INFO20003 Week 2 Lab

Modelling with MySQL Workbench

Objectives:

In this laboratory session you will familiarise yourself with the MySQL Workbench database modelling tool. This is the tool you will use to complete modelling tasks in labs and to complete Assignment 1.

In this lab you will:

- Launch the MySQL Workbench modelling tool
- Create a simple ER model using MySQL Workbench
- Define relationships between tables
- Create meaningful relationship labels

Creating the model

This section of the lab introduces you to the basic process of creating data models (entity-relationship models, or ER models) in MySQL Workbench.

◆ **Task 1** Launch MySQL Workbench.

When you launch MySQL Workbench you will be in the default "MySQL Connections" view. This is circled in red below.
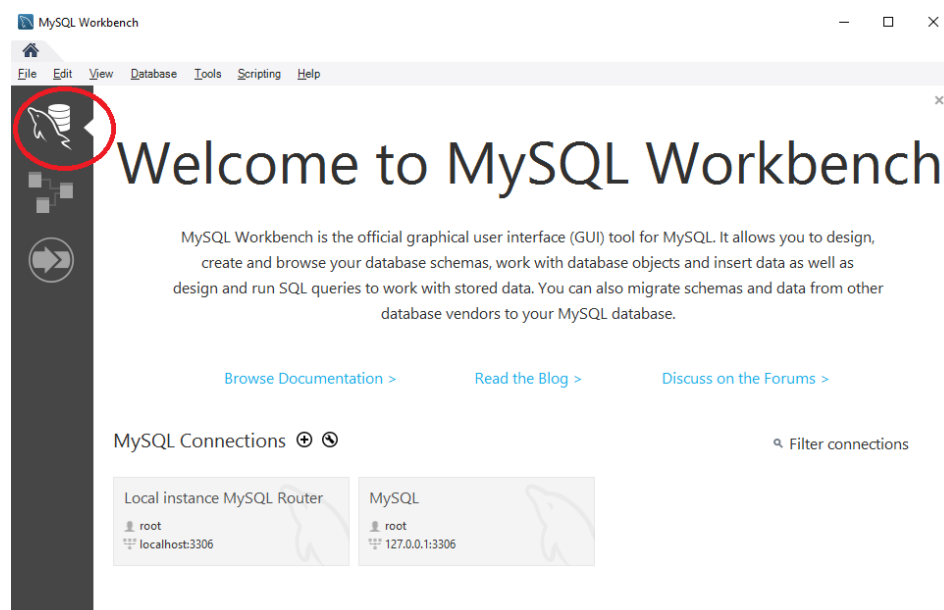

Figure 1: The default MySQL Workbench window

You will need to select the "Models" view:

Figure 2: The Models icon
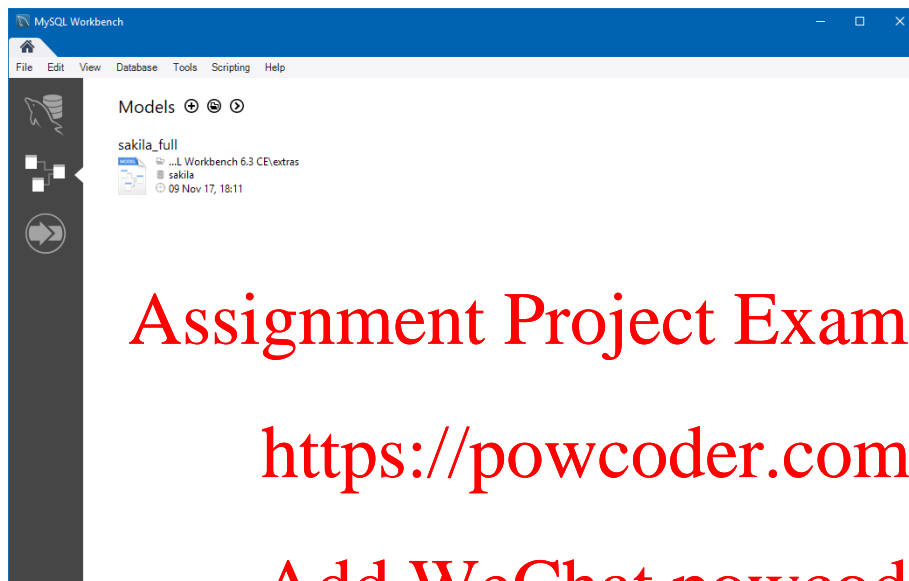
Your window will have changed to a different view:

