# **OpenHyPE: OpenHygrisC Data Processing for Education**

- Project funded by the Federal State of Northrhine-Westphalia (NRW)
- Funded by Ministerium für Landwirtschaft und Verbraucherschutz NRW (MLV)
- Project duration: 15.12.2021 30.06.2023

**ZUSAMMENFASSUNG, EINLEITUNG UND ZIELE AUF DEUTSCH** 

## 0. Abstract

The State Agency for Nature, Environment and Consumer Protection (LANUV) of the federal state of North Rhine-Westphalia (NRW) provides extensive quantitative and qualitative groundwater monitoring data. This contributes to the fulfillment of the European Water Framework Directive as well as the EU INSPIRE Directive for an open interoperable spatial data infrastructure.

NRW operates its own water-related data portal **ELWAS-WEB**, which also provides the groundwater database **HygrisC**. ELWAS and HygrisC are not easy to use and provide only limited exploratory data analysis capabilities to the public.

However, the state publishes much of its groundwater data as an **open data archive** called **OpenHygrisC**, which contains several data tables in **csv format**. In particular, the **big data** in the measurement table, which contains all time series with **more than 3.6 million individual measurements** (table rows), and the table with the spatial coordinates of the groundwater monitoring wells require the use of a **spatially enabled object-relational database management system (Spatial ODBRMS)** and extensive **data engineering** before insertion into the database.

The goal of the **OpenHyPE project** is to develop a first set of **Open Educational Resources (OER)** to train the setup, filling and use of a **geospatial-temporal database** with the OpenHygrisC data. Due to the **graduated level of difficulty**, the project addresses **students** from **secondary schools** as well as **universities** in the state of North Rhine-Westphalia and beyond.

All software products used are **Free and Open Source Software (FOSS)**. The database, which we call **OpenHyPE DB**, is based on **PostgreSQL / PostGIS** and establishes the center of the system for environmental data analysis and presentation. The OER demonstates how the geographic information system **QGIS** as well as **Python programs** in the **JupyterLab development environment** interoperate with the OpenHyPE DB to select, analyze and display the data in the form of time dependent maps or time series. We use Python and Jupyter from the **Anaconda distribution**.

The start-up funding for the OpenHyPE project is used to raise awareness of the **NRW's valuable** open environmental data collection among young people as well as to contribute to interdisciplinary STEM promotion in general and education for sustainable development (ESD) in particular by linking environmental science and computer science.

# 1. Introduction

### **1.1 Problem Description**

The state of North Rhine-Westphalia (NRW) operates comprehensive and professional measurement networks for the collection of environmental data through the LANUV. As part of Open.NRW and driven by the INSPIRE Directive of the European Union as well as other directives such as the EU Water Framework Directive (WFD), the state of NRW makes extensive data products openly accessible and freely usable on various platforms (Free and Open Data).

The state of NRW is a pioneer in Germany in providing open and (cost)free geodata. These data are a real treasure and form the basis for potentially massive knowledge gains in the field of environmental and nature conservation. Nevertheless, it seems that only a comparatively small group of people really uses this potential. Therefore, the OpenHyPE project has set itself the task of integrating this data stock into university teaching and developing corresponding freely accessible teaching material that can be used not only by students but also, to some extent, by pupils to learn the basics of environmental data processing. The start-up funding will be used to implement the first steps of developing such training material.

We follow the paradigm of "problem based learning": the necessary knowledge and skills are identified and taught based on a concrete socially relevant problem. The solution of the problem identified as significant is the motivation for learning.

At the beginning we want to develop the material on the basis of the problem area "groundwater protection". The Ministry for Environment, Agriculture, Nature and Consumer Protection NRW (MULNV) operates its own water-related data portal called ELWAS-WEB via the "Landesbetrieb Information und Technik Nordrhein-Westfalen" (IT.NRW). Data from the statewide groundwater database HygrisC are also held in this portal. ELWAS and HygrisC offer limited exploratory data analysis capabilities to outsiders. From the point of view of usability engineering, which deals with the user-friendliness of technical systems, improvements are desirable with regard to usability as well as data analysis possibilities, because exploratory data analysis and data mining in particular help to identify structures and relationships between data. ELWAS and HygrisC are therefore only suitable to a limited extent for teaching the basics of environmental data analysis, but they can be used in the classroom as supporting material.

On the portal OpenGeodata.NRW extensive data with spatial reference - also called geodata - are made available, which often have a time reference, such as land use changes or measurement data series on water quality. Excerpts of the HygrisC groundwater database of the state of NRW, published under the name OpenHygrisC, are also located there. These groundwater data can ideally serve as a basis for building one's own environmental database, which the learners can use to learn about concepts of data management and data analysis.

### 1.2 Project Goals

The following components are to be realized:

- Development of OpenHyPE geodatabase based on PostgreSQL/PostGIS to manage spatial and temporal data on groundwater quality and quantity.
- Problem-related free online course material (OER), tutorials, video tutorials, instructions,

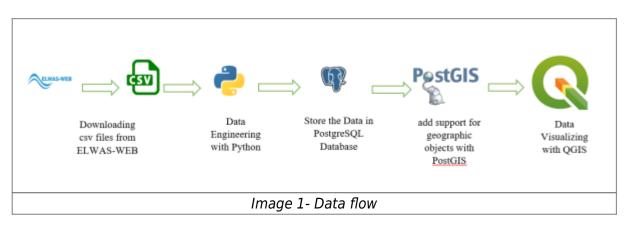
#### program code,

using Free and Open Source Software (FOSS):

- Introduction to the State Agency for Nature, Environmental and Consumer Protection (LANUV).
- Introduction to groundwater protection
- $\circ\,$  Introduction to the Geographic Information System QGIS
- $\circ\,$  Introduction to the relational database PostgreSQL and the query language SQL
- Introduction to the geodatabase extension PostGIS
- Introduction to the processing of geodata with the programming language Python
- $\circ~$  Installation of the OpenHyPE database management system
- Discussion of the data model and upload of the OpenHygrisC data of the LANUV
- $\circ\,$  Automatic creation of diagrams on time series of water quality
- $\circ\,$  Automatic generation of groundwater chemistry maps
- $\circ\,$  Creating simple dashboards with interactive online graphs and maps
- Introduction to data mining (descriptive statistics, searching for correlations)

### 2. Implementation

### 2.1 Data Flow



## 2.2 PostgreSQL/PostGIS

PostgreSQL is an open-source object-relational database management system (DBMS) known for its robustness, scalability, and extensive features. It is often referred to as "Postgres" and is one of the most popular and widely used databases in the world. PostgreSQL supports a wide range of data types, including numeric, text, Boolean, date/time, JSON, XML, and more. It provides support for complex queries, indexing, and advanced features such as full-text search, spatial data storage and querying, and transactional processing with ACID (Atomicity, Consistency, Isolation, Durability) properties.

PostgreSQL follows the SQL (Structured Query Language) standard, and it also provides additional features beyond the standard SQL specification. It supports stored procedures, triggers, and views, allowing developers to define custom business logic within the database itself.

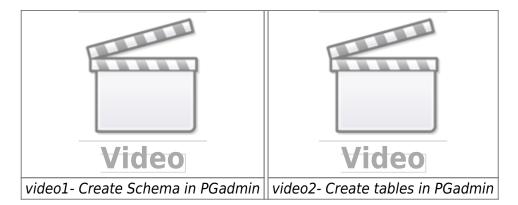
PgAdmin is an open-source administration and development platform for PostgreSQL. It is a graphical user interface (GUI) tool that provides a convenient way to manage and interact with PostgreSQL databases.

**PostGIS:** PostGIS is an open-source spatial database extension for PostgreSQL. It adds support for geographic objects and spatial functions to the PostgreSQL database, enabling the storage, retrieval, and analysis of geospatial data. Geographic data such as points, lines, polygons, and multi-dimensional geometries can be stored and manipulated within your PostgreSQL database using PostGIS. The capabilities of the database are extended by PostGIS to handle spatial data types, indexing, and spatial operations. PostGIS has gained popularity in a variety of applications, including mapping, geolocation-based services, environmental analysis, urban planning, and transportation. Its combination with the power and versatility of PostgreSQL makes it a robust solution for managing and analyzing geospatial data in a relational database environment.

The below image shows the PGadmin tool.



Watching the below videos to understand how we can create schemas and tables in the PostgreSQL database.



Create a database and schema based on the above video.

# 2.3 Data Engineering

# 2.3.1 Downloading the Data

In the first step, The data must be downloaded from here. To complete the process, download and extract the initial zip file, which contains four CSV files and one set of instructions. Refer to image 3 for guidance on selecting the appropriate zip file.

