COSC210

Database Mangement Sytems

Lecture 1 - Databases and Users

Dr. Edmund Sadgrove

Reading

- Chapter 1 from ***Fundamentals of Database Systems***

Summary

- Welcome to COSC210
- House Keeping
- The Basics
- The Database Approach
- Database Users
- Advantages of the DB Approach
- When not to use a DBMS

Welcome to COSC210

- Welcome to Databases!
- In this unit you will be coving:
  - The fundamentals of designing and implementing effective and productive databases using database management system software.
  - The use of *Structured Query Language* to create, modify and query database.
  - The use of *Relational Algebra* and *Relational Calculus* to describe data queries in a language-independent manner.
  - The use of *Entity-Relationship* modelling for database design.
  - Data Normalisation.

Welcome to COSC210

- Further topics in the unit, that relate the functionality and implementation of the under-lying *Database Management System* (DBMS):
  - Query Processing and Optimisation
  - Transaction Management
  - Roll-back and Recovery
  - Security

Welcome to COSC210

- Practical Topics:
  - Applying SQL
  - Developing data-driven applications

  - Creating Data-driven web application
  - Using/developing Spatial Databases

Welcome to COSC210

- Theres lots to learn - but its all very useful for a Computer Scientist / Software Engineer / Data Scientist to know.
- The topics are only a sub-set of those presented in the text - so we won't be crawling cover to cover!

House Keeping

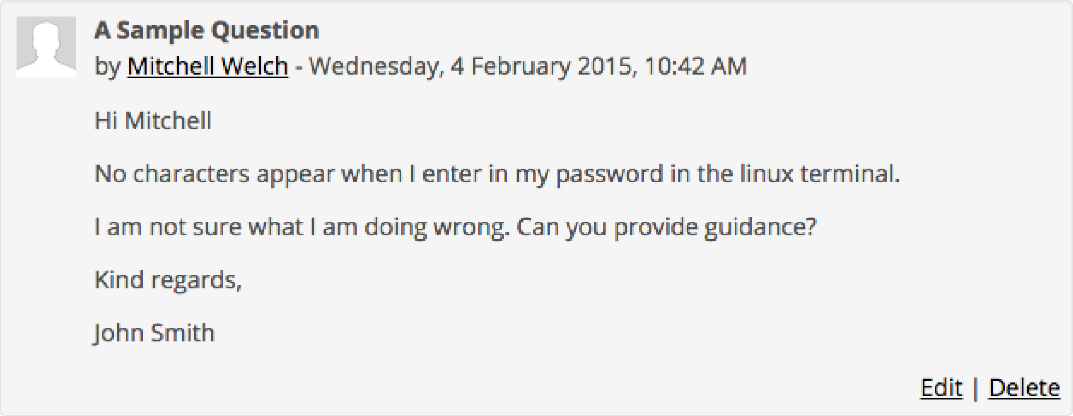
- All content will be presented through the moodle homepage for this unit.
- This doesn't necessarily mean it will actually be *hosted* on moodle however.
  - Lectures and Prac. material will be hosted on *turing.une.edu.au* (This is because moodle can't run the web-based slideshow application I am using to present the slides and notes.)
  - Some videos will be hosted via youtube (to get access to their bandwidth - sorry in advance if you get spammed with adds)

House Keeping

- The textbook for this unit is ***Fundamentals of Database Systems*** by Ramez Elmasri and Shamkant Navathe.
- United Campus Bookstore link: http://orders.ucb.net.au/ViewBook.aspx?acc=9781292097619
- Buy it from the bookstore, they have free delivery and the correct edition.
- If you are buying elsewhere, make sure you get the International Edition

House Keeping

- Questions about assignments, practicals and lectures should be communicated via the relevant discussion board
- Remember your discussion board etiquette!
- Make sure you use a descriptive subject line for your post.
- An Example post:



A Sample Question
by Mitchell Welch - Wednesday, 4 February 2015, 10:42 AM

Hi Mitchell

No characters appear when I enter in my password in the linux terminal.

I am not sure what I am doing wrong. Can you provide guidance?

