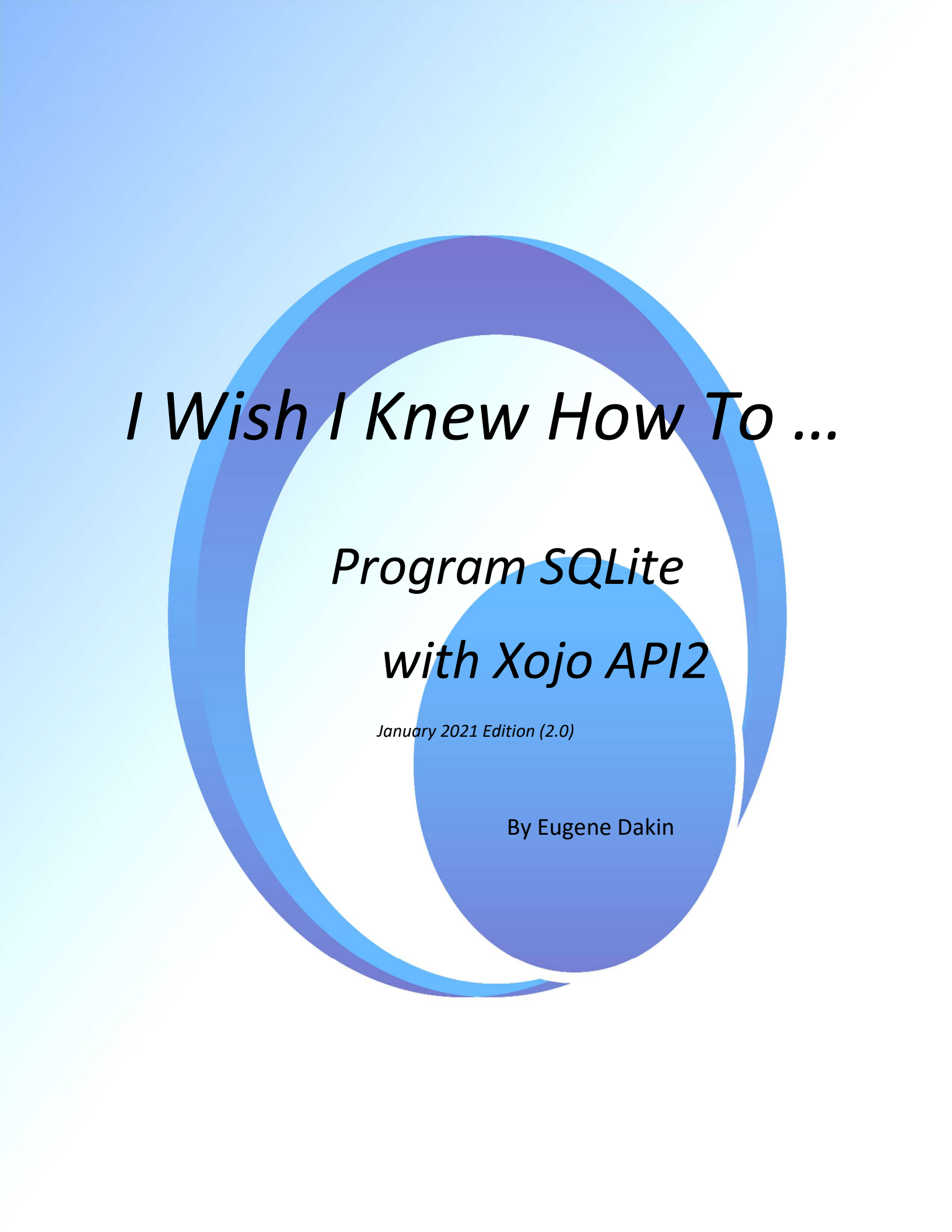




I Wish I Knew How To ...

