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,

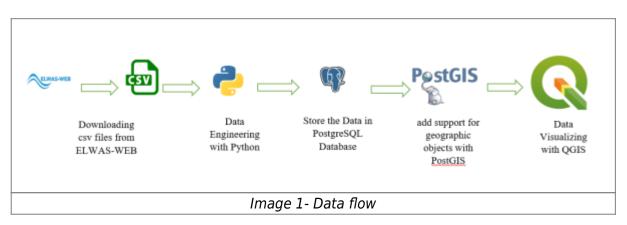
program code,

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
- $\circ~$ Introduction to the geodatabase extension <code>PostGIS</code>
- $\circ\,$ Introduction to the processing of geodata with the programming language Python
- $\circ~$ Installation of the OpenHyPE database management system
- $\circ\,$ Discussion of the data model and upload of the OpenHygrisC data of the LANUV
- Automatic creation of diagrams on time series of water quality
- 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.

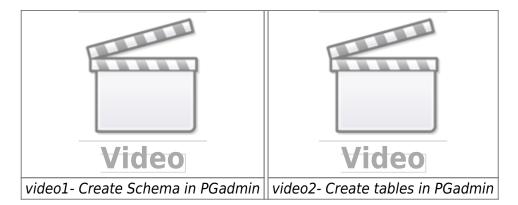
PgAdmin is an open-source administration and development platform for PostgreSQL. It is a graphical user interface (GUI) tool that provides a convenient way to manage and interact with PostgreSQL databases.

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 iwwybs 💽 Arangaaasseet 4	🔅 Debbil desentregen 🚺 Überbick für Stude	BM92 peine 🖬 Betmanische Am	n
		Information und Nordrheim West	Tachsia alise
OpenGeocata.NRW 3 Dryges		(J.4	SCH 45 XML
Grundwassermessstellen NRW			
Linestinformation: Date: Date: states unter ter Deteritance Des Detersatizations/busic_B_Doperty.pht_mail.pp Meladatan thr CP 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_gavmentelen-messwerte_EPS628022_CEV.zp (d) OpenhypteC_geveneentend_EPS628022_CEV.zp Daten der Grundwassermessstellen in NRW als Sqilte Date	nbank	29-10-2021 08-46-00 29-10-2021 08-49-08	41.7 MB 100.4 MB
Detai Detai DetainypteC_gaverassetiles-messwerks_EPS628022_CEV.ep (d) OpenHypteC_gaveseentand_EPS628032_CEV.ep Deten der Grundwassermessstellen in NRW als Sqilte 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_gav-messatelen-messwerks_EPS628022_CBV.ap CoverHypteG_gav-researchard_EPS628032_CBV.ap Detan der Grundwassermessstellen in NRW als Sollte Detai Detai Detai CoperHypteG_gav-messatelen-messwerke_EPS025632_Solte.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.

Filo		dit. V	ow Ir	ment Cell	Kernel V	Vidgets	Help						Tountest	Python 3 O		
8	+	9< 19	6 +	↓ ► Run	. c »	Mariedo	neti 🔊	e) (m)						- Jahren er		
		-	Read	the file		~	Text									
			on't forge	et to add the "de	limiter", norm	al csv fik	e's delimit	ter is *,* but in c	ur case it's *;*	, so we need	to assign it.					
	Tn	tselt d	f = nd.	read csulma s	tation ofna	me, del	imitera	't", headers	infer')							
	In		f = pd. f.head(read_csv(gw_s)	tation_pfna	une, del	limiter≏	';', header=	infer')			Python Code				
	In Out	d	f.head(tation_pfna	•32		';', header=		gemeinde_id			, beobachtung_w	vasserstand	-	
		[30]-	f.head()	name	•32				gemeinde_id 05111000				vasserstand		
		[3 <u>91</u> -	f.head(sl_nr) messstelle_id	name	• 32 343064	n 32 5678019	gw_stockwerk			gwhorizont_id NaN	gwhorizont		vasserstand -	₽	
		[39]-	f.head(sl_m 0 67530) messstelle_id 32505929	name UWB-Ddorf 01285	•32 343064 292077	n32 5578019 5545349	gw_stockwerk 1.0	grundstueck	05111000	gwhorizont_id NaN	gwhorizont NaN		vasserstand -	₽	

