

WEEK 02

Instructor: Yanan Wu

TA: Vanchy Li

Spring 2025



WEEK 02

LECTURE SESSION

Instructor: Yanan Wu

TA: Vanchy Li

Spring 2025

2.1.1

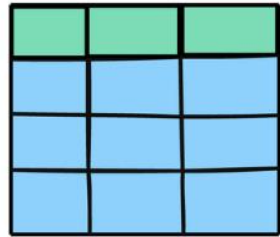
DATABASE AND DBMS

DATABASE

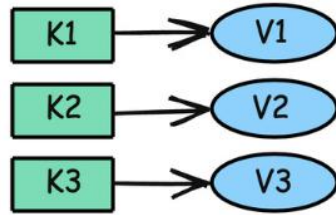
- A database is an organized collection of structured data, or information, typically stored electronically in a computer that can be easily accessed, managed, and updated



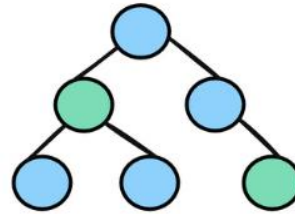
TYPES OF DATABASE



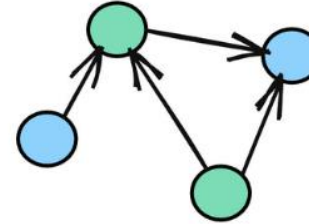
Relational



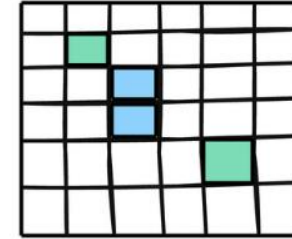
Key-Value



Document

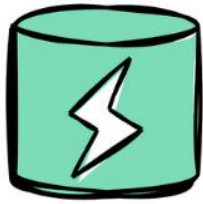


Graph

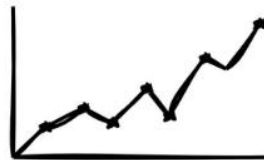


Wide-Column

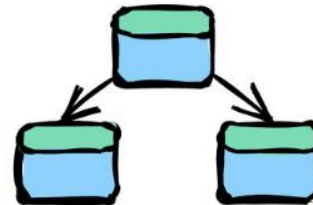
blog.algomaster.io



In-Memory



Time-Series



Object-Oriented



Text-Search



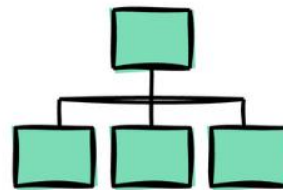
Spatial



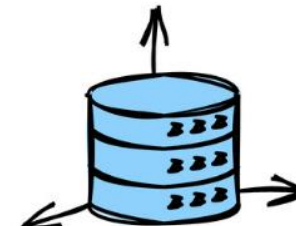
Blob



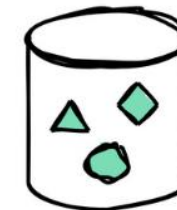
Ledger



Hierarchical



Vector



Embedded

DATABASE MANAGEMENT SYSTEM (DBMS)

- A software system that is designed to manage and organize data in a structured manner.

Key features of DBMS



Store large amounts of data systematically



Implements access control to protect data from unauthorized access



Ensure the consistency of data through constraints



Provide tools to back up the database

OPEN-SOURCE TECHNOLOGIES

- MySQL



- PostgreSQL



- MongoDB



- SQLite



- DynamoDB



- MariaDB





KEY ATTRIBUTES OF A ROBUST DBMS

- Reliability
 - The system can function correctly and consistently over time
 - The system must be able to offer continual uninterrupted service
- Consistency
 - Refers to ensuring that the database remains accurate, valid, and follows defined rules or constraints
- Technology-Proof
 - The system is adapted to evolving technologies and its long-term viability

