Course Outline

COMP9311
Database Systems
Semester 2 2016

Contents

- Course Staff
- Course Details
- 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 Staff

Contact Details

Staff Name	Role	Email	Phone
Raymond Wong	Course Convenor	wong at cse.unsw.edu.au	55932
Mohammad Ebrahimi	Course Admininstrator	z5023577 at cse.unsw.edu.au	

Consultations

Staff Name	Day/Time	Location
Raymond Wong	Wednesday 1500-1600	K17-213
Mohammad Ebrahimi	Thursday 1800-1900	K17-201B

Course Details

Course Code: COMP9311
Course Title: Database Systems
Units of Credit: 6
Course WebSite: http://www.cse.unsw.edu.au/~cs9311/
Handbook Entry: http://www.handbook.unsw.edu.au/postgraduate/courses/2016/COMP9311.html

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. 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 Web Applications Engineering and COMP9322 Service-Oriented Architectures. COMP9311 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;
- 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

There is no official pre-requisite. However, COMP9021 is the course co-requisite. Therefore, before/during this course, students should/will have:

- experience with procedural programming; and
- an understanding of a range of in-memory searching data structures (e.g., binary search trees, and hash-tables).

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 lot of 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, using real database schemas).

Following the style of COMP3311, we recognise that students already have significant demands on their time, and formal classes are not necessarily the best way for students to learn. Lab exercises will be provided for you to practise in the computer lab.

Natually, sometimes students will "get stuck" on understanding various aspects of the material. Hints or solutions will be provided for difficult lab work during the lab/lecture. There will also be face-to-face consultations each week where you can ask questions, if you have a problem that can't be handled easily before or after each lecture.

Teaching Strategies

- Lectures: the main way to introduce concepts, show examples, show/discuss benefits of different techniques.
- Lab Work: lab exercises to practise the knowledge.
- Consultations: weekly consultation sessions to provide personalized advice / assistance to students.

The lectures will serve as the main method of delivering the course contents to the students. Lab exercises, a written assignment and programming assignments give students the hands-on experience on various subjects of the course. The consultation will give students the opportunity to seek assistance from the lecturer on all contents of the course.

Assessment

- 1 Written assignment: 25 points.
- 2 Programming assignments: 25 points and 50 points respectively.
- Final exam: 100 points.
- **Final Mark** : 2 * (Assgt1 + Assgt2 + Assgt3) * FinalExam / (Assgt1 + Assgt2 + Assgt3 + FinalExam)

The written assignment will be marked according to the quality of the solution as well as the amount of effort demonstrated by the student. The programming assignments will be based on the result of auto marking scripts, the quality of the programming code and documentation.

Late written assignments will receive **Zero** marks.
Late programming assignments will attract a 10% penalty for the first day and a 30% penalty for each subsequent day.

Academic Honesty and Plagiarism

Plagiarism is the presentation of the thoughts or work of another as one's own.*

Examples include:

- direct duplication of the thoughts or work of another, including by copying material, ideas or concepts from a book, article, report or other written document (whether published or unpublished), composition, artwork, design, drawing, circuitry, computer program or software, web site, Internet, other electronic resource, or another person's assignment without appropriate acknowledgement;
- paraphrasing another person's work with very minor changes keeping the meaning, form and/or progression of ideas of the original;
- piecing together sections of the work of others into a new whole;
- presenting an assessment item as independent work when it has been produced in whole or part in collusion with other people, for example, another student or a tutor; and
- claiming credit for a proportion a work contributed to a group assessment item that is greater than that actually contributed.**

For the purposes of this policy, submitting an assessment item that has already been submitted for academic credit elsewhere may be considered plagiarism.

Knowingly permitting your work to be copied by another student may also be considered to be plagiarism.

Note that an assessment item produced in oral, not written, form, or involving live presentation, may similarly contain plagiarised material.

The inclusion of the thoughts or work of another with attribution appropriate to the academic discipline does not amount to plagiarism.

The Learning Centre website is the main repository of resources for staff and students on plagiarism and academic honesty.

The Learning Centre also provides substantial educational written materials, workshops, and tutorials to aid students, for example, in:

- correct referencing practices;
- paraphrasing, summarising, essay writing, and time management;
- appropriate use of, and attribution for, a range of materials including text, images, formulae and concepts.

Individual assistance is available on request from The Learning Centre. Students are also reminded that careful time management is an important part of study and one of the identified causes of plagiarism is poor time management. Students should allow sufficient time for research, drafting, and the proper referencing of sources in preparing all assessment items.

* Based on that proposed to the University of Newcastle by the St James Ethics Centre. Used with kind permission from the University of Newcastle

** Adapted with kind permission from the University of Melbourne.

There are several on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- UNSW: Academic Integrity and Plagiarism
- 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 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
1	Introduction, Data Modelling, ER Notation

2	Relational Model, ER-Relational Mapping, SQL Schemas
3	DBMSs, Databases, Data Modification
4	SQL Queries
5	More SQL Queries, Stored Procedures, PLpgSQL
6	Extending SQL: Queries, Functions, Aggregates, Triggers
7	More Triggers, Programming with Databases
8	Catalogs, Privileges
9	Relational Design Theory, Normal Forms
10	Relational Algebra, Query Processing
11	Transaction Processing, Concurrency Control
12	Future Directions of Database Systems

Resources for Students

Textbook:

- Fundamentals of Database Systems, Elmasri and Navathe, 6th edition, 2010, Addison-Wesley

Reference books:

- Database System Concepts, Silberschatz, Korth, Sudarshan, 6th edition, 2010, McGraw-Hill
- Database Management Systems, Ramakrishnan and Gehrke, 3rd edition, 2003, McGraw-Hill
- 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

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 textbook and reference books give the greatest detail on topics covered in the course. The Lecture Slides are less detailed, but have the advantage that they cover precisely the syllabus (but no more). They also contain examples that will be discussed in detail. All 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) can be found under the **Documentation** link on the course web page. The manuals provided with PostgreSQL are actually very good, 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.

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.

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.

For programming interaction with the database, we will look briefly at some programming languages which are commonly used for interaction with databases (e.g. PHP, Java, Ruby, Python), but these will simply be for illustrative purposes; we won't expect you to use them in any assessable work.

Course Evaluation and Development

This course is evaluated each session using the CATEI system.

The CATEI evaluation from the last time showed that students were in general satisfied with all aspects of the course. COMP9311 will follow the style of COMP3311 -- to put more emphasis on practical work. Therefore, your feedback is important and will be considered seriously. Student feedback via the CATEI system will enable improvements to future offerings of this subject.