UNIVERSITY OF HELSINKI
DEPARTMENT OF COMPUTER SCIENCE
Introduction to databases, course exam 4.3.2004

Please, write the name of the course, date of exam, your name, date of birth and signature on each separate answer paper.

1. Let's consider the tables:

```
Hotel(hotelID, name, address, phone)
RoomReservation(reservationNumber, hotelID->Hotel, name, address, phone,
   dateOfBirth, arrivalDate, duration, whenMade)
Room(hotelID->Hotel,roomNumber, facilities)
Occupation((hotelID,roomNumber)->Room, reservationNumber->RoomReservation,
   arrival, departure)
```

Table Hotel has 20, RoomReservation 20000, Room 1000 and Occupation X rows. Personal information about customers is stored in RoomReservation table. Old reservations are kept for some time as well as old occupation records.

a) What is the cardinality of the natural join Hotel*RoomResrvation?
0. Join is based on columns hotelID, name, address, and phone. In Hotel name. address and phone are hotel information in RoomReservation they are personal information. Thus there will be no matches.

b) Is the number of rows in table Occupation less, equal or greater than the number of rows in RoomReservation? Why?
Occupation is probably smaller. There must be one row in roomReservation for each Occupation. There may be future reservations. There may also be more than one occupation for a reservation but their number is probably less than the number of future reservations. Any number was accepted. The points were given on the basis of the motivation.

c) How are the cardinalities of $\pi_{\text{hotelID,roomNumber}}$ (Room) and $\pi_{\text{hotelID,roonNumber}}$ (Occupation) related to each other?
$|\pi_{\text{hotelID,roomNumber}} (\text{Room})| >= |\pi_{\text{hotelID,roonNumber}} (\text{Occupation})|$
All the rooms need not be occupied.

d) What is the cardinality of Hotel $\bowtie_{\text{Hotel.hotelID=Occupation.hotelID}}$ Occupation?
Same as the cardinality of Occupation =X.

e) What is the cardinality of Hotel $\bowtie_{\text{Hotel.hotelID} \neq \text{Room.hotelID}}$ Room?
Each Room row matches with 19 hotels, thus the result contains 19000 rows.

A recipe database has the following tables (tasks 2-4)

```
course (courseID, name, easeOfPreparation, noOfServings, cookingTime)
categories (courseID->course, category)
material (materialID, name, type, unit, unitPrice)
ingredients(courseID->course, materialID->material, amount )
instruction(courseID->course, phaseNo, description)
biggest_courseid(highvalue)
```

Column *noOfServings* in table *Course* indicates the number of servings in the recipe. *CookingTime* is expressed in minutes. Categories of courses include *soup*, *salad*, *appetiser* , *dessert* and *main course*.

Column *type* in table *Material* may contain values like '*fish*', '*pork*' and '*vegetable*'. *Amount* in *Ingredients* is expressed in units specified in table *Material*. (for example *kg*, *table spoon*, *apiece*). *Amount* contains the amount needed for the whole recipe. Table *biggest_courseid* contains the gratest *courseID* value in use.

2. Express the following queries in SQL. Specify an appropriate order for the result.

About evaluation of SQL tasks:
3p, the solution is essentially correct (small keyword and column and table name errors are allowed as well as some non-essential errors)
2p, some essential part is missing or incorrect, but basic idea is OK.
1p, There is something positive in the solutions.
0p No idea.
Missing ordering -1 point but only once. Same error counted only once if possible.

a) Make a list of soups that suit for appetisers.

```
select courseID, name
from  course, categories soup, categories appetiser
where course.courseID = soup.courseID and
      course.courseID = appetiser. courseID and
      soup.category='soup' and
      appetiser.category= 'appetiser'
order by nimi.
```

b) What materials are used in some appetiser but in no dessert?

```
select name
from  material
where materialID in
       (select materialID
        from ingredients, categories
        where ingredients.courseID= categories.courseID and
              categories.category='appetiser'
       )
and
materialID not in
   (select materialID
        from ingredients, categories
        where ingredients.courseID= categories.courseID and
              categories.category='dessert'
   )
order by name
```

c) List the recipes (courses) in which the amount of some material is missing or the unit for the amount is not specified. (9p)

```
select name
from course
where courseID in
   (select courseID
    from  ingredients
    where amount is null or
    materialID in
        (select materialID from material where unit is null)
   )
order by name
```

