

## CS 434 – Project

### University Student Database

Consider the data that are usually maintained by a typical university concerning students, courses, and enrollments. Students are admitted to the university, and they pursue a degree program in a particular department. The university catalog consists of courses that are offered every term. Students choose courses to take and enroll in them during registration. Instructors are assigned courses to teach, and they in turn assign grades. A possible database design results in the following relational tables:

COURSES (Cno, Ctitle, Hours, DeptId)  
DEPARTMENTS (DeptId, DeptName, College)  
INSTRUCTORS (LastName, FirstName, DeptId, Office, Phone, Email)  
SECTIONS (Term, LineNo, Cno, InstrLname, InstrFname, Room, Days, StartTime, EndTime, Capacity)  
STUDENTS (Sid, LastName, FirstName, Class, Phone, Street, City, state, Zip, Degree, DeptId, hours, Gpa)  
ENROLLMENTS (Sid, Term, LineNo, Grade)

Note: the primary key attributes are underlined. Foreign keys (Referential Integrity Constraints) are to be determined.

The COURSES table maintains the list of courses in the university catalog. Information about departments, instructors, and students is maintained in the DEPARTMENTS, INSTRUCTORS, and STUDENTS tables, respectively. Notice that some of the columns in the STUDENTS table are computed columns (gpa, hours) – i.e., their values are determined by other values in other tables. The SECTIONS table maintains the information about the schedule of classes for each term. The ENROLLMENT table keeps information about the enrollment of students in sections.

The following is an outline of a possible application program for the student database:

#### MAIN MENU

- (1) Students functions
- (2) Administrative functions
- (3) Reporting functions
- (4) Quit

#### STUDENT FUNCTIONS MENU

- (1) Register for courses
- (2) Add/Drop a course
- (3) Request Transcript
- (4) Compute GPA
- (5) Quit

#### ADMINISTRATIVE FUNCTIONS MENU

- (1) Create a new course/drop course
- (2) Prepare term schedule (add sections)
- (3) Add/drop instructors
- (4) Alter term schedule (add/drop/update sections)
- (5) Add/drop students
- (6) Print the list of instructors and the number of courses each of them is teaching (for a term)
- (7) Quit

#### REPORTING FUNCTIONS MENU

- (1) Print schedule of classes (for a term)
- (2) Print the catalog
- (3) Print the honors list of students for a department (e.g. students whose GPA > 3.8)
- (4) Quit

You have to implement the database system above. You can add additional functionality if you want and it will be appreciated. In particular you can add some stored procedures that are called from the application program.

There are three phases in the project.

### **Phase 1.**

Write SQL commands for creating tables including primary keys, secondary keys and foreign keys. Specify meaningful referential triggered actions for foreign key constraints both for deletions and updates.

Write SQL command files that populate each table. Each table is required to have a sufficient number of tuples. Keep in mind the relationships between tables, and plan your data accordingly. You should be able to run these command files successfully with no errors and no integrity violations.

#### **Deliverable 1**

1. Description of implementation, problems faced.
2. The SQL commands that create your tables
3. The SQL commands that populate your tables

### **Phase 2.**

Define and implement as check constraints five integrity constraints that should hold on the schema above (e.g. a student cannot enroll in more than 10 courses in a semester).

Write SQL triggers that correctly maintain the values of the computed attributes gpa and hours in table STUDENTS when the values of other attributes (that affect the value of attributes gpa and hours) are changed. Consider any type of data change: insertions, deletions, and updates.

#### **Deliverable 2**

1. Description of implementation, problems faced.
2. The SQL statements that define the constraints
3. The triggers

### **Phase 3.**

The application program for the student database that implements the functions described in the menus above.

#### **Deliverable 3**

1. Description of implementation, problems faced.
2. Users guide (Less than 3 pages)
3. Deliverable 1
4. Deliverable 2

5. The source code.
6. A printout of the use of the program.

*You should demonstrate your program at the end of the semester. Please send an email to the TA of the course to fix an appointment for the demonstration. Check the dates on the web page of the course. If you miss a task mention which one. In this case if you have implemented a task with a similar functionality it helps (mention it).*