COMP9120

Database Management Systems

Semester 2, 2022

Part A: Unit outline

Dr. Mohammad Polash
School of Computer Science

THE UNIVERSITY OF SYDNEY

Contents developed by Dr Lijun Chang

Acknowledgement of Country

I would like to acknowledge the Traditional Owners of Australia and recognise their continuing connection to land, water and culture. I am currently on the land of the Gadigal people of the Eora nation and pay my respects to their Elders, past, present and emerging.

I further acknowledge the Traditional Owners of the country on which you are on and pay respects to their Elders, past, present and future.

Copyright warning

Teaching Team

- Lecturer/Coordinator: Dr Mohammad Polash

  Professor Athman Bouguettaya
- Lecture: Thursday, 6-8 PM
- Tutors:
  - Iwan Budiman
  - Dhruvil Shah
  - Tommy Zhou
  - Dipankar Chaki
  - Mahdie Rezaein
  - Abbey Lin
  - Mehnaz Tabassum
- Tutorials: Thursdays, 8-9 PM, Friday 5-6 PM, 6-7 PM, 7-8 PM

Platforms used

- Canvas
  - Upload lecture slides, tutorial sheets, zoom recordings
  - Receive assignment submissions, conduct quizzes
  - Publish grades
- Ed
  - Discussion Forum
  - Conduct SQL challenge

What this unit of study is (and is not)

- We'll cover **how database management systems work**
  - But not **how to build** them

- Discuss **fundamentals of data management**
  - How to design databases, query databases, build applications with them.
  - Not how to be a DBA or how to tune a DBMS (e.g., oracle, postgresql).

THE UNIVERSITY OF SYDNEY

Outline of Lectures

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

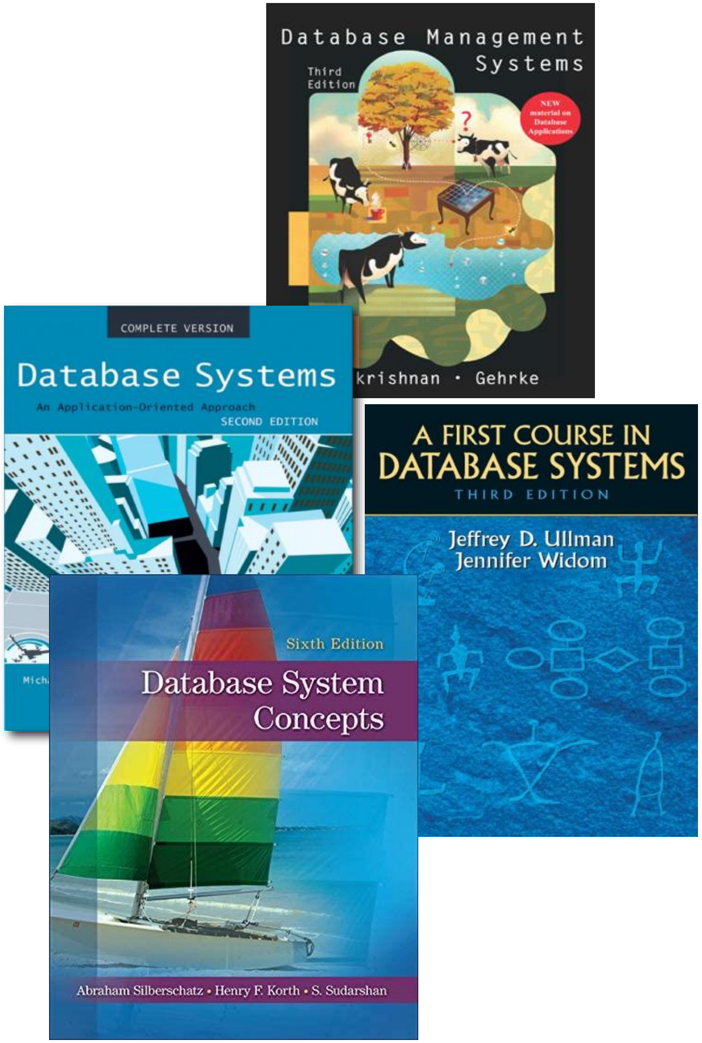
Textbooks

Prescribed Textbook:

- R. Ramakrishnan and I. Gehrke: ***Database Management Systems***, 3rd ed., McGraw-Hill, 2003.
  - Click on ReadingList in Canvas

Several Recommended Texts:

- A. Silberschatz, H. Korth, S Sudarshan: ***Database System Concepts***, McGraw-Hill, 2010
- J.D. Ullman, and J.Widom: ***A First Course in Database Systems***, 3rd ed., Prentice-Hall, 2008.
- M. Kifer, A. Bernstein, and P.M. Lewis: ***Database Systems: An Application-Oriented Approach***. Complete version, 2nd edition, Pearson/Addison Wesley, 2006.
- Suggested additional SQL reference:
  - J.S. Bowman. *The Practical SQL Handbook*. 4th edition, A-W. 2001.



Software

- We will be using various software/languages in the labs & assignments:
  - PostgreSQL
  - PowerPoint / Other drawing package for conceptual modelling
  - Java/Python

How Much Programming is Involved?

- Although you do not need to be a hard-core programmer, you will need some programming skills for the practical assignment
  - it is not a programming course,
  - but covers database design, creation and usage

- The DB programming assignment (**Assignment 2**) assumes basic programming experience and expects some competence in **Java or Python**!

