

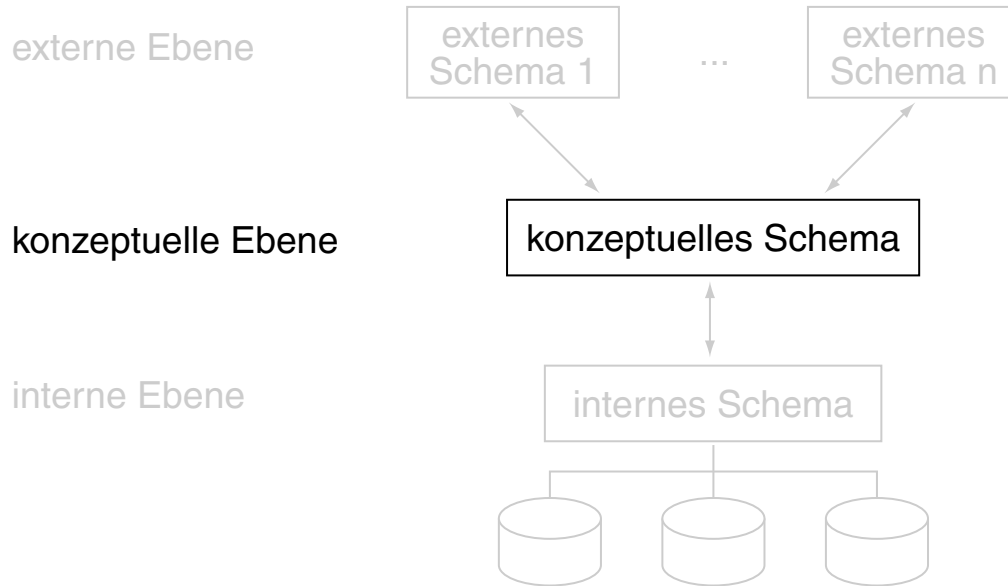
Kapitel DB:II

II. Konzeptueller Datenbankentwurf

- ❑ Entwurfsprozess
- ❑ Datenbankmodelle
- ❑ Einführung in das Entity-Relationship-Modell
- ❑ ER-Konzepte und ihre Semantik
- ❑ Charakterisierung von Beziehungstypen
- ❑ Weitere ER-Konzepte
- ❑ Konsolidierung, Sichtenintegration
- ❑ Konzeptuelle Modellierung mit UML

Entwurfsprozess

ANSI/SPARC-Schema-Architektur



*“Usually, a **representational** [= implementational, logical] **data model** is used to describe the conceptual schema when a database system is implemented. This implementation conceptual schema is often based on a conceptual schema design in a **high-level data model**.”*

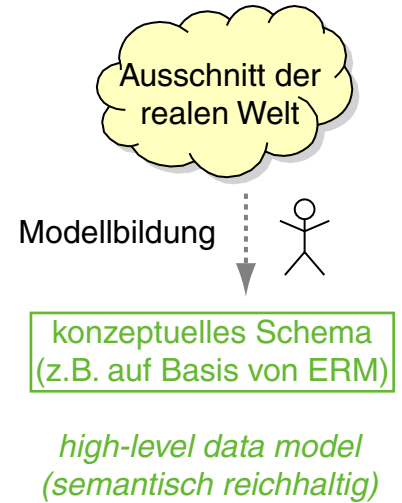
[p.37 Elmasri/Navathe 2015]

Bemerkungen zu [p.30 Elmasri/Navathe 2010]:

- ❑ Aussage 1: In einem Datenbanksystem wird das konzeptuelle Schema oft durch ein implementierungsnahes Datenmodell beschrieben, bspw. durch das relationale Datenmodell.
- ❑ Aussage 2: Im Entwurfsprozess entsteht dieses implementierungsnahes Datenmodell auf Grundlage eines semantisch reicheren Modells, etwa des ER-Modells oder UML.
- ❑ Idealerweise sollte eine Datenbankbeschreibung jedoch direkt auf dem semantisch reicheren Modell beruhen, ohne dass eine (manuelle) Transformation in ein implementierungsnahes Datenmodell erfolgen muss.

