I Wish I Knew How To …

Program SQLite

with Xojo Desktop

March 2015 Edition (1.2)

By Eugene Dakin

New to Version 1.0 – January 2014

This version has the following material added:

1) Wrote the book!

New to Version 1.1 – February 2014

This version has the following material added:

1) Prepared Statements
    a. Select
    b. Update
    c. Delete
    d. Insert
2) SQL Injection
3) SQLite Reserved Words
4) Delete Field Workaround
5) Add Table field
6) Rename existing Table

New to Version 1.2 – March 2015

This version has the following material added:

1) Maintenance (Analyze) Other-1
2) Added more items to book index (SQL: INSERT INTO, etc.)
3) Add data with a recordset (Example 6-3)
4) Updated saved examples to Xojo 2015 r1 format
5) Added multiuser (WAL) example (Example 25-2)

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 16
SQLite Database Browser 18
Open a Database 19
Create a New Database 22
Adding Data Manually 25
Editing The Database 27

Chapter 3 – Desktop Connection to a SQLite File 29
Front End and Back End 29
Connect to a SQLite Database 29
SQL Fundamentals 31

Chapter 4 – Getting Data from the SQLite Database File 32
Getting Database Information into TextFields 33
Populating a ComboBox 38
Putting Data into a Listbox 41
Sorting Data 44
Finding Data 48
Filtering Numerical Data 52
User-Based Searching 56
Plural Searches 59
Excluding Filtered Data 61
Wildcard Filtering 65

Chapter 5 – Editing Data on the SQLite Database File 69
Updating a Single Item in a Field 69
Updating Multiple Items in Fields 74
User-Based Updating 78
Multiple User-Based Updates 83
Updating Multiple Records 88

Chapter 6 – Adding Data to the SQLite Database File 93
  Add a New Row with Data in a Table 93
  Inserting User-based Data 98
  Inserting Recordset Data 103

Chapter 7 – Building a Menu 108

Chapter 8 – File Handling of a SQLite Database 110
  Opening a File 110
  Copying/Backing up a file 116

Chapter 9 – Value of Fields 121
  Counting Records in a Database 121
  Maximum Field Value 124
  Minimum Field Value 127
  Average Field Value 129
  Sum Field Value 132

Chapter 10 –Calculations with the Database 134
  Multiplication 134
  Addition 141

Chapter 11 – Formatting Numbers 147

Chapter 12 – Printing a Report 152
  Creating a Report with Xojo's Report Builder 152
  Vertical and Horizontal Resolution 157
  Adding a Picture or Logo 157
  Changing Report Picture at Runtime 162
  Add a Date 166
  Date Formats 169
  Custom Date Formats 170
  Page Number 174
  Formatting Page Number 178
    Page Number Example A 178
    Page Number Example B 182
    Page Number Total –Not working 186
  Page Setup 190
    Portrait and Landscape 190
  Page Margins 194
  Print Without Printer Dialogue 200
  Report Addition 203
  Grouping Data 207
  Grouping Descending Data 211

Subgouping Ascending Data ... 215
Add A Single Column ... 219
Font Properties ... 223
Font Code ... 224
Label Behaviours ... 229
Label Behaviours with Code ... 231
Field Format - Code ... 234
Print Without Printer Dialogue – Version 2 ... 238

Chapter 13 Working with Tables ... 241
Create New Table Only ... 241
Create New Table with Fields ... 243
Add Table Field ... 246
Delete Table Field - Workaround ... 248
Table Rename ... 255

Chapter 14 – Deleting Data ... 258
Deleting a Record ... 258
Delete All Records in a Table ... 260

Chapter 15 – Threads and Progress Bars ... 263
Creating a Thread ... 263
Adding a ProgressBar Thread ... 266
Database ProgressBar Thread ... 268

Chapter 16 – Database Encryption ... 272
Create Encrypted Database ... 272
Open Encrypted Database ... 275

Chapter 17 - Joins ... 279
Understanding Joins ... 279
The Concept ... 279
Inner Join ... 281
Full Outer Join (Not Supported by SQLite) ... 284
Left Join ... 286
Right Join (Not Supported by SQLite) ... 290
Left Outer Join ... 293
Right Outer Join (Not Supported by SQLite) ... 297

