SQL PROJECT BY FRANCESCO ORMA

In this project I will interrogate two different datasets using pgAdmin 4, to analyse and compare countries on key issues such as economy, population and CO emissions.

Main dataset: **Global Country Information Dataset**

Secondary dataset: **Global Data on Sustainable Energy**

-------------------------------------> Script on Google Drive <--------------------------------------

The project shows 16 queries.

The first 10 queries relate to the first table, with draws in which all the main functions explained in the SQL course will appear progressively.

The remaining 6 queries relate to both tables with different uses of the JOIN function.

QUERY 1: List of the 20 countries with the highest population

```
SELECT country, population, urban_population
FROM global_country_2023
ORDER BY population DESC LIMIT 20;
```

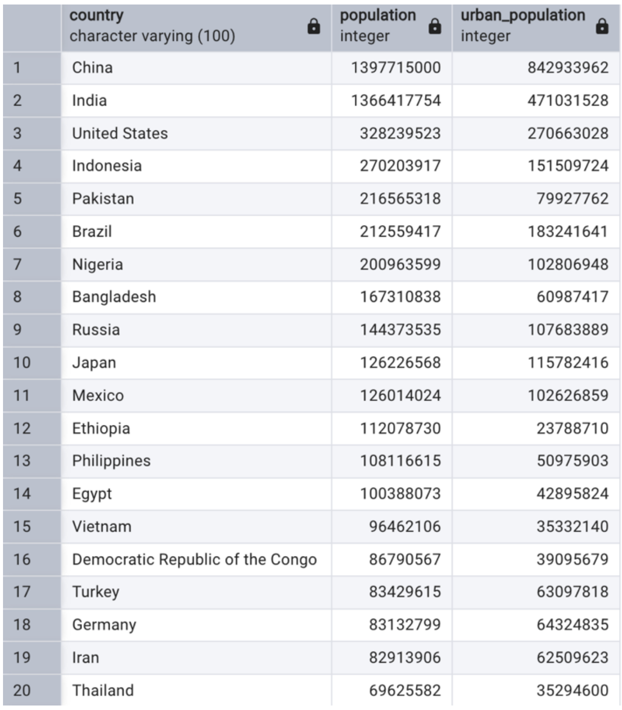
Comments:

SELECT to select the three columns of interest

FROM to select the table of interest

ORDER BY (DESC) to order a specific column of interest (from the largest to the smallest)

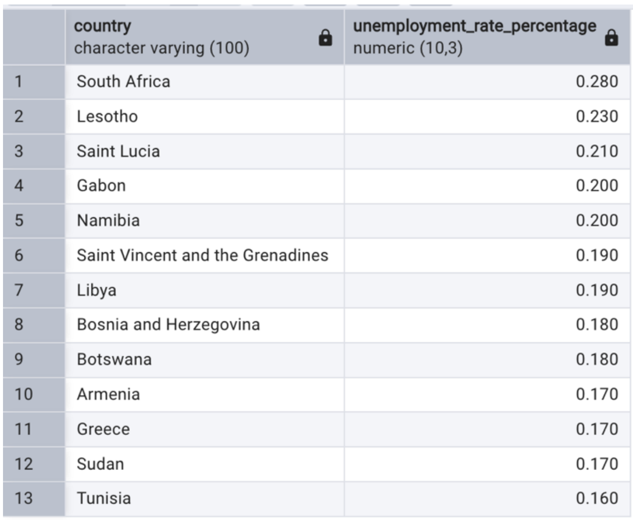
LIMIT (20) to obtain a limited amount of data in output (the first 20)



	country character varying (100)	population integer	urban_population integer
1	China	1397715000	842933962
2	India	1366417754	471031528
3	United States	328239523	270663028
4	Indonesia	270203917	151509724
5	Pakistan	216565318	79927762
6	Brazil	212559417	183241641
7	Nigeria	200963599	102806948
8	Bangladesh	167310838	60987417
9	Russia	144373535	107683889
10	Japan	126226568	115782416
11	Mexico	126014024	102626859
12	Ethiopia	112078730	23788710
13	Philippines	108116615	50975903
14	Egypt	100388073	42895824
15	Vietnam	96462106	35332140
16	Democratic Republic of the Congo	86790567	39095679
17	Turkey	83429615	63097818
18	Germany	83132799	64324835
19	Iran	82913906	62509623
20	Thailand	69625582	35294600

QUERY 2: Countries with an unemployment rate above 15%

```
SELECT country, unemployment_rate_percentage
FROM global_country_2023
WHERE unemployment_rate_percentage > 0.15
ORDER BY unemployment_rate_percentage DESC;
```