Car 🔜 UNI 🔜 Hame 🔝 Cyber 🔥 Drojski far Joveybo 💽 Arangaanseet 4	🔅 Debbil desentregen 🚺 Überbick für Stude	BM92 peine 🖬 Betranishi An	n
		Information und Nordrheim West	Tachsia alise
OpenGeocata.NRW 3 Dryges		(J.4	SCH 45 XML
Grundwassermessstellen NRW			
Linestenbereitige: Diese Daten sichen unter der Detentioner Des Datensatzbeschweitbung: BropertrygtetC_meta.pp Meladaten im G* Despend XRW	dicitions - Zero - Venitor 2.9. Jack Nutzing in ohr	n Einschränkungen nörr Sedingunge	en Andineig
Daten der Grundwassermessstellen in NRW als CSV			
Liaten der Grundwassermessstellen in Nikw als GGV			
Daten der Grundwassermessstellen in NRW als CSV		Letzte Änderung	Dataigroile
		Letzte Änderung 25-10.2021 08-48.00	Dateigröße 41.7 MB
Datai		and the second se	
Detai @ Openthypted_gav-menadeles-mesoweris_EPS625602_GDV.op	nbank	25-10-2021-00-48-00	41.7 MB
Datai ClosentrygetsG_gav-messaciellan-messawarka_RPSG28002_CBV.sp @ OpentrygetsG_gav-messaritant_RPSG28002_CBV.sp	nbank	25-10-2021-00-48-00	41.7 MB
Detai DenthypteC_ga-meastellen-messwerle_EPS628022_CEV.zp (d) OpenhypteC_ge-waserstand_EPS628022_CEV.zp Daten der Grundwassermessstellen in NRW als Spite Date	nbank	29-10-2021 08-46-00 29-10-2021 08-49-08	41.7 MB 100.4 MB
Detai Detai DetainypteC_gaverassetiles-resources_EPS62802_CEV.ep (d) OpenHypteC_gaveseentant_EPS62802_CEV.ep Deten der Grundwassermessstellen in NRW als Sollte Date Detai	nbank	25.10.2021 08-49-00 25.10.2021 08-49-00 Letzte Anderung	41.7 MB 100.4 MB Detelgrolle
Detai Detai DetaingsteG_ga-messatelen-messwerks_EPS628022_CBV.ep CoverHypteG_ga-messatelen in NRW als Solite Date Detai Detai CoverHypteG_ga-messatelen-messwerks_EPS025632_Solite.ap		28:10.2021 08:48:00 28:10.2021 08:49:00 Lettle Antisrung 28:10.2021 09:46:20	41.7 MB 300.4 MB Detelgrofile 86.8 MB
Datai         @ OpenHypteC.gx+measurates-measurate.EPS622002_CDV.zp           @ OpenHypteC.gx+measurates.text         EPS622002_CDV.zp           Daten der Grundwassermessstellen in NRW als Sqilte Date         Daten           Datei         @ OpenHypteC.gx+measurates.enswerte_EPS622832_Sqite.zp           @ OpenHypteC.gx+measurates.texter.EPS622832_Sqite.zp         @ OpenHypteC.gx+measurates.texter.EPS622832_Sqite.zp		28:10.2021 08:48:00 28:10.2021 08:49:00 Lettle Antisrung 28:10.2021 09:46:20	41.7 MB 300.4 MB Detelgrofile 86.8 MB

After the above zip file has been extracted, the four CSV files and the instruction file can be considered, with a strong recommendation to read the instruction file first.

# 2.3.2 Python

Python is a high-level, interpreted, and general-purpose programming language known for its simplicity and readability. It was created by Guido van Rossum and first released in 1991. Python emphasizes code readability and has a design philosophy that emphasizes clear, concise syntax, making it easier to write and understand code. Python is used in several ways such as:

- AI and machine learning
- Data analytics
- Data visualisation
- Programming applications

In this project, we have used Python for data engineering, data pre-processing, and data analysis. Jupyter Notebook is used in this project to write Python codes. The Jupyter Notebook is an opensource web application that data scientists can simply write the code for and make it easier to document. Simply, we can combine Python codes, text, images, comments, and the result of the codes on the same page. The below image shows how code, text, and the result of the code can be seen on a single page.

														The second second		
File	Ed	at V)	ew In	sert Cell	Kernel V	Vidgets	Help						Trusted	Python 3 O		
1	+ 3	R 21	15 +	🔶 🕨 Run	• C »	Markdo	in	< m								
		B	ead t	he file		~	Text									
				t to add the "de	imiter* norm	al cay file	Sec.	leris * * but in c	ur case if s **	so we need	to assign it					
		_	unnorge	. 15 655 616 66	annost , ristro	ar oprine	e o denne	with , but it c	di cube irb ,	, oo no nood	to aborget it.					
	In [			read_csv(gw_s	tation_pfna	me, del	imiter=	';', header=	infer')		-	Python Code				
		df	= pd.r .head()		tation_pfna	me, del	imiter=	';', header=	infer')			Python Code				
	In [: Out[:	df	.head()		tation_pfna	me, del •32		';', header= gw_stockwerk	an a	gemeinde_id	-		beobachtung_t	wasserstand	-	
			.head()	)	nan sa Tanta	•32			an a	gemeinde_id 05111000	gwhorizont_id		., beobachtung_t	wasserstand	٦.	
		d1	.head()	messstelle_id	name	<b>•32</b> 343064	<b>n32</b> 5578019	gw_stockwerk	an a		gwhorizont_id NaN	gwhorizont .	-	wasserstand -	₽	
		d1 0 1	.head() sl_nr 67530	messstelle_id 32505929	name UWB-Ddorf 01285	<b>•32</b> 343064 292077	<b>n32</b> 5678019 5645349	gw_stockwerk 1.0	grundstueck	05111000	gwhorizont_id NaN S	gwhorizont . NaN	-	wasserstand -	₽	

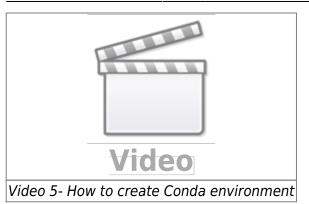
### 2.3.3 Anaconda

Anaconda is an open-source distribution for python and R. It is used for data science, machine learning, deep learning, etc. With the availability of more than 300 libraries for data science, it becomes fairly optimal for any programmer to work on anaconda for data science Anaconda is used in this project. An environment on Anaconda has been created to install all the packages needed for this project.



**Environment in Anaconda:** A conda environment is a directory that contains a specific collection of conda packages that are used in the project.

**How to create Conda environment:** The below video shows how to create a Conda environment, how to activate it, how to install different packages on the environment and how to deactivate the environment.



**Python packages:** Python packages are collections of modules that provide additional functionality and tools to extend the capabilities of the Python programming language. Packages are typically distributed and installed using package managers such as pip (the default package manager for Python) or conda.

To get more details about conda environment, I highly recommend visiting the below webpage.

https://towardsdatascience.com/manage-your-python-virtual-environment-with-conda-a0d2934d5195

**openhype environment:** For this project, an "openhype" environment was created to install all the required packages. This environment ensures that all the necessary dependencies are properly installed and configured.

Several packages are related to data science but in this project, we have used the below packages. There are two ways to install the below packages:

**install a package manually:** In this way you need to install each package manually into openhype environment by the command prompt.

• **pandas:** Pandas is a popular open-source Python library for data manipulation and analysis. It provides easy-to-use data structures, such as DataFrame, Series, and Index, that are designed to handle structured data efficiently.

In this project, pandas was utilized to read the CSV files, clean the data, and perform data engineering tasks. To install Pandas, execute the following Python code within the openhype environment in the Anaconda prompt:

conda install pandas

• **sqlalchemy:** SQLAlchemy is a popular open-source SQL toolkit and Object-Relational Mapping (ORM) library for Python. It provides a set of high-level APIs that allow developers to interact with relational databases using Python code.

conda install sqlalchemy

• **psycopg2:** Psycopg2 is a PostgreSQL adapter for Python. It provides a Python interface for interacting with PostgreSQL databases, allowing developers to connect to a PostgreSQL database server, execute SQL queries, and perform database operations using Python code.

conda install psycopg2

• geopandas: Geopandas is an open-source Python library built on top of Pandas and other

geospatial libraries. It extends the capabilities of Pandas by adding support for geospatial data, enabling users to work with spatial data in a tabular format.

conda install --channel conda-forge geopandas

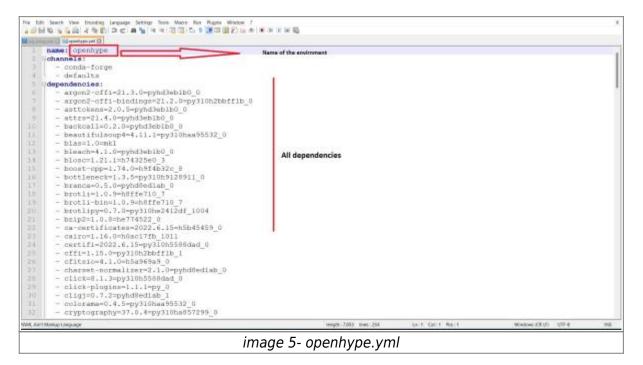
Certain packages require the specification of a channel for installation. This is why the above code includes the channel specification to ensure the correct installation.

