



# Weshalb Datenbanken?

## Probleme und abgeleitete Anforderungen

---

Prof. Dr. Inga Marina Saatz  
© 2016

**Fachhochschule  
Dortmund**  
University of Applied Sciences and Arts

# Weshalb Datenbanken?

## Ein Blick zurück...

# Weshalb Datenbanken?



Verkauf

Kundenliste

Kunden

Datei Kunden

Kunden-number	Name	Vorname	Anrede	Strasse	Wohnort
4812	Sklodowska	Marie	Frau	Kampstr. 1	Dortmund
4815	...	...	...	...	...

Buchhaltung

Zahlungen

Belege

Datei Belege

Kunden-number	Anrede	Vorname	Nachname	Datum	Betrag
4812	Frau	Marie	Sklodowska	29.09.2008	47,11
7562	...	...	...	...	...

# Das Arbeiten ohne Datenbanken: Probleme

# Problem: Inkonsistenz



# Problem: Inkonsistenz

Datei Kunden

Kunden-nummer	Name	Vorname	Anrede	Strasse	Wohnort
	Curie				
4812	Skłodowska	Marie	Frau	Kampstr. 1	Dortmund
4815	...	...	...	...	...

## Redundanz

Datei Belege

Kunden-nummer	Anrede	Vorname	Nachname	Datum	Betrag
4812	Frau	Marie	Skłodowska	29.09.2008	47,11
7562	...	...	...	...	...

## Inkonsistenz

## Anforderungen

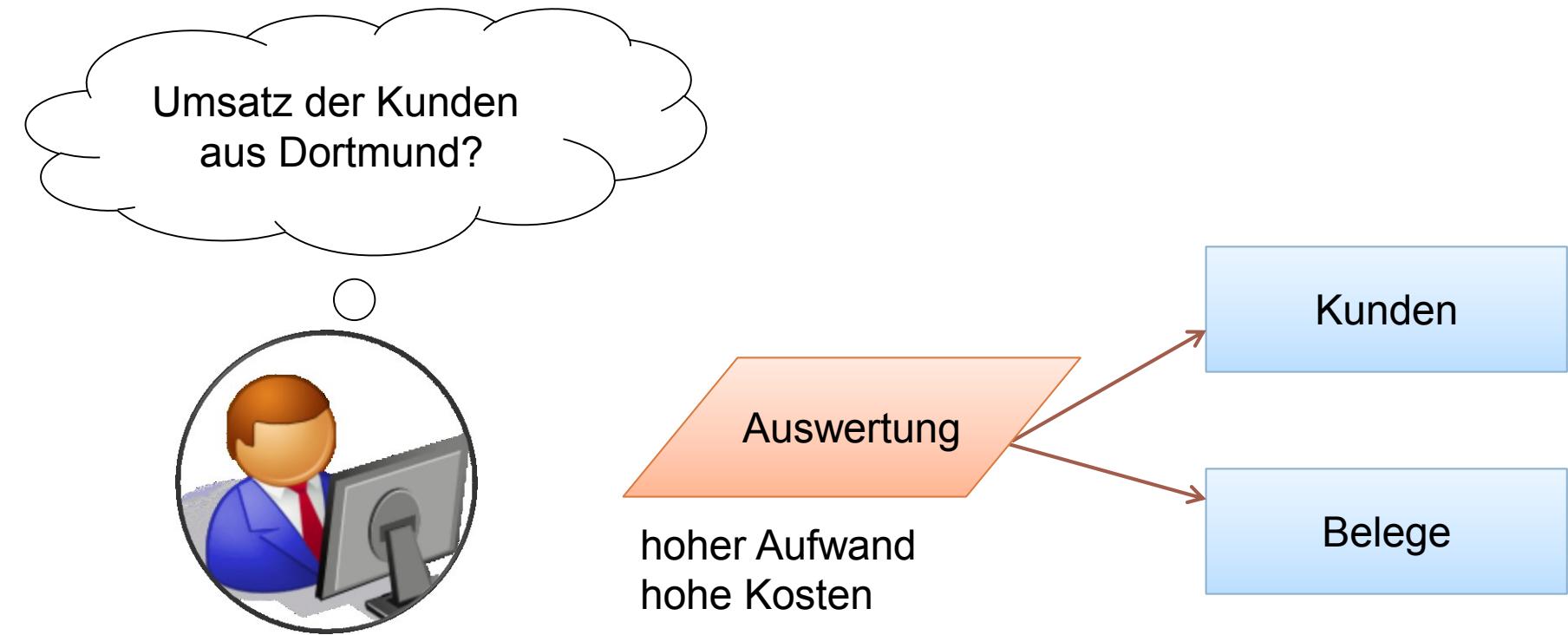
- Datenintegrität
  - Einheitliche Verwaltung aller von Anwendungen benötigten Daten
- Konsistenzüberwachung
  - Korrektheit der Datenbankinhalte
  - Korrekte Ausführung von Änderungen

# Problem: Inflexibilität

Umsatz der Kunden  
aus Dortmund?



# Problem: Inflexibilität



## Anforderungen

- Katalog der Datenbeschreibungen
- Flexible Operationen

# Problem: Datenschutz



# Problem: Datenschutz

Aushilfe

Kundenliste

Kunden

Kunden- nummer	Name	Vorname	Anrede	Bankleitzahl	Konto
4812	Curie	Marie	Frau	44350013	9987342
4815	...	...	...	...	...

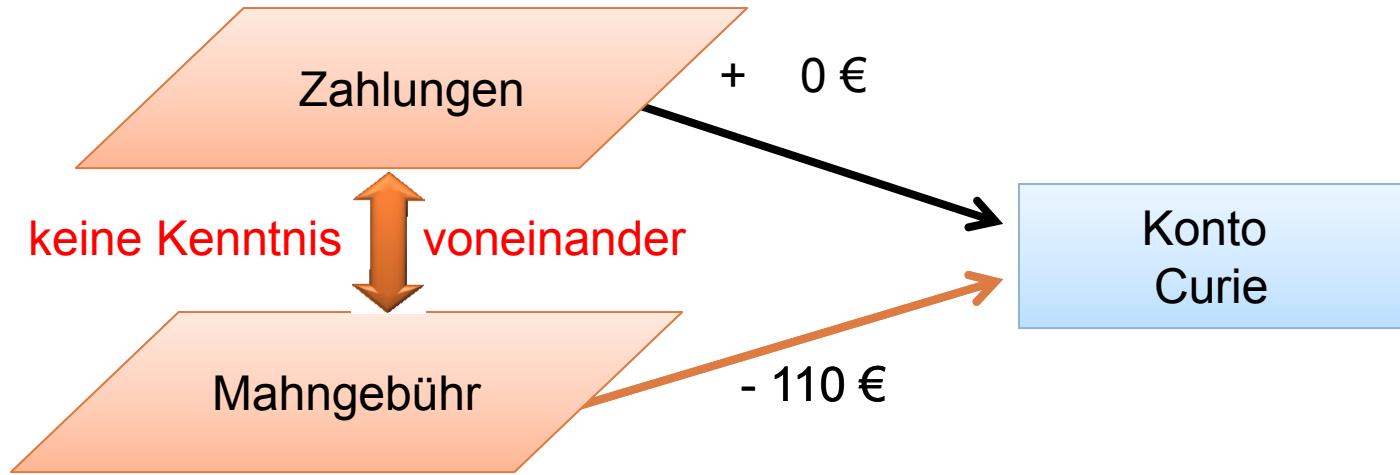
## Anforderungen

- Benutzersichten  
Verschiedene Nutzer (Aushilfe, Sachbearbeiter, ...) sehen unterschiedliche Ausschnitte des Datenbestandes
- Datenschutz  
Rechte der Nutzer zur Datensicht und Datenmanipulation können verwaltet werden

# Problem: Synchronisation

(Gleichzeitige Daten-Manipulation)

