How to convert a postgres database to sqlite

Asked 11 years, 8 months ago Modified 10 months ago Viewed 46k times

27

We're working on a website, and when we develop locally (one of us from Windows), we use sqlite3, but on the server (linux) we use postgres. We'd like to be able to import the production database into our development process, so I'm wondering if there is a way to convert from a postgres database dump to something sqlite3 can understand (just feeding it the postgres's dumped SQL gave many, many errors). Or would it be easier just to install postgres on windows? Thanks.

postgresql sqlite

Share Edit Follow

edited Feb 17, 2021 at 0:16
Shadow
33k 10 52 62

asked May 27, 2011 at 6:02
luqui
59k 11 143 202

22 I'd just switch your development environment to PostgreSQL, developing on top of one database (especially one as loose and forgiving as SQLite) but deploying on another (especially one as strict as PostgreSQL) is generally a recipe for aggravation and swearing. – mu is too short May 27, 2011 at 7:15

12 To echo mu's response, DON'T DO THIS..DON'T DO THIS..DON'T DO THIS. Develop and deploy on the same thing. It's bad engineering practice to do otherwise. – Kuberchaun May 27, 2011 at 14:38

2 Developing on SQLite and running on postgres is perfectly acceptable IF you are using an ORM which deals with all the differences in dialect and schema for you. – jwg Feb 23, 2018 at 9:11

1 @Zaz You meant serverfault.com/questions/274355/… – Andrew Grimm Dec 10, 2018 at 4:02

The same question on ServerFault. – Zaz Dec 11, 2018 at 20:59

9 Answers

Sorted by:

Highest score (default)

26

I found this blog entry which guides you to do these steps:

1. Create a dump of the PostgreSQL database.

```
ssh -C username@hostname.com pg_dump --data-only --inserts YOUR_DB_NAME >
dump.sql
```

2. Remove/modify the dump.

    1. Remove the lines starting with `SET`
    2. Remove the lines starting with `SELECT pg_catalog.setval`
    3. Replace true for ‘ `t` ’
    4. Replace false for ‘ `f` ’

3. Add `BEGIN;` as first line and `END;` as last line
4. Recreate an empty development database. `bundle exec rake db:migrate`
5. Import the dump.

```
sqlite3 db/development.sqlite3
sqlite> delete from schema_migrations;
sqlite> .read dump.sql
```

Of course connecting via ssh and creating a new db using rake are optional

Share Edit Follow

edited May 29, 2014 at 10:39
a_horse_with_no_name
537k 99 852 900

answered May 29, 2014 at 10:36
tutuDajuju
10k 5 65 88

4 For me, `t` and `f` did not work, I had to use `1` and `0` as described in these posts: stackoverflow.com/questions/5768364/… – neRok Jun 11, 2016 at 3:35

1 blobs saved in different format. sqlite uses x'010101', while postgres '\x0101' – Evgen Bodunov Oct 18, 2016 at 11:57

1 I created a gist that was based from this and other sources: gist.github.com/jeffreycastro/918a24588ac1628e2af6b0def5bb7416 – Jeffrey M Castro Nov 29, 2019 at 3:40

answer is not complete. check here - medium.com/@andreypu/… more detailed explanation. helped me out – Kirguduck Dec 15, 2021 at 15:26

13

STEP1: make a dump of your database structure and data

```
pg_dump --create --inserts -f myPgDump.sql \
   -d myDatabaseName -U myUserName -W myPassword
```

STEP2: delete everything except CREATE TABLES and INSERT statements out of myPgDump.sql (using text editor)

STEP3: initialize your SQLite database passing structure and data of your Postgres dump

```
sqlite3 myNewSQLiteDB.db -init myPgDump.sql
```

STEP4: use your database ;)

Share Edit Follow

edited Feb 17, 2021 at 0:15
dfrankow
19.6k 40 145 203

answered Dec 4, 2014 at 13:37
Artem Zaytsev
1,571 19 19

5

Taken from https://stackoverflow.com/a/31521432/1680728 (upvote there): The `sequel` gem makes this a very relaxing procedure:

First install Ruby, then install the gem by running `gem install sequel`.

In case of sqlite, it would be like this: `sequel -C postgres://user@localhost/db sqlite://db/production.sqlite3`

Credits to @lulalala .