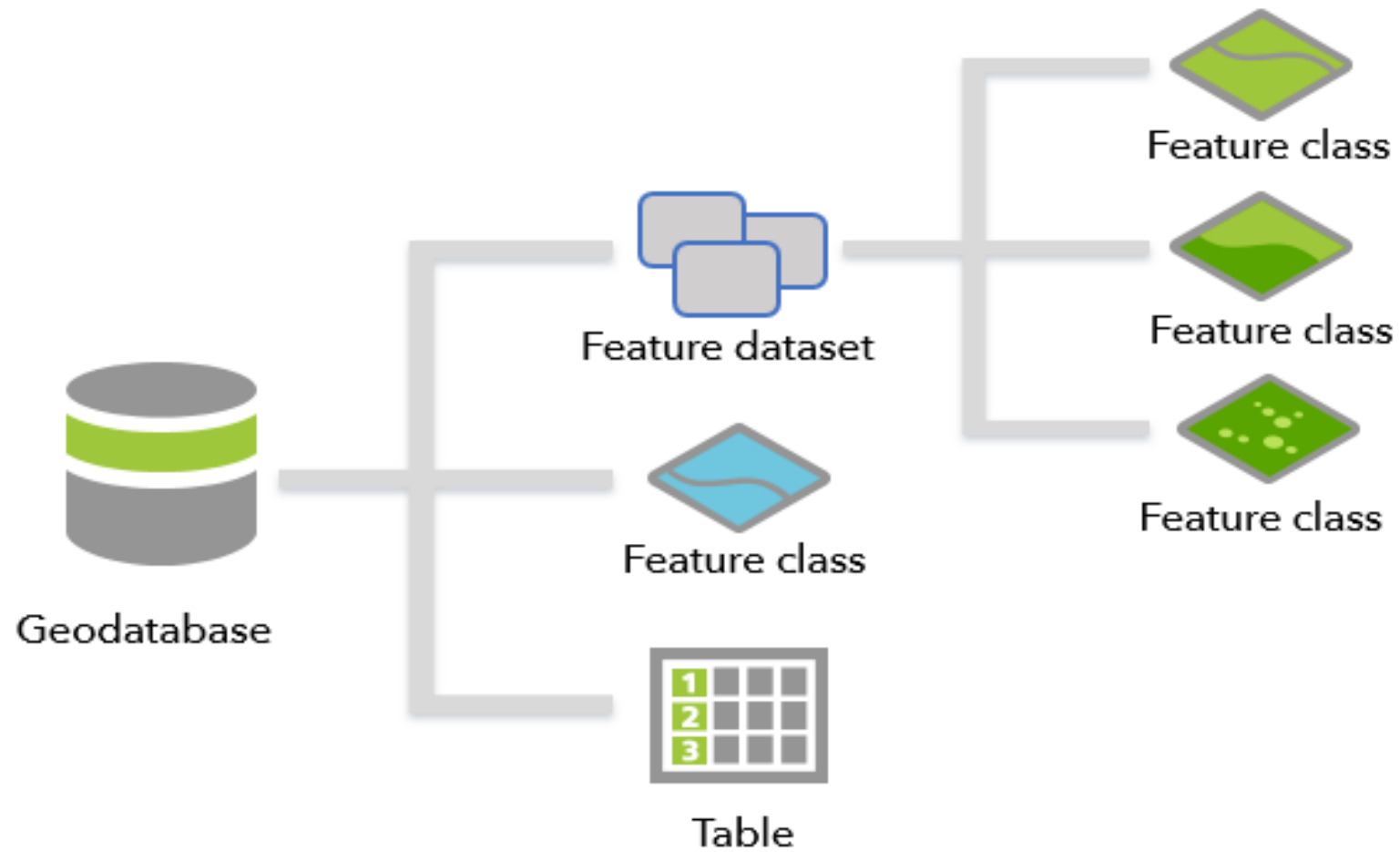
2.1.2 GEODATABASE



GEODATABASE

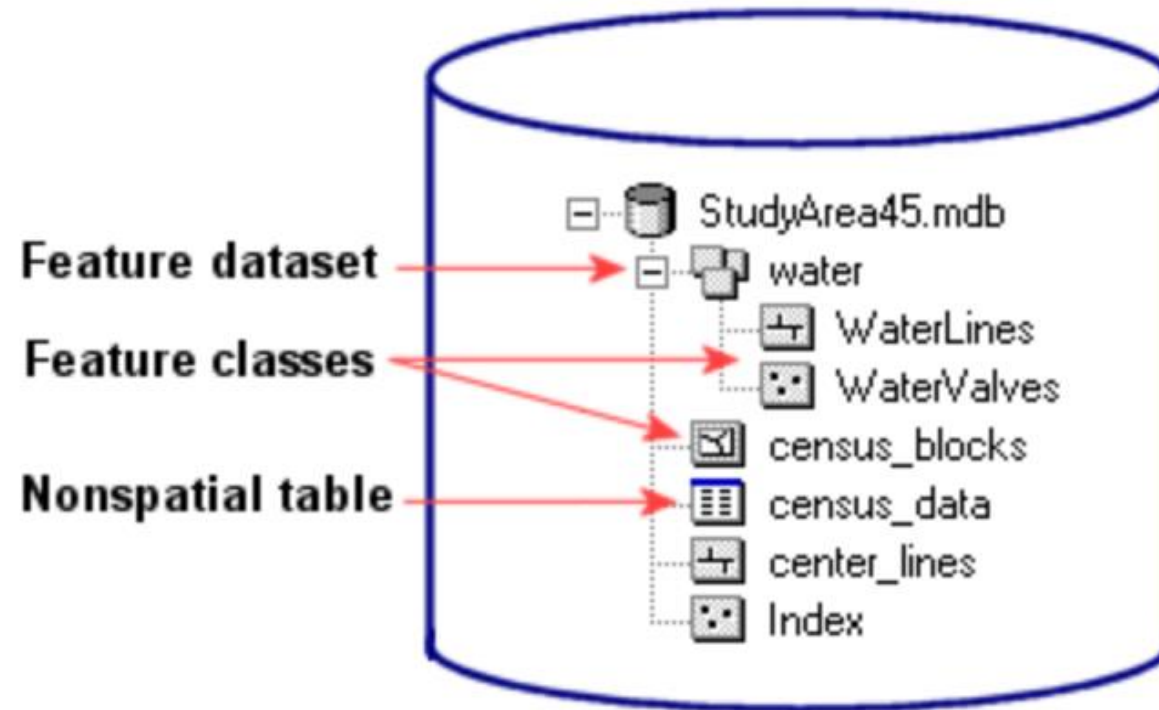
- A **Geodatabase** is a specialized type of database designed to store, manage, and analyze spatial and geographic data.
- It serves as a comprehensive framework for organizing spatial data and associated attribute information, allowing users to perform advanced geographic analysis, mapping, and data management tasks.

GEODATABASE IN GIS



FEATURE DATASETS AND FEATURE CLASS

- A feature dataset is composed of feature class
- A feature class is a collection of geographic features with the same geometry type, attributes, and spatial reference.

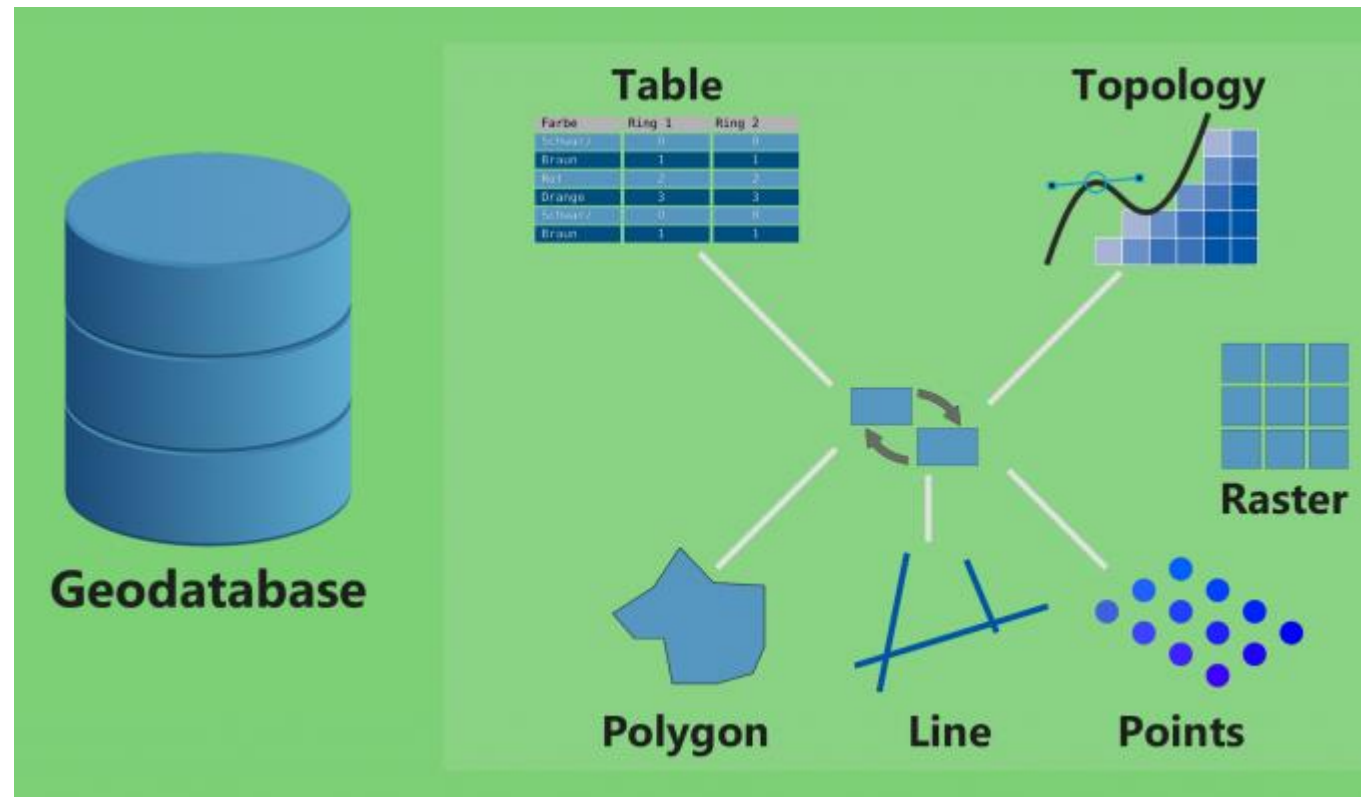


ADVANTAGES OF USING A GEODATABASE

- Centralized Data Management

Combines spatial (maps, shapes) and non-spatial (tabular) data in a single database.

