Course Outline

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	Raymond Wong (/users/z3002286)
Classes	**There are no on-campus classes in COMP3311 in 21T1.** The lectures will run as pre-recorded videos and online overview/problem-solving sessions. All labs and tutorials will also be held live online via Blackboard Collaborate. Timetable for all classes (/COMP3311/21T1/timetable)
Consultations	Schedule via email to cs3311@cse.unsw.edu.au
Units of Credit	6
Course Website	https://webcms3.cse.unsw.edu.au/COMP3311/21T1/ (https://webcms3.cse.unsw.edu.au/COMP3311/21T1/) or http://cgi.cse.unsw.edu.au/~cs3311/21T1/ (http://cgi.cse.unsw.edu.au/~cs3311/21T1/)
Handbook Entry	http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP3311.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP3311.html)
Issues	email cs3311@cse.unsw.edu.au (mailto:mailto:cs3311@cse.unsw.edu.au) (mailto:mailto:cs3311@cse.unsw.edu.au)
Student Reps	email stureps@cse.unsw.edu.au ... to raise major issues about the course

If you encounter any problems with the course or the way it's being run, contact the class account and we will try to fix the problem.

Course Summary

This course aims to explore in-depth the practice of developing database applications and the theory behind relational database systems. It will also give a very brief overview of the technologies used in implementing database management systems and the past, present, and future of database systems.

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 and efficient) 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.

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 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

On successfully completing this course, students should be able to:

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;

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	assignments, prac work
Entrepreneurial leaders capable of initiating and embracing innovation and change, as well as engaging and enabling others to contribute to change	-
Professionals capable of ethical, self-directed practice and independent lifelong learning	assignments, prac work
Global citizens who are culturally adept and capable of respecting diversity and acting in a socially just and responsible way	-

Teaching/Assessment Strategies

Summary of teaching strategies in COMP3311 21T1. Everything is online.

- Pre-recorded lectures (videos + slides) ... introduce concepts
- Live lectures (online topic overview / problem-solving sessions) ... provide overview / work through examples
- Tutorials (written ex) ... reinforce concepts and provide additional examples
- Labs (prac ex) ... introduce technology required for the assignments
- Assignments .. allow students to solve significant problems
- Quizzes ... review concepts from previous week
- Forums ... ask questions, get answers
- Live lectures / consultations ... ask questions (live), get answers

Pre-recorded Lectures

The COMP3311 lectures will consist of pre-recorded videos and live online lectures. Most course content will be made available via pre-recorded videos and accompanying slides (made by Dr. John Shepherd). Each video will be 20-40 minutes in duration and will cover a single topic. Videos will be available at the start of each week. There will be a few topics per week (all the pre-recorded videos in total per week should be around 2-3 hours). Students are expected to watch these videos in one of the timetabled lecture slots or before then.

Online Overview / Problem-solving Sessions (Live online lectures)

In one of the timetabled lecture slots, we will be running live online lectures where we will overview the topics and their background, and/or work through exercise problems, related to the content videos for that week. These live online sessions are intended to be interactive, so please ask questions or pose alternative solutions. These overview / problem-solving sessions will be recorded. The first live lecture will start from the first timetabled lecture Monday 9am in week 1. For the subsequent weeks, we will run the live lectures on

Wednesday 16:00 (the second timetabled lecture slot) instead of the Monday 9am slot. This is because from last year's experience, many students had difficulty attending 9am lectures (esp. those from other Asian countries in different time zones).

Labs (Prac Exercises) & Tutorials (Written Exercises)

The MyExperience feedback from the previous offerings of the course showed that having occasional labs and tutorials, where problems with the prac and written exercises can be discussed, would be useful for the course. Labs would be helpful at the beginning when everyone needed to set up the database and perhaps query the database the first time. On the other hand, tutes would be more useful during the second half of the course when the materials would focus more on the theories and the fundamental concepts behind a database system. Therefore, a 2-hour lab will be run every week (starting from week 2) in the first half of the course, and a 1-hour tute per week will be in the second half (starting from week 7).

Prac exercises are available on the course web site. You should work through each exercise according to its schedule, since exercises are often a pre-cursor to the assignment work that follows.

There will be a set of tutorial problems available each week, covering topics from the previous week. During the tute in the second half of the course, you can either run a Q&A with the tutor or work through some of the problems together. Discussion is encouraged. Solutions will be made available to the tutorial problems two weeks after they are released.

Assignments

In the assignments, you will consider the major stages of developing a database application, understanding the requirements, formulating SQL queries, querying and updating a database, testing a database application against its requirements, and building components of a database application using Python. The assignments contribute 40% of the overall mark for this course.

Assignments are completed *individually* ; this means that you should do them *yourself* without assistance from others, except by asking advice from the Lecturer or Course Admin or your Tutor. Assignments are a 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. The assignments also lay the foundation for successful performance in the final exam.

Assignments are to be submitted on-line (via `give` or WebCMS) before the due date stated in the assignment specifications. Assessment of assignments will be based on how accurately they satisfy the requirements, as determined by automatic marking. We provide sanity testing for the second assignment so that you can determine whether your code is working correctly. The supplied tests will use one instance of the database; the auto-marking may use different tests, and may also run on one or more different database instances.