Chapter 18 – Deleting Tables ... 300

Chapter 19 – Handling Apostrophes ... 303
Identify Apostrophe Issue ... 303
Making Apostrophe’s Work ... 307

Chapter 20 – Alias 309
Column Alias/Field 309
Table Alias 313

Chapter 21 – Transactions 317

Chapter 22 – Prepared Statements 324
SQL Injection Attack 324
Select Prepared Statement 325
Update Prepared Statement 331
Delete Prepared Statement 336
Insert Prepared Statement........ 339

Chapter 23 – Reserved Words 344

Chapter 24 – Specifications 345

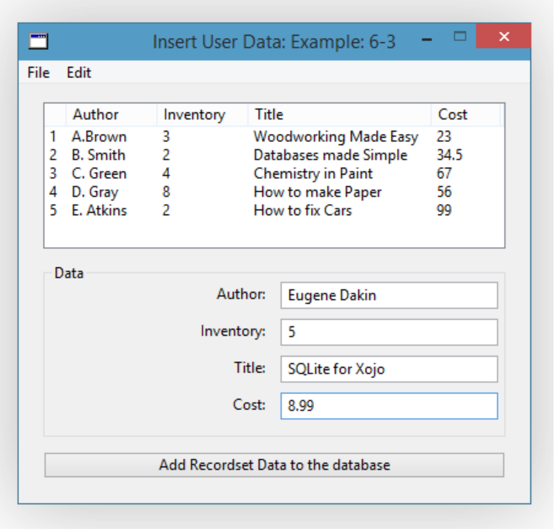
Chapter 25 - Other 346
Periodic Maintenance 346
Multiuser 346

Index........ 352

Inserting Recordset Data

User-based data is entered into the database by modifying inserting text field 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 Fields. This example uses data addition with a recordset instead of an SQL command.

Figure 63. Screengrab of Example 6-3 Prior to New Data



Insert User Data: Example: 6-3
File
Edit
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

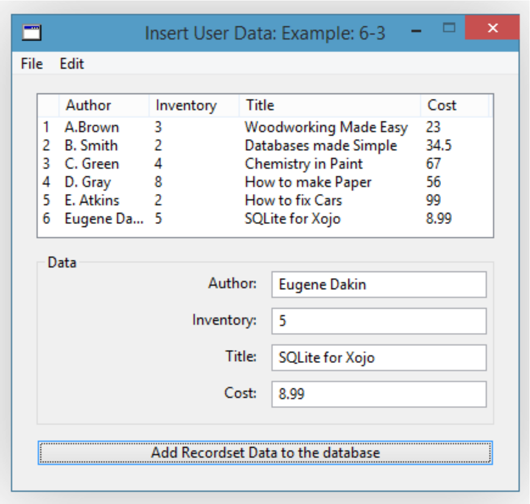
Inventory: 5

Title: SQLite for Xojo

Cost: 8.99

Data from the Insert.sqlite database is shown in the listbox and 5 entries are shown. There is data in the textfields, 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 64. Example 6-3 with Data


Insert User Data: Example: 6-3
File
Edit
Data
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
6	Eugene Da...	5	SQLite for Xojo	8.99

Author: Eugene Dakin

Inventory: 5

Title: SQLite for Xojo

Cost: 8.99

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.

```
Call ChooseDatabase
Call RefreshListbox
```

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 35. Example 6-3: ChooseDatabase Method

```
//Get the file name
Dim f as new FolderItem
Dim dlg as new OpenDialog
dlg.Title = "Open *.sqlite File"
f = dlg.ShowModal()

//Connect to a SQLite database
db = new SQLiteDatabase
db.DatabaseFile = f

If db.Connect = True then
  'The connection was successful!
```

```
 Else 'It did not connect
  MsgBox "An error occured connecting to the database" + EndOfLine +_
  db.ErrorMessage
  return
 End If

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

A folderitem variable called f is created to hold the file name and to remember the directory where the file exists. An OpenDialog 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 36. Example 6-3: RefreshListbox

