

*(i)* Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)

X

# Python and SQL for Extracting, Transforming, and Loading (ETL), and Visualization with Power BI for Non-Profit Organizations



 Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)



## Project Context:

Hi Yedou,

I remember you mentioned that you had some programming knowledge with Python and SQL. I have a small project in preparation for our Nonprofit Organization and would appreciate your help. Could you use your Python and SQL skills to create a database for managing and analyzing our activities?

Best, Sydney

## Conversation:

Yedou (Developer): "Hello Sydney! Have you collected or saved your activity data into a CSV file or database that I can use to extract the data needed for your project?"

Sydney (Client): "No, we don't have data stored this way."

Yedou (Developer): "Okay, no problem. I can pull the data I need from the API to complete the project for you. What are the objectives of the project?"

Sydney (Client): "Currently, our budgets are limited, so we want to identify which countries should be our priorities for humanitarian aid and determine the priority sectors within these countries."

## Data-Driven Decision-Making:

Before initiating a humanitarian response, an organization needs to gather comprehensive information about the country or continent to ensure an effective and appropriate intervention. Key aspects to consider include:

### 1. Geopolitical Context

- **Government and Political Stability:** Understand the current political situation, government structure, stability, and any ongoing conflicts or tensions.
- **Legal Framework:** Familiarize with local laws and regulations that might affect the operation, including humanitarian laws, import/export restrictions, and NGO regulations.

### 2. Demographics and Population

- **Population Size and Distribution:** Data on population density, urban vs. rural distribution, age distribution, and key demographic groups.
- **Vulnerable Populations:** Identify groups particularly vulnerable to crises, such as children, elderly, disabled, and marginalized communities.

### 3. Economic Conditions

- **Economic Indicators:** GDP, unemployment rates, inflation, poverty levels, and overall economic stability.
- **Infrastructure:** Condition of transportation, communication, and utilities infrastructure.

### 4. Health and Nutrition

- **Health Infrastructure:** Availability and quality of healthcare facilities, medical supplies, and personnel.
- **Disease Burden:** Prevalence of communicable and non-communicable diseases, vaccination coverage, and existing public health threats.
- **Nutritional Status:** Rates of malnutrition, food security status, and availability of essential food items.

Collecting this information enables the organization to create a contextually appropriate, efficient, and effective humanitarian response that aligns with the needs and conditions of the affected country or continent.

To achieve this, we gather data from the UNICEF API at the following base URL: <https://rdmapi.unicef.org/api/doc/index.html>. Additionally, we utilize resources from websites such as <https://www.learndatasci.com/tutorials/geospatial-data-python-geopandas-shapely/> and <https://datascience.quantecon.org/tools/maps.html>.

ⓘ Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)

```
import ast
def get_data(endpoint):
    """
    Function to make a GET request to the specified endpoint and return the response as pandas DataFrame.
    """
    try:
        response = r.get(base_url + endpoint)
        response.raise_for_status() # Raise an exception for HTTP errors
        return pd.DataFrame(response.json())
    except requests.RequestException as e:
        print("Error:", e)
        return None
# Base URL for the UNICEF API
# https://rdmapi.unicef.org/api/doc/index.html
base_url = "https://rdmapi.unicef.org"
# Endpoint: /api/countries
df_countries = get_data("/api/countries")
if df_countries is not None:
    #print("\nDataFrame from /api/countries:")
    df_countries
df_languages = get_data("/api/languages")
if df_languages is not None:
    #print("\nDataFrame from /api/countries:")
    df_languages
# Endpoint: /api/countries/current
df_current = get_data("/api/countries/current")
if df_current is not None:
    #print("\nDataFrame from /api/countries/current:")
    df_current
# Endpoint: /api/countries/organizations
df_organizations = get_data("/api/countries/organizations")
if df_organizations is not None:
    #print("\nDataFrame from /api/countries/organizations:")
    df_organizations
```

ⓘ Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)

```
import sqlite3
try:
    conn = sqlite3.connect('unicef.db')
    print("Opened database successfully")
except sqlite3.Error as e:
    print("Error connecting to SQLite database:", e)

DB = "/content/unicef.db"
def create_and_load_table(df, table_name, db_name):
    try:
        # Connect to SQLite database
        conn = sqlite3.connect(db_name)
        print("Opened database successfully")

        # Convert lists to strings
        df_copy = df.copy() # Create a copy to avoid modifying the original DataFrame
        df_copy['Regions_Lists'] = df_copy['Regions_Lists'].apply(json.dumps)
        df_copy['Sector_Lists'] = df_copy['Sector_Lists'].apply(json.dumps)

        # Write DataFrame to SQLite database
        df_copy.to_sql(table_name, conn, if_exists='replace', index=False)
        print("Table created and loaded successfully")

    except sqlite3.Error as e:
        print("Error connecting to SQLite database:", e)
    finally:
        # Close connection
        conn.close()

create_and_load_table(CurrentcountriesTable, "Actualcountries", DB)
```

ⓘ Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)

```
try:  
    # Connect to SQLite database  
    conn = sqlite3.connect(db_name)  
    print("Opened database successfully")  
    # Convert lists to strings  
    df_copy = df.copy() # Create a copy to avoid modifying the original DataFrame  
    df_copy['Regions_Lists'] = df_copy['Regions_Lists'].apply(json.dumps)  
    df_copy['Sector_Lists'] = df_copy['Sector_Lists'].apply(json.dumps)  
  
    # Write DataFrame to SQLite database  
    df_copy.to_sql(table_name, conn, if_exists='replace', index=False)  
    print("Table created and loaded successfully")  
except sqlite3.Error as e:  
    print("Error connecting to SQLite database:", e)  
finally:  
    # Close connection  
    conn.close()  
create_and_load_table(CurrentcountriesTable, "Actualcountries", DB)  
def create_and_load_table(df, table_name, db_name):  
    try:  
        # Connect to SQLite database  
        conn = sqlite3.connect(db_name)  
        print("Opened database successfully")  
  
        # Write DataFrame to SQLite database  
        df.to_sql(table_name, conn, if_exists='replace', index=False)  
        print("Table created and loaded successfully")  
  
    except sqlite3.Error as e:  
        print("Error connecting to SQLite database:", e)  
    finally:  
        # Close connection  
        conn.close()  
# Example usage to load DimTablelanguages and DimTableOrganizations to database DB  
create_and_load_table(DimTablelanguages, "DimTablelanguages", DB)  
create_and_load_table(DimTableOrganizations, "DimTableOrganizations", DB)
```

ⓘ Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)

```
try:  
    # Connect to SQLite database  
    conn = sqlite3.connect(db_name)  
    print("Opened database successfully")  
    # Convert unsupported types to strings  
    df_copy = df.copy() # Create a copy to avoid modifying the original DataFrame  
    for column in df_copy.columns:  
        if df_copy[column].dtype == 'object':  
            # Convert lists, dicts, or other unsupported types to JSON strings  
            df_copy[column] = df_copy[column].apply(lambda x: json.dumps(x) if isinstance(x, (list, dict)) else x)  
    # Write DataFrame to SQLite database  
    df_copy.to_sql(table_name, conn, if_exists='replace', index=False)  
    print(f"Table '{table_name}' created and loaded successfully")  
except sqlite3.Error as e:  
    print(f"Error connecting to SQLite database: {e}")  
finally:  
    # Close connection  
    conn.close()  
  
# Create and load tables  
create_and_load_table(DataCleanAndPrepareForML, "DataCleanAndPrepareForML", DB)  
  
def create_and_load_table(df, table_name, db_name):  
    try:  
        # Connect to SQLite database  
        conn = sqlite3.connect(db_name)  
        print("Opened database successfully")  
        # Convert GeoPandas geometries to WKT (Well-Known Text)  
        df_copy = df.copy() # Create a copy to avoid modifying the original DataFrame  
        if isinstance(df_copy, gpd.GeoDataFrame):  
            df_copy['geometry'] = df_copy['geometry'].apply(lambda geom: geom.wkt)  
        # Write DataFrame to SQLite database  
        df_copy.to_sql(table_name, conn, if_exists='replace', index=False)  
        print(f"Table '{table_name}' created and loaded successfully")  
    except sqlite3.Error as e:  
        print(f"Error connecting to SQLite database: {e}")  
    finally:  
        # Close connection  
        conn.close()  
create_and_load_table(WordGeoPandasBase, "WordGeoPandasBase", DB)  
create_and_load_table(GisDataWithImputationTable, "GisDataWithImputationTable", DB)
```

ⓘ Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)

DBeaver 23.3.4 - <.db> Script

Fichiers Edition Navigation Rechercher Editeur SQL Base de données Fenêtres Aide

Navigateur de ... Projets

Enter a part of object name here

Actualcountries

DataCleanAndPrepareForML

DimTableLanguages

DimTableOrganizations

GisDataWithImputationTable

WordGeoPandasBase

Views

Indexes

Sequences

Table Triggers

Data Types

DBeaver Sample Database (SQLite)

OneDrive

OneDrive 2

Actualcountries 1

SELECT \* FROM Actualcountries;

Entrez une expression SQL pour filtrer les résultats (utilisez Ctrl+Espace)

