

DBMS
Klausurvorbereitung

Inhalte

! Punkteaufteilung 100P

- 10% Grundlagen (Definitionen, 3-Schicht, ...)
- 31% SQL-Abfragen, einfach bis komplex
 - DB etwas komplexer
 - Beziehungen verstehen
 - MySQL-Diagramm
- 14% Relationenalgebra und Normalformen
- 14% Multiple Choice und Querschnittsthemen (aus allen Bereichen)
- 23% ER-Modellierung (Chen min:max)
 - einfache Notation

! Einleitung

- Einführung
 - geschichtliche Entwicklung
- Definition
 - Datenbanksystem und Bestandteile
 - Vorteile und Probleme
 - Eigenschaften (Persistenz, ...)
 - Datenmodelle
- DB-Architektur
 - ANSI-SPARC
 - Anwendungsarchitektur (3-Tier)
- Transaktionen
- Relationen und Tabellen als Grundelement

Modelle und Modellierung

- Entwicklung einer DB
 - Schritte
 - Vorgehensmodell
 - Bezug SWE
 - Was gibt es sonst für Ansätze? Agil, DevOps ...
- ER-Modell
 - Abstraktionskonzepte
 - Sprachbestandteile des ERM
 - Entitytyp
 - Beziehungstyp
 - identifizierend und nicht-identif.
 - ! weak entities
 - Grad einer Beziehung
 - Attribute
 - Kardinalitäten
 - 1:1, 1:n, M:n, vom Grad >2
 - Indizes
 - Modellarten des ER-M
 - ! Chen
 - Feinheiten kennen und selbst modellieren
 - Krähenfuß
- Datenarten (strukturiert, semi-strukturiert, unstrukturiert)
- Data Dictionary / Metadatenhaltung
- Krähenfußnotation
 - Notationselemente (Keys, gestrichelte Linien / durchgezogene...)
 - Modalität und Kardinalität
- Fallstudie Bergflöhe

Relationenmodell

- formale Basis von Datenbanken
 - mathematisch
 - Mengenlehre (reine)
 - von Mengen zu Mengen mittels Abfragen
 - keine Duplikate in Mengen
- ! Definition einer "Relation"
 - Kreuzprodukt
- Bestandteile einer Relation
 - Name, Attributnamen, Domänen
 - Integritätsbedingungen
 - ! funktionale Abhängigkeiten
 - Keys (Primär- und Fremdschlüssel)
 - ! Relation versus Tabelle
- Anomalien
 - 3 Arten

ER zu Relationen

- ! Bestimmung der Normalformen
 - Normalformen = Qualitätskriterien
 - Definition Key, SA, NSA, funktionale Abhängigkeiten
- Referenzielle Integrität (DDL)
 - ! identifizierende und nicht-identifizierende Beziehungen
 - wie sieht das im Code aus?
 - Master / Slave
 - ! RESTRICT, DELETE, UPDATE, SET NULL
- ER-Modell in Relationen übersetzen
 - binäre Beziehungen
 - 1:1 / 1:n
 - nicht in eigene Relation
 - m:n
 - eigene Tabelle mit dem Namen des Bn
 - Grenzfälle: z.B. sehr viele NULLs in 1:n
 - Beziehungen vom Grad > 2
 - BT wird immer zu einer eigenen Tabelle
 - Informationsverlust

SQL

- 6 - 7a Relationenalgebra und SQL
 - ! Relationenalgebra = Mengenlehre, aber auch Operatoren auf Wertebene (Spalten und Zeilen)
 - klassische Mengenoperationen
 - Schnittmenge
 - Vereinigungsmenge
 - Mengendifferenz
 - Tupel
 - müssen bei Mengenoperationen miteinander kompatibel sein
 - Relationen müssen entsprechend vom Format kompatibel sein
 - Projektion, Selektion, Verbund / Join
- 7b SQL (Verbund und Subselects)
 - Subselects -- korrelierte und unkorrelierte, mengenwertige und skalare, ANY, ALL

- ! DML, VDL, DDL
- SQL
 - DML im Detail kennen, auch Sonderfälle (LEFT JOIN, GROUP, ...), auch INSERT, UPDATE, DELETE
 - DDL und VDL nur kennen und interpretieren können
- SQL (Subselects, Präzisierung)
 - Übungsaufgaben