

1. Übersicht Datenbanksysteme

1. Übersicht

2. Konzeptueller Datenbankentwurf (ER-Modell)
3. Logischer Datenbankentwurf (relationales Datenmodell)
4. Datenbanksprachen
 - 4.1 SQL DDL und DML
 - 4.2 SQL-Anfragen 1
 - 4.3 SQL-Anfragen 2
 - 4.4 SQL Programmiersprachen-Anbindung

Literaturhinweise zur Vorlesung

- DAS Datenbank-Buch im deutschsprachigen Raum
 - A. Kemper, A. Eickler: Datenbanksysteme – Eine Einführung, De Gruyter Verlag, 10. Auflage, 2015
- Vorlesungsvideos zu Datenbanksysteme
 - <https://db.in.tum.de/teaching/bookDBMSeinf/aufzeichnungen/index.shtml?lang=de>
- Dokumentation Datenbanksystem Oracle
 - <http://oracle19c.in.htwg-konstanz.de>

Einführung

- Informationsexplosion im Internetzeitalter ("Datenzeitalter")
 - Daten als dritter Produktionsfaktor neben Kapital, Arbeit, Rohstoffe (s. A. Gadatsch)
 - Daten als "das neue Öl"
 - Daten als Quelle von Wertschöpfung und Innovation
 - Ziele von Datenbanksystemen
 - Effiziente Verwaltung großer Datenmengen
 - Daten mehreren Benutzern zur Verfügung stellen
- Steigende Bedeutung von Datenbanken



Bedeutung von Datenbanken

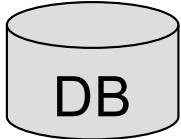
"If any technology was essential to the rebuilding of our daily lives around digital infrastructures, it was the database management system."

Thomas Haigh : How Charles Bachman Invented the DBMS, a Foundation of Our Digital World, Communication of the ACM, 2016

- Datenbanksysteme als „wichtigste Vorlesung im Studium“
 - Ziel sollte nicht nur das Bestehen der Vorlesung sein

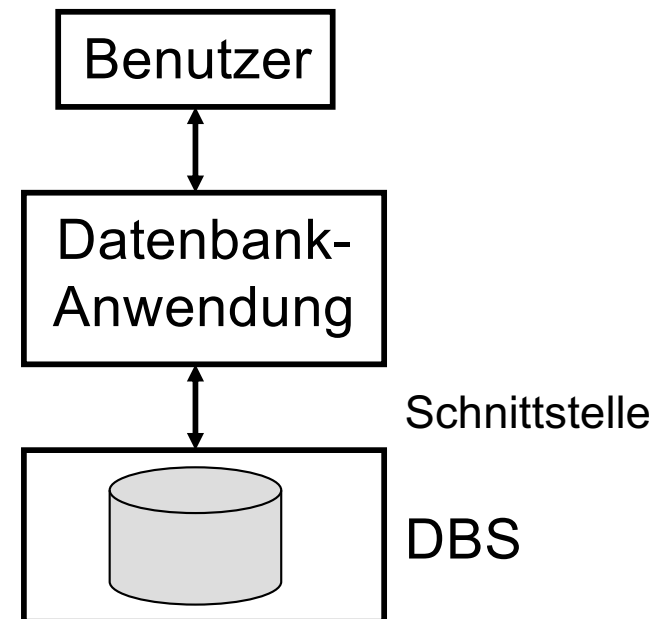
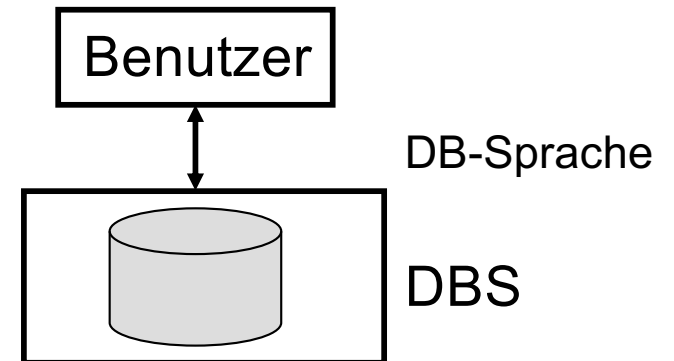
Datenbanken und Datenbanksysteme

