Modul Datenbanken

Vorlesung 2

Datenbanksysteme und Datenbankentwurf

IFI Wintersemester 2016/17

by Renzo Kottmann

Beim letzten Mal besprochen

- Was sind Daten?
- Welche Kategorien von Daten gibt es?
- Was ist eine Datenbank?

Ist das eine Datenbank?

Z.B. in einem Excel-File gespeichert:

```
Viereck;Axel;26123;Oldenburg;
Huber;Ina;12345;FFM;0123/65235
Lustig;Olga;12345;Frankfurt;0123/45456
Mustermann;Erika;12345;Frankfurt;0123/45456
Henseler;Herwig;26197;Großenkneten;04435/388486 (Fax:388487)
Lustig;Peter;Frankfurt;0123/45456
Huber;Ina;3454;Dresden;0283/11111
mustermann;erika;;Bremen;436654
```

- Ja
- Sehr simpel
- Wir haben sogar schon ein einfaches Datenbankprojekt gemacht

Schritte im Projekt

1. Problem in der realen Welt
    - Wer sind die Teilnehmerinnen in meinem Kurs?
2. Datenerfassung
    - Liste der Teilnehmerinnen
    - Implizites Model (Tabelle)
3. System zur Verarbeitung und Speicherung
    - CSV Datei

Was sind die Nachteile von diesem System?

Was wäre besser?

Was wäre besser?

1. Hohe Datenintegrität bzw. Konsistenz
2. Mehrere Nutzer können gleichzeitg an den selben Daten arbeiten
3. Effiziente Suche in grossen Daten

Für Datenintegrität

- Strukturierte Daten
- (relational) Datenbank (DB)
  - Schema („Modell“, „Metadaten“)
  - + Daten („Extension“, „Nutzdaten“)
  - Modell und Daten sind konzeptionell getrennt!

Für Effizienz und Mehrbenutzer

- Datenbankmanagementsystem (DBMS)
- Softwaresystem
  - ermöglicht Erstellen und die Pflege einer DB
  - stellt Werkzeuge für alle Aspekte der Datenverwaltung bereit

Datenbanksystem

- Datenbank (DB)

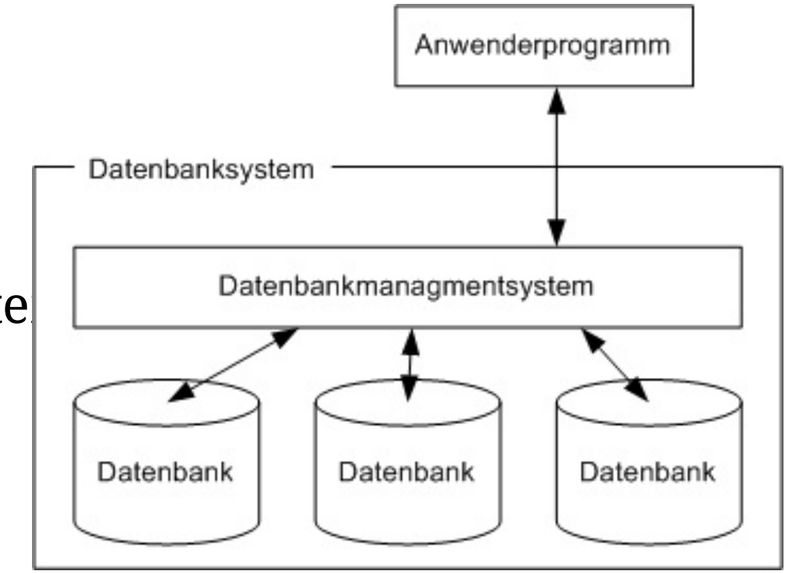
  Schema + Daten

- Datenbankmanagementsyste (DBMS)

  Softwaresystem zur Verwaltung

- Datenbanksystem

  DBMS + DB(s)



Definition: Datenbanksysteme

Ein Datenbank Management System oder vereinfacht ein Datenbanksystem ist eine spezialisierte Software sowohl zur langfristigen digitalen Erfassung als auch zur effizienten, sicheren Verarbeitung und Speicherung von Daten:

"Specialized software called a database management system, or DBMS, or more colloquially a "database system". A DBMS is a powerful tool for creating and managing large amounts of data efficiently and allowing to persist over long periods of time, safely. These systems are among the most complex types of software available.”

