COMP9120

Database Management Systems

Semester 2, 2016

Dr. Matthew Sladescu
Based on material by Dr. Bryn Jeffries

Welcome to COMP9120

- Lecturer: Matthew Sladescu (matthew.sladescu@sydney.edu.au)
- Lectures: Thursdays, 6-8pm, New Law School Lecture Theatre 024

- Tutors: Waiho Wong, Iwan Budiman, James Phillips, Archie Qi Huang
  Introductions in tonight's tutorial, after the lecture!
- Tutorials: Thursdays, 8-9pm, SIT Labs 115, 116, 117, 118

- Course Website:
  - USYD E-Learning Website (elearning.sydney.edu.au)

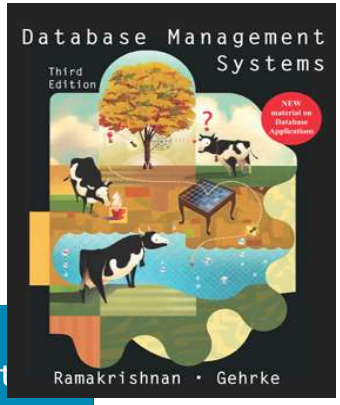
Outline of Lectures

	Week	Topic
DB Foundations	Week 1	Introduction
	Week 2	Conceptual Database Design
	Week 3	Relational Data Model / Logical Database Design
	Week 4	Relational Algebra and introduction to SQL
	Week 5	Database Integrity
	Week 6	Complex SQL
DB Applications	Week 7	Schema Refinement and Normalisation
	Week 8	Database Application Development and Security
	Week 9	Transaction Management
	Study Vacation (September 26 – October 30)	
	Week 10	Storage and Indexing
	Week 11	Query Evaluation
	Week 12	Query Optimisation
	Week 13	Unit of Study Review

Textbooks

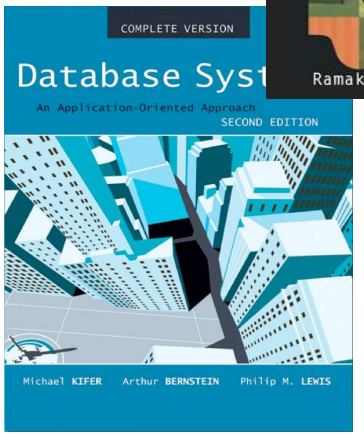
Prescribed Textbook:

- R. Ramakrishnan and I. Gehrke: ***Database Management Systems***, 3rd ed., McGraw-Hill, 2003.
  - Available from your Campus bookshop
  - Copies are available in SciTech library

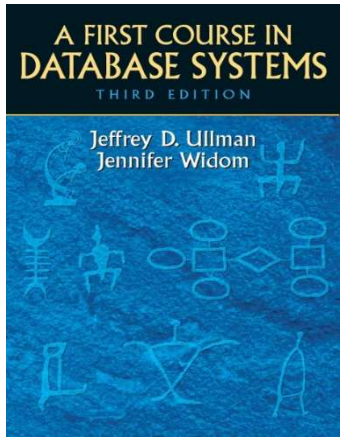


Several Recommended Texts:

- M. Kifer, A. Bernstein, and P.M. Lewis: ***Database Systems: An Application-Oriented Approach***. Complete version, 2nd edition, Pearson/Addison Wesley, 2006.



- J.D. Ullman, and J.Widom: ***A First Course in Database Systems***, 3rd ed., Prentice-Hall, 2008.



- Suggested additional SQL reference:
  - J.S. Bowman. *The Practical SQL Handbook*. 4th edition, A-W. 2001.

Of course, there are many more good text books on databases…

e.g. Silberschatz/Korth/Sudarshan: *Database System Concepts* (6thed), McGraw-Hill, 2010

General Resources

- We will use eLearning (Blackboard) to
  - Publish lecture slides
  - Distribute tutorial handouts and background material
  - Forming Teams (signup sheets)
  - Receive assignment submissions
  - Publish grades
  - Publish weekly homework questions – starting next week!

- Supporting resources
  - Centre for English Teaching (http://sydney.edu.au/cet/)
  - External self-study Mini Courses on Databases:
    - https://class.stanford.edu/courses/DB/2014/SelfPaced/about

  (Introduction to Databases, Jennifer Widom, Stanford)

Assessment Package

- Assessment tasks

  1. Online E-Learning Assessments:

     - Weekly auto-marked homework (beginning in Week 2) 10%

  2. Assignments

     - Week 6: Conceptual Design – Form Groups by Week 2! 10%
     - Week 9: DB Schema Implementation 10%
     - Week 12: DB Application Programming 10%

  3. Exam 60%

- Marks will be published on eLearning

  -Report any errors or omissions within **10 days**! After that marks are fixed

- You must obtain at least **40% in the exam**, as well as an **overall mark of at least 50%**, to pass the unit

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."

