

Assignment 2 (40 marks)

Manohar Kaul

November 22, 2021

Abstract

Read the following instructions carefully before proceeding.

- For this assignment, you are expected to connect to a sqlite3 database "assignment2.db" (provided). You will read the sqlite3 documentation (<https://sqlite.org/cli.html>) to understand how to write SQL queries in sqlite3.
- sqlite3 comes pre-installed with most Linux and MacOSx operating systems. In case, you don't have it installed on your machine, go ahead and install it on your local machine.
- For each question, you **must provide the query and the output of your query** on the database. Just providing one without the other, will result in a zero for that question.

1 Counting (10 marks)

Find all the artists who have albums to their name, including those artists that do not. Report for each *first alphabet of the artist name*, the count of artists per alphabet.

This count should account for: (i) repetitions caused due to the same artist having multiple albums and (ii) artists having no albums.

For e.g., AC/DC is counted twice because of their two albums ...

```
1 | AC/DC | | For Those About To Rock We Salute You |
1 | AC/DC | | Let There Be Rock |
```

Your final output must look something like the following:

```
A | 200
B | 2
...
Z | 56
```

2 Full Outer Join (10 marks)

- Using the following DDL statements create the instructor and student tables and populate them with values.

```
-- create and insert data into the instructor table
CREATE TABLE instructor (
  name TEXT,
  course_id TEXT
);
```

```
INSERT INTO instructor (name, course_id)
VALUES
('Amy', 'CS1000'),
('Aaron', 'CS700'),
('Anne', 'CS400');
```

```
-- create and insert data into the student table
CREATE TABLE student (
  name TEXT,
  course_id TEXT
);
```

```
INSERT INTO student (name, course_id)
VALUES
('Jack', 'CS800'),
('Jones', 'CS1000'),
('Jason', 'CS450');
```

- Verify you have the tables setup correctly and that the data is populated correctly.
- Write a query that "emulates" a FULL OUTER JOIN on the student and instructor tables listing the instructor's name, the course they teach, the student's name, and the course they take.

But there is a caveat, you are ONLY allowed to use a LEFT OUTER JOIN clause to achieve the output one would get from a FULL OUTER JOIN!

3 Track counts (10 marks)

Classify the tracks based on its play length.

- a. “short”: less than a minute,
- b. “medium”: between 1 and 5 minutes,
- c. “long”: greater than 5 minutes.

Report the count of tracks per play length classification.

4 Generate some combinations! (10 marks)

- Using the following DDL statements, create table X and populate it with values.

```
create table X (  
  id_num integer,  
  id_str text  
);  
  
insert into X (id_num, id_str)  
values  
(1, 'A'),  
(2, 'B'),  
(3, 'C'),  
(4, 'D'),  
(5, 'E');
```

- Verify you have the tables setup correctly and that the data is populated correctly.
- For all entries in table X, generate all possible combinations of size 3 for both the numeric IDs (id_num) and corresponding string IDs (id_str). Note that a valid row should look like:

1,2,3 | A,B,C

but

3,2,1 | C,B,A

is **NOT** valid. Order does matter!