

# COMP9120

## Lecture 1

### Database Management Systems Introduction

Presenter: Dr Armin Chitizadeh

*Version 1.1*



THE UNIVERSITY OF  
**SYDNEY**

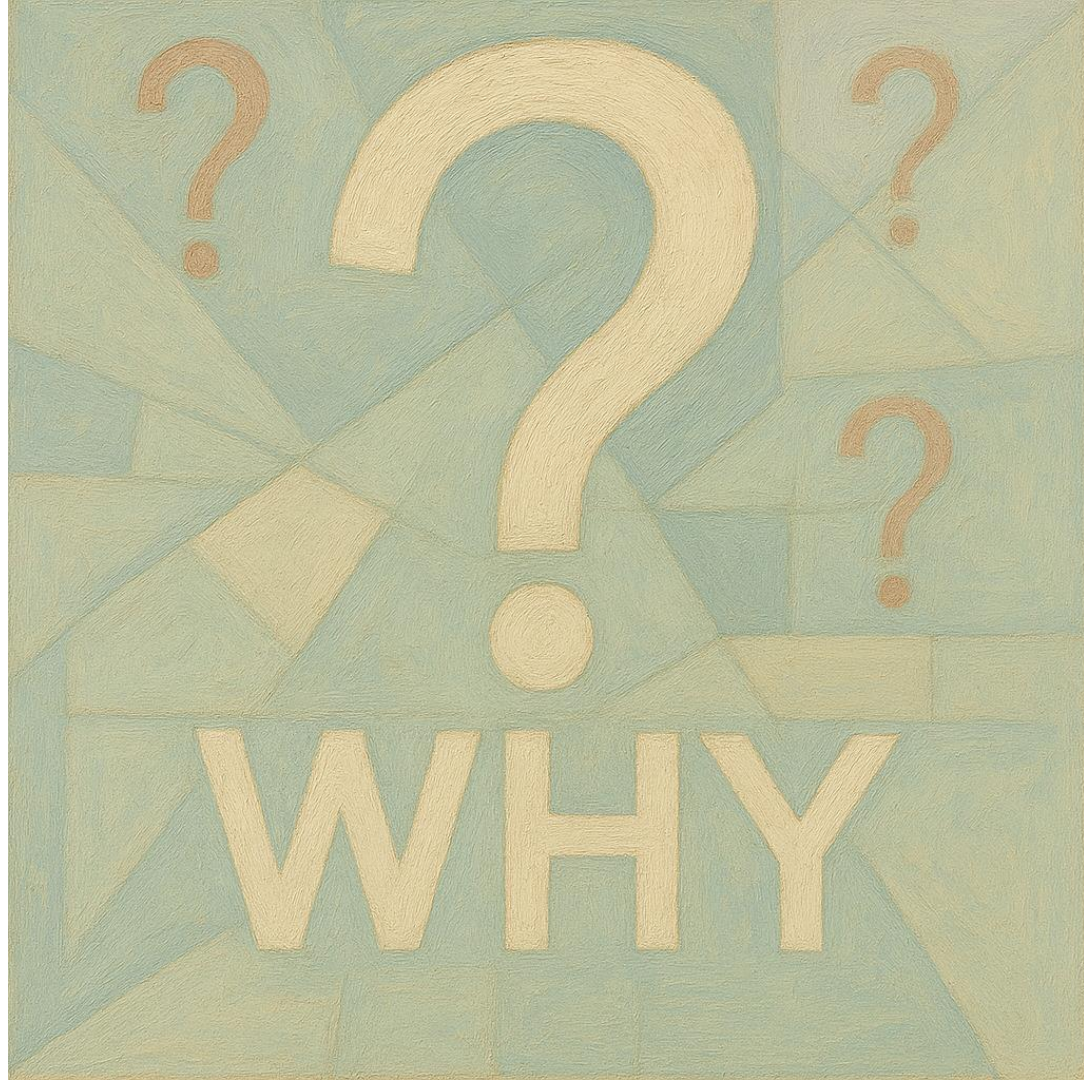
Inspired by the work of  
Professor Athman Bouguettaya  
and Dr Lijun Chang.

CRICOS 00026A TEQSA PRV12067



Who am I?





# Why?

- Discuss with people around you on why you picked this unit?

Let's menti!

<https://www.menti.com/alpxd7pz7jdr>



We recognise and pay respect to the Elders and communities – past, present, and emerging – of the lands that the University of Sydney's campuses stand on. For thousands of years they have shared and exchanged knowledges across innumerable generations for the benefit of all.



# COMMONWEALTH OF AUSTRALIA

## Copyright Regulations 1969

### WARNING

This material has been reproduced and communicated to you by or on behalf of the University of Sydney pursuant to Part VB of the Copyright Act 1968 (**the Act**).

The material in this communication may be subject to copyright under the Act. Any further copying or communication of this material by you may be the subject of copyright protection under the Act.

**Do not remove this notice.**





## Part1: The Structure of the Unit



# Teaching Team

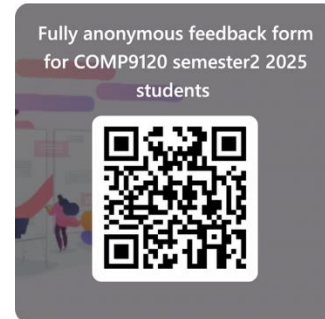
- Lecturer/Coordinator: Dr Armin Chitizadeh
- Lecturer: Dr Imdad

- Tutors:

- |                    |                  |                       |                   |
|--------------------|------------------|-----------------------|-------------------|
| - Abbey Lin (TA)   | - Dipankar Chaki | - Yan Rong            | - Amanveer Bagga  |
| - Thilina Lokuruge | - Hengzhi Chen   | - Qixuan Hu           | - Yuan Liang      |
| - Tommy Zhou       | - Muhammad Umair | - Aliasghar Smailpour | - Mehraz Tabassum |

