LECTURE 1:
INTRODUCTION

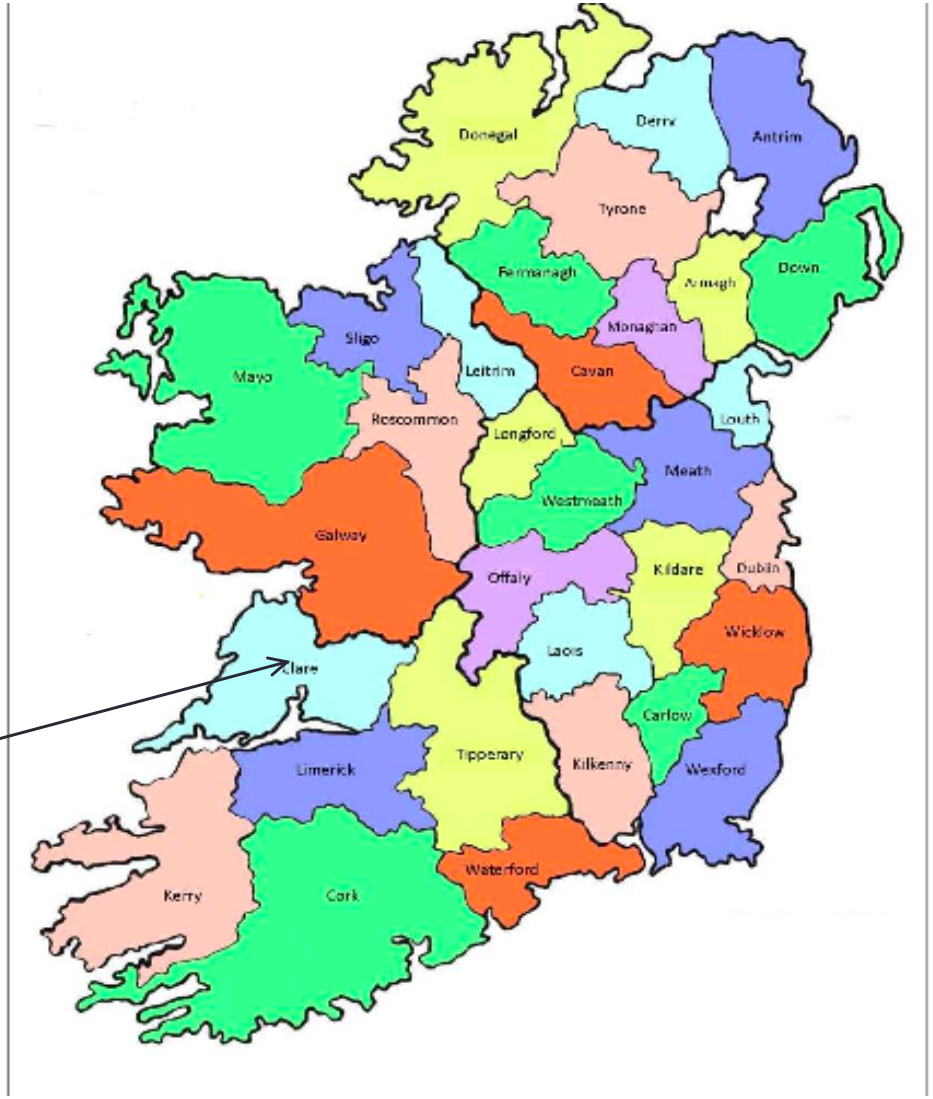
COMP2013J: Databases and Information Systems

Dr. David Lillis (david.lillis@ucd.ie)
UCD School of Computer Science
Beijing-Dublin International College

About Me

- My name is David Lillis:
  - “Lillis” is my family name.
  - “David” is my given name.
  - You can call me “David”, or “Dr. Lillis” (if you prefer to be more formal): I don’t mind.