*Program SQLite
with Xojo API2*

January 2021 Edition (2.0)

By Eugene Dakin

Table of Contents

Chapter 1 - Introduction to Xojo and the Environment	9
Preview of Chapter 1	9
What is Xojo?.....	9
Describing and Defining the User Interface	10
Setting the run-time properties	13
Chapter 2 - Database Fundamentals (SQLite Browser)	16
Database Terms and Structure.....	17
SQLite Database Browser	19
Open a Database	20
Create a New Database	25
Adding Data Manually	31
Editing The Database.....	34
Chapter 3 – Desktop Connection to a SQLite File	38
Front End and Back End	38
Connect to an SQLite Database.....	39
SQL Fundamentals.....	42
Create a New Database with Code	43
New Database Table.....	45
Chapter 4 – Getting Data from the SQLite Database File	47
Getting Database Information into TextColumns.....	48
Get Second Record.....	51
Populating a ComboBox	54
Putting Data into a Listbox	57
Sorting Data	60
Finding Data	64
Filtering Numerical Data	68
User-Based Searching.....	72
Plural Searches	75
Excluding Filtered Data.....	78
Wildcard Filtering.....	82
Chapter 5 – Editing Data on the SQLite Database File.....	86
Updating a Single Item in a Column.....	86
Updating Multiple Items in Columns	92
User-Based Updating.....	97
Multiple User-Based Updates	103
Updating Multiple Records	109

Chapter 6 – Adding Data to the SQLite Database File	114
Add a New Row with Data in a Table	114
Inserting User-based Data	119
Inserting RowSet Data	124
Chapter 7 – Building a Menu	130
Chapter 8 – File Handling of a SQLite Database	132
Opening a File	132
Copying/Backing up a file	138
Chapter 9 – Value of Columns	143
Counting Records in a Database	143
Maximum Column Value	146
Minimum Column Value	149
Average Column Value	151
Sum Column Value	154
Chapter 10 – Calculations with the Database	157
Multiplication	157
Addition	165
Chapter 11 – Formatting Numbers	172
Chapter 12 – Printing a Report	177
Creating a Report with Xojo's Report Builder	177
Vertical and Horizontal Resolution	182
Adding a Picture or Logo	183
Changing Report Picture at Runtime	188
Add a Date	193
Date Formats	197
Custom Date Formats	198
Page Number	203
Format Page Number Text	207
Formatting Page Number	211
Page Number Example A	211
Page Number Example B	215
Page Number Total –Not working	220
Page Setup	224
Portrait and Landscape	224
Page Margins	228
Print Without Printer Dialogue	235
Report Addition	238
Grouping Data	243

Grouping Descending Data	248
Subgrouping Ascending Data.....	252
Add A Single Column	256
Font Properties.....	260
Font Code	261
Label Behaviours	267
Label Behaviours with Code	272
Column Format - Code	275
Print Without Printer Dialogue – Version 2	280
 Chapter 13 Working with Tables	 283
Create New Table Only.....	283
Create New Table with Columns.....	285
Add Table Column	288
Delete Table Column - Workaround	290
Table Rename.....	297
 Chapter 14 – Deleting Data.....	 300
Deleting a Record	300
Delete All Records in a Table.....	303
 Chapter 15 – Threads and Progress Bars.....	 306
Creating a Thread	306
Adding a ProgressBar Thread	309
Database ProgressBar Thread	312
 Chapter 16 – Database Encryption	 317
Create Encrypted Database.....	317
Open Encrypted Database	322
 Chapter 17 - Joins.....	 327
Understanding Joins.....	327
The Concept	327
Inner Join	329
Full Outer Join (Not Supported by SQLite).....	332
Left Join	335
Right Join (Not Supported by SQLite).....	339
Left Outer Join.....	342
Right Outer Join (Not Supported by SQLite)	346
 Chapter 18 – Deleting Tables.....	 350
 Chapter 19 – Handling Apostrophes.....	 353
Identify Apostrophe Issue	353

