

## Bases de Datos

### Final Exam: Practice (40%)

Juan Miguel Manjarrez Zuluaga – 2415330

### Final Project Report – Geopolitical World Database (Docker, PostgreSQL, pgAdmin)

<https://github.com/mzmiguelwd/docker-postgresql-pgadmin-geo-world-database>

---

## Desarrollo

### Modelo Relacional

1. **airport** (iatacode\*, name, country, city, province, island, latitude, longitude, elevation, gmtoffset)
2. **borders** (country1\*, country2\*, length)
3. **city** (name\*, country\*, province\*, population, latitude, longitude, elevation)
4. **citylocalname** (city\*, country\*, province\*, localname)
5. **citiothername** (city\*, country\*, province\*, othername)
6. **citypops** (city\*, province\*, country\*, year\*, population)
7. **continent** (name\*, area)
8. **country** (name, code\*, capital, province, area, population)
9. **countrylocalname** (country\*, localname)
10. **countryothername** (country\*, othername\*)
11. **countrypops** (country\*, year\*, population)
12. **desert** (name\*, area, coordinates)
13. **economy** (country\*, gdp, agriculture, service, industry, inflation, unemployment)
14. **encompasses** (country\*, continent\*, percentage)
15. **ethnicgroup** (country\*, name\*, percentage)
16. **geo\_desert** (desert\*, country\*, province\*)
17. **geo\_estuary** (river\*, country\*, province\*)
18. **geo\_island** (island\*, country\*, province\*)
19. **geo\_lake** (lake\*, country\*, province\*)
20. **geo\_mountain** (mountain\*, country\*, province\*)
21. **geo\_river** (river\*, country\*, province\*)
22. **geo\_sea** (sea\*, country\*, province\*)
23. **geo\_source** (river\*, country\*, province\*)
24. **island** (name\*, islands, area, elevation, type, coordinates)
25. **islandin** (island, sea, lake, river)
26. **ismember** (country\*, organization\*, type)
27. **lake** (name\*, river, area, elevation, depth, height, type, coordinates)
28. **lakeonisland** (lake\*, island\*)
29. **language** (country\*, name\*, percentage)
30. **located** (city, province, country, river, lake, sea)
31. **locatedon** (city\*, province\*, country\*, island\*)

32. **mergeswith** (sea1\*, sea2\*)
  33. **mountain** (name\*, mountains, elevation, type, coordinates)
  34. **mountainonisland** (mountain\*, island\*)
  35. **organization** (abbreviation\*, name, city, country, province, established)
  36. **politics** (country\*, independence, wasdependent, dependent, government)
  37. **population** (country\*, population\_growth, infant\_mortality)
  38. **province** (name\*, country\*, population, area, capital, capprov)
  39. **provincelocalname** (province\*, country\* localname)
  40. **provinceothername** (province\*, country\*, othername\*)
  41. **prov pops** (province\*, country\*, year\*, population)
  42. **religion** (country\*, name\*, percentage)
  43. **river** (name\*, river, lake, sea, length, area, source, mountains, sourceelevation, estuary, estuaryelevation)
  44. **riveronisland** (river\*, island\*)
  45. **riverthrough** (river\*, lake\*)
  46. **sea** (name\*, area, depth)
  47. **sublanguage** (sublang\*, suplang)
- 

## Consultas requeridas

En mayúscula, sin renombres, cree las consultas en SQL para obtener:

1. ¿Cuál es el nombre y la superficie terrestre de cada continente?
  2. ¿Cuál es la superficie y la profundidad del lago Malawi?
  3. ¿Cuáles son los nombres y la superficie de todos los lagos en orden decreciente de profundidad?
  4. ¿Cuál es el nombre de los países donde la agricultura representa al menos el 50% del producto interno bruto? -- use INNER JOIN ... ON
  5. ¿Cómo se llama el río más largo? -- use una subconsulta
- 

## Recursos del diccionario de datos para dar respuesta a las consultas requeridas

**Continent:** Information about continents.

- name: name of the continent
- area: total area of the continent

**Lake:** information about lakes.

- name: the name of the lake
- area: the total area of the lake
- depth: the depth of the lake