# Platform used

- Canvas
  - Lecture slides, tutorial sheets, lecture recordings
  - Assignment submissions
  - Publish grades
- Ed
  - Discussion Forum
  - SQL practice
- Anonymous feedback form:
  - <https://forms.office.com/r/Tf3sAha9hc>



# What this unit of study is (and is not)

We will cover **how database management systems work**. We will discuss the:

- **Design, query, efficient access** and **storage** of databases.
- **Theoretical** aspects underpinning **relational databases**
- This UoS is **\*not\*** an SQL programming course, although we will be using SQL quite extensively!
- Note the following:

Unit code	COMP9120
Prohibitions ?	INFO2120 OR INFO2820 OR INFO2005 OR INFO2905 OR COMP5138 OR ISYS2120. <u>Students who have previously studied an introductory database subject as part of their undergraduate degree should not enrol in this foundational unit, as it covers the same foundational content</u>
Assumed knowledge ?	Some exposure to programming and some familiarity with data model concepts

# Outline of Lectures

	Week	Topic
Database Design and Query	Week 1	Introduction
	Week 2	Conceptual Database Design
	Week 3	Relational Data Model / Logical Database Design
	Week 4	Relational Algebra and SQL
	Week 5	Advanced SQL
	Week 6	Database Integrity
	Week 7	Database Application Development and Security
	Week 8	Schema Refinement and Normalisation
Database Internals	Week 9	Transaction Management
	Week 10	Quiz
	Week 11	Storage and Indexing
	Week 12	Query Evaluation and Optimisation
	Week 13	Review





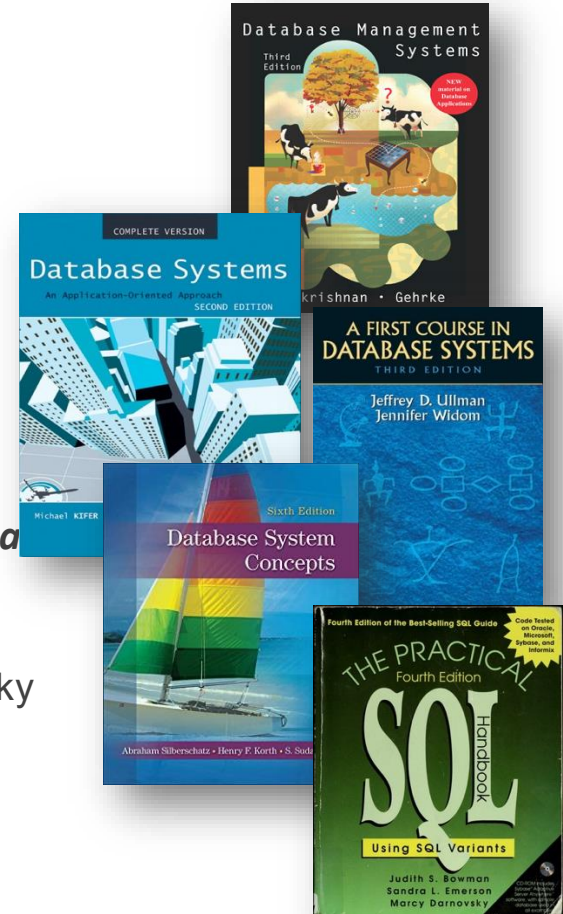
# Textbooks

## Click on Reading List in Canvas

- R. Ramakrishnan and I. Gehrke
  - ***Database Management Systems***
- A. Silberschatz, H. Korth, S Sudarshan
  - ***Database System Concepts***
- J.D. Ullman, and J.Widom:
  - ***A First Course in Database Systems***
- M. Kifer, A. Bernstein, and P.M. Lewis
  - ***Database Systems: An Application-Oriented Approach***

## Suggested additional SQL reference:

- Judith S. Bowman, Sandra L. Emerson, and Marcy Darnovsky
  - ***The Practical SQL***



# Software

- We will be using the following software in labs and assignments:
  - PostgreSQL (server) and pgadmin (client)
  - Java/Python

# How Much Programming is involved?

- Although you do not need to be a highly experienced programmer, you will need some *minimum programming skills* for the practical assignments:
  - As previously noted, this UoS is ***not*** a programming course,
  - It covers database design, creation and usage
- The *DB application programming* assignment (**Assignment 2**) assumes some programming experience and expects some competence in **Java or Python!**
  - It is **your responsibility to learn Java/Python**

# Assessments

## › Assessment tasks:

### 1. Two (2) **Group Assignments**

- *DB Schema Design* **12.5%**
  - Released on **Week 3** and **due** on **Week 6** (Sunday **14 Sept** 11:59pm)
- *DB Usage* **12.5%**
  - Released on **Week 7** and due on **Week 11** (Sunday **26 Oct** 11:59pm)

2. **Quiz** (Week 10) (Your Lecture Time and Location) **20%**

3. Weekly in-lab checkpoints **5%**

4. **Final Exam** (Exam Period) **50%**

› You must obtain  $\geq 40\%$  in the **final exam** and an **overall mark of  $\geq 50\%$** , to pass the unit

For example:

*Progressive Mark **44%**, Exam Mark **50%**, total **47%**: **Fail***

*Progressive Mark **75%**, Exam Mark **35%**, total **55%**: **Fail***



# Forming Groups

- › All the assignments are **group assignments**.
  - Each group should consist of 3 members. Please find others to form a group *as soon as possible*.
  - First assignment is released on Week 3, *groups must be formed by the end of Week 2*.
    - *At the start of week 3, we will randomly create groups of unassigned students.*
  - To ensure the **contribution** of each member, each one of you will *assess* the *contribution* of the *other teammates*. There will be a system to make sure the fairness.
    - In addition, we **will conduct random interviews to** ensure fairness.
- › Each group must organize itself:
  - Arrange internal communication, meetings and upload meeting minutes in Canvas
  - Set internal deadlines
  - Follow up if deadlines are missed
  - Have a fallback plan
  - Have a dispute resolution mechanism in place
- › Let your unit coordinator/lecturer know asap if there are issues (unresolvable disagreements, member who doesn't contribute, etc)

# Forming Groups pt2

- Each group should have **three** members
- Resolve team issues as soon as possible
  - First, have a professional discussion among team members, either in person or via video call.
  - If the issue isn't resolved, ask your tutor for help.
  - If it still isn't resolved, contact the unit coordinator. I will do everything I can to help fix the problem.