Grille	123 countryId	ABC iso3	ABC m49	123 isCurrent	123 isSOWC	123 isCountDown	123 isPublished	ABC cndcountryCode	ABC Regions_Lists	ABC Language1	ABC Value1
1	ABW	533		1	0	0	1	ABW	["DEVPING", "SDS", "AMERICAS", "CARIBB", "GLOBAL", "LA"]	English	Aruba
2	AFG	4		1	0	1	1	AFG	["DEVPING", "LDC", "LLDC", "SOUTHCEASIA", "ASIA", "GL"]	English	Afghanistan
3	AGO	24		1	1	1	1	AGO	["MIDDLEAFR", "DEVPING", "SSA", "LDC", "SSA", "AFRICA"]	English	Angola
4	AIA	660		1	1	0	0	AIA	["DEVPING", "SDS", "AMERICAS", "CARIBB", "GLOBAL", "LA"]	English	Anguilla
5	ALA	248		1	0	0	0	ALA	["NORTHEUR", "DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEU"]	English	Åland Islands
6	ALB	8		1	1	0	0	ALB	["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEU"]	English	Albania
7	AND	20		1	1	0	0	AND	["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEU"]	English	Andorra
8	ARE	784		1	1	0	0	ARE	["DEVPING", "WESTASI", "WASINAFR", "ASIA", "GLOBAL", "LA"]	English	United Arab Emirates
9	ARG	32		1	1	0	0	ARG	["SOUTHAME", "DEVPING", "AMERICAS", "GLOBAL", "LATIN"]	English	Argentina
10	ARM	51		1	1	0	0	ARM	["CAUCSEN", "DEVPING", "LLDC", "WASINAFR", "ASIA", "G"]	English	Armenia
11	ASM	16		1	0	0	0	ASM	["OCEAN", "DEVPING", "SDS", "GLOBAL", "OCEANIA", "PO"]	English	American Samoa
12	ATF	260		1	0	0	0	ATF	["DEVPING", "SSA", "SSA", "AFRICA", "EASTAFR", "GLOBAL"]	English	French Southern and Antarctic Terr
13	ATG	28		1	1	0	0	ATG	["DEVPING", "SDS", "AMERICAS", "CARIBB", "GLOBAL", "LA"]	English	Antigua and Barbuda
14	AUS	36		1	1	0	0	AUS	["DEV", "AUSNZ", "GLOBAL", "OCEANIA", "IND", "EAP", "W"]	English	Australia
15	AUT	40		1	1	0	0	AUT	["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "WESTEUR"]	English	Austria
16	AZE	31		1	1	1	1	AZE	["CAUCSEN", "DEVPING", "LLDC", "WASINAFR", "ASIA", "G"]	English	Azerbaijan
17	BDI	108		1	1	1	1	BDI	["DEVPING", "SSA", "LDC", "LLDC", "SSA", "AFRICA", "EAST"]	English	Burundi
18	BEL	56		1	1	0	0	BEL	["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "WESTEUR"]	English	Belgium
19	BEN	204		1	1	1	1	BEN	["WESTAFR", "DEVPING", "SSA", "LDC", "SSA", "AFRICA", "G"]	English	Benin
20	BES	535		1	0	0	0	BES	["DEVPING", "SDS", "AMERICAS", "CARIBB", "GLOBAL", "LA"]	English	Bonaire Sint Eustatius and Saba
21	BFA	854		1	1	1	1	BFA	["WESTAFR", "DEVPING", "SSA", "LDC", "LLDC", "SSA", "AFF"]	English	Burkina Faso
22	BGD	50		1	1	1	1	BGD	["DEVPING", "LDC", "SOUTHCEASIA", "ASIA", "GLOBAL", "LA"]	English	Bangladesh
23	BGR	100		1	1	0	0	BGR	["DEV", "NORTHAMEUR", "EASTEUR", "EUROPE", "GLOBAL"]	English	Bulgaria
24	BHR	48		1	1	0	0	BHR	["DEVPING", "WESTASI", "WASINAFR", "ASIA", "GLOBAL", "LA"]	English	Bahrain
25	BHS	44		1	1	0	0	BHS	["DEVPING", "SDS", "AMERICAS", "CARIBB", "GLOBAL", "LA"]	English	Bahamas
26	BIH	70		1	1	0	0	BIH	["DEV", "NORTHAMEUR", "EUROPE", "GLOBAL", "SOUTHEU"]	English	Bosnia and Herzegovina
27	BLM	652		1	0	0	0	BLM	["DEVPING", "AMERICAS", "CARIBB", "GLOBAL", "LATINAM"]	English	Saint Barthélemy
28	BLR	112		1	1	0	0	BLR	["DEV", "NORTHAMEUR", "EASTEUR", "EUROPE", "GLOBAL"]	English	Belarus
29	BLZ	84		1	1	0	0	BLZ	["DEVPING", "SDS", "AMERICAS", "CENTRALAME", "GLOBAL"]	English	Belize
30	BMU	60		1	0	0	0	BMU	["DEV", "NORTHAMEUR", "AMERICAS", "GLOBAL", "NORTF"]	English	Bermuda
31	BOL	68		1	1	1	1	BOL	["SOUTHAME", "DEVPING", "LLDC", "AMERICAS", "GLOBAL"]	English	Bolivia (Plurinational State of)
32	BRA	76		1	1	1	1	BRA	["SOUTHAME", "DEVPING", "AMERICAS", "GLOBAL", "LATIN"]	English	Brazil

Régénérer Save Cancel Exporter les résultats ... 200 249 249 row(s) fetched - 0,015s (0,015s fetch), on 2024-06-05 at 21:22:58

Actualcountries

Taper ici pour rechercher

66°F Clear

PST fr

FRA FR 21:26 05/06/2024

ⓘ Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)

Fichiers Edition Navigation Rechercher Editeur SQL Base de données Fenêtres Aide

Appliquer (commit) Retour arrière (rollback) Auto .db < N/A Insertion avancée 2 : 40 : 126 Sel: 0 | 0

Navigateur de ... Projets .db Script

Enter a part of object name here

Tables Actualcountries DataCleanAndPrepareForML DimTableLanguages DimTableOrganizations GisDataWithImputationTable WordGeoPandasBase Views Indexes Sequences Table Triggers Data Types DBeaver Sample Database (SQLite) OneDrive OneDrive 2

-- Get all columns from DataCleanAndPrepareForML table using SELECT statement in SQL.  
SELECT \* FROM DataCleanAndPrepareForML;

DataCleanAndPrepareForML 1

Entrez une expression SQL pour filtrer les résultats (utilisez Ctrl+Espace)

Grille	ABC name	ABC continent	123 Area	123 pop_est	123 gdp_md_est	ABC Sector_Lists	123 Total_Organization_By_country	123 Total_Sector_By_country	123 pop_density	ABC GDP_Category	ABC Priority
1	Fiji	Oceania	1,6395109959	889 953	5 496 [7]		5	1	542 816 121 529,6047	Poor	Yes
2	United Republic of Tanzania	Africa	76,3019635909	58 005 463	63 177 [2, 7, 18]		5	3	760 209,3	Middle-income	Yes
3	Western Sahara	Africa	8,6039842075	603 253	907 []		2	0	70 113,22	Poor	No
4	Canada	North America	1712,995276494	37 589 262	1 736 425 []		5	0	21 943 588 279,332867	Rich	No
5	United States	North America	1122,2819207781	328 239 523	21 433 226 []		5	0	292 475,1	Rich	No
6	Kazakhstan	Asia	330,5868222022	18 513 930	181 665 []		5	0	56 003 230 487,74754	Middle-income	No
7	Uzbekistan	Asia	49,9588206562	33 580 650	57 921 []		5	0	672 166 587 580,2726	Middle-income	No
8	Papua New Guinea	Oceania	37,991090783	8 776 109	24 829 [7]		5	1	231 004 396 533,91388	Middle-income	Yes
9	Indonesia	Asia	148,1358213808	270 625 568	1 119 190 [2, 7, 18]		5	3	1 826 874 590 342,227	Rich	Yes
10	Argentina	South America	278,9233922225	44 938 712	445 445 []		5	0	161 114 891 232,05298	Rich	No
11	Chile	South America	86,5084692111	18 952 038	282 318 []		5	0	219 077 255 358,154	Rich	No
12	Democratic Republic of Congo	Africa	189,5152324776	86 790 567	50 400 [2, 7, 18]		5	3	457 960,9	Middle-income	Yes
13	Somalia	Africa	39,5281864804	10 192 317,3	4 719 [7, 18]		5	2	257 849 352 766,4034	Poor	Yes
14	Kenya	Africa	48,0331948213	52 573 973	95 503 [2, 7, 18]		5	3	1 094 534 169 455,3323	Middle-income	Yes
15	Sudan	Africa	156,4445432974	42 813 238	30 513 [7, 18]		5	2	273 663 990 431,43903	Middle-income	Yes
16	Chad	Africa	107,1673639782	15 946 876	11 314 [2, 7, 18]		5	3	148 803 473 445,99567	Poor	Yes
17	Haiti	North America	2,4471136749	11 263 077	14 332 [2, 7]		5	2	4 602 596 567 406,481	Middle-income	Yes
18	Dominican Republic	North America	4,1292064987	10 738 958	88 941 []		5	0	2 600 732	Middle-income	No
19	Russian Federation	Europe	2 931,8319455266	144 373 535	1 699 876 []		5	0	49 243,46	Rich	No
20	Bahamas	North America	1,3997566129	389 482	13 578 []		5	0	278 249 801 713,14484	Middle-income	No
21	Falkland Islands (Malvinas)	South America	2,12875	3 398	282 []		2	0	1 596,242	Poor	No
22	Norway	Europe	90,4962551806	5 347 896	403 336 []		5	0	59 095 218 794,69693	Rich	No
23	Greenland	North America	677,5095646642	56 225	3 051 []		2	0	82 987 758,302522	Poor	No
24	French Southern and Antarctic Lands	Oceania	1,432928125	140	16 []		2	0	97,7024	Poor	No
25	Timor-Leste	Asia	1,2089391926	1 293 119	2 017 []		5	0	1 069 631 134 437,5223	Poor	No
26	South Africa	Africa	112,7185236204	58 558 270	351 431 []		5	0	519 508 844 856,76666	Rich	No
27	Lesotho	Africa	2,561879916	2 125 268	2 376 [2, 7, 18]		5	3	829 573 621 606,1439	Poor	Yes
28	Mexico	North America	175,2073411732	127 575 529	1 268 870 []		5	0	728 140 317 327,8298	Rich	No
29	Uruguay	South America	17,027171031	3 461 734	56 045 []		5	0	203 306 467 862,373	Middle-income	No
30	Brazil	South America	710,1852431534	211 049 527	1 839 758 []		5	0	297 175 320 150,12714	Rich	No
31	Bolivia	South America	92,0771733122	11 513 100	40 895 []		5	0	125 037 504 800,05258	Middle-income	No
	Bolivia	South America	100 000 000 000 000	22 510 452	226 010 []		5	0	200 000 000 000 000	Middle-income	No

