COMP9311 DATABASE SYSTEMS

Welcome and Hi

22T1; Week1; Feb14th

Course Schedules

Lectures:

- Live and Online.
- Starts Week 1. (6PM – 8PM (Monday) and 4PM – 6PM (Thursday))
- Recorded.

Labs:

- (starting Week 2, every week): guides you through the practical skills on the database application programming part of the course
- Serve also as consultations where tutors will answer all sorts of questions related to the course.

Course Schedules

For Face to Face (f2f) Labs:

- You must wear an appropriate face mask when indoors on campus
- Important: *If you have possible covid symptoms*, It is vital that you don't come to campus, and rest at home.
- Feel welcome to attend any of the alternate online classes.
- Unfortunately, there will be no recordings for f2f labs.

Course Information

Practices questions:

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

How to access Consultations/Labs/Lectures?

- Log into Moodle (https://moodle.telt.unsw.edu.au/).
- Go to course (COMP9311 – Database Systems 2022 T1).
- Click "Lectures and Recordings"
- Click the corresponding lab or consultation session to join.

Course Help

Consultation (Live and Online):

- Tutor present to answer any course related questions.
- Hour before the Thursday Lecture 3PM – 4PM (Thursday)

Course Forum (Use the Webcms3 Q&A Section):

- https://webcms3.cse.unsw.edu.au/COMP9311/22T1/forums/
- Tutors will visit the forum regularly to answer questions

For other course related help and or queries:

- comp9311unsw@gmail.com

If and only if you've exhausted the above options: send me an email.

For other Enrolment Issues

The course enrolment process isn't something lecturers have direct control over.

Who to contact: CSE school office at cse.admin@unsw.edu.au

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

Instructor/ Lecturer

- Michael Yu
- Office: K17 501K
- Email: mryu@cse.unsw.edu.au
  (for course queries, contact the course email)

Tutors

- PhD candidates from the CSE *Data and Knowledge Research Group*
  - Course Admin – QingShuai (K17-201 closed area)
  - TAs - ShunYang, Yuting, JianWei (K17-201 closed area)
- We have in 22T1 ~240 students, each tutor looks after 3-4 labs.

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 8)

i. Normalisation, functional dependencies

ii. Storage and indexing, data access operations

iii. Query processing: translation, optimisation, evaluation

iv. Transaction processing: transactions, concurrency control, recovery

Tentative Weekly Outline

Week	Monday	Thursday
Week 1	Subject Introduction, 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*	*Quiet Week*
Week 7	Relational Database Design	Disk, File, Index
Week 8	Transaction Management	Transaction Management
Week 9	Research Topics I	Research Topics II
Week 10	Research Topics III	Revision

Course Textbook

Text Book:

- Elmasri & Navathe, *Fundamentals of Database Systems*, Benjamin/Cummings, 6th Edition, 2010.

Reference Books:

- 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.

Note: Lecture notes will be sufficient

Course Assessments (1)

Two Assignments (12.5% relative weighting)

- Ass 1: Data Modelling + Relational Algebra (week 2-4)
- Ass 2: DB Design Theory + Database Storage Structures + Transaction (week 8-10)
- *20% reduction per day*.

One Project (25% relative weighting)

- Proj 1: SQL PLpgSQL (50%) (week 4-7)
- *20% reduction per day*.

One Exam (50% relative weighting)

- Date to be decided later

Course Assessments (2)

COMP9311 22T1 Assessment Summary:

Number	Name	Full Mark
1	Assignment 1: Data Modelling + Relational Algebra	25
2	Assignment 2: DB design Theory + Transaction	25
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)

Plagiarism

All submissions are checked for plagiarism. The university regards plagiarism as a form of academic misconduct and has very strict rules regarding plagiarism. Not knowing the rules *is not considered a valid excuse.*

For UNSW policies, penalties, and information to help avoid plagiarism, please see: https://student.unsw.edu.au/plagiarism.

For guidelines in the online ELISE tutorials for all new UNSW students: https://subjectguides.library.unsw.edu.au/elise/plagiarism.

Special Consideration

The process for assessing the impact of short-term events , beyond your control (exceptional circumstances), on your performance in a specific assessment task. Details in the **UNSW Application and Assessment Information document**.

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.

Pathways from COMP9311 (1)

In COMP9311, we introduce the foundations & technology of databases

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

Pathways from COMP9311 (2)

After COMP9311, you can go on to study ...

