



we
focus
on
students

Datenbanken 1

Datenbankarchitektur

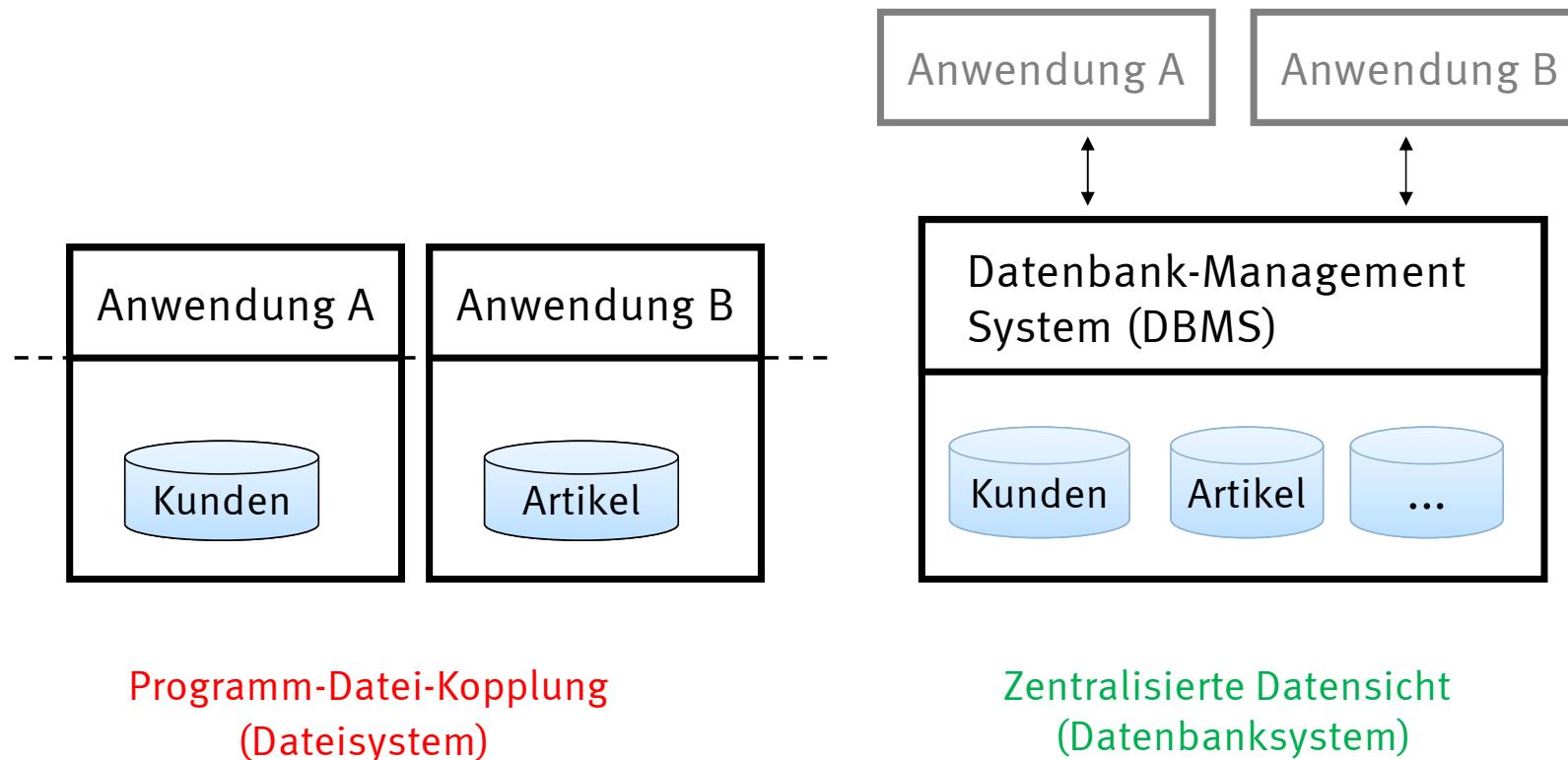
1	Was ist eine Datenbank?	2
2	Wie werden Daten relational gespeichert?	7
3	Wie ist eine (relationale) Datenbank aufgebaut?	19
4	Datenbankzugriff und Benutzerverwaltung	32
5	Wochenaufgaben	37

Was ist eine Datenbank?



Lösungsidee - Datenbankkonzept

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.

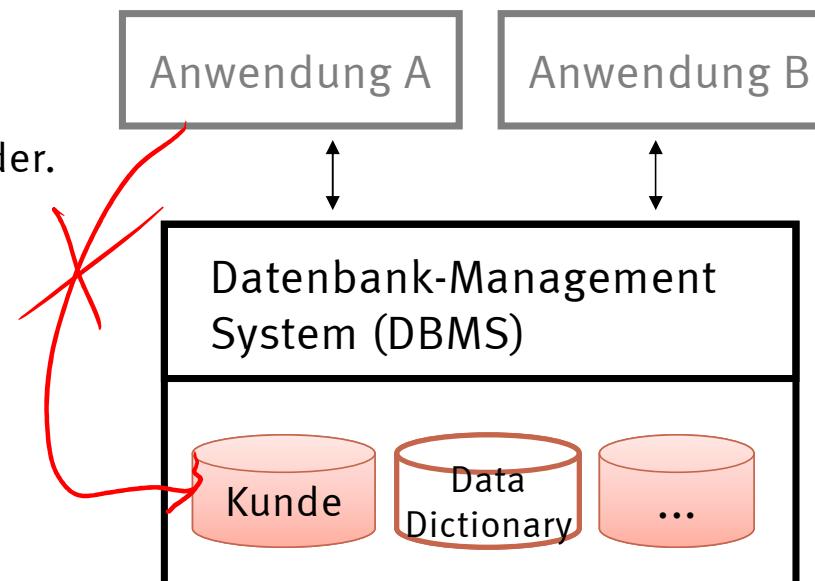
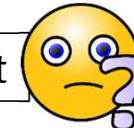


Datenbank-Managementsystem (DBMS)

Ein DBMS ...