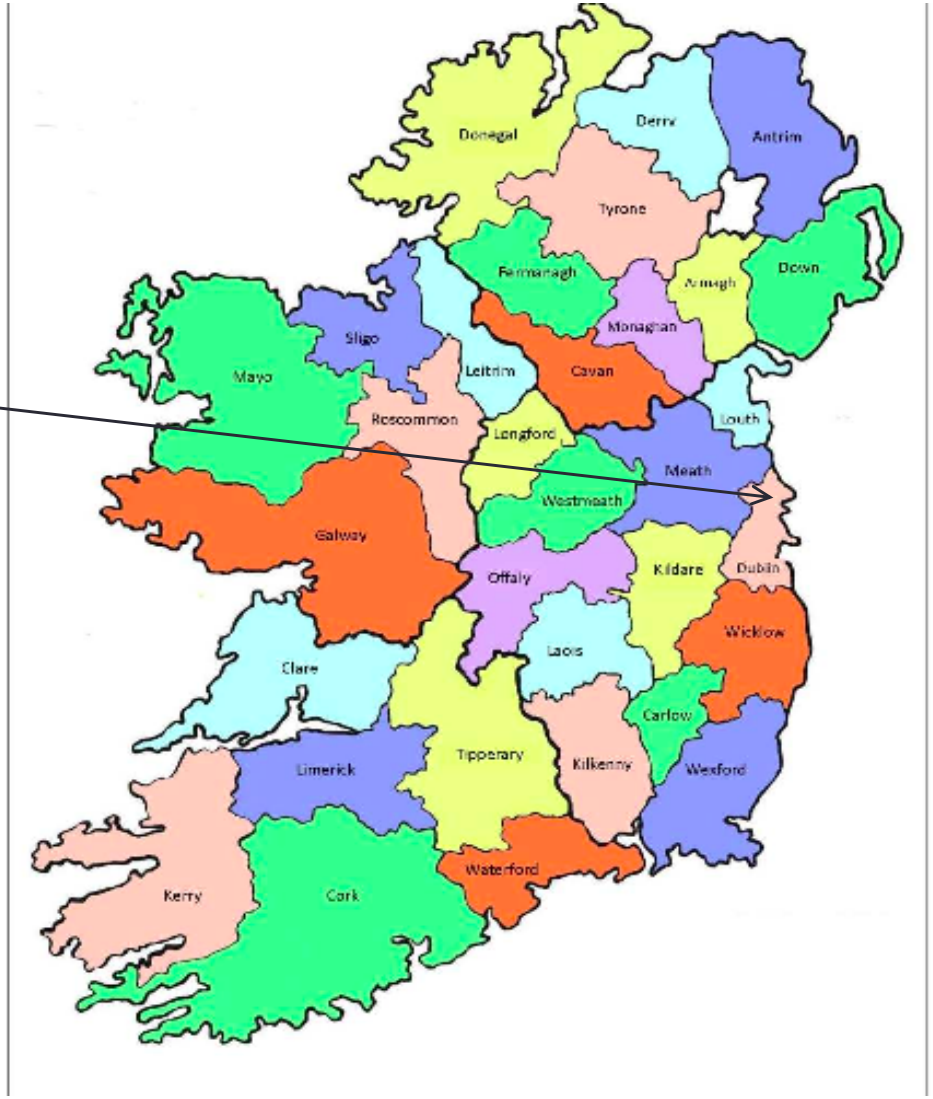
- Originally from Ennis: a town with a population of 25,000 located in the west of Ireland




ALMA
GameStop
DV8
MILANO
08-MN-2056
PARTNER
SUIT YOURSELF

About Me

- I live in Dublin: Ireland's capital city (and biggest city): population of 1,300, 000
  - (The entire country of Ireland has about 4,700, 000 people)



About Me

- Assistant Professor in School of Computer Science in UCD
- I did my PhD in UCD also, in the area of Multi Agent Systems.
  - I have also done research on Information Retrieval, Digital Forensics and Sensor Systems.
- Previously taught in:
  - Griffith College Dublin.
  - Fudan University, Shanghai.
- This is my 6th semester lecturing in BDIC, and my 9th semester in China.

Module Timetable

- Lectures:
  - Mondays 08:00-09:35 in Room 502, Teaching Building 4
- Labs:
  - Tuesdays 15:25-17:00 in Room 214, Teaching Building 4
- Labs begin in Week 3.