Heutige Anforderungen an Datenbanksysteme

1. Datenhaltung
2. Skalierbarkeit
3. Anfrageoptimierung (Performanz)
4. Datenintegrität
5. Mehrbenutzerfähigkeit
6. Schichtentrennung
7. Schnittstellen
8. Datensicherheit

Datenhaltung

- Große Datenmengen, welche nicht alle im Hauptspeicher gehalten werden können
- Kein Limit für die Anzahl der Zeilen in einer Tabelle
- Permanente Erreichbarkeit der Datenbank
- Verwendung mehrerer Tabellen.
  - Einfache Verwaltungsmöglichkeiten notwendig: Anlegen, ändern, löschen von Tabellen.
- Komplexe Datenmanipulationen: Anlegen, ändern, löschen von vielen Datensätzen gemäß nicht-trivialer Bedingungen.

Skalierbarkeit

Wird die Datenbank größer, so muss dies durch Einsatz von mehr Hardware ausgeglichen werden können:

- Nutzung vieler:
  - CPU- und Plattenressourcen durch
  - Multithreading, Multiprocessing und dynamisches Hinzufügen von Speicherplatz.
- Verteilbarkeit, Replizierbarkeit:
  - Zur Leistungssteigerung oder Datensicherheit können mehrere Instanzen einer Datenbank auf verschiedenen Rechnern gleichzeitig laufen.

Anfrageoptimierung (Performanz)

- Effiziente Abarbeitung von Suchanfragen durch schnelles Suchen in den Daten
- Hilfsstrukturen sollen dabei vermeiden, immer alle Daten durchsuchen zu müssen.

Datenintegrität

- Pflege von strengen Konsistenzbedingungen hinsichtlich
  - Datentypen
  - Redundanzfreiheit
  - Beziehungen der Daten untereinander
- Transaktionen
  - Werden mehrere Operationen hintereinander ausgeführt, so muss die Datenintegrität gewahrt sein
  - Beispiel: Kontobuchungen

Mehrbenutzerfähigkeit

- Viele Programme greifen gleichzeitig auf die gleichen Daten zu
  - Erkennen und Auflösen von Schreibkonflikten
- Zugriffsrechteverwaltung:
  - Nicht jeder Benutzer darf alle Daten sehen oder modifizieren

Schichtentrennung

- Physische Trennung von Datenbank und Anwendungssystemen auf verschiedener Hardware
- Plattformunabhängigkeit:
  - Eine Datenbank darf nicht von einem bestimmten Betriebssystem abhängen
- Möglichkeit, eine zentrale Datenbank für viele Anwendungen zu schaffen
- Trennung von Datenbank und Anwendungslogik:
  - Die Daten müssen unabhängig von einer konkreten Anwendung gespeichert sein

  Programme sind vergänglich und austauschbar, Daten und deren Struktur sind wesentlich langlebiger und wertvoller

Schnittstellen

- Komplexe Suchanfragen erfordern eine eigene Suchanfragesprache
- Vielfältige und offene Schnittstellen zu allen möglichen Programmiersprachen oder Anwendungen

