THE UNIVERSITY *of* EDINBURGH

Advanced Database Systems

Spring 2024

Lecture #01:

Course Introduction

Essential Questions

Why take this course?

What is this course about?

Who is running this course?

How will this course work?

Why? Reason #1: Utility

Data processing backs essentially every application

Databases of one form or another back most applications

The **principles** taught in this course back nearly everything in computing

Knowing how to manage data is a vital, core asset in today’s world

This material will empower you as a computer scientist

WHY? REASON #2: CENTRALITY

Data is at the centre of modern society

Much cheaper to generate data

Sensors, smart devices, social networks,

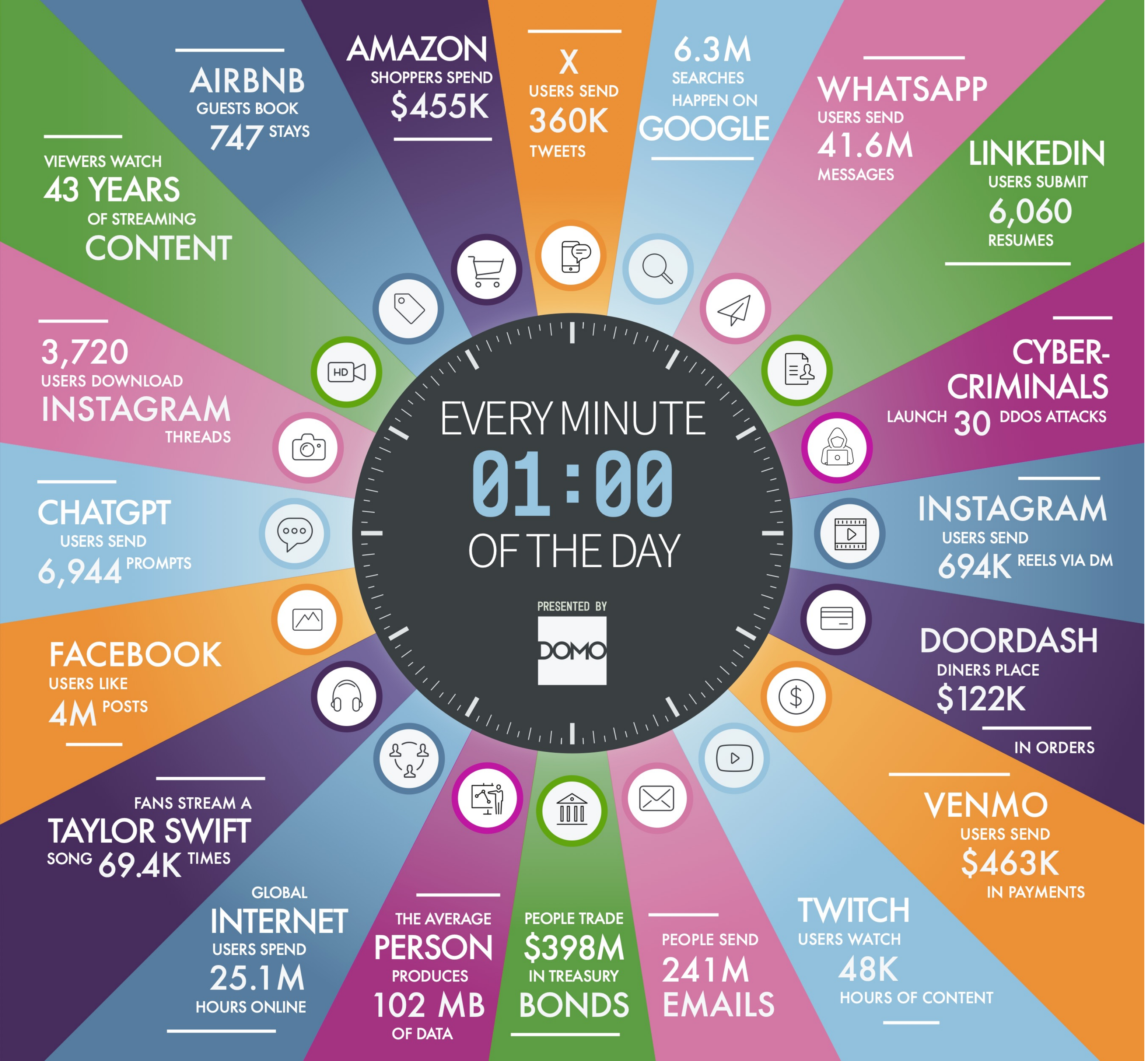
online games, software logs, audio & video

Much cheaper to process data

Cloud computing, open-source software,

heterogenous architectures (CPU, GPU, FPGA)

