


LEARNING MADE EASY

SQL

Essentials

for
dummies[®]

A Wiley Brand



Brush up on
core concepts

Jump-start your
skills expansion

Find solutions to
common problems

Richard Blum
Allen G. Taylor

SQL Essentials

**for
dummies**[®]
A Wiley Brand



SQL Essentials

by Richard Blum
and Allen G. Taylor

for
dummies[®]
A Wiley Brand

SQL Essentials For Dummies®

Published by: **John Wiley & Sons, Inc.**, 111 River Street, Hoboken, NJ 07030-5774, www.wiley.com

Copyright © 2025 by John Wiley & Sons, Inc. All rights reserved, including rights for text and data mining and training of artificial technologies or similar technologies.

Media and software compilation copyright © 2025 by John Wiley & Sons, Inc. All rights reserved, including rights for text and data mining and training of artificial technologies or similar technologies.

Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the Publisher. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Trademarks: Wiley, For Dummies, the Dummies Man logo, Dummies.com, Making Everything Easier, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002. For technical support, please visit <https://hub.wiley.com/community/support/dummies>.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2024948584

ISBN 978-1-394-29694-1 (pbk); ISBN 978-1-394-29696-5 (ebk); ISBN 978-1-394-29695-8 (ebk)

Contents at a Glance

Introduction	1
CHAPTER 1: Getting to Know SQL.....	5
CHAPTER 2: Creating a Database with SQL.....	13
CHAPTER 3: Drilling Down to the SQL Nitty-Gritty.....	27
CHAPTER 4: Values, Variables, Functions, and Expressions.....	51
CHAPTER 5: SELECT Statements and Modifying Clauses.....	79
CHAPTER 6: Querying Multiple Tables with Subqueries.....	109
CHAPTER 7: Querying Multiple Tables with Relational Operators.....	127
CHAPTER 8: Cursors.....	151
CHAPTER 9: Assigning Access Privileges.....	163
CHAPTER 10: Ten Retrieval Tips.....	175
Index	179

Table of Contents

INTRODUCTION	1
About This Book	1
Foolish Assumptions	2
Icons Used in This Book	3
Where to Go from Here	3
CHAPTER 1: Getting to Know SQL	5
Knowing What SQL Does	5
Knowing What SQL Does Not Do	6
Choosing and Using an Available RDBMS Implementation	7
Microsoft Access	7
Microsoft SQL Server	10
IBM DB2	11
Oracle Database	11
Sybase SQL Anywhere	11
MySQL	12
PostgreSQL	12
CHAPTER 2: Creating a Database with SQL	13
First Things First: Planning Your Database	13
Building Tables	14
Locating table rows with keys	17
Using the CREATE TABLE statement	17
Setting Constraints	20
Column constraints	20
Table constraints	20
Working with Keys and Indexes	20
Ensuring Data Validity with Domains	21
Establishing Relationships between Tables	21
Altering Table Structure	24
Deleting Tables	25
CHAPTER 3: Drilling Down to the SQL Nitty-Gritty	27
SQL's Data Types	27
Exact numerics	28
Approximate numerics	30

Character strings.....	30
Binary strings.....	32
Booleans	32
Datetimes.....	33
Intervals.....	34
XML	34
ROW	35
Collection	35
REF	36
JSON.....	36
User-defined.....	37
Handling Null Values.....	38
Applying Constraints.....	38
Column constraints	38
Table constraints.....	40
Foreign key constraints	42
Assertions	43
Operating on Data with the Data Manipulation Language	43
Retrieving data from a database.....	44
Adding data to a table	45
Updating data in a table.....	47
Deleting data from a table.....	49

CHAPTER 4: Values, Variables, Functions, and Expressions	51
Entering Data Values.....	51
Row values.....	52
Column references	52
Literal values.....	52
Variables.....	54
Special variables.....	55
Working with Functions	56
Summarizing data with set functions.....	56
Dissecting data with value functions.....	59
Using Expressions	69
Numeric value expressions	69
String value expressions	70
Datetime value expressions	71
Interval value expressions	71
Boolean value expressions.....	72

	Array value expressions	73
	Conditional value expressions	74
	Converting data types with a CAST expression.....	77
	Row value expressions.....	78
CHAPTER 5:	SELECT Statements and Modifying Clauses	79
	Finding Needles in Haystacks with the SELECT Statement	79
	Modifying Clauses	80
	FROM clauses	80
	WHERE clauses	82
	GROUP BY clauses	101
	HAVING clauses.....	104
	ORDER BY clauses.....	105
CHAPTER 6:	Querying Multiple Tables with Subqueries	109
	Introducing Subqueries	109
	Subqueries that return multiple values	110
	Subqueries that return a single value	112
	Quantified subqueries: Returning a single value.....	116
	Correlated subqueries.....	119
	Using Subqueries in INSERT, DELETE, and UPDATE Statements	125
CHAPTER 7:	Querying Multiple Tables with Relational Operators	127
	UNION.....	128
	UNION ALL.....	130
	UNION CORRESPONDING.....	131
	INTERSECT	132
	EXCEPT	133
	JOINS	134
	Cartesian product or cross join	134
	Equi-join	137
	Natural join	139
	Condition join	140
	Column-name join	140
	Inner join.....	142
	Outer join.....	143
	ON versus WHERE	147
	Join Conditions and Clustering Indexes.....	148