- It is your responsibility to learn Java/Python

Assessment Package

› Assessment tasks

1. SQL Challenge	
- Week 6	10%
2. Quiz	
- Week 9	14%
2. **Group** Assignments	
- Week 8: DB Schema Implementation	13%
- Week 13: DB Application Programming	13%
3. Final Exam	50%

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

*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 early as possible.
  - First assignment is released on Week 4, groups must be formed by Week 3.
    - At the start of week 4, we will randomly create groups among unassigned students.
  - To ensure the contribution of each member, we will conduct random interviews

- Group must organize itself
  - Arrange internal communication and meetings
  - Set internal deadlines
  - Follow up if deadlines are missed
  - Have fall-back mechanism
  - Have dispute resolution mechanism

- Let your unit coordinator / lecturer know if there are problems (unresolvable disagreements, member who doesn't contribute etc.)

Late Submission

- Late submission will incur **a penalty of 5% per day**.
  - A good 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.**
- Warning: submission sites get very slow near deadlines
- Submit early; you can resubmit if there is time before the deadline

Special Consideration (University policy)

- If your performance on assessments is affected by illness or misadventure
- Follow proper bureaucratic 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/
- Also, notify coordinator by email *as soon as anything begins to go wrong*
- There is a similar process if you need special arrangements eg for religious observance, military service, representative sports

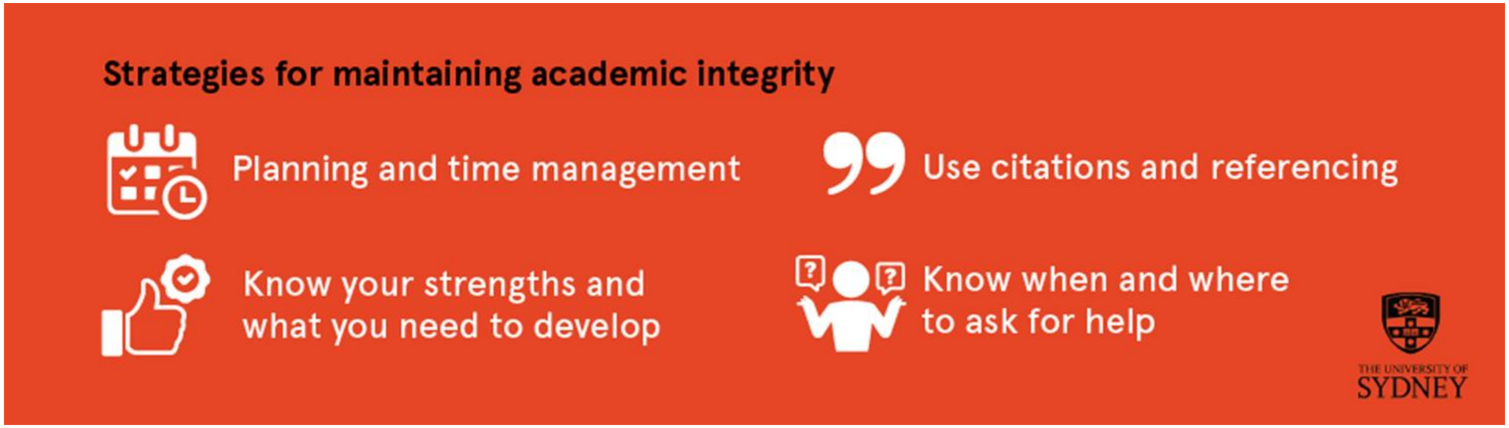
Academic integrity

- 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 by census date. Continuing students can self-enrol at any time.



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

THE UNIVERSITY OF SYDNEY

Academic integrity

- “The University of Sydney is unequivocally opposed to, and intolerant of, plagiarism and academic dishonesty.
  - Academic dishonesty means seeking to obtain or obtaining academic advantage for oneself or for others (including in the assessment or publication of work) by dishonest or unfair means.
  - Plagiarism means presenting another person’s work as one’s own work by presenting, copying or reproducing it without appropriate acknowledgement of the source.”

[source: http://sydney.edu.au/elearning/student/EI/index.shtml]

- **Submitted work is compared against other work (from students, the internet, etc)**
  - TurnitIn for textual tasks (through Canvas), other systems for code
- **Penalties for academic dishonesty or plagiarism can be severe**

Source of Help

Lecturer	Teaching Assistants / Tutors	Classmates	Private tutors	Online forums/ Online tutors	Students outside course/UoS	Hired coders Tutorial Company outside University	Relatives	Other
🟩	🟩	🟨	▨	🟥	🟥	🟥	🟥	▨

- Individual assessment
- A student needs to gain an understanding of fundamental knowledge/skills
- It is important to master the knowledge/skills themselves

🟩 Encouraged

🟨 Attribution required

🟥 Not acceptable

▨ Ask Lecturer/Coordinator

THE UNIVERSITY OF SYDNEY

Types of Help

Understanding General Concepts	**Explained using similar material (not assignment)**	**Sharing approach/concept to derive assignment solution**	**Designing code/solution**	**Implementing code/solution**

- Individual assessment
- A student needs to gain an understanding of fundamental knowledge/skills
- It is important to master the knowledge/skills themselves
- Students are encouraged to obtain help through relevant teaching materials and practices but not directly on assessment materials