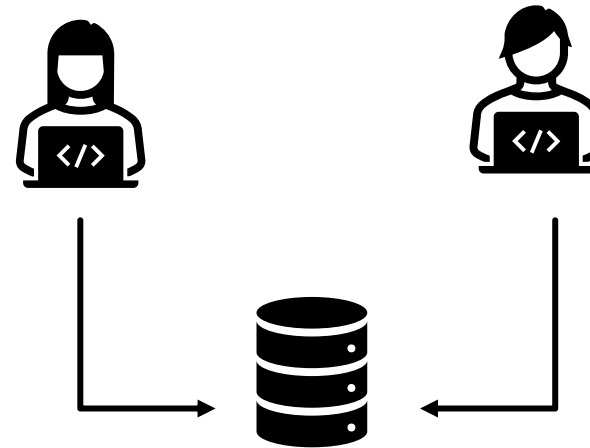
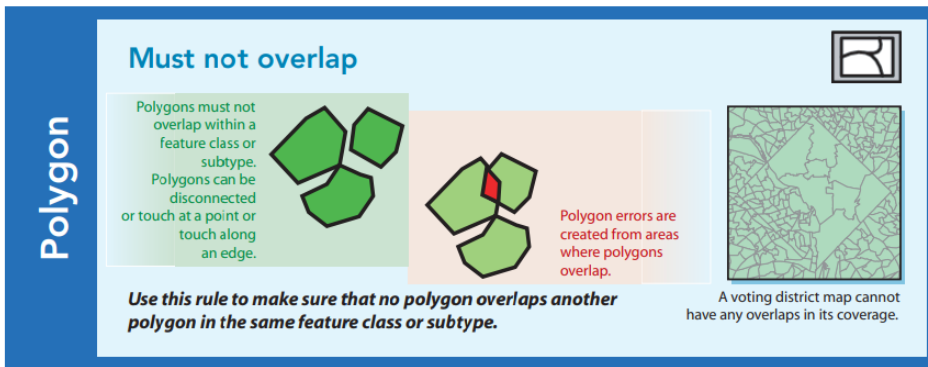
Provides a unified framework for managing diverse data types, including vector, raster, and attribute data.



ADVANTAGES OF USING A GEODATABASE

■ Data Integrity and Consistency

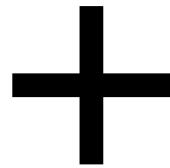
- Supports [topology rules](#) to ensure data accuracy (e.g., no overlapping polygons or gaps).
- Enforces data validation and constraints to maintain reliable relationships between features.
- Versioning support allows multiple users to work on the same dataset without conflicts.



ADVANTAGES OF USING A GEODATABASE

- Cross-Platform Compatibility

- Compatible with various GIS software tools, such as ArcGIS, QGIS, and GeoServer.
- Works with relational database systems like PostgreSQL/PostGIS, Oracle, or SQL Server.



ADVANTAGES OF USING A GEODATABASE

- Scalability

- Supports various levels of implementation:

- ❑ File Geodatabases: A file geodatabase is stored as multiple files in a folder with a `.gdb` extension. For team collaboration on medium-scale projects.
 - ❑ Enterprise Geodatabases: Also known as multiuser geodatabases, enterprise geodatabases are stored in relational databases. They can be virtually unlimited in size and number of users; the limits differ depending on the database management system (DBMS) vendor. For large organizations and multi-user environments.
 - ❑ Mobile Geodatabases: A mobile geodatabase is stored in an SQLite database that is entirely contained in a single file and has a `.geodatabase` extension.

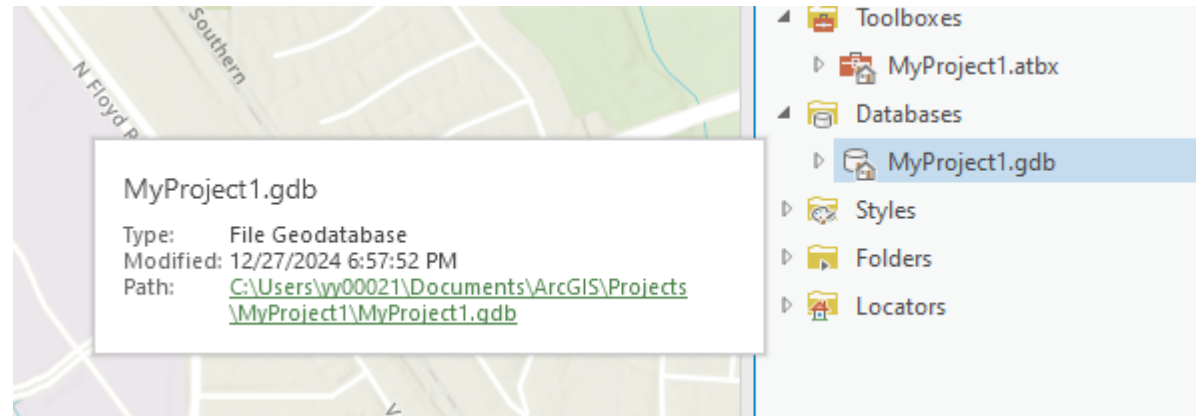
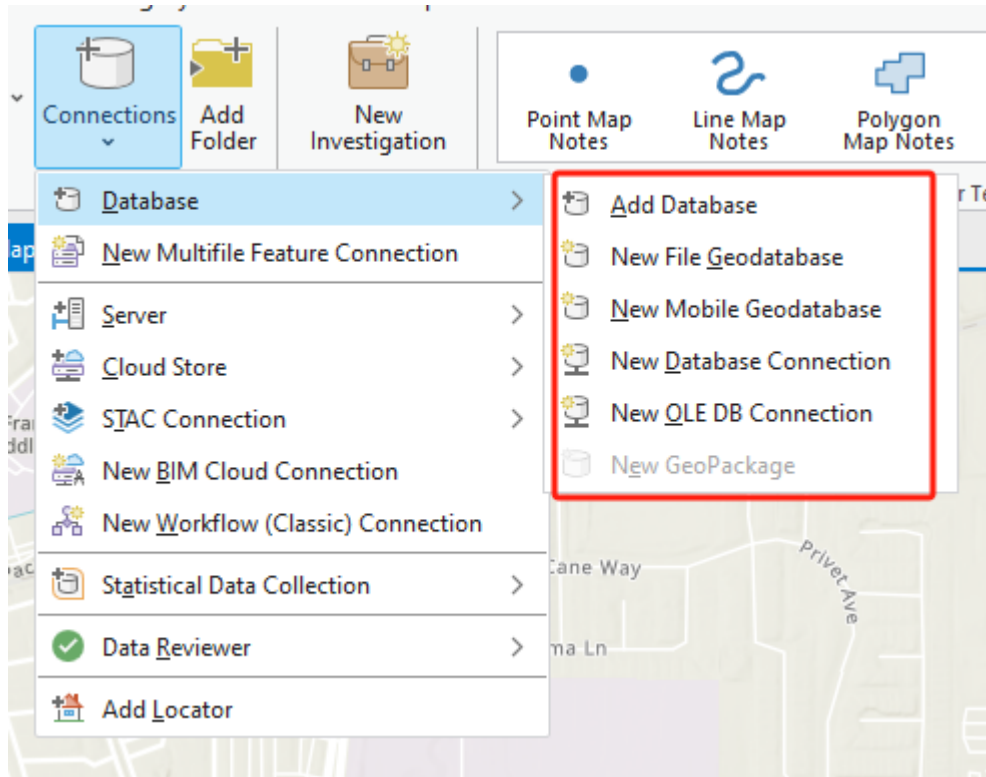
- Capable of handling large datasets efficiently in enterprise environments.

COMPARING TYPES OF GEODATABASES

More detailed [comparison of geodatabases](#)

Key characteristics	Enterprise geodatabase	Mobile geodatabase	File geodatabase
Description	A collection of various types of GIS datasets held as tables in a relational database.	A collection of various types of GIS datasets contained within a relational database.	A collection of various types of GIS datasets held in a file system folder.
Number of users	Multiple editors and can support multiple readers.	Single editor and can support multiple readers.	Single editor and can support multiple readers.
Storage format	<ul style="list-style-type: none">• Oracle• Microsoft SQL Server• IBM Db2• PostgreSQL• SAP HANA	All the datasets that belong to one mobile geodatabase are contained in an SQLite database that is stored in a single file.	Each dataset is a separate file on disk. All the datasets that belong to one geodatabase are contained in a single folder.

