Introduction to Databases
(INFR10080)

Dr Paolo Guagliardo

THE UNIVERSITY *of* EDINBURGH
informatics

v26.0

Changelog

v26.0 Initial version

Data

The **most important asset** of any enterprise

Must be effectively, efficiently and reliably

- collected and stored
- maintained and updated
- processed and analysed

to be turned into meaningful information

$\Longrightarrow$ Enable and **support decision making**

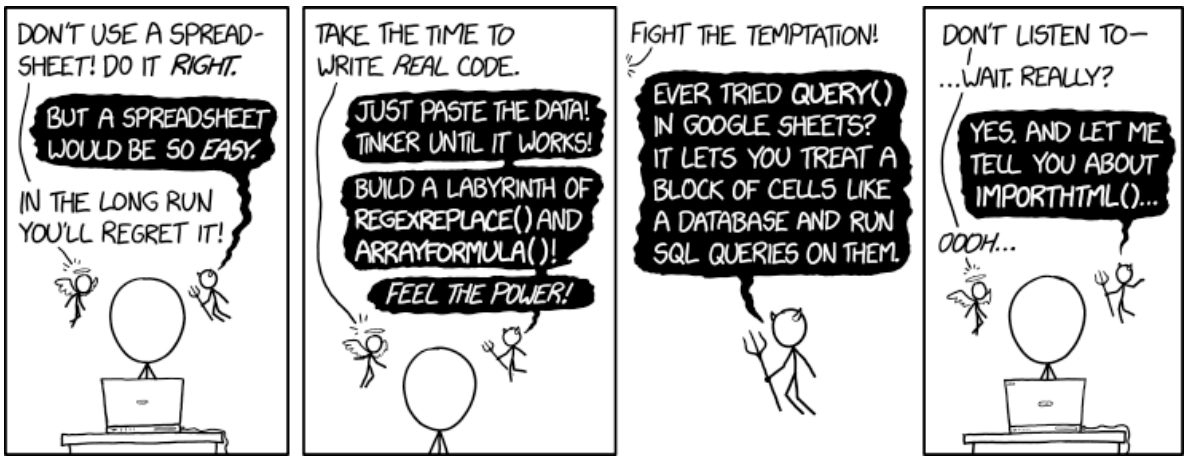
What is a database?

A collection of data items related to a specific enterprise, which is structured and organized so as to be more easily accessed, managed, and updated

Database **M**anagement **S**ystem (DBMS)

- software package for creating and managing databases
- mediates interaction between end-users (incl. applications) and the database
- ensures that data is consistently organized and remains easily accessible

Why use a DBMS?



Source: https://xkcd.com/2180

Why use a DBMS?

- Uniform data administration
- Efficient access to resources
- Data independence
- Reduced application development time
- Data integrity and security
- Concurrent access
- Recovery from crashes

Different kinds of data(bases)

- A **data model** is a collection of concepts for describing data
- A **schema** is a description of a particular collection of data, using a given data model

Relational databases ⟸ main focus of this course

Data organised in tables (relations) with typed attributes

Document stores

Text documents structured using tags (or other markers)

Graph databases

Data organised in graph structures with nodes and edges

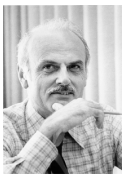
Key-value stores

Data organised in associative arrays (a.k.a. dictionaries or maps)

The relational model

First proposed by Edgar F. Codd in 1970

Simple idea: Organise data in **tables** (relations)



Schema

- Set of **table names**
- List of distinct (typed) **column names** for each table
- **Constraints** within a table or between tables

Instance

- Actual data (that is, the rows of the tables)
- Must satisfy typing and constraints

Example: relational database

Customer

CustID	Name	City	Address
cust1	Renton	Edinburgh	2 Wellington Pl
cust2	Watson	London	221B Baker St
cust3	Holmes	London	221B Baker St

Account

Number	Branch	CustID	Balance
243576	Edinburgh	cust1	-120.00
250018	London	cust3	5621.73
745622	Manchester	cust2	1503.82

Query languages

Used to ask questions (queries) to a database

Procedural

Specify a **sequence of steps** to obtain the expected result

Declarative

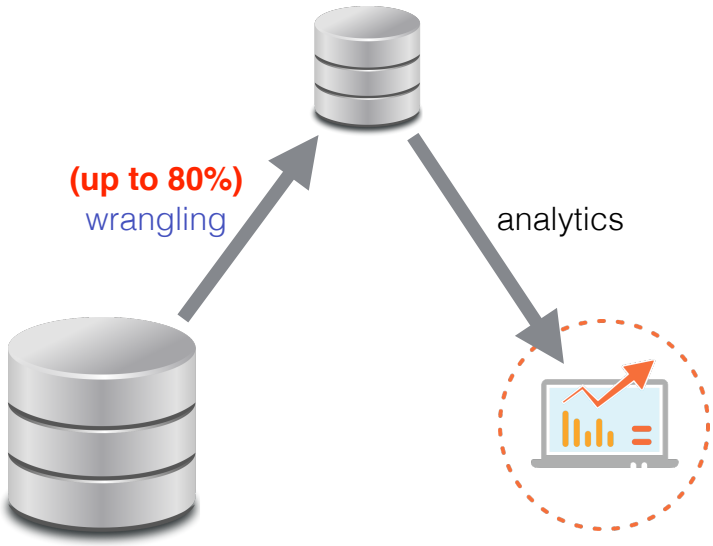
Specify **what** you want not **how** to get it