• **jupyter notebook:** Jupyter Notebook is an open-source web-based interactive computing environment that allows users to create and share documents containing live code, visualizations, explanatory text, and more. It supports various programming languages, including Python, R, and Julia.

conda install jupyter notebook

To ensure the successful installation of all the aforementioned packages, it is crucial to install them within the openhype environment using the Anaconda prompt.

**Load all the packages into the environment by a YAML file:** To streamline the installation process for all the required packages in this project, it is recommended to create an environment named "Openhype" and load a YAML file containing the package specifications. The contents of the "openhype.yml" file, as shown in the image below, encompass all the necessary packages.



The following code allows you to create an environment based on the yml file. Keep in mind that the YAML file should be in the same directory as your Anaconda installation. If the file is located elsewhere, you will need to provide the full path to the file in the code. openhype.yml can be found from here

```
conda env create -f openhype.yml
```

How to create a yml file for the environment: With the below code you can export the yml file from the existing environment.

conda env export > openhype.yml

Since the four CSV files have been downloaded in the previous chapter, it is time to read the CSV files and initiate the cleaning process to make them ready for importation into our database.

Refer to the code section in our Notebook for more information Github (the below image).

Search or jump to	Pull requests issues Marketplace Exp	skare		0 +*	6-
E sina7272 / LANUV_Groundwater	(Rade)			St Me Staventh I + 🛛 👘 Stavent I	
⊖ Code _ ☉ Issaes _ □\ Poli requests	🔿 Actions 🛛 Projects 🗔 Wiki	© Security: I≓ Insights © Settings			
	Primain - Prisent Satage		Go to file Add file - Code -	About ()	
	ena7272 update		Milleli 2 hours app (5)22 mmmite	Ground water in NRW	
	Apyrib, checkpoints				
	🐚 Natebook				
	📷 data				
	<b>a</b> 494			Relation	
	D giatributes				
	D README.ord			Examine a name remainer	
	READMEINS	vater		Packages No initiajn politikod Facility yar kni politige	
	Ground water in NRW			Languages	
	Demosaria ina m			Inggive Rodeback 5175.      Idek 575	
		image	6- Github		

There below four notebooks should be run separately, in order to import data into the database.

• import\_gemeinde.ipynb:

In this notebook, we will import the data of all geminde into the database.

• import\_katalog\_stoff.ipynb:

In this notebook, we will import the data of all the catalogue substances into the database.

• import\_messstelle.ipynb:

In this notebook, we will import the data of all stations into the database.

• import\_messwert.ipynb:

In this notebook, we will import the data of all values into the database.

### 2.4 Observation Data in the Database

After successfully downloading, cleaning, and importing the data into the database in the previous section, it is now time to examine the data within the database. Since we are working with a PostgreSQL database, we can utilize SQL commands to retrieve and analyze the data. Let's run some basic SQL commands to gain insights into the data.

To view our tables, we can use the following code. It selects all the columns from the "hygrisc" schema, which contains our tables. We limit the output to the first 100 rows to avoid overwhelming results.

select \* from hygrisc.messwert limit 100;

Image 7 illustrates the outcome of the aforementioned command executed in PGAdmin. It displays the resulting data obtained from the execution of the provided SQL command.

