

# Open source GIS Data Model template backup/restore overview

- Install PostgreSQL 10+, compatible PostGIS and PG Admin 4, if not already installed. Installation files (“pre-built binary packages”) are available from <https://www.postgresql.org/download/>
- Create a new database with spatial extensions for import / export.
- If generating an export file, create the GIS Data Model template tables in a new schema. Load data into them, which will be a site-specific process. Then generate a backup file from the template.
- If importing data from another site, restore from the backup file into the database created for this purpose.
- Step-by-step instructions provided below.

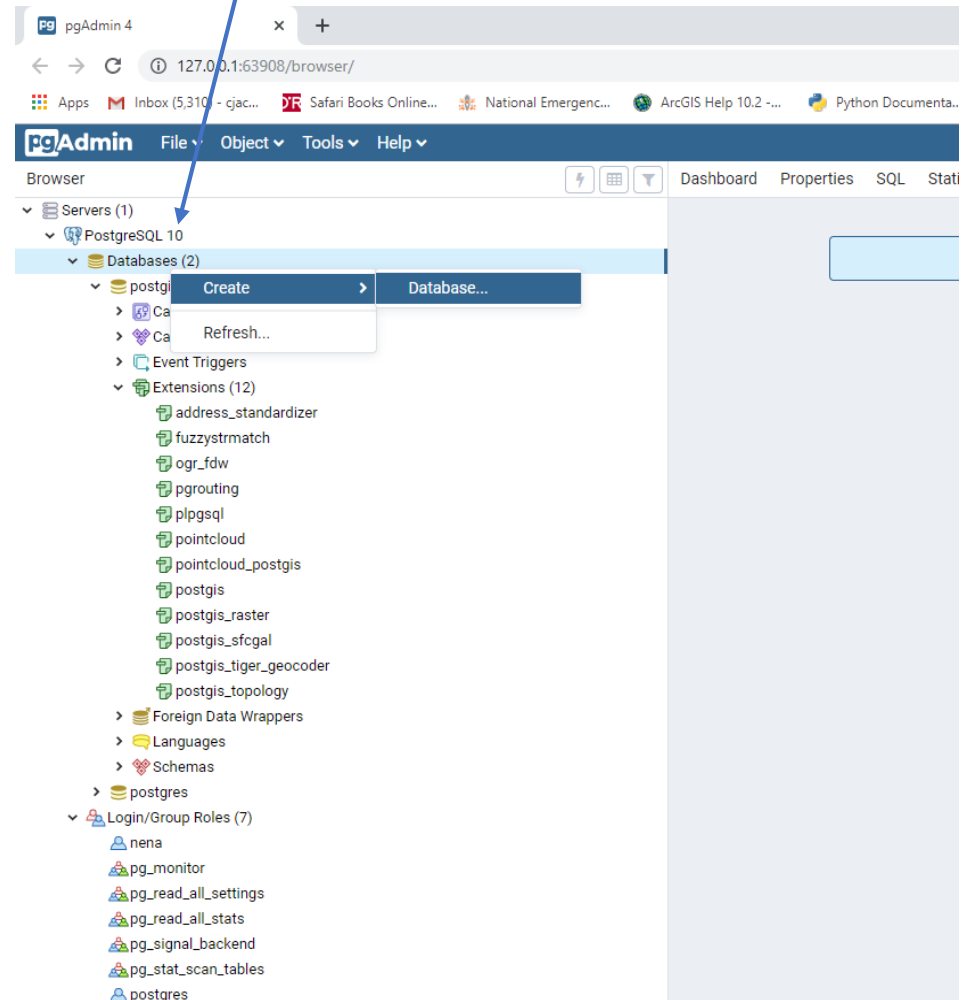
# Step 1: Create the Database

We're assuming a Postgres 10 or later environment with PostGIS and PGAdmin 4 installed and connections to one or more servers.

The first step is to create a database for the NENA GIS Data Model compliant import/export.