Software

- **One** objective of the course is to give some hands-on experience with existing database software
  - This course includes both theoretical and practical work.
  - However note: The focus is on teaching the principles, not software!

- We will be using various software/languages in the labs & assignments:
  - Oracle 12c & SQLDeveloper
  - Java/JDBC
  - For SQL tutorials/feedback, we will use SIT's **Challenge** system (will be setup soon)
    - continuing set of exercises each week – please use it regularly: will start later in semester
      Uses PostgreSQL and SQLite, so be aware of syntax differences
- **It is your responsibility to learn how to use them**
  - Installed in the SIT labs
  - Documentation is available on-line
  - Systems introduced in tutorials

How Much Programming Is Involved?

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

- However:
  The DB programming project, assumes basic programming experience and expects some competence in Java!

Applying for Special Consideration

› In case of **Illness** or **Misadventure**

- you can apply for special consideration
- the application has to be lodged with your faculty **within 3 working days** from the due date of the assignment

http://sydney.edu.au/engineering/student-policies/special-consideration.shtml

The first thing you should do:

- **Let your lecturer know** (best by email and as soon as possible)
- Get a certificate from a Professional Practitioner
- Lodge the application for special consideration at your Faculty

› Please note: No special consideration for missing out a few days or being on holiday etc.

- time management is your responsibility!

THE UNIVERSITY OF SYDNEY

Disability Services

Website: sydney.edu.au/disability
Queries: 8627 8422

Do you have a disability?

You may not think of yourself as having a 'disability' but the definition under the **Disability Discrimination Act** 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 ÷ asperger's disorder ÷ 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.

Any Questions about the Unit's Organisation?

Academic dishonesty and plagiarism

- Please read the University policy on Academic Honesty carefully:

  http://sydney.edu.au/elearning/student/EI/academic_honesty.shtml
- All cases of academic dishonesty and plagiarism will be investigated
- There is a new process and a centralized University system and database
- Three types of offenses:
  - Plagiarism – when you copy from another student, website or other source. This includes copying the whole assignment or only a part of it.
  - Academic dishonesty – when you make your work available to another student to copy (the whole assignment or a part of it). There are other examples of academic dishonesty.
  - Misconduct - when you engage another person to complete your assignment (or a part of it), for payment or not. This is a very serious matter and the Policy requires that your case is forwarded to the University Registrar for investigation.