- Queries are typically asked in a declarative way
- DBMSs figure out internally how to translate a query into procedures that are suitable for getting the results

SQL

- **S**tructured **Q**uery **L**anguage
- Declarative language for querying relational databases
- Implemented in all major (free and commercial) RDBMSs
- First standardized in 1986 (ANSI) and 1987 (ISO); several revisions afterwards (latest June 2023)
- Multi-billion-dollar business
- Most common tool used by data scientists

Accessing data is at the core of data analysis



Studying SQL is not enough

DBMSs encompass many areas of Computer Science:

Operating systems, Algorithms and data structures,
Formal logic, (Programming) languages, Multimedia, ...

Goals of this course

- **Create and modify a relational database**
  using standard software tools available on the market
- Compare strengths and weaknesses of different **database designs**
- Process and analyse data by means of **complex SQL statements**
- Formulate and manipulate queries
  in both **declarative and procedural database languages**
- Reason about the correctness and consistency
  of **concurrent database interactions** among multiple users

Syllabus

Core topics

- Query languages: **SQL**, relational algebra and calculus
- Database design: constraints and normal forms
- Scheduling and concurrency control: serializability, locking
- Database access from applications: embedded/dynamic SQL

Advanced topics (if time allows)

- Deductive databases: Datalog and recursive queries
- Incomplete data: missing values and certain answers

Prerequisites

- Some background in discrete mathematics
- Familiarity with **predicate logic** is a plus
  (but this will be introduced during the course)
- Familiarity with **Unix command line** is a plus
  (knowing the basics will make your life easier)
- No specific programming requirements
  (we will see some very simple Python programs)

The course is overall self-contained

Course links and resources

Handouts **Informatics Open Course Materials**

https://opencourse.inf.ed.ac.uk/idb

Announcements **Learn**

https://www.learn.ed.ac.uk/ultra/courses/_134859_1/outline

Discussions **EdStem**

https://edstem.org/eu/courses/3607/discussion

Video recordings **echo360**

https://echo360.org.uk/section/5432dd61-d995-41eb-8167-ea4cea5b8561/home

Timetable **Course Timetable Browser**

https://browser.ted.is.ed.ac.uk/generate/?courses[]=INFR10080_SV1_SEM1&period=SEM1

Lectures

Weeks 1-11 Three hours per week

Tuesdays 13:10-14:00
Usha Kasera Lecture Theatre, Old College

Thursdays 13:10-15:00
Lecture Theatre G.04, Robson Building

Week 11 is dedicated to **revision** (we typically solve a past exam)

Office hours

- Once a week, **from Week 2**, on Wednesdays 13:30-14:30
- In person in my office (Informatics Forum, room 5.11)

Tutorials

Weeks 3-10

Two sessions to discuss formative **exercises** with detailed solutions

Mondays 13:10-14:00
Lecture Theatre C, 40 George Square

Tuesdays 11:10-12:00
G.07 Lecture Theatre, Medical School (Teviot)

You will attend **only one** of these (your choice which one)

Exercises will be made available in advance for self-study

Strongly adviced to

- review the material covered after each lecture
- attempt the assigned exercises
- attend tutorials every week, in person

Coursework

Formative assignment (optional, but strongly advised)

- Requires writing queries in SQL and Relational Algebra
- Released in Week 4, due in **Week 6**
- Submission via Learn
- Feedback: individual automarked results + class discussions

Summative assignment (accounts for **20%** of the final mark)

- Similar to the formative assignment but **marked**
- Released in Week 7, due in **Week 9**
- Submission via Learn
- Feedback: individual automarked results

Exam

Accounts for **80%** of the final mark

Diets

December 2026 main examination, open to all students

August 2027 resit examination, only for selected students

Structure

- 6-8 questions, all compulsory for full marks
- For sample questions, search for
  - Database Systems (up until 2019), or
  - Introduction to Databases (from 2020)

  at `https://exampapers.ed.ac.uk/`

Software: REAL

- A cross-platform interpreter for relational algebra
- Written in Java, open-source
- Public Git repository hosted at
  `https://git.ecdf.ed.ac.uk/pguaglia/real`
- This is an **educational tool** not meant for production
- Used in the coursework
- More details later...

Software: PostgreSQL

- Open-source, commercial-level relational DBMS
- Available for Windows, Mac, Linux (and more)
- Installed on all **DICE machines**
- Each enrolled student has their own **personal database** (hosted on the university's central PostgreSQL server)
- Instructions for using PostgreSQL on DICE are on Learn
- Used in the coursework
- More details later...