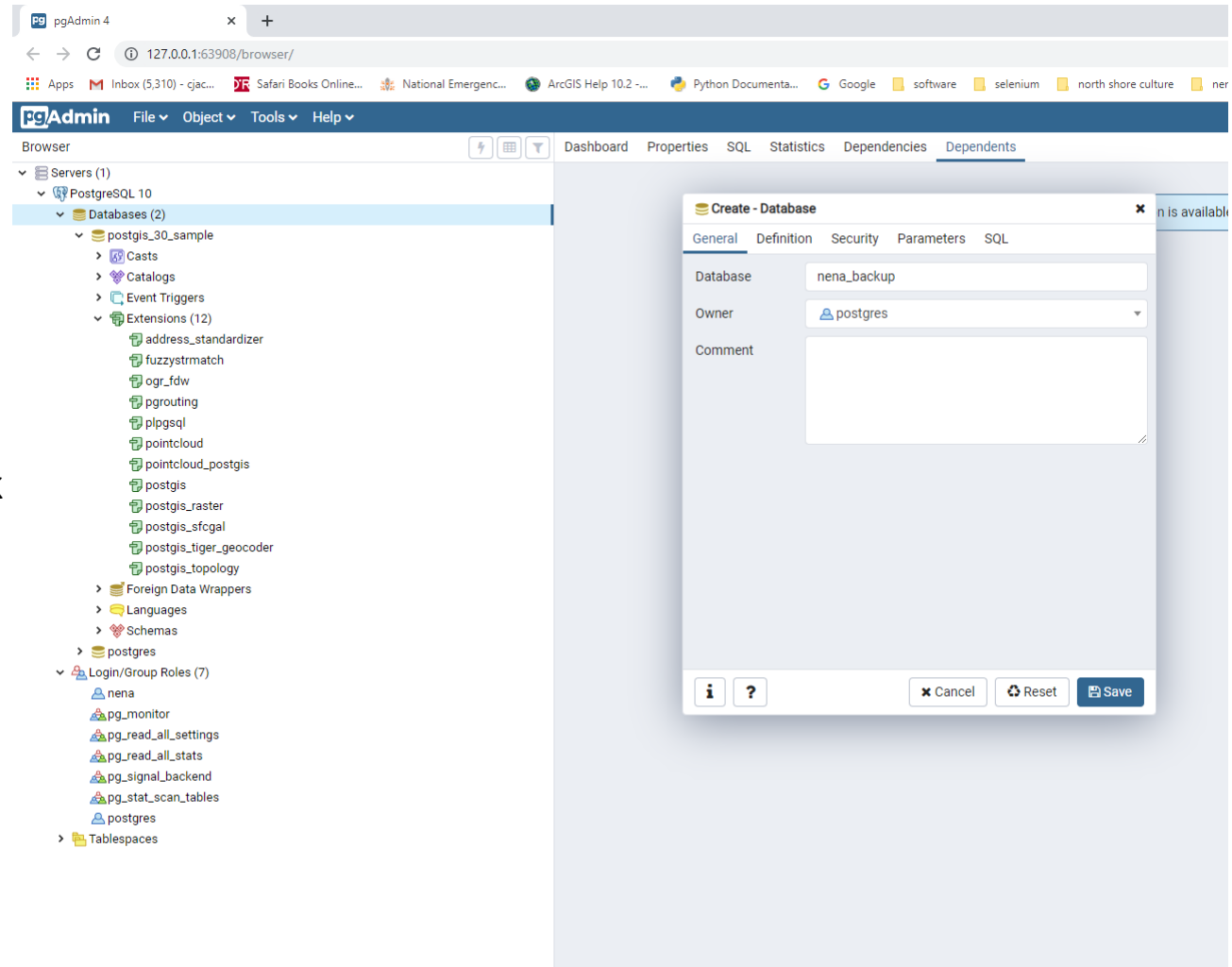
Under Server(s), click on the active connection, in this case PostgreSQL 10, right click on Databases, choose Create and then Database.

Server: PostgreSQL 10



# Name the Database and Save

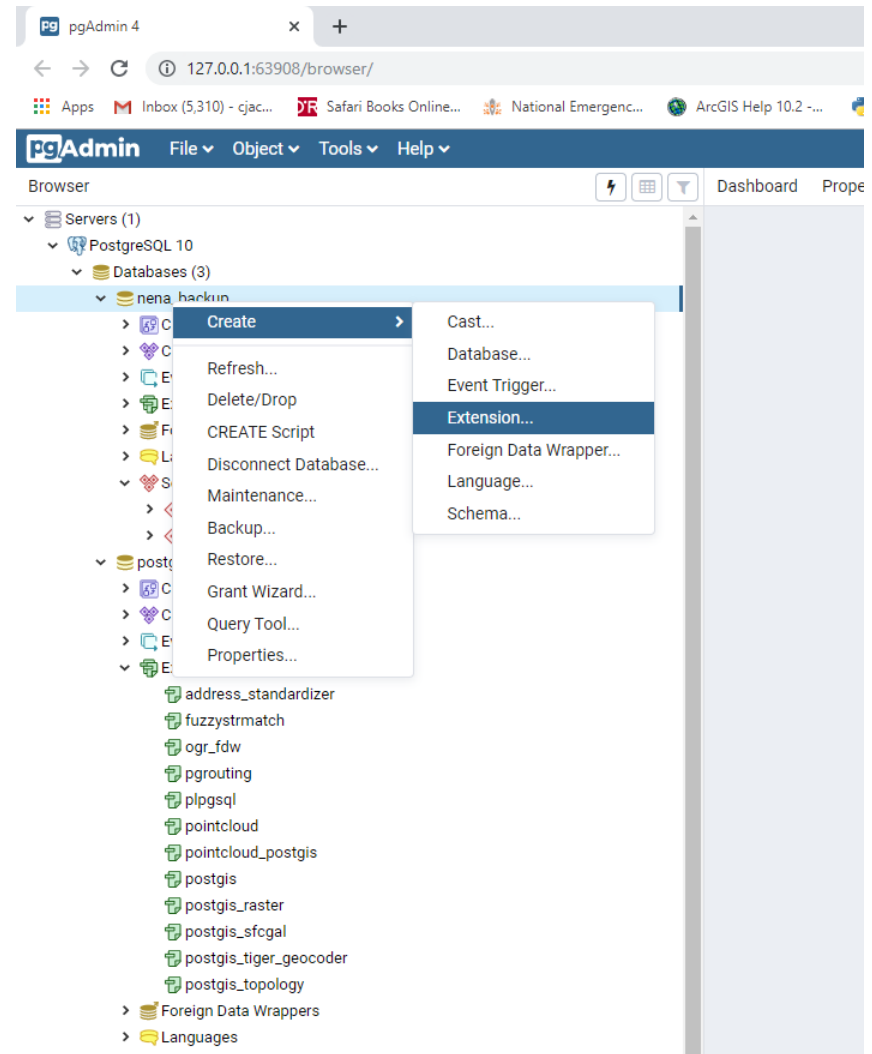
Fill in the dialog to create the database, in this case it will be called “nena\_backup,” and click Save



