# SYLLABUS FOR CS340: INTRODUCTION TO DATABASES

## COURSE INFORMATION

**CREDITS: 4**

**INSTRUCTOR: AMENEH SARBAZIAZAD**

**EMAIL: SARBAZIA@ONID.ORST.EDU**

**COURSE CATALOG DESCRIPTION:** *Design and implementation of relational databases, including data modeling with ER or UML diagrams, relational schema, SQL queries, relational algebra, user interfaces, and administration. PREREQS: CS 162 or ECE 152*

**INSTRUCTORS DESCRIPTION:** In this class you will have the opportunity to learn the basic skills needed to: design and implement a simple to moderately complex database, make good decisions regarding database design and document your design and queries in standardized notation. Additionally you will have the opportunity to practice these skills by building a simple website driven by a database backend.

**REQUIRED TEXT:** *None*

**OPTIONAL TEXT:**
*PHP and MySQL Web Development (4th Edition)*
ISBN: 9780672329166

## COURSE SPECIFICS

**COURSE STRUCTURE:** The course is essentially broken into two pieces. Around 3/4 of the course is dedicated to learning the basics of database design and use. You will complete 3 medium homework assignments and discuss them with your peers. The second part of the class will involve creating a simple website with a robust database backend.

There are two kinds of prepared material, they are both equally valuable. One kind are lectures which more formally go over the material. The second are "Real Coding" videos that demonstrate actual implementation and use of MySQL and web programming.

There will also be small quizzes which will help you keep informed of your understanding of the material. *If you ever get a wrong answer on a quiz that means you are not understanding critical information* and it is your responsibility to get clarification from your peers.

**PREREQUISITES:** You should be familiar with good coding practices. Good coding style is required and not taught in this class. You should understand basic control structures. If you are unable to code a simple sorting algorithm, you will have trouble in this class as this level of coding experience is expected.

**EVALUATION:** There are two different stages of evaluation. *Formative evaluation* will let you know if you are on the right track. Its purpose is to help you learn. This will come primarily from your peers and myself on the discussion forums, the exception is for the final project where you get several optional turn ins for ungraded feedback. The other form is *summative evaluation*. This is the grade you receive on completed projects. Its purpose is to measure the extent to which you understand the material. In other words, you will not be getting much feedback after assignments are submitted so it is your responsibility to get feedback early by discussing concepts using the methods of communication listed for this class.

## COMMUNICATIONS

**PIAZZA FORUM:** The Piazza Forum is the main place for our class discussion. You can ask your questions regarding course materials, quizzes and assignments there. .

**GOOGLE CHAT:** This form of communication will be used during office hours.

**EMAIL:** All email sent to myself or a TA should include CS340 as the first word in the subject. Additionally *email should only be used for grading related or personal questions*. If you are confused about a topic and need help, use Piazza. If you email us questions, we will simply ask you to post it to Piazza. The one exception is if you already posted a question in Piazza and a TA or the instructor did not respond within 24 hours, you can email us asking us to respond.

## ECAMPUS ELECTRONIC FORUM PARTICIPATION EXPECTATIONS

This program strives to prepare students for careers in computer science, which includes preparing students to communicate professionally. Therefore, students in this class are expected to communicate in a professional manner in discussion forums, email messages and all communications for this course. Critiques, disagreements, problems, or other topics of a sensitive nature can be addressed, but should be addressed civilly and professionally. If a student's communications become unprofessional, disruptive, abusive, inflammatory, or if they otherwise obstruct the learning process of the class, the instructor may restrict the student from participating in the electronic forums associated with the class and notify Ecampus and the OSU Office of Student Conduct and Community Standards. Productive learning communities and workplaces depend on civil, professional discourse. It is our hope that this policy strengthens your learning community and prepares you for the professional workplace.

## FINAL EXAM

The final exam covers all of the material for the class. It is proctored and needs to be scheduled in advance. If you do not contact the instructor at least 2 weeks before the exam window opens about any issues with scheduling, you will get no leniency.  Emergences are handed on a case by case basis. The earlier you contact the instructor the more likely the outcome will be in your favor.

## GRADING SCALE

**GRADE  AVERAGE**
A        93 OR GREATER
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        LESS THAN 60

**WEIGHTS:**

QUIZZES 15% Even split between all quizzes

ASSIGNMENTS 25% Even split between all assignments

FINAL 30%

PROJECT 30%

## LATE POLICY

You can discuss circumstances with me before an assignment is due to consider possible options if you are not going to get it done on time. Those options are more forgiving than the chart below.

