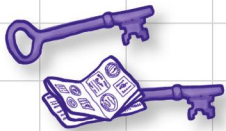


A Brain-Friendly Guide

Head First SQL



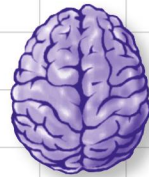
Help Greg
improve his data
relationships



Stop misplacing
your primary
and foreign keys



Finally be
able to explain
what's normal



Load important SQL
query concepts directly
into your brain



Avoid
embarrassing
ALTER
scenarios



Put your SQL knowledge
to the test with dozens
of exercises



O'REILLY®

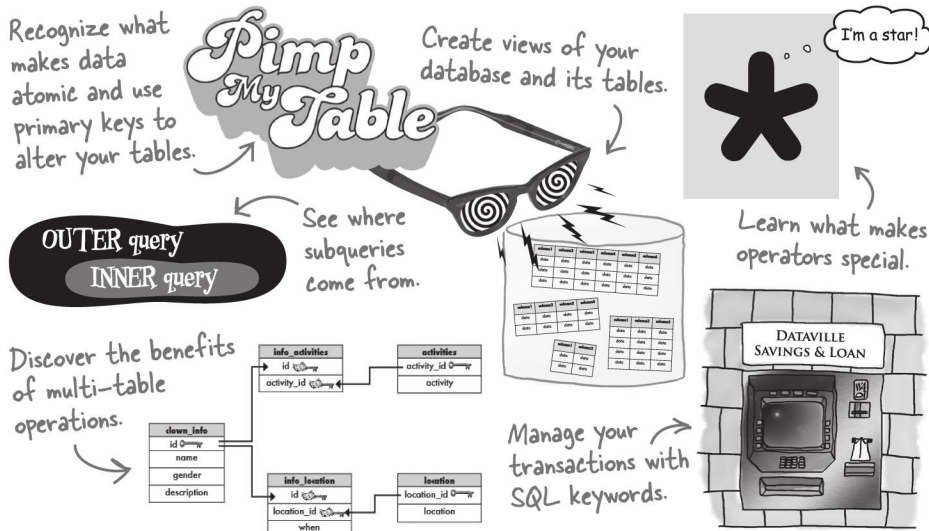
Lynn Beighley

Head First SQL

Database/SQL

What will you learn from this book?

In today's world, data is power, but the real secret to success is having power over your data. *Head First SQL* takes you to the heart of the SQL language, from basic syntax queries using INSERT and SELECT to hardcore database manipulation with subqueries, joins, and transactions. By the time you finish reading, you will not only understand effective and efficient database design and creation, you'll be querying, normalizing, and joining data like a pro. And you'll be a true master of your data.



“This is not SQL made easy; this is SQL made challenging, SQL made interesting, SQL made fun. It even answers that age-old question ‘How to teach non-correlated subqueries without losing the will to live?’ This is the right way to learn—it’s fast, it’s flippant, and it looks fabulous.”

—Andrew Cumming,
Author of *SQL Hacks*,
Zoo Keeper at sqlzoo.net

“There are books you buy, books you keep, books you keep on your desk, and thanks to O’Reilly and the Head First crew, there is the penultimate category, Head First books. They’re the ones that are dog-eared, mangled, and carried everywhere. *Head First SQL* is at the top of my stack.”

— Bill Sawyer,
ATG Curriculum Manager,
Oracle

US \$44.99

CAN \$53.99

ISBN-10: 0-596-52684-9

ISBN-13: 978-0-596-52684-9



O'REILLY®

www.oreilly.com
www.headfirstlabs.com

Praise for *Head First SQL*

“There are books you buy, books you keep, books you keep on your desk, and thanks to O’Reilly and the Head First crew, there is the ultimate category, Head First books. They’re the ones that are dog-eared, mangled, and carried everywhere. Head First SQL is at the top of my stack. Heck, even the PDF I have for review is tattered and torn.”