🟩 Encouraged

🟨 Attribution required

🟥 Not acceptable

⬜ Ask Lecturer/Coordinator

WHS Induction

School of Computer Science

Keeping our campus COVID safe

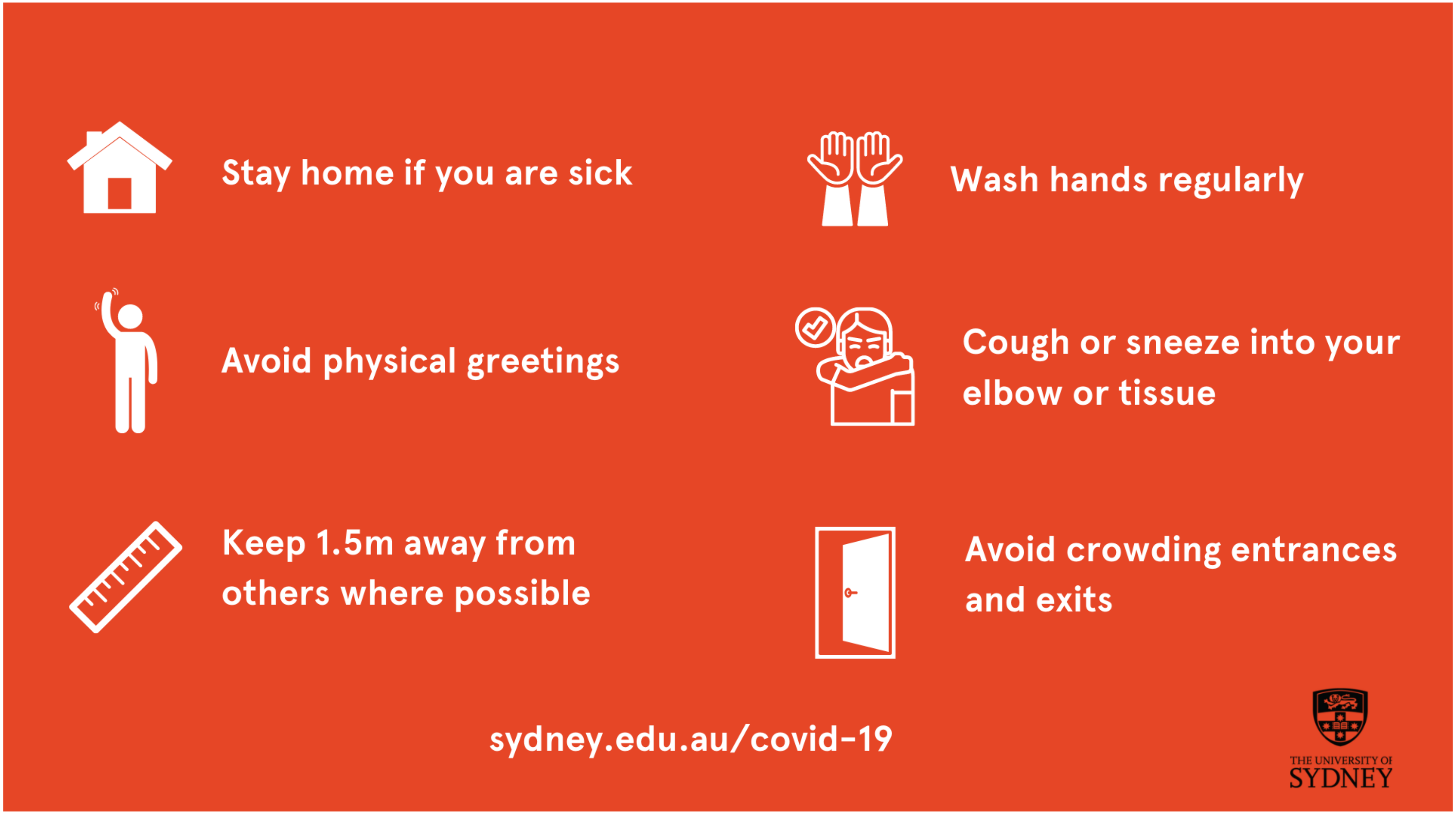
- The University is following NSW Government and NSW Health guidance as a minimum standard in our response to the COVID-19 pandemic.

- NSW Government restrictions can change at short notice.

- Check your student email for updates about University operations and COVID safety precautions.

- Visit our website: sydney.edu.au/covid-19

Follow COVID safety precautions



Stay home if you are sick

Wash hands regularly

Avoid physical greetings

Cough or sneeze into your elbow or tissue

Keep 1.5m away from others where possible

Avoid crowding entrances and exits

sydney.edu.au/covid-19

THE UNIVERSITY OF SYDNEY

Feeling unwell?

- Stay at home
  - if you are feeling unwell with any COVID-19 symptoms
  - If you have been directed to self-isolate
- Get tested
  - If you are feeling unwell with COVID-19 symptoms, please get tested as soon as possible
- Did you test positive?

  Yes? If you have visited campus within the infectious period, i.e. 72 hours before taking the test, you must advise the University via:

    - email covid19.taskforce@sydney.edu.au, or
    - call +61 2 9351 2000 (select option 1)
- Stay informed
  - Monitor the list of confirmed COVID case locations on campus page to check for potential exposure and follow NSW Health isolation and testing requirements.