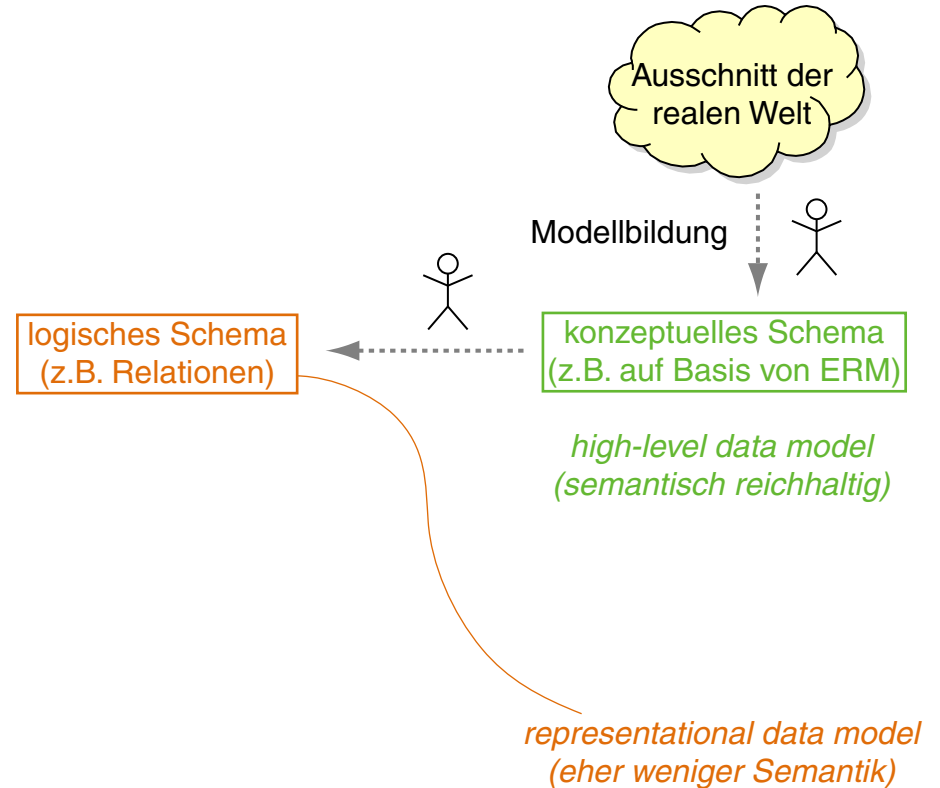
Entwurfsprozess

ANSI/SPARC-Schema-Architektur: Entwurfspraxis



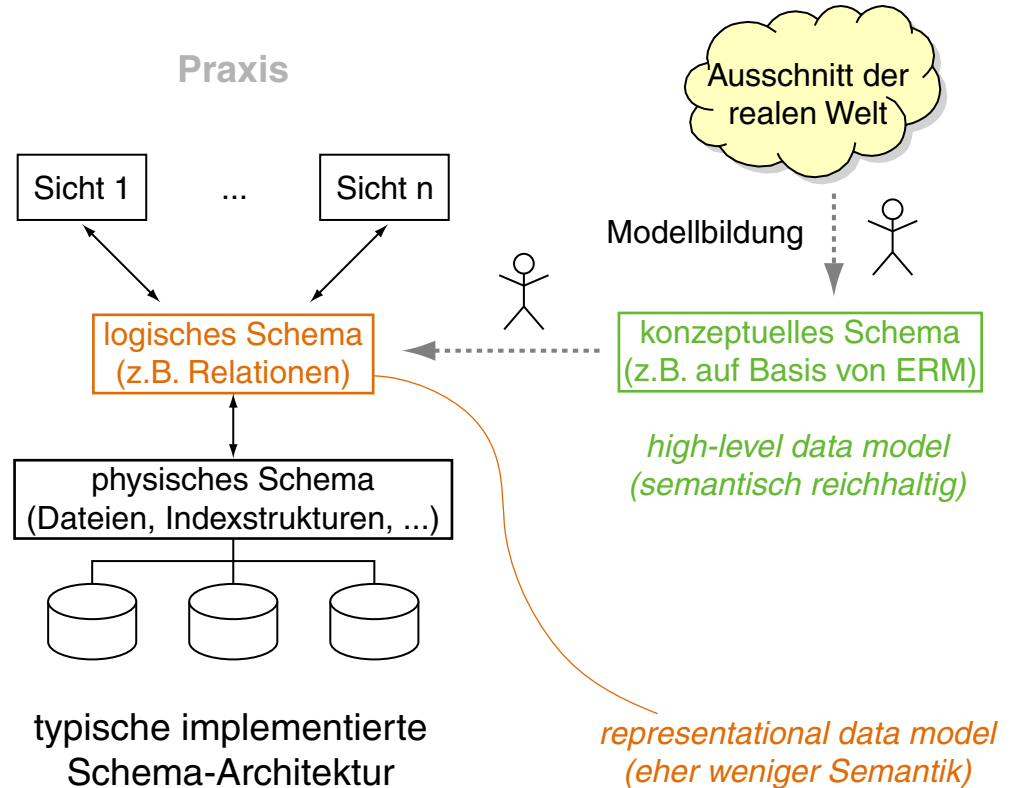
Entwurfsprozess

ANSI/SPARC-Schema-Architektur: Entwurfspraxis



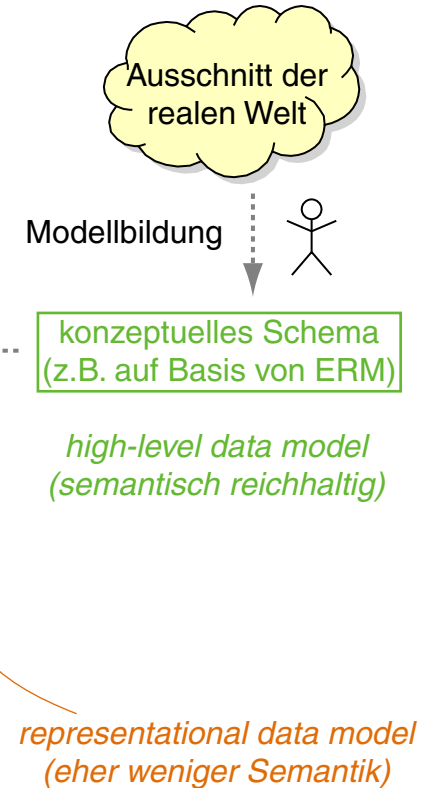
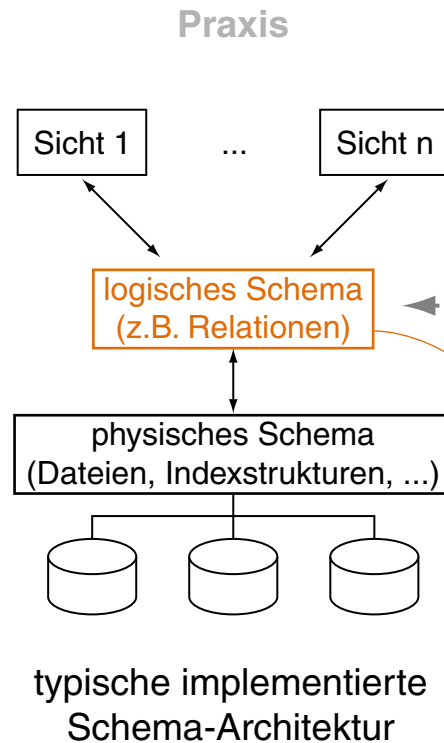
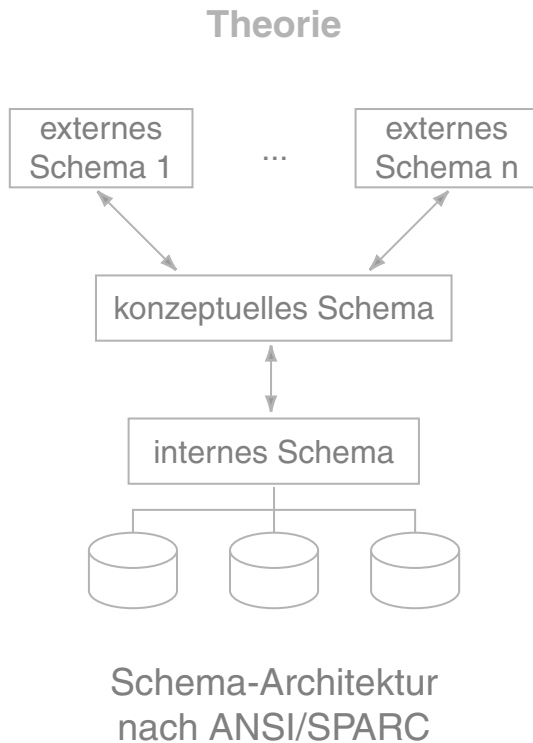
Entwurfsprozess

ANSI/SPARC-Schema-Architektur: Entwurfspraxis



Entwurfsprozess

ANSI/SPARC-Schema-Architektur: Entwurfspraxis



Entwurfsprozess

Zentrale Anforderung an den Entwurfsprozess ist die Informationserhaltung:

1. Erhaltung aller **Fakten**
2. Erhaltung aller **Beziehungen** zwischen den Fakten

Weitere, zum Teil informelle Gütekriterien:

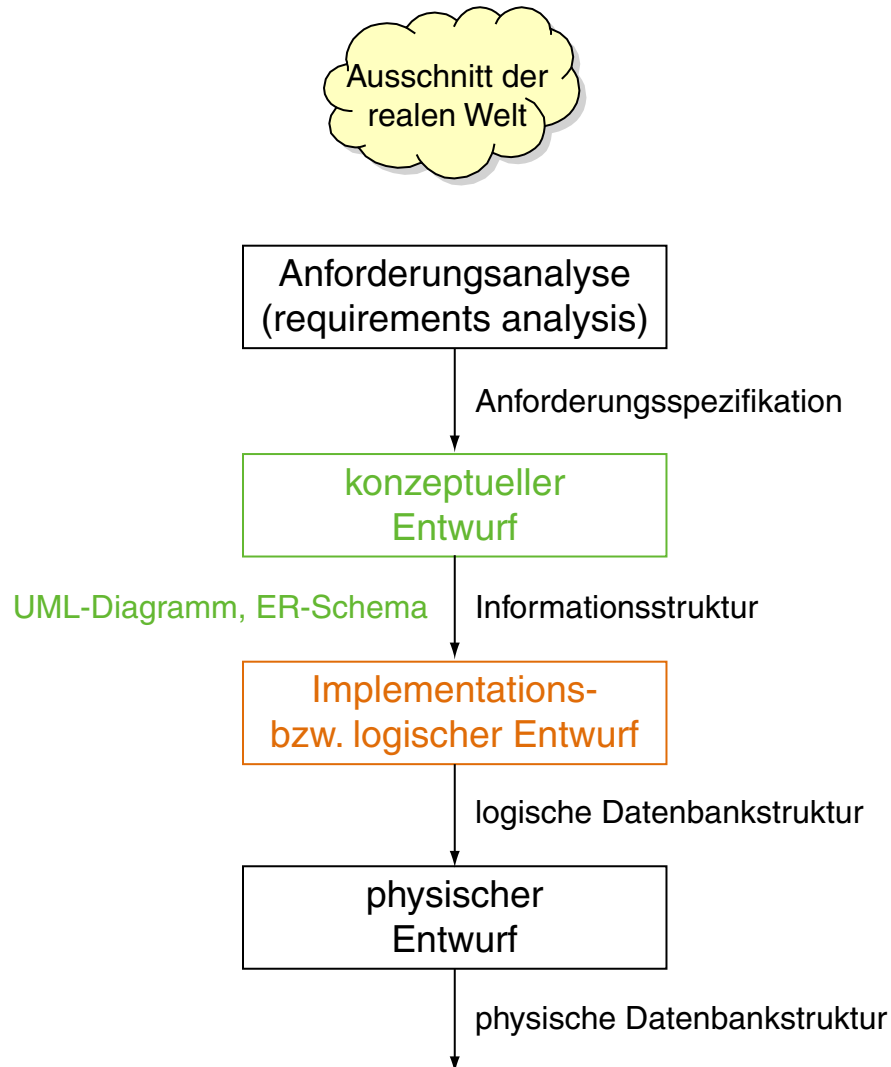
- ❑ Redundanzfreiheit
- ❑ Vollständigkeit bzgl. der Anforderungsanalyse
- ❑ Konsistenz der Beschreibung
- ❑ Ausdrucksstärke und Verständlichkeit des benutzten Formalismus
- ❑ formale Semantik
- ❑ Lesbarkeit der Dokumente
- ❑ Unterstützung von Erweiterbarkeit, Modularisierung, Wiederverwendbarkeit, Werkzeugeinsatz

