

Assignment 4

Due February 21, 2018 by 11:45pm

In this assignment, you will be required to use PostgreSQL. Your solutions should include the PostgreSQL statements for solving the problems, as well as the results of running these statements. Turn in a file .sql with your solutions (where necessary, include comments explaining your solutions).

Consider a database with the following relations:

Student(sid int, sname text, major text)	sid is primary key
Course(cno int, cname text, total int, max int)	cno is primary key total is the number of students enrolled in course cno max is the maximum permitted enrollment for course cno
Prerequisite(cno int, prereq int)	meaning: course cno has as a prerequisite course prereq cno is foreign key referencing Course prereq is foreign key referencing Course
HasTaken(sid int,cno int)	meaning: student sid has taken course cno in the past sid foreign key referencing Student cno foreign key referencing Course
Enroll(sid int,cno int)	meaning: student sid is currently enrolled in course cno sid foreign key referencing Student cno foreign key referencing Course
Waitlist(sid int,cno int, position int)	meaning: student sid is on the waitlist to enroll in cno, where pos is the relative position of student sid on the waitlist to enroll in course cno sid foreign key referencing Student cno foreign key referencing Course

1. Notice the primary and foreign key constraints listed for these various relations. Your task is to write insert and delete triggers on each these relations such that these constraints remain satisfied when insertions and deletions are applied to these relations.

Implement delete triggers on the Student and Course relations that have cascading deletion effects on the relations that hold foreign keys that reference the primary keys of these relations.

2. Furthermore (1) inserts and deletes in the Enroll relation and (2) insert and deletes in the Waitlist are governed by the following constraints:
 - A student can only enroll in a course if he or she has taken all the prerequisites for that course. If the enrollment succeeds, the total enrollment for that course needs to be incremented by 1.
 - A student can only enroll in a course if his or her enrollment does not exceed the maximum enrollment for that course. However, the student must then be placed at the next available position on the waitlist for that course.

- A student can drop a course, either by removing him or herself from the Waitlist relation or by dropping the course directly from the Enroll relation. When the latter happens and if there are students on the waitlist for that course, then the student who is at the first position for that course gets enrolled in that course and removed from the waitlist. If there are no students on the waitlist, then the total enrollment for that course needs to decrease by 1.

3. Use triggers to maintain a “view”

Major(major text, numberOfStudents int)

which maintains for each major the number of students with that major who are enrolled in some course. (A major will only appear in this relation if there is at least one student with that major who is enrolled in some course.)