

CSCE 156 – Computer Science II

Lab 08 - SQL II - Worksheet

Dr. Chris Bourke

Names _____

For each question, write an SQL query to get the specified result. You are highly encouraged to use a GUI SQL tool such as MySQL Workbench and keep track of your queries in an SQL script so that lab instructors can verify your work. If you do, write your queries in the script file provided rather than hand-writing your queries here.

1 Inserting & Manipulating Data

1. Choose your favorite album and insert it into the database by doing the following.
 - (a) Write a query to insert the band of the album
 - (b) Write a query to insert the album
 - (c) Write two queries to insert the first two songs of the album
 - (d) Write two queries to associate the two songs with the inserted album
2. Update the musician record for “P. Best”, his first name should be “Pete”.
3. Pete Best was the Beatle’s original drummer, but was fired in 1962. Write a query that removes Pete Best from the Beatles.
4. Attempt to delete the song “Big in Japan” (by Tom Waits on the album *Mule Variations*). Write a series of queries that will allow you to delete the album *Mule Variations*.

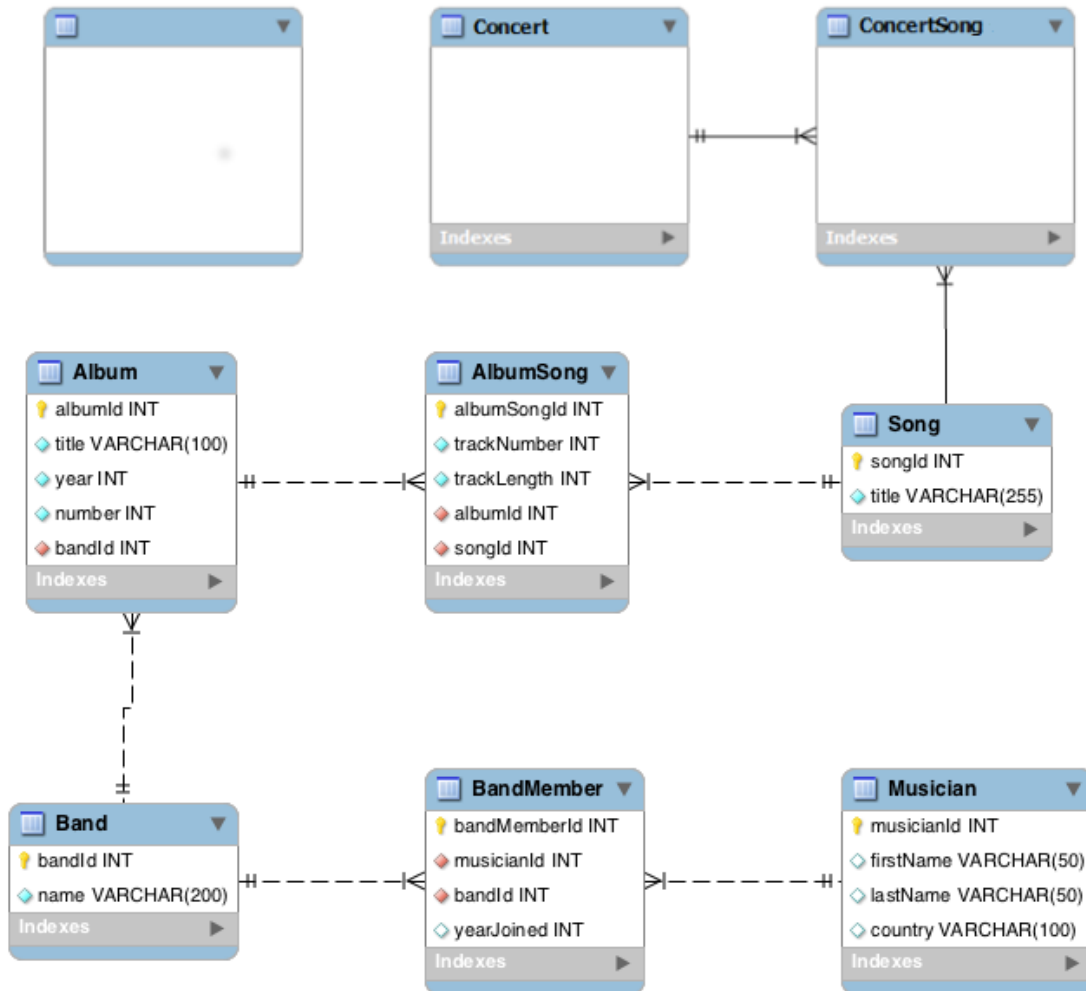


Figure 1: An incomplete ER diagram for the new database being designed.

2 Designing Tables

In this section you will design the ER by completing the diagram in Figure 1 as instructed in the following steps. All 6 information items in List 1 of the handout should be supported by the resulting design.

1. Use the tables from the original albums database as an example to list the fields and field types associated with each of the **Concert** and **ConcertSong** entities in Figure 1. If your design needs an additional entity then use the blank item.

Note: Pay attention to how the primary/foreign keys must be designed to support the relations between **Concert**, **ConcertSong** use **BandMember** and **AlbumSong** as examples to complete **ConcertSong**.

2. How does **Band** from the original albums database relate to the **Concert** and/or **ConcertSong** entities? Indicate the relation(s) by drawing the appropriate line(s) onto Figure 1 and make sure any primary/foreign keys are updated to reflect this/these relation(s).
3. Every concert takes place at a concert hall; it is possible that the name of a concert hall or its seating capacity changes. In such an event your current design should not require that any previously stored concerts be updated to reflect the modifications to the concert halls; if so add an entity to your design to solve the problem by filling the blank item in Figure 1.
4. Specify the relations between the entity you described in the blank item and the other entities in Figure 1. Make sure the primary/foreign keys of the entities match their relations.

3 Create the New Tables

Write and run a new SQL script (or simply just modify `albums.sql`) to generate the new tables you designed. Alternatively, you may write a new script that modifies the existing database). Use Figure 1 as a blueprint for your script making sure the following items are satisfied:

- Naming – Use a uniform naming conventions for the tables and their fields
- Field Types – Make sure to use appropriate types and a uniform typing conventions for each field
- Primary keys – Make sure to specify which fields are primary keys
- Foreign keys – Enforce appropriate foreign key restrictions to reflect each relation.
- Note: Some of the new relations may require you to use a `alter table` statement since you may need to modify the table from the original albums database design to enforce the constraints.
- Null/Default values – Some entity fields are so essential that if they are not provided a value the entity itself is unable to server its purpose (i.e. a primary key should never be nullable). Make sure all fields are able to store valid values.

Write SQL statements to do the following:

1. Write a statement to create the table **ConcertSong**
2. Write a statement to create the table **Concert**

3. Write statement(s) to create any other tables/entities that your design requires
4. Write statement(s) to alter the original tables that you made if needed.

4 Using The New Tables

You will now make sure that your design makes sense by writing several queries to insert and query data out of it.

1. Write queries to insert at least two **Concert** records.
2. Write queries to associate at least 3 songs with each of the two concerts
3. Write a select-join query to retrieve these new results and produce a playlist for each concert
4. Modify the query to include the name of the band playing the concert. If such a query is not possible, explain why and sketch an alternative design in which it would be possible.