

Texts in Computer Science

Series Editors:

David Gries

Fred B. Schneider

For other titles published in this series, go to
<http://www.springer.com/series/3191>

Peter Revesz

Introduction to Databases

From Biological to Spatio-Temporal

 Springer

Prof. Peter Revesz
University of Nebraska-Lincoln
Dept. Computer Science & Engineering
358 Avery Hall
Lincoln NE 68588
USA
revesz@cse.unl.edu

Series Editors

David Gries
Department of Computer Science
Upson Hall
Cornell University
Ithaca, NY 14853-7501
USA

Fried. B. Schneider
Department of Computer Science
Upson Hall
Cornell University
Ithaca, NY 14853-7501
USA

ISSN 1868-0941 e-ISSN 1868-095X
ISBN 978-1-84996-094-6 e-ISBN 978-1-84996-095-3
DOI 10.1007/978-1-84996-095-3
Springer London Heidelberg Dordrecht New York

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2009942531

© Springer-Verlag London Limited 2010

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licenses issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

The use of registered names, trademarks, etc., in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Cover design: SPi Publisher Services

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

In memory of
Paris C. Kanellakis
and
Alberto O. Mendelzon

I believe that there are a collection of planet-threatening problems, such as climate change and ozone depletion, that only scientists are in a position to solve. Hence, the sorry state of DBMS support...for this class of users is very troubling.

—Michael Stonebraker (2009), *CACM*, 52(5):11.

Preface

Introduced forty years ago, relational databases proved unusually successful and durable. However, relational database systems were not designed for modern applications and computers. As a result, specialized database systems now proliferate trying to capture various pieces of the database market. Database research is pulled into different directions, and specialized database conferences are created. Yet the current chaos in databases is likely only temporary because every technology, including databases, becomes standardized over time.

The history of databases shows periods of chaos followed by periods of dominant technologies. For example, in the early days of computing, users stored their data in text files in any format and organization they wanted. These early days were followed by information retrieval systems, which required some structure for text documents, such as a title, authors, and a publisher. The information retrieval systems were followed by database systems, which added even more structure to the data and made querying easier.

In the late 1990s, the emergence of the Internet brought a period of relative chaos and interest in unstructured and “semistructured data” as it was envisioned that every webpage would be like a page in a book. However, with the growing maturity of the Internet, the interest in structured data was regained because the most popular websites are, in fact, based on databases. The question is not whether future data stores need structure but what structure they need.

Today we see another period of relative chaos as specialized geographic, moving objects, biological, and other types of databases are unable to communicate with each other in spite of the need of many advanced applications. Database interoperability and data integration become important challenges in the current environment.

What are possible solutions to the above challenges? There are two observations that may be made. First, instead of expecting a radical movement away from relational data, future databases will probably provide various extensions of relational databases. Second, constraint databases, which are an extension of relational databases, appear to be suitable as a common standard for the various types of databases. In fact, geographic information systems sometimes convert internally their “vector data” into a constraint data to evaluate certain queries.

Purpose and Goals

This textbook provides comprehensive coverage of databases. The primary audience of the book is undergraduate computer science and non-computer science students with no or little prior exposure to databases. For them the extensive set of exercises at the end of each chapter will be useful. The text and the exercises assume as prerequisite only basic discrete mathematics, linear algebra, and programming knowledge. Many database experts will also find the bibliographic notes after each chapter a valuable reference for further reading. For both students and database experts the MLPQ constraint-relational database system is suggested for use. The MLPQ system, slides, solutions (for instructors), and other course aid is available free from the author’s web page: <http://cse.unl.edu/~revesz>

Topics Coverage and Organization

This book has three parts:

- **Part I: Data and Queries:** The first part of the book defines databases in general (Chapter 1), describes eleven different types of databases (Chapters 2-12), and presents the MLPQ and the DISCO database systems (Chapters 13-14) that implement several different types of databases. The first part of the book is enough to start using a database system for simple applications.
- **Part II: Database Design and Applications:** The second part of the book describes database design (Chapter 15) and discusses the following advanced database application issues: database interoperability, which allows translating data from one database to another database (Chapter 16), data integration, which allows combining data from different sources (Chapter 17), interpolation and approximation, which allows a simple representation of complex data (Chapter 18), and prediction and data mining, which allows the discovery of new information (Chapter 19). The second part of the book enables the readers to develop more complex and rewarding database applications.

