EECS484 Database Management Systems
Homework 2 (100 points)
Due 11:55 PM, Oct. 4th 2017

Please read the following instructions before starting the homework. Failing to follow the submission rules will lead to -4 points (e.g., wrong file format).

You should have access to the following files:
booktown.sql
booktownqueries_problems.sql
drop_booktown.sql

There are two parts to this homework:

Part 1: You are asked to write SQL queries given a schema. We will not be providing you with a database for this question. It is advised that you think of some way to be able to check if your queries really work the way they should. Please submit this part in separate files specified in each of the problem.

Part 2: This part requires you to look at the database associated with a sql file called ***booktown.sql***. This file should be attached along with this document on canvas. You will need your Oracle account to load and see the database. Please answer question 5 in the file named ***booktownqueries_problems.sql***. Question 6 is to be answered in a pdf file called ***hw2q6.pdf***. Please neatly write the relational algebra questions and scan them with a high quality scanner (either a good phone or a scanner found in the library). Alternatively, you can write them using an equation writing software (e.g. LaTeX).
Correct solutions that are hard to read may lose points. (-5)

================ *Working individually!!* =======================
This assignment is to be completed *individually, not in groups*
COE honor code applies.
You may discuss concepts with other students, but you may not copy solutions
==

No late days for homework! *If you miss the due date, you get 0 points. No exceptions on this.*

By submitting this homework, you are agreeing to abide by the Honor Code:
I have neither given nor received unauthorized aid on this assignment, nor have I concealed any violations of the Honor Code.

Part 1 (40 points)

Write some queries for the given relational schema. You are given the following relations (primary keys are underlined):

Student (**sid**, name, major) : student ID, name, and majors of students
Project (**pid**, ptitle) : project ID and title of projects
Course (**cid**, title) : course ID and title of courses
Member (**pid**, **sid**) : Relationship. Student sid is a member of project pid
Enrolled (**sid**, **cid**) : Relationship. Student sid is enrolled in course cid.

Please submit the answers to these questions in separate files as shown. The files should execute within sqlplus and should only answer the query (no need to create tables).

Question 1 (10 points):
Find the student IDs of all students who are enrolled in (‘EECS484’ and ‘EECS485’) or enrolled in (‘EECS482’ and ‘EECS486’) or enrolled in ‘EECS281’. For example, student A will be in the output if student A is enrolled in (‘EECS484’ and ‘EECS485’) or (‘EECS482’ and ‘EECS486’) or (‘EECS281’). *Note that ‘EECS484’, ‘EECS485’, ‘EECS486’, ‘EECS482’ and ‘EECS281’ are course titles.*
Correct solutions that use views, nested queries will only receive 50% of the grade. (This query can be done without views, nested queries)

FIle: q1.sql

Question 2 (10 points):
Find the unique student IDs of all students who have a project partner who is enrolled in either 'EECS482' or ‘EECS483’ and either ‘EECS484’ or ‘EECS485’ and ‘EECS280’. For example, student A will be the output if students A and B are project partners and B is enrolled in (‘EECS482’ or ‘EECS483’) and (‘EECS484’ or ‘EECS485’) and (‘EECS280’).
Correct solutions that use views, nested queries, or set operation like intersections, minus, or unions will receive only 50% credit. (This query can be done without views, nested queries, intersections, minus, or unions)

FIle: q2.sql

Question 3 (10 points):
We would like to know the **CS majors** who are taking heavy CS-based courses. We define heavy CS-based courses as courses where less than 20 non-CS majors are enrolled. Return all the student IDs (sid) and names of these CS students. The result of the query should be output in increasing order by IDs.
You may use views or nested queries for this part.

File: q3.sql

Question 4 (10 points):

The school is interested in seeing the possible combinations of students who take a course together but are not project partners. Define a VIEW called **StudentPairs** that contains all student ID pairs (sid1, sid2) with the following property: The students who are enrolled in a common course but are not members of the same project. List each pair of students only once. For example, if you list (1, 2), do not list (2, 1). Student IDs can be assumed to be integers.

You may use views or nested queries for this part. Correct solutions that use extra tables than needed will receive only 50% credit.

Hint: Try not to lose potential student pairs when joining your tables!

FIle: q4.sql

Part 2 (60 points)

We have provided you a sample database in **booktown.sql**. Please look through the beginning of this file to understand the schema. Feel free to add more sample data to test your answers. However, don't change the schema! We *will* be testing your answers using the schema that we have given you.

To build the database in SQLPlus, log into CAEN and navigate to the directory where booktown.sql is located at. Run sqlplus as in Project 1 to connect to Oracle and load booktown.sql:

```
SQL> START booktown.sql
```

Question 5 (40 points):

- Please provide queries to the questions listed in **booktownqueries_problems.sql**.
- There should be 5 questions in total in this file.
- Your queries should work for any database with the schema provided in **booktown.sql**, *not* just with the same data we have given you.
- If you ever make intermediate view, make sure you ***drop*** them at the end of that question.

Clarifications:

For problem 3 in question 5, *List titles, publication, author's id, author's last name, and author's first name of all books by authors who have published a book after 1995-10-01 but before 2005-10-01*. Here ***who have published a book*** we mean **at least one book** rather than exactly one book.

IMPORTANT: Ensure that once you are done, the entire **booktownqueries_problems.sql** can run completely without errors by:

```
SQL> START booktownqueries_problems.sql
```

Question 6 (20 points):

Write Relational algebra expressions for your solutions to Q5.1, Q5.2, Q5.3 and Q5.4 (in booktownqueries_problems.sql).

These do not involve aggregation and should be doable with relational algebra.

Submission

Any missing files will **not be accepted** after the deadline, and you will be given a 0 for it.

Instruction on submitting homework:

1. Prepare the .zip file (q1.sql, q2.sql, q3.sql, q4.sql, booktownqueries_problems.sql)
2. Submit *homework2.zip* on Canvas.
3. Submit *hw2q6.pdf* on Gradescope.

Files to be submit

- ❑ **q1.sql**
- ❑ **q2.sql**
- ❑ **q3.sql**
- ❑ **q4.sql**
- ❑ **booktownqueries_problems.sql**
- ❑ **hw2q6.pdf** that contains RA expressions for Question 6

How to create a zip file?

Log into a Linux machine. Put all your submission files into one folder

```
% zip homework2.zip q1.sql q2.sql q3.sql q4.sql booktownqueries_problems.sql
```

You MUST create the zip file using the above command as exactly typed. That ensures that you include the correct set of files with exactly the right names. You can add in a README.txt file if you wish to include any additional information.

Make sure your .sql files can run in a CAEN environment using SQLplus from start to finish. Failure to run (in any way) will result in you getting a 0. MAKE SURE IT WILL RUN!!