Régenerer Save Cancel Exporter les résultats ... 200 177 177 row(s) fetched - 0,002s (0,001s fetch), on 2024-06-05 at 21:29:18

.db

66°F Clear 21:29 FRA FR 05/06/2024

*i* Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)

Screenshot of the DBeaver application interface showing a database connection named ".db".

The left sidebar shows the project structure under "Projets" and the database structure under ".db", including tables like "Actualcountries", "DataCleanAndPrepareForML", "DimTablelanguages", "DimTableOrganizations", "GisDataWithImputationTable", "WordGeoPandasBase", "Views", "Indexes", "Sequences", "Table Triggers", and "Data Types".

The main area displays the "DimTablelanguages" table with the following data:

	name	isO6392
1	English	eng
2	French	fre
3	Spanish	spa
4	Arabic	ara

The status bar at the bottom shows the following information:

- File: .db
- Language: PST | fr | Inscriptible
- Time: 2 : 33 : 112
- Selection: Sel: 0 | 0
- System: FRA FR
- Date: 21:31 05/06/2024
- Weather: 66°F Clear

ⓘ Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)

SQL Appliquer (commit) Retour arrière (rollback) Auto .db < N/A Insertion avancée Sel: 0 | 0

Navigateur de ... Projets Enter a part of object name here

Tables

- Actualcountries
- DataCleanAndPrepareForML
- DimTableLanguages
- DimTableOrganizations
- GisDataWithImputationTable
- WordGeoPandasBase
- Views
- Indexes
- Sequences
- Table Triggers
- Data Types

DBeaver Sample Database (SQLite)

OneDrive OneDrive 2

Script

```
-- Get all columns from DimTableOrganizations table using SELECT statement in SQL.  
SELECT * FROM DimTableOrganizations;
```

DimTableOrganizations 1

Entrez une expression SQL pour filtrer les résultats (utilisez Ctrl+Espace)

	ABC organization	ABC code
1	United Nations Children's Fund	UNICEF
2	United Nations Statistical Division	UNSD
3	World Bank	WB
4	UNICEF Private Fundraising and Partnerships Division	PFP
5	UNICEF Europe and Central Asia Regional Office	ECARO
6	United Nations Children's Fund Programme Division	UNICEF_PD
7	United Nations Population Fund	UNFPA
8	World Health Organization	WHO
9	UNICEF Middle East and North Africa Office	MENARO

Grille

Texte

Régenerer Save Cancel Exporter les résultats ... 200 9 9 row(s) fetched - 0,001s, on 2024-06-05 at 21:32:34

.db

Taper ici pour rechercher

PST fr Inscriptible Insertion avancée 2 : 37 : 120 Sel: 0 | 0

66°F Clear 21:33 FRA FR 05/06/2024

ⓘ Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)

SQL > Appliquer (commit) Retour arrière (rollback) Auto .db < N/A >

Navigateur de ... Projets

Enter a part of object name here

Tables

- Actualcountries
- DataCleanAndPrepareForML
- DimTableLanguages
- DimTableOrganizations
- GisDataWithImputationTable
- WordGeoPandasBase
- Views
- Indexes
- Sequences
- Table Triggers
- Data Types
- DBeaver Sample Database (SQLite)
- OneDrive
- OneDrive 2

Script

```
-- Get all columns from GisDataWithImputationTable table using SELECT statement in SQL.
SELECT * FROM GisDataWithImputationTable;
```

GisDataWithImputationTable 1

Entrez une expression SQL pour filtrer les résultats (utilisez Ctrl+Espace)

Grille	123 pop_est	ABC continent	ABC name	ABC iso_a3	123 gdp_md_est	ABC geometry	123 area
1	889 953	Oceania	Fiji	FJI	5 496	MULTIPOLYGON (((180 -16.067132663642447, 180 -16.555216566639196, 179.36414266196414 -16.801354076946883, 178.72505936299711 -17.01204167436804, 1, 1, 16395109959	
2	58 005 463	Africa	Tanzania	TZA	63 177	POLYGON ((33.90371119710453 -0.9500000000000001, 34.072619999999997 -1.0598199999999451, 37.6988899999999484, 37.7669 -3.677120000, 19,8798157027	
3	603 253	Africa	W. Sahara	ESH	907	POLYGON ((-8.66558965454809 27.656425889592356, -8.665124477564191 27.589479071558227, -8.684399786809053 27.395744126896005, -8.6872936670174 25, 19,8798157027	
4	37 589 262	North America	Canada	CAN	1 736 425	MULTIPOLYGON (((-122.84000000000003 49.000000000000114, -122.974210000000014 49.02537777778, -124.91024 49.98456, -125.62461 50.41656000000004, 171,225276494	
5	328 239 523	North America	United States of America	USA	21 433 226	MULTIPOLYGON (((-122.84000000000003 49.000000000000114, -120.49.000000000000114, -117.03121 49, -116.04818 49, -113.49, -110.050000000000149, -107.0, 19,8798157027	
6	18 513 930	Asia	Kazakhstan	KAZ	181 665	POLYGON ((87.35997033076265 49.21498078062912, 86.59877648310336 48.549181626980626, 85.7682328633083 48.45575063739699, 85.72048383987067 47.4529, 330,5868222022	
7	33 580 650	Asia	Uzbekistan	UZB	57 921	POLYGON ((55.9681913592891 41.30864166926936, 55.928917270741096 44.9958546615911, 58.50312706892845 45.586804307632974, 58.68998904809581 45.50, 49,9588206562	
8	8 776 109	Oceania	Papua New Guinea	PNG	24 829	MULTIPOLYGON (((141.00021040259185 -2.60015105551566, 142.73524661679147 -3.28915292726321, 144.58397098203324 -3.861417738463416, 145.2731795595, 37,991090783	
9	270 625 568	Asia	Indonesia	IDN	1 119 190	MULTIPOLYGON (((141.00021040259185 -2.60015105551566, 141.01705691951895 -5.859021905138071, 141.03385176001382 -9.117892754760483, 140.143415155, 148,1358213808	
10	44 938 712	South America	Argentina	ARG	445 445	MULTIPOLYGON (((-68.63401022758323 -52.63637045887449, -68.25 -53.1, -67.75 -53.85, -66.45 -54.45, -65.05 -54.699999999999996, -65.5 -55.2, -66.45 -55.25, -278,9233922225	
11	18 952 038	South America	Chile	CHL	282 318	MULTIPOLYGON (((-68.63401022758323 -52.63637045887449, -68.633350000000001 -54.869499999999995, -67.56244 -54.87001, -66.9592000000001 -54.89681000, 86,5084692111	
12	86 790 567	Africa	Dem. Rep. Congo	COD	50 400	POLYGON ((29.33997592900346 -4.4999834122940925, 29.51998660572928 -5.419978936386315, 29.419992710088167 -5.939998874539434, 29.62003217949001, 19,8798157027	
13	10 192 317,3	Africa	Somalia	SOM	4 719	POLYGON ((41.58513 -1.683250000000001, 40.993 -0.85829, 40.98105 2.784519999999997, 41.8550309264397 3.918911920483727, 42.128609999999995 4.23413, 39,5281864804	
14	52 573 973	Africa	Kenya	KEN	95 503	POLYGON ((39.20222 -4.67677, 37.7669 -3.6771200000000004, 37.698689999999994, 30.476199999999997 -10.059819999999451, 33.903711197, 48,0331948213	
15	42 813 238	Africa	Sudan	SDN	30 513	POLYGON ((24.667369012152085 8.229187933785468, 23.805813429466752 8.666318874542526, 23.459012892355986 8.954285793488893, 23.394779087017184 9.2, 156,4445432974	
16	15 946 876	Africa	Chad	TCD	11 314	POLYGON ((23.83766000000014 19.58047000000105, 23.88689000000108 15.610839999999996, 23.024590000000103 15.68072000000065, 22.567950000000114, 107,1673639782	
17	11 263 077	North America	Haiti	HTI	14 332	POLYGON ((-71.71236141629296 19.714455878167357, -71.62487321642283 19.169837958243306, -71.7013026597825 18.785416978420453, -71.94511206733556 1, 2,4471136749	
18	10 738 958	North America	Dominican Rep.	DOM	88 941	POLYGON ((-71.70830481635805 18.044997056546094, -71.68773759630588 18.31666006110447, -71.94511206733556 18.6169001327206, -71.7013026597825 18., 19,8798157027	
19	144 373 535	Europe	Russia	RUS	1 699 876	MULTIPOLYGON (((180.00000000000006 71.51571433642829, 180.00000000000006 70.83219920854673, 178.9034249999997 70.78114000000001, 178.7253 71.098, 19,8798157027	
20	389 482	North America	Bahamas	BHS	13 578	MULTIPOLYGON (((-78.98 26.79000000000003, -78.51 26.87000000000005, -77.85000000000001 26.84, -77.82000000000001 26.580000000000005, -78.910000000000005, 1,3997566129	
21	3 398	South America	Falkland Is.	FLK	282	POLYGON ((-61.2 -51.85, -60 -51.25, -59.15 -51.5, -58.550000000000004 -51.10000000000001, -57.75 -51.5, -58.050000000000004 -51.900000000000006, -59.4000, 19,8798157027	
22	5 347 896	Europe	Norway	NOR	403 336	MULTIPOLYGON (((15.14282000000002 79.67431, 15.52255 80.01608, 16.99085000000002 80.05086, 18.25183 79.70175, 21.54383 78.95611, 19.02737 78.5626, 18., 90,4962551806	
23	56 225	North America	Greenland	GRL	3 051	POLYGON ((-46.76379 82.62796, -43.40644 83.22516000000002, -39.89753 83.18018, -38.62214 83.54905, -35.08787 83.64513000000001, -27.10046 83.51966, -20.8, 677,5095646642	
24	140	Seven seas (open ocean)	Fr. S. Antarctic Lands	ATF	16	POLYGON ((68.935 -48.62500000000001, 69.58 -48.94000000000005, 70.525 -49.06500000000005, 70.56 -49.4255, 70.28 -49.71, 68.745 -49.77500000000006, 68.7, 19,8798157027	
25	1 293 119	Asia	Timor-Leste	TLS	2 017	POLYGON ((124.96862848911623 -8.892790215697083, 125.08624637258026 -8.65688730228468, 125.94707238169826 -8.43209482181503, 126.64470421763855, 1,2089391926	
26	58 558 270	Africa	South Africa	ZAF	351 431	POLYGON ((16.344976840895242 -28.5767050106977, 16.824017368240902 -28.08216155366447, 17.21892863815404 -28.35594329194681, 17.387497185951503, 112,7185236204	
27	2 125 268	Africa	Lesotho	LSO	2 376	POLYGON ((28.978262566857243 -28.95559661226171, 29.32516645683259 -29.257386976846256, 29.018415154748027 -29.743765557737, 28.84839969250774, 2,561879916	
28	127 575 529	North America	Mexico	MEX	1 268 870	POLYGON (((-117.12775999999985 32.53533999999996, -115.9913499999995 32.61239000000012, -114.72138999999993 32.720899999992, -114.815 32.52528000, 175,2073411732	
29	3 461 734	South America	Uruguay	URY	56 045	POLYGON ((-57.62513342958296 -30.2162948545426, -56.976025763564735 -30.1096837463127, -55.97324459494094 -30.883075860316303, -55.601510179249, 17,027171031	
30	211 049 527	South America	Brazil	BRA	1 839 758	POLYGON ((-53.373661668498244 -33.76837778090764, -53.6505439927181 -33.2020408298183, -53.209588995971544 -32.72766611097424, -53.787951626182, 710,1852431534	
31	11 513 100	South America	Bolivia	BOL	40 895	POLYGON ((-69.85267810736496 -10.95173407502194, -68.78615759954948 -11.03638030359628, -68.27125362819326 -11.0452117273682, -68.04819230820539, 19,8798157027	
32	32 510 453	South America	Peru	PER	226 848	POLYGON ((-69.89363521999663 -4.2981869441943275, -70.9746788463023 -4.251264743673303, -70.92884334988358 -4.401591485210368, -71.74840572781655, -108,0498212638	

