University of Ottawa
School of Electrical Engineering and Computer Science
CSI2132-2014
Database Project Specification: Music Radio Station Database

This document contains the requirements as created in class. Note that I made *some modification* to the design, in order to limit the scope of this project. (For example, we do not consider advertisements in this database.)

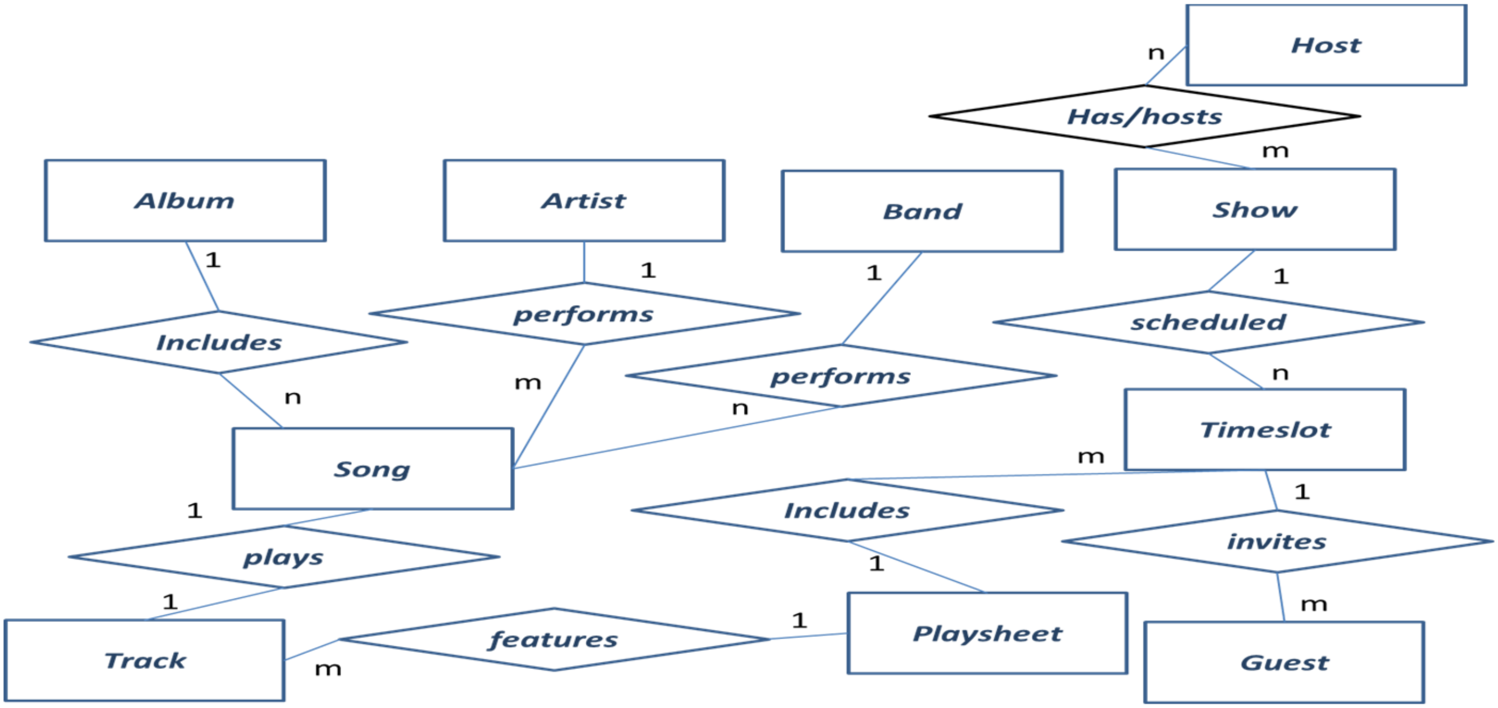
Instructions
1. Complete this project in a group of two (2) to three (3) students.
2. Demonstrate the project on **Monday March 31, 2014** in a 15 minute timeslot, as allocated by the TA.
3. Use PostgreSQL to complete this project, together with a language such as Java and JSP, or PhP, to create your Web-based front-end. You are also welcome to create a mobile front end, for use on say Android.

Deliverables:
Submit all your source code via BlackBoard Learn**, before Monday March 31, 2014 at 13h55**. (That is, before the demonstrations will start.) Note that all group members should submit the source code, not just one per group. All group members should attend the project demonstrations.

Your task:
Consider a Music Radio Station that broadcasts music shows. Each music show has one or more hosts and the show broadcasts some music according to a daily list of play sheets. Very often, a show also has a guest appearing on the show. The tracks that are played on a show have an artist or band (sometimes called group) associated with it. A track corresponds to a song that is from an album. For each track, it is also recorded whether the track is considered to have Canadian content (so-called Cancon content).

(You may want to take a look at http://ottawastart.com/radio.php to see the list of radio stations that broadcast in the Ottawa and Gatineau area. The region has three eclectic university/college stations, a Christian station, jazz stations, ethnic stations, oldies stations, hard rock stations, and so on.)

Here is a high level summary of the EER diagram, which only shows the entities and relationships.



Here is a list of some assumptions that were made.

1. A show may have one or more hosts, and a host may host many shows.
2. A show is scheduled in many time slots, but there is only one show at a specific time.
3. A guest is invited to appear in many timeslots, but there is maximum one guest for one show (timeslot). Some timeslots do not have any guests.
4. A timeslot is associated with only one play sheet, but a play sheet may be reused.
5. A play sheet features many tracks, but a track is associated with only one specific play sheet.
6. A track corresponds to exactly one song.
7. An album may contain many songs, but a song is associated with only one album.
8. An artist may perform many songs, but a song is associated with only one artist.
9. A band (or group) may perform many songs, but a song is associated with only one band.
10. The artist and band entities do not overlap.