How much data is generated *every minute?*



EVERY MINUTE

01:00

OF THE DAY

PRESENTED BY DOMO

AIRBNB GUESTS BOOK 747 STAYS

AMAZON SHOPPERS SPEND $455K

X USERS SEND 360K TWEETS

6.3M SEARCHES HAPPEN ON GOOGLE

WHATSAPP USERS SEND 41.6M MESSAGES

LINKEDIN USERS SUBMIT 6,060 RESUMES

CYBER-CRIMINALS LAUNCH 30 DDOS ATTACKS

INSTAGRAM USERS SEND 694K REELS VIA DM

DOORDASH DINERS PLACE $122K IN ORDERS

VENMO USERS SEND $463K IN PAYMENTS

TWITCH USERS WATCH 48K HOURS OF CONTENT

PEOPLE SEND 241M EMAILS

PEOPLE TRADE $398M IN TREASURY BONDS

THE AVERAGE PERSON PRODUCES 102 MB OF DATA

GLOBAL INTERNET USERS SPEND 25.1M HOURS ONLINE

FANS STREAM A TAYLOR SWIFT SONG 69.4K TIMES

FACEBOOK USERS LIKE 4M POSTS

CHATGPT USERS SEND 6,944 PROMPTS

3,720 USERS DOWNLOAD INSTAGRAM THREADS

VIEWERS WATCH 43 YEARS OF STREAMING CONTENT

https://www.domo.com/learn/data-never-sleeps-11

(2023 edition)

Scale of Scientific Data

Large Hadron Collider, CERN

Raw data: **600,000,000 GB/sec**

(19 ZettaBytes/year) Zetta = 10^{21}

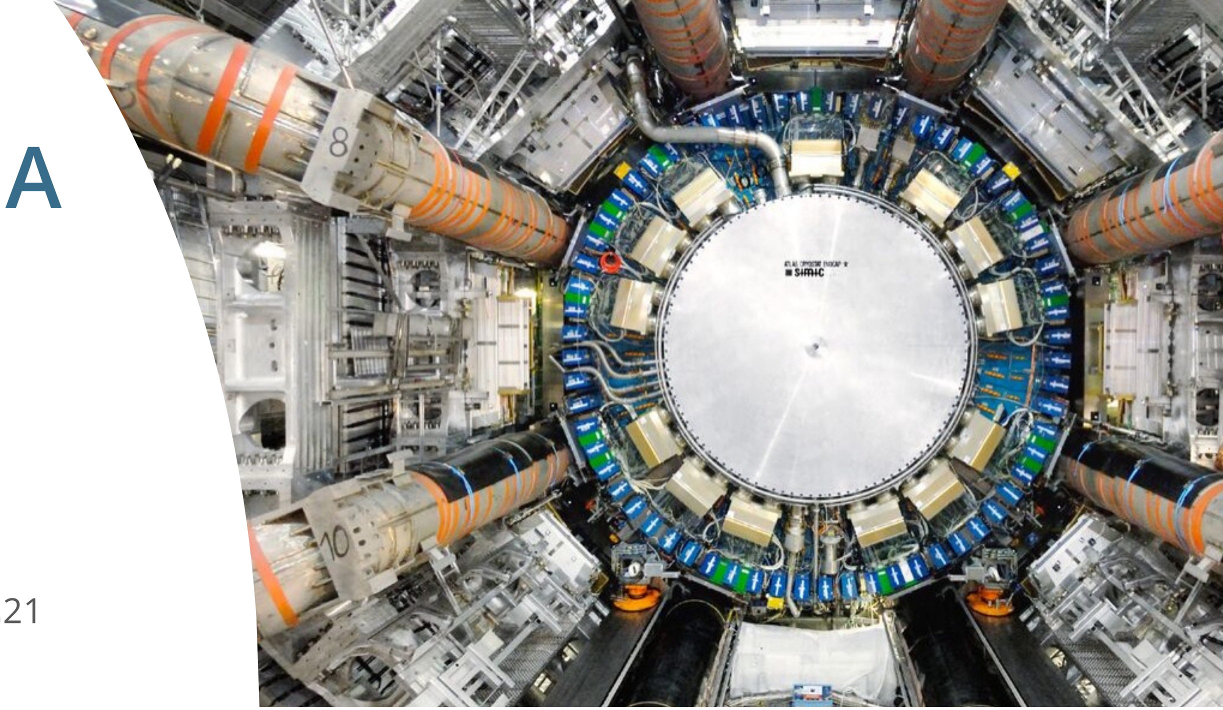
Downsampled: **25GB/sec**

(788 PetaBytes/year) Peta = 10^{15}

Downsampled further: **1050MB/sec**

(33 PetaBytes/year)

https://home.cern/science/computing/processing-what-record



Why? Reason #2: Centrality

Data is at the centre of modern society

Much cheaper to generate data

- Sensors, smart devices, social networks, online games, software logs, audio & video

