COMP 3005 Winter 2022 Assignment 02

ER Modeling

Due: Wed. Feb. 9 by 10:00pm submitted to brightspace

Revisions:

Revisions will be noted here

Marking: This assignment is based on itemized requirements totaling 54 marks.

Marks are awarded, or deducted, based on requirements as follows:

Req Type	Assignment Grading
R0.x	**Critical Submission and Intent Requirements.** Assignment (or problem in some cases) gets 0 if **any** critical submission requirement (shown in red) is not met.
R0.x	**Good Practice Requirements.** You lose 2 marks for any good practice requirement (shown in amber) not met.
Rx.x	**Design Requirements.** You earn 2 marks for each design requirement (green) satisfied, well implemented, and demonstrated as requested; 1 mark if it's partly met, met but not well implemented, or met but not demonstrated; and 0 if it's not attempted or met.

Submission Summary:

Submission Requirement R0.0 Submit your answers to this assignment as a single organized .pdf document. Label the problems clearly in your document. Individual loose files will not be graded. 0 marks for the assignment if this requirement is not met.

Question	Devliverable to Submit
Problem 1,2,3	NOTE hand in a **single** PDF document for your whole assignment. ER Model and schema diagram consistent with numbered requirements. (NO HAND DRAWN ER DIAGRAMS OR SCHEMAS)
Problem 4	ER Model and schema diagram consistent with numbered requirements and also a revised scenario description. (NO HAND DRAWN ER DIAGRAMS OR SCHEMAS)

Assignment 2 Marking Guide

There are many possible answers for each question. You will have to use your judgement. Use the specific numbered requirements to judge their answers.

Some solutions will be efficient in that they would not require many nulls in tables but probably have expensive tables. Others might build tables that would have a lot of null values in them. In this assignment we are not judging efficiency yet, we just care that all the necessary information is there and that the data and relationships are clear. (I may provide sample ER diagrams and table schema for some questions but their answers might be different and work just as well.)

Marks Breakdown

1A E-R diagram 6 marks (requirements ER1.1-ER1.6)
1B Schema 5 marks (requirements SR2.1-SR2.5)

2A E-R diagram 6 marks (requirements ER1.1-ER1.6)
2B Schema 5 marks (requirements SR2.1-SR2.5)
2C Design 5 marks (requirements R2.1-R2.5)

3A E-R diagram 6 marks (requirements ER1.1-ER1.6)
3B Schema 5 marks (requirements SR2.1-SR2.5)

4A E-R diagram 6 marks (requirements ER1.1-ER1.6)
4B Schema 5 marks (requirements SR2.1-SR2.5)
4C Scenerio Description marks (requirement DR4.1)
(For the Scenerio, rate their description from 1-5 as to how well they described their application area and to what extent it allows you to understand their E-R model.)

Total = 54 Marks.

I have numbered important requirements below so you can refer to a requirement number in the marking comments for the students (e.g. "ER1.3 was not met").

Creating E-R diagrams and Relational Schemas.

In this assignment you will produce E-R diagram data models and initial relational table schema for the situations described in the problems below. Problem 4 is based on your proposed database from assignment 1.

For each question you must provide the following. (Question 4 will also require a revised scenario description.)

1) Write down any important assumptions you make about the data (e.g. what is unique, what pertains to an individual or group, etc.) that will help the marker understand your diagram.

2) An E-R diagram for the data model satisfying the following requirements.

3) Provide a Schema for your tables.

Mark the E-Rdiagrams out of 5 [1 mark per requirement ER1.1 - ER1.5]. 0 Marks for Hand Drawn ER diagrams or schemas.

ER1.1 All strong entities (boxes) should show attributes and identify the key by underlining the primary key attributes. (You can use "bubbles" for attributes or list them in the entity box.)

ER1.2 Weak Entities, if any, should be identified as such with a double bordered box, show the descriminator attributes and also show the with a double bordered diamond the relationship that provides the borrowed portion of the key. (If their design does not require weak entities then consider this requirement met.)

ER1.3 Relationships should be shown with a diamond shape, have a name and show relationship cardinalities (1:1, 1:N, N:N). If inheritance is used then it should appear on the E-R model.

ER1.4 Relationships must correctly show compulsory vs. optional participation of entities in relationships.

ER1.5 There should be no attributes on the ER model the refer to other entities. For example, a playlist entitiy that has a user attribute that is meant to refer to instances of a user entity would be wrong (see the video discussion on this.)

ER1.6 There should be no unnecessary entities or relationships. That is, the design should not be more complicated than it needs to be. It is common when learning E-R modeling that people come up with models that have redundancies or are way more complicated than necessary.

Mark the Table Schema out of 5 [1 mark per requirement SR2.1 - SR2.5]

SR2.1 Their schema must correspond to their E-R diagram. Strong and weak entities each map to their own table (except if a table completely contains another which can happen if you have a chain of weak entities).

