

CS 4604: Introduction to Database Management Systems

Fall 2008

From the Undergraduate Catalog Emphasis on introduction of the basic database models, corresponding logical and physical data structures, comparisons of models, logical data design, and data base usage. Terminology, historical evolution, relationships, implementation, data base personnel, future trends, applications, performance considerations, data integrity. Senior standing required. A grade of C or better required in CS prerequisite 2604.

Instructor's Perspective CS 4604 is intended to be a first course in database systems for advanced undergraduates in computer science. It offers students an introduction to the design and programming of database systems. In particular, we will cover the entity-relationship (ER) approach to data modeling, the relational model of database management systems (DBMSs) and the use of query languages such as SQL. We will also cover relational algebra and the use of SQL in a programming environment. We will briefly touch upon query processing and mention the role of transaction management. We will try to devote a class or two to current topics of research such as XML databases, data warehousing and data mining. Topics outside the scope of this course are how to implement a DBMS, active database elements, and object oriented DBMSs.

Meeting Times 5:00pm–6:15pm, Tuesdays and Thursdays, McBryde 218

Instructor Zaki Malik, 231-8573, CRC Knowledge Works II, #2205
zaki AT cs DOT vt DOT edu, <http://people.cs.vt.edu/~zaki>
Please include "CS 4604" in the subject of all email messages

Office Hours 2pm–4pm Mondays and 2pm–3pm Wednesdays

Teaching Assistant Haiyan Cheng, hcheng04 AT vt DOT edu

Office Hours 3pm–5pm, Tuesdays and 4pm–5pm Thursdays, McBryde 106

Listserv CS4604_91770@listserv.vt.edu

Course Web Page <http://courses.cs.vt.edu/~cs4604>

If you are unable to attend scheduled office hours and need to meet, please send an email to set up an alternative time. If you need any accommodations because of a disability, if you have emergency medical information to share with the instructor, or if you need special arrangements in case the building must be evacuated, please meet the instructor as soon as possible.

Approach The course achieves a balance between establishing a theoretical foundation for DBMS and pragmatic applications of DBMS in a real world environment. The course places particular emphasis on the logical design of relational database systems. A significant semester-long project reinforces lectures. In this project, students will apply concepts presented in class and obtain practical, hands-on experience. Since students are more likely to use rather than implement database systems, this introductory course covers logical aspects of database design and implementation rather than the construction of database management systems.

Textbook and References The required textbook is “A First Course in Database Systems,” by Jeffrey D. Ullman and Jennifer Widom, Prentice Hall, Second Edition *or Higher*, 2002. The authors maintain a webpage for the book with errata lists and solutions to selected exercises. Your local bookstore has copies for purchase (3rd Ed). The instructor and TA will maintain other online resources on the course webpage.

Evaluation There will be 5–6 homeworks, a midterm exam, a final exam, and a semester-long group project. Both exams are closed book and closed notes. Projects, performed by groups of 2–3 students, will involve designing and programming a real-world database application from scratch. Student groups will do assignments related to their projects at regular intervals and turn in these assignments. The first part of the project consists of modeling a database in relational terms, loading data into a real DBMS, and writing interesting SQL queries. The second part consists of re-designing the database from scratch using E/R diagrams and redundancy-removal techniques, and evaluating the differences between the initial ad-hoc design and the later more formal design. The third part of the project involves embedding SQL queries into host language programs, and creating web-based interfaces to the database.

Homeworks account for 30%, the project for 30%, the midterm exam for 15%, and the final exam for 25% of the grade.

The homeworks expand on material taught in class, teach you some design and modeling, and will improve your vocabulary. Many of them will be in tune with an appropriate step of your database project. The project will help you to use the concepts learnt in realistic situations and encourages the use of programming skills in a realistic computing environment. The instructor designs all the assignments (homeworks and projects) and exams (midterm and final). The instructor grades both the exams. The TA grades the homeworks and project assignments. If you feel that an exam, homework, or project has been graded incorrectly, you may request that it be regraded. You must make requests for regrades to the instructor within one week of the date you received the graded assignment back. Note that the *complete* homework/exam will be regraded. Requesting only a partial regrade is not an option.

