Course Outline Public

Edit Resource (/COMP3311/19T3/resources/25116/edit)

Contents

- Course Details
- Course Summary
- Assumed Knowledge
- Student Learning Outcomes
- Teaching Strategies
- Teaching Rationale
- Student Conduct
- Assessment
- Course Schedule
- Resources for Students
- Course Evaluation and Development

Course Details

Course Code	COMP3311
Course Title	Database Systems
Convenor	John Shepherd (/users/z9300035)
Admin	Hayden Smith (/users/z3418003)
Classes	**Lectures** : ...fill in times/locations of lectures... Timetable for all classes (/COMP3311/19T3/timetable)
Consultations	...coming soon ...
Units of Credit	6
Course Website	http://cse.unsw.edu.au/~cs3311/19T3/ (http://cse.unsw.edu.au/~cs3311/19T3/)
Handbook Entry	http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP3311.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP3311.html)

Course Summary

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. Hence, knowledge of how to manage them is clearly important to the IT industry. In the context of further study, understanding how to use databases effectively is essential for courses such as COMP9321 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP9321.html) Data Services Engineering and COMP9322 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP9322.html) Software Service Design and Engineering. COMP3311 also provides a foundation for further study in advanced database topics, such as COMP9315 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP9315.html) Database Systems Implementation and COMP9318 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP9318.html) Data Warehousing and Data Mining. Database concepts are also relevant in courses such as COMP9319 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP9319.html) Web Data Compression and Search and COMP6714 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP6714.html) Information Retrieval and Web Search.

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

Assumed Knowledge

The official pre-requisite for this course is that students have taken either COMP2521 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP2521.html) or COMP1927 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP1927.html) .

Whatever the formal pre-reqs, we assume primarily that students have some experience with procedural programming and some knowledge of elementary data structures.

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 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/INFS1603.html) Introduction to Business Databases or COMP1531 (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP1531.html) Software Engineering Fundamentals. On the other hand, half of the class hasn't seen this material before, so we need to cover it. 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 have seen them previously.

Student Learning Outcomes

After completing this course, students will:

1. develop accurate, non-redundant data models;
2. realise data models as relational database schemas;
3. formulate queries via the full range of SQL constructs;
4. use stored procedures and triggers to extend DBMS capabilities;
5. write applications in Python that interact effectively with databases;
6. analyze performance issues in relational database applications;
7. understand the overall architecture of relational DBMSs;
8. understand the concepts behind transactions and concurrency control;
9. explain how query and transaction processing techniques work in RDBMSs

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 capabilities:

Graduate Capability	Acquired in
Scholars capable of independent and collaborative enquiry, rigorous in their analysis, critique and reflection, and able to innovate by applying their knowledge and skills to the solution of novel as well as routine problems	Exercises, Prac work, Assignments
Entrepreneurial leaders capable of initiating and embracing innovation and change, as well as engaging and enabling others to contribute to change	Assignments
Professionals capable of ethical, self- directed practice and independent lifelong learning	Exercises, Prac work, Assignments
Global citizens who are culturally adept and capable of respecting diversity and acting in a socially just and responsible way	Assignments

Teaching Strategies

- Lectures ... introduce concepts, show examples
- Theory Exercises ... provide additional examples to reinforce lectures
- Prac Exercises ... introduce technology required for the assignments
- Assignments .. allow students to solve significant problems
- Videos ... bite-size explanations of various concepts

Lectures/Theory+Prac Exercises

Each week there will be four hours of lectures during which theory, practical demonstrations and case-studies will be presented. You will get maximum benefits from lectures if you read the relevant textbook chapters before attending each lecture.

There are no tutorial or lab classes in this course. Tutorial-type questions (aka theory exercises) and practical exercises will be made available. You are expected to work through these yourself. Note: none of these are assessable, but we assume 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.

If you run into problems with the theory questions, ask via the forum on the page containing the exercise. If you encounter problems with the practical exercises, you can either ask via the forum or by attending a Help Session.

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 components of a database application using Python. The assignments contribute 30% of the overall mark for this course.

#	Description	Due (tentative)	Marks
1	Collaborative ER Design	Friday 4 October (end week 3)	5%

2	SQL and PLpgSQL	Sunday 27 October (end week 6)	12%
3	Python and SQL	Sunday 17 November (end week 9)	13%

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, and interaction of programming languages and DBMSs. Completion of the assignments will also aid in your understanding of database performance issues. The assignments also lay the foundation for successful performance in the final exam.

Assignments are completed *individually* ; this means that you should do them *yourself* without assistance from others, except for asking advice from the Lecturer or Course Admin. As noted above, assignments 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 `give` or WebCMS) before midnight on the due date. Assessment of assignments will be based on how accurately they satisfy the requirements, as determined by automatic marking. We provide testing harnesses for the second and third assignments so that you can determine whether your code is working correctly. The supplied tests will use one instance of the database; the auto-marking will use these same tests, but will also run tests using one or more different database instances.

