Francesc
de Borja Moll
Centre Integrat de
Formació Professional

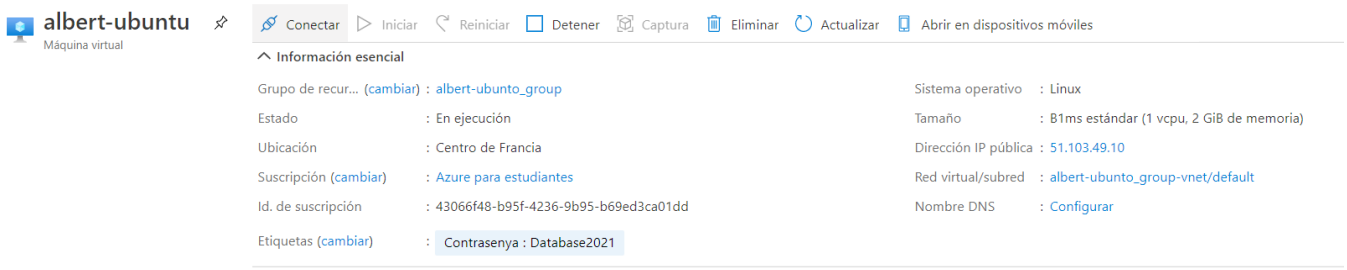
P5: Create the Government of the Balearic Island Database for Telephones calls Management using the Shell

Albert Perelló Puertas
IFC31W Group

INDEX

1. Create database “TELEFONIA”

To begin we must start our virtual machine and access it from Windows PowerShell.



Once inside, we start the container and access the MySQL console.

```
PS C:\Users\apere> ssh aperellop@51.103.49.10
aperellop@51.103.49.10's password:
```

```
aperellop@albert-ubuntu:~$ sudo docker ps -a
[sudo] password for aperellop:
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
          NAMES
d56e6ff17f3c        mysql:latest        "docker-entrypoint.s…"   6 weeks ago         Up 30 minutes       3306/tcp, 33
060/tcp   dbms
aperellop@albert-ubuntu:~$ sudo docker start dbms
dbms
```

```
aperellop@albert-ubuntu:~$ sudo docker exec -it dbms mysql -u root -p
Enter password:
```

Now we can create and use the “TELEFONIA” database with the following commands:

```
mysql> CREATE DATABASE TELEFONIA;
Query OK, 1 row affected (0.06 sec)

mysql> USE TELEFONIA;
Database changed
```

2. Creation of tables and relationships described above

Next, we will create within our TELEFONIA database, all the necessary tables for the management of telephone calls from the Govern de les Illes Balears (CAIB): USERS, PHONES, COMPANIES and CALLS.

2.1. Users table

```
mysql> CREATE TABLE USERS (NIF VARCHAR(9) PRIMARY KEY NOT NULL, Name VARCHAR(20) NOT NULL, Lats_Name1 VARCHAR(20) NOT NULL, Last_Name2 VARCHAR(20), Fec_Nacim DATE);
Query OK, 0 rows affected (0.24 sec)
```

```
mysql> DESCRIBE USERS;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| NIF        | varchar(9)  | NO   | PRI | NULL    |       |
| Name       | varchar(20) | NO   |     | NULL    |       |
| Lats_Name1 | varchar(20) | NO   |     | NULL    |       |
| Last_Name2 | varchar(20) | YES  |     | NULL    |       |
| Fec_Nacim  | date        | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

2.2. Phones table

```
mysql> CREATE TABLE PHONES (Numero VARCHAR(13) PRIMARY KEY NOT NULL, NIF_Usuario INT NOT NULL, ID_Compania INT);
Query OK, 0 rows affected (0.23 sec)
```

```
mysql> DESCRIBE PHONES;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| Numero      | varchar(13) | NO   | PRI | NULL    |       |
| NIF_Usuario | int         | NO   |     | NULL    |       |
| ID_Compania | int         | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