COVID-19 support and care

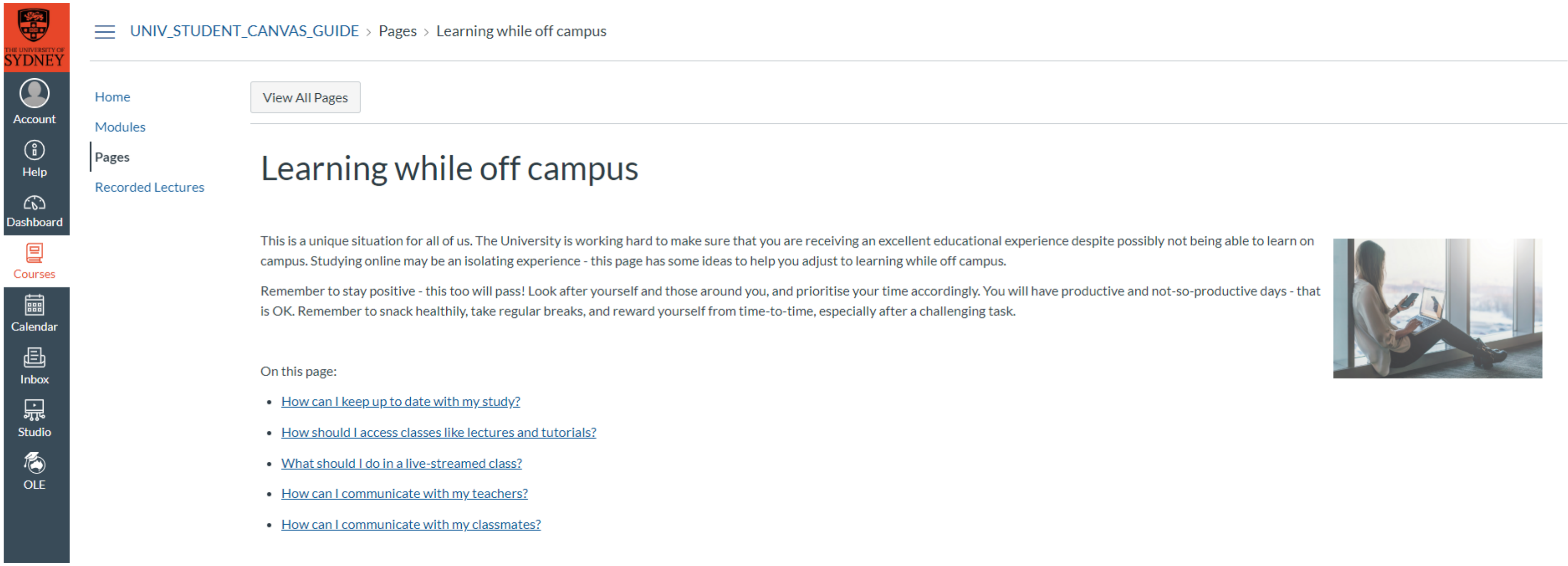
- Most large lectures will be delivered online and accommodations will be made for international students who have not yet returned to Australia.
- If you become infected with COVID-19 during the semester, or need to isolate, please notify your unit of study coordinator, as with any unexpected absence.
- If COVID-19 isolation or illness impacts assessment, use the usual mechanisms including simple extensions and special consideration to arrange reasonable adjustments. Visit https://www.sydney.edu.au/covid-19/students/study-information/test-exams-assessment.html#consideration.
- Further information on student support can be found on the University website at https://www.sydney.edu.au/students/support.html
- Other helpful study information can be found on the website at https://www.sydney.edu.au/covid-19/students/study-information.html.

Tips for students learning online

- Remember that you are still in a space with other students.
- Mute your microphone when not speaking.
- Use earphones or headphones - the mic is better, and you'll disturb others less.
- If you have a webcam, please switch it on so we can see you, if you are comfortable doing so.
- Try not to talk over someone else.
- Some classes may use breakout rooms – engaging fully in these is a great way to meet classmates and your teachers.
- Help your teachers know you're there by participating in chat, polls and other activities during class - we're all in this together.

Tips for learning online

- For tips and guides on learning online and the tools you will use, refer to Learning while off campus resources in Canvas. This is especially useful if it's your first time learning online at university.



Learning while off campus

This is a unique situation for all of us. The University is working hard to make sure that you are receiving an excellent educational experience despite possibly not being able to learn on campus. Studying online may be an isolating experience - this page has some ideas to help you adjust to learning while off campus.

Remember to stay positive - this too will pass! Look after yourself and those around you, and prioritise your time accordingly. You will have productive and not-so-productive days - that is OK. Remember to snack healthily, take regular breaks, and reward yourself from time-to-time, especially after a challenging task.

On this page:

- How can I keep up to date with my study?
- How should I access classes like lectures and tutorials?
- What should I do in a live-streamed class?
- How can I communicate with my teachers?
- How can I communicate with my classmates?

Emergency procedures (on campus)

- In the unlikely event of an emergency, we may need to evacuate the building.

- If we need to evacuate, we will ask you to take your belongings and follow the green exit signs.

- We will move a safe distance from the building and maintain physical distancing whilst 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/emergency.

School of Computer Science Safety Contacts

CHIEF WARDEN

Greg Ryan
Level 1W 103
9351 4360

FIRST AID OFFICERS

Julia Ashworth
Level 2E Reception
8627 9058

Will Calleja
Level 1W 103
9036 9706

Cecille Faraizi
Level 2E 237
9351 6060

Orally REPORT all
INCIDENTS
& HAZARDS
to your SUPERVISOR

OR

Coursework Postgraduates:	Julia Ashworth 8627 9058 or Keiko Narushima 8627 0872
CS School Manager:	Priyanka Magotra 8627 4295

Do you have a disability?

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.

The types of disabilities we see include:
Anxiety // Arthritis // Asthma // Autism // ADHD
Bipolar disorder // Broken bones // Cancer
Cerebral palsy // Chronic fatigue syndrome
Crohn’s disease // Cystic fibrosis // Depression Diabetes // Dyslexia // Epilepsy // Hearing impairment // Learning disability // Mobility impairment // Multiple sclerosis // Post-traumatic stress // Schizophrenia // Vision impairment and much more.

Students needing assistance must register with Disability Services. It is advisable to do this as early as possible. Please contact us or review our website to find out more.

