

Beginning SQL Queries

From Novice to Professional



Clare Churcher

Beginning SQL Queries: From Novice to Professional

Copyright © 2008 by Clare Churcher

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-59059-943-3

ISBN-10 (pbk): 1-59059-943-8

ISBN-13 (electronic): 978-1-4302-0550-0

ISBN-10 (electronic): 1-4302-0550-4

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Jonathan Gennick

Technical Reviewer: Darl Kuhn

Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell,

Jonathan Gennick, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann,

Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Project Manager: Beth Christmas

Copy Editors: Marilyn Smith, Kim Wimpsett

Associate Production Director: Kari Brooks-Copony

Production Editor: Ellie Fountain

Compositor: Susan Glinert

Proofreaders: Linda Seifert, Liz Welch

Indexer: Broccoli Information Management

Artist: April Milne

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <http://www.apress.com>.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at <http://www.apress.com/info/bulksales>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

To Mark and Ali

Contents at a Glance

About the Author	xiii
About the Technical Reviewer	xv
Acknowledgments	xvii
Introduction	xix
■ CHAPTER 1 Relational Database Overview	1
■ CHAPTER 2 Simple Queries on One Table	17
■ CHAPTER 3 A First Look at Joins	41
■ CHAPTER 4 Nested Queries	61
■ CHAPTER 5 Self Joins	77
■ CHAPTER 6 More Than One Relationship Between Tables	95
■ CHAPTER 7 Set Operations	107
■ CHAPTER 8 Aggregate Operations	133
■ CHAPTER 9 Efficiency Considerations	153
■ CHAPTER 10 How to Approach a Query	169
■ CHAPTER 11 Common Problems	191
■ APPENDIX Sample Database	209
■ INDEX	211

Contents

About the Author	xiii
About the Technical Reviewer	xv
Acknowledgments	xvii
Introduction	xix
CHAPTER 1 Relational Database Overview	1
What Is a Relational Database?	1
Introducing Data Models	2
Introducing Tables	4
Inserting and Updating Rows in a Table	5
Designing Appropriate Tables	7
Maintaining Consistency Between Tables	9
Retrieving Information from a Database	10
Relational Algebra: Specifying the Operations	11
Relational Calculus: Specifying the Result	13
Why Do We Need Both Algebra and Calculus?	14
Summary	15
CHAPTER 2 Simple Queries on One Table	17
Retrieving a Subset of Rows	20
Relational Algebra for Retrieving Rows	20
Relational Calculus for Retrieving Rows	20
SQL for Retrieving Rows	21
Retrieving a Subset of Columns	22
Relational Algebra for Retrieving Columns	22
Relational Calculus for Retrieving Columns	22
SQL for Retrieving Columns	23
Using Aliases	23
Combining Subsets of Rows and Columns	24
Saving Queries	25
Specifying Conditions for Selecting Rows	25
Comparison Operators	26
Logical Operators	27

Dealing with Nulls	29
Comparing Null Values.....	31
Finding Nulls	32
Managing Duplicates	32
Ordering Output	35
Performing Simple Counts	35
Avoiding Common Mistakes	36
Misusing Select to Answer Questions with the Word “both”.....	38
Misusing Select Operations to Answer Questions with the Word “not”	39
Summary	39
 CHAPTER 3 A First Look at Joins	 41
Joins in Relational Algebra	41
Cartesian Product.....	41
Inner Join.....	43
SQL for Cartesian Product and Join	44
Joins in Relational Calculus	45
Extending Join Queries	46
An Algebra Approach	47
Order of Algebra Operations	50
A Calculus Approach	51
Expressing Joins Through Diagrammatic Interfaces	53
Other Types of Joins	54
Summary	58
 CHAPTER 4 Nested Queries	 61
IN Keyword	61
Using IN with a Nested Query	62
Being Careful with NOT and <>	64
EXISTS Keyword	67
Different Types of Nesting	69
Inner Queries Returning a Single Value	70
Inner Queries Returning a Set of Values	72
Inner Queries Checking for Existence.....	72
Using Nested Queries for Updating	73
Summary	75