	5 III Ta Q	Dashb	oard Proj	perties SC	L Stat	istics Dep	endencies	Deper	ndents		etwas/etwa	sizelwa	17										
X 🚱 Domains		8		I Q			8 2'-	т		Nole	nit. 🛩		- 4		- 15	6	đ× I	£	18.×				
ETS Configure		6 1	was/etwood	Selwas +	1	1000									-								
FTS Dictioner	ries	Reading and	Editor Qu																	cratch P	and a		
> An FTS Parsers																			1	ciarci r	P.C.		
<ul> <li>FTS Template</li> <li>Honsign Table</li> </ul>		1	elect =	from sina	. nessae	ert limit	100;																
<ul> <li>b) Functions</li> </ul>	50 																						
> Materialized \	Views																						
> () Procedures																							
<ul> <li>Cr Sequences</li> </ul>																							
🛩 🛅 Tableo (6)																							
⇒ 🛅 gw_statio	ris.																						
> 🖽 katalog.g	e																						
🔉 🖽 katalog_at																							
measurem																							
> 📑 mesisteli																							
> 🖂 messwert																							
						No. 10 Contractor																	
X by Trigger Funct	tors	Data O	httput Exc	ilain Mes	19398	NOULCHIOUI																	
Этрена	5075	Data O						1	the Party				and the second		-	autorio b	and the second se		Restinger man	-	mananthill	manuel	1
> 😨 Турна ~ 🧱 Viewa (7)		-	sLor (PK)bigint	Aain Mes messate bagint		pha.id		1	w, Note	1	probesgut sist	/ mes	ergebeias le precision	1	messer tixt	gebnia,3	inveis	/	bestimmungsg double prectain	prentae on	/ massesthalt	/ trensverfa	abram.
Этрена		-	sLer	<ul> <li>messate bagint</li> </ul>	au /	, piane ,	, datam_pn date		agen	-17	probesgut sint Grundwasser	/ mes	se precision	/		gitnis,)	inveis	1	Bestimmungsg double precisie		/ massestheit text	/ trenswerfa	
> C Types > Venue (7) > groupby_s	station_result	-	sl.or [PK]bigint	<ul> <li>messate</li> <li>bigmt</li> </ul>	20900271	pre_id /	datum, pri date 1992-05-21		agen. 1	244 (	893	/ mes	ie precision		[12.0]	gebnia_8	inveis	'	Bestimmungsg double precisio	3	and a second	004	filter
> Types Weens (7) > Groupby.s > retrat	station,result		sl.or [PK] bigint 122890	Messeta bagint 151	199002710 299002710 289002810	pre_id text 0 3/1992/16	datum_pn date 1992-05-21 1994-03-29		1	244 (	ann Grundwasser	/ mes (0)/	le pecisión 1	05.248	(n.i) (n.i)	gebnis,ð	invelle	'	bestimmungsg double prectain	1	high (jun)	Membran	fiher filter
Types     Weess (7)     Weess (7)     intrat     intrat.geo     Ficular     Ecolum     Ecolum     Ecolum	station, result on	1 2	sLer [PK]bigint 122090 122090	messate     tagint	299002714 289002714 289002914 8030029	<ul> <li>pna.id hext</li> <li>3/1992/16</li> <li>3/1994/16</li> </ul>	datum.pn date 1992-05-21 1994-03-29 1993-01-11		1 1 1	244 ( 244 ( 244 (	ann Grundwasser Grundwasser	/ mes	le pectatori 1	06.248 61.978	(mail) (mail) (mail)	gebnic.3	inweis	,	bestimmunging couble precisio		figer (luc) figer (luc)	Membran	fiher filter filter
Types	station, result on	1 2 3	<b>sLor</b> [PK]bigint 122090 122090 122090	messata     tagint     163     172	6.14 209002710 209002010 8030030 8030030	<ul> <li>preuid</li> <li>hixt</li> <li>3/1992/16.</li> <li>3/1993/16.</li> </ul>	datum.pn date 1992-05-21 1994-05-29 1995-01-11 1992-11-02		1 1 1	244 ( 244 ( 244 ( 244 (	ann Grundwasser Grundwasser Grundwasser	/ mes	1	06.248 d1.978 3.0989	(mi) (mi) (mi) (mi) (mi)	getinic_3	inveis	,	Bestimmunging double precisio		indi ingi ingi ingi ingi	Membrari Membrari Membrari	fiher fiher fiher fiher
Prove     Venes (7)     Venes (7)     Venes (7)     venes (7)     retust     venes (7)     vene	station_result con ons	4 1 2 3 4	sLer [P4]bigint 122090 122090 122090 122090	messsta bigint     10     1	209002710 2090022710 209002010 80300000 80300000	<ul> <li>preuid</li> <li>text</li> <li>3/1992/16</li> <li>3/1993/16</li> <li>3/1992/16</li> </ul>	datam.pn date 1992-05-21 1994-03-29 1993-01-11 1992-11-02 1993-12-22		1 1 1 1 1	244 ( 244 ( 244 ( 244 ( 244 (	ann Grundwasser Grundwasser Grundwasser Grundwasser	/ mes	1 1 2 3	05.249 61.978 3.0989 25777	(mir) (mir) (mir) (mir) (mir) (mir)	gabria.3	inviii	'	Sestimmungeg Couble prechin		inali mgi inali mgi inali mgi inali mgi	Membran Membran Membran	the film film film film
Types     Venes(7)     Venes(7)     Groupbyz     fritut_poo     Ficture     Ficture     Sector     Sector	station_result con ons	4 7 3 4 5	9Lav (PK)bigint 122890 122890 122890 122890 122890	<ul> <li>messetta lagint</li> <li>193</li> <li>1953</li> <li>1953</li> <li>1953</li> <li>1953</li> <li>1953</li> <li>1954</li> <li>1954<td>269002710 2890022710 803002070 80300300 80300300 80300301</td><td>phajid feat 3/1992/16. 4/1994/16. 7/1993/16. 3/1992/16. 0//1993/16.</td><td>datar.ph date 1992-05-21 1994-03-29 1993-01-11 1992-11-02 1993-12-22 1993-04-21</td><td></td><td>1 1 1 1 1</td><td>244 ( 244 ( 244 ( 244 ( 244 ( 244 ( 244 (</td><td>ann Grundwasser Grundwasser Grundwasser Grundwasser</td><td>/ mes dout</td><td>1 2 3 1</td><td>05.248 61.978 3.0989 25777 .05403</td><td>(nur) (nur) (nur) (nur) (nur) (nur)</td><td>gebrie.3</td><td>inveis</td><td>'</td><td>bestimmungsg Guldle prectain</td><td></td><td>inuli mgri inuli mgri inuli mgri inuli mgri inuli mgri</td><td>Membran Membran Membran Membran</td><td>filter filter filter filter filter</td></li></ul>	269002710 2890022710 803002070 80300300 80300300 80300301	phajid feat 3/1992/16. 4/1994/16. 7/1993/16. 3/1992/16. 0//1993/16.	datar.ph date 1992-05-21 1994-03-29 1993-01-11 1992-11-02 1993-12-22 1993-04-21		1 1 1 1 1	244 ( 244 ( 244 ( 244 ( 244 ( 244 ( 244 (	ann Grundwasser Grundwasser Grundwasser Grundwasser	/ mes dout	1 2 3 1	05.248 61.978 3.0989 25777 .05403	(nur) (nur) (nur) (nur) (nur) (nur)	gebrie.3	inveis	'	bestimmungsg Guldle prectain		inuli mgri inuli mgri inuli mgri inuli mgri inuli mgri	Membran Membran Membran Membran	filter filter filter filter filter
<ul> <li>Types</li> <li>Venes (7)</li> <li>groupby, t</li> <li>retuit</li> <li>retuit</li> <li>retuit</li> <li>retuit</li> <li>retuit</li> <li>temp</li> <li>suffat, peo</li> <li>suffat, peo</li> <li>suffat, peo</li> <li>temp</li> <li>temp</li> <li>temp</li> </ul>	station_result con ons	1 2 3 4 5 6 7	sl.sv [PK]bigint 122890 122890 122890 122890 122890 122890 122890	<ul> <li>messate bagint</li> <li>163</li> <li>172</li> <li>177</li> <li>177</li> <li>179</li> <li>181</li> <li>197</li> <li>197</li> </ul>	1099002714 2099002714 209900207 80390030 80390030 80390030 80390030 80390035	<ul> <li>pha.id host</li> <li>3/1992/16.</li> <li>3/1992/16.</li> <li>3/1993/16.</li> <li>3/1992/16.</li> <li>3/1992/16.</li> <li>3/1992/16.</li> <li>3/1992/16.</li> <li>3/1993/16.</li> </ul>	datum_pri date 1992-05-21 1994-05-29 1993-01-11 1992-11-02 1993-12-22 1993-12-22 1993-04-21 1993-07-19	-	1 1 1 1 1 1	244 ( 244 ( 244 ( 244 ( 244 ( 244 ( 244 ( 244 (	aux Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser	/ mes doz	1 2 3 1 1	06.248 61.978 3.0989 25777 05403 1.0675 06.248	(m) (m) (m) (m) (m) (m) (m) (m)	gebrie_8	inveis	,	bestimmungsg double precisio		tign (jun) tign (jun) tign (jun) tign (jun) tign (jun) tign (jun)	Membran Membran Membran Membran Membran	the film film film film film film
<ul> <li>Types</li> <li>Vress(7)</li> <li>groups(7)</li> <li>groups(7)</li> <li>metad</li> <li>metad</li> <li>ficial peop</li> <li>ficial p</li></ul>	station_result con ons	1 2 3 4 5 6	sl.sv [PK]bigint 122890 122890 122890 122890 122890 122890 122890 122890 122890	<ul> <li>messate bigint</li> <li>163</li> <li>1672</li> <li>1679</li> <li>1670</li> <li>1671</li> <li>1670</li> <li>1670<td>6030036 8030030 8030030 8030030 8030030 8030030 8030036 8030036</td><td>pha.id hert 3/1992/16. 3/1993/16. 3/1993/16. 3/1992/16. 3/1992/16. 3/1993/16. 3/1993/16. 3/1993/16. 3/1993/16.</td><td>datum_pri date 1992-05-21 1994-05-21 1993-05-21 1992-05-21 1992-11-02 1993-12-22 1993-12-22 1992-04-21 1992-05-10</td><td></td><td>1 1 1 1 1 1 1</td><td>244 ( 244 ( 244 ( 244 ( 244 ( 244 ( 244 ( 244 (</td><td>aux Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser</td><td>/ mes dout</td><td>1 2 3 1 1</td><td>06.249 61.978 3.0989 25777 05403 1.0675 06.249 53.124</td><td>1001 (1001) (1001) (1001) (1001) (1001) (1001) (1001)</td><td>gebrie.3</td><td></td><td></td><td>couble precise</td><td></td><td>tion fight (juid fight (juid fight (juid fight (juid fight (juid fight (juid) fight (juid)</td><td>Menibrari Menibrari Menibrari Menibrari Menibrari Menibrari</td><td>filter filter filter filter filter filter</td></li></ul>	6030036 8030030 8030030 8030030 8030030 8030030 8030036 8030036	pha.id hert 3/1992/16. 3/1993/16. 3/1993/16. 3/1992/16. 3/1992/16. 3/1993/16. 3/1993/16. 3/1993/16. 3/1993/16.	datum_pri date 1992-05-21 1994-05-21 1993-05-21 1992-05-21 1992-11-02 1993-12-22 1993-12-22 1992-04-21 1992-05-10		1 1 1 1 1 1 1	244 ( 244 ( 244 ( 244 ( 244 ( 244 ( 244 ( 244 (	aux Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser	/ mes dout	1 2 3 1 1	06.249 61.978 3.0989 25777 05403 1.0675 06.249 53.124	1001 (1001) (1001) (1001) (1001) (1001) (1001) (1001)	gebrie.3			couble precise		tion fight (juid fight (juid fight (juid fight (juid fight (juid fight (juid) fight (juid)	Menibrari Menibrari Menibrari Menibrari Menibrari Menibrari	filter filter filter filter filter filter
<ul> <li>Types</li> <li>Wrenst (7)</li> <li>groupby,t</li> <li>motion</li> <li>motion</li> <li>mitian</li> <li>mitian</li> <li>mitian</li> <li>groupby,t</li> <li>Transit</li> <li>groupby,t</li> <li>Transit</li> <li>groupby,t</li> <li>Transit</li> <li>groupby,t</li> <li>Transit</li> <li>groupby,t</li> <li>transit</li> <li>temp</li> <li>temp</li> </ul>	station_result con ons	1 2 3 4 5 6 7	sl.sv [PK]bigint 122890 122890 122890 122890 122890 122890 122890	<ul> <li>messate bigint</li> <li>163</li> <li>1672</li> <li>1679</li> <li>1670</li> <li>1671</li> <li>1670</li> <li>1670<td>6030036 8030030 8030030 8030030 8030030 8030030 8030036 8030036</td><td><ul> <li>pha.id host</li> <li>3/1992/16.</li> <li>3/1992/16.</li> <li>3/1993/16.</li> <li>3/1992/16.</li> <li>3/1992/16.</li> <li>3/1992/16.</li> <li>3/1992/16.</li> <li>3/1993/16.</li> </ul></td><td>datum_pri date 1992-05-21 1993-05-21 1993-05-21 1993-05-22 1993-12-22 1993-12-22 1993-12-22 1993-04-21 1993-05-10</td><td></td><td>1 1 1 1 1 1 1</td><td>244 ( 244 ( 244 ( 244 ( 244 ( 244 ( 244 ( 244 (</td><td>aux Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser</td><td>/ mes dout</td><td>1 2 3 1 1</td><td>06.248 61.978 3.0989 25777 05403 1.0675 06.248</td><td>1001 (1001) (1001) (1001) (1001) (1001) (1001) (1001)</td><td>gebrie_8</td><td></td><td></td><td>couble precise</td><td></td><td>tign (jun) tign (jun) tign (jun) tign (jun) tign (jun) tign (jun)</td><td>Menibrari Menibrari Menibrari Menibrari Menibrari Menibrari</td><td>filter filter filter filter filter filter filter</td></li></ul>	6030036 8030030 8030030 8030030 8030030 8030030 8030036 8030036	<ul> <li>pha.id host</li> <li>3/1992/16.</li> <li>3/1992/16.</li> <li>3/1993/16.</li> <li>3/1992/16.</li> <li>3/1992/16.</li> <li>3/1992/16.</li> <li>3/1992/16.</li> <li>3/1993/16.</li> </ul>	datum_pri date 1992-05-21 1993-05-21 1993-05-21 1993-05-22 1993-12-22 1993-12-22 1993-12-22 1993-04-21 1993-05-10		1 1 1 1 1 1 1	244 ( 244 ( 244 ( 244 ( 244 ( 244 ( 244 ( 244 (	aux Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser	/ mes dout	1 2 3 1 1	06.248 61.978 3.0989 25777 05403 1.0675 06.248	1001 (1001) (1001) (1001) (1001) (1001) (1001) (1001)	gebrie_8			couble precise		tign (jun) tign (jun) tign (jun) tign (jun) tign (jun) tign (jun)	Menibrari Menibrari Menibrari Menibrari Menibrari Menibrari	filter filter filter filter filter filter filter