GEODATABASE IN ARCGIS PRO



2.1.3 SQL

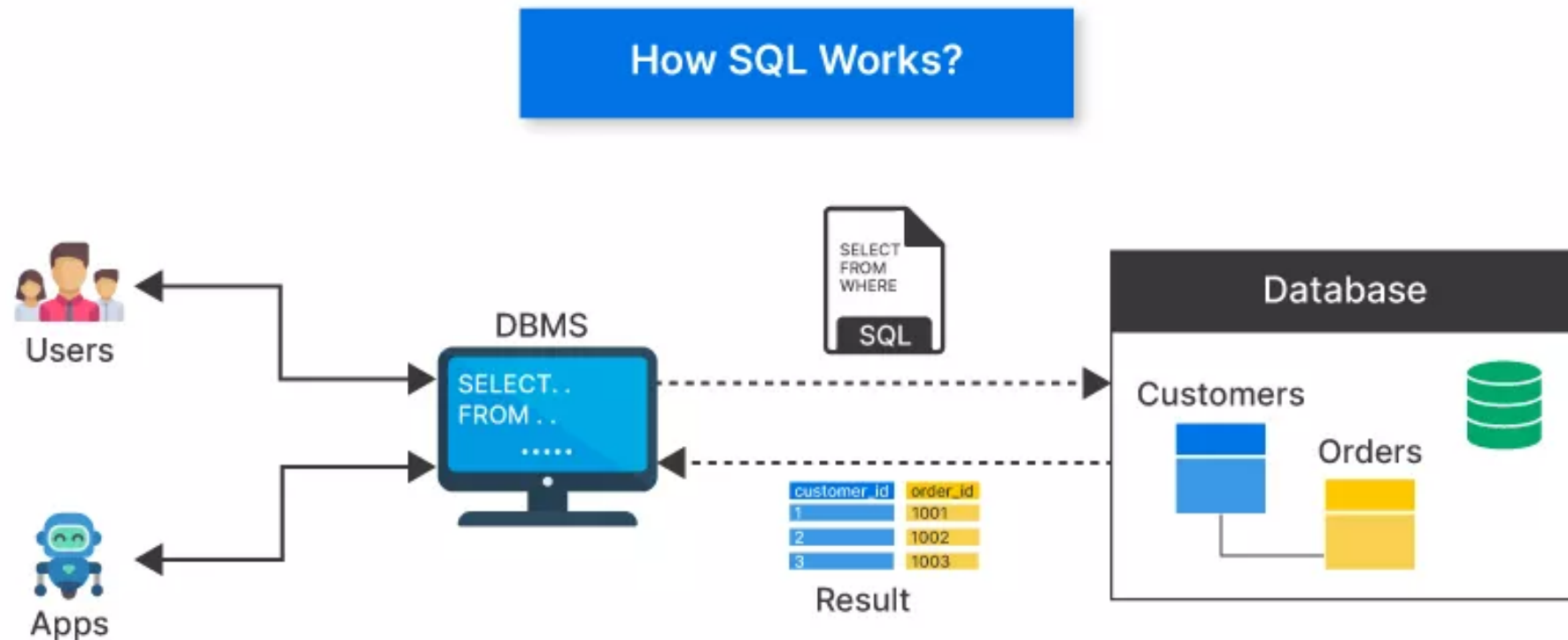


STANDARD QUERY LANGUAGE (SQL)

- SQL is a standard programming language specifically designed for managing and manipulating relational databases.
- It allows users to interact with a DBMS to perform various operations such as querying, updating, and managing database structures.

HOW SQL WORKS

- It serves as a standard interface for interacting with relational database management systems (RDBMS), allowing users to perform a variety of routine tasks and operations on data
 - such as querying, updating, inserting, and deleting records.



KEY POINTS BETWEEN DBMS AND SQL

- SQL is Independent of DBMS: While SQL is a standard, each DBMS may implement it with slight variations or extensions (e.g., PL/SQL in Oracle, T-SQL in SQL Server).
- DBMS Without SQL: Some DBMS types, such as NoSQL databases (e.g., MongoDB), do not rely on SQL but use alternative query languages.
- SQL Requires a DBMS: SQL commands need a DBMS to interpret and execute them. Without a DBMS, SQL commands would have no database to operate on.

2.1.4 SOFTWARE

SOFTWARES

- PostgreSQL
- What is PostgreSQL?
 - An open-source, powerful, object-relational database management system.
 - Known for its reliability, extensibility, and SQL compliance.
- Why PostgreSQL for GIS?
 - Supports structured queries for data management and manipulation.
 - Extensible via plugins, making it adaptable for spatial needs.

POSTGIS REFERENCE USERS - GOVERNMENT



Natural Resources Canada Ressources naturelles Canada

Canada



Ordnance Survey



POSTGIS REFERENCE USERS - PRIVATE SECTOR



Ball Aerospace
& Technologies Corp.



SOFTWARE

- What is PostGIS?
 - A spatial database extension for PostgreSQL.
 - Adds support for geographic objects like points, lines, and polygons.
- Key Features of PostGIS:
 - Spatial data types: GEOMETRY and GEOGRAPHY.
 - Spatial functions:
 - ❑ Query and manipulate spatial data (ST_Intersects, ST_Buffer, ST_Distance).
 - ❑ Perform spatial analyses (e.g., clipping, union, or proximity searches).

POSTGIS 3RD PARTY INTEGRATION

DESKTOP



AND MORE...

MIDDLE



mapnik



AND MORE...

LANGUAGE



AND MORE...

POSTGIS IS POWERFUL

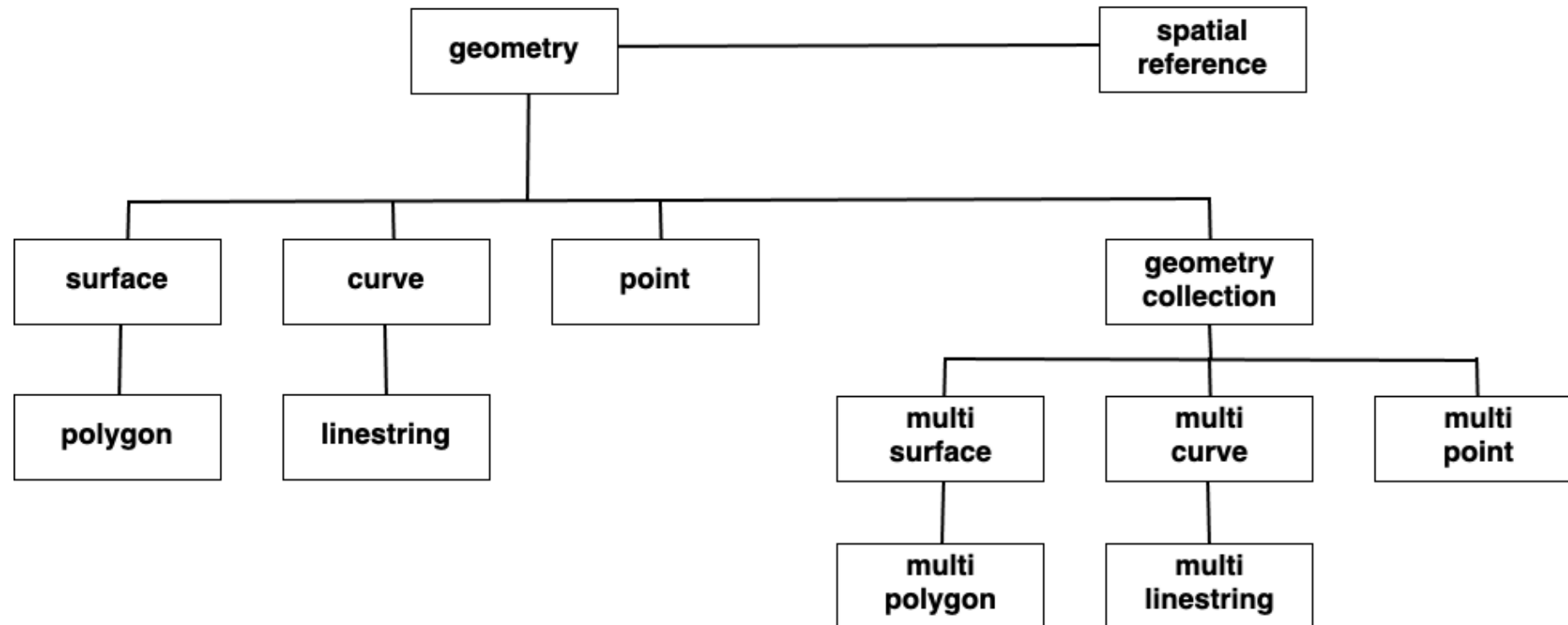
- Functions to work with [GeoJSON](#) (Geographic JavaScript Object Notation), [Keyhole Markup Language](#) (KML), Mapbox Vector Tiles (MVT) allowing web applications to talk directly to PostGIS
- GeoJSON and KML are two of the older, more popular vector formats used by web mapping applications

