

---

## FINAL PROJECT

### UNIVERSITY DATABASE

#### Introduction

Each student must complete the project individually. You will utilize MySQL to perform several 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. Your task is to design a relational database schema and implement it on MySQL. Then, you will load data into your database, create queries, and update transactions.

#### The UNIVERSITY Database

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

Assume 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), name, and 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 an associated department (department ID), a name, room ID, and salary.
- **Student:** Students are identified by their unique ID. Each student has an address, a name, birthdate, phone, student type, and 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, day, maximum count, term ID, and an end time. A section can be used in only one course 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 specifying 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 [30 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 all course sections offered in Winter 2017. This statement should return these columns: `course_id`, `section_id`, and `term_id`.
2. Modify the title of course, CIS253 from **Database Systems** to **Introduction to Relational Databases**.
3. 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.
4. 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`.
5. 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.
6. Display names of faculty members who work in department 4 or 5. This statement should return these columns: `faculty_id`, `name`, and `department_id`.
7. Display Winter 2017 course sections with the faculty member assigned to teach the class. This statement should return these columns: `course_id`, `section_id`, and `name`.
8. Display names of students, who received an 'A' as their final grade in Winter 2017.
9. Display 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 one section.
11. Create a view that lists all course sections offered by the **Accounting** department in the Winter 2017.
12. 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.

---

**PART 3: Java application (University. Java) [20 points]**

Create a Java application that accesses the university database and has the following interactions with the users:

- a. Display the titles of all courses
- b. Display the list of all departments
- c. Display the section ID, course ID, and course title of all sections
- d. Display all course sections offered in Spring 2017
- e. Display names of students and their majors
- q. Quit

QUERY OPTIONS

- (a) Display the titles of all courses
- (b) Display the list of all departments
- (c) Display the section ID, course ID, and course title of all sections
- (d) Display all course sections offered in Spring 2017.
- (e) Display names of students and their majors
- (q) Quit.

Type in your option: **a**

The Titles of All Courses

-----  
Accounting  
Basic English  
College Algebra  
English Literature  
Introduction to Relational Databases

QUERY OPTIONS

- (a) Display the titles of all courses
- (b) Display the list of all departments
- (c) Display the section ID, course ID, and course title of all sections
- (d) Display all course sections offered in Spring 2017.
- (e) Display names of students and their majors
- (q) Quit.

Type in your option: **b**

The List of All Departments

-----  
Computer Science  
Telecommunications  
Accounting  
Math and Science  
Liberal Arts

QUERY OPTIONS

- (a) Display the titles of all courses
- (b) Display the list of all departments
- (c) Display the section ID, course ID, and course title of all sections
- (d) Display all course sections offered in Spring 2017.
- (e) Display names of students and their majors
- (q) Quit.

Type in your option: **c**

Section ID	Course id	Title
-----		
1104	AC101	Accounting
1311	EN100	Basic English
1103	MA150	College Algebra
1207	LA123	English Literature
1102	CIS253	Introduction to Relational Databases

## QUERY OPTIONS

- (a) Display the titles of all courses
- (b) Display the list of all departments
- (c) Display the section ID, course ID, and course title of all sections
- (d) Display all course sections offered in Spring 2017.
- (e) Display names of students and their majors
- (q) Quit.

Type in your option: **d**

Section ID	Course ID
-----	
1205	AC265
1206	AC265
1207	LA123
1208	CIS253
1209	CIS253

## QUERY OPTIONS

- (a) Display the titles of all courses
- (b) Display the list of all departments
- (c) Display the section ID, course ID, and course title of all sections
- (d) Display all course sections offered in Spring 2013.
- (e) Display names of students and their majors
- (q) Quit.

Type in your option: **e**

Last name	First name	Major
-----		
Rickles	Deborah	BS-Computer Science
Diaz	Jose	AAS-Accounting
Lee	Brian	BS-Telecommunications
Patel	Rajesh	BS-Accounting
Tyler	Mickey	BS-Computer Science

## QUERY OPTIONS

- (a) Display the titles of all courses
- (b) Display the list of all departments
- (c) Display the section ID, course ID, and course title of all sections
- (d) Display all course sections offered in Spring 2017.
- (e) Display names of students and their majors
- (q) Quit.

Type in your option: **q**

---

**Submission Instructions:**

- Documentation describing any assumptions you may have made.
  - One or more SQL/Spool files of the SQL part of the project, including creating the tables and query results. You will need to label your project with your first initial, last name, and name of the project.
  - Submit the source code for the programming part of the project.  
Upload the file: University.java
- After completing Part 3, take a screenshot of the output (a-e). Copy the photos onto a word document and submit it along with Part 3.
- Zip the files to upload to Canvas (yourname\_project\_part1.zip and yourname\_project\_part2.zip).
  - The files you submit should be sent via Canvas latest by 11:59 PM of the due date.

**Due Dates:**

- Due Date for Part 1: **Friday, March 16<sup>th</sup>, 2018.**
- Due Date for Part 2 and Part 3: **Friday, March 23<sup>rd</sup>, 2018.**

**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 (30 points)
- Part 3 – Java Application (20 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 and Part 3 will not be accepted late.