

Homework 4

CS430 and CS630
100 points

Due Date: **Mon, Aug 14, 2023, before 10:00 am**

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 **<studentId>_Q1.sql** that will contain the answers (SQL statements), where <studentId> is replaced by your student id (e.g. if your student id is 12345, then the file will be named 12345_Q1.sql).
- For **Question 2** problems create an SQL file named **<studentId>_Q2.sql** that will contain the answers (SQL statements), where <studentId> is replaced by your student id (e.g. if your student id is 12345, then the file will be named 12345_Q2.sql).
- **The two SQL files** 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 SQL script files. 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 will be accepted. If any of the SQL files from the cs Unix machine is uploaded or modified after the due date, that file will receive no credit. All submissions must be electronically. No handwritten homework will be accepted.

All 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 in SQLPlus!)
- SQL queries that do not run successfully against the Oracle DB will receive 0 points
- An SQL statement ends with a semicolon ;
- In the each SQL file, 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, city:string, 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), 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.

Notes:

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

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.

- k) Write the SQL statement to update the record you inserted in Authors table at point (j). Update the name of the author and the rating to use different values.
- l) Write the SQL statement to alter table Authors to add an additional column phone of type string.
- m) 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: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 both CS430 and 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 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.
 - b) 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.

- 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 both Honda and Toyota cars (i.e. both make Honda and make Toyota).
- e) 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.
- f) 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'.