Definitionen

- Sehr allgemeine Definition von Datenbanken
 - Eine Datenbank ist eine selbständige und auf Dauer ausgelegte Datenorganisation, welche einen Datenbestand sicher und flexibel verwalten kann
 - Datenbank (DB)
 - Sammlung von Daten von einem Datenbankmanagementsystem
 - Symbol
- 
- Datenbankmanagementsystem (DBMS)
 - Programm zur Kontrolle des Datenbankzugriffs
 - Datenbanksystem (DBS)
 - Datenbank + Datenbankmanagementsystem

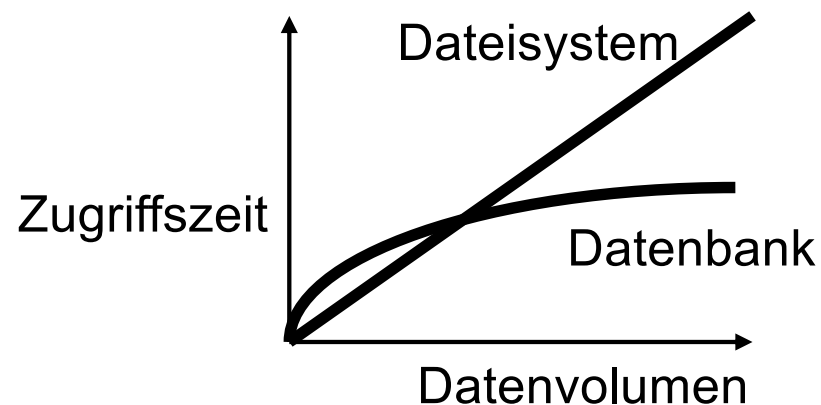
Nutzung von Datenbanksystemen

- Direkte Nutzung
 - Direkte Nutzung des Endbenutzers
 - Verwendung einer Datenbank-sprache, z.B. SQL
- Programmorientierte Systeme
 - Kommunikation mit Datenbank erfolgt über ein Programm
 - Programmierung des Programms durch Datenbankanweisungen



Eigenschaften von Datenbanksystemen

- Dauerhafte Speicherung und Ausfallsicherheit
 - Daten bleiben nach Programmende erhalten
 - Daten bleiben nach Systemausfall erhalten (Recovery)
- Verwaltung großer Datenmengen
 - Datenmenge im Bereich TB
 - Effiziente Suchmethoden



Eigenschaften von Datenbanksystemen

- Zugriffskontrolle und Sicherheit
 - Einschränkung der für Benutzer zugreifbaren Daten
 - Zugriffsrechte
- Konsistenz- und Integritätsprüfungen
 - Konsistenzbedingungen werden formuliert und überprüft
 - Beispiel: Kontostand von Sparbuch wird niemals negativ
 - "Alles-oder-Nichts"-Eigenschaft
- Redundanzfreiheit
 - Jedes Datum wird genau einmal gespeichert (Ausnahme Speicherungen aus Gründen der Ausfallsicherheit)

Eigenschaften von Datenbanksystemen

- Mehrbenutzerbetrieb (Concurrency control)
 - Gleichzeitiger Zugriff auf gleiche Datenbestände zugreifen durch mehrere Benutzer
 - Vermeidung von Anomalien durch Synchronisationsverfahren
 - Verwendung von Sperrprotokollen
- Hohe Verfügbarkeit und Fehlertoleranz
 - Keine separaten Zeiträume für Reorganisation der Daten oder Erstellen von Sicherheitskopien