Figure 4: The full screen of the Models view

◆ **Task 2** Click the Add model (+) symbol next to the word "Models" at the top left of the MySQL Workbench window.
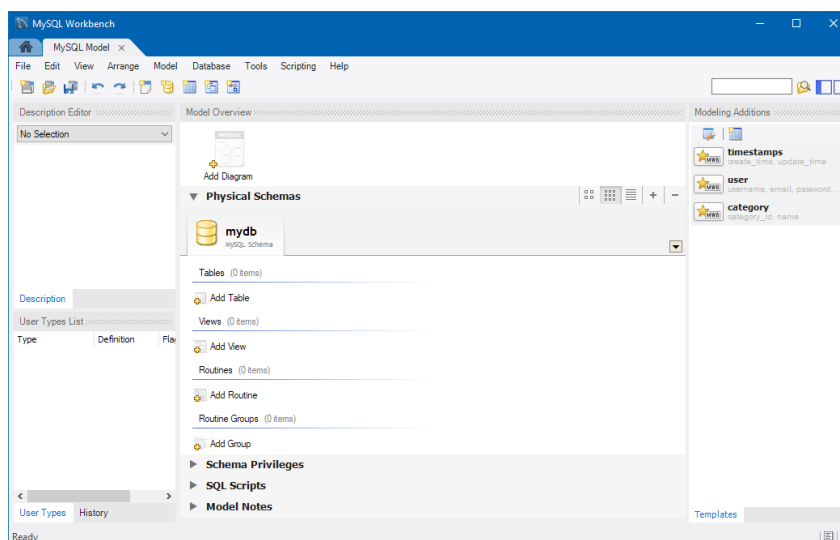
This will launch a new modelling window:

Figure 5: The new modelling window

◆ **Task 3** Click the "Add Diagram" icon to add a new diagram:
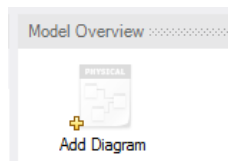
Figure 6: The add diagram icon

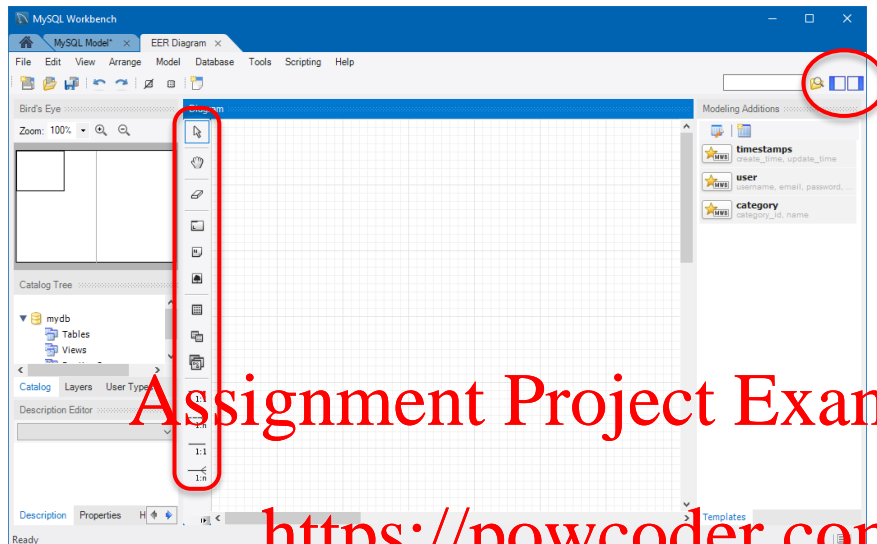This will bring up a second tab containing an empty diagram window:

Figure 7: The diagram with modelling tools on the left-hand side of the canvas. Note the buttons to hide the side panels in the top-right corner of the canvas.

◆ **Task 4** Make more space for your diagram. Hide the left- and right-hand panels of your modelling canvas using the buttons in the top-right corner:
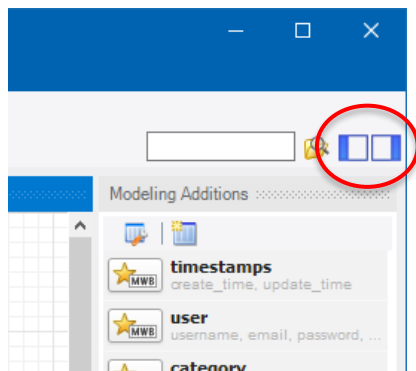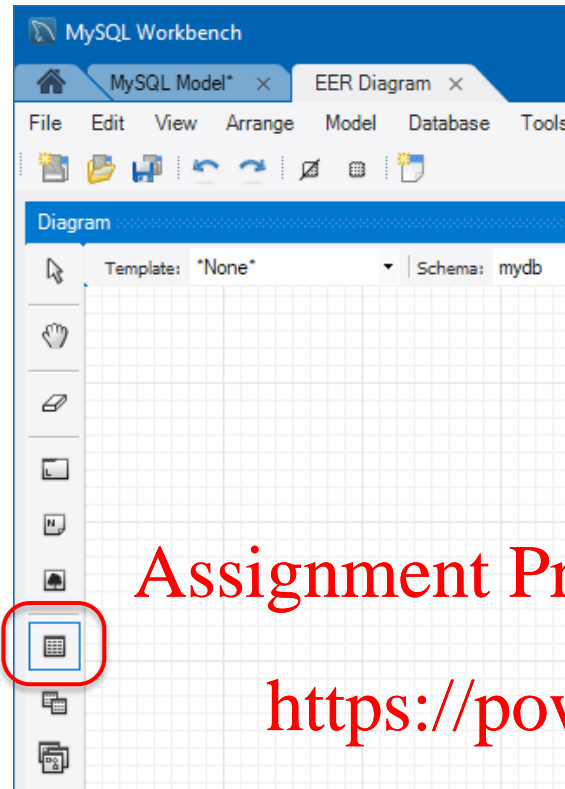
Figure 8: Buttons to minimize the left and right panels of the modelling canvas