Bemerkungen:

- ❑ Die Informationserhaltung bezieht sich auf die Transformation der Modelle ausgehend von der Anforderungsanalyse hin zur Implementierung:
 1. Die Erhaltung der Fakten fordert, dass alle gewünschten Eigenschaften der Objekte des Weltausschnitts auf jeder Abstraktionsstufe vorhanden sind.
 2. Die Erhaltung der Beziehungen fordert, dass alle gewünschten Regeln und Einschränkungen (*Constraints*) des Weltausschnitts auf jeder Abstraktionsstufe abbildbar sind.

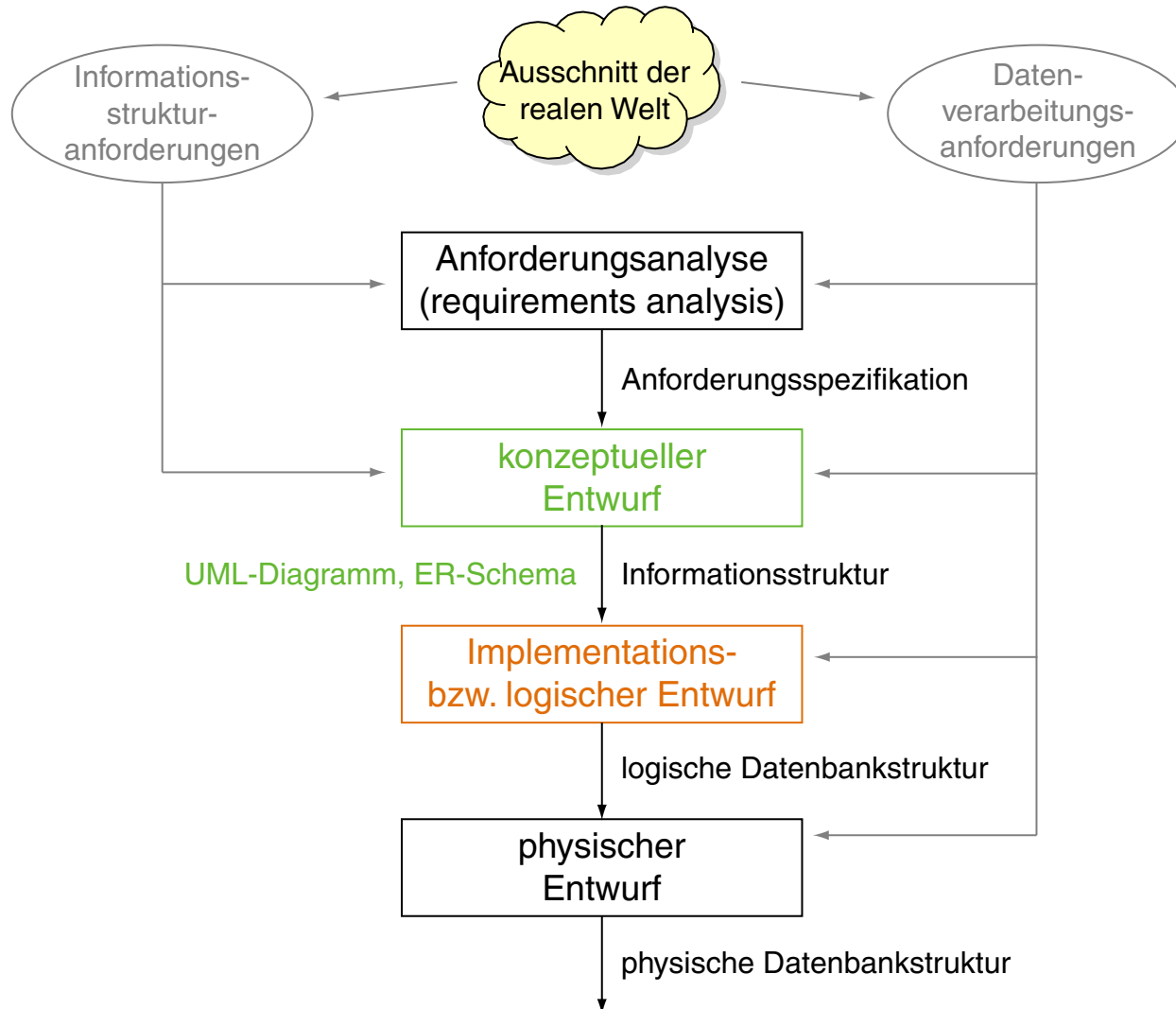
Entwurfsprozess

Phasenmodell des Datenbankentwurfs



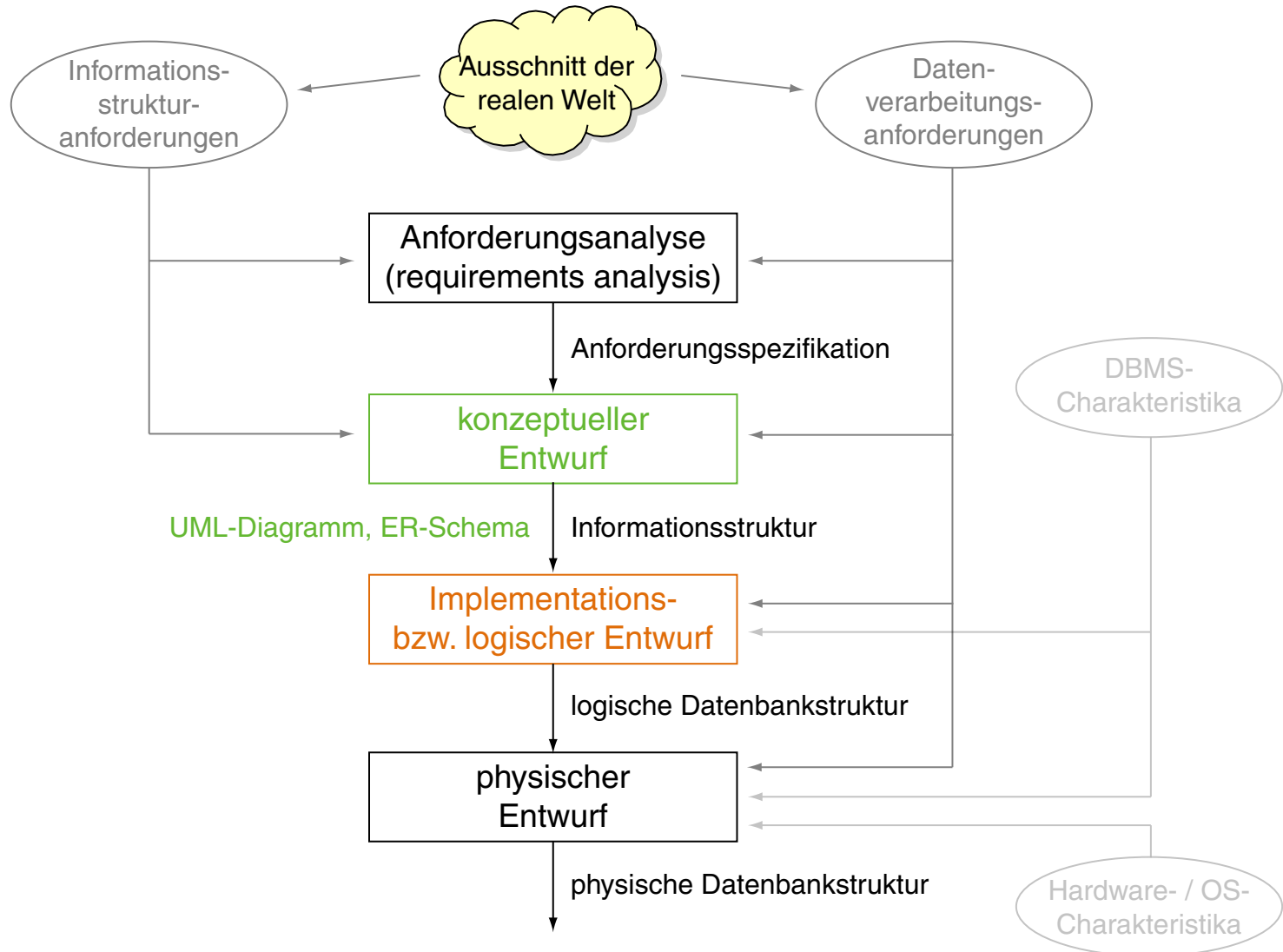
Entwurfsprozess

Phasenmodell des Datenbankentwurfs



Entwurfsprozess

Phasenmodell des Datenbankentwurfs



Bemerkungen:

- ❑ Verschiedene Phasen dieses Modells lassen sich weiter aufschlüsseln. Beispielsweise gliedern [Heuer/Saake 2018] den konzeptuellen Entwurf noch in einen Sichtenentwurf, eine Sichtenanalyse und eine Sichtenintegration.
- ❑ Sollen die Daten auf mehreren Rechnern *verteilt* vorliegen, muss Art und Weise der verteilten Speicherung festgelegt werden. Dies geschieht im sogenannten Verteilungsentwurf, einer Entwurfsphase zwischen konzeptuellem und logischen Entwurf.

