Course Outline

Contents

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

Course Details

Course Code	COMP9315
Course Title	DBMS Implementation
Units of Credit	6
Course Website	http://cse.unsw.edu.au/~cs9315 (http://cse.unsw.edu.au/~cs9315)
Handbook Entry	http://www.handbook.unsw.edu.au/postgraduate/courses/current/COMP9315.html (http://www.handbook.unsw.edu.au/postgraduate/courses/current/COMP9315.html)

Course Summary

This course aims to introduce students to the detailed internal structure of database management systems (DBMSs) such as Oracle or SQL Server. DBMSs contain a variety of interesting data structures and algorithms that are also potentially useful outside the DBMS context; knowing about them is a useful way of extending your general programming background. While the focus is on relational DBMSs, given that they have the best-developed technological foundation, we will also consider more recent developments in the management of large data repositories.

Relational DBMSs need to deal with a variety of issues: storage structures and management, implementation of relational operations, query optimisation, transactions, concurrency, recovery, security. The course will address most of these, along with a look at emerging database systems trends. The level of detail on individual topics will vary; some will be covered in significant detail, others will be covered relatively briefly.

An important aim of this course is to give you a chance to undertake an in-depth exploration of the internals of a *real* DBMS: PostgreSQL. Lectures will discuss the general principles of how DBMSs are implemented, and will also illustrate them with extensive examples from PostgreSQL where possible.

Course Timetable

The course timetable is available here (/COMP9315/16s1/timetable) .

Student Learning Outcomes

After completing this course, students will:

- understand the internal architecture of large-scale data management systems (primarily relational DBMSs)
- be familiar with a variety of data-structures and algorithms for large-scale data manipulation
- be able to analyse and tune the performance of data-intensive systems
- be familiar with techniques for implementing transactional behaviour and reliable storage
- be able to install and administer the PostgreSQL database management system
- be familiar with some modern trends in database technology (e.g. MapReduce, big data)

At the end of this course, you should be in a position where you could make contributions to the further development of PostgreSQL. Some of you might even be at the stage where you could build a database management system "from scratch".

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

Graduate Capability	**Acquired in**
scholarship: rigorous in their analysis, critique, and reflection	doing quizzes and writing assignment reports
scholarship: able to apply their knowledge and skills to solving problems	carrying out assignment and prac work
scholarship: capable of effective communication	writing reports for assignments
leadership: enterprising, innovative and creative	solving assignment problems
leadership: collaborative team workers	assignments are group-based
professionalism: capable of independent, self-directed practice	understanding PostgreSQL for assignments

Assumed Knowledge

This course assumes that you have a solid understanding of the following topics:

- the C language and programming in C (or C++) (e.g. completed COMP9021 or COMP1927)
- developing applications on RDBMSs (SQL, relational algebra e.g. COMP9311 or COMP3311)
- file organisation/manipulation (Unix open,close,lseek,flock)
- sorting/searching techniques (quick/merge sorts, binary trees, hash tables e.g. COMP9024 or COMP1927)

If you do not have this prerequisite knowledge, you should reconsider your enrolment in the course, or else quickly set about developing your knowledge on these topics.

Teaching Rationale

Since this course is about the understanding the detailed implementation of a specific technology (relational database management systems), practical experience with 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 various database techniques and technologies. However, the primary learning focus in this course is the assignment work.

There are no tutorial or 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 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 when you like.

In order to provide some motivation to work each week, and to help you review how well you're understanding the course, there'll be a number of on-line quizzes which will contribute 10% of your final mark.

Natually, sometimes people will "get stuck" on understanding various aspects of the material; use the Webcms comment facility to ask questions. I'll try to respond 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 Forum. 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 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, exercises 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 generally be released 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 benefit 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: the questions and pracs are not 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. The quizzes will provide some assessment-based motivation to work consistently.

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 implement modifications/enhancments to the PostgreSQL DBMS and perform experiments to analyse your implementation. The assignments contribute 25% of the overall mark for this course.

