# Datenbanken: Was muss ich wissen?

Prof. Dr. Jens Dittrich

18. Juli 2013, v0.04

## 1 Über dieses Dokument

Warum sind in diesem Dokument für jedes Kapitel die wichtigsten Lernziele als Fragen formuliert? Warum sollten Sie jede dieser Fragen spontan beantworten können?

Dieses pdf enthält Links zu den öffentlich zugänglichen Videos auf youtube aus der Vorlesung Informationssysteme, SoSe 2013, Informatik, Uni Saarland. Zusätzliches Material ist frei verfügbar auf datenbankenlernen.de. Z.B. für die Fotoagentur das Datenbankschema, Beispieldaten, das E/R-Modell und das Relationale Modell.

Alle Angaben ohne Gewähr. Bugs, Themenwünsche, Lob, etc. an Prof. Dr. Jens Dittrich.

## 2 Einführung

Welche Themen werden in dieser Vorlesung behandelt? Welche Termine und administrativen Randbedingungen gibt es? Was ist eine Studienarbeit? Wo finde ich weiterführende Literatur?

13.01 Vorlesung Informationssysteme

13.02 Administratives: Wie funktioniert ein Flipped Classroom?

13.19 Studienarbeiten

13.03 Literatur zu Datenbanksystemen

13.05 Das neue iPad 11

## 3 Motivation

Welche Vor- und Nachteile haben Datenbanksysteme gegenüber Dateisystemen? Welche Abstraktionsebenen sind dabei entscheidend? Was sind physische und logische Datenunabhängigkeit? Was sind Metadaten? Wie gelange ich von einem Szenario aus der realen Welt zu einer konkreten Datenbank? Was sind die verschiedenen Modellierungsschritte auf dem Weg dorthin?

13.04 Warum Datenbanksysteme?

13.06 Physische und Logische Datenunabhängigkeit

13.07 Übersicht über die Modellierungsschritte: von der Realität zum Datenbankschema

# 4 Entity Relationship Modellierung

Was macht dieser Modellierungsschritt prinzipiell? Warum ist ein Pflichtenheft in der Praxis wichtig? Was sind die wesentlichen Elemente von E/R? Was kann ich modellieren und was nicht? Was kann ich mit E/R ausdrücken und was nicht? Was sind Funktionalitäten? Was ist eine Rolle? Warum ist die Rekursion hier nicht gleich der Rekursion aus Programmiersprachen? Was sind schwache Entitätstypen und was ist deren Schlüssel? Was hat das mit N:M-Beziehungen zutun? Was ist eine Generalisierung und wie unterscheidet sich das von einer Komposition (ist-Teil-von)? Wie unterscheidet sich die Chen-Notation von min/max? Wie lese ich die beiden Notationen? Wozu braucht man nochmal mehrstellige Beziehungstypen? Warum brauche ich Sichtenkonsolidierung?

13.08 Entity Relationship Modellierung: Grundlagen, Funktionalitäten, Rollen, Rekursion

13.10 Entity Relationship Modellierung III: schwache Entitätstypen, N:M, Generalisierung, Teil-von

13.09 Entity Relationship Modellierung II: Chen versus min/max, mehrstellige Beziehungstypen

13.11 Entity Relationship Modellierung IV: Sichtenkonsolidierung

# 5 Relationales Modell

Was sind die wesentlichen Elemente des Relationalen Modells? Was ist der Unterschied zwischen einer Ausprägung und dem Relationenschema? Und warum ist Schema Singular von Schemata? Was ist ein Schlüssel? Was kann ich im Relationalen Modell modellieren und was nicht? Wie erfolgt die Übersetzung eines E/R-Modells zum Relationalen Modell? Was macht dieser Modellierungsschritt prinzipiell?Wie fasse ich Relationen zusammen? Was kann (und darf aber nicht) dabei schiefgehen? Welche Beziehungstypen werden wie umgesetzt? Wie werden schwache Entitätstypen umgesetzt? Welche Optionen gibt es für die Umsetzung der Generalisierung? Wie unterscheiden sich Relationen von Tabellen?