# Late Submission

- Late submission will incur **a penalty of 5% per day**.
  - An assignment that would normally get 9/10 and is 2 days late loses 10% of the full 10 marks, i.e., new mark = 8/10
  - An average assignment that would normally get 5/10 and is 5 days late loses 25% of the full 10 marks, i.e., new mark = 2.5/10
- Assessments more than **10 days late get 0 marks**.
- Warning: submission site gets very slow near deadlines
- Submit early; you can resubmit before the deadline

## Special Consideration (University Policy)

- If your performance on assessments is affected by illness or misadventure
  - Follow proper university procedures
    - Have professional practitioner sign special USyd form
    - Submit application for **special consideration** online, upload scans
    - Note you have only **3 working days** for applying
    - [http://sydney.edu.au/current\\_students/special\\_consideration/](http://sydney.edu.au/current_students/special_consideration/)
- Also, notify coordinator by email *as soon as anything begins to go wrong*
- There is a similar process if you need special arrangements, e.g., for religious observance, military service, representative sports, etc.





# Work, Health and Safety (WHS) Induction

School of Computer Science

# Staying Safe in the event of an emergency

- In the unlikely event of an emergency, we may need to evacuate the building.
- If we need to evacuate, please take your belongings and follow the green exit signs.
- We will move a safe distance from the building while waiting until the emergency is over.
- In some circumstances, we might be asked to remain inside the building for our own safety. We call this a lockdown or shelter-in-place.
- More information is available at [www.sydney.edu.au/about-us/campuses/emergencies-and-personal-safety.html](http://www.sydney.edu.au/about-us/campuses/emergencies-and-personal-safety.html)



# Supporting your health and wellbeing

## **Student life, wellbeing and support page**

Everything you need to know about the student services, resources and events available to support you while you study, including peer support, counselling, after hours support, and learning support.



<https://www.sydney.edu.au/students/support.html>

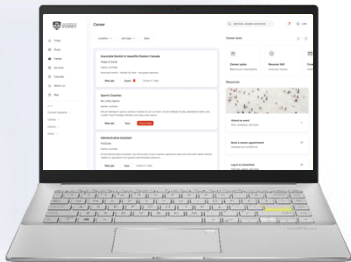
## **Safer Communities Office**

Support for people who have experienced sexual misconduct, domestic/family violence, bullying/harassment or issues relating to modern slavery



[www.sydney.edu.au/students/health-wellbeing/safer-communities.html](https://www.sydney.edu.au/students/health-wellbeing/safer-communities.html)

# Accessing tools and support through the Student Portal



Experience the University of Sydney's enhanced Student Portal where you can:

- View your integrated **Student calendar** including student Outlook calendar, class timetable, and events.
- Explore available job opportunities in the improved **Career section** and have access to exciting new tools like the Resume Builder.
- Access University services via the new **Study and Support** page.

[myuni.sydney.edu.au](https://myuni.sydney.edu.au)

# Understanding your academic integrity responsibilities

- Academic integrity refers to behaving honestly, ethically and responsibly in relation to all elements of your study at the University, including assessments.
- Always submit your own work, sit your own tests, and take your own examinations.
- Acknowledge any contributions in your assignment which are not your original thoughts, ideas or words.
- Academic Honesty Education Module – all commencing students must complete the module by census date. Continuing students can self-enrol at any time.
- Your coordinator will advise you on whether digital tools including grammar checkers and generative AI are allowed in assessments. Failure to comply with the coordinator's instructions on digital tools could amount to an academic integrity breach.
- For University's guidelines on generative AI, see next slide.

## Strategies for maintaining academic integrity



Planning and time management



Use citations and  
referencing



Know your strengths and  
what you need to develop



Know when and where  
to ask for help



# Using generative AI in COMP9120

- Generative AI can create digital content, including text, images, and video.
- Common examples are ChatGPT, Microsoft Copilot, Notion AI, Claude, etc.
- Different guidelines for use of generative AI apply to learning and assessment.

## Generative AI for learning

You are free to use generative AI to help you learn.  
Use it to explain things, apply knowledge, plan your study, make practice questions, and more.  
However, be aware of **its limitations** and **flaws**.

## Generative AI for assessment

*The use of generative AI tools are **NOT permitted** in the Final Exam and the Quiz.*

To learn more about using generative AI visit the Canvas resource developed by students: <https://bit.ly/students-ai>

## Be Careful to Learn from doing your Assignments and In-Lab Checkpoints

Remember, this is Armin when you do not learn from doing your assignments or in-lab checkpoint



# Managing access to our University systems

MultiFactor Authentication (MFA) is an additional security step to verify your identity at the University



More information

- ✓ Enter your UniKey and password on the login screen.
- ✓ Confirm your identity using the Okta Verify app on your phone.

- ✓ Set up a secondary email, phone number, and security question in your Okta profile to be able to reset your Unikey easily.

- ✓ Always have your Okta-enabled device with you.
- ✓ Do not uninstall the Okta Verify app.

Changing your mobile device?

- ✓ Install Okta Verify on your new device before removing it from your old one.



## The Learning Hub introduces *Studiosity*!

- **Writing Feedback:** Get writing and referencing feedback within 24 hours by uploading a draft into Studiosity.
- **Connect Live:** Enter a question in Connect Live and you will be put in touch with a Subject Specialist in an interactive online classroom.
- Available to Postgraduate Coursework students via [Canvas](#)



Access your Studiosity account to  
receive feedback on your draft  
assignment:



## Do you have a disability, medical condition, or caring responsibilities that impact on your studies?

You may not think of yourself as having a 'disability' but the definition under the **Disability Discrimination Act (1992)** is broad and includes temporary or chronic medical conditions, physical or sensory disabilities, psychological conditions and learning disabilities.