Régénérer Save Cancel Exporter les résultats ... 200 177 177 row(s) fetched - 0,018s (0,015s fetch), on 2024-06-05 at 21:35:18

Project - General

Name

- Bookmarks
- Diagrams
- Scripts

Record

PST fr Inscriptible Insertion avancée 2 : 42 : 130 Sel 0 | 0

66°F Clear 21:36 FRA FR 05/06/2024

Taper ici pour rechercher

Windows Taskbar icons



ⓘ Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)

Navigateur de ... Projets .db Enter a part of object name here

SQL Appliquer (commit) Retour arrière (rollback) Auto .db < N/A >

\*<.db> Script X  
--This SQL query selects specific columns (name, continent, gdp\_md\_est, Armed\_Conflict, and Priority) from the DataCleanAndPrepareForML table.  
--It filters the rows to include only those where both the Priority and Armed\_Conflict columns are set to "Yes".  
SELECT name,continent,gdp\_md\_est,Armed\_Conflict,Priority FROM DataCleanAndPrepareForML  
WHERE Priority = "Yes" AND Armed\_Conflict = "Yes";

DataCleanAndPrepareForML 1 X  
SELECT name,continent,gdp\_md\_est,Armed\_Conflict,Priority FROM Data Entrez une expression SQL pour filtrer les résultats (utilisez Ctrl+Espace)

	ABC name	ABC continent	123 gdp_md_est	ABC Armed_Conflict	ABC Priority
1	Fiji	Oceania	5496	Yes	Yes
2	Papua New Guinea	Oceania	24829	Yes	Yes
3	Indonesia	Asia	1119190	Yes	Yes
4	Democratic Republic of Congo	Africa	50400	Yes	Yes
5	Somalia	Africa	4719	Yes	Yes
6	Kenya	Africa	95503	Yes	Yes
7	Sudan	Africa	30513	Yes	Yes
8	Chad	Africa	11314	Yes	Yes
9	Haiti	North America	14332	Yes	Yes
10	Lesotho	Africa	2376	Yes	Yes
11	Mauritania	Africa	7600	Yes	Yes
12	Nigeria	Africa	448120	Yes	Yes
13	Ghana	Africa	66983	Yes	Yes
14	Côte d'Ivoire	Africa	58539	Yes	Yes
15	Guinea	Africa	12296	Yes	Yes
16	Guinea-Bissau	Africa	1339	Yes	Yes
17	Liberia	Africa	3070	Yes	Yes
18	Sierra Leone	Africa	4121	Yes	Yes
19	Burkina Faso	Africa	15990	Yes	Yes
20	Congo	Africa	12267	Yes	Yes
21	Iraq	Asia	234094	Yes	Yes
22	Cambodia	Asia	27089	Yes	Yes
23	Lao People's Democratic Republic	Asia	18173	Yes	Yes
24	Viet Nam	Asia	261921	Yes	Yes
25	Pakistan	Asia	278221	Yes	Yes
26	Afghanistan	Asia	19291	Yes	Yes
27	Syrian Arab Republic	Asia	98830	Yes	Yes
28	Ukraine	Europe	153781	Yes	Yes
29	Philippines	Asia	376795	Yes	Yes
30	Ethiopia	Africa	95912	Yes	Yes
31	Somalia	Africa	17836	Yes	Yes
32	Uganda	Africa	35165	Yes	Yes
33	Others	Others	11200	Yes	Yes

Régenerer Save Cancel Exporter les résultats ... 200 33 33 row(s) fetched - 0,001s, on 2024-06-05 at 22:01:46

.db X

Taper ici pour rechercher

PST fr Inscriptible Insertion avancée 2 : 113 : 257 Sel 0 | 0

FRA FR Latest games 22:06 05/06/2024

ⓘ Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)

Navigateur de ... X Projets

Enter a part of object name here

Tables

- Actualcountries
- DataCleanAndPrepareForML
- DimTableLanguages
- DimTableOrganizations
- GisDataWithImputationTable
- WordGeoPandasBase

Views

Indexes

Sequences

Table Triggers

Data Types

DBeaver Sample Database (SQLite)

OneDrive

OneDrive 2

Script X

```
-- This SQL query selects specific columns (name, continent, gdp_md_est, GDP_Category, Armed_Conflict, and Priority) from the DataCleanAndPrepareForML table.  
-- It filters the rows to include only those where the Priority and Armed_Conflict columns are both set to "Yes" and the GDP_Category column is set to 'Poor'.  
  
SELECT name, continent, gdp_md_est, GDP_Category, Armed_Conflict, Priority  
FROM DataCleanAndPrepareForML  
WHERE Priority = "Yes" AND Armed_Conflict = "Yes" AND GDP_Category = 'Poor';
```

DataCleanAndPrepareForML 1 X

Entrez une expression SQL pour filtrer les résultats (utilisez Ctrl+Espace)

	ABC name	ABC continent	123 gdp_md_est	ABC GDP_Category	ABC Armed_Conflict	ABC Priority
1	Fiji	Oceania	5496	Poor	Yes	Yes
2	Somalia	Africa	4719	Poor	Yes	Yes
3	Chad	Africa	11314	Poor	Yes	Yes
4	Lesotho	Africa	2376	Poor	Yes	Yes
5	Mauritania	Africa	7600	Poor	Yes	Yes
6	Guinea	Africa	12296	Poor	Yes	Yes
7	Guinea-Bissau	Africa	1339	Poor	Yes	Yes
8	Liberia	Africa	3070	Poor	Yes	Yes
9	Sierra Leone	Africa	4121	Poor	Yes	Yes
10	Congo	Africa	12267	Poor	Yes	Yes
11	South Sudan	Africa	11998	Poor	Yes	Yes