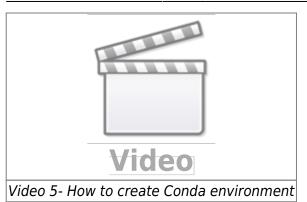
2.3.3 Anaconda

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

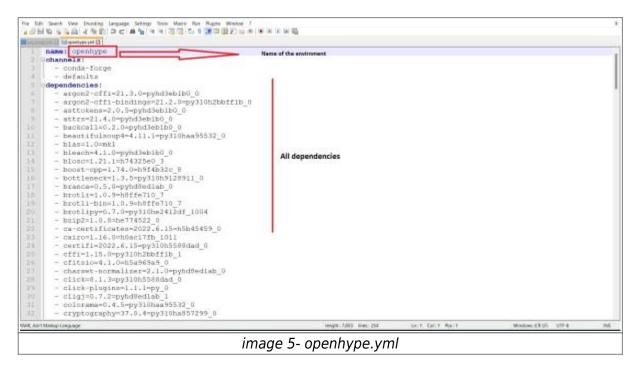
Certain packages require the specification of a channel for installation. This is why the above code includes the channel specification to ensure the correct installation.

• **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	isplare		0 +• \$ *
E sina7272 / LANUV_Groundwater	(Peee)			St We Schwands (+ 🕅 🖅 🚽 🗮 Staved () +
⊖ Code _ ○ Issoes _]. Poli requests	🛇 Actions 🛛 Projects 🗔 Wild	© Security ⊨ Insights © Setting		
	Primin - Prisent Song		Go to file Add file - Code +	About 0
	atna7272 update		atticit 2 hours ago (5) 22 commits	Ground water in NRW
	Read Contraction and the			
	Apyrib_checkpoints			
	Notebook			
	tata 🖿			
	1 (9) ¹			Refeases
	D granibute			
	D READMEntd			
	READMEINS	water		Packages Ni janiapa pukkhuć Fablutyov fen pukage
	Ground water in NRW			Languages
	0 e.m. m.			fragetor Postsheem 93.5% or (200), 5/25
		image	e 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.

