Course Outline

COMP3311
Database Systems

Semester 2 2013

Contents

- Course Aims
- Student Learning Outcomes
- Assumed Knowledge
- Course Staff
- Course Details
- Resources for Students
- Course Schedule
- Teaching Rationale
- Teaching Strategies
- Assessment
- Academic Honesty and Plagiarism
- Course Evaluation and Development
- Other Matters

Course Aims

This course aims to explore in depth the practice of developing database applications and the theory behind relational database management systems (RDBMSs). It will also give an overview of the technologies used in implementing database management systems and the past, present and future of database systems and database research.

Large data resources are critical to the functioning of just about every significant modern computer application, and so knowledge of how to manage them is clearly important in industry. In the context of further study, understanding how to use databases effectively is essential for courses such as COMP9321 Web Applications Engineering and COMP9322 Service-Oriented Architectures. COMP3311 also provides a foundation for further study in advanced database topics, such as COMP9315 Database Systems Implementation and COMP9318 Data Mining. Database concepts are also relevant in courses such as COMP9319 Web Data Compression and Search and COMP6714 Information Retrieval and Web Search

Student Learning Outcomes

By the end of the course, you should be able to:

- develop accurate, non-redundant data models
- realise data models as relational database schemas
- formulate queries via the full range of SQL constructs
- use stored procedures and triggers to extend DBMS capabilities
- write applications in PHP that interact effectively with databases
- know about other interfaces between programming languages and DBMSs
- build (Web) applications that interact with relational DBMSs
- understand principles and techniques for administering RDBMSs
- understand performance issues in relational database applications
- understand the overall architecture of relational DBMSs
- understand the concepts behind transactions and concurrency control
- appreciate query and transaction processing techniques within RDBMSs
- appreciate the past, present and future of database technology

Glossary:

- **DBMS**: DataBase Management System ... software system to support database manipulation
- **RDBMS**: Relational DBMS ... the most popular style of DBMS (refers to underlying data model)
- **SQL**: Structured Query Language ... the ANSI standard language for manipulating RDBMSs

This course contributes to the development of the following graduate attributes:

Graduate Attribute	Where Acquired
an in-depth engagement with relevant disciplinary knowledge in its interdisciplinary context	Assignments
the capacity for analytical and critical thinking and for creative problem solving	Assignments
the ability to engage in independent and reflective learning	Assignments

the skills to locate, evaluate and use relevant information (Information Literacy)	Assignments, Exercises
the capacity for enterprise, initiative and creativity	Assignments
the skills required for collaborative and multidisciplinary work	Assignment 1

Assumed Knowledge

The formal pre-requisite for this course has, in the past, been COMP2011 Data Organisation or COMP2911 Engineering Design 2. COMP2011 traditionally dealt with file manipulation, which is a useful background if you intend to deal with database internals. The file manipulation material, however has gradually disappeared and the most useful background from the current first- and second-year courses is now (a) two-semesters' experience in procedural programming, (b) some data structures and algorithms, especially data structures for searching. With the changes to the courses leading up to COMP3311, and especially with the move to 12-week semesters, it's become clear that many students are not well-prepared for an in-depth study of database internals. Thus, I have de-emphasised the DBMS internals aspect of this course, so that we can focus on the knowledge and skills required to produce solid database applications. (If you are keen to understand how DBMSs work internally (and, as CSE students, why wouldn't you be?) you can study this in detail in COMP9315 Database Systems Implementation.)

Since the pre-requisite for COMP3311 can be satisfied via first-year computing courses, you may wonder "Why isn't databases a second-year course?", You're not alone in thinking this; I believe the core material on database applications belongs in second-year (or even first-year). If you wish to bring this to the attention of CSE via the teaching committee, you have my full support.

A perpetual problem for COMP3311 is that around half of the class has already covered basic data modelling techniques (specifically ER diagrams) in courses such as INFS1603 Business Data Management. On the other hand, half of the class hasn't seen this material before, so we need to cover it in lectures. Those who have seen it before should treat this as revision. Some of the other material in the course (e.g. basic SQL) may also be familiar to students who have taken the above courses, but don't make the mistake of thinking "I know all this stuff"; we will definitely cover these areas in more depth than you've seen them previously.