# Step 2: Add Spatial Extensions

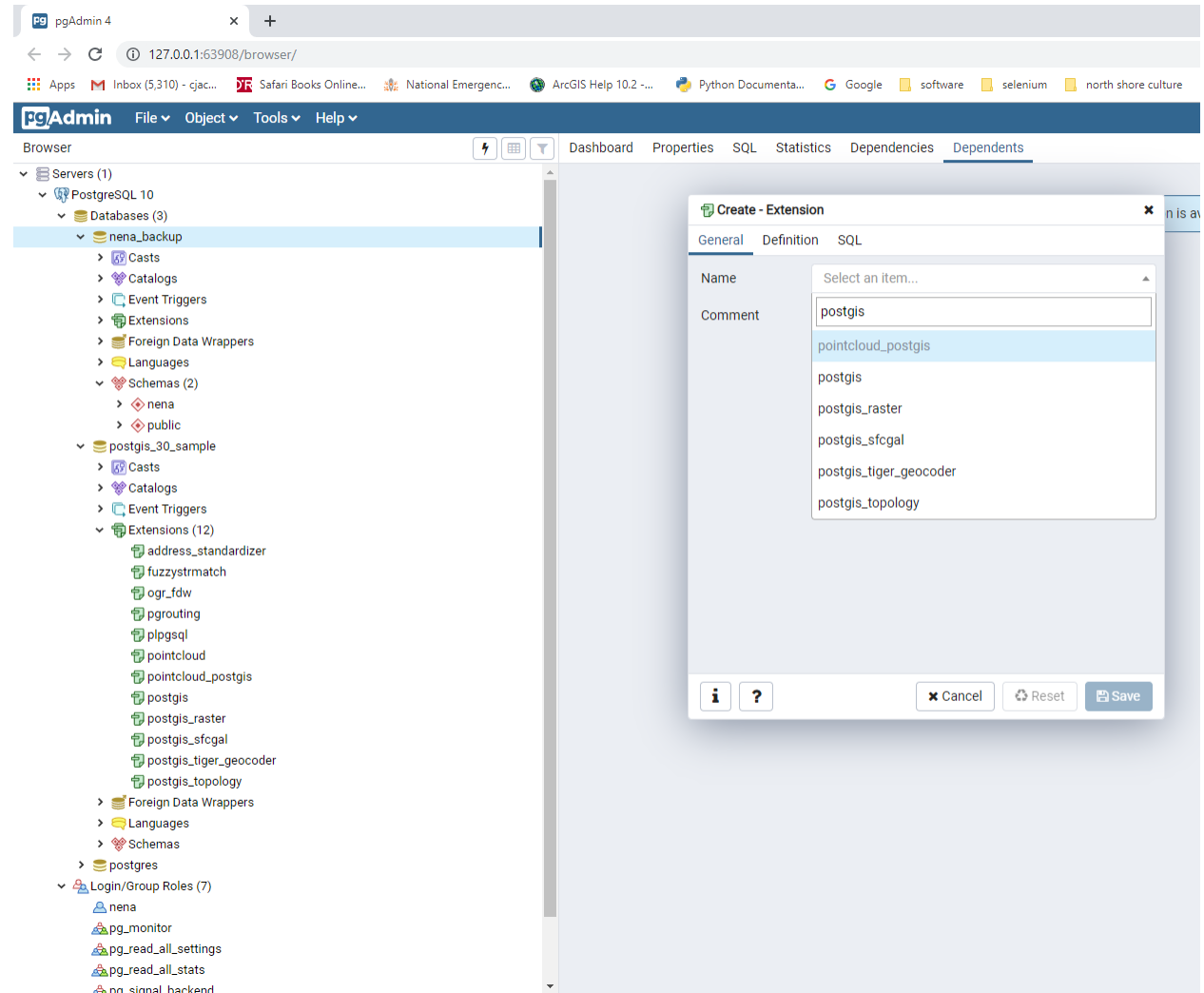
The next step is to add the minimum extensions needed to handle spatial data.

Right click on the database again, then choose Create and Extension.



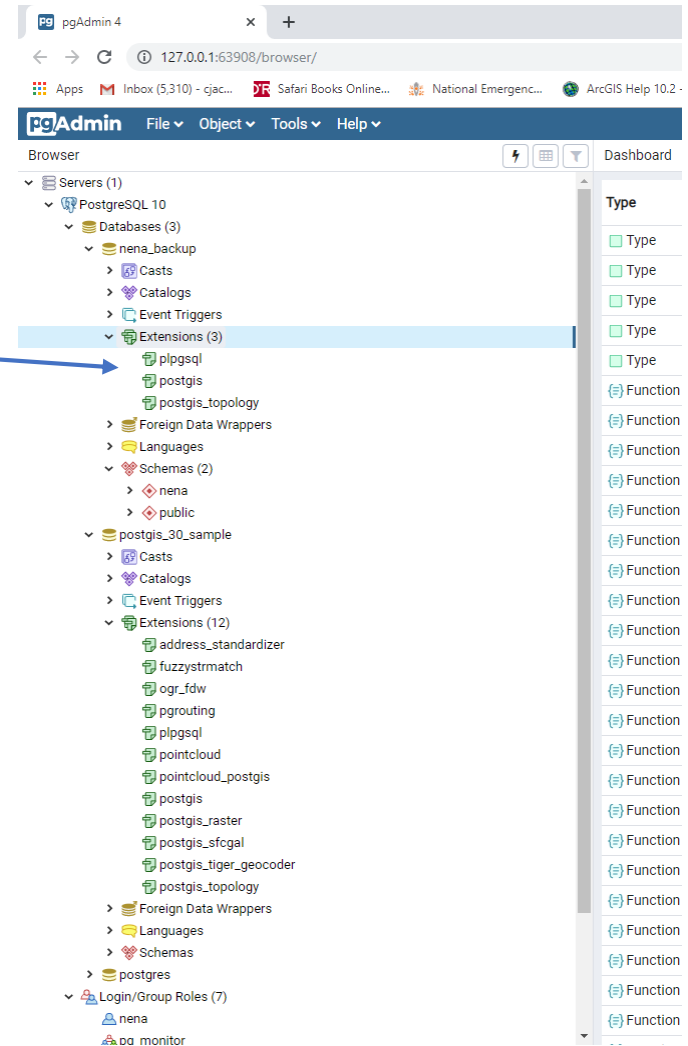
# Add Spatial Extensions (2)

Pick from the listing the “postgis” extension and then Save. Repeat for “postgis\_topology.”



