CHALMERS UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering

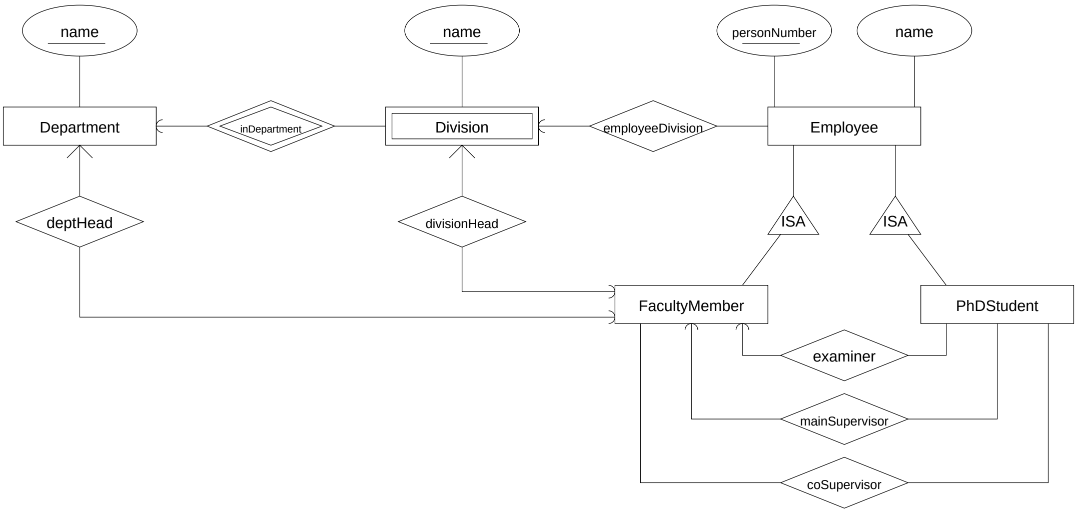
Examination in Databases, TDA357/DIT620

Saturday 18 December 2010, 08:30-12:30

Solutions

Updated 2011-12-07

Question 1. a) E-R diagram:
12 p



b) $Departments(\underline{name}, head)$
$head \rightarrow FacultyMembers.personNumber$

$Divisions(\underline{dept}, \underline{name}, head)$
$dept \rightarrow Departments.name$
$head \rightarrow FacultyMembers.personNumber$

$Employees(\underline{personNumber}, name, dept, division)$
$(dept, division) \rightarrow Divisions.(dept, name)$

$FacultyMembers(\underline{personNumber})$
$personNumber \rightarrow Employees.personNumber$

$PhDStudents(\underline{personNumber}, examiner, mainSupervisor)$
$personNumber \rightarrow Employees.personNumber$
$examiner \rightarrow FacultyMembers.personNumber$
$mainSupervisor \rightarrow FacultyMembers.personNumber$

$CoSupervisors(\underline{student}, \underline{supervisor})$
$student \rightarrow PhDStudents.personNumber$
$supervisor \rightarrow FacultyMembers.personNumber$

Question 2.
12 p

```
a) i)   AB    is neither
   ii)  ABD   is a key (and superkey)
   iii) ABE   is a superkey
   iv)  ACD   is a key (and superkey)
   v)   AE    is a key (and superkey)
   vi)  BCDEF is neither

b) i)   All violate BCNF, since none has a superkey on the left hand side.

  ii)   Decompose on AB -> C
        {AB}+ = {ABCF}

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

        Decompose R1 on B -> F
        {B}+ = {BF}

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

        Decompose R2 on E -> B
        {E}+ = {BDE}

                R21(B,D,_E)
                R22(A,E)
                    E -> R21.E

        Should update references to decomposed relations.


c) i)   B -> F
        E -> F


 ii)    Compute the minimal basis

                Remove E->F since we have E->B and B->F

        Group together FDs with the same LHS

                E -> BD

        For each group, create a relation with the LHS as the key.

                R1(_A,_B,C)
                R2(_C,_D,E)
                R3(B,_E,D)
                R4(_B,F)

        If no relation contains a key of R, add one relation containing only
        a key of R.

                R5(_A,_E)
```

