CS 2300 – Homework 5

Submission is done electronically via Canvas
Total possible points: 100 points

Homework Guidelines

No Handwritten Answers

This homework will require some written responses and some diagrams. For the written responses, use a standard word processor like Microsoft Word, LibreOffice Writer, Google Docs, or LaTeX. Drawings can be made using Microsoft PowerPoint or Google Drawings[1], among others (e.g. draw.io, lucidchart). You can explore other tools as well, just search around and verify the tool you are considering is compliant with the design aspects that are presented in class.

Please create your answers in a digital format, export to a PDF, and upload that PDF to Canvas. No need to turn a printed copy. For those of you considering LaTeX, ShareLaTeX and Overleaf are good online editors.

One Problem per Page

In the past, many students turned in a single piece of paper with all answered questions smooshed onto it. This did not leave my grader or I with adequate room to provide suggestions on how a design could be improved nor explanations on errors. The process of design work is to work in iterative stages: the first stage will be rather messy (and can be hand-written), the second stage will have things cleaned up, and so on. The final stage is what you should turn in, and it should be free of scratched-out work and high-density drawings. Only answer one problem on *at least* one page, and refrain from super-compact answers that leaves more than 1/2 of the page empty. Ideally, leave somewhere between 1/4 to 1/2 of a page empty so that we can make annotations.

[1] https://docs.google.com/drawings/

Problem 1

Write ONE SQL file (extension .sql) that creates a simple database, detailed below. Submit your .sql file through Canvas.

Construct your SQL statements in accordance with the assumptions at the bottom of this page.

Indicate which database system (e.g. MySQL, PostgreSQL, SQLite, SQL*Plus, MariaDB, etc.) you have used to test your code as a comment (e.g. -- Tested on MySQL). It is expected that your SQL file can be directly used to build a database in that system. You can choose to install the database system onto your own computer (recommended so that reusing it for the course database project is possible), or test your .sql file on a web database platform.

AlbumTrack(rid:int, cat#:str, sid:int, trackno:int)

Label(lid:int, lname:str, labbr:str)

Song(sid:int, stitle:str, duration:int, remixof:int, artist:int)

Release(rid:int, rtitle:str, year:int, aid:int)

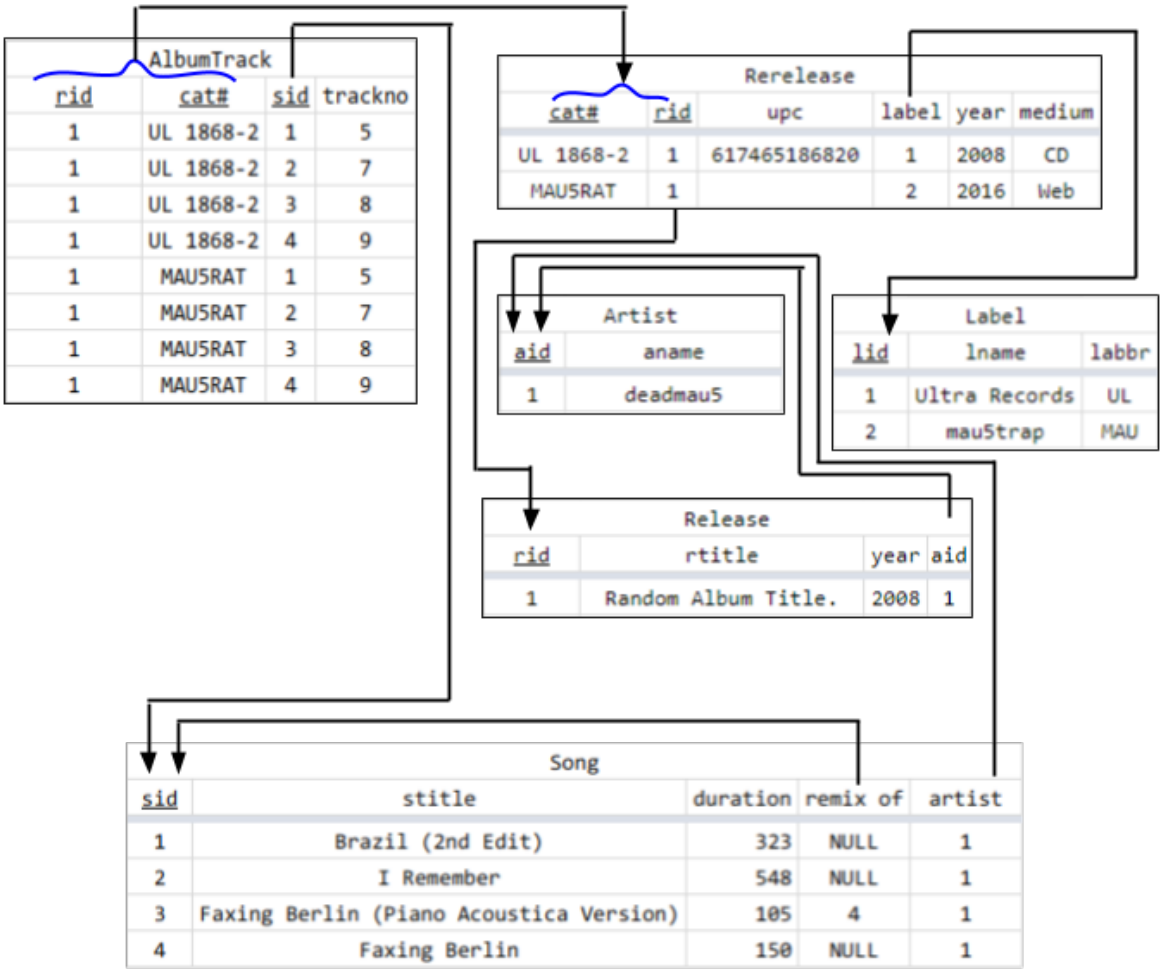
Rerelease(catno:str, rid:int, upc:str, label:int, year:int, medium:str)