Module Format

- Lectures will run from **Week 1** to **Week 15** (inclusive).
- Labs will run from **Week 3** to **Week 15** (inclusive).

- If you have any questions about the module, please ask me or a teaching assistant.

- After a lecture, during labs or by email:
  - My email address is david.lillis@ucd.ie

- Assessment is split between
  - 70% Final exam at end of term (Week 17/18)
  - 30% Practical work during semester

Virtual Learning Environment (VLE)

- We will use **Moodle** as the VLE for this module.
- Located at: https://csmoodle.ucd.ie/moodle/course/view.php?id=747 (or scan the QR code)
- Enrolment key:
  - BDIC-Data-2019

Assignments

- Assignments are **individual** (not group assignments).

- This means that you must submit **your own work** only.

  If you submit somebody else's work and pretend that you wrote it, this is called **plagiarism**.

- Plagiarism is a **very serious** academic offence.

Plagiarism & UCD Computer Science

- **Plagiarism is a serious academic offence**
  - [Student Code, sections 6.2 & 6.3] or [UCD Registry Plagiarism Policy] or [CS Plagiarism policy and procedures]
- Our staff and demonstrators are **proactive** in looking for possible plagiarism in all submitted work
- Suspected plagiarism is investigated by the CS Plagiarism subcommittee
  - Usually includes an interview with student(s) involved
  - 1st offence: **usually** 0 or NG in the affected components
  - 2nd offence: may be referred to the **University disciplinary committee**
- Student who enables plagiarism is equally responsible
  - **http://www.ucd.ie/students/guide/academicregs.html**
  - **http://libguides.ucd.ie/academicintegrity**

Asking for help

- If you find things difficult, help is available.
  - There is a lecturer and many TAs in every lab.
  - You can ask a question after class.
  - You can email a TA with a question outside class.
  - You can email me with a question outside class.
  - You can get help from your classmates.
    - Getting help to understand something is not the same as copying a solution!

Module Content

- Based on sections of the book
  - Fundamentals of Database Systems (6th Edition) by Elmasri and Navathe
  - ISBN-10: 0136086209 | ISBN-13: 978-0136086208
  - Chinese version available from Amazon.cn
  - ISBN: 9787302260448, 7302260443
- All materials will be covered in lecture notes
- Practical work will using MySQL Relational DataBase Management System (RDBMS)
  - Free, cross platform, open source database management system

Topics

- Introduction to Databases
- Database Models
- Relational Database Model
- Structured Query Language
- Database Design (Entity-Relationship Model)
- Database Normalisation
- Programmatic DB Use
- NoSQL Databases

INTRODUCTION TO DATABASES AND INFORMATION SYSTEMS

Why Study Databases?

- A huge amount of information being stored.
- The College, Medical records, Employers, Companies, Government Agencies, etc.
- Managing that data is a **really big** task
- Data Base Management Systems (DBMS)
- Storing is easy, **managing** is the issue

What is a Database?

- Initial Definition: A database is a collection of related data
- For example, consider the names, telephone numbers, and addresses of the people you know
- You may have recorded this data in an indexed address book (The contacts in your phone)
- This is a collection of related data with an implicit meaning and hence is a database

More Specific Properties of a Database

- A database represents some aspect of the real world
  - Changes to the real world are reflected in the database
- A database is a logically connected collection of data with some meaning
  - A random assortment of data cannot correctly be called a database
- A database is designed, built, and populated with data for a specific purpose
  - It has an intended group of users and some applications in which these users are interested

Database Management Systems

- A database management system (DBMS) is a collection of programs that enables users to create and maintain a database

- The DBMS is a general-purpose software system that allows the processes of **defining**, **constructing**, **manipulating**, and **sharing** databases among various users and applications

DBMS Models

- Nowadays most databases are relational.
- This is a specific model and this module we will concentrate most on this model.
- Other models that have been used are
  - Hierarchical
  - Network
- Other models have been suggested in recent years
  - Object-Oriented
  - Trans-relational
  - NOSQL (Not Only SQL)

Managing Data

- Businesses have always maintained data
- For centuries this was done on paper
- Huge files that had to be manually searched in order to find information
- The advent of computers allowed electronic storage
- First attempts stored electronic versions of documents
- The first major problem was how to store financial information

Spreadsheets

Spreadsheets

- Spreadsheets have been used by accountants for centuries.
- Very early electronic versions of spreadsheets were developed for mainframe computers in the 1960s.
- Modern computerised spreadsheets began with Daniel Bricklen and Bob Frankston.
- They developed VisiCalc in 1978, which became the basis for all electronic spreadsheets since.