Project - General X

Name

- Bookmarks
- Diagrams
- Scripts

Record

Régénérer Save Cancel Exporter les résultats ... 200 11 11 row(s) fetched - 0,000s, on 2024-06-05 at 22:09:32

.db X

Taper ici pour rechercher

PST fr Inscriptible Insertion avancée Sel: 0 | 0

22:12 FRA FR 05/06/2024

ⓘ Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)

SQL    Appliquer (commit)    Retour arrière (rollback)    Auto    .db    < N/A >    Insertion avancée    Sel: 0 | 0

Navigateur de ...    Projets

Enter a part of object name here

.db

- Tables
  - Actualcountries
  - DataCleanAndPrepareForML
  - DimTablelanguages
  - DimTableOrganizations
  - GisDataWithImputationTable
  - WordGeoPandasBase
- Views
- Indexes
- Sequences
- Table Triggers
- Data Types

DBeaver Sample Database (SQLite)

OneDrive

OneDrive 2

Script X

```
--This SQL query calculates the sum of the Total_Sector_By_country column for each continent.  
--It groups the results by the continent column and orders them in descending order of the summed values, providing the total sector count for each continent.  
  
SELECT continent, SUM(Total_Sector_By_country) AS Total_Sector  
FROM DataCleanAndPrepareForML  
GROUP BY continent  
ORDER BY Total_Sector DESC;
```

DataCleanAndPrepareForML 1 X

SELECT continent, SUM(Total\_Sector\_By\_country) AS Total\_Sector FROM [ ] Entrez une expression SQL pour filtrer les résultats (utilisez Ctrl+Espace)

Grille	continent	Total_Sector
1	Africa	88
2	Asia	37
3	Oceania	3
4	North America	3
5	Europe	2
6	South America	0
7	Antarctica	0

Project - General X

Name

- Bookmarks
- Diagrams
- Scripts

Régenerer    Save    Cancel    Exporter les résultats ...    200    7    7 row(s) fetched - 0,000s, on 2024-06-05 at 22:24:04

.db X

PST fr Inscriptible Insertion avancée 3 : 1 : 256 Sel: 0 | 0

Taper ici pour rechercher

WMO: 80% chance E... FRA FR 22:27 05/06/2024

*i* Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)

SQL | Appliquer (commit) Retour arrière (rollback) Auto .db < N/A > ...

Navigateur de ... Projets .db Script X

Enter a part of object name here

.db Tables Actualcountries DataCleanAndPrepareForML DimTableLanguages DimTableOrganizations GisDataWithImputationTable WordGeoPandasBase Views Indexes Sequences Table Triggers Data Types DBeaver Sample Database (SQLite) OneDrive OneDrive 2

--This query prints a list of continents and the total number of armed conflicts (where Armed\_Conflict is "Yes") for each, sorted from highest to lowest number of conflicts.

```
SELECT continent,
       COUNT(CASE WHEN Armed_Conflict = 'Yes' THEN 1 END) AS TotalConflicts
  FROM DataCleanAndPrepareForML
 GROUP BY continent
 ORDER BY TotalConflicts DESC;
```

DataCleanAndPrepareForML 1 X

SELECT continent, COUNT(CASE WHEN Armed\_Conflict = 'Yes' THEN 1 ELSE 0 END) AS TotalConflicts

continent	TotalConflicts
Africa	27
Asia	25
South America	7
North America	6
Europe	4
Oceania	3
Antarctica	1

Project - General X

Name Bookmarks Diagrams Scripts

Régenerer Save Cancel Exporter les résultats ... 200 7 row(s) fetched - 0,001s, on 2024-06-05 at 22:39:38

.db X

PST fr Inscriptible Insertion avancée 1 : 174 : 173 Sel 0 | 0

Taper ici pour rechercher

66°F Clear FRA FR 22:44 05/06/2024

ⓘ Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)

Screenshot of the DBeaver SQL Workbench interface showing a query execution and results.

The interface includes:

- Toolbar:** Includes icons for file operations, database connection, and various tools.
- Project Navigator:** Shows the current project structure, including a database named ".db" containing tables like "Actualcountries", "DataCleanAndPrepareForML", etc.
- Script Editor:** Displays a SQL query:

```
-- This query prints a result where each row represents a continent and two columns represent the number of armed conflicts for each status ("Yes" and "No").  
SELECT  
    continent,  
    COUNT(CASE WHEN Armed_Conflict = 'Yes' THEN 1 END) AS Conflict_Yes,  
    COUNT(CASE WHEN Armed_Conflict = 'No' THEN 1 END) AS Conflict_No  
FROM  
    DataCleanAndPrepareForML  
GROUP BY  
    continent  
ORDER BY  
    continent;
```
- Result Grid:** Shows the output of the query:

continent	Conflict_Yes	Conflict_No
Africa	27	24
Antarctica	1	0
Asia	25	23
Europe	4	35
North America	6	12
Oceania	3	5
South America	7	5
- Status Bar:** Shows the number of rows fetched (7), the execution time (0,001s), and the date and time (2024-06-05 at 22:48:47).
- System Tray:** Shows the Windows logo, a search bar, and system icons for battery, signal, and date/time (FRA FR, 22:52, 05/06/2024).

ⓘ Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)

Fichiers Edition Navigation Rechercher Editeur SQL Base de données Fenêtres Aide

Appliquer (commit) Retour arrière (rollback) Auto .db < N/A > Insertion avancée Sel: 0 | 0

Navigateur de ... Projets Script

Enter a part of object name here

.db Tables Actualcountries DataCleanAndPrepareForML DimTableLanguages DimTableOrganizations GisDataWithImputationTable Colloons pop\_density ABC GDP\_Catégorie ABC Priority ABC Organization ABC Armed\_Conflict 123 pop\_est ABC continent ABC name ABC iso\_a3 ABC gdp\_md\_est ABC geometry Clefs Clefs étrangères Indexes Références Triggers WordGeoPandasBase Colloons pop\_density ABC continent ABC name ABC iso\_a3 ABC gdp\_md\_est ABC geometry Clefs Clefs étrangères Indexes Références Triggers Project - General

--Using "LEFT JOIN" to join DataCleanAndPrepareForML and Word GeoPandas Base tables, based on name and Order by name.

SELECT \* FROM DataCleanAndPrepareForML  
LEFT JOIN WordGeoPandasBase  
ON DataCleanAndPrepareForML.name = WordGeoPandasBase.name  
Order by name ;

SELECT \* FROM DataCleanAndPrepareForML LEFT JOIN WordGeoPandasBase Entrer une expression SQL pour filtrer les résultats (utilisez Ctrl+Espace)

	pop_density	ABC GDP_Catégorie	ABC Priority	ABC Organization	ABC Armed_Conflict	123 pop_est	ABC continent	ABC name	ABC iso_a3	123 gdp_md_est	ABC geometry
147	7849352766,4034	Middle-income	Yes	Yes	Yes	10192317,3	Africa	Somalia	SOM	4719	POLYGON ((41.58513 -1.6832500000000001, 40.993 -0.85829, 40.98105 2.78
148	508844856,76666	Rich	No	Yes	Yes	58558270	Africa	South Africa	ZAF	351431	POLYGON ((16.344976840895242 -28.5767050106977, 16.824017368240902
149	216073,3	Poor	Yes	Yes	Yes	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
150	3765213449,4559	Rich	No	Yes	No	47076781	Europe	Spain	ESP	1393490	POLYGON ((-7.453725551778092 37.0977875839607, -7.537105475281024
151	69421875691,618	Middle-income	Yes	Yes	No	21803000	Asia	Sri Lanka	LKA	84008	POLYGON ((81.78779500188914 7.523055324733164, 81.63732221876059 6.4
152	663990431,43903	Middle-income	Yes	Yes	Yes	42813238	Africa	Sudan	SDN	30513	POLYGON ((24.567369012152085 8.229187933785468, 23.805813429466752
153	71368550,817726	Poor	No	Yes	No	581363	South America	Suriname	SUR	3697	POLYGON ((-54.524754197799716 2.3118488631237852, -55.097587449755
154	9464357266,8271	Rich	No	Yes	No	10285453	Europe	Sweden	SWE	530883	POLYGON ((11.027368605196868 58.85614940045936, 11.468271925511146
155	6197771096,4639	Rich	No	Yes	No	8574832	Europe	Switzerland	CHE	703082	POLYGON ((9.59422610844635 47.52505809182027, 9.632931756232978 47.
156	933102	Middle-income	Yes	Yes	Yes	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
157	3077159474,7727	Poor	Yes	Yes	No	9321018	Asia	Tajikistan	TJK	8116	POLYGON ((67.8299962755952 37.144994004864685, 68.39203250516596 :
158	1414688804,0208	Rich	No	Yes	Yes	69625582	Asia	Thailand	THA	543548	POLYGON ((105.21877689007889 14.273211778210694, 104.2814180847366
159	9631134437,5223	Poor	No	Yes	No	1293119	Asia	Timor-Leste	TLS	2017	POLYGON ((124.96868248911623 -8.892790215697083, 125.0862463725802
160	4142356555,9683	Poor	Yes	Yes	No	8082366	Africa	Togo	TGO	5490	POLYGON ((0.8995630224740694 10.9973938236426, 0.7723556461714843
161	3056338028,1604	Middle-income	No	Yes	No	1394973	North America	Trinidad and Tobago	TTO	24269	POLYGON ((-61.68000000000001 10.76000000000002, -61.1050000000000
162	5063177148,8971	Middle-income	No	Yes	Yes	11694719	Africa	Tunisia	TUN	38796	POLYGON ((9.482139926805274 30.307556057246188, 9.05560265466815 3:
163	667336800,50092	Middle-income	No	Yes	No	5942089	Asia	Turkmenistan	TKM	40761	POLYGON ((52.50245975119615 41.7831553808637, 52.944293247291654 :
164	16079932637,846	Middle-income	Yes	Yes	Yes	44269594	Africa	Uganda	UGA	35165	POLYGON ((33.90371119710453 -0.9500000000000001, 31.86617000000006
165	0081756901,0203	Middle-income	Yes	Yes	Yes	44385155	Europe	Ukraine	UKR	153781	POLYGON ((32.159440000000013 52.06125000000014, 32.41205813978769 5:
166	77091426778,381	Rich	No	Yes	Yes	9770529	Asia	United Arab Emirates	ARE	421142	POLYGON ((51.57951867046327 24.245497137951105, 51.757440626844186
167	4053589087,8003	Rich	No	Yes	No	66834405	Europe	United Kingdom	GBR	2829108	MULTIPOLYGON ((((-6.197884894220991 53.867565009163364, -6.95373023
168	760209,3	Middle-income	Yes	Yes	No	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
169	292475,1	Rich	No	Yes	No	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
170	03306467862,373	Middle-income	No	Yes	Yes	3461734	South America	Uruguay	URY	56045	POLYGON ((-57.62513342958296 -30.21629485445426, -56.9760257635647:
171	2166587580,2726	Middle-income	No	Yes	No	33580650	Asia	Uzbekistan	UZB	57921	POLYGON ((55.96819135928291 41.30864166926936, 55.928917270741096 :
172	5003172822,8969	Poor	No	Yes	Yes	299882	Oceania	Vanuatu	VUT	934	MULTIPOLYGON (((167.2168013857696 -15.89184620530842, 167.84487674
173	3378397	Rich	Yes	Yes	Yes	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
174	70113,22	Poor	No	Yes	Yes	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]	[NULL]
175	7932523204,1531	Middle-income	Yes	Yes	No	29161922	Asia	Yemen	YEM	22581	POLYGON ((52.0000098002224 19.000003363516058, 52.78218427919205