image 7- messwert table in databse

Now with the above SQL command, we are able to see the other three tables that we have (The below codes).

select \* from hygrisc.messstelle limit 100;

select \* from hygrisc.katalog\_stoff limit 100;

select \* from hygrisc.katalog\_ge limit 100;

Now we want to see more details for our tables and we will run the below codes.

**Count the rows of each table:** With the below code, we can see how many rows we have in each table.

select count (\*) from hygrisc.messwert;

select count (\*) from hygrisc.messstelle;

**Filter the data based on Nitrate only:** To determine the substance number of Nitrate, we need to retrieve the corresponding information from the database.

select \* from hygrisc.katalog\_stoff where name like 'Ni%';

	III To Q Dashboard Properties SQL Stat	ticz Dependencies Dependents 🛃 elusar; visus: Belvar *		
<ul> <li>ab beind</li> </ul>	s = a - III Q - #	- 5 8 2 - T - home - 8	A B-	
+ @ sha + E-Colutions	S chemisterageterm +			
> Re-Donwins	Query Elifier Query History		Scritch Part.	
<ul> <li>ID.FTS Configurations</li> </ul>			scheck (ver	
+ Its FTS Dictionaries	1 select + from size Astalo	Lucoff where name 14bg (Nrb)		
<ul> <li>Av PTS/Packers</li> </ul>				
+ 📰 FTS Templates				
<ul> <li>IIII Foreign Tablee</li> </ul>				
<ul> <li>BitPlanctiona</li> </ul>				
<ul> <li>Materialized Value</li> </ul>				
<ul> <li>Ephocedures</li> </ul>				
+ 1.15equerices				
- I Tables (8)				
<ul> <li>Impoundations</li> <li>Implementation</li> </ul>	1			
> III katalog_stuff				
<ul> <li>El riespuramenta</li> </ul>				
· mirrennriele				
<ul> <li>Immessivert</li> </ul>				
> Bit Trigger Panchers				
+ 🐑 Types				
🛥 📴 views (7)	Dete Output Explain Messages	ioth atom		
# groupby_stations,result	status , surger	a bescheitung a ces.M a		
e nitzał	🖌 biger 🏪 text	bescheitung     diske precision     text     text		
- Writtat_geom	7 1143 Not	(ull 746431		
+ Dolumna	2 1188 100kel	Male THERE &		
> 2+11/00mm	1 1244 fitter	INCOME.		
+ Danitat	4 1245 Istnet-Dickatoff	INC. INC.		
+ matta_peon	m T248 Mercel	hell hell		
+ terrer	6 1347 term-Sticketert	NE INT		
+ 📻 terrip2	7 1967 Nontoniale Texase	but but		
F 🐵 tipel	8 2010 Neokergol	buil 08000		
+ - toper_darte	h ziniš terzchen	Juli 1826-7845		
> (8) topology	10 2507 Temberressophure (M72			
<ul> <li>m printiples</li> </ul>	11 2768 Hispatheren	148 11001A.		
<ul> <li>miteregiate_poolges</li> </ul>	12 BETa tatuorettan	140 76424		
<ul> <li>Internap</li> <li>Technology Roles</li> </ul>	13 4164 Hidssemit	hall \$5457		
	14 4158 titlenstrem	10044		
Indiespiecko				

According to the information presented in "Image 8," the substance number (stoff\_nr) of Nitrate is **1244** 

select \* from hygrisc.messwert where stoff\_nr = '1244';

Now we can filter the messwert table based on Nitrate. In this step, We can save this new table as a new "view".

**What is a view:** a view is a virtual table that is derived from one or more existing tables or other views. A view does not store data physically but rather provides a way to present data from underlying tables in a structured and organized manner. It acts as a predefined query that can be used to retrieve and manipulate data. The below code creates views:

```
create view hygrisc.nitrat as (select * from hygrisc.messwert where stoff_nr
= '1244');
```

We now have a "nitrat" view, which can be accessed just like a table using the following code. This view is filter of our messwert table based on "1244" which is "Nitrate"

select \* from hygrisc.nitrat ;

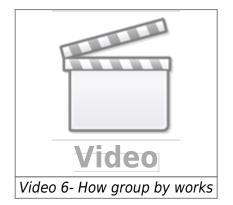
**Group by the two tables:** the Group by clause is used to group rows based on one or more columns in a table. When working with two tables, you can perform a GROUP BY operation to group the data based on common values from both tables. In this section, we want to group by our two tables (messwert and messstelle) only in Nitrate. These two tables have a column **messstelle\_id** which means station id.

```
select messstelle_id, count(*) from hygrisc.messwert where stoff_nr =
'1244' group by messstelle_id;
```

<ul> <li>an broket</li> </ul>		8 - 10	Q - # - 1	B 2- T - she v B F + 4 D - 5 5 d- 4 R-		
+ @ 1010 + EL Colladiante	A day	voluce@elwas -	1			
+ A Domana	and the second sec	or Query Hator			Scrietzh Paul	
+ D.FTS Configurations	-	_			ochron Piel	
+ DLF75Dictionatee	1 sels	out mesostalle	the count(+)	from sine.mesowert where stoff,nr = "1244" group by mesostelle_id		
+ A+FTS Farsers						
+ 💭 FTS Templates						
<ul> <li>III Foreign Tables</li> </ul>						
+ Wehnstone						
+ 🛄 bitaterizitzent Veivez						
+ C)Procedures						
<ul> <li>N.R Secondors</li> </ul>						
- III Tables (0)						
<ul> <li>III gw_stations</li> </ul>	1.1					
s minatelog.ge						
> Ittikatalog_stoff						
<ul> <li>Esmeasurements</li> </ul>						
ittrazzatelie						
<ul> <li>messuet</li> </ul>						
+ Big Trigger Functions + Int Types						
- Views (7)						
encoded and the station result	Dyla Outpi	ut Explain M	ranges tectfication	- 679		
e Criteri		essenteta de 1	auti a			
- Cathat, peom	4.50					
+ 🛅 Column		10131310	24			
e em Faules	- 2	18290408				
+ 2+ Triggers	- T	18290411	28			
> Cartat		(820069)	- 59			
> suitet_peore	- A.S.	10200710	1.1			
> torp	4	18280806	20			
+ 🚺 tergil	T. T.	12200940	1.12			
<ul> <li>ib tiper</li> </ul>		10391039	- 10			
<ul> <li>Ide tiger_chata</li> </ul>		18221117	0			
<ul> <li>B topology</li> </ul>	18	10201202	-			
> Tr postpres	11	(8281294	28			
<ul> <li>Sternete.postgo</li> <li>Sternep</li> </ul>	12	10291306	28			
Ga Login/Ursup Roles	- 13	18291912	42		L an and a station of	
		10401214			<ul> <li>Buccessifully run. Total guery runt</li> </ul>	me. 971 mper. 2632 rows affected

Image 9 shows that each station id has how many single measurements for the Nitrate only. I highly recommend opening the below website to get more deep into how "group by" works and how we can use it.

https://www.w3schools.com/sql/sql\_orderby.asp



#### Maximum date in nitrat table:

select \* from hygrisc.nitrat where datum\_pn = (select max(datum\_pn) from hygrisc.nitrat); 2024/05/19 02:17