Example: [Analyze Boston](#) has GeoJSON and KML to store vector data

- Mapbox Vector Tiles is a relatively new standard that has gained quite a bit of popularity in the last few years

POSTGIS IS POWERFUL

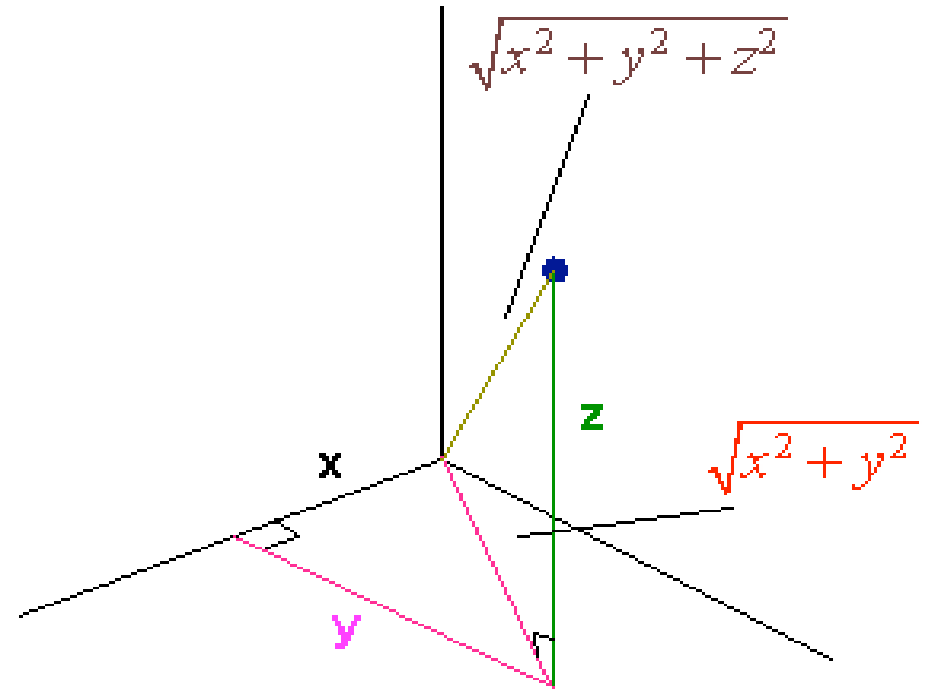
- Comprehensive geometry processing functions that go far beyond basic geometric operations, including functions for fixing invalid geometries and for simplifying and deconstructing geometries



POSTGIS IS POWERFUL

■ Built-in 3D and Topology Support

- `ST_3DClosestPoint(g1, g2)`
- `ST_3DDistance(g1, g2)`
- `ST_3DDWithin(g1, g2, r)`
- `ST_3DDFullyWithin(g1, g2, r)`
- `ST_3DIntersects(g1, g2)`
- `ST_3DLongestLine(g1, g2)`
- `ST_3DMaxDistance(g1, g2)`
- `ST_3DShortestLine(g1, g2)`





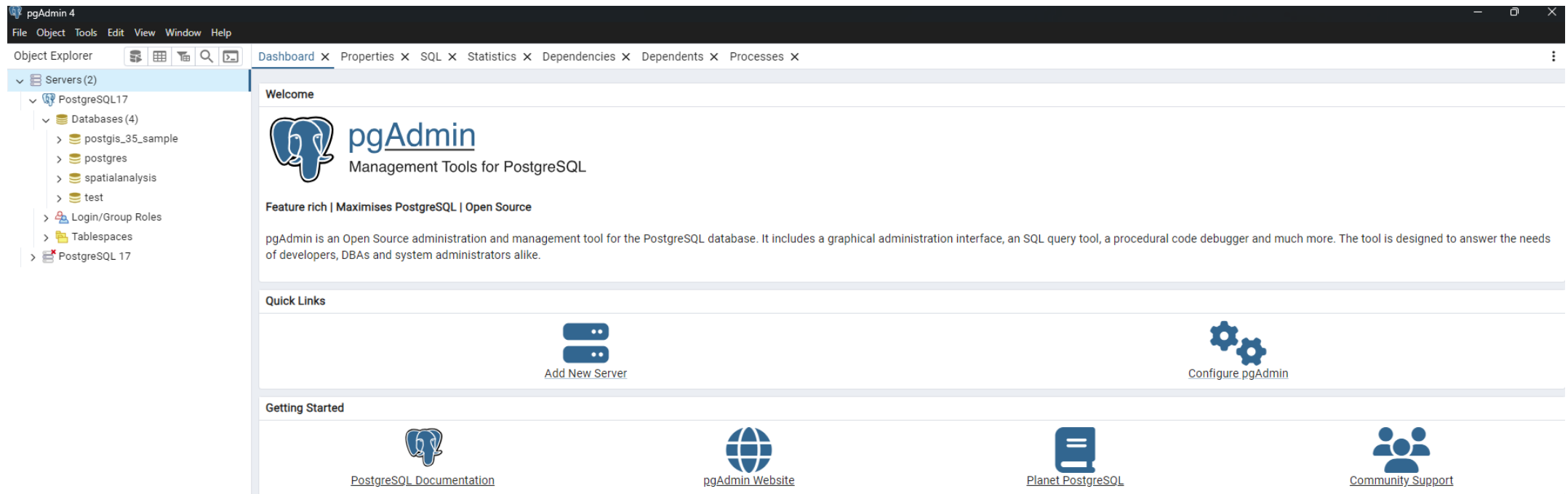
POSTGIS IS POWERFUL

- Over [300 seamless operations](#) for working with vectors and rasters in tandem, as well as for converting between the two families

2.1.5 PGADMIN

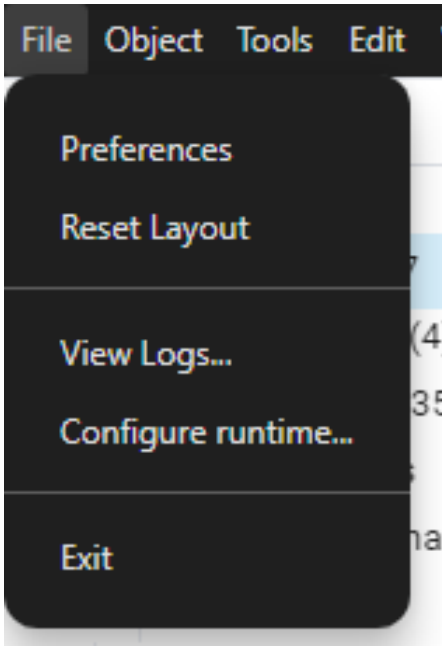
USER INTERFACE

- Add New Server button: Open the Register - Server dialog to add a new server definition.
- Configure pgAdmin button: Open the Preferences dialog to customize your pgAdmin client.
- PostgreSQL Documentation: The Documentation page for the PostgreSQL open-source project.
- pgAdmin Website: The website for pgAdmin releases and other project information.
- Planet PostgreSQL: Postgres related blogs.
- Community Support: information about obtaining support for PostgreSQL features.



