LECTURER: DR. TAI LE QUY

ANALYTICAL SOFTWARE AND FRAMEWORKS

TOPIC OUTLINE

UNIT 6

DATABASE TECHNOLOGY

STUDY GOALS

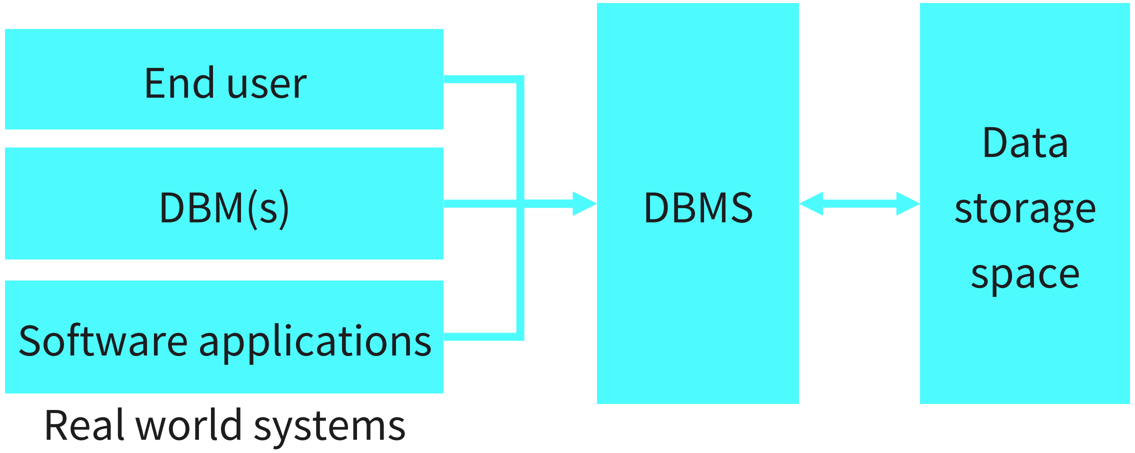
- Know common database approaches
- Know differences between relational and NoSQL-databases
- Know advantages of centralized and distributed databases
- Know differences between on-disk and in-memory databases

EXPLAIN SIMPLY

1. Why are relational databases so common and still dominating the market?
2. What benefits do NoSQL databases promise?
3. What are the advantages and challenges with in-memory databases?

- *A **database** (DB) is a collection of data records representing objects or entities of a domain and consisting of several fields.*
- *A **DBMS** is a software system used to **access, modify, manage, control, and organize a DB**. It is the interface between the DB and other systems or users.*

