AWS Cloud Practices- MySQL and MariaDB Setup

Prepared by Andres Pascasio
GitHub Profile

Step-by-Step Guide

Note: This guide is based on practical exercises and knowledge gained from AWS training. For more detailed and specific information, refer to the official AWS documentation and training resources.

Disclaimer: This guide is for educational purposes and reflects the author's understanding of AWS cloud practices. It is not an official AWS document or tutorial. For official AWS guidance, please refer to AWS's official documentation and training materials.

Overview

This guide walks you through the process of setting up MySQL or MariaDB on an AWS EC2 instance running Amazon Linux 2. It includes steps for installation, configuration, table creation, data insertion, and managing user passwords.

1. Connect to the Linux Server Using SSH

Using PuTTY on Windows

1. **Convert Your PEM File to PPK Format:**

   PuTTY does not directly support PEM files, so you'll need to convert your PEM file to PPK format using PuTTYgen.
   - Open PuTTYgen.
   - Click Load and select your PEM file.
   - Click Save private key to save the file as PPK.
2. **Connect to Your EC2 Instance:**
   - Open PuTTY.

- In the **Session** category, enter your EC2 instance’s public IP address or DNS name in the **Host Name** field.
- In the **Connection > SSH > Auth** category, browse for your PPK file under **Private key file for authentication**.
- Click **Open** to start the SSH session.

Note: You may need to enter the username for your EC2 instance, which is typically `ec2-user` for Amazon Linux 2.

Using SSH on Linux/Mac

For Linux and macOS users, connect using the terminal:

```
ssh -i /path/to/your/key.pem username@server_ip
```

Replace `/path/to/your/key.pem`, `username`, and `server_ip` with your specific details.

2. Install MySQL or MariaDB

Option A: Install MySQL

1. Update and install MySQL:

   ```
   sudo yum update -y
   sudo yum install mysql-server
   ```

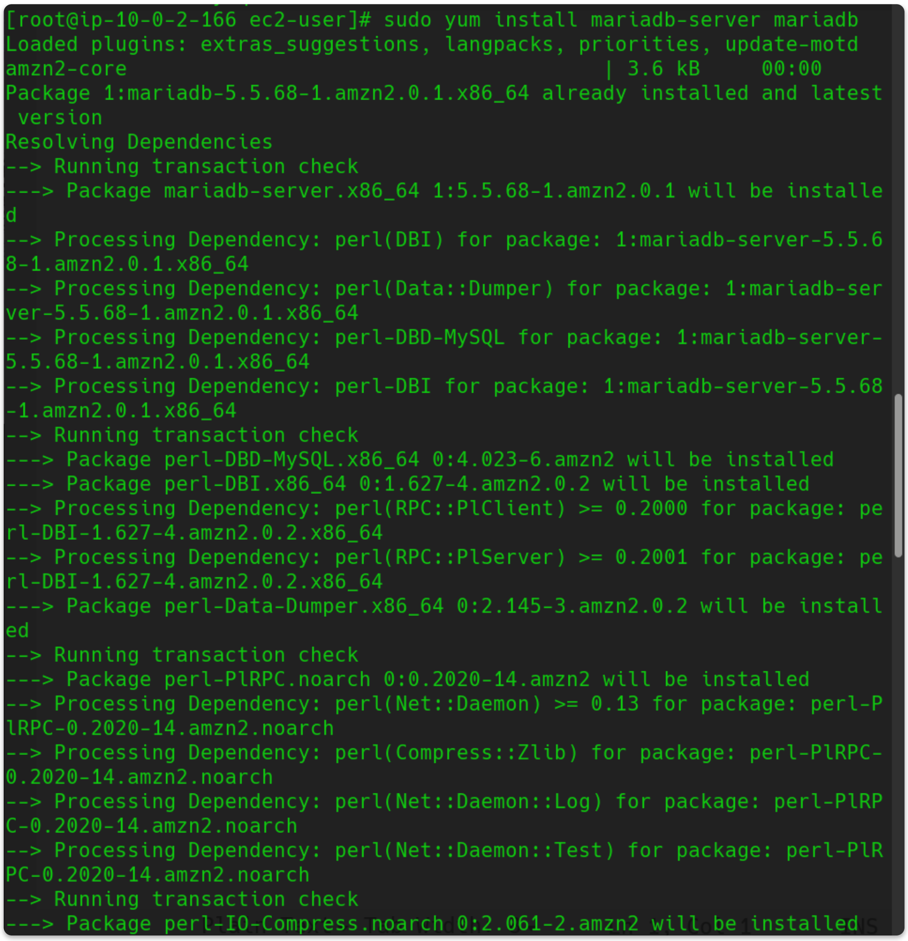
2. Start the MySQL service:

