INFO2120: DATABASE SYSTEMS 1

Semester 1, 2017 | 6 Credit Points | Mode: Normal-Day
Coordinator(s): Bryn Jeffries

WARNING: This unit is an archived version! See Overview tab for delivered versions.

1. INTRODUCTION

The ubiquitous use of information technology leaves us facing a tsunami of data produced by users, IT systems and mobile devices. The proper management of data is hence essential for all applications and for effective decision making within organizations.

This unit of study will introduce the basic concepts of database designs at the conceptual, logical and physical levels. We will place particular emphasis on introducing integrity constraints and the concept of data normalization which prevents data from being corrupted or duplicated in different parts of the database. This in turn helps in the data remaining consistent during its lifetime. Once a database design is in place, the emphasis shifts towards querying the data in order to extract useful information. The unit will introduce different query languages with a particular emphasis on SQL, which is industry standard. Other topics covered will include the important concept of transaction management, application development with a backend database, an overview of data warehousing and OLAP.

2. LEARNING OUTCOMES

Learning outcomes are the key abilities and knowledge that will be assessed in this unit. See assessment summary table below for details of which outcomes are assessed where. Outcomes are listed according to the course goals that they support.

Design (Level 2)

1. Ability to design a schema which says how information about a particular domain will be stored in a relational DBMS (given a conceptual data model); also ability to apply normalization theory to evaluate or improve a relational schema. Ability to capture business rules as integrity constraints in a database schema.

Engineering/IT Specialisation (Level 2)

2. Understanding of the concept of a DBMS: differences from other ways to store and share data; DBMS role in organizations; the types of work done with a DBMS.
3. Ability to work with data stored in a relational database management system (understand table definitions including integrity constraints, extract information through SQL queries, modify information through SQL queries, use views and permissions for security)
4. Experience of how application software can use data stored in a DBMS (eg a dynamic content web site) and understand the basic architectural alternatives for data management applications.
5. Understanding of the basic concepts of transaction management.

Maths/Science Methods and Tools (Level 2)

6. Understanding of the relational data model
7. Ability to connect general database concepts to both theoretical abstract formulations, and details of specific software platforms.

Professional Conduct (Level 2)

8. Understanding of the SQL mechanisms for basic concepts of data security and privacy

Project and Team Skills (Level 2)

9. Work effectively in a team with members whose skills and interests differ

For further details of course goals related to these learning outcomes, see online unit outline at http://cusp.eng.usyd.edu.au/students/view-unit-page/alpha/INFO2120 .

3. ASSESSMENT TASKS

ASSESSMENT SUMMARY

Assessment name	Team-based?	Weight	Due	Outcomes Assessed
Weekly Homework	No	10%	Multiple Weeks	1, 2, 5, 6, 7
SQL Tutorials	No	0%	Multiple Weeks	3, 6
SQL Quiz	No	10%	Week 9	3, 6
DB Design	Yes	10%	Week 5	1, 3, 6, 9
DB Schema	Yes	10%	Week 7	1, 4, 6, 8, 9
DB Programming	Yes	10%	Week 12	4, 5, 7, 8, 9
Final Exam	No	50%	Exam Period	1, 2, 3, 5, 6, 7, 8

ASSESSMENT DESCRIPTION

* indicates an assessment task which must be repeated if a student misses it due to special consideration

NOTE: All text-based written assignments make use of text-based similarity detecting software (Turnitin)

DB concept exercises: Each week a set of questions will be posed on that week's topic, based on the recommended reading and supplied

resources. Answers are submitted online and are mostly limited to simple formats such as multiple choice. Students can review their answers and will receive an overall score based upon their best 8 out of 10 weeks' submissions.

DB design assignment: Students work together in small groups to model a database based upon a brief scenario description, generating an conceptual Entity-Relationship diagram and then mapping this to a relational model. Students then implement this database in PostgreSQL.

DB application development assignment: The theme of the previous assignment is extended with the development of a client interface for the database, and the implementation of more advanced back-end features such as stored procedures and indexes.

SQL: Students work through weekly online tutorials introducing increasingly sophisticated usage of SQL. Solutions are provided for each week, and the topics are assessed in an SQL quiz.

Final Exam: Understanding of all of this unit`s material is reviewed in a written examination.

ASSESSMENT FEEDBACK

SQL tutorials provide simple feedback and allow multiple attempts, and example solutions are available after the submission deadline has passed.

Homework exercises include solutions after the submission deadline.

Design and Application assignments are worked on progressively and draft submissions can receive formative feedback that can be used to make an improved final submission.