Much cheaper to process data

- Cloud computing, open-source software, heterogenous architectures (CPU, GPU, FPGA)

The infrastructure determines what's possible

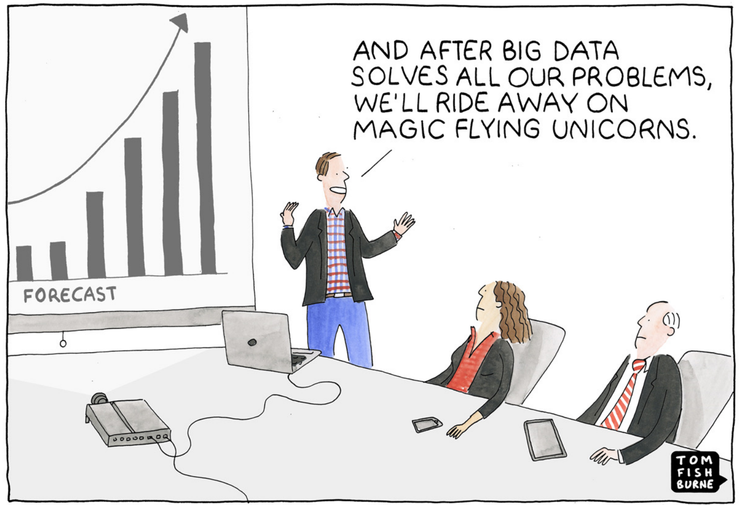
Why? Reason #3: The Core of Computing

Data growth will continue to outpace computation

Philosophy: more data → more value?

Systems for managing data at scale: the core of modern computing

Techniques you learn in this course underlie many topics in computing


FORECAST
AND AFTER BIG DATA SOLVES ALL OUR PROBLEMS, WE'LL RIDE AWAY ON MAGIC FLYING UNICORNS.
TOM FISH BURNE

© marketoonist.com

Essential Questions

Why take this course?

What is this course about?

Who is running this course?

How will this course work?

What is a Database?

A database is an organised collection of inter-related data that models some aspect of the real world

Databases are the core component of most computer applications

- Banking
- Web and mobile apps
- Online retailers
- Human resources

Sometimes confused with a Database Management System

What is a DBMS?

A database management system (DBMS) is software that **stores**, **manages**, and **facilitates** access to databases

Mediates interactions between users and databases

Traditionally, DBMS refers to relational databases

SQL, ACID transactions, prevent data loss

This will be the focus of this course!

Warning: market and terms in rapid transition

The tech remains (roughly) the same

Good time to focus on fundamentals!

ORACLE®

Microsoft® SQL Server®

MySQL®

SQLite

PostgreSQL

Why using a DBMS?

Consider one typical scenario:

1. Create a database that models a university organisation to keep track of students, instructors, and courses

2. Build an application to support typical operations on the DB:

    Add new students, instructors, and courses

    Register students for courses and generate class rosters

    Assign grades to students, compute GPA, and generate transcripts

Flat File Strawman

Store our database as comma-separated value (CSV) files

Instructor(name, dept, salary)

```
“Jones”, “CS”, 95000
“Smith”, “Physics”, 75000
“Gold”, “CS”, 62000
```

instructor.csv

Course(name, instructor, year)

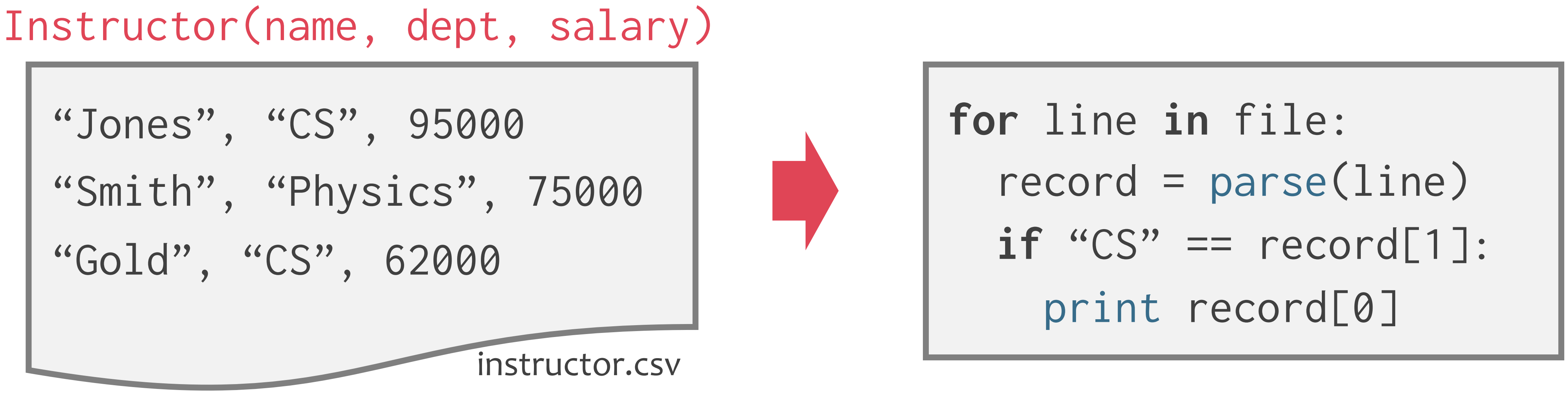
```
“Databases”, “Jones”, 2018
“Quantum M.”, “Smith”, 2017
“Compilers”, “Jones”, 2017
```