Here is a partial definition of the relations you will need to create. You may want to add additional information/attributes to personalise the project. You are also welcome to make changes to this design, should you prefer to change the cardinalities and/or connectivities of the relationships as described above.

- Host(<u>employee-number</u>, first-name, last-name, stage-name, date-of-birth, rating, contract-start-date, salary, …)
  Contract start date is used to show when this host joined the radio station.
- Show(<u>Show-number</u>, Show-name, description, category, …)
  Category refers to the type of show, in terms of music being played (e.g. rock, gospel, jazz, and so on.)
- HostsShow(<u>Contract-number</u>, Employee-number, Show-number, host-start-year, host-start-month, …)
  The month and year attributes are used to indicate when a host joined the show.
- Timeslot(<u>Timeslot-number</u>, start-time, end-time, Show-number, Playsheet-number, …)
- PlaySheet(<u>Playsheet-number</u>, date, day-of-week, ….)
  None that we store the date as well as the day of the week (e.g. Friday).
- Guest(<u>Guest-number</u>, firstname, lastname, description, topic, rating, timeslot-number, ..)
  The description attribute is used to describe the guests' areas of specialization and the topic attribute indicates what will be discussed. The rating is an integer value to indicate how popular a guest was.
- Track(<u>Track-id</u>, title, start-time, end-time, type, song-id)
  Type refers to the style of music such as "new" or "hit" or "golden oldie".
- Song(<u>Song-id</u>, title, cancon, instrumental, album-id, artist-id)
  Instrumental is a yes/no field, while Cancon indicated whether this track is considered to be Canadian.
- Artist(<u>Artist-id</u>, first-name, last-name, stage-name, nationality, date-of-birth, …)
- Band(<u>Artist-id</u>, name, start-year, nationality, …)
  Note that a specific song is either performed by an individual artist or a band.
- Album(<u>Album-id</u>, type, label, album-recording-date, …)

Requirements and Mark Allocation

You are required to complete the following tasks: (**Total 100 marks**)

1. **(10 marks)** Transform the description into a relational model and create all the tables in PostgreSQL. Add all other relevant attributes and remember to enforce entity and referential integrity.

2. **(10 marks)** Populate the tables with your own data, using the Music Radio Station of your own choice. It follows that your data, and the attribute values you choose, should be sufficient in order to implement and test the queries specified below. Your database should include at least 40 different songs. There should be 10 play sheets that contain around 8 songs each. The number of artists and/or bands should be at least 10. The number of shows should be at least 5, with an average of 2 hosts per show. Every week, there should be at least 3 timeslots per show. About 50% of the shows should have invited guests.

3. **(10 marks)** Provide the user with the ability to add data to, and delete data from, the following tables in your database: HostsShow, Song and Guest.

4. **(40 marks)** Create a number of SQL queries to explore this data. The following is a suggested list of "typical" queries that should be implemented.

   Shows and Play Sheets

   a. Display all the information about a user-specified show. That is, the user should select the name of the show from a list, and the information as contained in the show table is then displayed on the screen.
   b. Display the play sheet of a specific show, based on the date of the show. That is, the user should select the name of the show from a list, together with a date, and all the information of the play sheet (including the tracks) should then be displayed on the screen. The play sheet should be ordered using the sequence in which the tracks that were played.
   c. For each user-specified category of show, list the host names together with the last date that they hosted the show. The user should be able to select the category (e.g. jazz or hip hop) from a list.
   d. Given a user-specified show, find the title of the song played the most frequency on this show. List this information together with the name of the artist or band, and the name of the album. The user should be able to select the show (e.g. morning rush hour) from a list.
   e. For each category of show (e.g. hip hop or classical), list the host names together with the average rating of the hosts for this category of show.

   Songs, Artists, Bands and Albums

   f. Find the total number of songs of each recording label, as contained in our music inventory. That is, the data should be grouped by the recording label of the albums the songs appeared on.
   g. Display the details of the songs that have not been played in January 2014. That is, you should display the title of the song and the date it was recorded, together with the details of the artist or band.
   h. Find the titles and recording dates of the albums that include songs that have been played more often than any song performed by Band X. Order your results by the dates of the albums. (Here, Band X refers to any band of your choice.)
   i. List the names of the Canadian artists who recorded the Canadian content songs that have been played the most often. Display this information together with the names of hosts that played these songs.
   j. Provide a query to determine whether instrumental songs are more popular than other songs.

   Guests and Hosts

   k. Find the names of the guests that have the highest overall rating. Display this information together with the names of the hosts that invited these guests and the dates the guests appeared on shows.
   l. Find all the details of the guests that appeared the most frequently on a specific show. For example, display the details of the guest that appears most often on the morning talk show. Display this information ordered by the topics that they prefer to discuss.
   m. Find the names and salaries of all hosts with salaries that are higher than that of a host with a stage-name called John. (Note that there may be more than one host with this stage-name).
   n. Find the names of the most diverse hosts, i.e. hosts that have been involved in broadcasting the highest number of different shows. Display this information together with the show names and the time slots. For example, John Cane may have worked on the morning chat show (07h00-09h00), the midnight blues show (0h00-02h00), the lazy Saturday show (12h00-15h00) and the Friday happy hour show (16h00-18h00).

Please note that, in the above, the queries may also return only one name. (That is, there may be only one person satisfying these queries.)

5. **(30 marks)** Create a web-based front-end, for the user to directly query the database. You may also choose to create a mobile application.

6. Additional effort, such as creating a superb front-end or including a multimedia component, may earn you up to **20 bonus** marks.