13/21

Admin File + Object + Tools + Help +		
	🎟 🐂 Q. Cashboard Properties IVL. Instatus Dependencies Dependences 🗈 envision wargetives."	
<ul> <li></li></ul>	5 5 8 - 2 0 - 5 - 5 E B' T - 100 - 0 E F - 5 B - 5 B - 5 B.	
> \$3 California	gg ofmassionagetwar =	
+ ilg Domaina	Curry Editor Query History Scratch Pa	1
<ul> <li>() FT3 Configurations</li> </ul>	1 whet + from sing_nitrat where detum_pn = (select max/detum_pn) from sing_nitrat)	
<ul> <li>M.PTS Sutionaries</li> <li>Au/PTS Partners</li> </ul>		
ALFISPERE     FT3 Templates		
· Ef Porsign Tables		
Info Functions		
Materialized Views		
<ul> <li>I / Procedulem</li> </ul>		
<ul> <li>L±Eequences</li> <li>Tableo (f)</li> </ul>		
+ Ellips, stations		
i- III) katalog_ge		
+ milkatang.staff		
<ul> <li>Interactive</li> <li>Interactive</li> </ul>		
+ minespert		
> NE Trigger Functions		
> ID Tuties		
~ 🔤 Wares (7)	Data Output Explain Messages Notifications	
<ul> <li>B goupty_station_result</li> <li>minut</li> </ul>	star a mananita, i pravi a delar, a poste a delar a poste a manapolita, i manapolita, i delar a delar a manapolita a delar a d	an a vision a feelant a staat, dat
+ nital geon	A side of a side of a side of a side of a	a mic a ped a dete
<ul> <li>Columna</li> </ul>	1 17721101 94190773 1/2021/11 102:0017 1/20 0wewsee 41 (Jul) (Jul 9aut Desergeset See	August 60.0001079 3152-04-0
* TR. Marian		
+ \$+ Triggers		
+ suitet		
> terra		
+ tervez		
+ -@ tipe:		
<ul> <li>Settiger_state</li> </ul>		
<ul> <li>Grapology</li> <li>Epology</li> </ul>		
<ul> <li>If temptate_portple</li> </ul>		
> 否filemad		
<ul> <li>/b_Login/Group Roles</li> </ul>		
> Tatletaret		
	Image 10- Maximum of the date in the nitrat table	
	maye to maximum of the date in the model table	

As we can see in Image 10, the maximum date is 2021-08-17

#### Minimum date in nitrat table:

select \* from hygrisc.nitrat where datum\_pn = (select min(datum\_pn) from hygrisc.nitrat);

ASH	🎟 🖡 🔍 Dashboard Properties 90k. Statistics: Dependencies: Dependencie	
<ul> <li>so patiet</li> </ul>	8 6 8 4 8 0, 4 8 4 6 8 8 7 4 4 10 4 8 6 4 5 6 4 4 8 4	
<ul> <li>Orientes</li> <li>Statistics</li> </ul>	S Johnan Johnangerinan -	
+ By Contains	Built mediate and the second	Solutch Piel
<ul> <li>D-PTS Contigurations</li> </ul>		3078001981
FTS Dictionaries	3 select + free wine.nitrat.etere detum.pn - (select ein(detum.pn) free wine.nitrat);	
+ ##FTS Parsero		
<ul> <li>If Its Templetes</li> </ul>		
<ul> <li>IE Foreign Tables</li> <li>Ib Functions</li> </ul>		
<ul> <li>Contractions</li> <li>Contractions</li> <li>Contractions</li> </ul>		
> () Propertures		
+ 1.3 Sequences		
- 🦲 Tabiya (1)		
<ul> <li>migw_stations</li> </ul>		
<ul> <li>El valanto-ter</li> </ul>		
Plofe, governa 🛅 🔹		
<ul> <li>Immenuements</li> <li>Immenuefale</li> </ul>		
+ III measured		
> 89 Tripper Functione		
> C) Types		
- 📫 Vience (7)	Data Output Explain Messages Notifications	
+ 🔤 proupito_station_result	ity a meaning a proj a destar a soft a present a meaning to the best a between a sector a between a meaning to work the	e a vertainer a rot,ort a herkunft a alsaal,dar
+ etal	A shot who are the shot the second burder and second burder. The second burder	• test • test • test • zee
<ul> <li>E Oslamas</li> </ul>	1 1221403 Totista Statute 1224 December 1.6499 Juli Juli regi Decempera	a pull integrate purche
> ======		
+ 3+Trippers		
+ 🔛 mattat		
+ 🧱 suffat,geim		
+ terrar		
+ -to tax		
<ul> <li></li></ul>		
+ for topology		
<ul> <li>Itt postgrep</li> </ul>		
> mit template_accepter.		
> III herrigo		
<ul> <li>-Ib.Logm/Group Roles</li> </ul>		
<ul> <li>M Tablespaces</li> </ul>		

As we can see in Image 11, the minimum date is 1951-04-30

**Create geometry column in messstelle table:** In this section, we want to create a geometry column from **e32** and **n32** columns from the messstelle table. With the below code, we are able to create a new column and we set the name as **geom** 

```
ALTER TABLE hygrisc.messstelle ADD COLUMN geom geometry(Point, 25832);
UPDATE hygrisc.messstelle SET geom = ST_SetSRID(ST_MakePoint(e32, n32),
```

#### 25832);

	5	Q Dash	sboard Pro	perties 5	QL Statistics	Depende	ncies Dependents	🖬 elwar	s/elwas@ietwas *								
	() public	8			Q ~ # ~ I		B'- T -	feritui		4 0		4. 7 F	-				
~	<li>sina</li>	100	elwas/ofwas	Selwas v	-	-	No. of Concession, Name		-		-	and the second	-				
	A Collations	Discourse of	y Editor Qu		NY									cratch Pad			
	B Domains     B FTS Configurations				a.mensatelle									Charlen and			
	> 📰 katalog_stoff > 🔝 measurements	1.0	Output Ex		ssages Notifica												
	til katalog_stoff     itil measurements     itil messstelle	1.0	Output Ex	e,om a	ssages Notifica sumpfrohrleinge, double precision		ecebeudarchmesser,mm double precision		historischer, ruhe, wsp double precision		embisulaenge.cm double precision	double precision		unterkente, filter, om double precision		geometry	•
	itikatalog_stoff     itikatalog_stoff     itikatalog_stoff     itikatalog_stoff     itikatalog_stoff     itikatalog_stoff     itikatalog_stoff     itikatalog_stoff     itikatalog_stoff	1.0	filterlaenge double pre	e,om a	sumpfrohrleinge, double precision		double precision	A Indi	Instantacher, ruhe, wap double precision	A (sur)	and/out-buendout	obehante, filter, double precision (not)	em e	ODDANE DAGISSION	100	geometry	
	til katalog_stoff     itil measurements     itil messstelle	siber	filteriaeng double pre	t,om	sumpfrahrleenge, double precision	cm 🍙	double precision		historischer,ruhe,wap double precision		accos precision			li Grade trenders	but		6
	Theoremain of the second	siber #4 med	filtertaenge double pre te.	tjam ciston (hui	sumpfrohrlaunge, double precision	in A	double precision	pul	lastorischer, ruhe, wop double precision	0.0	and the frequency	pall	Jul	9	(huit) 9349	0101000020E86	6
	the late of the set of the s	iber Nimed I Padet	filtertaenge double pre te.	icision (hul) 100	aunisfishtionge, double precision	a no Rođ Rođ	double precision	tul  115	Nationacher, ruhe, wop double precision	indi indi	and a free of the	(1000) (1	)rul 1034	9 1]	(nult) 9349 (nult)	0101000020EB6 0101000020EB6	6 6 6
	Katalog_stoff     Katalog_stoff     Kite measurements     Timessurements     Timessurement     Timessurement     Timessurement     Timessurement     Timessurement     Timessurement     Timessurement     Timessurement     Timessurement	iber Nimed I Padet	A filteriseng double pre to.	tjem cision  hul 100  hul	sumpfrohrbenge, double precision	ind Jud Jud	double precision	(null) 115 (null)	historischer, nuhe, wep double precision	lter) Heri Jteri	and the provident	(m) 5000 (m)	jud 1054 jud	9 11 5	(huit) 9349 (huit) 7505	0101000020EB6 0101000020EB6 0101000020EB6	6 6 6
	the late of the second se	siber RV med 1 Padet 1 Delbri	filteriaenge double pre	t,em (tston (tul 100 (tul (tul (tul	sumpfrohrlaunge, double precision	ind Bud Bud Bud Bud	double precision	101  115  741   741	Restorischer, nahe, wop double precision	164) 164) [64] [64]	Sandare precision	[168] 2000 [168] 635	(nu) 1034 (nu) 759	00000000000 9 9 11 5	(nul) 9349 (nul) 7505 (nul)	0101000020EB6 0101000020EB6 0101000020EB6 0101000020EB6	5 5 5
	the late of the set of the s	siber R4 med 1 Padet 1 Debu	filteriaenge double pre	totation (Fruit) (Fruit) (Fruit) (Fruit) (Fruit) (Fruit)	sumpfrohrlaunge, double precision	a m Buri Buri Buri Buri Buri	double precision	(rul) (rul) (rul) (rul)	Bistorischer, nahe, wep double precision	164) 164) [64] [64] [64]	Sandare precision	(ull) (ull) (ull) (ull) (ull)	(nu) 1034 (nu) 750 (nu)	000000 (1000000) 10 10 10 10 20	(hult) 9349 (hult) 7505 (hult) 3452	0101000020EB6 0101000020EB6 0101000020EB6 0101000020EB6 0101000020EB6	5 5 5 5 5
	the late of the second se	siber R4 med 1 Padet 1 Debu	fiberiaenge double pre      te.      L      L      L      L	ecomini (hul) 1000 (hul) (hul) (hul) (hul) (hul) (hul) (hul) (hul) (hul) (hul) (hul)	satisfishtionge, double precision 1 1 1 1 1 1 1	ind Bod Bod Bod Bod Bod Bod	double precision	(rul) 113 (rul) (rul) (rul) (rul)	Battorischer, ruhe, wsp double precision	164) 164) 164) 164) 164)	annexe factorian	1000 (1000) 2000 [000] 1035 [000] 2000	jnul 1034 jnul 750 jnul 345	00000 (140000) 10 10 10 10 10 10 10 10 10 10	(hult) 9349 (hult) 7505 (hult) 3452 3100	0101000020E86 0101000020E86 0101000020E86 0101000020E86 0101000020E86 0101000020E86	0 0 0 0 0