Late penalties (t = time past due date):

| | |
|---|---|
| 1 second ≤ t > 6 hours | -5% |
| 6 hours ≤ t > 24 hours | -10% |
| 24 hour ≤ t < 48 | -25% |
| 48 ≤ t | No Credit |

## COURSE OUTCOMES

**General Database**

| KNOWLEDGE | SKILLS | ACTIVITIES | ASSESSMENT |
|---|---|---|---|
| • *Differences between databases and flat file systems* <br><br> • *Appropriate uses of a database* <br><br> • *Role and place for a DBMS* | | • *Recorded lectures* <br><br> • *Class discussion* | FORMATIVE <br><br> • *Quizzes* <br><br> SUMMATIVE <br><br> • *Final* |

**Database Design**

| KNOWLEDGE | SKILLS | ACTIVITIES | ASSESSMENT |
|---|---|---|---|
| • *Relationship cardinalities* <br><br> • *Entities vs Relationships* <br><br> • *Prevention of data anomalies through design* <br><br> • *Proper use of keys* <br><br> • *Basic database normalization* | • *ER Diagramming* <br><br> • *Schema Creation* <br><br> • *ER ←→ Schema conversion* | • *Model a moderately complex data set by using an ER diagram, and derive a relational schema from that diagram* <br><br> • *Create backend database design for simple website* | FORMATIVE <br><br> • *Quizzes* <br><br> • *Optional critiques* <br><br> SUMMATIVE <br><br> • *Final project* <br><br> • *Final* |

**MySQL**

| KNOWLEDGE | SKILLS | ACTIVITIES | ASSESSMENT |
|---|---|---|---|
| • *Indices: appropriate use and application* <br><br> • *Joins: types and application* | • *Basic syntax: data creation, data manipulation and selection* <br><br> • *Debugging* <br><br> • *Database creation from schemas* | • *Create a relational database from a relational schema* <br><br> • *Implement a database backend for a simple website* <br><br> • *Automated queries checks* | FORMATIVE <br><br> • *Quizzes* <br><br> • *Optional critiques* <br><br> SUMMATIVE <br><br> • *Final project* |

| | | | • *Final* |
|---|---|---|---|
| **Relational Algebra** | | | |
| KNOWLEDGE | SKILLS | ACTIVITIES | ASSESSMENT |
| • *Selection*<br><br>• *Projection*<br><br>• *Joins* | | • *Formulate simple queries in relational algebra by using projection, selection, product, and join operations* | FORMATIVE<br><br>• *Quizzes*<br><br>SUMMATIVE<br>• *Relational Algebra Assignment*<br><br>• *Final* |
| **Web Database Development** | | | |
| KNOWLEDGE | SKILLS | ACTIVITIES | FORMATIVE |
| • *Components of database driven website*<br><br>• *Roles of server side languages in database driven websites*<br><br>• *Security concerns* | • *Use of a server side language to:*<br>  ○ *display dynamic data on a website*<br>  ○ *store input from users*<br>  ○ *display related data from different tables* | • *Create database driven website* | • *Quizzes*<br><br>• *Optional critiques*<br><br>SUMMATIVE<br>• *Final project* |

## STUDENT OUTCOMES

At the end of the course, the following objectives should be met at the proficient level

| SKILLSET | LEARNING | BASIC | PROFICIENT |
|---|---|---|---|
| *DATABASE DESIGN* | *I don't yet know how to model simple entities or relationships in either ER diagrams or schemas.* | *I am able to create ER diagrams but do not know the situations that are difficult or impossible to capture. I have some difficulty converting them to schemas if they include more complex relations or unusual hierarchies.* | *I am able to create ER diagrams which correctly capture an external schema. I know the limitations of my ER diagrams and note them. I can convert those to schemas to be used in database implementation even if they include many-to-many relationships and class hierarchies. I know what situations will be difficult to implement in practice.* |
| *DATABASE IMPLEMENTATION* | *I don't yet know how to create tables or write queries to select data.* | *I can create tables but sometimes make incorrect decisions regarding best practices in the use of* | *I can create tables that include complex foreign key relationships. I know when to use indices. I can* |