Régénérer Save Cancel Exporter les résultats ... 200 177 177 row(s) fetched - 0.005s (0.002s fetch), on 2024-06-05 at 23:28:22

Project - General

Name Bookmarks Diagrams Scripts

Record

PST fr Inscriptible Insertion avancée 1 : 118 : 117 Sel: 0 | 0

65°F Clear FRA FR 23:41 05/06/2024

ⓘ Auto recovery contains some recovered files that haven't been opened.

View recovered files

File Edit Navigate Search SQL Editor Database Window Help

Auto Commit Rollback SQL

Script Script-1 Script-2 Script-3

--Joining DataCleanAndPrepareForML with WordGeoPandasBase to Retrieve iso\_a3, Geometry and Name Columns

```
SELECT DataCleanAndPrepareForML.*,
WordGeoPandasBase.iso_a3,
WordGeoPandasBase.geometry,
WordGeoPandasBase.name
-- List all other columns you need, excluding WordGeoPandasBase.pop_est, WordGeoPandasBase.continent, and WordGeoPandasBase.gdp_md_est
FROM DataCleanAndPrepareForML
LEFT JOIN WordGeoPandasBase
ON DataCleanAndPrepareForML.name = WordGeoPandasBase.name
ORDER BY DataCleanAndPrepareForML.name;
```

DataCleanAndPrepareForML(+) 1

SELECT DataCleanAndPrepareForML.\* , WordGeoPandasBase.iso\_a3 , WordGeoPandasBase.name Enter a SQL expression to filter results (use Ctrl+Space)

Grid	country	123 Total_Sector_By_country	123 pop_density	ABC GDP_Category	ABC Priority	ABC Organization	ABC Armed_Conflict	ABC iso_a3	ABC geometry	ABC name
1	5	2	598,201,922,732,577	Middle-income	Yes	Yes	Yes	AFG	POLYGON ((66.51860680528867 37.36278432875879, 67.07578209825962 37.35614390720929, 67.07578209825962 37.35614390720929, 67.36278432875879 37.36278432875879))	Afghanistan
2	5	0	896,089,543,974,2131	Middle-income	No	Yes	No	ALB	POLYGON ((21.0200403174764 40.84272695572588, 20.999989861747224 40.58000397395398, 20.999989861747224 40.58000397395398, 21.0200403174764 40.84272695572588))	Albania
3	5	0	201,556,626,112,36038	Middle-income	No	Yes	No	DZA	POLYGON ((-8.684399786809053 27.395744126896005, -8.665124477564191 27.58947907155822, -8.665124477564191 27.58947907155822, -8.684399786809053 27.395744126896005))	Algeria
4	5	0	307,195,629,890,5071	Middle-income	No	Yes	No	AGO	MULTIPOLYGON (((12.995517205465177 -4.781103203961884, 12.63161176926579 -4.991271254, 12.63161176926579 -4.991271254, 12.995517205465177 -4.781103203961884)))	Angola
5	5	0	744,754,0213920657	Poor	No	Yes	Yes	ATA	MULTIPOLYGON (((-48.66061601418252 -78.04701873159873, -48.1513964503784 -78.04701873159873, -48.1513964503784 -78.04701873159873, -48.66061601418252 -78.04701873159873)))	Antarctica
6	5	0	161,114,891,232,05298	Rich	No	Yes	Yes	ARG	MULTIPOLYGON (((-68.63401022758323 -52.63637045887449, -68.25 -53.1, -67.75 -53.85, -66.4°, -66.4°, -68.25 -53.1, -67.75 -53.85, -68.63401022758323 -52.63637045887449)))	Argentina
7	5	0	975,968,370,616,215	Middle-income	No	Yes	Yes	ARM	POLYGON ((46.50571984231797 38.770605373688294, 46.14362308124882 38.7412014837122, 46.14362308124882 38.7412014837122, 46.50571984231797 38.770605373688294))	Armenia
8	5	0	36,466,783,216,193054	Rich	No	Yes	No	AUS	MULTIPOLYGON (((147.6892597448818 -40.808258152022674, 148.289067824496 -40.87543751 Australia	Australia
9	5	0	872,044,427,885,5905	Rich	No	Yes	No	AUT	POLYGON ((16.97966782304037 48.123497515976305, 16.90375410326726 47.71486562762833, 16.90375410326726 47.71486562762833, 16.97966782304037 48.123497515976305))	Austria
10	5	0	1,039,462,946,870,2782	Middle-income	No	Yes	Yes	AZE	MULTIPOLYGON (((46.404950799348825 41.860675157227305, 46.68607059101661 41.82713715, 46.68607059101661 41.82713715, 46.404950799348825 41.860675157227305)))	Azerbaijan
11	5	0	278,249,801,713,14484	Middle-income	No	Yes	No	BHS	MULTIPOLYGON (((-78.98 26.790000000000003, -78.51 26.870000000000005, -77.850000000000001, -77.850000000000001, -78.98 26.790000000000003)))	Bahamas
12	5	2	13,750,843,739,415,682	Rich	Yes	Yes	No	BGD	POLYGON ((92.67227098182556 22.041238918541254, 92.65225711463799 21.324047552978485, 92.65225711463799 21.324047552978485, 92.67227098182556 22.041238918541254))	Bangladesh
13	5	0	334,504,027,829,09174	Middle-income	No	Yes	No	BLR	POLYGON ((28.1670942557794 56.16912995057879, 29.229513380660308 55.91834422466636, 29.229513380660308 55.91834422466636, 28.1670942557794 56.16912995057879))	Belarus
14	5	0	2,998,449,768,496,4253	Rich	No	Yes	No	BEL	POLYGON ((6.15665815595878 50.80372101501058, 6.04307335778111 50.128051662794235, 6.04307335778111 50.128051662794235, 6.15665815595878 50.80372101501058))	Belgium
15	5	0	208,454,715,220,83926	Poor	No	Yes	Yes	BLZ	POLYGON ((-89.14308041050333 17.808318996649405, -89.15090938999533 17.95546763760042, -89.15090938999533 17.95546763760042, -89.14308041050333 17.808318996649405))	Belize
16	5	2	1,224,034,323,888,1914	Middle-income	Yes	Yes	No	BEN	POLYGON ((2.6917016943562544 6.258817246928629, 2.6917016943562544 6.258817246928629, 2.6917016943562544 6.258817246928629, 2.6917016943562544 6.258817246928629))	Benin
17	5	2	212,316,594,243,95123	Poor	Yes	Yes	No	BTN	POLYGON ((91.69665652869668 27.771741848251665, 92.1037117858974 27.45261404063208, 92.1037117858974 27.45261404063208, 91.69665652869668 27.771741848251665))	Bhutan
18	5	0	125,037,504,800.05258	Middle-income	No	Yes	Yes	BOL	POLYGON ((-69.52967810736496 10.95173407502194, -68.78615759954948 11.036380303596, -68.78615759954948 11.036380303596, -69.52967810736496 10.95173407502194))	Bolivia
19	5	0	579,461.8	Middle-income	No	Yes	No	[NULL]	[NULL]	[NULL]
20	5	0	44,440,597,064,8422	Middle-income	No	Yes	No	BWA	POLYGON ((29.43218834810904 -22.09131275067588, 28.01723595552525 -22.82775359465908, 28.01723595552525 -22.82775359465908, 29.43218834810904 -22.09131275067588))	Botswana
21	5	0	297,175,320,150,12714	Rich	No	Yes	No	BRA	POLYGON ((-53.373661668498244 -33.76837778090764, -53.6505439921781 -33.202004082981, -53.6505439921781 -33.202004082981, -53.373661668498244 -33.76837778090764))	Brazil
22	5	0	496,856,211,592,4785	Middle-income	No	Yes	No	BRN	POLYGON ((115.45071048386981 5.447729803891534, 115.40570031134361 4.955227565933839, 115.40570031134361 4.955227565933839, 115.45071048386981 5.447729803891534))	Brunei
23	5	0	575,579,301,146,765	Middle-income	No	Yes	No	BGR	POLYGON ((22.657149692482994 44.23492300066128, 22.94483239105184743.82378530534713, 22.94483239105184743.82378530534713, 22.657149692482994 44.23492300066128))	Bulgaria
24	5	2	900,186,741,565,0015	Middle-income	Yes	Yes	Yes	BFA	POLYGON ((-5.404341599946974 10.370736802609146, -5.470564947929006 10.9512698427604, -5.470564947929006 10.9512698427604, -5.404341599946974 10.370736802609146))	Burkina Faso
25	5	2	5,399,900,398,579,621	Poor	Yes	Yes	No	BDI	POLYGON ((30.46967364576123 -2.41385475710134, 30.5276600000000026 -2.80761999999986, 30.5276600000000026 -2.80761999999986, 30.46967364576123 -2.41385475710134))	Burundi
26	5	1	1,086,871,101,222,8049	Middle-income	Yes	Yes	Yes	KHM	POLYGON ((102.5849324890267 12.186594956913282, 102.34809939983302 13.394247341358223, 102.34809939983302 13.394247341358223, 102.5849324890267 12.186594956913282))	Cambodia
27	5	3	688,078,854,410,264	Middle-income	Yes	Yes	No	CMR	POLYGON ((14.495787387762846 12.85939626713733, 14.893360000000003 12.21049999999982, 14.893360000000003 12.21049999999982, 14.495787387762846 12.85939626713733))	Cameroon

