COMP 3380 – Databases: Concepts and Usage

Department of Computer Science
The University of Manitoba
Fall 2006

Assignment 3 (Part B)

Course Number: COMP 338 (Section A01)
Instructor: Dr. Carson K. Leung **Due Date: Thu, Nov 30, 2006**
Hand-in: In class (at 11:30 in EITC E2-105)

Instructions:
- Submit all answers on 8.5" x 11" paper in a letter-size folder.
- Clearly indicate your name, student number, and section on the outside of the folder.
- Include a *signed* Honesty Declaration as the *first page* of your assignment.
- Optional: You may check your answers using HSQLDB or (other available systems).

Questions:

2. Write each of the eight queries from Assignment 3 (Part A) in **SQL** for the Students-Courses-Professors database. In other words, write each of the following queries in ***SQL*** for the Students-Courses-Professors database created for Assignment 2:
   - Students (studentID, studentName)
   - Courses (courseID, courseName)
   - Profs (profID, profName, profOffice)
   - Section (courseID, sectNum, profID)
     where foreign key (courseID) references Courses on delete cascade,
     foreign key (profID) references Profs
   - Enrolled (studentID, courseID, sectNum, grade)
     where foreign key (studentID) references Students,
     foreign key (courseID, sectNum) references Section

   a) Display the student ID of each student enrolled in 'COMP 3380'.

   b) Display the result of performing a Cartesian-product of Students and Enrolled.

   c) Display the student name, course name, and grade for each student who received an 'A+' in the corresponding course (i.e., display all courses in which a specific student received an 'A+').

   d) Display the name of each professor who teaches a course.

   e) Display the names of all professors who do not teach any courses.

   f) Display the names of all professors who teach *all* the courses.

   g) Display the student IDs of all students who are not enrolled in 'COMP 4380'.

   h) Display the student IDs of all students who are enrolled in 'COMP 3380' but not in 'COMP 4380'.

3. Use the following data (which is not normalized) to populate the Students-Courses-Professors database (using an appropriate collection of `INSERT` statements). Hand in (a) a printed copy of the insert statements and (b) a printed copy of each relation (use the `SELECT * FROM` *`relationName`* statement to display the contents of each relation). Also, write an SQL query to display the following table containing information from different relations.

STUDENTID	STUDENTNAME	COURSEID	COURSENAME	SECTNUM	GRADE	PROFID	PROFNAME	PROFOFFICE
1000000	Joe Dirt	COMP 3020	HCI	A01	B	1001	Joe Sql	E2-400 EITC
1000000	Joe Dirt	COMP 3350	SE	A01	A	1002	Joe Cool	100 Armes
1000000	Joe Dirt	COMP 3380	DB	A01	B+	1003	Joe Jock	500 Frank Kennedy
2000000	Clint Eastwood	COMP 3020	HCI	A02	A+	1001	Joe Sql	E2-400 EITC
2000000	Clint Eastwood	COMP 3350	SE	A02	A	1002	Joe Cool	100 Armes
2000000	Clint Eastwood	COMP 3380	DB	A02	A+	1001	Joe Sql	E2-400 EITC
3000000	Neo Matrix	COMP 3020	HCI	A03	B	1002	Joe Cool	100 Armes
3000000	Neo Matrix	COMP 3380	DB	A01	A	1003	Joe Jock	500 Frank Kennedy

4. Use the ***Java program*** provided for Assignment 1 (named DBTest.java) to perform the following operations:

   a) Ensure that appropriate statements have been added (in Question 3) to incorporate information about *Section* and *Professors*.
   Hint: *Check the sample solutions for Assignment 2*.

   b) Display the student ID of each student enrolled in 'COMP 3380'.

   c) Delete Student 1000000. Then display all students, sections, and enrolments.
   Note: Student 1000000 is enrolled in some courses. So, you will violate integrity constraints when you try to delete this student. Alternatively, you could *explicitly* drop the courses this student is enrolled in and delete him.

   ~~d) Delete Prof 'Joe Sql'. Then display all professors, sections, and enrolments.~~

   e) Insert Student 1500000, Mickey Mouse, who received a grade of C in Section A01 of 'COMP 3020'. Then display all students, courses, professors, sections, and enrolments.

   f) At the end of your processing, rollback the transaction so that the database is restored to its original state.
   Hint: *Use the* `ROLLBACK` *command*.

5. Write each of the following view in **SQL** for the Students-Courses-Professors database.

   a) Create a view that contains course and section information, including the number of students enrolled in each section, in one relation. Display the contents of the view.

   After the view has been created, print the contents of the view using a `SELECT *` statement.

--- End of Assignment 3 (Part B) ---