# NoSQL

## CS460 Databases for Data Scientists

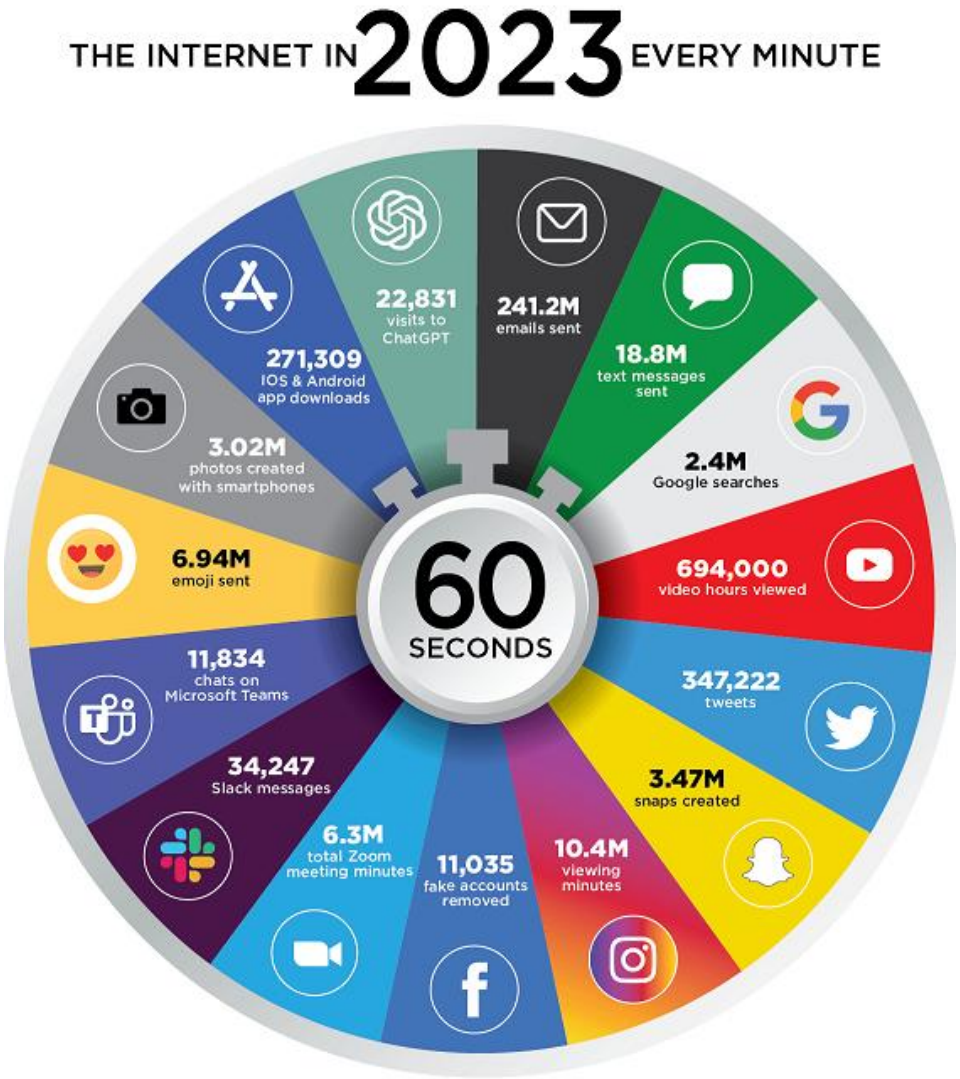
UNIVERSITY OF MANNHEIM
Data and Web Science Group



Slides partially from https://whytin.github.io/cs377_s16/slides/nosql-24.pdf

# Big Data







https://ediscoverytoday.com/2023/04/20/2023-internet-minute-infographic-by-ediscovery-today-and-ltmg-ediscovery-trends/

