

	Juan Diego Rafael Orrillo Salazar		Guía
	Date: 14/09	Course: 2024/2025	Module: ASIR2

GUÍA DE INSTALACIÓN DE MYSQL/MARIADB EN DISTROS BASADAS EN DEBIAN



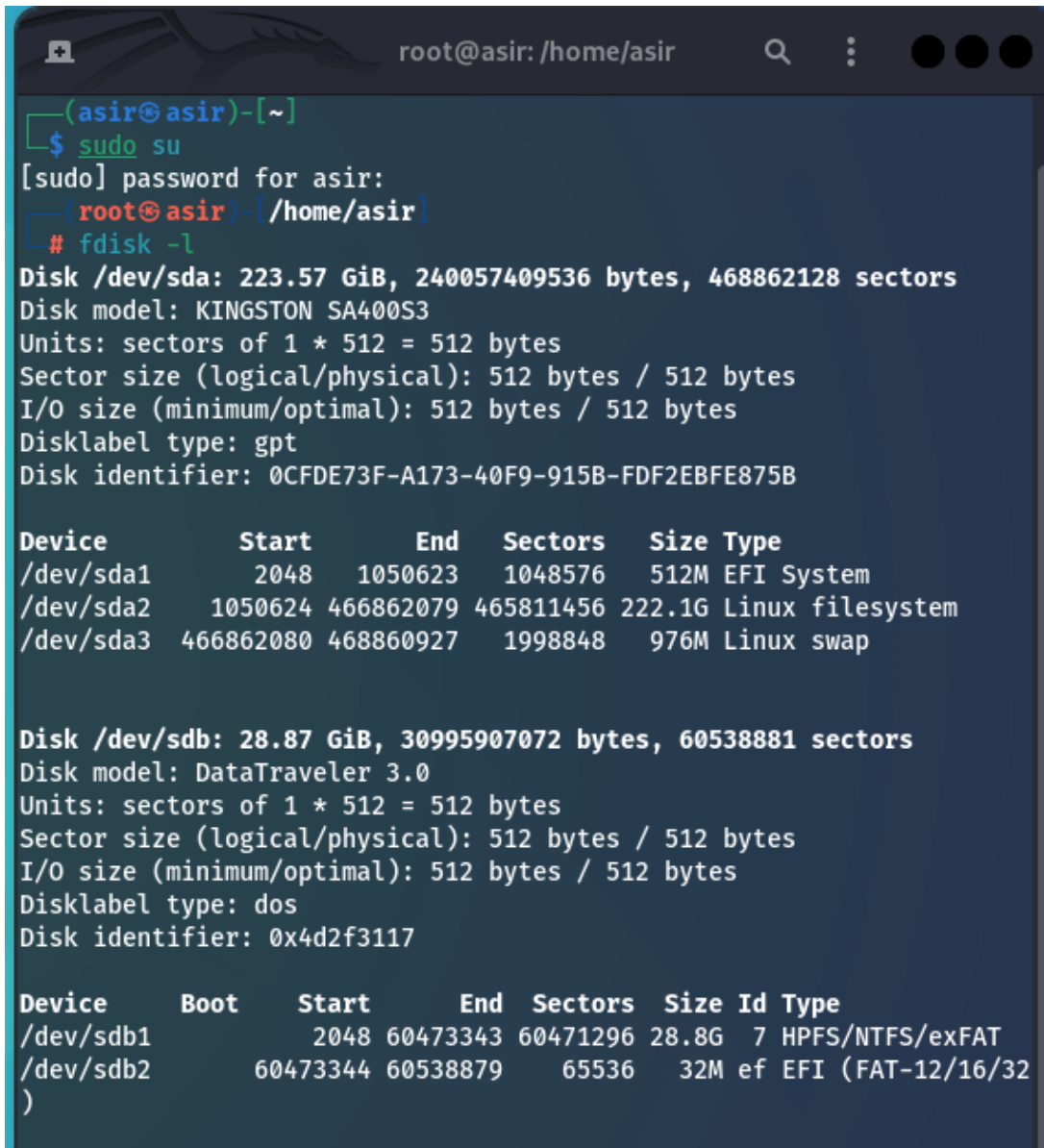
ÍNDICE.

