Head First SQL

Lynn Beighley

O'REILLY®

Beijing • Cambridge • Köln • Sebastopol • Taipei • Tokyo

Head First SQL

by Lynn Beighley

Printed in the United States of America.

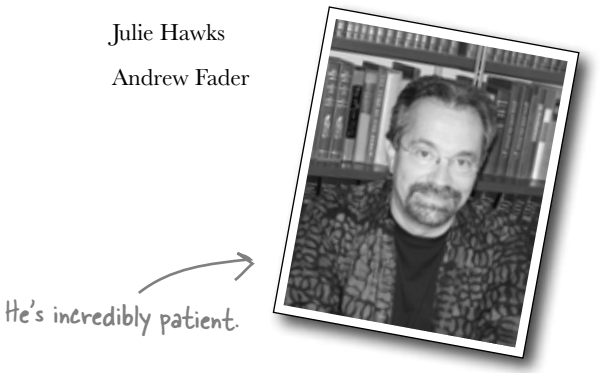
Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

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.



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

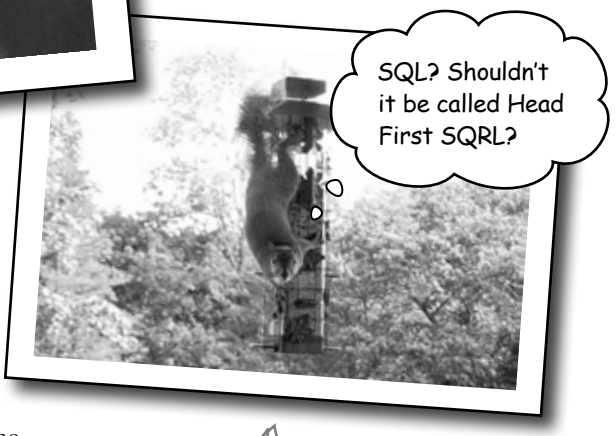
This book uses RepKover™, a durable and flexible lay-flat binding.

ISBN: 978-0-596-52684-9

[M] [6/09]

Author of Head First SQL

Lynn Beighley



The view from Lynn's window.

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.

Table of Contents (Summary)

Table of Contents (the real thing)

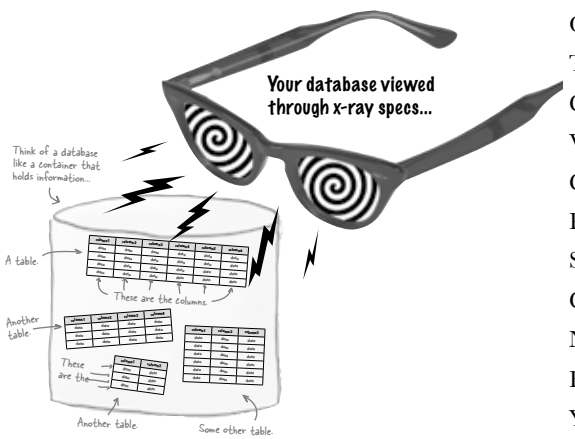
Intro

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?

1 data and tables

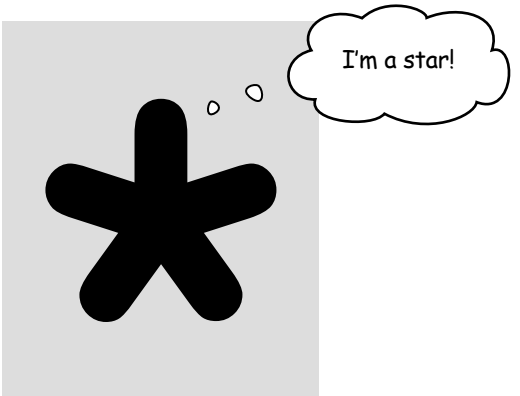
A place for everything

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**.



2 the SELECT statement
Gifted data retrieval

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.



3 DELETE and UPDATE

A change will do you good

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.