```
  //Format the Listbox
 Listbox1.HasHeading = 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
 Dim rs as RecordSet
 rs = db.SQLSelect("SELECT * FROM Library") //Find data
 if rs <> Nil then //Continue if there is data
  do until rs.EOF //continue until we reach the End Of File
   Listbox1.AddRow
   Listbox1.Cell(Listbox1.LastIndex, 0) = Cstr(rs.Field("ID").IntegerValue)
   Listbox1.Cell(Listbox1.LastIndex, 1) = rs.Field("Author").StringValue
   Listbox1.Cell(Listbox1.LastIndex, 2) = Cstr(rs.Field("Inventory").DoubleValue)
   Listbox1.Cell(Listbox1.LastIndex, 3) = rs.Field("Title").StringValue
   Listbox1.Cell(Listbox1.LastIndex, 4) = Cstr(rs.Field("Cost").DoubleValue)
   rs.MoveNext //move to the next recordset
  loop //get the next row of data
 else
  MsgBox "No RecordSet exists."
  Return
 end if
 rs = Nil

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

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 recordset variable is created with the variable name rs. A SQLSelect 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 recordset is checked to see if there is data (rs <> Nil). If there is data, then then each field and its data is placed in the appropriate TextField. There is more than one record in this recordset, and a do-until loop is used to start from the first recordset and stop at the last recordset (rs.EOF). The IntegerValue ‘ID’ is place in the cell (0) of the new row (addrow). The column to the right has a value of 1, and contains the Author StringValue. Each field from the database populates the listbox until the end of the database file has been reached. If there is no data in the recordset, 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 37. Pushbutton Update Code for Example 6-3

```
//Insert data in the database
Dim row as new DatabaseRecord
row.Column("Author") = TFAuthor.Text
row.Column("Inventory") = TFInventory.Text
row.Column("Title")  = TFTitle.Text
row.Column("Cost") = TFCost.Text

db.InsertRecord("Library", row)

If db.Error then
  MsgBox "An error occured: " + db.ErrorMessage
End If

call RefreshListbox
```

Adding data to a database by a recordset requires a few more lines of code and is a little easier to prevent mistakes. A new DatabaseRecord 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 recordset. Once all data has been entered in the recordset then the *InsertRecord* 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

-, 134

*, 134

/, 134

?, 327

+, 134

ACID, 16

Action Button Caption, 322

Add, 134

Add Column, 246

Add Field, 23

Add Manual Data, 25

Add New Row, 93

Add Record, 25, 103

Add Recordset Data, 103

Add Table, 23

Add User Based Data, 98

Addition, 141

AddRow, 42

Albernate Action Button, 322

Alias, 309

- Column, 309
- Field, 309
- Table, 313

ALTER TABLE, 31

ANALYZE, 346

Apostrophe, 38

Average Value, 129

AVG, 130

Back End, 29

Backup a File, 116

BEGIN TRANSACTION, 321

BindType

- SQLITE_BLOB, 329
- SQLITE_BOOLEAN, 329
- SQLITE_DOUBLE, 329
- SQLITE_INT64, 329
- SQLITE_INTEGER, 329
- SQLITE_NULL, 329
- SQLITE_TEXT, 329

Calculations, 134

Caption Action Button, 322

chr, 42

Column Add, 246

Column Delete, 248

ColumnCount, 42

ColumnWidths, 42

Commit, 322

Connect to SQLite, 29

Connection, 29

CopyFileTo, 119

Copying a File, 116

Count, 121

COUNT, 123

Create a Report, 152

Create a Thread, 263

CREATE DATABASE, 31

Create Encrypted Database, 272

Create Encryption, 272

CREATE INDEX, 31

Create New Database, 22

CREATE TABLE, 31

CRUD, 18

Data Manual Add, 25

Database

    Definition, 16

Database Edit, 27

Database Encryption, 272

Database New, 22

Database ProgressBar Thread, 268

Database Specifications, 345

Database Terms, 16

Date

    Custom Date Formats, 170
    Default Date Formats, 169

Decrypt, 276

Delete, 119

    All Records, 260
    One Record, 258

DELETE, 31

Delete Column, 248

Delete Data, 258

Delete Prepared Statement, 336

Deleting Tables, 300

Dialogue Box

    Labels, 113
    Open, 111

Divide, 134

DROP INDEX, 31

DROP TABLE, 31

Edit Database, 27

Employees.sqlite, 17

Encrypt, 273

Encryption, 272

Encryption Key, 276

End-of-File, 40

EOF, 40, 43, 47, 51, 55, 58, 61, 64, 68, 73, 77, 82, 87, 91, 96, 101, 106, 115, 137, 140, 144, 146, 278, 306, 312, 315, 321, 329, 330, 334, 342, 350

