

SQLite Diff - How to compare SQLite databases

1

1.1 SQLite Diff - How to compare SQLite database files

4





SQLite Diff - How to compare SQLite database files

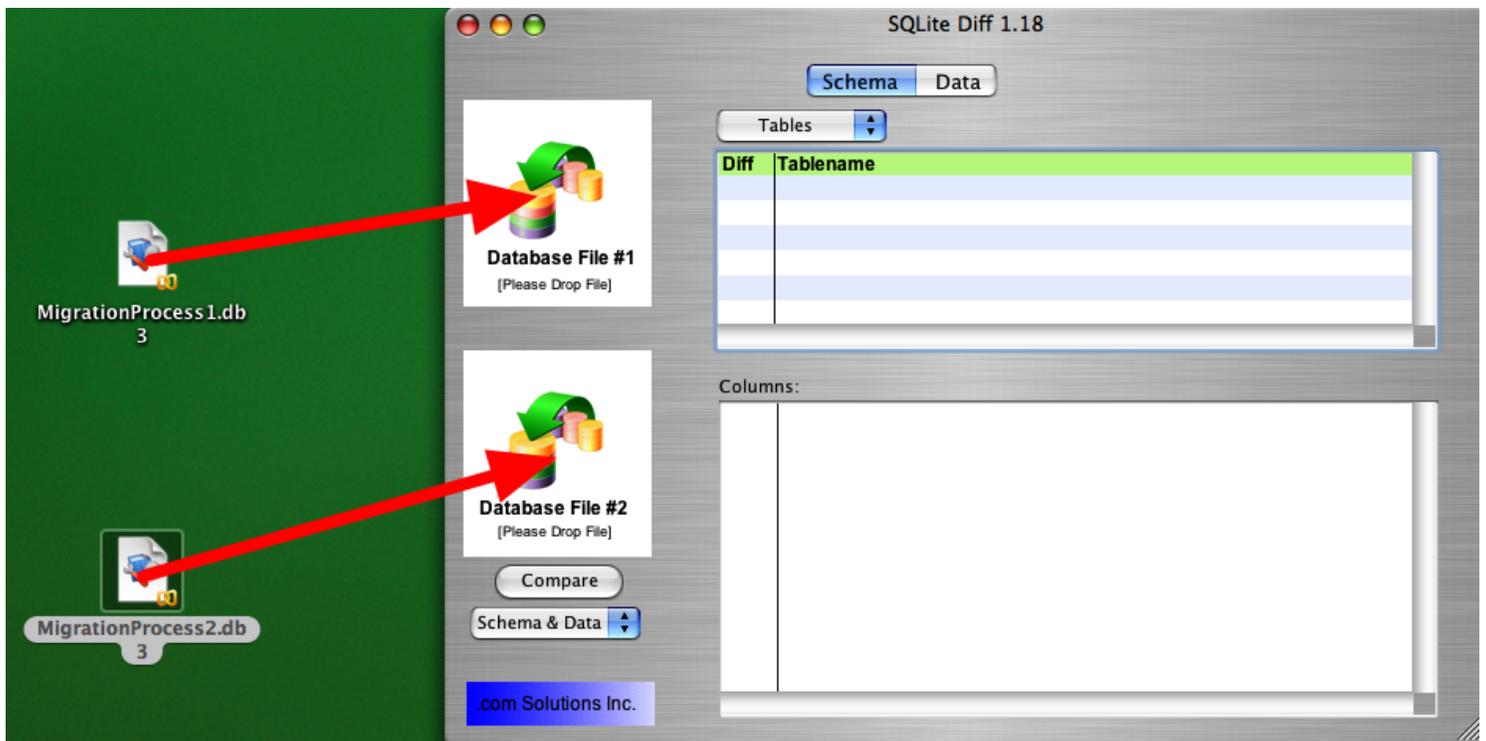
This document explains the process of comparing SQLite database files.

Document Version 01

SQLite Diff 1.18

10/12/2008

Step 1 - Selecting SQLite database files



SQLite Diff uses a Drag & Drop technique for selecting the two SQLite database files which will be compared.

You can think of the Database #1 file as being the first version of the file you want to compare, and Database #2 as the second version of the file, after you have made changes.