CHAPTER 8: Cursors	151
Declaring a Cursor	153
The query expression	153
Ordering the query result set	154
Updating table rows	156
Sensitive versus insensitive cursors	156
Scrolling a cursor	158
Holding a cursor	158
Declaring a result set cursor	159
Opening a Cursor	159
Operating on a Single Row	160
FETCH syntax	160
Absolute versus relative fetches	161
Deleting a row	162
Updating a row	162
Closing a Cursor	162
CHAPTER 9: Assigning Access Privileges	163
Working with the SQL Data Control Language	163
Identifying Authorized Users	164
Understanding user identifiers	164
Getting familiar with roles	164
Classifying Users	166
Granting Privileges	167
Looking at data	168
Deleting data	168
Adding data	169
Changing data	169
Using certain database facilities	169
Responding to an event	170
Defining new data types	170
Executing an SQL statement	170
Doing it all	170
Passing on the power	171
Revoking Privileges	171
Granting Roles	173
Revoking Roles	174

CHAPTER 10: Ten Retrieval Tips..... 175

- Verify the Database Structure..... 175
- Try Queries on a Test Database 176
- Double-Check Queries That Include Joins 176
- Triple-Check Queries with Subselects..... 176
- Summarize Data with GROUP BY 176
- Watch GROUP BY Clause Restrictions 177
- Use Parentheses with AND, OR, and NOT..... 177
- Control Retrieval Privileges 178
- Back Up Your Databases Regularly 178
- Handle Error Conditions Gracefully 178

INDEX..... 179

Introduction

SQL is the internationally recognized standard language for dealing with data in relational databases. Developed by IBM, SQL became an international standard in 1986. The standard was updated in 1989, 1992, 1999, 2003, 2008, 2011, 2016, and 2023. It continues to evolve and gain capability. Database vendors continually update their products to incorporate the new features of the ISO/IEC standard. (For the curious out there, ISO is the International Organization for Standardization, and IEC is the International Electrotechnical Commission.)

SQL isn't a general-purpose language, such as C++ or Java. Instead, it's strictly designed to deal with data in relational databases. With SQL, you can carry out all the following tasks:

- » Create a database, including all tables and relationships.
- » Fill database tables with data.
- » Change the data in database tables.
- » Delete data from database tables.
- » Retrieve specific information from database tables.
- » Grant and revoke access to database tables.
- » Protect database tables from corruption due to access conflicts or user mistakes.

About This Book

In this book, we cover how to use SQL to build and work with databases. Using SQL isn't just about knowing the terms and keywords, it's also about knowing when and how to use them to maximize performance of your application.

Here are some of the things you can do with this book:

- » Find out about the capabilities and limitations of SQL.
- » Discover how to develop reliable and maintainable database systems.

- » Create databases.
- » Speed database queries.
- » Control access to sensitive information.

Within this book, you may note that some web addresses break across two lines of text. If you're reading this book in print and want to visit one of these web pages, simply key in the web address exactly as it's noted in the text, pretending as though the line break doesn't exist. If you're reading this as an e-book, you've got it easy — just click the web address to be taken directly to the web page.

Foolish Assumptions

We know that this is a *For Dummies* book, but we don't really expect that you're a dummy. In fact, we assume that you're a very smart person. After all, you decided to read this book, which is a sign of high intelligence indeed. Therefore, we assume that you may want to do a few things, such as re-create some of the examples in the book. You may even want to enter some SQL code and execute it. To do that, you need at the very least an SQL editor and more likely also a relational database management system (RDBMS) of some sort. Many choices are available, both proprietary and open source. We mention several of these products at various places throughout the book but don't recommend any one in particular. Any product that complies with the ISO/IEC international SQL standard should be fine.

That said, take claims of ISO/IEC compliance with a grain of salt. No RDBMS available today is 100 percent compliant with the ISO/IEC SQL standard. For that reason, some of the code examples we give in this book may not work in the particular SQL implementation that you're using. The code samples we use in this book are consistent with the international standard rather than with the syntax of any particular implementation, unless we specifically state that the code is for a particular implementation.

Icons Used in This Book

For *Dummies* books are known for those helpful icons that point you in the direction of really great information. This section briefly describes the icons used in this book.



TIP

The Tip icon points out helpful information that's likely to make your job easier.



REMEMBER

This icon marks a generally interesting and useful fact — something that you may want to remember for later use.



WARNING

The Warning icon highlights lurking danger. When you see this icon, pay attention, and proceed with caution.