# Check Extensions

Check that extensions are listed  
for nena backup database



*Next steps depend on whether you are exporting or importing data.*

*If you are exporting data:*

If exporting data, then generate the NENA GIS Data Model template tables as detailed in Steps 3 & 4 below.

The process to populate these template tables will be specific to each site, and beyond the scope of this guidance. It will likely require the development of custom ETL scripts.

After the template is loaded, a backup file of the “nena” schema can be generated for transfer to another site. The “Create Backup” slides provide instructions for that operation.

*If you are importing data:*

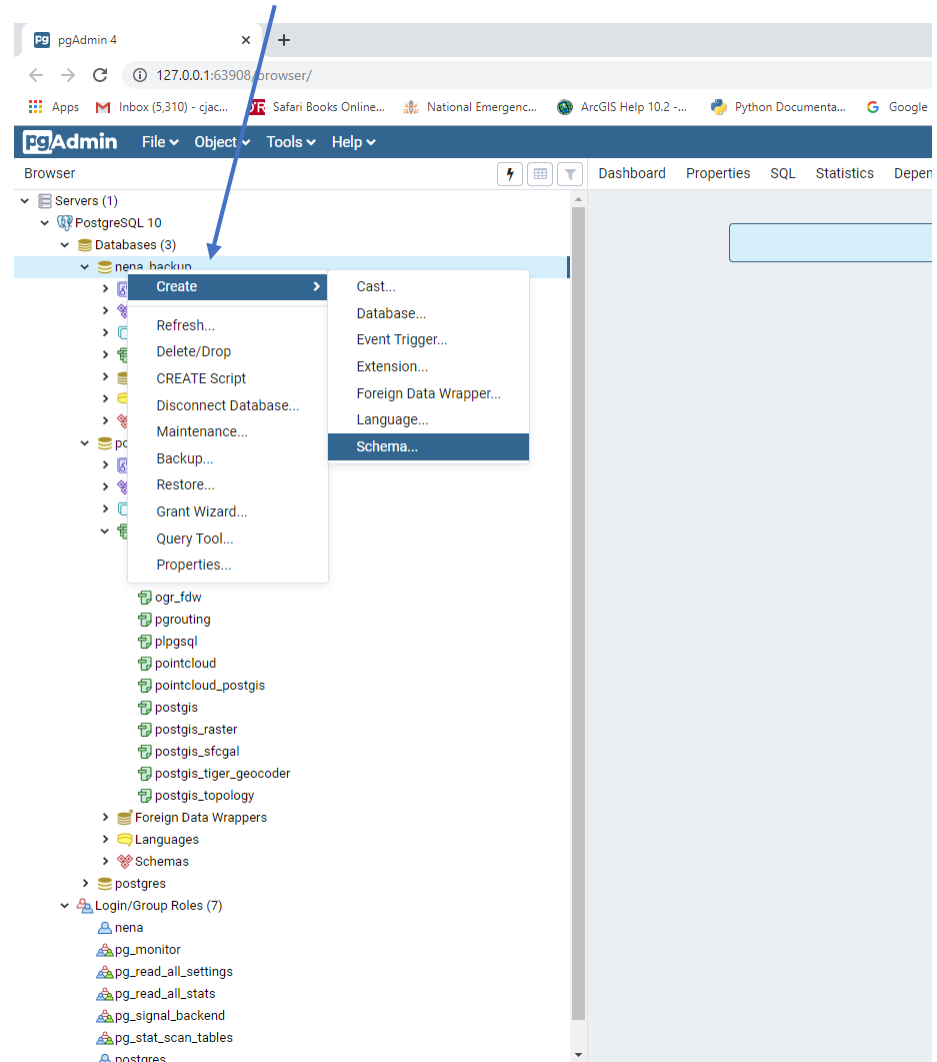
If a backup file is received, then it should be restored into the new database created in steps 1-2. The restore operation is detailed in the “Restore from Backup” slides at the end.

# Step 3: Create a Schema

Newly created database: Nena\_backup

Next, create a schema for the data to export.