Making Apostrophe's Work	358
Chapter 20 – Alias	360
Column Alias/Column.....	360
Table Alias	364
Chapter 21 – Transactions	368
Chapter 22 – Prepared Statements	375
SQL Injection Attack	375
Select Prepared Statement	376
Update Prepared Statement	382
Delete Prepared Statement	387
Insert Prepared Statement.....	390
Chapter 23 – Reserved Words	395
Chapter 24 – Specifications	396
Chapter 25 - Other	397
Periodic Maintenance	397
Multiuser	397
Index.....	403

Inserting RowSet Data

User-based data is entered into the database by modifying inserting text Column data into the SQL command. The following screenshot is of the program before the push button has been pressed. Data has been manually typed in for the Author, Inventory, Title, and Cost in the Text Columns. This example uses data addition with a RowSet instead of an SQL command.

Figure 65. Screengrab of Example 6-3 Prior to New Data

The screenshot shows a window titled "Insert User Data: Example: 6-3" with a menu bar containing "File" and "Edit". Inside the window, there is a table with 5 rows and 4 columns: Author, Inventory, Title, and Cost. Below the table, there is a section labeled "Data" with four input fields: Author (Eugene Dakin), Inventory (5), Title (SQLite for Xojo), and Cost (8.99). At the bottom of the window is a button labeled "Add Recordset Data to the database".

	Author	Inventory	Title	Cost
1	A. Brown	3	Woodworking Made Easy	23
2	B. Smith	2	Databases made Simple	34.5
3	C. Green	4	Chemistry in Paint	67
4	D. Gray	8	How to make Paper	56
5	E. Atkins	2	How to fix Cars	99

Data

Author: Eugene Dakin

Inventory: 5

Title: SQLite for Xojo

Cost: 8.99

Add Recordset Data to the database

Data from the Insert.sqlite database is shown in the listbox and 5 entries are shown. There is data in the textColumns, such as Eugene Dakin for a new author, and the pushbutton has not yet been pressed. Once pressed, the following screen shot adds the data and updates the listbox from the database.

Figure 66. Example 6-3 with Data

	Author	Inventory	Title	Cost
1	A. Brown	3	Woodworking Made Easy	23
2	B. Smith	2	Databases made Simple	34.5
3	C. Green	4	Chemistry in Paint	67
4	D. Gray	8	How to make Paper	56
5	E. Atkins	2	How to fix Cars	99
6	Eugene Da...	5	SQLite for Xojo	8.99

Data

Author:

Inventory:

Title:

Cost:

When the program runs for the first time, the `PushButton1.Open` method is called. Within this method are the following calls to two other methods.

```
Sub Open() Handles Open
    Call ChooseDatabase
    Call RefreshListbox
End Sub
```

The `ChooseDatabase` method contains code for the user to choose the Database name and get the location of the database file. A global property is created called `db`, which is a SQLite database type.

Code 38. Example 6-3: ChooseDatabase Method

```
Public Sub ChooseDatabase()
    //Get the file name
    Var dlg as new OpenFileDialog
    dlg.Title = "Open *.sqlite File"
    f = dlg.ShowDialog()

    //Connect to a SQLite database
```

```
db = new SQLiteDatabase
db.DatabaseFile = f

If f <> Nil Then
  Try
    db.Connect
    //The connection was successful!
  Catch err as DatabaseException //It did not connect
    MsgBox.Show "An error occured connecting to the database" + EndOfLine +
err.Message
    return
  End Try
End If

Exception err as OLEException //Let the user know of errors
  MsgBox.Show err.Message
End Sub
```

A folderitem variable called f is created to hold the file name and to remember the directory where the file exists. An OpenFileDialog is added to prompt the user for the database filename and to get the location directory of the database. When the dialogue is open, the title at the top of the window shows 'Open *.sqlite File'. This provides a strong hint as to what action the user is to perform and the file extension name to open the Inventory SQLite database. A new instance of SQLite database variable called db is created. The filename and location of the file stored in the variable f is loaded into the databasefile.

