CS315: Introduction to Database Systems

Assignment #1

Due: 24 Jan. 2013, 18:00 HRS 13 Jan. 2013
Max Marks: 175

This assignment is on creating, modifying and querying a database system for academic administration - a kind of mini-OARS. You are required to implement the descriptions given below in PostgreSQL. You have to create, modify and query tables. The assignment will be graded by the TAs by asking you to do a demo on data provided by them.

To test, for each table create a file of comma separated values (CSV) - one line for each row in a table. See the PostgreSQL documentation on how to load CSV files into tables.

The following entities with the attributes shown are present in the system. You must choose the relevant domain type and constraint for each attribute based on your knowledge of the system as it operates at IITK.

`Student: roll no., name, hall no., room no, gender, program (BTech, MTech, MDes, MBA, MSc, PhD), dept.`

`Course: course no., title, hrs of lectures, hrs of tutorials, hrs of labs, units, syllabus, pre-reqs.`

`Faculty: emp no., name, gender, off bldg, room no, dept.`

`Course offering: course no, semester, acad year.`

`Instructor: emp no, course offering.`

`Registration: roll no., course offering, regn type (repeat, substitute, normal), course type (compulsory, dept elective, open elective, hss elective, science elective).`

`Time table: course offering, meeting type (lect|tut|lab), start time, duration, location.`

`Grade Points: letter grade, points`

`Grade: registration, grade`

1. In the first part design and create tables for the above description of mini-OARS. Pay particular attention to domain definitions and constraints for the columns and table constraints - especially key constraints. Where necessary add extra attribute(s) so that a table has a key. Also, if necessary appropriately break up an entity into multiple tables or merge entities into a single table to get a proper table design. Your choice of keys should be based on what you know about the application as it operates at IITK.

   [100]

2. In this part alter the tables designed in part 1 to model the new situations given below:

   (a) `Roll no` is not a candidate key. That is the same student has multiple roll nos simultaneously.

   (b) `Course no` is not a candidate key. Same course has multiple course nos.

(c) An instructor is not an IITK employee (could be an adjunct faculty member or a senior PhD student).

(d) All attributes must have atomic values. Situations that you have to consider (amongst others) are multiple pre-reqs, multiple instructors for a course.

[5,5,5,10=25]

3. Write SQL queries for each of the following:

(a) For each course output the roll nos and names of students registered in the course.

(b) How many (only a single number as output) courses that have a registration of less than 10 students.

(c) List all the time table clashes for students and instructors. Output the course no, student/instructor name and clash data first for all student clashes and then instructor clashes.

(d) List all students (roll no and name) and courses (course no and name) where the the student has an 8am class in some course for which (s)he is registered.

(e) Find all faculty who are not teaching any course.

(f) Calculate the SPI of all students in CSE.

(g) Find the class room or location with the maximum occupancy (that is number of hours/week).

(h) Design and implement 3 queries that you think will be useful and where each query involves at least 3 tables.

[10x5=50]