Course Outline Public

Edit Resource (/COMP9315/20T1/resources/40353/edit)

Contents

- Course Details
- Course Summary
- 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
Course Convenor	John Shepherd (jas@cse.unsw.edu.au)
Course Admin	as above
Course Email	cs9315@cse.unsw.edu.au
Lectures	Mathews B: Monday 12-2, Tuesday 2-4 (campus map (https://student.unsw.edu.au/maps))
Consultations	TBA
Units of Credit	6
Course Website	http://www.cse.unsw.edu.au/~cs9315/ (http://www.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) (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 brief 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 aspect 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 examples from PostgreSQL where possible.

Student Learning Outcomes

After completing this course, students will be able to:

- describe the internal architecture of large-scale (relational) data management systems
- implement, test and analyse data-structures and algorithms for large-scale data manipulation
- choose appropriate data structures for given scenarios
- analyse and tune the performance of data-intensive systems
- describe techniques for implementing transactional behaviour and reliable storage
- install and administer the PostgreSQL database management system
- add new data types to PostgreSQL
- describe modern trends in database technology (e.g. noSQL, graphs)

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 COMP9024 or COMP2521)
- developing applications on RDBMSs (SQL, relational algebra e.g. COMP9311 or COMP3311)

- file organisation/manipulation (Unix open,close,lseek,read,write)
- sorting/searching techniques (quick/merge sorts, binary trees, hash tables e.g. COMP9024 or COMP2521)

If you *do not* have this prerequisite knowledge, you should *not* enrol in the course.

In particular, do *not* enrol in this course unless your C programming is strong. The exam and assignments require you to write complex C code. Also, PostgreSQL contains >100,000 lines of C code, some of which you will need to understand/modify.

Teaching Rationale

Since this course is about 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 some 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, backed up by the Prac Exercises.

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 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.

Naturally, sometimes people will "get stuck" on understanding various aspects of the material. Use the Webcms3 Comment facility to ask questions. The course staff will try to respond promptly if we're near a computer at the time.We will also run face-to-face consultations each week where you can come if you have a problem that can't be handled easily via the website. Many debugging problems fall into this category, because it is often much easier to show the whole environment where the bug occurs, rather than trying to explain it verbally and giving only partial information.

Teaching Strategies

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

Lectures

Each week there will be four 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.

Exercises

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 Webcms3 Forum, (b) consultations, (c) email to cs9315@cse.unsw.edu.au, in order to get your questions answered.

Quizzes

Over the course of the term, there will be a number of on-line quizzes which you do in your own time, via the Webcms3 quiz interface. These quizzes test your knowledge of content from the previous could of weeks. They are useful both for review and for assessment. The quizzes are worth a title of 15%, which is calculated by summing the marks for all quizzes and then scaling this into a mark out of 15.

Assignments

In the assignments, you will implement extensions to the PostgreSQL DBMS as well as building DBMS components in isolation and performing experiments to analyse your implementation. The assignments contribute 30% of the overall mark for this course.

#	Description	Due	Marks
1	New PostgreSQL Data Type	week 5	10%
2	Query Processing	week 10	15%

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 1-2) which you will form via the Groups component of Webcms3 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). There will also be peer assessment to ensure that everyone in a group makes a contribution to each submitted assignment; marks will be adjusted down to reflect lack of contribution.

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. There will be exam questions directly related to the assignment work to test your individual knowledge of assignment topics.

Assignments are to be submitted on-line (via Webcms3 or Give) before the specified 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 (equivalent to around 20% per day late).