Adding tables to your model

◆ **Task 5** Add a new table to your model by clicking on the table icon, then clicking anywhere in your diagram.

Figure 9: The add table tool
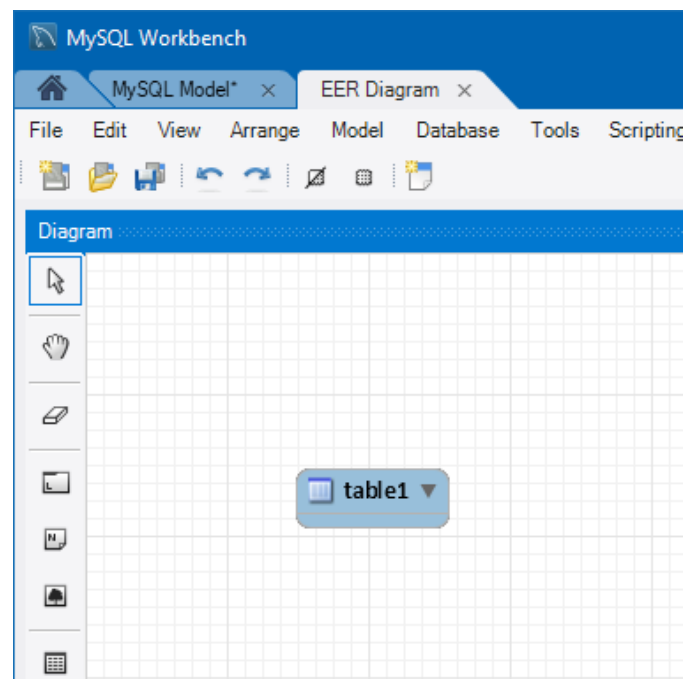
A new table will appear:

Figure 10: The table has now been placed on the diagram canvas

◆ **Task 6** Double click the table.

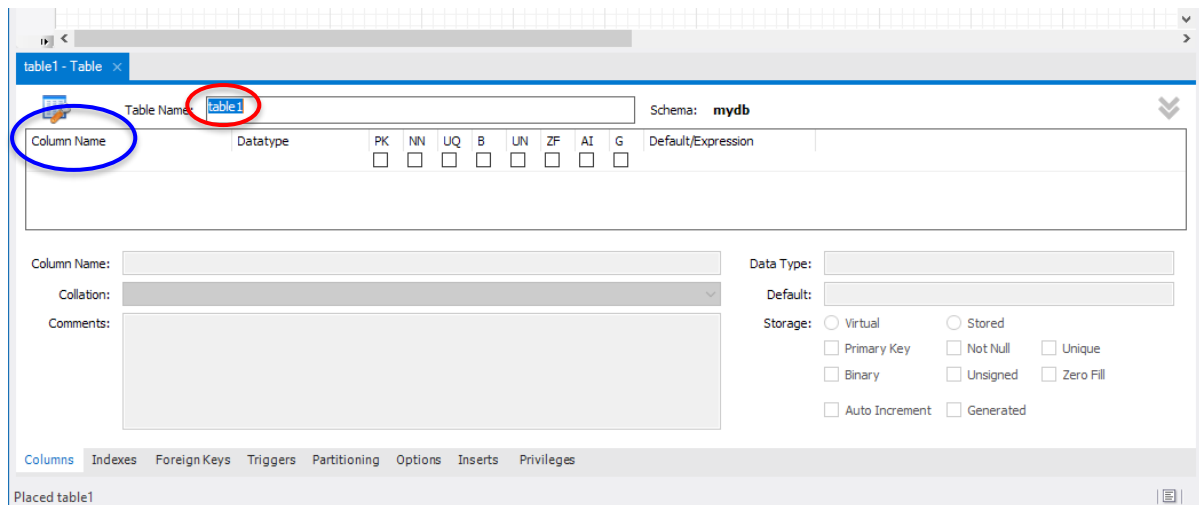The table editor appears at the bottom of the window. The table name is highlighted.

Figure 11: The table editor

◆ **Task 7** Change the name of the table from 'table1' to 'Song'.