# Problem: Synchronisation



## Programm Zahlungen:

Alter Kontostand:	-100 €
Zahlungseingang	100 €
Neuer Kontostand:	0 €

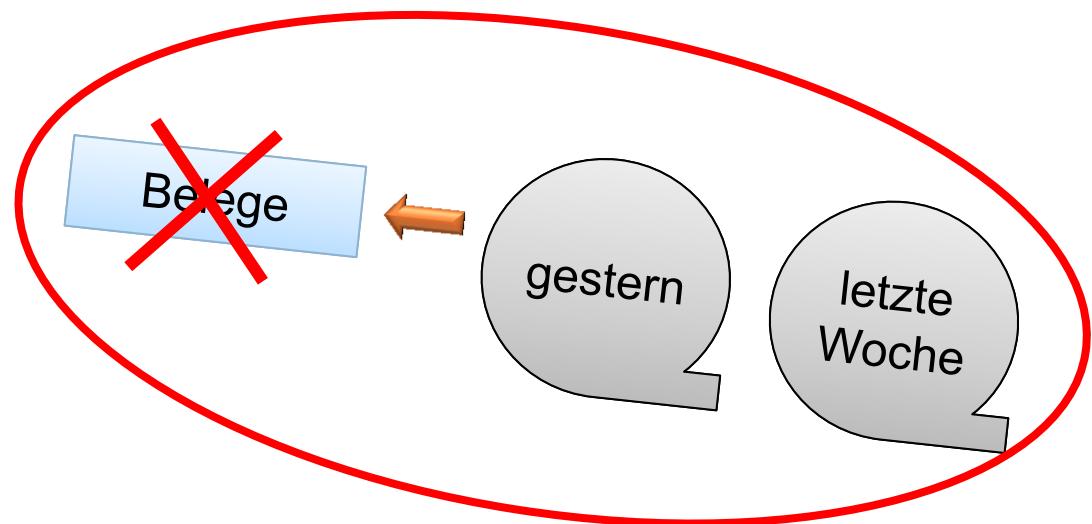
## Programm Mahngebühr

Alter Kontostand:	-100 €
Mahngebühr:	- 10 €
Neuer Kontostand:	-110 €

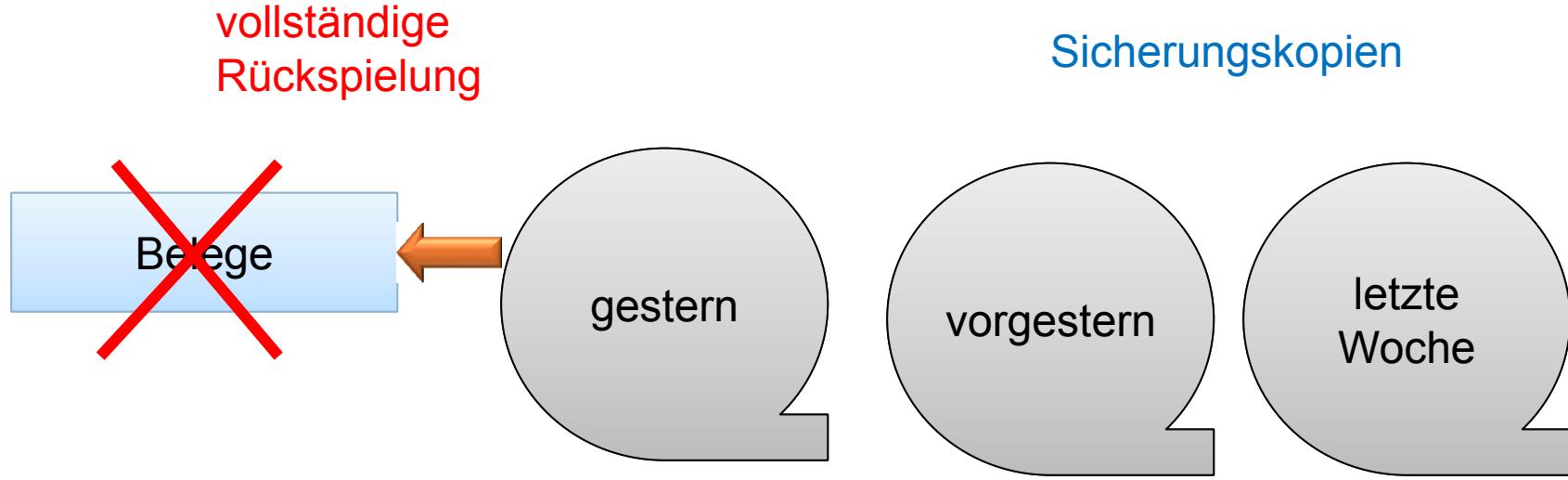
## Anforderung

- Synchronisation von Änderungen (Transaktionen) mehrerer Anwender

# Problem: Datensicherung



# Problem: Datensicherung

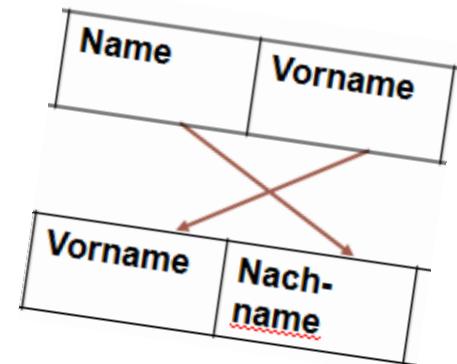


## Anforderung Datensicherung

- Mechanismen zum Vermeidung von Datenverlust durch das Rückspielen von Sicherungskopien

# Problem: Änderungen der Daten-Struktur

Physische und logische  
Datenunabhängigkeit



# Problem: Physische Datenabhängigkeit

## Änderung der Organisationsform

Sequenziell



Index

-

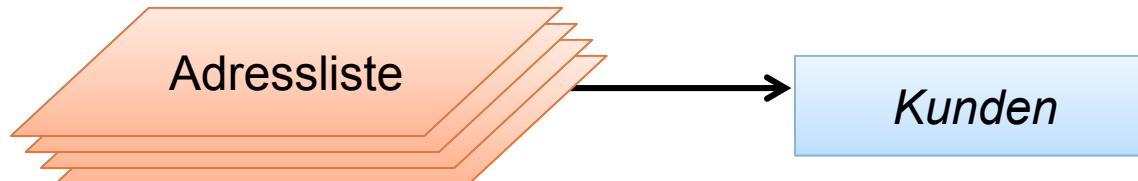
Sequenziell

Kunden- nummer	Name	Vorname
2720	Einstein	Albert
4812	Meitner	Lise
...	...	...

Schlüssel	Adresse
2720	Adresse
4812	Adresse
...	...

Quellcodeänderung

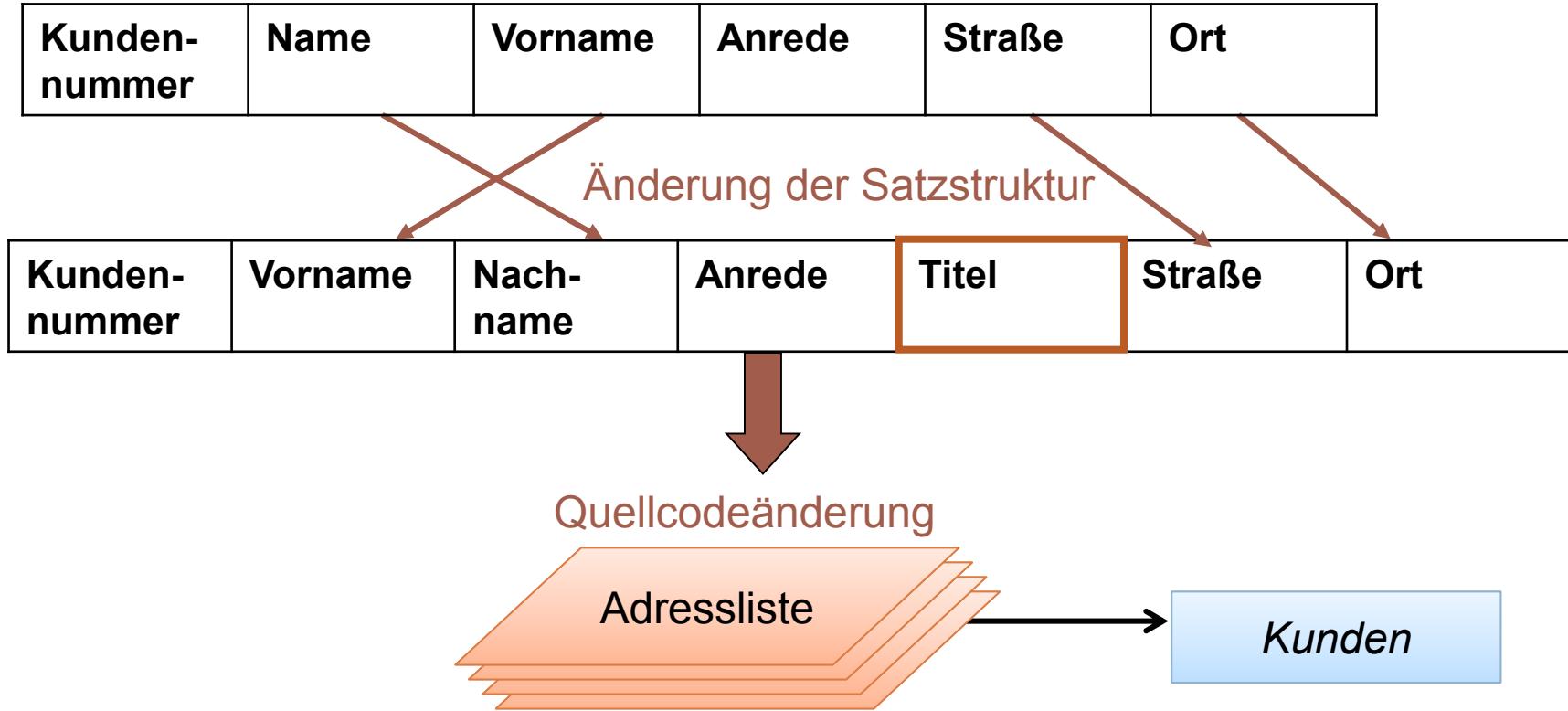
Kunden- nummer	Name	
2711	...	...
2720	Einstein	...
...	...	...
4812	Meitner	...



## Anforderung

- Physische Datenunabhängigkeit – Unabhängigkeit der Anwendungsprogramme von einer Änderung der Speicherorganisationform

# Problem: Logische Datenabhängigkeit



## Anforderung:

Logische Datenunabhängigkeit: Unabhängigkeit der Anwendungsprogramme von der Änderung der Satzstruktur der Daten

## Probleme

## Sich daraus ergebende Anforderungen

---

Inkonsistenz

- » **Datenintegrität**

Einheitlichkeit der Daten und ihrer Verwaltung für alle zugreifenden Programme

- » **Konsistenzüberwachung**

Korrekte, datenbankweite, vollständige Ausführung von Änderungen zur Gewährleistung korrekter Daten

Inflexibilität

- » **Katalog der Datenbeschreibungen**

Abbildung der Struktur der gespeicherten Daten

- » **Flexible Operationen**

Auswertungs-Funktionen für die Daten

## Probleme

## Sich daraus ergebende Anforderungen

---

Datenschutz

### » Benutzersichten

Jeder Nutzer sieht individuell angepasste „Ausschnitte“ des Datenbestandes

### » Datenschutz

Rechteverwaltung (Datensicht und -manipulation)

Synchronisation

### » Synchronisation aller Anwender-Interaktionen

Datensicherung

### » Datensicherungsmechanismen

Mechanismen zur Wiederherstellung von Daten nach Datenverlusten und dem Rückspielen von Sicherungskopien

## Probleme

## Sich daraus ergebende Anforderungen

---

Änderungen der  
Daten-Struktur

### » **Physische Datenunabhängigkeit**

Unabhängigkeit der zugreifenden Anwendungen  
von der gesamtheitlichen Speicher-Organisationsform,  
Vermeidung von Quellcode-Änderungen

### » **Logische Datenunabhängigkeit**

Unabhängigkeit der zugreifenden Anwendungen  
von der Änderung der Satzstruktur der Daten,  
Vermeidung von Quellcode-Änderungen

# 10 Anforderungen

1. Datenintegration
2. Konsistenzüberwachung
3. Katalog der Datenbeschreibungen
4. Flexible Operationen
5. Benutzersichten
6. Datenschutz
7. Datensicherheit
8. Synchronisation
9. Physische Datenunabhängigkeit
10. Logische Datenunabhängigkeit