Examples

01-01 REALStudio Program, 15
03-01 SQLite Connection, 29
04-01 Populating Textfields, 33
04-02 SELECT a Specific Recordset, 36
04-03 Populating a Combobox, 39
04-04 Populating a Listbox, 41
04-05 Database Sorting, 46
04-06 Finding Data, 49
04-07 Filtering Numerical Data, 53
04-08 User Based Searching, 56
04-09 Plural Searching, 59
04-10 Excluding Data, 63
04-11 Wildcard Searching, 66
05-01 Pushbutton Update, 73
05-01 RefreshListbox, 72
05-02 Pushbutton Update, 77
05-02 RefreshListbox, 76
05-03 Pushbutton Update, 82
05-03 RefreshListbox, 80
05-04 Pushbutton Update, 87
05-04 RefreshListbox, 85
05-05 Pushbutton Update, 91
05-05 RefreshListbox, 90
06-01 Pushbutton Update, 97
06-01 RefreshListbox, 95
06-02 Pushbutton Update, 102
06-02 RefreshListbox, 100
06-03 Insert Recordset, 107
08-01 Displaying Data from a Database, 113
08-01 Opening a SQLite Database, 112
08-02 Copying or Backing Up a Database, 117
09-01 Counting Records, 122
09-02 Maximum Field Value, 124
09-03 Minimum Field Value, 127
09-04 Average Field Value, 129
09-05 Sum Field Value, 132
10-01 Multiplication Action, 138
10-01 Multiplication RefreshListbox, 135
10-02 RefreshListbox, 141
11-01 Formatting Database Numbers, 149
12-01 Gathering Recordset Data, 154
12-01 Print Report Method, 155
12-02 Gathering Recordset Data, 159
12-02 PrintMyReport Method, 161
12-03 Change Logo Picture at Runtime, 162
12-03 Gathering Recordset Data, 163
12-03 PrintMyReport Method, 165
12-04 Add a Date Recordset Data, 166
12-04 PrintMyReport Method, 168
12-05 Gathering RecordSet Data, 171
12-05 PrintMyReport Method, 173
12-06 Page Number RecordSet Data, 175
12-06 PrintMyReport Method, 177
12-08 Page Number Recordset Data, 179
12-08 PrintMyReport Method, 180
12-09 Gather Recordset Data, 183
12-09 PrintMyReport Method, 185
12-09 Single Label Page Number, 182
12-10 Gather Recordset Data, 187
12-10 PrintMyReport Method, 188
12-10 Total Page Numbers, 186
12-11 Gather Recordset Data, 191
12-11 Page Orientation, 190
12-11 PrintMyReport Method, 192
12-12 Page Margins Recordset Data, 197
12-12 PrintMyReport Method, 198
12-13 No Printer Dialogue Recordset Data, 200
12-13 PrintMyReport Method, 201
12-14 PrintMyReport Method, 205
12-14 Sub Total Recordset Data, 204
12-15 Group PrintMyReport Method, 210
12-15 Group Recordset Data, 208
12-15 SQL Command, 207
12-16 Descending Recordset Data, 212
12-16 Descending SQL Command, 211
12-16 Group PrintMyReport Method, 213