course.csv

Apps have to parse the files each time they want to read/update records

Flat File Strawman

Example: Get the names of all computer science instructors



```
Instructor(name, dept, salary)
```

```
“Jones”, “CS”, 95000
“Smith”, “Physics”, 75000
“Gold”, “CS”, 62000
```

instructor.csv

```
for line in file:
  record = parse(line)
  if “CS” == record[1]:
    print record[0]
```

Tight coupling between application logic and physical storage

Flat File: Drawbacks

Data redundancy

- Duplication of information in different files
- Ex: changing string “CS” to “Computer Science” requires rewriting several files

Storage format needs to be exposed

- Developers need to be aware of the physical layout of data
- Data may be stored in various file formats such as CSV, JSON, binary, etc.

Difficulty in accessing data

- Need to write a new program to carry out each new task
- Programming complex logic on several files can be error-prone and inefficient

Flat File: Drawbacks (Cont.)

Search is expensive (no indexes)

Cannot find tuple with given key quickly

Always have to read the entire file

No atomicity of updates

Failures may leave database in an inconsistent state with partial updates carried out

Ex: moving money between two accounts should either complete or not happen at all

Integrity problems

Integrity constraints (e.g., course mark must be ≥ 0) become “buried” in program code

Hard to add new constraints or change existing ones

Flat File: Drawbacks (Cont.)

Concurrent access by multiple users

Concurrent access needed for performance

Uncontrolled concurrent accesses can lead to inconsistencies

No security

Hard to provide user access to some, but not all, data

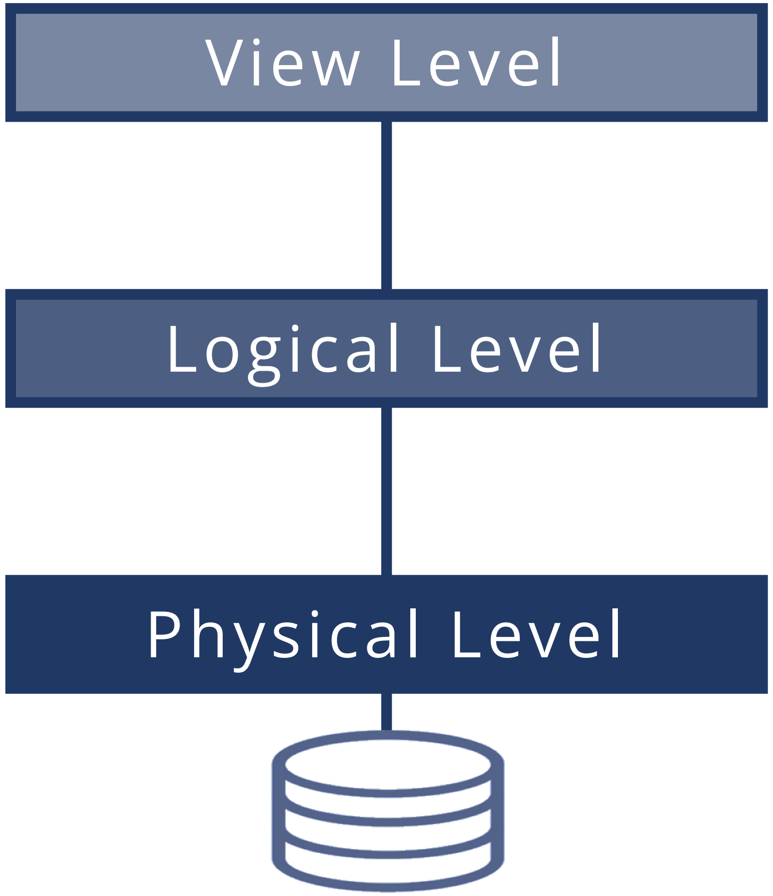
Storing data in raw CSV files is insecure

No application programming interface

How can a payroll program access the data?