Do not submit your quiz answers late (as there are heavy late penalties). Late submissions will have marks deducted from the maximum achievable mark at the rate of 2 percent per hour that they are late (i.e., 48 percent off the maximum assignment mark per day).

Quizzes

Each week, starting in Week 2, there will be a quiz on topics from the previous week. This gives you a chance to review what you've learned on those topics. Quizzes are released on Monday mornings and are due before midnight on the following Friday (i.e. 5 days later). Each quiz will have 4 questions.

Do not submit your quiz answers late (there are heavy late penalties), or forget to submit a quiz. Late submissions will have marks deducted from the attained mark at the rate of 2 percent per hour that they are late. Submissions will not be accepted after 2 days late.

Forums/Consultations/Labs/Tutes

You can make use of the forums to help you with quick questions regarding prac work, tute exercises and assignments. If you want your question answered, you need to provide details (e.g. what you tried, what you observed, the context where you tried it). It is difficult to provide assistance without enough information on what you're doing. In particular, problems in setting up and running your PostgreSQL server are very difficult to diagnose with a lot of context. Such questions are probably best handled in your lab session, where you can share your screen.

One thing you should *not* post on the forum are examples of assignment work (e.g. "Tried this query for Q3, but it gave me this unexpected output"). Save such questions for labs/tutes/consultations (e.g., via direct messages).

Final Exam

The Final Exam will be held online. The paper will be made available via Webcms3 and submission will be via give. The paper itself will be primarily based on SQL and PLpgSQL. It should take approximately 3-hours to complete. The exam will be open for a slightly longer period. Further details will be provided later.

Note: anyone caught colluding, collaborating, posting exam solutions onto any internet platform, reading solutions from any internet platform, will be reported to the Student Conduct Investigation Unit, and will most likely end up with a mark of 00FL for the course, if not exclusion from further study at UNSW.

Teaching Rationale

COVID has thrown up some challenges in running CSE courses. We have adopted the above strategies because we believe they are the best way for helping you to learn in a completely online environment.

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

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 held online, 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 at least 45% (22.5 out of 50) on the exam. Note that partial marks are available for all questions on the final exam, and the hurdle will be enforced after any required scaling.

Item	Topics	Due	Marks	Learning outcomes
Quizzes	All topics	Weeks 2-10	10%	1,2,3,4,5,6,7,8
Assignment 1	SQL/PLpgSQL	Week 5	20%	3,4
Assignment 2	Python/SQL	Week 9	20%	5
Final Exam	All topics	Exam period	50%	1,2,3,4,5,6,7,8

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:

```
quizzes = mark for on-line quizzes   (out of 10)
ass1    = mark for assignment 1      (out of 20)
ass2    = mark for assignment 2      (out of 20)
exam    = mark for final exam        (out of 50)
okExam  = exam >= 22.5               (after scaling)
mark    = ass1 + ass2 + quizzes + exam
grade   = HD|DN|CR|PS  if mark >= 50 && okExam
        = FL           if mark <  50 && okExam
        = UF           if !okExam
```

Course Schedule

The following is a tentative course schedule and may be changed depending on the class progress.

Week	Lectures	Tutes	Prac Work	Assignments	Quizzes	Notes
1	Course intro, Data Modelling, ER diagrams	-	01: SQLite	-	-	-
2	Relational Model, SQL DDL, Mapping ER to SQL	Data modelling, ER	02: PostgreSQL	-	Quiz 1	-
3	SQL (i)	ER -> SQL	03: Schema & constraints	-	Quiz 2	-
4	SQL (ii), PLpgSQL(i)	SQL	04: SQL (i)	-	Quiz 3	-
5	PLpgSQL (ii), Triggers, Aggregates	PLpgSQL functions	05: SQL (ii)	Ass1 due	Quiz 4	-
6	Flexibility Week	-	-	-	-	-
7	DB/PL interaction, Python, Psycopg2	Triggers	-	-	Quiz 5	-
8	Functional dependencies, Normalization	Psycopg2	-	-	Quiz 6	-
9	Relational Algebra, Query execution, Tuning	Normalization	-	Ass2 due	Quiz 7	-
10	Transactions, Concurrency control, Database futures, Course review	Relational algebra	-	-	Quiz 8	-

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 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 exactly 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 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).

Course Evaluation and Development

This course is evaluated each session using the myExperience system.

In the previous offerings of this course, students noted that the Webcms3 forums were less than optimal and that it was difficult to get engaged (with other students) online. The Webcms3 forums had been rewritten to incorporate the requested features from other forum platforms like Piazza. This term will be the second time the new forum is used, so feedback is welcome.

The evaluation from the previous offerings of the course showed that having occasional labs and tutorials, where problems with the prac and written exercises can be discussed, would be useful for the course. To address this, this is the second time that we are running labs in the first half of the course, and tutorials in the second half. Labs would be helpful at the beginning when everyone needed to set up the database and

perhaps query the database the first time. On the other hand, tutes would be more useful during the second half of the course when the course materials focus more on the theories and the fundamental concepts behind database systems. Your feedback is particularly important.

Resource created 5 months ago (Monday 01 February 2021, 03:58:18 PM), last modified about a month ago (Tuesday 04 May 2021, 01:43:33 AM).

Comments

(/COMP3311/21T1/forums/search?forum_choice=resource/57063)

(/COMP3311/21T1/forums/resource/57063)

There are no comments yet.