13.12 Das relationale Modell: Relationen, Domänen, Relationenschema, Schlüssel

13.13 Umsetzung ER nach Relationalem Modell: Grundlagen, binäre und mehrstellige Beziehungstypen

13.14 Verfeinerung des Relationalen Modells: Zusammenfassen von Relationen

13.15 Verfeinerung des Relationalen Modells II: schwache Entitätstypen, Generalisierung

13.16 Relationen versus Tabellen

# 6 Relationale Algebra

Wie kann ich Anfragen an das relationale Modell formulieren? Wie kann ich Daten filtern und auswählen? Wie funktionieren die typischen Mengenoperationen? Wie verbinde (joine) ich Daten aus verschiedenen Relationen? Was ist ein äusserer Verbund (outer join)? Wie werden Daten gruppiert und aggregiert? Und warum wird dabei auch noch projiziert?

13.17 Relationale Algebra: Selektion, Projektion, Vereinigung, Differenz, Kreuzprodukt, Umbenennung

13.18a Relationale Algebra: Differenz, Theta Join, Equi Join, Natural Join

13.18b Relationale Algebra: Semi Joins, Anti Semi Joins

13.18c Relationale Algebra: Äussere Joins

13.18d Relationale Algebra: Gruppierung und Aggregation

# 7 PostgreSQL und andere DBMS

Welche Datenbanksysteme (DBMS) gibt es eigentlich? Welches DBMS ist gut für was? Wie unterscheiden sich DBMS von Analytischen DBMS? Welche Benchmarks geben Hinweise auf die wirklich Leistung eines DBMS? Wie installiere ich PostgreSQL? Wie lade ich das Beispielschema und die Beispieldaten der Fotoagentur in PostgreSQL? Wie stelle ich Anfragen im PostgreSQL-Interface?

13.20 Übersicht über Datenbanksysteme: Welches DBMS für was?

13.22 Postgres Installieren

13.23 Postgres Übersicht, Schema und Daten laden

# 8 SQL

## 8.1 Grundlagen

Welche verschiedenene SQL-Standards gibt es? Was für SQL-Teilsprachen muss ich kennen? Was sind die Grundelemente einer SQL-Anfrage? Wie drücke ich Anfragen aus Relationaler Algebra in SQL aus? Was ist die *konzeptuelle* Ausführungsreihenfolge einer SQL-Anfrage? Wie kann ich das Ergebnis einer Anfrage sortieren? Wie kann ich die Anzahl der Ergebnisse einer Anfrage begrenzen und wann macht das Sinn?

13.21a SQL Standards

13.21b SQL Teilsprachen

13.24 SELECT FROM WHERE

13.25 ORDER BY

13.26 OFFSET, FETCH (oder LIMIT)

13.30 LIKE

## 8.2 Binäre Operationen

Wie setze ich die wichtigsten binären Operatoren der relationalen Algebra in SQL um? Wofür sind die ALL-Varianten gut?

13.27 UNION, UNION ALL

13.28 EXCEPT, EXCEPT ALL

13.29 INTERSECT, INTERSECT ALL

## 8.3 Joins

Wie setze ich die Verbundoperatoren (Joins) der relationalen Algebra in SQL um?

13.31 JOIN mit WHERE, ON, USING, NATURAL

13.32 OUTER JOIN

## 8.4 Gruppierung und Aggregation

Wie setze ich Gruppierung und Aggregation der relationalen Algebra in SQL um? Was ist bei der Auswahl der Gruppierungsattribute zu beachten? Was ermöglicht HAVING? Und wie ist die *konzeptuelle* Ausführungsreihenfolge einer SQL-Anfrage mit HAVING?

13.38 GROUP BY

13.39 HAVING

## 8.5 Sichten

Was ist eine Sicht? Wie definiere ich Sichten? Welche unterschiedlichen Arten von Sichten gibt es? Was sind lokale Sichten? Wie parametrisiere ich eine Sicht? Wann wird die Sicht berechnet? Was kann eine Sicht und was kann sie nicht? Wie unterscheiden sich lokale Sichten von globalen Sichten? Wie unterscheiden sich Tabellenfunktionen von anderen Funktionen?