Turn for the tasks 3 - 5

3. Express the following queries in SQL. Specify an appropriate order for the result.

a) What are the material costs for one serving of cabbage casserole?

```
 select sum (unitprice*amount/noOfServings)
   from course, ingredients, material
   where course.name='cabbage casserole' and
     course.courseID=ingredients.courseID and
     ingredients.materialID=material.materialID;
```

b) Which recipe includes the highest number of ingredients? Give the name of the course and the number of its ingredients.

```
select  course.courseID, name, count(*)
from course, ingredients
where course.courseID= ingredient.courseID
group by course.courseID, name
having count(*) >=
  (select  count(*)
   from course, ingredients
   where course.courseID= ingredient.courseID
   group by course.courseID, name)
```

c) Make a report that shows for each type of course (appetiser, main course and dessert) the number of recipes in categories determined by the time of preparation considered on the precision on 10 minutes (1-9 minutes ids the first category, 10-19 the next one, etc). Hint: function *trunc*(*expression*) truncates the *expression* to an integer.. (9p)

```
select category, trunc(cookingTime /10) timegroup, count(*)
from course, categories
where course.courseID=categories.courseID and
    category in ('appetiser','main course','dessert')
group by category, timegroup
```

.

4. A new delicious variant (pea soup with tomatoes) has been developed for the pea soup (course 10335). This adds 2 tomatoes to the recipe of the original pea soup. Tomatoes are already stored in the database as material (materialID A332). What operations are needed to register the variant? You need not consider how the instructions change. Give the operation also in SQL (8p)

```
// increade the counter
update biggest_courseid
    Set highvalue=highvalue+1;
// cpy the basic information of pea soup to the new course record
// replace the courseId and name – new courseid from biggest_courseid
insert into course
     select highvalue, 'pea soup with tomatoes', easeOfPreparation,
         noOfServings, cookingTime
     from biggest_courseid, course
     where courseid=10335';
// copy the ingredients of the original pea soup
// as ingredients of the new course,
// substitute the courseid with the courseid of the new course
insert into ingredients
    select highvalue, materialID, amount
```

```
    from biggest_courseid, ingredients
    where  ingrediets.courseid=10335;
// append the tomatoes to the ingredients of the new course
insert into ingredients
    select highvalue, 'A332', 2
    from biggest_courseid;
// end the transaction.
commit;
```

Grading

If needed operations are explained correctly but no SQL : 4p.
If SQL statements are OK but no explanation 8p.
Commit missing: -2p.
Biggest_courseid not used or updated :-2p
Multiple operations trine in a single statement -3p
Try to insert with update -2p
Major syntactic errors: -2p;

5. A video rental service has designed the following table to run their business:
copy (copy_id, movie_id, movie_name, director, yearMade, medium, dateRented, whoRentedID, whoRentedName, chargeForDay)

If the copy is not rented whoRentedID and whoRentedName are null and dateRented is '1.1.2200'.

a) What does the functional dependency ***director→movie_name*** mean in practice?
There is only a single movie title for each director ('there only one movie for each director' is also a correct interpretation).
(2p if student clearly understands the dependency, 1p: if not explained well, 0p: if not understood)

b) How would you express the rule 'There is only one movie made on the same year for each director'?
director, yearMade -> movie_id (3p)
1 point of solution:
director, movie_id-> yearMade

c) Let the be the following dependencies: (4p)

- copy_id -> movie_id,
- copy_id -> medium,
- copy_id -> chargeForDay,
- movie_id -> movie_name,
- movie_id -> director,
- movie_id -> yearMade,
- whoRentedID -> whoRentedName,
- copy_id, dateRented -> whoRentedID.

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

copy(copy_id, movie_id->movie, medium, chargeForDay)
movie(movie_id, movie_name, director, yearmade)
renter(whoRentedID, whoRentedName)

rent(copy_id->copy,dateRented, whoRentedID->renter)

The key for the original relation is copy_id, dateRented, thus there is no need for additional tables (if this is not mentioned -1p), otherwise grading depends on how much mess there is. Keys were not explicitly requested thus they need not be marked.

Turn for tasks 1 and 2