4 smart table design

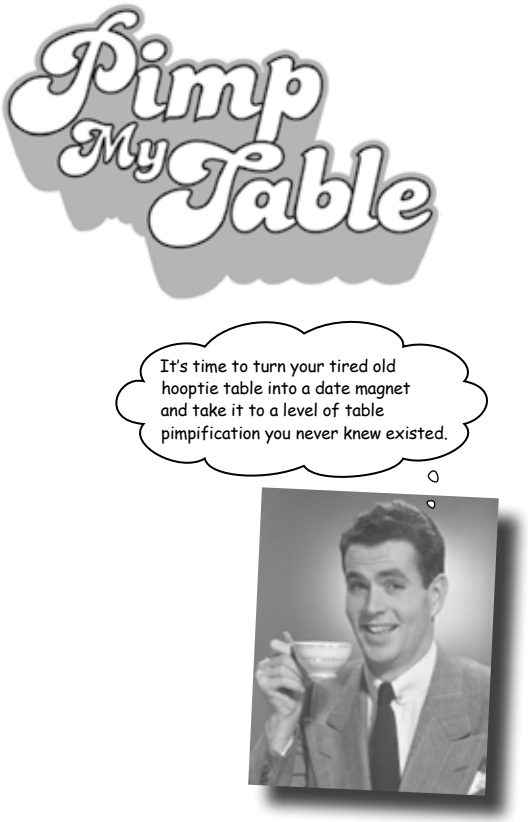
Why be normal?

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*.

5 ALTER
Rewriting the Past

Ever wished you could correct the mistakes of your past?

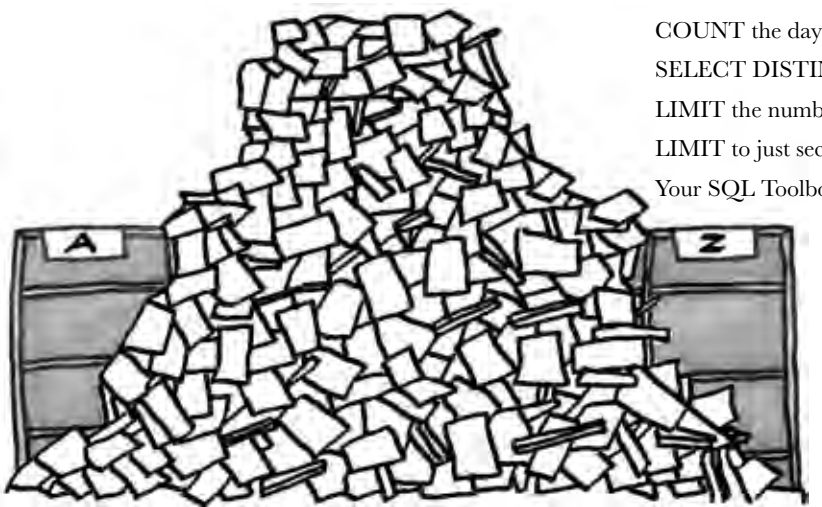
Well, now is your chance. By using the **ALTER command**, you can apply all the lessons you've been learning to tables you designed days, months, even years ago. Even better, you can do it without affecting your data. By the time you're through here, you'll know what **normal** really means, and you'll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



6 advanced SELECT

Seeing your data with new eyes

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.