ser 🖬 🏗 Th	🖬 🔍 Dashb	board Proper	rties SQL Sta	atistics Dep	endencies	Dependen	ta 🖪	etwais/etwa	s@elwas	ť.,									
> 🚱 Domains	8		E Q -	8 - 6	8 2'*	Τ.	Non	nit. 🗵		- 4 0		5.1	s dr	*	18.4				
 ETS Configurations 	67 0	elwas/elwas@e	with w									-		(internet					
D) FTS Dictionaries	BL. OB	Editor Query							_			_			_	Scratch F	-		
An FTS Parsers			1.02.53													acratch r	-940		
FTS Templates	1	select + fr	on sina.messa	wert limit	100;														
Foreign Tables																			
 b) Functions Materialized Views 																			
> () Procedures																			
> 13 Sequences																			
 Tables (6) 																			
> m gw_stations																			
> El katalog ge	- 1																		
> 🖽 katalog_stoff																			
measurements																			
> mesisitelle																			
> 📰 mennivert																			
X 102 Trigger Functions	Date (Detruit Evaluation	in Massauri	Batificatio															
X Vi Trigger Functions Types	Data C		in Messages																
K Trigger Functions Types Types Weens (7)	_	ster ,	messstelle_id	Notifications		/ stoff.)	• /	probesgut text	/ mess	ergebnia_z	/	messergeb	nia, hirweit	,	bestimmur double pre	ngsignerate ctalion	/ massesthat	/ terroverfal	hee
big Trigger Functions C Types Wrens (7) J Groupby_station_result	_		messstelle_id bigint	/ pre_id /	datum, pri date	/ stoff.) Light		probengut teat Grundwasser	 mess- doubl 	e precisioni			nia, hirweis	,	bestimmur double pre	clain	rent	terroverfal bot Membrant	
by Trigger Functions c Trigger Venet wass(7) for outply_station_result in retrat	1	sLer [PK]bigint 12289051	messatelle_id bigint 2090027	/ prejid / text 710 3/1992/16.	datam, pn date 1992-05-21	/ stoff.) Digitit	1244	ann Brundwasser	/ mess (5-0)	104	248	(12.87)	nia, hirweis	,	bestimmur double pre	clain	figer (her)	Membrant	the
big Trigger Functions Constructions Construction Types Difference Types Types		slov [PK]bigint 12209053 12209053	messatelle_id tagint 20990023 3099003	Pha.id Host 710 3/1992/16. 916 3/1994/16.	datum, pn date 1992-05-21 1994-03-29	≠ stoft, bişint	1244 1244	Grundwasser Grundwasser	1000	106 d1	248 .978	(n.8) (n.8)	nia, hirweis		bestimmur double pre	ctation	han hgm (hun) hgm (hun)	Membranh Membranh	ihe ibe
 X Trigger Functions Types Wreas (7) groupby_station_result retrait initiat 	1 2 3	sLor [PK]bigint 12209051 12209053 12209072	messstelle, id bigin 2999023 3090028 0 809002	Prejid text 710 3/1992/16. 816 3/1994/16. 207 3/1093/16.	datan, pr date 1992-05-21 1994-03-29 1995-01-11	/ stoff,/ bight	1244 1244 1244	sex Grundwasser Grundwasser Grundwasser	1000	106 61 31	248 .978 9989	(m/l) (m/l) (m/l)	nia, hirovatia	,	bestimmur double pre	ctation	inuti mg/l inuti mg/l inuti mg/l	Membranh Membranh Membranh	ihe libe
 Ingger Functions Types Weres (7) groupbg_station_result intrast intrast, peon iii Columns 	4 1 2 3 4	sL.vr [PHQ:bigint 12289051 12289052 12289072 12289077	rreesstelle_id bigint 2990020 20900000 20900000 803000 7 803000	Pre_id hext 710 3/1992/16. 3/1993/16. 3/1993/16. 303 3/1992/16.	datum,pn date 1992-05-21 1994-03-29 1995-01-11 1992-11-02	oşn	1244 1244 1244 1244	Grundwasser Grundwasser Grundwasser Grundwasser		106 d1 31 2.25	248 .978 1989 5777	(m)() (m)() (m)() (m)()	nia, birweis		bestimmur double pre	ctaton	ten (inn) ten (inn) ten (inn) ten (inn)	Membranh Membranh Membranh	ihe ibe ibe
No Trager Functions Trans Trans	4 1 2 3 4 5	sl_er [PR]bigint 12289051 12289053 12289075 12289077 12289077	messatella.id tegint 2990020 803000 7 803000 9 803000	Pre_id htt 3/1992/16. 3/1992/16. 3/1992/16. 303 3/1992/16. 303 3/1992/16.	daturs.pn date 1992-05-21 1994-03-29 1993-01-11 1992-11-02 1993-12-22	oşn	1244 1244 1244 1244 1244	soor Grundwasser Grundwasser Grundwasser Grundwasser		106 61 31 225 3.00	248 .978 9989 5777 5403	(null) (null) (null) (null) (null)	ris, Mrweis		bestimmur double pre	ctation	inger (jun) junij mgel junij mgel junij mgel junij mgel	Membranh Membranh Nembranh Membranh	ihe ibe ihe
 > 02 Trager Functions > 27 Trate > 9 providey_station_result > 10 rotat > 11 rotat > 12 columns > 10 columns > 2 thiggers 	1 7 9 4 5 6	sLer [PhQ]bigint 12289051 12289053 12289075 12289077 12289077 12289077 12289077	messatella.id tagint 2990020 803000 7 803000 9 803000 9 803000 9 803000	 pha.id ixrt 2/1992/16. 3/1992/16. 3/1993/16. 3/1993/16. 3/1993/16. 3/1992/16. 	datum, pri date 1992-05-21 1994-03-29 1993-01-11 1992-11-02 1993-12-22 1993-04-21	oşn	1244 1244 1244 1244 1244 1244	enn Brundwasser Brundwasser Brundwasser Brundwasser Brundwasser		106 61 31 2.27 3.07 11.0	248 .978 .989 5777 5403 .9675	(naf) (naf) (naf) (naf) (naf) (naf)	nia, Mrwein		bestimmur double pre	claton	inen ingn (jun) ingn (jun) ingn (jun) ingn (jun) ingn (jun)	Menibranfi Menibranfi Menibranfi Menibranfi Menibranfi	ine ibe ibe
 Ingger Functions Types Wrems (7) groupby_station_result intent_peon intent intent_peon intent intent intent intent intent intent 	1 2 3 4 5 6 7	BLor [P4]bigitt 12289051 12289051 12289055 12289055 12289055 12289055 12289055	messatella.id isignn 2990027 2990027 803000 803000 803000 803000 803000 803000 803000 803000 803000 803000 803000 803000 803000 803000	pres.1d hext 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10.	datum_pn date 1992-05-21 1994-05-29 1993-01-11 1992-11-02 1993-12-22 1993-04-21 1993-07-19	oşn	1244 1244 1244 1244 1244 1244 1244	Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser		106 61 31 2.27 3.07 11.0 106	248 978 989 5777 5403 248	(rad) (rad) (rad) (rad) (rad) (rad) (rad)	nia, Mrwein	,	bestimmur double pre	claton	huli mgi huli mgi huli mgi huli mgi huli mgi huli mgi huli mgi	Membranfi Membranfi Membranfi Membranfi Membranfi Membranfi	iter ibe ibe iter ibe
 Ingger Functions Types Weens (7) groupby_station_result instat instat_pecon \$ Columns \$ Rules \$ 1 higgers sulfat_peren sulfat_peren temp2 	1 7 9 4 5 6	ULAR [PV[bigin] 12289051 12289053 12289053 12289075 12289075 12289075 12289075 12289075 12289075	messatelia_k4 isgint 2990027 2090027 2090027 803000 803000 803000 803000 803000 803000 803000 9 803000 9 803000 9 803000 9 803000	pres.1d hext 3/1992/10. 3/1992/10. 3/1994/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10.	datam_pn date 1992-05-21 1994-03-29 1993-01-11 1992-11-02 1993-12-22 1993-03-10 1992-03-10 1992-05-10	- Opti	1244 1344 1344 1344 1344 1344 1344	Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser		106 61 9.1 9.0 9.0 11.0 106 53	248 .978 9089 5777 5603 248 124	(rall) (rall) (rall) (rall) (rall) (rall) (rall) (rall)	nia, Mirweis		bestimmur double pre	claton	inen ingn (jun) ingn (jun) ingn (jun) ingn (jun) ingn (jun)	Menibranfi Menibranfi Menibranfi Menibranfi Menibranfi	ihe ibe ihe ihe ibe
 Ingger Functions Types Wrems (7) groupby_station_result intent_peon intent intent_peon intent intent intent intent intent intent 	1 2 3 4 5 6 7	BLor [P4]bigitt 12289051 12289051 12289055 12289055 12289055 12289055 12289055	messatelia_k4 isgint 2990027 2090027 2090027 803000 803000 803000 803000 803000 803000 803000 9 803000 9 803000 9 803000 9 803000	pres.1d hext 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10. 3/1992/10.	datam_pn date 1992-05-21 1994-03-29 1993-01-11 1992-11-02 1993-12-22 1993-03-10 1992-03-10 1992-05-10	- Opti	1244 1344 1344 1344 1344 1344 1344	Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser Grundwasser		106 61 9.1 9.0 9.0 11.0 106 53	248 978 989 5777 5403 248	(rall) (rall) (rall) (rall) (rall) (rall) (rall) (rall)	16		contre the	clainin	tign (jun) tign (jun) tign (jun) tign (jun) tign (jun) tign (jun) tign (jun)	Membranfi Membranfi Membranfi Membranfi Membranfi Membranfi	ite ibe ite ite

