Here’s an updated step-by-step guide for using pgAdmin 4’s GUI to back up your entire database (schema + data) with full control and precision.

1. Launch the Backup Dialog

- In pgAdmin 4, navigate to your server → Databases.
- Right-click the target database → select Backup…
- The Backup dialog appears, titled “Backup - your_database”

2. General Tab Settings

- Filename: Click the Browse (…) button to specify the file path, e.g., C:\backup\mydb.backup or /backups/mydb.backup.
- Format: Choose:
  - Custom – compressed, flexible, selective restore.
  - Directory – multi-file, parallelizable.

  Avoid Plain as it’s not restorable via pgAdmin.
- Compression ratio: (Custom/Directory only) – choose 0–9; default ~6.
- Encoding: Typically set to server encoding (e.g. UTF8).
- Rolename: Choose the role that will own the backup objects (optional for proper ownership).

3. Data/Objects Tab

- Sections:
  - Pre-data (schema definitions)
  - Data (tables, sequences)
  - Post-data (indexes, constraints, triggers)
- Type of objects: By default, include both schemas and data. Only toggle if you want one or the other.

- Do not save toggles:

  - E.g., Owner, Privileges — leave these OFF to preserve ownership, GRANTs, comments, tablespaces, etc.

4. Table Options (Optional)

- Found under Table Options tab:

  - Use column inserts – includes explicit column names (safe but slower).

  - Load via partition root – useful for partitioned tables.

5. Query Options Tab

- Include CREATE DATABASE statement – ON (to restore the DB itself).

- Include DROP DATABASE – ON (adds DROP DATABASE IF EXISTS).

- Include IF EXISTS – ON (safe dropping).

- Use INSERT commands / Use Column INSERTS – typically OFF for speed (COPY is faster), but enable if you want purely portable SQL.

6. Options Tab

- Disable toggles: leave defaults (things like disabling triggers, etc.).

- You typically don’t disable triggers—keep defaults for consistency.

7. Execute Backup

- Click Backup (bottom right).

- A progress dialog appears; a success message confirms completion.

- Verify the backup file exists and is non-zero in size.

8. Test Your Backup

- Always validate by restoring it:

1. Create a new empty database (can be temporary).

2. Right-click → Restore…

3. Format: match the backup file (e.g. Custom).

4. Enable on Restore dialog:

   - Clean before restore – ON (drops existing objects).

   - Single transaction – ON (rolls back fully on failure).

5. Select the backup file and click Restore.

6. Monitor the Logs and Processes tab for errors.

If it restores cleanly and everything works, your backup is reliable.

9. Best Practices Recap

Task | Recommendation

-----|----------

Format | Custom (default compressed)

Include statements | CREATE/DROP DB, IF EXISTS

Sections | Pre-data, Data, Post-data

Test restore | Must restore on a test DB

Scheduling | Use OS cron or Task Scheduler

Secure copies | Store backups offsite and timestamp them

Automation | Consider scripting pg_dump for recurring backups

Following these exact selections ensures you fully capture every model (schema, roles, ownership) and data row. Restoration on another system or newly installed server will be robust, consistent, and error-free.

Let me know if you’d like help automating this via CLI or scheduling on your OS!