

Washington State University
School of Electrical Engineering and Computer Science
CptS 451 – Introduction to Database Systems
Spring 2020

Dr. Sakire Arslan Ay

Homework-4

Due Date: Wednesday, March 11 th 11:59pm

Name: _____

Student Number: _____

Question:	Max points:	Score:
1	10	
2	10	
3	10	
4	10	
5	10	
6	10	
7	10	
8	10	
9	10	
10	10	
ExtraCredit	5	
Total	100	

Consider the following database schema (also in **schema.sql**):

```
CREATE TABLE Majors (  
    major      VARCHAR(12),  
    description VARCHAR,  
    PRIMARY KEY(major)  
);  
CREATE TABLE Course (  
    courseMajor VARCHAR(12),  
    courseNo     VARCHAR(6),  
    credits      INTEGER NOT NULL,  
    enroll_limit INTEGER,  
    PRIMARY KEY(courseNo, courseMajor),  
    FOREIGN KEY (courseMajor) REFERENCES Majors(major)  
);  
CREATE TABLE Tracks (  
    trackMajor VARCHAR(12),  
    trackCode   VARCHAR(10),  
    title       VARCHAR,  
    PRIMARY KEY(trackMajor, trackCode),  
    FOREIGN KEY (trackMajor) REFERENCES Majors(major)  
);  
CREATE TABLE Student (  
    sID         CHAR(8),  
    sName       VARCHAR(30),  
    studentMajor VARCHAR(12),  
    trackCode    VARCHAR(10),  
    PRIMARY KEY(sID),  
    FOREIGN KEY (studentMajor, trackCode) REFERENCES Tracks(trackMajor, trackCode)  
);  
CREATE TABLE Enroll (  
    courseMajor VARCHAR(12),  
    courseNo     VARCHAR(6),  
    sID          CHAR(8),  
    grade        FLOAT NOT NULL,  
    PRIMARY KEY (courseMajor, courseNo, sID),  
    FOREIGN KEY (courseMajor, courseNo) REFERENCES Course(courseMajor, courseNo),  
    FOREIGN KEY (sID) REFERENCES Student(sID)  
);  
CREATE TABLE Prereq (  
    courseMajor VARCHAR(12),  
    courseNo     VARCHAR(6),  
    preMajor     VARCHAR(12),  
    preCourseNo  VARCHAR(6),  
    PRIMARY KEY (courseMajor, courseNo, preMajor, preCourseNo),  
    FOREIGN KEY (courseMajor, courseNo) REFERENCES Course(courseMajor, courseNo),  
    FOREIGN KEY (preMajor, preCourseNo) REFERENCES Course(courseMajor, courseNo)  
);  
CREATE TABLE TrackRequirements (  
    trackMajor VARCHAR(12),  
    trackCode   VARCHAR(10),  
    courseMajor VARCHAR(12),  
    courseNo     VARCHAR(6),  
    PRIMARY KEY (trackMajor, trackCode, courseMajor, courseNo),  
    FOREIGN KEY (trackMajor, trackCode) REFERENCES Tracks(trackMajor, trackCode),  
    FOREIGN KEY (courseNo, courseMajor) REFERENCES Course(courseNo, courseMajor)  
);
```

Please complete the following before you answer the homework questions:

1. On PostgreSQL Server create a database named hw4.
2. Download and extract HW4-DB.zip – attached to HW4 on Blackboard. There are 8 sql script files in this archive: schema.sql, majors.sql, course.sql, tracks.sql, student.sql, enroll.sql, prereq.sql, and trackReq.sql.
3. Create the tables Majors, Course, Tracks, Student, Enroll, Prereq, and TrackRequirements by running the CREATE TABLE statements in schema.sql file. Make sure to create the tables in the given order, otherwise you will get errors due to foreign key definitions.
4. Populate your DB by running the schema.sql, majors.sql, course.sql, tracks.sql, student.sql, enroll.sql, prereq.sql, and trackReq.sql script files in order. These files contain the INSERT statements for the above tables.
 - See Appendix-1 for instructions on how to execute script files on command line.
 - You may alternatively copy and paste the INSERT statements on the DBMS client (pgAdmin or command line) and execute them. Follow the order specified above when running insert statements. Otherwise there would be violations for the foreign key constraints.
5. Check if the data is inserted correctly by running a “select * from” on each table. The tables sizes are given below:
 - Majors: 7 ; Course.sql: 101 ; Tracks.sql: 6 ; Student: 50 ; Enroll: 136; Prereq: 94; TrackRequirements: 75

Part A. Write the following queries in SQL for the above database schema:

(Note: For each of the below you should write a single query. Answers that involve more than one query won't be accepted.)

1. Find the distinct courses that “Software Engineering (SE)” track students in 'CptS' major are enrolled in. Return the major, courseNo, and credits for those courses – sorted by major and courseNo.

coursemajor	courseno	credits
CHE	211	3
CptS	121	4
CptS	122	4
CptS	223	3
CptS	260	3
CptS	317	3
CptS	322	3
CptS	323	3
CptS	355	3
CptS	360	3
CptS	421	3
CptS	422	3
CptS	423	3
CptS	451	3
CptS	460	3

CptS	484	3
CptS	487	3
MATH	171	4
MATH	172	4
MATH	216	3
MATH	220	3

(21 rows)

2. Find the sorted names, ids, majors and track codes of the students who are enrolled in more than 18 credits (19 or above).

sname	sid	studentmajor	trackcode	sum
Aaron	12584489	ME		63
Ali	12582389	CptS	SE	30
Alice	12583589	CptS	SYS	52
Ally	12579189	CHE		24
Amir	12582989	CHE		21
Bill	12581189	CptS	SE	27
Jack	12584789	CptS	SE	52
Nick	12582689	CHE		36
Sam	12567189	ME		20