The "messstelle" table has been enhanced with an additional column called "geom," which contains the geometry information representing the location of each station.

#### Merge two tables:

In this section, we aim to merge the "messwert" and "messstelle" tables based on the common column, "messstelle\_id." To achieve this, we will select the desired columns from each table and then perform the merge based on the "messstelle\_id" column.

```
select t1."messstelle_id", t1."name", t1.geom, t2."stoff_nr",
t2."messergebnis_c", t2."masseinheit",
t2."datum_pn", t2."messergebnis_cm" from hygrisc.messstelle t1 ,
hygrisc.nitrat t2
where t1."messstelle_id" = t2."messstelle_id";
```

nter BIII 1,					B dans i dans baan '				
* @ puble	8 10 1	9 - II Q -	# * % B	B 1	10.002 - # P	* * = *	5.5 4	a (8)	
* @ una * El Collations	5 stant	alvanjjetena -							
<ul> <li>Sp Domains</li> </ul>	Gump Rither	Query History						Scratch Pad	
F D FTS Configurations	1 miles	a of function in a	and the state of the	the second state of the	off_ort_ t2.tessings	and the second second	Second shadoo 1		
5 Br F71Detoniaries					starti, sina.ettrat		second to be a		
6 da F75 Painera		tt. Transfalle To			and the property of the	1000			
+ CFTS Templates									
k Ellinwige Tables									
<ul> <li>bis Parachieres</li> </ul>									
Fill RAsteriolized Weeves .									
<ul> <li>I() Procedures</li> </ul>									
<ul> <li>t.iflequences</li> </ul>									
+ CTables (S)									
- No. 2011 Alastic Additionals									
<ul> <li>211 ge_stations</li> <li>01 katalog op</li> </ul>									
<ul> <li>Imitiatulog.ge</li> </ul>	-								
	Data Gidad	Explain Messager	Notifications						
<ul> <li>(E)-Latalog.ge</li> <li>(E)-Latalog.scott</li> </ul>		sostalia, id a suma a				a 1000.00	a rassignetik.cm		
Initiating pr     (if hashing cont     (if hashing cont     (if hashing cont     (if research method     (if research     (if research		enter a trans	Storety + 💽	nged 🔹 anam	hardened		e menegebrik.am	•	
+ (0) kurtalog, pr + (1) kurtalog, stort + (1) kurtalog, stort + (1) reseaucement + (1) reseaucement + (1) reseaucement + (1) rigger Functions		interiorational de la companya de la comp	ponety + 💽	1344	100.245 repl	1800-09-22	1.	•	
<ul> <li>I formation, get</li> <li>If handling, not1</li> <li>If transactionments</li> <li>If transactionments</li> <li>If transaction</li> <li>If transaction</li> <li>If transaction</li> <li>If transaction</li> <li>If transaction</li> <li>If transaction</li> </ul>		and alle. M A Harris A and A A A A A A A A A A A A A A A A A A A	promety a set of a construction of a constructio	134	100.245 reg/t 42.0805 reg/t	1800-09-22 1802-05-38		•	
<ul> <li>IS Forman, pre-</li> <li>IS Forman (pre-</li> <li>IS researching, and it</li> <li>IS researching</li> <li>IS resp</li></ul>		INTERNAL IN TRANSPORT	Geom promety • • • • • over1800010658-40. over1800010658-40. over1800010658-40.	134 134 134	100.245 reg/ 42.0965 reg/ 31.7025 reg/	1890-09-22 1890-01-88 1890-01-88	1.	•	
El Hardalog, pe     El Inatalog, and     El Inatalog, and     El Inatalog, and     El Inatalog, and     El Inatalog     E		10000000000000000000000000000000000000	Security & Security Contractions	134 134 134 134	100.248 (mp) 40.0000 (mp) 80.0000 (mp) 80.0000 (mp) 84.02 (mp)	1860-09-22 1860-09-38 1864-01-11 1860-02-04	-	•	
* (5) fundada, gan * (1) fundada, ganott * (1) fundada versetz *		10000000000000000000000000000000000000	Consty Const	134 134 134 134 134	100.240 mpl 40.0000 mpl 80.0000 mpl 80.0000 mpl 80.000 mpl 80.001 mpl 40.01 mpl	1895-09-22 1892-01-38 1893-02-08 1893-02-08 1990-02-05	1.	•	
El Hardalog, pe     El Inatalog, and     El Inatalog, and     El Inatalog, and     El Inatalog, and     El Inatalog     E		2000/2114 Parties 2000/2116 Parties 2000/2116 Parties 2000/2116 Parties 2000/2116 Parties 2000/2116 Parties 2000/2116 Parties	CONTRACTOR CONTRACTOR	134 134 134 134 134	100243 reg1 400060 reg4 80060 reg4 800705 reg4 8007 reg1 4007 reg1	1800-09-32 1800-07-38 1890-02-08 1990-02-08 1990-02-15 1990-07-15	-		
El totalog pr     El totalog pr     El totalog soot     El totalog soot     El totalog soot     El totalog soot     El totaloger Faultocos     El Topper Faultocos     El Toppe     El totaloger Faultocos     El Toppe     El totaloger Soltions.vesult     Solt soltable     Solt soltable     Solt soltable     Solta		Bit         Bit           100002713         Rosten           000002713         Rosten           000000713         Rosten           000000713         Rosten           000007173         Rosten           000007174         Rosten           000007173         ROsten           000007174         ROsten           000007174         ROsten           000007174         ROsten           000007174         ROsten           000007174         ROsten	Secondary         Image: March 1           COULDINGSHAD,         Image: March 1 <td>134 134 134 134 134 134</td> <td>100.246 mg/ 40.046 mg/ 40.046 mg/ 40.17 mg/ 40.17 mg/ 40.17 mg/ 40.1946 mg/ 30.3014 mg/</td> <td>180-09-22 1802-09-38 1894-09-17 1809-02-04 1809-02-04 1809-02-15 1809-09-15</td> <td>-</td> <td>•</td> <td></td>	134 134 134 134 134 134	100.246 mg/ 40.046 mg/ 40.046 mg/ 40.17 mg/ 40.17 mg/ 40.17 mg/ 40.1946 mg/ 30.3014 mg/	180-09-22 1802-09-38 1894-09-17 1809-02-04 1809-02-04 1809-02-15 1809-09-15	-	•	
* El tetalog pr * El tetalog pr * El tetalog pr * El tetalog pr * El tetalog * El		Status         Home         Home           20000         File         Home           00000000         File         Home           000000000         File         Home           000000000         File         Home           000000000         File         Home           000000000         File         Home           0000000000         File         Home	CONTRACTOR CONTRACTOR	134 134 134 134 134	100243 reg1 400060 reg4 80060 reg4 800705 reg4 8007 reg1 4007 reg1	1800-09-32 1800-07-38 1890-02-08 1990-02-08 1990-02-15 1990-07-15		Control query randomer 1 vans 480 marse: 73899 vana	fleet

In order to have the above SQL command available as a new view for the subsequent section in QGIS, we should store it accordingly.

```
create view hygrisc.nitrat_geom as (select t1."messstelle_id", t1."name",
t1.geom, t2."stoff_nr", t2."messergebnis_c", t2."masseinheit",
t2."datum_pn", t2."messergebnis_cm" from hygrisc.messstelle t1 ,
hygrisc.nitrat t2
where t1."messstelle_id" = t2."messstelle_id")
```

# 2.5 **QGIS**

QGIS is a free and open-source geographic information system (GIS) software. It provides a wide range of tools and functionalities for visualizing, analyzing, and managing geospatial data. QGIS supports various data formats and allows users to create, edit, and publish maps.

