

# CS 450 - Database Concepts

Spring 2012

Instructor: Dr. Jessica Lin

## Homework 3 – Due March 21 at 4:35pm, both in class and electronically

Consider the following schema for a discussion forum like Piazza. Note this is a modified version from the one given in HW1. In HW1, the courses do not have unique identifiers. For simplicity, in this assignment each course is given a unique identifier. Also, login names are used instead of email addresses.

For each relation, the attribute(s) of the primary key is(are) underlined. The foreign key constraints are also specified.

**Users(Login: string, First\_name: string, Last\_name: string, Age: integer, Member\_since: date)**

**Schools(SID: string, Sname: string, City: string, State: string, Zip: integer)**

**Courses(CID: string, SID: string, Code: string, Number: string, Sec: string, Semester: string, Title: string, Status: string)**

Foreign Key: SID References Schools(SID)

**Enroll\_in(Login: string, CID: string, Role: string, Notification\_Preference: string)**

Foreign Key: Login References Users(Login)

Foreign Key: CID References Courses(CID)

**Posts(PID: string, Posted\_By: string, CID: string, Content: string, Post\_Time: date, Original\_PID: string)**

Foreign Key: Posted\_By References Users(Login)

Foreign Key: CID References Courses(CID)

Foreign Key: Original\_PID References Posts(PID)

*(Note the self-referencing foreign key in the last relation.)*

Write the following queries in SQL. Note “dsmith” is a unique user ID.

1. Print the number of posts for this class (Code = “CS”, Number = “450”, Section = “001”, Semester = “Spring12”, SID = “GMU”) that are made by the instructor.
2. Print the full names of all users who are enrolled as both an Instructor (Role = “Instructor”) and as a Student (Role = “Student”), in different classes, of course. Use Intersect for this query.

3. Find the full name of the youngest *active* user registered on this site (i.e. the user must currently be enrolled in an active course), and his/her age
4. Print the CIDs of courses that have posts but none of them made by the instructor.
5. Print the number of posts made for *this class* that have no reply.
6. A new post has just been made for *this class*. In order to determine whom to send the notification to, list the names of members for this class who set their notification preference to “Real-time”.

For the following questions, you can consider the example posts hierarchy and the corresponding relation instance provided below. **However, your query should be general and not be based on this (or any) relation instance.**

```

P01 (dsmith)
  |___ P03 (erin)
    |___ P04 (jane)
  |___ P05 (chris)
    |___ P08 (dsmith)
      |___ P09 (chris)

```

```

P02 (dsmith)
  |___ P06 (erin)

```

```

P07 (chris)

```

PID	Posted by	CID	Content	Post Time	Original PID
P01	dsmith	C001	XXX	2/17/12	Null
P02	dsmith	C001	XXX	2/18/12	Null
P03	Erin	C001	XXX	2/18/12	P01
P04	Jane	C001	XXX	2/19/12	P03
P05	Chris	C001	XXX	2/19/12	P01
P06	Erin	C001	XXX	2/19/12	P02
P07	Chris	C001	XXX	2/19/12	Null
P08	dsmith	C001	XXX	2/19/12	P05
P09	Chris	C001	XXX	2/20/12	P08

7. For posts made by *dsmith*, print the full names of users that posted follow-ups on those posts. In this example, *dsmith* has 3 posts: P01, P02, and P08. P01 has two direct follow-ups (P03 and P05); P02 has one direct follow-up (P06); and P08 has one direct follow-up (P09). Your query should print the full names of users who posted P03, P05, P06, and P09, respectively.

8. Print the PIDs of all original posts (i.e. not a follow-up) made for *this class*. In this example, the original posts are P01, P02, and P07.
9. Print the PIDs of all *original* posts made for *this class* that have been followed up on (i.e. have at least one reply). In this example, the query should print P01 and P02.
10. Print the full name(s) of user(s) in *this class*, if any, who replied to every single *original* post made for this class.

**Database:** A testbed will be provided on Blackboard. You may use this testbed to **test** your queries. You may want to add more tuples to the tables to test your queries. Save the file and run the script to install the test database. Your queries, however, must work in general on any valid database on the above schemas. For data types of the attributes and the domain constraints, consult the given database.

#### **Grading of SQL queries:**

- Your script file will be graded only if it executes as one SQLPLUS script file without syntax errors or other similar problem. **Make sure your script runs without errors, otherwise you will not receive any credit for this assignment.**
- Your script file will be graded by running it on different databases obtained by inserting and/or deleting some rows into/from the given database. The correctness of the query answers is the basis for grades

#### **Instructions for Submission:**

- Submit all your SQL queries electronically in one script file. Bring a hardcopy to class.
  - The file must execute without problems as one SQLPLUS script file.
  - Put your name, Student Id, Course-Section and Instructor's name as the first several lines of the file. Be sure to add the comment symbol in the beginning of the line.
  - Please name your script file in this way: Initial letter of your first name + your last name + ".sql". For example, if your name is John Smith, your script file should be JSmith.sql.