Information Management II

Prof Yvette Graham
Director of Integrated Computer Science (Yrs 1;2)
FI ADAPT Research Centre, TCD

What is Data?

What is Data?

Data = Information

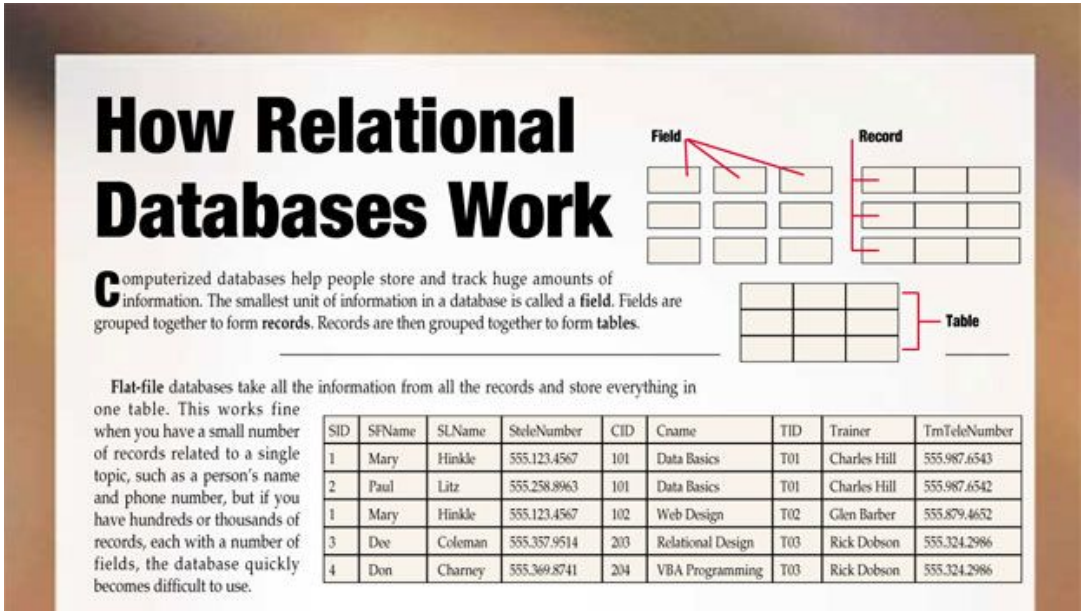
What is Data?

- Computer Scientists usually interested in data needed for a particular application
  - Flight/ticket booking system
  - Web hosting
  - Stock-keeping system
  - Online shopping
  - Internet Blog
  - Social Media: Twitter, Instagram, …
- Behind all of the above application lies at least one (possibly multiple) database(s)

Data needed for an Application

- Consider the application being developed
- What data do I need to store?
- What kind of storages should I use?
    - Relational database
    - NoSQL database
    - File storage

Relational Databases



How Relational Databases Work

Computerized databases help people store and track huge amounts of information. The smallest unit of information in a database is called a **field**. Fields are grouped together to form **records**. Records are then grouped together to form **tables**.

Flat-file databases take all the information from all the records and store everything in one table. This works fine when you have a small number of records related to a single topic, such as a person's name and phone number, but if you have hundreds or thousands of records, each with a number of fields, the database quickly becomes difficult to use.