Entwurfsprozess

Phasenmodell: Anforderungsanalyse

Ziel:

Sammlung des Informationsbedarfs in den verschiedenen Abteilungen bzw. Benutzergruppen.

Ergebnis:

- ❑ informelle Beschreibung des Problems bzw. der Aufgabenstellung (User Stories, Use-Cases, Texte, tabellarische Aufstellungen, Formblätter, Maskenentwürfe, etc.)
[Use-Case-Diagramm: uml-diagrams.org, Wikipedia]
- ❑ Trennen der Informationen über Daten (Datenanalyse) von den Informationen über Funktionen (Funktionsanalyse)

Bemerkungen:

- ❑ Im klassischen Datenbankentwurf wird nur die Datenanalyse einschließlich ihrer Folgeschritte behandelt; die Funktionsanalyse wird weitgehend ignoriert. Mittelfristig wird sich eine integrierte objektorientierte Betrachtung von Daten und Funktionen durchsetzen.

Entwurfsprozess

Phasenmodell: konzeptueller Entwurf

Ziel:

Formale, semantisch reichhaltige Beschreibung der Aufgabenstellung (der zu speichernden Daten) in einem abstrakten („high-level“) Datenmodell. Hierfür wird das Entity-Relationship-Modell am häufigsten eingesetzt.