Eigenschaften von Datenbanksystemen

- Anfragesprache
 - Inhaltsbezogener Zugriff durch deklarative Sprachen
 - Suchmechanismen
 - Beispiel SQL
- Kopplung mit Anwendungssystemen
 - Integration mit Programmiersprachen
 - Verkürzung der Entwicklungszeit von Anwendungen, da auf eine Schnittstelle zur Datenverwaltung aufgesetzt werden kann
- Skalierbarkeit
 - Anpassbarkeit an Größe und Leistungsanforderungen
 - Lineare Leistungssteigerung durch
 - Vergrößerung von Haupt- oder Externspeicher
 - mehr oder leistungsfähigere Prozessoren

Zeittafel der Datenbanksystem-Generationen

- 1. Generation
 - Filesysteme auf Band
- 2. Generation
 - Filesysteme auf Platte
- 3. Generation
 - Prärelationale Systeme (Netzwerk-, hierarchische Systeme)
- 4. Generation
 - Relationale Systeme
- 5. Generation
 - Postrelationale Systeme (u.a. objektbasierte Systeme, NoSQL)

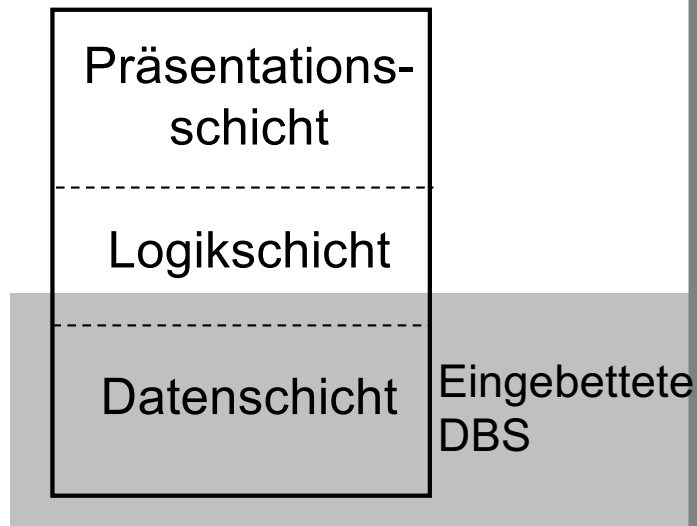
Datenmodelle

- Relationales Datenmodell
 - Basiert auf Relationenkalkül der Algebra
 - Mächtige Datenzugriffssprache SQL
 - Datenunabhängigkeit
 - Vertreter DB2 von IBM, Oracle, Informix

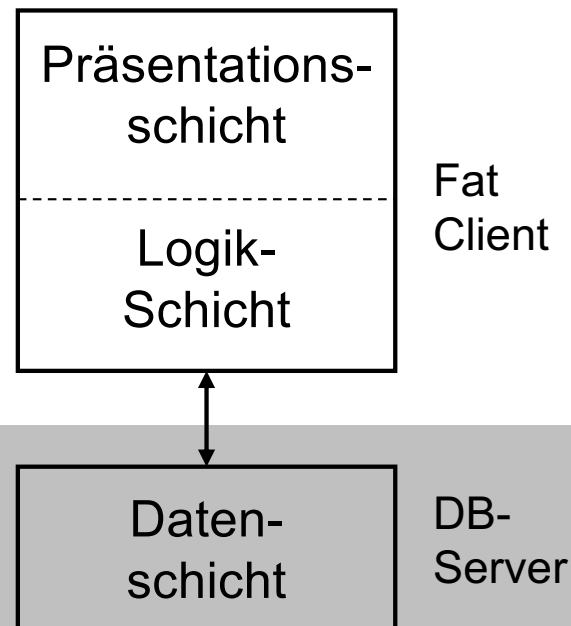
Pers							
<u>pnr</u>	name	jahrg	eindat	gehalt	beruf	<u>anr</u>	<u>vnr</u>
406	Coy	1950	01.03.86	80.000	Kaufmann	K55	123
123	Mueller	1958	01.09.80	68.000	Programmierer	K51	
829	Schmidt	1960	01.06.90	74.000	Kaufmann	K53	123
874	Abel		01.05.94	62.000	Softw.Entwickler	K55	829
503	Junghans	1975		55.000	Programmierer	K51	123
...