If you have ongoing caring responsibilities for another person (who has a disability, medical condition, mental health condition, or who is aged and frail) as defined by **Carer Registration Act 2010**

which impact on your studies, you can register with IDS.

To get assistance, students need to register with Inclusion and Disability Services. It is advisable to do this as early as possible. Please contact us or review our website to find out more.

Inclusion and Disability Services Office  
[sydney.edu.au/disability](http://sydney.edu.au/disability)  
02-8627-8422



# Support in COMP9120

- Tutorials
  - Your tutorials will be run by a friendly tutor - [you can meet them here](#). You can also ask your peers for help.
- Ed Discussion
  - Active class discussion enhances this unit! Keep ED questions concise. Post a question on ed if it can be answered within three sentences. For more complex queries, consult your lecturer, tutor, or visit our friendly help desk.
- Ask your lecturers
  - Meet with your lecturer after the lecture or their consultation time. You can ask any questions related to the course.
- Help Desks
  - TBA

# Menti Game!

- <https://www.menti.com/alaocyxoarij>



Time for a break!



## Part B: Database Introduction



Photo by Kateryna Hliznitsova  
From Unsplash

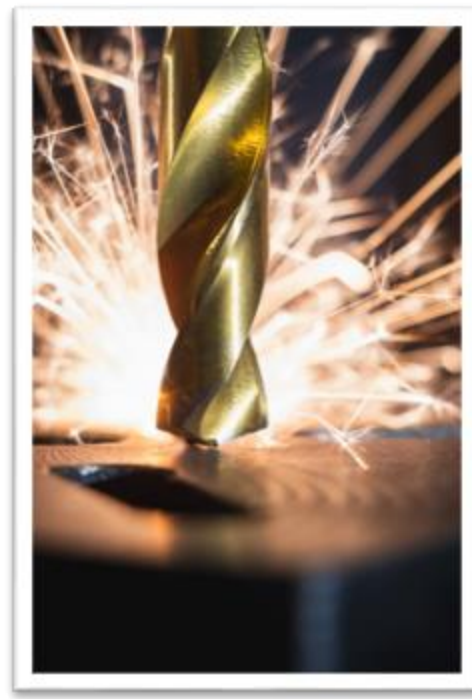


Photo by Greg Rosenke  
From Unsplash



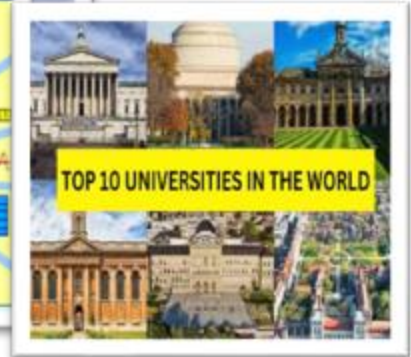
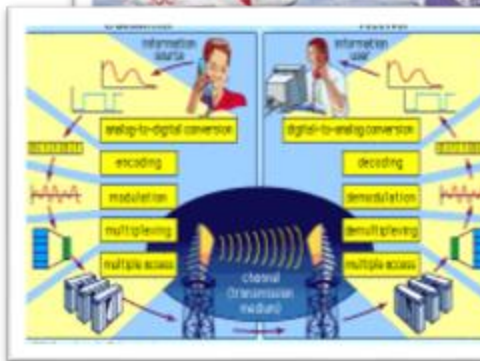
# Why do we study Databases

- They are **everywhere!**
  - They are the *power engine* behind everything that is *digital!*

## Examples

- **Banking systems:** accounts & loans, customers, all types of transactions (banks, ATMs, internet)
- **Social networks:** Facebook, Youtube, Whatsapp, Viber, etc
- **Large Language Models (LLMs):** ChatGPT, Gemini, Co-Pilot, etc
- **Airlines reservation systems:** reservations by customers, flight schedules, frequent flyer info
- **Corporate records:**
  - Universities: student enrolments, course offerings, timetabling, grades
  - Sales: customers, products, purchases - and reports on these
- **Telecommunication:** calls, bills, calling/SIM cards
- **Healthcare:**
  - patients, prescriptions, drugs

**Databases touch *all aspects* of our lives**





# Career Opportunities

- Database Designer
- Database administrator
- Database Analyst
- Database Application Programmer
- Data Scientist



Government of South Australia

### DATABASE ANALYST

TAFE SA

Regency Park, Adelaide SA  
ASO6 Salary range, \$98,470 - \$104,165 + Super • Based on experience

• Full-time, permanent role  
• \$104,165 + Super • Based on experience

3d ago



Wipro Limited

### Database Architect - L2

Richmond, Melbourne VIC (Hybrid)  
\$105k - \$113k + Super. Salary Packaging available.

Biological & Biomedical Sciences (Science & Technology)

- High level of proficiency with EDC systems
- Clinical trials database design, development and management
- Hybrid working arrangements, inner Melbourne location

The successful candidate will be responsible for leading the design and build of ALLG clinical trial databases.



McM Community Services & Development

### Donor Database and Insights Manager

Johnson Recruitment

South Melbourne, Melbourne VIC (Hybrid)  
\$90k - \$110k + Super + pbi Salary Packaging

Fundraising (Community Services & Development)

- Leading Community Support Organisation
- Salary \$90k - \$110k + Super + PBI Salary Packaging
- Hybrid working, South Melbourne office



McM Community Services & Development


### Donor Database and Insights Manager

Johnson Recruitment

South Melbourne, Melbourne VIC (Hybrid)  
\$90k - \$110k + Super + pbi Salary Packaging

Fundraising (Community Services & Development)

- Leading Community Support Organisation
- Salary \$90k - \$110k + Super + PBI Salary Packaging
- Hybrid working, South Melbourne office



McM Community Services & Development

### Donor Database and Insights Manager

Johnson Recruitment

South Melbourne, Melbourne VIC (Hybrid)  
\$90k - \$110k + Super + pbi Salary Packaging

Fundraising (Community Services & Development)

