

CS430 Homework 3

Brendan Nguyen (#01763771)

Due: Wednesday, Mar. 15, 2023

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 a 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 a 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 SQL files 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 `<studentId>_Q1.sql` and `<studentId>_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 will be accepted. If any of the two SQL files from the cs Unix machine is uploaded or modified after the due date, March 15, 2023 7pm, that file will not be graded and the student will receive no credit.

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 using SqlPlus CLI)
- SQL queries that do not run successfully against the Oracle DB will receive no credit.
- An SQL statement ends with a semicolon ; Queries must contain the ; at the end.
- In the 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, name: string, city: string, state: string, age: int)

- *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).

Using this db schema:

- 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.
- Write SQL statements to insert two records in each table. The statements should be written in such an order such that if executed in that order do not cause an error.
- Write the SQL query that extracts the id, name of customers from city 'Boston' that have some account with an amount smaller than 5000. The result should not contain duplicates. Sort the result by name in descending order.
- Write the SQL query to extract the id, name, and age of customers who did not opened any account between Jan 1st 2020 and Dec 1st 2021 (including these dates).
- Write the SQL query to extract the id, name, and age of customers who have both 'savings' and 'checking' types of accounts.
- Write the SQL query to extract the name and id of customers who have at most 10,000 across all their accounts.
- Write the SQL query to extract all accounts ids of type checking that have at least 2 owners.
- Write the SQL to get the number of accounts each customer older than 25 and younger than 35 has. In the result include only customers who have at least 2 accounts.
- Write the SQL to extract the id, name, and age of customers who opened accounts both in year 2018 and 2020.
- Write the SQL query to extract the id and the name of the customers who are from 'MA' state and who have at least 2 accounts of type 'savings'.

Question 2 (60 points)

Given the following db schema:

- *Articles* (aid: integer, aname: string, first_author: string, pubyear: integer, pubcompany: string)
- *Students* (sid: integer, sname: string, age: real, state: string)
- *Reads* (sid: integer, aid: integer, year: integer)

Primary keys are underlined in each relation. An article is uniquely identified by aid. An article has an id (aid), a name (aname), one first author (first_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 an article, a record will be present in the Reads relation, with that sid, aid, and the year the article was read.

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 number of articles that have a first_author whose name contain string 'an'. The query has to be case insensitive with regards string 'an'.
- d) Write the SQL statement to find information about the newest articles (hint: pubyear is max). Sort the result by the name of the article in ascending order.
- e) Write the SQL statement to find the id, name and age of students who read all articles. The result should contain no duplicates.
- f) Write the SQL statement to find the id and name of students who read some articles published in 2020 and did not read any article published in 2018. The result should be sorted by the name of the students in descending order.
- g) Write the SQL statement to find the ids and names of the students who read some articles in the same year when that article was published.
- h) Write the SQL that the number of articles read by each student. The result should contain information only about those students who read at least 3 articles.
- i) Write the SQL statement that extracts the minimum, maximum and average age of students for each state. Show this information only for states where there are at least 2 students.
- j) Write the SQL statement to find the articles whose first author name (first_author) starts with letter 'B' and that are published either before 2018 or after 2020 (note that 2018 and 2020 are included).
- k) Write the SQL statement to find the id, name of the students who read all articles published in year 2022 by the publishing company 'penguin'. The result should contain no duplicates.
- l) Find the id, name, age, and city of the students who did not read all articles published by 'simon' publishing company. The result should contain no duplicates.