Architekturen von Datenbankapplikationen

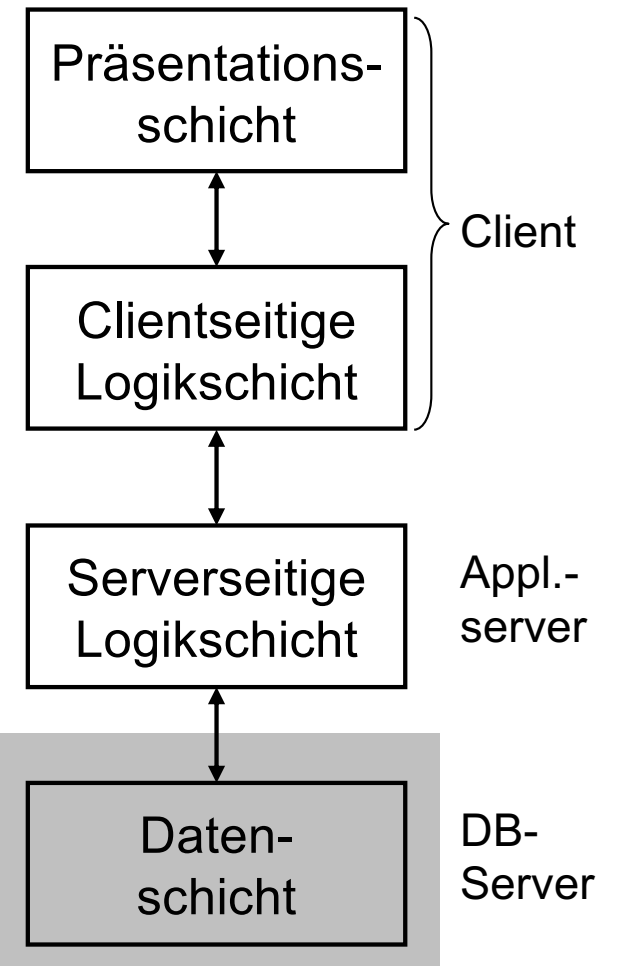
1-tier Architektur



2-tier Architektur

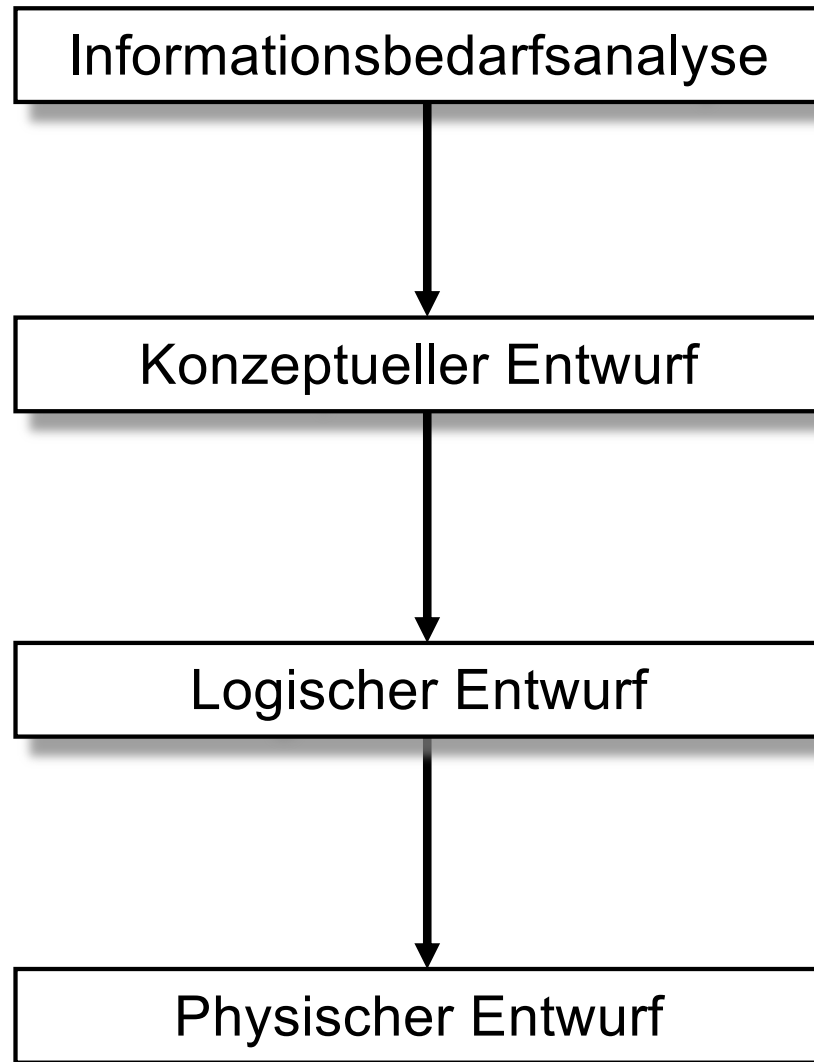


n-tier Architektur



Datenbankentwurf

Entwurfsschritte



Sammlung aller für eine Miniwelt bedeutsamen Gegenstände, Eigenschaften, Beziehungen und Operationen