# Das Konzept der Datenbank - 1

## Grundidee, Bestandteile, Funktionsprinzip

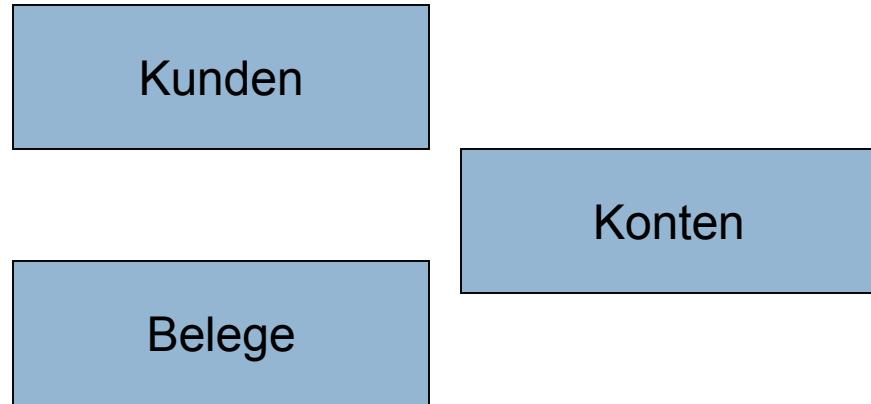
---

Prof. Dr. Inga Marina Saatz  
© 2016-2017

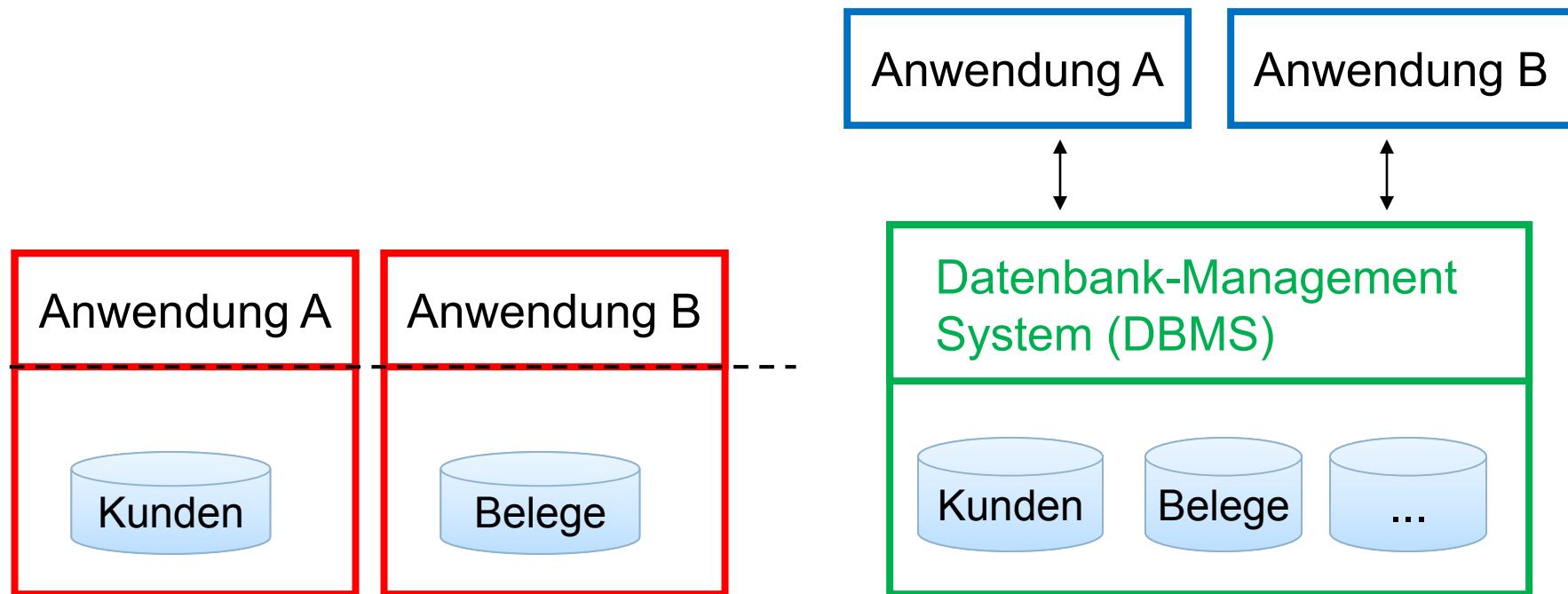
**Fachhochschule  
Dortmund**  
University of Applied Sciences and Arts



# Die Grundidee: Was ist eine Datenbank?







Programm-Datei-Kopplung  
(Dateisystem)

Zentralisierte Datensicht  
(Datenbanksystem)

Eine Datenbank (DB) ist  
eine Sammlung von Daten,  
die nach einheitlichen Regeln  
zentral gespeichert und verwaltet werden.

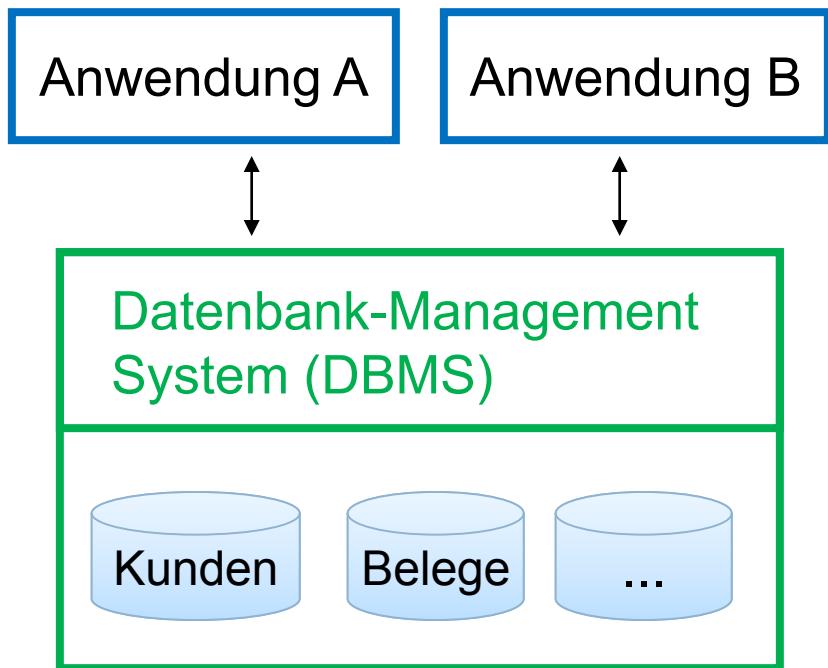
Sie enthält die Daten,  
die zur Beschreibung und Darstellung  
eines Anwendungsbereichs benötigt werden  
und über die Anwendungen hinweg gültig sind.

# Woraus besteht eine Datenbank?

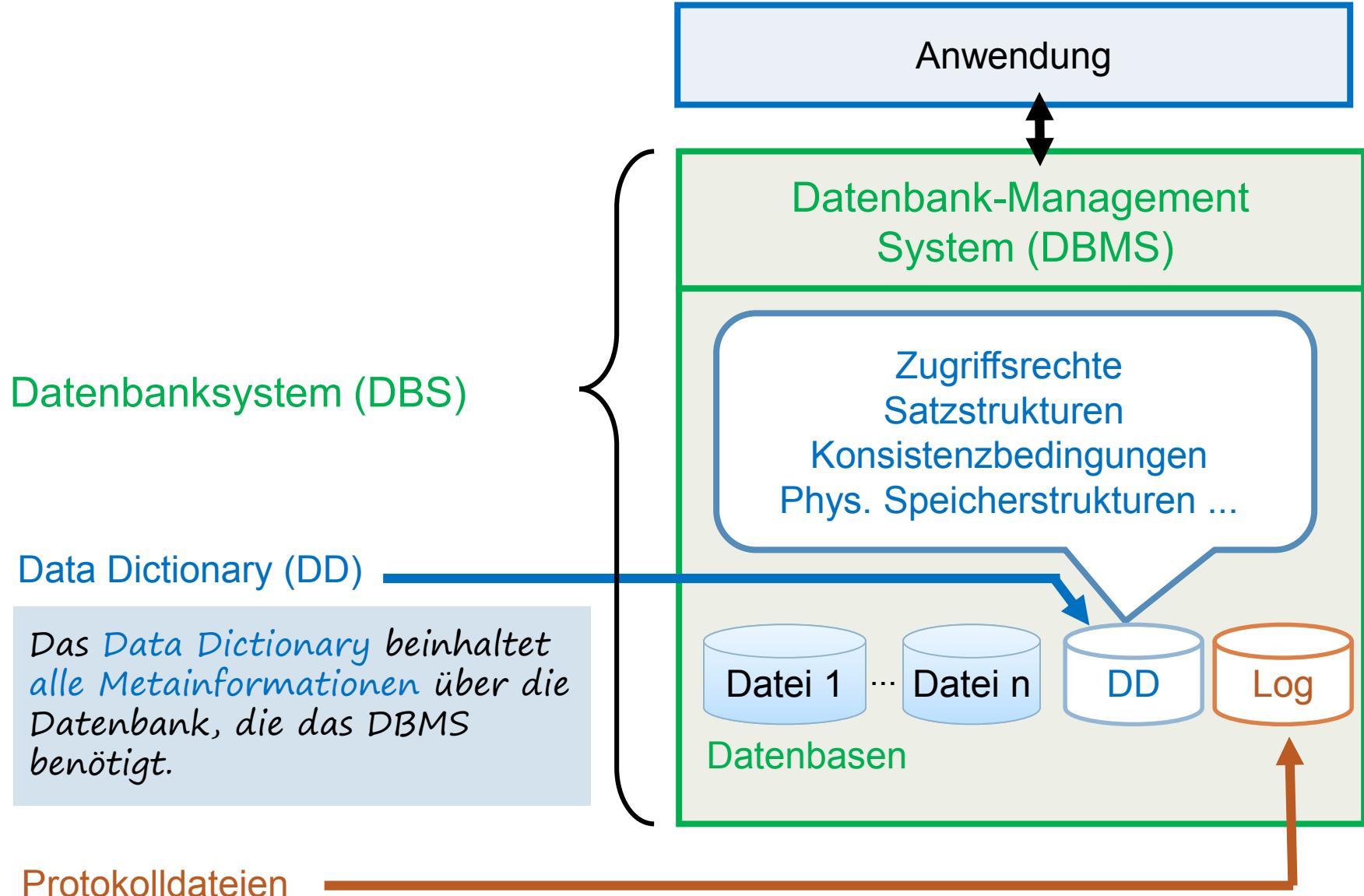
Datenbank-Managementsystem,  
Data Dictionary, Logdateien und  
Datenbasen

## Ein DBMS ...

- ▶ ... ist ein Softwaresystem zur Datenverwaltung
- ▶ ... stellt Funktionen bereit zum Einfügen, Löschen, Ändern, Sichern und Suchen von Daten.
- ▶ ... bietet eine Schnittstelle zum Anwender. Der Zugriff auf die Datenbank erfolgt ausschließlich über das DBMS
- ▶ ... kontrolliert die Datenbank.



