**University of Helsinki, Department of Computer Science**
**Introduction to Databases, 16.4.2004, H. Laine**

**1.** Let's consider the tables of a recipe database:

```
course (courseID, name, easeOfPreparation, noOfServings, cookingTime) [800 rows]
categories (course->course, category) {2400 rows]
material (materialID, name, type, unit, unitPrice) [400 rows]
ingredients(course->course, materialID->material, amount ) [8000 rows]
instruction(course->course, phaseNo, description) [8000 rows]
```

Categories of courses include *soup*, *salad*, *appetiser*, *dessert* and *main course*.

a) Is the natural join **course * material** possible, and, if it is, give your estimate on the number of rows in the result?
b) How many rows there are in **course** $\rhd\lhd_{\text{courseID=course}}$ **instruction**?
c) What can you tell about the result of **instruction – ingredients**?
d) Give your estimate on the number of rows in **instruction** $\rhd\lhd_{\text{instruction.course=ingredients.course}}$ **ingredients**?
e) What is the relation of cardinalities of $\pi_{\text{materialID}}$**(ingredients)** and $\pi_{\text{materialID}}$**(material)** (10p)

A passage control system uses personal key cards. A card is controlled each time when one wants to pass a gate. Each control is registered as a control record. If the gate is not opened, the reason for the denial is also registered. This system uses the following tables:

```
person (personID, name, address, task)
keycard (cardNo, dateAdmitted, owner->person,
    whoAdmitted->person, expirationDate)
gate (gateNo, location)
control(cardNo->keycard, whenControlled,
  gateNo->gate, action, reason)
permission(gateNo ->gate, cardNo ->keycard, startingDate,
    endingDate, typeOfPermission, whoAdmitted->person, dateAdmitted)
biggest_cardNo(cNo)
biggest_gateNo(gNo)
```

Tables biggest_cardNo and biggest_gateNo contain the biggest card and gate numbers that are in use. TypeOf Permission has values like 'officehours','evening', and 'week-end'. A previous value in the list is assumed to be contained in the succeeding value, i.e. permission 'evening' contains permission 'officehours'. Action in table control is 'denied' if passage is denied. The reason may be, for example, 'no_permission' or 'card_unreadable'. All permissions have some non null endingDate.

**2.** Give the following queries in SQL. Define a proper order for the results.
a) Make a list of persons to whom Bob Cop has during this year admitted a week-end type permission to pass gate 31.
b) Make a list of gates that have no week-end type permissions.
c) Make a list of persons that have admitted permissions for themselves.(9p)

***Tasks 3-5 on the other side.***

**3.** Give the following queries in SQL. Define a proper order for the results.

a) Make a report on gate passage denials listing the number of denials for each gate and reason ordered so that most frequent gate reason pairs are first.
b) Which gate has the biggest number of card reading failures?
c) List the persons that have been admitted more than one key card this year. Give also the number of cards they have achieved (9p)

**4.** A new gate is included in the system. Its location will be the 'coffee room'. All owners of valid key cards are admitted week-end type passage through this new gate up to the end of this year. The permission applies from now on. Pseudo-person '010101-0101' should be registered as the one who admitted this permission. This 'person' has already a record in the database. What operations are needed to register the change. Give the operations in SQL. (8p)

**5.** The following table has been designed for registering the results of competitions of a fisherman society:

```
catch (competitionID, location, dateOfCompetition, fishermanID,
        firhermanName, fishType, yearOfBirth,
        fishType_total_weight_in_catch,
        fishType_total_number_in_catch,
        ranking_in_competition)
```

a) Explain the meaning of the functional dependency
***fishermanID -> ranking_in_competition*** ?
b) How would you express the rule 'a fisherman may participate in only one competition on the same date'?
c) Let there be functional dependencies:
- competitionID -> location
- competitionID -> dateOfCompetition
- fishermanID -> fishermanName
- fishermanID -> yearOfBirth
- competitionID, fishermanID -> ranking_in_competition
- competitionID, fishermanID,fishType -> fishType_total_weight_in_catch
- competitionID, fishermanID,fishType -> fishType_total_number_in_catch

Give the database schema in Boyce-Codd normal form (9p)

***Tasks 1-2 on the other side***