— **Bill Sawyer, ATG Curriculum Manager, Oracle**

“This is not SQL made easy; this is SQL made challenging, SQL made interesting, SQL made fun. It even answers that age-old question ‘How to teach non-correlated subqueries without losing the will to live?’ This is the right way to learn—it’s fast, it’s flippant, and it looks fabulous.”

— **Andrew Cumming, Author of *SQL Hacks*, Zoo Keeper at sqlzoo.net**

“Outrageous! I mean, SQL is a *computer* language, right? So books about SQL should be written for *computers*, shouldn’t they? *Head First SQL* is *obviously* written for *human beings*! What’s up with *that*?!”

— **Dan Tow, Author of *SQL Tuning***

“Even many of the more advanced concepts are covered in a way that almost anyone should be able to easily follow...if your database does ever grow to the point where you need to use the more advanced commands this same book will show you how. The book not only teaches you how to code SQL, it also teaches many of the concepts of proper database design.”

— **Stephen Chapman, Ask Felgall (www.felgall.com/hfsql.htm)**

“Jammed with exercises, thoughtful questions, cartoons and side comments that make you giggle while making an important point, this book provides the most enjoyable way to learn SQL than I can imagine—well, short of learning it pool side in the Bahamas anyway. It even includes crossword puzzles to help the reader quiz himself on what he’s learned. If you want to have a fun time learning SQL, this is the way to do it. It gets down to hardcore SQL and keeps you thinking not just about what you’re doing, but why you’re doing it with its lively examples. This is gain without the pain. This is a fun way to learn.

— **Sandra Henry-Stocker, ITworld.com**

Praise for other *Head First* books

“This book’s admirable clarity, humor and substantial doses of clever make it the sort of book that helps even non-programmers think well about problem-solving.”

— **Cory Doctorow, co-editor of *Boing Boing***
Author, *Down and Out in the Magic Kingdom*
and *Someone Comes to Town, Someone Leaves Town*

“If you thought Ajax was rocket science, this book is for you. Head Rush Ajax puts dynamic, compelling experiences within reach for every web developer.”

— **Jesse James Garrett, *Adaptive Path***

“I received the book yesterday and started to read it...and I couldn’t stop. This is definitely très ‘cool.’ It is fun, but they cover a lot of ground and they are right to the point. I’m really impressed.”

— **Erich Gamma, IBM Distinguished Engineer, and co-author of *Design Patterns***

“*Head First Design Patterns* managed to mix fun, belly-laughs, insight, technical depth and great practical advice in one entertaining and thought provoking read. Whether you are new to design patterns, or have been using them for years, you are sure to get something from visiting Objectville.”

— **Richard Helm, co-author of *Design Patterns***

“One of the funniest and smartest books on software design I’ve ever read.”

— **Aaron LaBerge, VP Technology, ESPN.com**

“I just finished reading HF OOA&D and I loved it! The thing I liked most about this book was its focus on why we do OOA&D—to write great software!”

— **Kyle Brown, Distinguished Engineer, IBM**

I *heart* *Head First HTML with CSS & XHTML*—it teaches you everything you need to learn in a ‘fun coated’ format!”

— **Sally Applin, UI Designer and Fine Artist, <http://sally.com>**

Praise for the *Head First* Approach

“It’s fast, irreverant, fun, and engaging. Be careful—you might actually learn something!”

— **Ken Arnold, former Senior Engineer at Sun Microsystems**
Co-author (with James Gosling of Java),
The Java Programming Language

“I feel like a thousand pounds of books have just been lifted off of my head.”

— **Ward Cunningham, inventor of the Wiki**
and founder of the Hillside Group

“This book is close to perfect, because of the way it combines expertise and readability. It speaks with authority and it reads beautifully.”

— **David Gelernter, Professor of Computer Science, Yale University**

“Just the right tone for the geeked-out, casual-cool guru coder in all of us. The right reference for practical development strategies--gets my brain going without having to slog through a bunch of tired, stale professor-speak.”