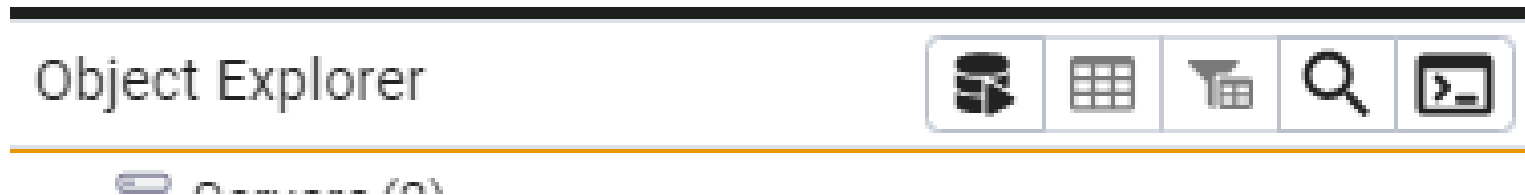
MENU BAR

Option	Action
Preferences	Click to open the Preferences dialog to customize your pgAdmin settings.
Reset Layout	If you have modified the workspace, click to restore the default layout.
Runtime	Click to open a submenu to Configure, View Log and Zoom settings. Only visible when pgAdmin4 runs in desktop mode. To know more about runtime menu click here



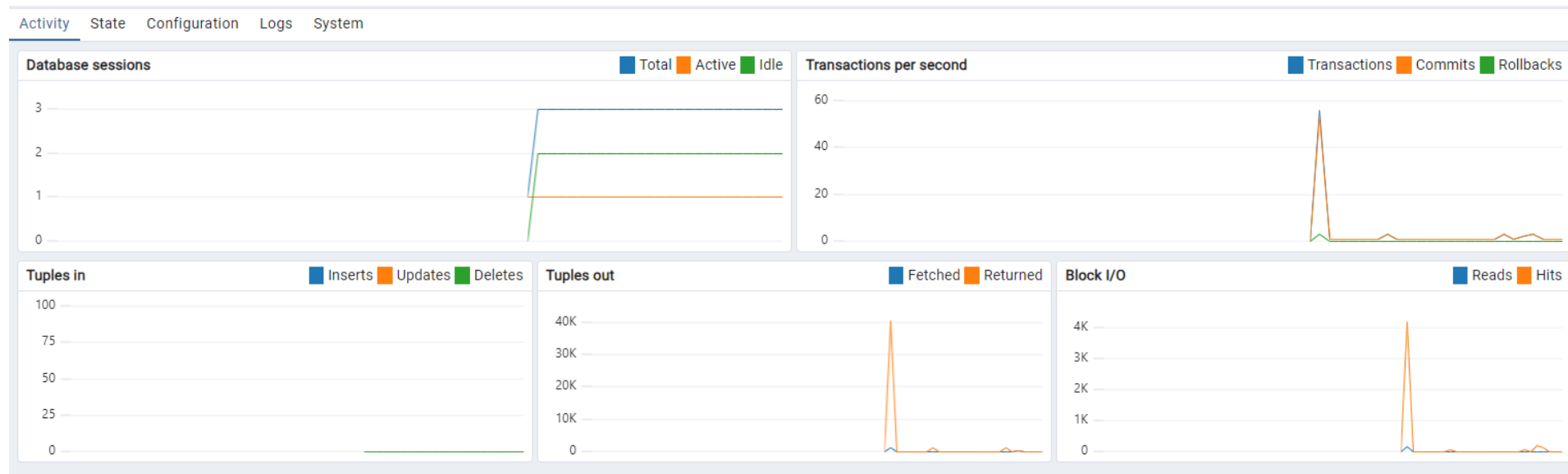
TOOLBAR

- Query button: Open the Query Tool in the current database context.
- View Data button: View/edit the data stored in a selected table.
- Filtered Rows button: Access the Data Filter popup to apply a filter to a set of data for viewing/editing.
- Search objects button: Access the search objects dialog. It helps you search any database object.
- PSQL Tool button: Open the PSQL in the current database context.



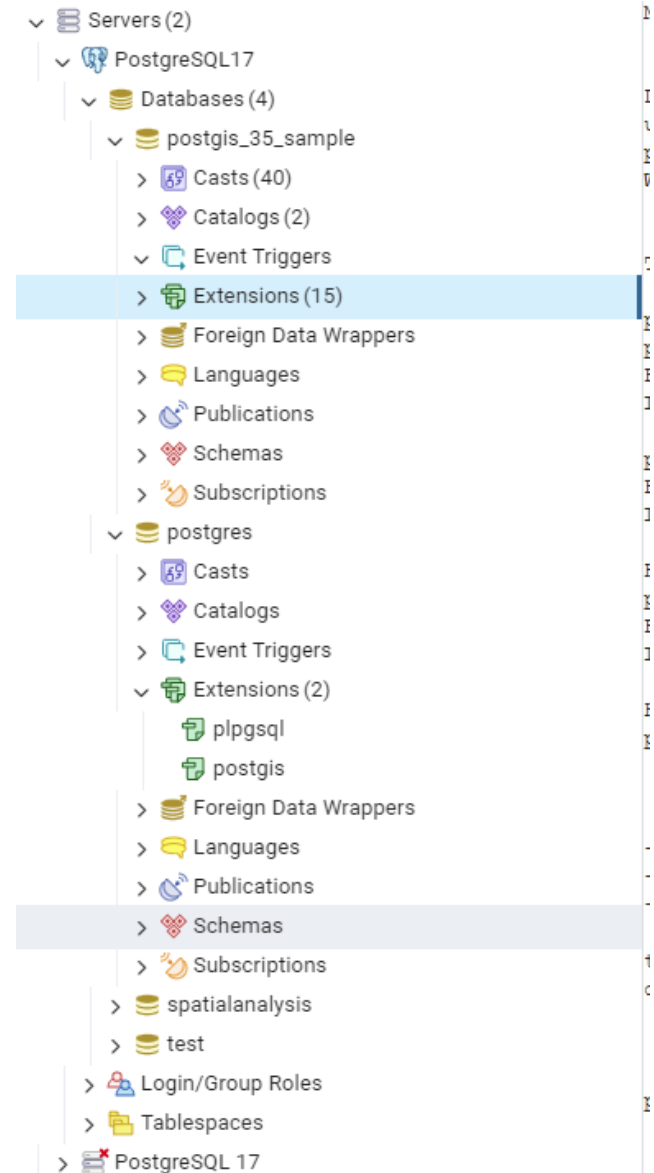
TABBED BROWSER

- The *Server sessions* or *Database sessions*: Display the interactions with the server or database.
- The *Transactions per second*: Display the commits, rollbacks, and total transactions per second that are taking place on the server or database.
- The *Tuples*: Display the number of tuples inserted, updated, and deleted on the server or database.
- The *Tuples out* graph: Display the number of tuples fetched and returned from the server or database.
- The *Block I/O* graph: Display the number of blocks read from the filesystem or fetched from the buffer cache (but not the operating system's file system cache) for the server or database.