Share Edit Follow

edited Nov 27, 2020 at 9:31

answered Oct 6, 2020 at 8:37
Felix
4,363 2 31 44

2

You can use pg2sqlite for converting pg_dump output to sqlite.

```
#  Making dump
pg_dump -h host -U user -f database.dump database

#  Making sqlite database
pg2sqlite -d database.dump -o sqlite.db
```

Schemas is not supported by pg2sqlite, and if you dump contains schema then you need to remove it. You can use this script:

```
# sed 's/<schema name>\.//' -i  database.dump
sed 's/public\.//' -i  database.dump
pg2sqlite -d database.dump -o sqlite.db
```

Share Edit Follow

edited Sep 23, 2021 at 2:24

answered Sep 23, 2021 at 2:17
caiiiycuk
1,396 13 20

1 Tried this tool. It ran for a few seconds, reported no errors and produced a sqlite db of zero bytes :/ – boxed May 11, 2022 at 7:08

This programme currently fails with exceptions against a dump from Postgresql 15. – Luís de Sousa Jan 19 at 9:13

Was fixed in 1.0.1 – caiiiycuk Jan 24 at 8:53

2

Even though there are many very good helpful answers here, I just want to mark this as answered. We ended up going with the advice of the comments:

> I'd just switch your development environment to PostgreSQL, developing on top of one database (especially one as loose and forgiving as SQLite) but deploying on another (especially one as strict as PostgreSQL) is generally a recipe for aggravation and swearing. – @mu is too short

> To echo mu's response, DON'T DO THIS..DON'T DO THIS..DON'T DO THIS. Develop and deploy on the same thing. It's bad engineering practice to do otherwise. – @Kuberchaun

So we just installed postgres on our dev machines. It was easy to get going and worked very smoothly.

Share Edit Follow

answered Sep 27, 2021 at 3:11

luqui
59k 11 143 202

1

In case one needs a more automatized solution, here's a head start:

```
#!/bin/bash

$table_name=TABLENAMEHERE

PGPASSWORD="PASSWORD" /usr/bin/pg_dump --file "results_dump.sql" --host
"yourhost.com" --username "username" --no-password --verbose --format=p --
create --clean --disable-dollar-quoting --inserts --column-inserts --table
"public.${table_name}" "memseq"

# Some clean ups
perl -0777 -i.original -pe "s/.+?(INSERT)/\1/is" results_dump.sql
perl -0777 -i.original -pe "s/--.+//is" results_dump.sql

# Remove public. prefix from table name
sed -i "s/public.${table_name}/${table_name}/g" results_dump.sql

# fix binary blobs
sed -i "s/'\\\\x/x'/g" results_dump.sql

# use transactions to make it faster
echo 'BEGIN;' | cat - results_dump.sql > temp && mv temp results_dump.sql
echo 'END;' >> results_dump.sql

# clean the current table
sqlite3 results.sqlite "DELETE FROM ${table_name};"

# finally apply changes
sqlite3 results.sqlite < results_dump.sql && \
rm results_dump.sql && \
rm results_dump.sql.original
```

Share Edit Follow

edited Dec 20, 2019 at 15:11

marc_s
722k 173 1320 1443

answered May 8, 2019 at 12:32

random_user
808 1 7 17

great script ! what about `boolean` values ? what do you do with those ? – Ricky Levi Jul 1, 2019 at 18:57

What's `memseq` ? I'm getting this error: `pg_dump: error: too many command-line arguments (first is "memseq")` – bmaupin Oct 7, 2021 at 19:08

0

Create a dump of the PostgreSQL database. ssh -C username@hostname.com pg_dump --data-only --inserts YOUR_DB_NAME > dump.sql. Remove/modify the dump. Remove the lines starting with SET. Add BEGIN; as first line and END; as last line. Recreate an empty development database. Import the dump.

Share Edit Follow

answered Mar 29, 2022 at 11:52
angelhart
1

0

when I faced with same issue I did not find any useful advices on Internet. My source PostgreSQL db had very complicated schema.
You just need to remove from your db-file manually everything besides table creating
More details - here

Share Edit Follow

answered Mar 29, 2022 at 11:48
Kirguduck
716 1 9 20

-1

It was VERY easy for me to do using the taps gem as described here: http://railscasts.com/episodes/342-migrating-to-postgresql

And I've started using the Postgres.app on my Mac (no install needed, drop the app in your Applications directory, although might have to add one line to your PATH envirnment variable as described in the documentation), with Induction.app as a GUI tool to view/query the database.

Share Edit Follow

answered Oct 14, 2012 at 20:30
jpw
18.4k 23 110 180

But this question is about going the other way around - from postgres to sqlite, not to postgres, as your comment and reference suggest. – nealmcb Dec 31, 2012 at 18:09