Database Management System

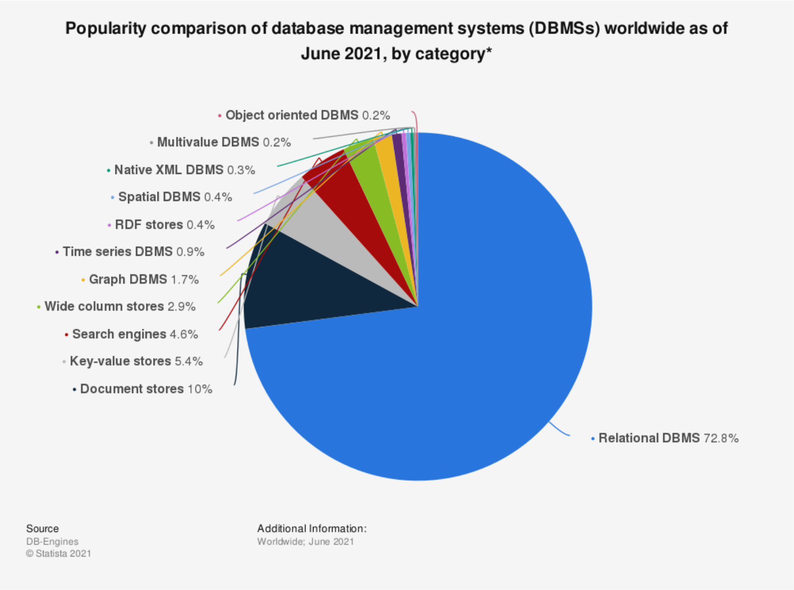


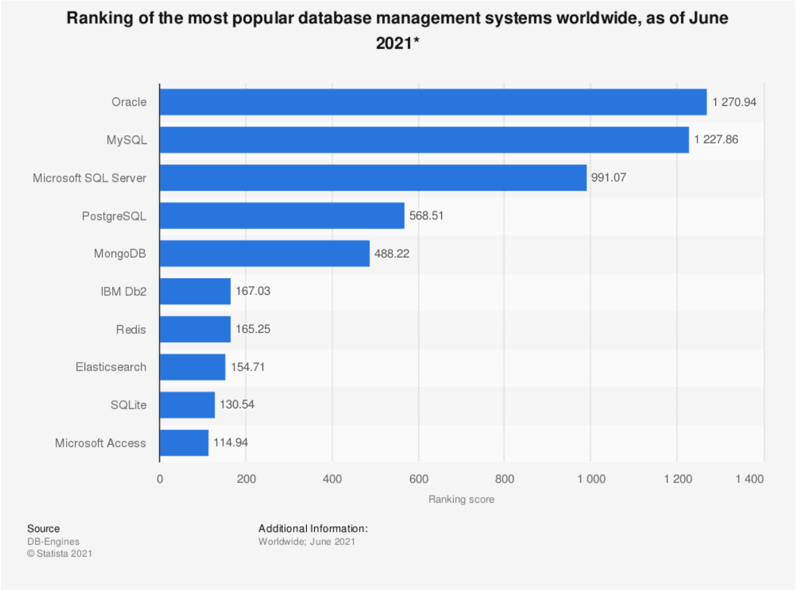
Database operations on data records:

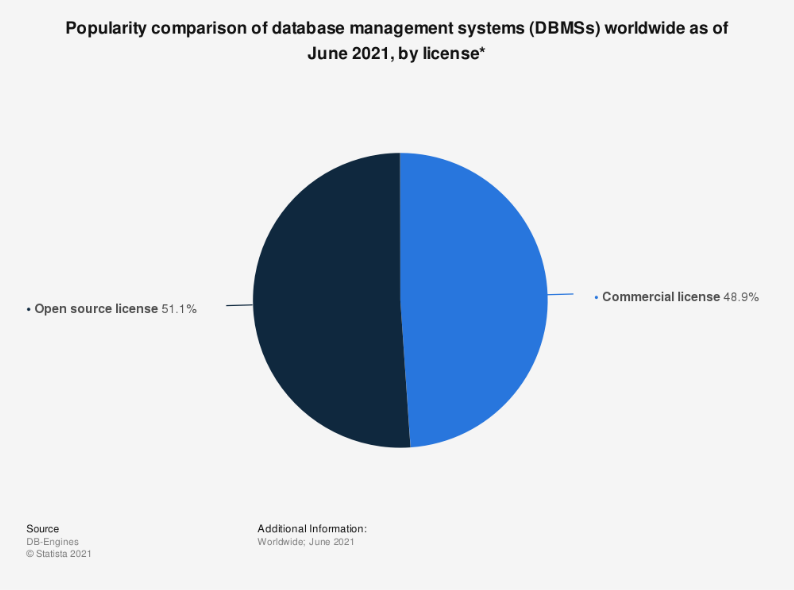
- **C**reate
- **R**ead (query)
- **U**pdate
- **D**elete

Source of the text: Course book DLMBDSA02, p. 140.
Source of the image: Course book DLMBDSA02, p. 140.

TODAY, WE MAY CHOOSE FROM A WIDE VARIETY OF HIGH-QUALITY DATABASE SYSTEMS.