Prólogo: BACKUP DE LA GUÍA.	3
INSTALACIÓN DE MARIADB:	5
CONFIGURACIÓN DE SERVICIOS:.....	6
CREACIÓN DE UN USUARIO, CONTRASEÑA Y PERMISOS:	8
CREACIÓN DE UNA BASE DE DATOS:	9
WEBGRAFÍA:.....	17

Prólogo: BACKUP DE LA GUÍA.

Antes de meterme con mysql y como esto es una guía, necesito tener un lugar donde resguardar mis archivos , con lo cual voy a enseñar donde y como haré el backup y por si es necesario, como formatear un usb (porque el que tengo actualmente está particionado porque tengo ventoy e isos dentro de sí mismo ya que es un usb booteable), para hacer un respaldo de archivos.

Usaremos `fdisk -l`, para hacer un mapeo de discos.

A terminal window with a dark blue background and light blue text. The window title is 'root@asir: /home/asir'. The user is logged in as 'asir' and runs 'sudo su'. The prompt changes to '[sudo] password for asir:' and then to 'root@asir: [/home/asir]'. The user runs '# fdisk -l'. The output shows details for two disks: /dev/sda and /dev/sdb. For /dev/sda, it lists sectors, size (223.57 GiB), model (KINGSTON SA400S3), units, sector size, I/O size, disklabel type (gpt), and disk identifier. It also shows a table of partitions: /dev/sda1 (512M EFI System), /dev/sda2 (222.1G Linux filesystem), and /dev/sda3 (976M Linux swap). For /dev/sdb, it lists sectors, size (28.87 GiB), model (DataTraveler 3.0), units, sector size, I/O size, disklabel type (dos), and disk identifier. It also shows a table of partitions: /dev/sdb1 (28.8G HPFS/NTFS/exFAT) and /dev/sdb2 (32M ef EFI (FAT-12/16/32)).

```
(asir@asir)-[~]
$ sudo su
[sudo] password for asir:
root@asir: [/home/asir]
# fdisk -l
Disk /dev/sda: 223.57 GiB, 240057409536 bytes, 468862128 sectors
Disk model: KINGSTON SA400S3
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 0CFDE73F-A173-40F9-915B-FDF2EBFE875B

Device            Start        End          Sectors      Size Type
/dev/sda1          2048        1050623      1048576      512M EFI System
/dev/sda2         1050624    466862079    465811456    222.1G Linux filesystem
/dev/sda3         466862080    468860927      1998848      976M Linux swap

Disk /dev/sdb: 28.87 GiB, 30995907072 bytes, 60538881 sectors
Disk model: DataTraveler 3.0
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x4d2f3117

Device    Boot      Start        End    Sectors    Size Id Type
/dev/sdb1                2048    60473343    60471296    28.8G   7 HPFS/NTFS/exFAT
/dev/sdb2            60473344    60538879      65536     32M  ef EFI (FAT-12/16/32)
)
```

O por otra parte podemos usar lsblk, la cual tiene menos información sin embargo es más legible si se es un poco inexperto y nos daremos cuenta de que el disco es el sdb

```
(asir@asir)-[~]
$ lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda          8:0    0 223.6G  0 disk
├─sda1       8:1    0   512M  0 part /boot/efi
├─sda2       8:2    0 222.1G  0 part /
├─sda3       8:3    0   976M  0 part [SWAP]
sdb          8:16    1  28.9G  0 disk
├─sdb1       8:17    1  28.8G  0 part /media/asir/Ventoy
└─sdb2       8:18    1    32M  0 part
```

