

Databases

01 Introduction & Course organization

Lucas Iacono. PhD

Graz University of Technology, Know Center Research, Austria

Institute of Human-Centred Computing

About Me



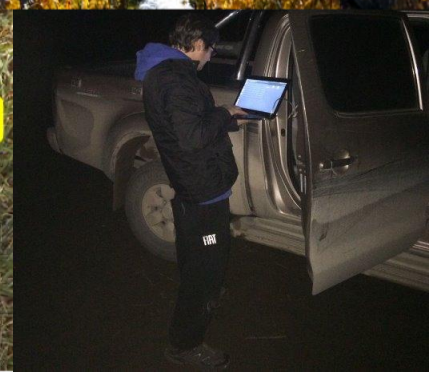
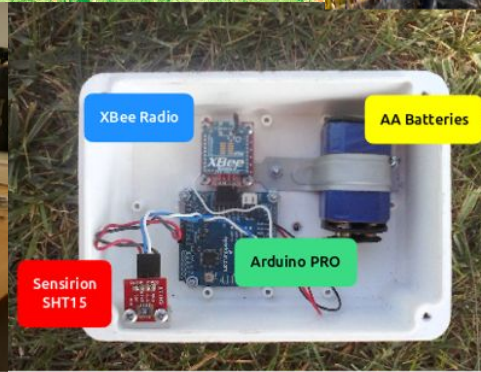
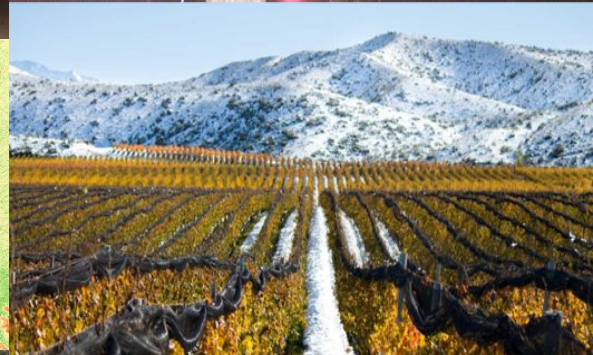
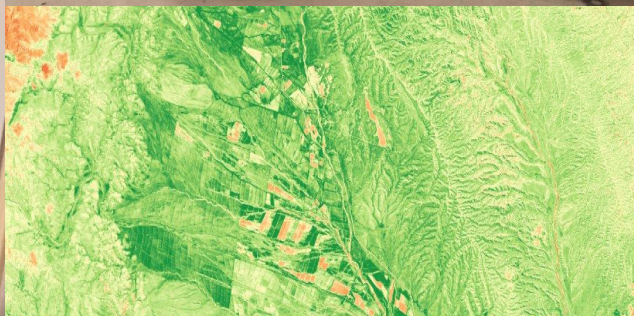
How to contact me?

- liacono@tugraz.at
- [lucasiacono](#)
- SAL Building - 2nd Floor - Office 02 067
- <https://lucasiacono.github.io>



About Me

- **Degree in Electrical and Electronics Engineering** – University of Mendoza, Argentina (2007)
- **Doctor in Engineering (PhD)** - University of Mendoza, Argentina (2015)
- **Data Engineer** – Fiat Petronas PSG16 Race Team (2015 - 2016)
- **Associate Professor** -
 - Introduction to Technology – National University of Cuyo (2015 – 2019) - Argentina
 - Industrial Robotics – Computer Engineering – University of Mendoza (2008 – 2019) - Argentina
 - Postdoc – IoT Devices and UAVs applied to frost damage mitigation - Argentine Research Council (CONICET) - 2017
- **Know Center Research GmbH**
 - Senior Researcher - HAI Area (2019 - 2023)
 - Research Area Manager - Data Management for AI Area (2023 - Now)



