

Zusammenfassung DBI 3. Klasse

Michael Briedl

Stand:

12.11.2017

Bis Folie 265

Inhalt

1.	Datenbanken	4
1.1	Grundlegendes zu Datenbanken	4
1.1.1	Arten von Datenbanken	4
1.1.2	DB-Geschichte	4
1.1.3	NoSQL	4
1.1.4	CAP-Theorem.....	5
1.1.5	Spaltenorientierte DB.....	5
1.1.6	Hadoop	5
1.1.7	SPRAIN	5
1.2	Datenbankmanagementsysteme	5
1.2.1	Filesystem (DB ohne DBMS)	5
1.2.2	Was ist ein DBMS?	5
1.2.3	Vorteile eines DBMS.....	6
1.3	Geschichte	6
1.3.1	Oracle	6
1.3.2	Festplattengeschichte	6
2	Datenbankmodellierung.....	7
2.1	3 Sichten auf Daten	7
2.1.1	Konzeptionelles Schema.....	7
2.1.2	Internes Schema	7
2.1.3	Externes Schema	7
2.1.4	Implementierung eines 3-Schema-Systems	7
2.2	Entity Relationship Diagrams	8
2.2.1	Entities.....	8
2.2.2	Entity-Sets.....	8
2.2.3	Attribute	9
2.2.4	Naming Conventions	9
2.2.5	Relationships	9
2.3	Von den Anforderungen zum ERD.....	10
2.3.1	Heuristiken zu Entities.....	10
2.3.2	Heuristiken zu Attributen	10
2.3.3	Heuristiken zu Relationships	11
2.4	Vom ERD zur Tabelle	11
2.4.1	Relationales Modell.....	11

2.4.2	Primary Keys	12
2.4.3	Foreign Keys	12
2.4.4	Vorgehen	12

1. Datenbanken

1.1 Grundlegendes zu Datenbanken

1.1.1 Arten von Datenbanken

- Nach Struktur
 - Hierarchisch
 - Netzartig
 - Relational
 - Objektrelational
 - Objektorientiert
 - XML
- Nach Nutzeranzahl
 - Single-User
 - Multi-User
 - Rechnernetzwerk
- Nach Nutzungsform
 - Abfrage-DB
 - Transaktions-DB
 - Analyse-DB
 - ETL
 - Extract, Transform, Load
 - Daten laden, verändern und in DB schreiben
- Nach Abfragesprache
 - SQL
 - NoSQL
- Nach Speichernutzung
 - Massenspeicher (Standard)
 - In-Memory
- Die BESTE Datenbank
 - Anforderungsbedingt
 - Relationale/objektrelationale am häufigsten verwendet

1.1.2 DB-Geschichte

- Edgar Frank Codd
 - "A Relational Model of Data for Large Shared Databanks"
 - Abfragesprachen
 - Alpha
 - Query by Example
 - SEQUEL (Structured English Query Language)

1.1.3 NoSQL

- In manchen Fällen schneller
- Ansonsten unpraktisch wegen fehlender Konsistenz

1.1.4 CAP-Theorem

- Consistency
- Availability
- Partition Tolerance (Verteilung ohne Ausfall)

1.1.5 Spaltenorientierte DB

- Zugriff über Spalten möglich
- Nicht an Zeilen gebunden

1.1.6 Hadoop

- Daten auf 64MB Blöcke aufgeteilt
- Verteilt auf verschiedene Server über Kontinente verteilt
- Mehrfachsicherung möglich
- Kein DBMS