12-17 Main and Sub Category Ordering, 215
12-17 PrintMyReport Method, 217
12-17 Subgroup Ascending Recordset Data, 216
12-18 Add A Column, 220
12-18 New SQL Command Snippet, 219
12-18 Old SQL Command Snippet, 219
12-19 Font Properties, 223
12-20 Change Text Bold Setting Snippet, 225
12-20 Change Text Font Setting Snippet, 224
12-20 Change Text Italic Setting Snippet, 225
12-20 Change Text Size Setting Snippet, 225
12-20 Change Text Underline Setting Snippet, 225
12-20 PrintMyReport Method, 227
12-20 Show Text Font Setting Snippet, 224
12-20 Show Text Size Setting Snippet, 224
12-21 Report Layout Behaviour, 230
12-22 Colour Alpha Channel Type, 233
12-22 Horizontal Alignment Code Snippet, 231
12-22 Text Backcolour Code Snippet, 233
12-22 Text Colour Code Snippet, 233
12-22 Vertical Alignment Code Snippet, 232
12-22 Word Wrapping Code Snippet, 234
12-23 Field Formatting Code Snippet, 235
12-23 Field Formatting Recordset Data, 235
12-23 General Formatting Code, 234
12-23 PrintMyReport Method, 237
12-25 No Printer Dialogue Recordset Data, 238
12-25 PrintMyReport Method, 239
13-01 Adding a New Table, 241
13-02 Create New Table with Fields, 244
13-03 Add a New Field, 247
13-04 Delete Field Workaround, 253
13-05 Rename Table, 255
14-01 Deleting a Record, 259
14-02 Deleting All Records, 261
15-01 PushButton Action Event, 265
15-01 Thread Run Event, 264
15-01 Timer Action Event, 264
15-02 PushButton Action Event, 267
15-02 Thread Run Event, 266
15-02 Timer Action Event, 267
15-03 PushButton Action Event, 270
15-03 PushButton Open Event, 270
15-03 Thread1 Run Event, 269
15-03 Timer Action Event, 269
16-01 Encrypted DB, 273
16-02 Open Encrypted DB, 276
17-01 Inner Join, 282
17-02 Full Outer Join, 284
17-03 Left Join, 287
17-04 Right Join, 291
17-05 Left Outer Join, 294
17-06 Right Outer Join, 297
18-01 Deleting a Table, 301
19-01 Apostrophe Error, 303
19-02 Apostrophe Safe SQL, 307
20-01 Column Alias, 310
20-02 Table Alias, 313
21-01 Choose Database, 319
21-01 Pushbutton Action, 321
21-01 RefreshListbox, 320
22-01 SQL Prepared Statement, 326
22-02 Choose Database Method, 332
22-02 Pushbutton Update, 335
22-02 RefreshListbox, 333
22-03 Prepared Statement Delete Record, 337
22-04 ChooseDatabase Method, 340
22-04 Pushbutton Insert, 342
22-04 RefreshListbox, 341
25-02 Multiuser, 348
Other-1 Analyze Maintenance, 346
Excluding Filtered Data, 61

Exists, 119

Extensions, 112, 118

Field, 42

Field Add, 23, 246

Field Delete, 248

Field Name Spaces, 16

Field Type

Blob, 243
Integer PRIMARY KEY, 243
Numeric, 243
Text, 243
Field values, 121

File

Adding Data, 93
Average Field Value, 129
Building a Menu, 108
Count Records, 121
Create New Database, 29
Editing Data, 69
Get Database, 32
Maximum Field Value, 124
Minimum Field Value, 127
Sum Field Value, 132
File Handling, 110

FileType, 112

Extensions, 112
FileType, 112
name, 112
Finding Data, 48

Finding Numerical Data, 52

Flat Database, 17

Formatting Numbers, 147

Front End, 29

Get Database Data, 32

Handling Apostrophes, 303

InitialValue, 42

INSERT INTO, 31, 97

INSERT Row, 97

INTEGER, 20

IsThreadRunning, 270

Joins, 279

Full Outer Join, 284
Inner Join, 281
Left Join, 286
Left Outer Join, 293
Right Join, 290
Right Outer Join, 297
The Concept, 279
Understanding, 279
Key Encryption, 276

LastIndex, 42

Listbox, 42

Maintenance, 346

Marker

VVV, 328

$VVV, 328
?, 328
?NNN, 328
@VVV, 328
MAX, 125

Maximum Value, 124

Message, 322

MessageDialog, 322

MessageDialogButton, 322

MIN, 128

Minimum Value, 127

MoveNext, 42

Multiple User Based Updates, 83

Multiplication, 134

Multiply, 134

Multiuser, 346

name, 112

New Database, 22

NONE, 20

NUMERIC, 20

Open Encrypted DB, 275

Opening a File, 110

OpenPrinter, 239

OpenPrinterDialog, 201

Page Orientation, 190

Page Orientation Values, 191

Plural Searches, 59

Populating

  Textfields, 33

Populating a Combobox, 38

Populating a Listbox, 41

Prepare, 327, 335

Prepared SQL Statement

  ?, 335
  Bind, 335
  Bind Type, 335
  Compiled SQL Statement, 324
  DELETE, 336
  Insert, 339
  Prepare, 335
  PreparedSQLStatement, 335
  SELECT, 325
  SQLExecute, 335
  SQLITE_BLOB, 329
  SQLITE_BOOLEAN, 329
  SQLITE_DOUBLE, 329
  SQLITE_INT64, 329
  SQLITE_INTEGER, 329
  SQLITE_NULL, 329
  SQLITE_TEXT, 329
  SQLSelect, 327
  UPDATE, 331

