ECE 656 Database Systems (Overview)

Paul A.S. Ward

This is an introductory graduate course on database systems. It covers what a database system is, why it is useful, and how it is used. In particular, this course will describe in detail the relational data model and how it enables effective problem solving, the logical and physical layers of database systems in general and of relational database systems specifically, and correct data modeling and database design. It also covers transactions, data persistence, and system dependability, both in terms of proper operation as well as system implementation and the implications of that realization. Data mining and system administration are also covered. Time permitting, the course will give a brief introduction to advanced topics in databases.

1 Learning Outcomes:

Students who complete the course requirements will be able to:

1. Understand Entity-Relationship Models (ERM) (entities, constraints, relationships, cardinality, primary keys)
2. Create an appropriate ERM given a broad set of requirements
3. Understand the Relational Model (RM) and Relational Algebra (RA) and be able to solve problems using RA
4. Transform an ERM to a RM
5. Recognize and create good relational designs for data, including functional dependency, multi-functional dependency, normal forms, and relational decomposition
6. Recognize poor relational designs and explain why they are poor
7. Understand the basics of SQL, especially DML and DDL, but also DCL and TCL, and its integration into client-application code
8. Transform a RM to SQL and RA to SQL
9. Understand and resolve performance problems in databases
10. Understand the basics of transactions and the relationship between transactions and data integrity
11. Understand the basics of modern database architecture, including indexing, query optimization, data storage, concurrency control, transaction management, and failure recovery
12. Understand the significance of database architecture for good relational design and apply that to ensure good relational design
13. Understand the basics of database administration, setting up and managing a database, and security basics
14. Understand the basics of data mining and be able to implement one of the three main data-mining algorithms

2 *Prerequisites:*

To be successful in this course students should already understand the following materials:

Set Theory: especially relations, functions, closure, and proofs
Logic: especially propositional logic, operators, and predicate logic
Data Structures: especially B-trees and abstract data types
Algorithms: especially sorting, hash functions, and Big-O notation
Systems Programming: especially memory and file management, but also processes, threads, inter-process communication, *etc*.
Language Arts/English: syntax, semantics, grammar, composition
Practice: recognizing that practice is essential to learning

It is implicitly assumed because of the above that students have knowledge of the C programming language and some variant of un*x. Knowledge of a scripting language is extremely helpful (*e.g.*, bash, Perl, *etc*.). It is likewise assumed that students have access to suitable computer equipment, though database-server machines will be accessible for students taking this course.

3 *Course Organization*

Most, though likely not all, lecture content will be in the form of pre-recorded videos available on the course website. In any week in which lecture content is in pre-recorded videos, the class time will be used for problem-solving exercises and addressing any student questions (especially as they pertain to upcoming assignment or project issues). At most half the time will be in problem-solving exercises. If there are few or no student questions it is likely that class time will end early.

A graduate course should take the average student 15 hours per week on average. With weekly scheduled time of 2.5 hours of lectures material and 2.5 (or fewer) hours of tutorial-like materials, students should expect to spend at least 10 hours per week in addition to the lecture and tutorial material. Note that this time does not include any time you may have to spend if you are missing some of the prerequisite material.

Lecture Schedule

The lecture schedule will be approximately as follows:

Week	**Topic**	**Objective(s)**
1, 2	Purpose; relational model and algebra; solving problems with RA	3
3, 4	Logical layer; SQL DML and DDL language basics; transforming RA to SQL	7, 8
5, 6	Database design: ER modeling; transforming ERM to RM; atomicity; functional dependencies, multi-valued dependencies, and relational decomposition	1, 2, 4, 5, 6
7, 8	Physical layer: storage, buffer-pool, file organization, indexes, query evaluation	9, 11
9, 10	Transactions, data persistence, and system dependability	10, 11, 12
11	Data mining: supervised and unsupervised analysis techniques; classification, clustering, and correlation	14
12	Advanced topics (text databases, key-value stores, warehousing, ...)	

Administration and security issues (13) will be covered throughout the term rather than as a distinct section, as it is necessary for students to learn the basics of administration from very early on in the course.

Course Website

All information will be posted on Learn and/or Piazza. Piazza will also be used for surveys and as a discussion forum. The Piazza course website is:

`https://piazza.com/uwaterloo.ca/<varies-from-term-to-term>/ece656`

The average response time for questions asked on Piazza is typically less than an hour, depending on the time of day, day-of-the-week, and the willingness of your fellow students to engage in the discussion. Outside of class time, do not expect any response to any other form of communication (e-mail, Learn, *etc.*). Questions on piazza can be directed just to the instructor if you wish and/or can be asked anonymously to the entire class. You will be enrolled into this Piazza course at the start of the course and will receive an e-mail requesting you sign up to Piazza (if you have not already done so in the past). If you do not receive such an e-mail within a week of the course starting, please notify the course staff so this can be resolved.

Any questions that you have should be asked in class or on Piazza. Questions asked elsewhere are likely to be missed. Even if they are seen, you will simply be redirected to Piazza. Piazza enables communication with any particular instructor, the entire instruction team, or the whole of the class, including the instruction team.

Course Materials

Course materials will be posted on Learn and/or Piazza and will include the following:

- Course Notes
- Presentation Slides
- Video Lectures
- Selections from "Readings in Database Systems"
  - 5th edition: `http://www.redbook.io/`
  - 4th edition: `http://www.redbook.io/archive/bib4.html`
  - This is also known as "The Red Book"
- The MySQL Reference Manual
  - `https://dev.mysql.com/doc/refman/8.0/en/`
- Additional reference materials may be added as appropriate to the course requirements

The presentation slides do not contain sufficient information to understand the course material. You will need to view the video lectures, read the course notes and selected materials from the redbook, as well as attend class, ask questions in class and/or on piazza, do assignments, and do the course project.

Reference Texts

You may find some of the following texts helpful:
C.J. Date and Hugh Darwen, *Databases, Types, and The Relational Model: The Third Manifesto*. Available at `https://www.dcs.warwick.ac.uk/~hugh/TTM/documents_and_books.html`
Silberschatz, Korth, and Sudarshan, *Database System Concepts*. McGraw-Hill. Both the 7^{th} (2019) and 6^{th}editions are reasonable texts, though there are substantial limitations in those texts in terms of materials on relational algebra and performance, as well as differences from materials taught in this course, especially with respect to issues database design, normalization, and performance analysis.

Course Deliverables

The course deliverables comprise several assignments (20%), a two-person project (30%), and a final exam (50%).

4 Note

This description is provided to give potential students an approximate overview of the course. The specific syllabus, included detailed information about the value and timing of various deliverables, will be provided at the start of the course.