Assessment

Your final mark in this course will be based on components from the quizzes, assignment work and 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 10)
ass2   = mark for assignment 2 (out of 15)
quiz   = mark for quizzes (out of 15)
exam   = mark for exam (out of 60)
okExam = exam > 24/60 (after scaling)
mark   = ass1 + ass2 + quiz + exam
grade  = HD|DN|CR|PS if mark ≥ 50 && okExam
       = FL if mark < 50 && okExam
       = UF if mark ≥ 50 !okExam
```

The Final Exam will be conducted in the CSE laboratories, and will involve written-answer analytical questions and some small implementation questions (C programming).

There is a hurdle requirement on the final exam. If you do not score at least 40% (24/60) on the exam (after scaling), you cannot pass this course. If you score very poorly (<40%) on the exam, but your overall course score exceeds 50%, you will be awarded a grade of UF (which counts as a fail).

Special Consideration

If your work in this course is affected by unforeseen adverse circumstances, you should apply for Special Consideration. If your request is reasonable and your work has clearly been impacted, then

- for an assignment, you may be granted an extension
- for the Final Exam, you may be offered a Supplementary Exam

Note the use of the word "may". None of the above is guaranteed. It depends on you making a convincing case that the circumstances have clearly impacted your ability to work.

UNSW handles special consideration requests centrally (in the Student Lifecycle division), so all special consideration requests must be submitted via the UNSW Special Consideration (https://iaro.online.unsw.edu.au/special_consideration/home.login) website.

Special consideration requests must be accompanied by documentation, which will be verified by Student Lifecycle. Do not email the course convenor directly about special consideration.

Extensions on assignments will only be awarded if the majority of the team are affected.

If you cannot attend the Final Exam because of illness or misadventure, then you must submit a Special Consideration request, with documentation, through MyUNSW within 24 hours of the exam. If your request is reasonable, then you will be awarded a Supplementary Exam (aka "Supp").

Note that UNSW expects you to be available to sit Supplementary Exams (held in the week Sep 9-13) if required. If you are awarded a Supp and do not attend, then your exam mark will be zero.

For further details on special consideration, see the UNSW Student website (https://student.unsw.edu.au/special-consideration) .

If you are registered with Disability Services, please forward your documentation to John Shepherd (mailto:jas@cse.unsw.edu.au) within the first two weeks of term.

Student Conduct

The **Student Code of Conduct** (Information (https://student.unsw.edu.au/conduct) , Policy (https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf)) sets out what the University expects from students as members of the UNSW community. As well as the learning, teaching and research environment, the University aims to provide an environment that enables students to achieve their full potential and to provide an experience consistent with the University's values and guiding principles. A condition of enrolment is that students *inform themselves* of the University's rules and policies affecting them, and conduct themselves accordingly.

In particular, students have the responsibility to observe standards of equity and respect in dealing with every member of the University community. This applies to all activities on UNSW premises and all external activities related to study and research. This includes behaviour in person as well as behaviour on social media, for example Facebook groups set up for the purpose of discussing UNSW courses or course work. Behaviour that is considered in breach of the Student Code Policy as discriminatory, sexually inappropriate, bullying, harassing, invading another's privacy or causing any person to fear for their personal safety, is serious misconduct and can lead to severe penalties, including suspension or exclusion from UNSW.

If you have any concerns, you may raise them with your lecturer, or approach the School Ethics Officer (mailto:ethics-officer@cse.unsw.edu.au) , Grievance Officer (mailto:grievance-officer@cse.unsw.edu.au) , or one of the student representatives (mailto:stureps@cse.unsw.edu.au) .

Plagiarism is defined as (https://student.unsw.edu.au/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:

- Plagiarism and Academic Integrity (https://student.unsw.edu.au/plagiarism)
