



*Small. Fast. Reliable.
Choose any three.*

[Home](#) [Menu](#) [About](#) [Documentation](#) [Download](#) [License](#) [Support](#)

[Purchase](#)

[Search](#)

sqldiff.exe: Database Difference Utility

► Table Of Contents

1. Usage

The `sqldiff.exe` binary is a command-line utility program that displays the differences between SQLite databases. Example usage:

```
sqldiff [options] database1.sqlite database2.sqlite
```

The usual output is an SQL script that will transform `database1.sqlite` (the "source" database) into `database2.sqlite` (the "destination" database). This behavior can be altered using command-line switches:

--changeset FILE

Do not write changes to standard output. Instead, write a (binary) changeset file into `FILE`. The changeset can be interpreted using the sessions extension to SQLite.

--lib LIBRARY

-L LIBRARY

Load the shared library or DLL file `LIBRARY` into SQLite prior to computing the differences. This can be used to add application-defined [collating sequences](#) that are required by the schema.

--primarykey

Use the schema-defined [PRIMARY KEY](#) instead of the [rowid](#) to pair rows in the source and destination database. (See additional explanation below.)

--schema

Show only differences in the schema not the table content

--summary

Show how many rows have changed on each table, but do not show the actual changes

--table TABLE

Show only the differences in content for TABLE, not for the entire database

--transaction

Wrap SQL output in a single large transaction

--vtab

Add support for handling [FTS3](#), [FTS5](#) and [rtree](#) virtual tables. [See below](#) for details.

2. How It Works

The sqldiff.exe utility works by finding rows in the source and destination that are logical "pairs". The default behavior is to treat two rows as pairs if they are in tables with the same name and they have the same [rowid](#), or in the case of a [WITHOUT ROWID](#) table if they have the same [PRIMARY KEY](#). Any differences in the content of paired rows are output as UPDATES. Rows in the source database that could not be paired are output as DELETES. Rows in the destination database that could not be paired are output as INSERTS.

The --primarykey flag changes the pairing algorithm slightly so that the schema-declared [PRIMARY KEY](#) is always used for pairing, even on tables that have a [rowid](#). This is often a better choice for finding differences, however it can lead to missed differences in the case of rows that have one or more PRIMARY KEY columns set to NULL.

3. Limitations

1. The sqldiff.exe utility is unable to compute differences for rowid tables for which the rowid is inaccessible. An example of a table with an inaccessible rowid is:

```
CREATE TABLE inaccessible_rowid(  
    "rowid" TEXT,  
    "oid" TEXT,  
    "_rowid_" TEXT  
);
```

2. The sqldiff.exe utility does not (currently) display differences in [TRIGGERS](#) or [VIEWS](#).
3. By default, differences in the schema or content of virtual tables are not reported on.

However, if a [virtual table](#) implementation creates real tables (sometimes referred to as "shadow" tables) within the database to store its data in, then sqldiff.exe does calculate the difference between these. This can have surprising effects if the resulting SQL script is then run on a database that is not *exactly* the same as the source database. For several of SQLite's bundled virtual tables (FTS3, FTS5, rtree

and others), the surprising effects may include corruption of the virtual table content.

If the `--vtab` option is passed to `sqldiff.exe`, then it ignores all underlying shadow tables belonging to an FTS3, FTS5 or rtree virtual table and instead includes the virtual table differences directly.