
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.

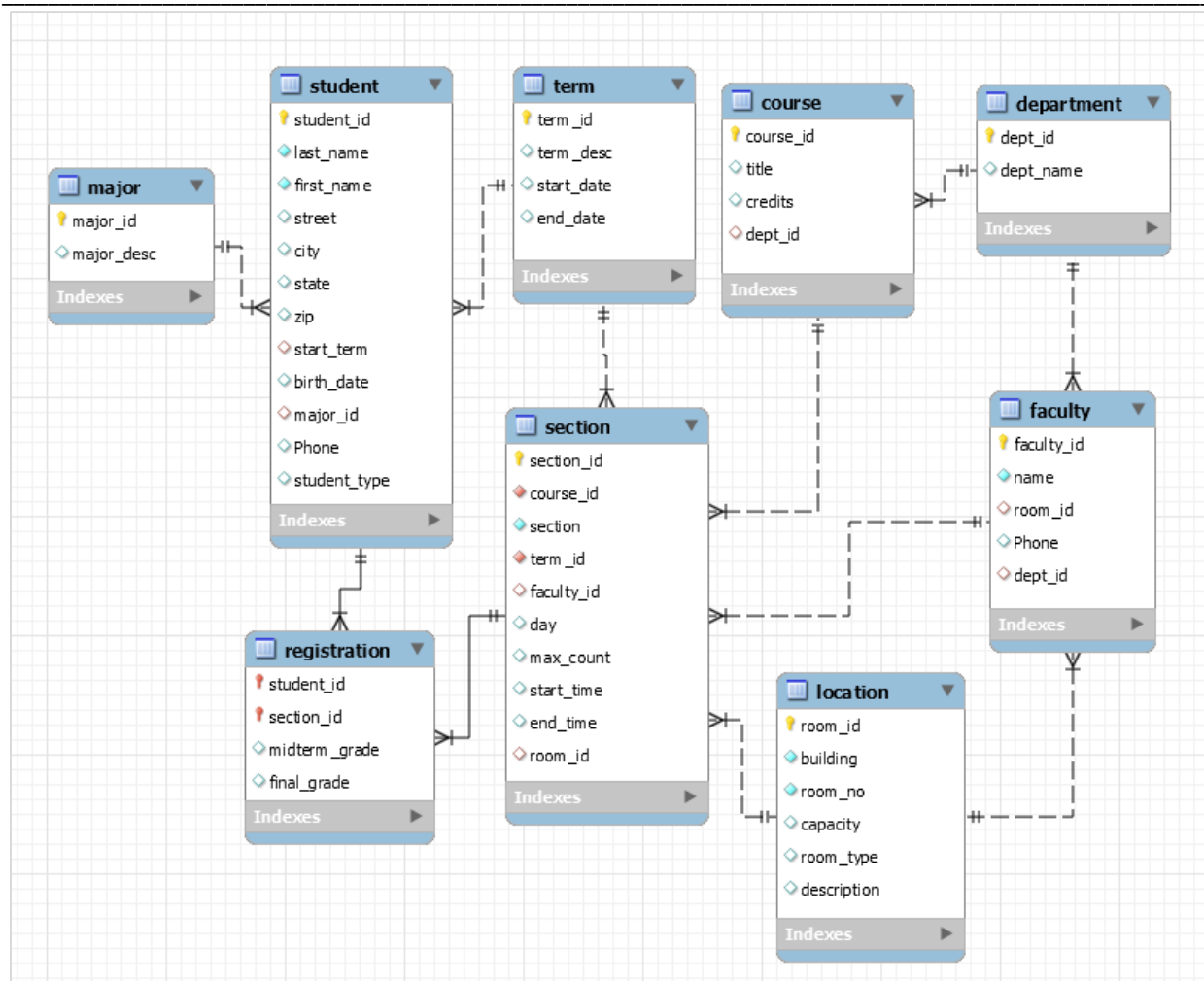
The goal of this project is to provide a realistic experience in the implementation, operation, and maintenance of a relational database. You will design a relational database schema and implement it on MySQL. Then, you will load some data into your database, create some queries, and update transactions.

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 (dept. ID) and has a name.
- **Course:** Each department runs a number of courses. Each course is associated with a course ID, title, credits, and department ID.
- **Faculty:** Faculties are identified by their unique ID. Each faculty has a name, an associated department (dept. ID), room ID, and phone number.
- **Student:** Students are identified by their unique ID. Each student has a name, address, birthdate, phone, an associated start term, 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 provides a set of sections used in different courses. Each course uses a number of sections. 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.



ER schema diagram for the University Database

Logical Design (map ER to Relational)

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

Schema

Department {dept_id, dept_name}

Course {course_id, title, credits}

Faculty {faculty_id, name, room_id, phone, dept_id}

Student {student_id, last_name, first_name, street, city, state, zip, start_term, birth_date, major_id, Phone, student_type}

Term {term_id, term_desc, start_date, end_date}

Section (section_id, course_id, section, 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. Capture your commands in a spool file to turn in.

1. Display names of faculty members who work in department 1 or 2.
2. Display Spring 2013 course sections with the faculty member assigned to teach the class. Display course ID, section ID, and faculty name. Use a join using JOIN...ON syntax.
3. Display a student's full name along with his/her major's description.
4. Display names of students, who received an 'F' as their final grade in winter 2013. Use a join using JOIN...ON syntax.
5. Display the titles of all courses taught by Professor Sen.
6. Find all New York and New Jersey students. Display student ID, last name, first name, and state.
7. For each course ID, display the maximum count in descending order.
8. Display all instructors and how many sections they teach.
9. Show the total enrollment for course CIS253 in a column named TOTAL ENROLLED.
10. Display course title, total capacity and number of sections in each course, where there is more than 1 section.
11. Create a view CSinstructors, showing all information about instructors from the Computer Science department.
12. Find students who are born in the month of October.

-
13. Create a new table, Sp13SECT, for Spring 2013 semester course sections using a subquery. Include Course ID, Section ID, Faculty ID and Room ID columns only.
 14. Delete rows from SP13SECT table for faculty 'Mobley'. Use a subquery
 15. Show all the sections whose registration is greater than 2. Display course and section ID. Use a join using JOIN...ON syntax.

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 Insight (yourname_project_part1.zip and yourname_project_part2.zip).
- The files that you submit should be sent via Insight latest by 11:55 pm of the due date.

Due Dates:

- Due Date for Part 1: **Wednesday, May 4th, 2016.**
- Due Date for Part 2: **Wednesday, May 18th, 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.