Late submissions will have marks deducted from the maximum achievable mark at the rate of 0.5% of the total mark per hour that they are late (i.e., approx 15% off the maximum assignment mark per day).

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, a some time will be spent during lectures to work through exercises, to illustrate the practice of using various database techniques and technologies. However, the primary learning focus in this course is assignment work, which has been designed to be challenging and relevant (i.e., dealing with real problems and using realistic database schemas).

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.

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 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. 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)

Assessment

Item	Topics	Due	Marks	Contributes to
Quizzes	All topics	Weeks 2,4,6,8,10	10%	1,2,...
Assignment1	Data Modelling	Week 3	5%	1,...
Assignment2	SQL and PLpgSQL	Week 6	12%	1,...
Assignment3	Python and SQL	Week 9	13%	2,...
Final Exam	All topics	Exam period	60%	1,2,...

Assessment

There will be three places where your learning in this course will be assessed: assignments, quizzes and the final exam.

Much as we dislike conflating the *learning* aspect of assignments with their *assessment* aspect, there will be marks for the assignment work. We 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 (but before indicated due dates), 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 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 final mark will be computed and how the hurdle will be enforced:

```
ass1    = mark for assignment 1      (out of 5)
ass2    = mark for assignment 2      (out of 12)
ass3    = mark for assignment 3      (out of 13)
quizzes = mark for on-line quizzes   (out of 10)
examP   = mark for exam (practical)  (out of 30)
examW   = mark for exam (theory)     (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
```

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 following is an **approximate** guide to the sequence of topics in this course. It is subject to change as the semester progresses.

Week	Lectures	Assignments	Quizzes
1	Introduction, Data Modelling, ER Model		
2	Relational Model, ER-Relational Mapping, SQL Schemas		Quiz 1
3	Data Modification, Simple SQL Queries	Assignment 1 Ends	
4	Complex SQL Queries, Stored Procedures		Quiz 2
5	PLpgSQL, Aggregates, Triggers		
6	No lectures	Assignment 2 Due	Quiz 3
7	Programming with Databases, Python and psycopg2		
8	Relational Design Theory, Normalisation		Quiz 4
9	Relational Algebra, Query Processing	Assignment 3 Due	
10	Transactions, Concurrency, Course Review		Quiz 5

Resources for Students

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

- Fundamentals of Database Systems (http://wps.aw.com/aw_elmasri_fundatasys_7) , Elmasri and Navathe, 7th edition, 2016, Addison-Wesley
- Database System Concepts (http://db-book.com/) , Silberschatz, Korth, Sudarshan, 6th edition, 2010, McGraw-Hill
- Database Management Systems (http://www.cs.wisc.edu/~dbbook/) , Ramakrishnan and Gehrke, 3rd edition, 2003, McGraw-Hill
- Database Systems: The Complete Book (http://www-db.stanford.edu/~ullman/dscb.html) , Garcia-Molina, Ullman, Widom, 2nd edition, 2008, 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

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

The textbooks give the greatest detail on the topics covered in the course, but cover many other topics. The Course Notes have less detail, but cover exctly the course syllabus. The Lecture Slides have even less detail, but include the examples discussed in lectures.

The software systems to be used in this course are PostgreSQL, SQLite, Python and psycopg2. The documentation and manuals provided with PostgreSQL (https://www.postgresql.org/docs/11/index.html) and Python (https://docs.python.org/3.7/) are actually very good, and the SQLite (http://www.sqlite.org/docs.html) and psycopg2 (http://initd.org/psycopg/docs/) 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 (http://www.postgresql.org/docs/books/) , SQLite (http://www.sqlite.org/books.html) , and Python (https://www.learnpython.org/) .

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, we have found O'Reilly books tend to be better than most.

PostgreSQL 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, SQL Server, 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 (although the developer base has expanded in recent years).

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 will be expecting whatever SQL statements you write for this course to work on both PostgreSQL and SQLite (although this may not always be possible).

For programming interaction with the database, we will be using Python via psycopg2.

Course Evaluation and Development

This course is evaluated each session using the myExperience system at the end of the semester.

The evaluation from the previous offering of the course showed that the course managed to fit the 10-week timeframe ok. Tutorials were criticised as not being particularly helpful. We will try Help Sessions as a less formal replacement for tutorials and run some tutorial-like activities in lectures.

Students are also encouraged to provide informal feedback during the semester, and let the lecturer in charge or the course admin know of any problems, as soon as they arise. Suggestions will be listened to very openly, positively, constructively and thankfully, and every reasonable effort will be made to address them as soon as possible.

Resource created about a year ago (Friday 15 March 2019, 01:57:25 PM), last modified 7 months ago (Saturday 14 September 2019, 06:02:46 PM).