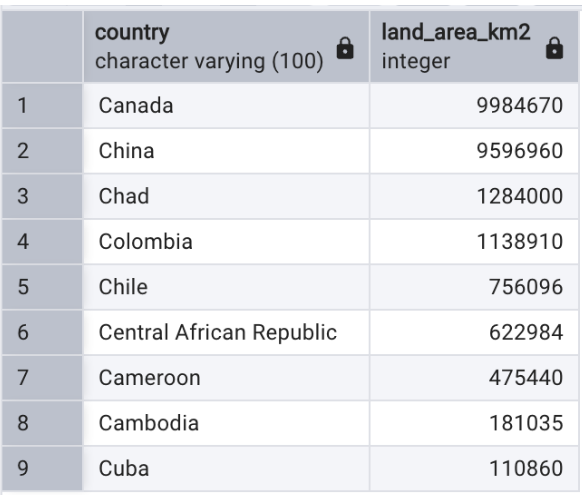
	country character varying (100)	unemployment_rate_percentage numeric (10,3)
1	South Africa	0.280
2	Lesotho	0.230
3	Saint Lucia	0.210
4	Gabon	0.200
5	Namibia	0.200
6	Saint Vincent and the Grenadines	0.190
7	Libya	0.190
8	Bosnia and Herzegovina	0.180
9	Botswana	0.180
10	Armenia	0.170
11	Greece	0.170
12	Sudan	0.170
13	Tunisia	0.160

Comments:

WHERE to impose a constraint (a value of interest that is greater than 0.15)

QUERY 3: List of countries starting with "C" and a land area greater than 100,000 km2

```
SELECT country, land_area_km2
FROM global_country_2023
WHERE land_area_km2 > 100000
AND country LIKE 'C%'
ORDER BY land_area_km2 DESC;
```



	country character varying (100)	land_area_km2 integer
1	Canada	9984670
2	China	9596960
3	Chad	1284000
4	Colombia	1138910
5	Chile	756096
6	Central African Republic	622984
7	Cameroon	475440
8	Cambodia	181035
9	Cuba	110860

Comments:

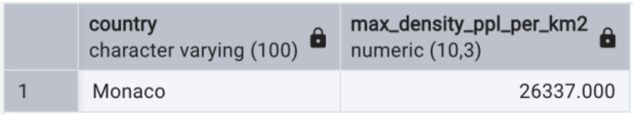
AND to impose two contemporary constraints

LIKE to check whether a specific character string matches a specified pattern

% as a wildcard character set after 'C' to indicate that C must be the first letter followed by any other string

QUERY 4: Country with the highest population density

```
SELECT country, density AS max_density_ppl_per_km2
FROM global_country_2023
WHERE density = (SELECT MAX(density) FROM
global_country_2023);
```



	country character varying (100)	max_density_ppl_per_km2 numeric (10,3)
1	Monaco	26337.000

Comments:

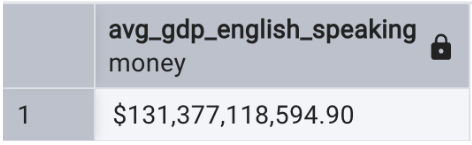
AS to rename a column name in output

The subquery (SELECT ...) to calculate a value derived (the maximum density)

MAX to calculate the maximum between the values of interest

QUERY 5: Average GDP grouped by English as official language

```
SELECT CAST(AVG(gdp::NUMERIC) AS MONEY) AS avg_gdp_english_speaking
FROM Global_Country_2023
WHERE official_language = 'English' AND gdp IS NOT NULL;
```



	avg_gdp_english_speaking money
1	$131,377,118,594.90

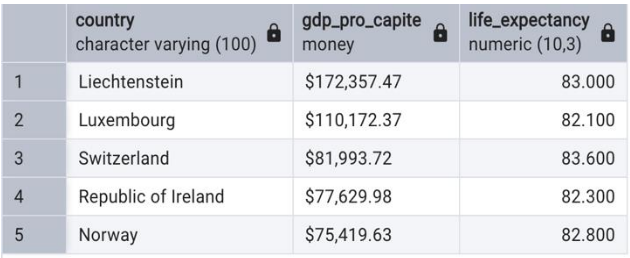
Comments:

CAST in combination with ::NUMERIC AS MONEY to transform the GDP value from MONEY to NUMERIC, allowing a calculation (the average).
Then, again from NUMERIC to MONEY to return the value in $