Assignment Project Exam Help

https://powcoder.com

◆ **Task 8** Now double-click the area under 'Column Name' and enter the following information for each column:

Add WeChat powcoder

Song Table:

Column Name	Data Type	Check Box (select)
SongID	INT	PK, NN *
SongTitle	VARCHAR(45)	NN
Artist	VARCHAR(45)	NN
AlbumID	INT	
LastPlayed	DATETIME	
Genre	VARCHAR(20)	

Table 1: The Music table column names and data types

* Note: PK stands for "primary key", and NN is short for "NOT NULL" (see below).

To change the column's data type, double-click in the data type column and select from the drop-down list.

For VARCHAR you need to type a number in the brackets. This number is the maximum length of the text that can be stored in this column. For DATETIME you **do not** need to type a number in the brackets. If brackets are shown, simply delete them.

The finished table should look like this in the table editor:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
SongID	INT	☑	☑	☐	☐	☐	☐	☐	☐	
SongTitle	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
Artist	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
AlbumID	INT	☐	☐	☐	☐	☐	☐	☐	☐	
LastPlayed	DATETIME	☐	☐	☐	☐	☐	☐	☐	☐	
Genre	VARCHAR(20)	☐	☐	☐	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

Figure 12: The Song table in the table editor

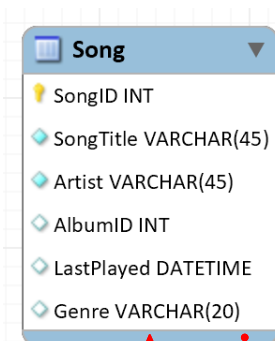And the Song table in the diagram should now look like this:

Figure 13: The Song table as drawn in the diagram

The yellow 'key' indicates that the SongID column is a Primary Key. The solid blue diamond indicates that this column must be populated for every row in the table and cannot be empty (or "null"). The blue outlined diamonds indicate the column can contain null or empty values for a row.

◆ **Task 9** Repeat Tasks 5 to 8 to add the Album and RecordCompany tables to the diagram:

Album Table

Column Name	Data Type	Check Box (select)
AlbumID	INT	PK, NN
AlbumTitle	VARCHAR(45)	NN
Rating	INT	
RecordCompanyID	INT	NN

Table 2: The Album table column names and data types

RecordCompany Table

Column Name	Data Type	Check Box (select)
RecordCompanyID	INT	PK, NN
RecordCompanyName	VARCHAR(45)	NN
Country	VARCHAR(45)	NN

Table 3: The RecordCompany table column names and data types

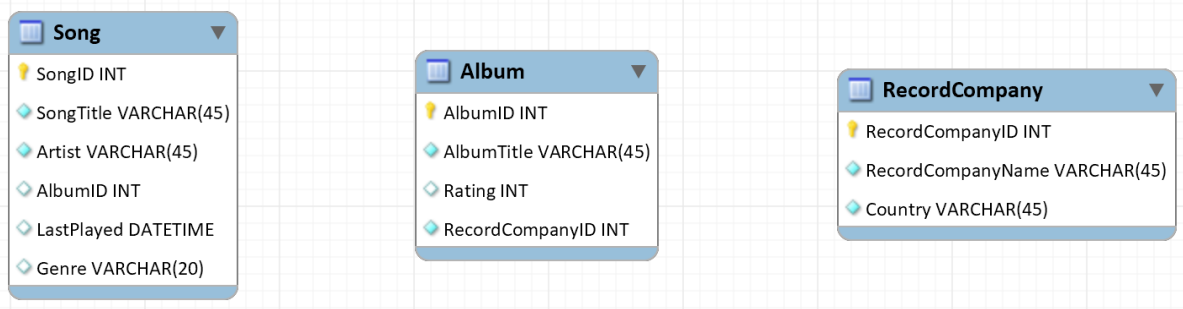After completing Task 9, your diagram should look similar to below:

Figure 14: The three tables (Song, Album and RecordCompany) in your diagram, without relationships

◆ **Task 10** Save your changes (**File** > **Save**).

Relating tables

Now that you have created the tables, it is time to link them with relationships.

When linking tables to each other, make sure you use the "eyedropper" tool that allows you to use existing columns.

Assignment Project Exam Help
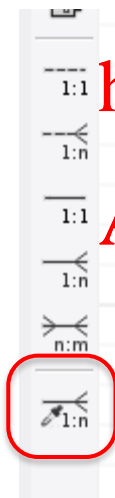
https://powcoder.com

Add WeChat powcoder

Figure 15: The "Place a Relationship Using Existing Columns" or "eyedropper" tool. This tool allows you to define relationships between two entities using an existing column in each entity.
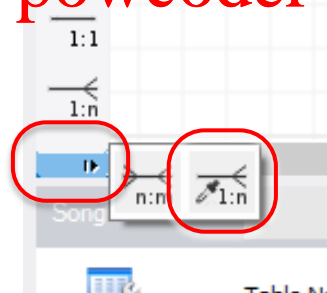
Figure 16: On Windows, this tool might be located in the "overflow" popout at the bottom of the toolbar.