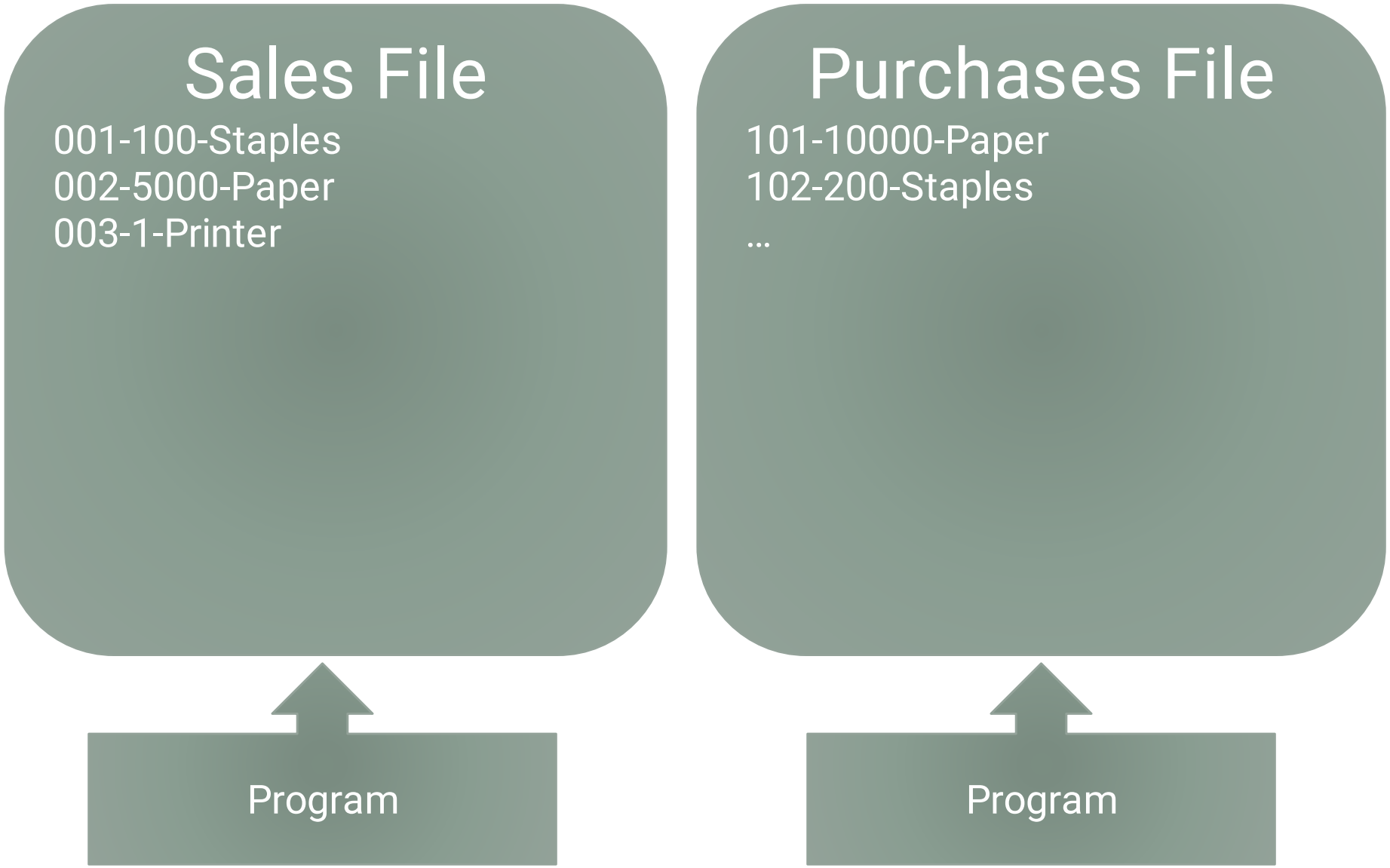
VisiCalc


A1
C
(C) 1979,1981 Software Arts, Inc.
VC-176Y2-IBM-TEST
43
#0000000
A B C D E F G H
ITEM NO. UNIT COST
MUCK RAKE 43 12.95 556.85
BUZZ CUT 15 6.75 101.25
TOE TONER 250 49.95 12487.50
EYE SNUFF 2 4.95 9.90
SUBTOTAL 13155.50
9.75% TAX 1282.66
TOTAL 14438.16

Storing Data

- Storing data as simply electronic copies of existing documents is not sufficient.
- How do we search for information?