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%';

Now B 0	🛯 🖕 🔍 Dashboard Properties DQL Ballatics Dependencies Dependents			
- (0 site	8 9 8 · E Q · # · 6 8 8 · 7 ·			
+ []+Colutions	Ø elven/elven@elven +			
 He Domano 	Query Editor Query History		Scretch Peel	
 IQLETE Configurations 				
+ ID FTS Dictionaries	1 select + from sine Astalog, stoff where name 14kg 'Nith			
 Aur PTS/Paralisers 				
+ TS Templates				
 IIII Foreign Tables 				
 Withfunctions 				
< 🦉 Materialized Vevia				
 II):Procedures 				
+ L/Sequences				
- I Tables (0)				
 Importations Importation or 				
> Illianing duff				
> El Nespuramenta				
· mirrennrielle				
+ III messivet				
 Bit Trigger Punchares 				
 Types 				
- 🔤 views (7)	Dete Output Explain Massages Notifications			
Provide the state of the sta	statur a neret a beachesturg a cesul			
+ nitrat	staff_ar a herne a bescheitung a besch			
- Writer from	1 1143 Non Pull 7440-05-1			
+ D Columna	2 1188 1604 (vul) 7445-654			
+ The Paler	1 1244 titlet pull bull	1		
+ De mighen	4 York Inter-Stational Ind. July	- ·		
+ metat_peom	5 T24 Meet Pull Pull			
+ term	6 1047 ternösckent pull pull			
+ term2	7 1967 Northanskie Tenskie (nali (nali			
F (e) tiger	8 2290 Ministered (1990)			
+ 🐵 toper_dette	 2163 Matches 2163 Matches 2163 Matches 			
> (e) topology	10 2507 Millionessipher (MA) [147-19-			
 mpostpes 	H 1768 Hosedana put 111914			
 Terndiala, postgis 				
s 📰 tseerap				
 In Login/Terrup Roles 	13 4184 Notement (no.0) 50457			
Interpretation in the second secon	14 4168 Mangaram (nutl. 180824-4.			