Lo desmontamos con “umount”

```
Device      Boot      Start        End    Sectors   Size Id Type
/dev/sdb1                2048 60473343 60471296 28.8G  7 HPFS/NTFS/exFAT
/dev/sdb2          60473344 60538879    65536    32M ef EFI (FAT-12/16/32)

(root@asir)-[/home/asir]
# umount /dev/sdb1
```

Y aquí le damos formato con “mkfs.vfat -F 32”, claro que dependiendo de lo que queramos, en este caso específico es para crear carpetas dentro de ese usb, sin embargo si se tratara de un disco normal para instalar el kernel de Linux se debe formatear en ext4.

```
(root@asir)-[/home/asir]
# mkfs.vfat -F 32 /dev/sdb1
mkfs.fat 4.2 (2021-01-31)
```

Cabe recalcar aquí que el “e2label” nos va a servir para darle nombre a este respaldo.

```
(root@asir)-[/home/asir]
# e2label /dev/sdb1 "Backup-GuiasASIR"
```

```
"Backup-GuiasASIR"
Partition 1
31 GB NTFS
```

Y ya estaría.

INSTALACIÓN DE MARIADB:

Para comenzar la instalación de mariadb/servicios de mysql primero tendremos que actualizar repositorios:

```
(root@asir)-[/home/asir]
# apt update && sudo apt upgrade -y
```

Ok luego de esto se debe seguir paso a paso la siguiente guía.

```
(root@asir)-[/home/asir]
# apt install mariadb-client mariadb-server
The following packages were automatically installed and are no longer required:
  g++-13 g++-13-x86-64-linux-gnu libstdc++-13-dev
Use 'sudo apt autoremove' to remove them.

Installing:
  mariadb-client mariadb-server

Installing dependencies:
  galera-4          liburing2
  gawk              mariadb-client-core
  libcgi-fast-perl  mariadb-common
  libcgi-pm-perl    mariadb-plugin-provider-bzip2
  libconfig-inifiles-perl mariadb-plugin-provider-lz4
  libdbd-mariadb-perl mariadb-plugin-provider-lzma
  libdbi-perl       mariadb-plugin-provider-lzo
  libfcgi-bin       mariadb-plugin-provider-snappy
  libfcgi-perl      mariadb-server-core
  libfcgi0t64       mysql-common
  libhtml-template-perl pv
  libmariadb3       rsync
  libsigsegv2       socat
  libterm-readkey-perl

Suggested packages:
  gawk-doc          libsql-statement-perl  netcat-openbsd
  libmldbm-perl     libipc-sharedcache-perl doc-base
  libnet-daemon-perl mariadb-test           python3-braceexpand

Summary:
  Upgrading: 0, Installing: 29, Removing: 0, Not Upgrading: 1
  Download size: 19.3 MB
  Space needed: 196 MB / 215 GB available

Continue? [Y/n]
```

CONFIGURACIÓN DE SERVICIOS:

Lo que necesitaremos saber aquí es la importancia de “systemctl” ya que es fundamental en las distros de linux ya que con el podemos gestionar y administrar muchos servicios, como por ejemplo bluetooth o networking ya que si llegamos a tener algún problema con alguno de estos dos, se podría arreglar muy fácil con “systemctl”, en este caso específico sólo lo usaremos para una vez instalado mariadb iniciar y habilitarle.

```
(root@asir)-[/home/asir]
# systemctl start mariadb

(root@asir)-[/home/asir]
# systemctl enable mariadb
Synchronizing state of mariadb.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable mariadb
Created symlink '/etc/systemd/system/multi-user.target.wants/mariadb.service' -> '/usr/lib/systemd/system/mariadb.service'.

(root@asir)-[/home/asir]
# systemctl status mariadb
● mariadb.service - MariaDB 11.4.3 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: >
   Active: active (running) since Sat 2024-09-14 22:50:08 CEST; 15s ago
 Invocation: 8b492bdbdddf42a28f2c03cf8c269b78
    Docs: man:mariadb(8)
          https://mariadb.com/kb/en/library/systemd/
   Main PID: 9460 (mariabdd)
    Status: "Taking your SQL requests now..."
     Tasks: 14 (limit: 61332)
    Memory: 210.8M (peak: 215.4M)
       CPU: 2.730s
    CGroup: /system.slice/mariadb.service
            └─9460 /usr/sbin/mariabdd

Sep 14 22:50:06 asir mariabdd[9460]: 2024-09-14 22:50:06 0 [Note] InnoDB: log s>
Sep 14 22:50:06 asir mariabdd[9460]: 2024-09-14 22:50:06 0 [Note] Plugin 'FEEDB>
Sep 14 22:50:06 asir mariabdd[9460]: 2024-09-14 22:50:06 0 [Note] Plugin 'wsrep>
Sep 14 22:50:06 asir mariabdd[9460]: 2024-09-14 22:50:06 0 [Note] InnoDB: Loadi>
Sep 14 22:50:06 asir mariabdd[9460]: 2024-09-14 22:50:06 0 [Note] InnoDB: Buffe>
Sep 14 22:50:08 asir mariabdd[9460]: 2024-09-14 22:50:08 0 [Note] Server socket>
Sep 14 22:50:08 asir mariabdd[9460]: 2024-09-14 22:50:08 0 [Note] mariabdd: Eve>
Sep 14 22:50:08 asir mariabdd[9460]: 2024-09-14 22:50:08 0 [Note] /usr/sbin/mar>
Sep 14 22:50:08 asir mariabdd[9460]: Version: '11.4.3-MariaDB-1' socket: '/run>
Sep 14 22:50:08 asir systemd[1]: Started mariadb.service - MariaDB 11.4.3 datab>
lines 1-24/24 (END)
```

Ahora es importante entender que necesitamos ejecutar un comando, este es “mysql_secure_installation”, esto para deshabilitar varias configuraciones por defecto, remover bases de datos que ya vienen integradas y que son de “test”, cuentas anónimas que también ya vienen por defecto, etc...

Cuando ejecutemos “mysql_secure_installation”, por mi sistema me aparecerá una alerta porque debo instalar algo antes, sin embargo luego de que lo instale sí me permitirá ejecutar este comando de manera correcta.

```

(root@asir)-[/home/asir]
# mysql_secure_installation
Command 'mysql_secure_installation' not found, but can be installed with:
apt install mariadb-client-compat
Do you want to install it? (N/y)y
apt install mariadb-client-compat
The following packages were automatically installed and are no longer required:
  g++-13 g++-13-x86-64-linux-gnu libstdc++-13-dev
Use 'sudo apt autoremove' to remove them.

Installing:
  mariadb-client-compat

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 1
  Download size: 27.3 kB
  Space needed: 103 kB / 215 GB available

Get:1 http://kali.download/kali kali-rolling/main amd64 mariadb-client-compat all 1:11.4.3-1 [27.3 kB]
Fetched 27.3 kB in 0s (64.4 kB/s)
Selecting previously unselected package mariadb-client-compat.
(Reading database ... 176972 files and directories currently installed.)
Preparing to unpack .../mariadb-client-compat_1%3a11.4.3-1_all.deb ...
Unpacking mariadb-client-compat (1:11.4.3-1) ...
Setting up mariadb-client-compat (1:11.4.3-1) ...
Processing triggers for kali-menu (2024.3.1) ...
Processing triggers for man-db (2.13.0-1) ...

(root@asir)-[/home/asir]
# mysql_secure_installation
/usr/bin/mysql_secure_installation: Deprecated program name. It will be removed in a future release, use 'mariadb-secure-installation' instead

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] Y
Enabled successfully!
Reloading privilege tables..
... Success!

```

Como lo mencioné antes el comando es para remover y configurar algunas cosas pero ya depende de las necesidades de cada uno.

CREACIÓN DE UN USUARIO, CONTRASEÑA Y PERMISOS:

```
(root@asir)-[/home/asir]
# sudo mysql -u root -p

Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 41
Server version: 11.4.3-MariaDB-1 Debian n/a

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

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> █
```

Ok vamos a crear un usuario con **"CREATE USER"**, un recordatorio de esto es que no se deben confundir las comillas con la comilla simple (comilla simple', comillas").