Kind regards,

John Smith

House Keeping

- Questions about your personal issue/progression/studies in COSC210 should be communicated by email
- The subject line of you email should be in for form: *Student Number, First Name Last Name - Subject*
- For example: **222211113333, John Smith - Extension on Assignment 2**

- Sample email:

  Hi Mitchell,

  Would it be possible for you to grant me an extension on assignment 2 till next Friday?

  Kind regards,

  John

House Keeping

- COSC210 has 2 Assignments for you to complete:
  - 1 Theory Assignment - This will be diagrams and interpretation questions (worth 20%).
  - 2 Practical Assignments - This will cover PSQL - DDL and DML (worth 10% each)..
- There are also 10 Quizzess to complete (worth 1% each)
- The Exam is worth 50% of the final mark
- Please review the moodle site and study guide for the due dates and submission methods.

House Keeping

- Two lectures per week
- One Practical session
- Please check the timetable page for timings and locations on these.

The Basics

- **Database**
  - A collection of related *data*.
- **Data**
  - Not information
  - known facts that have *real world* meaning.
- **UOD**
  - *Miniworld* or *Universe of Discourse* (UOD)
  - *UOD* determines the meaning of the data.

The Basics

- Databases can be of any size and have any level of complexity:
  - The Australian Taxation Office (*ATO*).
  - *Amazon.com* and *Ebay.com*
  - **UNE**'s student database.
- Large and Complex databases need to be efficiently managed.
- Databases are constructed for a particular purpose.

The Basics

- *Database Management System* (**DBMS**) is responsible for organising and retrieving data items.
- A DBMS is a *collection* of one or more programs.

- Users and Developers use a DBMS to:
    - **Define** data items.
    - **Construct** the database.
    - **Manipulate** and **retrieve** data.

The Basics

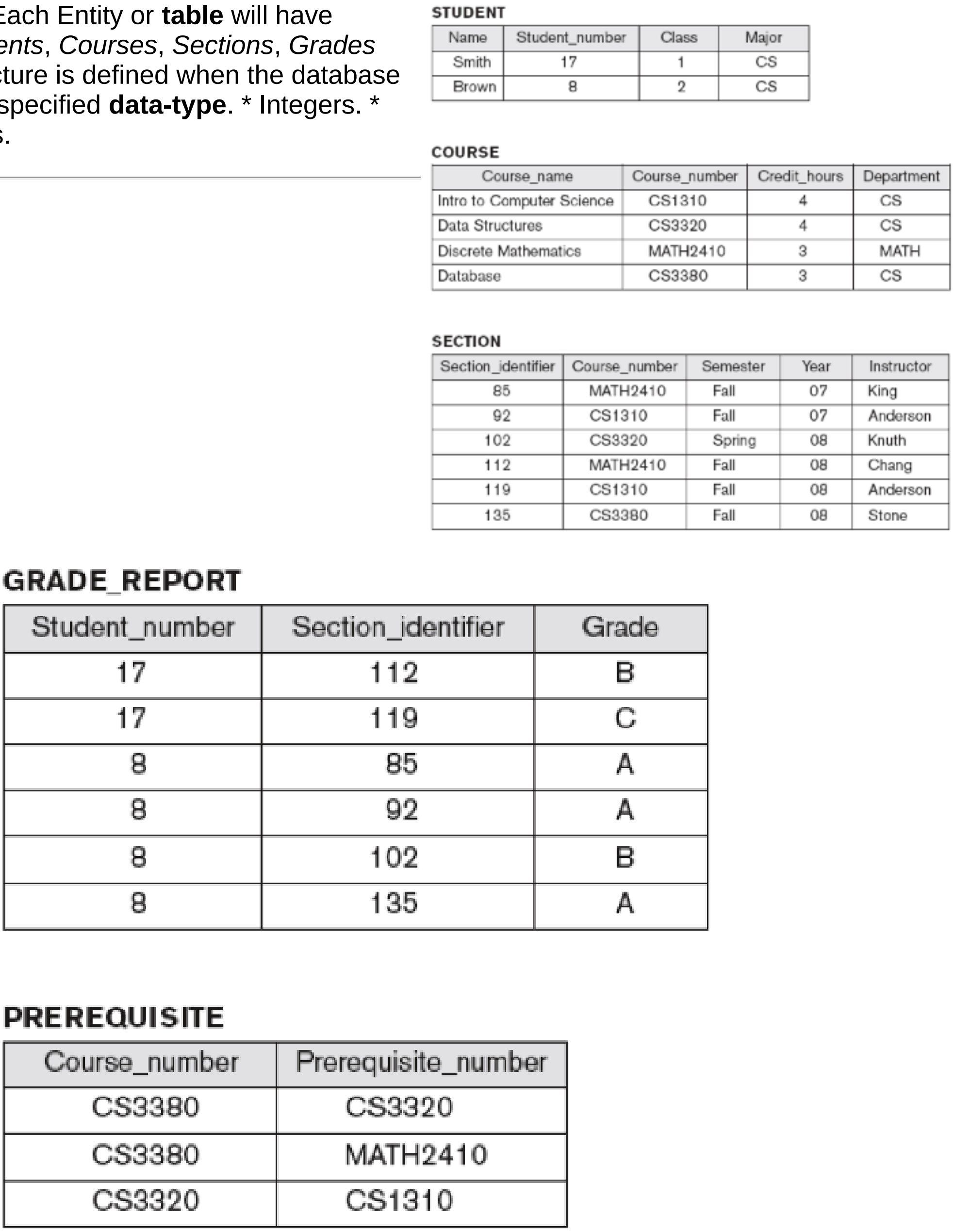
- DBMS allow *Application Programs* to access data through **structured requests** called ***queries***.
- The DBMS is responsible for protecting the data from:
    - **Security** threats (**Restricted** Access, auditing)
    - **System** failure (**Recovery**)