Datensicherheit

- Alle Änderungen müssen sofort dauerhaft gespeichert sein ohne Datenverlust

Relationale Datenbank PostgreSQL

http://www.postgresql.org/

- Open Source
- Sehr nah am SQL-Standard
- Sehr gute Dokumentation

Client / Server Sicht

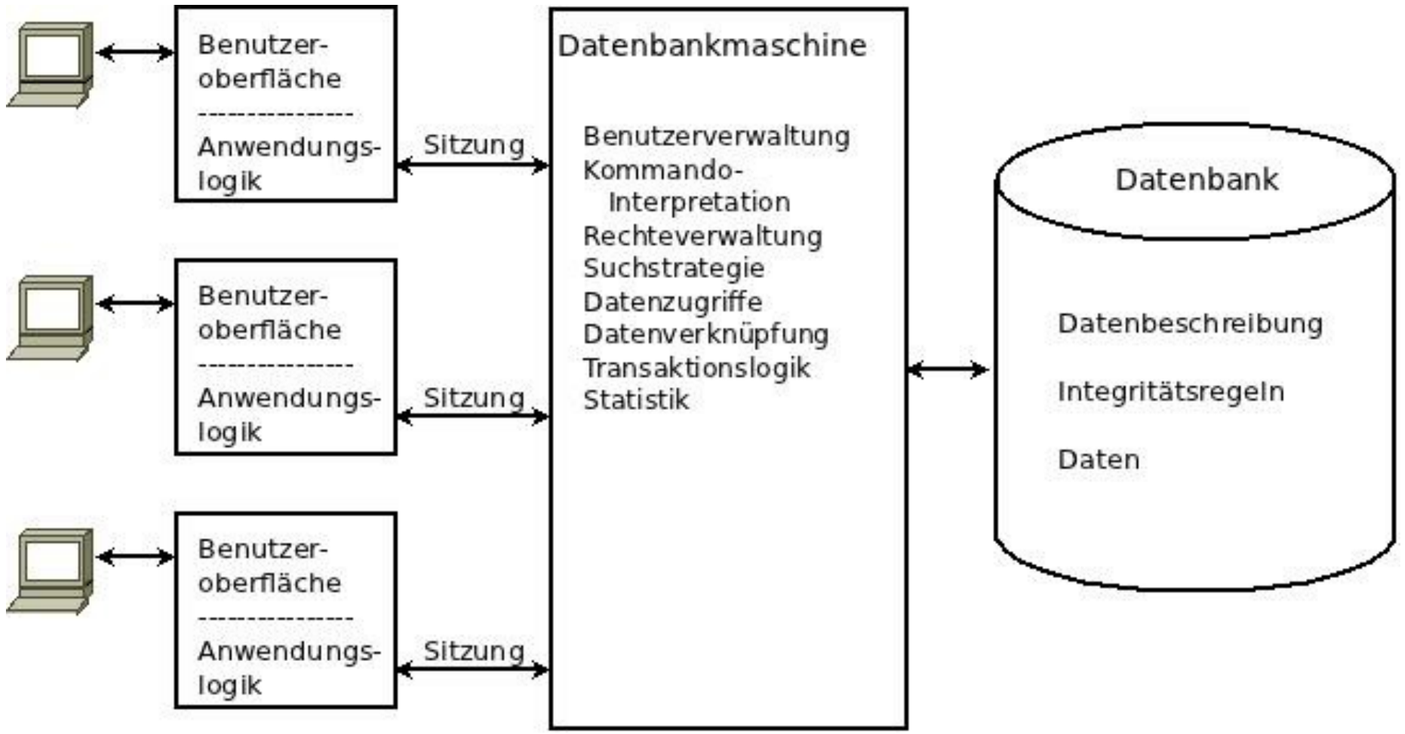


Abb. 2.1 aus Matthiessen

Datenmodell

System von Konzepten zur abstrakten Darstellung eines Ausschnitts der realen Welt mittels Daten

- besteht aus
  - generischer Datenstruktur
  - Operatoren
  - Integritätsbedingungen

Relationales Datenbankmodell

Generische Datenstruktur

- Relationen mit eindeutigen Namen
  - jede Relation ist eine Menge von Tupeln (Datensätzen) gleichen Typs
  - Die Struktur ist insofern generisch, als die Relationen und ihre Attribute (Spalten) beliebig gewählt werden können bzw. beim Einrichten der Datenbank angegeben werden müssen.