1.1.7 SPRAIN

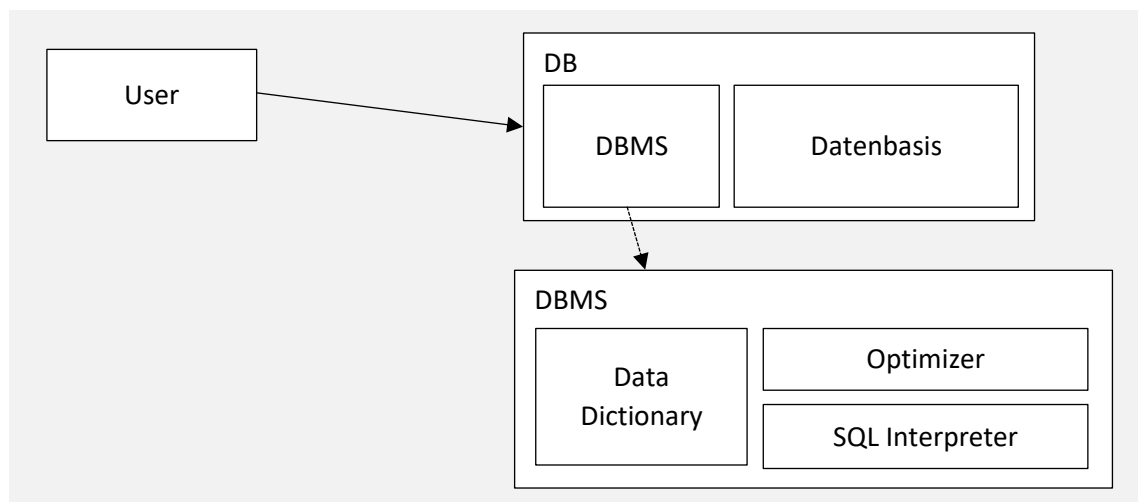
- Scalability
- Performance
- Relaxed consistency (siehe CAP-Theorem)
- Agility („Mut“ zu neuen Systemen)
- Intricity (Datenkapazität)
- Necessity (Gebundenheit an Hersteller, Sicherheit dass Hersteller bestehen bleibt)

1.2 Datenbankmanagementsysteme

1.2.1 Filesystem (DB ohne DBMS)

- Redundant
- Schwierige Aktualisierung
- Sperre auf Dateiebene
- Bei Änderungen muss jedes Programm geändert werden
- Routinen (Suchen, Filtern) sind in jedem Programm zu implementieren
- Datenschutz schwierig

1.2.2 Was ist ein DBMS?



1.2.3 Vorteile eines DBMS

- Wenig Redundanz
- Integrität (der Daten)
- Datensicherheit, -unabhängigkeit (von DBMS verwaltet, nicht Programm)
- Datensichten möglich (Views; nur bestimmte Teile von Daten sichtbar für bestimmte User)
- Effizient

1.3 Geschichte

1.3.1 Oracle

- 1979: Queries, Joins
- 1983: Transaktionen
- 1985: Client-Server Architektur
- 1992: Foreign Key Beziehungen, Stored Procedures
- 1997: objektorientierte Daten, BLOB
- 2003: Regex, Recycle-Bin

1.3.2 Festplattengeschichte

- „Geburtstag“: 4.11.1959
- Kapazität erster Festplatte: 5MB
- GB-Grenze geknackt: 1997
- TB-Grenze: 2007
- Erste SSD: 1978
- HDD-Höchstgrenze 2017: 12TB
- SSD-Höchstgrenze: 15TB
- 60TB SSD bis Ende 2017 angekündigt (von Seagate)

2 Datenbankmodellierung

2.1 3 Sichten auf Daten

2.1.1 Konzeptionelles Schema

- Logisches Schema (Modell)
- Unabhängig von anderen Schichten
- Aufgaben des konzeptionellen Schemas
 - Beschreibung der logischen Dateien und des Satzaufbaus
 - Beschreibungen der Beziehungen (**Relationships**) zwischen Tabellen
 - **Relationen** sind Referenzen in Datensätzen
 - Beschreibung der Felder
 - Beschreibung der Gültigkeitsbereiche
 - Zum Beispiel: „Alter“ gilt nur zwischen 0 und 150

2.1.2 Internes Schema

- Dazu gehören: DBMS, Datenbasis
- Aufgaben
 - Repräsentation von Zahlen (binär, dezimal, ...)
 - Repräsentation von Zeichen (Unicode, ASCII, EBCDIC, ...)
 - Speicherung von Datensätzen und Zugriff auf Datensätze
 - Hashes, sequentielle Indizes, Pfade, Verzweigungen, VSAM, Clustering, ...

2.1.3 Externes Schema

- UI, Druckvorlagen
- Nur Teilsichten
- Auch Teil d. externen Schemas: Views
 - Vorgefilterte Tabellen

2.1.4 Implementierung eines 3-Schema-Systems

- Ziel: gutes konzeptionelles Schema
 - Andere können fast automatisch generiert werden
- Vorgehensweise
 - Anforderungsanalyse
 - Ergebnis: unstrukturierte Daten
 - Hauptsächlich Anforderungen für externe Sichten
 - Ordnen
 - Synonyme, Homonyme beseitigen
 - Wörter mit gleicher Bedeutung
 - Dabei hilft CASE-Software (Computer Aided Software Engineering)

2.2 Entity Relationship Diagrams

- Kurz ERD
- Von **Peter Chen** (chinesischer Name: Chen Pin-Shan) und **Matt Flavin** entwickelt

