

# Overview Practical 1

Database Server and Database Software

Client Software for Accessing Database

Users and Schemata

Specific Schemata

Objects in the Database

GUI of DBeaver for PostgreSQL

# Database Server and Database Software

## A Virtual Unix Server

- ▶ Server machine's name:
- ▶ sc02.geo.uzh.ch
- ▶ Server machine is also called 'host'
- ▶ Operating system Ubuntu 20.04

## PostgreSQL 14.11

- ▶ Database we use for this course is named:
- ▶ **geodb** (= geodb.sc02.geo.uzh.ch)
- ▶ Connection to the database on port 5432
- ▶ Once connected you can do SQL: `SELECT version();`

# Client Software for Accessing Database

## DBeaver for PostgreSQL (and other databases)

- ▶ Grafical User Interface (GUI); Community edition is free and open source under Apache License 2.0
- ▶ Connects to our database `geodb.sc02.geo.uzh.ch:5432`

## Other SQL Clients

- ▶ DataGrip (professional software); pgAdmin IV (open source) and many others
- ▶ Integrated Development Environment (IDE ) like PyCharm, Visual Studio, Eclipse etc.
- ▶ Self written software program
- ▶ Web server, Web applications
- ▶ On the server -> commandline tool: `psql`

## Graphical Geo Clients

- ▶ QGIS / Grass
- ▶ ArcGISPro
- ▶ Web server, web applications and other software

# User accounts for this course

## userxx

- ▶ **Your personal user** (synonym = role) to access the database
- ▶ **xx** stands for a decimal number e.g 07 or 23 (user07, user23...)
- ▶ Use this account for exercises and for testing out your ideas
- ▶ Password:
- ▶ We will maintain a list with your name (real) <-> user (database)

## userdemo

- ▶ This user owns all sample data of geo874 and geo875
- ▶ Created and filled by us
- ▶ You are granted to query these tables
- ▶ Owns the three tables later needed for the assignment (`userdemo.city`, `userdemo.country`, `userdemo.neighbor`)

# User and Schemata

## A user...

- ▶ ... can log in the database
- ▶ ... owns a previously defined **schema**; the schema has the same name as the **user**; a schema is similar to a directory
- ▶ ...can create and query **objects** in its own schema
- ▶ ...sees objects in other schemata and can eventually query or manipulate these objects

## Objects of a user...

- ▶ ...reside in the **schema** of the user
- ▶ ...are for example: **tables, views, sequences** etc. Do you find other types in the schema of **userdemo**?

# Specific schemata already in the DB

## public

- ▶ Created in every database during creation
- ▶ Objects in the public **schema** are accessible to all users; this happens through the **role** also called public; all users of the database are members of the role public;
- ▶ We installed PostGIS (a spatial extension to the database -> geo875); therefore public contains all functions, views, types, operators etc. for storing geometries and for handling spatial objects

## topology

- ▶ A schema that is installed with the extension postgis\_topology; contains all functionality for handling topology in a database

## information\_schema, pg\_catalog

- ▶ Views and tables containing metadata about the database system

# GUI of DBeaver for PostgreSQL

## Start DBeaver on the Course Server (rds)


- ▶ Klick Windows Button to get to Start Screen
- ▶ Just type: DBea...
- ▶ Rightclick the DBeaver in the appearing list -> Pin to Taskbar
- ▶ Switch back to normal desktop and start DBeaver from taskbar

## Installation of DBeaver on Your Personal Laptop

- ▶ Download and install from <https://dbeaver.io/download/> Community Edition
- ▶ You need to be within University network (Virtual Private Network VPN) to be able to connect to the course database
- ▶ It would also work with a SSL (Secure Socket Layer) connection – instead of VPN – from an outside net; you can configure that directly in a DBeaver connection option;

