

## Datos Generales:

**Nombre:** Tomás Alfredo Villaseca Constantinescu

**País:** Chile

**Fecha:** 07/10/2023

**Contacto:** [tomas.villaseca.c@gmail.com](mailto:tomas.villaseca.c@gmail.com)

Este laboratorio demuestra cómo usar funciones de base de datos comunes con **SELECT** y la cláusula **WHERE**.

Después de completar este laboratorio, podrá realizar lo siguiente:

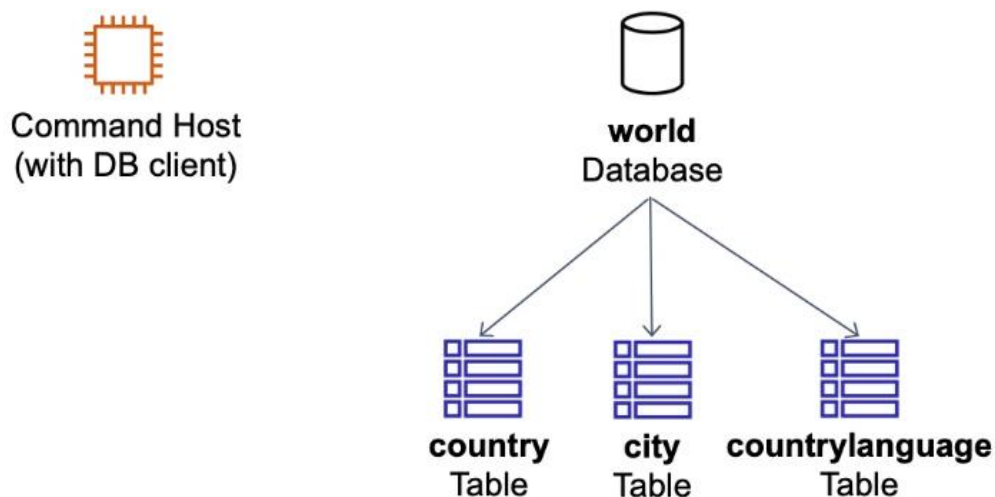
- Use las funciones agregadas **SUM()**, **MIN()**, **MAX()** y **AVG()** para resumir datos.
- Use la función **SUBSTRING\_INDEX()** para dividir las cadenas.
- Use las funciones **LENGTH()** y **TRIM()** para determinar la longitud de una cadena.
- Use la función **DISTINCT()** para filtrar los registros duplicados.
- Use las funciones en **SELECT** y la cláusula **WHERE**.

## Situación

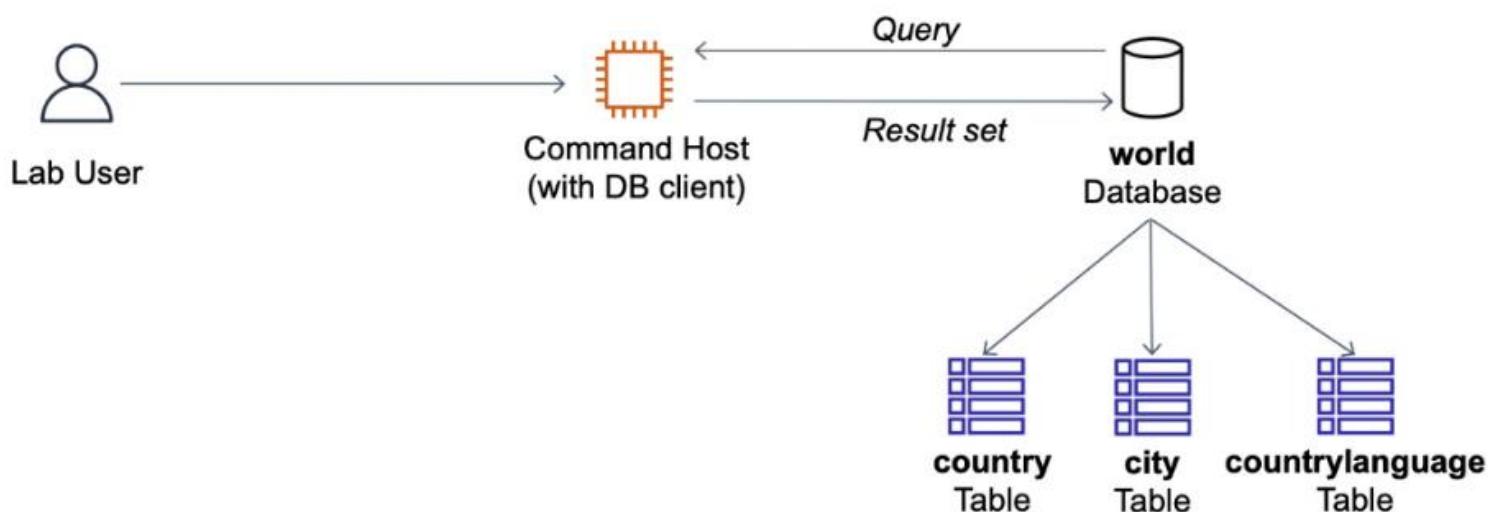
El equipo de operaciones de base de datos creó una base de datos relacional llamada **world** que contiene tres tablas: **city**, **country** y **countrylanguage**. Según los casos prácticos específicos definidos en el ejercicio de laboratorio, escribirá algunas consultas usando funciones de base de datos con **SELECT** y la cláusula **WHERE**.

