

**Mandatory deliverable: Part 3**

**Counts for 40% of the overall semester project.**

**Due date: 21.03.2011**

In this part of the semester project you are going to build on the data model you created in Part 1, and the stored procedures you created in Part 2.

## **1. PL/SQL packages, procedures, functions**

In the file skiklubben3.sql you will find the DDL queries to create a PL/SQL package called SKIKLUBBEN\_PKG, which includes the specification for a number of procedures and functions that constitute this package. The assignment is to create the procedures and functions themselves in a package body. You have already implemented several of these procedures in Part 2, but here you will give them a slightly different name. (For example, in Part 2 you created LEGG\_TIL\_NY\_MEDLEM\_SP; in Part 3, this procedure is called LEGG\_TIL\_NY\_MEDLEM\_PP. \_PP is a package procedure). A short description of the procedures and functions included in the package :

1. **LEGG\_TIL\_NY\_MEDLEM\_pp.** Add a new member to the database. You must create and insert a new identifier (primary key value) for the member.
2. **LEGG\_TIL\_NY\_SESONG\_pp.** Add a new season to the database. The season shall be the combination of two years, e.g., *2009-2010, 2010-2011, etc.*
3. **LEGG\_TIL\_NY\_ORGANISASJON\_pp.** Add a new organization to the database. You must create and insert a new identifier (primary key value) for the organization.
4. **LEGG\_TIL\_NY\_GRUPPE\_pp.** Add a new group to the database. The season and the group's administrator must be specified. Use the administrator's username to look up that individual's primary key value and insert this value as a foreign key.
5. **MEDLEM\_SESONG\_ALDER\_pf.** Given a season and a member username, return the age that will be used to determine what medals a member shall receive. Determine the age using the business rules shown on PPT slide (G).
6. **REGISTRER\_AKTIVITET\_pp.** Given a date, member username, activity type, course, description, distance, competition, add a new cross-country skiing activity.
7. **REGISTRER\_AKTIVITET\_pp.** Same name as previous, but with different parameters. Given a date, member username, activity type, course, description, number of jumps, competition, construction point, and longest jump, insert a new ski jump activity.
8. **REGISTRER\_AKTIVITET\_pp.** Same name as previous, but with other parameters. Given a date, member username, type of activity, description, hours, minutes, and competition, insert a new alpine skiing activity. The procedure must calculate the number of kilometers that corresponds to the number of hours (1 hour is equal to 4 kilometers).
9. **BLI\_MED\_I\_GRUPPE\_pp.** Given a username, group name, season, add the member to the group.
10. **BEREGN\_DISTANSE\_pf.** Given a username and a season, calculate the number of kilometers the member has registered in the season.

11. **GI\_UT\_MERKE\_pp.** Given a username and a season, determine what kind of medal the member will receive, and store this information in the database.
12. **GI\_UT\_ALLE\_MERKER\_pp.** Given a season, give each member the medal s/he has earned in that season. Store the member, season, and medal earned in the database.
13. **POKAL\_FORTJENERE\_pp.** Given a season, find all of the members who have earned the 3-year cup, i.e. have earned gold in the last three years in a row.
14. **REGISTRER\_GRUPPE\_AKTIVITET\_pp.** Given a date, group name, activity type, course, description, distance, competition, insert new cross-country skiing activities for all of the group's members.

Forklaring av parameterene finnes i skiklubben3.sql filen.

## 2. Database trigger

Create a trigger that updates the total number of kilometers for a given member for a given season, whenever a new activity is inserted for that member. The trigger may use the function `BEREGN_DISTANSE_pf` and store the result in an appropriate place.

## 3. Spørringer

Create sql-queries for the following :

1. Write a query that lists the total kilometers for each municipality in a given fylke between two dates (e.g. between 1-FEB-2005 and 15-FEB-2005), broken down by type of activity. The output should include the following columns  
kommune, grentype, total kilometer.
2. Write a query that lists the total kilometers for a given group. The total consists of the kilometers of all of the group members. The output should include the following columns:  
gruppe, sesong, total
3. Write a query that lists the total kilometers in a given season for all organizations a give member is associated with. The total consists of all of the kilometers of all members associated with the same organizations. The output should include the following columns:  
organization, season, total.

## General

The deliverable should consist of

1. The following SQL-scripts
  - a. SQL script for creating database tables and constraints or other changes that you have made to the database schema since the last assignment.
  - b. SQL-script for creating the package body, procedures, the trigger, and any other Oracle objects created as part of the solution.

- c. SQL-script with queries for task 3
- 2. A document that explains any assumptions you have made
- 3. All database objects shall be created in the group's user account on the class's Oracle server (IS203/dbadb on oracle.ist.unomaha.edu).
- 4. Each member of the group will also submit a reflection. Answer two questions among those found at <http://escalate.ac.uk/resources/reflection/09.html>