- **Part III: Query Evaluation, Safety, and Verification:** The third part of the book describes various issues related to the evaluation of database queries. The topics include indexing methods, which enable fast search in database tables (Chapter 20), data visualization, which is essential for a convenient interaction with the database system (Chapter 21), the safety of queries, that is, whether they are guaranteed to be evaluated precisely or approximately, or may enter an infinite loop and not terminate (Chapter 22), general evaluation algorithms (Chapter 23), the efficient implementation of the evaluation algorithms (Chapter 24), the complexity of the evaluation of different types of queries (Chapter 25). Computer software written in many programming languages can be translated into database queries. Software verification can be aided by translating the software to database queries, which are evaluated approximately by the above query evaluation algorithms (Chapter 26). The third part of the book gives students a deeper understanding and appreciation of the inside of database systems. It also enables students to read further scientific literature and begin research projects in database systems.

Figure 1 shows a dependency diagram of the chapters of the book. A special strength of the book is that it allows a flexible course design aided by the dependency diagram. We suggest the following courses.

- **Basic introductory course:** A basic introductory course, which is focused on a survey of various types of databases, would cover Chapters 1, 2, 3, and 4, any subset of Chapters 5-12, and the MLPQ system in Chapter 13. Such an introductory course would be suitable even for non-computer science majors. The topics could be easily fitted to the interest of the students. For example, geography majors may be interested in Chapters 5-8, while biology majors in Chapters 10-11. All of these students can be given a hands-on experience with a database system using the exercises in Chapter 13.
- **Enhanced introductory course:** An enhanced introductory course can add to the basic introductory course various applications-oriented chapters from the second part of the book (Chapters 15-19). These chapters enable more serious applications. For example, Chapter 18 discusses the triangulated irregular network (TIN) representation of surface data, which is important for geographic applications and enhances the material in Chapter 6. While Chapter 6 enables students to understand and use given geographic databases, Chapter 18 enables students to build geographic databases from measurement data. Many students may already have such measurement data from other projects outside of the database class.

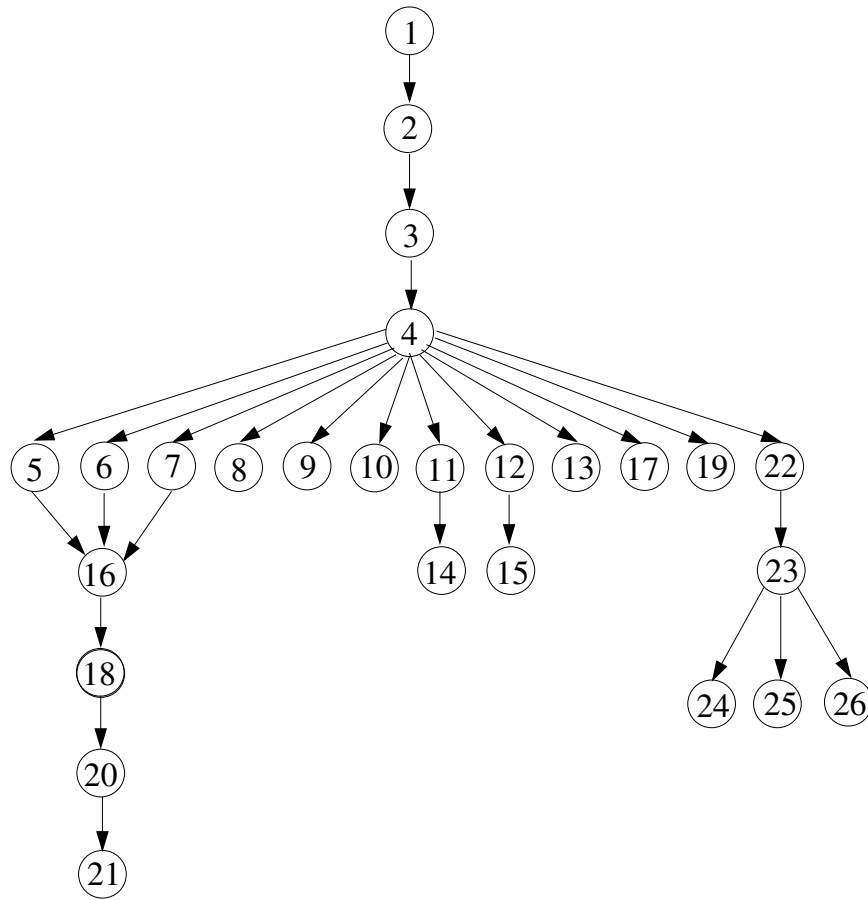


Figure 1: A dependency diagram of the chapters.