Artist(aid:int, aname:str)

Figure 1: Sample Relational Model

Assumptions:

- Song durations are in units of seconds and must be > 0
- Years cannot be before 1900
- Unless otherwise stated, assume anything that is a string will not exceed 80 characters in length
- UPCs may not be longer than 12 characters
- Rerelease.medium can only be one of the following values: CD, Web, LP, 45, Tape
- Label.labbr will not be longer than 5 characters
- Any column in the relations in Figure 2 that does not have any NULL values (empty cells) should be assumed to have a NOT NULL constraint. Likewise, any column with a least one null value should be assumed to be nullable.



AlbumTrack

rid	cat#	sid	trackno
1	UL 1868-2	1	5
1	UL 1868-2	2	7
1	UL 1868-2	3	8
1	UL 1868-2	4	9
1	MAU5RAT	1	5
1	MAU5RAT	2	7
1	MAU5RAT	3	8
1	MAU5RAT	4	9

Rerelease

cat#	rid	upc	label	year	medium
UL 1868-2	1	617465186820	1	2008	CD
MAU5RAT	1		2	2016	Web

Artist

aid	aname
1	deadmau5

Label

lid	lname	labbr
1	Ultra Records	UL
2	mau5trap	MAU

Release

rid	rtitle	year	aid
1	Random Album Title.	2008	1

Song

sid	stitle	duration	remix of	artist
1	Brazil (2nd Edit)	323	NULL	1
2	I Remember	548	NULL	1
3	Faxing Berlin (Piano Acoustica Version)	105	4	1
4	Faxing Berlin	150	NULL	1

Figure 2: Sample Database State for Problem 3.2.

To-do:

(1) Create 6 tables as shown in Figure 1 below.

a. You can rename invalid attribute names appropriately - e.g. rename cat# to catno or catalog_number or whatever.

b. You need to create constraints on the tables. For example, create primary keys, foreign keys, and domain / not null constraints. See the assumptions.

(2) Insert values into the tables as shown in Figure 2 below.

(3) Create SELECT queries to show the contents of each table.

(4) Implement the following SQL queries.

a. Find the song titles from songs included on rereleases that were released both after 2008 and on the 'Ultra Records' label.

i. *Sample Output*

stitle

b. For each rerelease, list its associated release's title and the rerelease's year and the number of songs it contains.

i. *Sample Output*

rtitle	year	album_track_count
Random Album Title.	2016	4
Random Album Title.	2008	4

c. Compute the runtime (in seconds) of each rerelease and include its catalog number and associated release title.

i. *Sample Output*

rtitle	catno	total_duration
Random Album Title.	MAU5RAT	1126
Random Album Title.	UL 1868-2	1126

d. For each remixed song, display it's title, duration, (remixing) artist name, and the title and (original) artist name who it remixed.

i. *Sample Output*

remix	duration	remixing_artist	original	original_artist
Faxing Berlin (Piano Acoustica Version)	105	deadmau5	Faxing Berlin	deadmau5

(5) Drop the tables in an order so that no referential integrity constraints are violated (and thus no CASCADEs are necessary).