Prepared Statements, 324

PreparedSQLStatement, 327

Primay Key, 27

Print a Report, 152

Progress Bar, 263

Progress Bar Thread, 266

ProgressBar, 269

ProgressBar Database Thread, 268

REAL, 20

Record Add, 25

Records, 16

Delete all Records, 260
Delete one Record, 258
Relation, 17

Relational database, 17

Rename Table, 255

Report, 152

A Simple Report, 152
Add a Date, 166
Add a Logo, 157
Add a Picture, 157
Add A Single Column, 219
Addition, 203
Change Picture at Runtime, 162
Characters for Custom Formatting, 234
Custom Date Formats, 170
Date Formats, 169
Field Format Code, 234
Font Code, 224
Font Properties, 223
Format a Page Number, 178
Grouping Data, 207
Grouping Descending Data, 211
Label Behaviours, 229, 231
Landscape, 190
Page Margins, 194
Page Number, 174
Page Number A, 178, 182
Page Setup, 190
Portrait, 190
Resolution Settings, 157
Resolutions, 157
Subgrouping Ascending Data, 215
Text Colour Choices, 230
Text Horizontal Alignment Type, 232
Text Horizontal Choices, 229
Text Vertical Alignment Type, 232
Text Vertical Choices, 229
Total Page Numbers, 186
Without Printer Dialogue, 200, 238
Reserved Words, 344

Return, 42

Rollback, 322

SaveAsDialog, 118

Select, 38, 303

SELECT, 31

SELECT AVG, 130

SELECT COUNT, 123

SELECT MAX, 125

SELECT MIN, 128

Select Prepared Statement, 325

SELECT SUM, 133

Self-Contained, 16

Serverless, 16

ShowModal, 41, 322

Sorting Data, 44

Spaces, 16

Spaces Field Name, 16

Specification, 345

SQL

=, 50
>, 54
ADD, 247

ALTER, 247
ALTER TABLE, 31, 247, 248
apostrophe, 38
CREATE DATABASE, 31
CREATE INDEX, 31
CREATE TABLE, 31, 249
DELETE, 259
DELETE, 31
DROP, 251
DROP INDEX, 31
DROP TABLE, 31
Greater Than, 54
INSERT INTO, 31, 97, 250, 252
ON, 314
ORDER BY, 45
RENAME, 256
RENAME TO, 256
SELECT, 31, 250
SQL Fundamentals, 31
UPDATE, 31
VALUES, 97
WHERE, 34

SQL Injection Attack, 324

SQLExecute, 259

SQLite

INTEGER, 20
NUMERIC, 20
Open, 19
REAL, 20
TEXT, 20

SQLite Database Browser, 18

SQLite Database Open, 19

SQLite Field Types, 243

SQLitePreparedStatement, 327

SQLSelect, 327

Subtract, 134

SuggestedFileName, 118

SUM, 133

Sum Field Value, 132

Table

Create a New Table, 241
Create with Fields, 243

Table Add, 23

Table Rename, 255

TEXT, 20

Thread

Kill, 270
Run, 270

Thread Creation, 263

Thread Database ProgressBar, 268

Thread Progress Bar, 266

Threads, 263

Timer

Mode, 270
ModeMultiple, 270
ModeOff, 270
Period, 270

Transactional, 16

Transactions, 317

UPDATE, 31

Update Prepared Statement, 331

Updating a Single Field, 69

Updating Multiple Fields, 74

Updating Multiple Records, 88

User Based Searching, 56

User Based Updating, 78

Value of Fields, 121

WAL, 346

While-Wend, 327

Wildcard Filtering, 65

Working with Tables, 241

Write-Ahead-Logging, 346

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 Desktop Apps” 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 350 pages with greater than 70 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 Great White Software. This book can be purchased at http://great-white-software.com/rblibrary/ 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: 978-1-927924-08-2