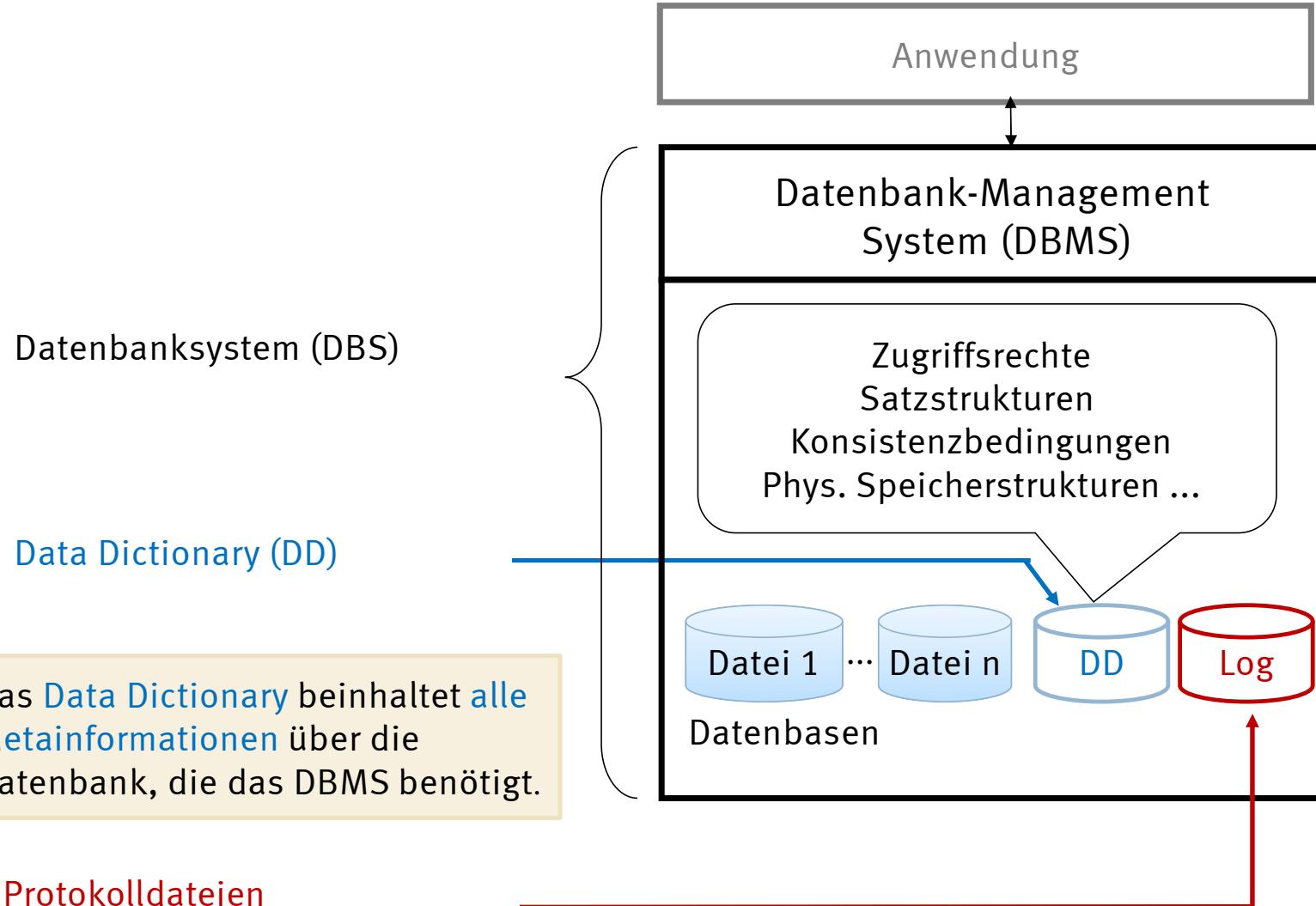
- ▶ ... ist ein Softwaresystem zur Datenverwaltung
- ▶ ... stellt Funktionen bereit zum Einfügen, Löschen, Ändern, Sichern und Suchen von Daten.
- ▶ ... bietet eine Schnittstelle zum Anwender. Der Zugriff auf die Datenbank erfolgt ausschließlich über das DBMS
- ▶ ... kontrolliert die Datenbank.

Wie ist ein DBMS aufgebaut



Zentralisierte Datensicht
(Datenbanksystem)

Begriffe



1	Was ist eine Datenbank?	2
2	Wie werden Daten relational gespeichert?	7
3	Wie ist eine (relationale) Datenbank aufgebaut?	19
4	Datenbankzugriff und Benutzerverwaltung	32
5	Wochenaufgaben	37

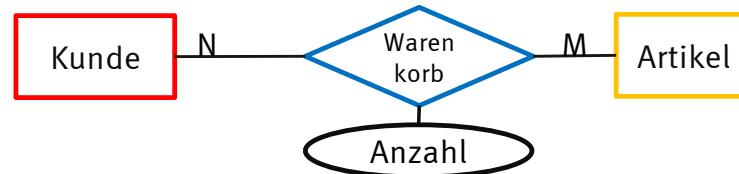
Reale Welt



ER-Modell



Datenbankmodell

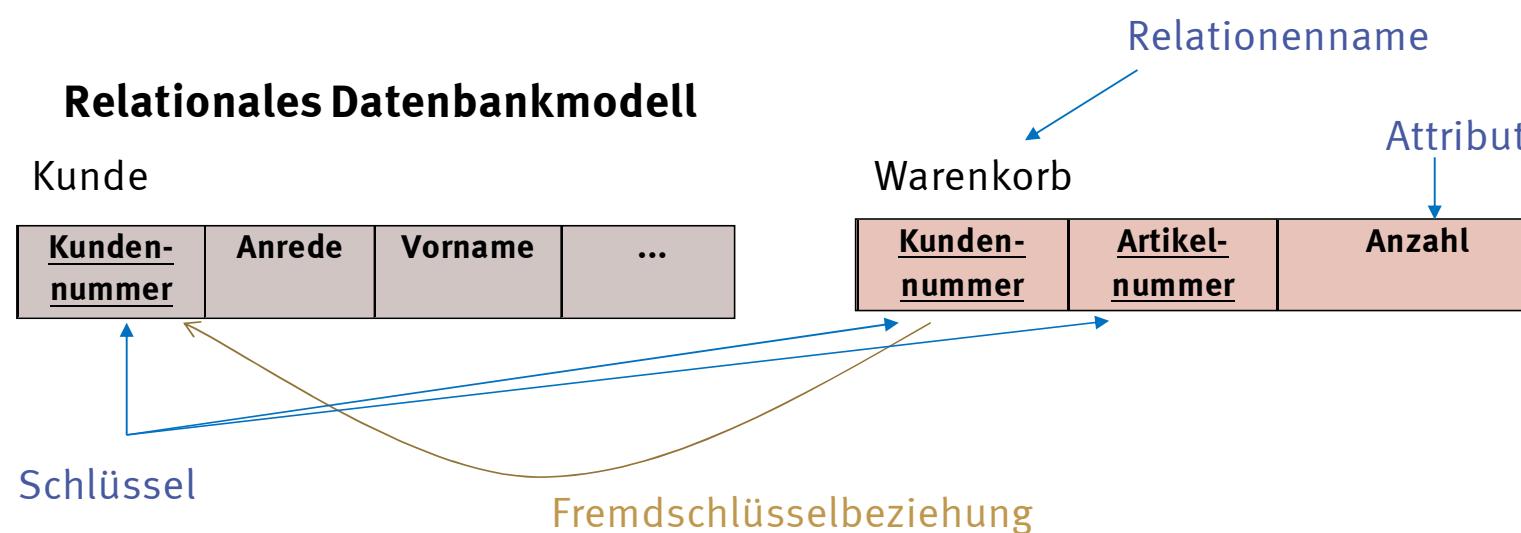


Durch ein **Datenbankmodell** wird beschrieben, wie Daten in einem Datenbanksystem gespeichert und bearbeitet werden können.