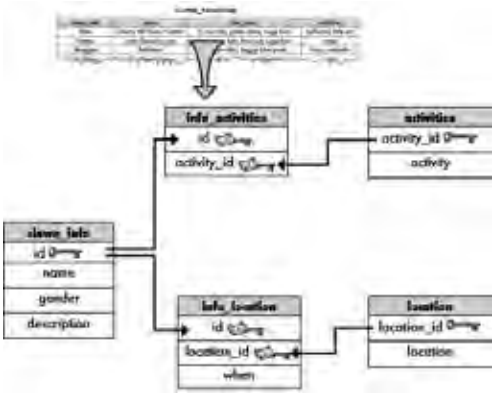


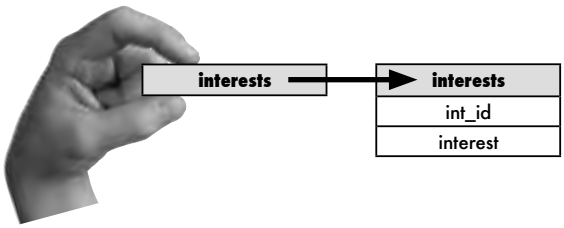
multi-table database design

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.





8 joins and multi-table operations

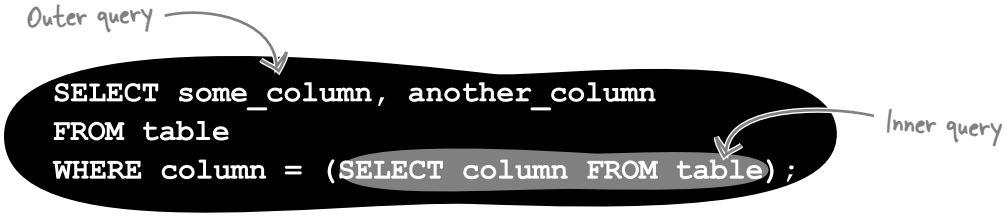
Can't we all just get along?

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.

subqueries

9 Queries within queries

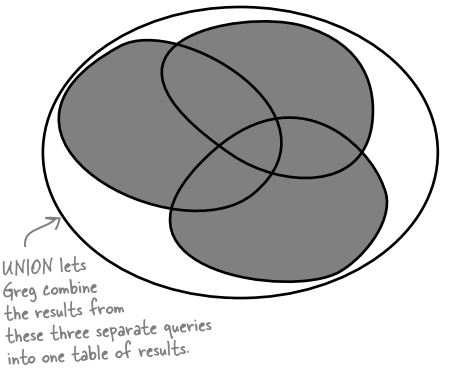
Yes, Jack, I'd like a two-part question, please. Joins are great, but sometimes you need to ***ask your database more than one question***. Or ***take the result of one query and use it as the input to another query***. That's where **subqueries** come in. They'll help you **avoid duplicate data**, **make 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!)

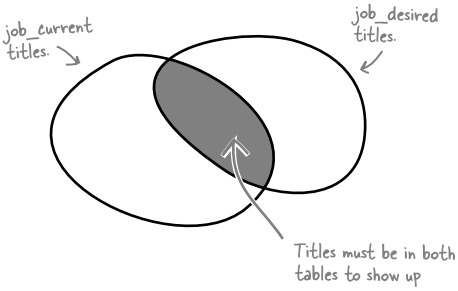


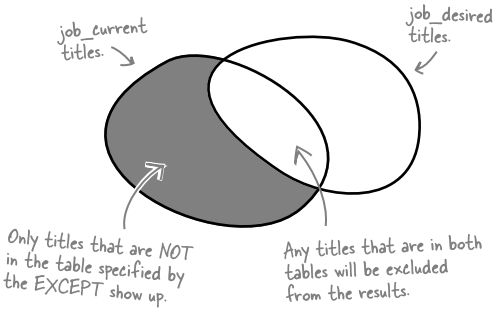
outer joins, self-joins, and unions

10 New maneuvers

You only know half of the story about joins. You've seen cross joins that return every possible row, and inner joins that return rows from both tables where there is a match. But what you haven't seen are **outer joins** that give you back rows that *don't have matching counterparts in the other table*, **self-joins** which (strangely enough) *join a single table to itself*, and **unions** that c*ombine the results of queries*. Once you learn these tricks, you'll be able to get at all your data exactly the way you need to. (And we haven't forgotten about exposing the truth about subqueries, either!)