ASSESSMENT GRADING

Final grades in this unit are awarded at levels of HD for High Distinction, DI (previously D) for Distinction, CR for Credit, PS (previously P) for Pass and FA (previously F) for Fail as defined by University of Sydney Assessment Policy. Details of the Assessment Policy are available on the Policies website at http://sydney.edu.au/policies . Standards for grades in individual assessment tasks and the summative method for obtaining a final mark in the unit will be set out in a marking guide supplied by the unit coordinator.
It is a policy of the School of Information Technologies that in order to pass this unit, a student must achieve at least 40% in the written examination. For subjects without a final exam, the 40% minimum requirement applies to the corresponding major assessment component specified by the lecturer. A student must also achieve an overall final mark of 50 or more. Any student not meeting these requirements may be given a maximum final mark of no more than 45 regardless of their average.
Students are required to pass the SQL Challenge milestone activities between Week 4 and Week 10 in order to pass this subject. Students with less than 40% in SQL Challenge will be assessed individually on their SQL skills by the lecturer whether they pass this milestone.

4. ATTRIBUTES DEVELOPED

Attributes listed here represent the course goals designated for this unit. The list below describes how these attributes are developed through practice in the unit. See Learning Outcomes and Assessment sections above for details of how these attributes are assessed.

Attribute	Method
Design (Level 2)	Schema design including integrity constraints that capture business rules as part of tutorial work and online quizzes/exam. Design of database-backed application to meet user needs, in the Practical Assignment.
Engineering/IT Specialisation (Level 2)	Theory and Practice of Relational Database Management Systems, throughout the unit
Maths/Science Methods and Tools (Level 2)	In-depth coverage of the relational data model and introduction to conceptual modelling using entity-relationship model and UML.
Information Seeking (Level 2)	Learning of database systems and SQL editors through online documentation for the lab tasks; topics on information security and integrity as part of the lecture.
Professional Conduct (Level 2)	The subjects covers basics of best-practices for SQL coding, schema naming conventions, database security, authorization mechanisms, and protection against SQL injection attacks.
Project and Team Skills (Level 2)	Team assignment (with disparate members), in the practical database project assignments.

For further details of course goals and professional attribute standards, see the online version of this outline at http://cusp.eng.usyd.edu.au/students/view-unit-page/alpha/INFO2120 .

5. STUDY COMMITMENT

A variety of learning situations will be employed during the unit of study, including lectures, on-line demos, tutorials, directed computer laboratory exercises, self-learning SQL exercises, assessed assignments and a small practical database project. To benefit fully from this unit it is necessary to participate fully in all aspects of the unit of study.

Pre-Lecture Preparation: Introductory content will be released each week in advance of the main lecture. All students are expected to review this content and answer review questions prior to the main lecture.

Independent Study: Preparatory study including introductory videos, reading reference material, attempting online homework questions and SQL tutorial questions.

Project work: Group activities working on assignment projects, mostly performed outside of classes.

Lectures: Attendance is expected, and students are required to take part in worksheet-based activities and in-class online questionnaires.

Laboratory: Laboratory are facilitated by tutors, and expose students to hands-on use of a DBMS and other practical activities relating to assignment projects.

Activity	Hours per Week	Sessions per Week	Weeks per Semester
Pre-Lecture Preparation	1.00	1	13
Lecture	1.00	1	13
Laboratory	2.00	1	12
Independent Study	5.00		13
Project Work - own time	3.00		3

Standard unit of study workload at this university should be from 1.5 to 2 hours per credit point which means 9-12 hours for a normal 6 credit point unit of study. For units that are based on research or practical experience, hours may vary. For lecture and tutorial timetable, see University timetable site at: web.timetable.usyd.edu.au/calendar.jsp

6. TEACHING STAFF AND CONTACT DETAILS

COORDINATOR(S)

Name	Room	Phone	Email	Contact note
Dr Jeffries, Bryn			bryn.jeffries@sydney.edu.au	

LECTURERS

Name	Room	Phone	Email	Contact note
Dr Jeffries, Bryn			bryn.jeffries@sydney.edu.au	

7. RESOURCES

RECOMMENDED REFERENCES

Kifer, Bernstein & Lewis, *1. Database Systems: An Application-Oriented Approach (Introductory Version)* (2nd). Addison-Wesley, 2006. 0321228383.

Ramakrishnan & Gehrke, *2. Database Management Systems* (3rd). McGraw-Hill, 2003. 0071230572.

Ullman & Widom, *3. First Course in Database Systems* (3rd). Prentice-Hall, 2008. 013600637X.

Silberschatz A., Korth H., & Sudarshan S., *4. Database Systems Concepts* (5th). McGraw-Hill, 2006. 0072958863.

COURSE WEBSITE(S)

University of Sydney LMS (http://elearning.sydney.edu.au/) will be used as the main gateway to all resources, including:

- Lecture slides

- Lecture recordings

- Introductory videos

- Activity Worksheets and solutions

Discussion forums and SQL tutorials are run through separate systems, but linked from the eLearning site.

NOTE ON RESOURCES

Some in-class activities may make use of students' own devices (smart-phones, tablets or laptops). It is not required to have a device, but students are encouraged to bring one to classes if they have one.

8. ENROLMENT REQUIREMENTS

ASSUMED KNOWLEDGE

None.

PREREQUISITES

INFO1003 OR INFO1103 OR INFO1903 OR INFS1000 OR DECO1012.

PROHIBITIONS

INFO2820 OR COMP5138 OR INFO2905.

