# COMP9311
# DATABASE SYSTEMS


- Wenjie Zhang
- School of Computer Science and Engineering
- wenjie.zhang@unsw.edu.au
- http://www.cse.unsw.edu.au/~zhangw


2025 Term 3; Week 1.1

# Outline

- **Course Introduction**
- Database Introduction
- Database Features

# Course Schedules

**Lectures**:

- Hybrid Mode: In-person & Online (Moodle-> Blackboard Collaborate).
  - Tutors are available to answer online questions. Post your questions online in the chatbox.
- Starts Week 1.
- Recorded (Moodle -> Lecture Recordings).

**Labs**:

- In-person, bring your own laptop, check your timetable (Week 2-5, 7-10)
- Guides you through the practical skills on the database application programming part of the course
- Not recorded.

# Course Help

Course Website:

- https://webcms3.cse.unsw.edu.au/COMP9311/25T3/

Course Forum (Use Discourse forums):

- https://discourse01.cse.unsw.edu.au/25T3/COMP9311/
- Tutors will visit the forum regularly to answer questions

Consultation:

- Tutor present to answer any course related questions.
- In-person: Friday 9AM – 11AM, Week 1-10, K17 G01
- online: Friday 9AM – 11AM, Week 1-10, access via Moodle

# Course Information

How to access **Online Consultations**?

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

**Practices** questions:

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

# 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 Staff

## Lecturer-in-Charge

- Professor Wenjie Zhang
- Email: wenjie.zhang@unsw.edu.au
  (for course queries, use the forums)

## Tutor Team

- Mostly research students from the *Data and Knowledge Research Group*

  Course Admin – Yiheng Hu (yiheng.hu@unsw.edu.au)

  - ~10 TAs - PhD and Master by Research Students

- We have ~400 students in 25T3

# 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:
  - Design and implementation detail of databases (COMP9315)

# Course Syllabus

**Data modelling and database design (Week 1 and Week 2)**

i. ER model, ER-to-relational

ii. Relational model (relational algebra), mapping of ER to relational model

**Essentials of Database application development (Week 3 and Week 4)**

i. SQL, views, stored procedures, triggers, aggregates

ii. PostgreSQL: PLpgSQL (procedural)

**Formal database design theory and system architecture (Week 5 to Week 9)**

i. Normalisation, functional dependencies

ii. Storage and indexing, data access operations

iii. Query processing: translation, optimisation, evaluation

iv. Transaction processing: transactions, concurrency control, recovery

v. NoSQL: NoSQL Databases for other purposes, i.e., MongoDB, Neo4j

# Tentative Weekly Outline

| Week | Tuesday | Wednesday |
|---|---|---|
| Week 1 | Subject Intro, Intro to DB | Conceptual DB Design (ER) |
| Week 2 | Relational Data Model | Relational Algebra |
| Week 3 | SQL | SQL |
| Week 4 | PLpgSQL | PLpgSQL |
| Week 5 | Functional Dependencies | Normal Forms |
| *Week 6* | *Quiet Week (No Lecture)* | *Quiet Week (No Lecture)* |
| Week 7 | Relational Database Design | Disk, File, Index |
| Week 8 | Transaction Management | Transaction Management |
| Week 9 | NoSQL | NoSQL/Guest Lecture |
| Week 10 | Advanced Topics/Guest Lecture | Revision |

# 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.

# Course Assessments (Cont.)

COMP9311 25T3 Assessment Summary:

| Number | Name | Full Mark |
|---|---|---|
| 1 | Assignment 1: Data Modelling + Relational Algebra (week 1-3) | 24 |
| 2 | Assignment 2: DB design Theory + Transaction (week 7-9) | 26 |
| 3 | Project 1: SQL, PLpgSQL (week 4-6) | 50 |
| 4 | Final Exam (Exam Week) | 100 |

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

- $\text{FinalMarks} = \sqrt{(Ass1 + Ass2 + Proj1) * FinalExam}$

# Submission

Assignments, OK to have either:

- Directly using word and then convert to PDF
- Hand-written and then convert to PDF/word

Submissions should be made through **Moodle**.

# Late Submission

**5% reduction per day** (of the full mark) for assignments and project

- 0 marks after 5 days late
- **1 second late = 1 day late**
- Submit wrong files = Late
- Please **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 (**within 3 working days**)
- Evidence is needed, application process and details in **here**
- **No other excuses are accepted** (e.g., network down, too busy, forgot to submit)

# 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.

# Generative AI

**Be careful using ChatGPT or other AI tools! There are not reliable.**

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

The impact of AI on Computer Science Education:

https://cacm.acm.org/news/the-impact-of-ai-on-computer-science-education/

# Final Exam

**DO NOT** go to the exam if you are not well enough to do so.

UNSW will consider your attendance **proof that you were OK** at the time of the exam. Go to the Doctor and apply for special consideration.

# Learning Summary/Approach

You'll mostly be fine in our exam if you…

- Follow lectures.
- Attempt all the practice exercise questions with solutions.
- Understand the theoretical component.
- Make the most of the practical component in the lab.

# Research Opportunities

- We are top-ranked worldwide for database research.
- Webpage: https://unswdb.github.io/

Research Degrees:

- https://research.unsw.edu.au/higher-degree-research-programs
- PhD (about 3.5 - 4 years) or MPhil (1.5 – 2 years).

Requirements:

- You received a degree previously in Computer Science or relevant fields from a world top 200 university (either overall ranking or by subject)
- WAM > 85 (UNSW standard) or equivalent

Reach out to me

- Email: wenjie.zhang@unsw.edu.au

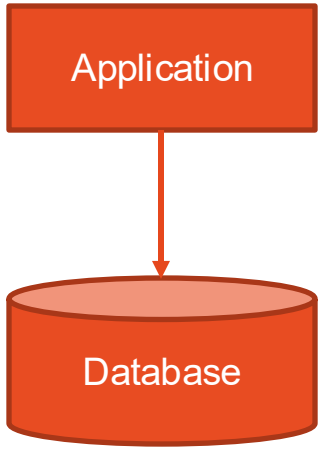
# Outline

- Course introduction
- **Database introduction**
- Database features

# Why Study Databases?

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)

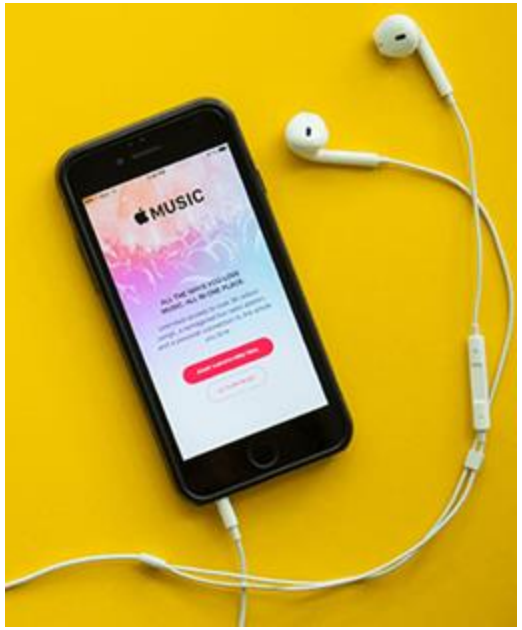



Red points are handled by databases; blue by networks.

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

Databases are the core component of most computer applications.

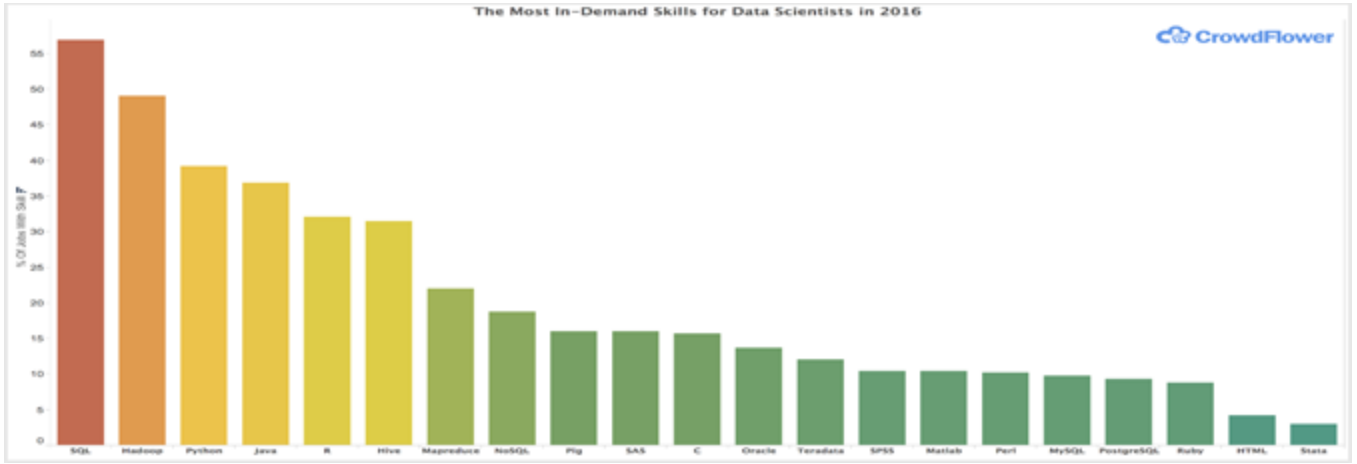
# Applications



# Data Science Skills Employers Want

Writing SQL Queries & Building Data Pipelines (KDnuggets 2022)

- “Learning how to write robust SQL queries and scheduling them on a workflow management platform like Airflow will make you extremely desirable as a data scientist, hence why it’s point #1.”




https://www.datanami.com/2016/01/07/what-data-science-skills-employers-want-now
https://www.kdnuggets.com/2021/10/11-most-practical-data-science-skills-2022.html

# What is Data?

- **Data**: known facts that can be recorded and have *implicit meaning* ...

For example - a student records database:

| Item | Type of data | Stored as |
|---|---|---|
| Name | String | Character strings? |
| Birthdate | Date | 3 integers? |
| WAM | Real number | Float number? |
| … | | |

# 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 relational databases

# Database Example

- Create a database that manages course enrollment 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** (strawman):
  - 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
- ***Data isolation* — multiple files and formats**
- ***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

# 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.**

# Database Management System (DBMS)

A database management system (**DBMS**) is software that allows applications to store and access information in a database.

DBMS contains:

- Collection of interrelated data
- Set of programs to access the data (i.e., define, create, query, update, and administrate)
- An environment that is both *convenient* and *efficient* to use

# Outline

- Course introduction
- Database introduction
- **Database features**

# DBMS Features

- Data Independence
- Efficient Data Access
- Data Integrity and Security
- Data Administration
- Concurrent Access and Crash Recovery
- Reduced Application Development Time

What you see is only a tip of the iceberg

# Database Management System (DBMS)

Recall Database Applications with DBMS

- 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

# Database Systems

## Frequent Terms

- Data ... defined by the scenario
- Relationships ... amongst data items
- Constraints ... on data and relationships
- Redundancy ... one source for each data item
- Data Manipulation ... declarative, procedural
- Concurrency ... multiple users sharing data
- Transactions ... multiple actions, atomic effect

# Summary

- *Data*: known facts that can be recorded and have implicit meaning ...
- *Database*: ... a collection of related data ...
- *Database Management System (DBMS):* ... a collection of programs that enables users to create and maintain a database ...
- *Database System:* ... The database and DBMS together ...

# 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. Jim 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

# Database requirements

Database Systems give you the ability to…

- *Define a database*
  - specifying the data items to be stored and their types,
- *Construct a database*
  - loading the data items and storing them on some storage medium,
- *Manipulate a database*
  - querying - i.e. retrieving relevant data,
  - updating - i.e. adding, deleting or modifying data items
- *Obtain Usage Reports*

# Database requirements (2)

Basic Expectations:

- *Timely - e.g., an airline database (fast response), a CAD system (must be interactive).*
- *Modifiable - must be able to be extended or reorganised, e.g., to cope with new laws, requirements, business conditions.*
- *Robust - e.g., power failure during an update - must be able to recover to a consistent state.*
- *Multi-user - e.g., trading system.*
- *Secure* - different classes of users may need different levels of access,
- *No redundancy*

# 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

# 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)

# Types of DBMS

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

# Top Database Management Systems

1. *Oracle* (Relational DBMS)
2. *MySQL* (Relational DBMS)
3. *Microsoft SQL Server* (Relational DBMS)
4. *PostgreSQL* (Relational DBMS)
5. *MongoDB* (Document Store)
6. Redis (Key-value Store)
7. Elasticsearch (Search Engine)
8. IBM DB2 (Relational DBMS)
9. SQLite (Relational DBMS)
10. Microsoft Access (Relational DBMS)

| Rank | | | DBMS | Database Model |
|---|---|---|---|---|
| Sep 2023 | Aug 2023 | Sep 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. | Elasticsearch | Search engine, Multi-model |
| 8. | 8. | 8. | IBM Db2 | Relational, Multi-model |
| 9. | ↑ 10. | ↑ 10. | SQLite | Relational |
| 10. | ↓ 9. | ↓ 9. | Microsoft Access | Relational |

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

# Data Model Concepts

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

**Database Instance** *(or State): 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.

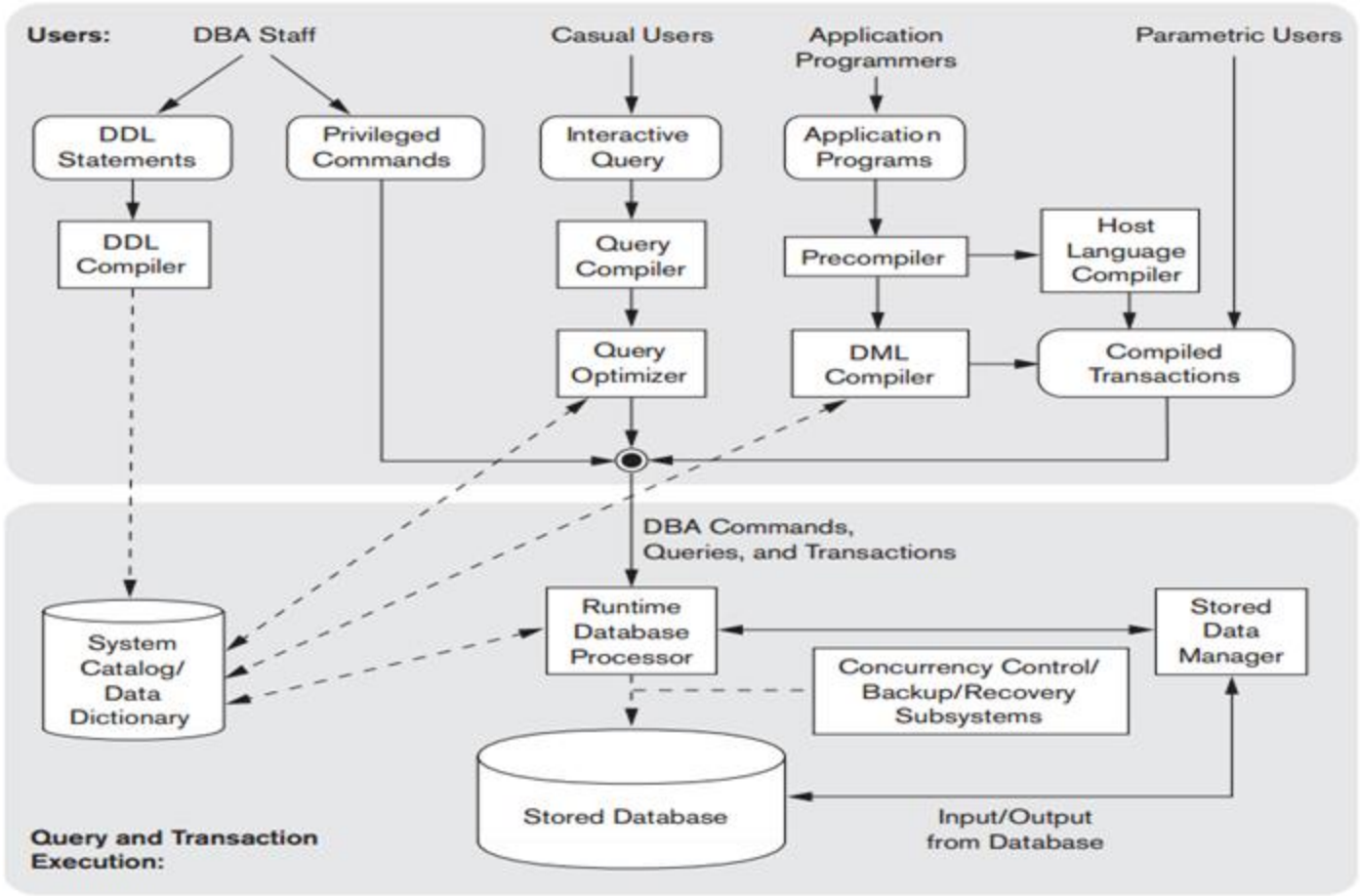
# Design a Database

- **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.

# In Conclusion

Hopefully, you now know…

- course structure,
- who to contact (where to seek help before emailing me),
- 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