CHALMERS UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering

Examination in Databases, TDA357/DIT620

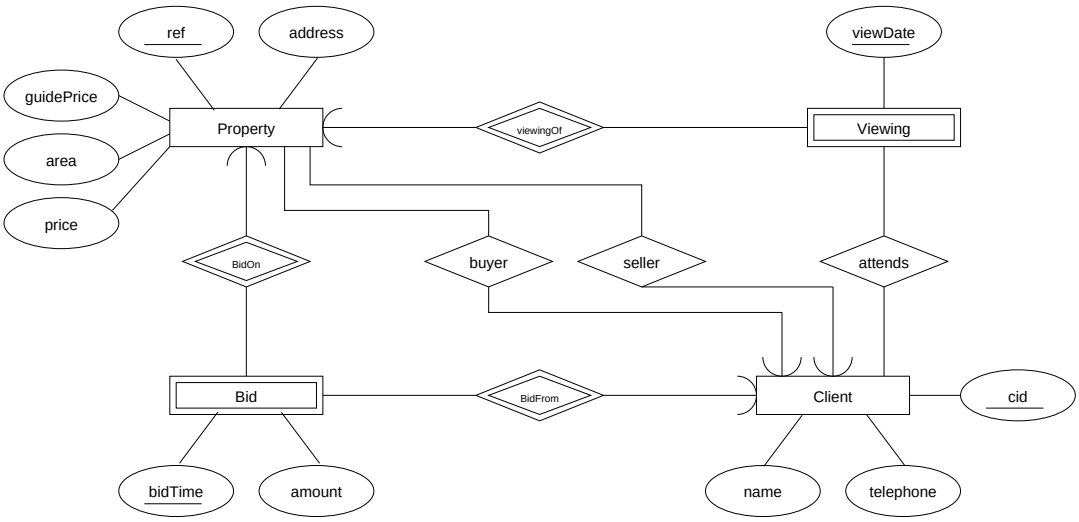
Thursday 20 December 2012, 14:00-18:00

Solutions

Updated 2012-12-22

Question 1. 12 p

a) (Here is one suggestion. Several other designs are also accepted. For example, modelling 'SoldProperties' as a subclass of property, or modelling 'Buyers' and 'Sellers' as subclasses of client, to model the different roles that clients can have.)
E-R diagram:



b) $Clients(\underline{cid}, name, telephone)$

$Properties(\underline{ref}, address, guidePrice, area, price, seller, buyer)$
$seller \rightarrow Clients.cid$
$buyer \rightarrow Clients.cid$

$Viewings(\underline{property}, \underline{viewDate})$
$property \rightarrow Properties.ref$

$Attends(\underline{property}, \underline{viewDate}, \underline{client})$
$(property, viewDate) \rightarrow Viewings.(property, viewDate)$

$Bids(\underline{property}, \underline{client}, \underline{bidTime}, amount)$
$property \rightarrow Properties.ref$
$client \rightarrow Clients.cid$

Question 2. 10 p

a) $ABCD$ — does not identify all attributes.
$ACDEG$ — this is a superkey but not a key, since attribute G can be removed and the resulting set of attributes is a key.

b)
```
Decompose on BC -> D
{BC}+ = {BCDG}

        R1(_B,_C,D,G)
        R2(B,C,A,E,F)
            B,C -> R1.(B,C)

Decompose R2 on FA -> B
{FA}+ = {FAB}

        R21(_F,_A,B)
        R22(F,A,C,E)
            F,A -> R21.(F,A)

Key of R22 is FACE
```

c) $BC \rightarrow G$
Left side is not a superkey of R, and G is not prime in R.

d) $R1(B, C, D, G)$
$R2(D, E, F)$
$R3(F, A, B)$
$R4(F, A, C, E)$

Question 3. 11 p

a) $Wards(\underline{number}, numBeds)$
$Patients(\underline{pid}, name, year, gender)$
$PatientInWard(\underline{pid}, ward)$
$pid \rightarrow Patients.pid$
$ward \rightarrow Wards.num$
$Tests(\underline{patient}, \underline{testDate}, \underline{testHour}, temperature, heartRate)$
$patient \rightarrow Patients.pid$

```
CREATE TABLE Wards (
    num         INT PRIMARY KEY,
    numBeds     INT
);

CREATE TABLE Patients (
    pid         CHAR(10) PRIMARY KEY,
    name        VARCHAR(30),
    year        INT,
    gender      CHAR(1) CHECK (gender IN ('F','M'))
);

CREATE TABLE PatientInWard (
    pid         CHAR(10),
    ward        INT,
    PRIMARY KEY (pid),
    FOREIGN KEY (pid) REFERENCES Patients(pid)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (ward) REFERENCES Wards(num)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE Tests (
    patient     CHAR(10),
    testDate    DATE,
    testHour    INT,
    temperature REAL,
    heartRate   INT,
    PRIMARY KEY (patient, testDate, testHour),
    FOREIGN KEY (patient) REFERENCES Patients(pid)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

b)

