CHALMERS UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering

Examination in Databases, TDA357/DIT620

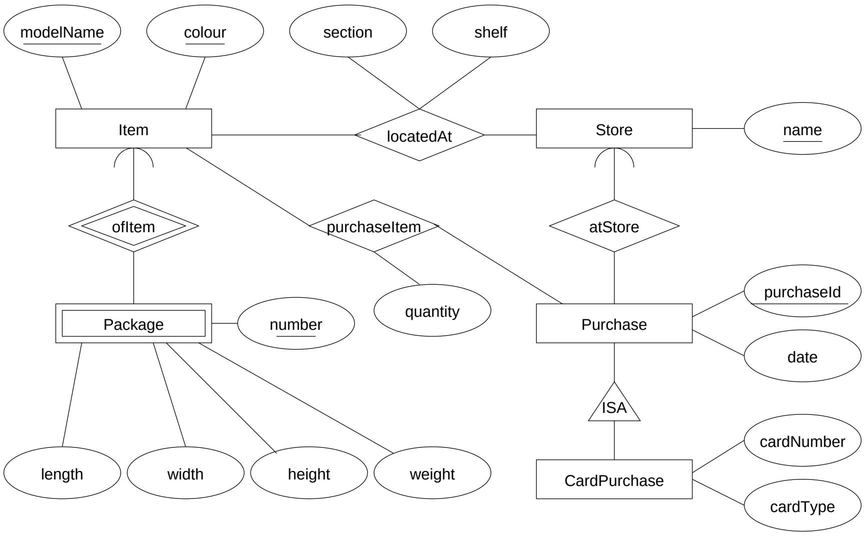
Thursday 16 April 2009, 08:30-12:30

Solutions

Updated 2011-12-12

Question 1. a) E-R diagram:

10 p



b) $Items(\underline{modelName}, \underline{colour})$

$Packages(\underline{modelName}, \underline{colour}, \underline{number}, length, width, height, weight)$
$(modelName, colour) \rightarrow Items.(modelName, colour)$

$Stores(\underline{name})$

$LocatedAt(\underline{modelName}, \underline{colour}, \underline{store}, section, shelf)$
$(modelName, colour) \rightarrow Items.(modelName, colour)$
$store \rightarrow Stores.name$

$Purchases(\underline{purchaseId}, date, atStore)$
$atStore \rightarrow Stores.name$

$PurchaseItems(\underline{purchaseId}, \underline{modelName}, \underline{colour}, quantity)$
$purchaseId \rightarrow Purchases.purchaseId$
$(modelName, colour) \rightarrow Items.(modelName, colour)$

$CardPurchases(\underline{purchaseId}, cardNumber, cardType)$
$purchaseId \rightarrow Purchases.purchaseId$

Question 2. 11 p

a) i) After considering the closures of all subsets of attributes, we find the following additional non-trivial FDs:

$AC \rightarrow D$
$BC \rightarrow A$
$BC \rightarrow D$
$ABC \rightarrow D$
$BCD \rightarrow A$

Superkeys are: BC, ABC, BCD, ABCD. There is one keys: BC.

FDs that violate BCNF:

$BD \rightarrow A$
$C \rightarrow D$
$AC \rightarrow D$

ii) — By first decomposing on $BD \rightarrow A$, we get $R_1(A, B, D)$ and $R_2(B, C, D)$. R_2 is not in BCNF, so we must decompose further.

— By first decomposing on $C \rightarrow D$, we get $R_1(C, D)$ and $R_2(A, B, C)$.

b) MVDs:

$course \twoheadrightarrow teacher$
$course \twoheadrightarrow author$

(Kemp,Kleinberg,TIN092) must also be added.

teacher	**course**
Kemp	TDA357
Broberg	TDA357
Holmström	TIN092
Kemp	TIN092

author	**course**
Garcia-Molina	TDA357
Ullman	TDA357
Widom	TDA357
Tardos	TIN092
Kleinberg	TIN092

Question 3. 5 p

a) i) $\pi_{name,city}(\sigma_{salary>30000ANDbranch=number}(Employees \times Branches))$

ii) $\pi_{name,city}(\sigma_{salary>30000}(Employees) \bowtie_{branch=number} Branches)$

b) $\tau_{avSal}(\gamma_{branch,AVG(salary)\rightarrow avSal}(Employees)$

Question 4. 8 p

a)
```
SELECT  SUM(quantity)
FROM    Sales
WHERE   item='p001' AND year=2008 AND month<7
```

b)
```
SELECT  N.item
FROM    Sales N, Sales D
WHERE   N.item = D.item
        N.country = 'Norway'
        AND D.country = 'Denmark'
        AND N.Year = 2009
        AND D.Year = 2009
        AND N.Month = 3
        AND D.Month = 3
        AND N.quantity > D.quantity
```

c)
```
CREATE VIEW V AS
  SELECT        item, SUM(quantity) AS quantity, country, year
  FROM          Sales
  GROUP BY      item, country, year
```

d)
```
SELECT  DISTINCT C.country
FROM    Sales C
WHERE   NOT EXISTS (
        SELECT  *
        FROM    Sales J, Sales D
        WHERE   J.item = 'p001'
                AND D.item = 'p001'
                AND J.country = C.country
                AND D.country = C.country
                AND J.quantity <= D.quantity )
```