Data Management for AI @Know Center

Area Manager



Lucas Iacono

Researchers



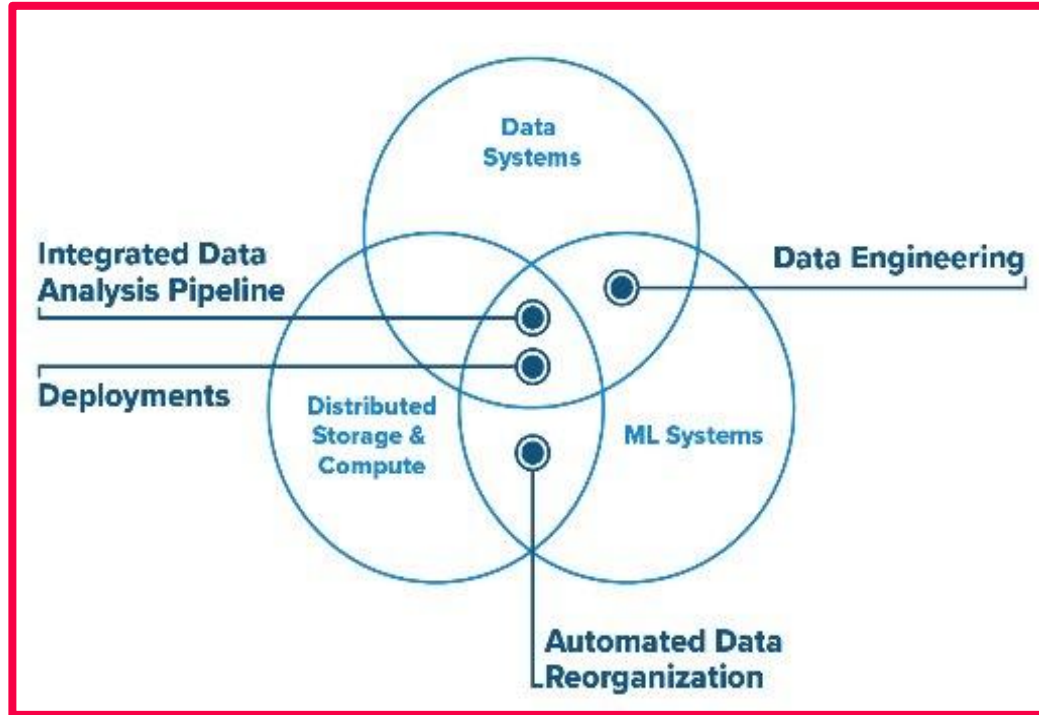
Shafaq Siddiqi - Mark Dokter

Data Scientists



Alexander Hiebl - Lorenz Dirry - Snehila Snehila

Our research



Agenda

- Short introduction to databases
- Course organization

Short introduction to databases

History of data storage and management

- Essential for humans to store and retrieve information/data to make informed decisions
 - Should be as efficient as possible

History of data storage and management

- Essential for humans to store and retrieve information/data to make informed decisions
 - Should be as efficient as possible
 - Categorization, no duplicates, relations, etc ...

History of data storage and management

- Essential for humans to store and retrieve information/data to make informed decisions
 - Should be as efficient as possible
 - Categorization, no duplicates, relations, etc ...

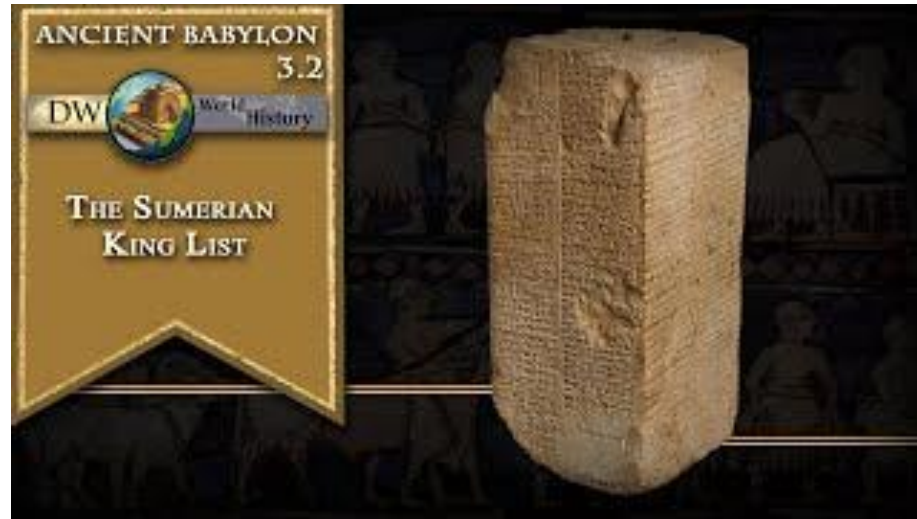
- Manual records of Sumerians are an early example of storing data (4000 BC using clay)



[archive.org]

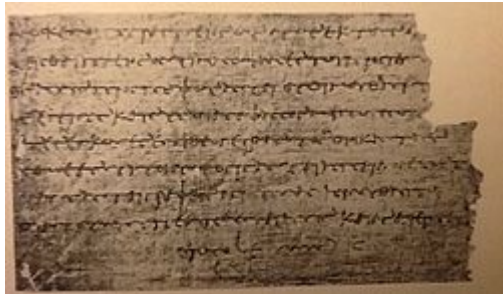
History of data storage and management

- Sumerians: the king list




History of data storage and management

- From clay, we moved to papyrus, parchment to paper



History of data storage and management

- From clay, we moved to papyrus, parchment to paper
- Data management in libraries  Dewey Decimal System




History of data storage and management

- From clay, we moved to papyrus, parchment to paper
- Data management in libraries ➔ Dewey Decimal System
 - Developed around 1876 and still used today in 135 countries
 - Organizes books in libraries based on **subjects**
 - **3-digit numbers** for **main** classes (e.g., 300 is social sciences)
 - **Fractional decimals** for **detailed** classes (e.g., 323 is civil and political rights)




History of data storage and management



- From clay, we moved to papyrus, parchment to paper
- Data management in libraries  Dewey Decimal System
 - Developed around 1876 and still used today in 135 countries
 - Organizes books in libraries based on subjects
 - 3-digit numbers for main classes (e.g., 300 is social sciences)
 - Fractional decimals for detailed classes (e.g., 323 is civil and political rights)
- Punch cards (1900 – 1955)

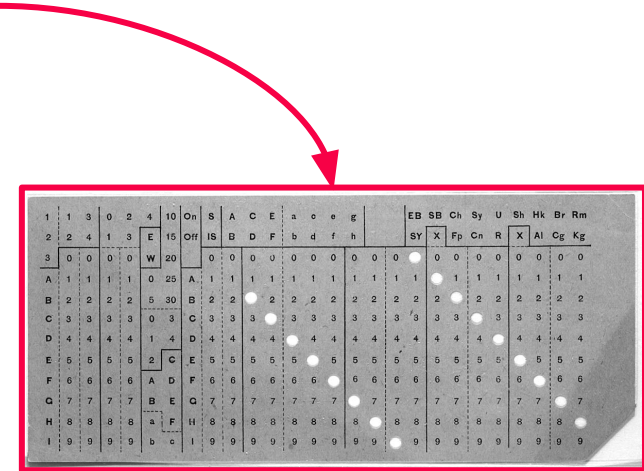


History of data storage and management

- From clay, we moved to papyrus, parchment to paper
- Data management in libraries  Dewey Decimal System
 - Developed around 1876 and still used today in 135 countries
 - Organizes books in libraries based on subjects
 - 3-digit numbers for main classes (e.g., 300 is social sciences)
 - Fractional decimals for detailed classes (e.g., 323 is civil and political rights)
- Punch cards (1900 – 1955)
- 1960 marks the start of computerized databases

