Homework 4

CS430 and CS630

100 points

Due Date: **Wed, Nov 9, 2022, before 8:30pm**

INSTRUCTIONS (please read carefully):
Homework MUST be submitted electronically (copy files to the users.cs Unix machine) before the due date following these instructions:

- For **Question 1** problems create an SQL file named **Q1.sql** that will contain the answers (SQL statements).
- For **Question 2** problems create an SQL file named **Q2.sql** that will contain the answers (SQL statements).
- **The SQL files** (named **Q1.sql** and **Q2.sql**) MUST be copied on the users.cs Unix machine before the due date, using the following instructions: Create a folder 'HW4' under your main folder for the course (cs630), and place the script files named Q1.sql and Q2.sql there. Ensure that the files are not readable by "others" (run for each filename the command chmod o-r filename) and that the files belong to the group CS630-1G and are readable by the group (run for each filename the command chmod g+r filename). DO NOT CHANGE PERMISSIONS FOR ANY OF THE DIRECTORIES (ESPECIALLY THE cs630 DIRECTORY IN YOUR HOMEDIR)!
- Students must have a cs unix account and must be enrolled in the cs630 class on the cs portal to be able to submit the homework.

No submission after the due date (November 9, 2022 before 8:30 pm) will be accepted. If any of the Q1.sql, Q2.sql files from the cs Unix machine is uploaded or modified after the due date, Nov 9, 2022 8:30pm, that file will not be graded and the student will receive no credit for that Question.

Submissions that do not follow the instructions from above and are not provided before Nov 9, 2022 8:30pm will not be accepted and will receive no credit. All submission must be electronically. No handwritten homework will be accepted.

The exercises starting with [CS630 only] are only for CS630 students (will not be graded for CS430 students)**. All the other exercises are for both CS430 and CS630 students.**

Important Notes:

- SQL statements must run against the Oracle database we use in class. (Please run and test your queries against the Oracle DB. Create the tables, insert some data, and test your queries!!!)
- SQL queries that do not run successfully against the Oracle DB will receive 0 points
- An SQL statement ends with a semicolon ;

- In the Q1.sql and Q2.sql files, before each SQL statement you MUST include a comment line with the problem number the sql statement is for (e.g., before writing the SQL query for (c) add a comment line such as --Answer for c)). Remember that a comment line starts with two dash symbols. Any other additional comments can be written in comment lines.

Question 1) (70 points)

Given the following db schema:

Books(bid:integer, bname:string, author:string, pubyear:integer,pubcompany:string)

Authors(aid:integer,name:string, rating: int, state:string)

Write(aid:integer,bid:integer)

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) 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.

Notes:

- For both CS430 and CS630 students, each problem (a through n) carries 5 points possible.

For this schema:

a) 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.
b) Write the SQL query to extract all the Books that have a pubcompany (hint: pubcompany is different than null).
c) Write the SQL query to extract all Books that don’t have a pubcompany.
d) Write the SQL query to find the number of authors for each state and rating.
e) Use INNER JOIN to write the SQL query to extract the name of the authors and the id and name of books they wrote. (Queries that do not use INNER JOIN or JOIN keyworks for joining tables, will not be given any credit)
f) Write the SQL statement to join Authors with Write. In the result also include the authors that did not write any books.

g) 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.

h) Write the SQL statement to update the record you inserted in Books at point (h). Update the name of the author and the pubyear to use different values.

i) Write the sql statement to update the all books published by 'penguin' company, to use pubyear 2000.

j) Write the SQL statement to update all Authors ratings to use rating 10.

k) Write the SQL statement to delete all authors that do not have any book they wrote.

l) Inside a comment line, describe in what case SELECT COUNT(*) FROM Books; could return a result different than the result of SELECT COUNT(bname) FROM Books;

m) Write the SQL statement to alter table Authors to add an additional column age of type real.

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 5 (note: including 1 and 5).

Question 2) (30 points)

Given the following db schema:

Cars(carid:integer, make:string, model:string, myear:integer, dailyfee:real)

Customers(custid:integer, name:string, city: string, state:string, dob: date)

Rents(carid:integer, custid:integer, rday:date)

Primary keys are underlines 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.

Notes:

- For CS430 students, each problem (a through e) carries 6 points possible.
- For CS630 students, each problem (a through f) carries 5 points possible.

a) Write the SQL statement to create table Cars with the constraint that no car older than 2010 can be inserted in the table (i.e. manufacturing year not before 2010). Please do not forget about the key constraints.

b) Write the SQL statement in create table Customers. In addition to the key constraints, add the constraints that no field in the table can be null.

c) Write the SQL statement to create table Rents. Add the key constraints. The day of the rental should always be provided.

d) Write the SQL statement to find the id and name of customers who rented all cars.

e) 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 2010 can be inserted in the table’.

f) [CS630 only] Write the SQL query to extract the carid, make and model for cars that were rented for some day in 2022 , but they were not rented for any day in 2021.