CHALMERS UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering

Examination in Databases, TDA357/DIT620

Thursday 4 April 2013, 14:00-18:00

Examiner:	Graham Kemp (telephone 772 54 11, room 6475 EDIT) The examiner will visit the exam room at 15:20 and 17:00.
Results:	Will be published by 22 April at the latest.
Exam review:	See course web page for time and place: http://www.cse.chalmers.se/edu/year/2012/course/TDA357/HT2012/
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 water supplier provides water to all customers in a district. Each customer is identified by a unique customer reference number, and the water company stores each customer's name and address in the database.

The water supplier sends out invoices to its customer. An invoice is identified by the customer number and the date on which it is sent out. The invoice shows the amount to be paid and also the amount of water used, and these are stored in the database.

Some customers register to receive electronic invoices (paperless invoices that are sent directly to the customer's bank). For these customers (and only these customers), the water supplier stores the customer's email address, the name of their bank and their bank account number.

The amount of water used is measured by a water meter. Each water meter is identified by a meter number. For each meter, the water supplier stores the meter's type and dimension (in mm). Meter readings are recorded in the database; whenever a meter is read, the amount of water used (in m^3) is recorded, together with the date on which the reading was taken. Exactly one water meter is associated with each customer identifier, and each customer has exactly one water meter.

In addition to paying for the amount of water used, customers pay an annual fee which depends on they type of their water meter. The annual fee associated with each water meter type 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.
(6p)

Question 2. Suppose we have relation $R(A, B, C, D)$ and functional dependencies $AB \to D$, $BC \to D$, $CD \to B$, $D \to A$.
12 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, with reasons, which FDs violate BCNF.
   ii) state, with reasons, which FDs violate third normal form (3NF).
(3p)

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) *Dependency preservation* is an important property of decomposition. Does the decomposition in your answer to part (c) give dependency preservation? Explain your answer.
(2p)

Question 3. 9 p

An estate agent uses a relational database to manage information about clients, properties for sale and bids made by clients for properties. This database has the following relations:

Clients(cid, name, telephone)
Properties(ref, address, area, seller, sold, saleDate)
Bids(property, client, amount)

A client has a unique client identifier (attribute *cid*). The client's name and telephone number are stored in the database. A client can be someone who is selling a property, or who is interested in buying a property or both (i.e. a seller of one property (or more than one property) and a prospective buyer for others).

Each property that the agent is trying to sell is identified by a reference code. The address of each property is stored in the database, and also the property's living area (in m^2). Attribute *seller* is the client identifier of the client who is selling he property. Attribute *sold* contains the value 'Y' or 'N' (default value is 'N') to indicate whether the property has been sold yet and, if it has been sold, attribute *saleDate* records the date of the sale.

During the lifetime of the database, the same client may sell more than one property, and the same property could be sold several times (each time a property is sold it will be assigned a new reference code).

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

b) A client is not allowed to bid on a property which they are selling.
Write an assertion that checks this.
(2p)

c) Each bid made on a property must be at least as high as the previous bids made on that property.
Write trigger code that enforces this constraint.
(4p)

Question 4. 6 p

Assume the same relations as in Question 3:

Clients(cid, name, telephone)
Properties(ref, address, area, seller, sold, saleDate)
Bids(property, client, amount)

a) Write a relational algebra expression that finds the seller's name and the address of properties with an area greater than $300m^2$. The results should be sorted by area.
(3p)

b) Write a relational algebra expression that finds the property identifier of properties (sold or unsold) that have received bids from at least three different clients.
(3p)

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

9 p

Clients(cid, name, telephone)
Properties(ref, address, area, seller, sold, saleDate)
Bids(property, client, amount)

a) Write an SQL query that finds the seller's name and the address of properties with an area greater than $300m^2$. The results should be sorted by area.
(3p)

b) Create a view *PricePerSquareMetre(property, address, price)* where *property* is a property identifier of a sold property, *address* is the property's address, and *price* is the price per square metre for that property calculated from the property's area and the amount of the highest bid on that property.
(3p)

c) Write an SQL query that finds the the client identifiers of clients who bid for a property that was sold in 2011, but did not bid for any properties that were sold in 2012.
(3p)

