COMP9311:
DATABASE SYSTEMS

Term 1 2024

Week 1 – Course Introduction

By Xiaoyang Wang, CSE UNSW

Disclaimer: the course materials are sourced from previous offerings of COMP9311 and COMP3311

Who's who in COMP9311

Lecturer in Charge

- Xiaoyang Wang
- Office: K17 501D
- Email: xiaoyang.wang1@unsw.edu.au

Course admin and Tutors

- Couse Admin: Xingyu Tan (xingyu.tan@unsw.edu.au)
- Tutors: Peiting Xie, Kaiyu Chen, Zeming Fei, Miao Ma, Qing Sima, Junhua Zhang, Yanping Wu, Qi Luo.
- Mostly research students from the Data and Knowledge Research Group
- Office: K17-201 closed area

Plan for the delivery of this course

Lectures

- describe all syllabus topics in some detail, with exercises and examples
- Start from Week 1: 4PM – 6PM (Mon) and 4PM – 6PM (Wed)
- In-person lecture and online streaming at the same time
- There will be admin/tutor to answer questions in the online chat panel
- In-person (O'Shane 104)
- Online streaming (Moodle -> Blackboard Collaborate)
- Recorded and uploaded to Echo360 (Moodle -> Lecture Recordings).

Online Lecture - BlackBoard Collaborate

Lecture Recordings - Monday

Lecture Recordings - Wednesday

Plan for the delivery of this course

Labs

- In-person, bring your own laptop, check your timetable
- Start from Week 2
- Guides you through the practical skills on the database application programming part of the course
- Not recorded.

Weekly Consultation

- Tutor present to answer any course related questions.
- 10:00 – 12:00 Friday via Blackboard Collaborate on Moodle

Plan for the delivery of this course

How to access Online Consultations

- Log into Moodle (https://moodle.telt.unsw.edu.au/).
- Go to course (COMP9311 – Database Systems 2024 T1).
- Click Blackboard Collaborate
- Click the corresponding consultation session to join.

Online Lecture - BlackBoard Collaborate

Practices questions

- Sample answers are provided.
- To be released on course website at every interval.

Support your learning

Couse Website

- https://webcms3.cse.unsw.edu.au/COMP9311/24T1/
- all course announcements, content and links to other tools/platforms

Online forum

- Ed forum (register yourself with https://edstem.org/au/join/mqzRT3)
- Tutors will visit the forum regularly to answer questions

Moodle

- access Echo360, submit assignment/project, MyExperience and Blackboard Collaborate

Email

- If you need any help, email me (xiaoyang.wang1@unsw.edu.au) or Xingyu (xingyu.tan@unsw.edu.au)

For Other Enrolment Issues

- The course enrolment process isn't something lecturers have direct control over.
- Matters such as the number of students that can take a course/lab, etc.
- Students always adjust their courses during prior to the census date.
- Checking daily for openings is still recommended.

Course Overview

This is an introductory level course of database systems.

We will be (mostly) learning:

- Theory behind relational database systems
- Practice of using relational database management systems

We will NOT be learning:

- How to build applications or websites over databases
- Design and implementation detail of databases

Syllabus Overview

Data modelling and database design (Week 1 to Week 3)

- ER model, ER-to-relational
- Relational model (relational algebra), mapping of ER to relational model

Database development (Week 3 to Week 5)

- SQL, views, stored procedures, triggers, aggregates
- PostgreSQL: psql (an SQL shell), PLpgSQL (procedural)
- PostgreSQL –> lab, project environment
- Functional dependencies

Syllabus Overview

Formal DB design theory and DBMS architecture (Week 7 – Week 10)

- Normalisation, functional dependencies
- DBMS architecture: client/server, file system, relational engine
- Storage and indexing, data access operations
- Query processing: translation, optimisation, evaluation
- Transaction processing: transactions, concurrency control, recovery
- Future of Databases, limitations of RDBMS's, modern/future technologies
- Course revision

Course Assessments

Two Assignments

- Ass 1: Data Modelling + Relational Algebra (week 2-4)
- Ass 2: DB Design Theory + Database Storage Structures + Transaction (week 8-10)

One Project

- Proj 1: SQL PLpgSQL (week 5-7)

Assignments and project are all individual work

Some are automarked (so you must follow the specification exactly)

One Exam

- Date to be decided later

Course Assessments

COMP9311 24T1 Assessment Summary

Number	Name	Full Mark
1	Assignment 1: Data Modelling + Relational Algebra	24
2	Assignment 2: DB design Theory + Transaction	26
3	Project 1	50
4	Final Exam	100

The equation for your final mark calculated by **Geometric Mean**

- Final Mark = sqrt ((ass1 + ass2 + proj1) * Final Exam)

5% reduction per day for assignments and project

- 0 marks after 5 days late
- **1 second late = 1 day late**
- Submit wrong files = Late
- Double check to make sure your submission is correct and on time!

Special Consideration

- We will grant no-penalty extensions due to extreme circumstances (e.g., medical emergencies)
- Apply via myUNSW as soon as possible
- Evidence is needed, application process and details in https://www.student.unsw.edu.au/special-consideration
- No other excuses are accepted (e.g., network down, too busy, forgot to submit)

Exam

Current plan is to do an online exam, comprising a mixture of week 1–10

Everyone gets exactly **one chance** to pass the Exam

If you attend the Exam:

- I assume that you are fit/healthy enough to take it (Fit-to-Sit rule at UNSW)
- no 2nd chance exams, even with a medical certificate

All Special Consideration requests

- must *document* how you were affected
- must be submitted via myUNSW
- Supplementary Exams are held shortly after the exam period; so be prepared to be available (!)

Plagiarism

★ We adopt a **zero-tolerance** policy for plagiarism.

All submissions are checked for plagiarism. The university regards plagiarism as a form of academic misconduct and has very strict rules regarding plagiarism.

For UNSW policies, penalties, and information to help avoid plagiarism, please see: https://student.unsw.edu.au/plagiarism. *Not knowing the rules is not considered a valid excuse.*

All assessments must be your own original work. They are NOT group project.

DO NOT: copy from others, copy from the Internet, pay someone to do it.

Be careful using ChatGPT or other AI tools!

https://www.student.unsw.edu.au/notices/2023/02/academic-integrity-reminder-chatgp

ChatGPT is not Reliable

Example:

- Hi ChatGPT, if you have 2L, 5L and 7L jugs, how to measure 2L milk?

ChatGPT is not Reliable

K

Hi ChatGPT, if you have 2L, 5L and 7L jugs, how to measure 2L milk?

You can measure exactly 2 liters of milk using 2-liter, 5-liter, and 7-liter jugs with the following steps, commonly known as the water jug problem:

1. Fill the 5-liter jug completely.
2. Pour the contents of the 5-liter jug into the 7-liter jug. This leaves you with 2 liters of milk in the 5-liter jug.
3. Empty the 7-liter jug.
4. Pour the 2 liters of milk from the 5-liter jug into the 7-liter jug.
5. Fill the 5-liter jug again.
6. Carefully pour the contents of the 5-liter jug into the 7-liter jug, which already has 2 liters of milk. This will leave you with 4 liters of milk in the 7-liter jug, as the 7-liter jug can hold up to 7 liters.

Now you have successfully measured out 2 liters of milk in the 7-liter jug.

Its response makes no sense at all!!

Student Learning Outcomes

Remark that: COMP9311 introduces foundations & technology of databases

- skills: how to build a database using DBMS
- theory: how do you know that what you built was good

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 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

What we expect from you

Things that we expect you to **do**:

- Follow the lecture content (i.e., weekly topics)
- Theory exercises: exercise/example questions (try them yourself)
- Prac work: lab exercises and project

You will show us your progress/learning outcomes through:

- Assignments: theoretical exercises
- Project: extended practical exercises
- Final exam: your learning outcomes on **both** practical and theoretical topics

What we expect from you

Importantly:

- We want you to feel welcome and safe in asking questions/help online
- This can only happen if we all behave respectfully towards each other when we interact online
- No judgement. Everybody in the class is here to learn something and everybody in the class will help each other to have the best learning experience.

Course Textbook

- ★ **Lecture notes will be sufficient**

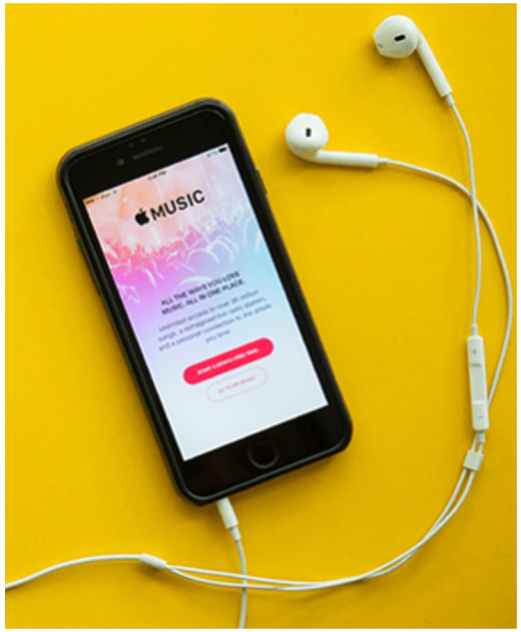
Reference Books:

- Elmasri & Navathe, Fundamentals of Database Systems, Benjamin/Cummings, 7th Edition, 2015.
- J. D. Ullman & J. Widom, A First Course in Database Systems, Prentice Hall, 1997.
- R. Ramakrishan, Database Management Systems, McGRAW-HILL, 1997.
- D. Maier, The Theory of Relational Databases, Computer Science Press, 1983.

Previous: Course Information

Next: Why study Databases

Why Study Databases?



Database Management System

Applications with Database management system

- Banking: transactions
- Airlines: reservations, schedules
- Universities: registration, grades
- Sales: customers, products, purchases
- Online retailers: order tracking, customized recommendations
- Manufacturing: production, inventory, orders, supply chain
- Human resources: employee records, salaries, tax deductions

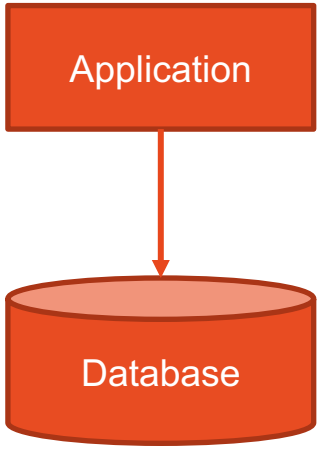
Databases touch all aspects of our lives

Why Study Databases?

All significant modern computer application has a large amount of data

In real-world applications, data will always have to be:

- stored (typically on a disk device)
- manipulated/accessed (efficiently, effectively)
- shared (by many users, concurrently)
- transmitted (all around the Internet)



Red points are handled by databases; blue by networks.

Database: Collection of related data that models some aspect of the real world.

★ Databases are the core component of most computer applications. ★

Database Example

Create a database that manages course enrolment for all UNSW students.

Things we need (simplified):

- Information about Students
- Information about Courses
- Information about course Selections

File Systems as Data Management?

File based system

- Contains various information on a storage device (hard disk)
- Files (such as TXT/CSV/EXCEL files, object files, source files)
- Stores files directly on the device and maybe in directories

STUDENT.csv (name, id, major)

"name","id","major"
"Smith","17","IT"
"Amy","8","IT"

COURSE.csv (code, name, department)

"code","name","department"
"COMP9311","Database Systems","CSE"
"COMP9312","Data Analytics for Graphs","CSE"

SELECTION.csv (student_id, course_code, term)

"student_id","course_code","term"
"17","COMP9311","2022T2"
"17","COMP9312","2022T2"
"8","COMP9311","2022T2"

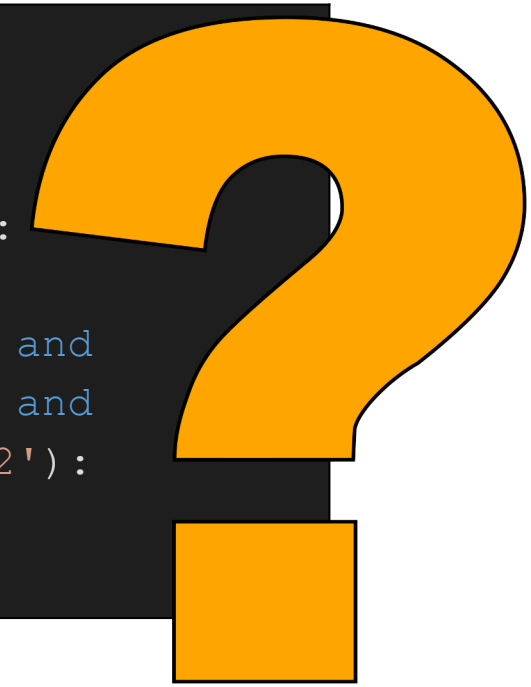
File Systems as Data Management?

Question: Does Amy select COMP9311 in 2022T2?

File Systems as Data Management?

Question: Does Amy select COMP9311 in 2022T2?

```
for line0 in open('STUDENT.csv'):
    student = parse(line0)
    if student[0] == 'Amy':
        for line1 in open('SELECTION.csv'):
            selection = parse(line1)
            if (selection[0] == student[1] and
                selection[1] == 'COMP9311' and
                    selection[2] == '2022T2'):
                return True
return False
```



Why Database Systems (1)

Drawbacks of using file systems to store data

- **Data redundancy and inconsistency**
  - Multiple file formats, duplication of information in different files
- **Difficulty in accessing data**
  - Would have to write a new program to carry out each new task
- **Integrity problems**
  - Integrity constraints (e.g., account balance >= 0) become “buried” in program code rather than being clearly kept and stated
  - Hard to add new constraints or change existing ones
- **Data isolation — multiple files and formats**

Why Database Systems (2)

Drawbacks of using file systems (cont.)

- **Atomicity of updates**
  - What is computer crashes?
  - Failures may leave the data in an inconsistent state.
  - Example: Transfer of funds from one account to another should either complete or not happen at all.
- **Hard to allow concurrent access by multiple users**
  - Uncontrolled concurrent accesses can lead to inconsistencies
  - Example: Two people reading a balance (say 100) and updating it by withdrawing money (say 50 each) at the same time

Database systems offer solutions to all the above problems.

Databases: Important Themes

The field of databases deals with:

- data ... representing application scenarios
- relationships ... amongst data items
- constraints ... on data and relationships
- redundancy ... one source for each data item
- data manipulation ... declarative, procedural
- transactions ... multiple actions, atomic effect
- concurrency ... multiple users sharing data
- scale ... massive amounts of data

What is Data?

Data (according to Elmasri/Navathe)

- Known facts that can be recorded (and have implicit meaning)

Example:

- Students – name, DOB, courses enrolled, etc.

  ... (who are also my thesis students)

- Movies – title, year published, director, etc.

  ... (that I like)

Two Types of Data

Data that is **Unstructured**

- No need to pre-define the data
- Requires expertise to prepare the data due to its non-formatted nature
- Can be a combination of various data

Data that is **Structured**

- Stored with a rigid and strict schema
- Can be organized into databases

What is Database?

Database (Elmasri/Navathe)

- ... a collection of related data ...
- Data items alone are relatively useless
- We need the data to have some structure

Example:

a student records database

- Contains information identifying students, courses they are enrolled in, results from past courses, ...

IMDb movie database

- Contains information about the movies, actors, theatres showing the movies, other movies the director also directed in, ...

What is Database Management System

Database can be manipulated and maintained by a database management system (DBMS)

According to Elmasri/Navathe:

- DBMS: ... a collection of programs that enables users to create and maintain a database ...
- Database system (in short): ... the database and DBMS together ...

Database Users

Database Administrator (DBA)

- Design of the conceptual and physical schemas
- Security and authorization
- Data availability and recovery from failures
- Database tuning

Application Programmer

- Implement the specific requirements
- E.g., Web Developer

End User

A Little Bit of History

- **Early 1960s:** First general-purpose DBMS, Integrated Data Store, by Charles Bachman (Turing Award)
- **Late 1960s:** IBM developed Information Management System (IMS) DBMS, adopting the hierarchical data model
- **1970s:** Edgar Codd (Turing Award), at IBM, proposed the relational model
- **1980s:** SQL became the standard. James Gray (Turing Award) presented the concepts of transaction
- **Late 1980s, 1990s:** ORACLE, DB2 by IBM, and POSTGRES by Michael Stonebraker (Turing Award). Data warehouse.
- **1998+:** NoSQL
- **Current:** Big data & large distributed data processing

Types of DBMS

- Relational
- Key/Value
- Graph
- Document
- Column-family

NoSQL

Top Database Management Systems

Rank			DBMS	Database Model
Feb 2023	Jan 2023	Feb 2022		
1.	1.	1.	Oracle	Relational, Multi-model
2.	2.	2.	MySQL	Relational, Multi-model
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model
4.	4.	4.	PostgreSQL	Relational, Multi-model
5.	5.	5.	MongoDB	Document, Multi-model
6.	6.	6.	Redis	Key-value, Multi-model
7.	7.	7.	IBM Db2	Relational, Multi-model
8.	8.	8.	Elasticsearch	Search engine, Multi-model
9.	↑ 10.	↑ 10.	SQLite	Relational
10.	↓ 9.	↓ 9.	Microsoft Access	Relational

Source: http://db-engines.com/en/ranking

Data Model

Data model: concepts used to describe the allowed structure of a database. i.e. the structure of the meta-data.

Levels of Data Models:

- **High-level or conceptual** (e.g. ER model – concerns entities, attributes and relationships)
- **Implementation or record-based** (e.g. Relational, Network, Hierarchical – that can be used to immediately derive a physical implementation)
- **Low-level or physical** (concerns record formats, access paths etc)

Data Model Concepts

Database Schema: a *formalism* of the data model, the *structural description* of what information will database holds.

Database Instance*: any combination of actual information populated* in the database at a particular time.

Workflow:

- We define a database by specifying its schema.
- The state is then an empty instance of the schema.
- To create the initial instance we load in data.
- After this, each change in state is an update to the instance.

Design a Database

Conceptual Design

- Requirements can be represented and manipulated using some computerized tools so that it can be easily maintained, modified, and transformed into a database implementation

Logical Design

- Translated by conceptual design that can be expressed in a data model implemented in a DBMS

Physical Design

- Further specifications are provided for storing and accessing the database

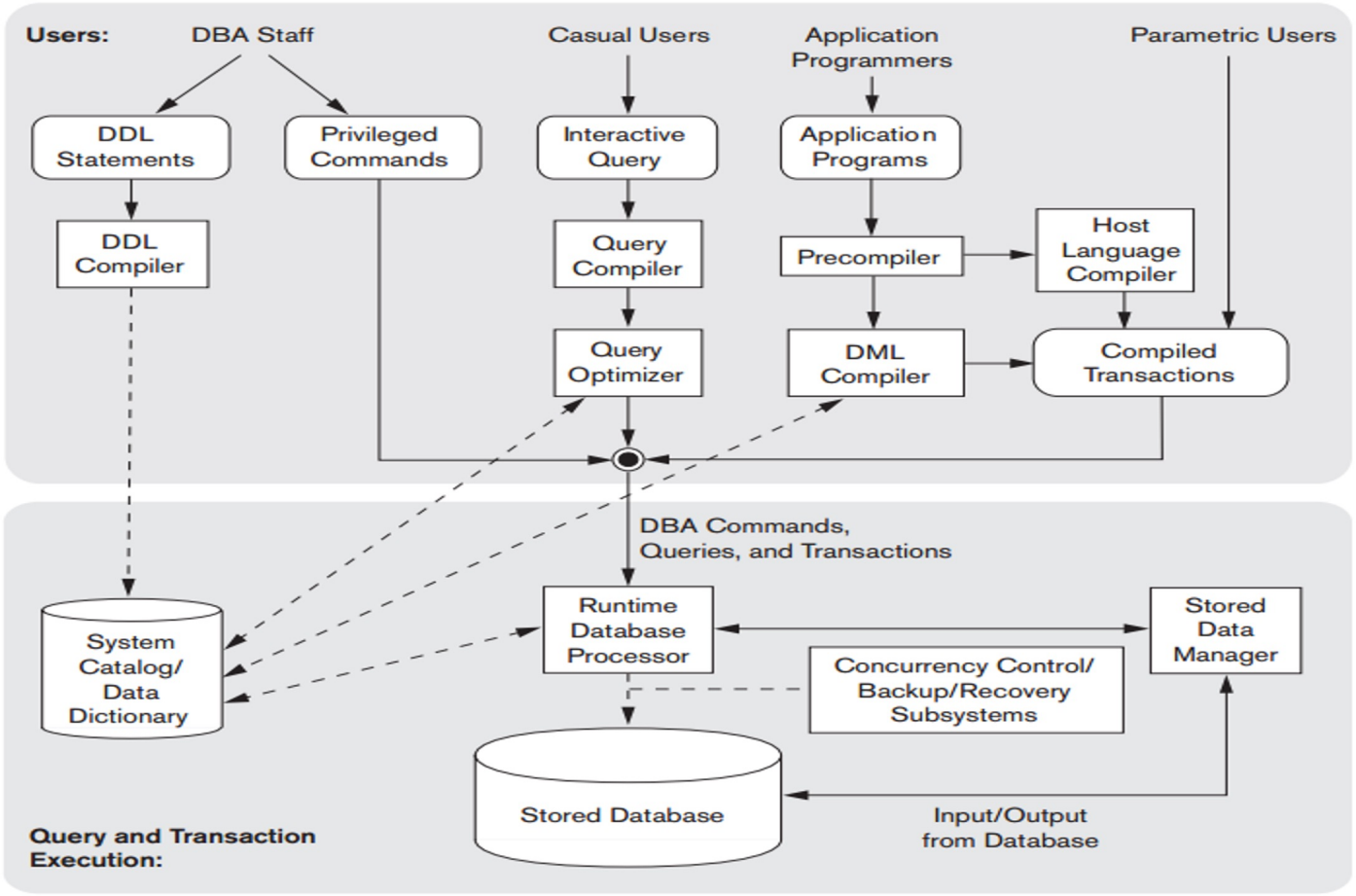
Database Languages

Data definition language (DDL): used to define the conceptual schema.

Data manipulation languages (DML): let users write requests to retrieve and manipulate data, as well as other tasks relating to data manipulation.

- Non-procedural DML (e.g., SQL, common for casual users)
  - interactive and/or embedded
  - set at a time/ set oriented.
- Procedural DML (also covered in this course)
  - embedded in a general-purpose language,
  - record at a time

Database System



Component modules of a DBMS and their interactions.

Learning Outcome

Hopefully, you now know…

- course structure,
- who to contact,
- how you're assessed and scored,
- the database applications around you,
- what goes on in databases (and is interested)

Next Lecture: Data Modelling, ER Diagram