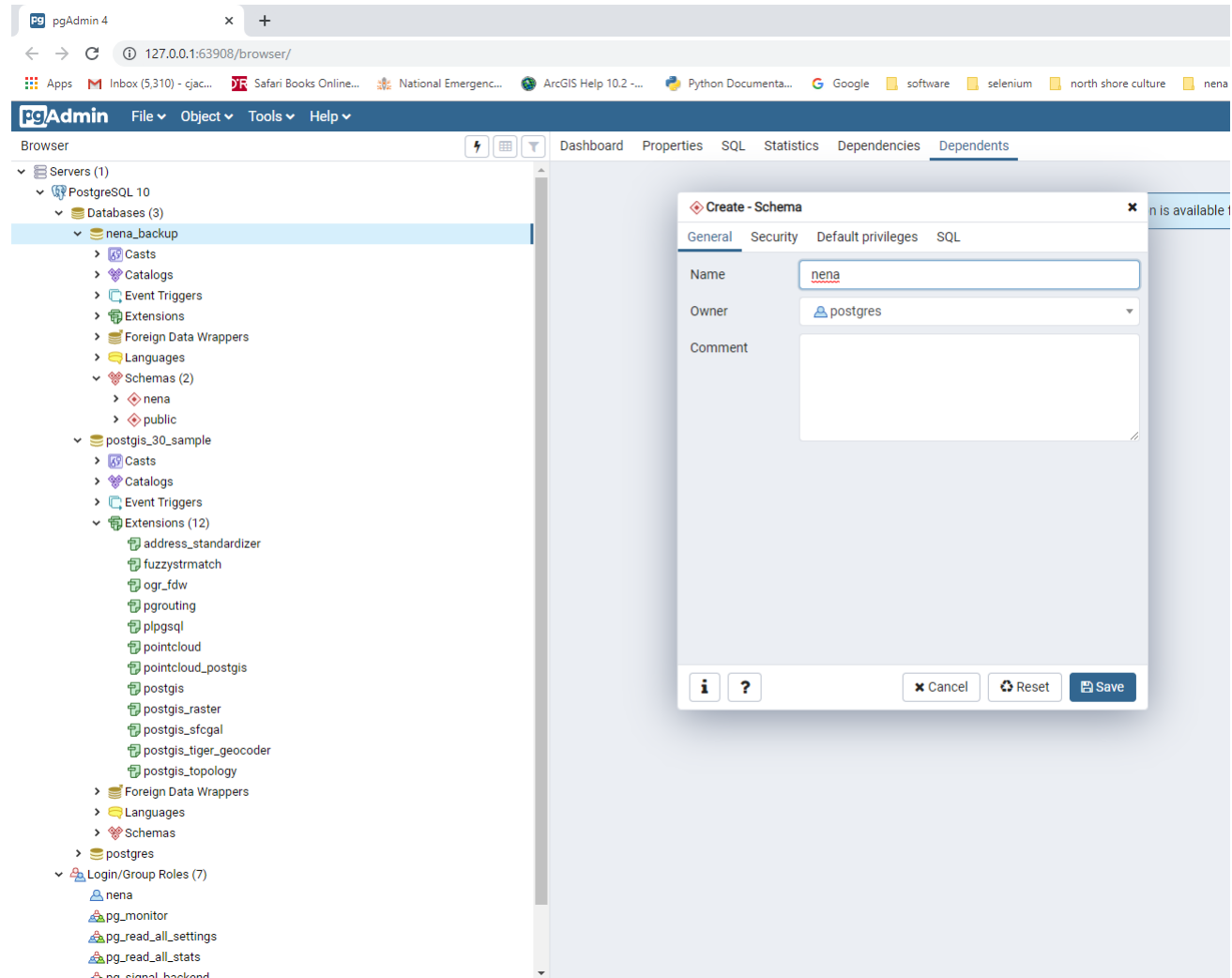
Under Databases, right click on the database that you have just created, choose Create and then Schema.





# Name the Schema

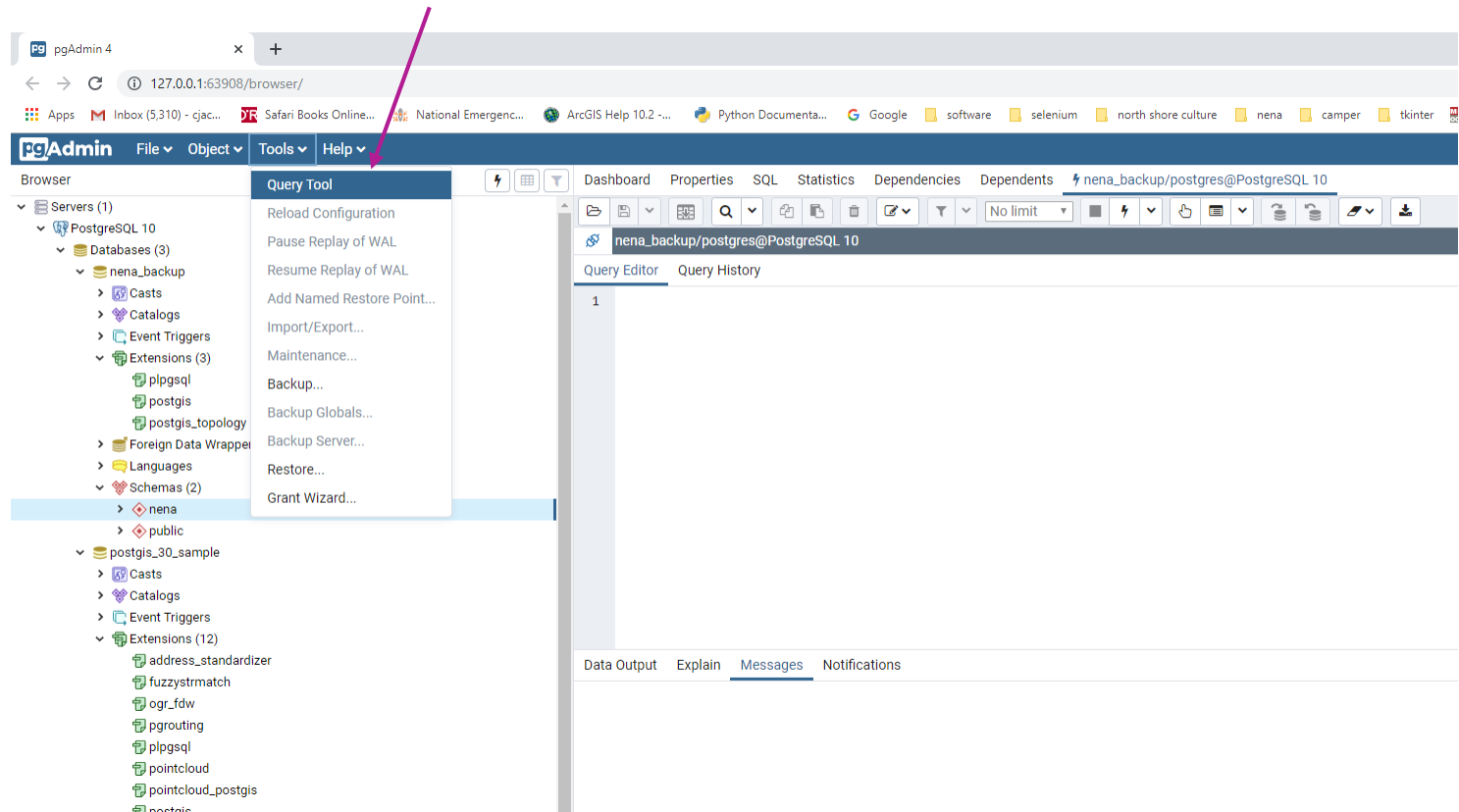
Fill in the dialog to create the schema, in this case it will be called “nena” and click Save



## Step 4: Open Query Tool to Create Template

If exporting data, we need to create the tables for the GIS Data Model template.