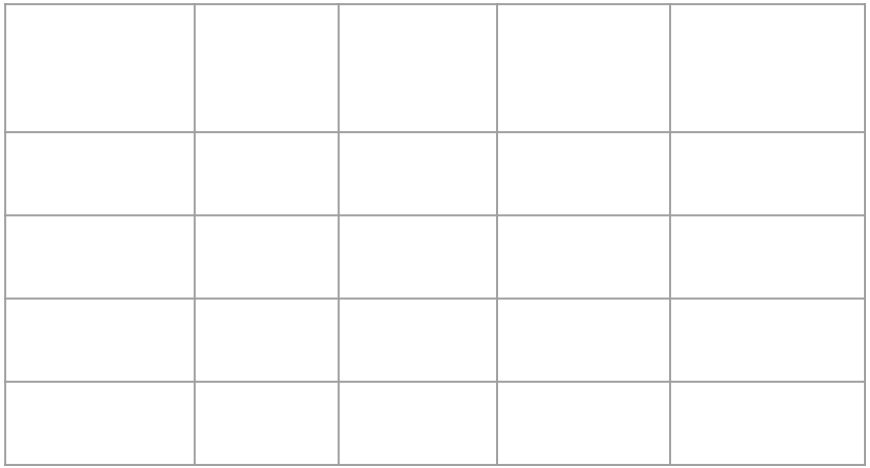
SID	SFName	SLName	SteleNumber	CID	Cname	TID	Trainer	TrnTeleNumber
1	Mary	Hinkle	555.123.4567	101	Data Basics	T01	Charles Hill	555.987.6543
2	Paul	Litz	555.258.8963	101	Data Basics	T01	Charles Hill	555.987.6542
1	Mary	Hinkle	555.123.4567	102	Web Design	T02	Glen Barber	555.879.4652
3	Dee	Coleman	555.357.9514	203	Relational Design	T03	Rick Dobson	555.324.2986
4	Don	Charney	555.369.8741	204	VBA Programming	T03	Rick Dobson	555.324.2986

- Proposed by E. F. Codd in 1970
- Information is organised in **2 dimensional tables** made of rows and columns
- Can leave cells with empty or "**null**" but generally they should be few
- Assumes for the most part that data fits a 2D structure
- Columns have headers containing name of data in that column
- Rows have a unique identifier of some kind

Relational Databases

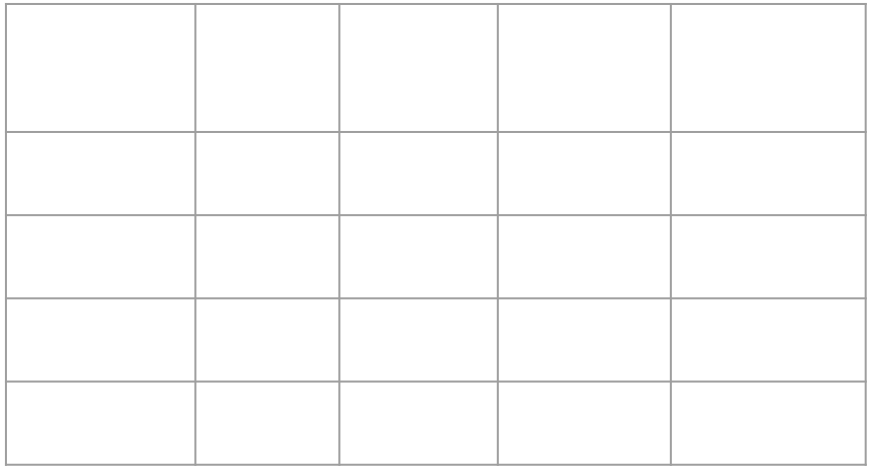
- Organise the data needed for our application into a table
- Or more likely a number of tables
- Think about how data in our application fits together in a meaningful way within a (2D) table
- E.g. we are storing student records? What would be the potential tables?

Relational Databases



- Organise the data needed for our application into a table
- Or more likely a number of tables
- Think about how data in our application fits together in a meaningful way within a (2D) table
- E.g. we are storing student records? What would be the potential tables?
  - students,
  - modules,
  - enrolments

Relational Databases



- Organise the data needed for our application into say 10 tables
- Think about how data in our application fits together in a meaningful way within a 2D table
- E.g. we are storing information about **people**?

Relational Databases

Persons table

name	age	phone		

- Organise the data needed for our application into say 10 tables
- Think about how data in our application fits together in a meaningful way within a 2D table
- E.g. we are storing information about people?