Computerized databases

- Flat files
 - Flat  no connection with other files/records
 - E.g. Hollerit's punch cards database
 - Stored as sequential files  retrieval is slow
 - Example: .csv file



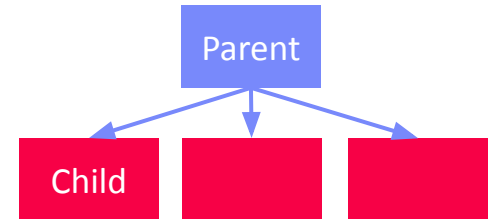
Computerized databases

- Flat files

- Flat → no connection with other files/records
- Stored as sequential files → retrieval is slow
- Example: .csv file

- Hierarchical model

- Tree of records
- E.g., IBM Information Management System
- Each child can only have one parent



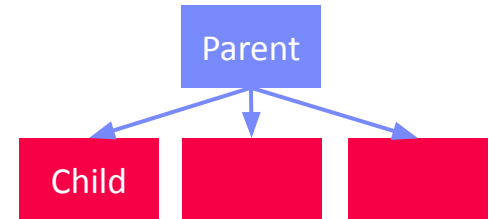
Computerized databases

- Flat files

- Flat → no connection with other files/records
- Stored as sequential files → retrieval is slow
- Example: .csv file

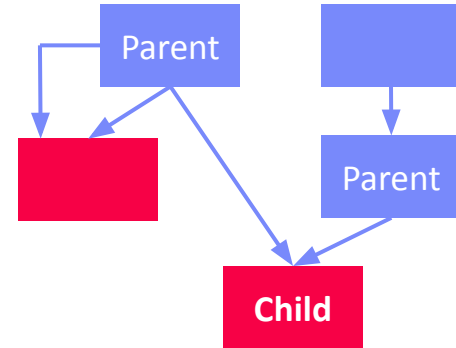
- Hierarchical model

- Tree of records
- E.g., IBM Information Management System
- Each child can only have one parent
- **Issue:** Duplicates!



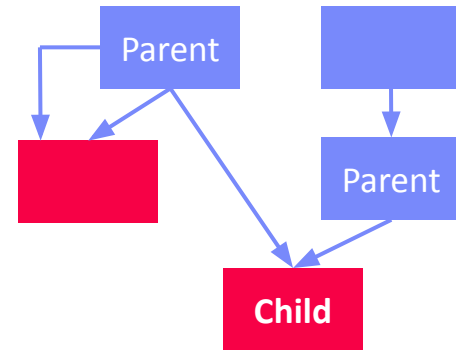
Computerized databases

- Network model
 - Graph of records
 - Developed by Charles Bachmann (Turing Award 1973)
 - Child can have multiple parents



Computerized databases

- Network model
 - Graph of records
 - Developed by Charles Bachmann (Turing Award 1973)
 - Child can have multiple parents
 - **Issue:** Complex array of pointers leads to complex implementations



Relational databases (1970/80s – now)

School Table

ID	Name
S001	University of Technology
S002	University of Applied Science

Student Table

School ID	ID	Name	DOB
S001	UT-1000	Tommy	05/06/1995
S001	UT-1000	Better	16/04/1995
S002	UAS-1000	Linda	02/09/1995
S002	UAS-1000	Jonathan	22/06/1995

Relational databases (1970/80s – now)

School Table

ID	Name
S001	University of Technology
S002	University of Applied Science

Student Table

School ID	ID	Name	DOB
S001	UT-1000	Tommy	05/06/1995
S001	UT-1000	Better	16/04/1995
S002	UAS-1000	Linda	02/09/1995
S002	UAS-1000	Jonathan	22/06/1995

← Tuple

↑ Attribute

Relational databases (1970/80s – now)

School Table

ID	Name
S001	University of Technology
S002	University of Applied Science

Student Table

School ID	ID	Name	DOB
S001	UT-1000	Tommy	05/06/1995
S001	UT-1000	Better	16/04/1995
S002	UAS-1000	Linda	02/09/1995
S002	UAS-1000	Jonathan	22/06/1995

