

Postgres Restore Read Me

The Postgres backup is available in two formats:

- PSQL method
- pg_restore method

The psql method is more version independent, in that the restore can be executed from the command line and is less likely to have any compatibility issues.

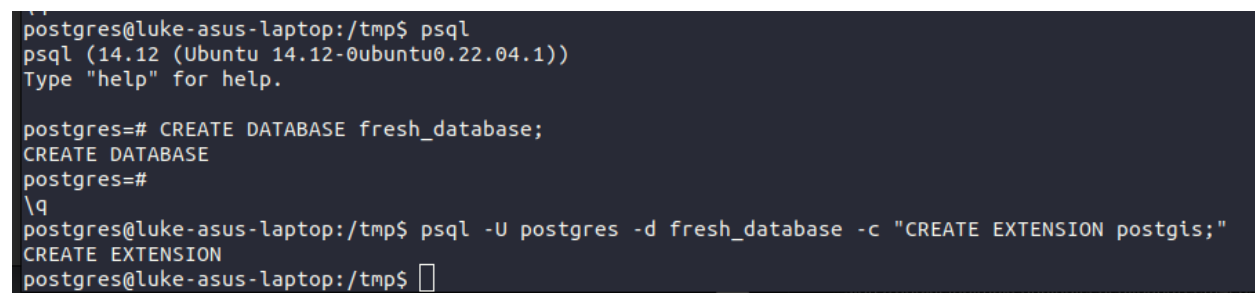
The pg_restore method should still be robust but may encounter issues if you are using an old version of Postgres. This method provides more flexibility in that it is possible to select which objects you wish to restore.

Postgres WA Geochemistry Database Restoration via PSQL

Restore on a clean database

1. Make a fresh database to test the restore with, ensure postgis is enabled

```
postgres=# CREATE DATABASE fresh_database;
CREATE DATABASE
postgres=#
\q
postgres@xxx:/tmp$ psql -U postgres -d fresh_database -c "CREATE EXTENSION postgis;"
```



```
postgres@luke-asus-laptop:/tmp$ psql
psql (14.12 (Ubuntu 14.12-0ubuntu0.22.04.1))
Type "help" for help.

postgres=# CREATE DATABASE fresh_database;
CREATE DATABASE
postgres=#
\q
postgres@luke-asus-laptop:/tmp$ psql -U postgres -d fresh_database -c "CREATE EXTENSION postgis;"
CREATE EXTENSION
postgres@luke-asus-laptop:/tmp$
```

2. Run the command to restore the database, *setting the backup file name accordingly* e.g.

```
postgres@xxx:/tmp$ psql -d fresh_database -U postgres -f "GSA_MmmYYYY_Backup.sql"
```

```
CREATE EXTENSION
postgres@luke-asus-laptop:/tmp$ psql -d fresh_database -U postgres -f "GSWA_November2023_Backup.sql"
SET
SET
SET
SET
SET
SET
set_config
-----

(1 row)

SET
SET
SET
SET
psql:GSWA_November2023_Backup.sql:23: ERROR: schema "public" already exists
ALTER SCHEMA
COMMENT
SET
SET
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
ALTER TABLE
ALTER SEQUENCE
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
ALTER TABLE
ALTER SEQUENCE
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
ALTER TABLE
ALTER SEQUENCE
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
ALTER TABLE
ALTER SEQUENCE
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
ALTER TABLE
ALTER SEQUENCE
CREATE TABLE
ALTER TABLE
```

The table data starts to come into postgres after the COPY commands:

The screenshot shows a PostgreSQL client interface. On the left is a tree view of the database structure, including a 'public' schema with various objects. The 'Tables (11)' folder is expanded, and 'gswa_dh_assay' is selected. The main window is split into two panes. The top pane, 'Query History', shows a SQL query: `1 SELECT * FROM public.gswa_dh_assay`, `2 LIMIT 100`, and `3`. The bottom pane, 'Data Output', shows a table with 15 rows of data. The table has the following columns: `ogc_fid` (integer), `dhassayid` (numeric(19)), `id` (numeric(19)), `dhgeochemid` (numeric(10)), `companyholeid` (character varying(300)), and `companysampleid` (character varying(300)).

ogc_fid	dhassayid	id	dhgeochemid	companyholeid	companysampleid
1	2399566642	146576201	5531422	MSD4	MSD4_017
2	2399567986	146576201	5531422	MSD4	MSD4_017
3	2399567047	146576201	5531422	MSD4	MSD4_017
4	2399567090	146576201	5531422	MSD4	MSD4_017
5	2399569031	146576201	5531422	MSD4	MSD4_017
6	2399568370	146576201	5531422	MSD4	MSD4_017
7	2399568391	146576201	5531422	MSD4	MSD4_017
8	2399567282	146576201	5531422	MSD4	MSD4_017
9	2399567346	146576201	5531422	MSD4	MSD4_017
10	2399567367	146576201	5531422	MSD4	MSD4_017
11	2399568690	146576201	5531422	MSD4	MSD4_017
12	2399566641	146576202	5531422	MSD4	MSD4_018
13	2399567089	146576202	5531422	MSD4	MSD4_018
14	2399569032	146576202	5531422	MSD4	MSD4_018
15	2399567240	146576202	5531422	MSD4	MSD4_018

Notes

The example above relates to the full Postgres database backup. The same instructions apply for a restore of the 'Full Flat Pivot' dataset too, changing the database and file names as required.

The command "CREATE SCHEMA public;" will error if public schema already exists - as seen in the screenshot of the restore. This schema is created by default when you make a database. The execution as seen in the screenshot continues regardless of the error and results in a fully formed database.

Indexes

Depending on your requirements, creating the following indexes may be useful.

Table	Column
gswa_dh_assay	anumber
gswa_dh_assay	companyholeid
gswa_dh_assay	dhgeochemid
gswa_dh_collar	anumber
gswa_dh_collar	company
gswa_dh_dhscollarsurvey	collarid
gswa_dh_dhssurvey	collarid
gswa_dh_nassay	aid
gswa_dh_nassay	anumber
gswa_ss_assay	surfacesampleid
gswa_ss_assay	anumber
gswa_ss_nassay	surfacesampleid
gswa_ss_sample	id
gswa_ss_sample	anumber
gswa_ss_sample	company
gswa_dh_alteration	collarid
gswa_dh_event	collarid
gswa_dh_geotech	collarid
gswa_dh_hyperspectral	collarid
gswa_dh_magsus	collarid
gswa_dh_mineralogy	collarid
gswa_dh_pxrreadings	collarid
gswa_dh_recovery	collarid
gswa_dh_regolith	collarid
gswa_dh_specgrav	collarid
gswa_dh_survey	collarid
gswa_dh_veining	collarid
gswa_dh_water	collarid
gswa_dh_weathering	collarid

Postgres WA Geochemistry Database Restoration via pg_restore

Steps to restore the PostgreSQL backup:

1: Create a new database to load the backup into

2: Reload the backup:

- Using pg_restore on the command line: `pg_restore -d "YOURDBNAME" -U postgres -f "Backup file path"`

- Using pgAdmin: Right click on the database where you would like to load the data and select the restore option. Simply indicate the filename and press the restore button.

The pg_restore command provides flexibility in allowing you to select objects that you wish to restore.

See the postgres documentation here:

<https://www.postgresql.org/docs/current/app-pgrestore.html>