



Datenbanken 1

Prof. Dr. Uta Störl

Hochschule Darmstadt – Fachbereich Informatik

Sommersemester 2016



Organisatorische Vorbemerkungen

- **Vorlesungsunterlagen**

- Online unter <https://www.fbi.h-da.de/organisation/personen/stoerl-uta/lehre/datenbanken-1.html>
- werden im Laufe des Semesters jeweils einige Tage vor der Vorlesung online bereit gestellt

- **Praktikum (gemeinsam mit Michael Roth)**

- Voraussetzung für Teilnahme: PAD 1 bestanden und PAD 2 begonnen (d.h. mindestens einmal die Klausur mitgeschrieben)
- Termine und Aufgabenstellungen auf der Homepage
- Teilnahmepflicht + Testatpflicht am Praktikumstermin
- **Vorlage der Vorbereitungsaufgaben(!) am Anfang des Praktikums**

- **Offenes Labor Datenbanken (Tutor Yorrik Schwappacher)**

- Dienstag, 16.00 – 19.15 Uhr D14/112
- Möglichkeit zum Üben, Fragen stellen, Praktika vor- und nachbereiten etc.



Organisatorische Vorbemerkungen

- **Klausur**
 - An- bzw. Abmeldung ausschließlich über das OBS
 - erlaubte Hilfsmittel: 1 (beidseitig) beschriebenes/bedrucktes A4-Blatt
 - Beispielklausur wird online bereitgestellt
 - bestandenes Praktikum ist Voraussetzung für die Teilnahme



Datenbanken 1

– Kapitel 1: Einführung –



Einführung

Inhalt des Kapitels

- Einführung in das Thema Datenbanken
- Überblick über die in der Lehrveranstaltung Datenbanken behandelten Themen
- Exemplarisches Mini-Beispiel, um erste Grundbegriffe für das Entwerfen von Datenbankmodellen kennen zu lernen

Lernziele

- Kennenlernen von Grundbegriffen für das Entwerfen von Datenbankmodellen



Einsatzgebiete für Datenbanken

- Finanzwesen
 - Kontoverwaltung, Überweisungen, Depots, ...
- ERP (Enterprise Resource Planning Systeme, z.B. SAP R/3)
 - Personalverwaltung, Buchhaltung, Produktbestände, Controlling
- eBusiness
 - Shop-Systeme, Produktkataloge, Aufträge
- Bibliothek
 - Volltextsuche, Entleihe
- Decision Support
 - statistische Auswertungen in großen Handelsketten u.ä.
- CAD-Systeme
 - Konstruktionsdaten
- Geographische Informationssysteme
 - Topologische Daten
 - Satellitendaten



Beispiele für große Datenbanken

- **Große Datenmenge**
 - Walmart: 2,5 Petabyte (2008)
 - eBay: 9,2 Petabyte (2013)
 - ...
- **Hoher Durchsatz**
 - UPS: 1,1 Milliarden SQL-Statements pro Stunde (2005)
 - ...
- **Große Anzahl Datenbanktabellen**
 - SAP R/3-Installation der Deutschen Telekom AG:
> 13.000 Datenbanktabellen
 - ...



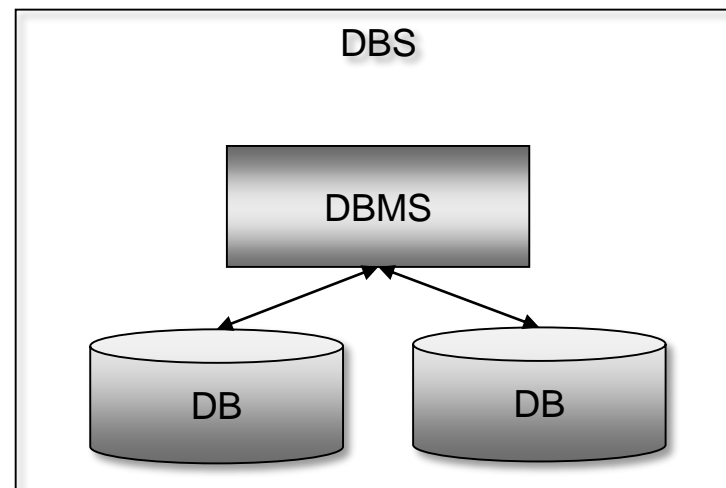
Kategorien von Datenbankanwendungen

- **OLTP (Online Transaction Processing)**
 - viele (kurze) Verarbeitungsvorgänge (Transaktionen) – mit (meist) einfachen Lese- und Schreiboperationen
 - viele parallele Benutzer
 - schnelle Antwortzeiten wichtig
 - hoher Datendurchsatz wichtig
 - z.B. Buchungssystem
- **DSS (Decision Support Systems)**
 - komplexe (lange) Verarbeitungsvorgänge – hauptsächlich Leseoperationen
 - wenige parallele Benutzer
 - relativ unkritische Antwortzeiten
 - z.B. Data Warehouses (OLAP, Online Analytical Processing)



Wichtige Grundbegriffe

Kürzel	Begriff	Erläuterung
DB	Datenbank	Strukturierter von DBMS verwalteter Datenbestand
DBMS	Datenbankmanagementsystem	Software zur Verwaltung von Datenbanken
DBS	Datenbanksystem	DBMS plus Datenbank(en)



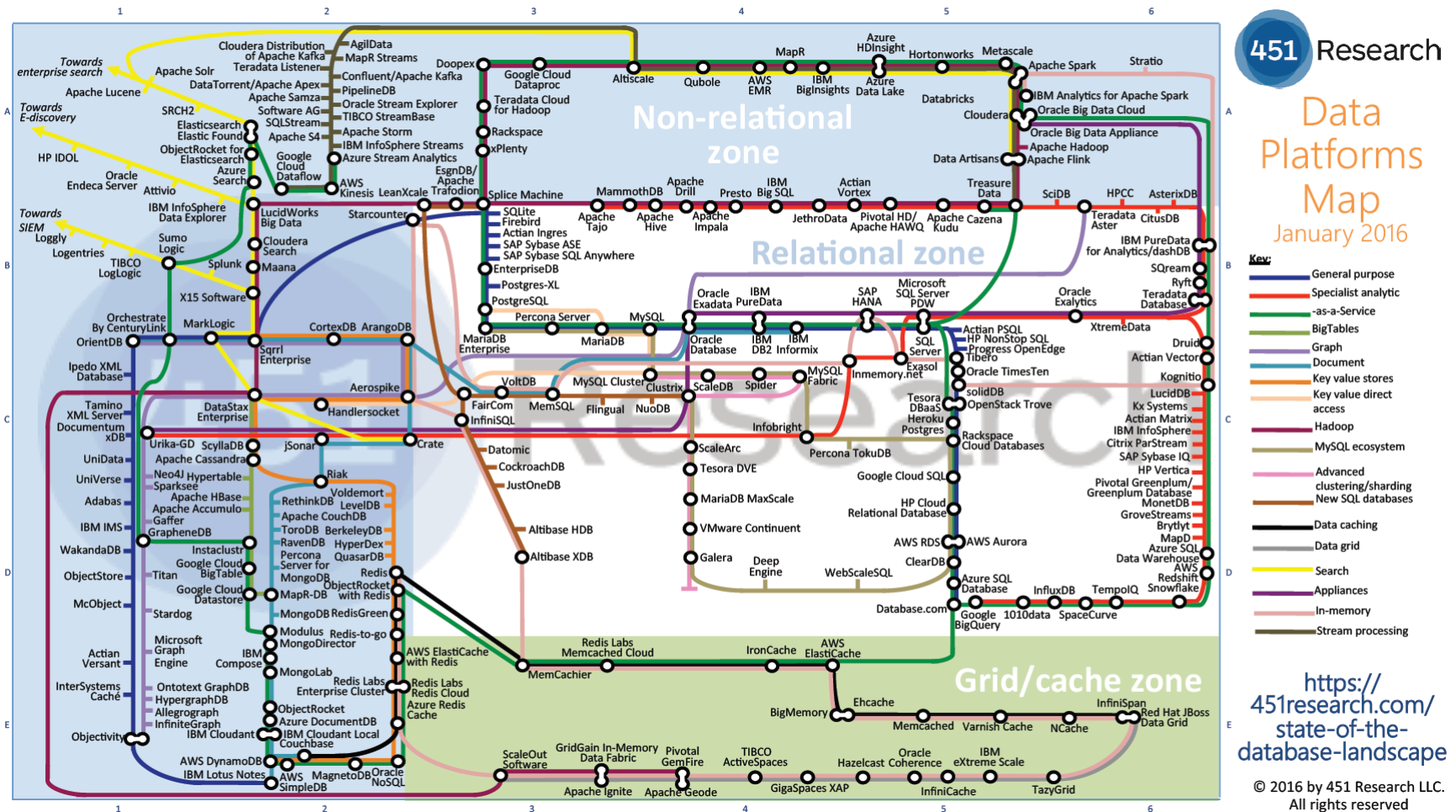


Konkrete Systeme (Auswahl)

- Hierarchische Datenbanken und Netzwerkdatenbanken
 - IMS (IBM), UDS (Siemens)
- **(Objekt-)Relationale DBMS**
 - **Oracle, IBM DB2, Microsoft SQL Server, Sybase** (gekauft von SAP), **Informix** (gekauft von IBM), **Teradata**, ...
 - **PostgreSQL, MySQL** (gekauft von Oracle), **MariaDB** (Fork von MySQL), **FireBird, SQLite**, ...
- Objektorientierte DBMS
 - Versant, db4o, ObjectStore, ...
- XML-DBMS
 - BaseX, Tamino (Software AG), ...
- NoSQL-DBMS
 - MongoDB, Couchbase, Cassandra, HBase, ...