Question 3. 10 p

a) $Exams(\underline{course}, \underline{examDate}, examTime)$
$Students(\underline{studentId}, name)$
$registeredFor(\underline{student}, \underline{course}, \underline{examDate})$
$\quad student \rightarrow Students.studentId$
$\quad (course, examDate) \rightarrow Exams.(course, examDate)$

```
CREATE TABLE Exams (
    course   CHAR(6),
    examDate DATE,
    examTime CHAR(2) CHECK examTime IN ('AM', 'PM'),
    PRIMARY KEY (course, examDate)
);

CREATE TABLE Students (
    studentId CHAR(10) PRIMARY KEY,
    name      VARCHAR(30)
);

CREATE TABLE registeredFor (
    student  CHAR(10),
    course   CHAR(6),
    examDate DATE,
    PRIMARY KEY (student, course, examDate),
    FOREIGN KEY (student) REFERENCES Students(studentId)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (course, examDate) REFERENCES Exams(course, examDate)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

b)

```
CREATE ASSERTION AtMostTwoExams CHECK
  ( NOT EXISTS
      ( SELECT   student
        FROM     registeredFor
        GROUP BY student, examDate
        HAVING   COUNT(course) > 2
      )
  )
```

c)
```
CREATE TRIGGER FixClash
BEFORE INSERT ON registeredFor
REFERENCING NEW ROW AS new
FOR EACH ROW
DECLARE et CHAR(2)
WHEN ( EXISTS (
        SELECT E1.course
        FROM (registeredFor NATURAL JOIN Exams) E1, Exams E2
        WHERE student = new.student
              AND E2.course = new.course
              AND E1.examDate = new.examDate
              AND E1.course <> E2.course
              AND E1.examDate = E2.examDate
              AND E1.examTime = E2.examTime
     ) )
BEGIN
    SELECT examTime INTO et
    FROM   Exams
    WHERE  course = new.course AND examDate = new.examDate;

    IF (et = 'AM') THEN
        INSERT INTO SpecialExams
          VALUES(new.student, new.course, new.examDate, 'PM');
    ELSE
        INSERT INTO SpecialExams
          VALUES(new.student, new.course, new.examDate, 'AM');
    END IF;
END;
```

Question 4. 6 p

a) $\pi_{name}(Students \bowtie_{studentId=student} (\sigma_{course='TDA357' \wedge examData='2010-12-18'}(registeredFor)))$

b) $\tau_{course}(\gamma_{course,AVG(nrSt)\rightarrow avgSt}(\gamma_{course,examDate,COUNT(student)\rightarrow nrSt}(registeredFor)))$

Question 5. 8 p

a)

```
SELECT   name
FROM     Students, registeredFor A, registeredFor B
WHERE    studentId = A.student
         AND A.student = B.student
         AND A.course = ’TDA357’
         AND B.course = ’TIN092’
         AND A.examDate = B.examDate
ORDER BY name
```

b)

```
SELECT  examDate
FROM    Exams A
WHERE   course = ’TDA357’
        AND NOT EXISTS (
                SELECT  course
                FROM    Exams B
                WHERE  A.examDate = B.examDate
                       AND B.course <> ’TDA357’ )
```

c)

```
CREATE VIEW V AS
    WITH ExamCounts AS (
        SELECT  course, examDate, COUNT(student) AS nrSt
        FROM    registeredFor
        GROUP BY course, examDate )
    SELECT  course, AVG(nrSt) as avgSt
    FROM    ExamCounts
    GROUP BY course
```

Question 6. 4 p

a) See section 6.6.4 of the course textbook.

b) See the fourth slide on page 7 of the lecture notes for lecture 10.

Question 7. See the first slide on page 4 of the lecture notes for lecture 12.
3 p

Question 8. 5 p

a)
```
<?xml version="1.0" standalone="yes" ?>

<!DOCTYPE Exams [

<!ELEMENT Exams (Exam*) >

<!ELEMENT Exam (Student*) >
  <!ATTLIST Exam
    course CDATA #REQUIRED
    room   CDATA #IMPLIED
    date   CDATA #REQUIRED >

<!ELEMENT Student (Points) >
  <!ATTLIST Student
    studentId CDATA #REQUIRED >

<!ELEMENT Points (#PCDATA) >

]>
```

b)
```
/Exams/Exam[@course="TDA357"]/Student
```

c)
```
for $e in /Exams/Exam[@course="TDA357"],
    $s in $e/Student[Points > 24]
order $s/@studentId
return <DatabasesPass studentId="{$s/@studentId}" date="{$e/@date}" />
```