2.2.1 Entities

- „Ding“/Objekt
 - Begriff „Daten“ nicht passend: gehört zu internem Schema

DEFINITION

Individuelles und identifizierbares Exemplar von Dingen, Personen oder Begriffen der realen oder der Vorstellungswelt

- Können sein:
 - Dinge
 - Organisationen
 - Personen
 - Ereignisse
 - Grundsätze
 - Selten: wichtige Beziehungen
- Stärke eines Entities
 - **Fundamental**
 - Auch Kernentität oder Regular Entity genannt
 - Können alleine bestehen
 - **Abhängig** bzw. **Weak**
 - Brauchen Fundamentalentität um bestehen zu können
- Grafische Darstellung verhindert Missverständnisse

2.2.2 Entity-Sets

- Enthalten Entities mit gleichen oder ähnlichen Merkmalen
- Werden oft (unsauber) mit Entities gleichgesetzt
- Können disjunkt oder überlappend sein:



2.2.3 Attribute

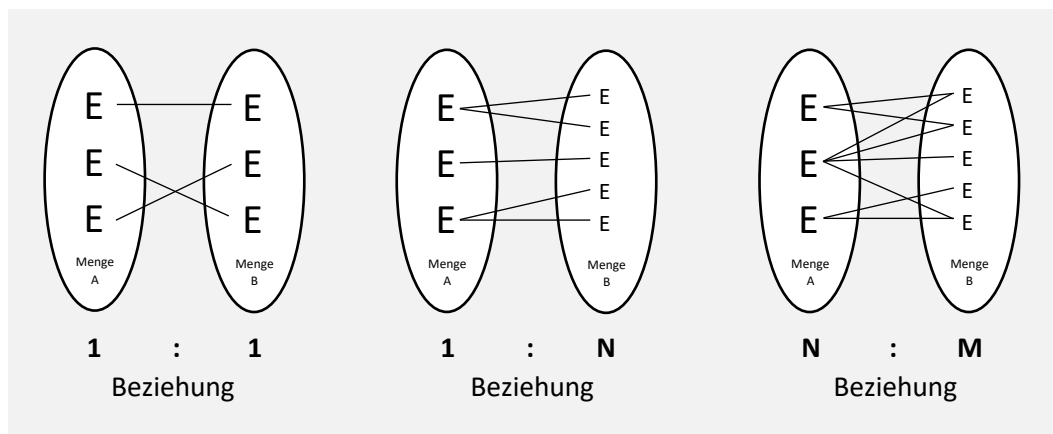
- Beschreiben Eigenschaft/Wert eines Entities
- Kriterien für Attribute:
 - Attribut zu Entity = 1:1
 - Attribut besitzt keine Attribute
 - Attribut ist nur Attribut des betroffenen Entities
- Wenn Kriterien nicht erfüllt -> Attribut wird zu Entity
- Haben Wertebereiche (z.B.: ASCII-Zeichen, alle Wochentage, 0 bis 100, ...)

2.2.4 Naming Conventions

- Singular
- Abkürzungen vermeiden
- Möglichst wenig Fachsprache
- Einheitlich bleiben
- Synonyme/Homonyme vermeiden
- Beschreibungen ins **Data Dictionary** (Metadatenkatalog) eintragen
- Keine...
 - Leerzeichen
 - Nicht-englischen Schriftzeichen (ß, Umlaute usw.)
- Hungarian Notation ist üblich

2.2.5 Relationships

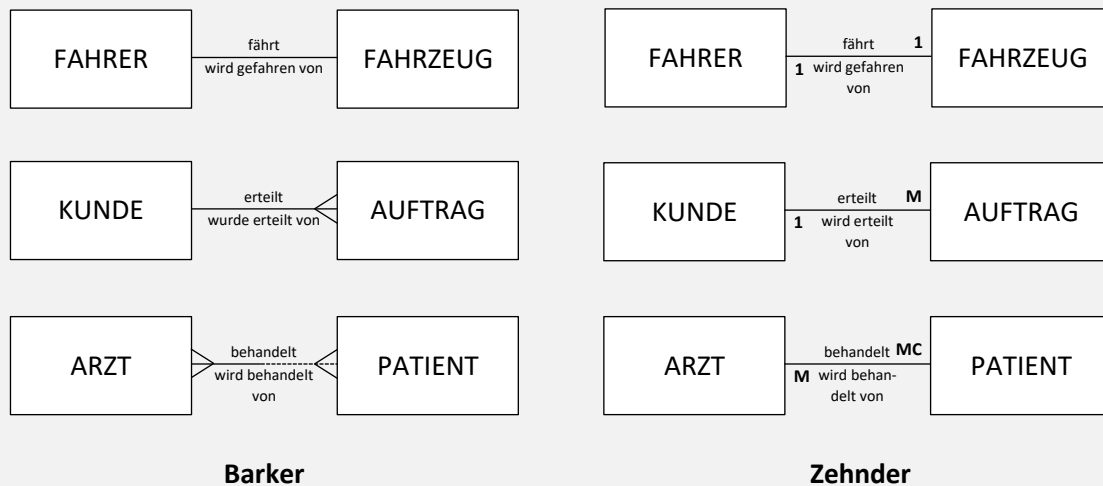
- Verbindungen zwischen Entities
- Legen fest, wieviele Entities aus Menge A Entities aus Menge B zugeordnet sind



