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
# Table of Contents (Summary)

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

# 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

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

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
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

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?

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 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
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.

We need to make some changes
Table altering
Extreme table makeover
Renaming the table
We need to make some plans
Retooling our columns
Structural changes
ALTER and CHANGE
Change two columns with one SQL statement
Quick! DROP that column
A closer look at the non-atomic location column
Look for patterns
A few handy string functions
Use a current column to fill a new column
How our UPDATE and SET combo works
Your SQL Toolbox
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.

Dataville Video is reorganizing
Problems with our current table
Matching up existing data
Populating the new column
UPDATE with a CASE expression
Looks like we have a problem
Tables can get messy
We need a way to organize the data we SELECT
Try a little ORDER BY
ORDER a single column
ORDER with two columns
ORDER with multiple columns
An orderly movie_table
Reverse the ORDER with DESC
The Girl Sprout® cookie sales leader problem
SUM can add them for us
SUM all of them at once with GROUP BY
AVG with GROUP BY
MIN and MAX
COUNT the days
SELECT DISTINCT values
LIMIT the number of results
LIMIT to just second place
Your SQL Toolbox
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 gregslist) lived happily ever after 339
Your SQL Toolbox 340
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.

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
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
- Greg's list gets more tables
- Greg uses an inner join
- But he wants to try some other queries
- Subqueries
- We combine the two into a query with a subquery
- As if one query wasn't enough: meet the subquery
- A subquery in action
- Subquery rules
- A subquery construction walkthrough
- A subquery as a SELECT column
- Another example: Subquery with a natural join
- A noncorrelated subquery
- SQL Exposed: Choosing the best way to query
- A noncorrelated subquery with multiple values: IN, NOT IN
- Correlated subqueries
- A (useful) correlated subquery with NOT EXISTS
- EXISTS and NOT EXISTS
- Greg's Recruiting Service is open for business
- On the way to the party
- Your SQL Toolbox

**SELECT** some_column, another_column
**FROM** table
WHERE column = (SELECT column FROM table);
outer joins, self-joins, and unions

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 combine 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!)

Cleaning up old data
It's about left and right
Here's a left outer join
Outer joins and multiple matches
The right outer join
While you were outer joining...
We could create a new table
How the new table fits in
A self-referencing foreign key
Join the same table to itself
We need a self-join
Another way to get multi-table information
You can use a UNION
UNION is limited
UNION rules in action
UNION ALL
Create a table from your union
INTERSECT and EXCEPT
We're done with joins, time to move on to...
Subqueries and joins compared
Turning a subquery into a join
A self-join as a subquery
Greg's company is growing
Your SQL Toolbox
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.

- Greg’s hired some help
- Jim’s first day: Inserting a new client
- Jim avoids a NULL
- Flash forward three months
- CHECK, please: Adding a CHECK CONSTRAINT
- CHECKing the gender
- Frank’s job gets tedious
- Creating a view
- Viewing your views
- What your view is actually doing
- What a view is
- Inserting, updating, and deleting with views
- The secret is to pretend a view is a real table
- View with CHECK OPTION
- Your view may be updatable if...
- When you’re finished with your view
- When bad things happen to good databases
- What happened inside the ATM
- More trouble at the ATM
- It’s not a dream, it’s a transaction
- The classic ACID test
- SQL helps you manage your transactions
- What should have happened inside the ATM
- How to make transactions work with MySQL
- Now try it yourself
- Your SQL Toolbox
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.

User problems 494
Avoiding errors in the clown tracking database 495
Protect the root user account 497
Add a new user 498
Decide exactly what the user needs 499
A simple GRANT statement 500
GRANT variations 503
REVOKE privileges 504
REVOKEING a used GRANT OPTION 505
REVOKEING with precision 506
The problem with shared accounts 510
Using your role 512
Role dropping 512
Using your role WITH ADMIN OPTION 514
Combining CREATE USER and GRANT 519
Greg’s List has gone global! 520
Your SQL Toolbox 522
How about a Greg’s List in your city? 524
Use SQL on your own projects and you too could be like Greg! 524