Legacy Databases

JUMP INTO THE EVOLVING WORLD OF DATABASE MANAGEMENT

Principles of Database Management provides students with the comprehensive database management information to understand and apply the fundamental concepts of database design and modeling, database systems, data storage, and the evolving world of data warehousing, governance and more. Designed for those studying database management for information management or computer science, this illustrated textbook has a well-balanced theory-practice focus and covers the essential topics, from established database technologies up to recent trends like Big Data, NoSQL, and analytics. One going case studies, dril-forw hoxes that reveal deeper insights on key topics, retention questions at the end of every section of a chapter, and connections boxes that show the relationship between concepts throughout the text are included to provide the practical tools to get started in database management.

KEY FEATURES INCLUDE:

- Full-color illustrations throughout the text.
- Extensive coverage of important trending topics, including data warehousing, business intelligence, data integration, data quality, data governance, Big Data and analytics.
- An online playground with diverse environments, including MySQL for querying; MongoDB; Neo4j Cypher; and a tree structure visualization environment.
- Hundreds of examples to illustrate and clarify the concepts discussed that can be reproduced on the book's companion online playground.
- · Case studies, review questions, problems and exercises in every chapter.
- · Additional cases, problems and exercises in the appendix.

Online Resources www.cambridge.org/

Solutions manual Code and data for examples

Cover illustration: @Chen Hanquan / DigitalVision / Getty Images Cover design: Andrew Ward.



WILFRIED LEMAHIEU Seppe vanden broucke Bart Baesens

DATABASE MANAGEMENT THE PRACTICAL GUIDE TO STORING, MANAGING

AND ANALYZING BIG AND SMALL DATA

www.pdbmbook.com

CAMBRIDG

NAGEM

Introduction

- Hierarchical Model
- CODASYL Model

- The hierarchical model originated during the Apollo program conducted by NASA
 - IBM developed the Information Management System or IMS DBMS (1966-1968)
- No formal description available and lots of structural limitations (legacy)
- Two key building blocks: record types and relationship types

• A record type is a set of records describing similar entities and has 0, 1 or more records

Examples: product record type, supplier record type

- A record type consists of fields or data items
 - Examples: product number, product name, product color

- A relationship type connects two record types
- Only hierarchical structures are allowed (1:N relationship types)
- A record type can be a parent in multiple parent/child relationship types, but it can participate in at most one relationship type as a child
- Relationship types can be nested
- Root record type sits as the top of the hierarchy, whereas leaf record type sits at the bottom





- All data needs to be retrieved by navigating down from the root node (procedural DML)
- The hierarchical model is also very rigid and thus limited in terms of expressive power
- No support for N:M or 1:1 relationship types
- N:M relationship type
 - assign one record type as the parent and the other as the child record type
 - put relationship type attributes in child record type
 - however: redundancy is introduced!



- Another option for an N:M relationship type is to create two hierarchical structures and connect them using a virtual child record type and a virtual parent/child relationship type
 - pointers can then be used to navigate between both structures
 - relationship type attributes can be put in the virtual child record type
 - no more redundancy



- 1:1 relationship types should be implemented in application programs
- The hierarchical model only allows relationship types of degree 2
 - Recursive relationship types or relationship types with more than 2 record types need to be implemented using virtual child record types
- A child can not be disconnected from its parent (on delete cascade)



- Model limitations
 - no guarantee that each department has exactly 1 manager
 - no guarantee that a department has at least 1 employee

- The CODASYL model was developed by the Data Base Task Group of the COnference on DAta SYstem Languages in 1969
- CA-IDMS (Computer Associates)
- Building blocks
 - record types
 - set types
- Lots of structural limitations (legacy)

- A record type is a set of records describing similar entities and has 0, 1 or more records or record occurrences
- A record type consists of various data items
- A vector is a multivalued attribute type
 - Example: e-mail address
- A repeated group is a composite data item for which a record can have multiple values or a composite multi-valued attribute type
 - Example: address

- A set type models a 1:N relationship type between an owner record type and a member record type
- A set occurrence has 1 owner record and 0, 1 or more member records
- A CODASYL set has both owner and member records and it is also possible to order the member records (↔ mathematical set)



- A member record can exist without being connected to an owner record
- A record type can be a member record type in multiple set types (network structures)
- Multiple set types may be defined between the same record types

Bachmann diagram



- 1:1 relationship types must be enforced in the application program
- N:M relationship types
 - introduce a dummy record type as a member record type in 2 set types having as owners the record types of the original N:M relationship type





- CODASYL allows to logically order the member records of a set (e.g., alphabetically)
- System can act as the owner for the root record type (singular or system owned set type)
- No support for recursive set types
 dummy record type needs to be introduced
- No set types with more than 2 participating record types





CODASYL

- Model limitations
 - an employee can be managed by multiple employees
 - no guarantee that a department must have exactly 1 manager
 - no guarantee that a department has minimal one employee

Conclusion

- Hierarchical Model
- CODASYL Model

More information?

JUMP INTO THE EVOLVING WORLD OF DATABASE MANAGEMENT

Principles of Database Management provides students with the comprehensive database management information to understand and apply the fundamental concepts of database design and modeling, database systems, data storage, and the evolving world of data warehousing, governance and more. Designed for those studying database management for information management or computer science, this illustrated textbook has a well-balanced theory-practice focus and covers the essential topics, from established database technologies up to recent trends like Big Data, NoSQL, and analytics. One-going case studies, drill down boxes that reveal deeper insights on key topics, retention questions at the end of every section of a chapter, and connections boxes that show the relationship between concepts throughout the text are included to provide the practical tools to get started in database

KEY FEATURES INCLUDE:

- Full-color illustrations throughout the text.
- Extensive coverage of important trending topics, including data warehousing, business intelligence, data integration, data quality, data governance, Big Data and analytics.
- An online playground with diverse environments, including MySQL for querying; MongoDB; Neo4j Cypher; and a tree structure visualization environment.
- Hundreds of examples to illustrate and clarify the concepts discussed that can be reproduced on the book's companion online playground.
- · Case studies, review questions, problems and exercises in every chapter.
- · Additional cases, problems and exercises in the appendix.



Code and data for examples

Cover illustration: @Chen Hanquan / DigitalVision / Getty Images. Cover design: Andrew Ward.



WILFRIED LEMAHIEU Seppe vanden broucke Bart baesens

DATABASE DATABASE MANAGEMENT THE PRACTICAL GUIDE TO STORING. MANAGING

AND ANALYZING BIG AND SMALL DATA

www.pdbmbook.com

CAMBRIDGE UNIVERSITY PRESS