**Economy:** economical information about the countries.

- country: the country code
- agriculture: percentage of agriculture of the GDP

**Country:** the countries (and similar areas) of the world with some data.

- name: the country name
- code: the car code

**River:** information about rivers.

- name: the name of the river
- length: the length of the river

---

## Consultas propuestas

### 1. ¿Cuál es el nombre y la superficie terrestre de cada continente?

Se realiza una consulta directa sobre la tabla `CONTINENT`, seleccionando las columnas `NAME` y `AREA`, que representan el nombre y la superficie de cada continente.

```
SELECT NAME, AREA  
FROM CONTINENT;
```

Query

Query History

1

SELECT NAME, AREA

2

FROM CONTINENT;

Data Output

Messages

Notifications

≡+

▼

▼

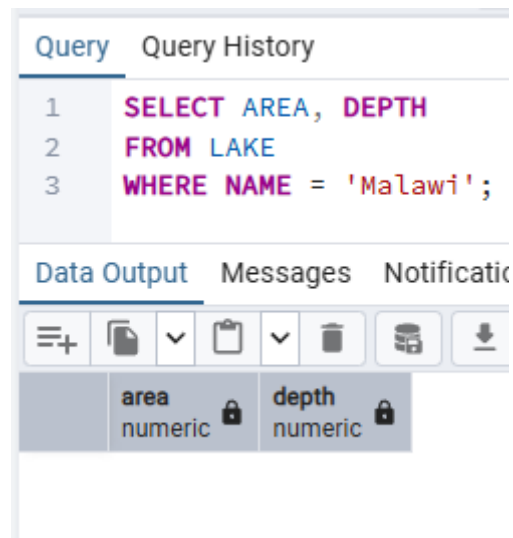
SQL

	name [PK] character varying (20)	area numeric (10)
1	Europe	10523000
2	Asia	44614500
3	Australia/Oceania	9000000
4	Africa	30221500
5	North America	24709000
6	South America	17840000

## 2. ¿Cuál es la superficie y la profundidad del lago Malawi?

En esta consulta se seleccionan los campos `AREA` y `DEPTH` de la tabla `LAKE`, filtrando únicamente por el nombre del lago que coincida exactamente con `'Malawi'`. Se usa la cláusula `WHERE` para identificar el lago específico.

```
SELECT AREA, DEPTH
FROM LAKE
WHERE NAME = 'Malawi';
```



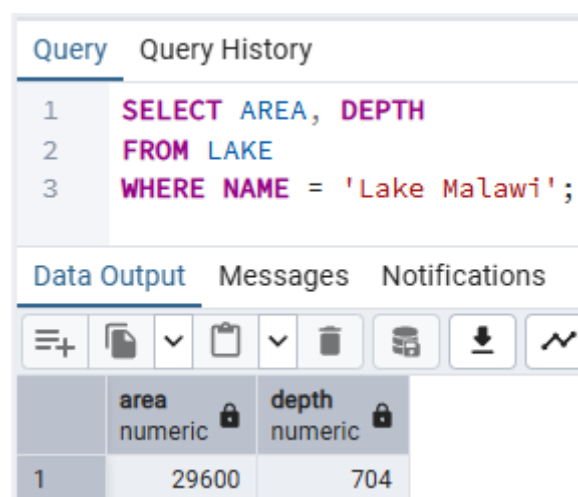
The screenshot shows a SQL query editor with two tabs: 'Query' and 'Query History'. The 'Query' tab is active, displaying the following SQL query:

```
1 SELECT AREA, DEPTH
2 FROM LAKE
3 WHERE NAME = 'Malawi';
```

Below the query, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with two columns: 'area' and 'depth'. Both columns are of type 'numeric' and have a lock icon. The table is currently empty.

	area numeric	depth numeric
--	-----------------	------------------

**Observación:** En la base de datos, el lago está registrado como `'Lake Malawi'` y no simplemente `'Malawi'`, por lo que es fundamental respetar el valor exacto al usar la cláusula `WHERE`.