Mac users note: If you cannot find this tool, you might need to close the Table editor section of the window.

In the Music table, the AlbumID column is the attribute that will identify which album each song is found on. In the following tasks, you will learn how to use the "eyedropper" tool to link the two tables together.

HINT: When drawing relationships between two tables, there must be a column of the same data type representing the same information in each entity.

◆ **Task 11** Select the "eyedropper" tool and select the AlbumID column in the Song table.
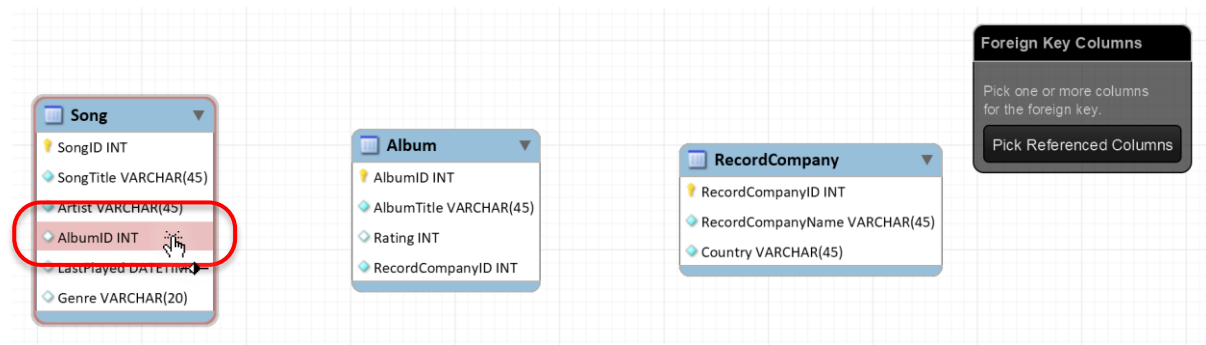

Figure 17: Select the AlbumID column of the Song table.

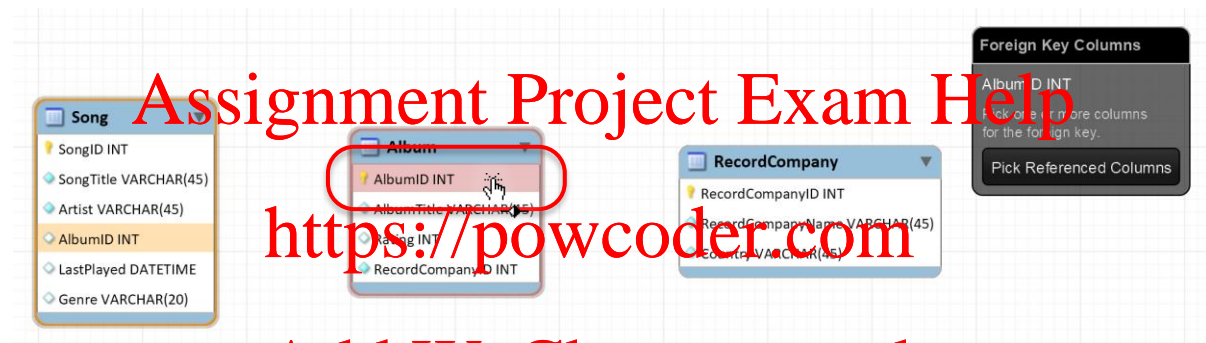◆ **Task 12** Then select the AlbumID column in the Album table.

Figure 18: Select the primary key of the Album table.

Note: The two attributes used to create a relationship between the two tables must have exactly the same data type. However, they may have different names.
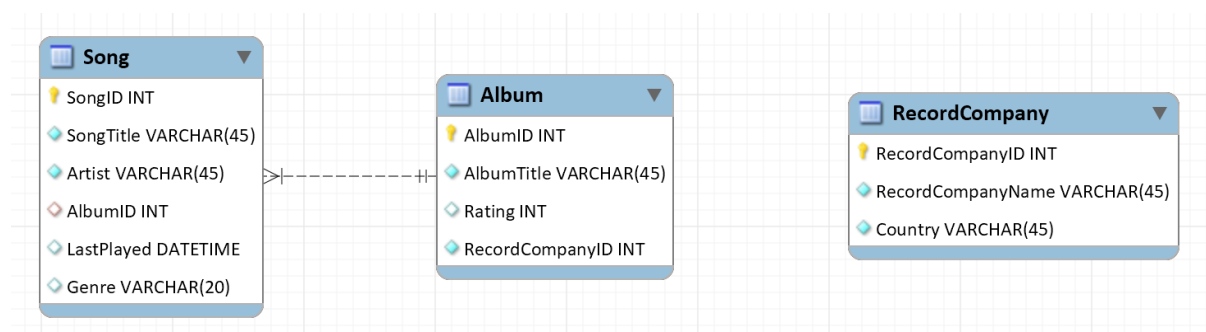

Figure 19: A relationship is now defined between the AlbumID in the Song table and the AlbumID in the Album table.

