_____

FINAL PROJECT
UNIVERSITY DATABASE

Introduction

Each student has to complete this project individually. You are going to use MySQL to perform some queries against a database. The schema and sample data of the database are provided.

The goal of this project is to provide a realistic experience in the implementation, operation, and maintenance of a relational database. You will design a relational database schema and implement it on MySQL. Then, you will load some data into your database, create some queries, and update transactions.

The UNIVERSITY Database

You have been approached by a university for the design and implementation of a relational database system that will provide information on the courses it offers, the academic departments that run the courses, the academic staff, and the enrolled students. The system will be used mainly by the students and the academic staff.

Assume that the following requirements were collected for this application:

- **Department:** The university is organized into departments. Each department is identified by a unique identification number (department ID), has a name, and has a budget.
- **Course:** Each department runs several courses. Each course is associated with a course ID, title, credits, department ID, and may also have associated prerequisites.
- **Faculty:** Faculties are identified by their unique ID. Each faculty has a name, an associated department (department ID), room ID, and salary.
- **Student:** Students are identified by their unique ID. Each student has a name, address, birthdate, phone, student type, and a major ID.
- **Term:** The university maintains a list of terms, specifying the term ID, term description, start date and end date.
- **Section:** The university maintains a list of all classes (sections) taught. Each section is identified by a section ID, course ID, section number, room ID, start time, end time, day, maximum count, and term ID. A section can be used in one course only but can be studied by many students. The department has a list of teaching assignments specifying the sections each faculty is teaching.
- **Registration:** The university has a list of all student course registrations specifying the associated sections each student has taken (or registered for), midterm grade, and final grade.
- **Location:** The university maintains a list of classrooms which specifies the room ID, building, room number, capacity, room type, and room description.
- **Major:** The university has a list of all majors. Each major is identified by a unique number (major ID) and a description.

_____

Logical Design

The full set of normalized tables for the University Database is as follows:

Schema

Department {dept_id, dept_name, budget}
Course {course_id, title, credits, dept_id, prerequisite}
Faculty {faculty_id, name, room_id, dept_id, salary}
Student {student_id, last_name , first_name, street, city, state, zip, birth_date, major_id, phone, student_type}
Term {term_id , term_desc , start_date , end_date}
Section (section_id , course_id , section_number , term_id , faculty_id , day , max_count , start_time , end_time, room_id}
Registration (student_id , section_id , midterm_grade, final_grade}
Major {major_id , major_desc}
Location {room_id , building, room_no, capacity, room_type , room_description}

PART 1: Implement the Database [50 points]

1. Create the tables for the university database using MySQL DBMS. To your report, you will add a list of the CREATE TABLE statements for the university database. Specify as many constraints (key, referential integrity) as you can in the relational schema. Choose appropriate data types for each attribute.
    a. Define the primary key, foreign key, NOT NULL, CHECK and UNIQUE constraints in the CREATE TABLE statement. If not possible, use ALTER TABLE statement to add a constraint.
    b. If a table does not have a foreign key, leave the entry blank. (Note: Some tables have a composite primary key. Identify all composite key attributes for such tables.)

Table	Primary key	Foreign key	Table Referenced

2. Load the records provided to you into each of the tables that you have created. Your data should be kept in a file so that it can be easily reloaded.

_____

PART 2: SQL Queries [50 points]

You should run a number of test queries to see that you have loaded your database in the way you intended. The second part of the project is to apply certain update transactions and retrieval queries.

Write SQL queries for the following and execute them.

1. Display courses that have the course ID **EN100** as a prerequisite.
   This statement should return these columns: course_id, title, credits, dept_id, and prereq.

2. Display all course sections offered in Winter 2013. This statement should return these columns: course_id, section_id, and term_id.

3. Modify the title of course CIS253 from **Database Systems** to **Introduction to Relational Databases**.

4. Display the number of enrollments for section **1103** of Course ID **MA150.** This statement should return these columns: section_id, course_id and the number of enrollments.

5. Display the student roster (use the format: <last name>, <first name> in a single column) for each section taught by Professor **Sen**. Identify the section using course ID and section ID. This statement should return these columns: course_id , section_id , last_name, and first_name.

6. Display a list of all departments, with the total number of instructors in each department, without using subqueries. This statement should return these columns: dept_name and the total number of instructors.

7. Display names of faculty members who work in department 4 or 5. This statement should return these columns: faculty_id, name, and department_id.

8. Display Winter 2013 course sections with the faculty member assigned to teach the class. This statement should return these columns: course_id, section_id, and name.

9. Display a student's full name along with his/her major's description.

10. Display names of students, who received an 'A' as their final grade in winter 2013.

11. Display all information about instructors from the **Computer Science** department.

12. Display the total enrollment for course **CIS253** in a column named TOTAL ENROLLED.

13. Display course title, total capacity and number of sections in each course, where there is more than one section.

14. Create a view that lists all course sections offered by the **Accounting** department in the Winter 2013.

15. Show all the sections whose registration is greater than two students. This statement should return these columns: course_id, section_id, and the number of enrollments.

_____

Submission:

- Documentation describing any assumptions you have made
- One or more SQL/Spool files of the SQL part of the project, including creating the tables and the query results. You will need to label your project with your first initial, last name, and the name of the project.
- Zip the files to upload to Canvas (yourname_project_part1.zip and yourname_project_part2.zip).
- The files that you submit should be sent via Canvas latest by 11:59 pm of the due date.

Due Dates:

- Due Date for Part 1: **Monday, November 21st, 2016.**
- Due Date for Part 2: **Monday, December 5th, 2016.**

Grading

The grade for this project will be broken down as follows:

- Part 1 – Create the Database / Populate Relations (50 points)
- Part 2 – Query the Database (50 points)

Important Note:

Plagiarism is not permitted and will result in a grade of zero.

Late policy:

For Part 1, there will be a 5% deduction for each day that it is late. Part 2 will not be accepted late.