Question 6. A system for bidding on properties has a transaction T with the following two steps:

4 p

T_1: find the highest bid on property 'p1' so far, and store this in m
T_2: make a bid of $m + 100000$ for property 'p1'

a) Suppose the highest bid so far is 3000000, and that users A and B run transaction T at the same time. Give all *schedules* (possible interleavings) of the steps and state the corresponding outcomes.
(2p)

b) Suppose transaction T operates with the same relations as in Question 3.
Which index(es) would improve the performance of transaction T?
(2p)

Question 7. Consider the following piece of XML:

8 p

```
<LabReports>
 <Labs>
  <Lab number="1" deadline="2012-11-13" />
  <Lab number="2" deadline="2012-11-27" />
 </Labs>
 <LabsPassed>
  <LabPassed lab="1" group="20" />
  <LabPassed lab="1" group="21" />
  <LabPassed lab="2" group="20" />
  <LabPassed lab="2" group="22" />
 </LabsPassed>
</LabReports>
```

Assume that this XML is in file *labreports.xml*.

a) Write an XQuery expression that constructs the following result:

```
<Result>
  <Lab number="1">
    <LabPassed lab="1" group="20"/>
    <LabPassed lab="1" group="21"/>
  </Lab>
  <Lab number="2">
    <LabPassed lab="2" group="20"/>
    <LabPassed lab="2" group="22"/>
  </Lab>
</Result>
```

(3p)

b) Write an XQuery expression that counts the number of groups who have passed each lab and constructs the following result:

```
<Result>
  <Lab numPassed="2" number="1">
    <Group group="20"/>
    <Group group="21"/>
  </Lab>
  <Lab numPassed="2" number="2">
    <Group group="20"/>
    <Group group="22"/>
  </Lab>
</Result>
```

(3p)

c) Write a Document Type Definition (DTD) that describes the result of the XQuery expression in part (b).

(2p)

CHALMERS UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering

Examination in Databases, TDA357/DIT620

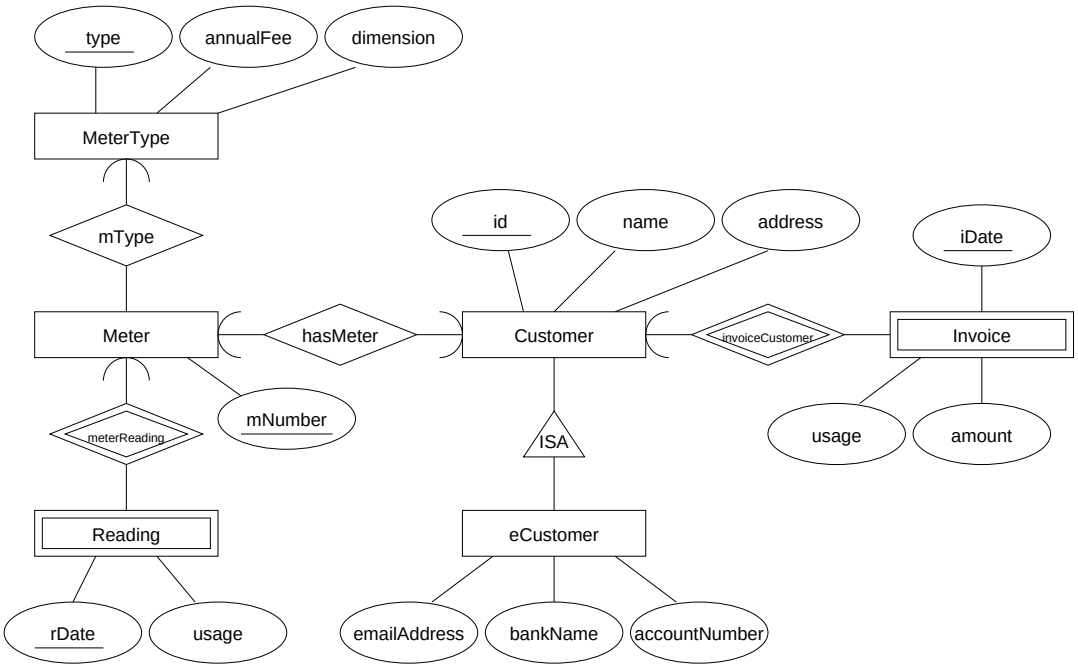
Thursday 4 April 2013, 14:00-18:00

Solutions

Updated 2013-12-08

Question 1. 12 p

a) (Here is one suggestion. Several other designs are also accepted.)
E-R diagram:



b) $Customers(\underline{id}, name, address)$

$eCustomers(\underline{id}, emailAddress, bankName, accountNumber)$
$id \rightarrow Customers.id$