Colocaremos el user (asir2 en mi caso), en comillas simples junto a localhost (localhost nos indica que nuestro usuario podrá conectarse a la database sólo desde el pc en donde esté instalado mariadb, SIN EMBARGO, si quisiéramos conectarnos desde otro pc a la base de datos debemos colocar '%' en vez de localhost pero ahora mismo no me es necesario). También necesitaremos usar **IDENTIFIED BY "** para crear una contraseña, aquí el ejemplo:

```
MariaDB [(none)]>
MariaDB [(none)]> CREATE USER 'asir2'@'localhost' IDENTIFIED BY 'Root12345';
Query OK, 0 rows affected (0.005 sec)
```

Usaremos **grant all privileges** para otorgar permisos, *.* para hacerle entenderle al sistema que queremos dar permisos al usuario en todas las bases de datos, tablas en el servidor y **TO** para indicar a quien queremos darle estos permisos/privilegios.

```
MariaDB [(none)]>
MariaDB [(none)]> GRANT ALL PRIVILEGES ON *.* TO 'asir2'@'localhost';
Query OK, 0 rows affected (0.004 sec)

MariaDB [(none)]> █
```


Y por ultimo para aplicar los privilegios:

```
MariaDB [(none)]> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.003 sec)  
  
MariaDB [(none)]> 
```

```
(root@asir)-[/home/asir]  
# mysql -u asir2 -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 43  
Server version: 11.4.3-MariaDB-1 Debian n/a  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Support MariaDB developers by giving a star at https://github.com/MariaDB/server  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> 
```

CREACIÓN DE UNA BASE DE DATOS:

A continuación voy a resolver los siguientes ejercicios:

1. Crea una base de datos llamada empresa.

Aquí no necesita tanta explicación, sólo es necesario saber que si se va a crear una base de datos y se quiere introducir tablas dentro de ella se debe usar “create database” seguido del nombre de la base de datos, posteriormente se debe usar “use” y el nombre de la base de datos.

```
MariaDB [(none)]> create database empresa;  
Query OK, 1 row affected (0.001 sec)  
  
MariaDB [(none)]> use empresa;  
Database changed  
MariaDB [empresa]> 
```

Y como podemos darnos cuenta antes salía “MariaDB [(none)]”, sin embargo cuando usamos una base de datos en contenido de los corchetes cambiará indicando que ya estamos en la base de datos (MariaDB [empresa]).

2. Crea una tabla empleados en la base de datos empresa con los campos: id (INT, AUTO_INCREMENT, PRIMARY KEY), nombre (VARCHAR(100)), apellidos (VARCHAR(100)), salario (DECIMAL(10,2)) y fecha_contratacion (DATE).

```
MariaDB [(none)]> use empresa;
Database changed
MariaDB [empresa]> create table empleados (
  -> id int AUTO_INCREMENT PRIMARY KEY,
  -> nombre VARCHAR(100),
  -> apellidos VARCHAR (100),
  -> salario DECIMAL (10,2),
  -> fecha_contratacion DATE
  -> );
Query OK, 0 rows affected (0.019 sec)

MariaDB [empresa]> 
```

3. Inserta tres empleados en la tabla empleados.

Bien, aquí agregaremos 3 empleados, Guillermo Palacios, Cipriano Castro, Vicente Gomez, junto a ellos irán un salario y fecha de contratación.

INSERT INTO para insertar datos en tablas e indicamos las columnas a donde queremos que se introduzcan estos datos y ya estaría.

```
MariaDB [empresa]> INSERT INTO empleados (nombre, apellidos, salario, fecha_contratacion) VALUES ('Guillermo', 'Palacios', '2001.00', '2024-08-28'), ('Cipriano', 'Castro', '2400.00', '2023-04-01')
Query OK, 3 rows affected (0.004 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

4. Realiza una consulta que muestre el nombre y apellido de los empleados que tienen un salario mayor a 2000.