Operatoren

- relationale Algebra
- Daten
  - eintragen
  - ändern
  - löschen
  - abfragen
  - ableiten

Integritätsbedingungen

- Bestimmung von ein-eindeutigen Tupeln
- Einschränkungen vom Wertbereich bestimmter Datentypen

https://de.wikipedia.org/wiki/Datenbankmodell

2. Runde

Mit dem relationalem Ziel vor Auge

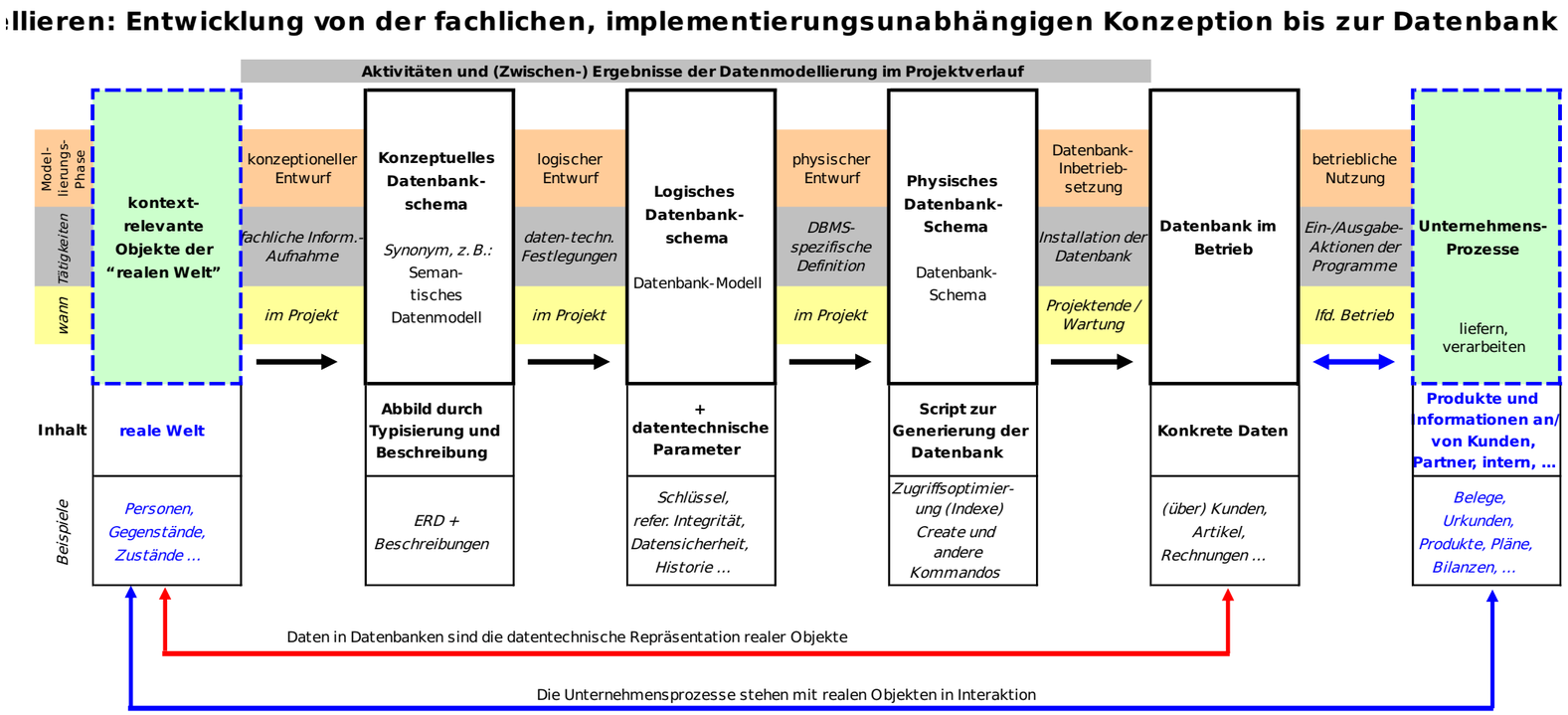
Schritte im Projekt

1. Problem in der realen Welt
    - Wer sind die Teilnehmerinnen in meinem Kurs?
2. Datenerfassung
    - Liste der Teilnehmerinnen
    - Implizites Model (Tabelle)
3. System zur Verarbeitung und Speicherung
    - CSV Datei

Schritte im Projekt

1. Problem in der realen Welt / Anwendungsdefinition
    - Wer sind die Teilnehmerinnen in meinem Kurs?
2. Konzeptioneller Entwurf / Datenbankmodellierung
3. Datenbank Realisierung
    - Implementierung des Modells
    - Verabreiten der Daten
4. System zur Verarbeitung und Speicherung
    - Datenbank im Betrieb: PostgreSQL

Alternativer Blick: Datenbankentwicklungszyklus



„DatMod v semMod zur DBK“ von VÖRBY, Konertierung zu SVG Perhelion - eigene Erstellung, aus Wikipedia-Text abgeleitet. Lizenziert unter Gemeinfrei über Wikimedia Commons.

Konzeptioneller Entwurf

In der Phase der Systemanalyse werden ausgehend von der Problemstellung die Anforderungen an die Lösung formuliert. Diese sollten möglichst vollständig und konsistent sein, d.h. alle (vollständigen) Anforderungen sollten widerspruchsfrei (konsistent) formuliert werden.

Ein guter Einstieg in die Anwendungsdefinition ist, sich zu verdeutlichen, welche genaueren Zwecksetzungen erfüllt werden sollen.

Anwendungsdefinition

Welchen Zwecken soll die Erfassung aller Kursteilnehmenrinnen dienen?

