Course Board/kursnmd

First of all I will use this document to reach all students with a request to join the course board. This is normally about 3-4 students that I will meet with after the course has completed to disscuss the course. It is a chance to help improve the course for next year by giving your opinion (and that of your classmates) to me. It is noramlly a very valuable source of feedback to us. If you want to do this please send me an email johnf@kth.se. Tha nk you.

Introduction

The group assignment is open ended. You are expected to think about the problem domain and design a database that covers the major needs. As the domain is very large it is not expected that you cover all but it should be reasonable to expect that the skeleton you create would be extendable to cover the whole domain. There are a few specific requirements below and your design should explicitly cover those. That is you will need to have relations containing information about Freinds Arena, Russian Men's Ice Hockey team and so on.

You will be split into groups on the first day, distribute tasks between you and then continue to work outside of class. The last recitation session is for you to meet together consolidate your work and get help, as needed, from the teachers. The final deliverable is due a week after that.

Missed first day? You will need to do the assignment none-the-less. If you can find others in your position you are free to form your own group of max five persons and form a CANVAS group as described. One person group is the fall back plan. If you send me an email before or at least before the end of the first day, I can connect you to others that may send such an email. After the first day you will not get help forming a group as the work will have started and it would not be fair to have you join a group then.

Winter Olympic Games - Stockholm 2022

Stockholm has plans to host the 2022 winter Olympic games. In order to demonstrated it is serious, the city is designing the database needed to keep track of all information on the games. That includes all teams, contestants, trainers, officials, reporters, housing, venues, sports, events, competitions, tickets, schedules, results, statistics, attendance, media coverage, and on and on.

Your task is to help with a small part of the total design. Namely data on:

1. National teams (ie. countries),
2. Sport (These each have a result of one gold one silver and one bronze medal winner)
3. Competitors: These are the atheletes that each can compete in several competitions, such as different distances for races;
4. Officials: Each official is qualified for certian sports and cna be assigned to competitions within those sports;
5. Teams: Only for sports that are team sports such as pair figure skating;
6. Competitions: These are moments within a sport in a specific place and time, with competitors and officials assigned. They can also fit into a structure such as rounds and groups, ie round 2 group C. For ex. competitor 3987203 can be in group C of round 2 for Speed skating 1,000 m on January 15, 14:30 at Ostermalms IP, main track. These competitions all take different lengths of time which is an important bit of information about them.

This part of the database should be sufficient to answer at least these questions:

1. What countries are competing and how many medals have they won?
2. What is going on at Friends Arena on Sunday?

3. What teams are competing in the women's slalom alpine ski race?
4. Where and when are the finals in the bobsleigh race being run?
5. Who are the goal keepers for Russian's men's ice hockey team?

Furthermore the database should enforce consistency:

1. contestants must belong to one and only one National team,
2. Officials should only be assigned to competitions in sports that they are qualified for.
3. No two competitions can take place at the same place at the same time. A place is not the same as arena as it includes competitions locations within the arena. The times of each competition can vary as stated above and the competition places should not stand idle just to make this constraint easier to implement. Notice that the TIMESTAMP type in SQL might make this overlapping time/date interval easier to implement in a trigger,

The above lists are required but not complete. You should try to imagine extensions to them. Some of these are very difficult indeed such as allowing atheletes to compete in all the sports they are good at. Your database design would need to work in practice with all that is implied but the entire lists of possible queries, relations, attributes, and constraints are way beyond the scope of this exercises, but do find some. It is also possible to satisfy the requirements above with a oversimple design that will run into other problems that you can imagine. Try to avoid these if you can.

Also the constraints do not have to be enforced absolutely only so that the normal way modifications would be made work. You can specify what normal is. In practice there would be an application layer and the administrators would not use SQL but rather your specified interface. I expect each of you to work productively during class and plus up to 4-8 hours outside class.

You will work in groups of 3-6 with 5 being preferred.

Each person should join the group in Canvas during the recitaion. One person should be first and then tell the others the group number. All of the group members can then join that group. (This is the first time I use Canvas in this way. If it does not work just email me a list of your names.

This assignment is much more about the process than the result. It should not take more than a few hours but all should be active in disscussions. Here is a suggested way to go about the work:

Phase 1

1. Make a list of all data (attributes) that seem to be needed without worrying about the best way of organizing it.
2. Test this list on the above criteria to see if the information is sufficient to, in principle, answer the questions and to ensure the consistency.
3. Repeat steps 1 and 2 until your list seems complete
4. Make a list of functional dependencies that you believe exist between the attributes.
5. Eliminate those FD's that follow from the others.
6. Repeat 4 and 5 until you feel you have formed a basis set of FD's for this database.
7. Create a 3NF Decomposition of your partial Olympic database.
8. List some other constraints that must be added, ie foreign keys, not null, value ranges,....

Some of these are best done separately by each team member and then combined into a more complete picture. So for example two memebers might focus on teams and contestants while others might think about the venues. Split the work as you see fit but do meet outside of class to combine your work and move on to the next step.

Phase 2

After this you are ready to create a preliminary ER diagrams over these attributes by attaching them to entities with relationships. Remember the design principles:

1. Faithful, should meet our specs.
2. Not Redundant, should not have data repeated anywhere.
3. Simple, Eliminate any unnecessary entities or relationships.

Also indicate:

1. multiplicity (pointy arrows);
2. referential integrity (Round arrows);
3. weak entities (double outlined shapes);
4. subclasses (isa triangles).

Phase 3

1. Decide how the ER diagram should best be turned into relations and create the SQL Schema.
2. Add some: checks, constraints and/or triggers to enforce consistency in the schema.
3. Implement in SQL
4. check by running queries

Phase 4
The final deliverable is in the form of one report per group containing:

1. group members names and *@kth.se email addresses;
2. the SQL database schema (list of relation names);
3. relation schemas (CREATE TABLE commands in a create.sql type file as in lab 1). Also fill the tables with som data as in 'create.sql';
4. ER diagram;
5. 3 paragraphs, one on each of the consistency requirements below explaining how your design satisfies them;
6. the SQL queries for the four required questions;
7. at least four additional SQL queries of your own devising;

It might very well also contain constraints, and triggers. It might also disscuss FD's and normal forms.

In addition, you should have implemented some part of the database in SQL enough to test at least one of the queries. Upload:

1. The report as a pdf with the diagrams, and explanations. This can also contain the schema, queries and output but also include the create and queries as text (.sql) files that can be run.
2. your sql commands in a text file(s) as in lab 1.
3. the output of running the code (create.sql followed by your queries.sql) in a separte text file if this was not in the pdf.

(You are free to use other database implementation but should provide code to show that your implementation worked on the query(ies)).

Grade:

The grade is pass fail and is based on exactly the stated requirements above. If the report is lacking in some requirement do not upload it until it covers all of them. If we do not agree that it covers all we will send it back for completion. The amount of work should not present a problem to make the deadline, but if you miss it your report will eventually be graded but will get lower priority. Besides that you are better off getting this done quickly than dragging it into exam time.