CHAPTER 5	Self Joins	77
	Self Relationships	77
	Creating a Self Join	79
	Queries Involving a Self Join	80
	A Calculus Approach to Self Joins	85
	Questions Involving “Both”	88
	A Calculus Approach to Questions Involving “Both”	90
	An Algebra Approach to Questions Involving “Both”	91
	Summary	92
	Self Relationships	92
	Questions Involving the Word “Both”	93
CHAPTER 6	More Than One Relationship Between Tables	95
	Representing Multiple Relationships Between Tables	95
	Algebra Approach to Two Relationships Between Tables	97
	Calculus Approach to Two Relationships Between Tables	101
	Business Rules	102
	Summary	105
CHAPTER 7	Set Operations	107
	Overview of Basic Set Operations	108
	Union-Compatible Tables	109
	Union	111
	Ensuring Union Compatibility	112
	Selecting the Appropriate Columns	113
	Uses for Union	114
	Intersection	117
	Uses of Intersection	117
	The Importance of Projecting Appropriate Columns	120
	Managing Without the INTERSECT Keyword	122
	Difference	123
	Uses of Difference	124
	Managing Without the EXCEPT Keyword	126
	Division	127
	Projecting Appropriate Columns	129
	SQL for Division	130
	Summary	132

CHAPTER 8	Aggregate Operations	133
	Simple Aggregates	133
	The COUNT Function	133
	The AVG Function	136
	Other Aggregate Functions	138
	Grouping	139
	Filtering the Result of an Aggregate Query	143
	Using Aggregates to Perform Division Operations	145
	Nested Queries and Aggregates	147
	Summary	150
CHAPTER 9	Efficiency Considerations	153
	Indexes	153
	Types of Indexes	154
	Indexes for Efficiently Ordering Output	157
	Indexes and Joins	158
	What Should We Index?	160
	Query Optimizer	161
	What Does the Query Optimizer Consider?	161
	Does the Way We Express the Query Matter?	162
	Summary	167
CHAPTER 10	How to Approach a Query	169
	Understanding the Data	169
	Determine the Relationships Between Tables	169
	The Conceptual Model vs. the Implementation	171
	What Tables Are Involved?	173
	Look at Some Data Values	174
	Big Picture Approach	174
	Combine the Tables	175
	Find the Subset of Rows	176
	Retain the Appropriate Columns	177
	Consider an Intermediate View	178

Spotting Key Words in Questions	179
And, Both, Also	179
Not, Never	182
All, Every	183
No Idea Where to Start?	183
Find Some Helpful Tables	183
Try to Answer the Question by Hand	183
Write Down a Description of the Retrieved Result	184
Is There Another Way?	185
Checking Queries	186
Check a Row That Should Be Returned	187
Check a Row That Should Not Be Returned	187
Check Boundary Conditions.	187
Check Null Values.....	188
Summary	188

■ CHAPTER 11 Common Problems

Poor Database Design	191
Data That Is Not Normalized	191
No Keys	194
Similar Data in Two Tables	195
Wrong Types	196
Problems with Data Values	197
Unexpected Nulls	197
Wrong or Inconsistent Spelling	197
Extraneous Characters in Text Fields	198
Inconsistent Case in Text Fields	199
Diagnosing Problems	200
Check Parts of Nested Queries Independently	201
Understand How the Tables Are Being Combined	201
Remove Extra WHERE Clauses	201
Retain All the Columns.....	202
Check Underlying Queries in Aggregates.....	202

Common Symptoms 202

 No Rows Are Returned..... 202

 Rows Are Missing..... 203

 More Rows Than There Should Be 205

 Statistics or Aggregates Incorrect..... 207

 The Order Is Wrong..... 207

Common Typos and Syntax Problems 207

Summary 208

APPENDIX **Sample Database** 209

INDEX 211

About the Author



CLARE CHURCHER holds a Ph.D. in physics and has designed several databases for a variety of large and small projects. She is a senior academic in the Applied Computing Group at Lincoln University, where she recently won an Excellence in Teaching Award for her contribution to developing and presenting courses in analysis and design, databases, and programming. She has supervised more than 70 undergraduate projects designing databases for small projects.

About the Technical Reviewer



■ **DARL KUHN** is a senior database administrator with Sun Microsystems. Before joining Sun, his work as a consultant ranged from database administration to custom application development. Darl is a coauthor of *RMAN Recipes for Oracle Database 11g* and *Oracle RMAN Pocket Reference*. He is an affiliate professor at Regis University, where he teaches database courses for the department of computer information technology. Darl currently lives near Aguilar, Colorado, with his wife, Heidi, and two daughters, Lisa and Brandi.