Anwendungsdefinition

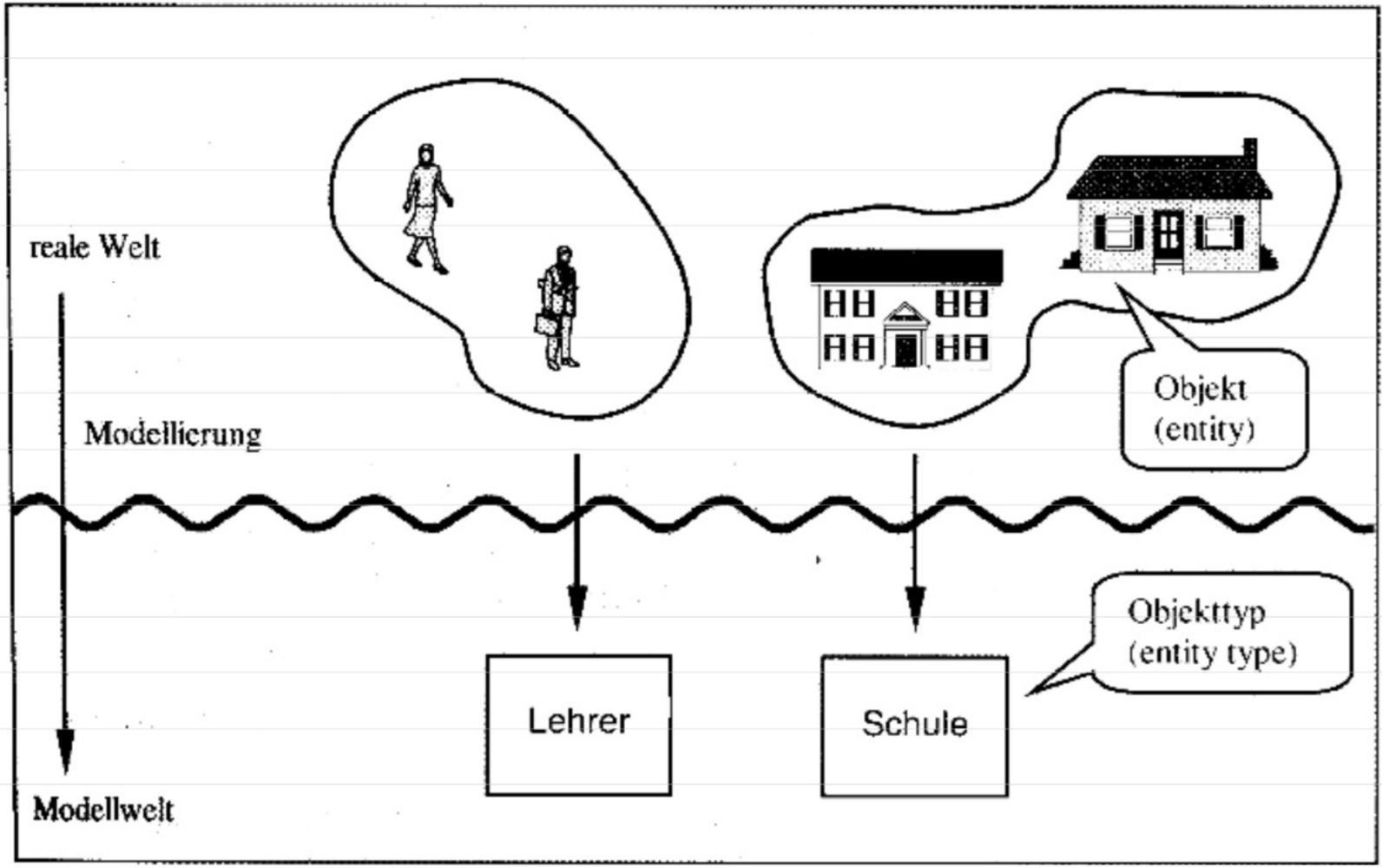
Welchen Zwecken soll die Erfassung aller Kursteilnehmenrinnen dienen?

- Informationsweitergabe an alle Teilnehmerinnen
- Namen lernen
- Notenvergabe
- Ermittlung des Wissenstands
- Wer macht welches Datenbankprojekt

Mini-Welt

(wenn) wir wissen was wir wollen, dann können wir folgende Fragen beantworten:

- Welchen Ausschnitt der realen Welt brauchen wir?
- Welche Aspekte müssen wir Berücksichtigen?


reale Welt
Modellierung
Modellwelt
Lehrer
Schule
Objekt
(entity)
Objekttyp
(entity type)

Entity Relationship Modellierung (ERM)

- Semantischer Datenbankentwurf
  - unabhängig von konkreten Datenbank-spezifischen Modellen
  - dienen zum Entwurf von relationalen, Netzwerk- oder Objekt-Datenbanken
- Graphisch
- Idee simpel
  - Leider sehr viele inkonsistente Varianten

Entity (Entität)

- Ein Entity-Relationship-Model (ERM) geht von Entitäten (~= Objekten) aus.

  "Eine Entität ist eine eigenständige Einheit, die im Rahmen des betrachteten Modells **eindeutig identifiziert** werden kann."

- Ein Entitätstyp wird durch Attribute genauer beschrieben und stellt somit eine abstrakte Beschreibung oder Charakterisierung von Entitäten da.

- Beispiel:

  Lehrer = Entitätstyp

  Renzo ~= Entität (ein spezieller Lehrender)

Attribute

- Eigenschaften von Entitäten werden durch Attribute beschrieben
- Attribute haben einen Namen und eine Domaine (= Bestimmung der Wertmenge).