IS NOT NULL to ignore the null values in the column of interest

QUERY 6: Top 5 countries with the best ratio of GDP per capita and life expectancy

```
SELECT country, gdp / population AS gdp_pro_capite, life_expectancy
FROM global_country_2023
WHERE life_expectancy IS NOT NULL AND gdp / population IS NOT NULL
ORDER BY gdp_pro_capite DESC LIMIT 5;
```



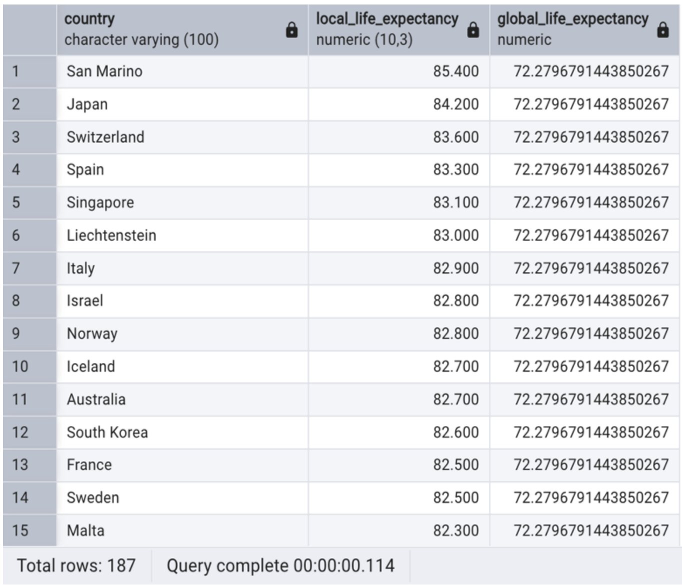
	country character varying (100)	gdp_pro_capite money	life_expectancy numeric (10,3)
1	Liechtenstein	$172,357.47	83.000
2	Luxembourg	$110,172.37	82.100
3	Switzerland	$81,993.72	83.600
4	Republic of Ireland	$77,629.98	82.300
5	Norway	$75,419.63	82.800

Comments:

Introduction of calculations within the query

QUERY 7: Average global life expectancy compared to local life expectancy

```
SELECT country, life_expectancy AS local_life_expectancy,
          (SELECT AVG(life_expectancy)
           FROM global_country_2023) AS global_life_expectancy
FROM global_country_2023
WHERE life_expectancy IS NOT NULL
ORDER BY life_expectancy DESC;
```



	country character varying (100)	local_life_expectancy numeric (10,3)	global_life_expectancy numeric
1	San Marino	85.400	72.2796791443850267
2	Japan	84.200	72.2796791443850267
3	Switzerland	83.600	72.2796791443850267
4	Spain	83.300	72.2796791443850267
5	Singapore	83.100	72.2796791443850267
6	Liechtenstein	83.000	72.2796791443850267
7	Italy	82.900	72.2796791443850267
8	Israel	82.800	72.2796791443850267
9	Norway	82.800	72.2796791443850267
10	Iceland	82.700	72.2796791443850267
11	Australia	82.700	72.2796791443850267
12	South Korea	82.600	72.2796791443850267
13	France	82.500	72.2796791443850267
14	Sweden	82.500	72.2796791443850267
15	Malta	82.300	72.2796791443850267

Total rows: 187 Query complete 00:00:00.114

Comments:

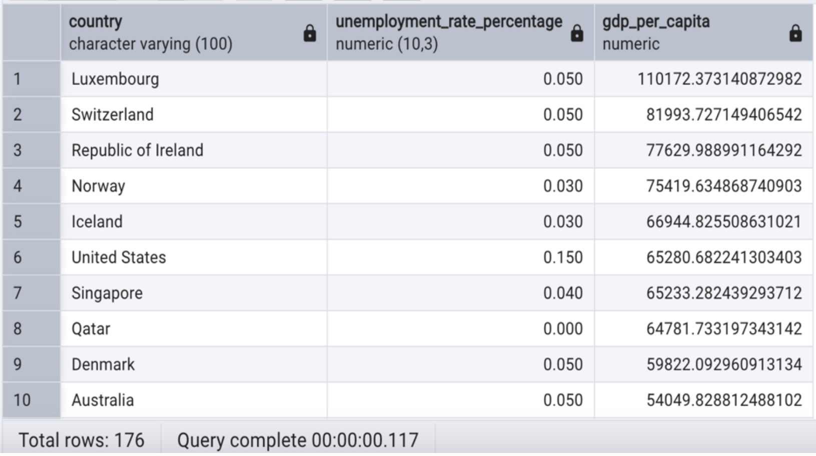
Subquery more complete than the previous one