Acknowledgments

Expecting your friends and family to put up with you writing a book is a tough call. Writing a second book the following year is really pushing their patience. So I am very indebted to my family and colleagues for putting up with me again. Special thanks to my two “first draft readers.” My husband, Neville Churcher, and my friend and colleague Theresa McLennan both gave me many valuable suggestions on the first attempts at every chapter, and always did it in Clare time (i.e., now!) and with very good grace. Thanks also to all my good friends in the Applied Computing Group at Lincoln University, especially Alan McKinnon for his advice on Chapter 9.

Many thanks to my editor Jonathan Gennick for his insight, expertise, and encouragement, and to both Jonathan and Donna for showing us around their lovely hometown of Munising. Beth Christmas, my Apress project manager, has been a pleasure to work with, and Dahl Kuhn has been a most careful and expert technical reviewer. Thanks to you both.

Introduction

As a query language, SQL is really quite small and should be easy to learn. A few basic ideas and a handful of keywords allow you to tackle a huge range of queries. However, many users often find themselves completely stumped when faced with a particular problem. You may find yourself in that group. It isn't really a great deal of help for someone to say, "This is how I would do it." What you need is a variety of ways to get started on a tricky problem. Once you have made a start on a query, you need to be able to check, amend, and refine your solution until you have what you need.

Two-Pronged Approach

Throughout this book, I approach different types of queries from two directions. The two approaches have their roots in relational algebra and calculus. Don't be alarmed though—I won't be delving into any complex mathematics. However, understanding a question and developing an appropriate SQL query do require logical thinking and precise definitions. The relational algebra and calculus approaches are both useful ways to grasp the logic and precision that are required to get accurate results.

The first approach, which has its roots in relational algebra, looks at *how* tables need to be manipulated in order to retrieve the subset of data you require. I describe the different types of operations that you can perform on tables, including joins, intersections, selections, and so on, and explain how to decide which might help in particular situations. Once you understand what operations are needed, translating them into SQL is relatively straightforward.

The second approach is what I use when I just can't figure out which operations will give me the required results. This approach, based on relational calculus, lets you describe *what* an expected row in your result might be like; that is, what conditions it must obey. By looking at the data, it is surprisingly easy to develop a semiformal description of what a "correct" retrieved row would be like (and, by implication, how you would recognize an "incorrect" row). Because SQL was originally based on relational calculus, translating this semiformal description into a working query is particularly straightforward.

I am always surprised at which approach my students take when confronting a new problem. Some will instantly see the algebra operations that are needed; others will find the calculus approach more obvious. The choice of approach changes from query to query, from person to person, and (I suspect) from day to day. Having more than one way to get started means you are less likely to be completely baffled by a new problem.

Who This Book Is For

This book is for anyone who has a well-designed relational database and needs to extract some information from it. You might have noticed in the previous sentence that the database must be “well designed.” I can’t overemphasize this point. If your database is badly designed, it will not be able to store accurate and consistent data, so the information your queries retrieve will always be prone to inaccuracies. If you are looking to design a database from scratch, you should read my first book, *Beginning Database Design* (Apress, 2007). The final chapter of this book outlines a few common design problems you are likely to come across and gives some advice about how to mitigate the impact or correct the problem.

For this book, you do not need any theoretical knowledge of relational theory, as I will explain the relevant issues as they come up. The first chapter gives a brief overview of relational database theory, but it will help if you have had some experience working with databases with a few or more tables.

Objective of This Book

In this book, you will be introduced to all the main techniques and keywords needed to create SQL queries. You will learn about joins, intersections, unions, differences, selection of rows, and projection of columns. You will see how to implement these ideas in different ways using simple and nested queries, and you will be introduced to a variety of aggregate functions and summary techniques. You can try out what you learn using the sample data provided through the Apress web page for this book (<http://www.apress.com/book/view/1590599438>). There you will find the Access database used for the examples in the book and some scripts to create the database on a number of other platforms.

Most important of all, you will learn different ways to get started on a troublesome problem. In almost all cases, there are several different ways to express a query. My objective is, for any particular situation, to provide you with a method of attack that matches your psyche and mood (just kidding).