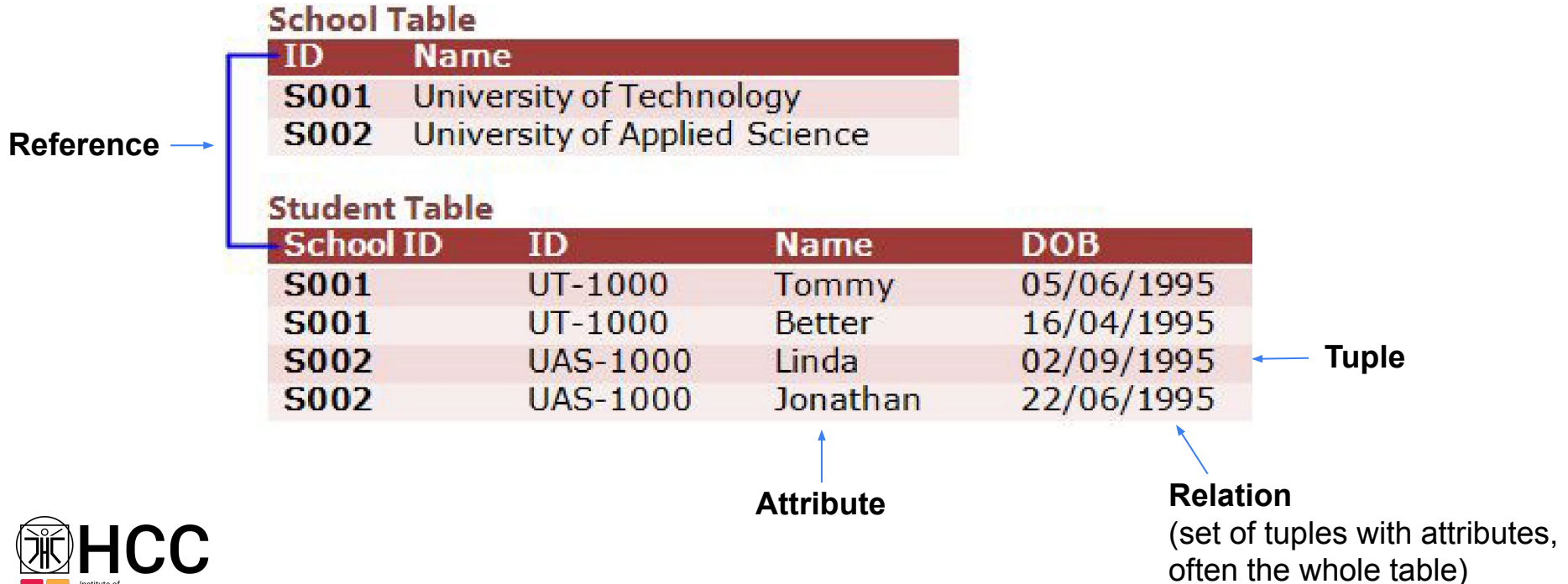
Reference →



← Tuple

↑ Attribute

Relational databases (1970/80s – now)



Relational databases (1970/80s – now)

- Organize a body of data into simple tables of related information
- There are no pointers to maintain (reference)

Relational databases (1970/80s – now)

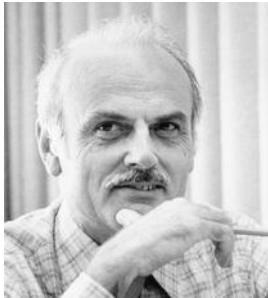
- Organize a body of data into simple tables of related information
- There are no pointers to maintain (reference)
- Tables connected only by having matching data field (keys)
- Easy to access, merge and change data

Relational databases (1970/80s – now)

- Organize a body of data into simple tables of related information
- There are no pointers to maintain (reference)
- Tables connected only by having matching data field (keys)
- Easy to access, merge and change data
- Independence between data scheme and physical storage
- SQL (structured query language)

Relational databases (1970/80s – now)

- Organize a body of data into simple tables of related information
- There are no pointers to maintain (reference)
- Tables connected only by having matching data field (keys)
- Easy to access, merge and change data
- Independence between data scheme and physical storage
- SQL (structured query language)



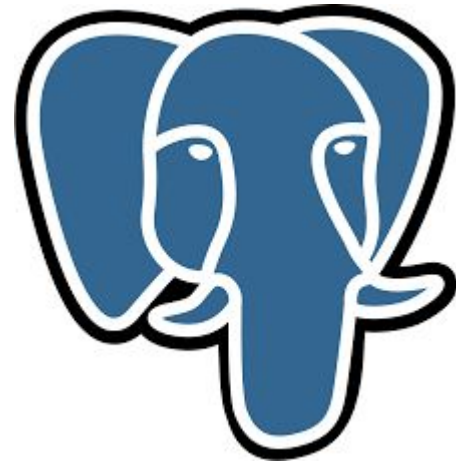
Edgar F. “Ted” Codd @ IBM
Research (**Turing Award ‘81**)

[E. F. Codd: A Relational Model of Data
for Large Shared Data Banks.
Comm. ACM 13(6), **1970**]



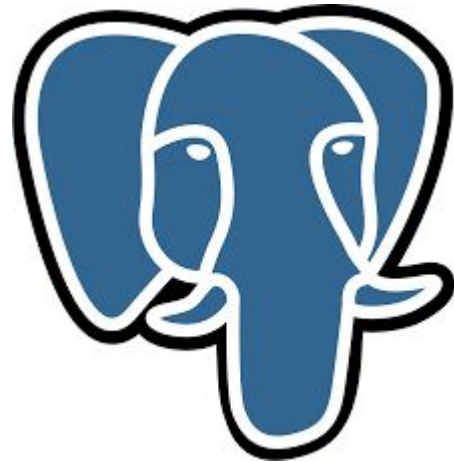
Our relational database: PostgreSQL

- Open-source and free relational database management system
- Easy to install, e.g., (<https://www.postgresql.org/download/>)
 - See also Tutorial pdf in TeachCenter