- Bestehen aus jeweils 2 **Assoziationen**:

BEZEICHNUNG		ANZAHL DER ENTITIES
1	Einfache Assoziation	1
C	Konditionelle Assoziation	0 oder 1
M	Multiple Assoziation	≥ 1
MC	Multiple konditionelle Assoz.	≥ 0

Notationen



Barker

Zehnder



Chen-Notation

Benannt nach Peter Chen
Eher selten

2.3 Von den Anforderungen zum ERD

- 2 Möglichkeiten:
 - Entities finden, ins Modell eintragen -> Attribute hinzufügen -> auf Normalform prüfen
 - Attr. existieren bereits (durch externe Schemen) -> Entities bilden -> auf NF prüfen
- Wichtigste Normalform: 3NF

2.3.1 Heuristiken zu Entities

- Beschrieben mit Hauptwort
- Man soll etwas „im Auge behalten“
- Wenn unklar ob Attribut, Entity oder Relationship -> Entity
- Wenn Attribut mit -name, -nummer oder -code endet -> Schlüsselattribut eines Entities
- Wenn zu Attribut Definition besteht, die auf Entity hinweist

2.3.2 Heuristiken zu Attributen

- Wenn etwas Wert annehmen kann
- Wenn Attributsdefinition sich auf Entity bezieht -> Entity zuordnen
- Wenn Def. sich auf mehrere Entity-Sets bezieht -> **Relationship-Set** erstellen, um Entity-Sets zu verbinden -> Attribut dem Relationship-Set zuordnen
 - Relationship-Sets werden wie Entity-Sets behandelt
- Wenn Attribut nicht zu allen Instanzen eines Entity-Sets passt -> Entity-Set unterteilen

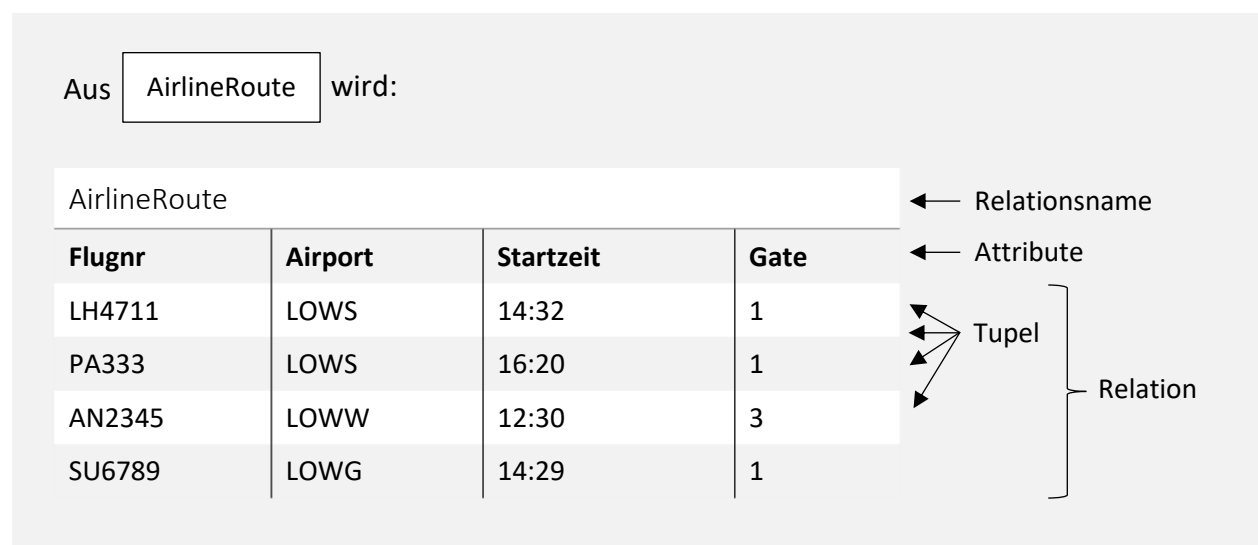
2.3.3 Heuristiken zu Relationships

- Beschrieben mit Verb
- Alleinstehendes Entity-Set -> Relationships zu anderen Entity-Sets suchen
- 2 Attribute von verschiedenen Entities sind nahe beisammen
- Eine Definition enthält mehr als ein Entity
- Mehrere eindeutige Schlüssel in Datensatz