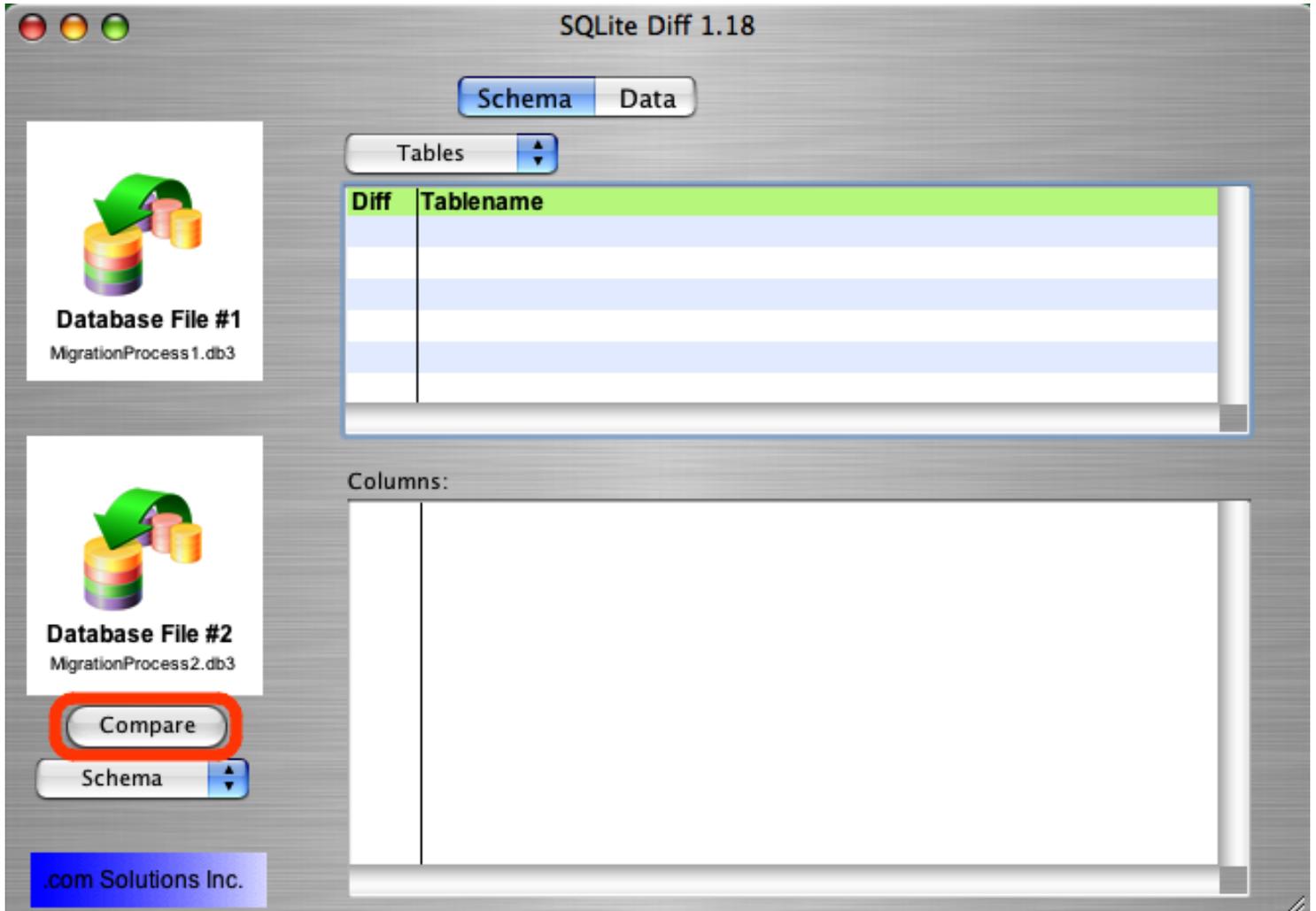
Selected files



The name of each SQLite database file will be displayed at the bottom of its Drag & Drop icon.