If there is a database connection, meaning there are no errors and the database information was successfully read by the program, then tell the database connection was successful (db.Connect = True). If something caused an error, then let the user know about the error and also the error message. To make sure that all errors are shown, an Exception err is created to trap and show the error message.

Now that the database has been chosen, and the program has only asked the user once for the database location, the next step is to show the data in the listbox, or RefreshListbox.

Code 39. Example 6-3: RefreshListbox

```
Public Sub RefreshListbox()  
    //Format the Listbox  
    Listbox1.HasHeader = true //show the heading  
    Listbox1.ColumnCount = 5 //make 5 columns  
    Listbox1.ColumnWidths = "5%, 20%, 20%, 40%, 15%"  
    Listbox1.InitialValue = "ID" + chr(9) + "Author" + chr(9) + "Inventory" + chr(9) + "Title" +  
    chr(9) + "Cost"  
  
    //Get the data from the database  
    Try  
        Var rs as RowSet  
        rs = db.SelectSQL("SELECT * FROM Library")  
        If rs <> Nil Then  
            For Each row as DatabaseRow in rs  
                Listbox1.AddRow  
                Listbox1.CellValueAt(Listbox1.LastAddedRowIndex, 0) =  
Cstr(rs.Column("ID").IntegerValue)  
                Listbox1.CellValueAt(Listbox1.LastAddedRowIndex, 1) = rs.Column("Author").StringValue  
                Listbox1.CellValueAt(Listbox1.LastAddedRowIndex, 2) =  
Cstr(rs.Column("Inventory").DoubleValue)  
                Listbox1.CellValueAt(Listbox1.LastAddedRowIndex, 3) = rs.Column("Title").StringValue  
                Listbox1.CellValueAt(Listbox1.LastAddedRowIndex, 4) =  
Cstr(rs.Column("Cost").DoubleValue)  
            Next  
            rs.Close  
        End If  
        rs = Nil  
    Catch error As DatabaseException  
        MessageBox("Error: " + error.Message)  
    End Try  
End Sub
```

The listbox is formatted by showing the heading, making 5 columns, showing the column width in percentage values and adding the heading labels. Each label is separated by a tab that is

shown as chr(9). A RowSet variable is created with the variable name rs. A SelectSQL command is used and the results are to be placed in the rs variable. This is the step where SQL is implemented. The SELECT command means 'get' information from the database (as opposed to updating or adding new data). The wildcard character (*) means select everything in the database.

Once data has been retrieved from the database, the data then needs to be placed on the screen so the user can see the data. To prevent an error, the rs RowSet is checked to see if there is data (rs <> Nil). If there is data, then each Column and its data is placed in the appropriate TextColumn. There is more than one record in this RowSet, and a do-until loop is used to start from the first RowSet and stop at the last RowSet. The IntegerValue 'ID' is placed in the CellValueAt (0) of the new row (addrow). The column to the right has a value of 1, and contains the Author StringValue. Each Column from the database populates the listbox until the end of the database file has been reached. If there is no data in the RowSet, then a messagebox shows that there is no data and to gracefully prevent an error from happening. An Exception error bit of code has been added to capture any other errors.

Code 40. Pushbutton Update Code for Example 6-3

```
Sub Action() Handles Action
    Try
        //Insert data in the database
        Var row as new DatabaseRow
        row.Column("Author") = TFAuthor.Text
        row.Column("Inventory") = TFInventory.Text
        row.Column("Title") = TFTitle.Text
        row.Column("Cost") = TFCost.Text

        db.AddRow("Library", row)

    Catch error As DatabaseException
        MessageBox("Error: " + error.Message)
    End Try

    call RefreshListbox
End Sub
```

Adding data to a database by a RowSet requires a few more lines of code and is a little easier to prevent mistakes. A new DatabaseRow variable is created called *row* and each column in the SQLite database is placed in quotation marks and the text (string) value is placed in the RowSet. Once all data has been entered in the RowSet then the *AddRow* adds a row to the table *Library*. If there is an error, error trapping shows some information to try and help the programmer track down the error. The listbox data is refreshed by calling the RefreshListbox method.