Note that the AlbumID diamond in the Song table has changed colour from blue to red. This indicates that this column is a "foreign key", used in linking the two tables.

◆ **Task 13** Now repeat this task by creating a relationship from the RecordCompanyID column in the Album table to the RecordCompanyID column in the RecordCompany table.
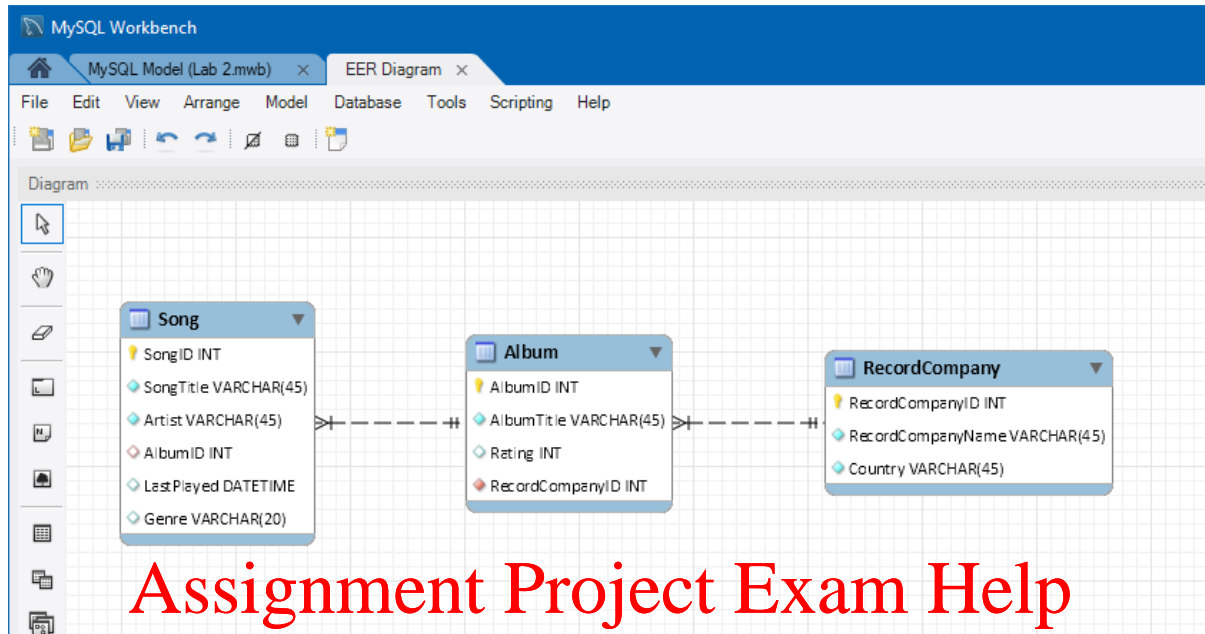
Your end result should look like this:

Figure 20: The finished relationships between the Song, Album and RecordCompany tables

◆ **Task 14** Save your changes.

Adding relationship labels

We are now going to label the relationship between each pair of tables. Relationship labels substantially improve the readability of the models.

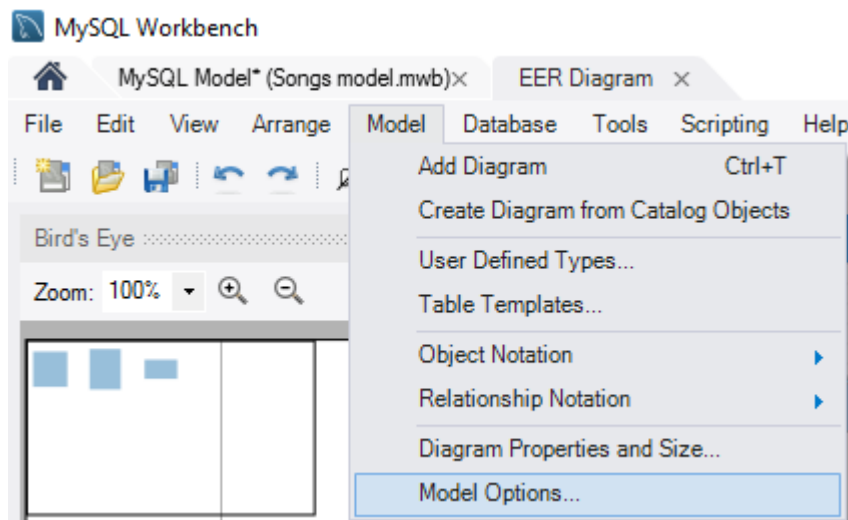◆ **Task 15** From the menu, select **Model** > **Model Options**.

