# Homework 6

CS430 and CS630

100 points

Due Date: **Monday, August 21, 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** create a Python file called **<studentId>_Q2.py ,** where <studentId> is replaced by your student id (e.g. if your student id is 12345, then the file will be named 12345_Q2.py ).
- For **Question 3** create a Python file called **<studentId>_Q3.py ,** where <studentId> is replaced by your student id (e.g. if your student id is 12345, then the file will be named 12345_Q3.py ).
- **The SQL and Python files** MUST be copied on the users.cs Unix machine before the due date, using the following instructions: Create a folder 'HW6' under your main folder for the course (cs630), and place the SQL and .py files 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 SQL or Python files from the cs Unix machine is uploaded or modified after the due date, that file will not be graded, and the student will receive no credit. All submission must be electronical. 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!!!)
- 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 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 (but not within the SQL statement).
- Python Code: up to 20% can be deducted for code formatting.
- Python code must be tested on the unix machine and run against the Oracle db. To run a Q2.py file you need to use python3 Q2.py command. To run a Q2.py file please use python3 Q3.py command.
- Python files that do not run (i.e. they give an error when run) could receive at most 50% credit.

## **Question 1)** (40 points)

Given the following db schema:

Articles(aid:integer, title:string, author:string, pubyear:integer)

Students(sid:integer,name:string, city: string, state:string, age:real, gpa:real)

Reads(aid:integer,sid:integer, rday: date)

Primary keys are underlined in each relation. An article is uniquely identified by aid. An article has an id (aid), a title, an author and a publication year (pubyear). A student is uniquely identified by sid. A student has an id (id), a name (attr. name), a city, a state, and age and a gpa. If a student reads an article, a record will be present in the Reads relation, with the sid of that student, the aid of that article and the date the article was read (rday attribute).

Notes:

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

For this schema:

a) Write the SQL statement to create the table Articles. Do not forget about the key constraints.
   Write the SQL statement to create table Students. Add the constraint that gpa should be between 1 and 4 (including 1 and 4). Do not forget about the key constraints.
   Write the SQL statement to create table Reads. Add the constraint that no attribute can be null. Do not forget about the key constraints.

b) Write the SQL statement to create an index on column rday. Explain when such an index will be useful.
c) Write the INSERT statements to inserts 3 students.
Write the INSERT statements to inserts 2 articles.
d) Write the INSERT statements to inserts some records into Reads following these conditions: one of the students from (c) read all articles inserted for (c). Another student from (c) read one article inserted for (c). One student from (c) read no article.
e) Write the SQL statement to create a View called MAStudents that contains all the information for Students from MA
f) Write the SQL statement to create a View called StudentsReads that contains information about the id, name and city of students and the id and title of article they read.
g) Write an SQL query that uses the view from (f) (view StudentsReads) to extract the count of articles read by each student. Queries that do not use the view StudentsReads are given no credit.
h) Write the SQL statements to drop the 2 views: StudentsReads, MAStudents.

## **Question 2)** (30 points)

- **Python Code: up to 20% can be deducted for code formatting.**

Using the schema from Question 1, write a Python file that uses Pandas library and does the following:

- Reads from the input: an Oracle username, Oracle password, Oracle hostname, Oracle db name.
- Connects to our Oracle DB
- Uses PANDAS library to run a query against the DB that extracts information about all Students. Saves the results in a pandas dataframe.
- Prints out the name of the columns of that dataframe.
- Prints out the shape of the dataframe.
- Prints out the first 3 records from the dataframe.
- Uses pandas aggregates to extract the average and min age of students. Prints the value.
- Uses pandas aggregates to get the minimum and maximum gpa for students. Prints the result.
- Uses pandas aggregates to get the sum of gpa values. Prints that result.
- Runs a second query against the DB to extract information about the id, name and state of students and the id and title of articles they read ( the resulted relation will have the SID,NAME,STATE, AID, TITLE columns). Save the result in a Pandas dataframe.
- Prints this news dataframe.
- Prints how many records are in the new dataframe.
- Prints how many columns are in the new dataframe.
- Prints the name of the columns from this new dataframe.

- Uses Pandas to filter this dataframe to keep only students from state MA. Saves the result into a third dataframe.
- Use pandas group by to extract how many articles each student from MA read. Prints the result

Note: please remember to close the connection.

## **Question 3)** (30 points)

- **Python Code: up to 20% can be deducted for code formatting.**

Using the schema from Question 1, write a Python file that uses the connection.cursor() to execute queries against the DB. The program should do the following:

- Reads from the input: an Oracle username, Oracle password, Oracle hostname, Oracle db name.
- Connects to our Oracle DB
- Uses the cursor to drop tables Students, Articles, Reads. Code must gracefully handle any exception, for the case these tables it tries to drop were not in the DB.
- Uses cursor to re-create the 3 tables from Schema from Question 1)
- Uses the cursor to Insert two records in each table.
- Uses the cursor to run a select query that extracts all articles. Prints all records extracted.
- Uses the cursor to run a select query that extracts all students. Prints all records extracted.
- Uses the cursor to run a select query that extracts all records from Reads. Prints all records extracted.

Note: please remember to commit the transaction and close the connection.