Refresh Save Cancel Export data 200 177 177 row(s) fetched - 0.005s (0.002s fetch), on 2024-06-07 at 00:24:56

PST en Writable Smart Insert 1:81:80 Sel: 0 | 0

Taper ici pour rechercher

ⓘ Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)

SQL | Commit | Rollback | Auto | .db | < N/A | Search |

Enter a part of object name | Tables |

Actualcountries | Columns | Keys | Foreign Keys | Indexes | References | Triggers | DataCleanAndPrepareForML | DimTableLanguages | DimTableOrganizations | GisDataWithImputation | WordGeoPandasBase | Views | Indexes | Sequences | Table Triggers | Data Types | DBeaver Sample Database (SC) | OneDrive | OneDrive 2 | Project - General |

--"Joining DataCleanAndPrepareForML with WordGeoPandasBase and Filtering for Non-Null Values"  
SELECT DataCleanAndPrepareForML.\*, WordGeoPandasBase.\*  
FROM DataCleanAndPrepareForML  
LEFT JOIN WordGeoPandasBase  
ON DataCleanAndPrepareForML.name = WordGeoPandasBase.name  
WHERE WordGeoPandasBase.pop\_est IS NOT NULL  
AND WordGeoPandasBase.continent IS NOT NULL  
AND WordGeoPandasBase.gdp\_md\_est IS NOT NULL  
ORDER BY DataCleanAndPrepareForML.name;

SELECT DataCleanAndPrepareForML.\*, WordGeoPandasBase.\* FROM DataCleanAndPrepareForML  
Enter a SQL expression to filter results (use Ctrl+Space)

	GDP_Category	Priority	Organization	Armed_Conflict	pop_est	continent	name	iso_a3	gdp_md_est	geometry
1	2.577	Middle-income	Yes	Yes	38,041,754	Asia	Afghanistan	AFG	19,291	POLYGON (((66.51860680528867 37.36278432875879, 67.07578209825962 37.35614390720929, 67.15279 15,279 POLYGON ((21.0200403174764 40.84272695572588, 20.99998861747224 40.5800397395398, 20.171,091 POLYGON ((-8.684399786809053 27.395744126896005, -8.665124477564191 27.589479071558227 88.815 MULTIPOLYGON (((12.995517205465177 -4.781103203961884, 12.63161176926579 -4.9912712540 898 MULTIPOLYGON (((-48.66061601418252 -78.04701873159873, -48.1513964503784 -78.047069600445,445 MULTIPOLYGON (((-68.63401022758323 -52.63637045887449, -68.25 -53.1, -67.75 -53.85, -66.45 13,672 POLYGON ((46.50571984231797 38.770605373686294, 46.14362308124882 38.7412048371224, 41.1396,567 MULTIPOLYGON (((147.68925947488418 -40.80825815202268, 148.289067824496 -40.87543751445,075 POLYGON (((16.979666782304037 48.123497015976305, 16.90375410326726 47.71486562762833, 48,047 MULTIPOLYGON (((46.404950799348825 41.860675157227305, 46.68607059101661 41.827137152 13,578 MULTIPOLYGON (((-78.98 26.790000000000003, -78.51 26.870000000000005, -77.850000000000000 302,571 POLYGON ((92.67272098182556 22.041238918541254, 92.65225711463799 21.324047552978485, 63,080 POLYGON ((28.17670942557794 56.16912995057879, 29.229513380660308 55.91834422466636, 533,097 POLYGON (((6.15665815595878 50.80372101501058, 6.04307357781111 50.128051662794235, 5.7 1,879 POLYGON (((-89.1430841050333 17.808318996649405, -89.1509093899553 17.95546763760042, 14,390 POLYGON ((2.6917016943562544 6.258817246928629, 1.8652405127123188 6.142157701029731, 2,530 POLYGON ((91.69665652869668 72.771741848251665, 92.10371178585974 27.45261404033208, 40,895 POLYGON (((-69.52967810736496 -10.95173407502194, -68.78615759954948 -11.0363803035986 18,340 POLYGON (((29.43218834810904 -22.091312758067588, 28.01723595552525 -22.82775359465908, 1,839,758 POLYGON (((-53.373661668498244 -33.768377780900764, -53.6505439927181 -33.2020040829818 13,469 POLYGON (((115.45071048386981 5.447729803891534, 115.40570031134361 4.955227565933839, 68,558 POLYGON (((22.65714969248299 44.23492300066128, 22.944832391051847 43.82378530534713, 15,990 POLYGON (((-5.404341599946974 10.370736802609146, -5.470564947929006 10.951269842976048 3,012 POLYGON (((30.469673645761223 -2.41385475710134, 30.52766000000026 -2.80761999999986, 27,089 POLYGON (((102.5849324890267 12.186594956913282, 102.3480993983302 13.39427431358223, 39,007 POLYGON (((14.495787378762846 12.85939626713733, 14.89336000000003 12.21904999999986 1,736,425 MULTIPOLYGON (((-122.84000000000003 49.000000000000114, -122.974210000000149.002537 11,314 POLYGON (((23.83766000000014 19.580470000000105, 23.886890000000108 15.610839999999996, 282,318 MULTIPOLYGON (((-68.63401022758323 -52.63637045887449, -68.63335000000001 -54.86949999 14,342,903 MULTIPOLYGON (((109.47520958866365 18.197700913968575, 108.65520796105616 18.50768199 323,615 POLYGON (((-66.87632585312258 1.253360500489336, -67.0650481838525 1.130112209473225, -12.267 POLYGON (((18.45306521980993 3.5043858911233485 18.393792351971143 2.90044342692872 1))

Refresh | Save | Cancel | Export data | Smart Insert | 6:44:313 | Sel: 0 | 0 |

Taper ici pour rechercher | FRA FR 00:27 07/06/2024 |

*(i)* Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)



# Humanitarian Response Dashboard

Countries

173

Gdp

86M

Pop

7.60bn

Continent

7

Area

214.05M

Organ

841

Sector

131

Conflicts

All

GDP

All

Countries

All

Continent

All

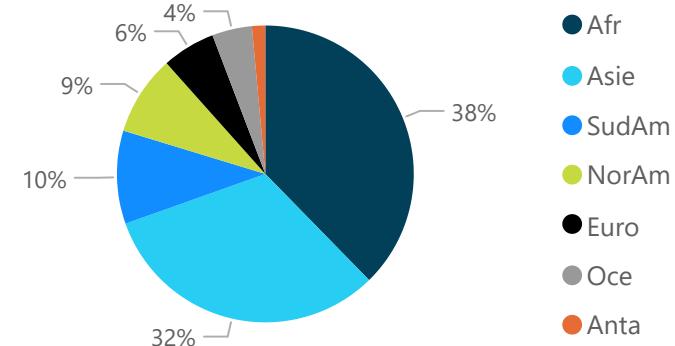
Sectors

All

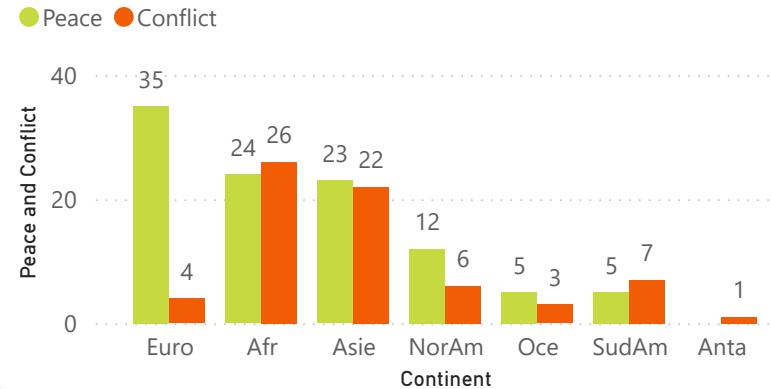
Priorities

All

## Ongoing Armed Conflicts by Continent

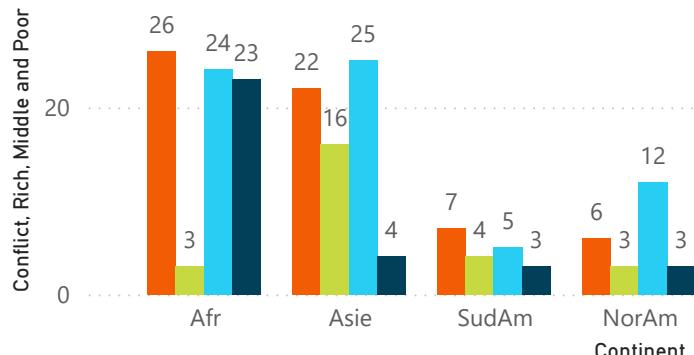


## Conflict and Peace by Continent



## Conflicts and Development by Continent

Conflict Rich Middle Poor



## Total Of Priorities



*(i)* Auto recovery contains some recovered files that haven't been opened.