1. Datenstruktur zur Speicherung der Datenobjekte
2. Datenoperationen zum Suchen und zur Bearbeitung der Datenobjekte

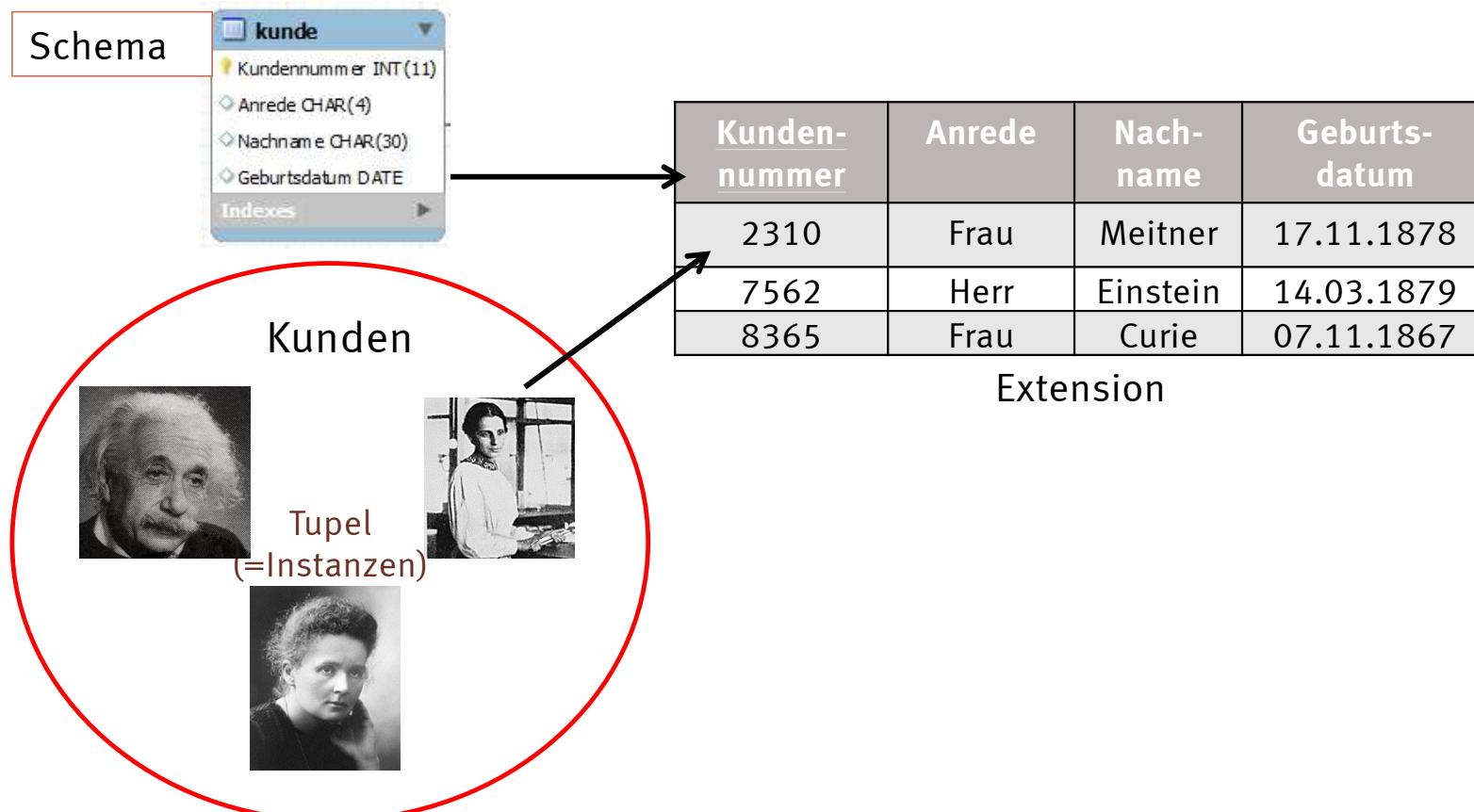
Relationales Datenbankmodell

Durch das **relationale Datenbankmodell** werden alle Datenobjekte des ER-Modells durch Relationen beschrieben.