9. POLICIES

ACADEMIC HONESTY

While the University is aware that the vast majority of students and staff act ethically and honestly, it is opposed to and will not tolerate academic dishonesty or plagiarism and will treat all allegations of dishonesty seriously.

All students are expected to be familiar and act in compliance with the relevant University policies, procedures and codes, which include:

- *Academic Honesty in Coursework Policy 2015*
- *Academic Honesty Procedures 2016*
- *Code of Conduct for Students*
- *Research Code of Conduct 2013* (for honours and postgraduate dissertation units)

They can be accessed via the University"s Policy Register: http://sydney.edu.au/policies (enter "Academic Honesty" in the search field).

Students should never use document-sharing sites and should be extremely wary of using online "tutor" services. Further information on academic honesty and the resources available to all students can be found on the Academic Integrity page of the University website: http://sydney.edu.au/elearning/student/EI/index.shtml

Academic Dishonesty and Plagiarism

Academic dishonesty involves seeking unfair academic advantage or helping another student to do so.

You may be found to have engaged in academic dishonesty if you:

- Resubmit (or "recycle") work that you have already submitted for assessment in the same unit or in a different unit or previous attempt;
- Use assignment answers hosted on the internet, including those uploaded to document sharing websites by other students.
- Have someone else complete part or all of an assignment for you, or do this for another student.
- Except for legitimate group work purposes, providing assignment questions and answers to other students directly or through social media platforms or document ("notes") sharing websites, including essays and written reports.
- Engage in examination misconduct, including using cheat notes or unapproved electronic devices (e.g., smartphones), copying from other students, discussing an exam with another person while it is in progress, or removing confidential examination papers from the examination venue.
- Engage in dishonest plagiarism.

Plagiarism means presenting another person's work as if it is your own without properly or adequately referencing the original source of the work.

Plagiarism is using someone else's ideas, words, formulas, methods, evidence, programming code, images, artworks, or musical creations without proper acknowledgement. If you use someone's actual words you must use quotation marks as well as an appropriate reference. If you use someone's ideas, formulas, methods, evidence, tables or images you must use a reference. You must not present someone's artistic work, musical creation, programming code or any other form of intellectual property as your own. If referring to any of these, you must always present them as the work of their creator and reference in an appropriate way.

Plagiarism is always unacceptable, regardless of whether it is done intentionally or not. It is considered dishonest if done knowingly, with intent to deceive or if a reasonable person can see that the assignment contains more work copied from other sources than the student's original work. The University understands that not all plagiarism is dishonest and provides students with opportunities to improve their academic writing, including their understanding of scholarly citation and referencing practices.

USE OF SIMILARITY DETECTION SOFTWARE

All written assignments submitted in this unit of study will be submitted to the similarity detecting software program known as **Turnitin**. Turnitin searches for matches between text in your written assessment task and text sourced from the Internet, published works and assignments that have previously been submitted to Turnitin for analysis.

There will always be some degree of text-matching when using Turnitin. Text-matching may occur in use of direct quotations, technical terms and phrases, or the listing of bibliographic material. This does not mean you will automatically be accused of academic dishonesty or plagiarism, although Turnitin reports may be used as evidence in academic dishonesty and plagiarism decision-making processes.

Computer programming assignments may also be checked by specialist code similarity detection software. The Faculty of Engineering & IT currently uses the MOSS similarity detection engine (see http://theory.stanford.edu/~aiken/moss/) . These programs work in a similar way to TII in that they check for similarity against a database of previously submitted assignments and code available on the internet, but they have added functionality to detect cases of similarity of holistic code structure in cases such as global search and replace of variable names, reordering of lines, changing of comment lines, and the use of white space.

IMPORTANT: School policy relating to Academic Dishonesty and Plagiarism.

In assessing a piece of submitted work, the School of IT may reproduce it entirely, may provide a copy to another member of faculty, and/or to an external plagiarism checking service or in-house computer program and may also maintain a copy of the assignment for future checking purposes and/or allow an external service to do so.

Other policies

See the policies page of the faculty website at http://sydney.edu.au/engineering/student-policies/ for information regarding university policies and local provisions and procedures within the Faculty of Engineering and Information Technologies.

10. WEEKLY SCHEDULE

Note that the "Weeks" referred to in this Schedule are those of the official university semester calendar https://web.timetable.usyd.edu.au/calendar.jsp

Week	Topics/Activities
Week 1	Introduction and Administrativa
Week 2	Conceptual Data Modeling
Week 3	The Relational Data Model
Week 4	Relational Algebra and SQL
Week 5	Complex SQL: Nested Queries and Grouping
	Assessment Due: DB Design
Week 6	Schema Refinement and Data Normalization
Week 7	Data Security and Integrity
	Assessment Due: DB Schema
Week 8	DB Application Development
Week 9	Transaction Management
	Assessment Due: SQL Quiz
Week 10	Indexing and Tuning
Week 11	Introduction to Data Warehousing and OLAP
Week 12	Further topics in database systems
	Assessment Due: DB Programming
Week 13	Revision
Exam Period	Assessment Due: Final Exam