Figure 21: The Model Options menu

◆ **Task 16** In the Model Options window, select "Diagram" and uncheck the "Use defaults from global settings" checkbox at the bottom of the window.
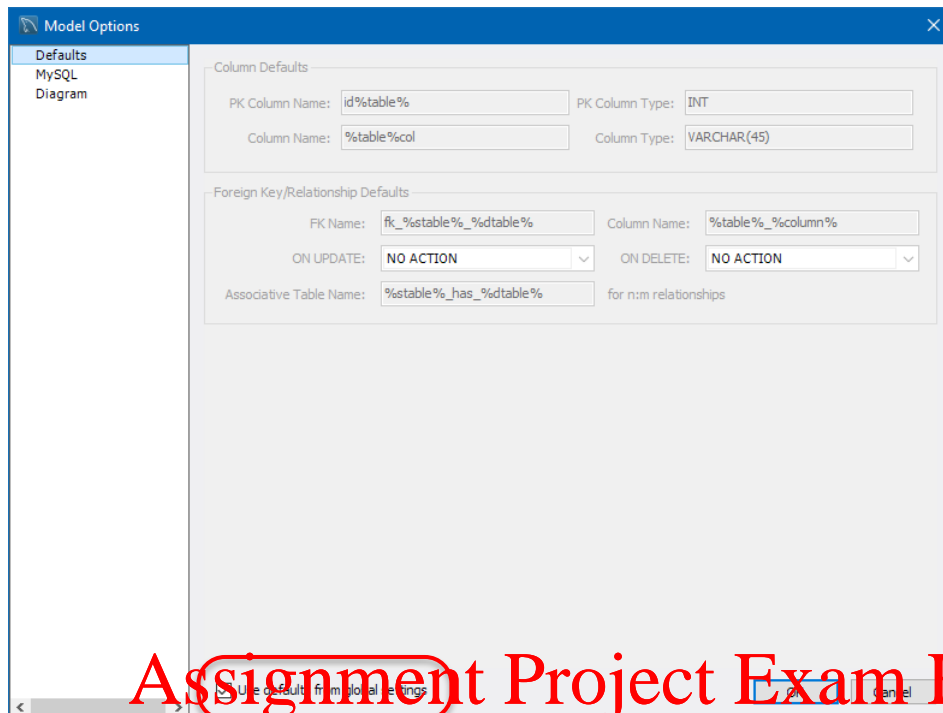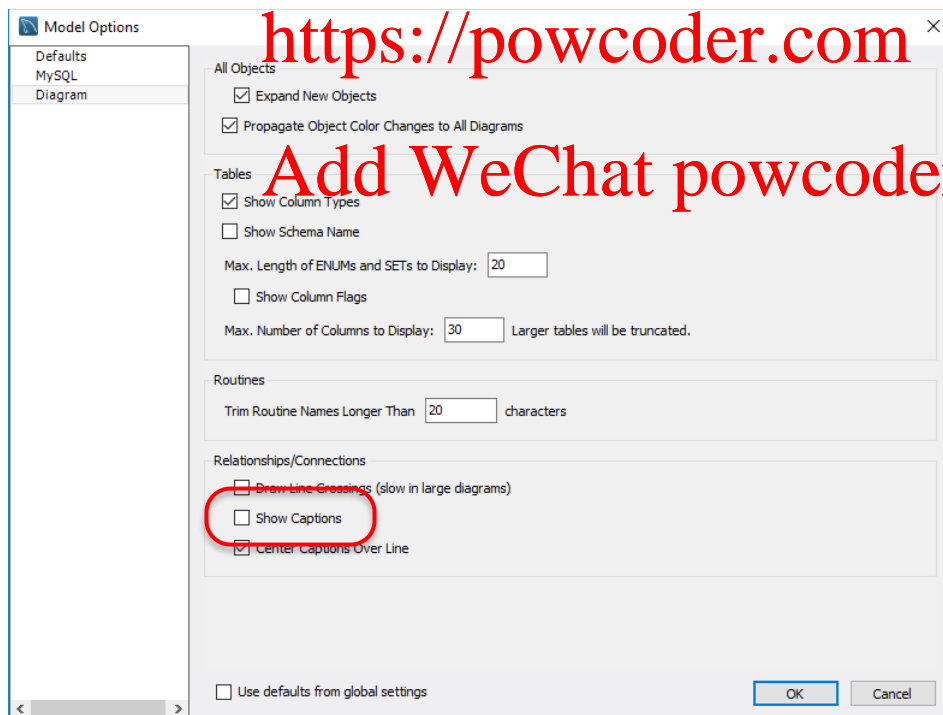

Figure 22: The Diagram menu in Model Options

Figure 23: After unchecking the "Use defaults from global settings" checkbox, all the diagram options are now available to be altered.

◆ **Task 17** Check the "Show Captions" checkbox in the "Relationship Connections" section of the Diagram Model preferences. Then click OK.