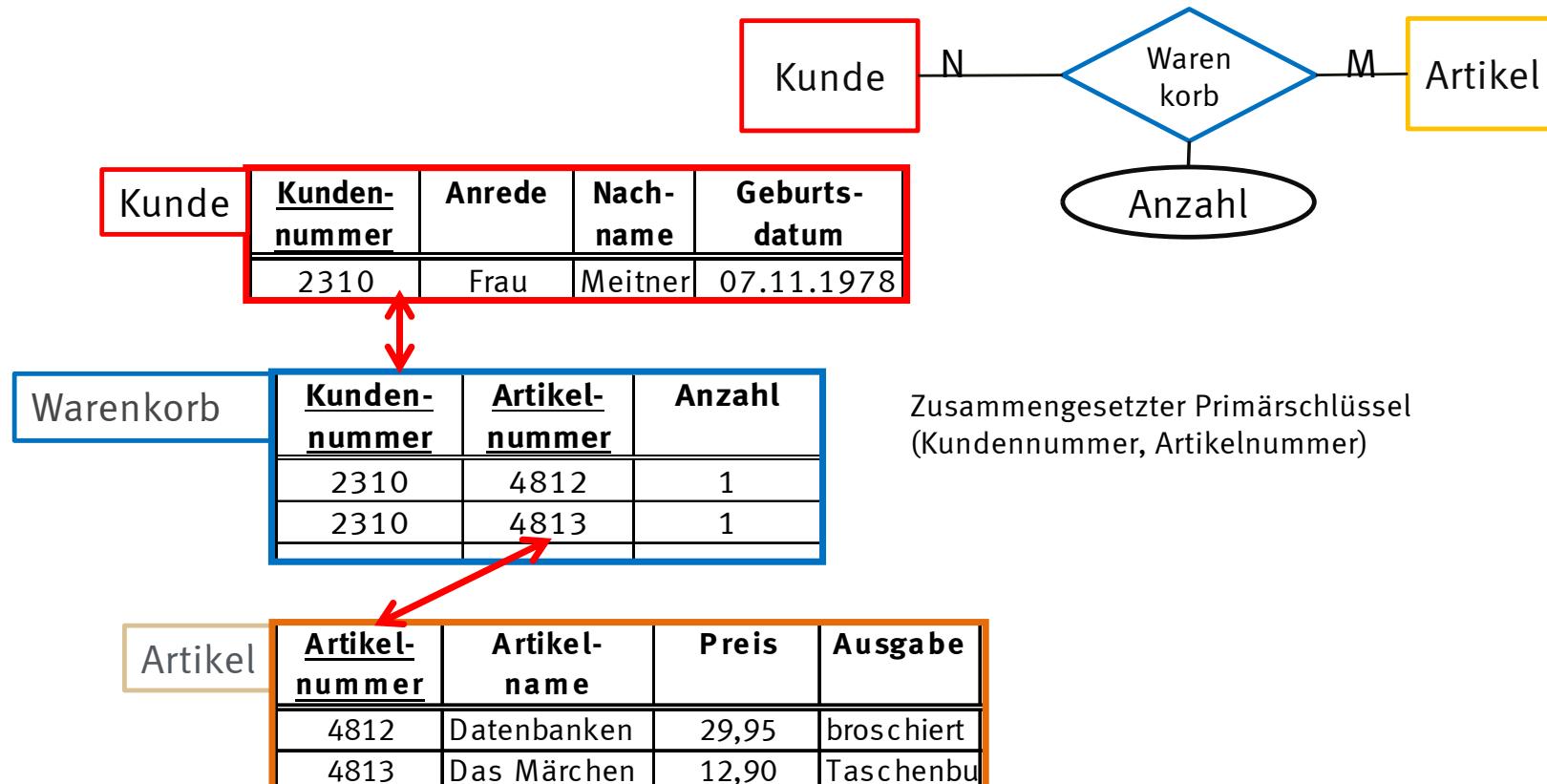
Relationales Datenmodell (T. Codd 1970)

Das Schema (=Tabellenkopf) besteht aus den Attributen der Relation. Ein Tupel stellt eine konkrete Kombination von Attributwerten (=Instanz) dar und wird im Datenbankbereich auch als Datensatz bezeichnet.



Beispiel N-M-Beziehung

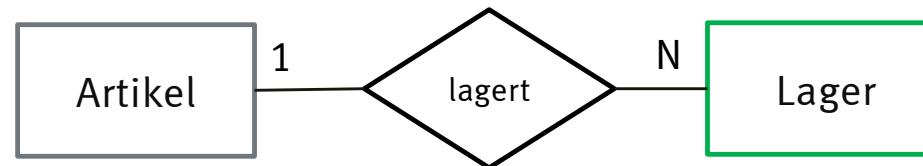
N-M-Beziehungen werden über eine Zwischentabelle (hier: Warenkorb) abgebildet. Der Primärschlüssel der Zwischentabelle wird aus den Fremdschlüsseln der referenzierten Entitäten (hier: Kundennummer & Artikelnummer) gebildet



Darstellung von Beziehungen

Beziehungen werden über **Fremdschlüssel** als **wertmäßige „Zeiger“** abgebildet.

Beispiel: Abbildung einer 1:N-Beziehung über einen Fremdschlüssel



Artikel	Artikel-nummer	Artikel-name	Preis	Ausgabe
	4812	Datenbanken	19,90	broschiert
	4813	Das Märchen	10,00	Taschenbuch

Lager	Lager-nummer	Standort	ANummer	Lager-bestand
	27135	R235	4812	18
	27432	R371	4813	0

Primär-schlüssel

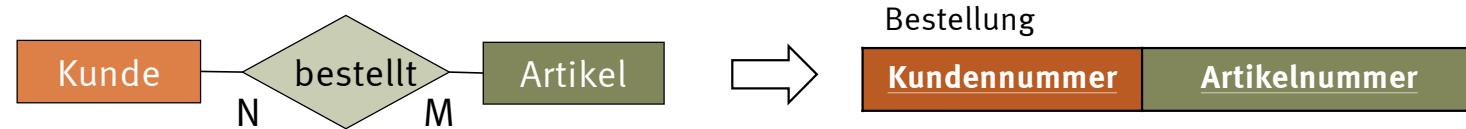
Fremdschlüssel

Möglichkeiten der Abbildung von Beziehungen zwischen Entitäten

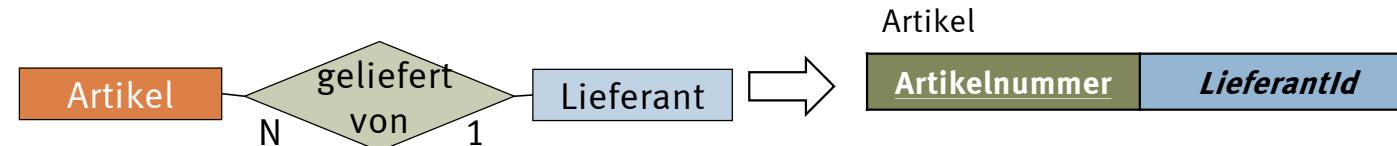
Sowohl Entitäten als **auch** deren Beziehungen (Relationships) werden durch Relationen abgebildet.

Eine Beziehung kann abgebildet werden durch

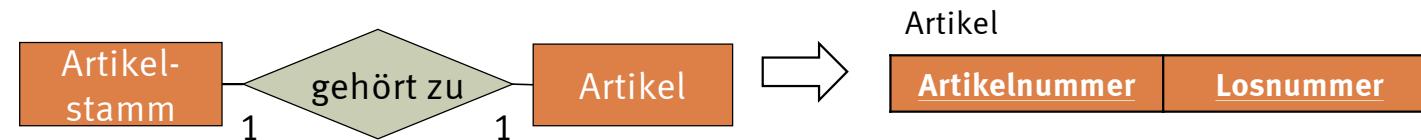
1. eine eigene Tabelle,



2. die Ergänzung eines Fremdschlüssels in der Tabelle der mehrwertigen Entität oder

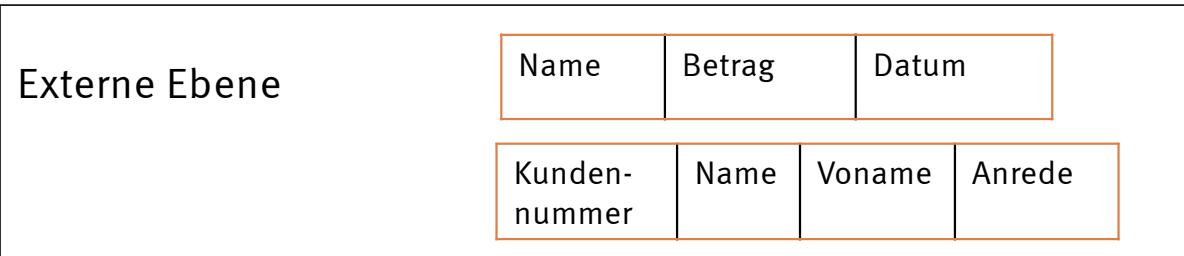


3. die Verschmelzung der Tabellen der beteiligten Entitäten .