## Entorno de laboratorio:

- Una instancia EC2 (Command Host) con un cliente de base de datos instalado.
- Una base de datos **world** que contiene 3 tablas: **city**, **country**, y **countrylanguage**.



Al final de este laboratorio habrá usado **SELECT** y la cláusula **WHERE** con algunas funciones de base de datos comunes.

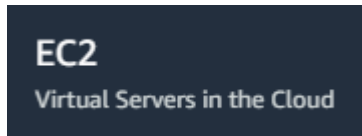


# Tarea 1: Conectarse a Command Host


En esta tarea, se conectará a una instancia de EC2 configurada con un cliente de base de datos (Command Host).

El cliente se usa para ejecutar las queries de SQL contra una base de datos relacional.

**Paso 1:** AWS Management Console → Services → Compute → EC2



**Paso 2:** Instances → Command Host

<input checked="" type="checkbox"/>	Name ▾	Instance ID	Instance state ▾	Instance type ▾	Status check
<input checked="" type="checkbox"/>	Command Host	i-0524ea72e85aedde0	Running 	t3.micro	2/2 checks passed

**Paso 3:** Command Host → Connect → Session Manager

## Connect to instance [Info](#)

Connect to your instance i-0524ea72e85aedde0 (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.

Cancel

Connect

**Paso 4:** Configurar la terminal para acceder a todas las herramientas y recursos necesarios.

```
sudo su  
cd /home/ec2-user/
```

- sudo = ejecutar comandos con privilegios.
- su = cambiar a otro usuario.
- cd = cambiar de directorio.

Session ID: user2741130=Tom\_\_sVillaseca-0ade39e12dc076495

```
sh-4.2$ sudo su  
[root@ip-10-1-11-97 bin]# cd /home/ec2-user/  
[root@ip-10-1-11-97 ec2-user]# pwd  
/home/ec2-user  
[root@ip-10-1-11-97 ec2-user]#
```

**Paso 5:** Conectarse a la base de datos relacional.

```
mysql -u root --password='re:St@rt!9'
```

- mysql = shell SQL para interactuar con los motores de bases de datos.
- -u = nombre de usuario de MySQL
- -p = contraseña de MySQL

Session ID: user2741130=Tom\_\_sVillaseca-0ade39e12dc076495

Instance ID: i-0524ea72e85aedde0

```
[root@ip-10-1-11-97 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 13  
Server version: 10.6.15-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.  
MariaDB [(none)]>
```

## Tarea 2: Consulte la base de datos world

En esta tarea, consultará la base de datos **world** usando varias **SELECT** y funciones de la base de datos.

Usará una función para procesar y manipular los datos en una consulta. Hay una amplia variedad de funciones SQL y este laboratorio revisa un subconjunto de funciones utilizadas con frecuencia.

