

Einführung in Datenbanksysteme (Introduction to Database Systems)

60 hours course, 30 hours practise

H. Schweppe
FU Berlin, SS 2005
hs@inf.fu-berlin.de

1 Introduction

- 1.1 *Databases vs. files*
- 1.2 *Basic concepts and terminology*
- 1.3 *Brief history of databases*
- 1.4 *Architectures & systems*
- 1.5 *Technical Challenges*
- 1.6 *DB lifecycle*

2 Conceptual Database Design

- 2.1 *Requirement analysis*
- 2.2 *Modeling languages*
 - 2.1.1 Overview
 - 2.1.2 Requirement Analysis (case study)
 - 2.2.1 Basic Modeling Primitives
 - 2.2.2 Modeling Languages: UML and Entity-Relationship Model (ERM)
 - 2.2.3 Conceptual DB design: basics
 - 2.2.4 From Requirements to Models
- 2.3 *Integrity Constraints*
 - 2.3.1 Constraint types
 - 2.3.2 Cardinality constraints
 - 2.3.3 Weak entities
- 2.4 *Modeling patterns*
 - 2.4.1 Modeling historical data
 - 2.4.2 N-ary relationships
 - 2.4.3 Generalization / specialization ...and more

3 Schema Design: Logical Design using the Relational Data Model

- 3.1 *Logical Schema Design*
 - 3.1.1 The Relational Data Model – Basics
 - 3.1.2 Keys, candidate keys and more
- 3.2 *From Conceptual to Logical Schema: Mapping ER to RDM*
 - 3.2.1 Relationships to relations: a simple step
 - 3.2.2 Mapping weak entities and multivalued attributes

- 3.2.3 Consolidation
- 3.2.4 Mapping generalization hierarchies and more

4 Logical Design : RDM - Schema Definition with SQL / DDL

4.1 *SQL history and standards*

4.2 *SQL/DDL – first steps*

- 4.2.1 Basis Schema Definition using SQL / DDL
- 4.2.2 SQL Data types, domains, user defined types
- 4.2.3 Creating simple tables

4.3 *SQL/DDL – Constraints*

- 4.3.1 Attribute and simple table constraints
- 4.3.2 Enforcing cardinality constraints and foreign keys
- 4.3.3 Deferred constraints
- 4.3.4 Assertions and triggers
- 4.3.5 Case study
- 4.3.6 Metadata management
- 4.3.7 Modifying and deleting definitions and more...

5 Normalization: - Quality of relational designs

5.1 *Functional Dependencies*

- 5.1.1 Design quality
- 5.1.2 Update anomalies
- 5.1.3 Functional Dependencies: definition
- 5.1.4.. Properties of Functional Dependencies

5.2 *Normal forms*

- 5.2.1 Informal introduction
- 5.2.2 Normal Forms and FDs
- 5.2.3 Normal forms (2NF, 3NF, BCNF, MV NF)
- 5.2.4 Lossless join and dependency preservation
- 5.2.5 Multivalued dependency

5.3 *Algorithms for finding Normal Forms*

- 5.2.1 Informal introduction
- 5.3.2 Minimal sets of Functional Dependencies
- 5.3.3 Synthesis and Decomposition

5.4 *Normal Forms: Critical review*

6 The Relational Data Model: Algebraic operations on tabular data

6.1 *Basic idea of relational languages*

6.2 *Relational Algebra operations*

- 6.3 *Relational Algebra: Syntax and Semantics*
- 6.4 *More Operators*
- 6.5 *Special Topics of RA*
 - 6.5.1 Relational algebra operators in SQL
 - 6.5.2 Relational completeness
 - 6.5.3 What is missing in RA?
 - 6.5.4 RA operator trees

7 The Relational Data Model: Logic foundation of data manipulation

- 7.1 *Logical foundations of the RDM*
- 7.2 *Relational Calculus Languages*
 - 7.2.1 Tuple calculus
 - 7.2.2 Brief overview of domain calculus
- 7.3 *Equivalence of relational languages*

8 SQL – Data Handling

- 8.1 *Update, Deletion, Insertion and bulk load**
- 8.2 *The query language SQL*
 - 8.2.1 Search predicates
 - 8.2.2 Arithmetic expressions and functions in predicates
 - 8.2.3 Different kinds of join
 - 8.2.4 Output improvement
- 8.3 *Advanced SQL*
 - 8.3.1 Subselects and Correlated subqueries
 - 8.3.2 Quantified expressions, SOME, ANY
 - 8.3.3 Grouping and Aggregation
 - 8.3.4 Transitive closure
 - 8.3.5 Final remarks: NULLS, temporary relations and more