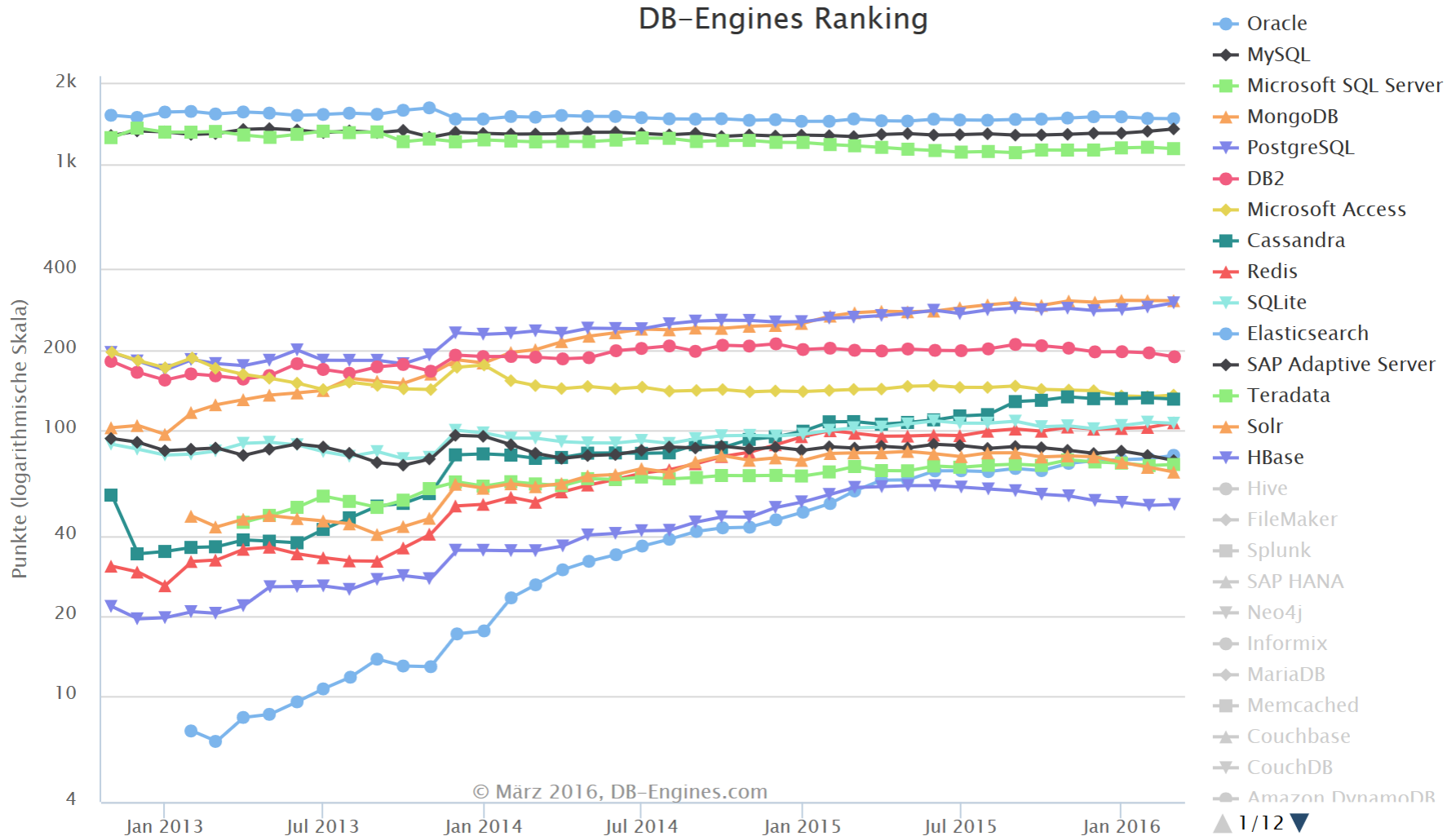


Übersicht Datenbanksysteme





„Popularität“ Datenbanksysteme



Quelle: http://db-engines.com/de/ranking_trend



Warum Datenbankmanagementsysteme?

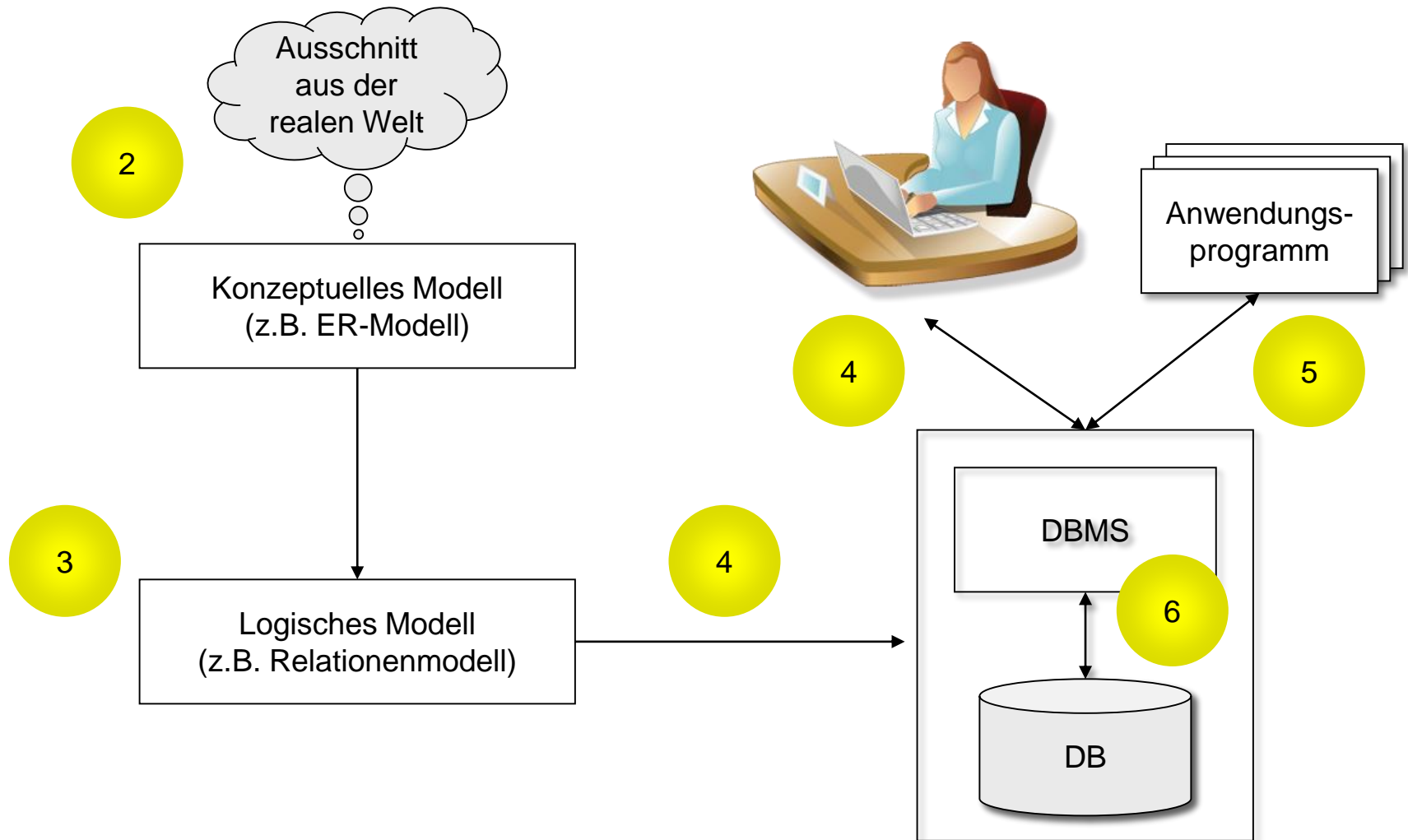
- Redundanzkontrolle
- Zugriffsbeschränkungen
- Aktionen anhand von Regeln
- Mehrbenutzerschnittstellen
- Beziehungen zwischen Daten
- Integritätsbedingungen
- Recovery

Bemerkung:

Es existieren auch formale Regeln die definieren, wann ein Softwaresystem wirklich ein Datenbankmanagementsystem ist. (Behandlung später)