Zentralisierte Datensicht  
(Datenbanksystem)



# Funktionsprinzip

Ablauf einer Datenbank-Anfrage,  
Transformationsregeln  
und ein SQL-Beispiel



# Funktionsprinzip

Kundenliste

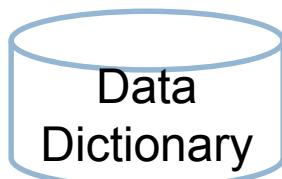
1. Liefere alle Kunden aus Dortmund in der Form

Kunden- nummer	Name	Vorname	Anrede

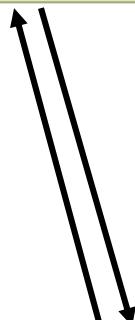
2. Ermittle die Satzstruktur der Datenbasis Kunden

Kunden- nummer	Anrede	Vorname	Name	Strasse	Wohnort

3. Ermittle die phys. Speicherstruktur  
**Sequentiell**



DBMS



# Funktionsprinzip

Kundenliste

7. Rückgabe aller Kunden aus Dortmund in der Form

Kunden-nummer	Name	Vorname	Anrede
2310	Meitner	Lise	Frau

6. Anwendung der  
Transformationsregeln

5. Interne Darstellung der Datensätze

Kunden-nummer	Anrede	Vorname	Nachname	Straße	Wohnort
2310	Frau	Lise	Meitner	...	Dortmund
7562	Hen	Albert	Einstein	...	Princeton
3365	Frau	Marie	Curie	...	Paris

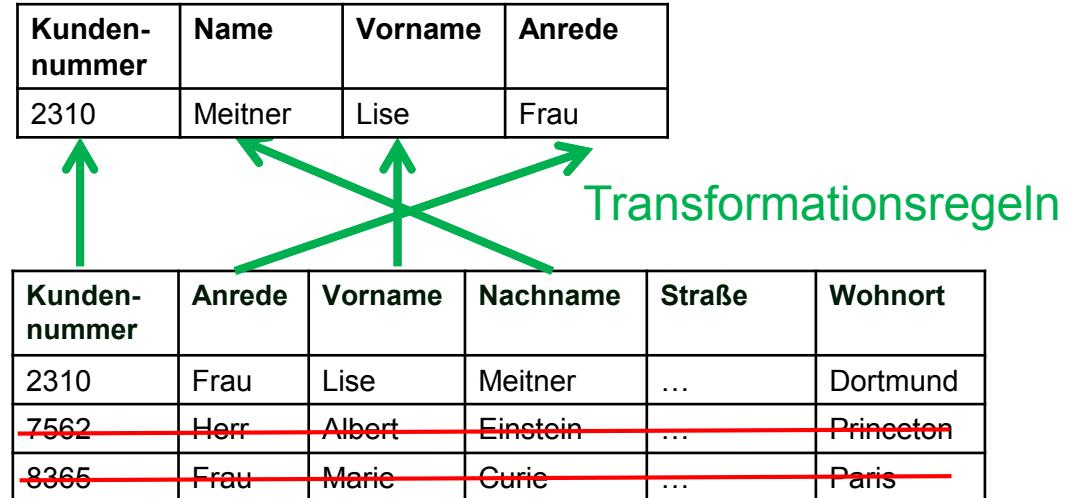
4. Auslesen der Datenbasis Kunden  
Sequentieller Dateidurchlauf

Kunden

Data  
Dictionary

# Transformationsregeln

Kunden aus Dortmund?



Transformationsregeln beschreiben, u.a. welche Daten in welcher Form weitergegeben werden, bspw. durch Auswahl und Sortierung der angefragten Attribute.

**SQL-Anfrage:**

```
SELECT Kundennummer, Name, Vorname, Anrede
FROM Kunde
WHERE Wohnort = 'Dortmund'
```

# Zusammenfassung

## Grundidee einer Datenbank:

*„Eine Datenbank ist eine Sammlung von Daten, die nach einheitlichen Regeln zentral gespeichert und verwaltet werden.“*

*Sie enthält Daten, die zur Beschreibung und Darstellung eines Anwendungsbereiches benötigt werden und die über die Anwendungen hinweg gültig sind.“*

- » **Zentralisierte Datensicht:**  
Entkopplung von Daten und Anwendungsprogrammen

## Bestandteile einer Datenbank

### Datenbank- Managementsystem (DBMS)

- » Einzige Schnittstelle zwischen Daten und Anwendung → Kontrolliert die Datenbank
- » Stellt Funktionen zur Manipulation und Verwaltung der Daten bereit (Einfügen, Löschen, Ändern, Sichern, Suchen)

### Data Dictionary

- » Metainformationen über die Datenbank: Zugriffsrechte, Satzstrukturen, Konsistenzbedingungen, physische Speicherstrukturen, ...

### Log/Protokolldateien

- » Protokollierung aller Änderungen der Datenbank → Schutz vor Datenverlust

### Datenbasen

- » Die eigentlichen Daten (in Dateien abgelegt)

## Funktionsprinzip einer Datenbank-Anfrage

---

### 1. Anwendung → DBMS

Daten-Anfrage an das DBMS

### 2. DBMS → Data-Dictionary

Das DBMS prüft auf Vorhandensein der angefragten Daten und ermittelt ihre **Ablage-Struktur**  
(Satzstruktur / die „Spalten“ der Tabellenstruktur)

### 3. DBMS → Data-Dictionary

Ermittlung der physischen **Speicherstruktur**  
z.B. sequentiell (ein Datensatz je „Zeile“ der Tabelle)

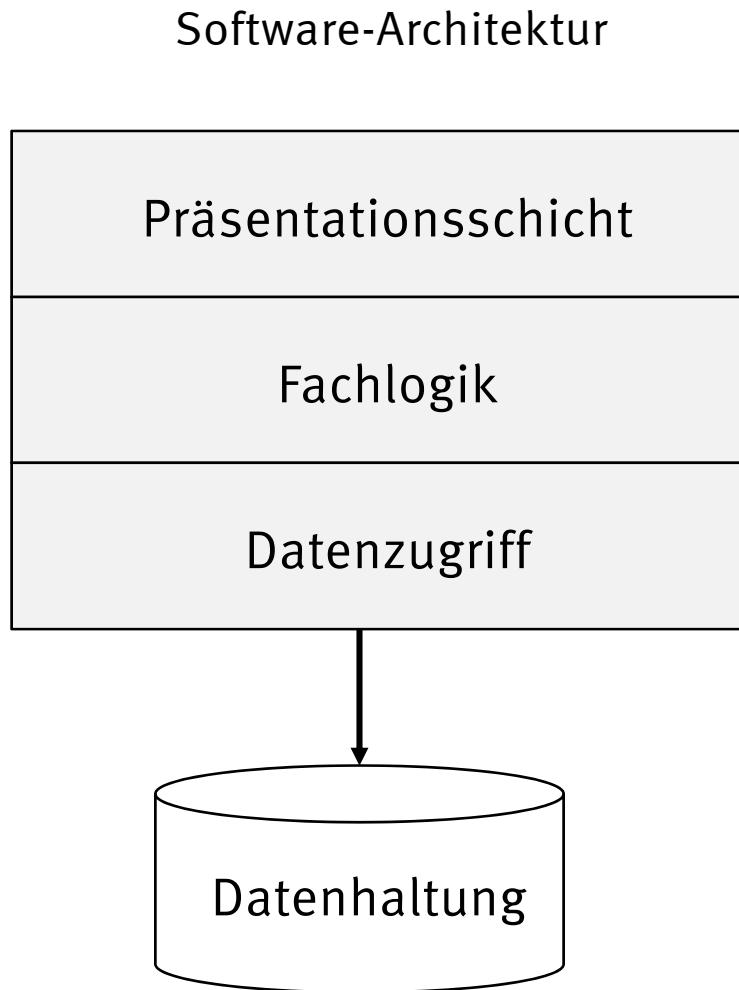
## Wiederholungs-Fragen:

- » Was ist die Grundidee einer Datenbank?
- » Welche Bestandteile hat eine Datenbank und welche Rolle spielen sie?
- » Wie funktioniert eine Datenbank?



# Datenbanken 1

## Drei-Schichten Architektur eines DBMS



Kundenliste

- Kundendaten anzeigen (GUI)
- Welche Kunden(daten) sollen angezeigt werden
- Auslesen der Kundendaten  
Wie ist die Datei organisiert?
- Speicherung der Daten in einer Datei

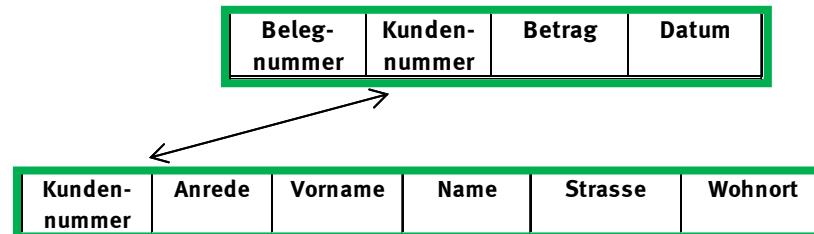
Kunden- nummer	Name	Vorname	Anrede
4812	Meitner	Liese	Frau
8888	Hahn	Otto	Herr