Relational Databases

Persons table

name	age	phone	cat's name	
Jack lynch	23	…	fluffy	

- Organise the data needed for our application into say 10 tables
- Think about how data in our application fits together in a meaningful way within a 2D table
- E.g. we are storing information about people?

Relational Databases

Persons table

name	age	phone	cat's name	
Jack lynch	23	…	fluffy	

- Organise the data needed for our application into say 10 tables
- Think about how data in our application fits together in a meaningful way within a 2D table
- E.g. we are storing information about people?

Relational Databases

Persons table

name	age	phone	person_id	dob
Jack lynch	23	…	1	1/1/00

Cats table

owner	name	color	dob
1	fluffy	brown	1/1/20

Relational Databases

Persons table

name	age	phone	person_id	dob
Jack lynch	23	…	1	1/1/00

Cats table

owner	name	color	dob
1	fluffy	brown	1/1/20
1	max	black	3/3/21

Relational Databases

Persons table

name	age	phone	Favorite color	

- Organise the data needed for our application into say 10 tables
- Think about how data in our application fits together in a meaningful way within a 2D table
- E.g. we are storing student records – potential tables – student, modules, enrolment tables

Relational Databases Achieve a lot

- Achieve 3 things here:
  - **Specify** the information needed about students (design the database)
  - **Store** information about students
  - **Model one or more relationships** between students and modules (who is enrolled in what module)

Students table

Student id	First name	surname	Date of birth	address

Relational Databases Achieve a lot

- Achieve 3 things here:
  - **Specify** the information needed about students (design the database)
  - **Store** information about students
  - **Model** the relationship between students and modules (who is enrolled in what module)

Enrolments table

Student id	Module id

Relational Databases

- Data needed for a particular application is organised into 2 dimensional tables
- Relations between data in each table is connected via specifying
  - *Which* **columns** in table A **relate** to another **column** in table B

Relational Databases

- Data needed for a particular application is organised into 2 dimensional tables
- Relations between data in each table is connected via specifying
  - *Which* **columns** in table A **relate** to another **column** in table B
  - *How* the column relates to it?
    - Pets example: the people were allowed have multiple pets but the pets were not allowed to have multiple owners
    - Students example: each student represents a unique individual real world person enrolled in TCD
    - Students example: modules represent individual modules that are available for students to attend
    - Students example: students are permitted to enrol in multiple modules

Relational Databases

- Data needed for a particular application is organised into 2 dimensional tables
- Relations between data in each table is connected via specifying
  - *Which* columns in table A relate to another column in table B
  - *How* the column relates to it?
    - Pets example: the people were allowed have multiple pets but the pets were not allowed to have multiple owners (one to many)
    - Students example: each student represents a unique individual real world person enrolled in TCD
    - Students example: modules represent individual modules that are available for students to attend
    - Students example: students are permitted to enrol in multiple modules (many to many)

What is a Database?

- An organised collection of Information, or Data…
  - “A database is a persistent collection of related data supporting several different applications within an organisation”
- Organised to:
  - model aspects of reality
  - in a way that supports processes that require this information
    - A collection of medical records in a Hospital
    - Finding records by a specific Doctor or Patient
      - mostly, to make the data more useful!

What is Metadata?

- Metadata adds Context to Data

Metadata	***Data***
Student Number:	89041258
Name:	John Patrick Smith
Account Balance:	132.56

- Metadata can include:
  - data type, name of element, size, restrictions etc.
  - Can be used at any level of aggregation

Database Management Systems (DBMS)

Database Management Systems

- Database Management System (DBMS)
- Goal of a DBMS is to simplify the storage of, and access to, data
- DBMS support:
  - Definition
  - Manipulation
  - Querying

