

Course Syllabus

Course Title

Database Design and Prototyping

Semester

Spring 2020

Course Number

IS455 – AGO, AUO

Instructor

Kevin Trainor

Class Times

Online meetings will be held on Tuesdays from 6:30 PM till 9:00 PM using Zoom. I look forward to joining you there. Please use a headset.

Online Lab Session

On Sunday mornings from 10:00 AM till 11:00 AM, I will be holding an optional Online Lab Session using Zoom. This will be a time when you can ask questions, discuss ideas, get help with your coding assignments, or get help with your Final Project. I will work with students on a first come, first served basis. If you need to arrange a meeting with me on a different day or at a different time, please send me an *Individual Meeting Request* via the Service Desk (see *Contacting Instructor* below).

Contacting Instructor

The preferred method for contacting me is by entering a request using the [Service Desk for this course](#). PLEASE, DO NOT send requests to my regular email address.

The Service Desk for this course has been implemented using the JIRA Service Desk product. If you are new to using the service desk in one of my courses, please visit the [iCourse JIRA Service Desk Introduction](#) for instructions and tips.

On an emergency basis, you may contact me using my mobile phone number: 847-650-9706.

Catalog Description

The course provides students with both theoretical and practical training in good database design. By the end of the course students will create a conceptual data model using entity-relationship diagrams, understand the importance of referential integrity and how to enforce data integrity constraints when creating a database. Students will be proficient in writing basic queries in the structured query language (SQL) and have a general understanding of relational database theory including normalization.

Course Topics

- An introduction to relational databases and SQL
- How to use MySQL Workbench and other development tools
- How to retrieve data from a single table
- How to retrieve data from two or more tables
- How to code summary queries
- How to code subqueries
- How to insert, update, and delete data
- How to work with data types
- How to use functions
- Introduction to Data Modeling
- How to design a database
- How to create databases, tables, and indexes
- How to create views
- Using MySQL Workbench to back up and restore a database
- Conceptual Data Modeling
- Logical Database Design Using Normalization

Learning Outcomes

After completing this course, you should be able to:

- Identify the advantages of relational database management systems over conventional files and spreadsheets.
- Code SQL queries that extract data from a single table.
- Code SQL queries that extract data from two or more tables.
- Code SQL summary queries.
- Code SQL subqueries.
- Code SQL statements that implement insert, update, and delete functions.
- Identify the standard data types available in relational databases and their typical use.
- Code SQL functions.
- Create databases, tables, and indexes using the Data Definition Language (DDL).
- Create views.
- Backup and restore a database using MySQL Workbench
- Create a non-trivial conceptual data model.
- Create a non-trivial logical database design.

Required Texts

Murach, J. (2019). *Murach's MySQL (3rd Edition)*. Fresno, CA: Mike Murach & Associates, Inc. ISBN 978-1-943872-36-7. Digital edition available directly from the publisher at <https://www.murach.com/shop/murach-s-mysql-3rd-edition-detail>

Oppel, Andrew J. (2010) *Data Modeling: A Beginner's Guide* New York : McGraw-Hill, ISBN-10: 0071623981, ISBN-13: 978-0071623988

Technology Requirements

You will be completing coding assignments and conducting your final project using your own computer. I recommend that you use a computer that runs Windows 10 or a recent release of MacOS. While the software that we will be using for this course does run on Linux computers, there will be substantially less technical support available for Linux. If you want to use a Linux computer for your coursework, please contact me first.

You will be installing the following software on your computers for this class:

- MySQL Server Community Edition
- MySQL Workbench

All of the software and services that you will be using are free for your use during this class. I will publish brief instructions and detailed tutorial videos to help you download, install, and begin using the required software on computers running Windows 10 and recent releases of MacOS. Links to these resources will be provided in our Weekly Schedule.

Course Schedule

The schedule for this course will be available via our Weekly Schedule at:

https://courseinfo.ligent.net/2020sp/illinois/is455ago_auo/index.html

The course schedule is always subject to reasonable change to account for changes in circumstances and to correct errors. When I make changes to the schedule, I will announce them via our Moodle Announcements Forum. Postings to this forum should result in you being sent an email copy of the announcement as well.

Course Elements

1. Readings

Required readings will be assigned from the textbooks listed above and from other resources that will be identified in the Weekly Schedule. Generally, readings are chosen to accompany any lecture or tutorial video provided for the week. So, you should expect to complete the readings before playing the videos.

This course has two required textbooks (Murach and Opperl). Most of the required reading assignments are from the Murach text. A few required reading assignments are from the Opperl text.

Please note that when we are covering the material from the Murach book, the readings and the recorded lectures will cover the same material. I have marked both the readings and the lectures as *required*. While many graduate-level courses do not include lectures on the material from the text, some graduate-level learners prefer learning by listening. I invite you to sample all of the content and then settle into a usage pattern that works best for you.

2. Live Lectures and Lecture Videos

The lectures for this course will be a combination of live lectures that I will deliver during our online sessions and pre-recorded lecture videos. I will be doing live lectures on weeks when we are covering material from the Opperl text. These weeks are at the very beginning and at the very end of the course schedule.

I have pre-recorded lecture videos for all of the chapters that we will be covering from the Murach text. While most of the material covered in the recorded lectures is from the textbook, I occasionally cover supplemental material in the lectures as well. I always include commentary that I believe adds value to the text.

As mentioned above, when we are covering the material from the Murach book, the readings and the recorded lectures will cover the same material. I have marked both the readings and the lectures as *required*. While many graduate-level courses do not include lectures on the material from the text, some graduate-level learners prefer learning by listening. I invite you to sample all of the content and then settle into a usage pattern that works best for you.