This example updates a SQLite database with a database record.

Index

-, 157

*, 157

/, 157

+, 157

ACID, 16

Add, 157

Add Column, 28, 288

Add Manual Data, 31

Add New Row, 114

Add Record, 31, 124

Add Recordset Data, 124

Add Table, 27

Add User Based Data, 119

Addition, 165

Alias, 360

Column, 360

Field, 360

Table, 364

ALTER TABLE, 42

ANALYZE, 397

Apostrophe, 53

Average Value, 151

Back End, 38

Backup a File, 138

BindType

SQLITE_BLOB, 381

SQLITE_BOOLEAN, 381

SQLITE_DOUBLE, 381

SQLITE_INT64, 381

SQLITE_INTEGER, 381

SQLITE_NULL, 381

SQLITE_TEXT, 381

Calculations, 157

Column Add, 288

Column Delete, 290

Column Add, 28

Connect to SQLite, 39

Connection, 38

Copying a File, 138

Count, 143

Create a Report, 177

Create a Thread, 306

CREATE DATABASE, 42

Create Encrypted Database, 317

Create Encryption, 317

CREATE INDEX, 42

Create New Database, 25

CREATE TABLE, 42

CreateDatabase, 44

CRUD, 19

Data Manual Add, 31

Database

 Definition, 17

Database Edit, 34

Database Encryption, 317

Database New, 25

Database ProgressBar Thread, 312

Database Specifications, 396

Database Terms, 17

Date

 Custom Date Formats, 198

 Default Date Formats, 197

Delete

 All Records, 303

 One Record, 300

DELETE, 42

Delete Column, 290

Delete Data, 300

Delete Prepared Statement, 387

Deleting Tables, 350

Dialogue Box

 Labels, 135

 Open, 133

Divide, 157

DROP INDEX, 42

DROP TABLE, 42

Edit Database, 34

Employees.sqlite, 17

Encryption, 317

Examples

 01-01 REALStudio Program, 15

 03-01 SQLite Connection, 40

 03-02 New SQLite database code, 44

 03-03 New table and field, 46

 04-01 Populating Textfields, 49

 04-02 SELECT a Specific Recordset, 51

 04-03 Populating a Combobox, 54

 04-04 Populating a Listbox, 57

 04-05 Database Sorting, 62

 04-06 Finding Data, 65

 04-07 Filtering Numerical Data, 69

 04-08 User Based Searching, 72

 04-09 Plural Searching, 75

 04-10 Excluding Data, 80

 04-11 Wildcard Searching, 83

 05-01 Pushbutton Update, 91

 05-01 RefreshListbox, 89

 05-02 Pushbutton Update, 96

 05-02 RefreshListbox, 94

 05-03 Pushbutton Update, 101

 05-03 RefreshListbox, 100

 05-04 Pushbutton Update, 107

 05-04 RefreshListbox, 106

 05-05 Pushbutton Update, 113

 05-05 RefreshListbox, 111

 06-01 Pushbutton Update, 118

 06-01 RefreshListbox, 117

 06-02 Pushbutton Update, 123

 06-02 RefreshListbox, 122

 06-03 Insert Recordset, 128

 08-01 Displaying Data from a Database,
 135

 08-01 Opening a SQLite Database, 134

- 08-02 Copying or Backing Up a Database, 139
- 09-01 Counting Records, 144
- 09-02 Maximum Field Value, 146
