

CS430 Homework 4

Brendan Nguyen

Due: Thursday, Mar. 30, 2023

Question 1 (70 points)

Given the following DB schema:

- *Books* (bid: int, bname: string, author: string, pubyear: int, pubcompany: string)
- *Authors* (aid: int, name: string, rating: int, city: string, state: string)
- *Write* (aid: int, bid: int)

Primary keys are underlined in each relation. A book is uniquely identified by *bid*. A book has an id (*bid*), a name (*bname*), one author (attribute *author*), a publication year (*pubyear*), and a publishing company (*pubcompany*). An author is uniquely identified by *aid*. An author has an id (*aid*), a name (attr. *name*), a rating (attr. *rating*), a city (attr. *city*) and a state (attr. *state*). If an author wrote a book, a record will be present in the *Write* relation, with the *aid* of that author and the *bid* of that book.

For this schema:

- Write the SQL statements to create the three tables. Do not forget about the keys constraints. Table statements should be written in an order such that if executed in that order will not cause an error.
- Use `INNER JOIN` to write the SQL query to extract the id and name of the authors and the id, name, and pubyear of books they wrote. (Queries that do not use `INNER JOIN` or `JOIN` keywords for joining tables will receive no credit)
- Write the SQL query to extract the id and name of all the *Books* that do not have a pubyear. Sort the result by name in descending order.
- Write the SQL query to find the number of books for each pubcompany and pubyear.
- Write the SQL statement to join *Authors* with *Write*. In the result also include the authors that did not write any books.
- Write the SQL statement to update all *Authors* ratings to use rating 8.
- Write the SQL statement to update all books published in year 2020 to use pubcompany 'simon'.
- Write the SQL statement to delete all authors that did not write any book.
- Write the SQL statement to delete all *Books* that do not have a pubyear.
- Write the SQL table to insert a record into each of these three tables. Statements need to be written in an order such that if executed in that that will not cause an error.
- Write the SQL statement to update the record you inserted in *Authors* table at part (j). Update the name of the author and the rating to use different values.
- Write the SQL statement to alter table *Authors* to add an additional column *phone* of type string.
- Inside a comment line describe under what conditions query `SELECT COUNT(name) FROM Authors;` could return a result different than the result of query `SELECT COUNT(*) FROM Authors;`

- n) Drop all tables (note that the drop statements should be written in an order such that they can execute successfully when run in that order). Rewrite the create statement for table *Authors* to include the constraint that we only allow ratings between 1 and 10 (note: including 1 and 10).

Question 2 (30 points)

Given the following DB schema:

- *Cars* (carid: int, make: string, model: string, myear: int, dailyfee: real)
- *Customers* (custid: int, name: string, city: string, state: string, dob: date)
- *Rents* (carid: int, custid: int, rday: date)

Primary keys are underlined in each relation. A car is uniquely identified by `carid`. A car also has a make (e.g. Toyota), a model (e.g. Corolla), a manufacturing year (attr. `myear`), and a daily fee (attr. `dailyfee`). A customer is uniquely identified by attribute `custid`. A customer also has a name, city, state and a date of birth (`dob`). Customers rent Cars. If a customer rents a car, a record is inserted in *Rents* table with the `carid`, the `custid`, and the day (attribute `rday`) the rental is made for.

For this schema:

- Write the SQL statement to create table *Cars* with the constraint that no car older than 2015 (i.e. manufacturing year before 2015), and no car newer than 2020 (i.e. manufacturing year after 2020) can be inserted into the table. Please do not forget about the key constraints.
- Write the SQL statement to create table *Customers*. In addition to the key constraints, add the constraints that no field in the table can be NULL.
- Write the SQL statement to create table *Rents*. Add the key constraints. The day of the rental should always be provided.
- Write the SQL statement to find the id and name of customers who rented both Honda and Toyota cars (i.e. both make Honda and make Toyota).
- Write the SQL query to extract the `carid`, `make` and `model` for cars that were rented for some day in 2020, but they were not rented for any day in 2022.
- Write an INSERT statement to the table *Cars*. Right after the statement, explain in a comment line what should be changed in the insert to cause it to fail due to the constraint “no car older than 2015 or newer than 2020 can be inserted in the table”.