TREE CONTROL

- The left pane of the main window displays a tree control (Object explorer) that provides access to the objects that reside on a server.





WEEK 02

LAB SESSION

Instructor: Yanan Wu

TA: Vanchy Li

Spring 2025

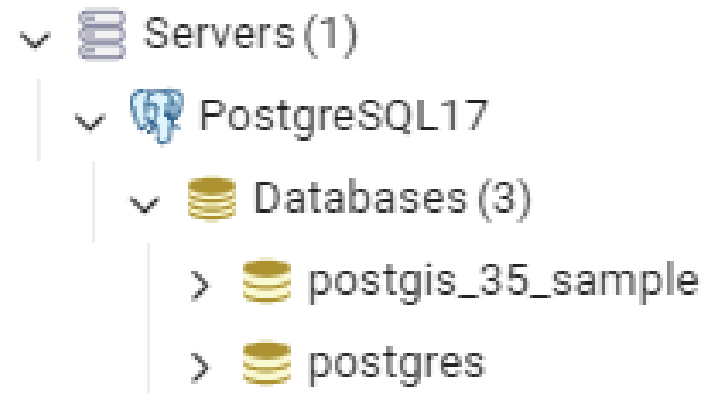


2.2.1 BASIC OPERATIONS IN POSTGRESQL

DATABASES UNDER SERVER

- The server contains two databases
 - One is the default postgresql database
 - The other one is the database we created while installing PostgreSQL
 - If one of the servers has a red cross on it means that it is not connected
 - You can click on it to have access
 - Sometimes you are required to provide the password

Object Explorer



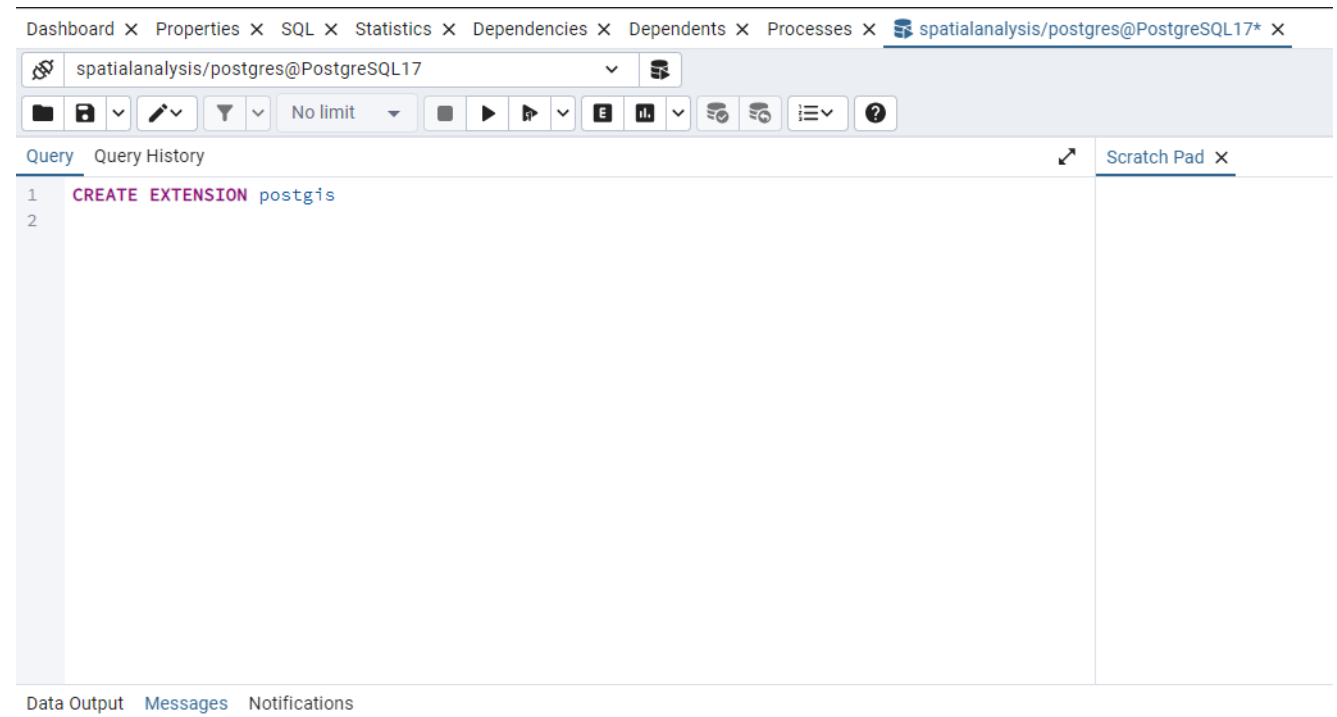
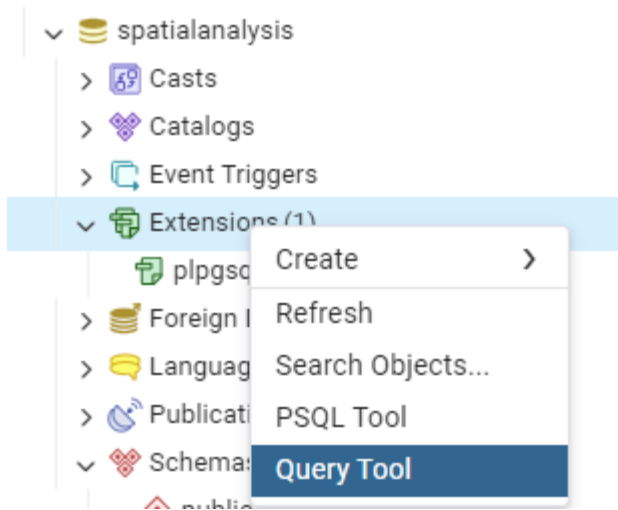
CREATE A DATABASE