|  |  | foreign keys. I don't always use indices where appropriate. I don't like to use sub-queries to compare data across tables. | construct correct queries that join tables and select data without needing anything more than the table definitions. I know when to calculate values rather than store them. I know how to string together many joins to select data across many tables. |
| --- | --- | --- | --- |
| *DATABASE COMMUNICATION* | *I don't know common terminology to discuss or write ideas about databases.* | *I am able to communicate basic ideas regarding database but may not use proper notation when documenting ideas in writing.* | *I am able to specifically and clearly talk about issues with databases and write definitions using diagrams, schemas and relational algebra where appropriate to communicate my ideas.* |
| *WEB APPLICATIONS* | *I don't yet know how to create a website that pulls data from a database.* | *I know how to connect to a database and run simple queries to insert and query data.* | *I know how to correctly insert and query data to prevent security issues. I am comfortable using the results of one query to then use in a future query. I produce clean code that throws no errors, warnings or notices.* |
| *DEBUGGING* | *I don't know the proper way to debug database applications.* | *I often don't know the best steps to take to debug issues. I have difficulty finding the code responsible for errors but with enough trial and error can fix errors.* | *I know how to isolate areas that are causing problems. I test early and often. I know when to and how to consult documentation. I know how to ask for help in public forums.* |

## ADACEMIC DISHONESTY

Do not copy work without citing it: this includes lectures, other students' work and Internet sources. This is plagiarism and will result in the strictest of punishments. Exams are summative evaluations of your performance. Do not share information with others regarding exams.

You will not get in trouble for posting too much code. I may ask you nicely to remove it if you posted more than I am comfortable with, but in almost all instances you should be able to rework your question or comment to include less code. After an assignment is due I encourage you to share code in a way that makes it easy for other students to access. This means providing links to source files or directly posting small snippets of code. Again, this is encouraged. Post your work for feedback!

## SCHEDULE

| TIME PERIOD | ACTIVITIES |
|---|---|
| Week 1 | Watch Week 1 videos. Read this document. Quiz1 due at end of week 1 (4/3) |
| Week 2 | Watch Week 2 videos. Project proposal, Quiz2 and Quiz3 due at end of week 2 (4/10). |
| Week 3 | Watch Week 3 videos. Quiz4 due at end of week 3 (4/17) |
| Week 4 | Watch Week 4 videos. Assignment1 due at end of week 4 (4/24) |
| Week 5 | Watch Week 5 videos. Quiz5 due at the end of week 5 (5/1) |
| Week 6 | Watch Week 6 videos. Assignment2 (SQL Assignment) due at end of week 6 (5/8) |
| Week 7 | Watch Week 7 videos. Project queries are due if feedback is desired. Quiz6 (Relational Algebra quiz) due at end of week 7 (5/15) |
| Week 8 | Watch Week 8 videos. Project interface is due if feedback is desired. Assignment 3 turn in due at end of week 8 (5/22) |
| Week 9 | Watch Week 9 videos. Project PHP is due if feedback is desired (5/29) |
| Week 10 | Watch Week 10 videos. Project is due (6/5) |
| Post-week 10 | Final exam occurs **Proctored Final (6/5-6/7)** |

## STUDENTS WITH DISABILITES

"Accommodations are collaborative efforts between students, faculty and Disability Access Services (DAS). Students with accommodations approved through DAS are responsible for contacting the faculty member in charge of the course prior to or during the first week of the term to discuss accommodations. Students who believe they are eligible for accommodations but who have not yet obtained approval through DAS should contact DAS immediately at (541) 737-4098."

Students with documented disabilities who may need accommodations, who have any emergency medical information the instructor should be aware of, or who need special arrangements in the event of evacuation, should contact the instructor as early as possible, and no later than the first week of the term. Class materials will be made available in an accessible format upon request.

## ABET REQUIREMENTS

1. Describe the difference between a relational database and a flat file

(Level 1; ABET Outcomes: A, j).

2. Model a moderately complex data set by using an ER diagram, and derive a relational schema from that diagram (Level 3; ABET Outcomes: A, b, C).

3. Create a relational database from a relational schema (Level 4; ABET Outcomes: A, K, c).

4. Create multiple indices in a relational database, and explain when and why such indices are appropriate (Level 5; ABET Outcomes: A, b, C).

5. Formulate SQL statements for data manipulation (Level 4; ABET Outcomes: A, c).

6. Formulate simple queries in relational algebra by using projection, selection, product, and join operations (Level 3; ABET Outcomes: A, I).

7. Describe the components and interfaces of a Web-based database system (Level 1; ABET Outcomes: A, B, I).

8. Design and implement a Web-based relational database system, using one or more scripting languages (e.g., PHP) and an open-source database development system (e.g., MySQL) (Level 4; ABET Outcomes: a, B, C, I, K).