# Minimal Configuration of DBeaver I

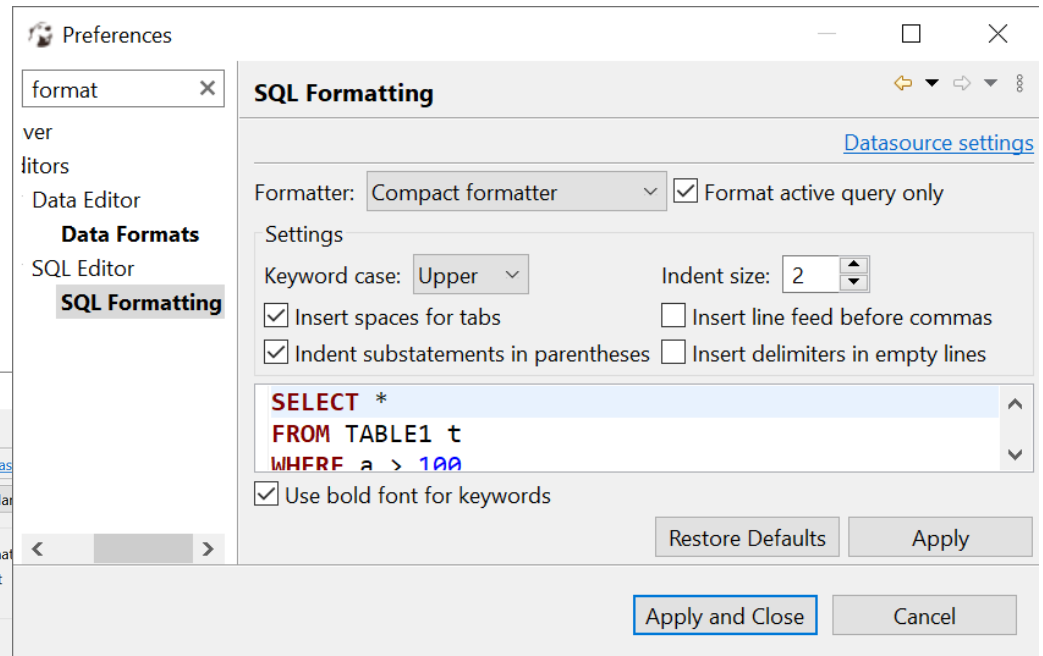
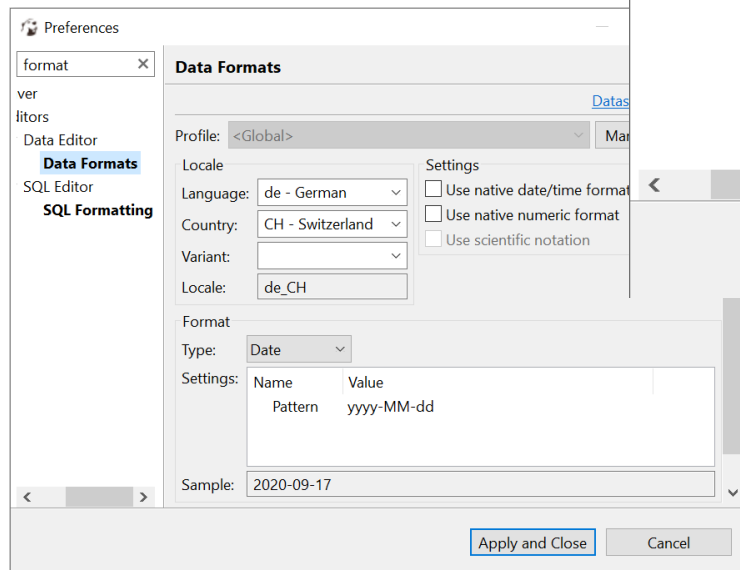
## GUI

- ▶ Menu – Window: Leave only Projects visible
  - ▶ In the Projects Window you can see a General folder
  - ▶ Create a new Project by using this icon on the Project Window line
- 
- ▶ Project Name: PROJECT\_GEO874\_USERxx
  - ▶ Uncheck Use default location
  - ▶ Put a path to your personal file storage e.g. similar to  
W:\VORLESUNG\_25\GEO874\PROJECT\_GEO874\_USERxx
  - ▶ Finish
  - ▶ Right-click the project folder PROJECT\_GEO874\_USERxx -> Set Active Project