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 ;

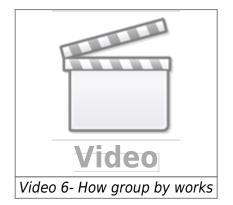
Group by the two tables: the Group by clause is used to group rows based on one or more columns in a table. When working with two tables, you can perform a GROUP BY operation to group the data based on common values from both tables. In this section, we want to group by our two tables (messwert and messstelle) 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;
```

 an broket 		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						
 III Foreign Tables 						
+ Wehnstone						
+ 🛄 bitaterizitzent Veivez						
+ C)Procedures						
 N.R Secondors 						
- III Tables (0)						
 III gw_stations 	1.1					
s minatelog.ge						
> Ittikatalog_stoff						
 Esmeasurements 						
ittrazzatelie						
 messuet 						
+ Big Trigger Functions + Int Types						
- Views (7)						
encoded and the station result	Dyla Outpi	ut Explain M	ranges tectficat	- 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			
 ib tiper 		10391039	- 10			
 Ide tiger_chata 		18221117	0			
 B topology 	18	10201202	-			
> Tr postpres	11	(8281294	28			
 Sternete.postgo Sternep 	12	10291306	28			
Ga Login/Ursup Roles	- 13	18291912	42		Lange and the second second	
		10601016			 Buccessifully run. Total guery runt 	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); 2024/05/18 23:26

13/20

Admin File* Object* Tools* Help*		
	🖥 🙀 Q. Dashboard Properties BQL Instatus Dependencies Dependence 🗰 environment@environ	
· · · · · · · · · · · · · · · · · · ·	8 2 2 - 2 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2	
> \$3 Collectors	(viewastikhous)olvez =	
 Re Domaina 	Guary Editor Quary History Scratch Pad	_
 (g) FT3 Configurations 	E select + from sing-infrast where detailor + (select max(datum.pn) from sing-infrast)	
 PTS Stationaries 	 start - the startstrate and networkst - startstratecht the startstratecht 	
Au FTS Partners BFTS Templates		
Bit Poreign Telden		
+ B-Functions		
 Materialized Views 		
 CProtochom 		
 + Elegances 		
~ 🛅 flableo (ki)		
+ CIII (av., whattores		
+ militatarg.staft		
 Interacting second Interacting second 		
> En mesostelle		
+ Itt mexavert		
> N0 Trigger Functionse		
> ID Tutes		
 Were (7) groupby_station_result 	Data Output Explain Messages Indiffications	
+ Softer Control And	La resultable i pradi den a den a den e polariti entre e polariti e secondariti e secondaritati e secondariti e secondariti e se	16,24
+ ntal geon		·
 Tolume 	1 T7721212 SUBT77 SQULVIT SCHW177 T344 Guernesse at [uu] Jul mul Decempted soc-Augue errors. S 2015	109.0
 Interview 		
+ \$+ Triggers		
+ mattat		
 astid, pion. statut 		
+ Etavla2		
+ -(E) Eiger		
 Shipecata 		
 Brispology 		
> 😤 postgree		
 mitemplate_postplit mitemplate_postplit 		
/ (b.Login/Group Roles		
> In Tatletracet		
	Image 10 Maximum of the date in the nitrat table	
	Image 10- Maximum of the date in the nitrat 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);

A2W 8 1	🗄 🐚 Q. Dashboard: Properties: 9(3). Statistics: Dependencies: Dependen	
 so patiet 	8 8 8 4 8 9 8 4 8 8 8 T - 100 - 1 8 9 4 8 4 5 8 4 4 8 4	
 Bit Contations 	S down/dwnsg/dwna -	
+ & Sortains	E-27 Introduction and the Dary Hallow Deep Hallow	South Pal
 B-PTS Coritiquations 		308001981
In FTS Dictionaries	3 select + from sine, nitrat where detum_pn = (select win(detum_pn) from sine.without);	
6 MA FTS Paraelo		
 FTS Templetes 		
> UP Foreign Tables		
 bis Functions 		
 Materialized Views 		
> 1)Procedures > 1.1Sequences		
= CTabres (5)		
> migw_stations		
 Etvatalio.or. 		
Profe position of the		
+ 🛄 messurementis		
 IIII measafete 		
+ IIII measuret		
> Ith Trigger Functione		
 (c) Types (c) Views (7) 		
+ Broutin_station_result	Data Output Eiplain Messages Notifications	
+ entit		envertatives a vertatives a rot_ort a hexault a alcoal_dat
- Citrat geom	National and the state that any second state and better and second states and second	
* Bröstames	1 122734201 10212144 (1193110) 10411044 10100 1044 10400 1041 10400 1041 10400 1041 10400 1041 10400 1041 1040	sampsat juli intr_Bek 2011-te28
> =====		
+ 3+Thopers		
+ withat		