- 09-03 Minimum Field Value, 149
- 09-04 Average Field Value, 151
- 09-05 Sum Field Value, 154
- 10-01 Multiplication Action, 161
- 10-01 Multiplication RefreshListbox, 159
- 10-02 RefreshListbox, 166
- 11-01 Formatting Database Numbers, 174
- 12-01 Gathering Recordset Data, 179
- 12-01 Print Report Method, 181
- 12-02 Gathering Recordset Data, 185
- 12-02 PrintMyReport Method, 187
- 12-03 Change Logo Picture at Runtime, 188
- 12-03 Gathering Recordset Data, 189
- 12-03 PrintMyReport Method, 191
- 12-04 Add a Date Recordset Data, 193
- 12-04 PrintMyReport Method, 195
- 12-05 Gathering RecordSet Data, 200
- 12-05 PrintMyReport Method, 202
- 12-06 Page Number RecordSet Data, 204
- 12-06 PrintMyReport Method, 206
- 12-07 Page Number RecordSet Data, 208
- 12-07 PrintMyReport Method, 210
- 12-08 Page Number Recordset Data, 212
- 12-08 PrintMyReport Method, 214
- 12-09 Gather Recordset Data, 216
- 12-09 PrintMyReport Method, 218
- 12-09 Single Label Page Number, 215
- 12-10 Gather Recordset Data, 221
- 12-10 PrintMyReport Method, 223
- 12-10 Total Page Numbers, 220
- 12-11 Gather Recordset Data, 225
- 12-11 Page Orientation, 224
- 12-11 PrintMyReport Method, 227
- 12-12 Page Margins Recordset Data, 231
- 12-12 PrintMyReport Method, 233
- 12-13 No Printer Dialogue Recordset Data, 235
- 12-13 PrintMyReport Method, 237
- 12-14 PrintMyReport Method, 241
- 12-14 Sub Total Recordset Data, 239
- 12-15 Group PrintMyReport Method, 247
- 12-15 Group Recordset Data, 245
- 12-15 SQL Command, 243
- 12-16 Descending Recordset Data, 249
- 12-16 Descending SQL Command, 248
- 12-16 Group PrintMyReport Method, 251
- 12-17 Main and Sub Category Ordering, 252
- 12-17 PrintMyReport Method, 255
- 12-17 Subgroup Ascending Recordset Data, 253
- 12-18 Add A Column, 258
- 12-18 New SQL Command Snippet, 256
- 12-18 Old SQL Command Snippet, 256
- 12-19 Font Properties, 260
- 12-20 Change Text Bold Setting Snippet, 262
- 12-20 Change Text Font Setting Snippet, 261
- 12-20 Change Text Italic Setting Snippet, 262
- 12-20 Change Text Size Setting Snippet, 262
- 12-20 Change Text Underline Setting Snippet, 262
- 12-20 PrintMyReport Method, 265
- 12-20 Show Text Font Setting Snippet, 261
- 12-20 Show Text Size Setting Snippet, 261
- 12-21 Report Layout Behaviour, 268
- 12-22 Colour Alpha Channel Type, 274
- 12-22 Horizontal Alignment Code Snippet, 272
- 12-22 Text Backcolour Code Snippet, 274
- 12-22 Text Colour Code Snippet, 274
- 12-22 Vertical Alignment Code Snippet, 273
- 12-22 Word Wrapping Code Snippet, 275
- 12-23 Field Formatting Code Snippet, 276
- 12-23 Field Formatting Recordset Data, 277