- COMP9315: how to build relational DBMSs (write your own PostgreSQL or Oracle)
- COMP9318: techniques for data mining (discovering patterns in DB)
- COMP9319: web data compression and search (dealing with a large amount of Web data)
- COMP6714: information retrieval, web search (dealing with text data)
- COMP932[1|2|3]: service-oriented computing, which relies on DB background

More Research Opportunities for you

Research Degrees:

- https://research.unsw.edu.au/higher-degree-research-programs
- PhD (3 – 3.5 years) or Master by Research or Master of Philosophy (1.5 – 2 years).

Basic Requirements:

- GPA > 80 with education from world top 400 universities.
- GPA > 75 if from world top 100 universities.

Reach out to Prof. Wenjie Zhang

- Office: K17 502, CSE Phone: (+61 2) 93857799
- Email: wenjie.zhang@unsw.edu.au

Previous: Course Admin

Next: Databases

Why Study Databases?

Most significant modern computer application rely on huge quantities of data.

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; brown by networks.

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

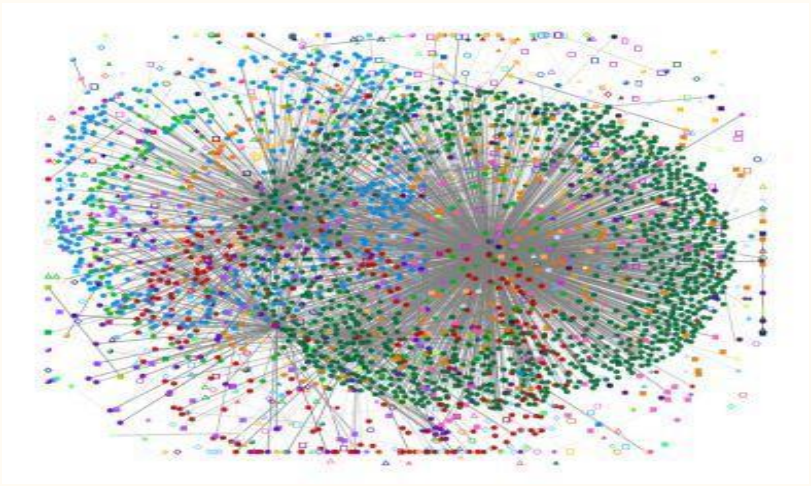
Big Data

There are many different types of data: text data, image data, audio data, video data, etc.

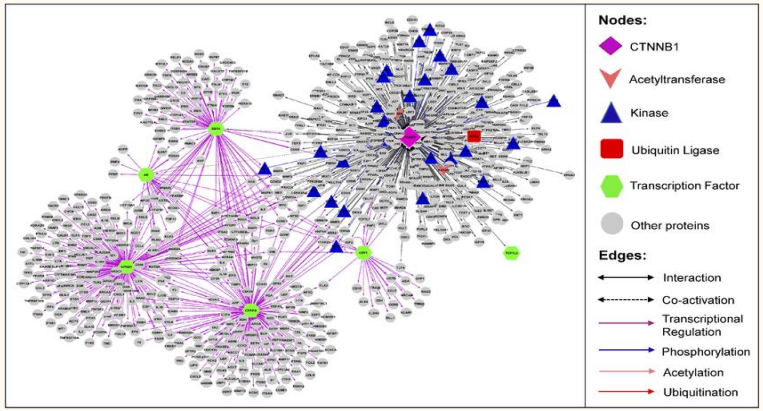
The amount of data grows very fast.

- Zettabyte 1,180,591,620,717,411,303,424 (2^{70}) byte
- Exabyte 1,152,921,504,606,846,976 (2^{60}) byte
- Petabyte 1,125,899,906,842,624 (2^{50}) byte
- Terabyte 1,099,511,627,776 (2^{40}) byte
- Gigabyte 1,073,741,824 (2^{30}) byte
- Megabyte 1,048,576 (2^{20}) byte
- Kilobyte 1,024 (2^{10}) byte
- Byte 1 byte

Internet of Things



Web Graphs



Biological Networks

Social Networks

The Vs of Big Data

- *Volume*: The amount of data matters.
- *Variety*: The many types of data that are available.
- *Velocity*: Desire for data to be received and acted on quickly.
- *Veracity*: The accuracy of your data, how well it conforms to facts.
- *Value*: Data has intrinsic value. Data is of no use until that value is discovered

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

File Systems as Data Management?

File based systems vs database approach.