Database systems offer solutions to all the above problems

Levels of Abstractions



Simplifies interaction with the database, hides info (e.g., salary) for security purposes

Describes data stored in the DB

```
type instructor = record
  name: string;
  dept: string;
  salary: integer;
end
```

Describes how a record is stored

Data independence:
Insulate users from changes in lower levels

Data Models

Data model

Collection of concepts for describing the data in a database

Schema

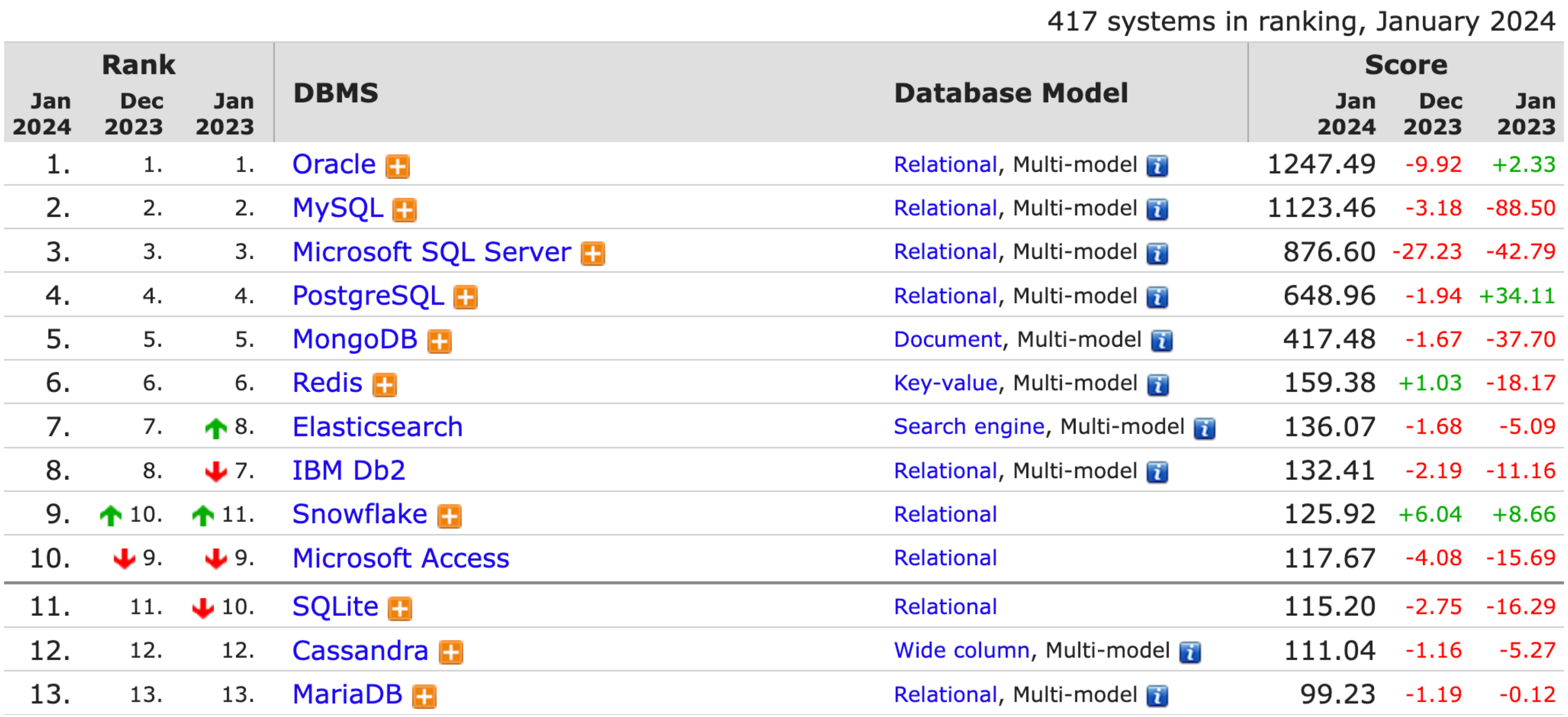
Description of a particular collection of data, using a given model