THE UNIVERSITY OF SYDNEY

Disability Services Office

sydney.edu.au/disability

02-8627-8422

Other support

- Student support
  - https://sydney.edu.au/campus-life/health-wellbeing-success.html
- Learning support
  - http://sydney.edu.au/study/academic-support/learning-support.html
- International students
  - http://sydney.edu.au/study/academic-support/support-for-international-students.html
- Aboriginal and Torres Strait Islanders
  - http://sydney.edu.au/study/academic-support/aboriginal-and-torres-strait-islander-support.html
- Student organization (can represent you in academic appeals etc)
  - http://srcusyd.net.au/ or http://www.supra.net.au/
- Please make contact, and get help
- You are not required to tell anyone else about this
- If you are willing to inform the unit coordinator, they may be able to work with other support to reduce the impact on this unit
  - e.g. provide advice on which tasks are most significant

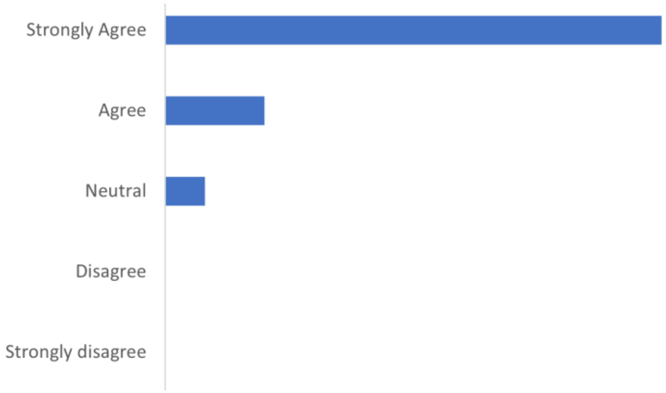
Two-way assessment

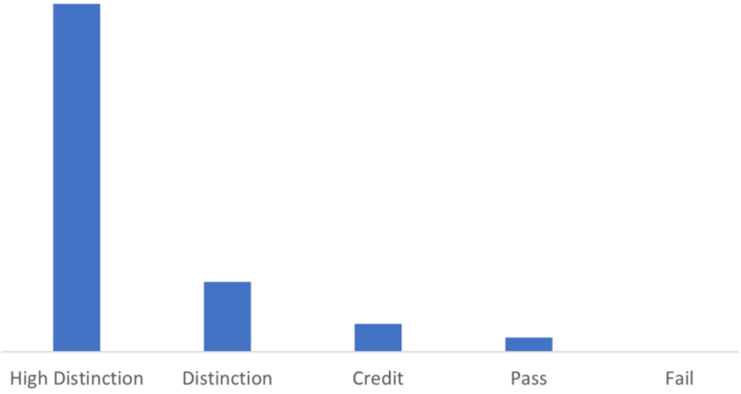
Teaching Team

- We aim to win your opinion to ‘Strongly Agree’ in all the teaching criteria via Unit of Sydney Survey

Students

>= 85% Mark High Distinction
>= 75% Mark Distinction
>= 65% Mark Credit
>= 50% Mark Pass





Some Advice from Past Students

- "If you skip any lectures/lab, set aside an equal amount of time to go over the slides/work *in full*. I mostly just scanned over the lecture notes, which was good enough for homeworks etc, but come revision time it meant I had a lot more to cram in than was pleasant."

- "Draft out your solution to the final assignment in the same week as it's released. The 'crunch' trying to get our submission up to scratch in the couple of days before the deadline was an entirely unnecessary suffering that we inflicted on ourselves ;)"

- "Submit your homework a day in advance. Actually, apply this 'trick' to all assignments, in all subjects."

Time for a break!

When you come back, please complete the poll

Part B: Introduction

THE UNIVERSITY OF SYDNEY

What is a Database?

- A large integrated collection of data, central to every enterprise/organisation

- Models a real-world *enterprise*
  - Entities (e.g., Students, Courses)
  - Relationships (e.g., Alice is enrolled in COMP9120)

Databases are Everywhere!

- Banking systems:
  - accounts & loans, customers, all transactions (banks, ATMs, internet)
- 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 this
  - Human resources: employee records, salaries, tax deductions
  - Manufacturing: production, inventory, orders, supply chain
- Telecommunication: calls, bills, calling/SIM cards
- Health care: patients, prescriptions, drugs, …

Databases touch all aspects of our lives

How to Manage a Database?

- Store database in files; write custom programs to access & manipulate database?