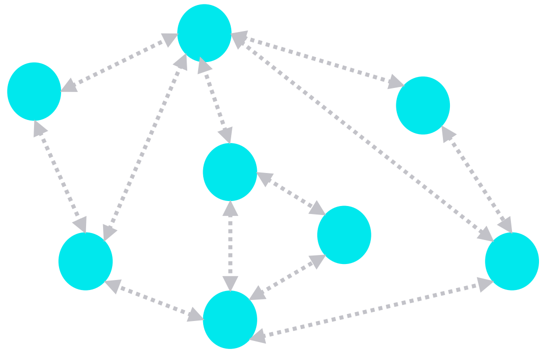




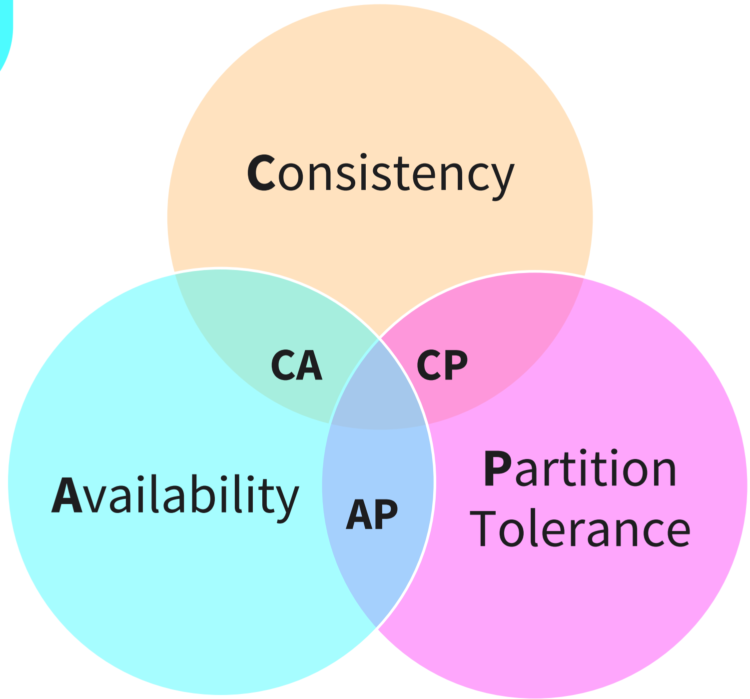
Source of the images: Statista, 2021a; Statista, 2021b; Statista, 2021c.

TODAY'S DATABASES ARE OFTEN DISTRIBUTED SYSTEMS TO MEET REQUIREMENTS OF SCALABILITY AND PERFORMANCE.

*"A distributed system is a **collection of autonomous computing elements** [or 'nodes'] **that appears to its users as a single coherent system**."*



- *As such, distributed DBMSs are designed to be either CA, AP, or CP according to the **CAP theorem**.*



Source of the text: van Steen and Tanenbaum, 2017, p. 16.
Source of the image: Own creation.

ACID STANDS FOR ATOMICITY, CONSISTENCY, INDEPENDENCE, AND DURABILITY. IT VALUES CONSISTENCY OVER AVAILABILITY.

ATOMICITY

Each **operation** is **considered as a single unit (atom)** that failed or succeeded completely. In other words, the operation **cannot fail or succeed partially**. If a part of the operation fails, then the whole operation fails, and the data is kept unchanged.

CONSISTENCY

The operation is applied **only** to **valid data that follows all the rules and constraints of the database**. If the data is not valid, then the data is kept unchanged.

INDEPENDENCE

The **operation** is **processed independently and securely without any interference or interruption by another operation**. The order of processing the operation is not considered.

DURABILITY

This characteristic ensures that the **operation committed to a database is not lost, even in the case of failures.**

Source of the text: Course book DLMBDSA02, p. 141.

BASE VALUES AVAILABILITY, FLEXIBILITY, AND SCALABILITY OVER CONSISTENCY

BASICALLY AVAILABLE
Database appears to work all/most of the time, available for querying by all users. To guarantee availability, the consistency is handled less strictly.

SOFT STATE
Data in the system may be in change due to eventual consistency operations, even if no new transaction hits the database system. Different replicas do not have to be mutually consistent all the time.

EVENTUALLY CONSISTENT
As data gets stored in the database (on *one* node of the database), the new state is gradually replicated across all nodes. Before full replication, the state of the overall database (on *all* nodes) is not **consistent** for a short period of time, but it will be **eventually**.

