



FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

Datenbanksysteme I

Prof. Dr. Viktor Leis

Professur für Datenbanken und Informationssysteme

1. Datenbankentwurf (ER-Modell)
2. relationales Modell (relationale Algebra)
3. **SQL** (SELECT, CREATE TABLE, INSERT, UPDATE)
4. Relationale Entwurfstheorie (funktionale Abhängigkeiten, Normalformen)
5. physische Datenorganisation (B+Bäume)
6. Transaktionen (ACID, grundlegende Funktionsweise)
7. Anfragebearbeitung (kanonische Übersetzung, logische Optimierung, Joinalgorithmen)
8. grafische Datenanalyse mit ggplot (geometrische Objekte, Abbildung von Attributen auf ästhetische Eigenschaften, faceting)

1. normalisierte Speicherung von Daten in Relationen
 - effiziente Änderungen der Daten
 - keine Verschwendung von Speicherplatz
2. deklarative Anfragen (SQL)
 - Beantwortung *beliebiger* Anfragen
 - automatische Anfrageoptimierung
3. Transaktionen
 - Sicherstellung von Persistenz
 - ermöglicht gleichzeitiger Datenzugriff

Hierarchisches Datenmodell

- bevor das relationale Modell erfunden wurde, gab es hierarchische Datenbanken

- moderne Inkarnation: JSON, XML

- Beispiel JSON 1 (Vorlesungen unter Studenten):

```
[{ "name": "Xenokrates", "semester": 18, "vorlesungen":  
  [ {"titel": "DBMS I", "SWS": 6} ] },  
 { "name": "Jonas", "semester": 12, "vorlesungen":  
  [ {"titel": "DBMS I", "SWS":6}, {"titel": "DBMS 2", "SWS":6}]]]
```

- Beispiel JSON 2 (Studenten unter Vorlesungen):

```
[ {"titel": "DBMS I", "SWS": 6, "studenten":  
  [{ "name": "Xenokrates", "semester": 18}]},  
 {"titel": "DBMS II", "SWS": 6, "studenten":  
  [{ "name": "Xenokrates", "semester": 18},  
   { "name": "Jonas", "semester": 12 }]}]
```

- Redundanz

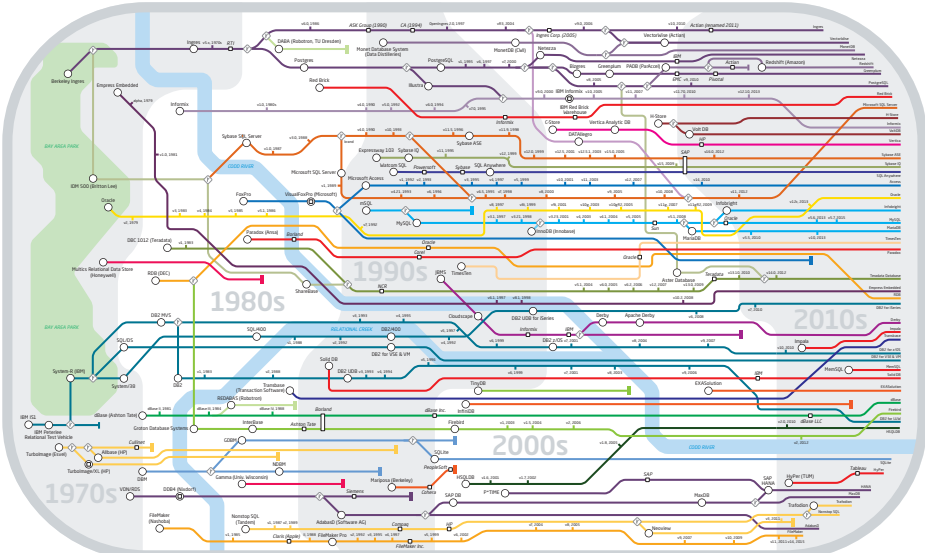
- Turing Award für Charles Bachman, 1973
- moderne Inkarnation: Modellierung in den meisten objektorientierten Programmiersprachen
- Beispiel:

```
class Vorlesung { String titel; int SWS; }  
class Professor { String name; String rang; Vorlesung v[]; }  
class Student { int matrnr; String name; int semester; }  
class Pruefung { Vorlesung v; Professor p; Student s; float note; }
```

- komplexe Anfragen, komplexe Datenbanksysteme

1. Idee durch Edgar F. Codd 1970 (IBM Research), Turing Award 1981
2. 1980: erste kommerzielle Produkte (Oracle, IBM, Ingres)
3. 1990: Technologie setzt sich durch
4. 2000: Dominanz der “Dinosaurier” (Oracle, Microsoft, IBM)
5. 2010-jetzt: viele neue Systeme (auch NoSQL)

Genealogy of Relational Database Management Systems



Key to lines and symbols

- DBMS name (Company)
- Acquisition
- ▲ Versions
- ⊥ Discontinued
- ◇ Branch (Intellectual and for code)
- Crossing lines have no special semantics

Felix Naumann, Jona Bauckmann, Claudia Exler, Jan-Peer Rautigh, Fabian Techner
Contact: Hasso Plattner Institut, University of Potsdam, Felix.naumann@uni-potsdam.de
Design - Alexander Seitz Castle Design, Hamburg
Version 6.0 - October 2024
<https://gitlab.hpi.uni-potsdam.de/rwthinfo/genealogy.html>

NoSQL, Key/Value Stores, Document Stores

- Hintergrund: relationale Datenbanksysteme und SQL sind (angeblich) zu kompliziert und ineffizient
- “Lösung”: hierarchisches Datenmodell, kein Schema, keine Transaktionen, keine deklarativen Anfragen, keine Normalisierung
- Datenbanksystem speichert komplexe Objekte (z.B. JSON)
- Zugriff zu Objekten über definierte Schlüssel
- “schema-free” bedeutet, dass das Schema implizit in den Anfragen steckt
- Beispiele: Redis, MongoDB