Introduction to Databases

(INFR10080)

(Course Introduction)

(Fall 2025)

THE UNIVERSITY *of* EDINBURGH

Changelog

v25.0 Initial version

Course Objectives

You will learn:

- What is a database/database system?
- How to query a database?
- How to design a database?
- How to update a database?
- How to use a database in your applications?

This course dose not cover:

- How to develop database systems* ⟸ check out ADBS
- non-relational databases ⟸ check out MLS

*But knowing how they work would help with using databases (advance topics)

Why you should take this course

SQL's enduring popularity (over half-century old)

- go-to/only language for accessing and operating databases
- popular language among professional developers for decades

$$$ ...

Oracle

Larry Ellison briefly overtakes Elon Musk as world's richest person

Oracle co-founder's shares rose by 40% in early trading, valuing his fortune at $393bn, just ahead of Musk's $384bn

Dan Milmo *Global technology editor*

Wed 10 Sep 2025 23.45 BST

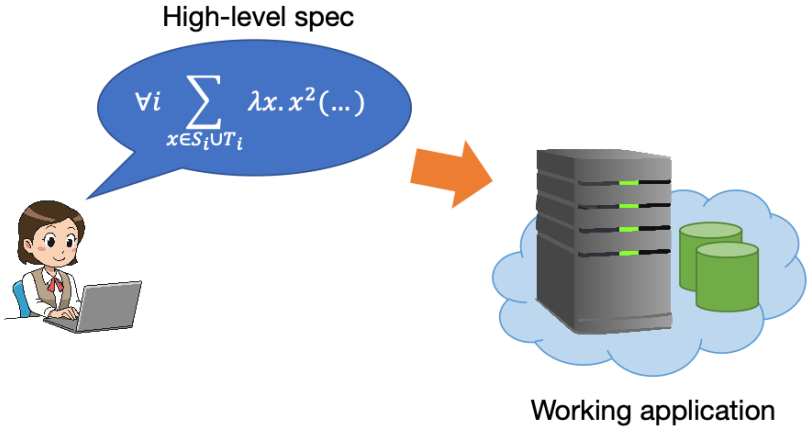
Share

Source: theguardian.com/technology/2025/sep/10/larry-ellison-dislodges-elon-musk-as-worlds-richest-person

Why you should take this course

In many ways, SQL systems are the highest-level successful programming abstractions

Programming: The Dream

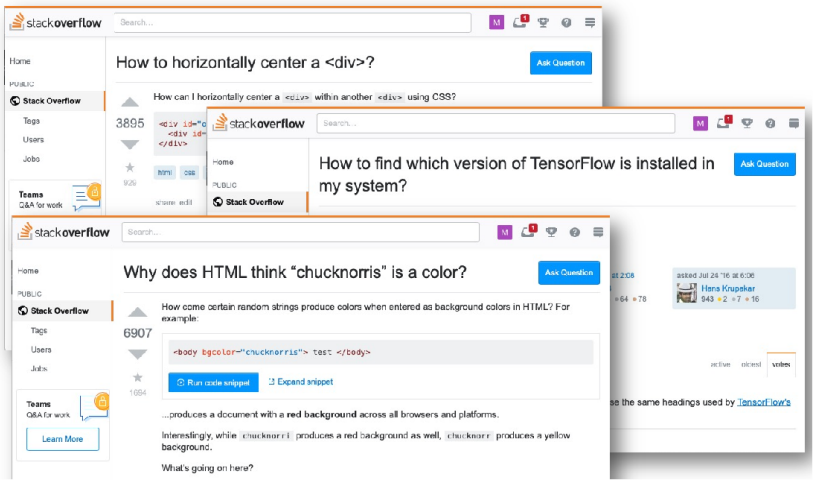


Source: CS 245: Principles of Data-Intensive Systems (Winter 2021), Stanford University https://cs245.stanford.edu