(9 rows)

3. Find the courses that only 'SE' track students in 'CptS' major have been enrolled in. Give an answer without using the set EXCEPT operator.

coursemajor	courseno
CptS	323
CptS	422
CptS	484
CptS	487

(4 rows)

4. Find the students who were enrolled in the courses that 'Ali' was enrolled in and earned the same grade as 'Ali' in those courses. Return the name, id, and major of the student as well as the courseNo, major, and student's grade for those courses.

sname	sid	studentmajor	coursemajor	courseno	grade
Bill	12581189	CptS	MATH	220	3
Jack	12584789	CptS	CptS	223	2.75
Jack	12584789	CptS	CptS	260	3
Jack	12584789	CptS	CptS	355	3.25

(4 rows)

5. Find the students in 'CptS' major who are not enrolled in any classes. Return their names and sIDs. (Note: Give a solution using OUTER JOIN)

sname	sid
Ally	12514189
Li	12576189
Mick	12565189
Sam	12254189
Tom	12354189

(5 rows)

6. Find the courses whose enrollments exceed their enrollment limits (i.e., the total enrollment is greater than the enrollment limit of the course). Return the course major, courseNo, enrollment limit, and the actual enrollment for those courses.

coursemajor	courseno	enroll_limit	enrollnum
CHE	110	4	6
CHE	211	4	7
CHE	321	4	5
CptS	260	3	4
EE	499	2	3
MATH	251	4	6

(6 rows)

7. Find the courses which are prerequisites for more than 5 courses. Return the major, courseNo, and the number of the successor courses.

premajor	precourseno	count
ME	220	7
EE	311	7
MATH	108	6
CptS	223	8

(4 rows)

8. Find the 'CptS' major students who passed a course but failed the prerequisite of that course, i.e., got a grade lower than "2". (For example, Alice (sid: 12583589) passed CptS355 but had a grade 1.75 in the prerequisite course CptS223.) Return the names and sIDs of those students and the courseno of the course (i.e., the course whose prereq had a low grade).

sname	sid	coursemajor	courseno
Alice	12583589	CptS	317
Alice	12583589	CptS	322
Alice	12583589	CptS	355
Alice	12583589	CptS	360
Alice	12583589	CptS	451

(5 rows)

9. For each 'CptS' course, find the percentage of the students who passed the course. Assume a passing grade is 2 or above. (Note: Assume that students who didn't earn a grade in class should be excluded in average calculation).

coursemajor	courseno	passrate
CptS	121	100
CptS	122	100
CptS	223	66
CptS	260	75
CptS	317	100
CptS	322	100
CptS	323	100
CptS	355	75
CptS	360	100
CptS	421	100
CptS	422	100

CptS	423	66
CptS	451	100
CptS	460	100
CptS	484	100
CptS	487	100

(16 rows)

10. Write the equivalent SQL query for the following relational algebra expression.

$R1 = \text{Student} \bowtie_{(\text{Student.sID}=\text{Enroll.sID AND Student.studentMajor}=\text{Enroll.courseMajor})} \text{Enroll}$

$R2 = \sigma_{\text{gpa} \geq 2} (\gamma_{\text{sID}, \text{avg}(\text{grade}) \rightarrow \text{gpa}} R1) \bowtie_{\text{Student.studentMajor}=\text{Majors.major}} \text{Student} \bowtie \text{Majors}$

$\pi_{\text{sID}, \text{sName}, \text{studentMajor}, \text{description}, \text{gpa}} R2$

sid	sname	studentmajor	description	gpa
12534189	John	CE	Civil Engineering	2.5
12567189	Sam	ME	Mechanical Engineering	3.5
12578189	Allison	CHE	Chemical Engineering	2.3125
12579189	Ally	CHE	Chemical Engineering	2.82142857142857
12582389	Ali	CptS	Computer Science	3.5
12582689	Nick	CHE	Chemical Engineering	3.9
12582989	Amir	CHE	Chemical Engineering	2.95833333333333
12583289	Tom	CHE	Chemical Engineering	3
12583589	Alice	CptS	Computer Science	2.16666666666667
12584189	Macy	CE	Civil Engineering	2.95
12584489	Aaron	ME	Mechanical Engineering	3.23684210526316
12584789	Jack	CptS	Computer Science	3.16071428571429

(12 rows)

Extra Credit: (5pts)

Find the students who took all the courses required by his/her track in his/her major.

sid	sname	studentmajor
12584789	Jack	CptS

(1 row)

Submission Instructions:

HW4 will be submitted online on Blackboard.

- Please include all your SQL queries in a text file (in order) and save it as HW4 . sql
- Please include the question numbers as comments. Also, include your name and the list of the students you collaborated with in the beginning of the file.
- Before you submit, make sure that the complete file can be run on the command line with the command. Make sure to include ; at the end of each query.
psql -U postgres -d hw4 < HW4.sql
- Submit your HW4 . sql file to "Homework-4 -Dropbox" under Homeworks on Blackboard.

Appendix-1

Running an SQL Script file on PostgreSQL Command Line:

Start a console window (on Windows: run cmd), and browse to the directory where the .sql script file is located.

Run the following in command line :

```
psql -U postgres -d hw4 < your_script_file.sql
```

If the database hw4 doesn't exist, you need to create it first.

If you would be running PostgreSQL client with another username (other than postgres), replace postgres with that username. You will be asked to enter your password for the username you specify.

How to run the script and create the output file

Let's assume you have created your sql script file and saved it as `your_script_file.sql`.

If you would like to save query results into a file, then run a statement like the following in the command line. This would save the query results into `your_output_file.txt`:

```
psql -U postgres -d db_name < your_script_file.sql >  
your_output_file.txt
```