Our relational database: PostgreSQL

- Open-source and free relational database management system
- Easy to install, e.g., (<https://www.postgresql.org/download/>)
 - See also Tutorial pdf in TeachCenter
- Nice GUI (pgAdmin4)
- Good support and user community
- More on this also in the next lectures

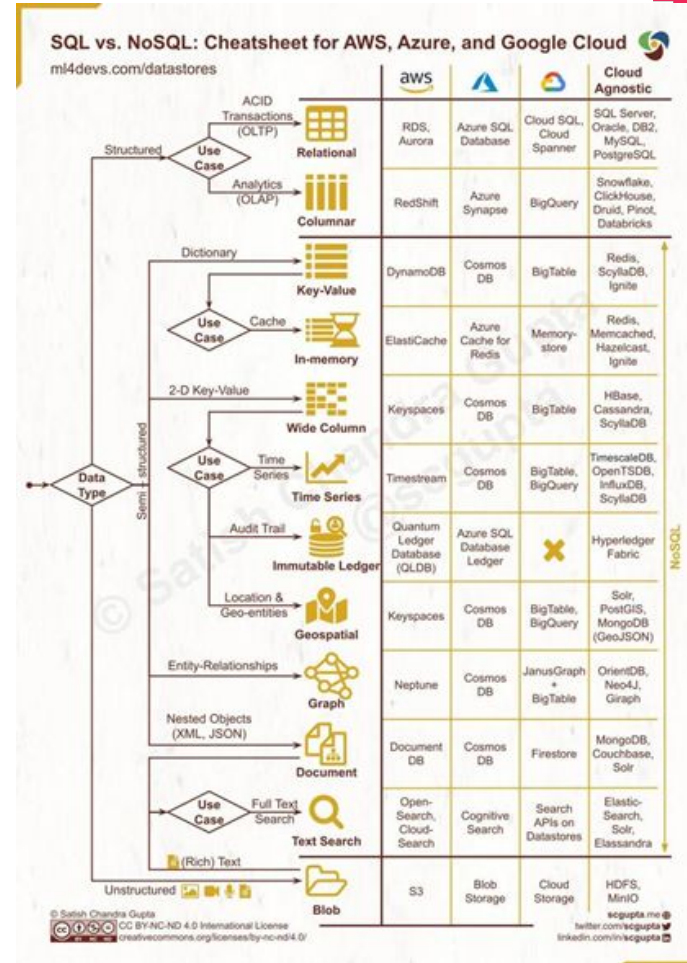


Beyond relational databases (NoSQL)

- Data became bigger (BigData) and in more variety
 - Graphs (nodes/edges/attributes)
 - Time-series (sequences of observations)
 - Key-value stores (simple put/get/delete operations --» high scalability)
 - RDF stores (subject/predicate/object triples --» semantic Web)
 - Document stores (store and search in documents, e.g., Apache Solr)
 - Geo-spatial databases (store and retrieve geo-locations)
- NoSQL

Beyond relational databases (NoSQL)

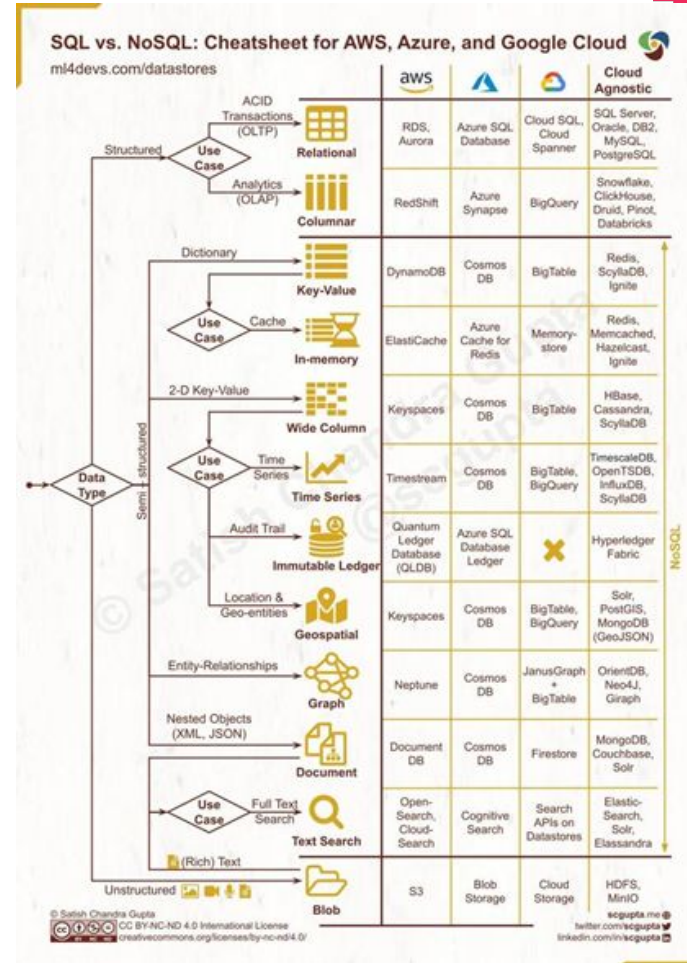
[ml4devs.com]



Beyond relational databases (NoSQL)

- Our course: relational (PostgreSQL)
- Excuse to other types in Data Management & Data Integration and Large-Scale Analysis courses