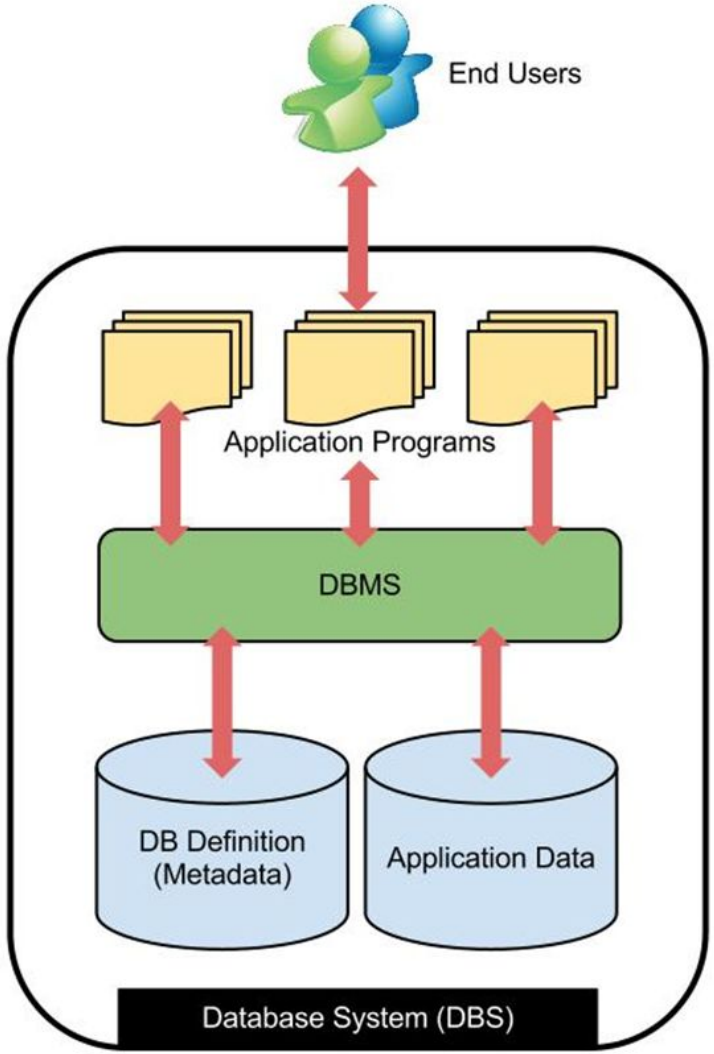
- A DBMS can manage a single, or set of, DBs

DBMSs provide

- Efficient, reliable and secure management of large amounts of persistent data.
- Language(s) for defining the DB
  - *data definition language*
  - *T*his data about data (e.g. student number is a seven digit number plus one check digit) is called *metadata*
- Languages for storing, retrieving and updating data in the DB
  - *data manipulation languages*

DBMS Examples

- MySQL, PostreSQL, SQLite …
- Oracle, IBM-DB2, SQLServer …


End Users
Application Programs
DBMS
DB Definition
(Metadata)
Application Data
Database System (DBS)

