2 AND dept = dept2 )
    THEN
        UPDATE Departments
        SET    departmentHead =
                   ( SELECT departmentHead
                     FROM   Departments
                     WHERE  city = city1 AND dname = dept1 )
        WHERE  city = city2 AND dept = dept2;
    ENDIF;

    UPDATE Employees
    SET    city = city2,
           dept = dept2
    WHERE  city = city1 AND dept = dept1;

    DELETE FROM Departments WHERE city = city1 AND dname = dept1;
END;
```

Question 4. 6 p

a) $\tau_{name}(\pi_{empId,name,salary+supplement}(Employees \bowtie Offices))$

b) If we assume that all sales departments have at least one employee:

$$\gamma_{city,AVG(salary)\rightarrow avgSalary}(\sigma_{dept=\text{"}sales\text{"}}(Employees))$$

If there can be sales departments with no employees, we might want to include those in the result, with '0' as the average. This can be done by forming the union of the relational algebra expression given above with:

$$\pi_{city,0}(\pi_{city}(\sigma_{dname=\text{"}sales\text{"}}(Departments)) - \pi_{city}(\sigma_{dept=\text{"}sales\text{"}}(Employees)))$$

Question 5. 10 p

a)
```
SELECT   empId, name, salary + supplement AS totalSalary
FROM     Employees NATURAL JOIN Offices
ORDER BY name
```

b) i)
```
SELECT  dname
FROM    Departments
WHERE   city = "London"
        AND dname NOT IN (
                SELECT  dname
                FROM    Departments
                WHERE   city = "Paris" )
```

ii)
```
( SELECT  dname
  FROM    Departments
  WHERE   city = "London" )
      EXCEPT
( SELECT  dname
  FROM    Departments
  WHERE   city = "Paris" )
```

c)
```
CREATE VIEW SalaryBill AS
SELECT   city, SUM(salary + supplement) AS amount
FROM     Employees NATURAL JOIN Offices
GROUP BY city
```

Question 6. 5 p

a) The result printed by transaction T1 could be different if transaction T1 and T3 are run concurrently. Good answers will discuss the concept of *phantoms* (see Example 6.47 in the course textbook) and the schedule of operations that causes different results to be printed.

b) i) task 1: 2
task 2: 30

ii) task 1: 4
task 2: 6

iii) It would be better to have an index on *city* (cost: 420 vs. 480).

Question 7. 7 p

a) Corrected DTD is:

```
<!DOCTYPE A [
<!ELEMENT A (B*) >
<!ELEMENT B (C) >
<!ELEMENT C (#PCDATA) >
<!ATTLIST A
  a1 CDATA #REQUIRED >
<!ATTLIST B
  b1 CDATA #REQUIRED
  b2 CDATA #IMPLIED >
<!ATTLIST C
  c1 CDATA #REQUIRED >
]>
```

b) i)

```
<B b1="B1" b2="15">
  <C c1="red">first</C>
</B>
<B b1="B4" b2="35">
  <C c1="red">fourth</C>
</B>
```

ii)

```
<C c1="blue">third</C>
<C c1="red">fourth</C>
```

c)

```
<Result>
  {
    for $b in (doc("exam.xml")//B)
    order by $b/C/@c1
    return <C c1="{$b/C/@c1}"><B b1="{$b/@b1}" /></C>
  }
</Result>
```