Aufbau der Vorlesung Datenbanken 1





Phasen des Datenbankentwurfs

Anforderungsanalyse

Datenbedarf

nicht formalisiert

Konzeptioneller Entwurf

Konzeptuelles Modell

Informationsstruktur

Logischer Entwurf

Logisches Modell

logische Datenbankstruktur

Physischer Entwurf

Internes Schema

interne (physische) Datenbankstruktur



Anforderungsanalyse

Aufgaben

- Identifikation der zu unterstützenden Aufgaben (Prozesse!)
- Sammeln der Anforderungen
- Filtern der relevanten Informationen
- Klassifikation (Einteilung) der Informationen

Ergebnis der Anforderungsanalyse ist im allgemeinen nicht formalisiert

Beispiel: Produktdaten eines PC-Herstellers

Bezeichnung: *Notebook*

Farbe: *graublau*

RAM: *4 GB*

Preis: *679,00*

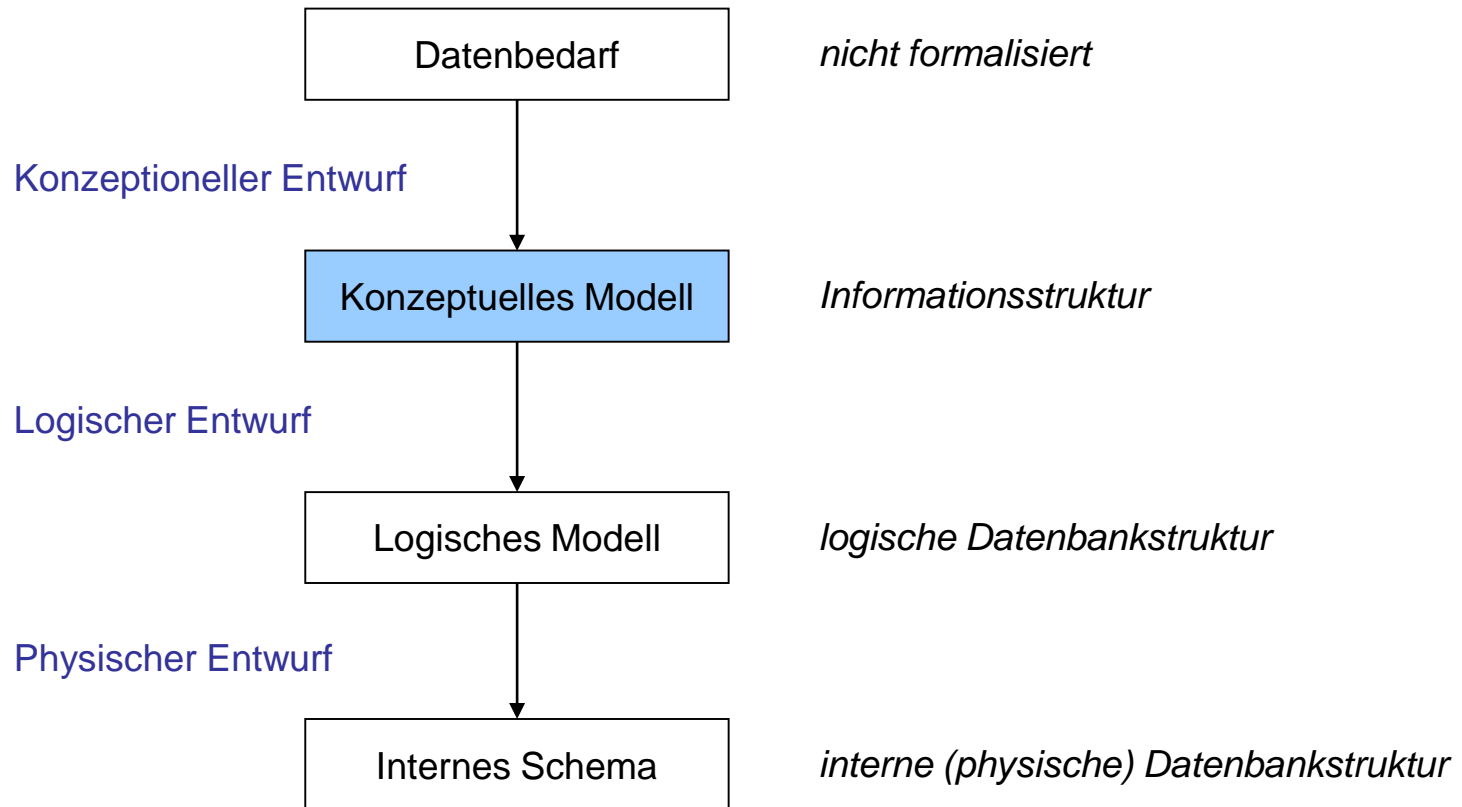
...





Phasen des Datenbankentwurfs

Anforderungsanalyse





2

Konzeptioneller Entwurf

- Ergebnis des konzeptionellen Entwurfs ist ein **konzeptuelles Modell**
- Im Datenbankbereich wird für die Repräsentation des konzeptuellen Modells i.a. das **Entity-Relationship-Modell** (ER-Modell bzw. ERM) verwendet
 - entwickelt 1976 von Peter Chen
 - viele Erweiterungen und Weiterentwicklungen im Laufe der Jahre
 - *Analogie zur objektorientierten Analyse-Phase (UML, 1997)*



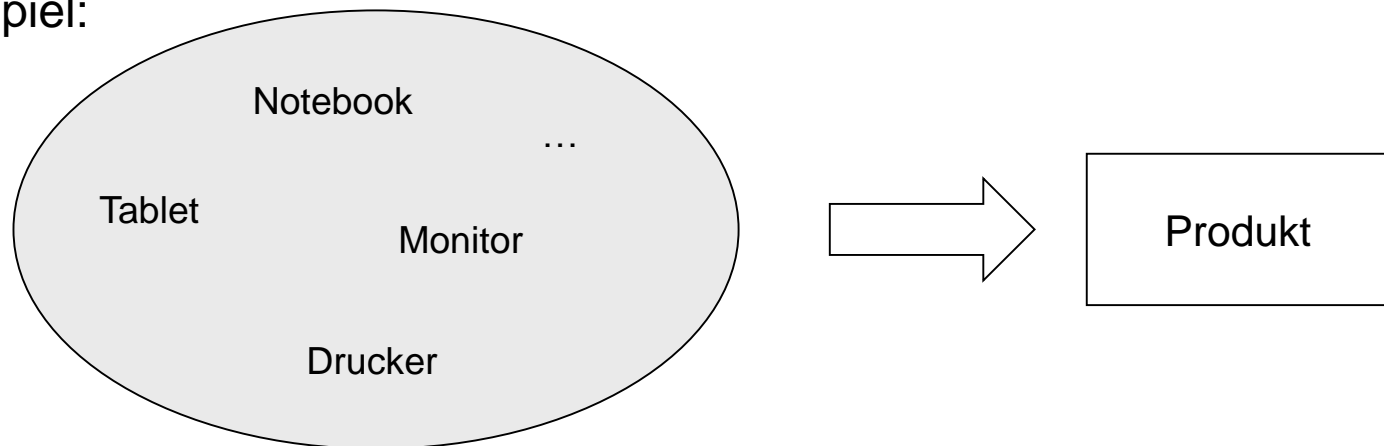
ER-Modell: Grundbegriffe – 1(2)

- **Entity.** Objekt der realen (bzw. der zu modellierenden) Welt.
- **Entity-Set.** Menge aller Entities mit gleichen oder ähnlichen Eigenschaften.
- **Entity-Typ.** Repräsentant der Objekte gleichen Typs.

Notation:

Entity-Typ

Beispiel:





ER-Modell: Grundbegriffe – 2(2)

- **Attribut:** repräsentiert eine Eigenschaft eines Entity-Typs, d.h. eine Eigenschaft, welche alle Entities dieses Entity-Typs besitzen.

Notation:

$E (A_1, A_2, \dots, A_m)$

Attribut

Beispiel:

Produkt

Bezeichnung

Preis

- Weitere wichtige Grundbegriffe: **Schlüssel, Beziehungen, Kardinalitäten von Beziehungen** ... → siehe Kapitel 2