OHS INDUCTION

School of Information Technologies

THE UNIVERSITY OF SYDNEY

General Housekeeping – Use of Labs

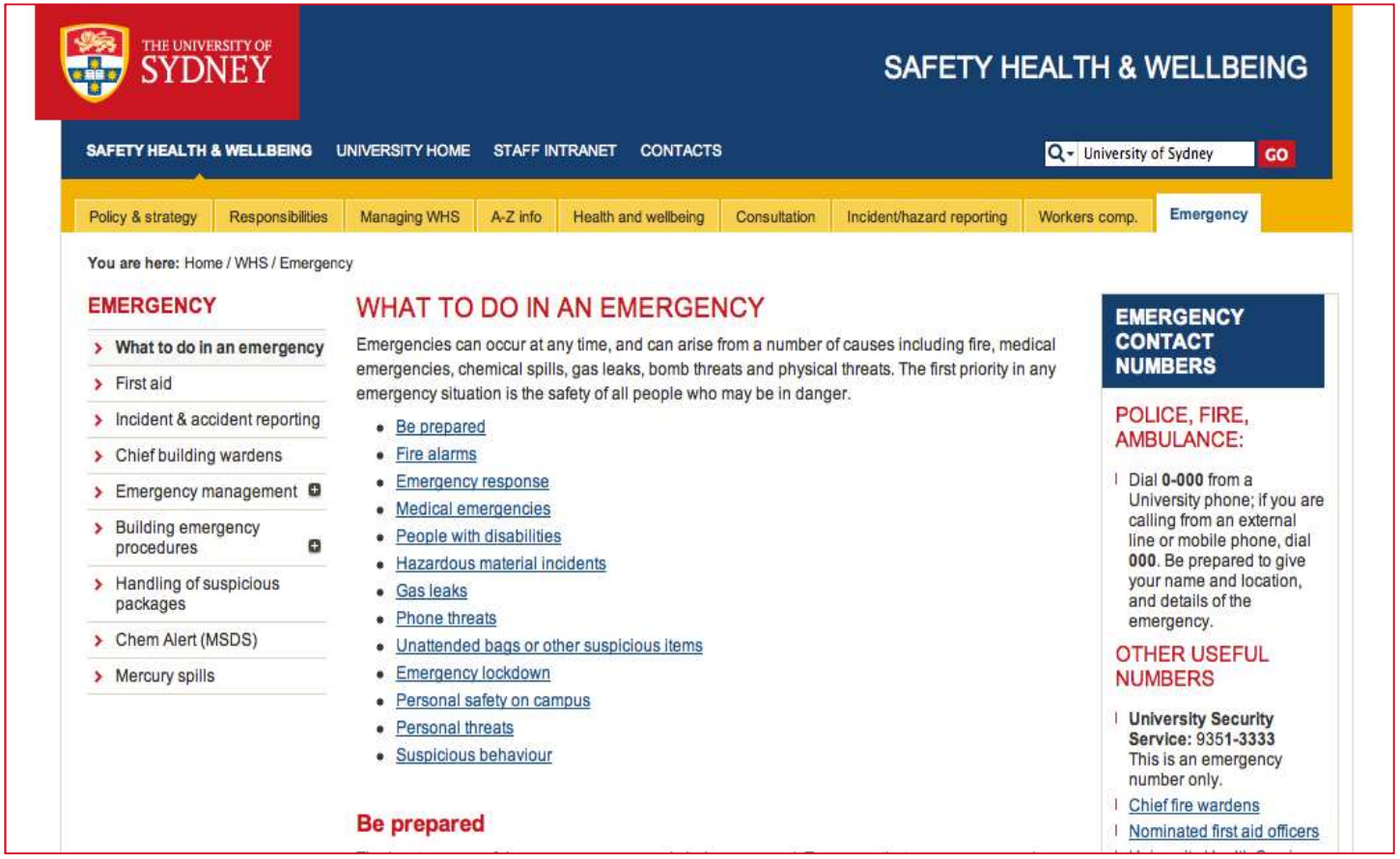
- Keep work area clean and orderly
- Remove trip hazards around desk area
- No food and drink near machines
- No smoking permitted within University buildings
- Do not unplug or move equipment without permission



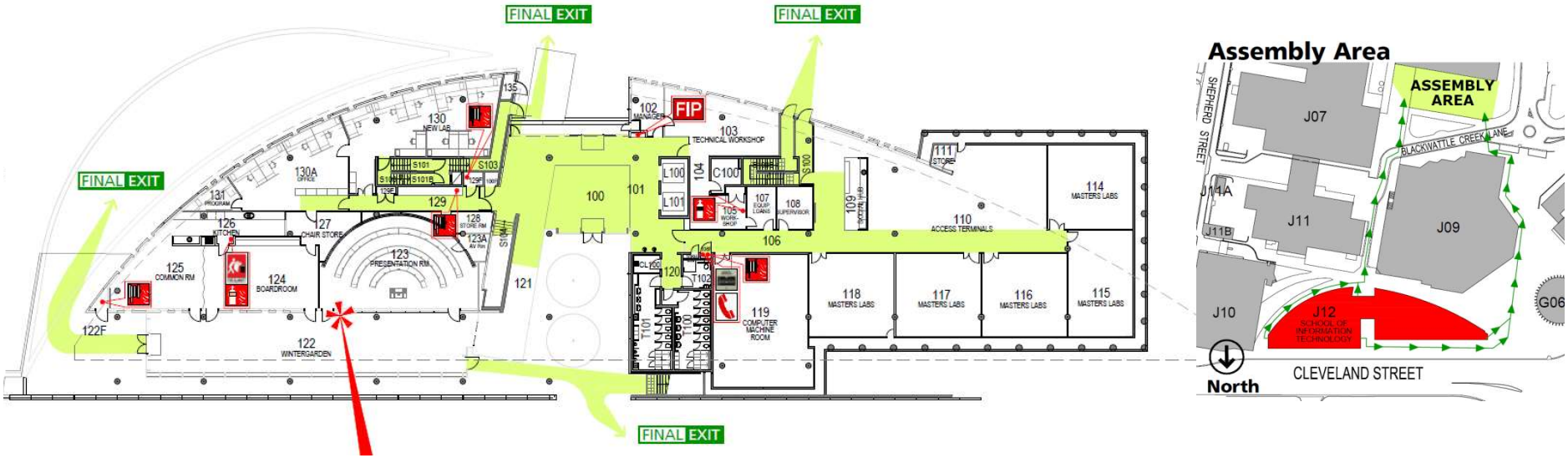


EMERGENCIES – Be prepared

www.sydney.edu.au/whs/emergency



WHERE IS YOUR CLOSEST SAFE EXIT ?



