

CSCE 156 – Lab09: SQL II

Worksheet- Summer 2018

Name _____

Select One: Driver/Navigator

Design Entity Relations (ER)

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.

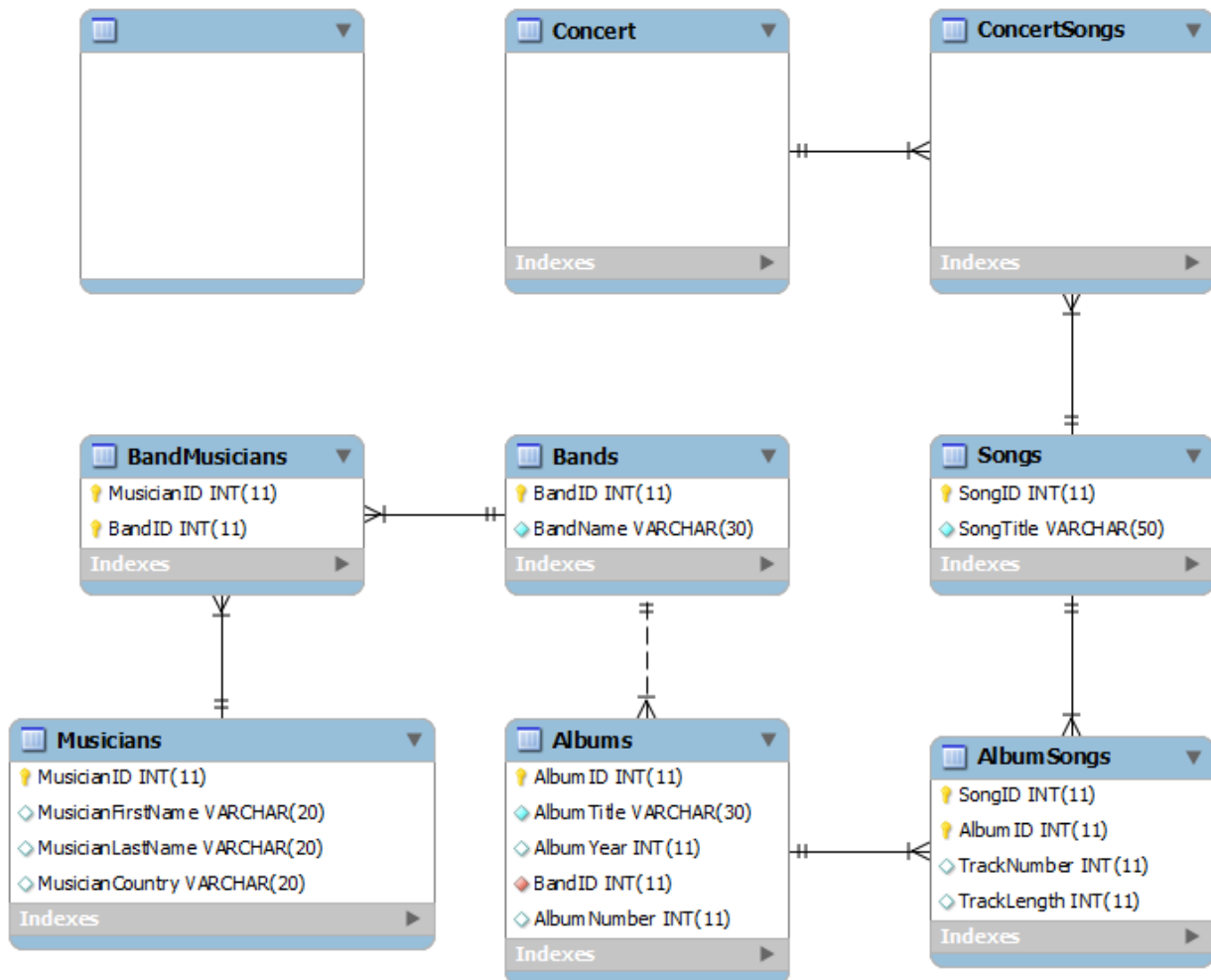


Figure 1 An incomplete ER diagram for the new database being designed.

1. Use the tables from the original *Albums* database as an example to list the fields and field types associated with each of the *Concerts* and *ConcertSongs* 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*, *Concert Songs*, and *Songs*; use *BandMusicians* and *AlbumSongs* as examples to complete *ConcertSongs*.

2. How does *Bands* from the original Albums database relate to the *Concert* and/or *ConcertSongs* 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.

Show your design to a lab instructor and get it signed off by a lab instructor before proceeding.

Date_____

Signature_____

Create the New Database

Write and run a new SQL script (or simply just modify `albums.sql`) to generate the new tables you designed in Activity 1 using the *CREATE TABLE* statement (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 the *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 and demonstrate your working script to a lab instructor to have them sign off on this lab.

1. Write a statement to create the table *ConcertSongs*
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.

Date_____

Signature_____

Use the new Database

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.

Date_____

Signature_____

Advanced Activity (Optional)

Consider the venues listed in Table 1.

Table 1 The concert halls		
	Name	Capacity
1.	The Mega Dome	12,000
2.	The Gorge	32,000
3.	Hotel Concert Hall	5,000
4.	Cruise Concert Hall	2,000

Say that one concert was held at each of the concert halls according to the following rules listed below. Write an SQL script to insert data into the newly designed tables using the rules bellow.

1. In descending order of concert hall capacity, bands are assigned a concert hall in descending order of the number of their album songs and ascending order of their band name. That means the band with the most number of album songs and the smallest lexicography band name gets the highest capacity concert hall and the band with the second highest number of album songs get the second highest capacity concert hall.
2. The songs played at each concert are those which have at least one album song of 5 minutes or longer.
3. Each concert is sold out.