- **Advanced course:** The textbook also allows several types of advanced courses. An advanced course, focused on query evaluation, may cover interoperability (Chapter 16), indexing (Chapter 20), safety (Chapter 22), general evaluation algorithms (Chapter 23), and implementation methods (Chapter 24). These chapters already allow students to implement either individually or in group projects various simple query processors or prototype database systems. The coverage of the rest of the chapters depends on the interest of the students. Computer graphics students may study Chapter 21, theoretical computer science students may study Chapter 25, and software engineering students may study Chapter 26.

Acknowledgments

I'd like to thank for reviewing various parts of the book Berthe Choueiry, Manolis Koubarakis, Franz Mora, French Roy, and James Van Etten. I'd like to thank for introducing me to the database area the late Paris Kanellakis, my Ph.D. thesis advisor at Brown University, and the late Alberto Mendelzon, my postdoctoral advisor at the University of Toronto. I thank the following coauthors on books, journal articles or conference papers that contributed to the material in this book and from whom I learned much: András Benczur, Jan Chomicki, Floris Geerts, Sofie Haesevoets, Mark Griep, Gosta Grahne, Bart Kuijpers, Gabriel Kuper, Ram Narayanan, Agnes Novak, Robert Powers, James Pustejovsky, Raghu Ramakrishnan, Divesh Srivastava, Paolo Terenziani, and Jan Van den Bussche.

I would like to thank my former Ph.D. students for their comments on the text and other contributions to our database research group: Scot Anderson (Southern Adventist University), Mengchu Cai (IBM, DB2 Group), Rui Chen, Yi Chen, Ying Deng, Jun Gao (University of the Ozarks), Lixin Li (Georgia Southern University), Min Ouyang, Thomas Triplet (Concordia University), and Shasha Wu (Spring Arbor University). In addition, the following M.S. students also contributed significantly to the database research group: Brian Boon, Jo-Hag Byon, Vijay Eadala, Pradeep Gundavarapu, Pradip Kanjamala, Yiming Li, Yuguo Liu, Prashanth Nandavanam, Andrew Salamon, Yonghui Wang, and Lei Zhang.

The research, which forms several chapters of this book, was supported in part by the U.S. National Science Foundation under grants IRI-9625055, IRI-9632871, and EIA-0091530, by a NASA grant, by a Gallup Research Professorship, by an Alexander von Humboldt Research Fellowship, and by a J. William Fulbright Scholarship. The latter two awards enabled research leaves from the University of Nebraska-Lincoln to the Max Planck Institut für Informatik and the University of Athens, respectively. Andreas Podelski at the former and Manolis Koubarakis at the latter place served as great hosts during my visits.

I am grateful to the editors at Springer-Verlag, especially Wayne Wheeler, Senior Editor for Computer Science, and Simon Rees, Senior Editorial Assistant, for their careful editing and attention to several details in the production of the book.

Finally, I would like to thank Lilla, my wife, and our children, Sophie, Claire, and Gregory, for their enthusiasm and patience during the writing of this book.

Peter Revesz
 Lincoln, Nebraska, USA
 July 2009

Contents

Preface	vii
1 Data Models, Queries, Evaluation	1
1.1 Data Models	1
1.2 Queries	4
1.3 Evaluation	5
2 Propositional Databases	7
2.1 Propositional Data Model	7
2.2 Implication Queries	10
2.3 Evaluation	10
2.4 Limitations	12
3 Relational Databases	15
3.1 Relational Data Model	15
3.2 Relational Algebra Queries	18
3.3 SQL Queries	24
3.4 Evaluation	34
4 Constraint Databases	43
4.1 Infinite Relational Data Model	43
4.2 Relational Algebra Queries	46
4.3 SQL Queries	51
4.4 Evaluation	55