Mini-Beispiel: Kundenverwaltung

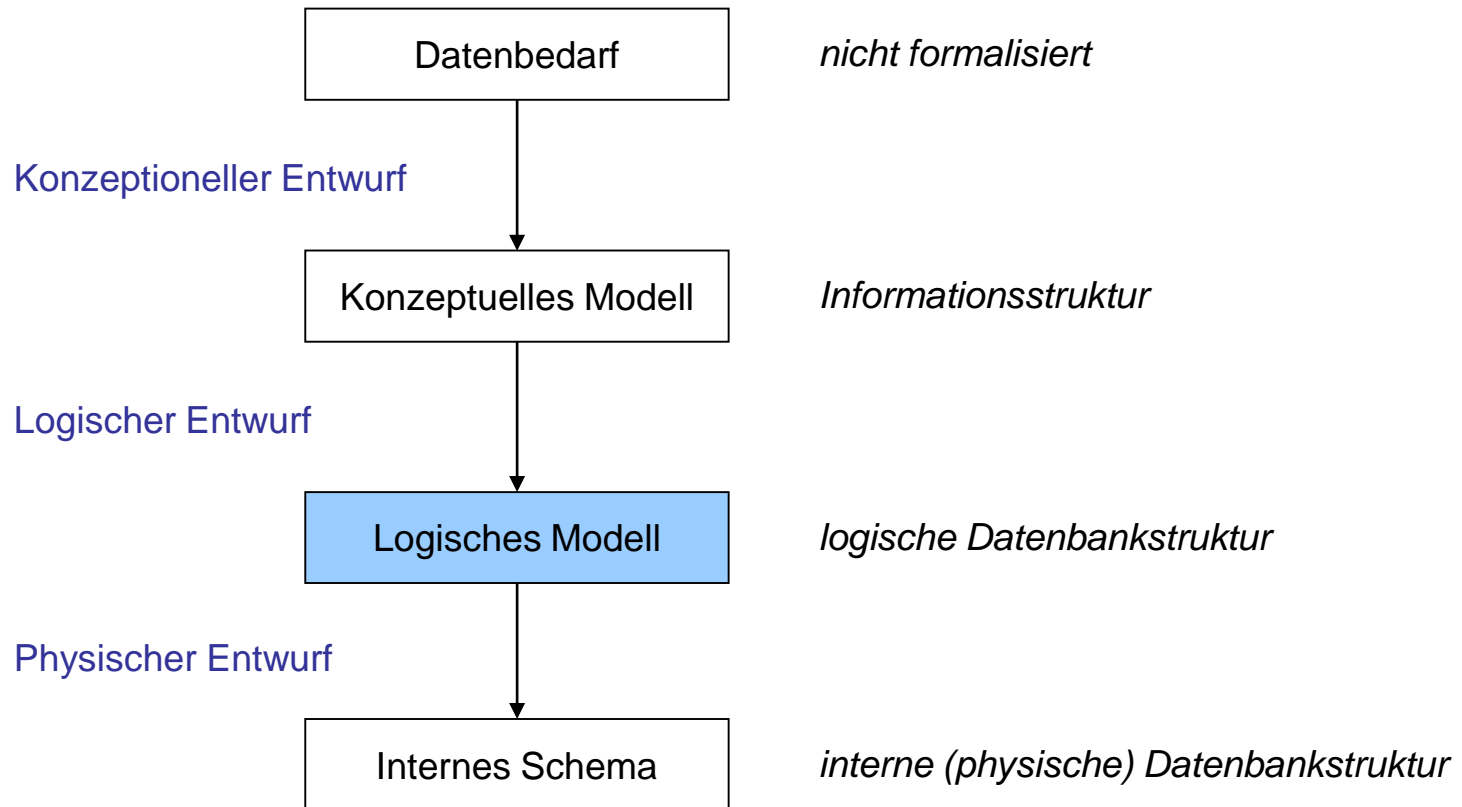
- Ein Unternehmen möchte seine Kunden in einer Datenbank verwalten
 - Was sind die betrachteten Entities?
 - Was ist der geeignete Entity-Typ?
 - Welche Eigenschaften hat der Entity-Typ?





Phasen des Datenbankentwurfs

Anforderungsanalyse





- Ergebnis des logischen Entwurfs ist ein **logisches Modell**
- Welches logische Modell verwendet wird, hängt vom Datenbankmodell des verwendeten Datenbankmanagementsystems ab
 - *Analogie zur objektorientierten Design-Phase (Entscheidung für eine bestimmte Programmiersprache)*
- Wichtigstes logisches Datenbankmodell ist das **Relationenmodell** (oder Relationales Modell)
 - Grundlagen 1970 von Edgar F. Codd



