Fundamentals of Physical Design and Query Compilation

Synthesis Lectures on Data Management

Editor
M. Tamer Özsu, ***University of Waterloo***

Synthesis Lectures on Data Management is edited by Tamer Özsu of the University of Waterloo. The series will publish 50- to 125 page publications on topics pertaining to data management. The scope will largely follow the purview of premier information and computer science conferences, such as ACM SIGMOD, VLDB, ICDE, PODS, ICDT, and ACM KDD. Potential topics include, but not are limited to: query languages, database system architectures, transaction management, data warehousing, XML and databases, data stream systems, wide scale data distribution, multimedia data

Fundamentals of Physical Design and Query Compilation
David Toman and Grant Weddell
2011

Methods for Mining and Summarizing Text Conversations
Giuseppe Carenini, Gabriel Murray, and Raymond Ng
2011

Probabilistic Databases
Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch
2011

Peer-to-Peer Data Management
Karl Aberer
2011

Probabilistic Ranking Techniques in Relational Databases
Ihab F. Ilyas and Mohamed A. Soliman
2011

Uncertain Schema Matching
Avigdor Gal
2011

Fundamentals of Object Databases: Object-Oriented and Object-Relational Design
Suzanne W. Dietrich and Susan D. Urban
2010

Advanced Metasearch Engine Technology
Weiyi Meng and Clement T. Yu
2010

Web Page Recommendation Models: Theory and Algorithms
Sule Gündüz-Ögüdücü
2010

Multidimensional Databases and Data Warehousing
Christian S. Jensen, Torben Bach Pedersen, and Christian Thomsen
2010

Database Replication
Bettina Kemme, Ricardo Jimenez Peris, and Marta Patino-Martinez
2010

Relational and XML Data Exchange
Marcelo Arenas, Pablo Barcelo, Leonid Libkin, and Filip Murlak
2010

User-Centered Data Management
Tiziana Catarci, Alan Dix, Stephen Kimani, and Giuseppe Santucci
2010

Data Stream Management
Lukasz Golab and M. Tamer Özsu
2010

Access Control in Data Management Systems
Elena Ferrari
2010

An Introduction to Duplicate Detection
Felix Naumann and Melanie Herschel
2010

Privacy-Preserving Data Publishing: An Overview
Raymond Chi-Wing Wong and Ada Wai-Chee Fu
2010

Keyword Search in Databases
Jeffrey Xu Yu and Lu Qin, Lijun Chang
2009

© Springer Nature Switzerland AG 2022

Reprint of original edition © Morgan & Claypool 2011

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopy, recording, or any other except for brief quotations in printed reviews, without the prior permission of the publisher.

Fundamentals of Physical Design and Query Compilation

David Toman and Grant Weddell

ISBN: 978-3-031-00753-8 paperback
ISBN: 978-3-031-01881-7 ebook

DOI 10.1007/978-3-031-01881-7

A Publication in the Springer series
SYNTHESIS LECTURES ON DATA MANAGEMENT

Lecture #18
Series Editor: M. Tamer Özsu, *University of Waterloo*
Series ISSN
Synthesis Lectures on Data Management
Print 2153-5418 Electronic 2153-5426

Fundamentals of Physical Design and Query Compilation

David Toman and Grant Weddell
University of Waterloo, Canada

SYNTHESIS LECTURES ON DATA MANAGEMENT #18

ABSTRACT

Query compilation is the problem of translating user requests formulated over purely conceptual and domain specific ways of understanding data, commonly called logical designs, to efficient executable programs called query plans. Such plans access various concrete data sources through their low-level often iterator-based interfaces. An appreciation of the concrete data sources, their interfaces and how such capabilities relate to logical design is commonly called a physical design. This book is an introduction to the fundamental methods underlying database technology that solves the problem of query compilation. The methods are presented in terms of first-order logic which serves as the vehicle for specifying physical design, expressing user requests and query plans, and understanding how query plans implement user requests.

KEYWORDS

query compilation, semantic query optimization, physical database design, database update, logical foundations and integrity constraints, bag semantics and query plans

Contents

Preface xi

Acknowledgments xiii

1 Introduction 1

1.1 Logical and Physical Design 1
1.2 Summary 5
1.3 Bibliographic Notes 5
1.4 Exercises with Topics for Discussion 6

2 Logical Design and User Queries 9

2.1 Logical Design in FOL 9
2.1.1 Logical Vocabularies 9
2.1.2 Logical Constraints 11
2.2 User Queries in FOL 15
2.3 Summary 17
2.4 Bibliographic Notes 19
2.5 Exercises with Topics for Discussion 19

3 Basic Physical Design and Query Plans 21

3.1 Access Paths and Simple Scanning 22
3.2 Query Plan Execution 26
3.3 Conjunctive Plans 29
3.4 General Query Plans 33
3.5 Summary 37
3.6 Bibliographic Notes 38
3.7 Exercises with Topics for Discussion 39

4 On Practical Physical Design 41

4.1 References, Pointers and Linked Structures 43
4.1.1 Efficient Search and Secondary Indices 46
4.2 Nulls, Partitions and Run-time Typing 47

4.2.1 Nulls Indicating Value Inapplicable 47
4.2.2 Horizontal Partitioning 49
4.2.3 Run-time Typing 50
4.3 Built-In Functions and Hashing 51
4.3.1 Hashing 52
4.4 Two-Level Store 54
4.4.1 Two-level References 55
4.4.2 ISAM Indexing 56
4.5 Summary 60
4.6 Bibliographic Notes 61
4.7 Exercises with Topics for discussion 61