QUERY 8: Comparison of unemployment rate and per capita GDP

```
SELECT country, unemployment_rate_percentage,
                              (gdp::NUMERIC / NULLIF(population, 0)) AS gdp_per_capita
FROM global_country_2023
WHERE gdp IS NOT NULL AND population IS NOT NULL AND unemployment_rate_percentage IS NOT NULL
ORDER BY gdp_per_capita DESC;
```

Comments:

NULLIF to return NULL if population is 0, otherwise returns population

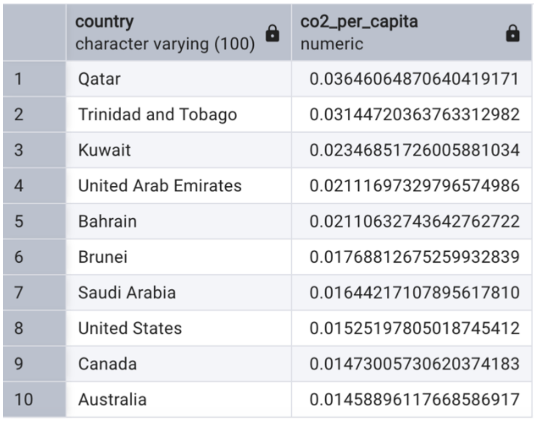


	country character varying (100)	unemployment_rate_percentage numeric (10,3)	gdp_per_capita numeric
1	Luxembourg	0.050	110172.373140872982
2	Switzerland	0.050	81993.727149406542
3	Republic of Ireland	0.050	77629.988991164292
4	Norway	0.030	75419.634868740903
5	Iceland	0.030	66944.825508631021
6	United States	0.150	65280.682241303403
7	Singapore	0.040	65233.282439293712
8	Qatar	0.000	64781.733197343142
9	Denmark	0.050	59822.092960913134
10	Australia	0.050	54049.828812488102

Total rows: 176 Query complete 00:00:00.117

QUERY 9: Which 10 countries have the highest environmental impact in terms of CO2 emissions per capita?

```sql
SELECT country, co2_emissions ::NUMERIC / population ::NUMERIC AS co2_per_capita
FROM global_country_2023
WHERE co2_emissions IS NOT NULL AND population IS NOT NULL
ORDER BY co2_per_capita DESC
LIMIT 10;
```



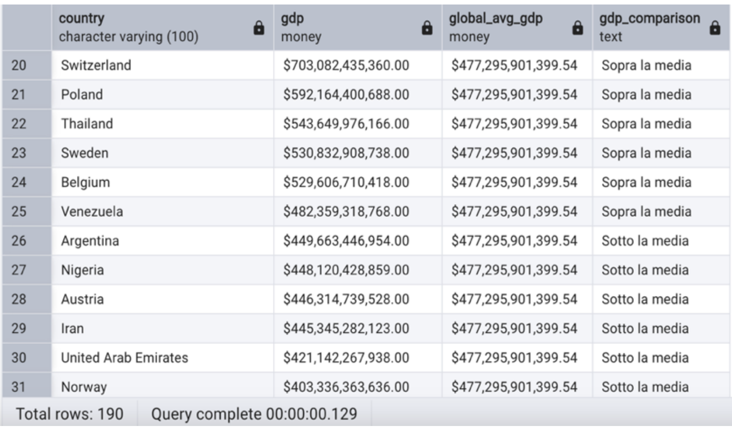
	country character varying (100)	co2_per_capita numeric
1	Qatar	0.03646064870640419171
2	Trinidad and Tobago	0.03144720363763312982
3	Kuwait	0.02346851726005881034
4	United Arab Emirates	0.02111697329796574986
5	Bahrain	0.02110632743642762722
6	Brunei	0.01768812675259932839
7	Saudi Arabia	0.01644217107895617810
8	United States	0.01525197805018745412
9	Canada	0.01473005730620374183
10	Australia	0.01458896117668586917

Comments:

I had to convert the data to NUMERIC as both are INTEGER and the ratio returned 0 as result