File based system:

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

Database Approach:

- Improvement to a shared file solution

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*
  - 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

Introduction

Smart Transportation

Business Services

Natural Disasters

Public Health

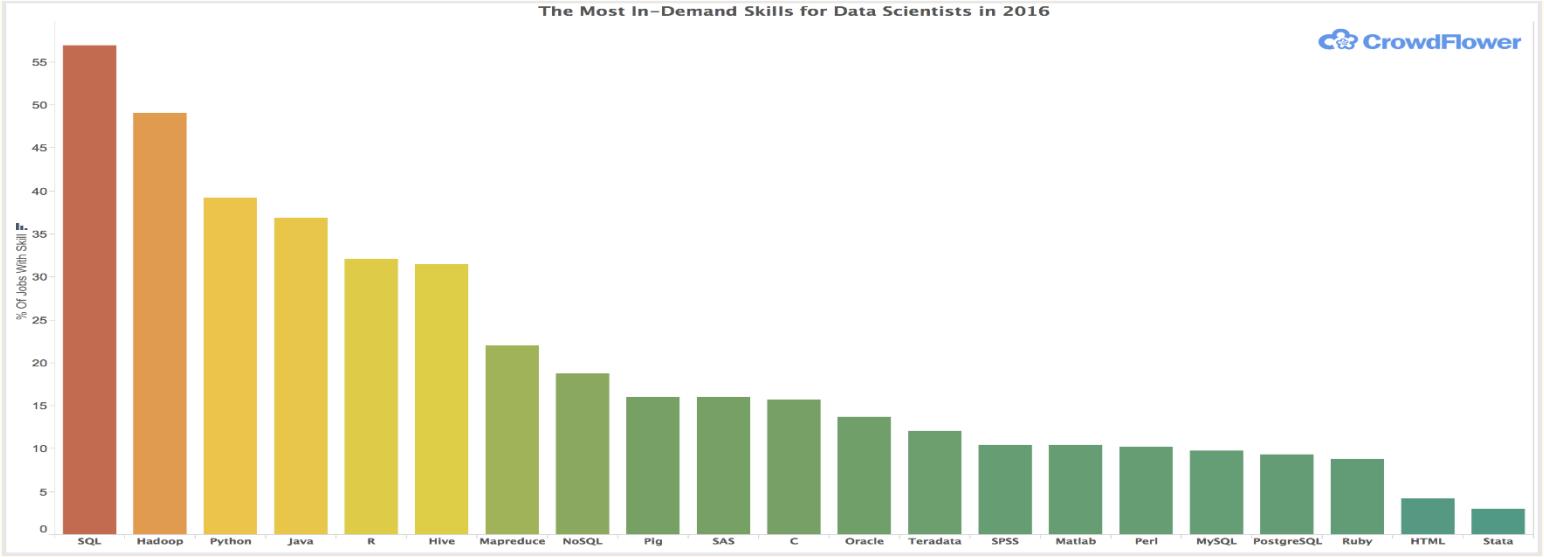
Modern Military

Tourism Development

Data Science Skills Employers Want

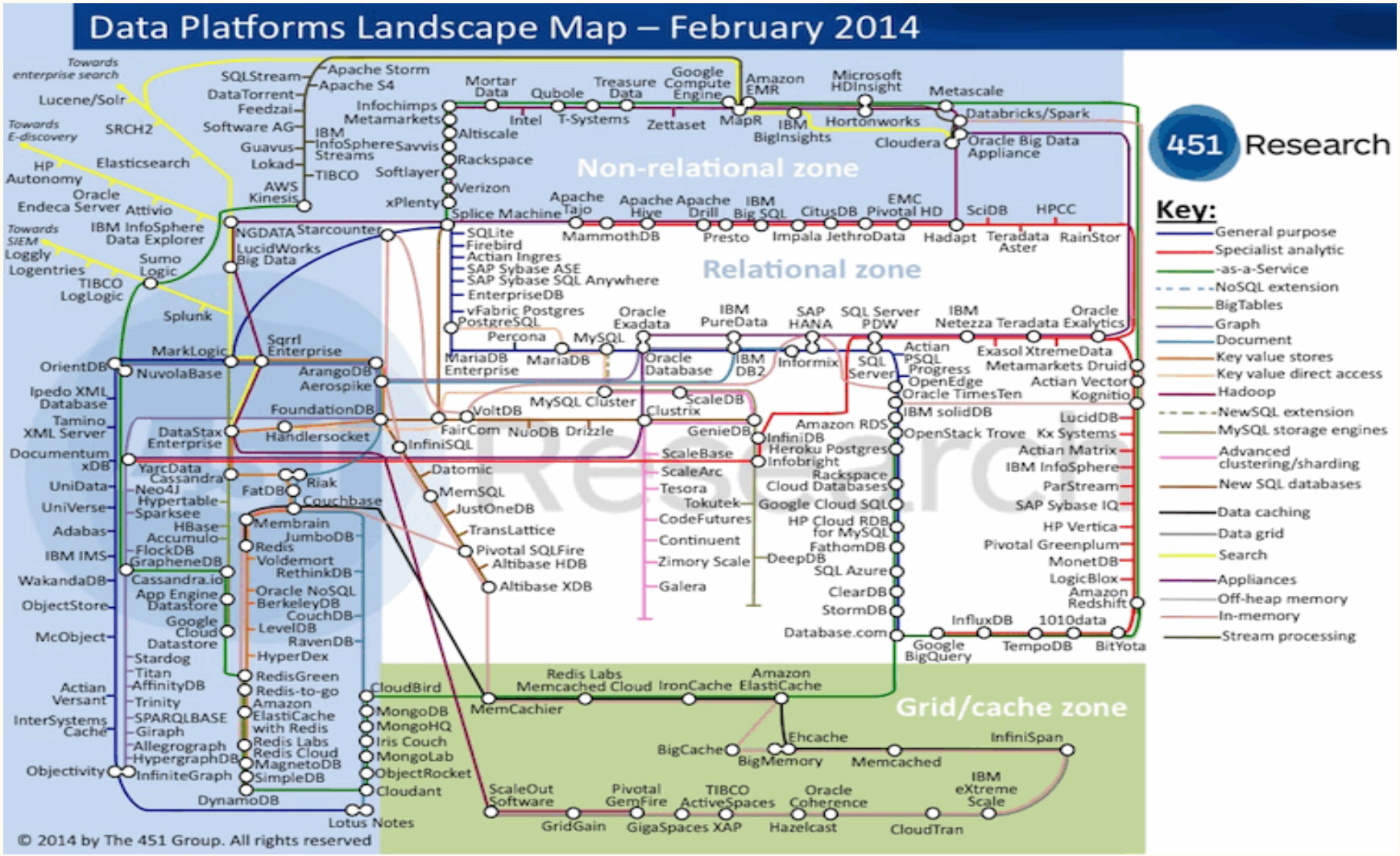
Writing SQL Queries & Building Data Pipelines (KDnuggets 2022)

- "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

Data Platform Landscape Map 2014



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. IBM DB2 (Relational DBMS)
7. Radis (Key-value Store)
8. Elasticsearch (Search Engine)
9. SQLite (Relational DBMS)
10. Cassandra (Wide column)

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

Back to the Databases

Elmasri/Navathe:

- *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 ...

What is Data

Elmasri/Navathe:

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

For Example - a student records database:

Item	Type of data	Stored as
Family name	String	Character strings?
Birthdate	Date	3 integers?
Weight	Real number	Floating point number?
…		

Database Management System (DBMS)

DBMS contains information about a particular enterprise

- Collection of interrelated data
- Set of programs to access the data
- An environment that is both *convenient* and *efficient* to use

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 Themes

- *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

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 (1)

There will likely be a great amount of people who will need regular access to the database.

For most users…

- *Naive User (aka parametric user)* - typically employs the database via "canned transactions" - standardised queries and updates, through a menu system of some kind.
- *Online User* - has an understanding of the database system. May be capable of designing their own queries etc.

Database Users (2)

The *Database Administrator* (DBA) to oversee the daily tasks.

Centralised control of the database:

- Authorise access
- Monitor DB usage,
- Recover data,
- Identify data,
- Choose the DB Structure,
- Manage definitions of views . . .

Database Users (3)

Systems analyst:

- determine end users requirements,
- develop specifications for canned transactions and reports,
- may also take part in database design.

Application programmer - Implements the specifications given by analyst:

- tests,
- debugs,
- maintains the resulting programs.

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 (cont) 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.

Checking understanding:

- 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.