— **Travis Kalanick, Founder of Scour and Red Swoosh Member of the**
MIT TR100

“The combination of humour, pictures, asides, sidebars, and redundancy with a logical approach to introducing the basic tags and substantial examples of how to use them will hopefully have the readers hooked in such a way that they don’t even realize they are learning because they are having so much fun.”

— **Stephen Chapman, Fellgall.com**

Other related books from O'Reilly

The Art of SQL

Learning SQL

SQL in a Nutshell

SQL Cookbook

SQL Hacks

SQL Pocket Guide

The Relational Database Dictionary

Database in Depth

Other books in O'Reilly's *Head First* series

Head First PMP

Head First Object-Oriented Analysis and Design

Head Rush Ajax

Head First HTML with CSS and XHTML

Head First Design Patterns

Head First Servlets & JSP™

Head First Java™

Head First EJB™

Head First JavaScript (2007)

Head First C# (2007)

Head First SQL

Wouldn't it be dreamy if there was a book that could teach me SQL without making me want to relocate to a remote island in the Pacific where there are no databases? It's probably nothing but a fantasy...



Lynn Beighley

O'REILLY®

Beijing • Cambridge • Köln • Sebastopol • Tokyo

Head First SQL

by Lynn Beighley

Copyright © 2007 O'Reilly Media, Inc. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly Media books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*safari.oreilly.com*). For more information, contact our corporate/institutional sales department: (800) 998-9938 or *corporate@oreilly.com*.

Series Creators:	Kathy Sierra, Bert Bates
Series Editor:	Brett D. McLaughlin
Editor:	Catherine Nolan
Design Editor:	Louise Barr
Cover Designers:	Louise Barr, Karen Montgomery
Production Editor:	Sanders Kleinfeld
Indexer:	Julie Hawks
Page Viewer:	Andrew Fader

Printing History:

August 2007: First Edition.

He's incredibly patient.



The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. The *Head First* series designations, *Head First SQL*, and related trade dress are trademarks of O'Reilly Media, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc., was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and the authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

No clowns, doughnuts, or Girl Sprouts were harmed in the making of this book. Just my car, but it's been fixed.



This book uses RepKover™, a durable and flexible lay-flat binding.

ISBN: 978-0-596-52684-9

[M]

[2011-02-25]

To our world, awash in data.
And to you, who want to master it.

Author of Head First SQL



↑
Lynn Beighley

Lynn is a fiction writer stuck in a technical book writer's body. Upon discovering that technical book writing actually paid real money, she learned to accept and enjoy it.

After going back to school to get a Masters in computer science, she worked for the acronyms NRL and LANL. Then she discovered Flash, and wrote her first bestseller.

A victim of bad timing, she moved to Silicon Valley just before the great crash. She spent several years working for Yahoo! and writing other books and training courses. Finally giving in to her creative writing bent, she moved to the New York area to get an MFA in creative writing.

Her Head First-style thesis was delivered to a packed room of professors and fellow students. It was extremely well received, and she finished her degree, finished *Head First SQL*, and can't wait to begin her next book.

Lynn loves traveling, cooking, and making up elaborate background stories about complete strangers. She's a little scared of clowns.



↑
The view from
Lynn's window.

Table of Contents (Summary)

	Intro	xxv
1	Data and Tables: <i>A place for everything</i>	1
2	The SELECT Statement: <i>Gifted data retrieval</i>	53
3	DELETE and UPDATE: <i>A change will do you good</i>	119
4	Smart Table Design: <i>Why be normal?</i>	159
5	ALTER: <i>Rewriting the past</i>	197
6	Advanced SELECT: <i>Seeing your data with new eyes</i>	235
7	Multi-table Database Design: <i>Outgrowing your table</i>	281
8	Joins and Multi-table Operations: <i>Can't we all just get along?</i>	343
9	Subqueries: <i>Queries Within Queries</i>	379
10	Outer Joins, Self Joins, and Unions: <i>New maneuvers</i>	417
11	Constraints, Views, and Transactions: <i>Too many cooks spoil the database</i>	455
12	Security: <i>Protecting your assets</i>	493

Table of Contents (the real thing)

Intro

Your brain on SQL. Here *you* are trying to *learn* something, while here your *brain* is doing you a favor by making sure the learning doesn't *stick*. Your brain's thinking, "Better leave room for more important things, like which wild animals to avoid and whether naked snowboarding is a bad idea." So how *do* you trick your brain into thinking that your life depends on knowing SQL?