Models in practice

Relational, key-value, graph, document, array, hierarchical, network

Most DBMSs implement the relational data model

Ranking of DBMS Technologies 2024



417 systems in ranking, January 2024

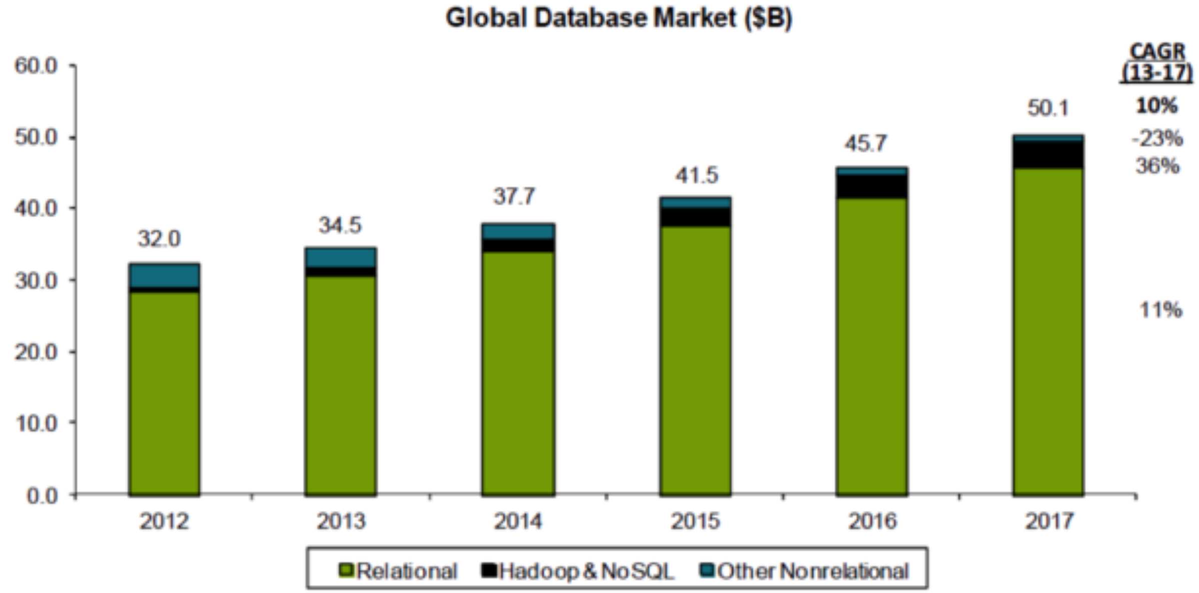
Rank			DBMS	Database Model	Score		
Jan 2024	Dec 2023	Jan 2023			Jan 2024	Dec 2023	Jan 2023
1.	1.	1.	Oracle	Relational, Multi-model	1247.49	-9.92	+2.33
2.	2.	2.	MySQL	Relational, Multi-model	1123.46	-3.18	-88.50
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	876.60	-27.23	-42.79
4.	4.	4.	PostgreSQL	Relational, Multi-model	648.96	-1.94	+34.11
5.	5.	5.	MongoDB	Document, Multi-model	417.48	-1.67	-37.70
6.	6.	6.	Redis	Key-value, Multi-model	159.38	+1.03	-18.17
7.	7.	↑ 8.	Elasticsearch	Search engine, Multi-model	136.07	-1.68	-5.09
8.	8.	↓ 7.	IBM Db2	Relational, Multi-model	132.41	-2.19	-11.16
9.	↑ 10.	↑ 11.	Snowflake	Relational	125.92	+6.04	+8.66
10.	↓ 9.	↓ 9.	Microsoft Access	Relational	117.67	-4.08	-15.69
11.	11.	↓ 10.	SQLite	Relational	115.20	-2.75	-16.29
12.	12.	12.	Cassandra	Wide column, Multi-model	111.04	-1.16	-5.27
13.	13.	13.	MariaDB	Relational, Multi-model	99.23	-1.19	-0.12

Based on #mentions (e.g., stack overflow), google trends, job postings, profile data on LinkedIn, tweets…