   ```
   sudo systemctl start mysqld
   ```

Option B: Install MariaDB

1. Install MariaDB:

   ```
   sudo yum update -y
   sudo yum install mariadb-server mariadb
   ```



```
[root@ip-10-0-2-166 ec2-user]# sudo yum install mariadb-server mariadb
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                       | 3.6 kB     00:00
Package 1:mariadb-5.5.68-1.amzn2.0.1.x86_64 already installed and latest
 version
Resolving Dependencies
--> Running transaction check
---> Package mariadb-server.x86_64 1:5.5.68-1.amzn2.0.1 will be installe
d
--> Processing Dependency: perl(DBI) for package: 1:mariadb-server-5.5.6
8-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Data::Dumper) for package: 1:mariadb-ser
ver-5.5.68-1.amzn2.0.1.x86_64
--> Processing Dependency: perl-DBD-MySQL for package: 1:mariadb-server-
5.5.68-1.amzn2.0.1.x86_64
--> Processing Dependency: perl-DBI for package: 1:mariadb-server-5.5.68
-1.amzn2.0.1.x86_64
--> Running transaction check
---> Package perl-DBD-MySQL.x86_64 0:4.023-6.amzn2 will be installed
---> Package perl-DBI.x86_64 0:1.627-4.amzn2.0.2 will be installed
--> Processing Dependency: perl(RPC::PlClient) >= 0.2000 for package: pe
rl-DBI-1.627-4.amzn2.0.2.x86_64
--> Processing Dependency: perl(RPC::PlServer) >= 0.2001 for package: pe
rl-DBI-1.627-4.amzn2.0.2.x86_64
---> Package perl-Data-Dumper.x86_64 0:2.145-3.amzn2.0.2 will be install
ed
--> Running transaction check
---> Package perl-PlRPC.noarch 0:0.2020-14.amzn2 will be installed
--> Processing Dependency: perl(Net::Daemon) >= 0.13 for package: perl-P
lRPC-0.2020-14.amzn2.noarch
--> Processing Dependency: perl(Compress::Zlib) for package: perl-PlRPC-
0.2020-14.amzn2.noarch
--> Processing Dependency: perl(Net::Daemon::Log) for package: perl-PlRP
C-0.2020-14.amzn2.noarch
--> Processing Dependency: perl(Net::Daemon::Test) for package: perl-PlR
PC-0.2020-14.amzn2.noarch
--> Running transaction check
---> Package perl-IO-Compress.noarch 0:2.061-2.amzn2 will be installed
```

2. Start and enable the MariaDB service:

```
sudo systemctl start mariadb
sudo systemctl enable mariadb
```

3. Connect to MySQL or MariaDB

Using MySQL Client

```
mysql -u root
```

Explanation:

- `mysql -u root`: This command connects to the MySQL server as the `root` user without prompting for a password. This is generally the case right after installation when the root user might not have a password set yet.

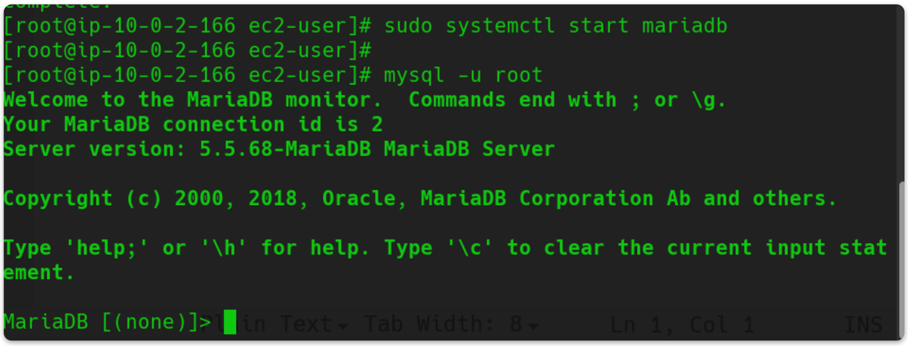
Best Practice:

- **Using Passwords**: For security reasons, it is highly recommended to use a password for the root user and any other MySQL user accounts. Instead of connecting without a password, use the following command to connect with a password prompt:

```
mysql -u username -p
```

```
- **`-u username`**: Replace `username` with the MySQL user you want to
connect as.
- **`-p`**: This flag tells MySQL to prompt for the user’s password. You
will be asked to enter the password interactively.
```

Note: If you are working in a production environment or any environment where security is a concern, always ensure that you have set up strong passwords for all MySQL user accounts and follow best security practices.



```
[root@ip-10-0-2-166 ec2-user]# sudo systemctl start mariadb
[root@ip-10-0-2-166 ec2-user]#
[root@ip-10-0-2-166 ec2-user]# mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 5.5.68-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 stat
ement.

MariaDB [(none)]>
```

If you encounter a socket error or the service is not found, verify the service status or try the correct service name.

4. Create a Database

1. Create a new database:

```
CREATE DATABASE students;
```

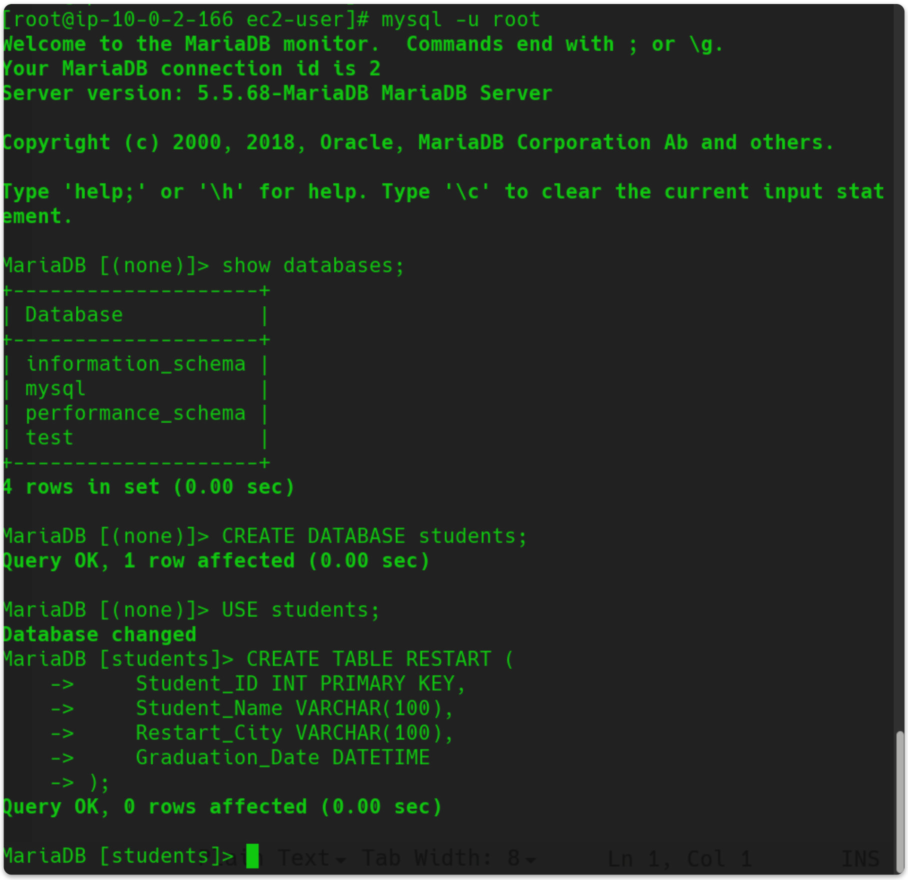
2. Select the database:

```
USE students;
```