+ mattat_givinn		
+ ternp		
+ interest		
+ do tiger data		
+ for topology		
 Im postgiwa 		
> mit temptate_postgin		
> stitthemap		
 -lb.Logn/Group Roles 		
8 Tabiwipacet		

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 De	shboar	rd Prop	inties S	QL Statistics	Depens	encies Dependents	E elwa	s/elwas@ielwas*							
3	() public	8		8 -	· ·	2 - 10 -	6 8	2- T -	Notest	*	4 6		4- 1 B	4			
~	r 🛞 sina		1000	n/otwasg	w and			A REAL PROPERTY OF A REAL PROPERTY.						-			
	Ali Collations	BLO	-	tor Due		S7									icratch Pad		
	 Image: Second sec													1	ichation Paig		
	(1): FTS Dictornation (1): FTS Paraties (1): FTS Paraties (1): FTS Paraties (1): Frontion (1): Frontion (1): Frontion (1): Frontedures (1):	1	sec		511	a.messstelle											
	> 🔤 katalog_pe																
	Hatalog_pe Hatalog_stoff measurements	De	ia Outp	ut Expl	ain Me	ssages Notific	cations										
	> 11 katalog_stoff > 10 measurements > 11 mesastelle	Do	. f	iut Expl Norfaenge, double prec	em a	ssages Notific aunofrohrlaung double precision	e,cm	eutoudurchmesser_m double precision	•	historischer, ruhe, wsp double precision		embaulaengecm double precision	oberkante, filte double precisik	,em (unterkente, filter, om double precision		geom 🔒
	 El katalog_stoff III measurements 	-	•	literisenge,	em a	sumpfrohrlaung double precision	e,cm	double precision	• •	double precision	A (hull	action buccooks	double precision	n fi	CONTRACT DECISION		geometry
	itikatalog_stoff itikatalog_stoff itikatalog_stoff itikatalog_stoff itikatalog_stoff itikatalog_stoff itikatalog_stoff itikatalog_stoff	ster	A dia.	literisenge,	am a	sumpfrohrlang double precision	e,cm	double precision	-	double precision		and one becomes	- Online building	er.	l)	but	0101000020586
	Theoremain of the second	sber N-m	A do	literisenge,	sion (bul	sumpfrohrleing double precision	e.cm 11	double precision	put	double precision	(n.f)	and the free second	(hell)	ju.	9	(huit) 9349	0101000020EB6 0101000020EB6
	the late of the second se	sber Pim t Pat	A do	literisenge,	ern e sion (kul) 1000	auripfichrieing double precision	e.cm ju ju	double precision	(nul) 115	double precision	(nat) (nat)	and as factoria	(1968) 2000)nu 1034	9 1]	(huit) 9349 (huit)	0101000020EB6 0101000020EB6 0101000020EB6
	tetraing_stoff til measurements til measurements til measurements til measurement	sber Pim t Pat	A do	literisenge,	am an sion (bul) 1000 (bul)	sumpfishrisong double precision	e.cm jn jn jn	double precision	pul 115 pul	double precision	(16.0) (16.0) [16.0]	and the second	2000 [mil]	ןיען 1034 זענן	0.500 (165550) 9 1] 5	(huit) 9349 (huit) 7505	0101000020EB6 0101000020EB6 0101000020EB6 0101000020EB6
	the late of the second se	siber R4m 1 Del	A f	literisenge,	am (nul) ston (nul) (nul) (nul)	sumpfishrieung double precision	ilian In In In In	double precision R R R R R	(rul) 119 (rul) (rul)	double precision	ituli ituli ituli	source previous	(mil) 2000 (mil) 635) 1034 (14) 750	00000000000 9 11 15	(hul) 9349 (hul) 7505 (hul)	0101000020EB6 0101000020EB6 0101000020EB6 0101000020EB6
	tabling_stoff tabling_stoff timessurements timessatele timessatele timessatele timeswent tight frigger Functions times(7)	ather 194 m 1 Def Suet	A f	literisenge,	erni andi stori juuli 1000 juuli juuli juuli	sumptishtiong double precision	i, em jou jou jou jou jou	double precision 4 4 4 4 4 4 4 4 5	tul 115 tul tul tul	double precision	540 540 [44] [44] [44]	andre benoon	(mil) (mil) (mil) (35 (mil)	1903 1934 1944 1944 1944	00000 (100000) 10 10 10 10 10 10 10 10 10 10	(hul) 9349 (hul) 7505 (hul) 3452	
	the late of the second se	ather 194 m 1 Def Suet	a f	literisenge,	ston bull 1000 bull bull bull bull	sampfischriteing double precision 1 2 1 1 1	i in In In In In In	double precision q q q q q q q q q q q q q	(rul) 115 (rul) (rul) (rul) (rul)	double precision	turi turi turi turi turi turi turi	andre brookin	1940 1940 1940 1940 1940 1940	500 1034 104 104 104 104 345	000000 (100000) 99 10 10 10 20 00	(hult) 9349 (hult) 7505 (hult) 3452 3452 3190	0101000020E86 0101000020E86 0101000020E86 0101000020E86 0101000020E86 0101000020E86

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";
```