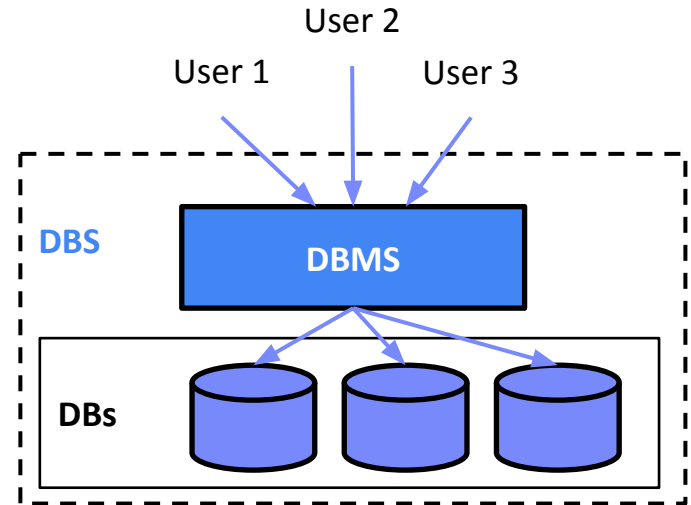
[ml4devs.com]



What is a database?

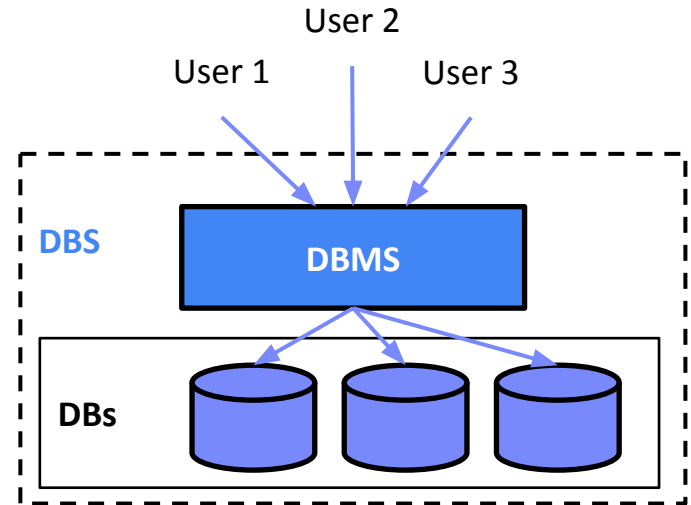
Definition of a database

- Database system (DBS): DBMS + DBs



Definition of a database

- Database system (DBS): DBMS + DBs
- DBMS: Database Management System (Software to handle DBs)
- DBs: Database (data/metadata collection to describe something)
- Note: DB also a short for DBS/DBMS



Why it is important?

- Important for data analytics / research

Why it is important?

- Important for data analytics / research
- Business
 - Market Volume: **10-100B U\$\$**
 - Foundation of many applications in various domains

Why it is important?

- Important for data analytics / research (see Last.fm / recommender systems example at the beginning)
- Business
 - Market Volume: **10-100B \$US**
 - Foundation of many applications in various domains
- Technical
 - Consistency and data integrity (no redundancy)
 - Performance and scalability
 - Multi-user operations and access control
 - Application development and maintenance costs

Readings (optional)

- Raghu Ramakrishnan, Johannes Gehrke: Database Management Systems (3. ed.). McGraw-Hill 2003, ISBN 978-0-07-115110-8, pp. I-XXXII, 1-1065
- Jeffrey D. Ullman, Jennifer Widom: A first course in database systems (2. ed.). Prentice Hall 2002, ISBN 978-0-13-035300-9, pp. I-XVI, 1-511
- Ramez Elmasri, Shamkant B. Navathe: Fundamentals of Database Systems, 3rd Edition. Addison-Wesley-Longman 2000, ISBN 978-0-8053-1755-8, pp. I-XXVII, 1-955
- Alfons Kemper, André Eickler: Datenbanksysteme - Eine Einführung, 10. Auflage. De Gruyter Studium, de Gruyter Oldenbourg 2015, ISBN 978-3-11-044375-2, pp. 1-879

Course organization

General information

- Title: Databases
- Type: VU (means continuous assessment)
- Semester hours: 2 (3 ECTS)
- Offered in: WS

General information

- Title: Databases
- Type: VU (means continuous assessment)
- Semester hours: 2 (3 ECTS)
- Offered in: WS

- Studies:
 - Computational Social Systems (master)
 - Electrical Engineering and Audio Engineering (master)

General information

- Teach Center & Discord:
 - Discord for questions, discussions: <https://discord.gg/3NWBzJTRjh>
 - Teach Center for material: <https://tc.tugraz.at/main/course/view.php?id=3781>
 - **E-mails only for important organizational questions!**
- Language of Instruction/Communication: English
 - Informal language (first name is fine) + ask questions when they appear
 - Please also use English when asking questions
 - Online availability: Mo-Fri 9:00 to 17:00
 - Hello and Thanks when asking questions are welcome!