The screenshot shows a SQL query editor with two tabs: 'Query' and 'Query History'. The 'Query' tab is active, displaying the following SQL query:

```
1 SELECT AREA, DEPTH
2 FROM LAKE
3 WHERE NAME = 'Lake Malawi';
```

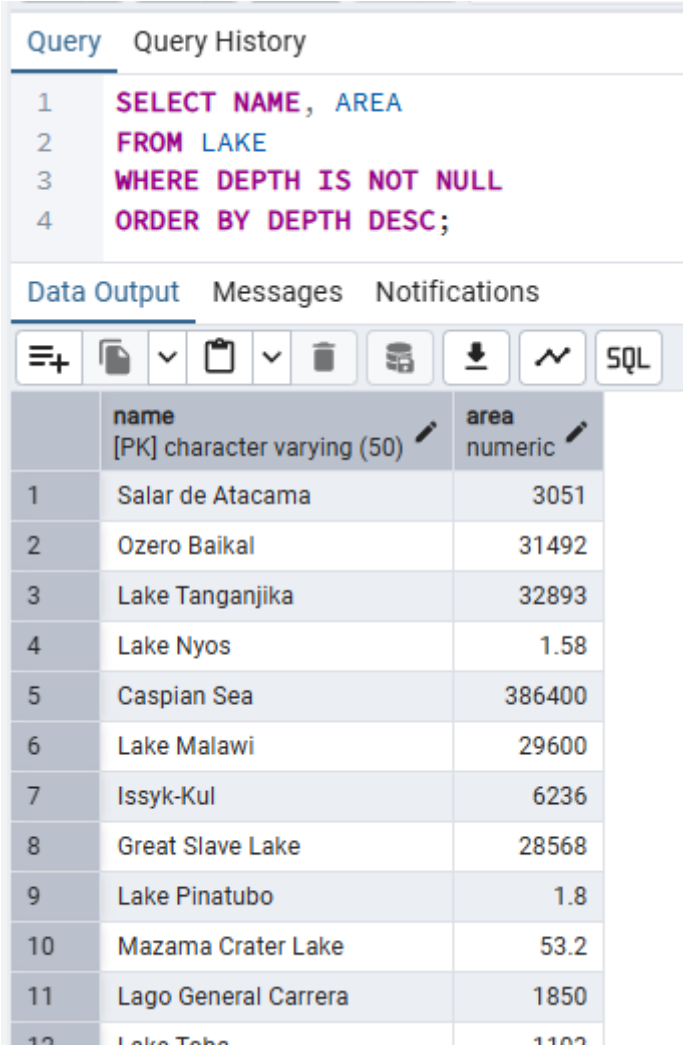
Below the query, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with two columns: 'area' and 'depth'. Both columns are of type 'numeric' and have a lock icon. The table contains one row of data.

	area numeric	depth numeric
1	29600	704

### 3. ¿Cuáles son los nombres y la superficie de todos los lagos en orden decreciente de profundidad?

Aquí se extraen los nombres y superficies de todos los registros en la tabla LAKE, y se ordenan por la columna DEPTH en orden descendente usando ORDER BY ... DESC, para mostrar primero los lagos más profundos.

```
SELECT NAME, AREA
FROM LAKE
WHERE DEPTH IS NOT NULL
ORDER BY DEPTH DESC;
```



The screenshot shows a database query interface. At the top, there are tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying the following SQL query:

```
1 SELECT NAME, AREA
2 FROM LAKE
3 WHERE DEPTH IS NOT NULL
4 ORDER BY DEPTH DESC;
```

Below the query, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table of results. The table has two columns: 'name' (character varying (50)) and 'area' (numeric). The results are ordered by depth in descending order.