  - How many BSc II students do we have?
  - Is Sean Russell in the list?
  - What modules is he registered for?
- This type of **query** requires that we change how things are done.
- People then began to look at how to model data for electronic use.

File Based Database



File Based Database Systems

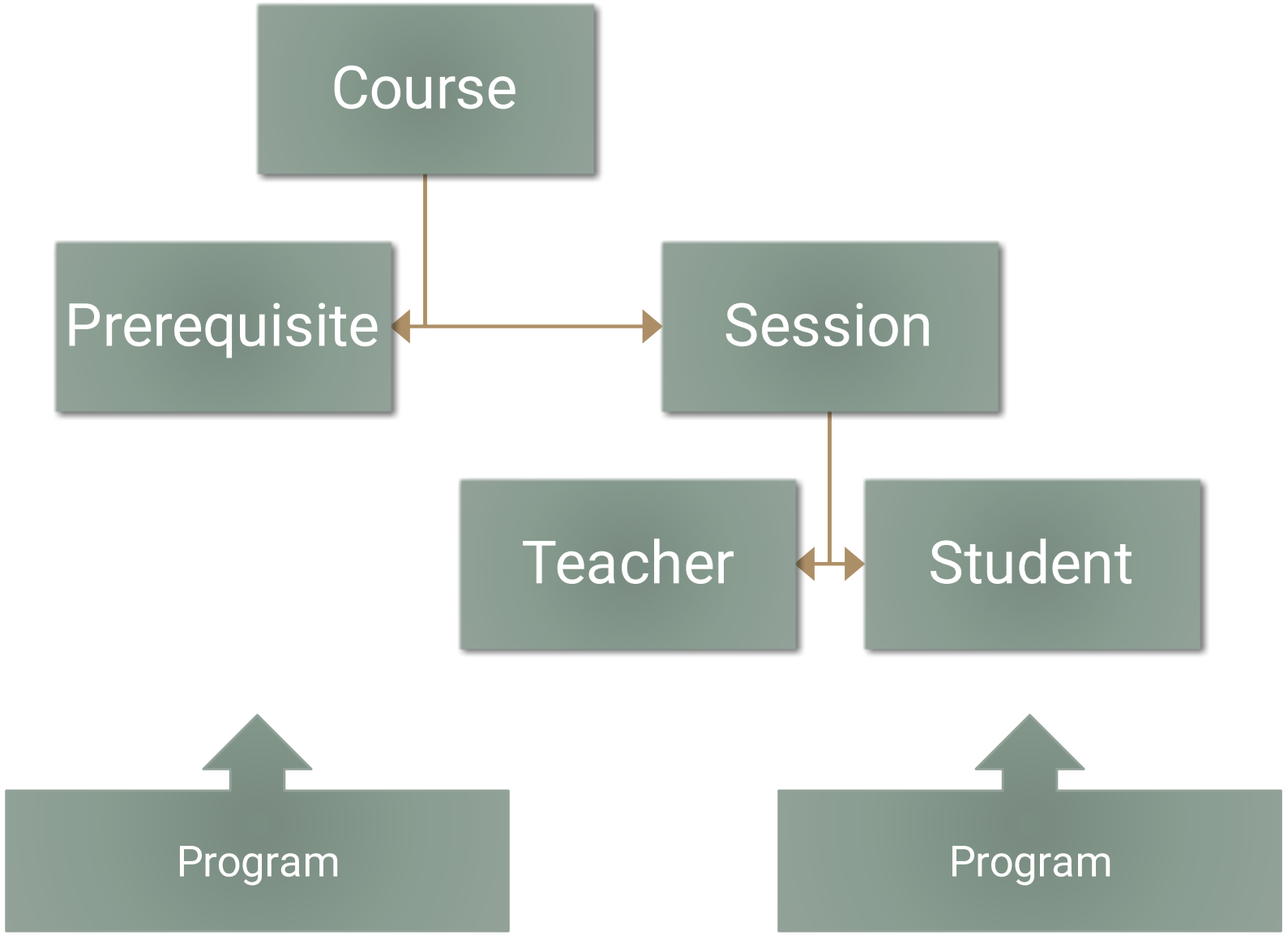
- Disadvantages
  - No query language
  - No scalability
  - Hard to update schema and modify data
  - Recovery?