Name	Betrag	Datum
Meitner	44,32	29.09.2008
Hahn	5233,91	29.09.2008

Externe  
Sichten

## Transformationsregeln

Satzstrukturen, Beziehungen, Regeln



Logische  
Gesamtsicht

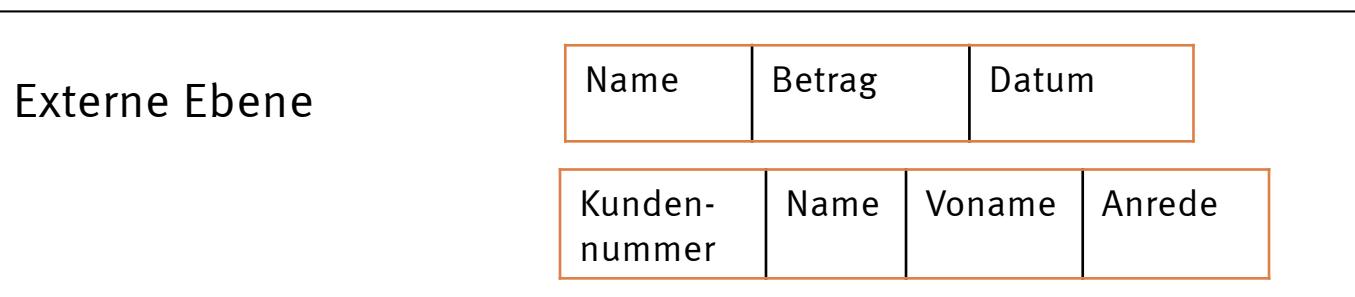
## Transformationsregeln

Phys. Speicherstrukturen  
Zugriffspfade

Datenbasen

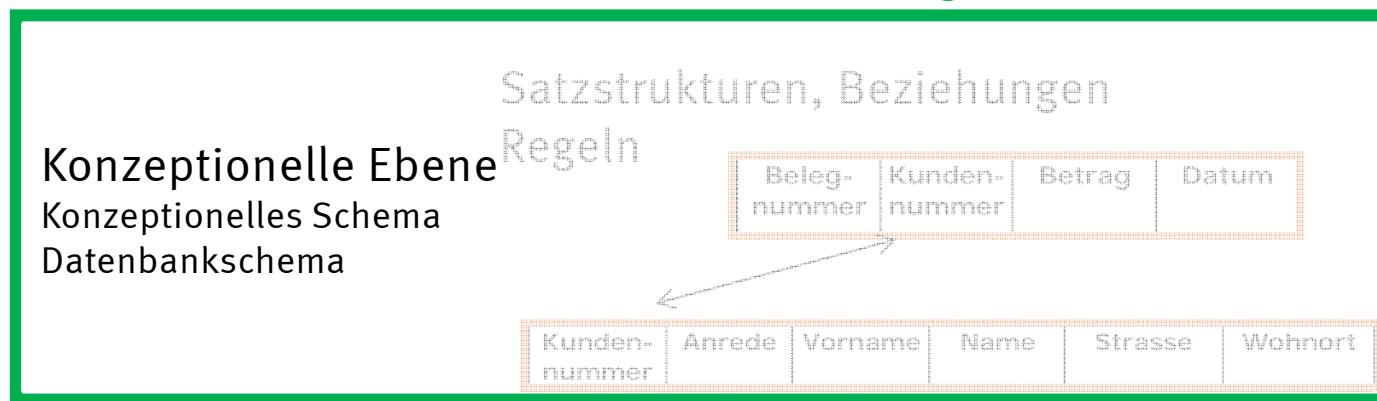
Interne Sicht

## Die Datenebenen kategorisieren die Schemaobjekte



Externe  
Sichten

## Transformationsregeln



Logische  
Gesamtsicht



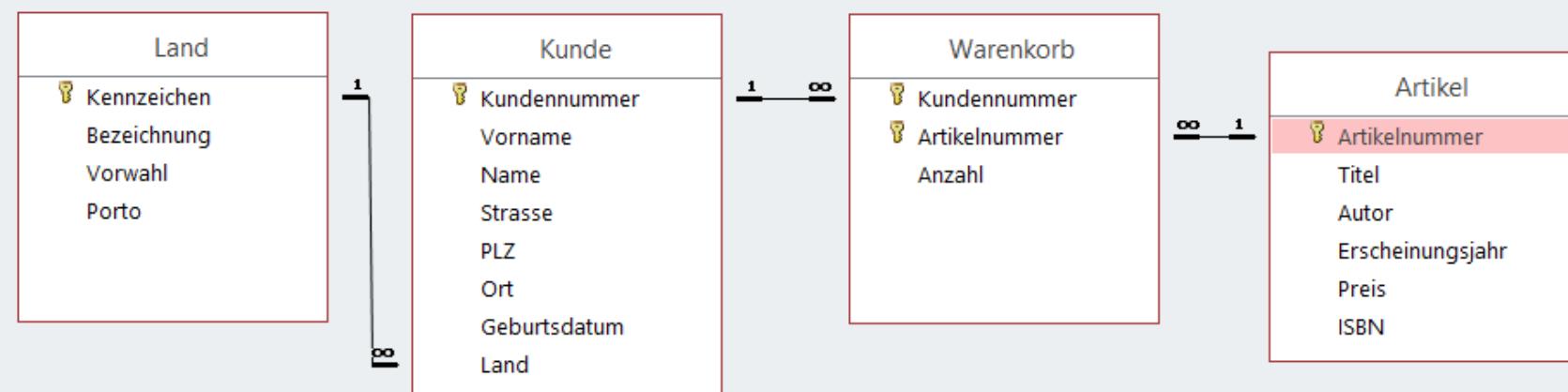
Interne Sicht

Konzeptionelles Schema umfasst das **gesamte** Datenbankmodell in einer für das DBMS verständlichen Form.

Konzeptionelle Ebene  
Konzeptionelles  
Schema

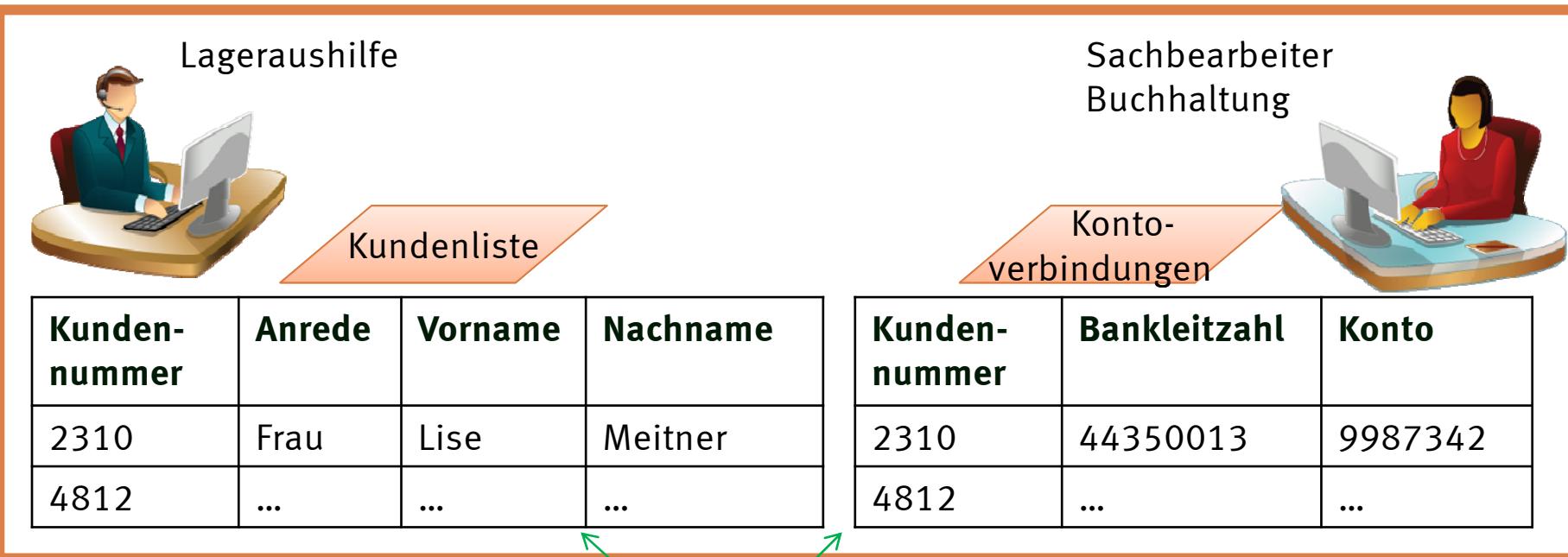
Logische  
Gesamtsicht

Beispiel: Tabellendefinition und Beziehungsübersicht in MS Access



# Logische und externe (Benutzer-)Sichten

Externe Sichten (=relevante Datensicht nach außen)

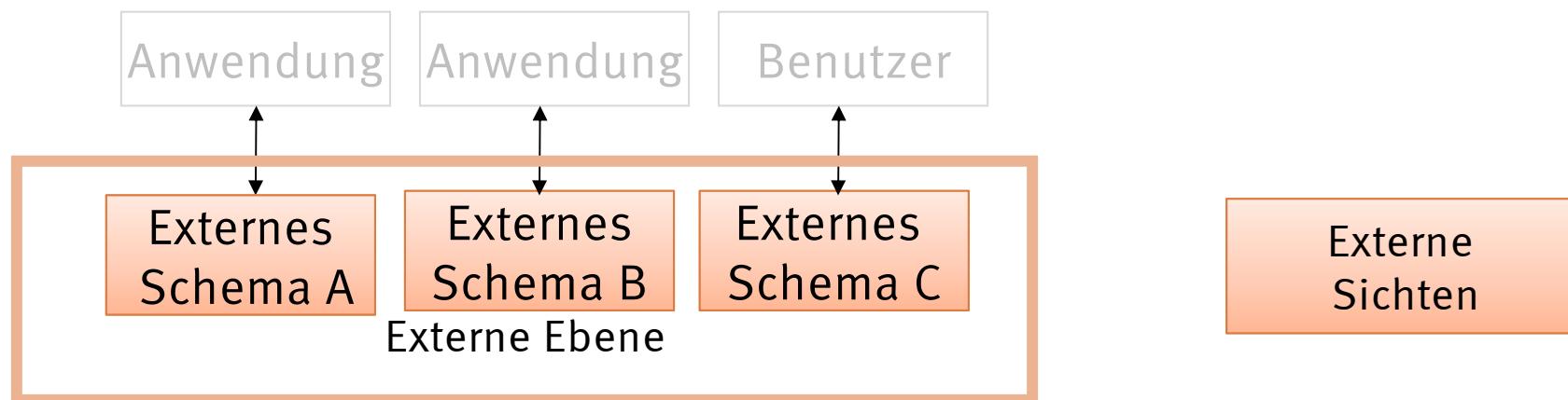


Logische Gesamtsicht

Kunden- nummer	Anrede	Vorname	Nachname	Bankleitzahl	Konto
2310	Frau	Lise	Meitner	44350013	9987342
4812	...	...	...	...	...

Sichtdefinition

Die **externe Ebene** umfasst alle Schemaobjekte, welche die Datensicht nach außen beschreiben. Hierzu gehören Benutzersichten (=gespeicherte Anfragen) und die Definition von Zugriffsrechten.



Beispiel: Gespeicherte Anfragen und Formulare in Access



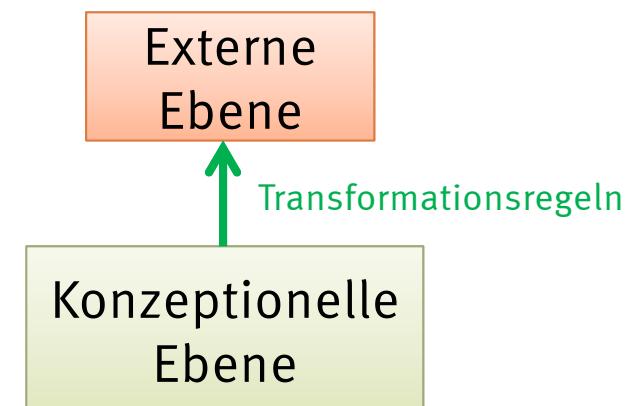
Isolierung von Anwendungsprogrammen vor Änderungen des **konzeptionellen** Modells.

- Konsequenzen:

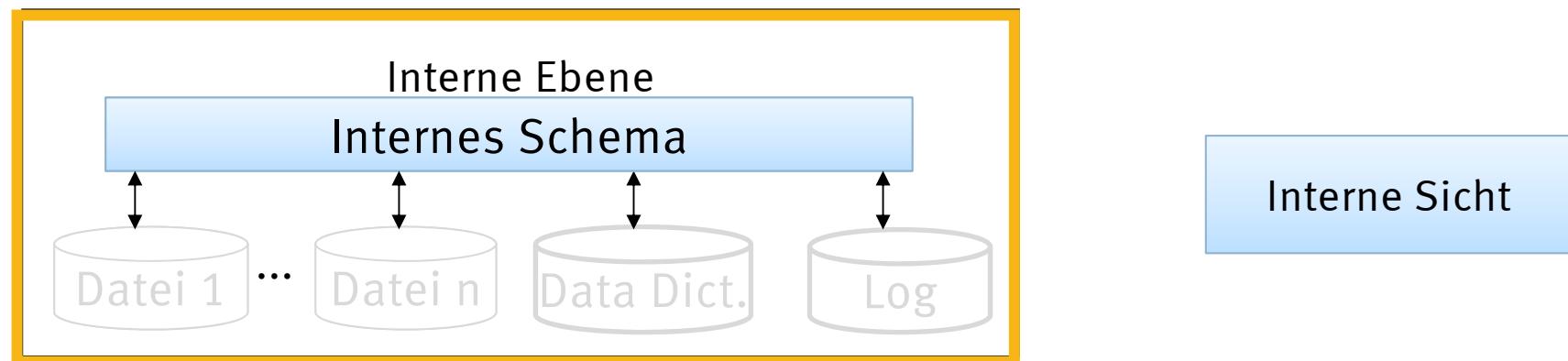
- Anwendungsprogramm muss konzeptionelles Modell nicht kennen
- Änderung des konzeptionellen Modells ist möglich, ohne dass bestehende Programme geändert werden müssen

- Realisierung in der 3-Schichten Architektur:

- Definition eines externen Schemas
- Transformationsregeln kapseln Änderungen am konzeptionellen Schema



Das interne Schema umfasst die Datenbankobjekte, welche die Dateiorganisation, die Zugriffsmethoden und die Zugriffspfade (Indexe, Verkettungen, ...) beschreiben.