13.40 CREATE VIEW Grundlagen

13.41 CREATE VIEW Beispiel mit Aggregation

13.42 Lokale Sichten mit WITH

13.43 Tabellenfunktionen

## 8.6 NULL-Semantik, PARTITION BY-Aggregation, Unteranfragen

Was ist beim Filtern und Gruppieren zu beachten, wenn es NULL-Werte in einer der Eingabetabellen gibt? Gibt es noch andere Möglichkeiten zu gruppieren und zu aggregieren als GROUP BY? Wie ist die *konzeptuelle*s Ausführungsreihenfolge einer SQL-Anfrage mit PARTITION BY? Wie funktionieren unkorrelierte und korrelierte Unteranfragen? Wie kann ich solche Unteranfragen lesen? Wie kann ich sie vereinfachen?

13.44 NULL != NULL

13.45 OVER, PARTITION BY

13.46 Unteranfragen mit IN, EXISTS, ALL, ANY

## 8.7 Tabellendefinition und Integritätsbedingungen

Wie lege ich eine Tabelle in SQL an? Wie ändere ich ihre Definition? Was ist der Datenbankkatalog? Was ist die psql-Shell? Was für Datentypen stehen hierfür zur Verfügung? Wie definiere ich Primärschlüssel, Schlüssel, Fremdschlüssel und einfache Integritätsbedingungen? Wie erzwinge ich Definiertheit von Werten oder lege mir selbst einen Datentyp an? Wie füge ich Tupel ein, ändere oder lösche sie? Was passiert mit

abhängigen Tupeln, die Tupel referenzieren, die gelöscht oder geändert werden? Wie kann ich beeinflussen, was mit diesen Tupeln passiert?

13.47 CREATE, ALTER, DROP, DESCRIBE TABLE, Katalog, psql Shell (Datendefinitionssprache, DDL)

13.48 SQL Datentypen

13.49 PRIMARY KEY, UNIQUE, FOREIGN KEY, REFERENCES, CONSTRAINT

13.50 NOT NULL, DEFAULT, CHECK, CREATE DOMAIN

13.51 INSERT, UPDATE, DELETE (Datenmanipulationssprache, DML)

13.52 ON DELETE NO ACTION, SET NULL, CASCADE, SET DEFAULT

### 8.8 Trigger

Wie kann ich den Zustand der Datenbank automatisiert überwachen? Was ist ein Trigger? Wann wird er ausgelöst und für was? Welche Typen von Triggern gibt es? Wie kann ich Trigger einsetzen, um die Konsistenz meiner Datenbank sicherzustellen? Was ist Atomicity und Consistency? Was ist eine Transaktion? Und was bedeutet dies für Trigger? Was genau wird für ein INSERT, DELETE oder UPDATE ausgeführt, wenn es Trigger für dieses INSERT, DELETE oder UPDATE gibt? Und was nicht? Wie kann ich die Überprüfung der Konsistenz an das Ende einer Transaktion verschieben?

13.53 Trigger Grundlagen

13.54 Trigger Zeitpunkt, ROW, STATEMENT, Anwendungsbeispiele, Vor- und Nachteile

13.55 Trigger für Konsistenzbedingungen, Transaktionen, Atomicity, Consistency, DEFERRABLE

### 8.9 Regeln

Wie kann ich in PostgreSQL Sichten zum Schreiben freischalten? Was bedeutet dies für die Implementierung Logischer Datenunabhängigkeit? Wie unterschieden sich Regeln von Triggern?

13.56 CREATE RULE, Änderbarkeit von Sichten

### 8.10 JDBC

Wie nutze ich ein DBMS aus einem Programm heraus? Welche Gefahren gibt es dabei? Was ist eine SQL-Injection-Attacke? Und wie vermeide ich sie durch Prepared Statements? Was ist JDBC?

13.57 JDBC, SQL Injection, Prepared Statements

## 9 Schemadesign, Normalformen

Was ist schlechtes Schemadesign? Warum ist das überhaupt ein Problem? Warum kann das in einem Software-Projekt (sehr) teuer werden? Was ist Datenredundanz? Was sind NULL-Werte? Und wieso sind zuviele davon garnicht gut? Was sind unechte Tupel?

13.58 Schlechtes, falsches, missverständliches Schemadesign, Redundanz, NULL-Werte, Unechte Tupel

Was sind funktionale Abhängigkeiten? Wie erkenne ich sie? Und wie erkenne ich sie nicht? Was ist der Unterschied zwischen funktionalen Abhängigkeiten einer Ausprägung versus einer beliebigen Ausprägung (und damit des Relationenschemas)?