- Leading Community Support Organisation
- Salary \$90k - \$110k + Super + PBI Salary Packaging
- Hybrid working, South Melbourne office



McM Community Services & Development

### Donor Database and Insights Manager

Johnson Recruitment

South Melbourne, Melbourne VIC (Hybrid)  
\$90k - \$110k + Super + pbi Salary Packaging

Fundraising (Community Services & Development)

- Leading Community Support Organisation
- Salary \$90k - \$110k + Super + PBI Salary Packaging
- Hybrid working, South Melbourne office

# What is a Database?

- A large integrated collection of data, central to every enterprise/organisation managed through a software system, called Database Management System (DBMS)
- Models a real-world Universe of Discourse (UoD) – e.g. University operations – It consists of:
  - **Entities** (e.g., Students, Courses)
  - **Relationships** (e.g., Alice is enrolled in COMP9120)

# Key Questions on How Database should be Managed

- Do we want to write *custom* programs to access & manipulate database?
- How do we answer questions such as:
  - Count of students in a course?
  - Average workload of staff?
- How do we protect against concurrency anomalies (i.e., issues related to concurrent operations)?
- What if program crashes in the middle of operations?
- How do we make sure only the *right people* have the *right permissions* to *access* and *manipulate* the data?



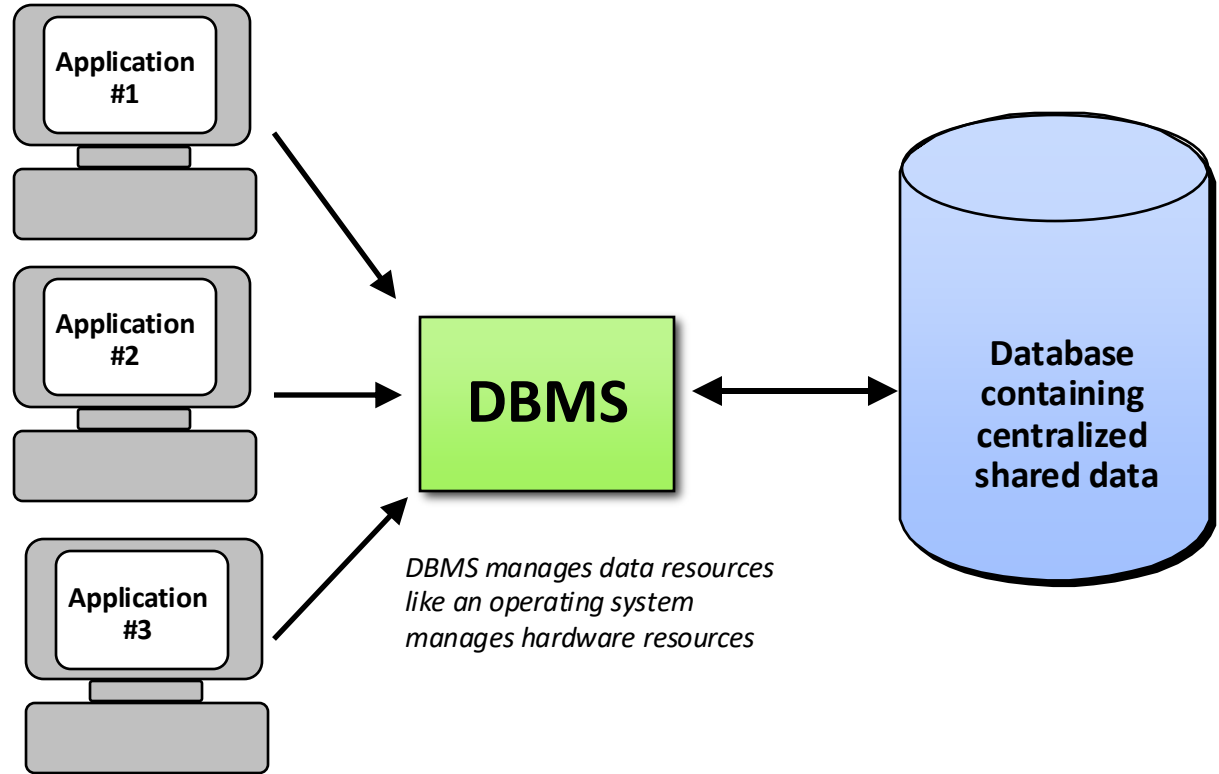
# Manage a Database by a DBMS!

## › Database Management System (DBMS)

- Stores the database on some mass (persistent) storage
  - Provide efficient storage along with fail safety (backup/recovery)
- Supports a high-level query language (e.g. SQL)
  - DBMS interprets statements of the query language to perform requested database access.
- Provides transaction management
  - Guarantee correct concurrent access to shared data

# The DBMS Approach

- Central repository of shared data
- Stored in a convenient and efficient form
- Data is centrally managed by a DBMS