* Invalid Data (Constraints)

The Basics

* University database example * Each Entity or **table** will have **records** of the same type. * *Students*, *Courses*, *Sections*, *Grades* and *Pre-requisites* * **Record** structure is defined when the database is created. * Columns will have a specified **data-type**. * Integers. * Floating point. * Character Strings.

The Basics



STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

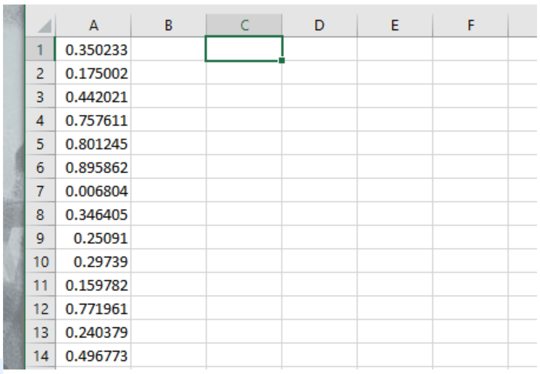
Figure 1.2
A database that stores student and course information.

The Basics

- A DBMS will access data in a structured way to **retrieve** and **manipulate**:
- Examples of Retrieval:
    - 'List all courses and grades for the student with last name smith'
    - 'List the names of all students who have taken Databases in 2009'
- Examples of updates:
    - 'Change the course name of Databases to Database Management Systems'
    - 'Change the Credit hours on Data Structures to 5'

The Traditonal Approach

- Traditional file processing architectures.
- For example, the dreaded MS excel **spreadsheet** file.
    - Data sharing is limited.
    - Users or departments will have their own data (file).

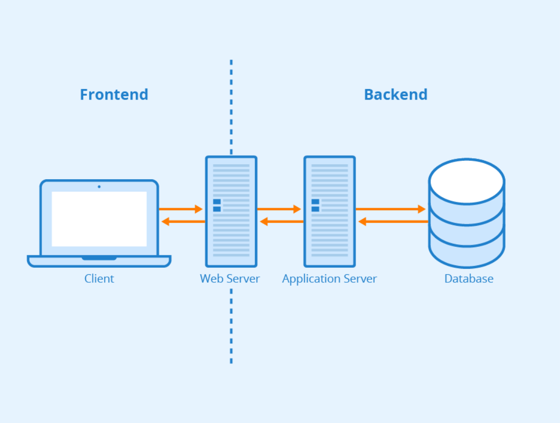


	A	B	C	D	E	F
1	0.350233					
2	0.175002					
3	0.442021					
4	0.757611					
5	0.801245					
6	0.895862					
7	0.006804					
8	0.346405					
9	0.25091					
10	0.29739					
11	0.159782					
12	0.771961					
13	0.240379					
14	0.496773					

* When scaled up, this system is inherently inefficient.