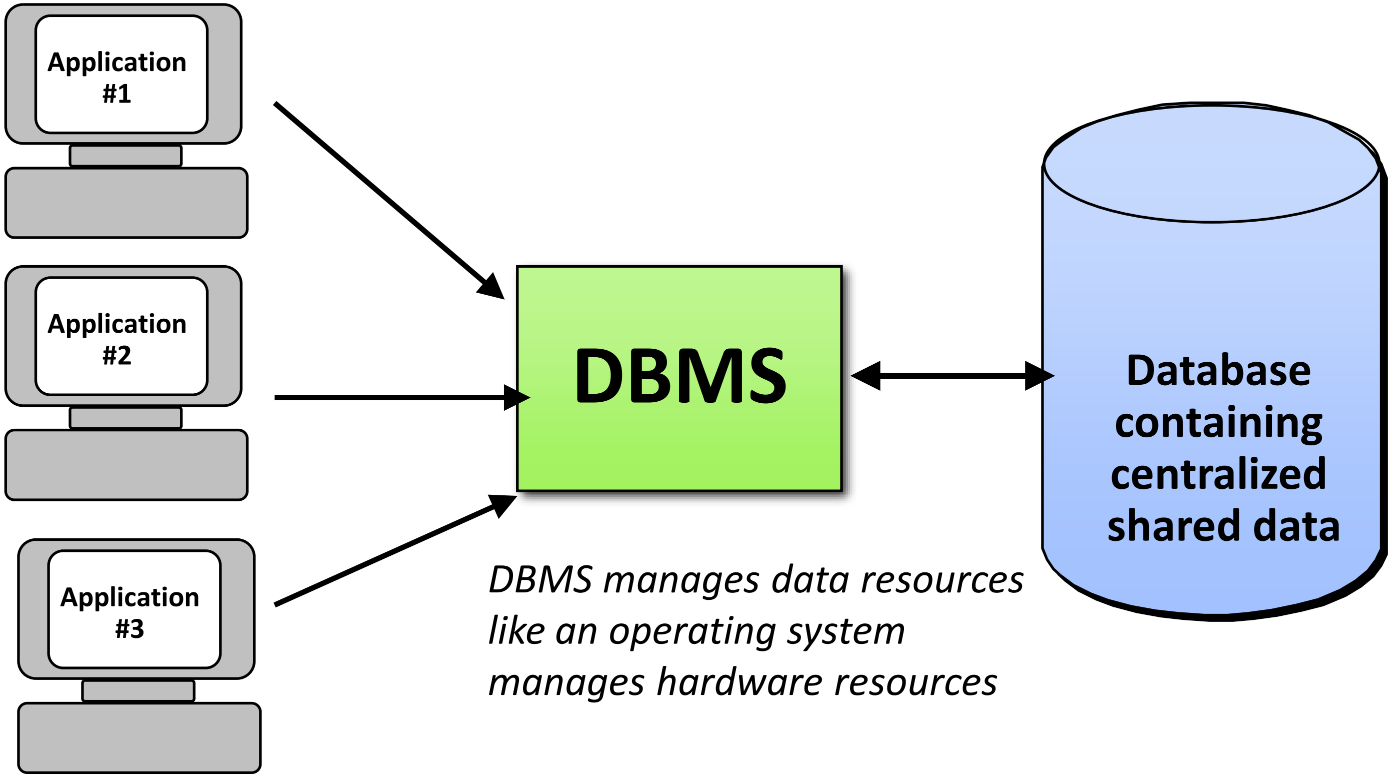
- How do we answer questions?
  - eg: Count of students in a course?
  - eg: Average workload of the staffs?
- How do we protect against concurrency anomalies?
- What if program crashes in middle of operations?
- How do we make sure only the right people have the right permissions to certain subsets of the data?

Manage a Database by DBMS!

- **Database Management System (DBMS)**
  - Stores the database on some mass (persistent) storage providing fail safety (backup / recovery)
  - Supports a high-level access language (e.g. SQL)
    - Application describes database accesses using that language.
    - DBMS interprets statements of language to perform requested database access.
- Provides transaction management to guarantee correct concurrent access to shared data

The DBMS Approach

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



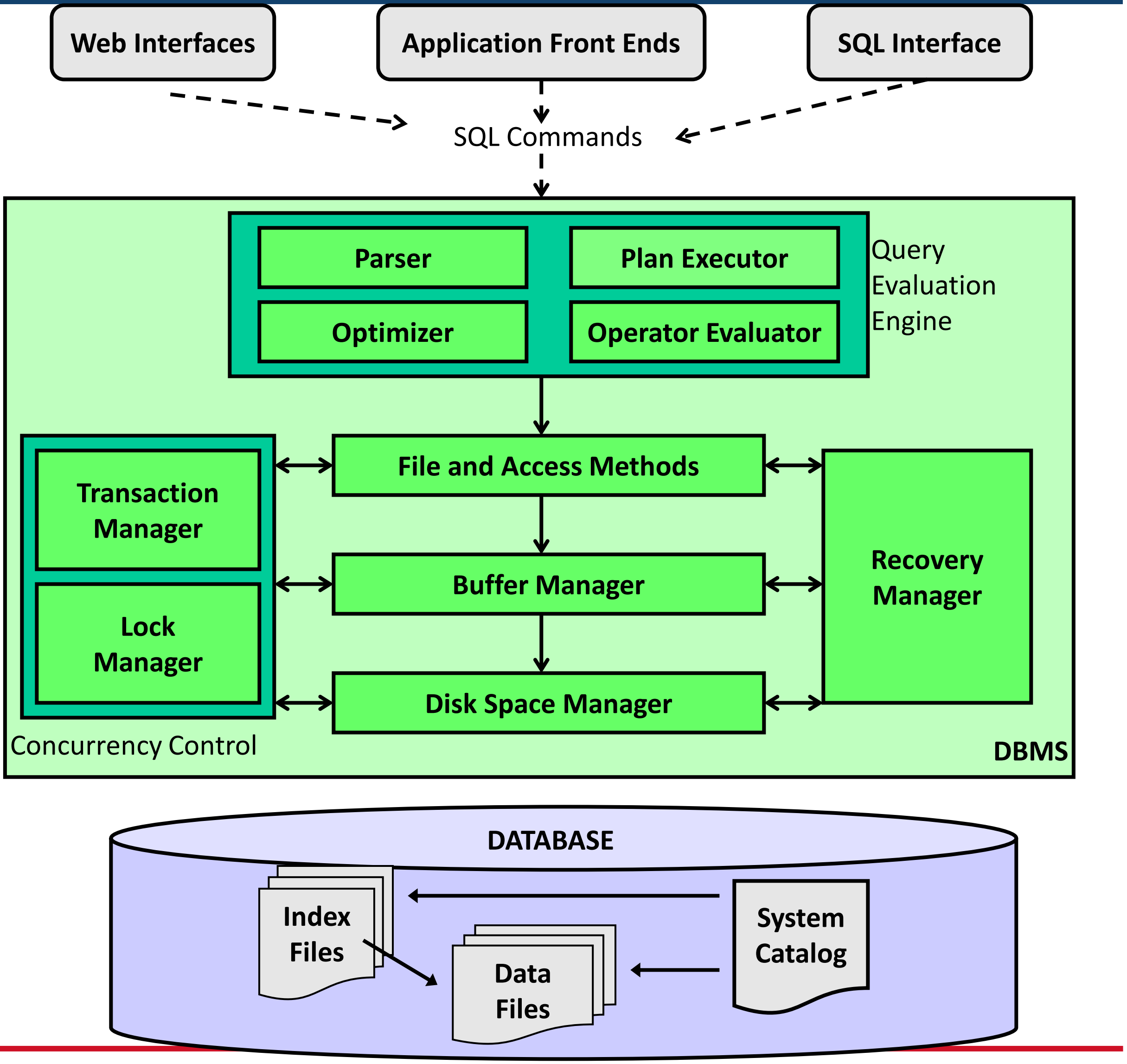
DBMS manages data resources like an operating system manages hardware resources