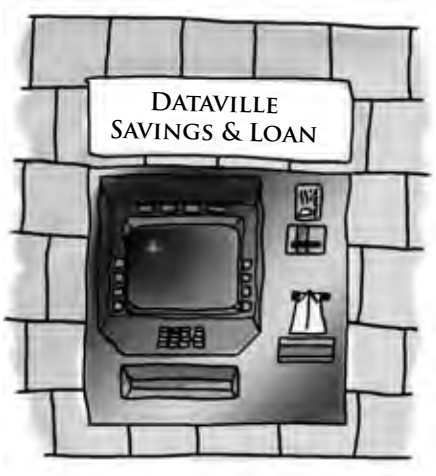


11 constraints, views, and transactions

Too many cooks spoil the database

Your database has grown and other people need to use it.

The problem is that some of them won't be as skilled at SQL as you are. You need ways to **keep them from entering the wrong data**, techniques for allowing them to **only see part of the data**, and ways to **stop them from stepping on each other when they try entering data at the same time**. In this chapter we begin protecting our data from the mistakes of others. Welcome to Defensive Databases, Part 1.

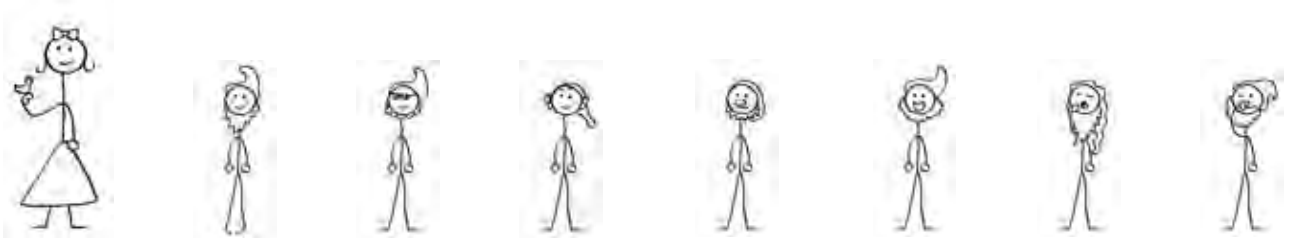


12

security

Protecting your assets

You've put an enormous amount of time and energy into creating your database. And you'd be devastated if anything happened to it. You've also had to give other people **access to your data**, and you're worried that they might insert or update something incorrectly, or even worse, **delete the wrong data**. You're about to learn how databases and the objects in them can be made more **secure**, and how you can have complete control over ***who can do what with your data.***



leftovers

The Top Ten Topics (we didn't cover)

Even after all that, there's a bit more. There are just a few more things we think you need to know. We wouldn't feel right about ignoring them, even though they only need a brief mention. So before you put the book down, take a read through these **short but important SQL tidbits**. Besides, once you're done here, all that's left is another appendix... and the index... and maybe some ads... and then you're really done. We promise!