Course Staff

Contact Details

Staff Name	Role	Email	Phone
John Shepherd	Course Convenor	jas at cse.unsw.edu.au	93856494
Ashesh Mahidadia	Lecturer	ashesh at cse.unsw.edu.au	56556
Rachid Hamadi	Lecturer	rhamadi at cse.unsw.edu.au	56559
Nathan Ellis	Lecturer	z5160405 at cse.unsw.edu.au	
Thomas Kunc	Lecturer	z5205060 at cse.unsw.edu.au	

Consultations

Staff Name	Day/Time	Location
John Shepherd	Tuesday 1600-1700	K17-410
John Shepherd	Wednesday 1600-1700	K17-410
John Shepherd	Friday 1000-1100	K17-410

Course Details

Course Code: COMP3311
Course Title: Database Systems
Units of Credit: 6
Course WebSite: http://www.cse.unsw.edu.au/~cs3311/
Handbook Entry: http://www.handbook.unsw.edu.au/undergraduate/courses/2013/COMP3311.html

Resources for Students

Any of the following books is a suitable text for this course:

- Database System Concepts, Silberschatz, Korth, Sudarshan, 6th edition, 2010, McGraw-Hill
- Database Management Systems, Ramakrishnan and Gehrke, 3rd edition, 2003, McGraw-Hill
- Fundamentals of Database Systems, Elmasri and Navathe, 6th edition, 2010, Addison-Wesley
- Database Systems: The Complete Book, Garcia-Molina, Ullman, Widom, 2nd edition, 2008, Prentice-Hall
- Database Systems: An Application-Oriented Approach Kifer, Berstein, Lewis, 2nd edition (Complete Version), 2006, Addison-Wesley

Choose the one that bests suits your learning style and preferences. Also, if you have access to an earlier edition of any of these (one or two editions less than the one given above), it'll be fine for this course.

The textbooks give the greatest detail on topics covered in the course. The Course Notes are less detailed, but have the advantage that they cover precisely the syllabus (but no more). The Lecture Slides contain a summary of the Course Notes, along with examples. All Course Notes, Lecture Slides and solutions to the examples from lectures will be posted on the course web site.

Documentation for the software systems to be used in this course (PostgreSQL, SQLite and PHP) can be found under the **Documentation** link on the course web page. The manuals provided with PostgreSQL and PHP are actually very good, and the SQLite documentation is reasonable, so you don't need to buy textbooks for these. However, if you feel more comfortable with a book, there are references to a range of books on the web sites for PostgreSQL, SQLite and PHP.

A general problem with technology textbooks is that they go out-of-date very quickly. Another problem is that many of them provide a brief introduction with some examples, and then give a summary of the manual. In general, I've found that O'Reilly books tend to be better than most.

PostgreSQL(v9) will be used as the primary DBMS in this course because it is a typical example of a full-featured client-server DBMS, and has the added bonuses that (a) it has a simple extensibility model and (b) has the source code available if you want to learn more about how DBMSs work. Commercial alternatives could have been Oracle, DB2, SQLserver, although none of these are available in source code form. (MS Access is not a full-featured relational database.) The only plausible open-source alternative is MySQL, but it was not a full-featured DBMS until version 5, and has a source code base that is largely cobbled together from a number of existing open-source systems. The PostgreSQL code base, on the other hand, is the result of coherent development by a relatively small team, and PostgreSQL is, IMO, technically superior to MySQL.

SQLite is a very widely-used example of a serverless RDBMS and will be used for contrast with PostgreSQL. It has a similar (90% compatible) dialect of SQL to PostgreSQL, and, like PostgreSQL, supports the SQL92 standard. As an exercise in portability, we'll be expecting whatever SQL statements you write for this course to work on both PostgreSQL and SQLite (although this may not always be possible). Because SQLite requires no server, it will be used as the DBMS in the Prac Exam.

For programming interaction with the database, we'll be using PHP(v5) and the Apache webserver. During lectures, we may look briefly at some other programming languages which are commonly used for interaction with databases (e.g. Java, Ruby, Python), but these will simply be for illustrative purposes; we won't expect you to use them in any prac work.

Course Schedule

The order that topics are covered in this course is probably not the most "natural order" for presenting them. The material is presented in lectures in an order that ensures that you are best prepared for the assignments.

The precise schedule is subject to change as the semester progresses.