Before playing my lecture videos, make sure that you have previously played [Tips on Playing My YouTube Videos](#) to assure that you get the most from your viewing experience.

3. Tutorial Videos

I have created tutorial videos for the following use cases:

- There are a number of tutorials that explain how to do activities necessary for the course. These include installing software, reading grading rubrics, submitting assignments, and related activities.

- Some tutorials are coordinated with the exercises in your coding assignments. They represent a demonstration of the skill you will be expected to use when doing the exercise.

Because tutorial videos do not duplicate material covered by other resources, I recommend that you always play these recordings. Before playing my tutorial videos, make sure that you have previously played [Tips on Playing My YouTube Videos](#) to assure that you get the most from your viewing experience.

4. Coding Assignments

There will be weekly Coding Assignments. As mentioned above, coding assignments will sometimes be paired with tutorial videos. These should allow you to complete your Coding Assignment using the same general approach that has been demonstrated in the video.

A Moodle submission activity will be provided for submitting each assignment. Instructions for each Coding Assignment and a grading rubric will be published in the Weekly Schedule.

Solutions to Coding Assignments will be posted to our Moodle site just before our next online class session. You can expect us to review your solutions and mine at the beginning of our next class.

A major goal for this course is to build your proficiency in self-evaluation of your work. To build this skill, I will expect you to be able to estimate your grade on each Coding Assignment. The solutions to coding assignments posted to our Moodle site and our review of those solutions during the next class will serve as your primary feedback for the Coding Assignment.

As secondary feedback, your Coding Assignment submissions will be graded and commented upon. This feedback will be published to the Moodle assignment submission activity within 2 weeks.

The grading rubric for Coding Assignments has been designed to promote two important behaviors:

- Submitting your work in a properly named and formatted file. This helps substantially with grading workflow.
- Submitting your work by the week-ending deadline. This assures that you will get the benefit of having tried to solve the problem on your own before seeing the solutions of others.

While separate grading rubric and assignment submission instructions documents will be published, the following is a summary of the coding assignment grading rubric features:

- 10 points will be awarded for submitting a single, properly named and properly formatted file to the proper Moodle assignment submission activity.
- A minimum of 75 points will be awarded for submissions that are submitted on time, and that demonstrate a good faith effort on all parts of the

assignment. Late submissions will be awarded 74 points or fewer in this category.

The implication of this grading scheme is that you can expect a score of 85 or higher on all Coding Assignments that meet both of these criteria.

5. *Participation*

Participation grades will be based upon the number of participation points earned during the semester. These are the activities through which you can earn participation points:

Activity	Points Earned
1 post or reply made in the <i>Open Discussion</i> forum.	1
1 speaking contribution during class	2
1 chat contribution during class	1
1 presentation of your Coding Assignment solution during class	5
1 presentation as spokesperson for your a breakout group during class	5

6. *Final Project*

The final project will require you to implement a MySQL database solution to a business problem. You will be given a logical database design document for a database that supports the activities of a small business. You will be responsible for creating a MySQL schema for the database, for loading test data, and for coding a series of SQL scripts. More detailed instructions for this project will be published separately.

Basis for Determining Grade

The various components of student work will contribute to the final grade based upon the following percentages:

- Class participation 10%
- Required coding assignments 45%
- Final project 45%

Letter grades will be determined as follows:

- A+ 97 - 100%;
- A 93 - 96%;
- A- 90 - 92%;
- B+ 87 - 89%;
- B 83 - 86%;
- B- 80 - 82%;
- C+ 77 - 79%;
- C 73 - 76%;
- C- 70 - 72%;
- D+ 67 - 69%;
- D 63 - 66%;
- D- 60 - 62%;
- F 0 - 59%;

ISCHOOL AND UNIVERSITY ACADEMIC POLICIES

Academic Integrity

The iSchool has the responsibility for maintaining academic integrity so as to protect the quality of education and research in our school and to protect those who depend on our integrity. Consequences of academic integrity infractions may be serious, ranging from a written warning to a failing grade for the course or dismissal from the University.

See the student code for academic integrity requirements:

<http://studentcode.illinois.edu/article1/part4/1-401/>

Statement of Inclusion

<http://www.inclusiveillinois.illinois.edu/mission.html>

As the state's premier public university, the University of Illinois at Urbana-Champaign's core mission is to serve the interests of the diverse people of the state of Illinois and beyond. The institution thus values inclusion and a pluralistic learning and research environment, one which we respect the varied perspectives and lived experiences of a diverse community and global workforce. We support diversity of worldviews, histories, and cultural knowledge across a range of social groups including race, ethnicity, gender identity, sexual orientation, abilities, economic class, religion, and their intersections.

Religious Observances

In keeping with our Statement of Inclusion and Illinois law, the University is required to reasonably accommodate its students' religious beliefs, observances, and practices in regard to admissions, class attendance, and the scheduling of examinations and work requirements.

If you anticipate the need for an accommodation, please communicate with your instructor in the first two weeks of class. If you are an undergraduate student and your instructor requires an absence letter, you must fill out the Religious Observance Accommodation Request form: <http://odos.illinois.edu/community-of-care/resources/docs/Religious-Observance-Accommodation-Request-Form.pdf>. Other accommodations may also be available.

Accessibility Statement

To obtain accessibility-related academic adjustments and/or auxiliary aids, students with disabilities must contact the course instructor and the Disability Resources and Educational Services (DRES) as soon as possible. To contact DRES you may visit 1207 S. Oak St., Champaign, call 333-4603 (V/TTY), or e-mail a message to disability@uiuc.edu.

Contact Hours

This course will require approximately 54 contact hours.

Last Revised

2020-01-13