Who is this book for?	xxvi
We know what you're thinking	xxvii
Metacognition	xxix
Bend your brain into submission	xxxi
Read me	xxxii
The technical review team	xxxiv
Acknowledgments	xxxv

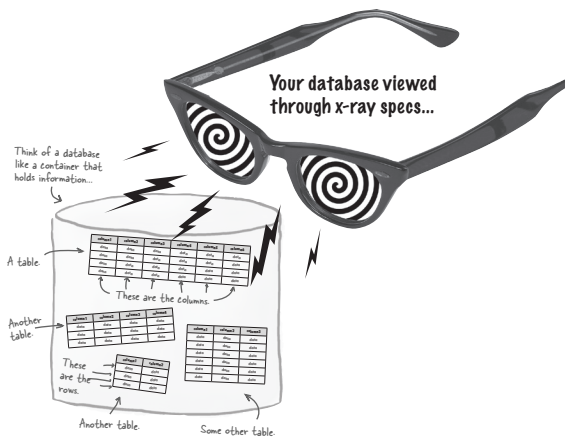
data and tables

A place for everything

1

Don't you just hate losing things? Whether it's your car keys, that 25% off coupon for Urban Outfitters, or your application's data, there's nothing worse than not being able to **keep up with what you need...** when you need it. And when it comes to your applications, there's no better place to store your important information than in a **table**. So turn the page, come on in, and take a walk through the world of **relational databases**.

Defining your data	2
Look at your data in categories	7
What's in a database?	8
Your database viewed through x-ray specs...	10
Databases contain connected data	12
Tables Up Close	13
Take command!	17
Setting the table: the CREATE TABLE statement	19
Creating a more complicated table	20
Look how easy it is to write SQL	21
Create the my_contacts table, finally	22
Your table is ready	23
Take a meeting with some data types	24
Your table, DESCRibed	28
You can't recreate an existing table or database!	30
Out with the old table, in with the new	32
To add data to your table, you'll use the INSERT statement	34
Create the INSERT statement	37
Variations on an INSERT statement	41
Columns without values	42
Peek at your table with the SELECT statement	43
SQL Exposed: Confessions of a NULL	44
Controlling your inner NULL	45
NOT NULL appears in DESC	47
Fill in the blanks with DEFAULT	48
Your SQL Toolbox	50

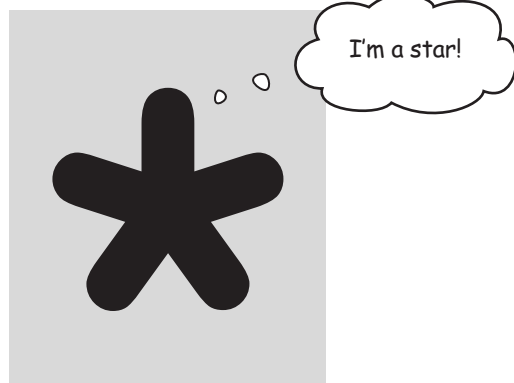


the *SELECT* statement**Gifted data retrieval**

2

Is it really better to give than retrieve? When it comes to databases, chances are you'll need to **retrieve your data** as often than you'll need to insert it. That's where this chapter comes in: you'll meet the powerful **SELECT** statement and learn how to **gain access to that important information** you've been putting in your tables. You'll even learn how to use **WHERE**, **AND**, and **OR** to selectively get to your data and even avoid displaying the data that you *don't* need.