```
MariaDB [empresa]> SELECT nombre,
-> apellidos FROM empleados WHERE salario > 2000;
+-----+-----+
| nombre | apellidos |
+-----+-----+
| Guillermo | Palacios |
| Cipriano | Castro |
+-----+-----+
2 rows in set (0.001 sec)
```

5. Actualiza el salario de un empleado cuyo id es 1, incrementándolo en un 10%.

La mayoría de los comandos simples como este se pueden llegar a formular de varias maneras, sin embargo esta se me hizo la más fácil, que es sólo sumar el salario actual más un 10% haciendo una operación matemática sencilla.

```
MariaDB [empresa]> UPDATE empleados
-> SET salario = salario + (salario * 0.10)
-> WHERE id = 1;
Query OK, 1 row affected (0.005 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [empresa]> SELECT id, nombre, apellidos, salario, fecha_contratacion FROM empleados;
+-----+-----+-----+-----+-----+
| id | nombre | apellidos | salario | fecha_contratacion |
+-----+-----+-----+-----+-----+
| 1 | Guillermo | Palacios | 2201.10 | 2024-08-28 |
| 2 | Cipriano | Castro | 2400.00 | 2023-04-01 |
| 3 | Vicente | Gomez | 1700.00 | 2022-05-03 |
+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)
```

6. Elimina de la tabla empleados al empleado cuyo id es 3.

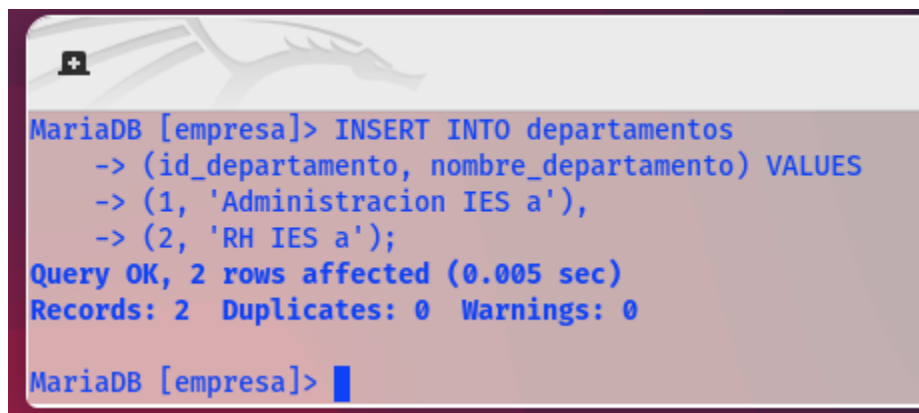
```
MariaDB [empresa]> DELETE FROM empleados WHERE id = 3;
Query OK, 1 row affected (0.002 sec)

MariaDB [empresa]> □
```

7. Crea una nueva tabla departamentos con los campos: id_departamento (INT, PRIMARY KEY), nombre_departamento (VARCHAR(50)).

```
MariaDB [empresa]> CREATE TABLE departamentos (  
  -> id_departamento int primary key, nombre_departamento varchar(50)  
  -> );  
Query OK, 0 rows affected (0.014 sec)
```

8. Inserta dos departamentos en la tabla departamentos.



```
MariaDB [empresa]> INSERT INTO departamentos  
  -> (id_departamento, nombre_departamento) VALUES  
  -> (1, 'Administracion IES a'),  
  -> (2, 'RH IES a');  
Query OK, 2 rows affected (0.005 sec)  
Records: 2  Duplicates: 0  Warnings: 0  
  