SR2.2 All 1:1 and 1:N relationships are implemented by adding attributes to existing entity tables. All N:N relationships are implemented as their own tables.

SR2.3 Inheritance relationships are implemented by producing individual tables for the sub-class (or chilld) entities. (If necessary you may produce a table for the superclass entities as well.)

SR2.4 Each table in your schema should have a name and clearly show all the attributes and have the attributes that make up the primary key underlined.

SR2.5 Each table should indicate any foreign key relationships with other tables. You can do this by drawing arrows on the schema if you want or using text.

Problem 1

Consider a database that keeps track of scenes filmed for different movies. The database should keep track of all the information mentioned in this scenario.

A movie uses a screenplay (or story) that is organized into scenes. The movie will also have the same scenes because the movie follows the screenplay.

Not all screenplays in the database become movies but we want the database to keep track of all the screenplays regardless. The database should keep track of the screenplay's title and author.

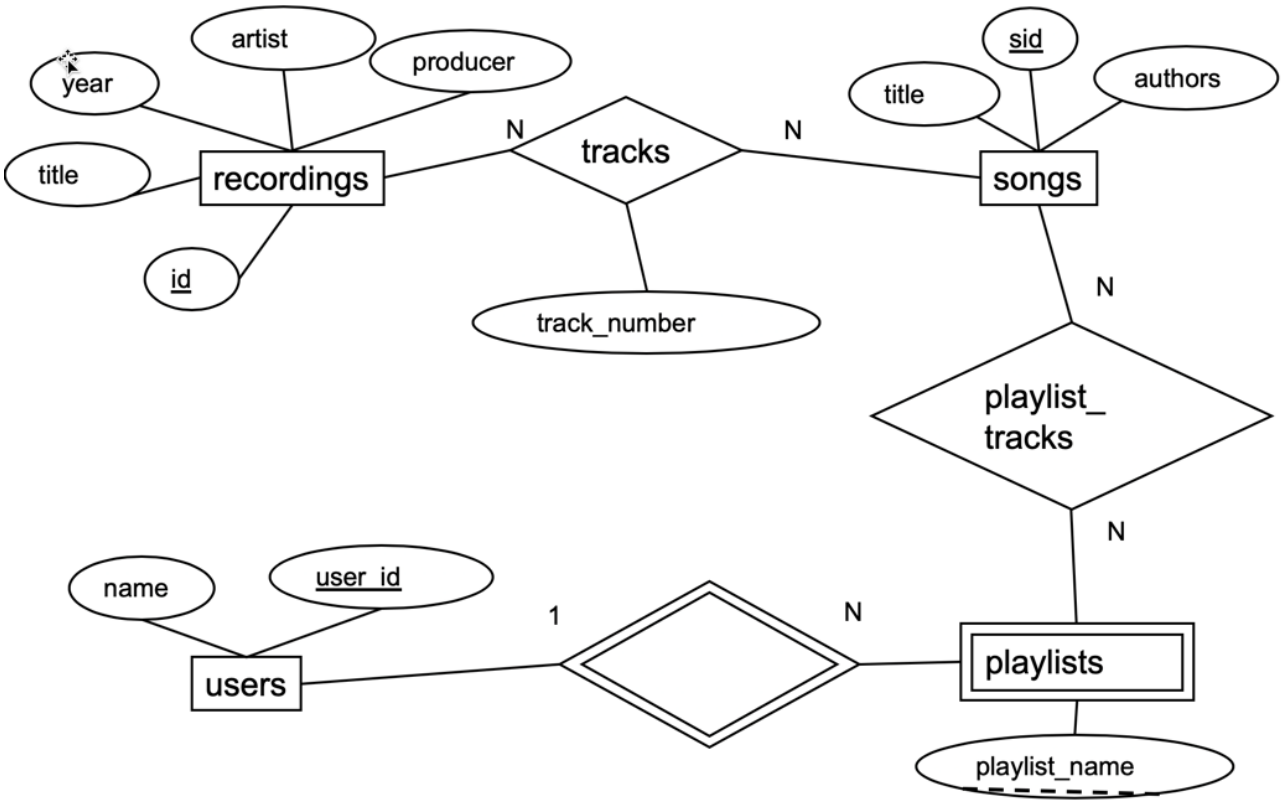
Every movie is of a particular screenplay. Also, a screenplay is used for only one movie. That is, there are not two different movies made of the same screenplay.

Scenes have a story-location where the story takes place and a filming-location where the filming will actually be done. Each scene has some actors that appear in that scene. Actors have a name, phone number, address and named agent that represents them. The database should keep track of which actors are needed for which scenes so it can be used to make up the shooting schedule.