The Database Approach

* In the ***Database Approach***. * Single storage point. * Defined once and used repeatedly. * External access. * Key *characteristics* of this approach include: * **Self-descriptive** nature. * Insulation between programs and the underlaying implementation - **abstraction** * Support for different ***Views*** * Sharing of data with multiuser ***transaction management***.



The Database Approach

- A DBMS not only stores data, but also a complete description of:
    - The **structure** of all tables/files.
    - Any **constraints** on the data held within.
    - Users/Applications **access privileges**.
- This ***Meta-Data*** is stored in the **DBMS catalogue**
- In the file processing approach, data structures are defined within the applications.

The Database Approach

- A DBMS **insulates** applications/programs from the underlaying implementation.
- This is referred to as **data abstraction**.
- Changes made to the underlaying **platform**, do not affect the **client applications**.
- The DBMS provides Client applications with a **conceptual representation**.

- Keeping the implementation details hidden.
- Allowing further data abstraction.

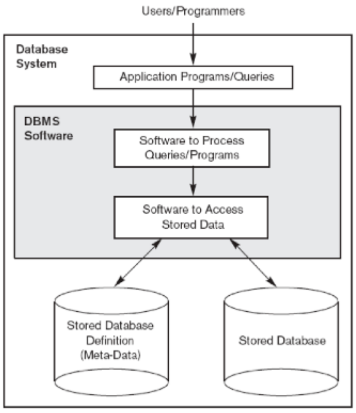

Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Stored Data
Stored Database Definition (Meta-Data)
Stored Database

Figure 1.1
A simplified database system environment.

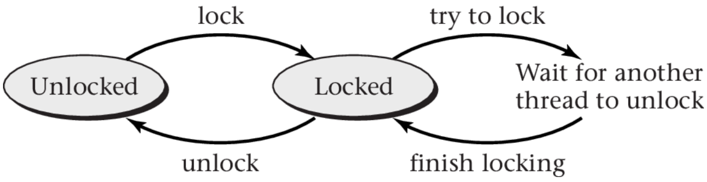
The Database Approach

- **Conceptual representations** mean the definition of different **Views**.
- **Client applications** may view data in different ways
- DBMS allows for the definition of different views:
  - Views may be a **subset** of data.
    - Historical.
    - User Privileges.
  - Views may contain **Virtual or Aggregate Data**.
    - Averages.
    - Total.
    - Minimum.

* Maximum.

The Database Approach

- A DBMS allows **multiple client applications** to access a single database.
  - Multiple applications, departments and even organisations.
- Multi-user support requires **concurrency controls**.
- A DBMS achieves this by using a **transaction management** process.



* Individual transactions should be: * Isolated (parallel processing)

* Atomic (stops partial operations)

Database Users

- **Database Designers** obtain data requirements:

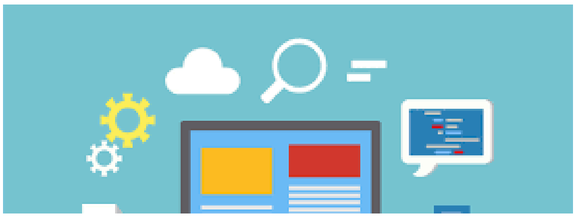
  - Data to be stored.
  - Appropriate structures for storage.
  - They also design.
    - Views, Constraints and Database functions.

- **Database Administrators** are responsible for maintaining a database.

  - **Authorising access** to users and applications.
  - **Acquiring** hardware and resources.
  - **Monitoring** usage and performance.

Database Users

- **End-users** are responsible for *using* the database - **querying**, **updating** and **generating reports**.
  - **Casual** end users (variable tasks).
  - **Naive** users such as bank tellers.
  - **Sophisticated** users such as scientists, engineers and analysts.
  - **Stand-alone** users such the ATOs E-Tax software, MS Access and OpenOffice Base.



Database Users

- Behind the scenes:
    - **DBMS System Designers and Engineers** construct the actual DBMS software and add-on modules.
    - **Tool Developer** build design tools for database development.
    - **Data Centre and Maintenance Personnel** make sure the DB and network hardware are capable of providing the required functionality.