1	Was ist eine Datenbank?	2
2	Wie werden Daten relational gespeichert?	7
3	Wie ist eine (relationale) Datenbank aufgebaut?	19
4	Datenbankzugriff und Benutzerverwaltung	32
5	Wochenaufgaben	37

Die Datenebenen kategorisieren die Schemaobjekte



Externe
Sichten

Transformationsregeln



Logische
Gesamtsicht

Transformationsregeln

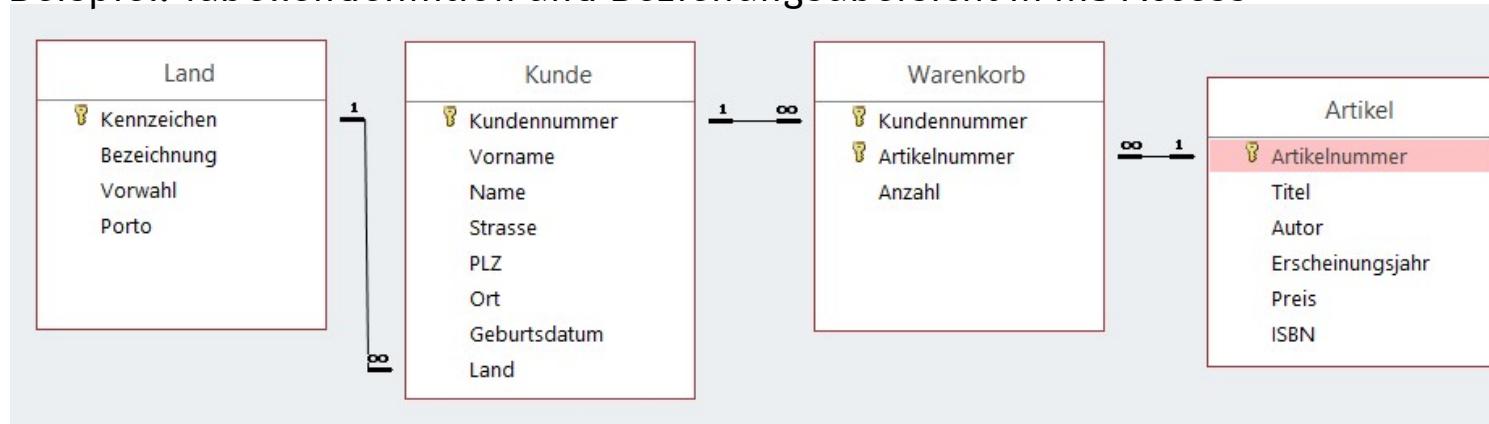


Interne Sicht

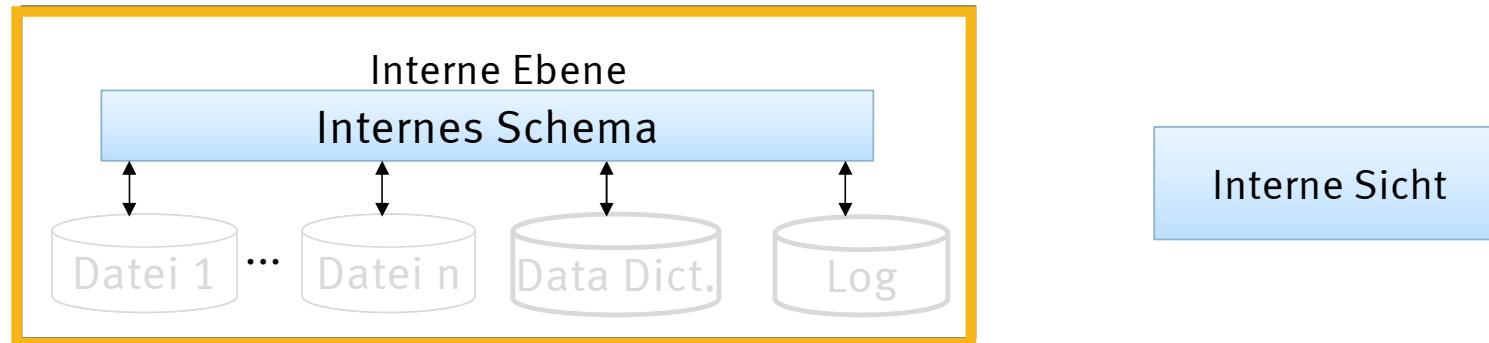
Konzeptionelles Schema umfasst das **gesamte** Datenbankmodell in einer für das DBMS verständlichen Form.



Beispiel: Tabellendefinition und Beziehungsübersicht in MS Access



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



Beispiele:

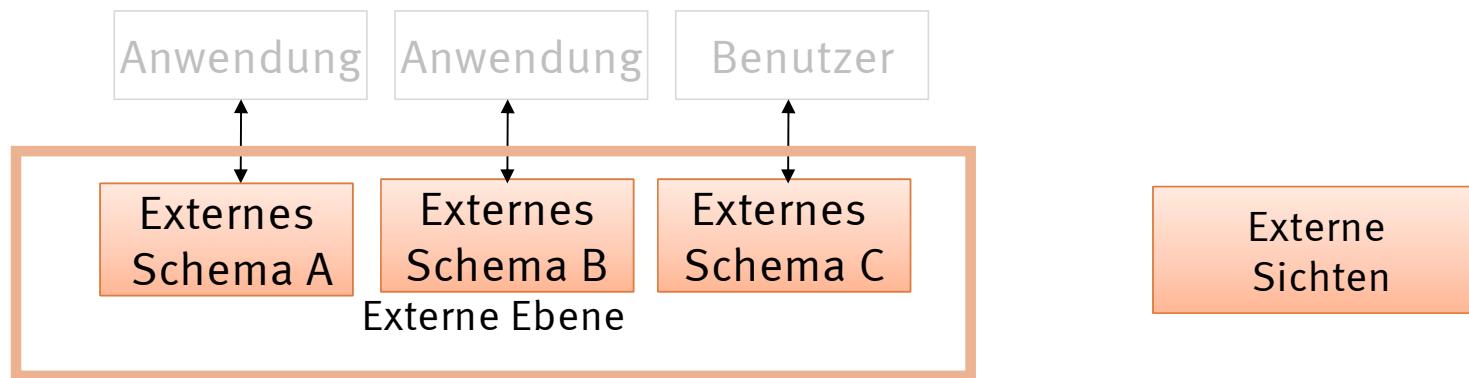
Index-Erstellung

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



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

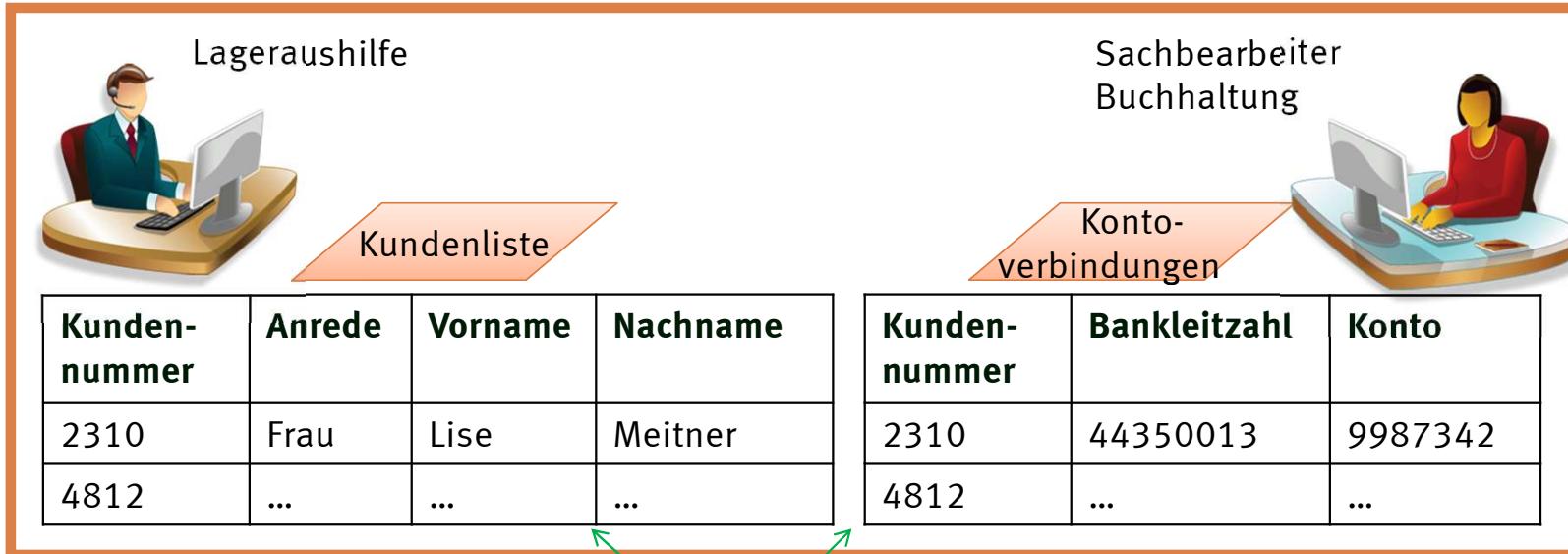


Beispiel: Gespeicherte Anfragen und Formulare in Access

Abfragen	
	Kunde Abfrage
	Warenkorb_Preis
	Warenkorb_Preis_pro_Kunde
Formulare	
	KundeGeteilt

Logische und externe (Benutzer-)Sichten

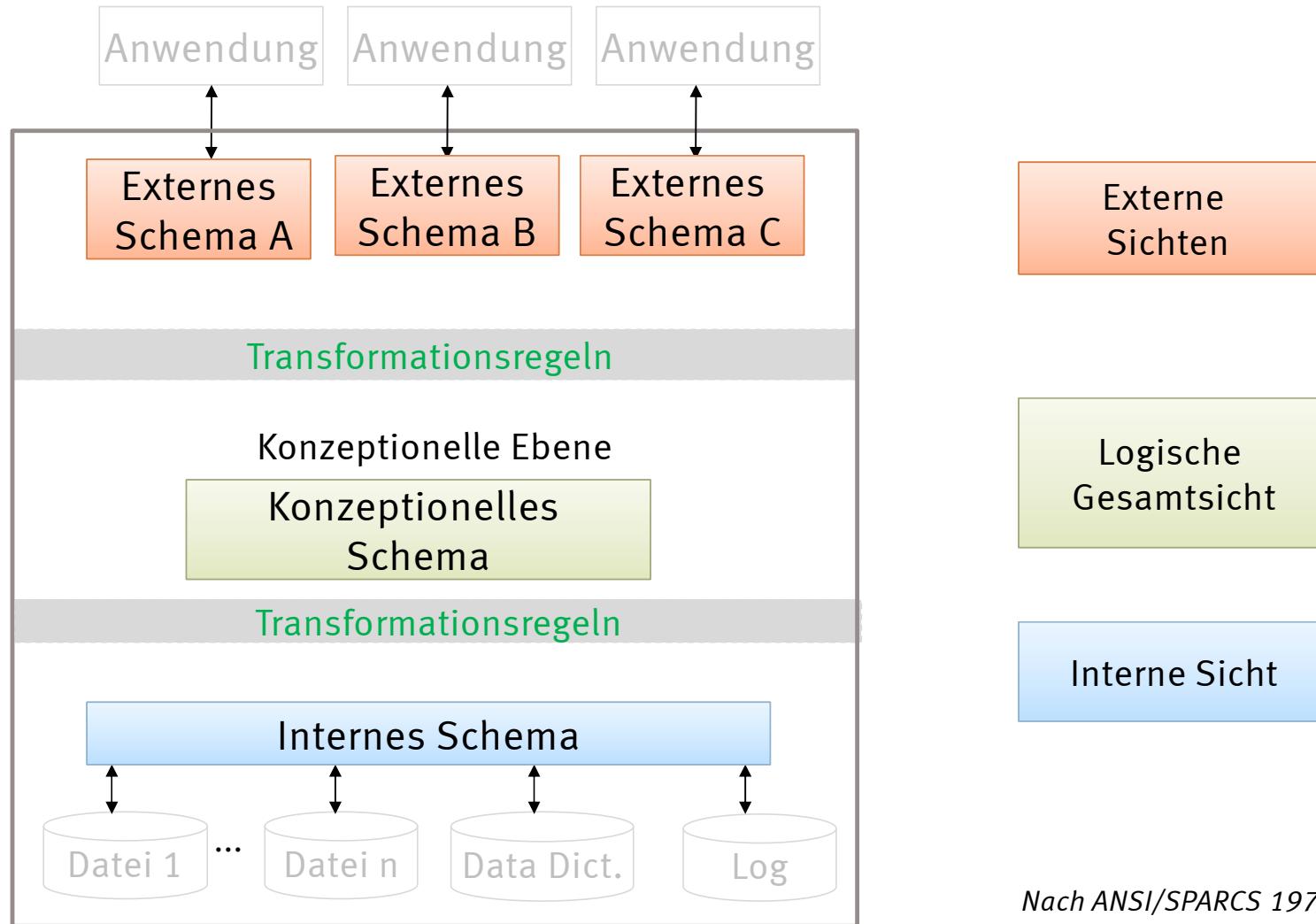
Externe Sichten (=relevante Datensicht nach außen)