```
CREATE ASSERTION NotOverFullWard CHECK
  ( NOT EXISTS (
        SELECT    num
        FROM      Wards JOIN PatientInWard ON num=ward
        GROUP BY  num, numBeds
        HAVING    numBeds < COUNT(pid)
  ) );
```

c)
```
CREATE TRIGGER WardFull
BEFORE INSERT ON PatientInWard
REFERENCING NEW AS new
FOR EACH ROW
DECLARE numAvailable INT;
        availableWard INT;
BEGIN
    SELECT  numBeds INTO numAvailable
    FROM    FreeBeds
    WHERE   ward = :new.ward;

    IF numAvailable = 0 THEN
        SELECT MIN(ward) into availableWard
        FROM FreeBeds
        WHERE numBeds > 0;

        :new.ward := availableWard;
    END IF;
END;
```

Question 4. 7 p

a) Unfortunately the attributes names in Questions 3 and 4 are inconsistent, so we accept either:

$\pi_{temperature,heartRate}(\sigma_{year<1950}(Patients) \bowtie Tests)$

or:

$\pi_{temperature,heartRate}(\sigma_{year<1950}(Patients) \bowtie_{patient=pid} Tests)$

(Similarly, we accept alternative solutions for Question 5(a).)

b)

$$
\begin{array}{l}
(\pi_{year} \\
\quad (\sigma_{m>f} \\
\quad\quad (\\
\quad\quad\quad \gamma_{year,COUNT(pid)ASm}(\sigma_{gender='M'}(Patients)) \\
\quad\quad \bowtie \\
\quad\quad\quad \gamma_{year,COUNT(pid)ASf}(\sigma_{gender='F'}(Patients)) \\
\quad\quad) \\
\quad) \\
) \\
\cup \\
(\\
\quad (\pi_{year}(\sigma_{gender='M'}(Patients))) \ \setminus \ (\pi_{year}(\sigma_{gender='F'}(Patients))) \\
)
\end{array}
$$

Question 5. 10 p

a)
```
SELECT  temperature, heartRate
FROM    Patients, Tests
WHERE   pid = patient and year < 1950
```

b)
```
CREATE VIEW FreeBeds AS
  SELECT   num as ward, numBeds - COUNT(pid) AS numBeds
  FROM     Wards LEFT OUTER JOIN PatientInWard ON ward = num
  GROUP BY num, numBeds
```

c)
```
WITH
  R1 AS
      ( SELECT    year, COUNT(pid) AS m
        FROM      Patients
        WHERE     gender = 'M'
        GROUP BY  year ),
  R2 AS
      ( SELECT    year, COUNT(pid) AS f
        FROM      Patients
        WHERE     gender = 'F'
        GROUP BY  year ),
  R3 AS
      ( SELECT    year
        FROM      R1 NATURAL JOIN R2
        WHERE     m > f ),
 R4 AS
      ( SELECT    year
        FROM      Patients
        WHERE     gender = 'M' ),
  R5 AS
      ( SELECT    year
        FROM      Patients
        WHERE     gender = 'F' )

SELECT year FROM R3 UNION (SELECT year FROM  R4 MINUS SELECT year FROM R5)
```

Using Oracle's NVL function, we could have:

```
WITH
  R1 AS
      ( SELECT    year, COUNT(pid) AS m
        FROM      Patients
        WHERE     gender = 'M'
        GROUP BY  year ),
  R2 AS
      ( SELECT    year, COUNT(pid) AS f
        FROM      Patients
        WHERE     gender = 'F'
        GROUP BY  year ),
  R3 AS
      ( SELECT year, m, NVL(f,0) AS f
        FROM R1 NATURAL LEFT OUTER JOIN R2 )

SELECT year FROM R3 WHERE m>f;
```

Question 6. 4 p

A dirty read can occur when one transaction reads a data value that has been modified by another, before that other transaction commits the change. For example, if T_B modifies the cinema seat bookings by booking a seat (in step T_2), and this modified value is read by T_A (in step T_1), and then transaction T_B rolls back, undoing the change, then transaction T_A will have performed a dirty read.

In this case, a dirty read can occur when T_A runs at isolation level READ UNCOMMITTED, and T_B runs at any isolation level.

Question 7. 6 p

a)

```
<!DOCTYPE Hospital [

<!ELEMENT Hospital (Patients, Tests) >

<!ELEMENT Patients (Patient*) >
  <!ELEMENT Patient EMPTY >
    <!ATTLIST Patient
      pid       ID    #REQUIRED
      name      CDATA #REQUIRED >

<!ELEMENT Tests (Test*) >
  <!ELEMENT Test EMPTY >
    <!ATTLIST Test
      pid       IDREF #REQUIRED
      time      CDATA #REQUIRED
      temp      CDATA #IMPLIED
      heartRate CDATA #IMPLIED >

]>
```

b)

```
<Result>
  {
    let $h := doc("hospital.xml")
    for $p in $h//Patient
    let $tests := $h//Test[@pid = $p/@pid]
    return <Patient pid="{$p/@pid}" name="{$p/@name}">{$tests}</Patient>
  }
</Result>
```