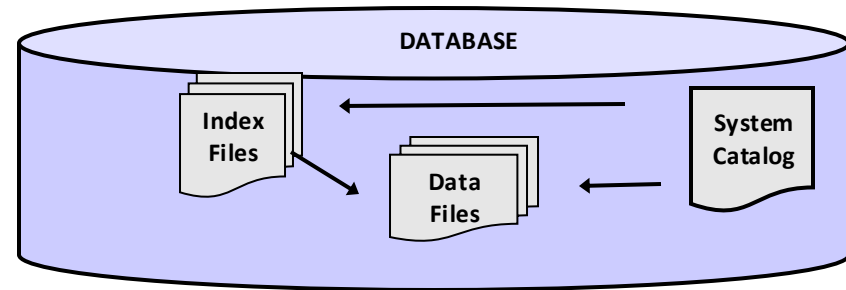
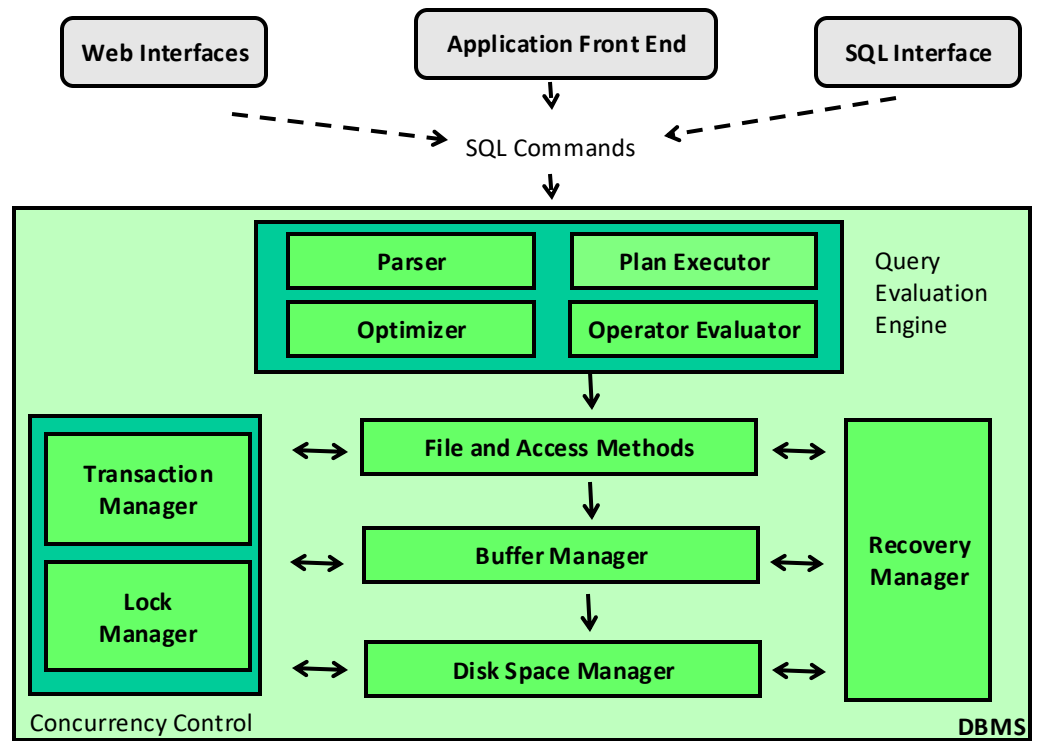


# Advantages of DBMS

- Improved Data Sharing & Security
  - Different users get different views of the data
- Enforcement of Standards
  - All data access is done in the same way
- Improved Data Quality
  - Integrity constraints, data validation rules
- Better Data Accessibility/ Responsiveness
  - Use of standard data query language (SQL)
- Backup/Recovery, Concurrency
  - Disaster recovery is easier

# Structure of a DBMS

- A typical DBMS has a **layered** architecture
- This is **one** of several possible architectures  
Each system has its own variations



# Relational Data Model

- › This unit focuses on **relational databases**, which use the *relational data model*: The most widespread among DB vendors
  - For other data models, please check COMP5338: Advanced Data Models
- › Definition: A **data model** is a *collection of concepts for describing data*
  - The relational data model is the *most widely used model*!
    - Main concepts: *relation* (essentially, a *table*), *functional dependencies* (e.g., *relationships*) and *schema* (collection of *related tables*): based on *set theory* and *functions*.

<i>Student</i>				
<u>sid</u>	name	email	gender	address
5312666	Jones	ajon@cs.com	M	123 Main St
5366668	Smith	paul@mail.com	M	45 George
5309650	Jin	jjin@it.com	F	19 City Rd



# Relational Database

- A relational database is a *collection of interrelated tables*, referred to as a *schema*
  - Tables are typically related to each other by *key attributes*
- Example: *Course management database* represented by the following schema:
  - Students(*sid: string, name: string, gpa: float*)
  - Courses(*cid: string, cname: string, credits: int*)
  - Enrolled(*sid: string, cid: string, grade: string*)

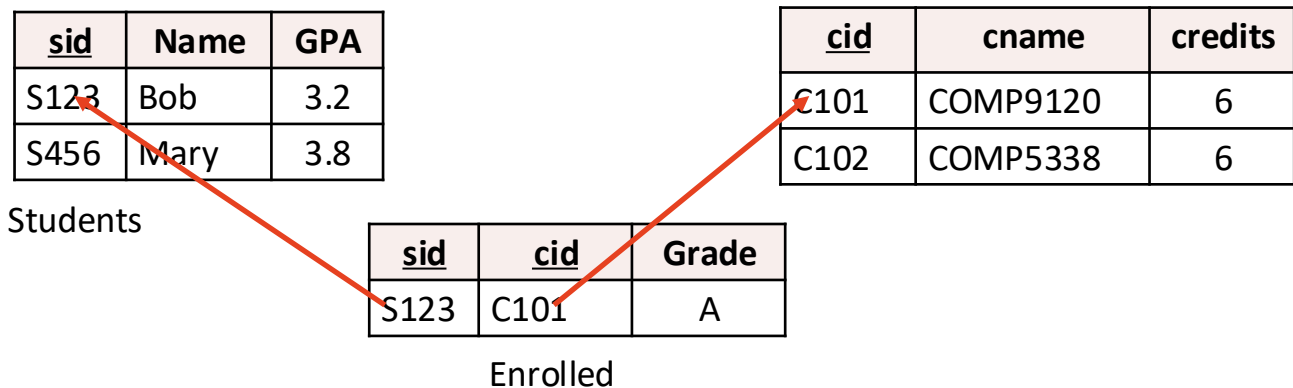
<u>sid</u>	Name	GPA
S123	Bob	3.2
S456	Mary	3.8