Why you should take this course

In many ways, SQL systems are the highest-level successful programming abstractions

Programming: The Reality

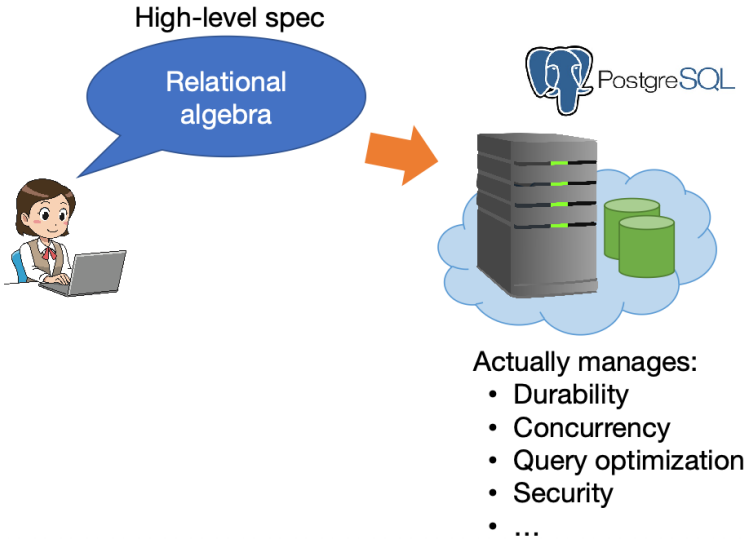


Source: CS 245: Principles of Data-Intensive Systems (Winter 2021), Stanford University https://cs245.stanford.edu

Why you should take this course

In many ways, SQL systems are the highest-level successful programming abstractions

Programming with Databases



Source: CS 245: Principles of Data-Intensive Systems (Winter 2021), Stanford University https://cs245.stanford.edu

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**

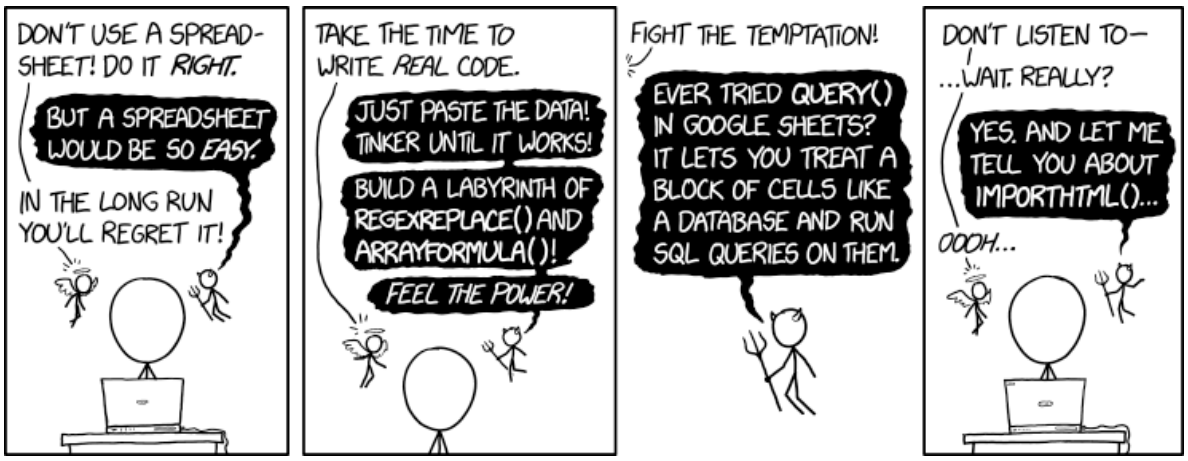
What is a database?

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 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?