Präzise Beschreibung einer Miniwelt durch relationale oder objektorientierte Modelle

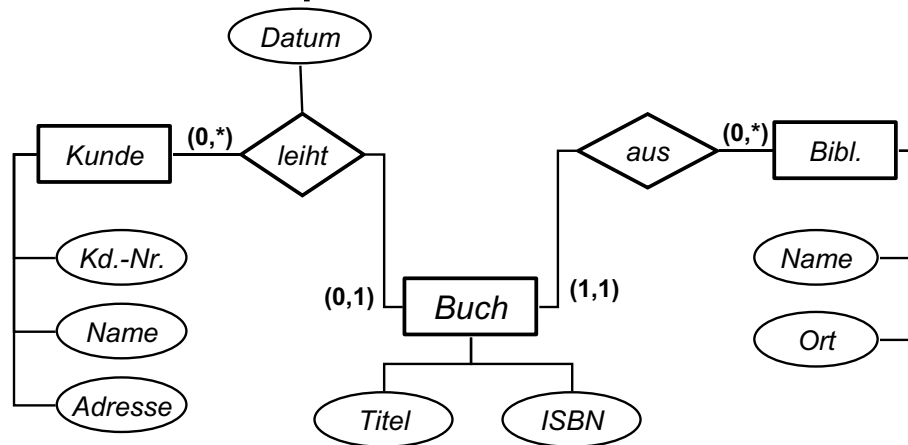
Abbildung auf ein rechnergestützt interpretierbares Schema, z.B. relationales Schema

Abbildung des logischen Datenbankschemas in eine effiziente physische Datenbasisstruktur