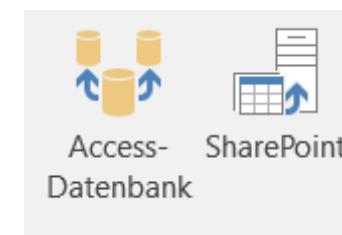


Beispiele:

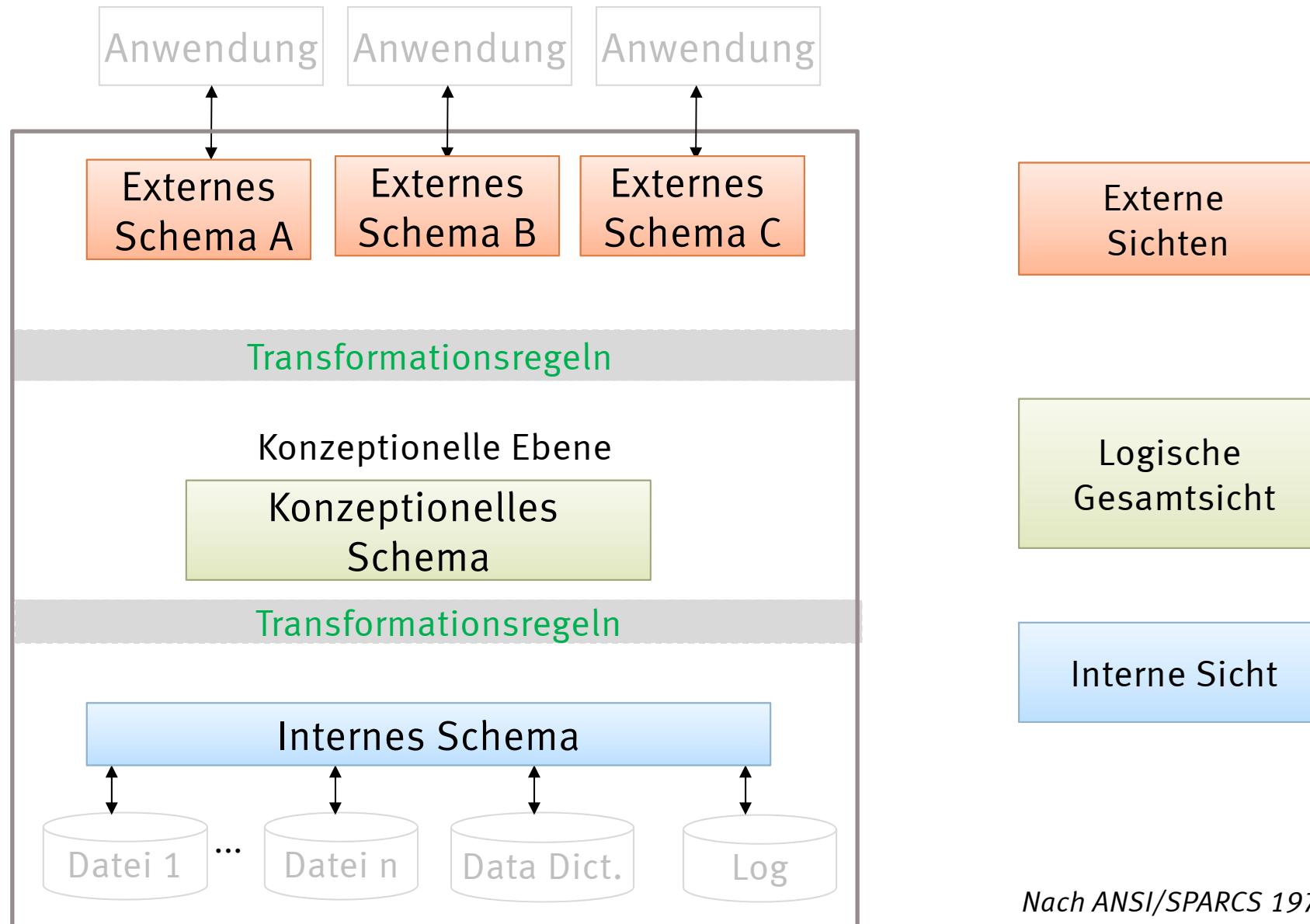
## Index-Erstellung

Allgemein	Nachschlagen
Feldgröße	Long Integer
Format	
Dezimalstellenanzeige	Automatisch
Eingabeformat	
Beschriftung	
Standardwert	0
Gültigkeitsregel	
Gültigkeitsmeldung	
Eingabe erforderlich	Ja
Indiziert	Ja (Ohne Duplikate)
Textausrichtung	Standard

Aufteilung in eine Backend (Daten-)Datei und eine Frontend (=Gui) Datei



# 3-Ebenen Schemaarchitektur



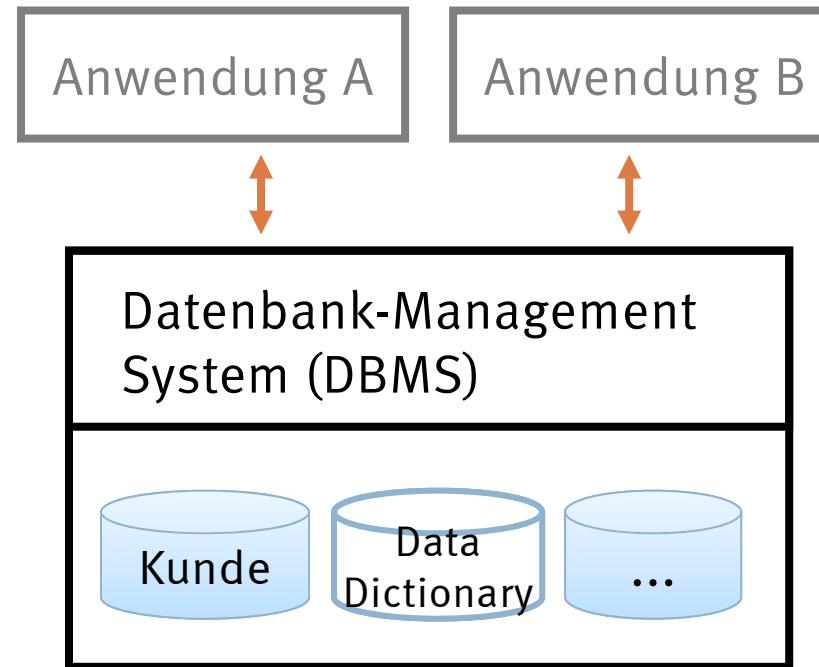
Nach ANSI/SPARCS 1978

we  
focus  
on  
students



# Drei-Schichten Architektur

## Datenbankzugriff



Wie kommunizieren die Anwendungen mit dem DBMS?

ORACLE®

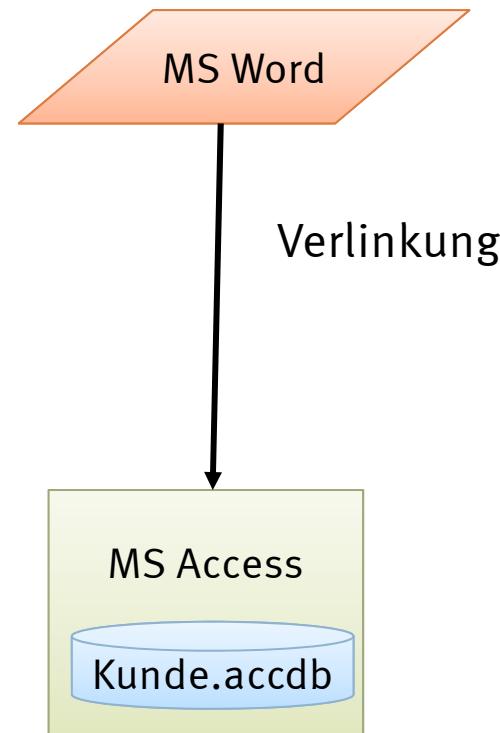
MySQL™

MariaDB®  
FOUNDATION



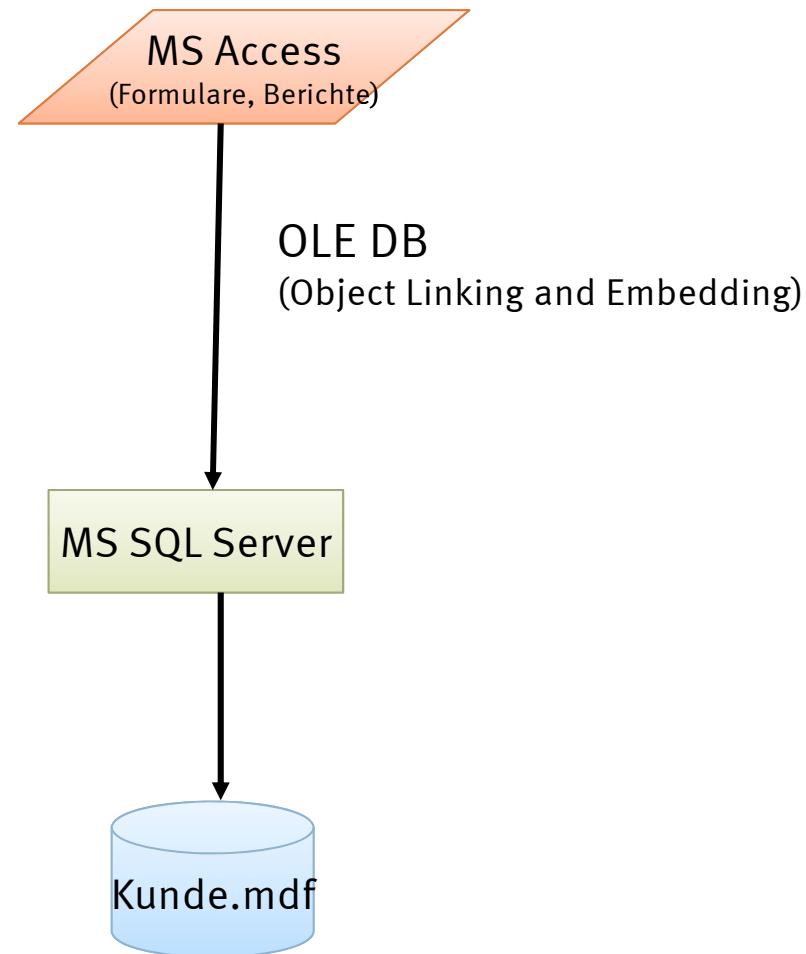
A  
Microsoft®  
SQL Server

	
Unternehmen	Microsoft Erste Version 1992
Lizenz	Kommerziell
GUI	integriert Formulare, Berichte