- 12-23 General Formatting Code, 275
- 12-23 PrintMyReport Method, 279
- 12-24 No Printer Dialogue Recordset Data, 280
- 12-24 PrintMyReport Method, 282
- 13-01 Adding a New Table, 283
- 13-02 Create New Table with Fields, 286
- 13-03 Add a New Field, 289
- 13-04 Delete Field Workaround, 295
- 13-05 Rename Table, 298
- 14-01 Deleting a Record, 301
- 14-02 Deleting All Records, 303
- 15-01 PushButton Action Event, 308
- 15-01 Thread Run Event, 307
- 15-01 Timer Action Event, 308
- 15-02 PushButton Action Event, 311
- 15-02 Thread Run Event, 310
- 15-02 Timer Action Event, 310
- 15-03 PushButton Action Event, 314
- 15-03 PushButton Open Event, 315
- 15-03 Thread1 Run Event, 312
- 15-03 Timer Action Event, 313
- 16-01 Encrypted DB, 319
- 16-02 Open Encrypted DB, 323
- 17-01 Inner Join, 330
- 17-02 Full Outer Join, 333
- 17-03 Left Join, 336
- 17-04 Right Join, 340
- 17-05 Left Outer Join, 343
- 17-06 Right Outer Join, 347
- 18-01 Deleting a Table, 351
- 19-01 Apostrophe Error, 353
- 19-02 Apostrophe Safe SQL, 359
- 20-01 Column Alias, 361
- 20-02 Table Alias, 364
- 21-01 Choose Database, 370
- 21-01 Pushbutton Action, 373
- 21-01 RefreshListbox, 371
- 22-01 SQL Prepared Statement, 377
- 22-02 Choose Database Method, 383
- 22-02 Pushbutton Update, 386
- 22-02 RefreshListbox, 384
- 22-03 Prepared Statement Delete Record, 388
- 22-04 ChooseDatabase Method, 391
- 22-04 Pushbutton Insert, 394
- 22-04 RefreshListbox, 392
- 25-02 Multiuser, 399
- Other-1 Analyze Maintenance, 397
- Excluding Filtered Data, 78
- Field Add, 288
- Field Delete, 290
- Field Name Spaces, 16
- Field Type
 - Blob, 285
 - Integer PRIMARY KEY, 285
 - Numeric, 285
 - Text, 285
- Field values, 143
- File
 - Adding Data, 114
 - Average Field Value, 151
 - Building a Menu, 130
 - Count Records, 143
 - Create New Database, 38
 - Editing Data, 86
 - Get Database, 47
 - Maximum Field Value, 146
 - Minimum Field Value, 149
 - Sum Field Value, 154
- File Handling, 132
- Finding Data, 64
- Finding Numerical Data, 68
- Flat Database, 17, 18
- Formatting Numbers, 172

Front End, 38

Get Database Data, 47

Handling Apostrophes, 353

INSERT INTO, 42

INTEGER, 23

Joins, 327

- Full Outer Join, 332
- Inner Join, 329
- Left Join, 335
- Left Outer Join, 342
- Right Join, 339
- Right Outer Join, 346
- The Concept, 327
- Understanding, 327

Maintenance, 397

Marker

- VVV, 380
- \$VVV, 380
- ?, 380
- ?NNN, 380
- @VVV, 380

Maximum Value, 146

Minimum Value, 149

Multiple User Based Updates, 103

Multiplication, 157

Multiply, 157

Multiuser, 397

New Database, 25

NONE, 23

NUMERIC, 23

Open Encrypted DB, 322

Opening a File, 132

Page Orientation, 224

Page Orientation Values, 225

Plural Searches, 75

Populating

- Textfields, 48

Populating a Combobox, 54

Populating a Listbox, 57

Prepared SQL Statement

- Compiled SQL Statement, 375
- DELETE, 387
- Insert, 390
- SELECT, 376
- SQLITE_BLOB, 381
- SQLITE_BOOLEAN, 381
- SQLITE_DOUBLE, 381
- SQLITE_INT64, 381
- SQLITE_INTEGER, 381
- SQLITE_NULL, 381
- SQLITE_TEXT, 381
- UPDATE, 382