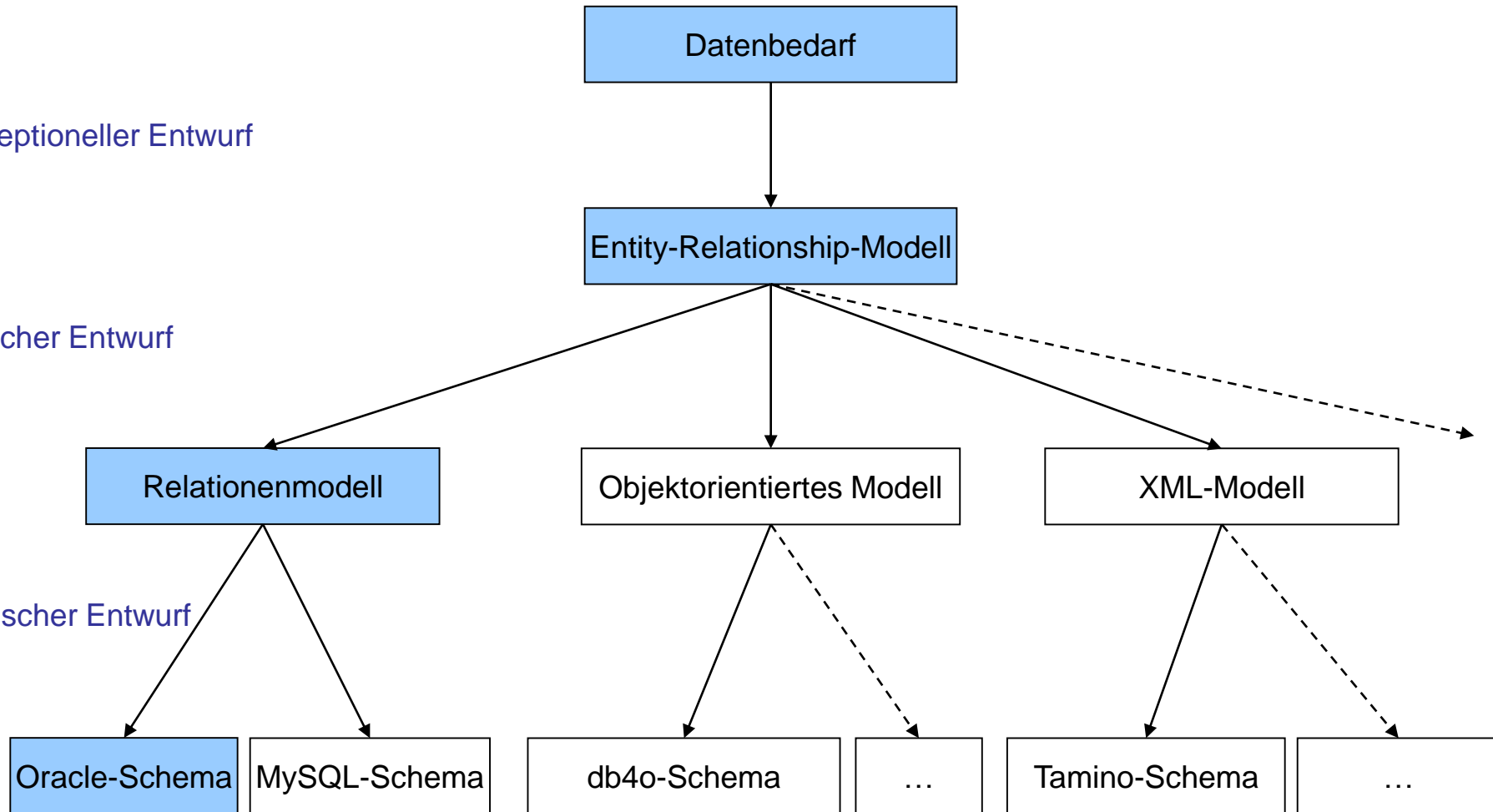
Phasen des Datenbankentwurfs: Beispiele

Anforderungsanalyse

Konzeptioneller Entwurf

Logischer Entwurf

Physischer Entwurf



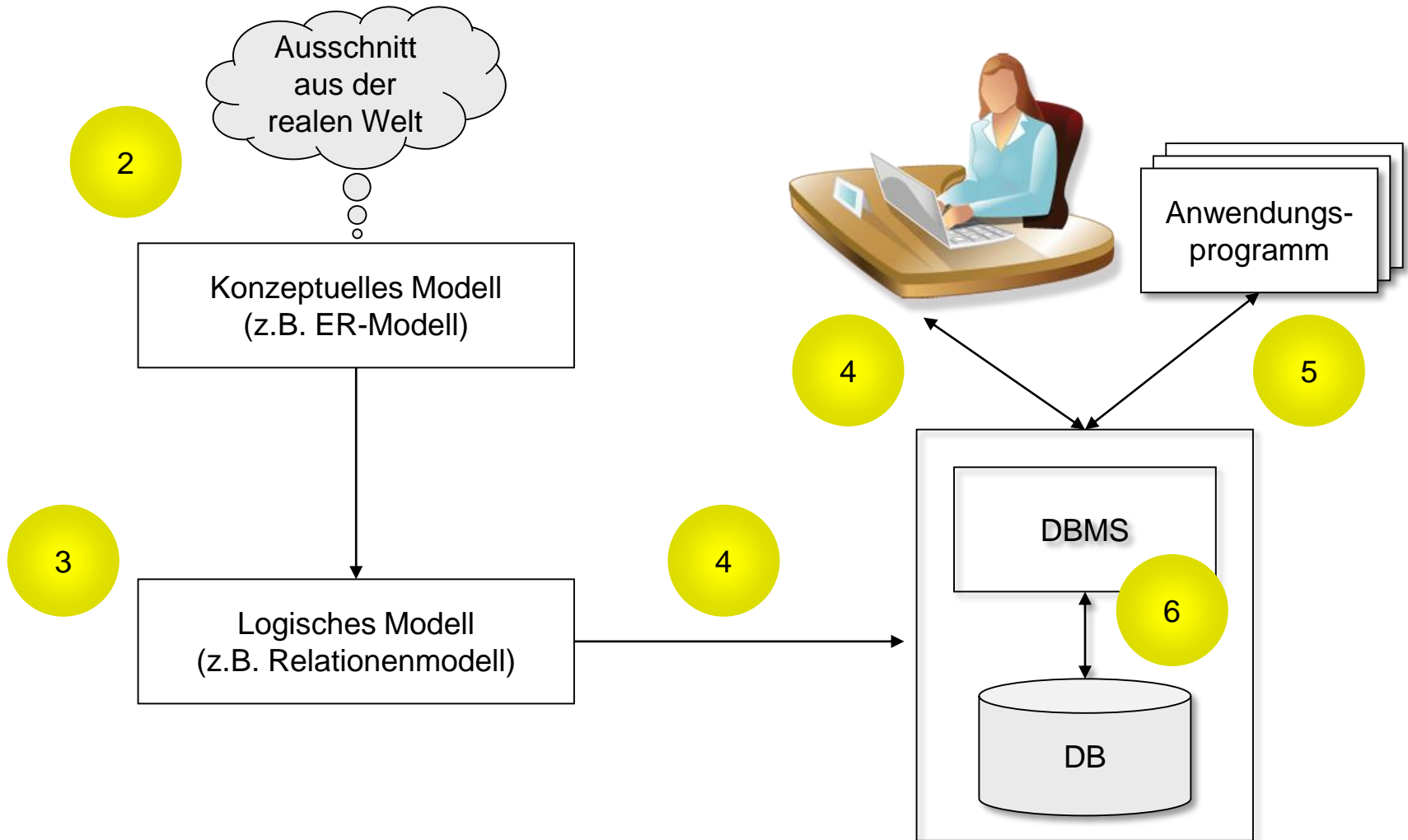


- extrem vereinfachte(!) Sicht: eine Relation ist eine Tabelle
- genaue Diskussion der Konzepte des Relationenmodells → Kapitel 3

KUNDE:	KNR	NAME	VORNAME	GEBURTSDATUM
	1001	Mario	Götze	03.06.1992
	1002	Podolski	Lukas	04.06.1985
	1003	Özil	Mesut	15.19.1988



Datenbanken 1





- wichtigste Datenbanksprache zur Datendefinition (und -manipulation) in relationalen DBMS: **Structured Query Language (SQL)**
- ausführliche Diskussion der wichtigsten Sprachkonzepte:
→ siehe Kapitel 4

- Beispiel:

```
CREATE TABLE Kunde (  
  Kundennummer integer           PRIMARY KEY,  
  Name          char(30)         NOT NULL,  
  Vorname       char(30)         NOT NULL,  
  Geburtsdatum  date  
);
```



Datendefinition: Datentypen in SQL

- **integer** (oder auch **integer4**, **int**),
- **smallint** (oder auch **integer2**),
- **float**(p) (oder auch kurz **float**),
- **decimal**(p, q) und **numeric**(p, q) mit jeweils q Nachkommastellen,
- **character**(n) (oder kurz **char**(n), bei $n = 1$ auch **char**) für Zeichenketten (Strings) fester Länge n ,
- **character varying**(n) (oder kurz **varchar**(n) für Strings variabler Länge bis zur Maximallänge n ,
- **bit**(n) oder **bit varying**(n) analog für Bitfolgen, und
- **date**, **time** bzw. **timestamp** für Datums-, Zeit- und kombinierte Datums-Zeit-Angaben
- **blob** (**binary large object**) für sehr große binäre Daten
- **clob** (**character large object**) für sehr große Strings

Achtung: Gelegentlich verwenden Hersteller andere Bezeichnungen (beispielsweise bei Neuimplementierung von Datentypen – Oracle beispielsweise **varchar2**(n))



4

Datenmanipulation – 1(2)

- Daten müssen **eingefügt**, **gelesen** und **verändert** werden können
- Beispiel für Einfügen (**INSERT**)

```
INSERT INTO Kunde  
VALUES (1001, 'Götze', 'Mario', '03.06.1991')
```

- Beispiele für Lesen (**SELECT**)

```
SELECT KNr, Name, Geburtsdatum  
FROM Kunde  
WHERE Name = 'Götze'
```

```
SELECT KNr, Name  
FROM Kunde
```



- Beispiel für Änderung (**UPDATE**)

```
UPDATE Kunde  
SET Geburtsdatum = '03.06.1992'  
WHERE KNr = 1001
```

- Beispiele für Löschen (**DELETE**)

```
DELETE  
FROM Kunde  
WHERE Name = 'Götze'
```

```
DELETE  
FROM Kunde
```

Transaktionen

- Um die Aktionen mehrere Benutzer voneinander zu isolieren und/oder mehrere Aktionen logisch zusammenzufassen, verwendet man in Datenbanksystemen **Transaktionen**
- Eine Transaktion wird mit **COMMIT** beendet – erst dann sind die Änderungen für andere Nutzer sichtbar:



```
INSERT INTO Kunde  
VALUES (1002, 'Götze', 'Mario', '03.06.1992');
```

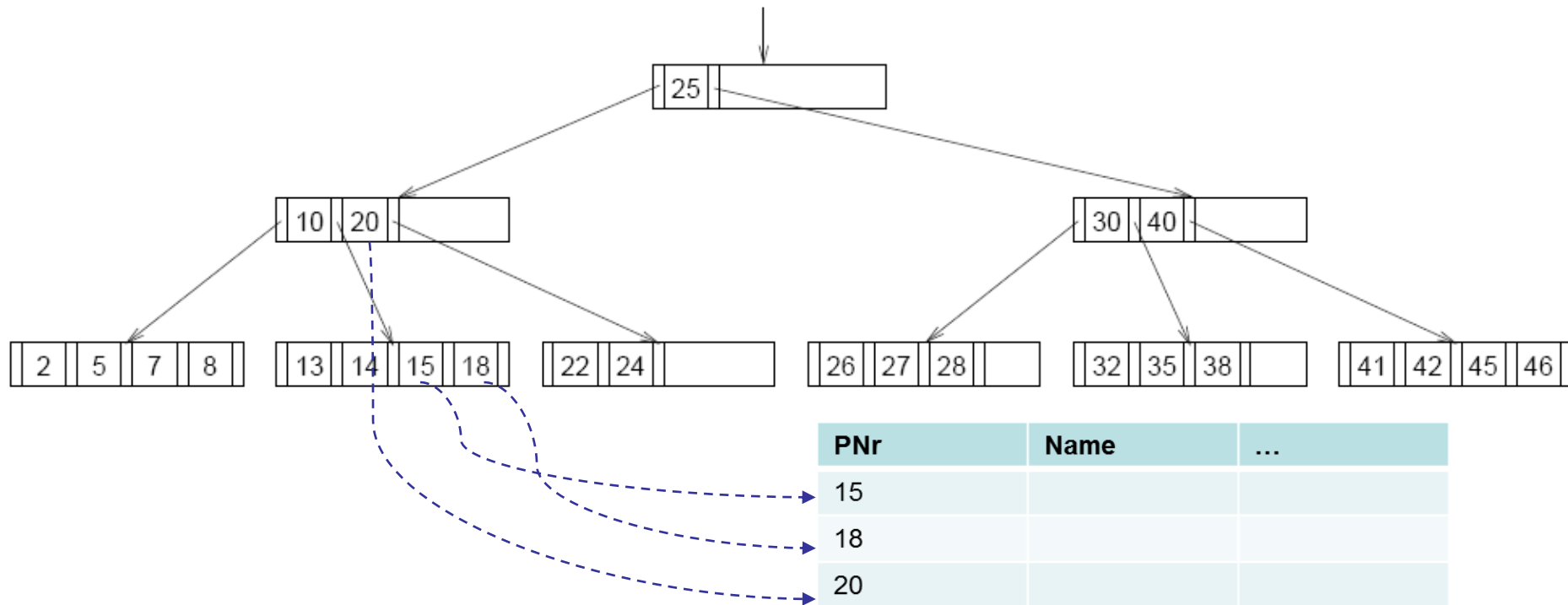
```
COMMIT;
```



```
SELECT *  
FROM Kunde;
```