- Advantages
  - More lightweight than a DBMS
  - Might be good enough for small data, for example a personal address book

Hierarchical Database Model

- The oldest DBMS model is the hierarchical model (Information Management System from IBM)
- Developed in the 1960s to overcome the problems with file processing
- The model was not standardised
- It was based on a tree structure consisting of nodes, branches and roots
- It allowed for 1 to many relationships
- The best known implementation is IMS by IBM

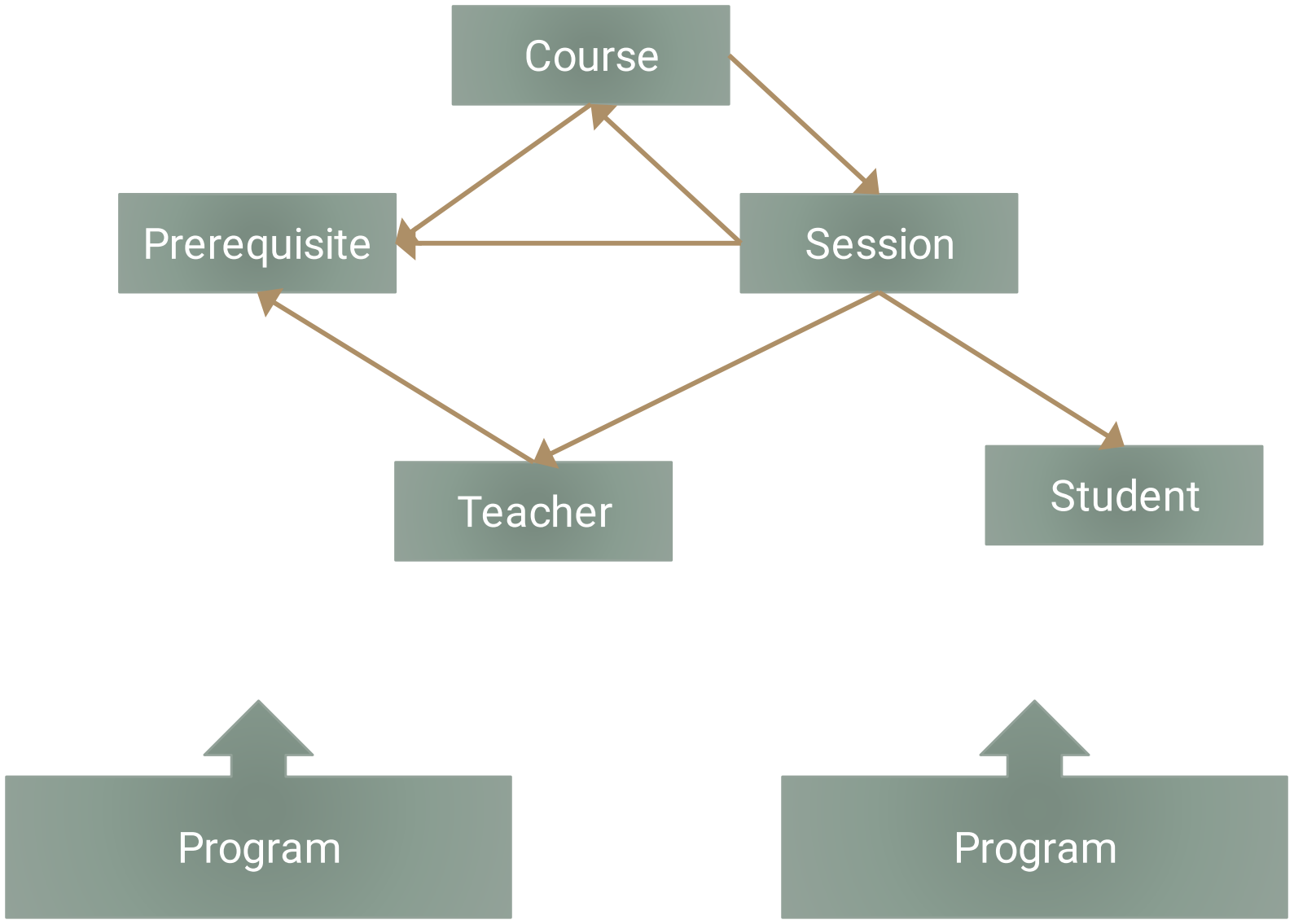
Hierarchical Based Database



Network Database System

- The network model was developed by committee
- The Database Task Group (DBTG) of CODASYL (COnference on DAta SYstems Languages) developed the model so that it could be standardised
- CODASYL are the people who developed the COBOL programming language and much of the language of the network model looks like COBOL
- The model allows the implementation of many-to-many (M:M) relationships, which ultimately get translated into one-to-many (1:M).

Network Based Database



Relational Database Model

- Formulated by Edgar Codd of IBM in 1970.
- Commercial RDBMS in 80s.
- “Codd's 12 Rules” (actually 13) that all RDBMSs should follow.
- Most widely used model at present
  - Access, Oracle, MySQL, MariaDB, MS SQL Server, DB2, Sybase ASE, PostgresSQL etc.

Relational Concepts

- Data is represented as collections of **relations**
- Each relation is **table** of values
- Each table consists of **rows** and **columns**
- Each **row** represents an **entity** or **record**
- Rows are unordered