Week	Lecture	Prac Work	Assignments	Quizzes
0	-	Fill out survey on background knowledge	-	-
1	Data Modelling, ER Model	SQLite: intro to DBs	Ass 1 released	-
2	Relational Model, ER->Rel	Install PostgreSQL server	Design Team discussions	-
3	SQL Schemas, Building DBs	Learn ER diagram tool	Finish design discussions	Quiz 1
4	SQL Queries	Schema definition; data constraints	Finish Ass 1	-
5	Stored Procedures, PLpgSQL	SQL queries, views	-	-
6	Triggers, Aggregates, Windows	PLpgSQL functions	-	Quiz 2
7	Programming with DBs, PHP	User-defined aggregates	Ass 2 released	-
8	Catalogs, Privileges	Updates and triggers	Work on Ass 2	-
9	Programming with DBs, PHP	-	Finish Ass 2	Quiz 3
10	Relational Design Theory	PHP Scripts and Databases	Ass 3 released	-
11	Relational Algebra, Query Processing	-	Work on Ass 3	-
12	Transactions, Concurrency	-	Work on Ass 3	Quiz 4
13	Database Trends, Course Review	-	Finish Ass 3	-

Teaching Rationale

Since this course is about the understanding and effective use of a specific technology (relational database management systems), practical use of the technology is critical to the learning outcomes of the course. Thus, I will spend a lot of time in lectures working through exercises, to illustrate the practice of using various database techniques and technologies. However, the primary learning focus in this course is the assignment work, which I've designed to be challenging and relevant (deal with real problems using real database schemas).

There are no allocated tutorials and no laboratory classes in this course, because I recognise that students

already have significant demands on their time, and formal classes are not necessarily the best way to force people to learn. However, don't make the mistake of thinking that "no classes" means that you can get by with "no work". I will be providing a host of exercises and prac work that I *expect you will do* in your own time. In other words, all of the material that would have been in tutes and labs is still available, but you can interact with it in your own time.

This semester, I will run a couple of tutorial-like sessions each week, at times to be determined once enrolments are stabilised and I get a feeling for the best timeslots.

Natually, sometimes people will "get stuck" on understanding various aspects of the material, so I run a MessageBoard where you can ask questions, and where I will respond relatively promptly if I'm near a computer at the time. I also run face-to-face consultations each week where you can come if you have a problem that can't be handled easily on the MessageBoard. Many debugging problems fall into this category, because it is often much easier to show me the whole environment where the bug occurs, rather than trying to explain it to me via email and giving me only partial information.

Teaching Strategies

There are three primary modes of learning for this course: lectures, exercises, assignments.

Lectures/Exercises

Each week there will be three hours of lectures during which theory, practical demonstrations and case-studies will be presented. There are two sources of material for lectures: Course Notes and Lecture Slides. The Course Notes are a detailed account of the various topics (although not as detailed as a textbook), and will be released well before the topic is discussed in lectures. The Lecture Slides will contain a summary of the Course Notes, along with exercises, and I will not make them available until after the lecture, in order to keep the exercises "fresh". You will get maximum benefits from lectures if you read the relevant course notes (or textbook chapters) before attending each lecture.

There will be no tutorials or lab classes in this course, but tutorial-type questions and practical exercises will be made available. You are expected to work through these yourself. Note: none of these are assessable, but I am assuming that you will be interested enough in the topics to actually do them without the need for assessment-based incentives. It is definitely in your own best interest to keep up-to-date with the theory/prac exercises.

Given the absence of tutorials and labs, you **must** make use of (a) the WebCMS MessageBoard, (b) consultations, (c) email direct to me, in order to get your questions answered.

Assignments

In the assignments, you will consider the major stages of the database application development process, conceptual design, implementation as a PostgreSQL application, and building a Web interface on the Apache/PHP platform. The assignments contribute 35% of the overall mark for this course.

#	Description	Due	Marks
1	ER/SQL data model	Fri 23 Aug (end week 4)	7%
2	SQL and PLpgSQL	Fri 13 Sept (end week 7)	10%
3	Web interfaces (HTML,PHP)	Fri 18 Oct (end week 11)	15%

An important outcome of doing the assignment work is that you will advance your knowledge and skills in key course objectives: data modelling, query formulation, use of stored procedures, interaction of programming languages and DBMSs. Completion of the assignments will also aid in your understanding of database performance issues and Web interface construction. The assignments also lay the foundation for successful performance in the final exam.