QUERY 10: Global average of GDP and comparison of countries above/below this average

```sql
WITH gdp_average AS (
  SELECT CAST(AVG(gdp::NUMERIC) AS MONEY) AS global_avg_ gdp
  FROM global_country_2023
  WHERE gdp IS NOT NULL
)
SELECT country, gdp,
  (SELECT global_avg_ gdp FROM gdp _Average) AS global_avg_ gdp,
        CASE
    WHEN CAST(AVG(gdp::NUMERIC) AS MONEY) >= (SELECT global_avg_ gdp FROM gdp _Average)
    THEN 'above average’
    ELSE 'below average’
  END AS gdp _comparison
FROM global_country_2023
WHERE gdp IS NOT NULL AND urban_population IS NOT NULL
GROUP BY country, gdp
ORDER BY gdp DESC;
```



	country character varying (100)	gdp money	global_avg_gdp money	gdp_comparison text
20	Switzerland	$703,082,435,360.00	$477,295,901,399.54	Sopra la media
21	Poland	$592,164,400,688.00	$477,295,901,399.54	Sopra la media
22	Thailand	$543,649,976,166.00	$477,295,901,399.54	Sopra la media
23	Sweden	$530,832,908,738.00	$477,295,901,399.54	Sopra la media
24	Belgium	$529,606,710,418.00	$477,295,901,399.54	Sopra la media
25	Venezuela	$482,359,318,768.00	$477,295,901,399.54	Sopra la media
26	Argentina	$449,663,446,954.00	$477,295,901,399.54	Sotto la media
27	Nigeria	$448,120,428,859.00	$477,295,901,399.54	Sotto la media
28	Austria	$446,314,739,528.00	$477,295,901,399.54	Sotto la media
29	Iran	$445,345,282,123.00	$477,295,901,399.54	Sotto la media
30	United Arab Emirates	$421,142,267,938.00	$477,295,901,399.54	Sotto la media
31	Norway	$403,336,363,636.00	$477,295,901,399.54	Sotto la media

Total rows: 190 Query complete 00:00:00.129

Comments:

CTE to calculate the global average of GDP.

CASE, THEN, ELSE and END to compare each country with the global average:
If it is >= then 'above average'
If < then 'below average'

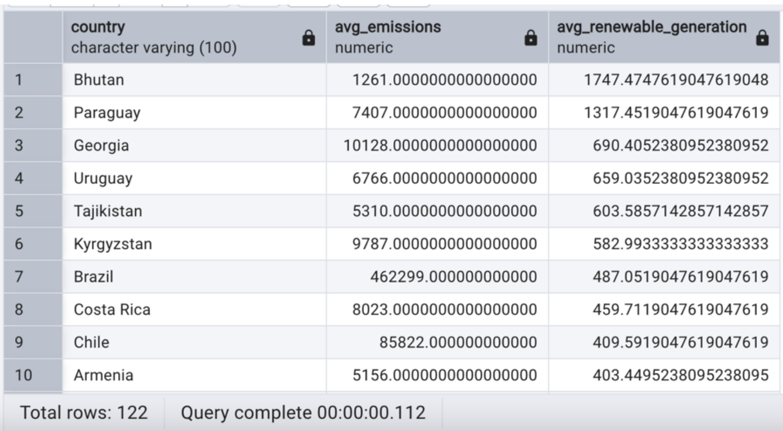
QUERY 11: Average CO2 emissions and renewable energy generation capacity for each country

```
SELECT global_country_2023.country,
        AVG(global_country_2023.co2_emissions) AS avg_emissions,
        AVG(global_sustainable_energy.renewable_electricity_generating_capacity_per_capita) AS
            avg_renewable_generation
FROM global_country_2023
INNER JOIN global_sustainable_energy
ON global_country_2023.country = global_sustainable_energy.country
WHERE global_sustainable_energy.renewable_electricity_generating_capacity_per_capita IS NOT NULL
GROUP BY global_country_2023.country
ORDER BY avg_renewable_generation DESC;
```

Comments:

INNER JOIN to return only the common values in both tables

GROUP BY to group data according to a particular value (in this case the countries)



	country character varying (100)	avg_emissions numeric	avg_renewable_generation numeric
1	Bhutan	1261.0000000000000000	1747.4747619047619048
2	Paraguay	7407.0000000000000000	1317.4519047619047619
3	Georgia	10128.0000000000000000	690.4052380952380952
4	Uruguay	6766.0000000000000000	659.0352380952380952
5	Tajikistan	5310.0000000000000000	603.5857142857142857
6	Kyrgyzstan	9787.0000000000000000	582.9933333333333333
7	Brazil	462299.000000000000	487.0519047619047619
8	Costa Rica	8023.0000000000000000	459.7119047619047619
9	Chile	85822.000000000000	409.5919047619047619
10	Armenia	5156.0000000000000000	403.4495238095238095