Open the Query Tool, under the Tools tab



# Open Template Table Creation Script

Click on file folder icon and browse to file location

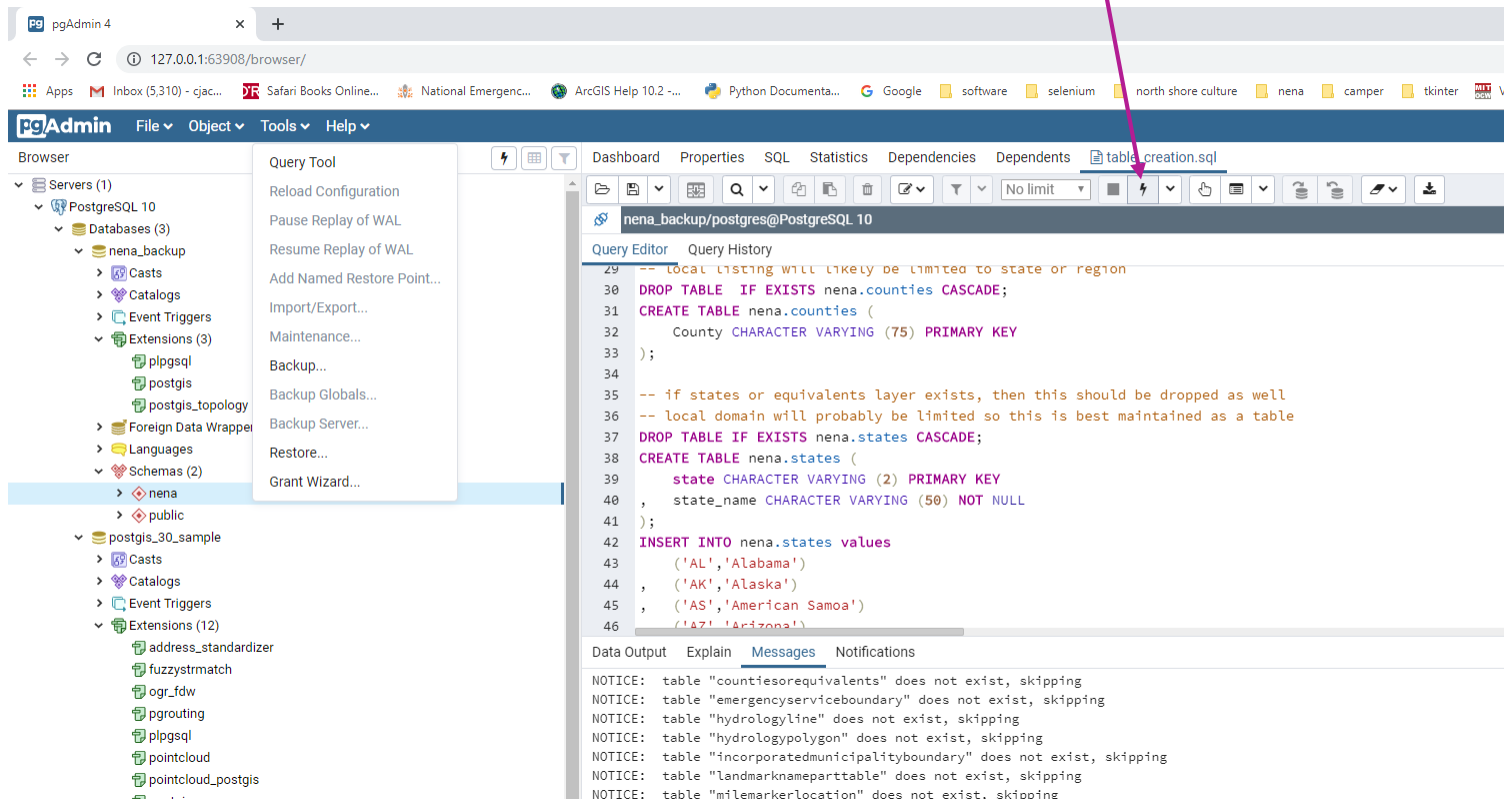
The screenshot shows the pgAdmin 4 web interface. On the left, the 'Browser' pane displays a tree structure of databases and schemas. The 'nena' database is selected. A right-click context menu is open over the 'nena' database, showing options like 'Query Tool', 'Reload Configuration', 'Pause Replay of WAL', 'Resume Replay of WAL', 'Add Named Restore Point...', 'Import/Export...', 'Maintenance...', 'Backup...', 'Backup Globals...', 'Backup Server...', 'Restore...', and 'Grant Wizard...'. A pink arrow points from the text 'Click on file folder icon and browse to file location' to the file folder icon in the top toolbar of the 'Query Editor' pane. The 'Query Editor' pane shows a single query line. A 'Select file' dialog box is open, displaying a list of files and folders. The dialog box has a text field with the path 'ers\cjacq\OneDrive\Documents\orgsProgs\nena\template\'. The list of files and folders is as follows:

Name	Size	Modified
after_db_create.sql	545.0 B	Mon Oct 28 17:38:58 2019
after_pg_install.sql	81.0 B	Mon Oct 28 17:39:51 2019
create_data_model_template.sql	12.5 KB	Fri Jul 12 10:57:01 2019
create_dm_template_v2.sql	41.8 KB	Tue Aug 20 18:52:12 2019
create_dm_template_v2_backup.sql	11.1 KB	Mon Jul 15 15:01:37 2019
create_dm_template_working_orig.sql	40.6 KB	Tue Aug 20 19:48:08 2019
dm_table_creation.sql	15.9 KB	Wed Jul 17 17:43:03 2019
http%3a%2f%2fcywin.mirror.constant.com%2f	0.0 B	Mon Sep 23 21:28:13 2019
nena_20191107	0.0 B	Thu Nov 7 15:17:02 2019
New Folder	0.0 B	Mon Nov 4 10:56:47 2019
ng911_gis_20190723.sql	95.1 KB	Mon Nov 4 11:57:33 2019
table_creation.sql	41.6 KB	Mon Oct 28 18:00:11 2019

The dialog box also has a 'Format' dropdown set to 'sql' and buttons for 'Cancel' and 'Select'.