	Bashooard Properties SQL Statistics Dependencies Dependents 🖀 disculoreogradieus.*	
 Option 	8 8 8 - 2 0 - 8 - 5 2 2 T - mm - 8 9 - 6	0 · · · · · · · · · · · · · · · · · · ·
* @ sma * El Collations	🔗 efenn (elemanisten) -	
 Sp. Domains 	Gwery Editor Query History	Scottch Pad
F D PTS Configurations] select ti."meanshelle td", ti."new", ti.goon, t2."stoff or", t2."meanergebria.c	ef. 42. Summal chatter
5 By F75 Dictoriation	2 U. "detumpe", 12. "essergebrits.ce" free site-mesostelle 11 , site-mitrat 12	s p tas minimum p
5 #aF75Painers	b where the "menantable fd" = the "menantable fd";	
🕴 🧮 FTS Templates		
 Etherwige Tables 		
 Ein Planchieren 		
Materialized Wewes.		
 Processare Coldecusercas 		
+ CTables IS		
> III de stations		
* (Thiatalog ge		
 Ithatalog.codt 		
+ ITTreesawheets	Data Dutjua Explain Mescages Notifications	
* Ell receptoile	annestatik. If a new a geom a too too too a new and a set	ters pro a resessigneria.cm
	1 200002717 BoxPec 01019002205540. 1344 100.245 rep/ 180	10/09/27 +
	E 80802019 HAHAL 01018000398940, 1344 40,0905 Hajd 189	10-17-38 a
* W Tegger Functions * C Types		
	8 8030201 HAVE, INTERCEPTION. 1244 RE201 April 189	nach II -
 Typen Verez (7) Typingraupby, station_result 		Nacio () =
 C) Types Veren (7) C) Unren (7) 	a accorde attole, citerancipalment, taka akalf regi regi	
 Types Weens (1) Typesolvy_station_insult Styniadity_station_insult Styniadit_spean 	8 8050179 17026. 0101800158888. 1348 84.07 rept 140 5 9050178 17026. 0101800008440. 1349 44.07 rept 140	40-00-04 -
 Types Wrens (7) Strandy Judien, reall Strandy Judien, reall Strandy Just Strandy Jean Strandy Jean Colores 	II 000017W ITEML OFFINIOEDEMAIL Table ALL?/ regil Here 3 000017W ITEML OFFINIOEDEMAIL Table ALL?/ regil Here 6 800017W ITEML OFFINIOEDEMAIL Table ALL.94 Here 5 800017W ITEML OFFINIOEDEMAIL Table ALL.94 Here	10-02-03 = 10-07-15 =
 Types Wrene (7) Stransfort, shifting, result Stratut, journ Stratut, journ Colomes Others 	# 000017W \$T004. OrtHon0008H44. T244 #4.27 mpl He 5 900017W \$T004. OrtHon0008H44. 1244 44.27 mpl He 6 900017W \$T004. OrtHon0008H44. 1244 44.04m mpl He 7 290008181 He/msc. 0018100209844. 1244 He/mml He/ml 109	00000
 Types Wren (1) Types (provide control of the control of t	II 000017W ITELE 0101000000000000000000000000000000000	000204 =