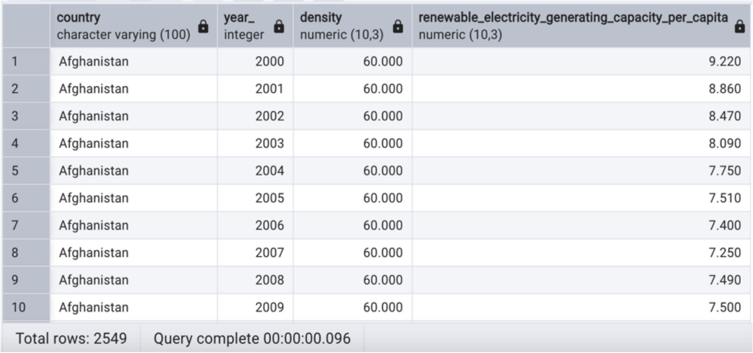
Total rows: 122 Query complete 00:00:00.112

QUERY 12: Combined data on population density and renewable energy generating capacity, including countries without generation capacity.

```
SELECT global_country_2023.country,
       global_sustainable_energy.year_,
       global_country_2023.density,
       global_sustainable_energy.renewable_electricity_generating_capacity_per_capita
FROM global_country_2023
LEFT JOIN global_sustainable_energy
ON global_country_2023.country = global_sustainable_energy.country
WHERE global_sustainable_energy.renewable_electricity_generating_capacity_per_capita IS NOT NULL;
```

Comments:

LEFT JOIN to return all the values of the left table (global_country_2023) and only the common ones in the right



	country character varying (100)	year_ integer	density numeric (10,3)	renewable_electricity_generating_capacity_per_capita numeric (10,3)
1	Afghanistan	2000	60.000	9.220
2	Afghanistan	2001	60.000	8.860
3	Afghanistan	2002	60.000	8.470
4	Afghanistan	2003	60.000	8.090
5	Afghanistan	2004	60.000	7.750
6	Afghanistan	2005	60.000	7.510
7	Afghanistan	2006	60.000	7.400
8	Afghanistan	2007	60.000	7.250
9	Afghanistan	2008	60.000	7.490
10	Afghanistan	2009	60.000	7.500

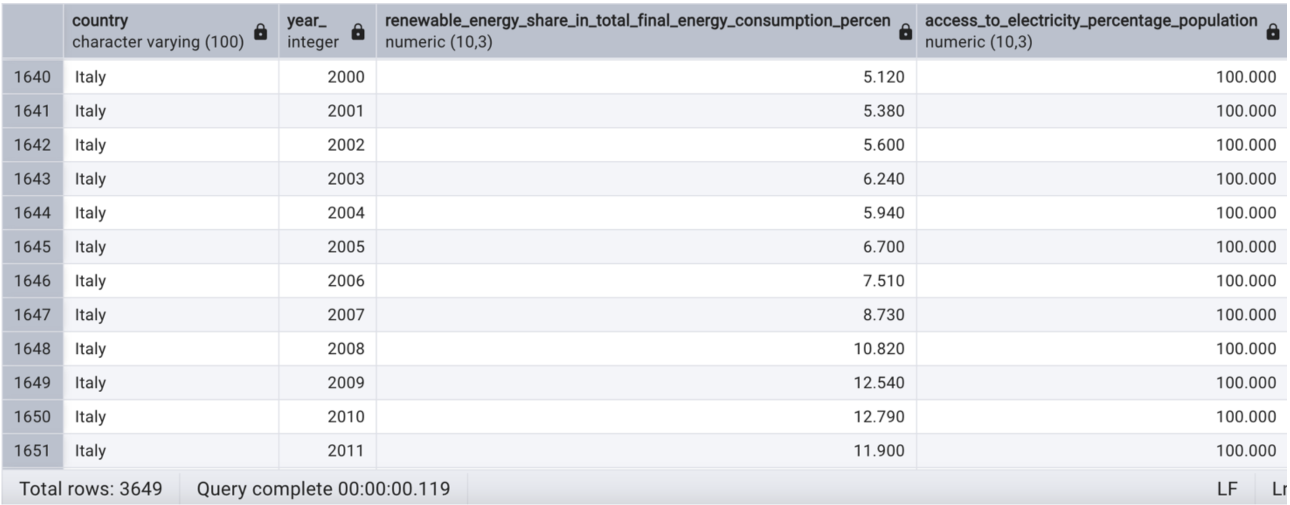
Total rows: 2549 Query complete 00:00:00.096