Date or no date?	54
A better SELECT	57
What the * is that?	58
How to query your data types	64
More punctuation problems	65
Unmatched single quotes	66
Single quotes are special characters	67
INSERT data with single quotes in it	68
SELECT specific columns to limit results	73
SELECT specific columns for faster results	73
Combining your queries	80
Finding numeric values	83
Smooth Comparison Operators	86
Finding numeric data with Comparison Operators	88
Text data roping with Comparison Operators	91
To be OR not to be	93
The difference between AND and OR	96
Use IS NULL to find NULLs	99
Saving time with a single keyword: LIKE	101
The call of the Wild(card)	101
Selecting ranges using AND and comparison operators	105
Just BETWEEN us... there's a better way	106
After the dates, you are either IN...	109
... or you are NOT IN	110
More NOT	111
Your SQL Toolbox	116



DELETE and UPDATE

A change will do you good

3

Keep changing your mind? Now it's OK! With the commands you're about to learn—**DELETE** and **UPDATE**—you're no longer stuck with a decision you made six months ago, when you first inserted that data about mullets coming back into style soon. With **UPDATE**, you **can change data**, and **DELETE** lets you **get rid of data** that you don't need anymore. But we're not just giving you the tools; in this chapter, you'll learn how to be selective with your new powers and avoid dumping data that you really do need.

Clowns are scary	120
Clown tracking	121
The clowns are on the move	122
How our clown data gets entered	126
Bonzo, we've got a problem	128
Getting rid of a record with DELETE	129
Using our new DELETE statement	131
DELETE rules	132
The INSERT-DELETE two step	135
Be careful with your DELETE	140
The trouble with imprecise DELETE	144
Change your data with UPDATE	146
UPDATE rules	147
UPDATE is the new INSERT-DELETE	148
UPDATE in action	149
Updating the clowns' movements	152
UPDATE your prices	154
All we need is one UPDATE	156
Your SQL Toolbox	158



smart table design

Why be normal?

4

You’ve been creating tables without giving much thought to them. And that’s fine, they work. You can `SELECT`, `INSERT`, `DELETE`, and `UPDATE` with them. But as you **get more data**, you start seeing **things you wish you’d done** to make your `WHERE` clauses simpler. What you need is to make your tables more *normal*.

Two fishy tables	160
A table is all about relationships	164
Atomic data	168
Atomic data and your tables	170
Atomic data rules	171
Reasons to be normal	174
The benefits of normal tables	175
Clowns aren’t normal	176
Halfway to 1NF	177
PRIMARY KEY rules	178
Getting to NORMAL	181
Fixing Greg’s table	182
The CREATE TABLE we wrote	183
Show me the table money	184
Time-saving command	185
The CREATE TABLE with a PRIMARY KEY	186
1, 2, 3... auto incrementally	188
Adding a PRIMARY KEY to an existing table	192
ALTER TABLE and add a PRIMARY KEY	193
Your SQL Toolbox	194

Wait a second. I already have a table full of data. You can’t seriously expect me to use the `DROP TABLE` command like I did in chapter 1 and type in all that data again, just to create a primary key for each record...



ALTER
Rewriting the Past

5

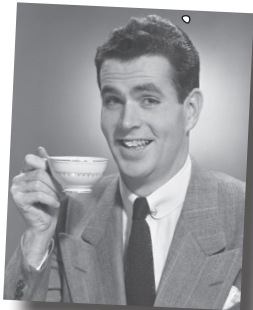
Ever wished you could correct the mistakes of your past?

Well, now is your chance. By using the **ALTER command**, you can apply all the lessons you've been learning to tables you designed days, months, even years ago. Even better, you can do it without affecting your data. By the time you're through here, you'll know what **normal** really means, and you'll be able to apply it to all your tables, past and present.