Workload The course will appear deceptively simple but history shows that it is possible to get a variety of grades. The course also moves at a fast pace and the workload is generally high! The instructor assumes that you will relish a design-oriented course and have the initiative to learn about various aspects of database design and programming. If you would like a career involving database technology (i.e., how to implement databases, not just use them effectively), you should also take a more advanced course in database system implementation or pursue a graduate program in this area. As mentioned earlier, this course does not cover these aspects.

Keeping in Touch Please use the listserv actively for discussions and exchanging ideas. Since it is created automatically by a central university system, any student registered in CS 4604 will be added to the mailing list. If you do not receive a test mail from the instructor by the end of the first week of classes, ensure that your email address is properly recorded in the university system (the system only records @vt.edu addresses, not, for instance, @myfavoriteisp.com).

Syllabus Here is a weekly breakdown of the schedule for the course. The schedule on the class website lists required reading for each class. This schedule is subject to change.

- Week 1: Course logistics, Introduction to Database Systems (Chapter 1)
- Weeks 2–5: SQL, Relational Algebra, Data Definition (Chapters 2, 5, 6 and 7)
- Weeks 6–7: Data Modeling using E/R diagrams (Chapter 4)
- Week 8: Midterm review, Midterm examination

- Weeks 9–12: The Relational Model (Chapter 3)
- Week 13: Programming with SQL
- Week 14: Students' choice (XML, Data mining, etc.)
- Week 15: Final review, Final examination

Electronic Accounts You are expected to have an account in the undergraduate Linux lab in the first floor of McBryde Hall that will give you access to a PostgreSQL server. The server will run on the machine cs4604.cs.vt.edu. You will be able to access the server from any machine in the Linux lab. If you require Linux lab accounts, please contact the lab administrators. We will provide accounts for you on the database server at the appropriate time in the semester. You may use your personal computer for project assignments and for testing homework solutions. However, we cannot guarantee that we will be able to grade work (especially the first few steps and the last step of the project) done at home (on other versions of software or because of file transfer problems). If you use your personal computer for project assignments, we will clearly specify with each assignment what you need to submit and/or demonstrate.

Programming Languages We assume that you are familiar with programming languages such as C, C++, and Java and server-side scripting languages such as Perl and PHP. The course does not teach these languages but you may be required to use them for some of your project assignments (particularly toward the end of the course). If you have never used these languages before, we expect that you would either (i) obtain the necessary expertise outside the classroom or (ii) come and talk to the instructor about it (in which case, you will most probably be advised to drop this course). We strongly believe this background is essential to fully appreciate this course and that you should have acquired such a background by the end of the junior year.

Honor Code The honor code is in effect for every aspect of this class. You are expected to do your own work. No one may give you answers to homeworks or exams. You are allowed to work on project assignments with the members of your group. In other words, students are encouraged to communicate about general principles of the course, but all assigned homework must be done on an individual basis. The instructor and the TA are available to provide any assistance that you may need. You may not exchange any code or solutions, either in pieces or in entirety, by any electronic means or hard copy.

More information about CS 4604

- CS 4604 is not easy. There is typically no correlation between the number of years of experience with ORACLE/MySQL/substitute-your-favorite-db-here and your grade in CS 4604.
- Homework problems are often tricky and difficult. Many of them will not involve straightforward applications of concepts taught in class but will require you to apply these concepts in creative ways.
- For most of the homework problems, there is more than one correct solution. Therefore, solution sketches posted by the instructor cannot cover all possible answers. Keep in mind that many questions are designed to address sources of learning/design difficulties.
- Since you are students at or above the senior level, you are expected to demonstrate initiative in learning the various software systems needed for this course. A one-semester course such as this cannot cover all nuances, nor should you as computer science students get bogged down by the details of different implementations.