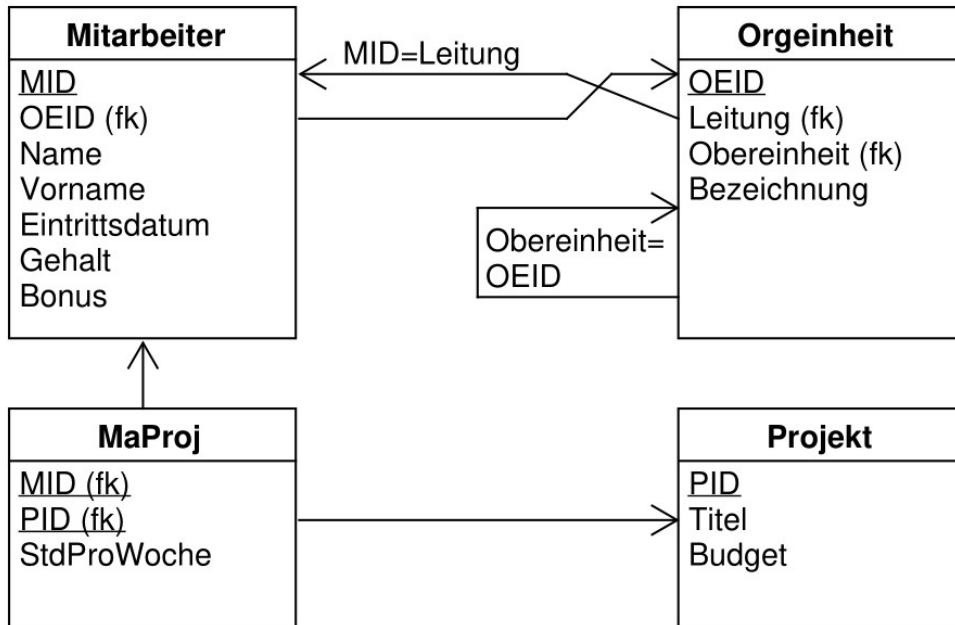


- Viele Aspekte von Datenbanksystemen sind alt, aber heutzutage immer noch relevant
- Immer mal wieder wird ein Ersatz für SQL vorgeschlagen. Nach einiger Zeit stellt sich heraus, dass dieser nicht wirklich besser ist
- Stattdessen werden sinnvolle Ideen dieser Ersatzvorschläge in SQL übernommen

- Dieser Überblick basiert auf Folien von Andy Pavlo (<http://www.cs.cmu.edu/~pavlo/>)
- History of Databases (<https://15721.courses.cs.cmu.edu/spring2023/schedule.html>)

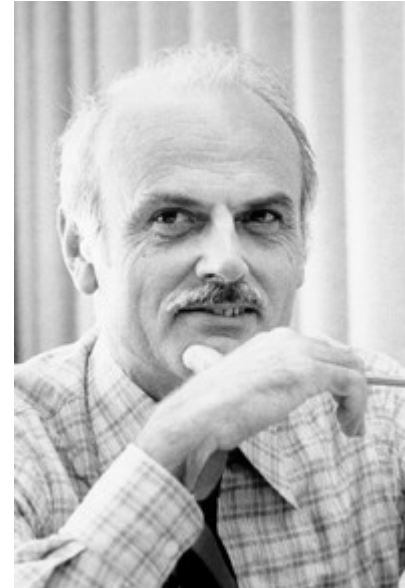
- Einzeltupel-Verarbeitung
- Programmorientierter Zugriff
- Keine deklarative Abfragesprache
- Vertreter:
 - Netzwerk-Datenmodell
 - Hierarchisches Datenmodell
- Noch in Großrechnersystemen im Einsatz
- Keine Neuentwicklung auf deren Basis

- Tabellen
- Referenzen zwischen Tabellen
- Normalisierung



Theoretische Grundlagen

- Ted Codd: A relational Model of Data for Large Shared Data Banks



- Reines Konzept
- Implementierbarkeit effizienter Systeme unklar

Erste Implementierungen:

- System R – IBM Research
- INGRES – U.C. Berkeley
- Oracle – Larry Ellison

- Das relationale Modell setzt sich durch
- SQL-Standard
- Kommerzielle Anbieter
 - Oracle
 - IBM DB2
 - Sybase
 - Informix
 - Terradata
 - ...
- Oracle setzt sich durch
- Stonebraker entwickelt Postgres als objektrelationales DBMS



- Vermeidung des "relational-object impedance mismatch"
- Direkte Umsetzung von Konzepten objektorientierter Sprachen in der Datenbank
- Führen nur noch ein Nischendasein
 - ObjectStore, letztes Update 2014, Rank 141 in DB Engines

Donnerstag, 2. März 2023 10:44

- Keine großen Veränderungen/Fortschritte im Bereich Datenbanksysteme
- Kommerzielle Anbieter beherrschen den Markt
- Neue Systeme:
 - Microsoft SQL Server
 - MySQL
 - SQLite
- Aufkommen dedzierter OLAP-Systeme - Data Cubes



- Kommerzielle Anbieter zu schwergewichtig und teuer
- Open-Source-Systemen fehlen wichtige Funktionen
- Viele Unternehmen bauen eigene Middleware zur horizontalen Skalierung von Datenbanksystemen (scale out statt scale up)

Aufkommen spezieller OLAP-Systeme

- Verteilt / Shared-Nothing
- Relational / SQL
- Kommerziell

- Entscheidende Leistungsgewinne durch Spaltenorientierung

- Systeme
 - Netezza
 - Greenplum
 - Vertica
 - MonetDB



Verteiltes Programmier- und Ausführungsmodell zur Analyse sehr großer Datenmengen (Big Data)

- Google entwickelt erste Lösung - für seine Suchmaschine
- Yahoo - Hadoop - Open Source, Java
- Kein explizites Datenmodell - steckt in den Analysefunktionen
- kein SQL

- Großer Hype
- Große Probleme
- Einführung von SQL in dieses Modell, keine gute Idee

MAPR-DB

Fokus auf hoher Verfügbarkeit und großer Skalierbarkeit

- Schemalos
- Nichtrelational (z.B. Schlüssel/Wert-Modell, Spaltenfamilien, Dokumente)
- Keine Transaktionen
- Kein SQL
- Open Source

Bekannteste Vertreter



Vereinigung der Vorteile von SQL- und NoSQL-Systeme

- Verteilt
- Hohe Verfügbarkeit
- Horizontale Skalierbarkeit
- Relational
- SQL
- Transaktionen

Bekannter Vertreter:



Datenbanken als Service
Containerbasiert
Cloud-Native, komplett neues Design



- Trennung von Ausführungs- und Speichermaschine
- Disaggregation
- Unabhängige Skalierung der beiden Komponenten
- Log Structured Storage

Graph-Strukturen - Knoten und Kanten
Abfragesprache Cypher
SQL:2023 ergänzt SQL um Graph Query Syntax

Native Graph Databases - Direkt Speicherung von Referenzen

- Behauptete bessere Performance als relationale Systeme
- Neuere Forschung zeigt, dass relationale Systeme vergleichbare bzw. besser Performance bringen können als native Systeme



Spezialisierte Systeme für Zeitreihen- bzw. Ereignisdaten



Database of Databases

<https://dbdb.io/>

DB-Engines

<https://db-engines.com/de/>