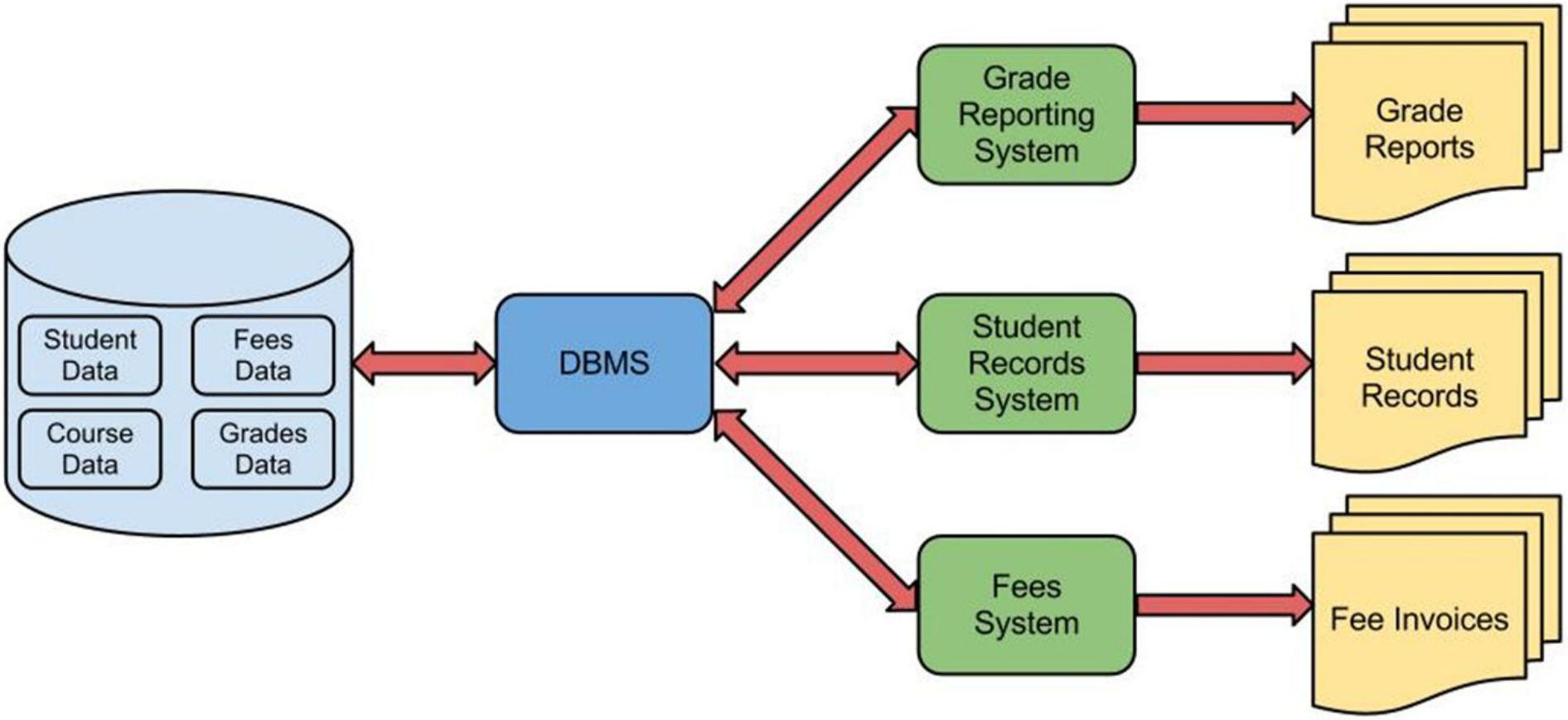
More about Databases

Why important to know about Databases?

- Ubiquity
- The majority of large corporations, web sites, scientific projects… all manage both day to day operations as well as business intelligence and data mining using databases

Databases can be helpful to manage data:

- Duplication of data
  - Wasteful of storage
  - Inefficiency
  - Most importantly, leads to inconsistencies
- DB approach aims to eliminate such *redundancy* (data duplication)
- Data from all applications is integrated and stored once in the DB
- All applications access the same physical copy of the data


Student Data
Fees Data
Course Data
Grades Data
DBMS
Grade Reporting System
Student Records System
Fees System
Grade Reports
Student Records
Fee Invoices

Data Independence

- DBMS support ***logical data independence***
  - by allowing the view of the data to be changed and data added without affecting it's underlying organisation

- DBMS support ***physical data independence***
  - as they ***insulate*** the way in which data is viewed by the applications/users from the way in which it is physically stored

Data Intregrity

- Data Integrity is concerned with the ***consistency*** and ***accuracy*** of the data in the Database
- Data Redundancy is a major threat to Data Integrity
- Support for Data Integrity is a key feature of any DBMS

Data Integrity

- Databases model parts of the real world in which many rules apply

–“A student has only one address”

–“A student must take 5 courses in the final year or 4 courses plus a project”