Where to Go from Here

If you're brand-new to the database world, start out in Chapter 1. It explains why databases are useful, and walks through a few of the different popular database software packages available.

If you're already familiar with database software packages, and you just want to dive into the nuts and bolts of working with databases, Chapters 2 through 5 cover all the things you'll need to know to get a database up and running.

If you're already an old hand at SQL and you want to dive into some more advanced topics, Chapters 6 through 8 cover the complicated world of database queries. There are plenty of ways to retrieve data from a database, but not all of them are efficient — especially if you have lots of data to work with!

Chapter 9 examines the dreaded security topic. If you're just creating a database for your own use, feel free to skip this chapter, but if you work in an environment where lots of people are going to need access to your database, this chapter is a must. Knowing how to protect your data from prying eyes has become a hot topic these days, and it's important to know just how to protect it.

Finally, Chapter 10 provides ten tips for improving your database experience when using SQL to retrieve data.

- » Understanding what SQL does
- » Getting clear on what SQL doesn't do
- » Weighing your SQL implementation options

Chapter 1

Getting to Know SQL

In the early days of the relational database management system (RDBMS), there was no standard language for performing relational operations on data. A number of companies came out with RDBMS products, and each had its own associated language. However, differences in syntax and functionality made it impossible for a person using the language of one RDBMS to operate on data that had been stored by another. The creation of SQL solved this problem, but SQL is a continually evolving language that changes with each official release (the most recent being in 2023). This chapter explores just what SQL is (and isn't). It also takes a look at using SQL in some different database packages.

Knowing What SQL Does

SQL (pronounced *ess cue el*) is a software tool designed to deal with relational database data. It does far more than just execute queries. Yes, you can use it to retrieve the data you want from a database using a query. But you can also use SQL to create and destroy databases, as well as modify their structure. In addition, you can add, modify, and delete data with SQL. Even with all that capability, SQL is still considered only a *data sublanguage*, which means that it doesn't have all the features of general-purpose programming languages such as C, C++, C#, or Java.

SQL is specifically designed for dealing with relational databases, so it doesn't include a number of features needed for creating useful application programs. As a result, to create a complete application — one that handles queries, as well as provides access to a database — you have to write the code in one of the general-purpose languages and embed SQL statements within the program whenever it communicates with the database.

Knowing What SQL Does Not Do

Before we can tell you what SQL doesn't do, we need to give you some background information. In the 1930s, computer scientist and mathematician Alan Turing defined a very simple machine that could perform any computation that could be performed by any computer imaginable, regardless of how big and complex. This simple machine has come to be known as a *universal Turing machine*. Any computer that can be shown to be equivalent to a universal Turing machine is said to be *Turing-complete*. All modern computers are Turing-complete. Similarly, a computer language capable of expressing any possible computation is said to be Turing-complete. Practically all popular languages, including C, C#, C++, BASIC, FORTRAN, COBOL, Pascal, Java, and many others, are Turing-complete. SQL, however, is not.

Because standard SQL is not Turing-complete, you can't write an SQL program to perform a complex series of steps the way you can with a language such as C or Java. On the other hand, languages such as C and Java don't have the data-manipulation capabilities that SQL has, so you can't write a program with them that will efficiently operate on database data. There are several ways to solve this dilemma:

- » Combine the two types of language by embedding SQL statements within a program written in a host language such as C.
- » Have the C program make calls to SQL modules to perform data-manipulation functions.
- » Create a new language that includes SQL, but also incorporates those structures that would make the language Turing-complete. (This is essentially what Microsoft and Oracle have done with their versions of SQL.)

All three of these solutions are offered by various vendors.

Choosing and Using an Available RDBMS Implementation

SQL by itself isn't all that useful — you need a platform that stores the data itself and uses SQL to create, read, update, and delete (often called CRUD) the data. This is where the RDBMS comes in.

The RDBMS is a program that stores data in a manner that makes it easy to retrieve the data as quickly as possible. Storing data in a typical file isn't efficient, because in order to find a specific data item, the program would have to read through the entire file until it got to that data.

An RDBMS system uses various methods to store and index data so it can quickly find a specific data record, based on the SQL statement it's processing. There are plenty of different RDBMS programs available these days, each with different features to help increase data retrieval performance. In the following sections, we fill you in on some of the more common RDBMS programs available today.

Microsoft Access

Microsoft Access is an entry-level RDBMS with which developers can build relatively small and simple databases and database applications. It's designed for use by people with little or no training in database theory. You can build databases and database applications using Access, without ever seeing SQL. However, you can opt to use SQL in Access if you so choose.

Access runs under any of the Microsoft Windows operating systems, as well as Apple's macOS, but not under Linux or any other non-Microsoft operating system.

To reach the SQL editor in Access, do the following:

- 1. Open a database that already has tables and at least one query defined.**

A great place to start is with the Northwind Traders Starter Edition database provided as a free download with Access. The database includes a built-in mini-application that uses Access forms to help query and insert data. After you