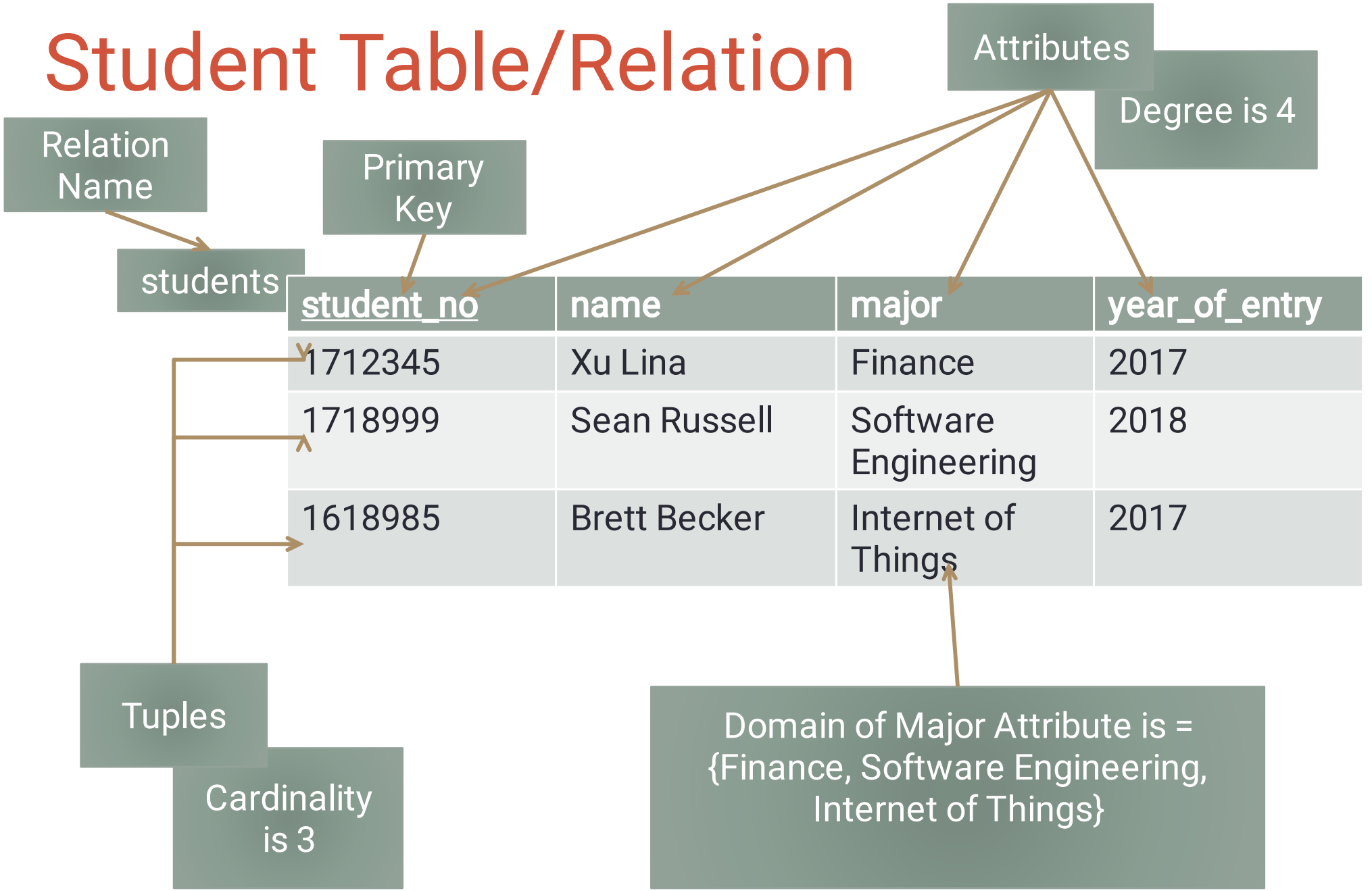
Relational Concepts

- No duplicate rows are allowed
- Each relation has a **primary key**, the value of which uniquely identifies the **record/entity**
- Each column represents an **attribute**
- Table name and column name are used to help interpret the values

Database Terminology

- **Relation** is a mathematical term for a **table**
- **Row** is called a **Tuple**
- **Column** is called an **Attribute**
- **Domain** is used to describe the types of values that can appear in a column
- **Degree** is the number of attributes
- **Cardinality** – the number of tuples/rows in a relation
- **Atomic Value** – precisely one value at each row intersection
- **Null Value** – Missing, not known or irrelevant data (not the same as zero or blank)

Student Table/Relation



Advantages of Database Approach

- Data can be **shared**
- **Redundancy** can be reduced
- **Integrity** can be maintained
- **Security** can be enforced
- Conflicting **requirements** can be balanced
- **Standards** can be enforced

Redundancy happens when the same data is stored in 2 or more places

Integrity means that the data in the database is accurate and consistent

NoSQL

- NoSQL databases were invented in the 2000s to deal with some new challenges that traditional relational databases struggled with. Especially useful for large, unstructured data, and data that does not need to be updated instantly.
- Different types:
  1. **Document databases** pair each key with a complex data structure known as a document. Documents can contain many different key-value pairs, or key-array pairs, or even nested documents. Typical example: *MongoDB*
  2. **Graph stores** are used to store information about networks of data, such as social connections. Graph stores include *Neo4J* and *Giraph*.

NoSQL (continued)

3. **Key-value stores** are the simplest NoSQL databases. Every single item in the database is stored as an attribute name (or 'key'), together with its value. Examples of key-value stores are *Riak* and *Berkeley DB*. Some key-value stores, such as *Redis*, allow each value to have a type, such as 'integer', which adds functionality.

4. **Wide-column stores** such as *Cassandra* and *HBase* are optimized for queries over large datasets, and store columns of data together, instead of rows.

MySQL

- We will mostly be using MySQL Community Server (latest version is 8.0.15)
- Download from here: https://dev.mysql.com/downloads/mysql/
- You need to install this before your first lab in Week 3.