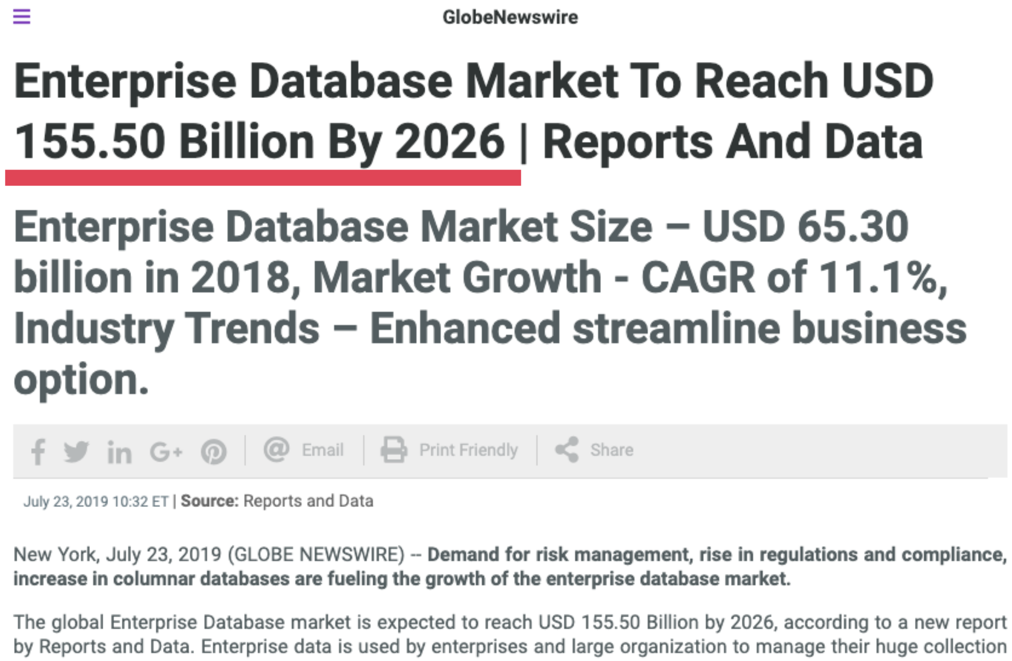
http://db-engines.com/en/ranking

Global Database Market

Huge and growing market



Source: IDC, Bernstein analysis



Enterprise Database Market To Reach USD 155.50 Billion By 2026 | Reports And Data

Enterprise Database Market Size – USD 65.30 billion in 2018, Market Growth - CAGR of 11.1%, Industry Trends – Enhanced streamline business option.

July 23, 2019 10:32 ET | **Source:** Reports and Data

New York, July 23, 2019 (GLOBE NEWSWIRE) -- **Demand for risk management, rise in regulations and compliance, increase in columnar databases are fueling the growth of the enterprise database market.**

The global Enterprise Database market is expected to reach USD 155.50 Billion by 2026, according to a new report by Reports and Data. Enterprise data is used by enterprises and large organization to manage their huge collection

http://www.infoworld.com/article/2916057/open-source-software/open-source-threatens-to-eat-the-database-market.html
https://www.globenewswire.com/news-release/2019/07/23/1886552/0/en/Enterprise-Database-Market-To-Reach-USD-155-50-Billion-By-2026-Reports-And-Data.html

What Is This Course About?

Big ideas in database management systems

- **Principles**: data independence, declarative programming, isolation, consistency
- **Core algorithms**: search, optimisation, evaluation, concurrency
- **System designs**: how to compose components into a technological stack

The heart of scalable computer systems

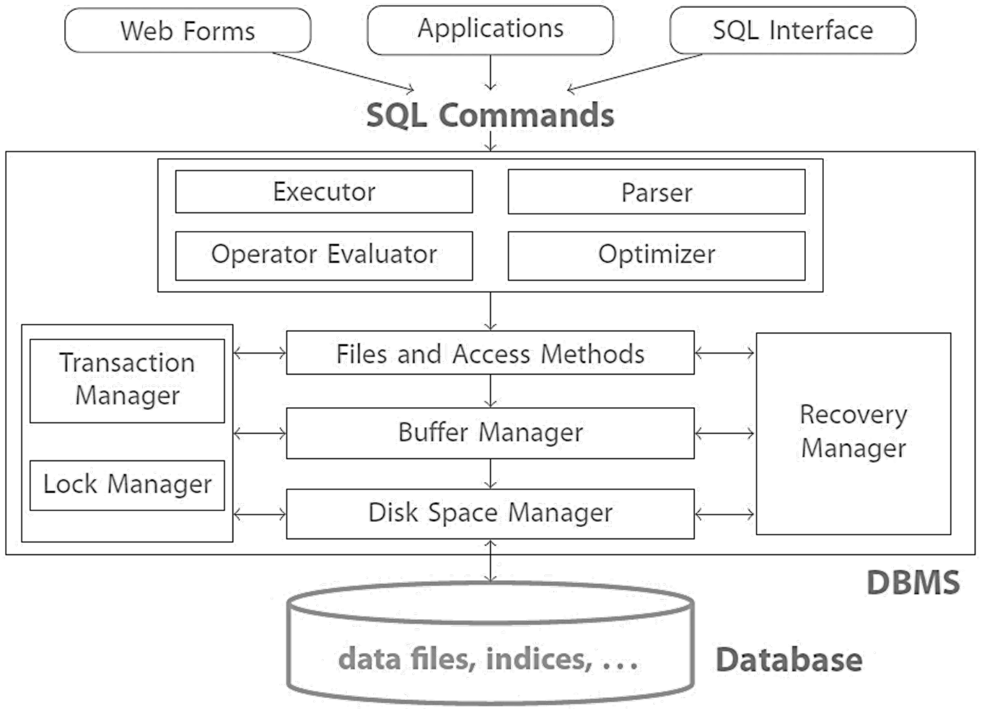
Many of the details and technologies will change in the future

- Be prepared to generalize from what you learn here
- Keep learning new things

What Is This Course About?