All assignments (except the design part of assignment 1) will be completed *individually*; this means that you should do them *yourself* without assistance from others, except for asking advice from the lecturer in this course. As noted above, they are the primary vehicle for learning the material in this course. If you don't do them, or simply copy and submit someone else's work, you have wasted a valuable learning opportunity.

Assignments are to be submitted on-line (via WebCMS) before midnight on the due date. Assessment of assignments will be primarily based on how accurately they satisfy the requirements; this means that most of the mark will be based on automatic marking. However, we will also manually examine submitted assignments to determine (a) whether they are written with good style, (b) how closely they satisfied the requirements, if they fail the auto-marking. Late submissions will have marks deducted from the maximum achievable mark at the rate of roughly 1% of the total mark per hour that they are late (i.e. around 25% per day).

Assessment

There will be three places where your learning in this course will be assessed: assignments, quizzes and the final exam. Much as I dislike conflating the *learning* aspect of assignments with their *assessment* aspect, there will be marks for the assignment work. I would rather use the assignments entirely as learning vehicles and have no assessment associated with them, but I suspect that wouldn't result in a satisfactory outcome (i.e. nobody would

do them).

There will also be a number of on-line quizzes during the semester, to be taken in your own time, primarily as a way for you to gauge your progress in the course, but also worth some marks. Lastly, the final exam will form the major assessment in this course and aims to test what you learned about databases during the course of the semester.

The Final Exam will be conducted in the CSE laboratories, and will involve both a practical component (writing SQL queries on a supplied database) and and a theory component (e.g. data modelling, schema normalization, etc.). In order to meet the exam hurdle requirement, you must score better than 40% on both the prac *and* the theory component. Note that partial marks are available for all questions on the final exam, and the hurdle will be enforced after any required scaling.

Your final mark in this course will be based on the marks from the above three assessment components. Note that the exam is a hurdle, so that if you fail the exam badly enough, you cannot pass the course. The following formula describes precisely how the mark will be computed and how the hurdle will be enforced:

```
ass1    = mark for assignment 1      (out of 7)
ass2    = mark for assignment 2      (out of 10)
ass3    = mark for assignment 3      (out of 15)
quizzes = mark for on-line quizzes   (out of 8)
examP   = mark for exam (practical)  (out of 30)
examW   = mark for exam (written)    (out of 30)

exam    = examP + examW              (out of 60)
okExam  = examP > 12 && examW > 12   (after scaling)

mark    = ass1 + ass2 + ass3 + quizzes + exam
grade   = HD|DN|CR|PS  if mark >= 50 && okExam
        = FL           if mark <  50 && okExam
        = UF           if !okExam
```

Academic Honesty and Plagiarism

Plagiarism is defined as "using the words or ideas of others and presenting them as your own". UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Learning Centre: Plagiarism and Academic Integrity
- MyUNSW: Plagiarism and Academic Misconduct
- CSE: Addendum to UNSW Plagiarism Guidelines
- CSE: Yellow Form (whose terms you have agreed to)

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism.

Course Evaluation and Development

Every semester since 2001, COMP3311 has been surveyed via Andrew Taylor's CSE-wide Course Evaluation Survey. I have also evaluated it using phpSurvey, and my own set of questions, aimed specifically at getting feedback on things that I felt didn't work. Since 2007, we have used UNSW's online CATEI evaluation. Normally, COMP3311 rates reasonably well. In the 09s1 offering, the major complaints were that assignment marking/feedback happened too slowly and that the final assignment had too few marks for the amount of work involved. In the 11s1, the main complaints were about the lecture timeslot (9am) and the 3-hour lecture duration; there were similar complaints about the 3-hour lecture timeslot in 12s1. This year, we have two separate lecture timeslots, so you won't need to concentrate for 3-hours straight, and I'll try to get the assignments marked faster (although with twice as many students as in 2012 this will be challenging).

Another issue related to course development was the move to 12-week semesters. When the semesters were shortened from 14 weeks, COMP3311 was changed to: cut back on the intro and review lecture content (which makes it imperative that you read this document); reduce the level of detail at which query and transaction processing was discussed.

Other Matters

By the end of this course, we want you to be capable of building high-quality (correct, efficient, usable) applications based on relational databases, to have a sound understanding of issues in managing relational database management systems, and an overview of how they work internally.

Enjoy the course! ... *John Shepherd*, July 2013