Project 1 – Database Design

Overview

This project is worth 100 points and counts 10% towards your final grade. It is due on Friday, July 20, at 5 p.m. Your submission should be a PDF document.

This project will give you hands-on practice in working with MySQL Workbench (or similar tool) to create a key-based, fully attributed, BCNF data model. In this project you will design a database, draw a data model to represent the design, then create a "physical model" of your design in the format of DDL (table create statements.)

Objectives

1. Become familiar with a data modeling tool of your choice
2. Demonstrate ability to create a complete data model.
3. Use the data modeling software to generate the DDL to create the database you have designed

Deliverables

1. A key-based, fully-attributed data model depicting your database design

2. The DDL necessary to create the database you have designed.

3. Documentation of any assumptions you made regarding unclear or missing requirements

Submission

Your results for this project assignment should be captured in a document (such as a .txt file, MS Word or similar tool.) Please then save your final deliverable document as a **PDF** for submission. If you are doing "PAIR PROGRAMMING" on this assignment, please be sure to identify the name of your "programming" partner on your submission. You must EACH submit your own final deliverable document for this homework.

The final deliverable document you submit for this project must consist of three sections:

The first section is a picture of your entire data model. The second section is text containing all DDL generated by your data modeling software tool necessary to create the database you have designed. The DDL must include create statements for all tables in your database (including definition of all data columns.) Primary and foreign keys must be defined. DDL must include all constraints, including foreign key references. DDL must include create statements for necessary secondary indexes. Third is list (bullet points) of any assumptions you found necessary to support decisions you made about the process and/or database design.

Case Study: City College Registrar's Office

Your task is to design a database for the registrar's office at City College. The database you are designing will be the "backend" for a new client/server, web-based system that is replacing an old PC-based system. Users will access the new system via the browser on their workstations.

The backend database will be a MySQL version 5.7 instance running on a central Linux database server. Application architecture will consist of web pages for the front-end presentation layer. The middle layer integrating the front-end and back-end will be programmed by the app dev teams using java script in a NodeJS + Express framework.

SCOPE:

Although the Registrar's office provides many different services, you should restrict your scope for this database design to the "registration" business process as described below. Scope includes the system's ability to create the following reports:

> A student schedule, a classroom schedule, an advisor's list of advisees, student grade reports, student transcripts, class rosters, a faculty member's teaching schedule.

The ADD/DROP process is out of scope for this project. Any "waitlist" process is out of scope for this project.

The online course catalog is in scope for this project. The database for the new system must support the online course catalog system which is used by students and advisors as students enroll in classes. The online course catalog allows students and advisors to look up courses/sections and see the day/time schedule information, classroom/building assignments, and current enrollment.

REQUIREMENTS:

During interviews with the staff and leadership team in the registrar's office, you have gathered the following requirements.

- The database must be large enough to support 3000 active students (at all levels). The database must also hold all historical information for all students dating back to 1963 when the college began. Historical data will be collected from the legacy system, converted as needed, and loaded into the database. Assume that there have been an average number of 700 new students joining the university every year throughout its history.

- Historical records include each student’s Social Security Number, which was used as an identifier for students prior to the implementation of the current PC-based system in 1997.
- Once a course is full, no more enrollments will be allowed.
- Course/Section size limits are determined by classroom capacity.
- Employee data for faculty members is brought into the registrar’s system via an interface with the PeopleSoft “Human Resources” system. Each faculty member is identified by a 5–digit EmployeeID number that migrates from PeopleSoft.
- During registration, students sign up for courses. Each student completes his/her form with assistance from their advisor. Students and advisors use the online Course Catalog to obtain information about classes, sections, building/classroom location, day/time course offerings, and enrollments.
- Completed, signed forms are then taken to the registrar’s office for input into the registration system.
- Each course is identified by an 8 character CourseID consisting of a 4-letter abbreviation representing the department followed by a 4 digit number.
- Each course carries a certain number of credit-hours ranging from 1 to 4.
- A student can sign up for one or more courses, but no student can take more than 18 credit-hours per semester.
- Most common inquiries by the staff members of the registrar’s office include:
    - Looking up student information by lastname, firstname
    - Looking up course and section enrollments by student name
    - Looking up course/sections offered by department by semester
    - Looking up course and section assignments by faculty member

- A student can choose to take off a semester and not sign up for any classes during that semester.

- Some courses offer more than one section. For example, ENGL1001 is required for all entering Freshmen. There are many sections of English 1001 taught by many different faculty members.

- There will never be more than 12 sections of any given course running at the same time.

- Different sections of a course may have the same instructor, or they may have different instructors.

- A section of a course is assigned to a classroom within a building.

- One faculty member is assigned to teach a section of a course.

- Classrooms have room numbers; buildings have names.

- A student cannot take two different sections of the same course at the same time.

- Faculty members may teach up to 3 different course sections per semester.

- Each student is assigned one faculty member to be their advisor.

- Each student's registration course selections must be reviewed and approved by their advisor or a representative from the student's department if their permanent advisor has not been assigned yet.

- Some faculty members don't advise any students.

- Students may declare a double major.

- A faculty member may take off a semester (for sabbatical) and not teach any courses during that semester.

- Semesters are identified by the year and the season (Fall, Spring, Summer)

- Students are assigned a letter grade at the end of the semester for each course they take. Numeric equivalents of letter grades are used to calculate each student's cumulative GPA.

- If a student fails a course they can take it again in a future semester and the grade for the retake overrides their original grade for cumulative GPA calculation, but the historical record of the original failing grade must be kept.