EMERGENCIES

Evacuation Procedures

ALARMS

BEEP... BEEP... Prepare to evacuate

1. Check for any signs of immediate danger.
2. Shut Down equipment / processes.
3. Collect any nearby personal items.

WHOOP... WHOOP... Evacuate the building

1. Follow the EXIT exit signs.
2. Escort visitors & those who require assistance.
3. DO NOT use lifts.
4. Proceed to the assembly area.

EMERGENCY RESPONSE

1. Warn anyone in immediate danger.
2. Fight the fire or contain the emergency, if safe & trained to do so.

If necessary...

3. Close the door, if safe to do so.
4. Activate the "Break Glass" Alarm  or 
5. Evacuate via your closest safe exit. EXIT 
6. Report the emergency to 0-000 & 9351-3333

MEDICAL EMERGENCY

› If a person is seriously ill/injured:

1. **call an ambulance 0-000**
2. **notify the closest Nominated First Aid Officer**

   If unconscious– send for Automated External Defibrillator (AED)
   AED **locations**. (http://sydney.edu.au/whs/docs/news/AED_locations.pdf)
   NEAREST to SIT Building (J12)
   - Electrical Engineering Building, L2 (ground) near lifts
   - Seymour Centre, left of box office
   - Carried by all Security Patrol vehicles

3. **call Security -** 9351-3333
4. **Facilitate the arrival of Ambulance Staff (via Security)**

Nearest Medical Facility
University Health Service in Level 3, Wentworth Building

First Aid kit – SIT Building (J12)
kitchen area adjacent to Lab 110

School of IT Safety Contacts

CHIEF WARDEN
Name: Greg Ryan

FIRST AID OFFICERS

Name: Will Calleja
Location: 1 West
Phone: 9036 9706

Name: Katie Yang
Location: 2E-227
Phone: 9351 4918

Orally REPORT all INCIDENTS & HAZARDS to your SUPERVISOR

OR

Undergraduates: to Katie Yang 9351 4918

Coursework Postgraduates: to Cecille Faraizi 9351 6060

SIT School Manager: Shari Lee 9351 4158

COMP9120
Lecture 1: Introduction

Welcome!

› Just one second...

How many databases did we just discuss?

THE UNIVERSITY OF SYDNEY

Databases are Everywhere!

- Database Application Examples:
  - Banking systems:
    - accounts & loans, customers, all transactions (banks, ATMs, internet)
  - Airlines reservation systems:
    - reservations by customers, flight schedules, freq.flyer info
  - Corporate records
    - Sales: customers, products, purchases - and reports on this
    - Human resources: employee records, salaries, tax deductions
    - Manufacturing: production, inventory, orders, supply chain
    - Universities: student enrollments, course offerings, timetabling, grades
  - Telecommunication: calls, bills, calling/SIM cards
  - Health care: patients, prescriptions, drugs, …

Databases touch all aspects of our lives

Databases on the Internet

- Ebay
  - More than 100 back-end databases (in 2005)
    - 12,000 “servers” (2007 - http://www.computerworlduk.com/in-depth/infrastructure/485/ebay-reveals-the-secrets-of-grid-computing/)
  - ca. 5 billion SQL/day (in 2005)
  - Processes >50 PB/day (2012 http://knowwpcarey.com/article.cfm?cid=25&aid=1171)
- Wikipedia: (as of Oct. 2006 - http://stats.wikimedia.org/EN/ChartsWikipediaEN.htm)
  - ca. 350 servers
  - 249 languages, millions of articles
    (engl.: 1.5M with 5GB data, 4.1 million updates/month)
    - 12GB English as of 2012
  - Behind each language at least one database cluster (2+ dbms)

    cf. *Brion Vibber “Scaling and Managing LAMP at Wikimedia”, Santa Clara, 2008*.
- 2008: Yahoo! claimed record with 2 Petabyte Database