Prepared Statements, 375

Primay Key, 34

Print a Report, 177

Progress Bar, 306

Progress Bar Thread, 309

ProgressBar Database Thread, 312

REAL, 23

Record Add, 31

Records, 17

 Delete all Records, 303

 Delete one Record, 300

Relation, 17

Relational database, 17

Rename Table, 297

Report, 177

 A Simple Report, 177

 Add a Date, 193

 Add a Logo, 183

 Add a Picture, 183

 Add A Single Column, 256

 Addition, 238

 Change Picture at Runtime, 188

 Characters for Custom Formatting, 275

 Custom Date Formats, 198

 Date Formats, 197

 Field Format Code, 275

 Font Code, 261

 Font Properties, 260

 Format a Page Number, 211

 Grouping Data, 243

 Grouping Descending Data, 248

 Label Behaviours, 267, 272

 Landscape, 224

 Page Margins, 228

 Page Number, 203, 207

 Page Number A, 211, 215

 Page Setup, 224

 Portrait, 224

 Resolution Settings, 182

 Resolutions, 182

 Subgrouping Ascending Data, 252

 Text Colour Choices, 268

 Text Horizontal Alignment Type, 273

 Text Horizontal Choices, 267

 Text Vertical Alignment Type, 273

 Text Vertical Choices, 267

 Total Page Numbers, 220

 Without Printer Dialogue, 235, 280

Reserved Words, 395

Select, 53, 353

SELECT, 42

Select Prepared Statement, 376

Self-Contained, 16

Serverless, 16

Sorting Data, 60

Spaces, 16

Spaces Field Name, 16

Specification, 396

SQL

 ALTER TABLE, 42, 290

 apostrophe, 53

 CREATE DATABASE, 42

 CREATE INDEX, 42

 CREATE TABLE, 42, 291

 DELETE, 42

 DROP, 293

 DROP INDEX, 42

 DROP TABLE, 42

 INSERT INTO, 42, 292, 294

 ORDER BY, 61

 SELECT, 42, 292

 SQL Fundamentals, 42

 UPDATE, 42

SQL Injection Attack, 375

SQLite

INTEGER, 23
NUMERIC, 23
Open, 20
REAL, 23
TEXT, 23
SQLite Database Browser, 19
SQLite Database Open, 20
SQLite Field Types, 285
Subtract, 157
Sum Field Value, 154
Table
 Create a New Table, 283
 Create with Fields, 285
Table Add, 27
Table Rename, 297
TEXT, 23
Thread Creation, 306
Thread Database ProgressBar, 312
Thread Progress Bar, 309
Threads, 306
Transactional, 16
Transactions, 368
UPDATE, 42
Update Prepared Statement, 382
Updating a Single Field, 86
Updating Multiple Fields, 92
Updating Multiple Records, 109
User Based Searching, 72
User Based Updating, 97
Value of Fields, 143
WAL, 397
Wildcard Filtering, 82
Working with Tables, 283
Write-Ahead-Logging, 397
Zero-Configuration, 16

The 'I Wish I Knew' series contains technical data and advice that makes sense and contains practical and numerous examples with explanations to allow you to ease into the steep programming curve. You can create Desktop SQLite database applications today!

This book "I Wish I Knew How to ... Program SQLite with Xojo API2" delves into the mystery of programming a back-end database. SQLite is a zero-configuration, serverless, transactional, and self-contained database.

The book is written as a guide and reference to Xojo programmers who program Desktop Applications in Windows and Mac. There are no dynamic link libraries (dll), COM, or Active X parts to add. Although the report maker has been explained and can make reports, a workaround is to set the default print margin properties to zero.

There are more than 20 chapters and over 400 pages with greater than 80 example programs.

Examples include opening recordsets, finding data, sorting, updating data, counting records, printing a report, deleting records, creating new records, creating reports, and more. Many screenshots have been added to show the results of the code with an index to help find topics quickly.

This is one of many books at Xojo Library. This book can be purchased online at <http://www.xdevlibrary.com/> where many great Xojo and Real Studio resources are available.

Happy programming!

Eugene

Eugene Dakin MBA, Ph.D., P.Chem., is an author of Xojo and Real Studio reference materials and has many years of experience in the programming industry.

ISBN: 978-1-927924-29-7