A	ABSOLUTE ACTION ADD ADMIN AFTER AGGREGATE ALIAS ALL ALLOCATE ALTER AND ANY ARE ARRAY AS ASC ASSERTION AT AUTHORIZATION
B	BEFORE BEGIN BINARY BIT BLOB BOOLEAN BOTH BREADTH BY
C	CALL CASCADE CASCADED CASE CAST CATALOG CHAR CHARACTER CHECK CLASS CLOB CLOSE COLLATE COLLATION COLUMN COMMIT COMPLETION CONNECT CONNECTION CONSTRAINT CONSTRAINTS CONSTRUCTOR CONTINUE CORRESPONDING CREATE CROSS CUBE CURRENT CURRENT_DATE CURRENT_PATH CURRENT_ROLE CURRENT_TIME CURRENT_TIMESTAMP CURRENT_USER CURSOR CYCLE
D	DATA DATE DAY DEALLOCATE DEC DECIMAL DECLARE DEFAULT DEFERRABLE DEFERRED DELETE DEPTH DEREF DESC DESCRIBE DESCRIPTOR DESTROY DESTRUCTOR DETERMINISTIC DICTIONARY DIAGNOSTICS DISCONNECT DISTINCT DOMAIN DOUBLE DROP DYNAMIC
E	EACH ELSE END END_EXEC EQUALS ESCAPE EVERY EXCEPT EXCEPTION EXEC EXECUTE EXTERNAL
F	FALSE FETCH FIRST FLOAT FOR FOREIGN FOUND FROM FREE FULL FUNCTION
G	GENERAL GET GLOBAL GO GOTO GRANT GROUP GROUPING
H	HAVING HOST HOUR
I	IDENTITY IGNORE IMMEDIATE IN INDICATOR INITIALIZE INITIALLY INNER INOUT INPUT INSERT INT INTEGER INTERSECT INTERVAL INTO IS ISOLATION ITERATE
J	JOIN
K	KEY
L	LANGUAGE LARGE LAST LATERAL LEADING LEFT LESS LEVEL LIKE LIMIT LOCAL LOCALTIME LOCALTIMESTAMP LOCATOR
M	MAP MATCH MINUTE MODIFIES MODIFY MODULE MONTH
N	NAMES NATIONAL NATURAL NCHAR NCLOB NEW NEXT NO NONE NOT NULL NUMERIC
O	OBJECT OF OFF OLD ON ONLY OPEN OPERATION OPTION OR ORDER ORDINALITY OUT OUTER OUTPUT
P	PAD PARAMETER PARAMETERS PARTIAL PATH POSTFIX PRECISION PREFIX PREORDER PREPARE PRESERVE PRIMARY PRIOR PRIVILEGES PROCEDURE PUBLIC
Q	
R	READ READS REAL RECURSIVE REF REFERENCES REFERENCING RELATIVE RESTRICT RESULT RETURN RETURNS REVOKE RIGHT ROLE ROLLBACK ROLLUP ROUTINE ROW ROWS
S	SAVEPOINT SCHEMA SCROLL SCOPE SEARCH SECOND SECTION SELECT SEQUENCE SESSION SESSION_USER SET SETS SIZE SMALLINT SOME SPACE SPECIFIC SPECIFICTYPE SQL SQLEXCEPTION SQLSTATE SQLWARNING START STATE STATEMENT STATIC STRUCTURE SYSTEM_USER
T	TABLE TEMPORARY TERMINATE THAN THEN TIME TIMESTAMP TIMEZONE_HOUR TIMEZONE_MINUTE TO TRAILING TRANSACTION TRANSLATION TREAT TRIGGER TRUE
U	UNDER UNION UNIQUE UNKNOWN UNNEST UPDATE USAGE USER USING
V	VALUE VALUES VARCHAR VARIABLE VARYING VIEW
W	WHEN WHENEVER WHERE WITH WITHOUT WORK WRITE
X	
Y	YEAR
Z	ZONE

```
File Edit Window Help
> SELECT CURRENT_DATE;
+--------------+
| CURRENT_DATE |
+--------------+
| 2007-07-26   |
+--------------+
1 row in set (0.00 sec)
```

```
File Edit Window Help
> SELECT CURRENT_TIME;
+--------------+
| CURRENT_TIME |
+--------------+
| 11:26:48     |
+--------------+
1 row in set (0.00 sec)
```

```
File Edit Window Help
SELECT CURRENT_USER;
+----------------+
| CURRENT_USER   |
+----------------+
| root@localhost |
+----------------+
1 row in set (0.00 sec)
```

mySQL installation

Try it out for yourself

All your new SQL skills won't do you much good without a place to apply them. This appendix contains instructions for getting your very own MySQL RDBMS for you to work with.

tools roundup

All your new SQL tools

Here are all your SQL tools in one place for the first time, for one night only (kidding)! This is a roundup of all the SQL tools we've covered. Take a moment to *survey the list and feel **great***—**you learned them all**!