Proof of Concept: SDSS SkyServer

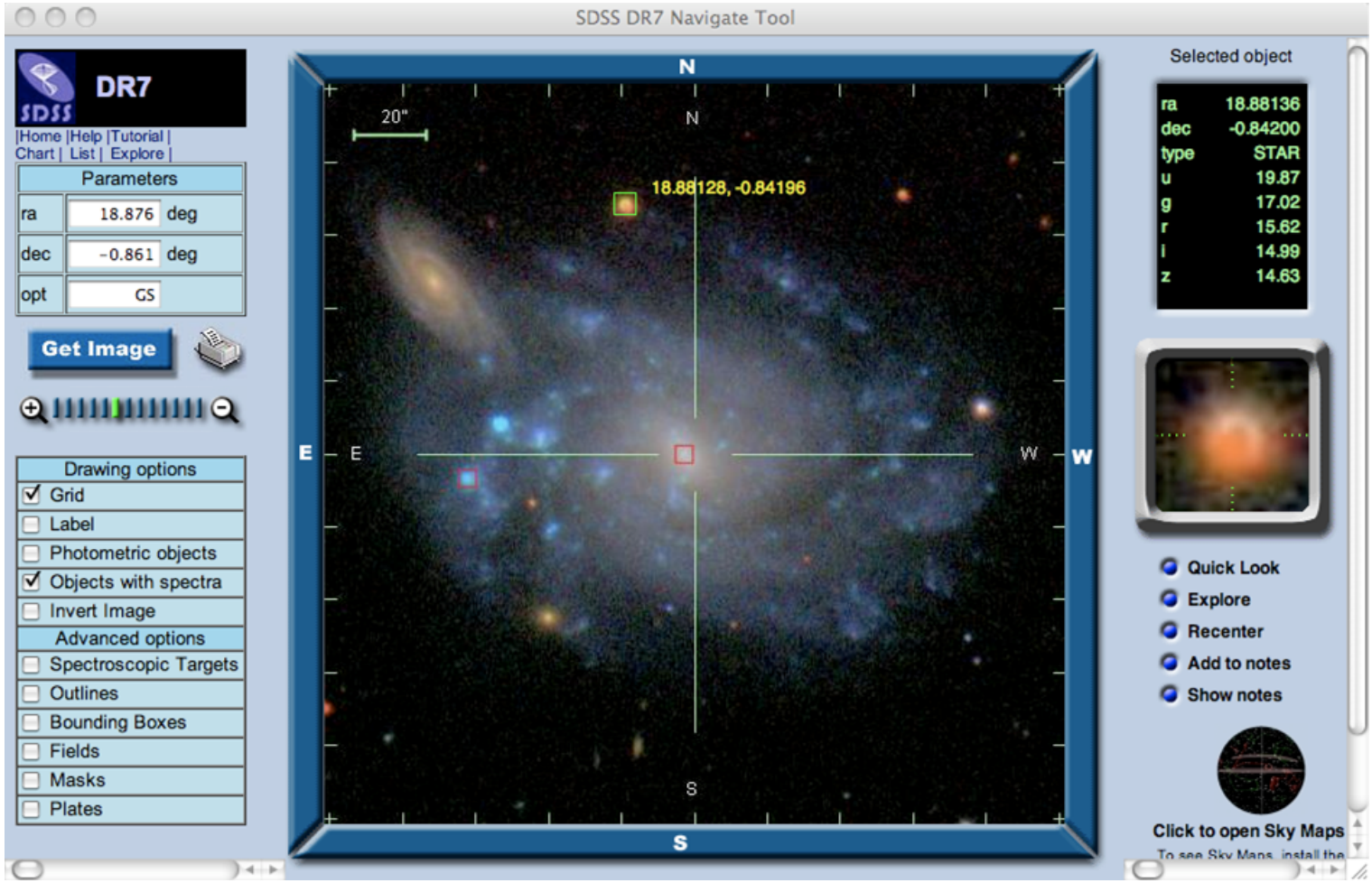
- Website to access data from the Sloan Digital Sky Survey (SDSS)
  - astronomy survey aimed at creating a map of a large part of the universe
  - used a 2.5-meter telescope in New Mexico to automatically take images of about 1/3 of the sky.
  - https://www.youtube.com/watch?v=08LBItePDZw

Jim Gray standing next to the main SDSS telescope at Apache Point Observatory, New Mexico.

- SDSS Data Release DR7:
  - 350 million celestial objects
  - 1.51 million spectra of stars, galaxies and quasars
  - over 70TB of raw and processed data

Example: SkyServer

A ‘Virtual Telescope’ over the SDSS data, backed by database systems, allowing to share and analyse the data using SQL.



[Screenshot of SkyServer website for SDSS DR7; Source: www.sdss.org]

THE UNIVERSITY OF SYDNEY

What is a Database?

- Collection of data central to some enterprise / organisation
- Essential to operation of enterprise
  - Contains the only record of enterprise activity
- An asset in its own right
  - Historical data can guide enterprise strategy
  - Of interest to other enterprises
- State of database mirrors state of enterprise
  - Database is persistent
  - Shared:
    all qualified users have access to the same data for use in a variety of activities.

File Storage & Custom Programs

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

- How do we answer questions using database?
  - eg: Count of students in a course?
  - eg: Count of female students in a course?
- 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?

THE UNIVERSITY OF SYDNEY

What is a DBMS?