BASE stands for **B**asically **A**vailable, **S**oft state, **E**ventually consistent.

Source of the text: Course book DLMBDSA02, p. 141f.

RELATIONAL DATABASE

- For **structured data following a defined** database **scheme**
- **Records**/entities are stored uniformly as **rows** in **tables** with a defined group of fields (**columns**).
- **Primary** and **foreign keys** are used to represent **relationships** between entities.

Relation between Two Data Tables in a Relational Database

Students

Name	ID	Grade
John	70	78
Mark	75	87
Emy	78	89
Mike	80	85
Adam	85	70
Heba	87	72

Primary key

Relation

Orders

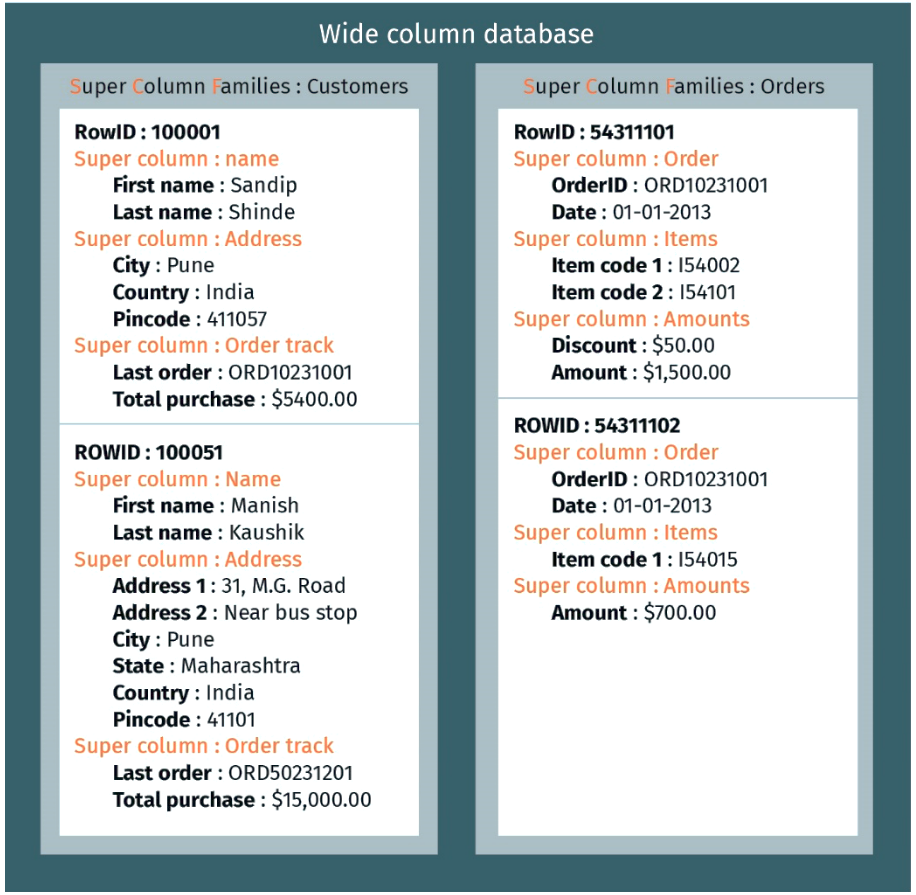
ISBN	Title	S-ID	Date
1	A	78	15-Jan
2	B	87	11-Feb
3	C	78	02-Mar
4	D	85	09-Mar
5	E	70	10-Apr
6	F	70	02-May

Foreign key

Typical: following **ACID model** and **using SQL** (Structured Query Language) for schema definition and DB interaction

Source of the text: Course book DLMBDSA02, p. 126f.
Source of the image: Course book DLMBDSA02, p. 144.

- **NoSQL** stands for **N**ot **o**nly **SQL**.
- Usually designed as **distributed** database and for horizontal **scalability**, following the **BASE** model.
- Allows for **structured**, **unstructured**, **semi-structured**, and **polymorphic** data.
- **Wide range of database technologies**, e.g., **key-value**, **document-oriented**, **wide-column**, **graph**, etc.