# Motivation for NoSQL

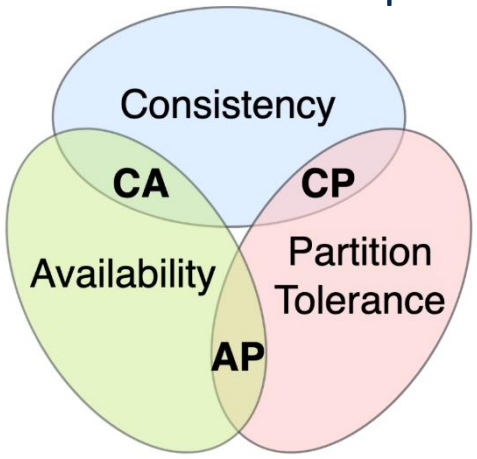
UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Scalability
- RDBMS usually scale with hardware added to one server
  - very expensive when reaching maximum capacity
- On > 50GB data:
  - MySQL
    - Writes 300 ms avg
    - Reads 350 ms avg
  - Cassandra
    - Writes 0.12 ms avg
    - Reads 15 ms avg
- NoSQL stands for
  - No Relational
  - Not only SQL
  - An umbrella term for a class of products that don‘t follow RDBMS principles

# CAP Theorem

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- "Of three properties of shared-data systems —
  data Consistency, system Availability, and tolerance to network Partitions —
  only two can be achieved at any given moment in time" — Brewer, 1999
  - Consistency: all nodes see the same data at the same time
  - Availability: guarantee that every request receives a response about whether it was successful or failed
  - Partition tolerance: system continues to operate despite arbitrary message loss or failure of part of the system





https://commons.wikimedia.org/wiki/File:CAP_Theorem_Venn_Diagram.png

# NoSQL Categories

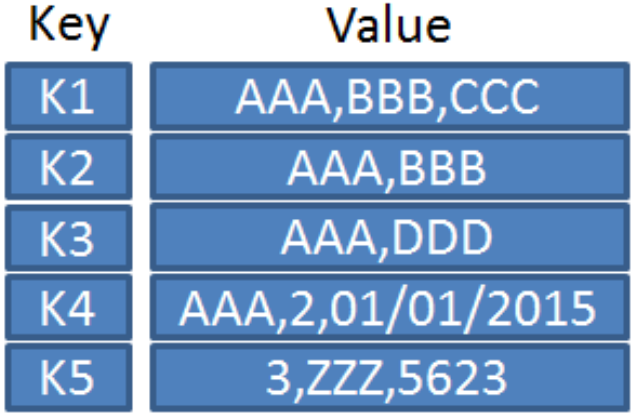
UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Four groups:
  - Key-value stores
  - Column-based families or wide column systems
  - Document stores
  - Graph databases

# Key-value stores

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Simplest NoSQL databases
  - collection of key, value pairs
- Queries are limited to query by key
- Examples:
  - Redis
  - Riak
  - Voldermort
  - DynamoDB
  - MemcacheDB





https://upload.wikimedia.org/wikipedia/commons/5/5b/KeyValue.PNG

# Column-Based Families

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Data is stored in a big table
- Access control, disk storage, and memory accounting performed on column families
- Example:
  - Cassandra
  - Hbase
  - DuckDB
  - Hypertable

| id | user_name | current_balance | number_of_ transactions |
|---|---|---|---|
| 1 | 'taggy@yahoo.com' | 1059298 | 1045 |
| 2 | 'fra@hotmail.com' | 3910 | 194 |
| ... | ... | ... | ... |

Column storage better

```
SELECT sum(current_balance)
            AS total_transactions
FROM table
WHERE user_id > 2
```

Row storage better

```
SELECT user_id, user_name, current_balance
FROM table
WHERE user_id = 1
```