ANSI-SPARC three level architecture

ANSI-SPARC Three Level Architecture (1975):

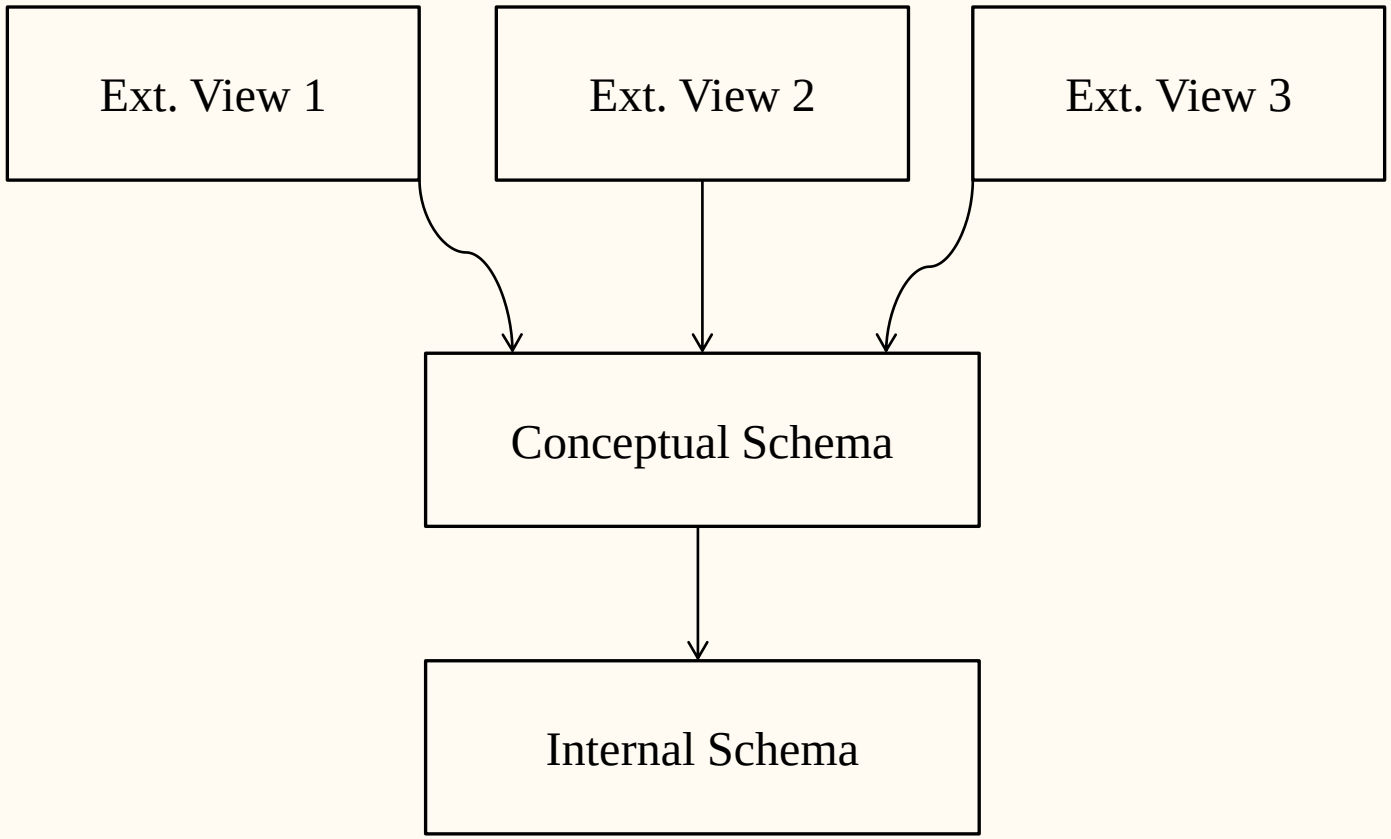
- *External/View level:*
  - includes a number of external schemas or user views.
- *Conceptual level:* has a conceptual schema,
  - describes the structure of the whole database for a community of users.
- *Internal level:* has an internal schema,
  - describes the physical storage structure of the database.

Three level architecture replaces the two-level approach (1971).

ANSI: American National Standard Institute.

SPARC: Standards Planning and Requirements Committee.

The Three Level Architecture



Data Independence

There are two kinds…

- *Logical data independence*: the ability to change the conceptual schema without changing external views.
- *Physical data independence*: the ability to change physical storage paths and access structures without changing the conceptual view

Conclusion: 3 levels of abstraction => 2 levels of data independence

Database Languages

The DBMS provides the following languages:

- *View definition language (VDL):* used to define external schemas.
- *Data definition language (DDL):* used to define the conceptual schema.
- *Storage definition language (SDL):* used to define the internal schemas.

Note: In the context of DBMS where conceptual and internal levels are together, DDL defines both schemas.