QUERY 13: Countries with information on renewable energies, including all energy data even if they do not correspond to a country.

```
SELECT global_country_2023.country,
            global_sustainable_energy.year_,
            global_sustainable_energy.renewable_energy_share_in_total_final_energy_consumption_percen,
            global_sustainable_energy.access_to_electricity_percentage_population
FROM global_country_2023
RIGHT JOIN global_sustainable_energy
ON global_country_2023.country = global_sustainable_energy.country;
```

Comments:

RIGHT JOIN to return all the values of the right table (global_country_2023) and the common ones in the left



	country character varying (100)	year_ integer	renewable_energy_share_in_total_final_energy_consumption_percen numeric (10,3)	access_to_electricity_percentage_population numeric (10,3)
1640	Italy	2000	5.120	100.000
1641	Italy	2001	5.380	100.000
1642	Italy	2002	5.600	100.000
1643	Italy	2003	6.240	100.000
1644	Italy	2004	5.940	100.000
1645	Italy	2005	6.700	100.000
1646	Italy	2006	7.510	100.000
1647	Italy	2007	8.730	100.000
1648	Italy	2008	10.820	100.000
1649	Italy	2009	12.540	100.000
1650	Italy	2010	12.790	100.000
1651	Italy	2011	11.900	100.000

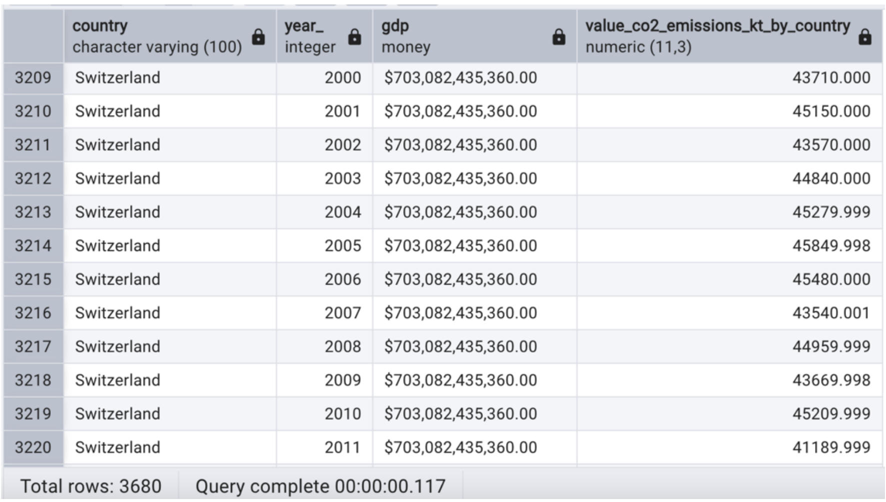
Total rows: 3649 Query complete 00:00:00.119 LF Lr

QUERY 14: Combined tables that include all countries and energy information, even when there is no match

```
SELECT global_country_2023.country,
            global_sustainable_energy.year_,
            global_country_2023.gdp,
            global_sustainable_energy.Value_co2_emissions_kt_by_country
FROM global_country_2023
FULL JOIN global_sustainable_energy
ON global_country_2023.country = global_sustainable_energy.country;
```

Comments:

FULL JOIN to return all rows from both tables without constraints on the matches



	country character varying (100)	year_ integer	gdp money	value_co2_emissions_kt_by_country numeric (11,3)
3209	Switzerland	2000	$703,082,435,360.00	43710.000
3210	Switzerland	2001	$703,082,435,360.00	45150.000
3211	Switzerland	2002	$703,082,435,360.00	43570.000
3212	Switzerland	2003	$703,082,435,360.00	44840.000
3213	Switzerland	2004	$703,082,435,360.00	45279.999
3214	Switzerland	2005	$703,082,435,360.00	45849.998
3215	Switzerland	2006	$703,082,435,360.00	45480.000
3216	Switzerland	2007	$703,082,435,360.00	43540.001
3217	Switzerland	2008	$703,082,435,360.00	44959.999
3218	Switzerland	2009	$703,082,435,360.00	43669.998
3219	Switzerland	2010	$703,082,435,360.00	45209.999
3220	Switzerland	2011	$703,082,435,360.00	41189.999