Question 5. a) i) $Papers(\underline{refNo}, title)$
10 p

$PaperAuthors(\underline{refNo}, \underline{email}, name, institute)$

  $refNo \rightarrow Papers.refNo$

$Reviewers(\underline{refNo}, \underline{email})$

  $refNo \rightarrow Papers.refNo$

  $email \rightarrow ProgrammeCommittee.email$

$ProgrammeCommittee(\underline{email}, name, institute)$

```
CREATE TABLE Papers (
    refNo            INT PRIMARY KEY,
    title            VARCHAR(100)
);

CREATE TABLE PaperAuthors (
    refNo            INT,
    email            VARCHAR(50),
    name             VARCHAR(30),
    institute        VARCHAR(30),
    PRIMARY KEY (refNo, email)
    FOREIGN KEY refNo REFERENCES Papers(refNo)
        ON DELETE CASCADE
        ON UPDATE CASCADE
};

CREATE TABLE ProgrammeCommittee (
    email            VARCHAR(50) PRIMARY KEY,
    institute        VARCHAR(30)
);

CREATE TABLE Reviewers (
    refNo            INT,
    email            VARCHAR(50),
    PRIMARY KEY (refNo, email)
    FOREIGN KEY refNo REFERENCES Papers(refNo)
        ON DELETE CASCADE
        ON UPDATE CASCADE
    FOREIGN KEY email REFERENCES ProgrammeCommittee(email)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

ii) Here are some suggestions, but some other policies will be accepted if these are well motivated.

If a paper's reference number is updated, then we want to propagate that change to the referencing relations. If a paper is deleted from the Papers relation, then information about the paper's authors should be deleted from the PaperAuthors relation, and records about reviewers assigned to review that paper should be deleted from the Reviewers relation.

If a programme committee member's e-mail address is updated, then we want to propagate that change to the Reviewers relation. If a programme committee member is deleted from the database, then we want all records about that programme committee member to be deleted from the Reviewers relation.

b)
```
CREATE ASSERTION NotReviewingOwnPaper CHECK
  ( NOT EXISTS (
        SELECT *
        FROM   Reviewers R, PaperAuthors A
        WHERE  A.refNo = R.refNo
               AND A.email = R.email
  ) );
```

c)
```
CREATE VIEW Conflicts AS
  SELECT  refNo, M.email as email
  FROM    ProgrammeCommittee M, PaperAuthors A
  WHERE   M.institute = A.institute
```

Question 6. 4 p

a) There are 6 possible orderings of transactions T1, T2 and T3, and all result in a balance of 400 in both accounts. In those cases where T3 executes before T2, the value printed is 1000. In other cases, the value printed is 800.

b) All of the outcomes in part (a) are still possible. Additionally, T2 may see data written by T1 as T1 writes it. So T2 can get value 500 or 600 into Y.

Question 7. 6 p

a) See lecture slides for "Lecture 12, Indexes" at http://www.cs.chalmers.se/Cs/Grundutb/Kurser/dbas/DbasHT2008/index.cgi/Lecture_Notes (in particular: page 3, slide 4; page 4, slide 1)

b) Various answers will be accepted. For example, consider the effect of using an index if the relation is so small that it fits in a single disc block. Also, consider the case where there are many more insert operations than data access requests. Solutions should discuss the disc blocks that might need to be transferred between disc and main memory in these cases (or any other case that is suggested in answer to this question).

Question 8. 6 p

a)

```
<!DOCTYPE Universities [

<!ELEMENT Universities (City*)>
<!ELEMENT City (University*)>
<!ELEMENT University EMPTY>

<!ATTLIST City
  name  ID      #REQUIRED >

<!ATTLIST University
  name  ID      #REQUIRED
  type  CDATA   #REQUIRED >
] >
```

b)

```
/Universities/City/University/@name

/Universities/City/University[@type="Technical"]
```

c)

```
FOR     $c IN /Universities/City
FOR     $u IN $c/University
WHERE   $u/@type = "Technical"
RETURN  <Result>{$c/@name}: {$u/@name}</Result>
```