Source of the text: Course book DLMBDSA02, pp. 145—148.
Source of the image: Course book DLMBDSA02, p. 147.

Relational DBMS

Still dominating, **highly mature, well known and documented.**

Following well-known and accepted **SQL standards—easy to migrate**.

Less efficient in handling complex semi-structured/**unstructured data**.

Upfront effort for schema definition provides **uniform records for data analysis**.

NoSQL DBMS

Increasingly utilized in large **unstructured** and **distributed** data analytics.

High **flexibility without** upfront effort for **fixed schema**, polymorphic records may be **more challenging at analysis**.

BASE or ACID.

Designed for **availability**, **fault-tolerance**, and **horizontal scalability** with relatively little effort.

Large **range of formats** and **specific constraints** of each DBMS.

Source of the text: Course book DLMBDSA02, pp. 142—148.

Centralized DB

DBMS and DB located in **one single physical (central) location**.

Maximizes data **integrity** and **consistency**. **Minimizes redundancy**.

Only one physical location is **easier to secure but results in higher risk of data**.

Potentially **lower power consumption and maintenance effort**.

Higher risk of latency under high traffic (bottleneck).

Distributed DB

DDBMS and DB **distributed over multiple computing resources** but **appear** to other systems and users **as one database system**.

Replication creates redundancy and **increases availability** and **fault tolerance**—associated with **overhead costs**.

Running queries in parallel boosts **performance**.

Easy **horizontal scaling** by adding additional nodes.

Source of the text: Course book DLMBDSA02, pp. 148—151.

On-disk DB

All data stored on disks (HDD or SDD).

Some data (frequently/last used query results) **cached** in working memory to increase performance.

Due to large amount of low-cost space in virtual storage **database size tends to be unlimited**.

Under high traffic and large data volume **may get not performant enough** due to relatively **slow performance at reading and writing to disks**.

In-memory DB

Keeps entire set of records and all **transactions in** volatile **RAM for highest performance**, **no latency caused by disk access** after startup.

Transactions may be written to **sequential transaction log (append-only) on disk to ensure recoverability**.

Needs warm-up phase reading all data to memory **after start-up and in** case of **recovery**. **Crash may result in data loss** unless persisted in transaction log.

Source of the text: Course book DLMBDSA02, pp. 151—153.

Row-oriented

Data is managed as sequence of **records stored in table rows**.

Efficient for **inserting**, **updating**, **deleting** complete records.

To join tables or reading subsets of columns of records **requires to read the complete records** (supported by indices), which makes it **less efficient in querying large tables**.

Fits well with **OLTP**.

NoSQL DBMS

Data is **organized according to tables' columns**, all values of single columns of single tables are stored together, followed by values of next single columns.

Reading and writing of single columns is much more efficient.

To join or select only subsets of columns, **only affected columns need to be read**, which boosts performance in large tables for that case.

Fits well with **OLAP**.

Source of the text: Course book DLMBDSA02, pp. 153—155.

TIME-SERIES DATABASE, OBJECT-ORIENTED, AND GRAPH-ORIENTED DATABASE

TIME-SERIES DATABASE

Optimized **for time-series (time-stamped) data and calculations on that data**, e.g., min, max, avg with sliding time windows;

much **more efficient than traditional RDBMS in that domain**;

applications in performance monitoring, real-time analytics, IoT applications

OBJECT-ORIENTED DATABASE

Optimized for storing and retrieving objects from object-oriented programming languages without the need of Object-Relational-Mapping **(ORM)**, high consistency between objects in programming language and the database

representing object-oriented design concepts like inheritance and associations between graphs of objects **in the database**

GRAPH-ORIENTED DATABASE

Natively stores and manages interconnected data, designed to **treat relationships between entities (**nodes) as **equally important** to the entities itself;

connections between data records are stored alongside with their own attributes; **accessing nodes** (data records) **and relationships is a very efficient and constant-time operation** (no JOINs required)

Source of the text: Course book DLMBDSA02, pp. 155—157; Neo4j, Inc., 2021a.

