



# Advanced Database Systems

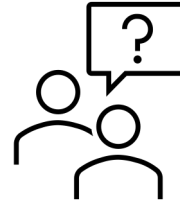
Spring 2024

Lecture #01:

## Course Introduction

1

## ESSENTIAL QUESTIONS



Why take this course?

What is this course about?

Who is running this course?

How will this course work?

2

## 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

3

## WHY? REASON #2: CENTRALITY

Data is at the **centre** of modern society

Much cheaper to generate data

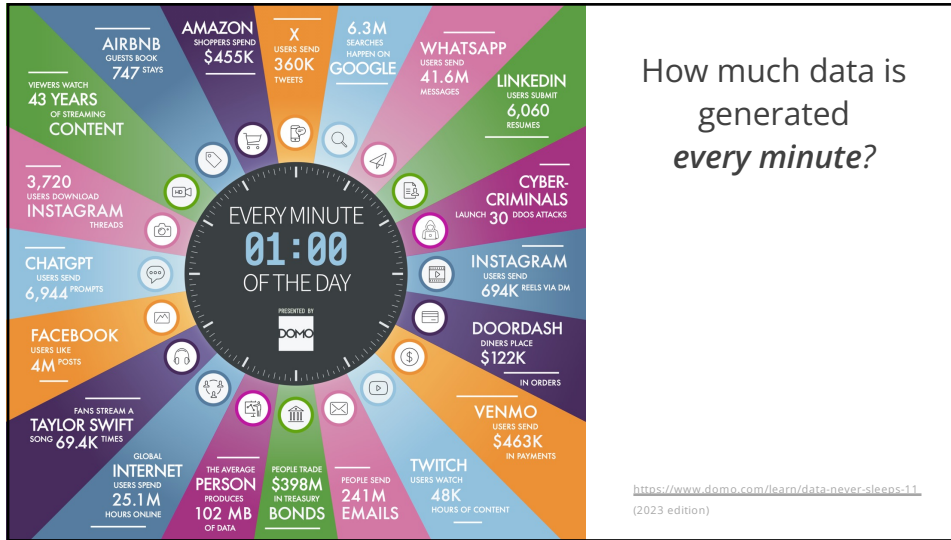
Sensors, smart devices, social networks,  
online games, software logs, audio & video

Much cheaper to process data

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



4



5

## 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>

6

## 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**

7

## WHY? REASON #3: THE CORE OF COMPUTING

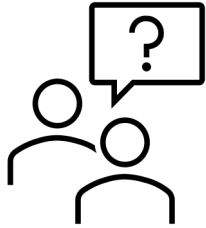
Data growth will continue to outpace computation  
Philosophy: more data → more value?

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

Techniques you learn in this course underlie many topics in computing

8

## ESSENTIAL QUESTIONS



Why take this course?

What is this course about?

Who is running this course?

How will this course work?

9

## 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

10

## 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®



11

## 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

12

## 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

13

## 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

14

## 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

15

## 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  $\geq 0$ ) become "buried" in program code

Hard to add new constraints or change existing ones

16

## 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

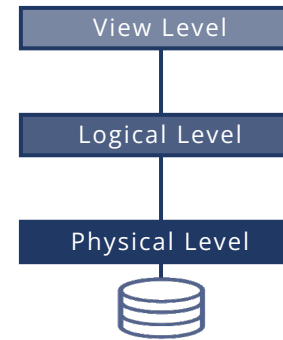
### No application programming interface

- How can a payroll program access the data?

Database systems offer solutions to all the above problems

17

## 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

18

## 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

19

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

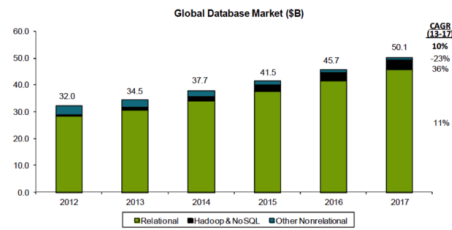
Based on #mentions (e.g., stack overflow), google trends, job postings, profile data on LinkedIn, tweets...

<http://db-engines.com/en/rankings>

21

## 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.

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/2019/07/23/1888597/0/en/Enterprise-Database-Market-to-Reach-USD-155.50-Billion-By-2026-Reports-And-Data.html>

22

## 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

23

## WHAT IS THIS COURSE ABOUT?

**Design and implementation** of disk-oriented DBMSs

Storage and file structure

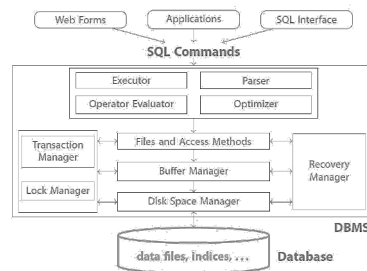
Indexing techniques

Query evaluation (theory & practice)

Query optimisation

Transaction management

Distributed and parallel databases



24

## 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

25

## 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

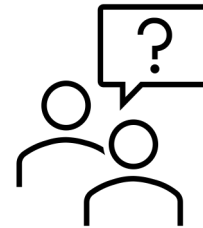
P-TIME, NP-complete

Solid programming skills

Coursework includes one programming assignment in Java

26

## ESSENTIAL QUESTIONS



Why take this course?

What is this course about?

Who is running this course?

How will this course work?

27

## 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



28

## 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

29

## LECTURE OVERVIEW

### Block 0: Databases and Query Languages

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

week 1, Milos

### Block 1: Theory of Query Evaluation

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

weeks 2-3, Andreas

### Block 2: DBMS Internals

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

weeks 4-11, Milos

30

## 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](#)

1<sup>st</sup> class or MSc distinction: 70% and above

31

## PROGRAMMING ASSIGNMENT

Involves coding in Java

Requires good programming skills

Java expertise is not mandatory

But experience with object-oriented 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**

32

## 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**

33



## TEXTBOOKS

**Database Management Systems**, 3<sup>rd</sup> edition  
Ramakrishnan and Gehrke

Most lectures will closely follow this book

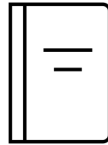
Old edition (2003) but still relevant and unbeatable



**Principles of Databases**, preprint  
Barcelo, Arenas, Libkin, Martins, and Pieris

Comprehensive material on database theory

<https://github.com/pdm-book/community>



34

## 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

35

## 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



36

## 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

37

## 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)