First set of Mandatory Exercises in INF3100/INF4100 Spring 2007

Formalities

All students are to hand in an individual assignment. If two students want to hand in an assignment together, they have to apply to the group teacher and get an acception for this in advance.
The assignment is to be delivered by email to the group teacher.
Detailed instructions for what to deliver is described in the section on “Deliverables” at the end of this file.

Notice: Assignments delivered after the deadline will be rejected. You must answer all questions in order to get your assignment approved.

Students that have had their assignment accepted and still choose to withdraw from the exam, have to deliver a paper copy of the assignment to their group teacher to get a written statement from the group teacher that it has been accepted. This only regards students that withdraw before the 14-days limit.

This mandatory exercise is due: Friday 23. March at 12:00 am.

Exercise 1

Prove the transitive rule for multivalued dependencies (MVDs):
If X -->> Y and Y -->> Z, then X -->> (Z \ Y)
Tip: Draw a Venn diagram before you start working on this exercise.

A natural first guess for a transitive rule for MDVs is likely to be

If X -->> Y and Y -->> Z, thenl X -->> Z

Which added constraints do we need for this rule to hold?

Exercise 2

In this exercise, you are to use the test database described in filmdatabasedok.pdf (http://www.uio.no/studier/emner/matnat/ifi/INF3100/v07/undervisningsmateriale/oracle/filmdatabasedok.pdf).
An English summary can be found in filmdatabasedok_eng.pdf (http://www.uio.no/studier/emner/matnat/ifi/INF3100/v07/undervisningsmateriale/oracle/filmdatabasedok_eng.pdf).

Use SQL against the film database to answer the following questions:

1. How many films are registered in the database?

2. The films are divided into rating categories. How many films are there in each category?
3. The films are divided into genres. Make a list over all the genres and the percentage of the films in each genre. The list should be sorted descending by percentage.
4. How many (distinct) actors are there in the database? What percentage of these is female, and male?
5. List title, production year, director and producer for all films from the first year that the database has registered films from (i.e. from the oldest films).
6. Find first and last names of all female directors that have made more than 5 films, and that have used one and the same actor in all their films (which may be a lot more than 5).

Exercise 3

In this exercise, we are looking at computing point scores in team sports like handball and soccer, i.e. computing points in *leagues.*

One league consists of a set of teams that play each other in pairs. Each team has its own game field; this is called the home field of the team. Each team meets each of the other teams for matches twice, once on their own home field, and once on the other team's home field. For each match, the number of goals scored by the home team is registered, as well as the number of goals scored by the visiting team.

In each match, points are given to the teams. If the teams score the same number of goals, one point is given to each of the teams. If not, the team scoring the highest number of goals get 3 points, while the other team get no points. The winner is the team having the highest point score after all matches have been played.

If more than one team get the same number of points, they are ranged according to goal difference (i.e. the sum of the number of goals the team has scored, minus the sum of the number of goals scored by the opponent team in the same match). If there are more than one team having the same point score and the same goal difference, the team that has scored the highest number of goals comes first.

You are now to create a small database in Oracle and then make a program in Java that uses SQL to register match results, and a sorted result list. It is allowed to use C++ if this is preferred, but we cannot give any help with C++ related questions, only Java. The database is to consist of two base relations:

1. One table for the teams participating in the series. The table should have three attributes:
    - A primary key (an unambiguous team number ranging from 1 to the number of teams in the series)
    - An unambiguous name for the team
    - The name of the home field of the team
2. One table for the match results. This table has four integer attributes:
    - The number of the home team (foreign key to the team table)
    - The number of the visiting team (foreign key to the team table)

    - The number of goals scored by the home team
    - The number of goals scored by the visiting team

The results are to be presented in a table having one row for each team in the league and four columns:

- The name of the team
- Sum of goals the team has scored in all its matches
- Sum of goals scored by the opponents in the team matches
- Sum of points the team has been given

The table should be sorted with the best team at the top, and the poorest at the bottom.

What to do/what has to be done

1. Define the two base relations in Oracle. Explain which keys and foreign keys they have, and how these can (or can not) be enforced. It is *not* required that you enforce all integrity constraints that should exist in the database.
2. The following is to be solved using Java against Oracle:
    1. Register five teams in the team table. You have to think up names for the teams and their home fields yourselves.
    2. Register the following data in the table of matches (the results from a small handball league):

Home team	**Visiting team**	**Home goals**	**Visiting goals**
1	2	19	19
1	3	22	17
1	4	17	7
1	5	22	13
2	1	22	22
2	3	14	14
2	4	16	15
2	5	13	13
3	1	13	20
3	2	16	16
3	4	21	20
3	5	17	17
4	1	20	19
4	2	16	15
4	3	13	14
4	5	17	15
5	1	19	20

5	2	17	17
5	3	14	13
5	4	18	17

3. Use SQL to create the result table using Java (java.sql).

Deliverables:

Email to the group is to have the following in the subject field:
Subject: Oblig 1 inf3100 (<username student >)

For Exercise 1 the following is to be delivered:

A document containing the solution and Venn diagram. Format: PDF or word (doc).

For Exercise 2 the following is to be delivered:

- • A SQL-file.
- • A result file.

For Exercise 3 the following is to be delivered:

- • A SQL-file for database creation.
- • Java code (.java files). The code should be readable and runable.
- • A file showing the results from running the program.

Common for exercises 2 and 3 (regarding delivery of SQL-parts):

The SQL-file should have the name <username student>.<nr>.sql (where <nr> stands for 2 and 3 for the two exercises).
It should be possible to run the file using the start command in SQL*plus.
The first line in the file should be the same as in the subject field of the email, but in the form of a SQL comment:
/* Oblig 1 inf3100 (<username student >)*/
All comments in the file should be enclosed by this kind of SQL comment parentheses.
Comments regarding the solution in general should be put first in the file (after the first line).

The results from one run of this SQL file is also to be delivered.
The result file should be named <username student>.res.
The first line in this file is to be identical to the first line in the SQL-file.

End of mandatory exercise set 1.