Vorgehensweise:

1. Modellierung von Sichten (z.B. einer Abteilung oder Benutzergruppe)
2. Analyse der vorliegenden Sichten hinsichtlich von Konflikten:
 - ☐ Namenskonflikt: Homonyme, Synonyme
 - ☐ Typkonflikt: verschiedene Strukturen für das gleiche Konzept
 - ☐ Wertebereichskonflikt: keine Vereinheitlichung möglich
 - ☐ Bedingungskonflikt: verschiedene Sichten fordern eigene Integritätsbedingungen
 - ☐ Modellierungskonflikt: gleicher Sachverhalt ist unterschiedlich modelliert
3. Integration der Sichten in ein Gesamtschema bzw. ER-Diagramm

Bemerkungen:

- ❑ Ein Homonym ist ein Wort, das für mehrere Konzepte steht. Beispiel: Jaguar
- ❑ Synonyme sind verschiedene Wörter für dasselbe Konzept. Beispiel: {Haus, Gebäude}
- ❑ Beispiel für einen Typkonflikt: eine Patientin hat aus Sicht der Krankenkasse andere Eigenschaften als aus Sicht einer Ärztin.
- ❑ Beispiele für einen Bedingungskonflikt: verschiedene Schlüssel wurden (von verschiedenen Benutzergruppen) für dieselbe Menge von Objekten vorgesehen.

Entwurfsprozess

Phasenmodell: Verteilungsentwurf

Ziel:

Festlegung von Art und Weise einer verteilten Speicherung.

Beispiel:

Kunde (KdNr, Name, Adresse, PLZ, Kontostand)

- horizontale Verteilung:

Kundenstamm_1 (KdNr, Name, Adresse, PLZ, Kontostand)
mit $PLZ \leq 50\,000$

Kundenstamm_2 (KdNr, Name, Adresse, PLZ, Kontostand)
mit $PLZ > 50\,000$

- vertikale Verteilung:

Kunde_adr (KdNr, Name, Adresse, PLZ)
Kunde_kto (KdNr, Kontostand)

Zusammenhang kann über das Attribut `KdNr` hergestellt werden.

Entwurfsprozess

Phasenmodell: logischer Entwurf

Ziel:

Umsetzung des konzeptuellen Entwurfs in das Datenmodell des Realisierungs-DBMS, zur Zeit meist das relationale Modell.

Vorgehensweise:

1. teilweise automatische Transformation des konzeptuellen Schemas, z.B. des Entity-Relationship-Modells in das relationale Modell
2. Verbesserung des relationalen Schemas anhand von Gütekriterien durch entsprechende Normalisierungsalgorithmen. Stichwort: **Normalformen**

Ergebnis:

Ein logisches (relationales) Datenbankschema, das Datenredundanzen „weitgehend“ vermeidet und die Konsistenzbedingungen des Entity-Relationship-Modells „weitgehend“ erhält.

Entwurfsprozess

Phasenmodell: physischer Entwurf

Ziel:

Effizienzsteigerung **ohne** die logische Struktur der Daten zu verändern.

Konzepte:

„Tuning“ der Abbildung der Relationen auf den Sekundärspeicher:

- ❑ Definition von Indexstrukturen, die direkten (assoziativen) Zugriff auf alle Tupel einer Relation mit bestimmten Attributwerten erlauben.
- ❑ Clusteranalyse zur Gruppierung von Daten im Sekundärspeicher, so dass zusammen benötigte Daten auf denselben Seiten / in denselben Blöcken im Speicher liegen. Typisch insbesondere bei objektorientierten DBMS.

Bemerkungen [Schuerr 2001]:

- ❑ Kritik an der Phasenmodell-Vorgehensweise aus Sicht der Softwaretechnik:
 - es handelt sich um das überholte Wasserfallmodell der Softwaretechnik
 - Rückgriffe von einer Phase zu vorgehenden Phasen fehlen
 - Testaktivitäten sind nicht explizit aufgeführt
 - eine Wartung mit Aktivitäten aus allen Phasen ist unstrukturiert
 - inkrementelle bzw. schrittweise Realisierung eines DBS wird nicht unterstützt
 - Modellierung von Funktionen bleibt nahezu unberücksichtigt
 - Verzahnung mit Entwicklung sonstiger Teile eines Informationssystems fehlen
- ❑ Alternative Vorgehensweise:
 - objektorientierter Entwurf für das gesamte Informationssystem
 - Abbildung von Klassendiagrammen (statt ER-Diagrammen) auf Relationen
 - Verwendung modern(er) Vorgehensmodelle der Softwaretechnik

Datenbankmodelle

Definition 1 (Datenmodell)

Datenmodelle dienen zur Erfassung und Darstellung der Informationsstruktur für eine Anwendung oder einen Anwendungsbereich.

Datenbankmodelle

Definition 1 (Datenmodell)

Datenmodelle dienen zur Erfassung und Darstellung der Informationsstruktur für eine Anwendung oder einen Anwendungsbereich.

Beispiele:

- ❑ Typsysteme in Programmiersprachen
- ❑ Formalismen zur Wissensrepräsentation in Expertensystemen (Frames, Regeln, logische Formeln)
- ❑ Repräsentationsmodelle in Graphiksystemen (BRep, CSG)
- ❑ SPO-Tripel im Resource Description Framework des Semantic Web
- ❑ Datenbankmodelle in Datenbanksystemen (hierarchisches Modell, Relationenmodell, Entity-Relationship-Modell)

Datenbankmodell = Datenmodell für Datenbanksysteme

Datenbankmodelle

Definition 2 (Datenbankmodell, Datenbankschema [Heuer/Saake 2018])

Ein Datenbankmodell ist ein System von Konzepten zur Beschreibung von Datenbanken. Es legt Syntax und Semantik von Datenbankbeschreibungen für ein Datenbanksystem fest.

