Homework 3

CS430 and CS630

100 points

Due Date: **Wed, Oct 26, 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 'HW3' 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 (Oct 26, 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, Oct 26, 2022 8:30pm, that file will not be graded and the student will receive 0 points for that Question.

Submissions that do not follow the instructions from above and are not provided before Oct 26, 2022 8:30pm will not be accepted and will receive 0 points.

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) (40 points)

Given the following db schema:

Customers(cid: int, firstname: string, lastname: string, city: string, state: string)
Has_account(cid: int, aid: int, since: date)
Accounts(aid: int, atype: string, amount: real)

Primary keys are underlined in each relation. Relation Customers contains information about customers. A customer is uniquely identified by cid. Relation Accounts contains information about bank accounts. An account is uniquely identified by aid. The type of the account is given in column atype (for example atype='savings'). Customers have accounts.
For a customer that has an account (i.e. is the owner of that account) there is a record in relation Has_account, that has the cid of that customer and the aid of that account, as well as the date when the account was opened (column since).

Notes:

- For CS430 students, each problem (a through h) carries 5 points possible.
- For CS630 students, each problem (a through j) carries 4 points possible.

Using this db schema:

a) Write the SQL statements to create the tables from this db schema. Do not forget to define the necessary key constraints. The statements should be written in an order such that if executed in that order will not cause an error.
b) Write the SQL query that extracts the id, first name and last name of customers from state 'MA' that have some account with an amount greater than 1000. The result should not contain duplicates. Sort the result by the last name in an ascending order.
c) Write the SQL query to find how many accounts of each type are in the database.
d) Write the SQL to get the number accounts each customer from state 'MA' has. In the result include only the ones that have at least 3 accounts.
e) Write the SQL query to extract the id and the first name of the customers who have at least 2 accounts of type 'checking'.
f) Write the SQL to extract the id, first name, last name of customers who opened accounts both in 2018 and 2020.
g) Write the SQL query to extract the id and the last name of customers who did not opened any account after Jan 1st 2020.

h) Write the SQL query to extract the id and the last name of customers who have both 'savings' and 'checking' types of accounts.
i) [CS630 only] Write the SQL query to extract the last name and id of customers who have at least 20,000 across all their accounts.
j) [CS630 only] Write the SQL query to extract all accounts ids of type checking that have at least 2 owners.

Question 2) (60 points)

Given the following db schema:

Books(bid:integer, bname:string, author:string, pubyear:integer,pubcompany:string)
Students(sid:integer,sname:string, age:real, state:string)
Reads(sid:integer,bid:integer,year: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). A student is uniquely identified by sid. A student has an id (sid), a name (attr. sname), age (attr. age) and a state (attr. state). If a student reads a book, a record will be present in the Reads relation, with that sid and bid and the year the book was read.

Notes:

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

For this db schema:

a) Write the SQL statements to create the tables from this db schema. Do not forget to define the key constraints. The statements should be written in an order such that if executed in that order will not cause an error.
b) For each table, write an insert statement to insert one record. The statements should be written in an order such that if executed in that order will not cause an error.
c) Write the SQL statement to find the students who live in 'MA' and they are either younger than 25 or older than 35.
d) Write the SQL statement to find the number of books that have an author whose name starts with letter B.
e) Write the SQL statement to find the oldest books (hint: pubyear is min).
f) Write the SQL statement that extracts the average age of students by state. Show this information only for states where there are at least 50 students.
g) Write the SQL statement to find the id and name of students who read all books. The result should contain no duplicates.

h) Write the SQL statement to find the id, name and state of the students who read all books published by 'penguin' publishing company. The result should contain no duplicates. Sort the result by the name of the students in descending order.
i) Write the SQL statement to find the id and name of students who read some books published by the publishing company 'penguin' and did not read any book published by company 'simon'.
j) Write the SQL statement to find the names of the students who read some book in the same year when that book was published.
k) [CS630 only] Find the name, age and state of the students who did not read all books published by 'penguin' publishing company in year 2020. The result should contain no duplicates. (note: 2020 is the year when the book was published).
l) [CS630 only] Write the SQL that extracts the id, name and author of books published by 'penguin' publishing company, that were read by all students. The result should contain no duplicates.