2.4 Vom ERD zur Tabelle

2.4.1 Relationales Modell

- Datenbanknahe
- Von Edgar F. Codd entwickelt



- Begriffe:
 - **Tupel:** Tabellenzeile/Datensatz
 - **Tabelle:** Menge von Tupeln
 - **Kardinalität:** Anzahl der Tupel
 - **Grad:** Anzahl der Attribute
- Tabelle (Relation)
 - Menge von Tupeln
 - Eindeutiger Name
 - Normalerweise >1 Zeile
 - Ordnung der Zeilen egal (über Werte angesprochen)
 - Spalten (Attribute) auch >1, Ordnung egal

- Mathematisch

DEFINITION

Eine Relation R ist eine Teilmenge eines kartesischen Produktes über n nicht disjunkte Wertebereiche W_i

Eine Relation ist eine Menge von Tupeln der Form (w_1, w_2, \dots, w_n) mit $w_i \in W_i$

- Begriffe:
 - Kartesisches Produkt: Menge aller Wertekombinationen von W_1 bis W_n

2.4.2 Primary Keys

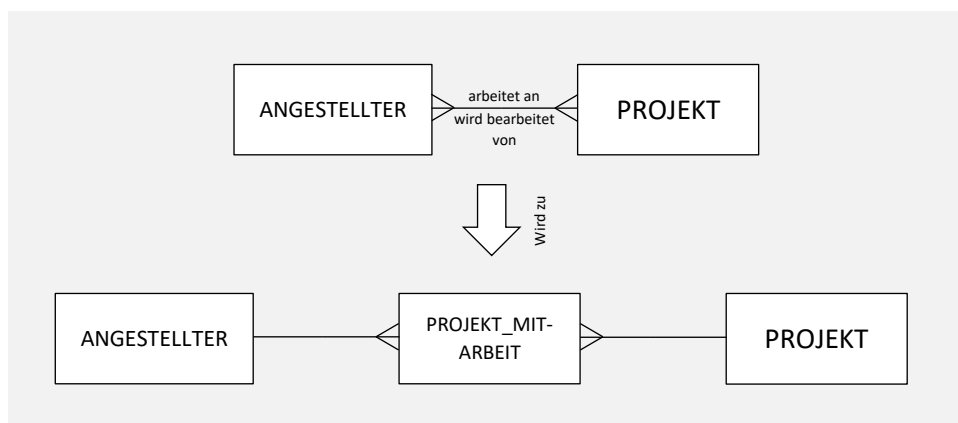
- **Candidate Keys (CK)**
 - „Kandidaten“ für Primary Key
 - Hat jede Tabelle min. einmal
 - Können aus mehreren (nach wie vor getrennten) Spalten bestehen
- **Primary Key (PK)**
 - Einer pro Tabelle
 - Aus CK ausgewählt
 - Sollte leicht schreibbar und sprechend sein
- **Alternate Keys (AK)**
 - Alle CKs, die nicht PK wurden

2.4.3 Foreign Keys

- Gewöhnliche Attribute
- Zeiger auf Zeile/Datensatz in anderer Tabelle
 - Dort ist er PK
- Können (natürlich) Teil des PK sein
- Können auch auf Zeilen in eigener Tabelle zeigen -> Hierarchie
- Kann NULL sein (außer wenn in PK)

2.4.4 Vorgehen

- Aus jeder Entitätsmenge wird Tabelle (braucht PK)
- 1:N Relationships werden zu FK in N-Tabelle
- Aus M:N Relationships werden assoziative Tabellen



Beispiel Bestellung

Verhältnis Bestellung zu Artikel = M : N

**Primary Key:**

BEST_ART(BestNr, ArtNr, ...) -> Nicht ideal: Datensatz muss gelöscht werden, um Bestellung zu bearbeiten

BEST_ART(BestNr, LfdNr, ArtNr, ...) -> Besser: ArtNr kann geändert werden

BEST_ART(LfdNr, BestNr, ArtNr, ...) -> Nicht ideal: Bestellung 119 könnte LfdNr 236, 354 und 512 haben (also völlig zusammenhangslos)

- Attribute werden zu Spalten
- Relationships zu Foreign Keys