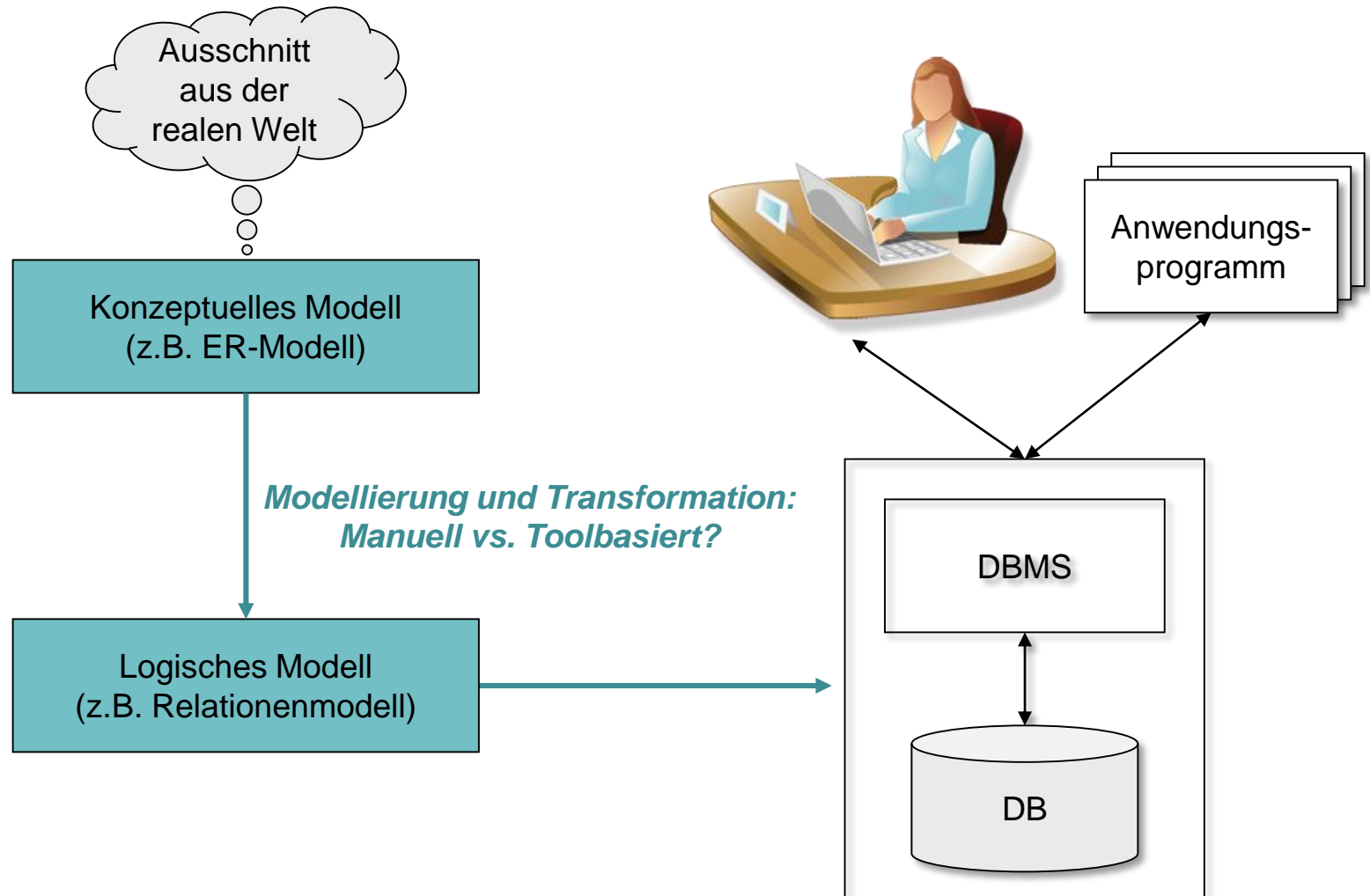


- Datenbanken speichern teilweise Millionen von Daten in einer Tabelle (z.B. Verbindungsdaten Telekommunikation)
- ⇒ Lineares Durchsuchen würde viel zu lange dauern
- ⇒ Um den Zugriff auf häufig benötigte Daten zu optimieren, werden in Datenbanksystemen Indexe verwendet; meistens B-Bäume:



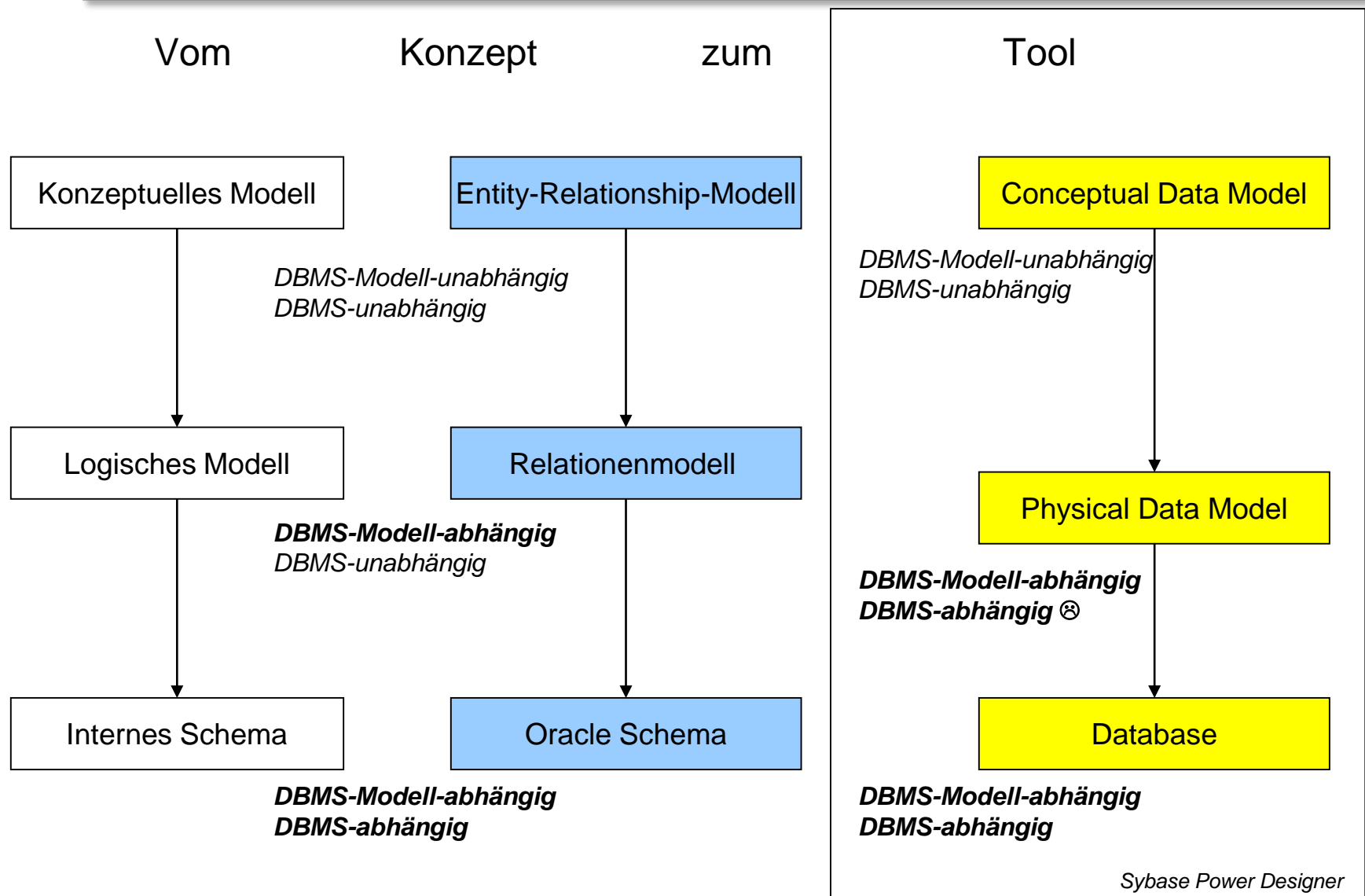


Datenbanken 1



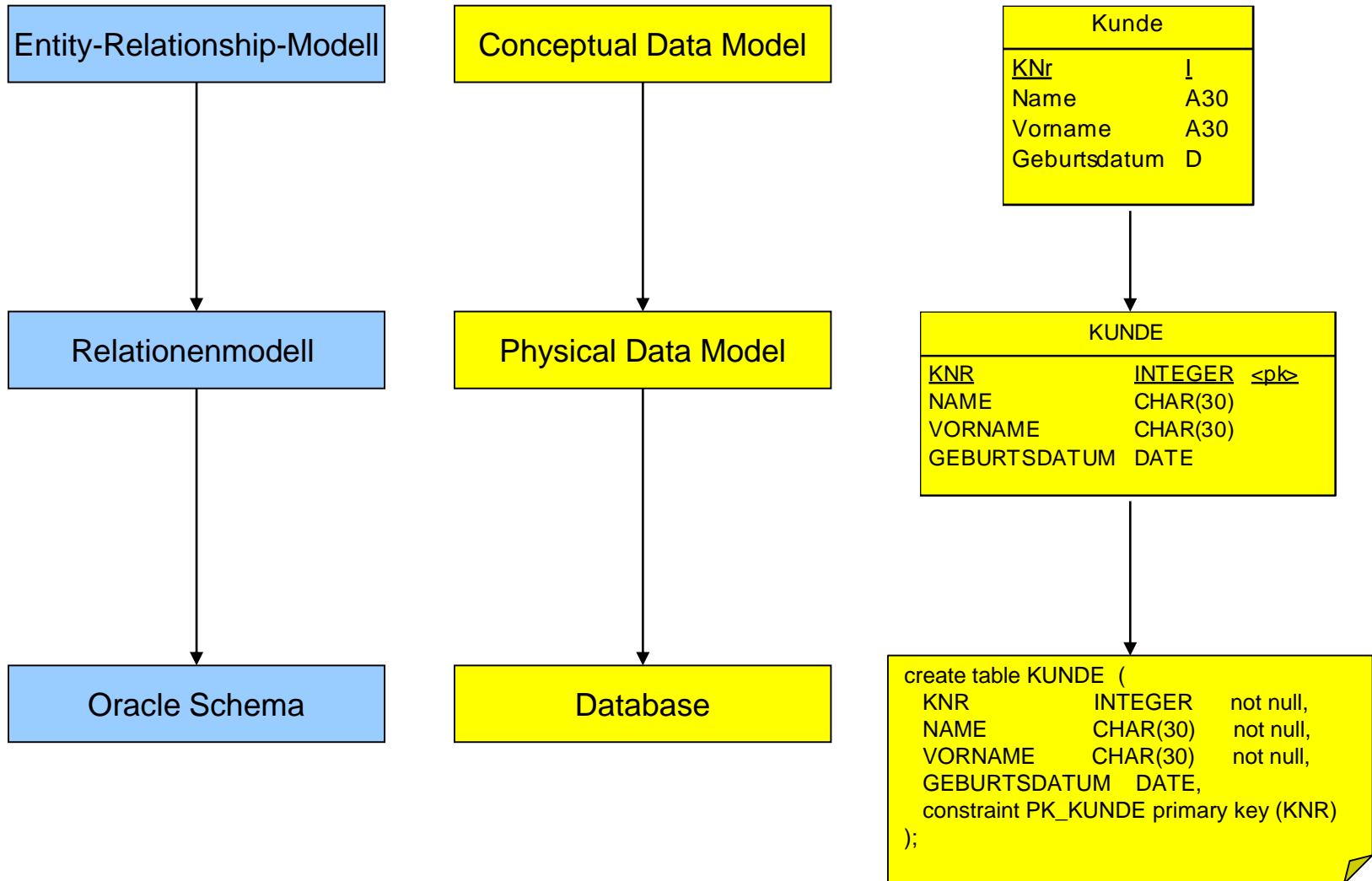


Toolbasierte Datenbankmodellierung





Toolbasierte Datenbankmodellierung



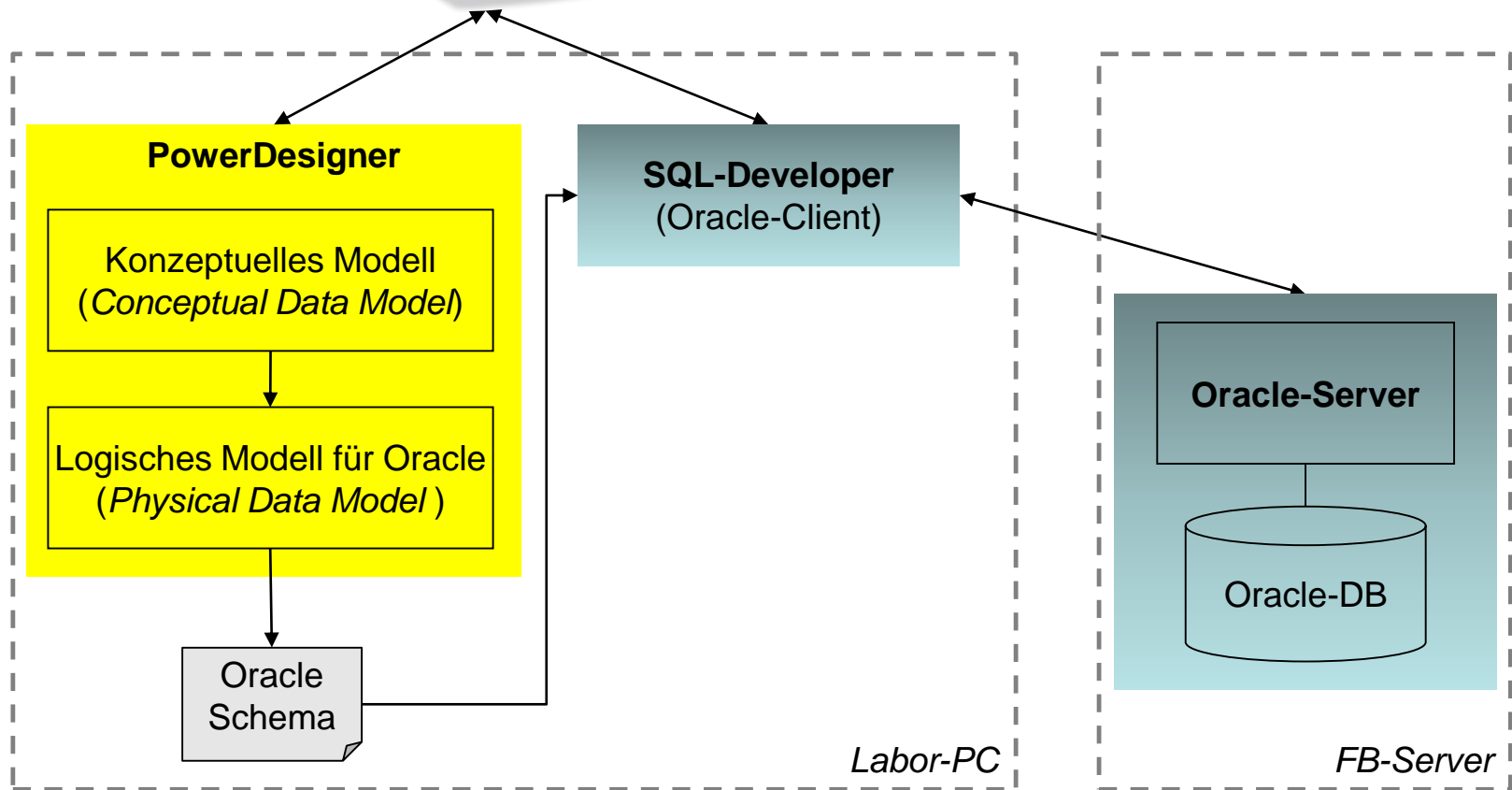


Praktikum

1. Datenbankentwurf mit dem Entity-Relationship-Modell und vertraut werden mit den verwendeten Software-Tools (PowerDesigner, Oracle)
2. Erweiterter Datenbankentwurf mit dem Entity-Relationship-Modell
3. Arbeit mit SQL (Datendefinition und Datenmanipulation)
4. Datenbankprogrammierung (Stored Procedures und Trigger)
5. Datenbankprogrammierung (JDBC und Transaktionen)