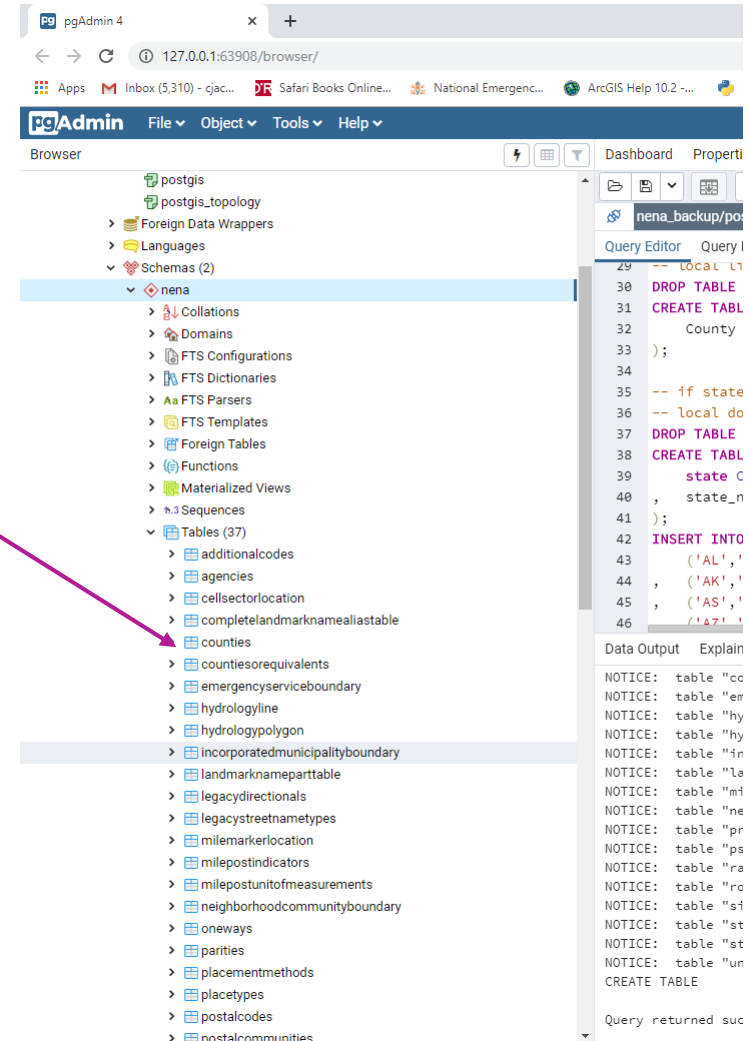
# Run Table Creation Script

Run the table creation script (in this case, “table\_creation.sql”) by clicking on lightning icon