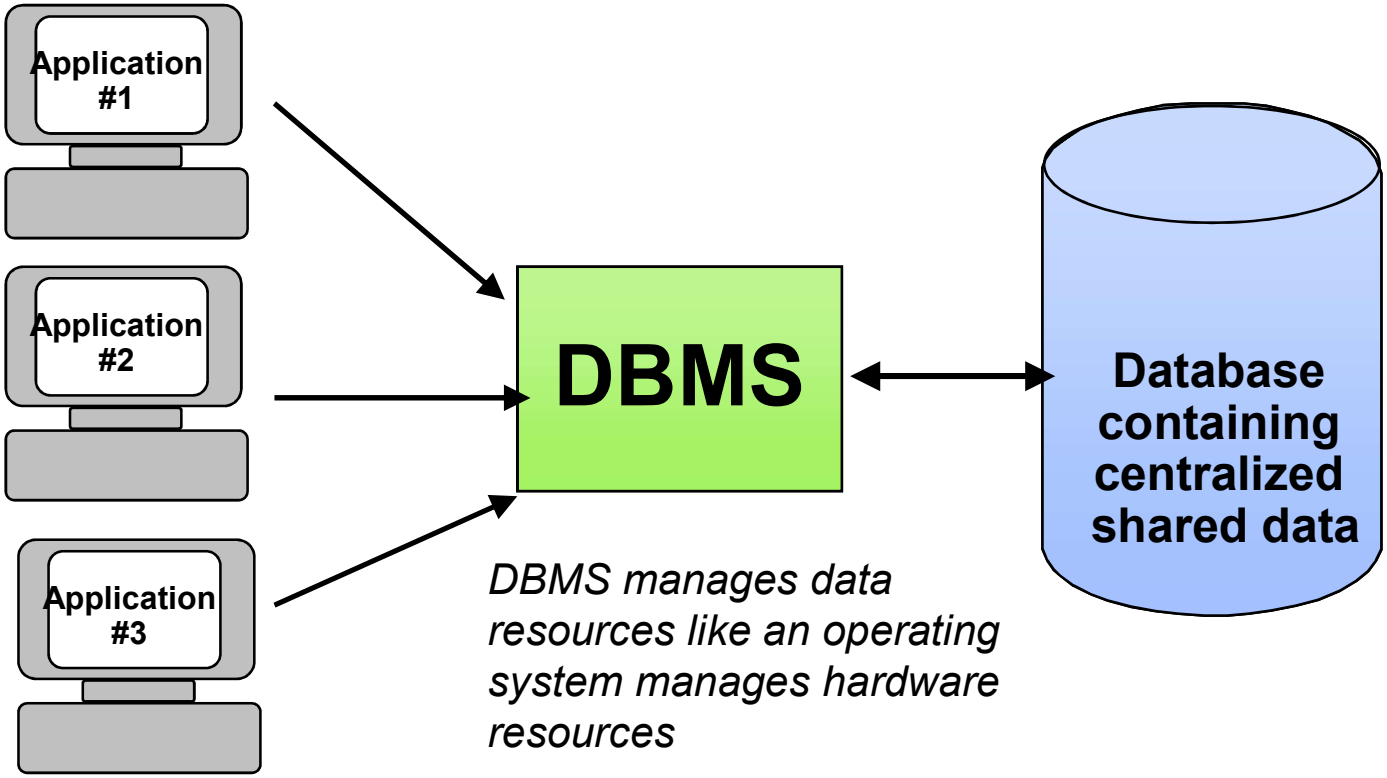
› A **Database Management System (DBMS)** is a program that manages a database:

- Stores the database on some mass (persistent) storage providing fail safety (backup / recovery)
- Provides transaction management to some guarantees for correct concurrent access to shared data
- 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.

[Source: Disney, FL]

The DBMS Approach

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



THE UNIVERSITY OF SYDNEY

Advantage of DBMS

- **Program-Data Independence**
  - Metadata stored in DBMS, so applications don't need to worry about data formats
  - Data queries/updates managed by DBMS so programs don't need to process data access routines
  - Results in:
    - Reduced application development time
    - Increased maintenance productivity
    - Efficient access

