Project schema

Name (or names, if working in a pair): Joachim Mjelde, Thor Madsen

Here you will plan what your project will look like, in terms of the database and the application that will use it. This is all preliminary, and it's OK if some details change by the time you are done with the project! (With your final submission you will turn in an updated version of this reflecting any changes that have happened.)

Project description (10 points)

Give a brief overall description of your project. What is it for?
Our project will be a database of the athletes on the Westminster Ski Team, their skis, and the brands of the skis. The user will be able to look up a person and get information about the person and their skis. You can also access information about the specific skis.

Database (30 points)

Tables

List the tables you intend to include in the database (meeting the requirements laid out). For each table, list the columns and say which is the primary key and what foreign keys the table will have (and to which tables). For each table, also give a few examples of what rows of that table will look like.

- Table `Person`
  - Columns: id, name, age, sex, brandId, countryId
  - Primary key: id
  - Foreign keys: brandId references brand table, countryId references Country table
  - Sample rows: 1, Frank, 21, M, 1, 4
- Table `Brand`
  - Columns: id, name, countryId
  - Primary key: id
  - Foreign keys: countryId references Country table
  - Sample rows: 1, Fischer, 1
- Table `Ski`
  - Columns: id, discipline, pairNum, brandID, personId
  - Primary key: id
  - Foreign keys: brandId references brand table, personId references Person table,
  - Sample rows: 1, SL, 2, 1, 1
- Table Country
  - Columns: id, Country
  - Primary key: id

- Foreign keys: n/a
- Sample rows: 1, Austria

Application (40 points)

Features

List the major features of your program. What will the user be able to do?

Display Name, age, sex, Brand, and number of pairs of everyone on team.
Input a brand, output how many athletes use this brand.
Input a nationality, output the athletes from that nationality, the brands they use, and number of pairs
Delete Person.
See a list of the people that have more than a decided amount of pairs. In this case 3.
Input a name, output all the skis and the ski info for that person

Language and format

What language do you intend to use (Python, Java, something else)? Do you intend to build a text-only, graphical, or web interface? If graphical or web, what toolkit/framework do you plan to use?

Language: Python
Interface: Text

Queries

For each of the major features you listed above, write down preliminary versions of whatever SQL queries will be needed to enact the feature. These do not need to be final at this point, but should use the tables you listed above. (I recommend you go ahead and make a database with your tables and some data now so that you can test your queries at this point. You will then be ahead when it comes to the rest of the project!)

To get full points, the queries you list below must meet the requirements — you must list at least one that uses a join, one that uses aggregation or a subquery, etc.

- *Feature:* Display Name, age, sex, Brand, and number of pairs of everyone on team
  - Select Person.name, Person.age, Person.sex, Brand.name as Brand, count(*) numberOfPairs
  - From Person
  - JOIN Brand on Brand.id = Person.brandId
  - join Ski on Person.id = Ski.personId

- group by Person.ida
- ;

- Feature: input a brand output how many athletes use that brand.
  - Select count(Person.name)
  - From Person
  - JOIN Brand on Brand.id = Person.brandId
  - Where Brand.name = 'fischer'
  - ;
- Feature: input a nationality output athletes from that nationality and what brands they use.
  - Select Person.name, Brand.name
  - From Person
  - JOIN Brand on Brand.id = Person.brandId
  - JOIN Country on Country.id = Person.countryId
  - Where Country.country = 'norway'
  - ;
- Feature: delete person.
  - Delete from Person where name = 'Joachim';
- Feature: See a list of the people that have more than a decided amount of pairs. In this case 3.
  - select Person.name, count(*) as count
  - FROM Person
  - JOIN Ski ON Person.id = Ski.personId
  - group by Person.id
  - having count >3
  - ;
- Feature: Input a name, output all the skis and the ski info for that person
  - select Person.name, Brand.name, Ski.discipline, Ski.pairNum
  - from Person
  - join Ski on Ski.personId = Person.id
  - join Brand on Ski.brandId = Brand.id
  - where Person.name = 'Mikkel'
  - ;

Procedural SQL

Part of the requirements of the project is that your application make use of at least one stored procedure or trigger. Write a preliminary version of the procedure or trigger below with a description of its purpose.

Purpose: Count the number of skis in each event
Code:
CREATE PROCEDURE TotalSkis(IN discipline ENUM, OUT total INTEGER)
SELECT COUNT(Person.name)
INTO total

```
FROM Person
        JOIN Ski on Ski.personId = Person.id
WHERE Ski.discipline = discipline
;
```

User input sanitization (10 points)

Write "yes, I will sanitize all user input that goes into the database" below:
Yes, I will sanitize all user input that goes into the database.

Project development plan (10 points)

If working alone, what exactly do you need to do to complete the project? Give yourself a checklist of tasks that will need to be completed.

If working with a partner, how do you intend to divide the work of the project? What will each of you contribute to the final product? (Sometimes it makes sense to divide tasks strictly between you: person A does tasks X, Y, and Z, while person B does tasks P, Q, and R. Sometimes people prefer to work together throughout the project, for example doing pair programming or deciding what each person does ad hoc *as needs arise. Whatever you want to do is fine, but* ***plan to keep a record of who does what as this will be asked on final submission.****)*

Name: Joachim
Tasks:

- Input values
- Make tables in database
- Develop queries
- Create interface
- Test application
- Create presentation video

Name: Thor
Tasks:

- Create Database
- Develop queries
- Create an interface
- Implement queries.
- Test application
- Create presentation video