# Document Databases

UNIVERSITY OF MANNHEIM
Data and Web Science Group

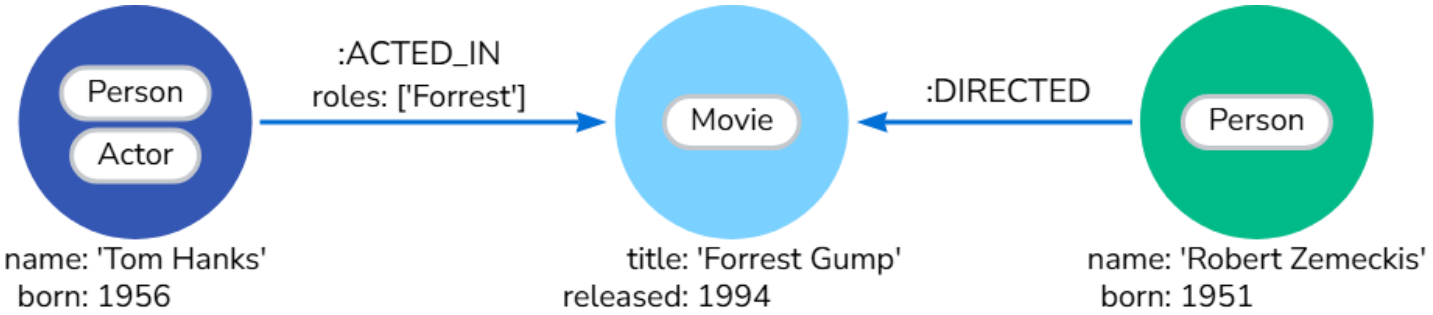
- Collections of similar documents
- Each document can resemble a complex model
- Good for single view or data hub applications
- Examples:
  - MongoDB
  - ArangoDB
  - CouchDB

```
{
   "firstName": "Bob",
   "lastName": "Smith",
   "address": {
      "type": "Home",
      "street1":"5 Oak St.",
      "city": "Boys",
      "state": "AR",
      "zip": "32225",
      "country": "US"
   },
   "hobby": "sailing",
   "phone": {
      "type": "Cell",
      "number": "(555)-123-4567"
   }
}
```

# Graph Database

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Collection of vertices (nodes) and edges (relations) and their properties
- Example:
  - Neo4j
  - Amazon Neptune
  - AllegroGraph
  - VertexDB






https://neo4j.com/docs/getting-started/appendix/graphdb-concepts/

# What's Next?

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Database Systems II (FSS, Moerkotte)
  - e.g., distributed DBMS, object-relational DBs, deductive DBs
- Query Optimization (FSS, Moerkotte)
  - more sophisticated query optimization
- Large-Scale Data Management (HWS, Gemulla)
  - e.g., parallel & distributed databases, MapReduce, SPARQL, NoSQL
- Knowledge Graphs (Hertling)
  - Graph Representation and Inference
  - Knowledge Modeling and Integration

# What's Next?

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Data Security and Privacy (FSS, Armknecht)
  - also covers aspects such as encryption
- Web Data Integration (HWS, Bizer)
  - dealing with multiple databases
  - automatically integrating them into a single one
  - can be accompanied with a practical project
- Data Mining (FSS/HWS, Bizer/Hertling)
  - finding patterns in data
  - entry point to more specific lectures in the data analytics field
  - includes a practical project

# Exam

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Date: 02.06.2025
- Time: 10:00-11:00
- Location: B 243

# Exam

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 6 questions
  - Each question 10 points
- Prepare for similar questions as the exercise
- Focus:
  - Be able to apply your knowledge to a specific database / example at hand

# Exam

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- SQL
  - Be able to write SQL queries (know the syntax)
    - **No** views, no roles
    - **No** temporary relations using WITH
- ER Models
  - Write and understand ER models
  - Be able to reduce a ER model to Relation Schema
- Normalization
  - Be able to normalize a given database

# Exam

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Indexing/Hashing
  - Understand differences between different options
  - Create and modify given indices
    - **Excluding** special cases like „merge siblings“ or „redistribute pointers“ for B+ tree
- Database Architectures
  - General understanding of concepts (no details)

# Exam

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Query Processing
  - Know the different join possibilities
  - Know how indices can be used for selections
- Query Optimization
  - Be able to apply equivalence rules (don‘t need to remember them)
  - Understand execution/evaluation plans
    - Be able to decide which one is better given a concrete example
  - Be able to compute the number of tuples for a given selection

# Exam

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Transactions & Concurrency
  - Understand and Write schedules
  - Know what Conflicts are (detect conflicts)
  - Know the Two-Phase Locking Protocol
  - Know/detect deadlocks
  - **No** Timestamp-based Scheduling or Validation Based Protocol
- Recovery
  - Know and apply log-based recovery mechanism (undo, redo operations)
    - Including checkpoints

# Exam

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Applications
  - Know the security implications
    - SQL Injection
    - Cross Site Scripting
    - Password storage
  - **No** HTML or servlet code etc
- NoSQL
  - Not exam relevant

# Course Evaluation

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- See Ilias

# Q & A

UNIVERSITY OF MANNHEIM
Data and Web Science Group