#	**Description**	**Due**	**Marks**
1	Storage Management	week 5	11%
2	Query Processing	week 11	14%

Assignments are important, *not* for the marks they provide, but for the understanding they give you about the details of DBMS internals. Assignments will be completed in small groups (size 1 or 2) which you will form via the Groups component of WebCMS once it's activated. If you insist on doing the assignment alone, you won't get any benefit (e.g. no more time to do it, same expectation on the standard, no "kinder" marking).

As noted above, assignments are an important vehicle for learning the material in this course. If you don't do them (e.g. by not contributing to your group, or by the whole group copying and submitting someone else's work), you have wasted a valuable learning opportunity. As well as group interviews on assignment submissions, there will be exam questions directly related to the assignment work to test your individual knowledge of assignment work.

Assignments are to be submitted on-line (via WebCMS or Give) before midnight on the due date. Assessment of assignments will be based on how accurately they satisfy the requirements and how good your analysis is. Late submissions will have marks deducted from the maximum achievable mark at the rate of roughly 0.8% of the total mark per hour that they are late (this is equivalent to around 20% per day late).

Assessment

Your final mark in this course will be based on components from the assignment work and the exam. 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 11)
ass2   = mark for assignment 2 (out of 14)
quiz   = mark for quizzes (out of 10)
exam   = mark for exam (out of 65)
okExam = exam > 26/65 (after scaling)
mark   = ass1 + ass2 + quiz + exam
grade  = HD|DN|CR|PS if mark >= 50 && okExam
       = FL if mark < 50 && okExam
       = UF if !okExam
```

The Final Exam will be conducted in the CSE laboratories, and will involve some multiple-choice questions, some long-answer questions and some small implementation questions (C programming). Satisfactory performance in the exam is needed in order to meet the hurdle requirement, where "satisfactory performance" means a score of better than 40% after the exam marks have been scaled.

Academic Honesty and Plagiarism

Plagiarism is defined as (https://student.unsw.edu.au/what-plagiarism) *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 (http://www.lc.unsw.edu.au/plagiarism/index.html)
- MyUNSW: Plagiarism (https://student.unsw.edu.au/plagiarism) and Academic Misconduct (https://www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf)
- CSE: Addendum to UNSW Plagiarism Guidelines (http://www.cse.unsw.edu.au/~chak/plagiarism/plagiarism.html)
- CSE: Yellow Form (http://www.cse.unsw.edu.au/people/studentoffice/policies/yellowform.html) (whose terms you have agreed to)

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

One common form of **Academic Dishonesty** is to join an assignment group, contribute little or nothing to the assignment work, and then put your name on the assignment submission as if you have made a contribution. To ensure that only people who contribute effort to an assignment are awarded marks for that assignment, we will interview groups and ask each member questions about the assignment. If it is clear that you did not contribute to the assignment (because you can't answer questions about it), you will receive no marks for that assignment.

Course Schedule

The following provides a *tentative* schedule for how the course will run:

Week	Lectures	Quiz	Assignments
1	Course Intro, DBMS review, Rel Alg (RA), PostgreSQL, Catalog	-	-
2	Storage: Disks, Files, Buffers, Pages, Tuples	Quiz 1	-
3	RA Operations: Cost Models, Scanning, Sorting, Projection	-	-
4	Selection: Heaps, Sorted Files, Hashed Files, Indexes, Btrees	Quiz 2	-
5	Selection: N-d Hashing, N-d Trees, Signatures, Similarity Hashing	-	Ass 1 due
6	Joins: Nested Loop, Sort-Merge, Hash Join	Quiz 3	-
7	Query Processing, Optimization, Execution	-	-
8	Transactions: Isolation, Concurrency Control	Quiz 4	-
9	Transactions: Durability, Recovery	-	-
10	Parallelism, Distribution, MapReduce	Quiz 5	-
11	Column-oriented DBMSs, Non-SQL Databases	-	Ass 2 due
12	Big Data, Graph Data, Course Review	Quiz 6	-

The assignment deadlines are firm, but I reserve the right to change the lecture topics depending on your/my interest as the semester progresses.

Resources for Students

The course web site will provide access to a large range of material for this course:

- all course notes, along with slides and examples from lectures
- a collection of exercises, both theoretical and practical
- PostgreSQL source code and documentation
- a collection of research/survey papers discussing aspects of the syllabus

All of the following books contain useful material for this course:

- Database System Concepts (http://db-book.com/) , Silberschatz, Korth, Sudarshan, 5th edition, 2005, McGraw-Hill
- Database Management Systems (http://www.cs.wisc.edu/~dbbook/) , Ramakrishnan and Gehrke, 3rd edition, 2003, McGraw-Hill
- Fundamentals of Database Systems (http://wps.aw.com/aw_elmasri_fundatasys_5) , Elmasri and Navathe, 5th edition, 2006, Addison-Wesley
- Database Systems: The Complete Book (http://www-db.stanford.edu/~ullman/dscb.html) , Garcia-Molina, Ullman, Widom, 2nd edition, 2009, Prentice-Hall
- Database Systems: An Application-Oriented Approach (http://wps.aw.com/aw_kifer_datacomp_2) Kifer, Berstein, Lewis, 2nd edition (Complete Version), 2006, Addison-Wesley

There's no need to buy any of them, but if you plan to be seriously involved with databases in the future, any of them would be a useful addition to your professional bookshelf.

These books do not have much information on some of the topics later in the course, and especially if you get an older edition than the one noted above. There will be material in the Readings that covers the later topics.

All of the lecture examples and assignment work will be done using the PostgreSQL relational database management system. PostgreSQL is a typical example of a full-featured DBMS, and has the added bonuses that (a) it has a powerful extensibility model, and (b) has the source code available.

An inevitable question, when it comes to open-source DBMSs is "Why aren't you using MySQL, since everyone uses it?". The primary reason for not using it is that its source code is built from a collection of disparate components gleaned from a variety of existing open-source systems, thus making its code base somewhat "haphazard". Also annoying is the fact that MySQL often flaunts SQL standards in its default behaviour. The PostgreSQL code base, on the other hand, is the result of coherent development by a relatively small team, and PostgreSQL is, IMHO, technically superior to MySQL.

Note: you will be required to *compile* your own PostgreSQL server from source code to carry out the assignment work. This is easy to do on Linux and Max OSX, but not so straightforward on Windows (as far as I know). Since I have no access to a Windows system, I will not be able to assist students who insist on trying to do their development on that system. If you are working on the assignments at UNSW, you have Linux access by default. If you want to work at home, and currently run Windows, your best option would be to add a Linux partition to your system and run it dual-boot, or run a virtual Linux on top of Windows.

PostgreSQL documentation is attached to the course web site, and also available as a downloadable tar-ball. The PostgreSQL manual is actually very good (comprehensive, well-structured) so you don't need to buy a PostgreSQL text. However, if you feel more comfortable with a book, there are references to a range of books on the PostgreSQL web site (http://www.postgresql.org/docs/books/) .

We will be using PostgreSQL 9.4.6 for this course. If you insist on downloading your PostgreSQL source code from the PostgreSQL web site (http://www.postgresql.org/docs/books/) (rather than the course web site), make sure you get *this* version.

And, yes, version 9.5.1 has just been released, but I'd rather not inflict the latest bleeding-edge bugs on you.

A variety of books cover PostgreSQL, but 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 simply give a summary of the documentation. In general, I've found that O'Reilly books tend to be better than most. However, you should consider the longevity of any technology book before outlaying the typically high cost for it.

Course Evaluation and Development

This course is evaluated each session using the CATEI system.

In the previous offering of this courses, students were reasonably satisfied with the course, but did mention that assignment marking had taken too long. We plan to mark the assignments more quickly this time.

Also, the course has been somewhat "modernised" by adding recent examples of special-purpose, but widely used, DBMS styles.

Other Matters

By the end of this course, we want you to have a deep understanding of how database management systems work. This should help you in a number of ways: (a) make you better database application developers, (b) able to implement non-DBMS code for manipulating large amounts of data. Some of you might also make contributions to open-source DBMS projects, or even develop your own DBMSs.

Enjoy the course! ... *John Shepherd* , February 2016

Resource created 6 years ago (Monday 21 December 2015, 11:42:32 AM), last modified 6 years ago (Sunday 28 February 2016, 12:49:49 PM).

Comments (/COMP9315/16s1/forums/search?forum_choice=resource/1997) (/COMP9315/16s1/forums/resource/1997)

There are no comments yet.