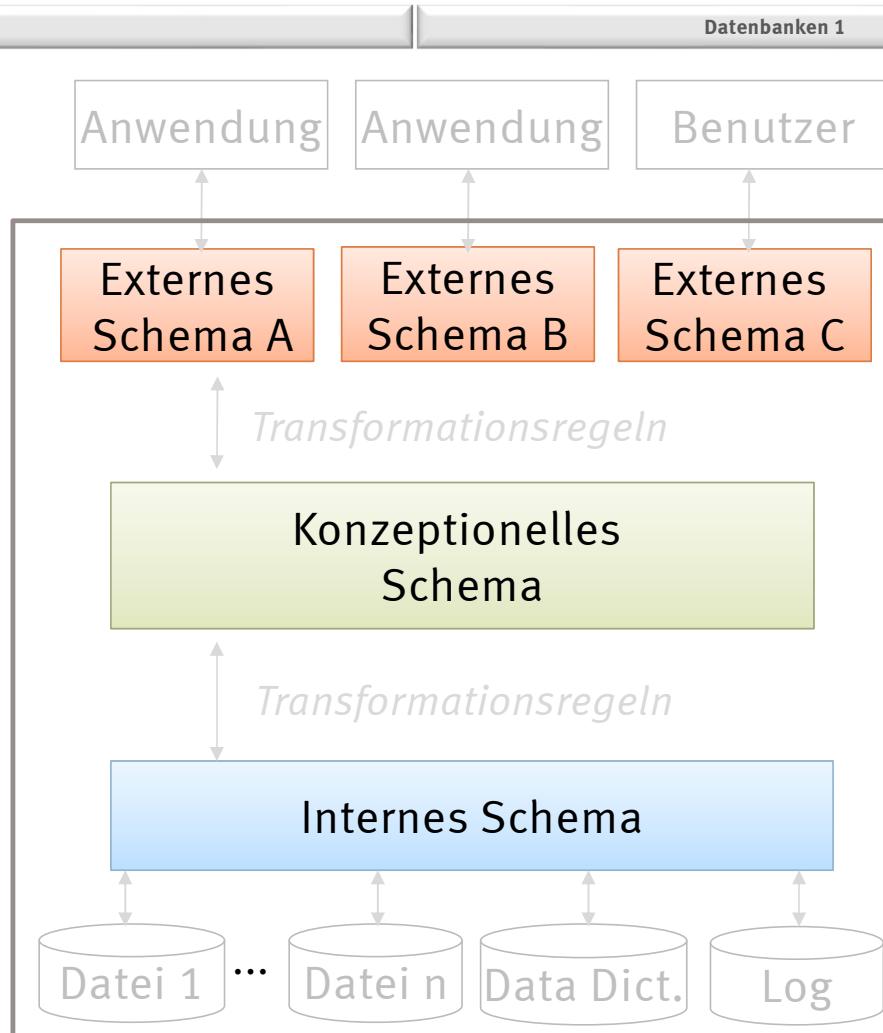
Logische Gesamtsicht

Kundennummer	Anrede	Vorname	Nachname	Bankleitzahl	Konto
2310	Frau	Lise	Meitner	44350013	9987342
4812

3-Ebenen Schemaarchitektur



Nach ANSI/SPARCS 1978



Anm.: ausgegraute Elemente gehören nicht
zur 3-Ebenen Schemaarchitektur

Vorteile eines DBMS

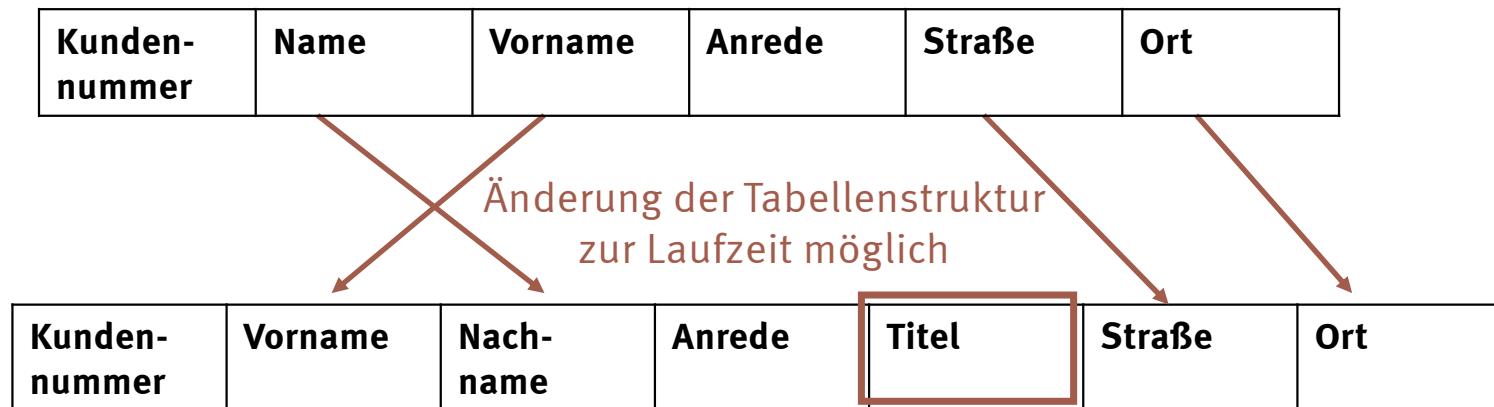
Codd'sche Regeln 1982

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

Überprüfung der Codd'schen Regel 9

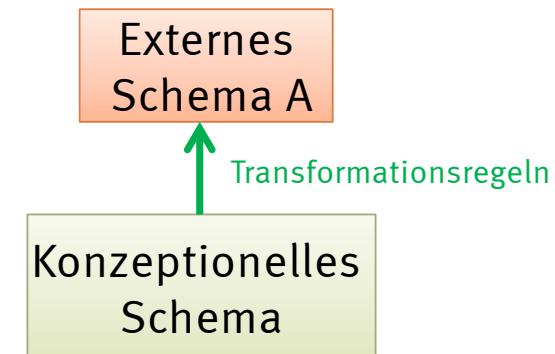
Anforderung:

Logische Datenunabhängigkeit –
Die Tabellenstruktur kann zur Laufzeit geändert werden, ohne dass
Anwendungsprogramme neu kompiliert werden müssen



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



Überprüfung der Codd'schen Regel 10

Anforderung

Physische Datenunabhängigkeit: Unabhängigkeit der Anwendungsprogramme von einer Änderung der Speicherorganisationform. Nur das DBMS greift physikalisch auf die Daten zu.
→ Eine Änderung der physikalischen Speicherung ohne Konsequenzen zur Laufzeit erfolgen.