**Paso 1:** Mostrar las bases de datos existentes.

```
SHOW DATABASES;
```

```
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| world |
+-----+
5 rows in set (0.003 sec)
```

**Paso 2:** Ver el esquema, los datos y la cantidad de filas de la tabla **country**.

```
SHOW COLUMNS FROM world.country;
```

```
MariaDB [(none)]> SHOW COLUMNS FROM world.country;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Code | char(3) | NO | PRI | | |
| Name | char(52) | NO | | | |
| Continent | enum('Asia','Europe','North America','Africa','Oceania','Antarctica','South America') | NO | | Asia | |
| Region | char(26) | NO | | | |
| SurfaceArea | decimal(10,2) | NO | | 0.00 | |
| IndepYear | smallint(6) | YES | | NULL | |
| Population | int(11) | NO | | 0 | |
| LifeExpectancy | decimal(3,1) | YES | | NULL | |
| GNP | decimal(10,2) | YES | | NULL | |
| GNPOld | decimal(10,2) | YES | | NULL | |
| LocalName | char(45) | NO | | | |
| GovernmentForm | char(45) | NO | | | |
| Capital | int(11) | YES | | NULL | |
| Code2 | char(2) | NO | | | |
+-----+-----+-----+-----+-----+
14 rows in set (0.001 sec)
```

**Paso 3:** Realizar una consulta utilizando las funciones **SUM()**, **MIN()**, **MAX()**, **AVG()**, y **COUNT()** para resumir datos de la tabla **country**.

- SUM() = Suma todos los valores de una columna.
- AVG() = Entrega el promedio de los valores de una columna.
- MAX() = Entrega la fila con el valor más alto de una columna.
- MIN() = Entrega la fila con el valor más bajo de una columna.
- COUNT() = Entrega el número total de filas de una columna.

```
SELECT sum(Population), avg(Population), max(Population), min(Population), count(Population) FROM world.country;
```

```
MariaDB [(none)]> SELECT sum(Population), avg(Population), max(Population), min(Population), co
+-----+-----+-----+-----+-----+
| sum(Population) | avg(Population) | max(Population) | min(Population) | count(Population) |
+-----+-----+-----+-----+-----+
|      6078749450 |    25434098.1172 |    1277558000    |              0   |             239   |
+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

**Paso 4:** Realizar una consulta en donde divida un string usando **SUBSTRING\_FUNCTION()**

```
SELECT Region, substring_index(Region, " ", 1) FROM world.country;
```

```
MariaDB [(none)]> SELECT Region, substring_index(Region, " ", 1) FROM world.country;
+-----+-----+
| Region                | substring_index(Region, " ", 1) |
+-----+-----+
| Caribbean              | Caribbean                        |
| Southern and Central Asia | Southern                        |
| Central Africa         | Central                          |
| Caribbean              | Caribbean                        |
| Southern Europe        | Southern                         |
| Southern Europe        | Southern                         |
| Caribbean              | Caribbean                        |
| Middle East            | Middle                          |
| South America          | South                           |
+-----+-----+
```

- En este caso la división del string ocurre en donde hay un espacio.

**Paso 5:** Realizar una consulta en donde utilice la función **SUBSTRING\_FUNCTION()** como parte de una condición de la cláusula **WHERE** para filtrar registros que incluyen **Southern** en la primera parte del nombre de la región.

```
SELECT Name, Region from world.country WHERE substring_index(Region, " ", 1) = "Southern";
```