$Invoices(\underline{iDate}, \underline{customer}, amount, usage)$
$customer \rightarrow Customers.id$

$MeterTypes(\underline{mType}, annualFee, dimension)$

$Meters(\underline{mNumber}, mType, customer)$
$customer \rightarrow Customers.id$
$mType \rightarrow MeterTypes.mType$
$UNIQUE \;\; customer$

$Readings(\underline{meter}, \underline{rDate}, usage)$
$meter \rightarrow Meters.mNumber$

Question 3. 9 p

a)
```
CREATE TABLE Clients (
    cid INT PRIMARY KEY,
    name VARCHAR(30),
    telephone VARCHAR(15)
);

CREATE TABLE Properties (
    ref         VARCHAR(10) PRIMARY KEY,
    address     VARCHAR(30),
    area        INT,
    seller INT,
    sold        CHAR(1) DEFAULT 'N' CHECK (sold IN ('Y','N')),
    saleDate DATE,
    FOREIGN KEY (seller) REFERENCES Clients(cid)
--reject deletion or update of referenced row
);

CREATE TABLE Bids (
    property    VARCHAR(10),
    client      INT,
    amount INT,
    PRIMARY KEY (property, client, amount),
    FOREIGN KEY (property) REFERENCES Properties(ref),
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (client) REFERENCES Clients(cid)
--reject deletion or update of referenced row
);
```

b)
```
CREATE ASSERTION NotBiddingOnOwnProperty CHECK
  ( NOT EXISTS (
        SELECT   seller
        FROM     Properties, Bids
        WHERE    ref = property AND seller = client
  ) );
```

c)
```
CREATE OR REPLACE TRIGGER BidHighEnough
BEFORE INSERT ON Bids
REFERENCING NEW AS new
FOR EACH ROW
DECLARE numHigherBids INT;
BEGIN
    SELECT COUNT(amount) INTO numHigherBids
    FROM   Bids
    WHERE  property = :new.property AND amount > :new.amount;

    IF ( numHigherBids > 0 ) THEN
        RAISE_APPLICATION_ERROR(-20000, 'bid not high enough');
    END IF;
END;
```

Question 5. 9 p

a)
```
SELECT   name, address
FROM     Properties, Clients
WHERE    area > 300 AND cid=seller
ORDER BY area;
```

b)
```
CREATE VIEW v AS
SELECT   property,address, MAX(amount) / area AS price
FROM     Bids, Properties
WHERE    property=ref and sold='Y'
GROUP BY property, address, area;
```

c)
```
-- Find those who bid in 2012
-- Find those who bid in 2011
-- Subtract these
```

Question 7. 8 p

a)
```
<Result>
  {
    let $l := doc("l1.xml")
    for $n in $l//Lab/@number
    let $p := $l//LabPassed[@lab=$n]
    return <Lab number="{$n}">{$p}</Lab>
  }
</Result>
```

b)
```
<Result>
  {
    let $l := doc("l1.xml")
    for $n in $l//Lab/@number
    let $p := $l//LabPassed[@lab=$n]
    return <Lab number="{$n}" numPassed="{count($p)}">
      {
        for $g in $p/@group
        return <Group>{$g}</Group>
      }
    </Lab>
  }
</Result>
```

c)
```
<!DOCTYPE Result [

<!ELEMENT Result (Lab*)>

<!ELEMENT Lab (Group*) >

<!ELEMENT Group EMPTY >

<!ATTLIST Lab
  number CDATA #REQUIRED
  numPassed CDATA #REQUIRED >

<!ATTLIST Group
  group CDATA #REQUIRED >

]>
```