Step 2 - Comparing Database Schema

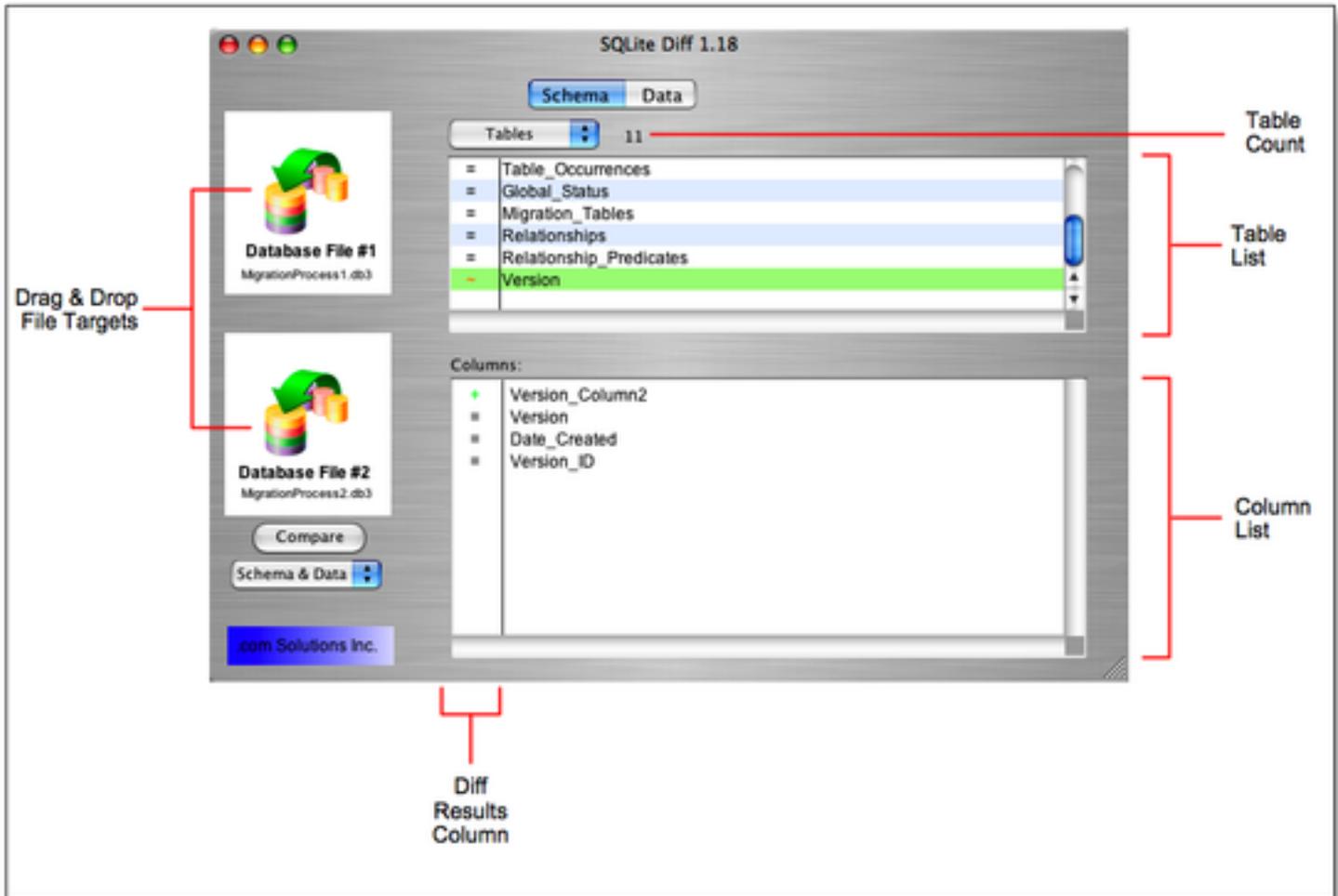


If you only want to compare database schemas, select Schema instead of Schema & Data from the comparison type menu.

Then click the Compare button.



Schema comparison results - Tables



Once the schema comparison has completed, the Tables results will be displayed.

Click on any individual table to see the results. The Diff Results column will be displayed with the following symbols as an indication of the type of changes which have occurred:

- = Object Matches
- + Object Added
- Object Deleted
- ~ Object Changed

This screenshot shows that the column named Version_Column2 was added to the Version table.



Schema comparison results - Triggers

Diff	Trigger
+	LayoutName2
-	LayoutName
=	fkDelCascade_RelationshipID_RelationshipPredicates
~	LayoutName3
-	LayoutName4
-	LayoutName5

Trigger SQL Code:

```
CREATE TRIGGER LayoutName4 AFTER INSERT ON Layouts FOR EACH ROW WHEN New.Layout_Name is NULL BEGIN UPDATE Layouts SET Layout_Name = 'Layout55' || New.Layout_ID WHERE Layout_ID=New.Layout_ID; END
```

Viewing other schema results can be done via selecting the object type from the Object Type menu. Selecting Triggers from the menu displays the list of triggers within the compared database files.