```
MariaDB [(none)]> SELECT Name, Region from world.country WHERE substring_index(Region, " ", 1) = "Southern";
+-----+-----+
| Name                               | Region                               |
+-----+-----+
| Afghanistan                        | Southern and Central Asia          |
| Albania                            | Southern Europe                     |
| Andorra                            | Southern Europe                     |
| Bangladesh                         | Southern and Central Asia          |
| Bosnia and Herzegovina             | Southern Europe                     |
| Bhutan                             | Southern and Central Asia          |
| Botswana                           | Southern Africa                     |
| Spain                              | Southern Europe                     |
| Gibraltar                          | Southern Europe                     |
| Greece                             | Southern Europe                     |
| Croatia                            | Southern Europe                     |
| India                              | Southern and Central Asia          |
| Iran                               | Southern and Central Asia          |
| Italy                               | Southern Europe                     |
| Kazakstan                          | Southern and Central Asia          |
| Kyrgyzstan                         | Southern and Central Asia          |
| Sri Lanka                          | Southern and Central Asia          |
| Lesotho                            | Southern Africa                     |
| Maldives                           | Southern and Central Asia          |
| Macedonia                          | Southern Europe                     |
| Malta                              | Southern Europe                     |
| Namibia                            | Southern Africa                     |
| Nepal                              | Southern and Central Asia          |
| Pakistan                           | Southern and Central Asia          |
| Portugal                           | Southern Europe                     |
| San Marino                         | Southern Europe                     |
| Slovenia                           | Southern Europe                     |
| Swaziland                          | Southern Africa                     |
| Tajikistan                         | Southern and Central Asia          |
| Turkmenistan                       | Southern and Central Asia          |
| Uzbekistan                         | Southern and Central Asia          |
| Holy See (Vatican City State)      | Southern Europe                     |
| Yugoslavia                         | Southern Europe                     |
| South Africa                       | Southern Africa                     |
+-----+-----+
34 rows in set (0.000 sec)
```



**Paso 6:** Realizar una consulta utilizando las funciones **LENGTH()** y **TRIM()** para determinar cuántos caracteres tiene un string.

- TRIM() = Borra los espacios en blanco iniciales y finales.
- LENGTH() = Arroja un conteo de los caracteres resultantes.

```
SELECT Region FROM world.country WHERE LENGTH(TRIM(Region)) < 10;
```

```
MariaDB [(none)]> SELECT Region FROM world.country WHERE LENGTH(TRIM(Region)) < 10;
```

Region
Caribbean
Caribbean
Caribbean
Polynesia
Caribbean
Caribbean
Caribbean
Polynesia
Caribbean
Caribbean
Caribbean
Caribbean
Melanesia
Caribbean
Caribbean
Caribbean
Caribbean
Caribbean
Caribbean
Caribbean
Melanesia
Polynesia
Polynesia
Melanesia
Caribbean
Polynesia
Melanesia
Caribbean
Polynesia
Polynesia
Caribbean
Polynesia
Caribbean
Caribbean
Caribbean
Melanesia
Polynesia
Polynesia

```
39 rows in set (0.000 sec)
```

- En este caso se arrojan las regiones que tienen menos de 10 caracteres en sus nombres.
- Se puede notar que el resultado entrega registros duplicados.



**Paso 7:** Realizar la misma consulta anterior agregando la función **DISTINCT()** para filtrar los registros duplicados.

```
SELECT DISTINCT(Region) FROM world.country WHERE LENGTH(TRIM(Region)) < 10;
```

```
MariaDB [(none)]> SELECT DISTINCT(Region) FROM world.country WHERE LENGTH(TRIM(Region)) < 10;
+-----+
| Region |
+-----+
| Caribbean |
| Polynesia |
| Melanesia |
+-----+
3 rows in set (0.001 sec)
```

## Desafío

Consulte la tabla **country** para arrojar un conjunto de registros basado en el siguiente requisito:

Escriba una consulta que arroje filas que tienen **Micronesia/Caribbean** como nombre de región. El resultado debe dividir la región **Micronesia** y **Caribbean** en dos columnas separadas: una llamada **Region Name 1** y una llamada **Region Name 2**.

```
SELECT Name, substring_index(Region, "/", 1) as "Region Name 1", substring_index(region, "/", -1) as "Region Name 2" FROM
world.country WHERE Region = "Micronesia/Caribbean";
```

```
MariaDB [(none)]> SELECT Name, substring_index(Region, "/", 1) as "Region Name
+-----+-----+-----+
| Name | Region Name 1 | Region Name 2 |
+-----+-----+-----+
| United States Minor Outlying Islands | Micronesia | Caribbean |
+-----+-----+-----+
1 row in set (0.000 sec)
```

Laboratorio Completado