9 Views in SQL: a tool for restricting column access and for simplifying application programming

- 9.1 *Views: not only access security*
 - 9.1.1 The general idea
 - 9.1.2 Query execution on views
 - 9.1.3 Generalization hierarchies and views
- 9.2 *Updatable views*
 - 9.2.1 Semantic characterization
 - 9.2.2 Some syntactic criteria
 - 9.2.3 WITH CHECK option
 - 9.2.4 Key preserved views

9.3 *View update using triggers*

10 **Extending the Relational Model: SQL 99**

10.1 *Motivation*

10.2 *Collection types*

10.3 *Types and objects*

10.4 *Functions*

10.5 *Triggers*

11 **Access Rights in SQL**

11.1 *The SQL security model*

11.2 *Granting and revoking privileges*

12 **Embedding SQL in Programming languages**

12.1 *Introduction: using SQL from programs*

12.2 *Embedded SQL*

12.2.1 Static and dynamic embedding

12.2.2 Cursors

12.2.3. ESQL / C

12.2.4 Positioned Update

12.3 *Transactions in application programs*

12.3.1 Definition

12.3.2 Isolation levels

12.4 *SQL and Java*

12.4.1 JDBC

12.4.2 SQLJ

12.5 *OR mapping and components*

13 **Physical schema design**

13.1 *Introduction*

13.2 *Technology*

13.2.1 Disk technology

13.2.2 RAID

13.3 *Index structures in DBS*

13.3.1 Indexing concept

13.3.2 Primary and Secondary indexes

13.3.3 Types of indexes and index definition in SQL

- 13.3.4 Implementing indexes: search trees
- 13.3.5 Criteria for indexing

13.4 *More index structures*

- 13.4.1 Clustered indexes
- 13.4.2 Implementation of rows and tables
- 13.4.3 B+ trees with data leafs
- 13.4.4 Bitmap indexes
- 13.4.5 Hash index and inversion
- 13.4.6 Case study ("Video store")

13.5 *Multi dimensional indexes (R-tree)*

14 Transactions: models

14.1 *Concepts: ACID properties*

14.2 *Modeling transactions: histories and schedules*

- 14.2.1 Correctness criteria
- 14.2.2 Serial execution
- 14.2.3 History

14.3 *Serializability*

- 14.3.1 Conflict graph
- 14.3.2 Serializability theorem

15 Concurrency control

15.1 *Serializability and Concurrency Control*

15.2 *Locking*

- 15.2.1 Lock protocols
- 15.2.2 Two phase locking
- 15.2.3 Strict transactional protocols
- 15.2.4 Lock conflicts and Deadlocks
- 15.2.5 Lock modes
- 15.2.6 Deadlock detection, resolution, avoidance

15.3 *Nonlocking concurrency control*

- 15.3.1 Optimistic cc: forward / backward oriented
- 15.3.2 Time stamp ordering
- 15.3.3 Multiversion cc

15.4a *Synchronizing index structures*

15.4b *Distributed transactions: Two Phase Commit (short)*

16 Logging and Recovery in Database systems

16.1 *Introduction: Fail safe systems*

- 16.1.1 Failure Types and failure model
- 16.1.2 DBS related failures

16.2 *DBS Logging and Recovery principles*

- 16.2.1 The Redo / Undo principle
- 16.2.2 Writing in the DB
- 16.2.3 Buffer management
- 16.2.4 Write ahead log
- 16.2.5 Log entry types
- 16.2.6 Checkpoints

16.3 *Recovery*

- 16.3.1 ReDo / UnDo
- 16.4.2 Recovery algorithm

17 Data Warehouses in a nutshell

17.1 *Introduction OLTP vs. OLAP*

17.2 *DWH methodology*

17.3 *Stars and Stripes*

17.4 *OLAP operators: Roll up and Drill down, SQL operators ROLLUP and CUBE*

17.5 *ROLAP and MOLAP ... and more*

For slides (pdf) see <http://www.inf.fu-berlin.de/lehre/SS05/19517-V/unterlagen.html>