Praktikumsumgebung

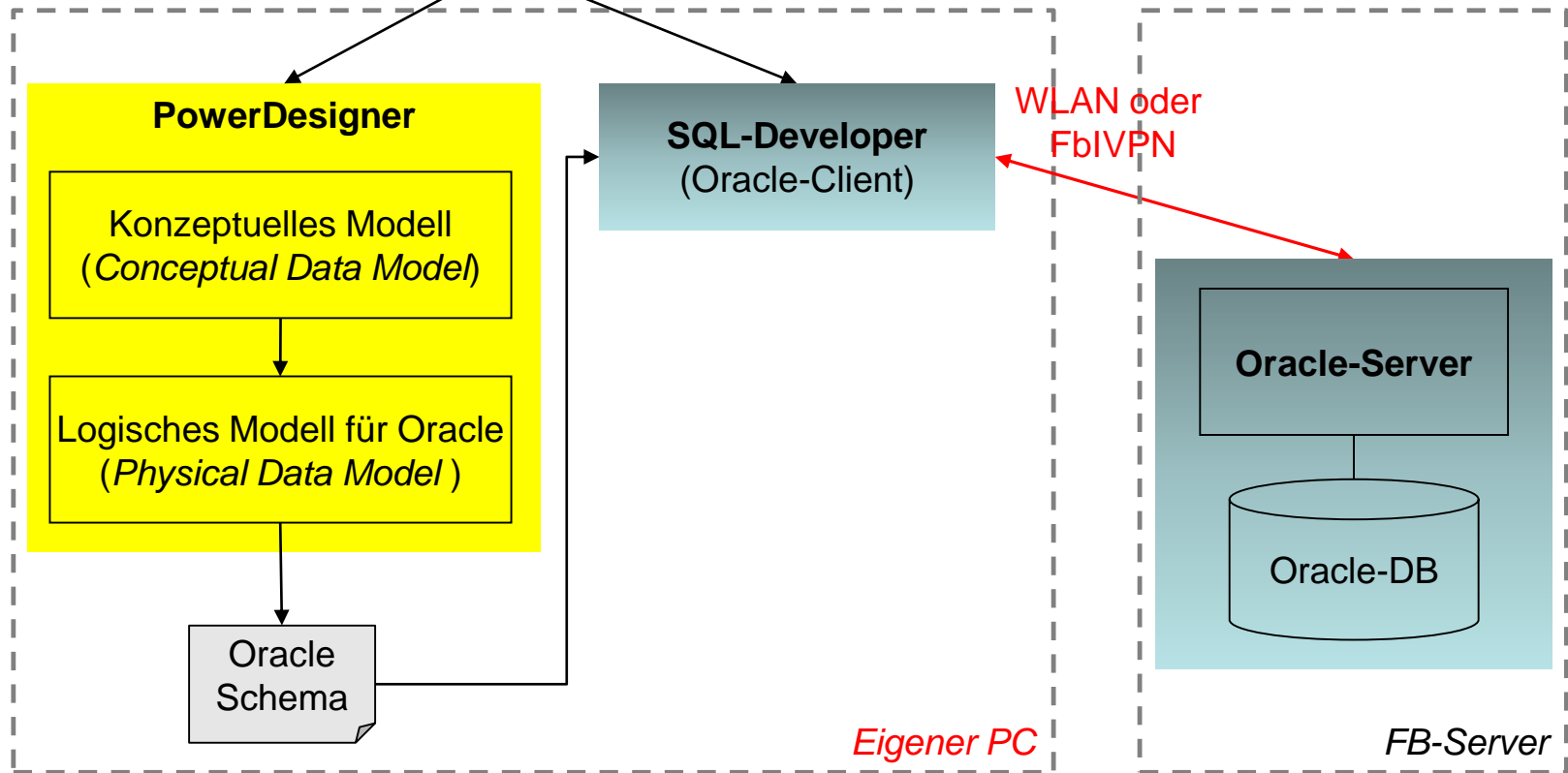




Arbeit zu Hause / außerhalb des Labors



Variante 1

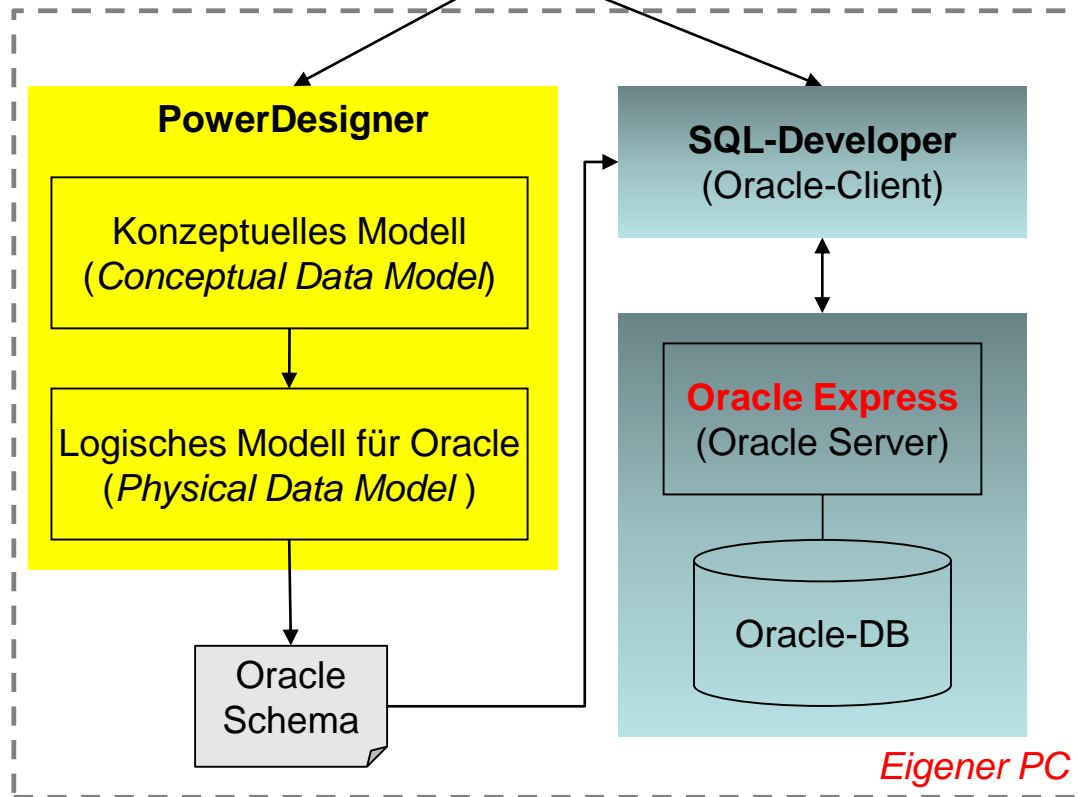




Arbeit zu Hause / außerhalb des Labors



Variante 2





Literaturempfehlungen

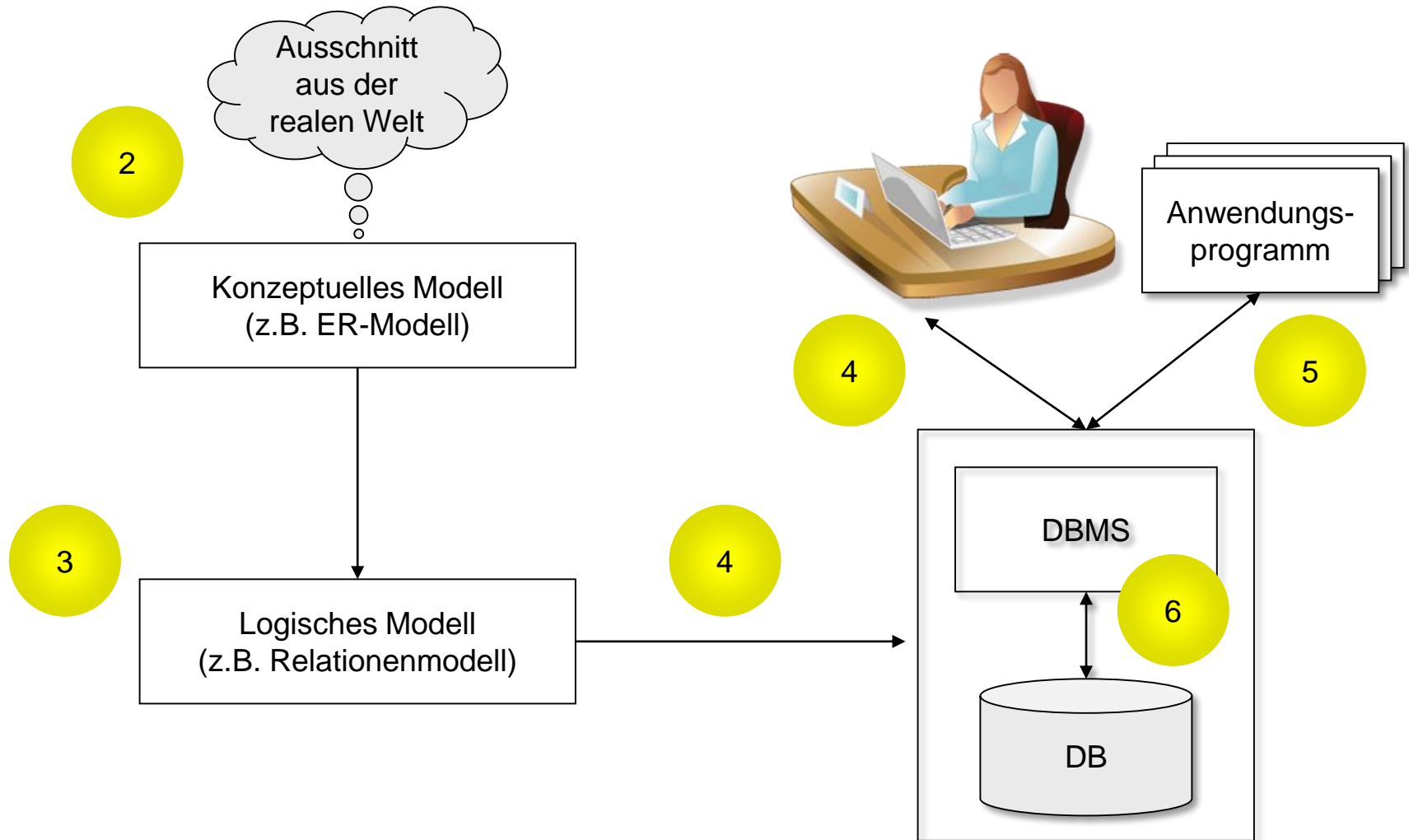
- G. Saake, A. Heuer und K.-U. Sattler: Datenbanken: Konzepte und Sprachen. MITP-Verlag, 5. Auflage 2013.
auch ältere Auflagen geeignet!
- A. Kemper und A. Eickler: Datenbanksysteme. Oldenbourg Verlag, 8. Auflage, 2011.
auch ältere Auflagen geeignet!
- R. Elmasri und S. B. Navathe: Grundlagen von Datenbanksystemen. (Bachelorausgabe). Pearson Studium, 3. Auflage, 2009.

Englische Literatur:

- C. J. Date: An Introduction to Database Systems. Addison-Wesley Systems Programming Series, 8th ed. 2003.



Datenbanken 1





Ausblick Datenbanken 2

