

LABORATORIO 273

ORGANIZACIÓN DE DATOS

Luz Basso, Agustín Esteche, Cristofer Gutiérrez,
Fiorella Pereira, Tatiana Rosa, Ignacio Suárez.

OBJETIVOS

Este laboratorio demuestra cómo usar funciones de base de datos comunes con las cláusulas GROUP BY y OVER.

Al completar este laboratorio aprenderemos lo siguiente:

- Usar la cláusula GROUP BY con la función agregada SUM()
- Usar la cláusula OVER con la función de ventana RANK()
- Usar la cláusula OVER con la función agregada SUM() y la función de ventana RANK()

TAREA 1: CONECTARSE A UN COMMAND HOST

Connect to instance [Info](#)

Connect to your instance i-01c762a64d20430ce (Command Host) using any of these options

EC2 Instance Connect | **Session Manager** | SSH client | EC2 serial console

Session Manager usage:

- Connect to your instance without SSH keys, a bastion host, or opening any inbound ports.
- Sessions are secured using an AWS Key Management Service key.
- You can log session commands and details in an Amazon S3 bucket or CloudWatch Logs log group.
- Configure sessions on the Session Manager [Preferences](#) page.

Nos conectaremos mediante la opción directa desde la página de instancias de Amazon.

```
sh-4.2$ sudo su
[root@ip-10-1-11-249 bin]# cd /home/ec2-user/
[root@ip-10-1-11-249 ec2-user]# mysql -u root --password='re:St@rt!9'
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 14
Server version: 10.6.18-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Conexión con la base
de datos.

TAREA 2: CONSULTE LAS BASE DE DATOS EXISTENTES

```
MariaDB [(none)]> SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
| world |  
+-----+  
5 rows in set (0.001 sec)
```

```
MariaDB [(none)]> █
```

En la consulta podemos verificar que existe la base de datos llamada “world”.

SELECCIONAMOS LA LISTA DE PAISES

Utilizamos **SELECT * FROM world.country** para seleccionar todos los paises de la base de datos world.

Code		Name	Continent	Region	SurfaceArea	IndepYear	P
GNPold	LocalName			GovernmentForm	Capital	Code2	
ABW	Aruba	North America	Caribbean	193.00	NULL	AW	
	793.00 Aruba		Nonmetropolitan Territory of The Netherlands	129	129	AW	
AFG	Afghanistan	Asia	Southern and Central Asia	652090.00	1919	1919	
	NULL Afganistan/Afghanistan		Islamic Emirate		1	AF	
AGO	Angola	Africa	Central Africa	1246700.00	1975	AO	
	7984.00 Angola		Republic		56	AO	
AIA	Anguilla	North America	Caribbean	96.00	NULL	AI	
	NULL Anguilla		Dependent Territory of the UK		62	AI	
ALB	Albania	Europe	Southern Europe	28748.00	1912	AL	
	2500.00 Shqipëria		Republic		34	AL	
AND	Andorra	Europe	Southern Europe	468.00	1278	AD	
	NULL Andorra		Parliamentary Coprincipality		55	AD	
ANT	Netherlands Antilles	North America	Caribbean	800.00	NULL	AN	
	NULL Nederlandse Antillen		Nonmetropolitan Territory of The Netherlands		33	AN	
ARE	United Arab Emirates	Asia	Middle East	83600.00	1971	AE	
	36846.00 Al-Imarat al-'Arabiya al-Muttahida		Emirate Federation		65	AE	
ARG	Argentina	South America	South America	2780400.00	1816	AR	
	323310.00 Argentina		Federal Republic		69	AR	
ARM	Armenia	Asia	Middle East	29800.00	1991	AM	
	1627.00 Hajastan		Republic		126	AM	
ASM	American Samoa	Oceania	Polynesia	199.00	NULL	AS	
	NULL Amerika Samoa		US Territory		54	AS	
ATA	Antarctica	Antarctica	Antarctica	13120000.00	NULL	AQ	
	NULL -		Co-administrated		NULL	AQ	
ATF	French Southern territories	Antarctica	Antarctica	7780.00	NULL		

MOSTRAMOS DOS REGIONES ORGANIZADAS

Para mostrar ambas regiones y ordenar los resultados por población en orden descendente.

```
SELECT Region, Name, Population FROM world.country WHERE Region =  
'Australia and New Zealand' ORDER By Population desc;
```

```
MariaDB [(none)]> SELECT Region, Name, Population FROM world.country WHERE Region = 'Australia and New Zealand' ORDER By Population desc;  
+-----+-----+-----+  
| Region | Name | Population |  
+-----+-----+-----+  
| Australia and New Zealand | Australia | 18886000 |  
| Australia and New Zealand | New Zealand | 3862000 |  
| Australia and New Zealand | Christmas Island | 2500 |  
| Australia and New Zealand | Norfolk Island | 2000 |  
| Australia and New Zealand | Cocos (Keeling) Islands | 600 |  
+-----+-----+-----+  
5 rows in set (0.000 sec)
```

AGRUPAMOS LOS REGISTROS RELACIONADOS.

Filtramos los registros con una condición donde la región es igual a Australia y Nueva Zelanda. Luego, agrupamos los resultados utilizando la cláusula **GROUP BY**. Posteriormente, aplicamos la función **SUM()** a los resultados agrupados para generar una población total para esa región.