MariaDB [empresa]> █
```

9. Realiza una consulta que muestre todos los empleados junto con el nombre de su departamento (usa un JOIN entre empleados y departamentos).

Aquí si lo podríamos hacer de varias maneras, por ejemplo agregando la primary key de departamentos (id_departamento) en la tabla empleados, también cabe recalcar que en el ejercicio dice “usa un JOIN”, podríamos usar desde un JOIN simple, un inner join, un left join y hasta un right join.

Usaremos ALTER TABLE y ADD COLUMN, todo esto con el fin de usar como clave foránea a id_departamento.

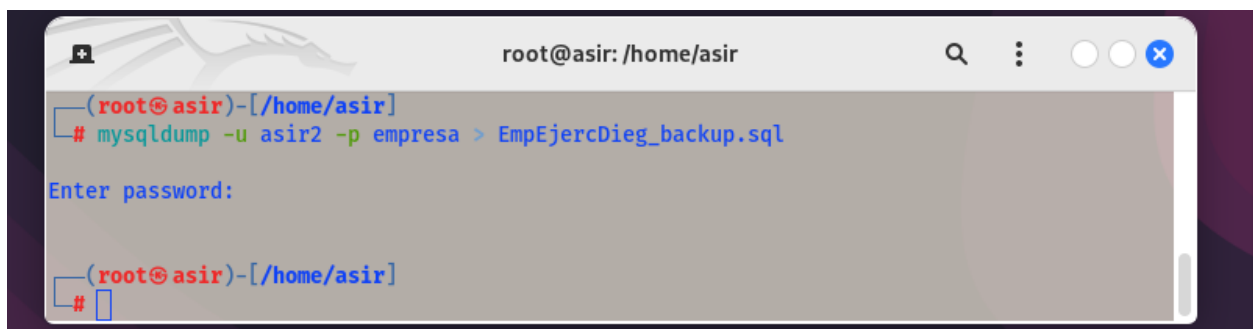
```
MariaDB [empresa]> ALTER TABLE empleados ADD COLUMN id_departamento INT;
Query OK, 0 rows affected (0.017 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [empresa]> UPDATE empleados SET id_departamento = 1 WHERE id = 1;
Query OK, 1 row affected (0.005 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
MariaDB [empresa]> UPDATE empleados SET id_departamento = 2 WHERE id = 2;
Query OK, 1 row affected (0.004 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

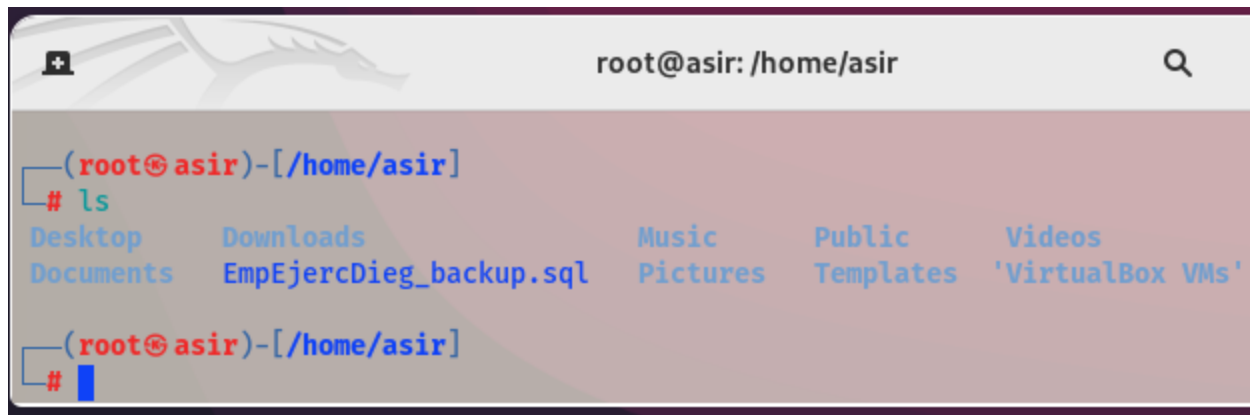
```
MariaDB [empresa]> SELECT empleados.nombre, empleados.apellidos, departamentos.nombre_departamento
-> FROM empleados
-> INNER JOIN departamentos
-> ON empleados.id_departamento = departamentos.id_departamento;
+-----+-----+-----+
| nombre | apellidos | nombre_departamento |
+-----+-----+-----+
| Guillermo | Palacios | Administracion IES a |
| Cipriano | Castro | RH IES a |
+-----+-----+-----+
2 rows in set (0.001 sec)
```