You can download QGIS for free from the below link.

#### https://qgis.org/en/site/

The below video shows how to download and install QGIS for Windows which is highly recommended to watch before installing it.



To gain a better understanding of QGIS, I recommend watching the following video, which provides valuable insights and guidance on using the software.

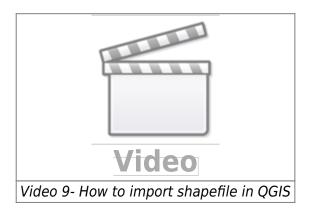


**Create a time series video:** In this section, our objective is to generate a time series video depicting the changes in nitrate concentration over time in North Rhine-Westphalia, the most populous state in Germany. To begin, we must download the shapefile for North Rhine-Westphalia and import it into QGIS for further analysis and visualization.

Three below shapefiles need to be downloaded

- entire state shapefile (dvg1bld\_nw.shp)
- kreis shapefile (dvg1krs\_nw.shp)
- Gemeinde shapefile (dvg1gem\_nw.shp)

All three shapefiles can be downloaded from here. After downloading the shapefiles, we can proceed to load them into QGIS for visualization and analysis. The below video shows how to load the shapefile in QGIS.



Now we can see the map of NRW, kreis and Gemeinde. There are two options to create a video for time series.

**Locally with shapefile:** In here, we need to have a shapefile that consists of the nitrate concentration over time. download the notebook from here and run the Python codes to create two shapefiles. then we should load these two shapefiles to the QGIS. The first shapefile consists of all stations in NRW and the second one consists of the nitrate concentration.

The below video shows how we can load shapefiles to QGIS.



**Connect to Database:** The below video shows how we can connect our QGIS to Database and load the file from Database.



## **3 Dashboard**

In this section, the process of creating an interactive dashboard for our data will be explored. An interactive dashboard is a versatile tool that allows data to be interacted with, analyzed, visualized, and key information to be monitored by users.

This section discusses two approaches to creating a dashboard. The objective of this dashboard is to provide a user-friendly and interactive interface for data exploration and visualization, accessible even to non-programmers. Such a dashboard plays a crucial role in enhancing data comprehension and is widely utilized by managers and decision-makers to facilitate informed decision-making processes.

One notable example of this type of dashboard is the Covid-19 dashboard, which has gained widespread usage worldwide, including in Germany. The Covid-19 dashboard provides users with valuable insights into the number of new cases and deaths reported over various time periods. It helps individuals track the progression of the pandemic and understand the impact it has had on different regions and countries.

In our specific case, the objective is to develop a straightforward dashboard that showcases the map of North Rhine-Westphalia (NRW) alongside the concentration rates of Nitrate and Sulfate at different time intervals. This dashboard will provide a visual representation of the spatial distribution of these pollutants and enable users to observe any temporal variations in their concentrations within NRW.

#### **Plotly Dash:**

**Plotly:** Plotly is a computing company located in Montreal, Canada. They develop online data analytics and visualization tools. Plotly offers online graphing, analytics, and statistics tools for their users, as well as scientific graphing libraries for Python, R, MATLAB, Perl, Julia, Arduino, and REST. Plotly offers several open-source and enterprise products such as Dash which have been used for creating simple and interactive dashboards in this project.

**Dash:** Dash is a framework to build data apps rapidly not only in Python but also in R, Julia, and F#. According to Plotly's official website, Dash is downloaded 800,000 times per month which shows that nowadays Dash getting more popular. Dash is a great framework for anyone who uses data with a customised user interface. Through a couple of simple patterns, Dash eliminated all of the technologies as well as protocols that are needed to make a full-stack web app with interactive data considerations. Another good feature is that Dash is running on web browsers so it means that no other application needs to run it.

To learn more about creating a dashboard with Plotly Dash, you can follow the link provided below. This resource contains comprehensive tutorials that guide you through the process of building a simple dashboard using Plotly Dash. These tutorials will provide you with step-by-step instructions and examples to help you create interactive and visually appealing dashboards using Plotly Dash.

### https://www.youtube.com/c/CharmingData

Dash is also offering some dashboards examples which could be really nice and helpful to get ideas. | Click here for Dash gallery

All the source codes of the dash gallery are available in | here

**Dashboard Design:** I have developed a web application dashboard that effectively visualizes time series data for Nitrate and Sulfate. The image below shows the main page of our dashboard.



In order to run the above dashboard on your local system, you need to do the following:

### Click here and pull the repository

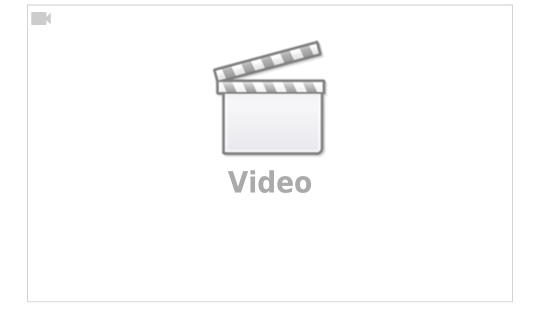
In this repository all the Python codes, SQL queries are available

There is a yml file inside the folder which name is "dashboard\_environment". This yml file will be create an environment with all the necessary packages which you will needed to run the application. First step needs to load this file to the Anaconda, if you are not sure how to do that please refer to the XX

### 4. Result

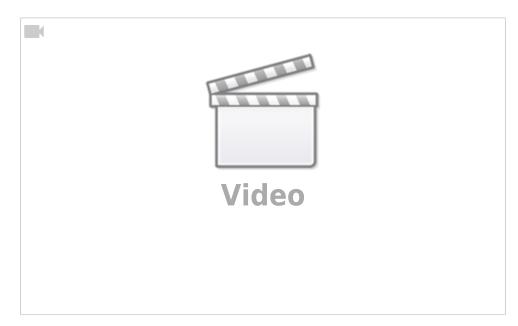
### Nitrate concentration 2000-2010

The video below shows the concentration of nitrate in North Rhine-Westphalia (NRW) from 2000 to 2010. This visualization was created using QGIS 3.16. By watching the video, you can observe the temporal changes in nitrate levels across NRW during the specified time period.



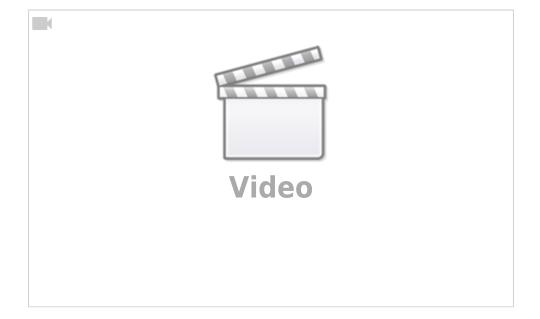
### Nitrate concentration 2010-2020

The video below shows the concentration of nitrate in NRW from 2010 to 2020. The video was created with QGIS 3.16



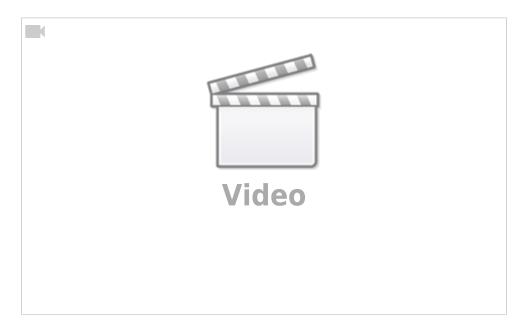
### Sulfat concentration 2000-2010

The video below shows the concentration of sulfate in NRW from 2000 to 2010. The video was created with QGIS 3.16



### Sulfat concentration 2010-2020

The video below shows the concentration of sulfate in NRW from 2010 to 2020. The video is created with QGIS 3.16



### 5. Project codes

All the codes are available in the below link.

Click here for project codes

### **Weitere Infos**

• EOLab-Wiki-Seiten zum Thema Grundwasserdaten in NRW

From: https://wiki.eolab.de/ - HSRW EOLab Wiki

Permanent link: https://wiki.eolab.de/doku.php?id=eolab:openhype:start&rev=1688022536



Last update: 2023/06/29 09:08