- **Minimal Data Redundancy**
  Leads to increased data integrity/consistency

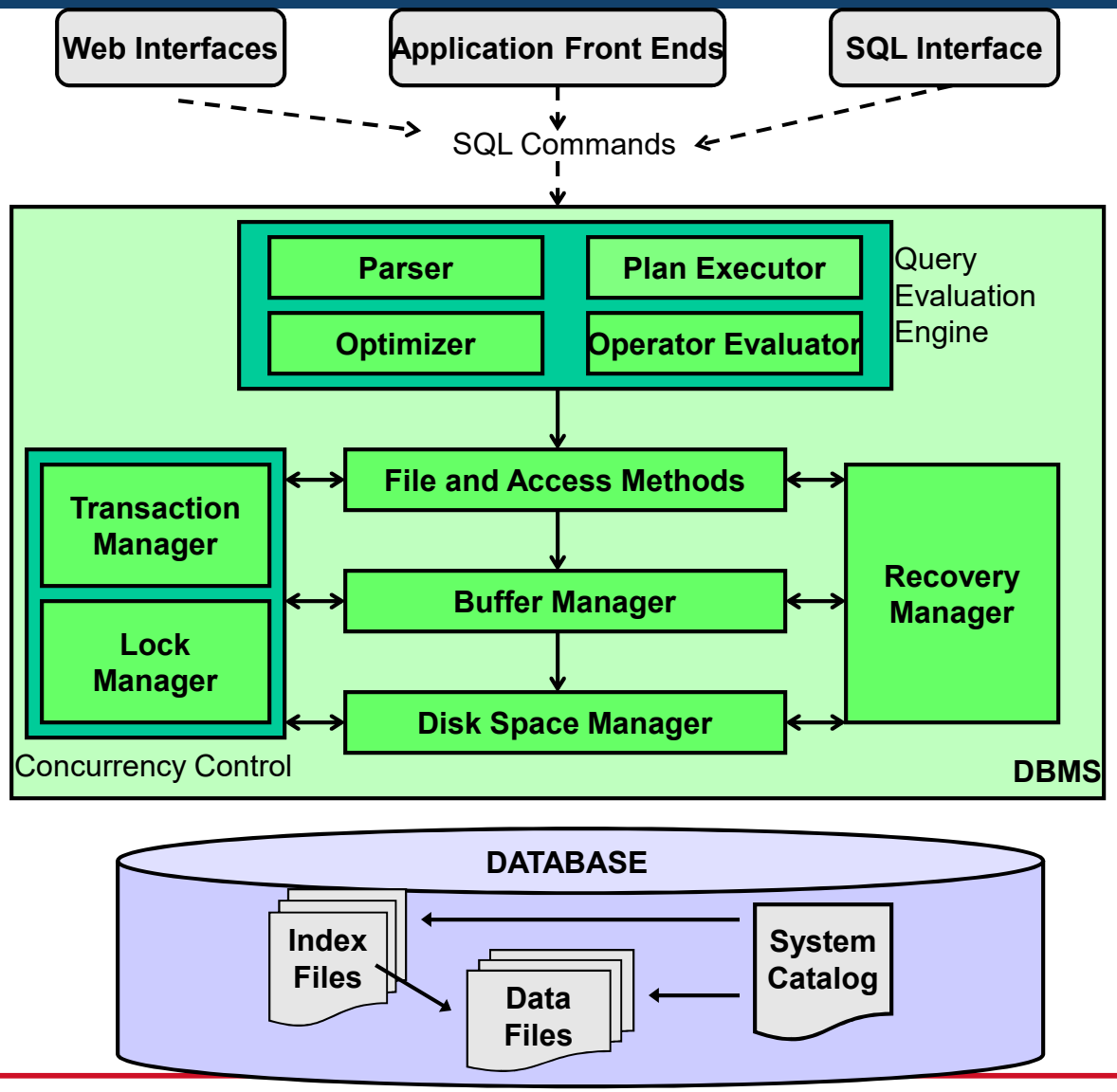
THE UNIVERSITY OF SYDNEY

Advantage of DBMS (continued)