- DBMS express such rules by means of “integrity constraints”
- Validation of data values being entered into the DB is another aspect of Data Integrity
- Many users/applications simultaneously updating the Database can threaten Data Integrity

–This requires “concurrency control”

Query Languages

- Query languages, such as SQL, are usually very easy to learn and intuitive
- With some assumptions in place we can use the same simple database interface to interact with a wide number of interfaces
- Users do not need years of training or a CS degree to query / add / remove information from it
- The same database can also be used in a range of application programs all at the same time

Metadata management

- With the Database approach:
  - Metadata is stored centrally in the catalog
  - Database catalog entry for patient record

Patient_ID	int(4)	Unique
Patient_Name	varchar(255)	Firstname followed by Surname
Patient_Address	varchar(255)	Truncate if necessary
Patient_Phone	int(10)	Home phone
Patient_Allergies	varchar(255)	Drug name or None

Advantages and Disadvantages of Databases

Advantages of Databases

- Search and Retrieval Capabilities
  - Filtered according to specific needs
- Reduced Data Redundancy
  - Ease of Update
- Greater Data Integrity
- Independence from Applications, Concurrent Access
- Improved Data Security
- Reduced Costs for Data Entry, Storage and Retrieval

Database Disadvantages

- Some training still required for management and querying
- Database systems can be complex and time-consuming to design
- Cost
  - Software
  - Hardware
  - Training
- Loss of autonomy brought about by centralised control of the data
- Inflexibility due to complexity or bad application database match

Database Languages (eg SQL)

- Programming languages which are used to:
  - Define a database
    - its entities and the relationships between them
  - Manipulate its content
    - insert new data and update or delete existing data
  - Conduct queries
    - request information based upon defined criteria

- The Structured Query Language (SQL) is the most commonly used language for Relational Databases
  - Supported by all relational DBMS and is a standard.

SQL

SQL

- SQL is split into four sets of commands which are divided based upon the tasks they are used for:
  - Data Definition Language
  - Data Modification Language
  - Data Query Language
  - Data Control Language

SQL Data Definition

- SQL uses a collection of **imperative verbs** whose effect is to modify the schema of the database
- Can be used to **add**, **change** or **delete** definitions of tables or other objects.
- These statements can be freely mixed with other SQL statements
  - so the DDL is not truly a separate language.

SQL Data Manipulation

- The data manipulation language comprises the SQL data change statements
  - Modifies stored data
  - Does NOT modify the schema or database objects
    - This is always the responsibility of the Data Definition Language

- Used for inserting, deleting and updating data in the tables of a database

SQL Data Query

- The data query language allows users of a database to formulate requests and generate reports
- There is one primary command used in SQL to query the database - the SELECT Statement
  - This statement is used to query or retrieve data from a table in the database.
  - A query may retrieve information from specified columns or from all of the columns in the table
  - A query may have specified criteria that must be met in order for data to be returned

More about Databases and Final Words

Transactions

- A way to group actions that must happen atomically
  - all or nothing
- Guarantees to move the DB content from one consistent state to another
- Isolates these actions from parallel execution of other actions/transactions
- Ensures the DB is recoverable in case of failure
  - e.g. the power goes out

Backup and Recovery

- Ensures that the DB can be returned to a stable state in case of errors, such as:
  - Transaction failure
  - System errors
  - System crash
  - Data Corruption
  - Disk failure

Database “Users”

- DBMS implementer
  - Builds the DBMS System
- Database designer
  - Designs the Database, Establishes the Schema
- Database application developer
  - Develops programs that operate upon the DB
- Database administrator
  - has overall responsibility for the DB including specifying access constraints, selection of appropriate backup and recovery measures, monitoring performance etc.

Emergent Databases

- XML Databases
  - Document-Oriented
- NoSQL Databases
  - Web Scale, Non-Relational, Open Source
- In Memory Databases
  - Stores data in main memory rather than on disk
- Others
  - Massively parallel processing (MPP) databases
  - Online analytical processing (OLAP) databases