# Minimal Configuration of DBeaver II

## Preferences

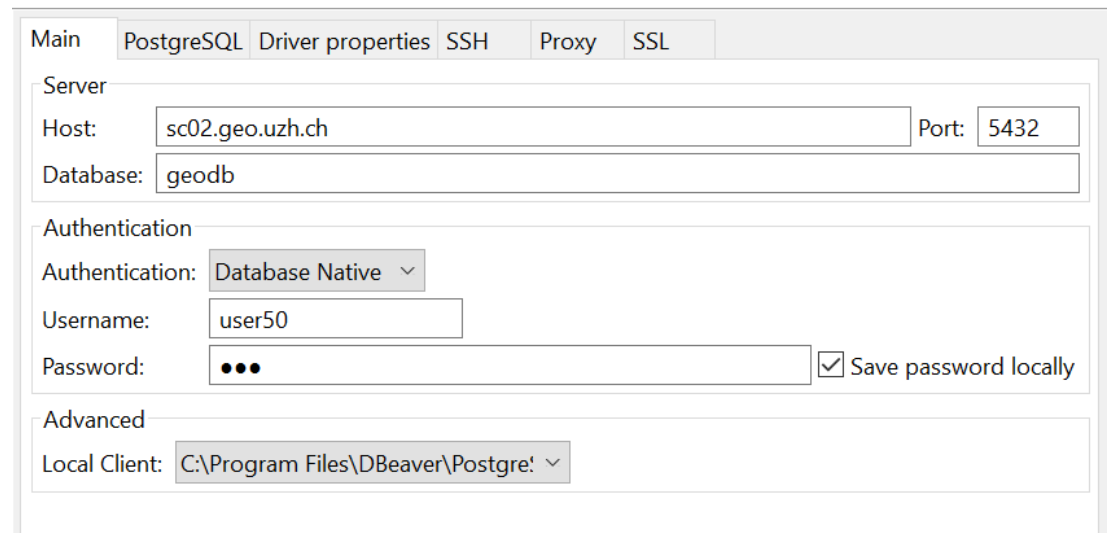
- ▶ Menu → Window → Preferences
- ▶ Type format in the search field



# DBeaver Connection

## Connection

- ▶ Under your Project Folder PROJECT\_GEO874\_USERxx: Right-click Connections -> Create -> Connection: choose PostgreSQL icon
- ▶ Next Button
- ▶ Adjust according to
- ▶ Use your Username and Password
- ▶ Test Connection Button
- ▶ If Success then OK Button
- ▶ Rename your connection «geodb» under the Project Connections by Right-click -> Rename -> userxx@geodb



The screenshot shows the 'PostgreSQL' tab of the DBeaver connection configuration dialog. The 'Server' section includes fields for 'Host' (sc02.geo.uzh.ch), 'Port' (5432), and 'Database' (geodb). The 'Authentication' section shows 'Authentication' set to 'Database Native', 'Username' as 'user50', and a masked 'Password' field with a 'Save password locally' checkbox checked. The 'Advanced' section shows 'Local Client' set to 'C:\Program Files\DBeaver\Postgre!'.

# Play with DBeaver and the Database

## Query Window - Scripts

- ▶ Double click your connection to log in to «geodb»
- ▶ Right click connection -> SQL Editor
- ▶ This opens (or reopens) a new file (=script) where you can enter SQL statements and then execute them
- ▶ You can save this file; default location is your Project-Scripts folder

## Explore the database through your connection

- ▶ Drill down your connection -> geodb -> Schemas
- ▶ Look at what you have in your schema; look at userdemo's schema
- ▶ On Schemas you can define Filters (right click on Filters) to include e.g. your user~~xx~~ and userdemo