General information

- Teach Center & Discord:
 - Discord for questions, discussions: <https://discord.gg/3NWBzJTRjh>
 - Teach Center for material: <https://tc.tugraz.at/main/course/view.php?id=3781>
 - **E-mails only for important organizational questions!**
- Language of Instruction/Communication: English
 - Informal language (first name is fine) + ask questions when they appear
 - Please also use English when asking questions
- Teaching and Learning Method:
 - Topics of the course are complemented by **practical exercises using PostgreSQL**

General information

- Teach Center & Discord:
 - Discord for questions, discussions: <https://discord.gg/3NWBzJTRjh>
 - Teach Center for material: <https://tc.tugraz.at/main/course/view.php?id=3781>
 - **E-mails only for important organizational questions!**
- Language of Instruction/Communication: English
 - Informal language (first name is fine) + ask questions when they appear
 - Please also use English when asking questions
- Teaching and Learning Method:
 - Topics of the course are complemented by **practical exercises using PostgreSQL**
- Objectives:
 - With successful completion of this course, students have gained a **basic understanding of**

Content

- This course covers, primarily from a user perspective, foundations of databases and data handling in terms of:

Content

- This course covers, primarily from a user perspective, foundations of databases and data handling in terms of:
 - Overview of databases and data storage systems

Content

- This course covers, primarily from a user perspective, foundations of databases and data handling in terms of:
 - Overview of databases and data storage systems
 - Data modeling
 - Relational databases and normalization

Content

- This course covers, primarily from a user perspective, foundations of databases and data handling in terms of:
 - Overview of databases and data storage systems
 - Data modeling
 - Relational databases and normalization
 - Query languages (SQL)

Content

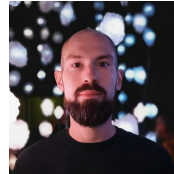
- This course covers, primarily from a user perspective, foundations of databases and data handling in terms of:
 - Overview of databases and data storage systems
 - Data modeling
 - Relational databases and normalization
 - Query languages (SQL)
 - Data handling and manipulation with Python (Pandas)
- + 2 individual exercises
- + group project

Content

- Not part of this course
 - Relational algebra (formal definitions)
 - Physical design (internal workings)
 - Query processing (optimization)
 - Transaction processing (concurrency)
 - Beyond-relational databases (NoSQL)
- See **Data Management, Data Integration and Large-Scale Analysis courses**
 - <https://tc.tugraz.at/main/course/view.php?id=1998>
 - https://online.tugraz.at/tug_online/wbLv.wbShowLVDetail?pStpSpNr=579568
 -

Course team

- Lecturer: Lucas Iacono
- Advisor Lecturer: Dominik Kowald
- Group Projects (Oct): Peter Müllner
- Student assistant (tutor): Margarita Dubina



Assessment

- 2 exercises (50 points) --> **individual** & 60% of the final grade
 - Exercise 1: Data modeling (25 points)
 - Exercise 2: Data ingestion and queries (25 points)

Assessment

- 2 exercises (50 points)
 - Exercise 1: Data modeling (25 points)
 - Exercise 2: Data ingestion and queries (25 points)
- Course project (30 points) --> **groups of 5**
 - Concept (10 points)
 - Objective 1: Research problem, motivation and methodology (10 points)
 - Full project (20 points)
 - Objective 2: Data modeling and data ingestion (5 points)
 - Objective 3: Database queries and data handling (5 points)
 - Objective 4: Presentation and discussion of results (5 points)
 - Objective 5: Reproducibility of results (5 points)
 - **Assign yourselves to groups via TeachCenter**

Assessment

- Grading scheme (> **40 points** needed for passing the course)
 - 0-40 points: 5
 - 41-50 points: 4
 - 51-60 points: 3
 - 61-70 points: 2
 - 71-80 points: 1

Course project (30 points)

- **Objective 1: Research problem and motivation (10 points)**
 - Choose a dataset, describe the **research problem** that you want to solve with it and why it is relevant (for you)
 - A list of potential dataset repositories is provided but you are also free to use an own **dataset from your field** / or just a dataset that you like

Course project (30 points)

- **Objective 1: Research problem and motivation (10 points)**
 - Choose a dataset, describe the **research problem** that you want to solve with it and why it is relevant (for you)
 - A list of potential dataset repositories is provided but you are also free to use an own **dataset from your field** / or just a dataset that you like
 - **Example: DTM dataset**
 - **Research problem/question:** how will increase the participation in the german automotive business of a car manufacturer if this company wins a DTM championship? This means that ...
 - **Relevance:** our research interest lies in the importance of promoting companies in sports to increase business participation, since ...
 - **Method:** a relational database is perfect for this task since it allows to save the relationship between participation of companies in the german automotive business and winnings in DTM drivers championship, and ...

Course project (30 points)

- **Objective 1: Research problem and motivation (10 points)**
 - This concept is the **first submission**
 - Important is the research problem / question you want to answer and why you use a relational database (e.g., multiple tables with connections)
 - You can also be creative
 - **You will get feedback on this, which you need to integrate for the final submission of the project**

Course project (30 points)

- **Objective 2: Data modeling and data ingestion (5 points)**
 - Describe your dataset (e.g., ER diagram), how you have transferred the dataset to a database scheme and how you ingested the data (**similar to Ex. 1 + first part of Ex. 2**)

Course project (30 points)

- **Objective 2: Data modeling and data ingestion (5 points)**
 - Describe your dataset (e.g., ER diagram), how you have transferred the dataset to a database scheme and how you ingested the data (**similar to Ex. 1 + first part of Ex. 2**)
- **Objective 3: Database queries and data handling (5 points)**
 - Describe your database queries (should contribute to your **research problem**) and any data handling you performed (**similar to second part of Ex. 2**)

Course project (30 points)

- **Objective 4: Presentation and discussion of results (5 points)**
 - For presenting/interpreting/discussing the results of your queries, you can use any library that you want (**Lecture 6 provides some examples, e.g., Pandas, Seaborn**)
 - You should interpret (process/analyze/visualize) the queried data

Course project (30 points)