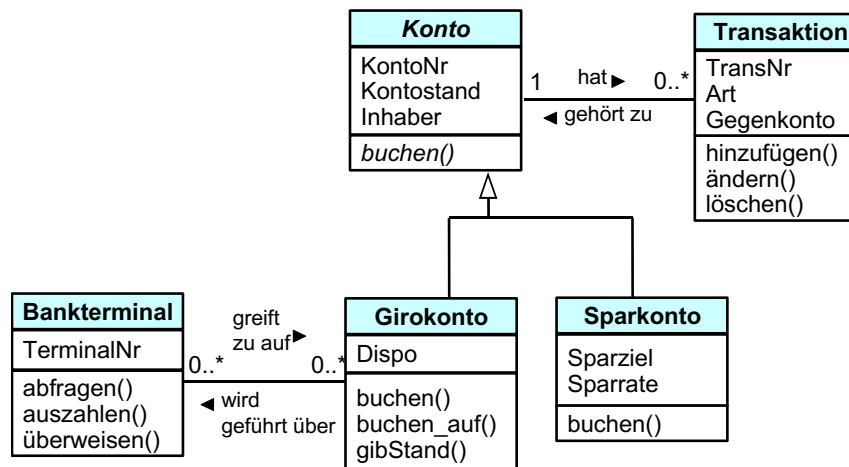
Konzeptueller Entwurf

Relationale und objektorientierte Modellierung

- Entity-Relationship-Modell

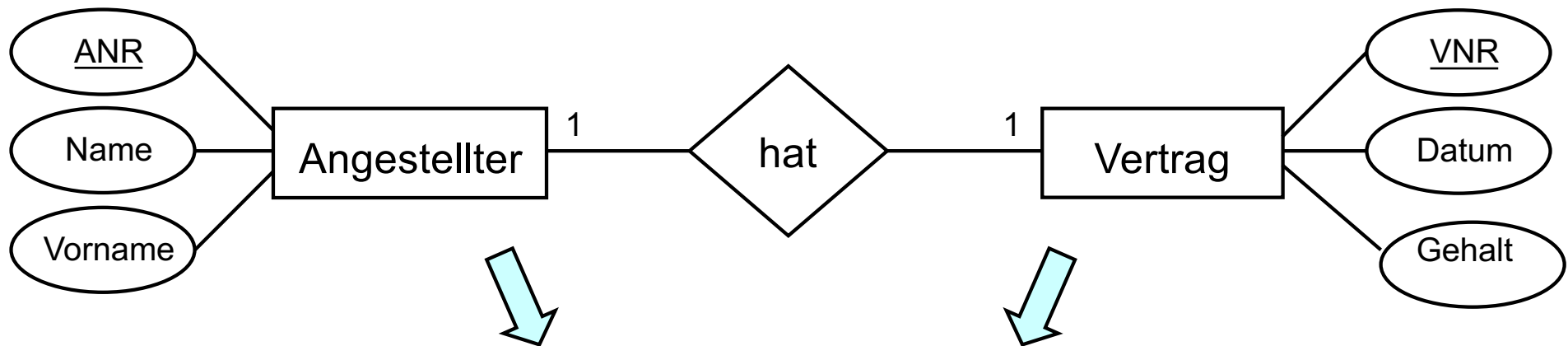


- Objektorientiertes Modell



Logischer Entwurf

- Abbildung auf relationales Schema, Normalisierung



Angestellter = ({ANR, Name, Vorname, VNR})

Vertrag = ({VNR, Datum, Gehalt})

Beispiel:

Angestellter	<u>ANR</u>	Name	Vorname	VNR
	2004	Müller	Hans	2004
	1208	Zimmer	Jochen	1208
	1001	Abel	Kai	1001

Vertrag	<u>VNR</u>	Datum	Gehalt
	2004	1.10.1999	3500
	1208	1.1.2002	4000
	1001	1.3.1990	5500

Physischer Entwurf

- Verwendung der Datenbanksprache SQL

```
CREATE TABLE Angestellter
( ANR          integer,
  Name         varchar(20),
  Vorname      varchar(20),
  VNR          integer,
  CONSTRAINT pk_Angestellter PRIMARY KEY (ANR),
  CONSTRAINT fk_Angestellter FOREIGN KEY (VNR)
    REFERENCES Vertrag(VNR)
    ON DELETE SET NULL );

CREATE TABLE Vertrag
( VNR          integer,
  Datum        date,
  Gehalt       integer,
  CONSTRAINT pk_Vertrag PRIMARY KEY (VNR) );
```