We need to make some changes	198
Table altering	203
Extreme table makeover	204
Renaming the table	205
We need to make some plans	207
Retooling our columns	208
Structural changes	209
ALTER and CHANGE	210
Change two columns with one SQL statement	211
Quick! DROP that column	215
A closer look at the non-atomic location column	222
Look for patterns	223
A few handy string functions	224
Use a current column to fill a new column	229
How our UPDATE and SET combo works	230
Your SQL Toolbox	232



It's time to turn your tired old hooptie table into a date magnet and take it to a level of table pimpification you never knew existed.



advanced SELECT

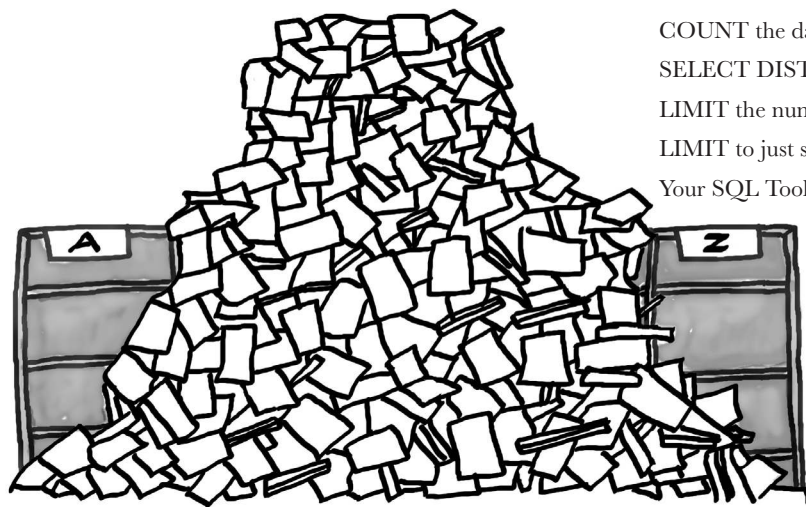
Seeing your data with new eyes

6

It's time to add a little finesse to your toolbox. You already know how to SELECT data and use WHERE clauses. But sometimes you need more **precision** than SELECT and WHERE provide. In this chapter, you'll learn about how to **order and group** your data, as well as how to **perform math operations** on your results.

Dataville Video is reorganizing	236
Problems with our current table	237
Matching up existing data	238
Populating the new column	239
UPDATE with a CASE expression	242
Looks like we have a problem	244
Tables can get messy	249
We need a way to organize the data we SELECT	250
Try a little ORDER BY	253
ORDER a single column	254
ORDER with two columns	257
ORDER with multiple columns	258
An orderly movie_table	259
Reverse the ORDER with DESC	261
The Girl Sprout® cookie sales leader problem	263
SUM can add them for us	265
SUM all of them at once with GROUP BY	266
AVG with GROUP BY	267
MIN and MAX	268
COUNT the days	269
SELECT DISTINCT values	271
LIMIT the number of results	274
LIMIT to just second place	275
Your SQL Toolbox	278

**DATAVILLE
Video**



multi-table database design

7 Outgrowing your table

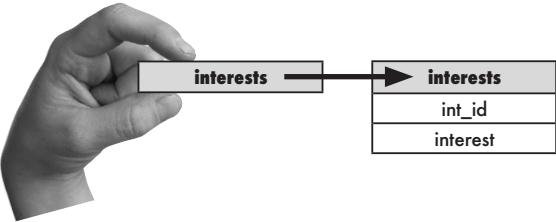
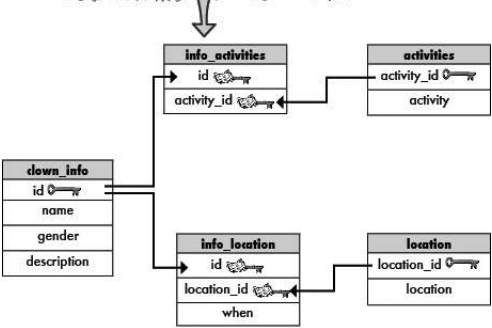
Sometimes your single table isn't big enough anymore.