5. Create the `RESTART` Table

1. Create the `RESTART` table:

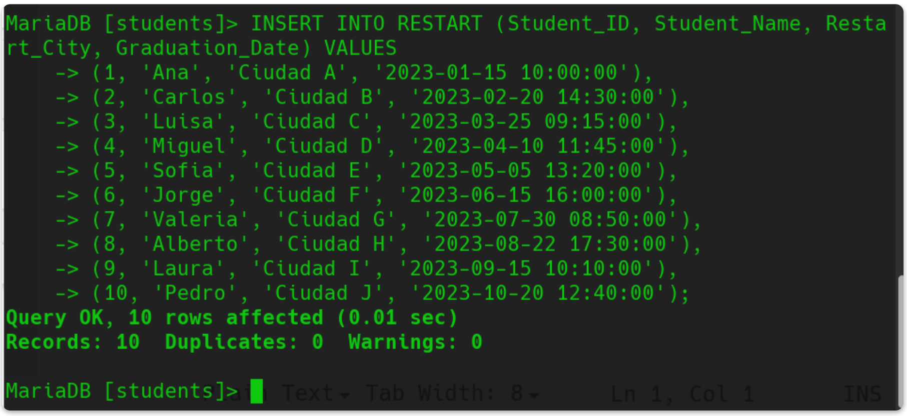
```
CREATE TABLE RESTART (
    Student_ID INT PRIMARY KEY,
    Student_Name VARCHAR(100),
    Restart_City VARCHAR(100),
    Graduation_Date DATETIME
);
```



```
[root@ip-10-0-2-166 ec2-user]# mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 5.5.68-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 stat
ement.

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
+--------------------+
4 rows in set (0.00 sec)

MariaDB [(none)]> CREATE DATABASE students;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> USE students;
Database changed
MariaDB [students]> CREATE TABLE RESTART (
    ->     Student_ID INT PRIMARY KEY,
    ->     Student_Name VARCHAR(100),
    ->     Restart_City VARCHAR(100),
    ->     Graduation_Date DATETIME
    -> );
Query OK, 0 rows affected (0.00 sec)

MariaDB [students]> 
```

2. Insert 10 sample rows:

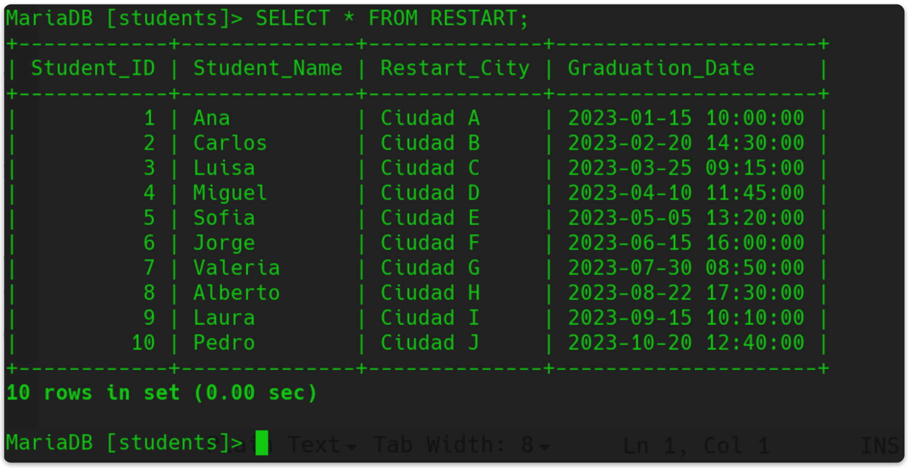
```
INSERT INTO RESTART (Student_ID, Student_Name, Restart_City,
Graduation_Date) VALUES
(1, 'Ana', 'City A', '2023-01-15 10:00:00'),
(2, 'Carlos', 'City B', '2023-02-20 14:30:00'),
(3, 'Luisa', 'City C', '2023-03-25 09:15:00'),
(4, 'Miguel', 'City D', '2023-04-10 11:45:00'),
(5, 'Sofia', 'City E', '2023-05-05 13:20:00'),
(6, 'Jorge', 'City F', '2023-06-15 16:00:00'),
(7, 'Valeria', 'City G', '2023-07-30 08:50:00'),
(8, 'Alberto', 'City H', '2023-08-22 17:30:00'),
(9, 'Laura', 'City I', '2023-09-15 10:10:00'),
(10, 'Pedro', 'City J', '2023-10-20 12:40:00');
```



```
MariaDB [students]> INSERT INTO RESTART (Student_ID, Student_Name, Resta
rt_City, Graduation_Date) VALUES
    -> (1, 'Ana', 'Ciudad A', '2023-01-15 10:00:00'),
    -> (2, 'Carlos', 'Ciudad B', '2023-02-20 14:30:00'),
    -> (3, 'Luisa', 'Ciudad C', '2023-03-25 09:15:00'),
    -> (4, 'Miguel', 'Ciudad D', '2023-04-10 11:45:00'),
    -> (5, 'Sofia', 'Ciudad E', '2023-05-05 13:20:00'),
    -> (6, 'Jorge', 'Ciudad F', '2023-06-15 16:00:00'),
    -> (7, 'Valeria', 'Ciudad G', '2023-07-30 08:50:00'),
    -> (8, 'Alberto', 'Ciudad H', '2023-08-22 17:30:00'),
    -> (9, 'Laura', 'Ciudad I', '2023-09-15 10:10:00'),
    -> (10, 'Pedro', 'Ciudad J', '2023-10-20 12:40:00');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

MariaDB [students]>
```