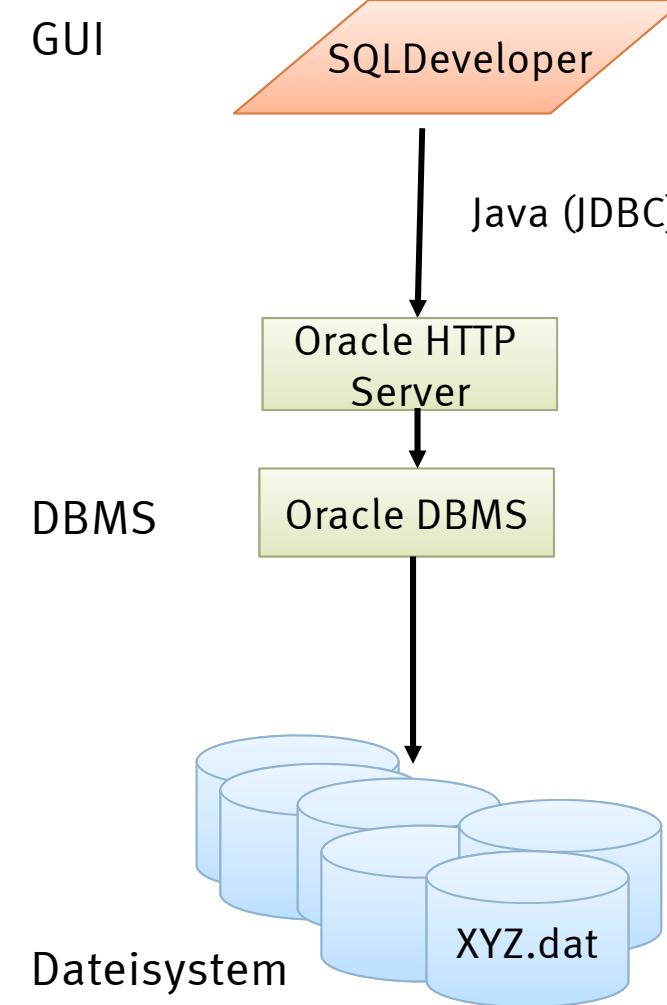


# MS Access als Frontend

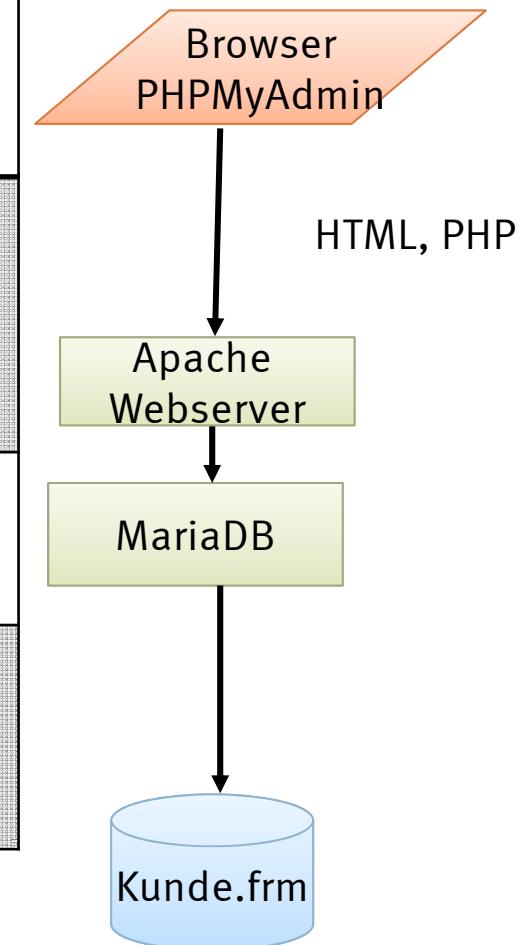
	
Unternehmen	Microsoft Erste Version 1989
Lizenz	Kommerziell
GUI	z.B. MS Access 



<b>ORACLE</b>	
Unternehmen	Gegründet 1977
Lizenz	Kommerziell
Version	Oracle Version 19c Express Edition (OTN)
GUI	SQLDeveloper 



		
Unternehmen	1995 gegründet 2008 von Sun übernommen 2010 von Oracle gekauft	MariaDB ist ein Fork der MySQL-DBMS (ab 2010) *
Lizenz	GPL (Community Version) und Kommerziell	GNU General Public Licence
GUI	MySQL Workbench oder MySQL Query Browser	PHPMyAdmin, MySQL Workbench oder MySQL Query Browser

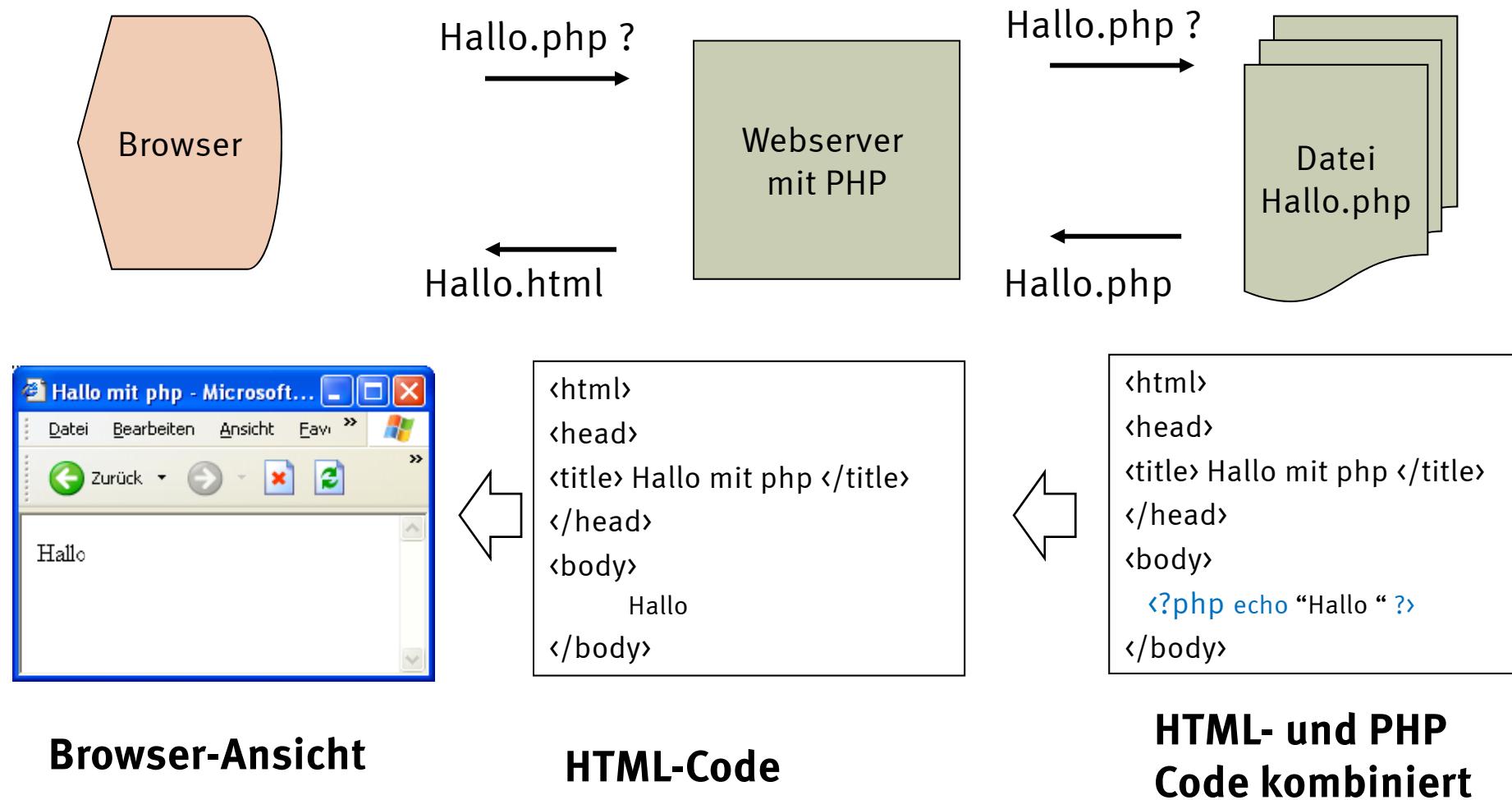


\*) Daten- und Tabellen-Definitionsdateien (.frm) sind kompatibel zu der entsprechenden MySQL-Version

The screenshot shows the phpMyAdmin configuration interface. The left sidebar lists databases: Neu, buchhandlung, information\_schema, mysql, performance\_schema, phpmyadmin, and test. The main content area is divided into several sections:

- Allgemeine Einstellungen**: Zeichensatz/Kollation der MySQL-Verbindung: utf8mb4\_unicode\_ci.
- Anzeige-Einstellungen**: Sprache - Language: Deutsch - German; Design: pmahomme; Schriftgröße: 82%.
- Datenbank-Server**: Server: 127.0.0.1 via TCP/IP, Server-Typ: MariaDB, Server-Version: 10.1.13-MariaDB - mariadb.org binary distribution, Protokoll-Version: 10, Benutzer: root@localhost, Server-Zeichensatz: UTF-8 Unicode (utf8).
- Webserver**: Apache/2.4.17 (Win32) OpenSSL/1.0.2d PHP/5.6.21, Datenbank-Client Version: libmysql - mysqld 5.0.11-dev - 20120503 - \$Id: 76b08b24596e12d4553bd41fc93cccd5bac2fe7a \$, PHP-Erweiterung: mysqli, PHP-Version: 5.6.21.
- phpMyAdmin**: Versionsinformationen: 4.5.1, aktuelle stabile Version: 4.6.4, Dokumentation, Wiki, Offizielle Homepage, Mitmachen, Unterstützung erhalten, Liste der Änderungen.