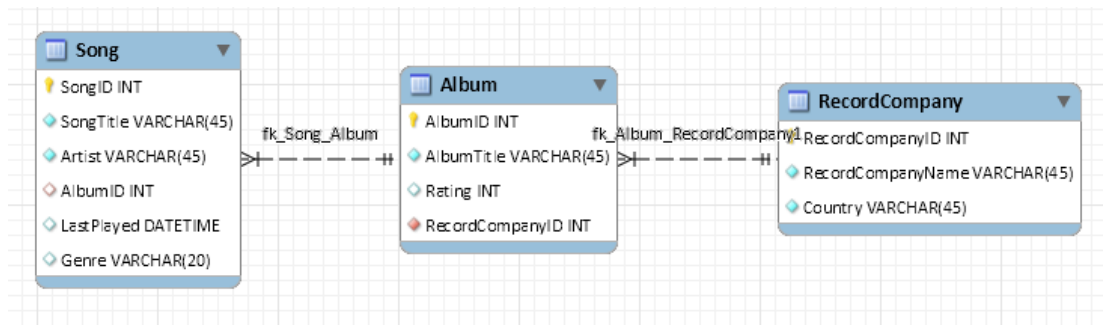The relationship connections are now labelled with default labels.

Figure 24: The relationships are labelled fk_Song_Album and fk_Album_RecordCompany1.

◆ **Task 18** Double-click on the fk_Song_Album relationship label.
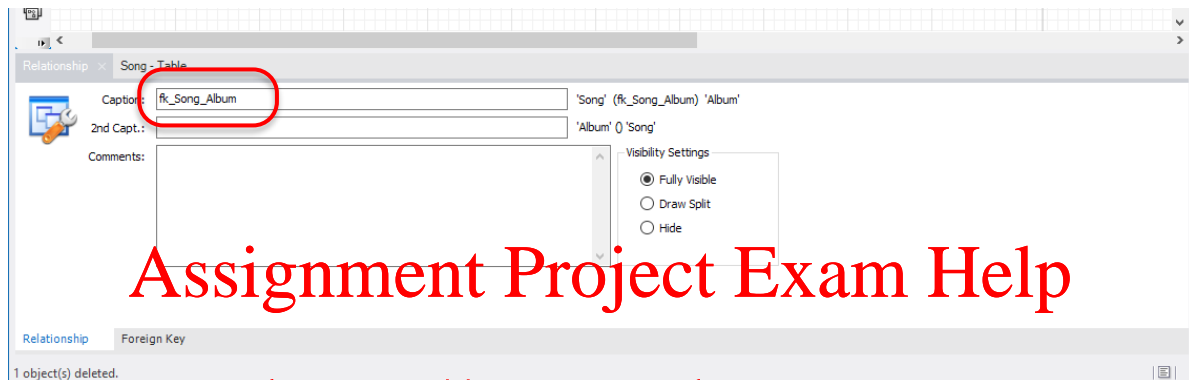
The Relationship editor appears.

Figure 25: The Relationship editor

◆ **Task 19** Change the "Caption" (circled in red in Figure 25) from 'fk_Song_Album' to 'recorded on' and close the Relationship editor.

The relationship between the Song and Album tables has been renamed.

◆ **Task 20** Repeat the process for the other relationship, changing the name to 'released by'.
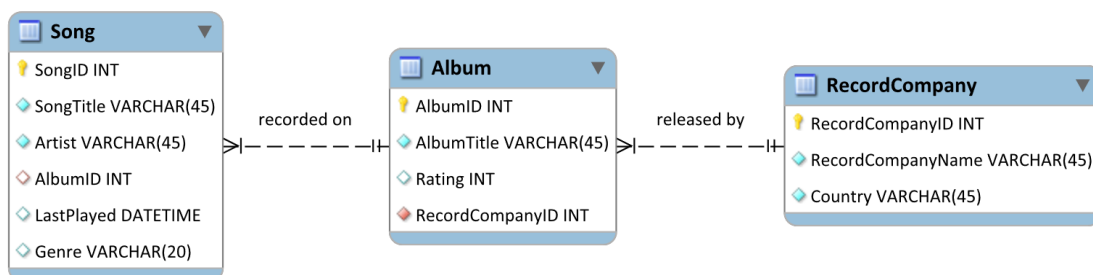
Your final diagram should look like this:

Figure 26: The final Music model

You have just modelled a simple music-related scenario with three entities (tables).

The model is not quite complete – it lacks participation constraints (an indication of whether the relationship is mandatory or optional). This will be covered in next week's lab.

◆ **Task 21** Save your changes to your diagram. <u>You will need this model again next week</u>, so save it in a safe place!

If you are working on a lab PC, save your file to a location where you can retrieve it later, such as your H: drive. Files saved to the lab PC's C: drive are deleted on machine reboot.

Congratulations, you have created your first ER model!

End of Lab 2

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder