>>

COMP3311 Course Overview

- Why Study Databases?
- Databases: Important Themes
- What is Data? What is a Database?
- Studying Databases in CSE
- Syllabus Overview
- Your Background
- Teaching/Learning
- Assignments
- Quizzes
- Exam
- Supplementary Assessment Policy
- Assessment Summary
- Textbook (options)
- Database Management Systems
- Further Reading Material
- Home Computing
- Overview of the Databases Field
- Database Application Development
- Database System Architecture

^ >>

COMP3311 Database Systems

Lecturer: *John Shepherd* (cs3311@cse.unsw.edu.au)

Web Site: http://webcms3.cse.unsw.edu.au/COMP3311/21T3/
or http://www.cse.unsw.edu.au/~cs3311/

(If Webcms3 is unavailable, try http://www.cse.unsw.edu.au/~cs3311/21T3/)

<< ^ >>

❖ Why Study Databases?

Every significant computer application involves Large Data.

This needs to be:

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

Green stuff handled by databases; blue by networks.

Challenges in building effective databases: efficiency, security, scalability, maintainability, availability, integration, new media types (e.g., music), ...

<< ^ >>

❖ 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? What is a Database?

According to the Elmasri/Navathe textbook ...

- Data = known recorded facts, with implicit meaning
  - e.g. a student's name, a product id, a person's address or birthday
- Database = collection of related data, satisfying constraints
  - e.g. a student *is enrolled in* a course, a product *is sold at* a store
- DBMS = database management system
  - software to manage data, control access, enforce constraints
- RDBMS = relational database management system
  - e.g. PostgreSQL, SQLite, Oracle, SQL Server, MySQL, ...

<< ^ >>

❖ Studying Databases in CSE

COMP3311 introduces foundations & technology of databases

- skills: how to build database-backed applications
- theory: how do you know that what you built was good

After COMP3311 you can go on to study ...

- COMP9313: managing Big Data (Hadoop, Spark, NoSQL techniques)
- COMP9315: how to build relational DBMSs (write your own PostgreSQL)
- COMP9318: techniques for data mining (discovering patterns in DB)
- COMP9319: Web data compression and search (XML data)
- COMP6714: information retrieval, web search (dealing with text data)
- COMP9321: data services (making data available via a network)

<< ^ >>

❖ Syllabus Overview

Core syllabus ...

- Data modelling and database design
  - ER model, ODL, ER-to-relational
  - Relational model (design theory, algebra)
- Database application development
  - SQL, views, stored procedures, triggers, aggregates
  - SQLite: `sqlite3` (an SQL shell)
  - PostgreSQL: `psql` (an SQL shell), `PLpgSQL` (procedural),
  - Programming language access to databases (Python, ORMs)

The brown stuff is not covered in tutes/pracs and is not examinable

❖ Syllabus Overview (cont)

More syllabus ...

- Database management systems (DBMSs)
  - DBMS architecture: query processing, index structures
  - Transaction processing: transactions, concurrency control, recovery
- Future of Databases
  - Limitations of RDBMS's, potential future technologies

Blue and green stuff is covered only briefly, and is not examinable

To learn more about the green stuff, take COMP9313, ...

To learn more about the blue stuff, take COMP9315, ...

<< ^ >>

❖ Your Background

We assume that you ...

- have experience with procedural programming
- have some background in data structures
- hopefully, have some knowledge of Python

You might have acquired this background in

- COMP1511, COMP1531, COMP2521

If you don't know Python, look at some online tutorials soon.

e.g. https://www.python.org/about/gettingstarted/

<< ^ >>

❖ Teaching/Learning

Stuff that is available for you:

- Textbooks: describe most syllabus topics in detail
- Topic Videos: summarize all syllabus topics
- Problem-solving sessions: work through examples
- Tutorial sessions: theory/prac questions (+ solutions)
- Prac exercises: lab-like exercises
- Assignments: more detailed practical exercises
- Quizzes: periodic progress check

All online. If you want on-campus, wait for ~~COMP3311 21T1~~ ???

<< ^ >>

❖ Teaching/Learning (cont)

On the course website, you can:

- find out the latest course news
  (important annoucements will also be emailed)
- view the topic-based slides/videos
- get details of tute/prac exercises
- get assignment specs/material
- do the quizzes
- get your questions answered (via the Forums)

URL: https://webcms3.cse.unsw.edu.au/COMP3311/21T3/

(If Webcms3 is unavailable, try http://www.cse.unsw.edu.au/~cs3311/21T3/)

<< ^ >>

❖ Assignments

Two assignments, which are critical for learning

1. SQL/PLpgSQL, 15%, due end week 5
2. Python/SQL/psycopg2, 20%, due end week 9

All assignments are done individually, and ...

- submitted via **`give`** or Webcms3
- automarked (so you must follow specifcation exactly)
- plagiarism-checked (copying solutions ⇒ **0** mark for assignment)
- rent-a-coder monitored (buying solutions ⇒ **exclusion**)

<< ^ >>

❖ Quizzes

Six quizzes, each worth 4 marks

- cover material in previous few weeks lectures
- aim to check your understanding of recent material
- done via Webcms3 in your own time
- primarily multiple-choice
- held in weeks 2, 3, 4, 7, 8, 10
- released Monday, due Friday 9pm
- can be submitted multiple times

6×4 = 24, which is mapped into a mark out of 15

Heavy penalties for late submission

<< ^ >>

❖ Exam

The Final Exam includes questions on ...

- SQL, PLpgSQL, (Python), design exercises, analyses
- 60% prac questions, 40% "theory" questions

Online, open-web exam during exam period

- exam is open for 4 hours
- content is what I'd put in a 3-hour in-lab exam
- can work on home machine, or via **`ssh`**, or via **`vlab`**
- all questions typed in and submitted online (**`give`**)

Sample exam will be available on the course website in Week 10

<< ^ >>

❖ Supplementary Assessment Policy

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

If you attempt the Exam

- I assume that you are fit/healthy enough to take it
- no 2nd chance exams, even with a medical certificate

All Special Consideration requests:

- must document how you were affected
- must be submitted to UNSW (useful to email lecturer as well)

Supplementary Exams are held ... (maybe) in mid-January!

❖ Assessment Summary

Your final mark/grade will be determined as follows:

```
quizzes = mark for on-line quizzes   (out of 15)
ass1    = mark for assignment 1      (out of 15)
ass2    = mark for assignment 2      (out of 20)
exam    = mark for final exam        (out of 50)
okExam  = exam >= 20                 (after scaling)

mark    = ass1 + ass2 + quizzes + exam

grade   = HD|DN|CR|PS  if mark >= 50 && okExam
        = FL           if mark <  50 && okExam
        = UF           if !okExam
```

<< ^ >>

❖ Textbook (options)

- Elmasri, Navathe
  Fundamentals of Database Systems (7th ed, 2016)
- Garcia-Molina, Ullman, Widom
  Database Systems: The Complete Book (2nd ed, 2008)
- Ramakrishan, Gehrke
  Database Management Systems (3rd ed, 2003)
- Silberschatz, Korth, Sudarshan
  Database System Concepts (7th ed, 2019)
- Kifer, Bernstein, Lewis
  Database Systems: Application-Oriented Approach (2nd ed, 2006)

Earlier editions of texts are ok

❖ Database Management Systems

Two example DBMSs for prac work:

- SQLite (open-source, free, no server needed)
- PostgreSQL (open-source, free, full-featured)

Comments on using a specific DBMS:

- the primary goal is to learn SQL (a standard)
- the specific DBMS is not especially important **
- but, each DBMS implements non-standard features
- we will use standard SQL as much as possible
- PG docs describe all deviations from standard

** Unless it seriously violates SQL standards ... I mean you, MySQL

<< ^ >>

❖ Further Reading Material

The on-line documentation and manuals provided with:

- SQLite are reasonably good
- PostgreSQL are very good
- Python are similarly comprehensive

Some comments on technology books:

- tend to be expensive and short-lived
- many provide just the manual, plus some examples
- generally, anything published by O'Reilly is useful

Aside: once you understand the concepts, the manual is sufficient

<< ^ >>

❖ Home Computing

Software versions that we'll be running this semester (TBC):

- PostgreSQL 12/13, SQLite 3.x, Python 3.7+, psycopg2 2.8+

If you install them at home:

- get versions "close to" these
- test all work at CSE before submitting

Alternative to installing at home:

- run them on the new AWS CSE server in a terminal window
- use **`vlab`** to log in to the AWS server with a window manager

Details on setting up a PostgreSQL server are in the first Prac Exercise.

❖ Home Computing (cont)

To access new server via **`ssh`**:

```
$ ssh d.cse.unsw.edu.au
```

To access via VLab, use a VNC service with:

```
d.cse.unsw.edu.au:5920
```

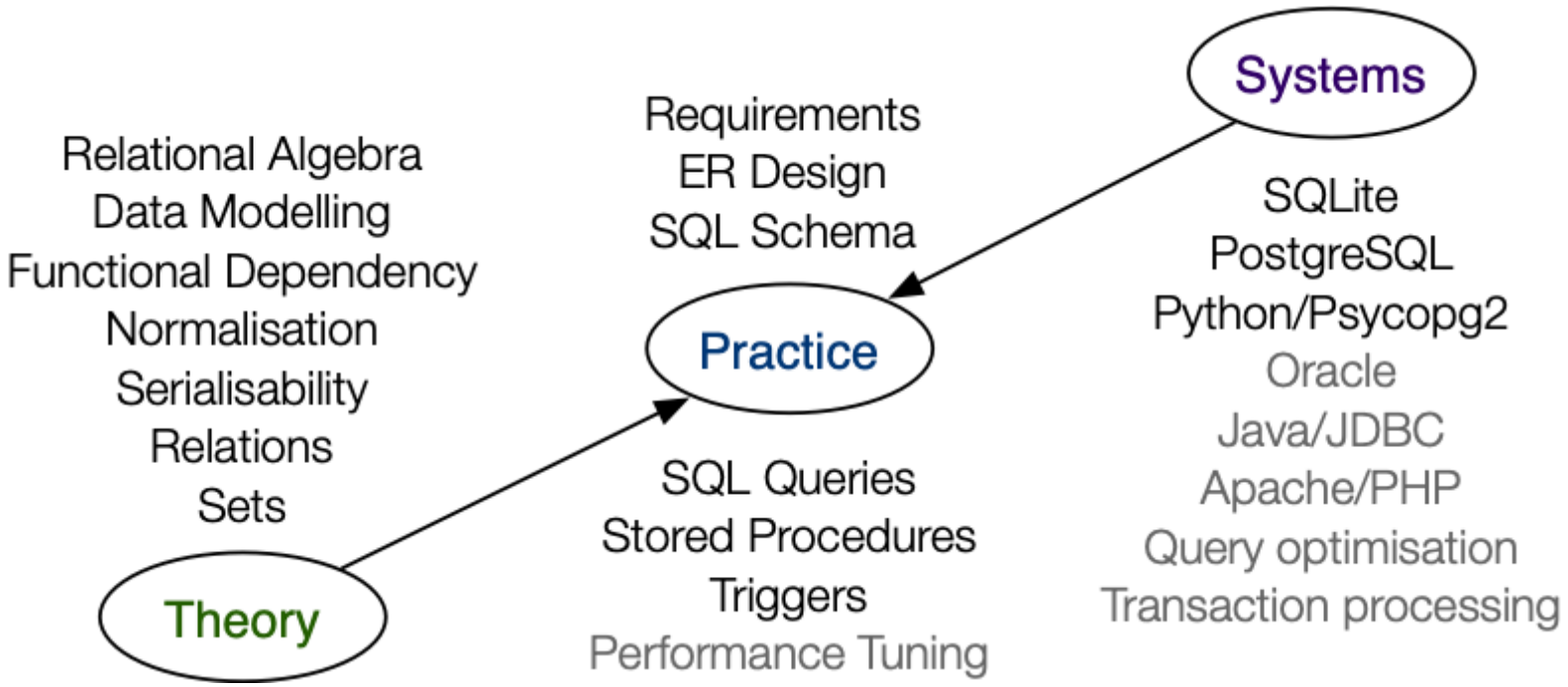
Hostname is **`nw-syd-vxdb`**

Your CSE home directory is accessible on this server

You have a special directory under **`/localstorage`**

More details on how to set up PostgreSQL in first Prac Exercise

❖ Overview of the Databases Field



❖ Database Application Development

A variation on standard software engineering process:

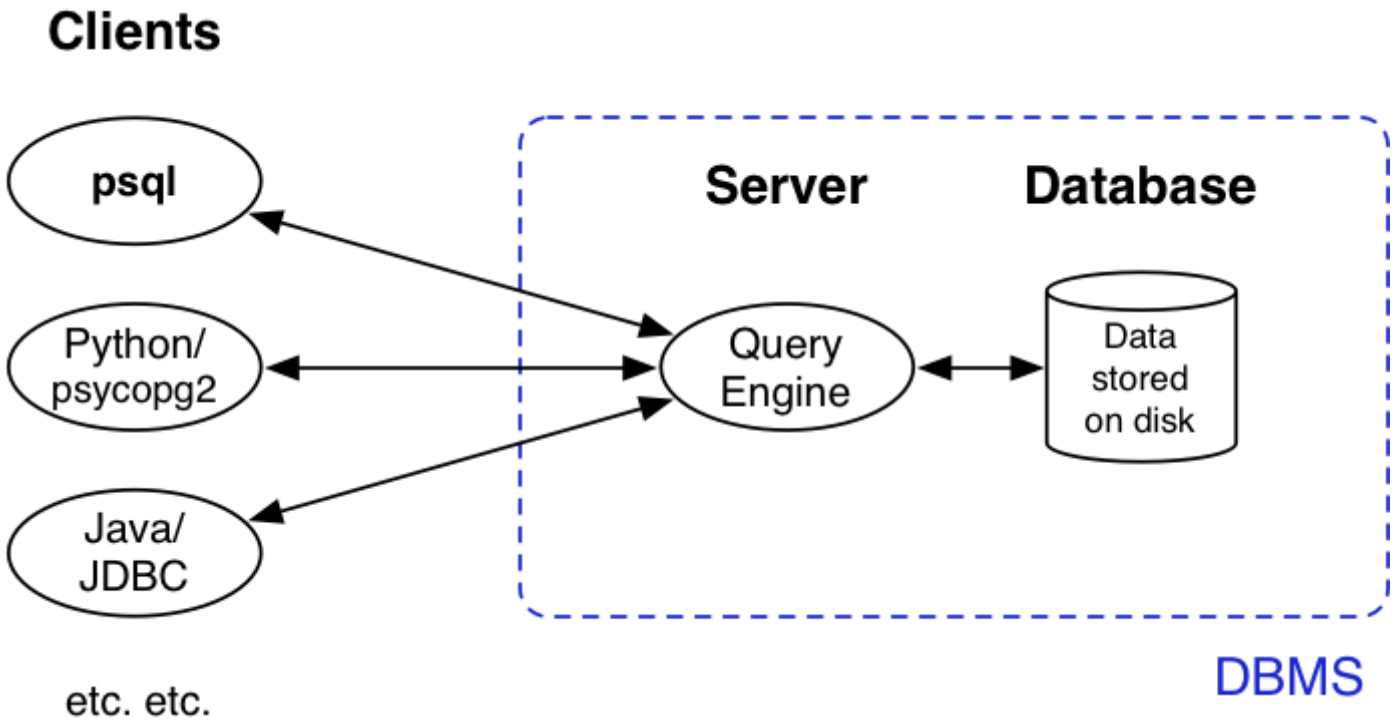
1. analyse application requirements
2. develop a data model to meet these requirements
3. check data model for redundancy (using relational theory)
4. implement the data model as relational schema
5. define operations (transactions) on this model
6. implement operations via SQL and procedural PLs
7. construct a program interface to these operations
8. monitor performance and "tune" the schema/operations

At some point, populate the database (may be via interface)

During the course, we consider these in the order 2, 4, 6, 7, 3

❖ Database System Architecture

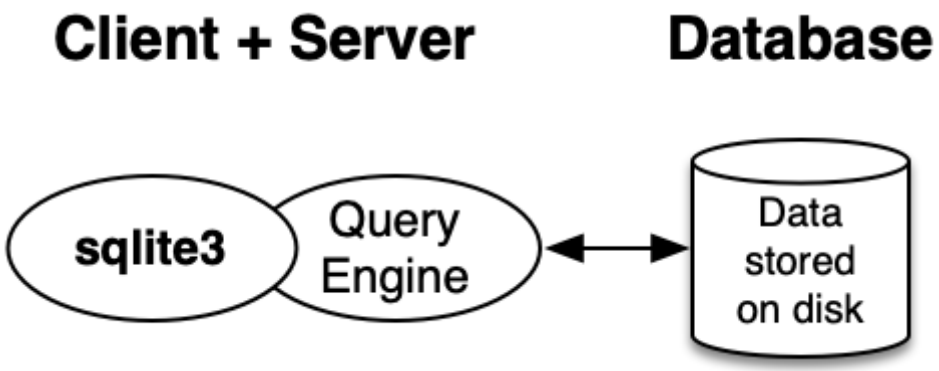
The typical environment for a modern DBMS is:



SQL queries and results travel along the client↔server links

❖ Database System Architecture (cont)

SQLite is not a client-server system:



Although it does have an API for use from programming languages.

Produced: 12 Sep 2021