CIS 560 Database System Concepts

Spring 2007

Homework 10 of 10: Machine Problem (MP10) / Project Part 3 of 3

Aggregation Queries

Assigned: Sat 28 Apr 2007
Due: Fri 04 May 2007 (before midnight)

The purpose of this assignment is to exercise your basic understanding of aggregation in SQL.

This homework assignment is worth a total of 20 points and the project design component (combined over any of MP6, 8, and 10 that you turn in) is worth 60 points.
Use your KSOL drop box to turn in a.zip file MP10-XYZ.zip, with your initials in place of XYZ.

Use C++ or Java **only** to solve the problem. Specify which you are using in a README.txt file and name the programs accordingly: each problem should have a source code file mp10_i.[LANGUAGE]. Put the compile line in your README.txt file. Indicate on the first line of the README.txt file whether you are using ODBC or JDBC. The file names should be given on the command line, e.g., "mp10_i file1 file2". The file format consists of a list of attributes for the relation, each beginning with @, then zero or more rows.

Write the following five queries and tie each to a radio button or drop-down menu entry.

1. **(25%) Aggregation.** Return either of the following counts:

   - (AMMDB) Times that specimens of a particular Monster species have been Killed-By a Player.
   - (GradMiner) Number of courses of a particular description that have been taken by a student. (For example, a student may have taken CS373 and CS473 at UIUC in satisfaction of an Algorithms requirement.)

2. **(25%) Selection/Aggregation.** Return the count of:

   - (AMMDB) Monsters encountered that occur at a **particular** depth. You may use a JSP pop-up dialog or an edit box with a “view monsters at this depth” box that displays all monster tuples from the Monster table for a specified depth, plus the county.
   - (GradMiner) Courses a student took that have a **particular** department and course number. You may use a JSP pop-up dialog or a pair of combo boxes with a “view courses from this department, at this level” box that displays all tuples from the KSU-Course table for a specified depth, plus the county.

3. **(25%) Aggregation with Multistage Join.** Return the number of:

   - (AMMDB) Monsters encountered out of depth (at a dungeon level lower than where they normally occur). Calculate this attribute and put it into the Encountered table when each encounter tuple is saved **or** when the table is loaded.
   - (GradMiner) Deficiency courses assigned to a student for which they have not had the prerequisite (by title). Calculate this attribute and put it into the Deficiency table when each deficiency tuple is saved **or** when the table is loaded.

4. **(25%) Aggregation with Multistage Join and Condition.** Return the number of

   - (AMMDB) Times a particular Monster has occurred out of depth but been killed by a particular Player.
   - (GradMiner) Times **any** KSU-Course has had prerequisites that a particular Student has not taken.

Extra Credit. Do either of the following for 25% extra credit.

(AMMDB) Average number of times a monster occurring out of depth has been killed by the player, versus the number of times it has killed the player.

(GradMiner) Total number of times that any student has gone on probation having taken a particular graduate-level class without the prerequisite(s).

Class participation. Post any unclear points about ACID properties or concurrency to the class web group before the due date of the assignment.

The Road Ahead

- Homework 10 (this assignment) will wrap up your project with aggregation queries.