5	Temporal Databases	67
5.1	Allen's Relations	67
5.2	Temporal Data Models	70
5.3	Indefinite Temporal Data Model	71
5.4	Temporal Database Queries	73
6	Geographic Databases	81
6.1	Rectangles Data Model	82
6.2	Vector Data Model	84
6.3	Worboys' Data Model	86
6.4	Constraint Data Model	87
6.5	Topological Data Models	89
6.6	Geographic Queries	91
7	Moving Objects Databases	111
7.1	Moving Points Data Model	112
7.2	Parametric Circles Data Model	112
7.3	Parametric Rectangles Data Model	113
7.4	Parametric Worboys Data Model	115
7.5	Periodic Parametric Data Models	115
7.6	Geometric Transformation Data Models	120
7.7	Moving Objects Queries	122
8	Image Databases	137
8.1	Affine Invariance	138
8.2	Affine-Invariant Similarity Measures	140
8.3	The Color Ratios Similarity Measure	141
8.4	Similarity of Patterned Objects	143
9	Constraint Objects Databases	153
9.1	The Constraint Objects Data Model	153
9.2	Closed, Open, and Possible Worlds	154
9.3	Syntax	156
9.4	Semantics	161
9.5	Projection Queries	167
9.6	Evaluation of Refinement Queries	168
9.7	Constraint XML	168
10	Genome Databases	179
10.1	Biological Basics	179
10.2	The Genome Map Assembly Problem	182
10.3	Sequence Similarity	190
10.4	Structure Similarity	198

11 Set Databases	205
11.1 Syllogisms	205
11.2 Boolean Algebras of Sets	219
11.3 Relations with Sets	230
12 Constraint Deductive Databases	235
12.1 Syntax	235
12.2 Datalog with Sets	240
12.3 Semantics	246
12.4 Datalog with Aggregation and Negation	254
13 The MLPQ System	261
13.1 The MLPQ System Architecture	262
13.2 Relational Databases	264
13.3 Geographic Databases	275
13.4 Moving Objects Databases	286
13.5 Topological Databases	302
13.6 Recursive Queries	307
13.7 Linear Programming	315
13.8 Web Accessible Applications	316
14 The DISCO System	335
14.1 DISCO Queries	335
14.2 Implementation	339
14.3 Using the DISCO System	346
14.4 Extensibility of the DISCO System	348
15 Database Design	351
15.1 Entity Relationship Diagrams	351
15.2 Constraint Automata	361
16 Interoperability	385
16.1 Data Interoperability	385
16.2 Query Interoperability	402
16.3 Other Types of Interoperability	410
17 Data Integration	417
17.1 Decision Trees	417
17.2 Support Vector Machines	421
17.3 Classification Integration	421
17.4 The Reclassification Problem	425
17.5 Wrappers	426
17.6 Data Fusion	430
17.7 Security	430

18 Interpolation and Approximation	435
18.1 Linear Interpolation	436
18.2 Triangulated Irregular Networks	436
18.3 Shape Functions	443
18.4 Spatiotemporal Interpolation	449
18.5 Inverse Distance Weighting	457
18.6 Piecewise Linear Approximation	460
19 Prediction and Data Mining	485
19.1 Prediction	485
19.2 Data Mining	492
19.3 Model-Based Diagnosis	495
20 Indexing	501
20.1 Minimum Bounding Parametric Rectangles	502
20.2 The Parametric R-Tree Index Structure	508
20.3 Indexing Constraint Databases	515
20.4 The MaxCount Operator	516
21 Data Visualization	537
21.1 Isometric Color Bands	538
21.2 Value-by-Area Cartogram	542
21.3 Animation of Moving Objects	548
22 Safe Query Languages	555
22.1 Safety Levels	555
22.2 Restriction	558
22.3 Safe Aggregation and Negation Queries	559
22.4 Safe Refinement and Projection Queries	560
22.5 Certification	560
23 Evaluation of Queries	571
23.1 Quantifier Elimination and Satisfiability	572
23.2 Evaluation of Relational Algebra Queries	590
23.3 Evaluation of SQL Queries	594
23.4 Evaluation of Datalog Queries	595
23.5 Constraint Logic Programming	614
24 Implementation Methods	621
24.1 Evaluation with Gap-Graphs	621
24.2 Evaluation with Matrices	628
24.3 Boolean Constraints	637
24.4 Optimization of Relational Algebra	641

25 Computational Complexity	655
25.1 Turing Machines	655
25.2 Complexity Classes	659
25.3 Complexity of Datalog	660
25.4 Complexity of Relational Algebra	667
26 Software Verification	685
26.1 Addition-Bound Matrices	685
26.2 Abstract Interpretation	689
26.3 Software Verification using MLPQ	695
Bibliography	701
Index	739
Colour Plate Section	745