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

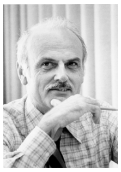
Data organised in graph structures with nodes and edges

Vector databases

(Unstructured) data represented as embedding vectors

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

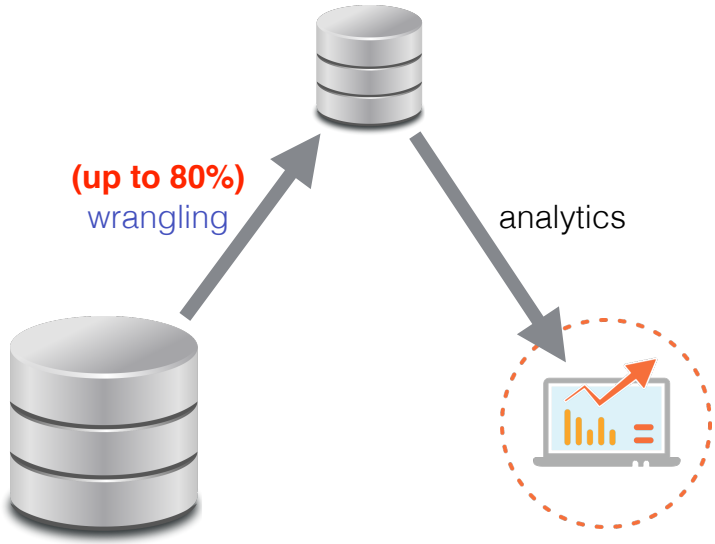
Procedural	**Declarative**
Specify a **sequence of steps** to obtain the expected result	Specify **what** you want not **how** to get it

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

Course Staff

Instructor: Yang Cao

lecturers, office hours (in-person)

Course TA: Tianjian Yang

office hours (online via Collaborate)

more details later

Textbook (1)

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

- Available from the library
- **Not mandatory**
  Lectures + materials posted on Learn are enough
  (this is what the assessment is based on)

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**

The link to the class is on Learn

All students enrolled in Learn are automatically signed up on Piazza

Rather than emailing questions, post them on Piazza

- You can post **privately** to instructors
- You can post **anonymously** to classmates

Most questions will be answered during office hours

Lectures

Weeks 1-11

Three hours per week

Thursdays 15:10-16:00
Lecture Theatre G.04 - Robson Building

Fridays 10:00-12:00
Usha Kasera Lecture Theatre - Old College

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

Tutorials

There will be no formal tutorials required to attend.

Instead, tutorials are replaced by **Exercises** and **Office Hours**.

Exercises

Formative exercises: questions with detailed solutions Exercises will be made available in advance for self-study

Strongly adviced to

1. check the lecture slides and exercise before each lecture
2. review the materials after each lecture
3. attempt the assigned exercises
4. ask questions on Piazza & during office hours

Office hours

Tailored 1-1 learning for your particular case. More effective than forced in-classroom tutorials.

Time

- Once a week, **from Week 2**, on Thursdays 13:45-14:45
- In person in my office (Informatics Forum, room 4.32A)
- Online on Blackboard Collaborate: https://eu.bbcollab.com/guest/f8d181c853b540bfae434e490627462c

During these sessions, we will answer questions:

- asked live (with priority to those who come in person)
- selected from those posted on Piazza

Coursework

Formative assignment (optional, but strongly advised)

- Requires writing queries in SQL and Relational Algebra
- Released in Week 5, 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 8, due in **Week 9**
- Submission via Learn
- Feedback: individual automarked results

Exam

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

Diets

December 2025 main examination, open to all students

August 2026 resit examination, still unclear who is allowed

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: 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…

(We may also use other DBMSs (e.g., SQLite, duckdb) for in-classroom demos)

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…

Other stuff

Video recordings

- All lectures will be recorded
- The recordings will be available on **echo360** (link on Learn)

Lecture slides

- Handouts for all topics will be made available on Learn
- Keep an eye out for the **latest version**

  In case of changes

  before class: updated version number next to the link

  after class: an announcement will also be sent out