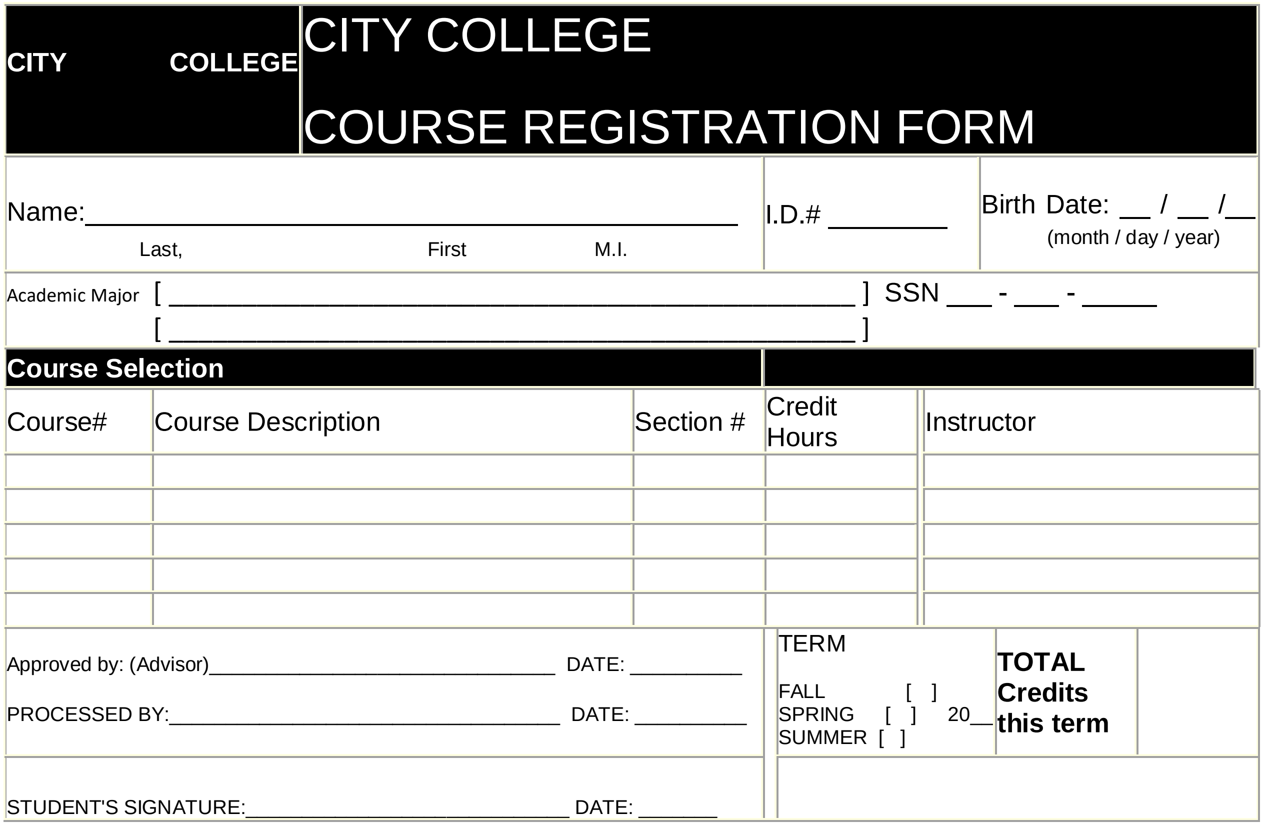
- The school operates three semesters per year: Spring (16 weeks), Summer (8 weeks), and Fall (16 weeks.)

Guidelines, Step-by-Step

1. Draw a DATA MODEL to depict your database design for this business application. In this data model be sure to include and account for the following:
    a. Identify all the ENTITIES described in this scenario.
    b. You can assume the existence of common non-key attributes based on your experience and knowledge as a student. For example, for a student you can assume that a student entity includes student name, address, phone number, etc.
    c. Define a unique identifier (primary key attribute) for each of the entities in this scenario. Surrogate keys (auto_increment) may be assigned where useful.
    d. Define the relationships between the entities described in this scenario and identify the cardinality and optionality of each relationship.
    e. Resolve any many-to-many relationships described in this scenario.
    f. Identify all non-key attributes, data type, length, and constraints.
    g. Identify all foreign keys.
    h. Identify and define any secondary indexes necessary.
2. Once the data model is complete, using the modeling tool, export the DDL necessary to create the database including all tables, keys, constraints and indexes. You should edit the DDL to remove any generated statements that you do not understand or cannot explain what the statement is for.
3. In a fictitious case study like this, the list of business requirements above cannot possibly be completely exhaustive. As you design the database, you may find it necessary to make certain assumptions about City College and its registration process. Your final deliverable for this project should include a section where you document any additional assumptions you found necessary to support decisions you made while creating your data model.
4. Feel free to email the instructor with specific questions whose answers are necessary for you to decide how to handle any specific database design issues.

Appendix A

Screen image from the current PC-based registration system. Completed registration forms are printed from this screen. Printed copies are signed by the student and the advisor and turned in to the registrar's office. The staff member completing the final enrollment then signs the paper form and files it.



CITY COLLEGE

CITY COLLEGE

COURSE REGISTRATION FORM

Name:____
Last, First M.I.

I.D.# ________

Birth Date: __ / __ /__
(month / day / year)

Academic Major [____] SSN ___ - ___ - _____
[____]

Course Selection

Course#	Course Description	Section #	Credit Hours	Instructor

Approved by: (Advisor)______________________________ DATE: __________

PROCESSED BY:__________________________________ DATE: __________

TERM

FALL [ ]
SPRING [ ] 20__
SUMMER [ ]

TOTAL Credits this term

STUDENT'S SIGNATURE:____________________________ DATE: _______