# Prinzip PHP-Webserver



- Java

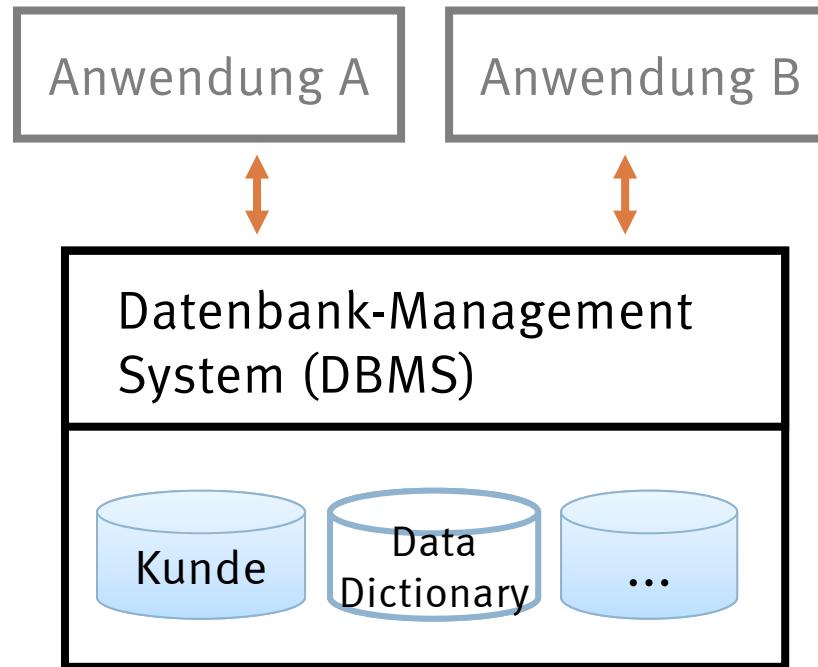
```
Connection con=
```

```
    DriverManager.getConnection ("localhost", "username", "password");
```

- PHP

```
<?php
```

```
    $conn = new mysqli("localhost", "username", "password");  
?>
```



Wie kommunizieren die Anwendungen mit dem DBMS?

- Direkte Verlinkung
- OLE
- JDBC
- PHP



we  
focus  
on  
students



# Datenbankadministration

## Benutzerverwaltung

**Fachhochschule  
Dortmund**

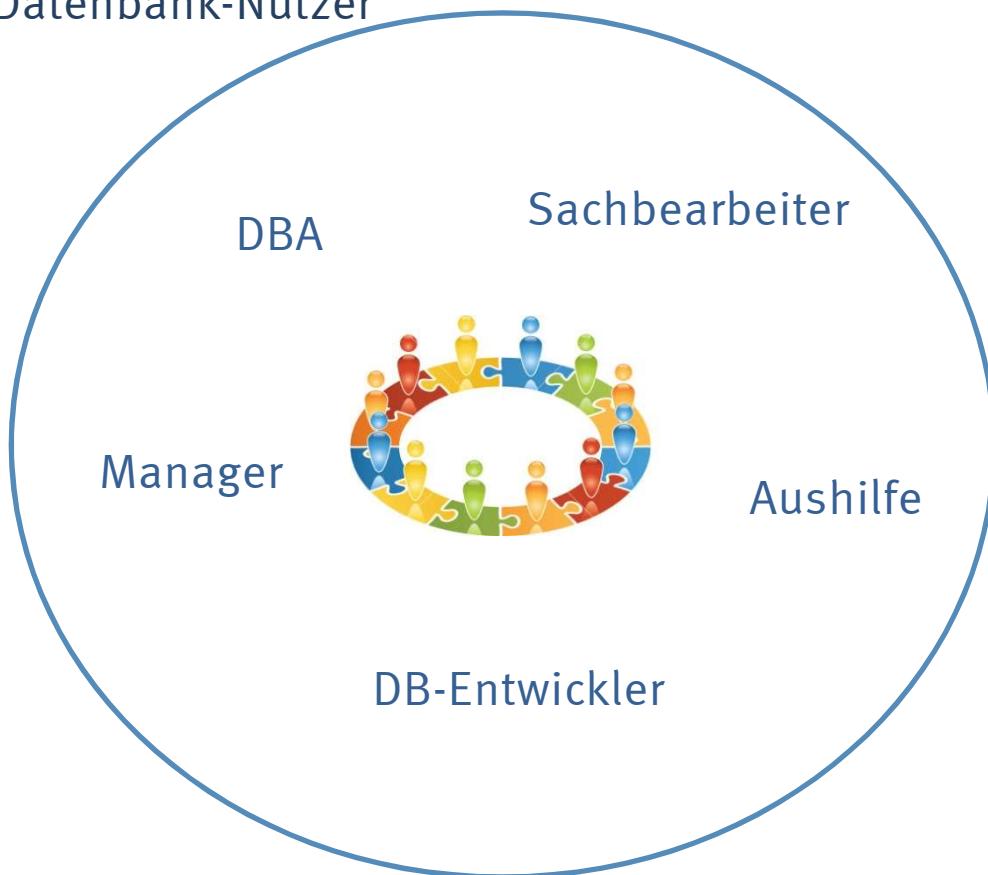
University of Applied Sciences

© 2020 - Prof. Dr. Inga Marina Saatz

# Datenbankbenutzer

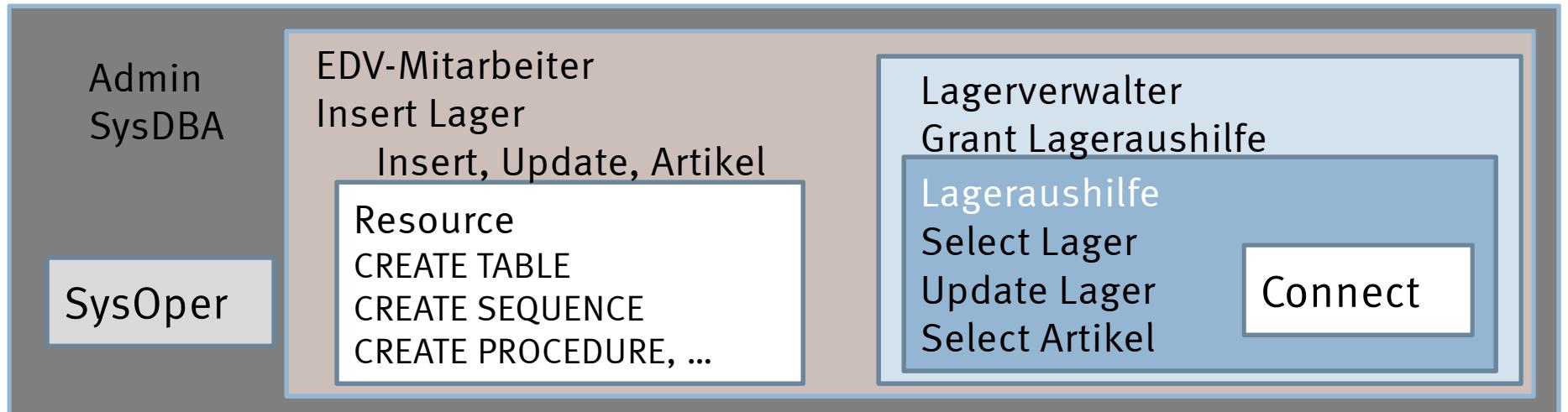
Der Datenzugriff erfolgt nur über das DBMS und das DBMS kontrolliert die Benutzerberechtigungen bei jedem Zugriff. Die Zugriffsrechte werden im Data Dictionary gespeichert.

## Datenbank-Nutzer



- ▶ Datenbankentwickler
  - ▶ Systemanalyse
  - ▶ Anwendungsentwicklung
  - ▶ Beantwortung von Ad-hoc-Anfragen
- ▶ Datenbankadministratoren (DBA)
  - ▶ Datenbankdesign
  - ▶ Verwaltung
  - ▶ Überwachung
  - ▶ Performance-Tuning

- Oracle
  - Rollen
    - eine Rolle ist eine benannte Sammlung von Einzelrechten
    - Rollen lassen sich Nutzern zuweisen
    - Rollen können auch an andere Rollen vergeben werden, dadurch lassen sich Rechte-Hierarchien schaffen
    - Beispiel für eine Rechte-Hierarchie mit unterschiedlichen Rollen:



# Oracle: Benutzer anlegen

- Anlegen eines neuen Benutzers (Common User)

```
CREATE USER C##myUSER
  IDENTIFIED BY geheim
  ACCOUNT UNLOCK          // nicht gesperrt
  CONTAINER=ALL;
ALTER USER C##Saatz quota 100M on USERS;
```

- Benutzer löschen:

```
DROP USER C##myUSER
```

# Objektprivilegien

- Vergabe von Objekt-Privilegien

```
GRANT {<system_priv>|<role>},...  
TO {<user>|<role>|public},...  
[WITH GRANT OPTION]
```

- Beispiel

```
GRANT CREATE TABLE  
TO C##myUSER
```

# Rechtevergabe (Oracle)

## Beispiele für benötigte Rechte

Bereich	Beispiel	Bemerkung
DB-Schema anlegen	GRANT CREATE TABLE TO ...; GRANT CREATE SEQUENCE TO ... ; GRANT SELECT ON ... TO ...;	Tabellen erzeugen Sequence erstellen Lesezugriff erteilen
Benutzer administrieren	GRANT CREATE TABLESPACE TO ...; GRANT CONNECT TO ...;  GRANT SELECT ANY DICTIONARY TO ...;	Tablespace erstellen Berechtigung, um sich mit der DB zu verbinden Zugriff auf das Data Dictionary
DB-Programme erstellen	GRANT CREATE TRIGGER TO ...; GRANT CREATE PROCEDURE TO ...; GRANT EXECUTE ON ... TO ...;	Prozeduren erstellen Trigger erstellen Prozedur ausführen

# Systemprivilegien

## Vergabe von System-Privilegien

```
GRANT {<system_priv>}|<role>},...  
TO {<user>}|<role>|public},...  
[WITH ADMIN OPTION]
```

*Weitergegebenes Recht bleibt erhalten,  
wenn dem Rechtgeber das Recht entzogen wird*

Oracle	Beschreibung
SYSDBA	Recht, Tabellen anderer Benutzer einzusehen
SYSOPER	DB hoch/runter fahren
CONNECT	Recht, sich mit der Datenbank zu verbinden
RESOURCE	Recht, Tabellen oder Indizes zu erstellen

# Entzug von Rechten

Prof. Dr. I.M. Saatz

Datenbanken 1

Fachbereich Informatik

8

- Entzug von Rechten

```
REVOKE priv_type [(column_list)],...
ON [object_type] {tbl_name | * | *.* | db_name.*}
FROM <user>
```

- Entzug aller Rechte

```
REVOKE ALL PRIVILEGES, GRANT OPTION
FROM <user>
```

- Beispiel

```
REVOKE CREATE TABLE FROM C##myUSER
```

# Zusammenfassung

Prof. Dr. I.M. Saatz

Datenbanken 1

Fachbereich Informatik

9

- Nutzer anlegen und löschen
  - CREATE USER
  - DROP USER
- Rechte vergeben und entziehen
  - GRANT
  - REVOKE
  - Admin-Option
  - Grant-Option