- UNSW Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism. In particular, you are also responsible for ensuring that your assignment files are not accessible by anyone but you by setting the correct permissions in your CSE directory and code repository, if using one (in particular, do not put assignment code in a public GitHub repository). Note also that plagiarism includes paying or asking another person to do a piece of work for you and then submitting it as your own work.

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW staff and students have a responsibility to adhere to this principle of academic integrity. Plagiarism undermines academic integrity and is not tolerated at UNSW. Plagiarism at UNSW is defined as using the words or ideas of others and passing them off as your own.

If you haven't done so yet, please take the time to read the full text of

- UNSW's policy regarding academic honesty and plagiarism (https://student.unsw.edu.au/plagiarism)

The pages below describe the policies and procedures in more detail:

- Student Code Policy (https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf)
- Student Misconduct Procedure (https://www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf)
- Plagiarism Policy Statement (https://www.gs.unsw.edu.au/policy/documents/plagiarismpolicy.pdf)
- Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)

You should also read the following page which describes your rights and responsibilities in the CSE context:

- Essential Advice for CSE Students (https://www.engineering.unsw.edu.au/computer-science-engineering/about-us/organisational-structure/student-services/policies/essential-advice-for-cse-students)

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, Catalogues	-	-
2	No Lectures	Quiz 1	-
3	Storage: Disks, Files, Buffers, Pages, Tuples; Cost Models	-	-
4	Relational Algebra Operations: Scanning, Sorting, Projection	Quiz 2	-
5	Selection: Heaps, Sorted Files, Hashed Files, Indexes, B-trees	-	Ass 1 due
6	Selection: N-d Hashing, N-d Trees, Signatures, Similarity Matching	Quiz 3	-
7	Joins: Nested Loop, Sort-Merge, Hash Join	-	-

8	Query Processing, Optimisation, Execution	Quiz 4	-
9	Transactions: Isolation, Concurrency Control, Durability, Recovery	-	-
10	Distributed and non-SQL databases	Quiz 5	Ass 2 due

The assignment deadlines are firm, but I reserve the right to change the schedule of lecture topics.

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 Systems: The Complete Book (http://db-book.com/) , Garcia-Molina, Ullman, Widom, 2nd edition, 2009, Prentice-Hall
- Database System Concepts (https://www.mheducation.com/highered/product/0078022150.html) , Silberschatz, Korth, Sudarshan, 7 th edition, 2020, McGraw-Hill
- Fundamentals of Database Systems (https://www.pearson.com.au/products/D-G-Elmasri-Navathe/D-G-Elmasri-Ramez-Navathe-Shamkant-B/Fundamentals-of-Database-Systems-Global-Edition/9781292097619?R=9781292097619) , Elmasri and Navathe, 7 th edition, 2019, Addison-Wesley
- Database Systems: An Application-Oriented Approach (https://www.pearson.com/us/higher-education/program/Kifer-Database-Systems-An-Application-Oriented-Approach-Compete-Version-2nd-Edition/PGM268925.html) Kifer, Berstein, Lewis, 2nd edition (Complete Version), 2006, Addison-Wesley
- Database Management Systems (http://www.cs.wisc.edu/~dbbook/) , Ramakrishnan and Gehrke, 3rd edition, 2003, McGraw-Hill

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 11.3 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, PostgreSQL version 12 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 MyExperience 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 way more quickly this time, using auto-marking with extensive test cases.

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* , May 2019

Resource created 2 months ago (Thursday 02 January 2020, 05:09:12 PM), last modified 6 days ago (Thursday 05 March 2020, 10:09:24 AM).

Comments

(/COMP9315/20T1/forums/search?forum_choice=resource/40353)

(/COMP9315/20T1/forums/resource/40353)



Guodong Chen (/users/z5224156) 28 days ago (Wed Feb 12 2020 14:10:58 GMT+1100 (Australian Eastern Daylight Time))

Hi, John. Are we required to write lots of coding in C or C++?

Thanks.

Edit Delete



Tao Xue (/users/z5219280) 24 days ago (Sun Feb 16 2020 23:20:16 GMT+1100 (Australian Eastern Daylight Time))

I think the language is C. Having some understanding on C++ is better. I think so.

Edit Delete



John Shepherd (/users/z9300035) 24 days ago (Sun Feb 16 2020 12:29:30 GMT+1100 (Australian Eastern Daylight Time)), last modified 24 days ago (Sun Feb 16 2020 12:30:30 GMT+1100 (Australian Eastern Daylight Time))

C ... PostgreSQL is written in C

And lots of C code ... don't do the course if your C programming is not strong.

Edit Delete



Shaowei Ma (/users/z5238010) 25 days ago (Sat Feb 15 2020 16:14:18 GMT+1100 (Australian Eastern Daylight Time))

As said in assumed knowledge, I think so

Edit Delete