DEVELOPING AN APPLICATION FOR ACCESSING A DATABASE

This project consists of the development of a desktop application for allowing the access to a database (DB). Unless a specific permission of the professor in charge, the application will be developed in teams of 3 people. Every team member must belong to the same lab group. The teams will be formed on the corresponding activity on the virtual campus. The maximum grade which can be achieved is 15 out of the 100 total points of the course.

Goal

The system to be developed consists of an application to manage the setlists of a series of concerts stored in a database (DB). The DB to be managed will contain a series of tables to store information about artists, songs, albums, venues, and setlists.

The following figure describes the relational schema of the DB to be used. A file containing this schema of the DB can be found on the virtual campus of the course.

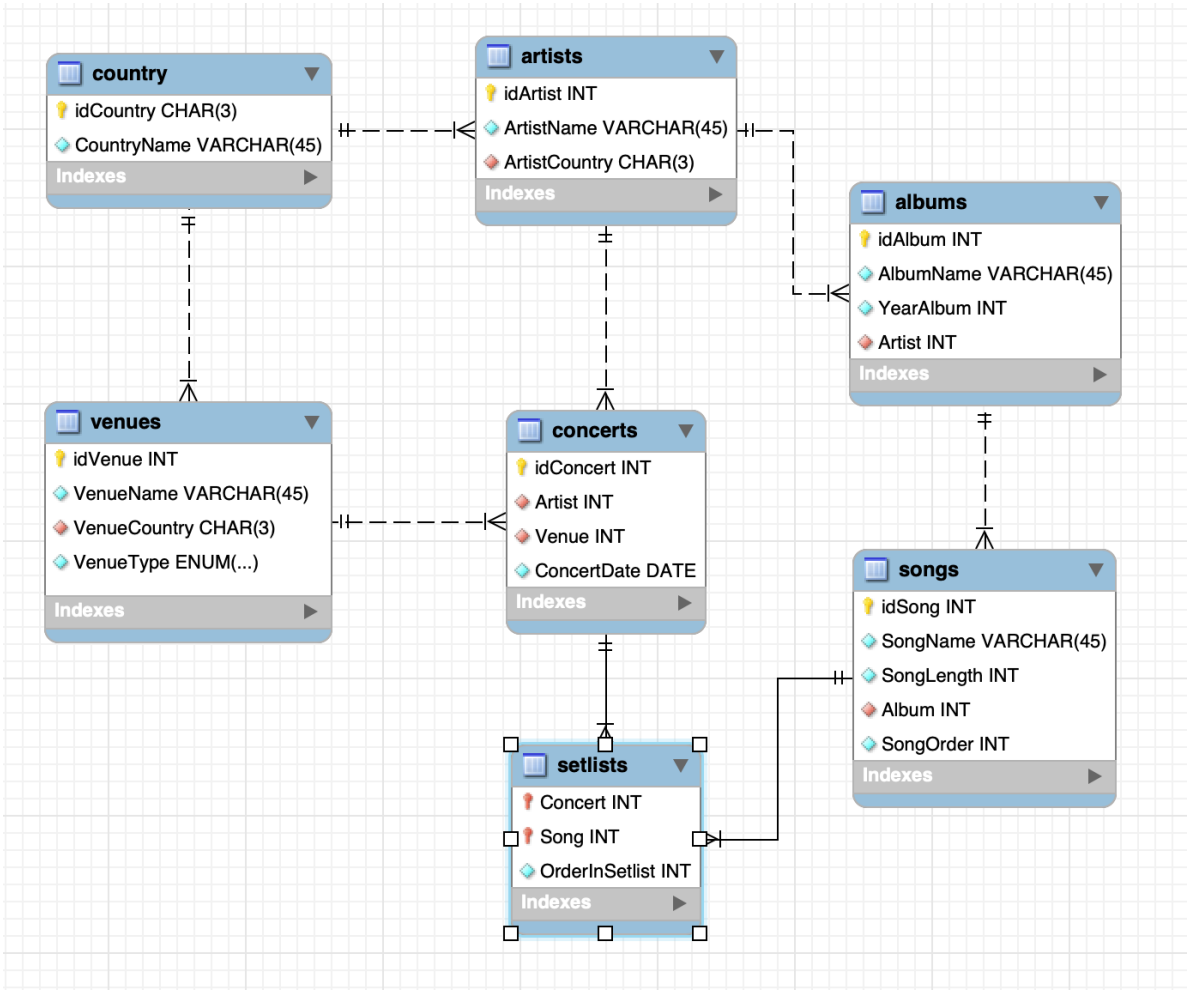

Figure 1. DB E/R schema

Tables and attributes names are self-explaining, so no further extra details about Figure 1's content is commented on. However, some considerations to be taken into account are:

- The possible values for the type of venue are: hall, pavilion, stadium and festival.
- The code of a country will be the first 3 letters of its name in capital letters, e.g., Spain - SPA, France - FRA, Uruguay - URU.
- If any questions remain, you are invited to discuss them with the laboratory practice teacher as soon as possible.

Functionality

The main functionalities of the system are:

- When starting the application, the system will connect to the DB whose schema is shown in Figure 1.
- After that, the management (inserting, updating, and deleting) of the information related to the tables ARTISTS, ALBUMS, SONGS, VENUES, and COUNTRIES, should be possible through the corresponding forms.
- To create a CONCERT and its SETLIST, the artist, venue and date of the concert will be chosen and then the songs that were part of the setlist for that concert will be selected.
- The application will allow you to modify the information of a concert by changing the artist, the venue and/or the date.
- The application will also allow you to modify the setlists by adding songs, removing them or changing the order in which they were played.
- The application will allow to visualize concerts and their setlists according to the following instructions:
    - Navigation by artist: it will show the list of concerts of an artist and display the setlists of the selected concerts. A filter can be applied so that only concerts in a selected country are displayed.
    - Navigation by venue: allows you to display the list of concerts in a specific venue and to view the setlists of the selected concerts.
- Finally, the application will also generate a series of reports, choosing the most appropriate format and information to be displayed, depending on the data contained in the DB:
    - Ordered list of songs performed the most times by an artist throughout all his/her concerts.
    - Ordered list of albums performed the most times by an artist.
    - List of artists who have ever performed a complete album in a concert.
    - List with the complete information of the songs that have been performed the most times.
    - Ordered list of artists who have performed the most concerts between 2 specific dates.
    - Ordered list of countries in which there have been more concerts between 2 specific dates.

Additional Considerations

- Any additional design/development decision taken by the development team MUST be agreed with the lab professor.
- When delivering the project, every table must contain at least 15 rows.
- The selection of the development language and the DBMS is free, although VB.NET and MySQL are recommended, as in the previous projects. Technical doubts will only be answered about these language and frameworks.
- The system must be developed in a 3-layer architecture (presentation, domain and persistence) following the principles of the Software Engineering. The use of design patterns (e.g., database broker) will be especially valued.
- Any modification performed on the proposed schema MUST be conveniently justified and agreed with the professor.
- The (UNEXTENDIBLE) defense date is the last session of the lab group of the members of the development team (check on Moodle). The delivery will be done some (2-3) days earlier so that the defense can be conveniently prepared.