Keys (Schlüssel)

Da die Definition einer Entität beinhaltet, dass diese zumindest im Rahmen eines Modells eindeutig identifiziert werden kann, braucht jeder Entitätstyp eine Menge von Attributen als Schlüssel.

Die Auswahl eines oder meherer Attribute als Schlüssel legt fest, dass es keine zwei Entitäten eines Entitätstyp geben kann die identische Attributwerte haben.

- Wichtige Eigenchaften:
  - Eindeutigkeit
  - Zuteilbarkeit

Relationship (Beziehung)

Verschiedene Entitäten können zueinander in Beziehung gesetzt werden.

- In jeder Beziehung haben Entitäten gewisse Rollen
- Beziehungen können Eigenschaften (Attribute) haben
- Beziehungen haben Kardinalitäten

Notationen

Es gibt verschiedene Formen ERM zu notieren (textuell und/oder graphisch):

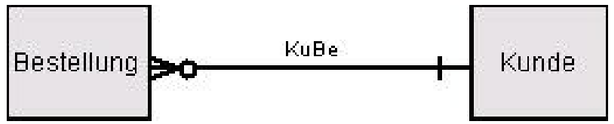
- Chen Notation
- Crow Foot's
- Unified Modelling Language (UML)

Notation

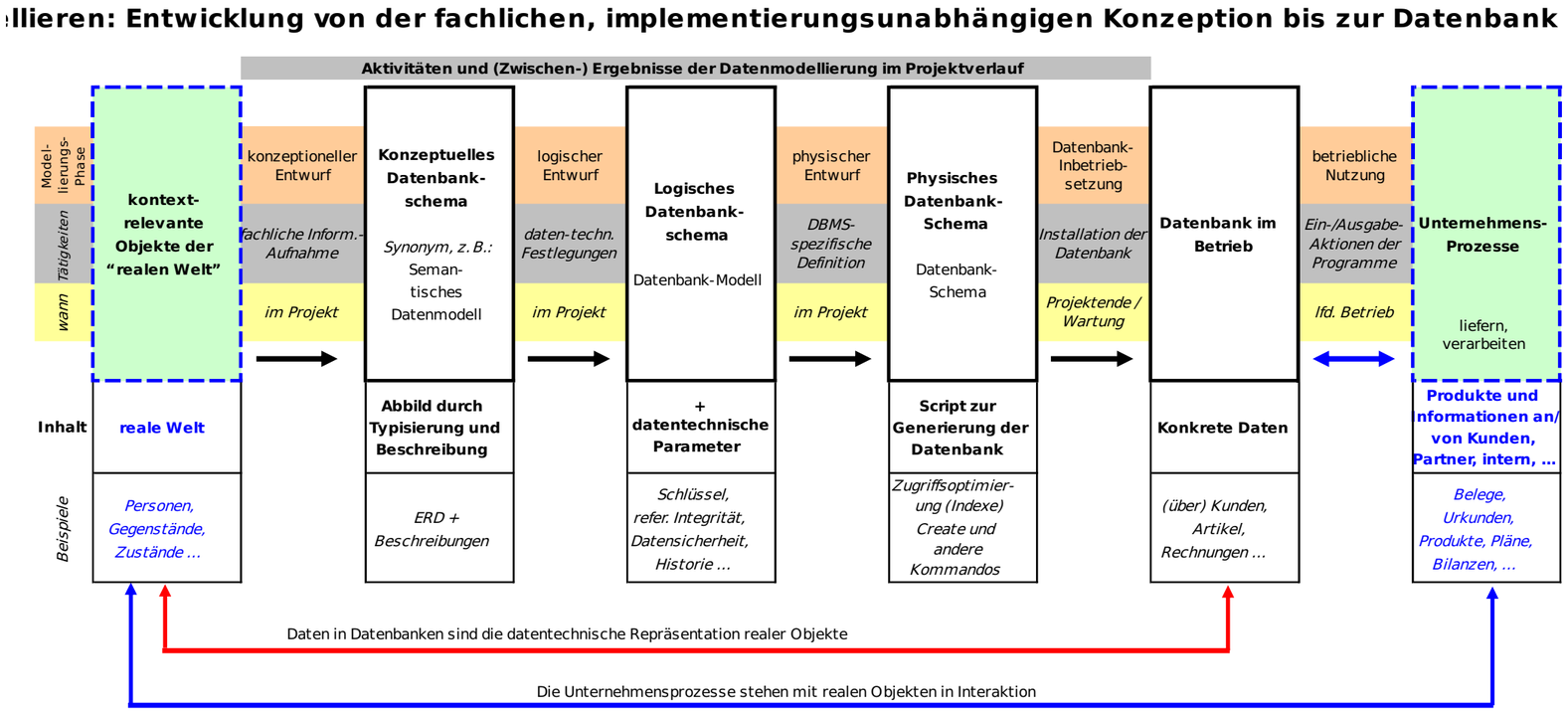
Entity

Attribute

Beziehungen/Relationship



Alternativer Blick: Datenbankentwicklungszyklus



„DatMod v semMod zur DBK“ von VÖRBY, Konertierung zu SVG Perhelion - eigene Erstellung, aus Wikipedia-Text abgeleitet. Lizenziert unter Gemeinfrei über Wikimedia Commons.

Structured Query Language (SQL)

SQL ist eine Datenbanksprache

1. zur Definition von Datenstrukturen/Modellen
2. zum Bearbeiten (Einfügen, Verändern, Löschen)
3. zum Abfragen von darauf basierenden Datenbeständen
4. zur Rechtevergabe

SQL Eigenschaften

- basiert auf relationaler Algebra
- an English angelehnt
- Deklarativ und funktional
- Fast alle Datenbanken verstehen SQL
- Standardisiert
  - PostgreSQL hat einer der besten Umsetzungen

Danke für die Zusammenarbeit

Referenzen:

- M. Unterstein and G. Matthiessen, Relationale Datenbanken und SQL in Theorie und Praxis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

Natural Keys

Ein natürlicher Schlüssel ist die Menge der Attribute einer Entität, die diese Entität eindeutig identifiziert (natürlich da es nur einer Auswahl der schon eh definierten Attribute braucht).

```
* Diese können an der Realität überprüft werden
```

"Intelligente Schlüssel" basieren häufig auf Industriestandards und können mit Hilfe externer Ressourcen überprüft werden. Z.B: ISBN, EAN, VIN, UPC...

Artificial Keys

Artifizielle, synthetische Schlüssel sind zusätzlich hinzugefügte Attribute die für die eindeutige Identifizierung einer Entität benutzt werden.

```
* Können nicht an der Realität überprüft werden
```

Auto-Numbers (Increment, serials) sind schlicht keine relationale Schlüssel, werden aber oft als artificial keys benutzt.

Natural KeyArtificial KeyTeil des DatenmodelsJaNein

Entitätstyp:

Entitätstyp mit Attributen:

Beziehungen: Kardinalitäten

Zweistellige Beziehungen

Zur Erläuterung einer Beziehung ist es wichtig, die Beziehung getrennt nach beiden Richtungen zu lesen.

Vom ERM zu einem Datenbankschema

Auf dem Papier sehen sich die Darstellung vom Datenbankschema und ERM sehr ähnlich.

Umsetzung von 1:m Beziehungen Von einem zu vielen.

Umsetzung von n:m von vielen zu vielen

Beziehungen bei abhängigen Entitätstypen (weak entities)

Eine Entität heißt abhängige Entität oder engl. weak entity, wenn diese nicht durch eigene Attribute identifiziert werden kann.

Beispiel: Kinder von Mitarbeitern, bei denen die Firma die Kindergeldzahlung verwaltet. Im Beispiel werden Kinder nur abhängig vom Mitarbeiter eindeutig identifiziert.