- 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



Short History of DBMS

› **Early Database Applications:**

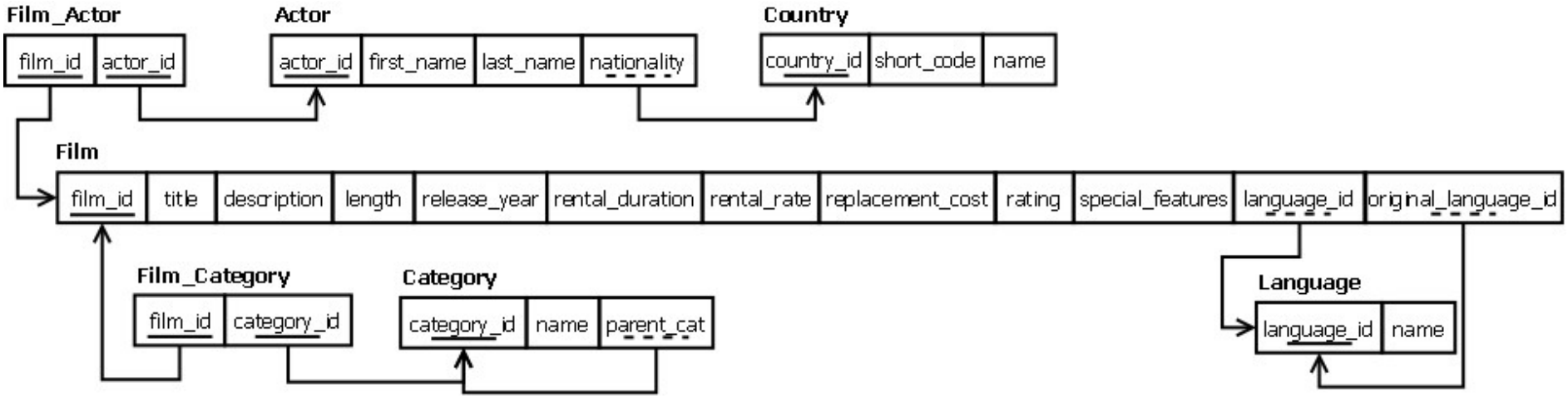
- Early 1960s: Charles Bachman: First General Purpose DBMS (IDS) – 1973 Turing Award!
- Late 1960s: IBM develops IMS – forms basis of hierarchical data model
- 1970: Edgar Codd proposes ***the relational data model*** : widespread adoption - 1981 Turing Award!
- Late 1980s: SQL Query language developed as part of IBMs System R project standardised.
- 1999: James (Jim) Gray wins Turing award for his contributions to database transactions
- Relational databases are now widespread with many DB vendors.
- May used with the dynamic web – where DB queries are generated in response to DB clicks, and query results are returned to the user in formatted HTML code.
- To express many different sorts of data that need to be exchanged between cooperating businesses, the recent standard is **XML (eXtended Markup Language)**. Main features: data is *semi-structured* and *self-describing.*

Relational Databases

- Stores data as rows with multiple attributes
- Rows of the same format ('type') form a table
- A relational database is a collection of such tables (which typically are related to each other by key attrbutes; more on that next week)
- Example:

Student				
sid	name	email	gender	address
5312666	Jones	ajon1121@cs	m	123 Main St
5366668	Smith	smith@mail	m	45 George
5309650	Jin	ojin4536@it	f	19 City Rd

Relational Schema Example



Central DBMS Services

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

Declarative Queries: “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
  - Programming is hard, and choosing between different computations is hard
- Users should be offered a way to express their requests declaratively
  - A query language can be based on logic
  - Select…where…

THE UNIVERSITY OF SYDNEY

Queries in a DBMS

- DBMS provides a specialized language for accessing data
  - **Query Language**
  - Can be further distinguished between
    - DML - Data Manipulation Language
    - DDL - Data Definition Language
    - DCL - Data Control Language
- Standard for relational DBMS: **SQL**
  - Based on formal query languages:
    Relational Algebra and Relational Calculus
- Queries are evaluated as efficiently as possible
  - Huge influence by physical design

SQL Example

- The *working-horse* command: **SELECT – FROM – WHERE**
- retrieves data (rows) from one or more tables of a relational database that fulfil a search condition
- Example 1:

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

- Example 2:

```
SELECT *
  FROM Student
```

- Example 3:

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

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 from an external perspective. This helps manage failure scenarios.
- Transactions prevent interference between two code executions that access & update the same data.
- As discussed later, transactions can achieve this via logging & locking.

THE UNIVERSITY OF SYDNEY

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

Role in Design, Implementation and Maintenance of a TP System

- 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
- DBMS Vendor's Software Engineers

THE UNIVERSITY OF SYDNEY

Summary

- DBMS used to maintain & query large datasets that are shared by many application programs/users

- Some powerful ideas:
  - Program-Data Independence
  - Controlled Data Redundancy
  - Declarative Queries
  - Transactions

- Every ‘knowledge worker’ or scientists needs database know-how, as do all IT experts (application developers, software engineers, system analysts, …) - not just DBAs

- Databases are one of the broadest and most useful areas in CS and IS

THE UNIVERSITY OF SYDNEY

References

- Ramakrishnan/Gehrke (3rd edition)
  - ***Chapter 1 Next Week's Homework Questions***
- 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/

THE UNIVERSITY OF SYDNEY

Next Week

› Conceptual Database Design using the
  - Entity Relationship Model

› Readings:
  - ***Ramakrishnan/Gehrke, Chapter 2 : Includes content on next week's Homework Questions!***
  - Kifer/Bernstein/Lewis book, Chapter 4
  - Ullman/Widom, Chapter 4