Students

<u>cid</u>	cname	credits
C101	COMP9120	6
C102	COMP5338	6

<u>sid</u>	<u>cid</u>	Grade
S123	C101	A

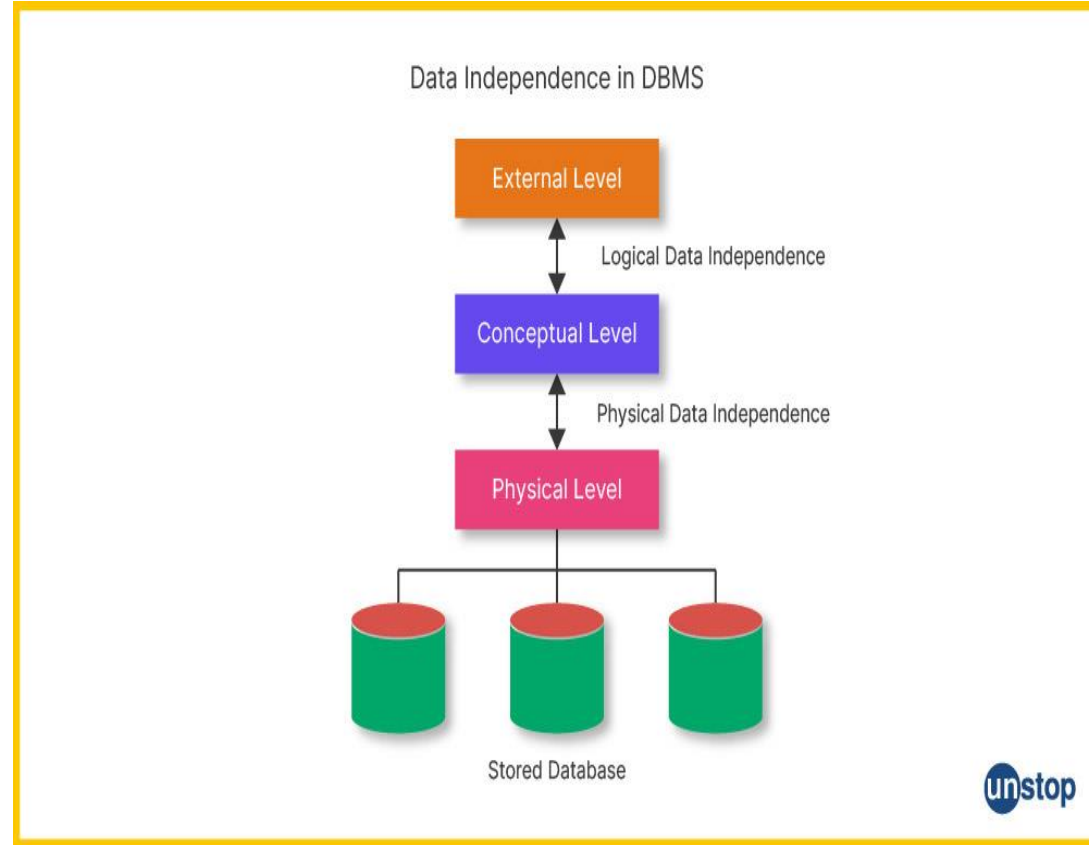
Enrolled



# Schema Levels in Databases

- Three (3) Levels:
  - External Schema/Application
  - Conceptual Schema
  - Physical Schema

## Schema Levels



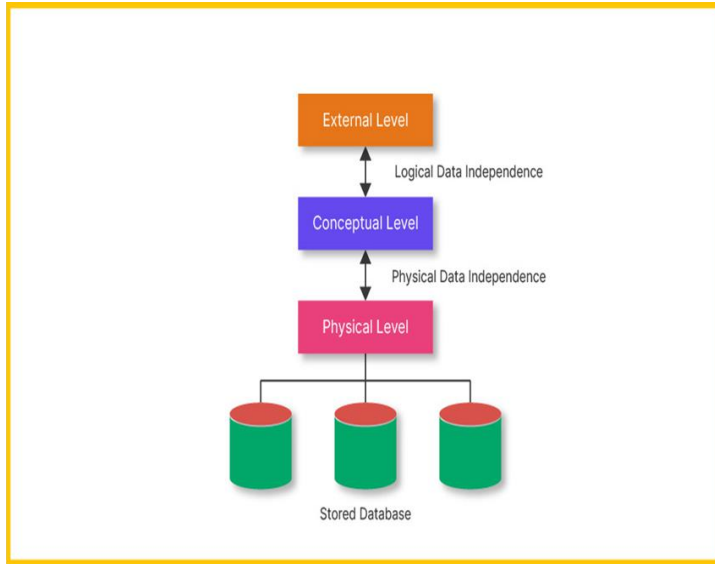
# DBMS Services

Main services:

- Data Independence
- Declarative Querying
- Transaction Management & Concurrency Control

# Data Independence

**Data Independence:** *Applications do not need to worry about how data is structured and stored*



<https://unstop.com/blog/data-independence-in-dbms>

**Logical data independence:** protection of the applications from changes in the *logical structure of the data*

*i.e., should not need to worry whether we can add a new entity or attribute without rewriting the application/view*

**Physical data independence:** protection of the conceptual schema (and applications) from *physical layout changes*

*i.e., should not need to worry which disks are the data stored on, or whether the data is indexed.*

# Declarative Querying: “What” not “How”

- It is convenient to indicate declaratively ***what*** information is needed, and leave it to the system to work out ***how*** to process through the data to extract what we need
- Users should be offered a way to express their requests *declaratively*
  - A query language which is based on *first-order logic* (e.g., *tuple calculus*)
    - SELECT...FROM...WHERE...