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.

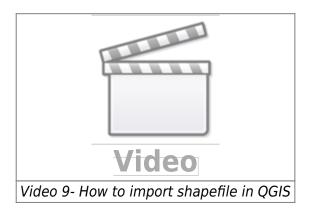


Create a time series video: In this section, our objective is to generate a time series video depicting the changes in nitrate concentration over time in North Rhine-Westphalia, the most populous state in Germany. To begin, we must download the shapefile for North Rhine-Westphalia and import it into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

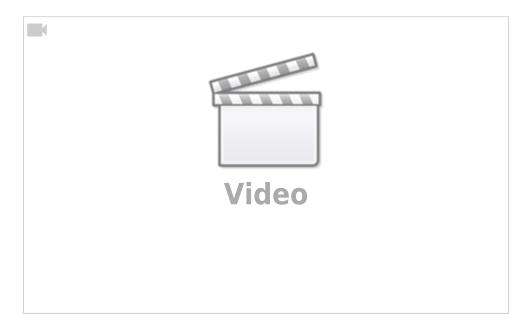
Dashboard Design: I have developed a web application dashboard that effectively visualizes time series data for Nitrate and Sulfate. The image below shows the main page of the our dashboard.



4. Result

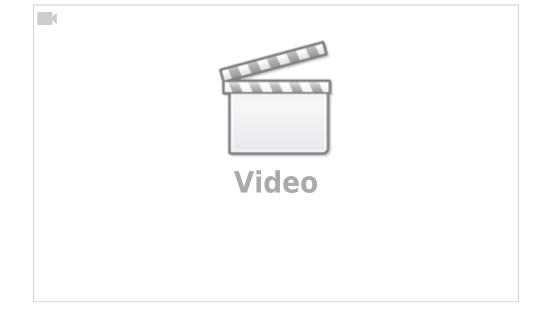
Nitrate concentration 2000-2010

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.



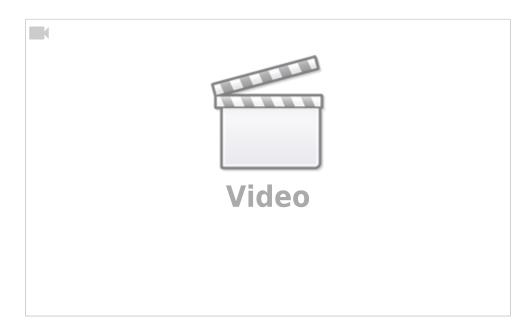
Nitrate concentration 2010-2020

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



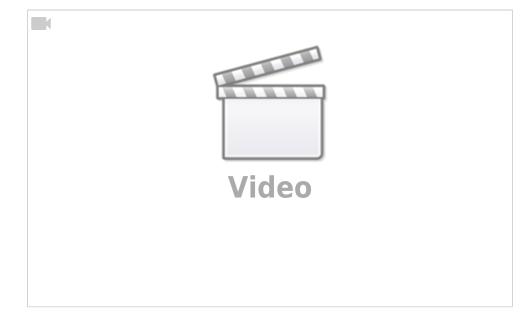
Sulfat concentration 2000-2010

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=1688022030