2.3. Companies table

```
mysql> CREATE TABLE COMPANIES (ID_Comapania INT PRIMARY KEY NOT NULL, Name VARCHAR(20) NOT NULL, Anio_Fundacion INT NOT NULL);
Query OK, 0 rows affected (0.07 sec)
```

```
mysql> DESCRIBE COMPANIES;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| ID_Comapania  | int         | NO   | PRI | NULL    |       |
| Name          | varchar(20) | NO   |     | NULL    |       |
| Anio_Fundacion | int        | NO   |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

2.4. Calls table

```
mysql> CREATE TABLE CALLS (Num_Llamante VARCHAR(13) NOT NULL, Num_Llamado VARCHAR(13) NOT NULL, Date DATE, Time TIME
NOT NULL, PRIMARY KEY (`Num_Llamante`, `Num_Llamado`));
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> DESCRIBE CALLS;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| Num_Llamante | varchar(13) | NO   | PRI | NULL    |       |
| Num_Llamado  | varchar(13) | NO   | PRI | NULL    |       |
| Date         | date        | YES  |     | NULL    |       |
| Time         | time        | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

3. In a PHONES relationship, modify the NUMERO attribute to contain 11 character strings

To change the “Numero” attribute so that it can contain 11 characters at most instead of 13 as it had in the beginning we use the following command:

```
mysql>      alter table PHONES
            modify column Numero VARCHAR(11) NOT NULL;
```

```
mysql> alter table PHONES modify column Numero VARCHAR(11) NOT NULL;
Query OK, 0 rows affected (0.73 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

4. In the USERS relationship, add the attribute ADDRESS after the LastName2 and contain Strings of a 100 characters

To add the ADDRESS attribute that can contain up to 100 characters we will use the following command:

```
mysql>      alter table USERS
            add Address VARCHAR(100) NULL;
```

```
mysql> alter table USERS add Address VARCHAR(100) NULL;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

5. In the USERS relationship change their identifier to CUSTOMERS

For this we will use the command `rename` as seen in the image

```
mysql> rename table USERS to CUSTOMERS;
Query OK, 0 rows affected (0.12 sec)
```

6. In the CUSTOMERS relationship, change their identifiers of the LastName1 and LastName2 by AP1 and AP2 respectively

This time we will use the `change` command to rename the two Last_ Name of the CUSTOMERS table.

```
mysql> alter table CUSTOMERS change Lats_Name1 AP1 varchar(20);
Query OK, 0 rows affected (0.13 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> alter table CUSTOMERS change Last_Name2 AP2 varchar(20);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

7. In the CUSTOMERS relationship, change their identifier back to USERS

```
mysql> rename table CUSTOMERS to USERS;
Query OK, 0 rows affected (0.05 sec)
```

8. In the USERS relationship, change the AP1 and AP2 attribute identifiers to LastName1 and LastName 2

```
mysql> alter table USERS change AP1 LastName1 varchar(20), change AP2 LastName2 varchar(20);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

9. In the CALLS relationship add a constrain so that a number cannot be called himself

So that a number cannot call itself we use the following command:

```
mysql>      alter table CALLS
            add unique(Num_Llamante);
```

```
mysql> alter table CALLS
    -> add unique(Num_Llamante);
Query OK, 0 rows affected (0.10 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

10. In the USERS relationship, delete the ADDRESS attribute

To delete the addres attribute we use the `drop` command.

```
mysql> alter table USERS drop column Address;
Query OK, 0 rows affected (0.11 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

11. Delete all relationships from the TELEFONIA database

```
mysql> drop table USERS;
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> drop table CALLS;
Query OK, 0 rows affected (0.10 sec)
```

```
mysql> drop table PHONES;
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> drop table COMPANIES;
Query OK, 0 rows affected (0.08 sec)
```

12. Delete the TELEFONIA database

```
mysql> drop database TELEFONIA;
Query OK, 0 rows affected (0.34 sec)
```

13. Bibliography

https://www.w3schools.com/