OTH Amberg-Weiden

OTH Amberg-Weiden

MODERN DATABASES AND NOSQL

Prof. Dr.-Ing. Michael Wiehl

Infos on the module

OTH Amberg-Weiden

- 5 ECTS
- 4 SWS (2 + 2)

- Work load
  - Contact time: 60 h
  - Self study: 90 h = approx 5-6h per week

Evaluation of learning success

OTH Amberg-Weiden

- Project for design and implementation of a database management system of a selected application
- Result: Project report (10 pages)

Moodle course

OTH Amberg-Weiden

- Slides of the lectures
- References to online learning material
- Examples
- Videos

Learnings

OTH Amberg-Weiden

- know the **basics of relational database systems** and can understand and compare them with other forms of data organization
- name examples of the use of relational database systems and list the possibilities of linking databases to application programs
- know the syntax of a common access language and can apply it
- learn about **distributed data models** as well as platforms and frameworks for distributed data, such as NoSQL databases

Learnings

OTH Amberg-Weiden

- be able to independently design, create, and query databases
- Students refine their knowledge of modern databases, including distributed data models
- By designing and building complex infrastructures, students deepen their ability to abstract
- Students learn a confident approach to modern database applications and infrastructures.

- Students will be able to model, discuss, and present modern databases to a larger audience in small groups
- Through independent learning, students will acquire time management skills.

Course structure

OTH Amberg-Weiden

Prof. Wiehl – Classic DBs / Relational Databases

- Design of databases systems
- Modelling
- Designing a database
- Normalisation
- Quering SQL Databases

Prof. Pirkl – Modern Databases / NoSQL

- Scalability
- Key value stores
- Comparison of DBs
- Time series & Elastic search
- Distributed Systems

Literature

OTH Amberg-Weiden

- „Advanced Data Management“, Lena Wiese, De Gruyter Graduate, 2015
  - Chapter 1: Entity Relationship Model, UML
  - Chapter 2: Relational DBMS, Mapping ER, Normalization, Transactions

- „Getting Started with SQL and Databases“, Mark Simon, apress, 2023
  - Querying and Working with SQL databases, joining tables, sorting

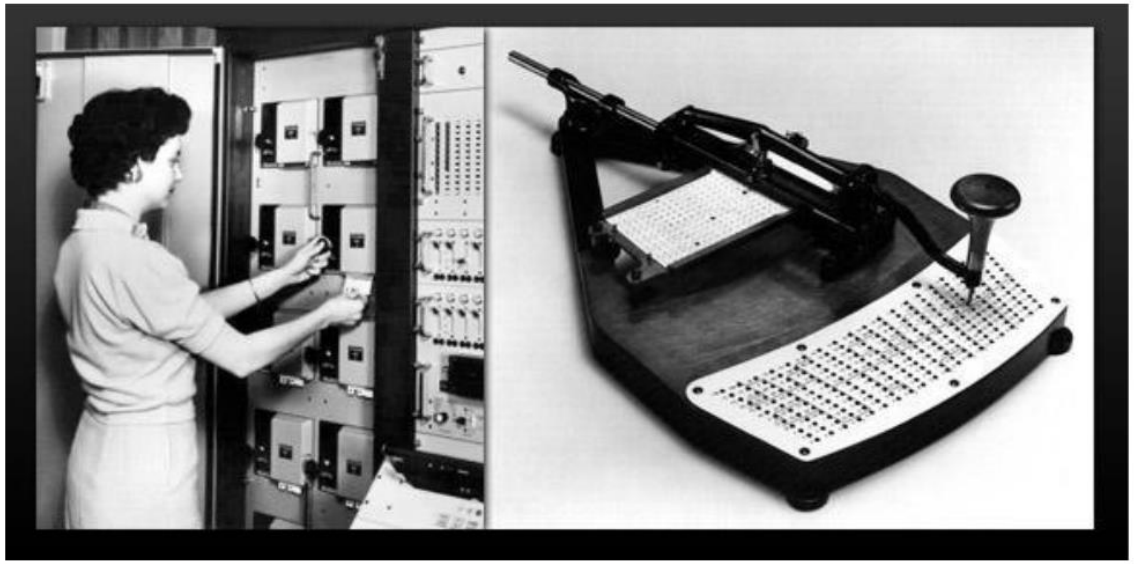
Your knowledge? – Stand up and sit down

OTH Amberg-Weiden

- You know what a database?
- Working with databases?
- You queried a database?
- You have built a new database by your own?