- **Objective 4:** Presentation and discussion of results (5 points)
 - For presenting/interpreting/discussing the results of your queries, you can use any library that you want (**Lecture 6 provides some examples, e.g., Pandas, Seaborn**)
 - You should interpret (process/analyze/visualize) the queried data
- **Objective 5:** Reproducibility aspects of your project (5 points)
 - What is needed to reproduce your project? Where is data? Where is the code? Other additional documentation?
 - **We should be able to reproduce all your project results!**
 - A GitHub repository could be good idea to fulfill this part

Course project (30 points)

- **Objective 4:** Presentation and discussion of results (5 points)
 - For presenting/interpreting/discussing the results of your queries, you can use any library that you want (**Lecture 6 provides some examples, e.g., Pandas, Seaborn**)
 - You should interpret (process/analyze/visualize) the queried data
- **Objective 5:** Reproducibility aspects of your project (5 points)
 - What is needed to reproduce your project? Where is data? Where is the code? Other additional documentation?
 - **We should be able to reproduce all your project results!**
 - A GitHub repository could be good idea to fulfill this part

Objectives 2 – 5 form the second submission to address your research problem

Course project (30 points)

- Format (see also pdf in TeachCenter)
 - Powerpoint / latex slides --> submitted as **.pdf**
 - Structured according to the 5 objectives
 - **6 slides (exactly!)**
 - 1 per objective
 - + 1 title slide at the beginning describing the roles in the group (who has done what?)

Course project (30 points)

- Format (see also pdf in TeachCenter)
 - Powerpoint / latex slides --> submitted as **.pdf**
 - Structured according to the 5 objectives
 - **6 slides (exactly!)**
 - 1 per objective
 - + 1 title slide at the beginning describing the roles in the group (who has done what?)
- Discussion
 - We will discuss selected projects together at the end of the course
 - You will be notified if this is needed

Potential Dataset Sources (just examples)

- Recommender Systems & Personalization Datasets (UC-San Diego)
 - <https://cseweb.ucsd.edu/~jmcauley/datasets.html>
- Stanford Network Analysis Project (SNAP)
 - <https://snap.stanford.edu/data/index.html>
- Data Science Challenges (Kaggle)
 - <https://www.kaggle.com/datasets>
- Open Machine Learning Platform (OpenML)
 - <https://www.openml.org/search?type=data&sort=runs&status=active>
- Open Data Platform Austria (data.gv.at)
 - <https://www.data.gv.at/suche/?typeFilter%5B0%5D=dataset>
- Dataset Search Engine (Google)
 - <https://datasetsearch.research.google.com/>
- GroupLens Research (MovieLens etc.)
 - <https://grouplens.org/datasets/>

Exercises (50 points)

- **Gift cards review dataset**

- This dataset contains Amazon gift card reviews collected in 2023.
- Datasets are shared for research and teaching purposes
- Original dataset: <https://www.kaggle.com/datasets/qwaazs/amazon-gift-and-card-reviews>
- **Cleaned dataset will be provided for Exercise 2**



Exercises (50 points)

- **Exercise 1: Data Modeling (25 points)**
 - Entity relationship (ER) modeling (15 points)
 - Resulting diagram needs to capture the main information of the dataset
 - Mapping ER diagrams into relational model (10 points)
 - Relationships, keys and data attributes need to match the dataset

Exercises (50 points)

- **Exercise 2: Data Ingestion and queries (25 points)**
 - Database and schema creation, and data ingestion (9 points)
 - Database should be complete
 - SQL query processing
 - Six SQL SELECT queries to answer questions on the data (16 points)

Course calendar

- **Part 1: Data Modeling and relational databases**
 - (1) Mon. 20.10.2025: Overview of databases and data storage systems + **Group Project**
 - (2) Mon. 27.10.2025: Data modeling + **Exercise 1 presentation**
 - (3) Mon. 03.11.2025: Relational databases and normalization (+ bit of SQL for Ex. 1)
 - (4) Mon. 10.11.2025: Query languages (SQL) + **Exercise 2 presentation**
 - Mon. 17.11.2025: no lecture (time for exercise 1)

Exercise 1 submission: Tue. 18.11.2025

- Mon. 24.11.2025: no lecture (time for project concept)

Course calendar

- **Part 2: Data handling and project presentations**

Course project concept submission: Tue. 25.11.2025

- (5) Mon. 01.12.2025: Data handling and manipulation with Python (Pandas)
- Mon. 08.12.2025: public holiday (**means no lecture – time for exercise 2**)

Exercise 2 submission: Fri. 12.12.2025

- Mon. 15.12.2024: No lecture (**time for group project full submission**)

Project submission: Fri. 19.12.2025

- (6) Mon. 12.01.2026: project discussions of selected groups (if needed)

Course calendar – Exercises and Projects Summary

- Exercise 1
 - Mon. 27.10.2025 – Tue. 18.11.2025
- Course Project Concept (in parallel to the full project submission)
 - Wed. 22.10.2025 – Fri. 25.11.2025
- Exercise 2
 - Wed. 19.11.2025 – Fri. 12.12.2025
- Course Project Submission
 - Wed. 22.10.2025 – Fri. 19.12.2025
- Course Project Discussions (for selected groups if needed)
 - Mon. 12.01.2026

Conclusions

- Summary
 - Short introduction to databases
 - DB system = DBMS + DBs
 - Course organization
 - 2 individual exercises and 1 group project
 - You could already start assigning yourself to a group in TeachCenter
- Next week
 - Data modeling + **Exercise 1**
- Questions / comments?