Design and implementation of disk-oriented DBMSs

- Storage and file structure
- Indexing techniques
- Query evaluation (theory & practice)
- Query optimisation
- Transaction management
- Distributed and parallel databases



Learning Outcomes

Gain insights into how DBMSs function internally

Learn data management techniques that can help **YOU, the future scientist,** to transform data into knowledge and build new DBMS technologies

Distinguish “hard” vs. “easy” in query evaluation

Learn fundamental concepts used in CS and beyond

Course Prerequisites

Recommended: Introductory course on Databases

- Developing applications using relational DBMSs
- Good knowledge of query languages is a plus ⇐ We will briefly revisit them

Design and analysis of algorithms

- Sorting & searching algorithms, big-O notation

Basic familiarity with complexity

- PTIME, NP-complete

Solid programming skills

- Coursework includes one programming assignment in Java

Essential Questions

Why take this course?

What is this course about?

Who is running this course?

How will this course work?

Who Is Running This Course?

Milos Nikolic

Lecturer, School of Informatics

Interests: database systems, in-database machine learning, stream processing, query compilation

Andreas Pieris

Reader, School of Informatics

Interests: database theory, knowledge-enriched data, knowledge representation and reasoning

How Will This Course Work?

In-person lectures weeks 1-11

- All lectures are live streamed and recorded for later viewing
- Check the course schedule and timetable for more information
- Lectures are followed by **short online quizzes**
- Guest lecture about a popular open-source DBMS in week 11

In-person tutorials weeks 3, 5, 7, 9, 11

- Discussing your answers to tutorial sheets
- To change your tutorial group, use the group change request form

No practical labs this year

Lecture Overview

Block 0: Databases and Query Languages week 1, Milos

- Crash course on SQL and relational algebra
- Covered in an introductory database course

Block 1: Theory of Query Evaluation weeks 2-3, Andreas

- This is not a theory database course…
  - … but understanding the fundamentals is essential for implementation

Block 2: DBMS Internals weeks 4-11, Milos

- How to implement different parts of a database system?
- Important for the coursework assignment

Assessment Structure

Programming assignment (40%)

Implement features in an educational database system in Java

Course engagement (10%)

Weekly online quizzes

Final exam (50%)

In-person exam

School of Informatics uses a Common Marking Scheme

1st class or MSc distinction: 70% and above

Programming Assignment

Involves coding in Java

Requires good programming skills

- Java expertise is not mandatory
- But experience with object-orient programming is expected

Released in week 2

- Some topics covered by then, others covered later
- Allows you to start early & better manage your time

Due: Thursday, 28 March @ 12 noon

Online Quizzes

Short online quizzes released after each lecture

Goals: engage & reinforce the basics

Marking rules

- Quizzes are auto-marked on Learn
- **2 attempts** for each quiz (higher mark counts)
- Each quiz counts equally for engagement
- No deadline per quiz. **Latest submission is Thursday, 4 April @ 12 noon**

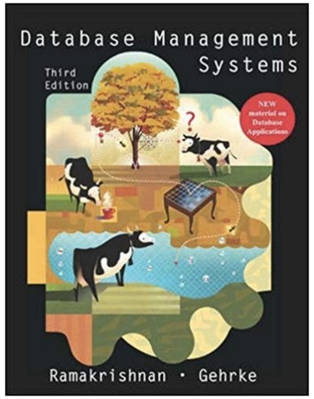
Max engagement mark is 100

Textbooks

Database Management Systems, 3rd edition

Ramakrishnan and Gehrke

- Most lectures will closely follow this book
- Old edition (2003) but still relevant and unbeatable


Database Management Systems
Third Edition
NEW material on Database Applications
Ramakrishnan · Gehrke

Principles of Databases, preprint

Barcelo, Arenas, Libkin, Martins, and Pieris

- Comprehensive material on database theory
- https://github.com/pdm-book/community

Recent Course Changes

Different exam format since 2022/23

- 6-8 smaller questions, all mandatory
- Same format this year

Coursework assignment released early

- Self-assess if your programming skills suffice before the end of week 2

Course content similar to previous years

- This 20-credit course replaces Advanced Databases (INFR11011)
- Content from INFR11011 (e.g., exam questions) still relevant

Plagiarism Policy

All assignments must be your own work

They are **not** group assignments

You may **not** copy source code from other people or the web

You may **not** use public repositories to host your code

We have the technology to detect cheating

See UoE Academic misconduct for more information

Staying in Touch

All class communication via Piazza

Announcements and discussion

- Read it regularly
- Post all questions/comments there
- Answer each other’s questions!

Piazza's Live Q&A for asking questions while watching the live stream

Sign up now on Learn

Acknowledgement

The lecture slides in this course incorporate content from various individuals, to which I am grateful:

D. Olteanu (Zurich)

T. Furche (Oxford)

J. Hellerstein (Berkeley)

A. Pavlo (CMU)

T. Grust (Tübingen)

R. Ramakrishnan (Microsoft)

J. Gehrke (Microsoft)