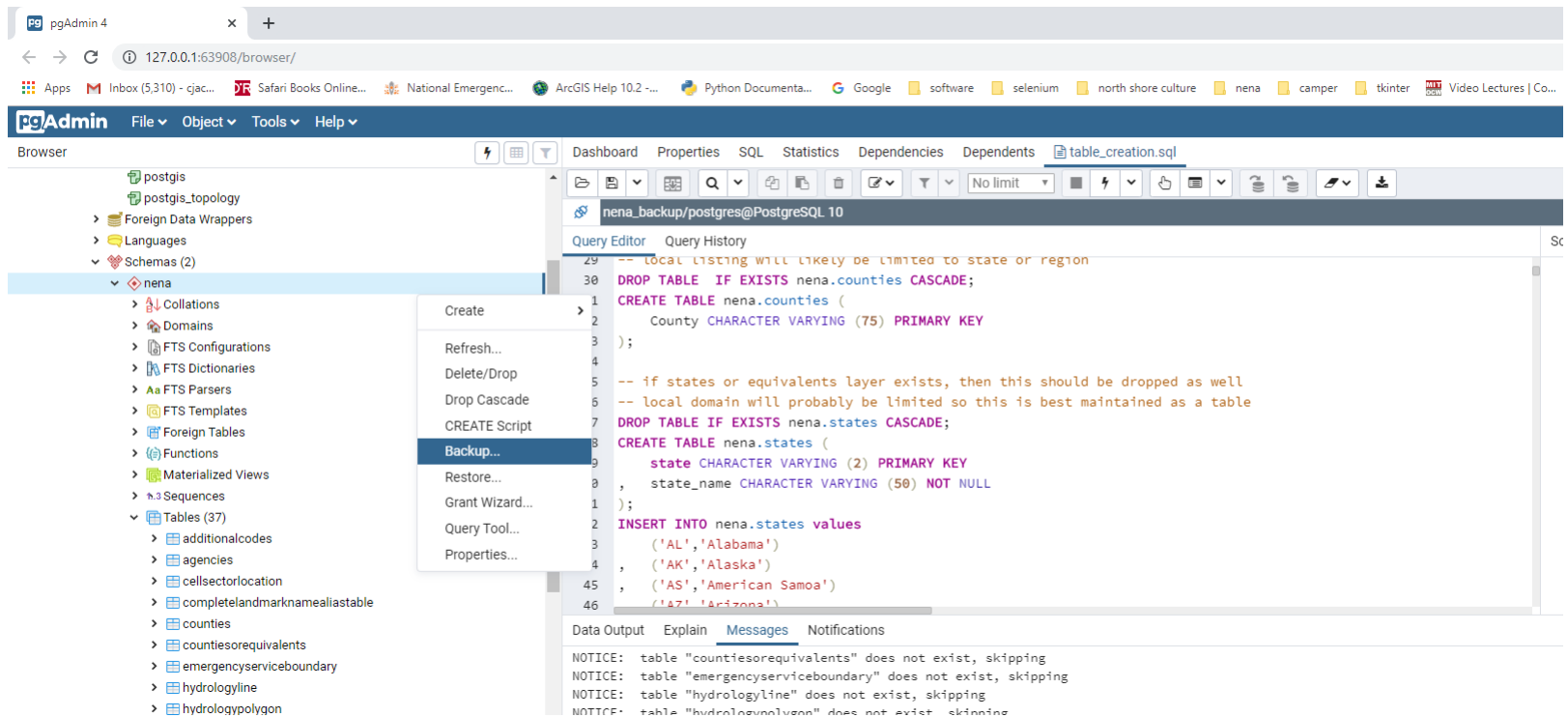
# After Running Table Creation Script

Data Model tables will be listed  
under nena schema



# Create Backup to Transfer Data (1)

Highlight the nena schema, right click and choose Backup.



# Create Backup to Transfer Data (2)

Fill in complete file path for the backup file and click Backup Confirmation should appear in lower right as shown below

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Schemas (2)' tree is expanded, showing the 'nena' schema. The main pane shows the 'Backup (Schema: nena)' dialog box. The 'General' tab is active, and the 'Filename' field is filled with 'C:\Users\progsProgs\nena\template\nena\_20191030.backup'. The 'Format' is set to 'Custom'. The 'Backup' button is highlighted. Below the dialog, the 'Messages' tab is selected, showing a list of notices indicating that various tables do not exist and are being skipped. At the bottom right, a notification box titled 'Backing up an object on the server' shows the operation completed successfully in 0.38 seconds.

pgAdmin 4

127.0.0.1:63908/browser/

Apps | Inbox (5,310) - cja... | Safari Books Online... | National Emergenc... | ArcGIS Help 10.2 ... | Python Documenta... | Google | software | selenium | north shore culture | nena | camper | tkinter | Video Lectures | Co...

pgAdmin | File | Object | Tools | Help

Browser

- postgis
- postgis\_topology
- Foreign Data Wrappers
- Languages
- Schemas (2)
  - nena
    - Collations
    - Domains
    - FTS Configurations
    - FTS Dictionaries
    - FTS Parsers
    - FTS Templates
    - Foreign Tables
    - Functions
    - Materialized Views
    - Sequences
    - Tables (37)
      - additionalcodes
      - agencies
      - cellsectorlocation
      - completelandmarknamealiastable
      - counties
      - countiesorequivalents
      - emergencyserviceboundary
      - hydrologyline
      - hydrologypolygon
      - incorporatedmunicipalityboundary
      - landmarknameparttable
      - legacydirectionals
      - legacystreetnametypes
      - milemarkerlocation
      - milepostindicators
      - milepostunitofmeasurements
      - neighborhoodcommunityboundary
      - oneways
      - parities
      - placementmethods
      - placetypes
      - postalcodes
      - postalcommunities

Dashboard | Properties | SQL | Statistics | Dependencies | Dependents | table\_creation.sql

Backup (Schema: nena)

General | Dump options

Filename: C:\Users\progsProgs\nena\template\nena\_20191030.backup

Format: Custom

Compression ratio:

Encoding: Select an item...

Number of jobs:

Role name: Select an item...

Cancel | Backup

Data Output | Explain | Messages | Notifications

NOTICE: table "countiesorequivalents" does not exist, skipping  
NOTICE: table "emergencyserviceboundary" does not exist, skipping  
NOTICE: table "hydrologyline" does not exist, skipping  
NOTICE: table "hydrologypolygon" does not exist, skipping  
NOTICE: table "incorporatedmunicipalityboundary" does not exist, skipping  
NOTICE: table "landmarknameparttable" does not exist, skipping  
NOTICE: table "milemarkerlocation" does not exist, skipping  
NOTICE: table "neighborhoodcommunityboundary" does not exist, skipping  
NOTICE: table "provisioningboundary" does not exist, skipping  
NOTICE: table "psap\_boundary" does not exist, skipping  
NOTICE: table "railroadcenterlines" does not exist, skipping  
NOTICE: table "roadcenterlines" does not exist, skipping  
NOTICE: table "sitestructureaddresspoints" does not exist, skipping  
NOTICE: table "statesorequivalents" does not exist, skipping  
NOTICE: table "streetnamealiastable" does not exist, skipping  
NOTICE: table "unincorporatedcommunityboundary" does not exist, skipping  
CREATE TABLE

Query returned successfully in 276 msec.

Backing up an object on the server

Backing up an object on the server 'PostgreSQL 10 (localhost:5432)' from database 'nena\_backup'

Fri Nov 15 2019 09:15:50 GMT-0500 (Eastern Standard Time)

0.38 seconds | More details... | Stop Process

Successfully completed.

# Restore from Backup

Create a new database, do not use the same one as used for backup operation. With restore database highlighted, right click and pick Restore, browse to the backup file, highlight and click Select. Confirmation should appear in lower right as shown below.

The screenshot displays the pgAdmin 4 interface with the 'Restore (Database: nena\_backup)' dialog box open. The 'General' tab is selected, showing the following options:

- Format: Custom or tar
- Filename: (empty)
- Number of jobs: 1
- Role name: Select an item...

A file selection dialog is open, showing a list of backup files. The file 'nena\_20191115.backup' is selected.

The 'Data Output' tab shows a list of tables being restored, including 'countiesorequivalents', 'emergencyseviceboundary', 'hydrologyline', 'hydrologypolygon', 'incorporatedmunicipalityboundary', 'landmarknameparttable', 'legacydirectionals', and 'legacystreetnametypes'.

A confirmation message 'Successfully completed.' is visible in the bottom right corner.