History of databases

- 1950s and early 1960s:
  - Data processing using magnetic tapes for storage
  - Tapes provided only sequential access
  - Punched cards for input



Punched card reader (L) and writer ® | Image from A Brief History of Communication Technology.

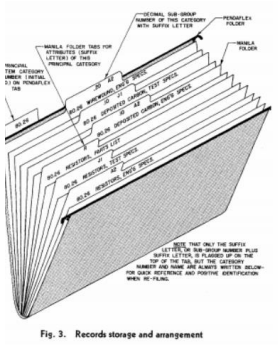


Fig. 3. Records storage and arrangement

- Data in a file system

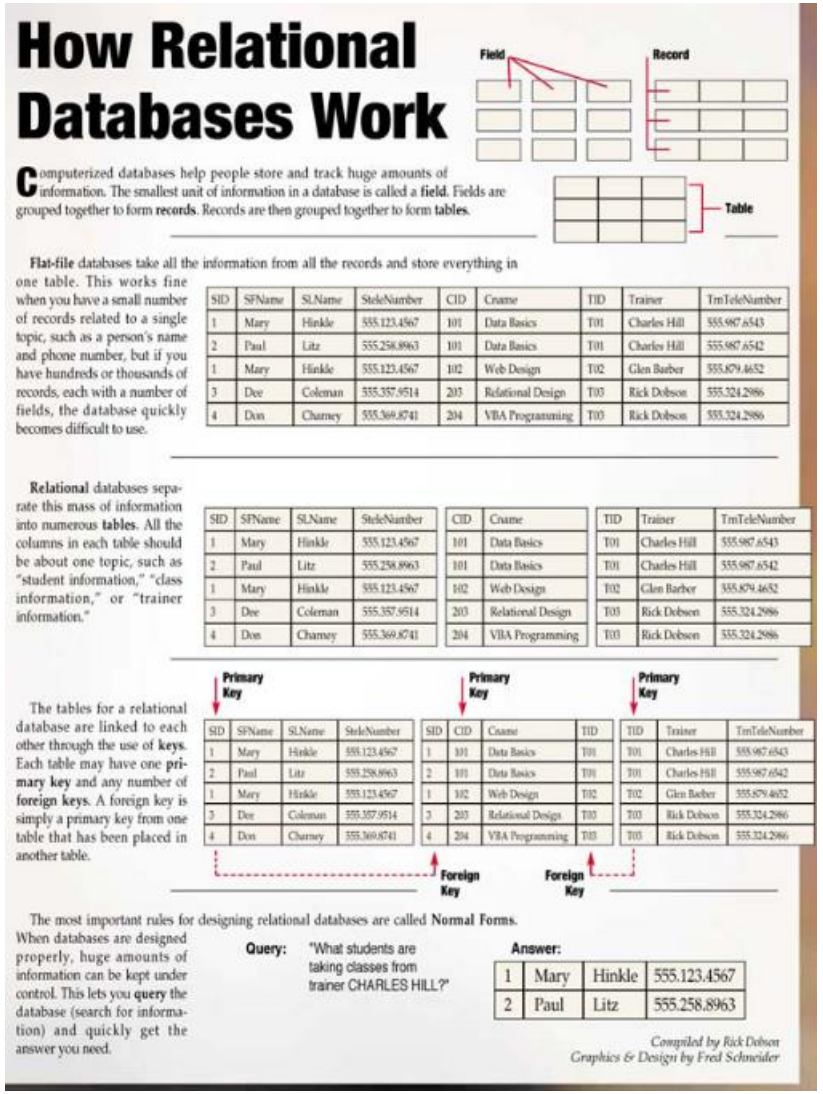
Make your own:

http://www.kloth.net/services/cardpunch.php

History of databases

OTH Amberg-Weiden

- Late 1960s and 1970s:
  - Hard disks allowed direct access to data
  - Network and hierarchical data models in widespread use
  - Ted Codd defines the relational data model
    - IBM Research begins System R prototype
    - UC Berkeley (Michael Stonebraker) begins Ingres prototype
    - Oracle releases first commercial relational database
  - High-performance (for the era) transaction processing



How Relational Databases Work

Computerized databases help people store and track huge amounts of information. The smallest unit of information in a database is called a **field**. Fields are grouped together to form **records**. Records are then grouped together to form **tables**.

Flat-file databases take all the information from all the records and store everything in one table. This works fine when you have a small number of records related to a single topic, such as a person's name and phone number, but if you have hundreds or thousands of records, each with a number of fields, the database quickly becomes difficult to use.

SID	SFName	SLName	SteleNumber	CID	Cname	TID	Trainer	TrnTeleNumber
1	Mary	Hinkle	555.123.4567	101	Data Basics	T01	Charles Hill	555.987.6543
2	Paul	Litz	555.258.8963	101	Data Basics	T01	Charles Hill	555.987.6542
1	Mary	Hinkle	555.123.4567	102	Web Design	T02	Glen Barber	555.879.4652
3	Dee	Coleman	555.357.9514	203	Relational Design	T03	Rick Dobson	555.324.2986
4	Don	Charney	555.369.8741	204	VBA Programming	T03	Rick Dobson	555.324.2986

Relational databases separate this mass of information into numerous **tables**. All the columns in each table should be about one topic, such as "student information," "class information," or "trainer information."

SID	SFName	SLName	SteleNumber
1	Mary	Hinkle	555.123.4567
2	Paul	Litz	555.258.8963
1	Mary	Hinkle	555.123.4567
3	Dee	Coleman	555.357.9514
4	Don	Charney	555.369.8741

CID	Cname
101	Data Basics
101	Data Basics
102	Web Design
203	Relational Design
204	VBA Programming

TID	Trainer	TrnTeleNumber
T01	Charles Hill	555.987.6543
T01	Charles Hill	555.987.6542
T02	Glen Barber	555.879.4652
T03	Rick Dobson	555.324.2986
T03	Rick Dobson	555.324.2986

The tables for a relational database are linked to each other through the use of **keys**. Each table may have one **primary key** and any number of **foreign keys**. A foreign key is simply a primary key from one table that has been placed in another table.

Primary Key — Primary Key — Primary Key

SID	SFName	SLName	SteleNumber
1	Mary	Hinkle	555.123.4567
2	Paul	Litz	555.258.8963
1	Mary	Hinkle	555.123.4567
3	Dee	Coleman	555.357.9514
4	Don	Charney	555.369.8741

SID	CID	Cname	TID
1	101	Data Basics	T01
2	101	Data Basics	T01
1	102	Web Design	T02
3	203	Relational Design	T03
4	204	VBA Programming	T03

TID	Trainer	TrnTeleNumber
T01	Charles Hill	555.987.6543
T01	Charles Hill	555.987.6542
T02	Glen Barber	555.879.4652
T03	Rick Dobson	555.324.2986
T03	Rick Dobson	555.324.2986

Foreign Key — Foreign Key

The most important rules for designing relational databases are called **Normal Forms**. When databases are designed properly, huge amounts of information can be kept under control. This lets you **query** the database (search for information) and quickly get the answer you need.

Query: "What students are taking classes from trainer CHARLES HILL?"