5 Query Compilation and Plan Synthesis 63

5.1 Beth Definability 63
5.2 Conjunctive Queries and Dependencies 65
5.2.1 The Chase 65
5.2.2 Nested Loops and Right-deep Join Plans 66
5.2.3 Chase and Plan Synthesis 67
5.2.4 Equality-generating Dependencies 70
5.2.5 Post-processing: Assignments and Comparisons 70
5.2.6 Post-processing: Duplicate Elimination 71
5.2.7 Post-processing: Cut Insertion 72
5.2.8 Extensions to Chase: Beyond Conjunctive Queries 73
5.3 First-order Queries and Constraints 74
5.3.1 Interpolation 75
5.3.2 Interpolants Constructively 76
5.3.3 Interpolants vs. Plans 78
5.3.4 Interpolants and Duplicates 80
5.4 Interpolation vs. Chase 81
5.5 Summary 82
5.6 Bibliographic Notes 82
5.7 Exercises with Topics for Discussion 83

6 Updating Data 85

6.1 User vs. Physical Updates 85
6.1.1 Updates, Definability, and Plans 86
6.2 Updates and Complements 88

6.2.1 Complements vs. a Cyclic Schema 89
6.2.2 Update Types 91
6.3 Progressive Updates 91
6.3.1 Constant Space Complements 93
6.4 Summary 94
6.5 Bibliographic Notes 95
6.6 Exercises with Topics for Discussion 95

A First-Order Logic 97
A.1 Signatures 97
A.2 Syntax 97
A.3 Semantics 99
A.4 Proofs and the Sequent Calculus 101

Bibliography 105

Authors' Biographies 109

Preface

Information abounds, and the arrival of commercially viable computer systems with mass storage to help manage information has prompted the development of database systems together with an underlying technology. This book is about an important part of the technology for database systems that require only a logical appreciation of data on the part of application developers, such as database systems that support the relational model of data: the part that enables users to formulate queries that are devoid of any information relating to concrete data sources and their low-level interfaces.

A fundamental problem that must therefore be addressed is to translate user requests over purely conceptual and domain-specific ways of understanding of data, commonly called *logical designs*, to efficient executable programs, called *query plans*, for evaluating the requests by accessing various concrete data sources through their low-level often iterator-based interfaces. An appreciation of the concrete data sources, their interfaces and how such capabilities relate to logical design is in turn called a *physical design*. There are several phrases used to refer to the problem of translating user requests to plans; *plan generation* and *query optimization* are examples. In this book, we use *query compilation* to emphasize our focus on the issue of determining when plans correctly implement user requests.

The lingua franca throughout the book is *first-order logic* (FOL). Indeed, we shall see that FOL suffices to satisfy a number of needs connected with capturing and reasoning about physical design and query compilation.

1. It is a convenient language in which to formulate logical designs.
2. User queries correspond naturally to well-formed formulae in FOL. In fact, a subset of well-formed formulae that satisfy a condition called *domain independence* constitute a standard metric of query expressiveness called *relational completeness*.
3. With the added notion of *binding patterns*, it is also a convenient language to capture physical designs. This remains true for very sophisticated designs that entail multi-level store, pointers, hashing, capabilities for runtime typing, user defined functions, and so on.
4. Additional syntactic conditions on well-formed formulae in FOL lead to a plan language with a small set of primitive operations that suffice to formulate query plans over sophisticated physical designs that are often "built-in" in existing relational database technology.
5. Query compilation, the problem of synthesizing plans for user queries, corresponds naturally to reasoning in FOL. In the book, we look at how both chase-based reasoning and more general reasoning based on interpolation in FOL can be used to synthesize plans for user queries that express both requests for information and for data update.

The book is organized into six chapters. A more thorough introduction to basic terminology and notation for the problems of physical design and query compilation are the subject of Chapter 1. Items 1–4 above are the subject of the next three chapters. In particular, Chapters 2 and 3 review how FOL in combination with binding patterns can be used to formally capture both logical and physical design, to express user queries, and, with some straightforward extensions relating to syntax, to express query plans that implement user queries by appealing to a small set of primitive operations. Chapter 4 presents a number of cases that demonstrate how such a minimalist approach can be coupled with constraints expressed in FOL in ways that manifest sophisticated data structures and algorithms that are often implemented as additional native data encoding options and query operations in existing relational database technology. (Some examples are listed in Item 3 above.) The last item is the subject of the remaining two chapters. In particular, approaches to the problem of query compilation are considered in Chapter 5 and applications of these techniques to the problem of database update in Chapter 6.

A brief overview of FOL is given in Appendix A for readers who might require a review of basic material. It is a good idea to read through Appendix A to at least align expectations of FOL terminology and notation with what is presumed in this book. The book also presumes that readers are familiar with the subject of computer science to the extent that is acquired during the first two years of an undergraduate degree. This includes an understanding of algorithms and data structures, basic computational and complexity issues and imperative programming. An introductory course on mathematical logic and on databases is also desirable, in particular on declarative programming in SQL.

Following each chapter is a brief bibliography relevant to the topic and a list of exercises on topics that can be used as the basis for a class discussion. Many are sufficiently general that they can also be used as the basis for a term project.

David Toman and Grant Weddell
July 2011

Acknowledgments

The authors would like to thank the National Institute of Informatics (NII, Tokyo) for providing the opportunity to deliver a lecture series based on an earlier version of the material discussed in this book. We would also like to express our thanks to Tamer Özsu for his support and encouragement throughout.

David Toman and Grant Weddell
July 2011