Advantage 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 and are now widespread with many DB vendors
  - For other data models, please check COMP5338: Advanced Data Models

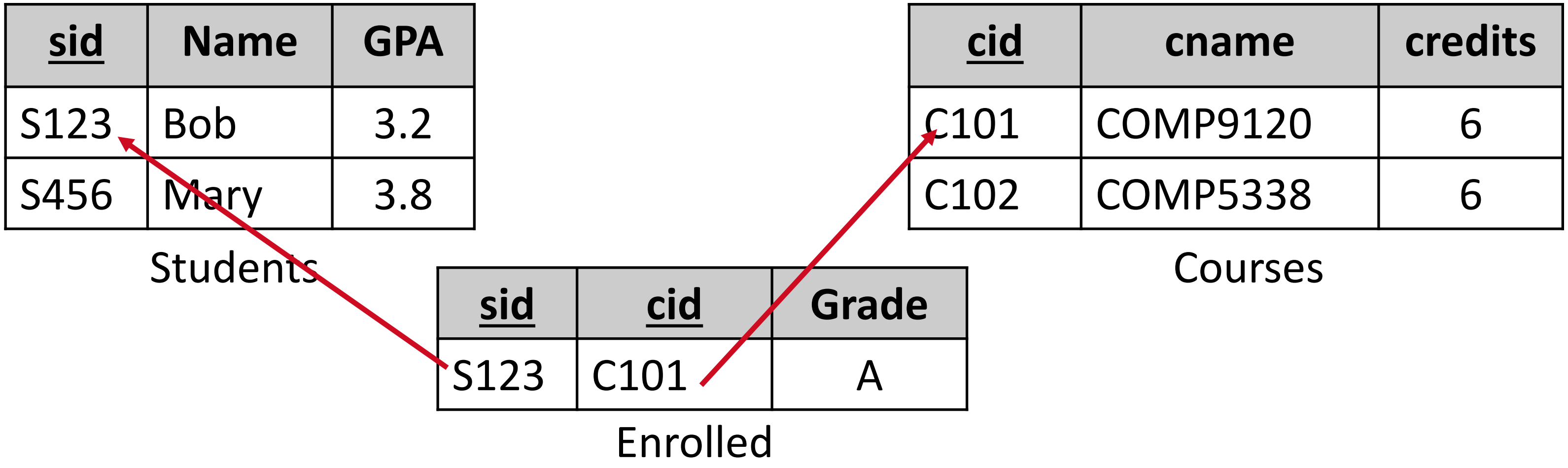
- A **data model** is a collection of concepts for describing data
  - The relational model of data is the most widely used model today
    - Main Concept: the *relation* (essentially, a table) and *schema*

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

Relational Database

- A relational database is a collection of such tables
  - which typically are related to each other by key attributes

- *Example: course management database with schema*
  - Students(sid: *string*, name: *string*, gpa: *float*)
  - Courses(cid: *string*, cname: *string*, credits: *int*)
  - Enrolled(sid: *string,* cid*: string,* grade*: string*)



sid	**Name**	**GPA**
S123	Bob	3.2
S456	Mary	3.8

Students

cid	**cname**	**credits**
C101	COMP9120	6
C102	COMP5338	6

Courses

sid	**cid**	**Grade**
S123	C101	A

Enrolled

Central DBMS Services

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

Data Independence

Applications do not need to worry about *how the data is structured and stored*

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

i.e. should not need to ask: can we add a new entity or attribute without rewriting the application?

Physical data independence: protection from *physical layout changes*

i.e. should not need to ask: which disks are the data stored on? Is the data 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 you need

- Users should be offered a way to express their requests declaratively
  - A query language can be based on logic
  - Select…from…where…

