

# Introduction to Databases

## (INFR10080)

Dr Paolo Guagliardo



Fall 2020 (v20.1.0)

## 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**

⇒ 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 Management System (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?

- ▶ 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

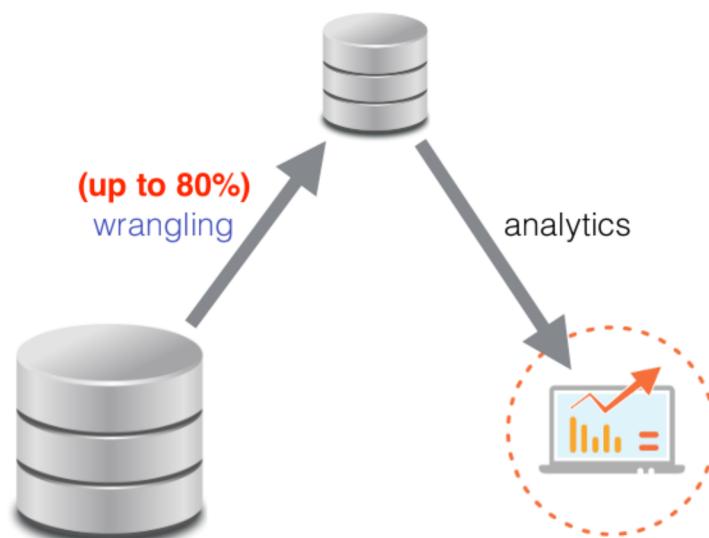
Specify **what** you want not **how** to get it

- ▶ 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

- ▶ Structured Query Language
- ▶ 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 Dec 2016)
- ▶ 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
- ▶ Understand the **how queries are optimized and executed**  
in relation with **how data is stored and organised**

# 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
- ▶ Query evaluation and optimisation: join strategies, query plans

## Advanced topics (if time allows)

- ▶ Deductive databases: Datalog and recursive queries
- ▶ Incomplete data: missing values and certain answers
- ▶ Storage and indexing: B+ trees, static hashing

# 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

## Textbook (1)

### Main text

*Ramakrishnan, Gehrke:*  
Database Management Systems  
McGraw-Hill, 3rd edition

### Not mandatory

Materials from lectures and tutorials are enough

### Availability

- ▶ **Main Library** (George Square): **3 copies** (3 hours loan)
- ▶ **Murray Library** (King's Buildings): **6 copies** (12 weeks loan)
- ▶ **Blackwell's** (Nicholson St): **10% student discount**

## Textbook (2)

### Further reading

*Abiteboul, Vianu, Hull*  
Foundations of Databases  
Addison-Wesley, 1995

- ▶ Mostly theoretical topics
- ▶ Out of print but freely available (for **personal use only**)  
<http://webdam.inria.fr/Alice/>

## Course website(s)

All course materials and announcements are on **Learn**

Class discussions will take place on **Piazza**:

<https://piazza.com/ed.ac.uk/fall2020/infr10080/home>

No need to signup for the class

I will enrol you using your **uun email address**

Rather than emailing questions, post them on Piazza

- ▶ You can post **privately** to instructors (tutors and me)
- ▶ You can post **anonymously** to classmates

## Tutorials

- ▶ They will start in week 3 or week 4
- ▶ Discuss **formative** exercises assigned throughout the course
- ▶ Tutorial sheets will be made available one week in advance
- ▶ Marks for submitting **attempts** at solving the exercises
- ▶ Solutions to tutorial exercises will be posted on Learn

## Coursework

Accounts for **60%** of the final mark

### Participation in tutorials **10%**

- ▶ Submitting attempts at solving assigned exercises (independently of correctness) **Max marks: 70/100**
- ▶ Actively contributing to Piazza discussions (for **extra marks**)

### SQL assignment **25%**

- ▶ Requires writing SQL queries to a given specification
- ▶ Marked automatically (details later on)
- ▶ Possibly there will be an optional dry-run

### Online test **25%**

- ▶ Taken from home via Learn
- ▶ Multiple-choice questions, plus 1 to 3 open-ended ones
- ▶ Open-ended questions marked only if the rest are correct

# Exam

Accounts for **40%** of the final mark

## Diets

- ▶ December 2020: open to all students

## Structure

- ▶ See past exam papers for sample questions:  
<https://exampapers.ed.ac.uk/>

## Software: PostgreSQL

- ▶ Open-source, commercial-level relational DBMS
- ▶ Installed on all **DICE machines**
- ▶ Available for Windows, Mac, Linux (and more)
- ▶ Very simple to compile and install on your laptop
- ▶ Each enrolled student has their own **personal database** (hosted on the university's central PostgreSQL server)
- ▶ Instructions will be posted on Piazza
- ▶ You will use it to write **SQL queries** for the coursework

# Other stuff

## Lecture recording

- ▶ Lectures will be recorded
- ▶ Links to the recordings will be **posted on Learn**

## Lecture slides

- ▶ Handouts for all topics are available on Learn
- ▶ Keep an eye out for the **latest version**  
*before class:* version number next to view/download links  
*after class:* an announcement will also be sent out

## Office hours

- ▶ **Weekly meetings** (interest must be registered in advance)
- ▶ I am usually available for quick questions after class  
(but do not forget to **use Piazza** where possible)