# Queries in a DBMS

- DBMS provides a query language combining 3 *specialized languages* for accessing data
  - **DML** - Data Manipulation Language - *Select, insert, update*
  - **DDL** - Data Definition Language - *Create, Alter, Drop*
  - **DCL** - Data Control Language - *Grant, Revoke*
- Standard Query Language for Relational DBMS: **SQL**
  - Based on formal query languages: Relational Algebra and Tuple Calculus

# SQL Example

- The *working-horse* command: **SELECT – FROM – WHERE**
  - *retrieves* data from one or more tables of a relational database that fulfill a *search condition*

Example 2:

```
SELECT *  
FROM Student
```

Student				
<u>sid</u>	name	email	gender	address
5312666	Jones	ajon@cs.com	M	123 Main St
5366668	Smith	paul@db.com	M	45 George
5309650	Jin	jin@it.com	F	19 City Rd

Example 2:

```
SELECT name, email  
FROM Student  
WHERE sid=5312666
```

name	email
Jones	ajon@cs.com

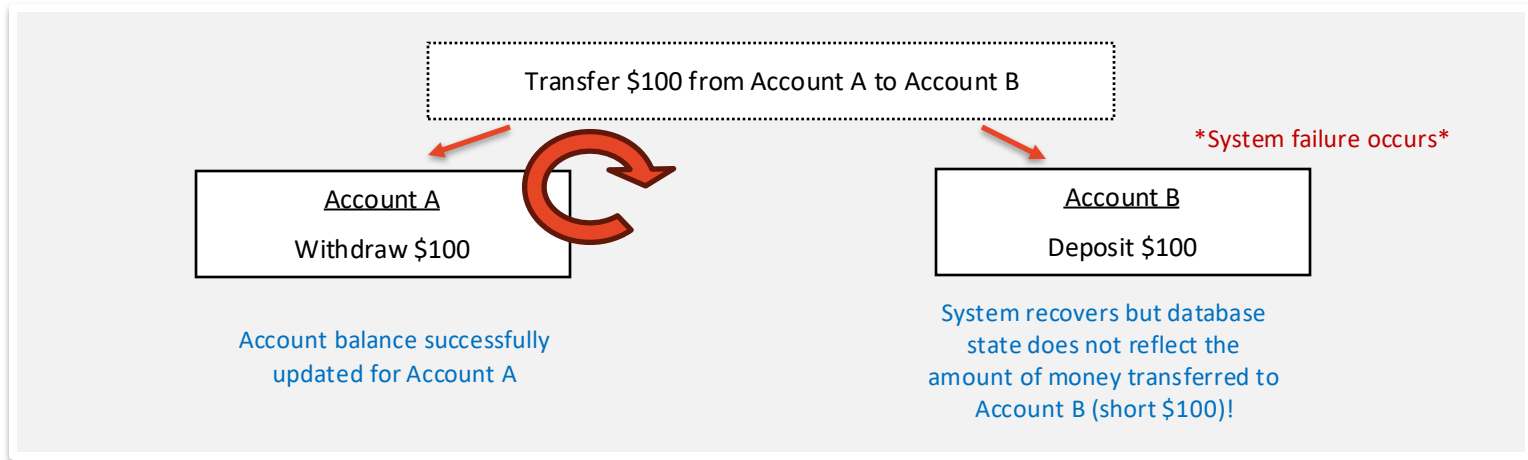
Example 1:

```
SELECT COUNT(*)  
FROM Student  
WHERE gender='F'
```

COUNT(*)
1

# What is a Transaction?

- “The basic unit of change as seen by a DBMS”  
(Ramakrishnan & Gherke, Database Management Systems, Ch1)
- This unit contains the execution of a piece of code that either executes *completely* or *not at all* (ie: **atomic execution**).
  - *No such thing as partially complete transactions!* This helps manage failure scenarios.



- Transaction atomicity is achieved via *locking* and *logging*.
- Concurrent control algorithms (e.g., *two-phase locking*) also help prevent interference between two different transaction executions that access & update the same data.



# Desirable DBMS Requirements

- **High Availability:** must be operational all the time or close to that.
- **High Reliability:** correctly tracks state, does not lose data, controlled concurrency
- **High Throughput:** accommodate many users => many transactions/sec
- **Low Response Time:** users don't wait for too long
- **Long Lifetime:** complex systems are not easily replaced – need to be *adaptive* to changes
  - Must be designed so DBMS can be easily extended as the needs of the enterprise change
- **Security:**
  - Sensitive information must be carefully protected since system is accessible to many users
    - Authentication, authorization, encryption

# Roles in DBMS

- **System Analysts**

- Specifies system using input from customer; provides complete description of functionality from customer's and user's point of view
- Conceptual database design

- **Database Designer**

- Specifies structure of data that will be stored in database ( logical & physical database schemas)

- **DB Application Programmer**

- Implements application programs (including transactions) that access data and support enterprise rules

- **Database Administrator (DBA)**

- Maintains database once system is operational: space allocation, performance optimization, database security, deals with failures and congestion

- **End-Users**

- Often unaware that they are dealing with data in a DBMS

# Summary

- DBMS is used to maintain & query large datasets that are shared by many application programs/users
- Some important concepts:
  - Logical and Physical Data Independence
  - Declarative Queries
  - Transactions
- Every 'knowledge worker' or scientists needs to have the database know-how, as do all IT experts- not just DBAs

# References

- Ramakrishnan/Gehrke (3rd edition)
  - **Chapter 1**
- Kifer/Bernstein/Lewis (2nd edition)
  - Chapters 1.1-1.3, 2.1, 2.2, 3.1, 3.2
  - Missing: comparison with file-based info system
- Ullman/Widom (3rd edition)
  - Chapters 1.1, 2.1, 2.2
  - Missing: comparison with file-based info system, roles of workers
- Silberschatz/Korth/Sudarshan (5th edition)
  - Chapters 1.1-1.5, 1.12, 2.1
- Tony Hey et. al (Ed.): *The Fourth Paradigm: Data-Intensive Scientific Discovery*, Microsoft Research, 2009
  - <http://research.microsoft.com/en-us/collaboration/fourthparadigm/>

## Next Week

- Conceptual Database Design using the
  - Entity Relationship Model
- Readings:
  - ***Ramakrishnan/Gehrke, Chapter 2***
  - Kifer/Bernstein/Lewis book, Chapter 4
  - Ullman/Widom, Chapter 4

See you next week!

- Winners of week1 Competition!



Denise  
*Entertainment*

<Thursday Winner to be Added>

# Please Share Your Opinion

Fully anonymous feedback form  
for COMP9120 semester2 2025  
students



<https://forms.office.com/r/Tf3sAha9hc>

# Updates

- 1.1

- Winner names added
- The placements of some text were fixed
- New break pages added, separating the two parts of the lecture
- Group formation pt2 is added to answer questions asked by students