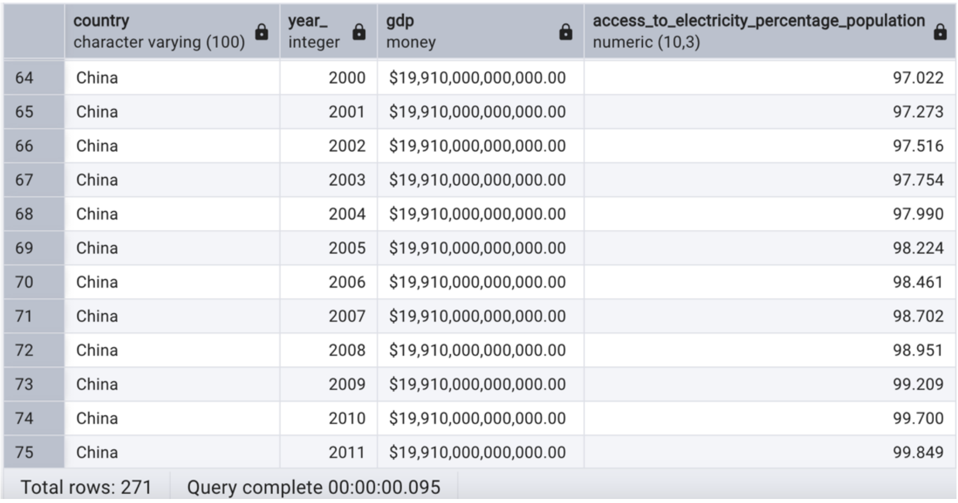
Total rows: 3680   Query complete 00:00:00.117

QUERY 15: Countries with electricity access above 90% and a GDP above $1,000,000,000.

```
SELECT global_country_2023.country,
            global_sustainable_energy.year_,
            global_country_2023.gdp,
            global_sustainable_energy.access_to_electricity_percentage_population
FROM global_country_2023
INNER JOIN global_sustainable_energy
ON global_country_2023.country = global_sustainable_energy.country
WHERE global_sustainable_energy.access_to_electricity_percentage_population > 90
AND global_country_2023.gdp::NUMERIC > 1000000000000;
```

Comments:

INNER JOIN in combination with WHERE and AND for the imposition of constraints (access_to_electricity_pourcentage_population > 90 and gdp > $1,000,000)



	country character varying (100)	year_ integer	gdp money	access_to_electricity_percentage_population numeric (10,3)
64	China	2000	$19,910,000,000,000.00	97.022
65	China	2001	$19,910,000,000,000.00	97.273
66	China	2002	$19,910,000,000,000.00	97.516
67	China	2003	$19,910,000,000,000.00	97.754
68	China	2004	$19,910,000,000,000.00	97.990
69	China	2005	$19,910,000,000,000.00	98.224
70	China	2006	$19,910,000,000,000.00	98.461
71	China	2007	$19,910,000,000,000.00	98.702
72	China	2008	$19,910,000,000,000.00	98.951
73	China	2009	$19,910,000,000,000.00	99.209
74	China	2010	$19,910,000,000,000.00	99.700
75	China	2011	$19,910,000,000,000.00	99.849

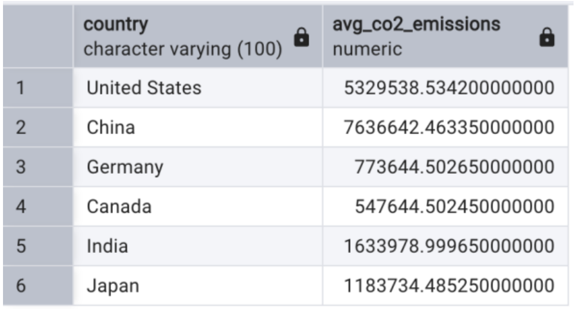
Total rows: 271   Query complete 00:00:00.095

QUERY 16: Average CO2 emissions per country filtering only those with an average emission above 500,000 kt

```
SELECT global_country_2023.country,
             AVG(global_sustainable_energy.value_co2_emissions_kt_by_country) AS avg_co2_emissions
FROM global_country_2023
INNER JOIN global_sustainable_energy
ON global_country_2023.country = global_sustainable_energy.country
GROUP BY global_country_2023.country
HAVING AVG(global_sustainable_energy.value_co2_emissions_kt_by_country) > 500000;
```

Comments:

HAVING to indicate only the groups that contain a clause condition (value_co2_emissions_kt_by_country greater than 500,000)



	country character varying (100)	avg_co2_emissions numeric
1	United States	5329538.534200000000
2	China	7636642.463350000000
3	Germany	773644.502650000000
4	Canada	547644.502450000000
5	India	1633978.999650000000
6	Japan	1183734.485250000000