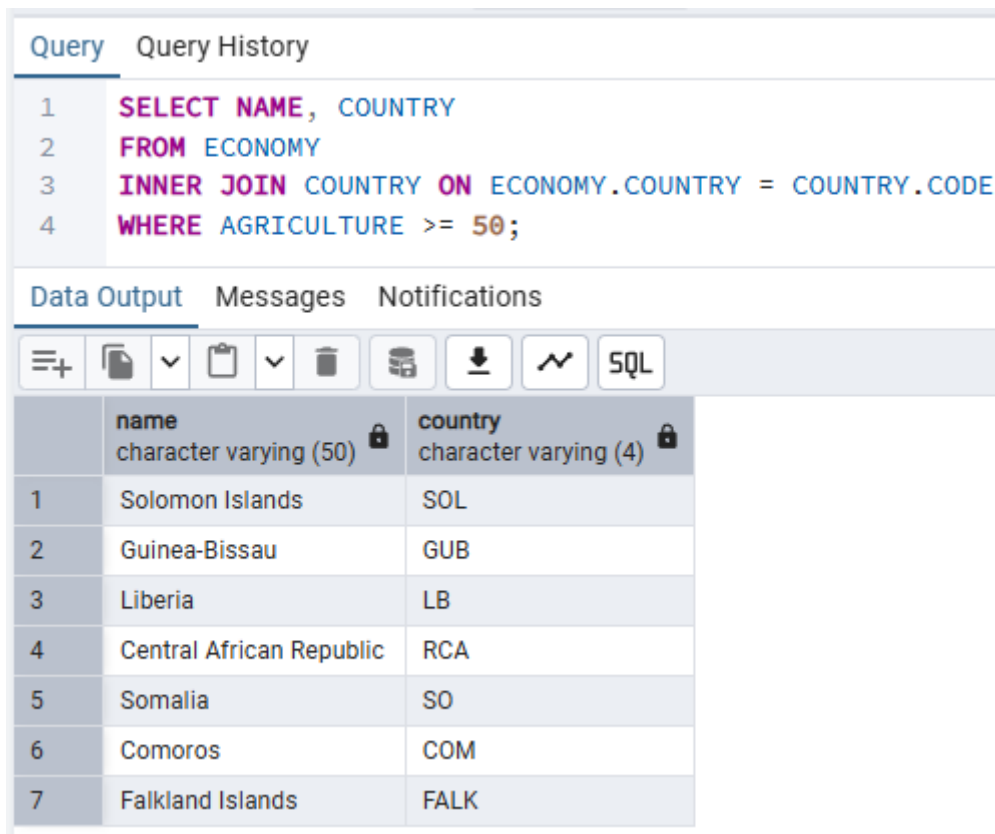
	name [PK] character varying (50)	area numeric
1	Salar de Atacama	3051
2	Ozero Baikal	31492
3	Lake Tanganjika	32893
4	Lake Nyos	1.58
5	Caspian Sea	386400
6	Lake Malawi	29600
7	Issyk-Kul	6236
8	Great Slave Lake	28568
9	Lake Pinatubo	1.8
10	Mazama Crater Lake	53.2
11	Lago General Carrera	1850
12	Lake Toba	1100

**Observación:** En la base de datos existen registros de lagos con DEPTH igual a NULL (es decir, sin información registrada). Para evitar que estos aparezcan al inicio del resultado (ya que SQL ordena los NULL antes que cualquier valor numérico en orden descendente), se añade la cláusula WHERE DEPTH IS NOT NULL.

**4. ¿Cuál es el nombre de los países donde la agricultura representa al menos el 50% del producto interno bruto? -- use INNER JOIN ... ON**

Se utiliza una combinación interna (INNER JOIN) entre las tablas ECONOMY y COUNTRY, relacionando la clave COUNTRY con el código de país. Luego se filtran los países cuya columna AGRICULTURE es mayor o igual al 50%.

```
SELECT NAME, COUNTRY
FROM ECONOMY
INNER JOIN COUNTRY ON ECONOMY.COUNTRY = COUNTRY.CODE
WHERE AGRICULTURE >= 50;
```



The screenshot shows a database query interface with two tabs: 'Query' and 'Query History'. The 'Query' tab is active, displaying the following SQL query:

```
1 SELECT NAME, COUNTRY
2 FROM ECONOMY
3 INNER JOIN COUNTRY ON ECONOMY.COUNTRY = COUNTRY.CODE
4 WHERE AGRICULTURE >= 50;
```


Below the query, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with 7 rows and 2 columns: 'name' and 'country'. The table has a toolbar with icons for expand, save, undo, redo, delete, refresh, download, and SQL.

	name character varying (50)	country character varying (4)
1	Solomon Islands	SOL
2	Guinea-Bissau	GUB
3	Liberia	LB
4	Central African Republic	RCA
5	Somalia	SO
6	Comoros	COM
7	Falkland Islands	FALK

**5. ¿Cómo se llama el río más largo? -- use una subconsulta**

Se utiliza una subconsulta para obtener la longitud máxima (MAX (LENGTH) ) de todos los ríos en la tabla RIVER. Luego, la consulta externa selecciona el nombre del río que tenga exactamente esa longitud, identificando así el río más largo.

```
SELECT NAME
FROM RIVER
WHERE LENGTH = (
    SELECT MAX (LENGTH)
    FROM RIVER
);
```

Query		Query History
1	SELECT NAME	
2	FROM RIVER	
3	WHERE LENGTH = (	
4	SELECT MAX(LENGTH)	
5	FROM RIVER	
6	);	
Data Output		Messages
		Notificati
<div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑</div> <div>🗄</div> <div>⬇</div> </div>		
	name	
	[PK] character varying (50) 	
1	Yangtze	