

COURSE OUTLINE UCLA EXTENSION

DEPARTMENT OF ENGINEERING, INFORMATION SYSTEMS & TECHNICAL MANAGEMENT

Course Number & Title: Part II: Introduction Structured Query language (SQL) Syntax

Course Author(s): Richard Patlan
Author(s) Affiliation: Capital Programs, UCLA
Phone: 310-206-5908
Cell: 626-221-8435
E-Mail: rpatlan@capnet.ucla.edu
rpatlan@dbwebsite.com

Project : Student Functionality Requirement

Students, Courses and Tutors Scenario

The following data model is designed to hold information relating to Students, Student Courses and Tutors who deal with these students. For this scenario we need to define the following facts:

The Entities required should include:

- Student Information
- Courses
- Student Courses
- Employees (Tutors)
- Contact Information (Contact between the Student and the Tutor by e-mail, phone, projects etc.)
- Contact Types (e-mail, phone, projects, assessments)

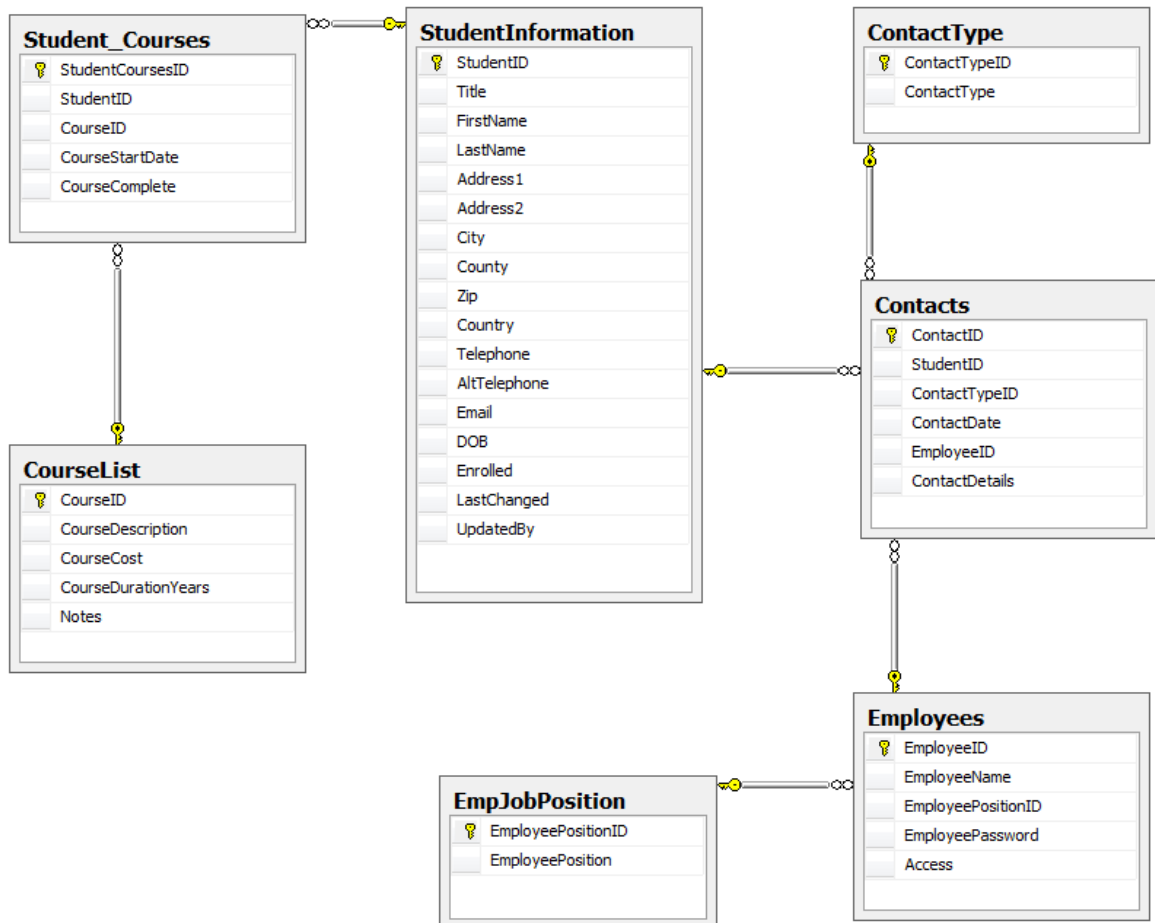
The Entities are related based on the ER diagram below. Use it to determine table relationships. For example

- One Student can enroll in or many Courses
- One Course can have one or many Students enrolled in it.
- One Student can have zero, one or many forms of contact with the Course Tutor
- One Employee (Tutor) can deal with many contacts
- One contact Type (Phone, E-mail, Assessments, Projects etc.) can have zero, one or many contacts

The design allows

- a Student to enroll in one or multiple Courses,
- a Course allowing one or more Students enrolled in it.
- a student may be in contact with the Course Tutor many times using many different forms of contact.
- a tutor will deal with many contacts involving many Students.

Use the following data model to create your tables.



ASSIGNMENT TO COMPLETE

Write the SQL code to create the following:

1. Write sql code to create all tables shown in the ER diagram. Assign appropriate datatype for each field including which should allow NULLs, Primary Key and set as Identity
2. Write sql code for each table to create a Foreign key constraint on appropriate referenced table.
3. Create INDEX on LastName column in Student Information table. Name the new INDEX "IX_LastName."
4. Write code to ALTER Student Information table to ADD new column named CreatedDateTime with datatype as datetime.
5. Write code to ALTER Student Information table to create a DEFAULT on the Country field. Set default as 'US'
6. Write code to ALTER Student Information table to DROP AltTelephone column.
7. Write sql code to create three stored procedures on the Student Information table:
 - a. Insert single student record stored procedure. Insert three sample student records.
 - b. Update Country with 'Bruin Nation' in single student record. Use StudentID in WHERE clause to update only one record.
 - c. Delete single student record. Use StudentID in WHERE clause
8. Write sql code to create a VIEW based on the Student Information table. Only return First Name and Last Name columns. Name the view DisplayName
9. Write sql code to create a VIEW based on the Student Information table. Concatenate First Name, Last Name and Title to create new virtual column with alias name of "StudentDisplayName."
10. Create a trigger on the Student Information table to set Country field to UPPER case when new record added.

When submitting completed project just submit the SQL code in a word document or as a sql text file.