- Create a new Database per you need



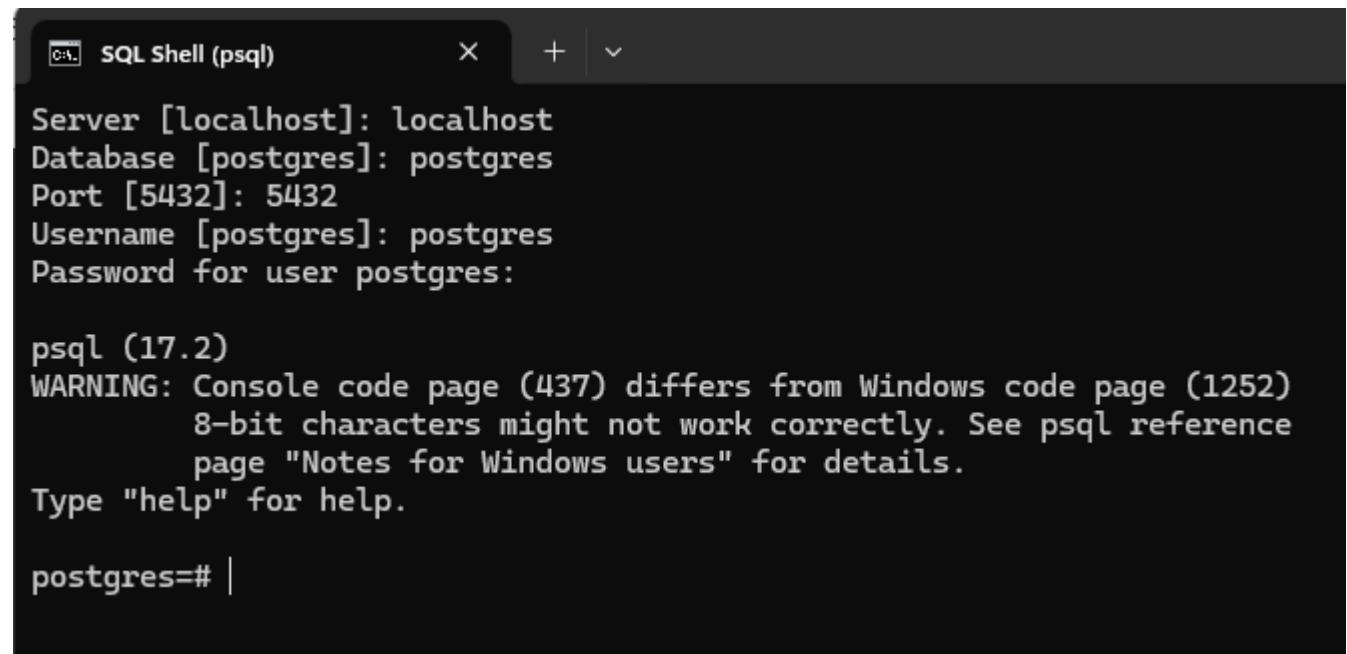
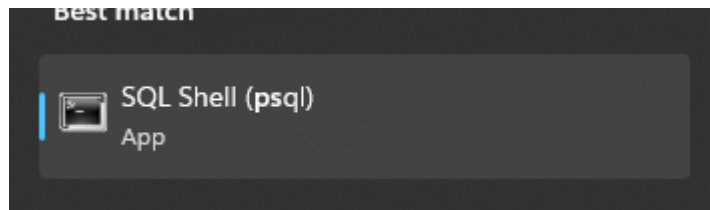
CHECK EXTENSION

- The default extension is plpgsql
- Adding postgis extension through Query tool
- Type “CREATE EXTENSION postgis” in Query
- Then the postgis extension was created under Extensions



SQL SHELL

- What if I do not want to use pgAdmin?
 - SQL shell (psql) is your second option
- The suggested server is localhost, Database is postgres, port is 5432, username is postgres
- Start executing SQL statements

A screenshot of a terminal window titled 'SQL Shell (psql)'. The terminal shows the following text:

```
Server [localhost]: localhost
Database [postgres]: postgres
Port [5432]: 5432
Username [postgres]: postgres
Password for user postgres:

psql (17.2)
WARNING: Console code page (437) differs from Windows code page (1252)
        8-bit characters might not work correctly. See psql reference
        page "Notes for Windows users" for details.
Type "help" for help.

postgres=# |
```


VERSION OF POSTGIS AND POSTGRESQL

- *SELECT postgis_full_version()*

POSTGIS="3.5.0 3.5.0" [EXTENSION]

PGSQL="170" GEOS="3.13.0-CAPI-1.19.0" PROJ="8.2.1 NETWORK_ENABLED=OFF

URL_ENDPOINT=https://cdn.proj.org

USER_WRITABLE_DIRECTORY=C:\WINDOWS\ServiceProfiles\NetworkService\AppData\Local\proj

DATABASE_PATH=D:\PostgreSQL\share\contrib\postgis-3.5\proj\proj.db" (compiled against PROJ 8.13.0)

LIBXML="2.12.5" LIBJSON="0.12" LIBPROTOBUF="1.2.1" WAGYU="0.5.0 (Internal)"

CHECK VERSION IN SQL SHELL

```
postgres=# SELECT version();
               version
-----
PostgreSQL 17.2 on x86_64-windows, compiled by msvc-19.42.34435, 64-bit
(1 row)
```

- psql waits for the semicolon and executes all line as one SQL statement
- A multiple lines SQL statement is not executed before we include a semicolon at the end
- I prefer the UPPERCASE syntax for SQL commands

CREATE DATABASE IN SQL SHELL

CREATE DATABASE test;

\1

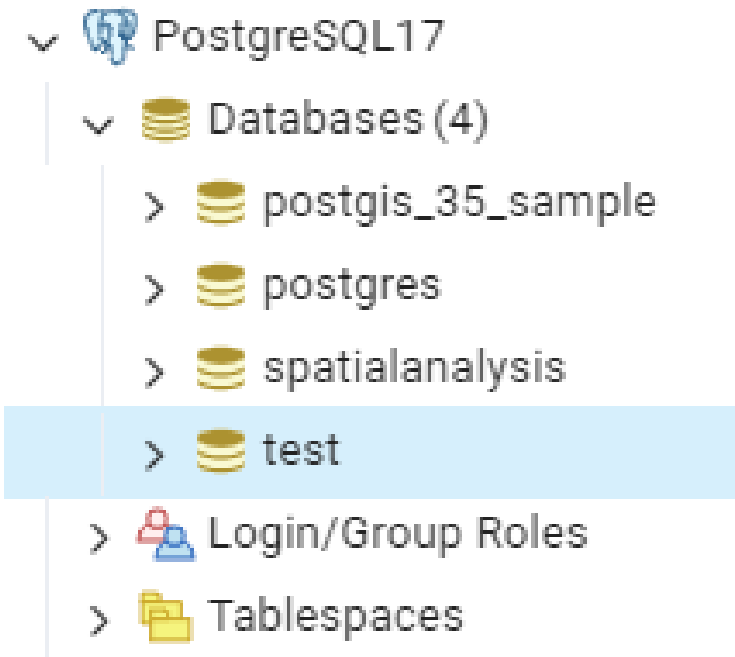
```
postgres=# CREATE DATABASE test;
CREATE DATABASE
postgres=# \1
```

List of databases									
Name	Owner	Encoding	Locale Provider	Collate	Ctype	Locale	ICU Rules	Access privileges	
postgres	postgres	UTF8	libc	English_United States.1252	English_United States.1252				
spatialanalysis	postgres	UTF8	libc	English_United States.1252	English_United States.1252				
template0	postgres	UTF8	libc	English_United States.1252	English_United States.1252				=c/postgres
template1	postgres	UTF8	libc	English_United States.1252	English_United States.1252				postgres=CTc/postgres =c/postgres
test	postgres	UTF8	libc	English_United States.1252	English_United States.1252				postgres=CTc/postgres

(6 rows)

CHECK NEW DATABASE IN PGADMIN

- Refresh the Database under your server



CONNECT DATABASE IN PSQL

```
psql (17.2)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

test=# \c postgres
You are now connected to database "postgres" as user "postgres".
postgres=#
```

CONNECT TO DATABASE USING COMMAND LINE

- Using command line to check the documentation for command in psql
- Open command line and type

psql --help

where psql

If you have this error message: 'psql' is not recognized as an internal or external command.

Then you need to add PATH of psql to your window path:

```
C:\Users\yy00021>psql --help
psql is the PostgreSQL interactive terminal.

Usage:
  psql [OPTION]... [DBNAME [USERNAME]]

General options:
  -c, --command=COMMAND      run only single command (SQL or internal) and exit
  -d, --dbname=DBNAME        database name to connect to
  -f, --file=FILENAME        execute commands from file, then exit
  -l, --list                  list available databases, then exit
  -v, --set=, --variable=NAME=VALUE
                              set psql variable NAME to VALUE
                              (e.g., -v ON_ERROR_STOP=1)
  -V, --version               output version information, then exit
  -X, --no-psqlrc             do not read startup file (~/.psqlrc)
  -1 ("one"), --single-transaction
                              execute as a single transaction (if non-interactive)
  -?, --help[=options]       show this help, then exit
  --help=commands            list backslash commands, then exit
```

DANGEROUS COMMAND



- DROP DATABASE test;