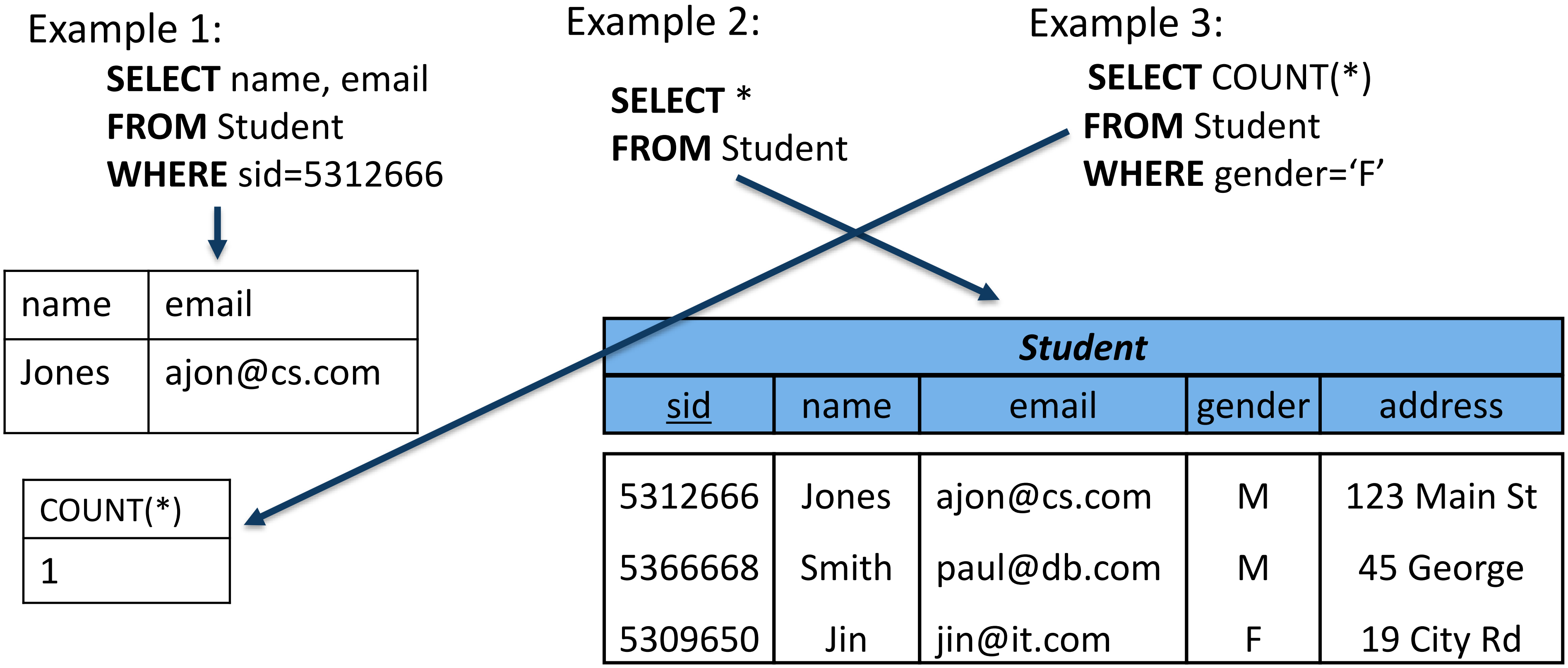
Queries in a DBMS

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

THE UNIVERSITY OF SYDNEY

SQL Example

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



Example 1:

SELECT name, email
FROM Student
WHERE sid=5312666

name	email
Jones	ajon@cs.com

Example 2:

SELECT *
FROM Student

Example 3:

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

COUNT(*)
1

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

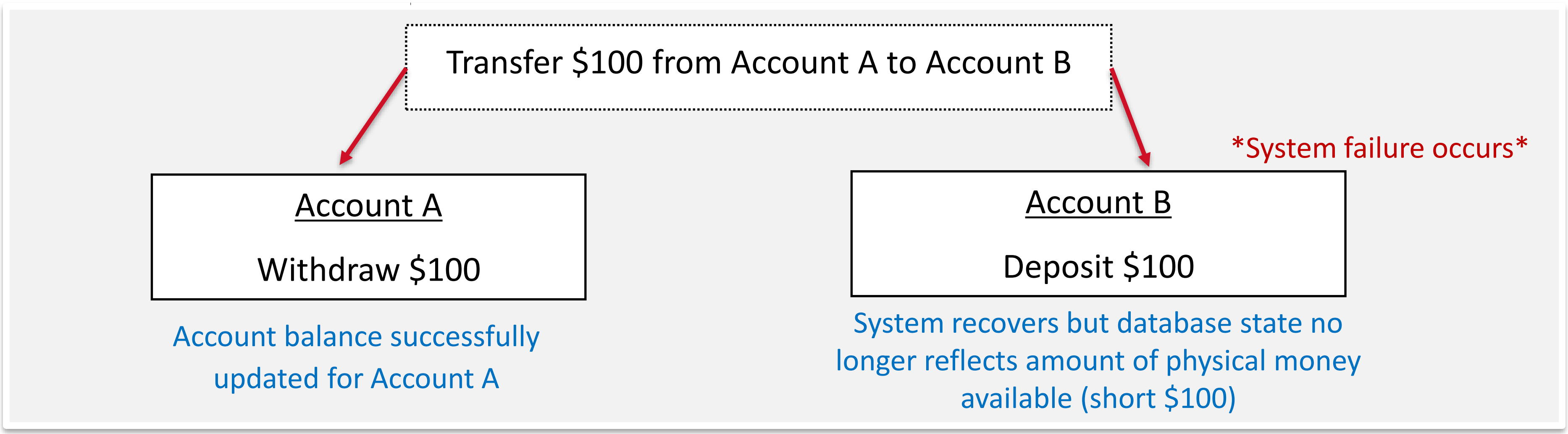
THE UNIVERSITY OF SYDNEY

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.



Transfer $100 from Account A to Account B

System failure occurs

Account A

Withdraw $100

Account balance successfully updated for Account A

Account B

Deposit $100

System recovers but database state no longer reflects amount of physical money available (short $100)

- Transactions help prevent interference between two code executions that access & update the same data.
  - Transactions achieve this via logging & locking.

DB System Requirements

- **High Availability**: on-line => must be operational while enterprise is functioning
- **High Reliability**: correctly tracks state, does not loose data, controlled concurrency
- **High Throughput**: many users => many transactions/sec
- **Low Response Time**: on-line => users are waiting
- **Long Lifetime**: complex systems are not easily replaced
  - Must be designed so they 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 (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:
  - Application-Data Independence
  - Declarative Queries
  - Transactions

- Every ‘knowledge worker’ or scientists needs 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!

THE UNIVERSITY OF SYDNEY