10. Realiza una copia de seguridad de la base de datos empresa utilizando una instrucción de MariaDB



```
root@asir: /home/asir
(root@asir)-[/home/asir]
# mysqldump -u asir2 -p empresa > EmpEjercDieg_backup.sql
Enter password:
(root@asir)-[/home/asir]
#
```

Para verificar la existencia de este archivo podemos usar un simple "ls" ya que en el momento en el que usé el comando estaba ubicado en /home/asir

A terminal window with a dark purple title bar. The title bar contains a window icon on the left, the text 'root@asir: /home/asir' in the center, and a search icon on the right. The terminal content shows a prompt '(root@asir)-[/home/asir]' followed by the command '# ls'. The output of the command is a list of files and directories: Desktop, Downloads, Music, Public, Videos, Documents, EmpEjercDieg_backup.sql, Pictures, Templates, and 'VirtualBox VMs'. Below the output, the prompt '(root@asir)-[/home/asir]' is shown again with a cursor on the '#' character.

```
(root@asir)-[/home/asir]
# ls
Desktop    Downloads  Music      Public     Videos
Documents  EmpEjercDieg_backup.sql  Pictures    Templates  'VirtualBox VMs'
```

```
(root@asir)-[/home/asir]
#
```

Y con un **"cat"** podremos ver el contenido del mismo

```

(root@asir)-[/home/asir]
# cat EmpEjercDieg_backup.sql
/*M!999999\ - enable the sandbox mode */
-- MariaDB dump 10.19-11.4.3-MariaDB, for debian-linux-gnu (x86_64)
--
-- Host: localhost    Database: empresa
--
-- Server version      11.4.3-MariaDB-1

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*M!100616 SET @OLD_NOTE_VERBOSITY=@@NOTE_VERBOSITY, NOTE_VERBOSITY=0 */;

--
-- Table structure for table `departamentos`
--

DROP TABLE IF EXISTS `departamentos`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `departamentos` (
  `id_departamento` int(11) NOT NULL,
  `nombre_departamento` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`id_departamento`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_uca1400_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `departamentos`
--

LOCK TABLES `departamentos` WRITE;
/*!40000 ALTER TABLE `departamentos` DISABLE KEYS */;
INSERT INTO `departamentos` VALUES
(1,'Administracion IES a'),
(2,'RH IES a');
/*!40000 ALTER TABLE `departamentos` ENABLE KEYS */;

```

```

/*!40000 ALTER TABLE `departamentos` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `empleados`
--

DROP TABLE IF EXISTS `empleados`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `empleados` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(100) DEFAULT NULL,
  `apellidos` varchar(100) DEFAULT NULL,
  `salario` decimal(10,2) DEFAULT NULL,
  `fecha_contratacion` date DEFAULT NULL,
  `id_departamento` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_uca1400_ai_
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `empleados`
--

LOCK TABLES `empleados` WRITE;
/*!40000 ALTER TABLE `empleados` DISABLE KEYS */;
INSERT INTO `empleados` VALUES
(1,'Guillermo','Palacios',2201.10,'2024-08-28',1),
(2,'Cipriano','Castro',2400.00,'2023-04-01',2);
/*!40000 ALTER TABLE `empleados` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!M!100616 SET NOTE_VERBOSITY=@OLD_NOTE_VERBOSITY */;

-- Dump completed on 2024-09-15  4:01:26

mysql> (root@asir)-[/home/asir]

```


WEBGRAFÍA:

<https://www.geeksforgeeks.org/how-to-install-mysql-mariadb-in-linux/>

<https://styde.net/crear-y-eliminar-tablas-en-mysql-mariadb/>

<https://es.stackoverflow.com/questions/371967/como-calcular-porcentaje-en-mysql>