Eine konkrete Datenbankbeschreibung wird Datenbankschema genannt.

Datenbankmodelle

Definition 2 (Datenbankmodell, Datenbankschema [Heuer/Saake 2018])

Ein Datenbankmodell ist ein System von Konzepten zur Beschreibung von Datenbanken. Es legt Syntax und Semantik von Datenbankbeschreibungen für ein Datenbanksystem fest.

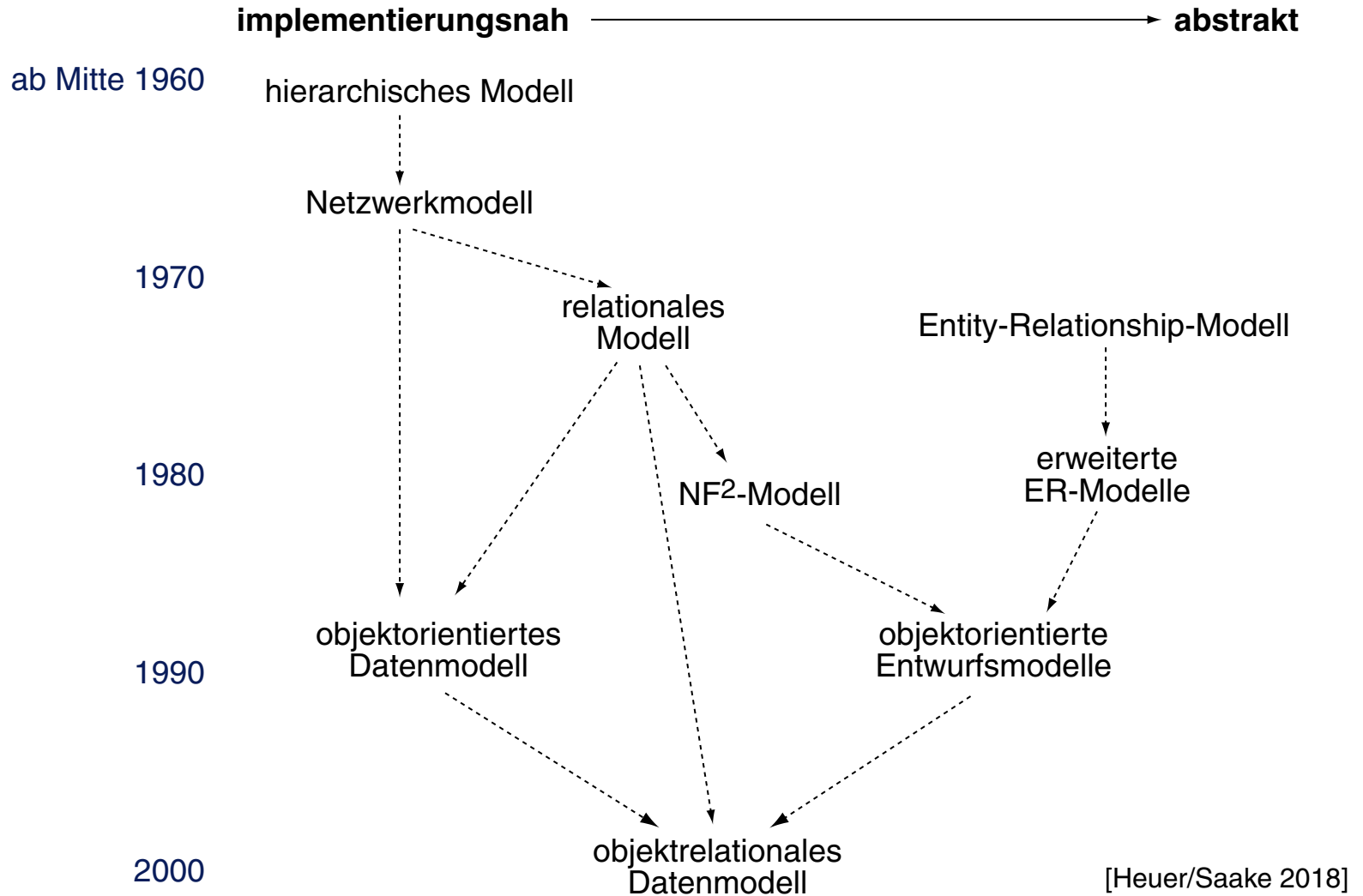
Eine konkrete Datenbankbeschreibung wird Datenbankschema genannt.