13.59 Funktionale Abhängigkeiten, Functional Dependencies (FD)

Wie kann man mit Hilfe von Funktionalen Abhängigkeiten Schlüssel erkennen und definieren? Wie unterscheiden sich Superschlüssel von Kandidatenschlüsseln und Primärschlüsseln? Was sind Prim- und was sind Nicht-Primattribute? Wann ist eine Funktionale Abhängigkeit trivial? Was sind die Erste Normalform (1NF) und die Zweite Normalform (2NF) und warum sind diese nicht wirklich wichtig?

13.60 Superschlüssel, Kandidatenschlüssel, Primattribute, Triviale FD, Erste und Zweite Normalform

Was sind die Kriterien der Dritten Normalform (3NF)? Wann ist ein Relationenschema in 3NF?

13.61 Dritte Normalform 3NF

Was sind die Kriterien der Boyce-Codd Normalform (BCNF)? Wann ist ein Relationenschema in BCNF? Wann kann eine Tabelle überhaupt in 3NF sein und trotzdem BCNF verletzten?

13.62 Boyce-Codd Normalform BCNF

Was sind Inferenzregeln? Wie wende ich sie an? Was ist die Hülle $F^+$ einer Menge F Funktionaler Abhängigkeiten?

13.63 Inferenzregeln, Hülle

Welche Gütemaße sollte ich beachten, wenn ich ein Relationenschema zerlege? Wann ist eine Zerlegung gültig, verbundtreu (verlustlos), und/oder abhängigkeitsbewahrend? Welche Gütemasse muss bzw. kann ich mit Zerlegungen immer erreichen?

13.64 Zerlegung, gültig, verbundtreu, verlustlos, abhängigkeitsbewahrend

# 10 NoSQL und MapReduce

Was ist mit NoSQL wortwörtlich gemeint? Welche Klassen von NoSQL-Systemen gibt es? Welche Systeme machen wann überhaupt Sinn? Und wieso sollte ich in vielen Fällen dann lieber doch ein SQL-DBMS benutzen?

13.65 Was ist eigentlich NoSQL?

Wie ist die Semantik der map()- und der reduce()-Funktion? Was ist die Semantik der Verarbeitung eines MapReduce-Jobs? Wie setzte ich eine SQL-Anfrage nach map() und reduce() um? Und warum sollte ich das wann nicht machen? Was ist Hadoop? Was ist HDFS? Was sind die wesentlichen Gemeinsamkeiten und Unterschiede der Architektur eines MapReduce-Systems zu einem DBMS? Wie wird ein MapReduce-Job prinzipiell verarbeitet? Und wie funktioniert dies auf einem grossen Cluster? Was ist Redundanz in HDFS? Was bedeutet dies für Failover und Lastbalancierung? Was ist eine Intervall-Partitionierung? Was passiert in der Map-Phase, Shuffle-Phase, Reduce-Phase? Was ist ein Map Task? Was ist ein Reduce Task?

13.66 MapReduce, Semantik von map() und reduce()

13.67a MapReduce, Einordnung der Architektur

13.67b MapReduce, Prinzipielle Verarbeitung

13.68 MapReduce, Verarbeitung eines MapReduce Jobs auf einem Cluster

# 11 B-Bäume

Was ist die Grundidee eines Index? Wie unterscheiden sich Indexe von Karteikästen? Was ist eine Seite? Wie unterscheiden sich B-Bäume von binären Suchbäumen? Sind B-Baum und $B^+$-Baum eigentlich dasselbe? Wie suche ich Daten in einem B-Baum? Wieso werden Bereichsanfragen auf Schlüsseln gut unterstützt? Was ist ISAM? Wie füge ich effizient ein? Wie genau funktioniert der split() prinzipiell? Und wie genau implementiere ich das? Was ist der Unterschied zwischen einem Clustered und einem Unclustered B-Baum? Und wann macht welcher von beiden Sinn? Welche und wieviele dieser Indexe kann ich für eine Tabelle erstellen? (Und wieso hat PostgreSQL keinen Clustered Index ;-)?)

13.69a B-Bäume: Grundidee, Intuition

13.69b B-Bäume, Definition, insert, find_key, find_range, ISAM, split, delete, merge

13.69c B-Bäume, Unclustered vs. Clustered