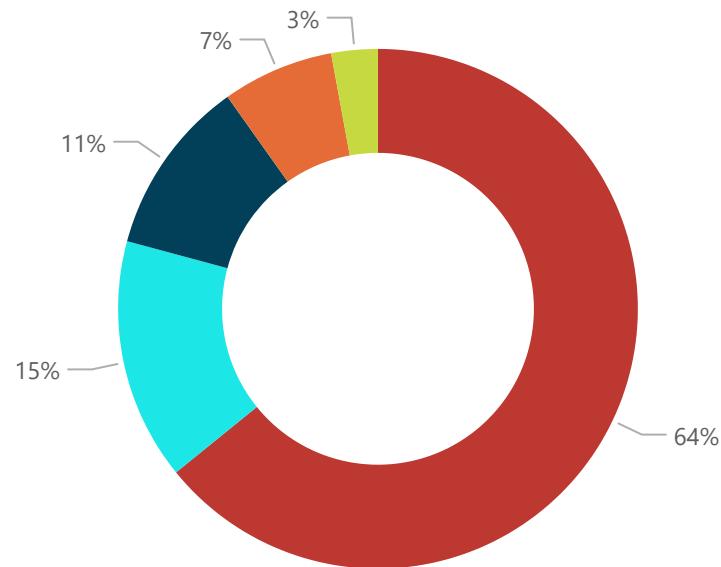
[View recovered files](#) X

All

All

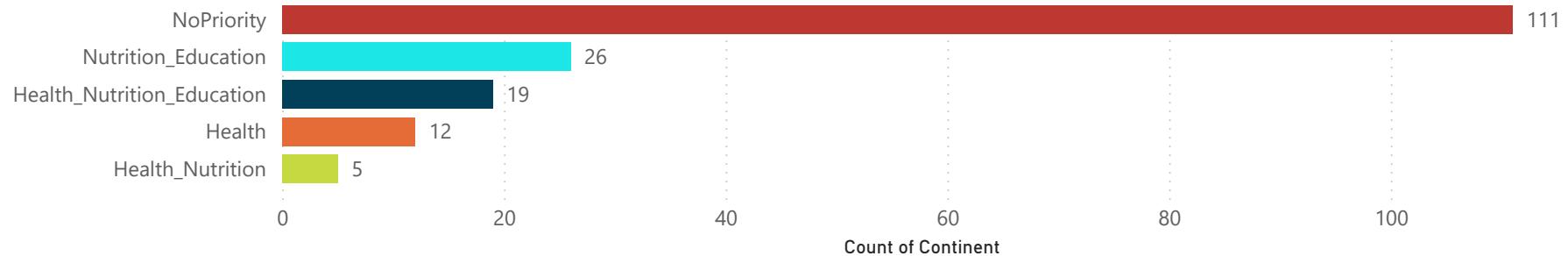
### Total areas of intervention in %

● NoPriority ● Nutrition\_Education ● Health\_Nutrition\_Education ● Health ● Health\_Nutrition



### Frequency Of Action Areas

Domains\_of\_Action



ⓘ Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)

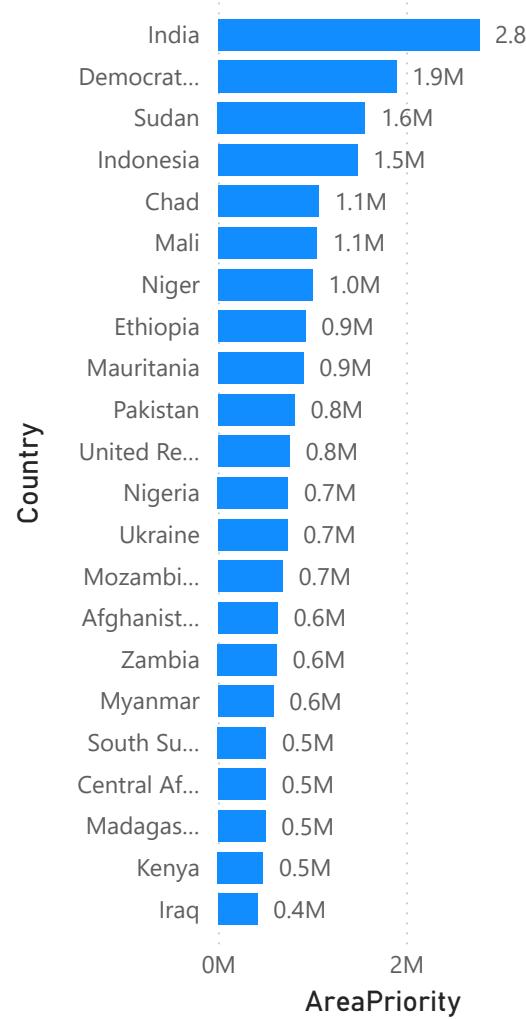


Continent

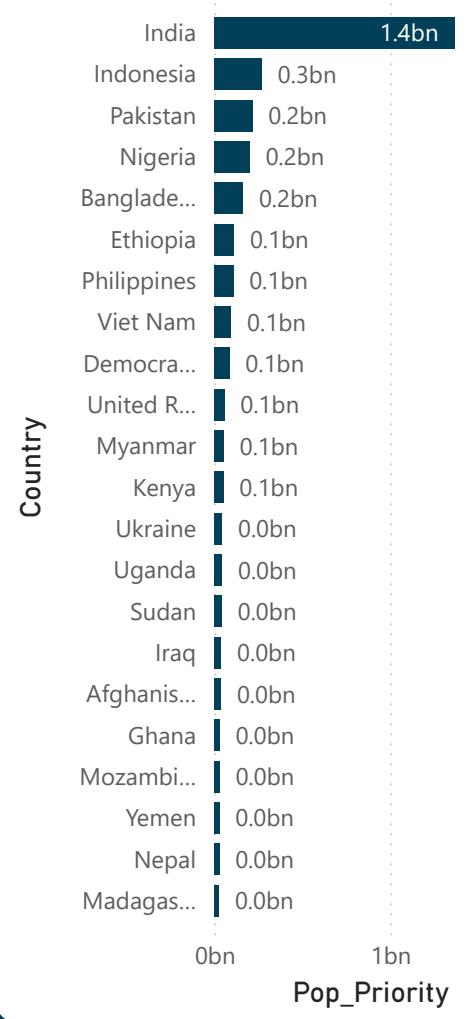
All



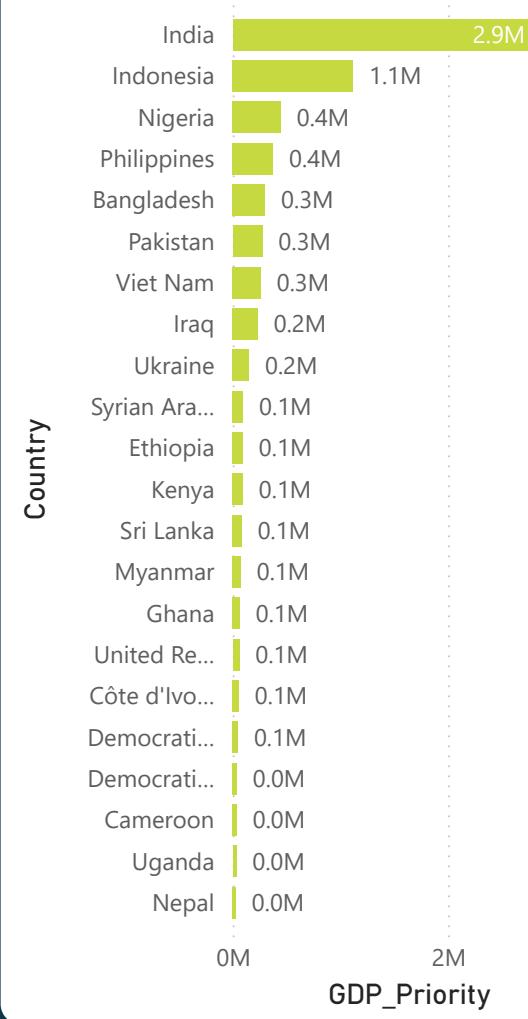
### Area Of Priority Countries



### Population Of Priority Countries



### GDP Of Priority Countries



*(i)* Auto recovery contains some recovered files that haven't been opened.

[View recovered files](#)



All

All

## Conflict Zone