SQLite Diff first compares lists of triggers by name between the two database files. Then the trigger creation SQL code is compared for triggers having the same name within both of the SQLite database files.

Changed trigger SQL code is displayed with yellow background, as shown in this first screenshot.

Add trigger SQL code is displayed with a green background. The green text background means that the



highlighted SQL code was added to the trigger in the 2nd database file.

Schema comparison results - Indexes

The screenshot displays a software interface for comparing database indexes. At the top, a dropdown menu labeled 'Indexes' shows a count of 8. Below this is a list of indexes with the following entries:

- = FMSD_Script_Name_idx
- = Migration_Columns_Repeat_Count_idx
- = sqlite_autoindex_Table_Occurrences_1
- = TO_Name_New_idx
- = Migration_Columns_Table_ID_idx
- Relationship_Predicate_RID_idx

The 'Relationship_Predicate_RID_idx' entry is highlighted in green. Below the list is a section titled 'Index SQL Code:' containing the following SQL code:

```
CREATE INDEX Relationship_Predicate_RID_idx ON  
Relationship_Predicates (Join_Type)
```

The 'Join_Type' column name in the SQL code is highlighted in yellow. Red lines and brackets on the right side of the interface label these elements as 'Index Count', 'Index List', and 'Index SQL Code'. At the bottom, two labels with red lines pointing to the interface are 'Diff Results Column' (pointing to the minus sign next to the selected index) and 'Yellow Highlighted Changed SQL' (pointing to the yellow highlight on 'Join_Type').

SQLite Diff first compares lists of indexes by name between the two database files.

Then the index creation SQL code is compared for indexes having the same name within both of the SQLite database files.

Changed, added or deleted index creation SQL code is displayed with a colored background in order to make it easy to spot changes.



Schema comparison results - Views

Diff	View
-	Triggers_list_onlyDB1
=	Triggers_list_only
=	Triggers_list_only_NEW
-	Triggers_list_only_NEW2

View SQL Code:

```
CREATE VIEW Triggers_list_only_NEW2 as SELECT name,sql,type FROM
sqlite_master where type='trigger'22'
```

Annotations: Red lines connect 'View Count' to the number 4, 'View List' to the table, and 'View SQL Code' to the code block. A red box highlights the value '22' in the SQL code. Red lines also connect 'Diff Results Column' to the 'Diff' column and 'Yellow Highlighted Changed SQL' to the yellow highlight in the SQL code.

SQLite Diff first compares lists of views by name between the two database files.

Then the view creation SQL code is compared for views having the same name within both of the SQLite database files.

Changed, added or deleted view creation SQL code is displayed with a colorized background in order to make it easy to spot changes.



SQLite Data Comparison - Added Records

SQLite Diff 1.18

Schema Data

Database #2 Tables: 53

Table Count

Table List

Changed Records List

DB#	Diff	ROWID	gnpk	gn_colName
DB2	+	410	410	gn_library
DB2	+	411	411	gn_gnDateCreated
DB2	+	412	412	gn_gnDateModified
DB2	+	413	413	gn_fdName
DB2	+	414	414	gn_fdLocation
DB2	+	415	415	gn_fdInstructor
DB2	+	416	416	gn_fdFrequency
DB2	+	417	417	gn_fdDescription
DB2	+	418	418	gn_fdTeachingAssist
DB2	+	419	419	gn_fdFinalGrade
DB2	+	420	420	gn_fdNotes
DB2	+	421	421	gn_fdStartTime
DB2	+	422	422	gn_fdEndTime
DB2	+	423	423	gn_fdCredits
DB2	+	424	424	gn_fdStartDate
DB2	+	425	425	gn_fdEndDate

Diff Results Columns

Selecting the Schema & Data menu selection (this is the default), compares records in each of the tables which exist within both of the database files.

SQLite Diff uses the SQLite ROWID value to compare each record between each of the database files. The tables list contains summary info displaying the difference count and number of records found in each source database file.

Clicking on an individual table displays the added/deleted/changed records as a comparison between first database (DB1) and the second database file (DB2).

