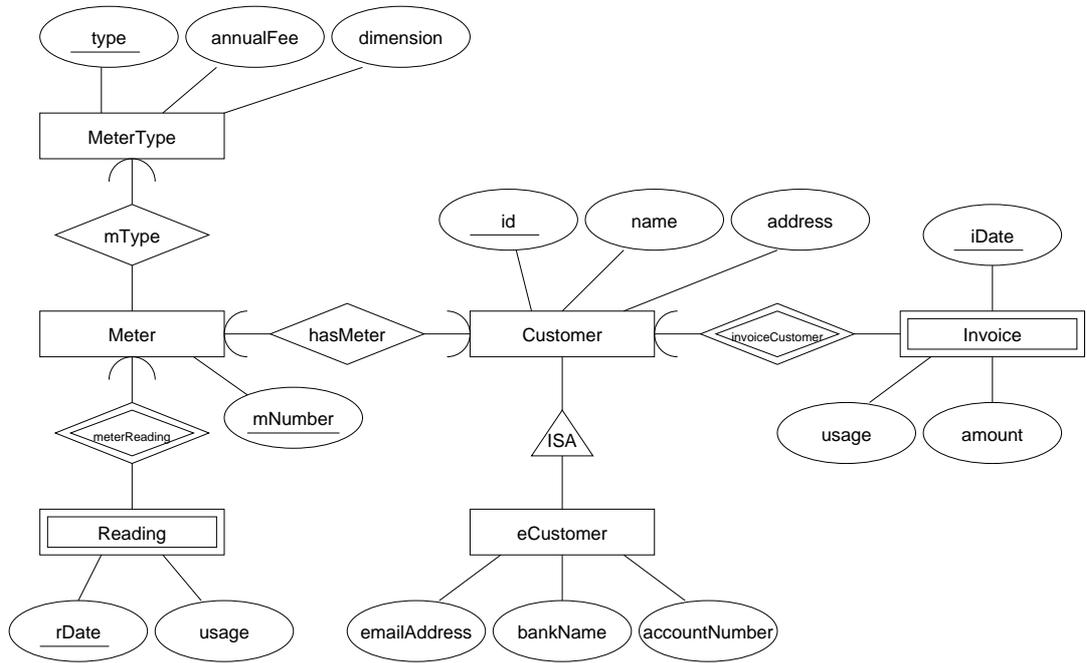


CHALMERS UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering
Examination in Databases, TDA357/DIT620
Thursday 4 April 2013, 14:00-18:00

Solutions

Updated 2013-12-08

Question 1. a) (Here is one suggestion. Several other designs are also accepted.)
 12 p E-R diagram:



b) $Customers(id, name, address)$

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

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

$MeterTypes(mType, annualFee, dimension)$

$Meters(mNumber, mType, customer)$
 $customer \rightarrow Customers.id$
 $mType \rightarrow MeterTypes.mType$
 UNIQUE customer

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

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 >
]>
```