```
MariaDB [(none)]> SELECT Region, SUM(Population) FROM world.country WHERE Region = 'Australia and New Zealand' GROUP By Region ORDER By SUM(Population) desc;
+-----+-----+
| Region | SUM(Population) |
+-----+-----+
| Australia and New Zealand |      22753100 |
+-----+-----+
1 row in set (0.002 sec)
```

AGRUPA LOS REGISTROS POR REGION Y POBLACIÓN

Se usa cláusula OVER() para agrupar los registros por Region y usa la función SUM() para agregar los registros. El resultado muestra la población de un país junto con el total continuo de la región.

```
MariaDB [(none)]> SELECT Region, Name, Population, SUM(Population) OVER(partition by Region ORDER BY Population) as 'Running Total' FROM world.country WHERE Region = 'Australia and New Zealand';
+-----+-----+-----+
| Region          | Name           | Population | Running Total |
+-----+-----+-----+
| Australia and New Zealand | Cocos (Keeling) Islands |      600 |          600 |
| Australia and New Zealand | Norfolk Island |     2000 |         2600 |
| Australia and New Zealand | Christmas Island |    2500 |         5100 |
| Australia and New Zealand | New Zealand | 3862000 | 3867100 |
| Australia and New Zealand | Australia | 18886000 | 22753100 |
+-----+-----+-----+
5 rows in set (0.001 sec)
```

SE AGRUPAN Y ORDENAN LOS REGISTROS

Se agrupa los registros por Region y los ordena por Population con la cláusula OVER(). La función RANK() genera un número de rango que indica la posición de cada registro en el conjunto de resultados.

```
MariaDB [(none)]> SELECT Region, Name, Population, SUM(Population) OVER(partition by Region ORDER BY Population) as 'Running Total', RANK() over(partition by region
RDER BY population) as 'Ranked' FROM world.country WHERE region = 'Australia and New Zealand';
+-----+-----+-----+-----+
| Region          | Name           | Population | Running Total | Ranked |
+-----+-----+-----+-----+
| Australia and New Zealand | Cocos (Keeling) Islands |      600 |          600 |    1 |
| Australia and New Zealand | Norfolk Island |     2000 |         2600 |    2 |
| Australia and New Zealand | Christmas Island |     2500 |         5100 |    3 |
| Australia and New Zealand | New Zealand    | 3862000 |      3867100 |    4 |
| Australia and New Zealand | Australia      | 18886000 |     22753100 |    5 |
+-----+-----+-----+-----+
5 rows in set (0.001 sec)
```

DESAFIO

Describir una consulta para calificar los países en cada región por su población de mayor a menor.
Determinar si usar la cláusula de agrupación GROUP BY o OVER y la función SUM() o RANK().

SELECT Region, Name, Population, RANK() OVER(partition by Region ORDER BY Population desc) as 'Ranked' FROM world.country order by Region, Ranked;

Region	Name	Population	Ranked
Antarctica	French Southern territories	0	1
Antarctica	Bouvet Island	0	1
Antarctica	South Georgia and the South Sandwich Islands	0	1
Antarctica	Antarctica	0	1
Antarctica	Heard Island and McDonald Islands	0	1
Australia and New Zealand	Australia	18886000	1
Australia and New Zealand	New Zealand	3862000	2
Australia and New Zealand	Christmas Island	2500	3
Australia and New Zealand	Norfolk Island	2000	4
Australia and New Zealand	Cocos (Keeling) Islands	600	5
Baltic Countries	Lithuania	3698500	1
Baltic Countries	Latvia	2424200	2
Baltic Countries	Estonia	1439200	3
British Islands	United Kingdom	59623400	1
British Islands	Ireland	3775100	2
Caribbean	Cuba	11201000	1
Caribbean	Dominican Republic	8495000	2
Caribbean	Haiti	8222000	3
Caribbean	Puerto Rico	3869000	4
Caribbean	Jamaica	2583000	5
Caribbean	Trinidad and Tobago	1295000	6
Caribbean	Guadeloupe	456000	7
Caribbean	Martinique	305000	8

DESAFIO

La mejor opción fue utilizar la función de ventana **OVER()** junto con **RANK()** para asignar un rango a cada país dentro de su región basado en su población. Esto nos permite evitar el uso de la cláusula **GROUP BY**, que se utiliza para realizar operaciones de agregación, como **SUM()**.

La función **RANK() OVER (PARTITION BY Region ORDER BY Poblacion DESC)** asignó un rango a cada país dentro de su región, ordenándolos por población de mayor a menor. El **PARTITION BY Region** asegura que el ranking se reinicie para cada región.

Este enfoque permite obtener el ranking de los países en cada región sin necesidad de agrupar los datos, lo que podría resultar en una consulta más eficiente y flexible.

MUCHAS
GRACIAS