Database Exam Answers
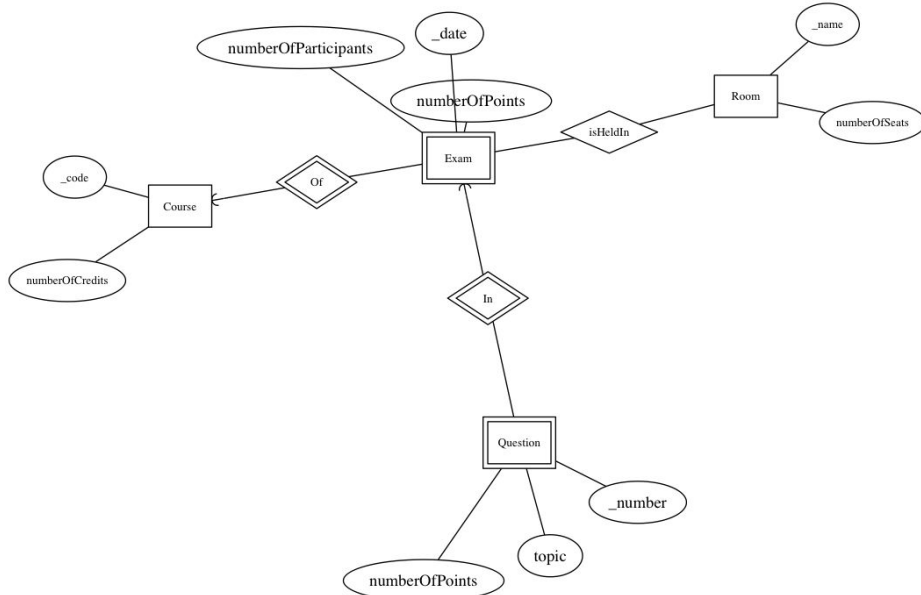
Gothenburg 17 March 2017
Aarne Ranta

Question 1.


```
Course(_code,numberOfCredits)

Exam(_date,numberOfPoints,numberOfParticipants,_courseCode)
  courseCode -> Course.code

Question(_number,topic,numberOfPoints,_examDate,_examCourseCode)
  examDate -> Exam.date
  examCourseCode -> Exam.courseCode

Room(_name,numberOfSeats)

isHeldIn(_examDate,_examCourseCode,_roomName)
  examDate -> Exam.date
  examCourseCode -> Exam.courseCode
  roomName -> Room.name
```

Question 2

a. R(country,population,river,length).

A country determines its population and a river determines its length, but there are no other dependencies.

b. We can derive all weakings of the left hand side and all trivial dependencies. Examples: AB -> B, AC -> B, AC -> D, ABC -> D, AC->ABCD. The only key is AC.

c. BCNF decomposition on A->B gives AB and ACD. The latter is decomposed on C->D to CD and CA.

3NF decomposition gives AB and CD as the base FDs. None of these contains the key, so we add AC. The result is the same as with BCNF (since we consider AC = CA).

Question 3

a.

```
INSERT INTO ExamQuestions VALUES
('TDA357',today(),3,12,'SQL queries')

INSERT INTO ExamQuestions VALUES
('TDA357',today(),4,8,'Relational algebra')

-- today() can be replaced by any
reasonable-looking indication of date
```

b.

```
SELECT topic
FROM ExamQuestions
WHERE courseCode='TDA357' AND date=today()
```

c.

```
SELECT SUM(points)
FROM ExamQuestions
WHERE courseCode='TDA357' AND date=today()
```
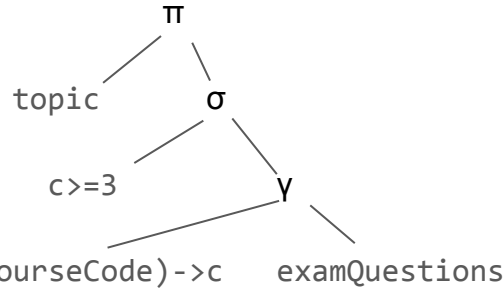
d.

```
SELECT topic
FROM ExamQuestions
GROUP BY topic
HAVING COUNT(DISTINCT courseCode) >= 3

-- leaving out DISTINCT is not quite correct
```

Question 4

a.

π
    topic   σ

c>=3   γ

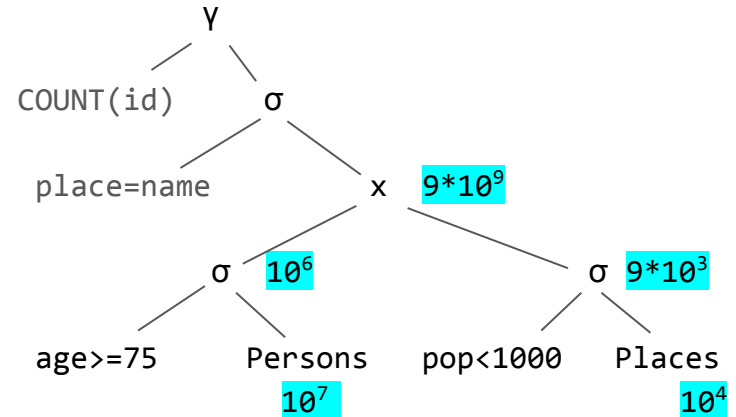topic,COUNT(δ courseCode)->c   examQuestions

-- leaving out δ is forgiven, because it is not in the cheatsheet grammar

-- the renaming of COUNT can be done in different ways or omitted

-- the formula can be a tree or use subscripts

b.

γ

COUNT(id)   σ

place=name   x $9*10^9$

σ 10^6 σ $9*10^3$

age>=75 Persons pop<1000 Places

10^7 10^4

Question 5

```
CREATE TABLE Courses (
  code Text PRIMARY KEY
  )


CREATE TABLE Exams (
  course Text REFERENCES Courses(code),
  date Date.
  PRIMARY KEY (course,date)                    a
  )


CREATE TABLE Questions (
  course Text,
  date Date,
  number INT,
  qtext TEXT,
  (course,date) REFERENCES Exams(course,date),
  PRIMARY KEY (course,date,number),            c
  (course,date,number,qtext) UNIQUE,           d
  CHECK number BETWEEN 1 AND 10                 e
  )
```

```
CREATE View ExamPoints AS (                    b
  SELECT (course, date, SUM(points))
  FROM Questions
  GROUP BY (course,date)
  )
```

Possible solutions to f:

- Exams has five columns for questions, which are NOT NULL; extra questions are added separately. This is a clumsy solution but should work.

- A trigger might check that a question is not deleted if its exam would have less than 5 questions after deletion. This is a good partial solution and gives a point, but doesn't solve how to get started with an exam.

- Checking the foreign key constraint of Questions could be made DEFERRABLE and a transaction could be used to insert into a view where Exams have columns for five questions. (complete code not required)

e could also be solved by a trigger on insert in Questions.

Question 6

a. Non-repeatable read, permitted by READ COMMITTED
b. A customer needs SELECT on Persons(id,name) and Employees(id,position). However, this can be minimized by creating a view Positions(name,position) or even Managers(name), and give the customer SELECT rights only on this.

   An administrator needs SELECT on Employees(id) and UPDATE on Employees(salary)

c. The element should have a root FT node. If we see this as the family tree of the current royal family of Sweden, we wrap the child FT nodes between <FT name="CarlGustav"> and </FT>