- Well-known **cloud service providers** offer **managed versions** of introduced database categories in a **wide technological range** and add their **own proprietary database services** optimized for the cloud and big data.
- **Database vendors** of introduced database categories **offer** their products as **specialized fully managed database services** (**DBaaS**) in the cloud too.

Source of the text: Amazon Web Services, Inc., 2021; EDB, 2021; Google Cloud, n.d.; Microsoft, 2021; MongoDB, Inc., 2021; Neo4j, Inc., 2021b; Redis Ltd., 2021.

REVIEW STUDY GOALS

- Know different database approaches
- Know differences between relational and NoSQL-databases
- Know advantages of centralized and distributed databases
- Know differences between on-disk and in-memory databases

SESSION 6

TRANSFER TASK

Please visit

1. https://aws.amazon.com/products/databases/
2. https://cloud.google.com/products/databases
3. https://azure.microsoft.com/en-us/product-categories/databases/

and investigate the service offerings of the respective cloud service provider in terms of databases.

What kind of database services are offered?

Please visit

1. https://www.influxdata.com/
2. https://neo4j.com/cloud/aura/.
3. https://www.mongodb.com/atlas.
4. https://www.enterprisedb.com/products/biganimal-cloud-postgresql.

and investigate **what kind of database management system** (DBMS) **is presented there! What features are to expect? What advantages and challenges are associated with this?**

TRANSFER TASK
PRESENTATION OF THE RESULTS

Please present your results.
The results will be discussed in plenary.

1. Online transactional processing (OLTP) is a system that provides online database modification. Which database fits the nature of OLTP in applying data transactions?

   a) Row-based database
   b) Column-based database
   c) Distributed database
   d) NoSQL database

2. Which type of database is the best fit for an organization when the data is in the form of documents?

a) Relational
b) Distributed
c) NoSQL
d) Column-based

3. If transaction response speed is crucial for an organization, which type of database is the best fit for this organization?

a) Row-based
b) Column-based
c) Relational
d) In-memory

How did you like the course?

LIST OF SOURCES

Amazon Web Services, Inc. (2021). *Purpose-Built Databases on AWS | Amazon Web Services*. Amazon Web Services, Inc. https://aws.amazon.com/products/databases/

EDB (2021). *BigAnimal—Fully-managed PostgreSQL in the cloud from EDB*. EnterpriseDB. https://www.enterprisedb.com/products/biganimal-cloud-postgresql

Google Cloud (n.d.). *Google Cloud Databases*. Google Cloud. https://cloud.google.com/products/databases

Microsoft (2021). *Azure Databases—Types of Databases on Azure | Microsoft Azure*. https://azure.microsoft.com/en-us/product-categories/databases/

MongoDB, Inc. (2021). *MongoDB Atlas | Multi-cloud application data platform*. https://www.mongodb.com/atlas

Neo4j, Inc. (2021b). Neo4j Aura—Fully Managed Cloud Solution. *Neo4j Graph Database Platform*. https://neo4j.com/cloud/aura/

Neo4j, Inc. (2021a). *What is a Graph Database?* Neo4j Graph Database Platform. https://neo4j.com/developer/graph-database/

Redis Ltd. (2021). *Redis Enterprise Cloud – Fully Managed Cloud Service*. Redis. https://redis.com/redis-enterprise-cloud/overview/

Statista (2021a). *Popularity comparison of database management systems (DBMSs) worldwide as of June 2021, by category.* https://www.statista.com/statistics/1131595/worldwide-popularity-database-management-systems-category/

Statista (2021b). *Ranking of the most popular database management systems worldwide, as of June 2021.* https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems/

Statista (2021c). *Popularity comparison of database management systems (DBMSs) worldwide as of June 2021, by license.* https://www.statista.com/statistics/1131575/worldwide-popularity-database-management-systems-license/

Steen, M. van, & Tanenbaum, A. S. (2017). *Distributed systems* (Third edition, Version 3.01). https://www.distributed-systems.net/index.php/books/ds3/