A scene can be filmed more than once (maybe the actor forgot their lines). Each filming of a scene is called a “Take”. The movie is typically created by using the best take of each scene and putting them together in the correct order. The database will be used to print a "cutlist" which is a list of takes that make up a particular movie and the order in which the takes should appear. (In reality a movie is cut with partial takes but for this exercise we will assume complete takes are used to assemble the movie.)

Problem 2

The following is the ER model provided with the Beatles sample database and explored in the first lecture videos on ER models.



Our design has been discussed with the intended client and a fundamental issue uncovered. The proposed database has a **songs** entity that represents songs as a composition (title, composer) but does not capture the notion of an audio recording of the songs. Also the playlist_tracks relationship is placing the written version of songs on playlists rather than an audio recording of the song. For this problem we want you to redesign the database ER model to incorporate the following requirements.

R2.1 There should be a concept of an audio recording (e.g. mp3, or wav file) of a song.

R2.2 A song composition (i.e. title, composer, etc.) could be recorded multiple times. That is, there could be many recordings of a composed song.

R2.3 The tracks that appear on an album should refer to audio recordings of songs (not written compositions of songs).

R2.4 It should be audio recordings of songs that are placed on playlists (not written compositions of songs).

R2.5 Given an audio recording of a song, it should be possible to determine the composer of the song. That is, to relate an audio recording of a song to the title and composer information about the recorded song.

Problem 3

This problem is based on a scenario not unlike the sample project proposal for a fakebook database presented in assignment #1. Here is the scenario.

Many music students, music teachers and musicians, make use of "fake books" or "real books" as a source of songs (google this). These books present on one or two pages the melody and chords of a song. These books are very popular with improvising musicians and students who must both learn to play and to analyse the harmony in these tunes. Initially these books were part of an illegal trade but legal versions of most have now been published by reputable publishers. Many musicians have .pdf versions of these books. Some are copies of the illegal books and some scanned copies of books that they bought but want the convenience of a digital pdf copy.

We want you to design a database that can provide indexing information for well known books. That is, represent which song appears in which books and on which page. Also we want to provide information like title, composer, publisher, about the books and the songs that appear in them.

Finally the database should maintain a collection of users with user id and login password. An application will allow users to upload their own pdf copies of books and the database needs to keep track of which user the books belongs to. Users who are authenticated based on user id and password would be allowed to search the entire indexing database and be shown extracted pdf pages from books they themselves have uploaded.

Here are the details and requirements.

The database should provide indexing that catalogs the songs that appear in different well known books. The database should be usable both to search for songs and discover in which books the song appears and on what page.

It is expected that some, non-database, application code will allow users (musicians) to upload their own pdf copies of books and the database will keep track of which books which users have uploaded. When a musician has uploaded a book they have the right to be shown pages of that book located using the indexing data in the database. For copyright and legal reasons they will not be allowed to see contents from other user's pdf books. If two musicians upload the same identical book in theory the app need only keep one copy but for legal reasons individual copies must be kept.

The target size of the database would be upwards of 50,000 songs, and represent the contents of a few hundred books.

The database should store information about books which includes some sort of book code (unique key), title, publisher, and date of publication.

The song indexing information should represent which books a particular song appears in and on what page and also the length of the song (in terms of number of pages). Song information should include the song's title, composer(s), book, page number, and song length in pages.

Each actual uploaded pdf book should also have some kind of offset information to account for introductory pages, missing forewords etc. That is, if the song is indexed to be on page 1 but that is the 10th page of a particular user's scanned .pdf book then an offset should be stored in the database to account for this.

Design an E-R model for this database. You will have to make decisions about attributes and keys. If you don't think it is clear what your attributes mean then provide some notes and assumptions to go with your design. It is expected that this question will require discussion to clarify what is required. Make sure to ask lots of questions in the Discord forum about this problem.

Problem 4

Complexity Requirement R0.1 The proposed ER model should contain at least three entities and two N:N relationships. If it is too simple give a mark of 0 for this question. You might need to revise your assignment 1 propsal now that we have a more formal complexity measure.

This course has a project component. That is, an ongoing design of your own choosing. You are going to build your own database about something that interests you. In assignment #1 you provided a brief scenario and proposal for something that interests you. Now you want to create the first model of your database. Later in the course you will build this with SQLite and populate it with data. We will use this modeling opportunity to discover if your proposal is too simple to use as a course project.

DR4.1 [5 marks] For this problem you must re-write the scenario to describe what your database will be about, and then come up with the proposed E-R diagram and initial table schema as for the previous problems. You wrote a description in the previous assignment but we want you to repeat it here. This will give you a chance to make modifications and to make sure the information is here for whoever is marking your E-R model. (If it has not changed from assignment #1 then just copy and paste it here again.)