Your data has become more complex, and that **one table** you've been using just **isn't cutting it**. Your single table is full of redundant data, wasting space and slowing down your queries. You've gone as far as you can go with a single table. It's a big world out there, and sometimes you need **more than one table** to contain your data, control it, and ultimately, be the master of your own database.

Finding Nigel a date	282
All is lost... But wait	293
Think outside of the single table	294
The multi-table clown tracking database	295
The clowntracking database schema	296
How to go from one table to two	298
Connecting your tables	303
Constraining your foreign key	305
Why bother with foreign keys?	306
CREATE a table with a FOREIGN KEY	307
Relationships between tables	309
Patterns of data: one-to-one	309
Patterns of data: when to use one-to-one tables	310
Patterns of data: one-to-many	311
Patterns of data: getting to many-to-many	312
Patterns of data: we need a junction table	315
Patterns of data: many-to-many	316
Finally in 1NF	321
Composite keys use multiple columns	322
Shorthand notations	324
Partial functional dependency	325
Transitive functional dependency	326
Second normal form	330
Third normal form (at last)	336
And so, Regis (and gregslis) lived happily ever after	339
Your SQL Toolbox	340

clown_tracking

clown_id	name	last_name	activities
Elmo	Cherry Hill Senior Center	F, red hair, green dress, huge feet	ballroom, tifo, air
Pablo	Jack Green's party	hair, like red, huge feet	none
Boogles	Ballroom	hair, long, blue hair	ball, ventrilo



joins and multi-table operations

8

Can't we all just get along?

Welcome to a multi-table world. It's great to have **more than one table** in your database, but you'll need to learn some **new tools and techniques** to work with them. With multiple tables comes confusion, so you'll need **aliases** to keep your tables straight. And **joins** help you connect your tables, so that you can get at all the data you've spread out. Get ready, it's time to **take control of your database** again.

Still repeating ourselves, still repeating...	344
Prepopulate your tables	345
We got the "table ain't easy to normalize" blues	347
The special interests (column)	348
Keeping interested	349
UPDATE all your interests	350
Getting all the interests	351
Many paths to one place	352
CREATE, SELECT and INSERT at (nearly) the same time	352
CREATE, SELECT and INSERT at the same time	353
What's up with that AS?	354
Column aliases	355
Table aliases, who needs 'em?	356
Everything you wanted to know about inner joins	357
Cartesian join	358
Releasing your inner join	363
The inner join in action: the equijoin	364
The inner join in action: the non-equijoin	367
The last inner join: the natural join	368
Joined-up queries?	375
Table and Column Aliases Exposed: What are you hiding from?	376
Your SQL Toolbox	377

...and that's where
little result tables
really come from.



subqueries

9

Queries within queries

Yes, Jack, I'd like a two-part question, please. Joins are great, but sometimes you need to *ask your database more than one question*. Or *take the result of one query and use it as the input to another query*. That's where **subqueries** come in. They'll help you **avoid duplicate data, make your queries more dynamic**, and even get you in to all those high-end concert afterparties. (Well, not really, but two out of three ain't bad!)

Greg gets into the job recruiting business	380
Greg's list gets more tables	381
Greg uses an inner join	382
But he wants to try some other queries	384
Subqueries	386
We combine the two into a query with a subquery	387
As if one query wasn't enough: meet the subquery	388
A subquery in action	389
Subquery rules	391
A subquery construction walkthrough	394
A subquery as a SELECT column	397
Another example: Subquery with a natural join	398
A noncorrelated subquery	399
SQL Exposed: Choosing the best way to query	400
A noncorrelated subquery with multiple values: IN, NOT IN	403
Correlated subqueries	408
A (useful) correlated subquery with NOT EXISTS	409
EXISTS and NOT EXISTS	410
Greg's Recruiting Service is open for business	412
On the way to the party	413
Your SQL Toolbox	414