Advantages of the DB Approach

- There are a series of advantages of the DB approach over file-processing architecture.

* **Controlling data Redundancy** through normalisation (more on this later).

```
* Improves efficiency.
* Prevents inconsistency.
```

* **Restricting Access** * *fine-grain access protocols* for data: * Retrieval, manipulation, creation and removal.

Advantages of the DB Approach

- **Persistent Storage of Objects**
    - **Object-Oriented DBMS**s can allow for *persistent* storage of class objects.
        - Objects can also be shared between applications.
- **Efficient Processing**
    - DBMSs typically have:
        - Highly efficient data structures and powerful search algorithms for query processing.
    - DBMSs can utilise:
        - Indexing (for searching), query optimisations and Caching (regular return values).

Advantages of the DB Approach

- **Backup and Recovery**
    - Most DBMSs provide **automated** back-up and recovery procedures.
        - Even if the **failure occurs during a complex transaction**.

* **Multiple User Interfaces and Views** * Mulitple views for different tasks. * This can include **specialised GUIs**: * Advanced data analysis and/or casual data retreival.

Advantages of the DB Approach

- **Representation of Complex Relationships**
    - A relational DBMS can provide the facilities to represent complex relationships between relations (tables) within the database.
- **Enforcement of Integrity Constraints**
    - An **Integrity Constraint** is a condition that must always be true.
        - A **data-type** for a particular attribute (column) of data.

        - Uniqueness constraint per table record - key (***Entity Integrity Constraint***).
        - The relationship between tables - keys (***Referential Integrity Constraint***).
    - Contraints not defined by the DBMS may be referred to as ***Business Rules***.

Advantages of the DB Approach

- **Inferencing and Triggering Actions**
    - DBMSs can allow users to define ***deduction rules*** and infer new information.
    - A simple example is the calculation of an individuals age.
        - This can be calculated from their DOB.
    - DBMSs allow for operations to be ***triggered* by changes** to data items.
    - The operations triggered can be combinations of standard functions or ***stored procedures***
    - These triggers and stored procedures become part of the database definition.
        - For Example: PL/pgSQL.

Advantages of the DB Approach

* Operations and development benefits:

- Enforcement of **standards**.
- **Reduced development** times.
- **Flexibility**.
- **Availability** of information.
- Scale-up, Scale-down and replication/distribution.

When not to use a DBMS

- Initial investment and training costs.
- Admin **overhead**.
- Database doesn't change and grow.
- DBMSs can have a large technology footprint (e.g. memory space required, **storage space**, processing overhead).
- Real-time data **processing requirements** and storage restrictions - e.g. imbedded sensing systems
- **Single user** access to data.

A quick Tour of the Technology

- PostgreSQL - A relational DBMS
- PostGIS - Spatial Database functionality for Postgres
- MongoDB - A NoSQL database
- JDBC/JDO - Connect Java programs to a database
- PHP - Web Programming with database-driven webpages

Summary

- Welcome to COSC210
- House Keeping
- The Basics
- The Database Approach

- Database Users
- Advantages of the DB Approach
- When not to use a DBMS

Questions?

ASOK, ACCORDING TO MY SPREADSHEET, YOU HAVE BEEN DOING A TERRIBLE JOB.
www.dilbert.com scottadams@aol.com
PERHAPS YOUR SPREAD-SHEET IS POORLY CONCEIVED AND DOES NOT CAPTURE THE COMPLEXITY OF THE REAL WORLD.
8-9-07 ©2007 Scott Adams, Inc./Dist. by UFS, Inc.
AND LET'S NOT FORGET THE NEAR CERTAINTY THAT YOUR FORMULAE ARE POINTING TO THE WRONG CELLS.
NUMBERS DON'T LIE.

Next Lecture

- Database System Concepts and Architecture

Reading

- Chapter 1 from ***Fundamentals of Database Systems***
- Start on Chapter 2 from ***Fundamentals of Database Systems*** for next lecture.
- Start on Chapter 4 (Basic SQL) from ***Fundamentals of Database Systems*** for the start of the practicals next week.