Änderung der Organisationsform

Sequenziell



Index - Sequenziell

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

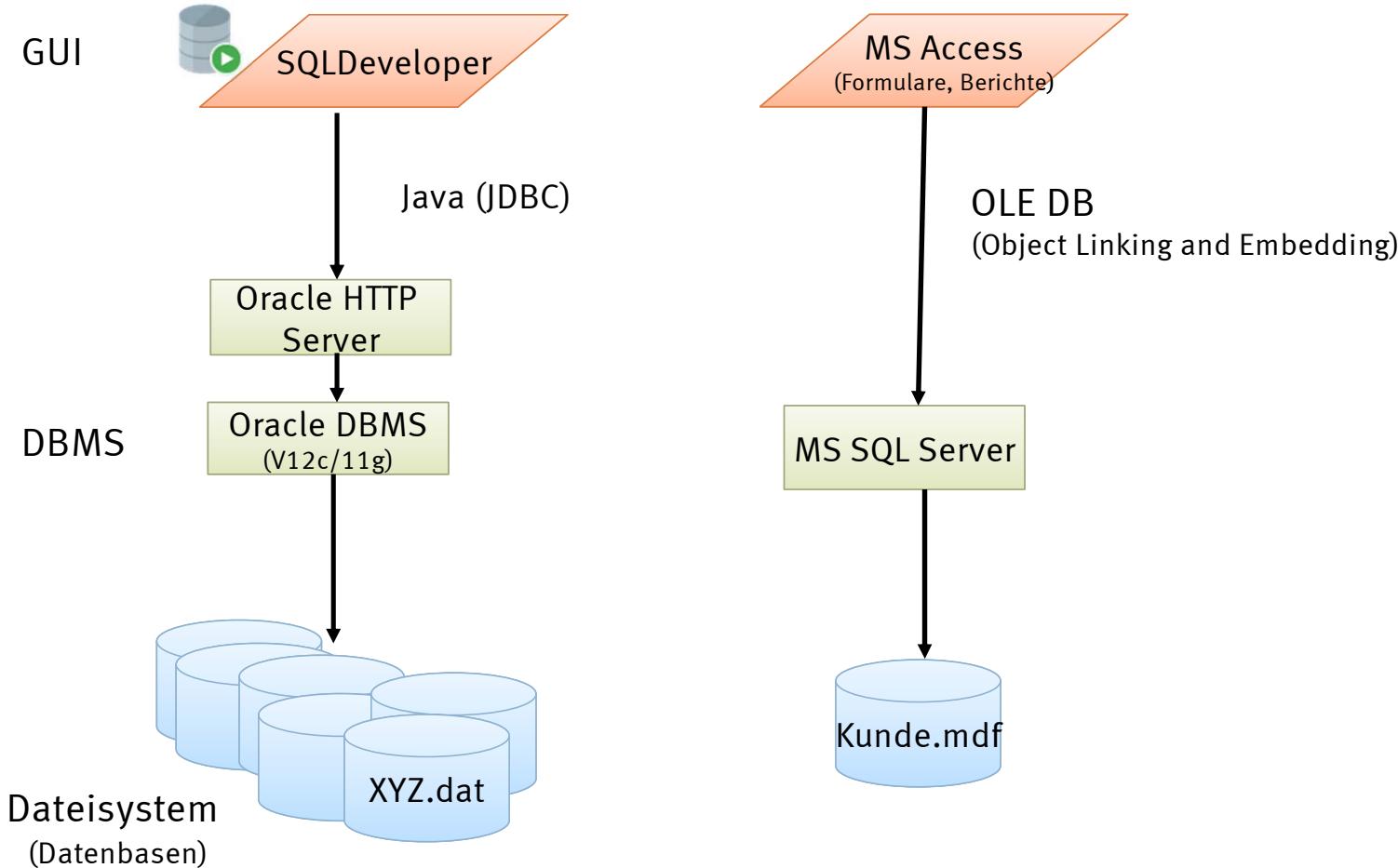
Schlüssel	Adresse
2720	Adresse
4812	Adresse
...	...

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

1	Was ist eine Datenbank?	2
2	Wie werden Daten relational gespeichert?	7
3	Wie ist eine (relationale) Datenbank aufgebaut?	19
4	Datenbankzugriff und Benutzerverwaltung	32
5	Wochenaufgaben	37

Zugriff auf Datenbanken (Beispiele)

Neben dem Zugriff über die Konsole können die Datenbanken bspw. über die JDBC- oder OLE DB-Schnittstellen angesprochen werden.



Oracle: Benutzer anlegen

Installation → Hinweise in ILIAS

Oracle 12c (mit dem bei der Installation angelegten Nutzer SYS, SYSTEM)

- Anlegen eines neuen Benutzers (Common User)

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

- Benutzer löschen:

```
DROP USER C##SAATZ
```

- Benutzerrechte vergeben

- Hier ist das Skript in ILIAS entsprechend anzupassen:
- GRANT CONNECT TO C##SAATZ;
- GRANT CREATE TABLE TO C##SAATZ;
- usw.

1	Was ist eine Datenbank?	2
2	Wie werden Daten relational gespeichert?	7
3	Wie ist eine (relationale) Datenbank aufgebaut?	19
4	Datenbankzugriff und Benutzerverwaltung	32
5	Wochenaufgaben	37

- Wochenaufgaben
 - Lernmodule Relationales Modell sowie Datenbankarchitektur
 - Praktikum
 - Relationales Modell
 - Datenbankzugriff (Praktikum)
 - Installation Oracle SQLDeveloper
 - Zugriff auf die Oracle-Praktikumsdatenbank herstellen.
- Möglichkeiten:
1. vpn und Oracle SQLDeveloper **oder**
 2. Oracle SQL-Developer Remote-App **oder**
 3. HomeLab VD (virtuelles Labor) **oder**
 4. Installation von Oracle auf dem eigenen Rechner **oder**
 5. Nutzung von Oracle in einem Docker-Container

**Vielen Dank
für Ihre aktive Mitarbeit**