Answer:

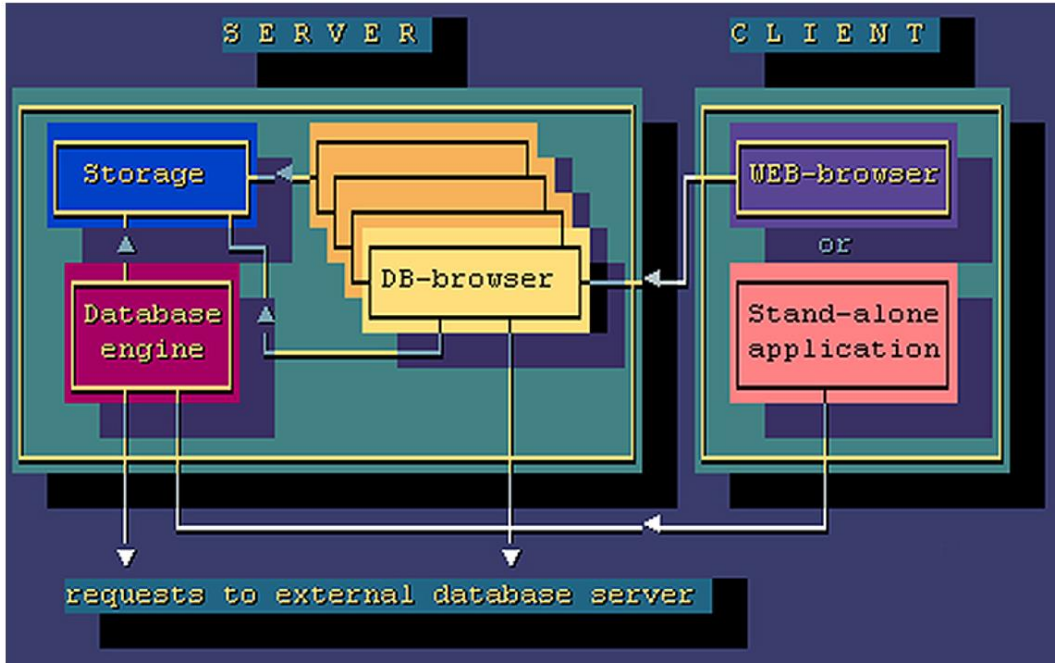
1	Mary	Hinkle	555.123.4567
2	Paul	Litz	555.258.8963

Compiled by Rick Dobson
Graphics & Design by Fred Schneider

History of databases

- Late 1960s and 1970s:
  - Hard disks allowed direct access to data
  - Network and hierarchical data models in widespread use
  - Ted Codd defines the relational data model
    - IBM Research begins System R prototype
    - UC Berkeley (Michael Stonebraker) begins Ingres prototype
    - Oracle releases first commercial relational database
  - High-performance (for the era) transaction processing
  - Advent of DBMS

Database Management Systems



History of databases

OTH Amberg-Weiden

- 1980s
  - Research relational prototypes evolve into commercial systems
  - SQL („Structured Query Language“) becomes industrial standard
  - Parallel and distributed database systems
  - Object-oriented database systems

History of databases

OTH Amberg-Weiden

- 1990s:
  - Large decision support and data-mining applications
  - Large multi-terabyte data warehouses
  - Emergence of Web commerce

History of databases

OTH Amberg-Weiden

- 2000s
  - Big data storage systems
  - Google BigTable, Yahoo PNuts, Amazon,
  - “NoSQL” systems.
  - Big data analysis: beyond SQL

History of databases

OTH Amberg-Weiden

- 2010s
  - SQL reloaded
  - SQL front end to Map Reduce systems
  - Massively parallel database systems
  - Multi-core main-memory databases

What is a ...