Database Languages

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

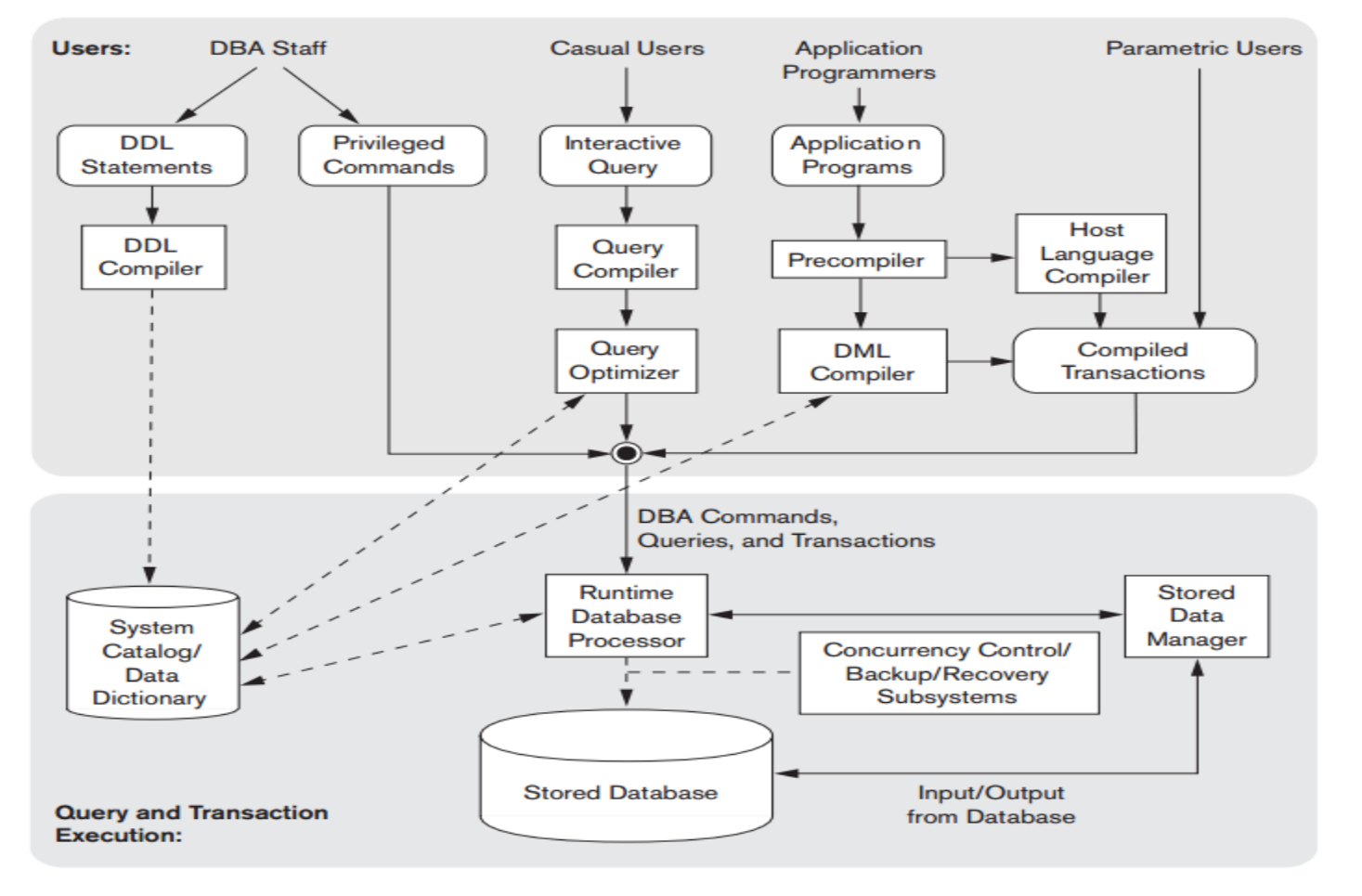
- *Non-procedural DML* (e.g., SQL, common for casual users)
  - interactive and/or embedded
  - set at a time/ set oriented.
- *Procedural DML* (also covered in this course)
  - embedded in a general purpose language,
  - record at a time

Database Query Processing

Components Involved

- *Run-time database processor*: Receives and update requests and carries them out with the help of the stored data manager.
- *Stored data manager or file manager*: Controls access to the DBMS information stored on disk:
  - may use the OS for disk access,
  - controls other aspects of data transfer, such as handling buffers.
- *Pre-compiler:* extracts DML commands from the host language program.
- *Query processor (or Complier):* parses high-level queries and converts them into calls to be executed by the data manager.

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