Gegenüberstellung mit Programmiersprachen:

Datenbank	Programmiersprache
Datenbankmodell <i>Relation, Attribut, ...</i>	Typsystem <i>class, int, String, ...</i>
Datenbankschema Kunde (KdNr, Name, ...)	Variablendeklaration <code>class Kunde { int KdNr; String Name; }</code>
Datenbank, DB (2305, "Meier", ...)	Werte 2305, "Meier"
Datenbankmanagementsystem DBMS	Entwicklungs- und Laufzeitumgebung
Datenbanksystem DBS = DB + DBMS	Programm zur Laufzeit

Datenbankmodelle

Historie



Datenbankmodelle

Datenbankschema

Das Datenbankschema einer Datenbank definiert für eine Anwendung:

1. statische Eigenschaften

- (a) identifizierbare Objekte (Basisdatentypen)
- (b) Beziehungen zwischen Objekten
- (c) Attribute von Objekten und Beziehungen

2. dynamische Eigenschaften

- (a) Operationen auf Daten
- (b) Abfolge und Koordination von Operationen

3. Integritätsbedingungen

- (a) an Objekte
- (b) an Operationen

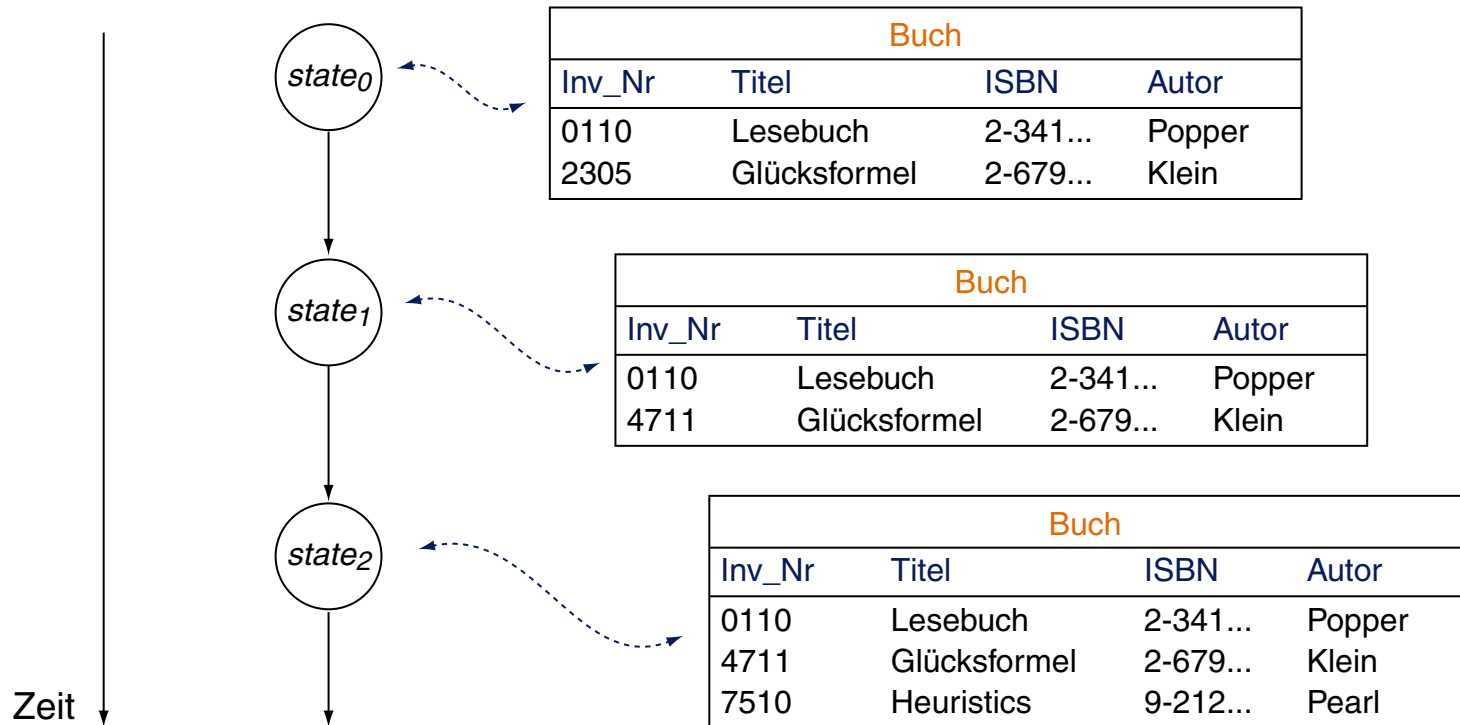
Bemerkungen:

- ❑ Beispiel für eine statische Eigenschaft: Sportlerinnen haben eine Nationalität.
- ❑ Beispiel für eine dynamische Eigenschaft: Ein Auto muss zugelassen werden, bevor man es fahren darf.
- ❑ Beispiel für eine Integritätsbedingung für Objekte: Studierende einer Universität haben unterschiedliche Matrikelnummern. (Schlüsselbedingung)
- ❑ Beispiel für eine Integritätsbedingung für Operationen: Bei einer Gehaltsänderung darf das Gehalt nur steigen. (Übergangsbedingung)

Datenbankmodelle

Datenbankzustand

Ein Datenbankzustand ist der zu einem Zeitpunkt t gültige bzw. gespeicherte Zustand aller Objekte und ihrer Beziehungen und muss den im Datenbankschema festgelegten Strukturbeschreibungen gehorchen.



Bemerkungen:

- ❑ Typischerweise ändert sich ein Datenbankschema selten; der Datenbankzustand ist Gegenstand laufender Modifikationen. Beispiel Flugbuchungssystem: Jede Reservierung entspricht einer Änderung des Datenbankzustands.
- ❑ Ein Datenbanksystem kann als kontinuierlich laufender Prozess aufgefasst werden, dessen jeweils aktueller Zustand *state* den Inhalt der Datenbank (Datenbasis) festlegt. Eine formale Definition der Semantik dieses Prozesses lässt sich durch eine lineare Folge von Zuständen modellieren, wobei die Zustandsübergänge den Änderungen der Datenbankinhalte entsprechen.