OTH Amberg-Weiden

... database

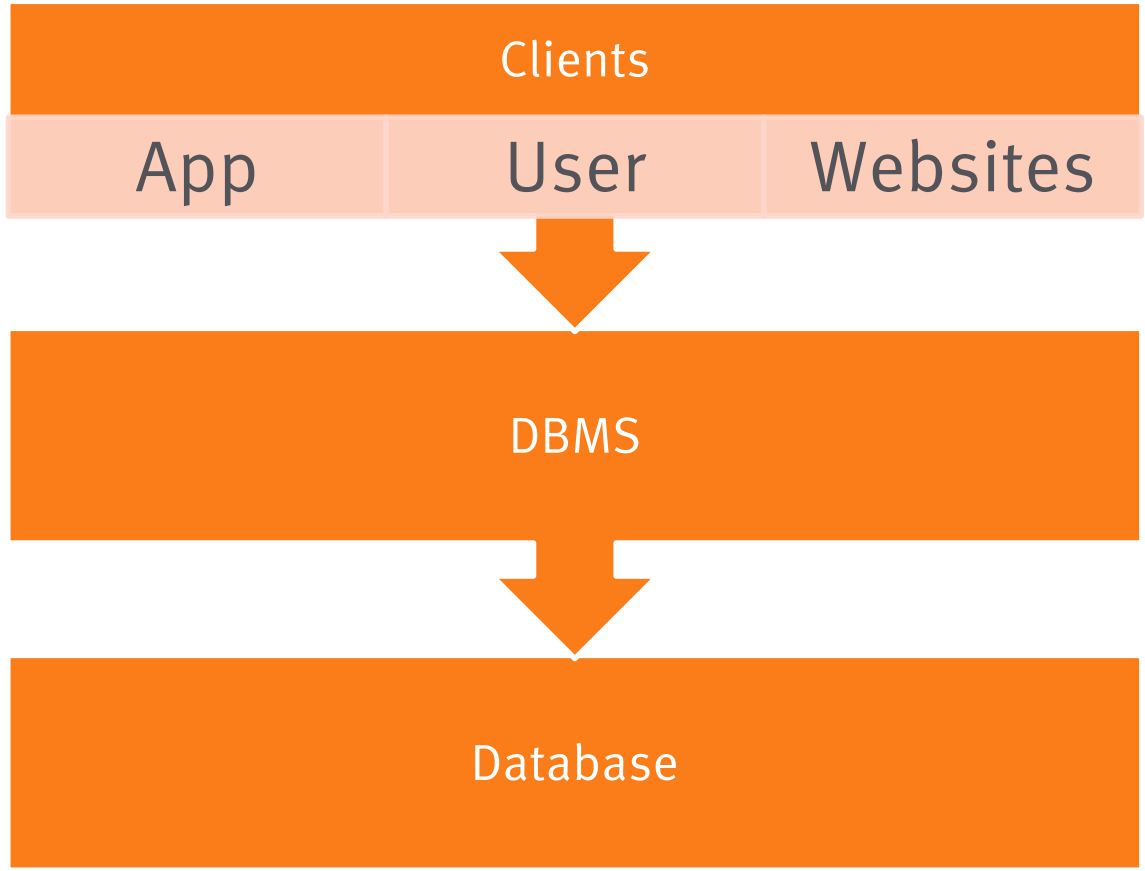
- ... database management system (DBMS)?

Database Management System

OTH Amberg-Weiden

- interacts with apps and users
- interacts with database itself

- organizes data and its storage
- handles user requests
- assures data integrity
- enables scalability



Types of DBMS

OTH Amberg-Weiden

- Relational DBMS (RDBMS): These systems store data in table form and use relational models to manage the data relationships.
  - Examples include MySQL, PostgreSQL and Oracle.
- NoSQL DBMS: These systems are used when large amounts of unstructured data need to be managed, such as in big data applications.
  - Examples include MongoDB, Cassandra and Couchbase
- In-memory DBMS: These systems store data in the main memory instead of on hard disks, which enables very fast data access. They are often used in real-time applications, such as those required in the manufacturing industry for process monitoring.

Applications of DBMS

OTH Amberg-Weiden

smart home

industry

healthcare

AI

Introducing of SQL Games