3. Select all rows:

```
SELECT * FROM RESTART;
```

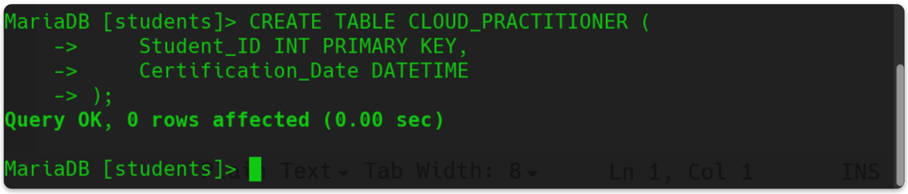


```
MariaDB [students]> SELECT * FROM RESTART;
+------------+--------------+--------------+---------------------+
| Student_ID | Student_Name | Restart_City | Graduation_Date     |
+------------+--------------+--------------+---------------------+
|          1 | Ana          | Ciudad A     | 2023-01-15 10:00:00 |
|          2 | Carlos       | Ciudad B     | 2023-02-20 14:30:00 |
|          3 | Luisa        | Ciudad C     | 2023-03-25 09:15:00 |
|          4 | Miguel       | Ciudad D     | 2023-04-10 11:45:00 |
|          5 | Sofia        | Ciudad E     | 2023-05-05 13:20:00 |
|          6 | Jorge        | Ciudad F     | 2023-06-15 16:00:00 |
|          7 | Valeria      | Ciudad G     | 2023-07-30 08:50:00 |
|          8 | Alberto      | Ciudad H     | 2023-08-22 17:30:00 |
|          9 | Laura        | Ciudad I     | 2023-09-15 10:10:00 |
|         10 | Pedro        | Ciudad J     | 2023-10-20 12:40:00 |
+------------+--------------+--------------+---------------------+
10 rows in set (0.00 sec)

MariaDB [students]>
```

6. Create the `CLOUD_PRACTITIONER` Table

1. Create the `CLOUD_PRACTITIONER` table:

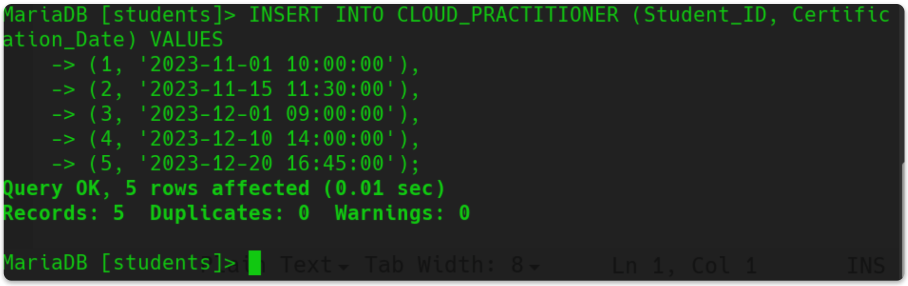
```
CREATE TABLE CLOUD_PRACTITIONER (
    Student_ID INT PRIMARY KEY,
    Certification_Date DATETIME
);
```



```
MariaDB [students]> CREATE TABLE CLOUD_PRACTITIONER (
    ->     Student_ID INT PRIMARY KEY,
    ->     Certification_Date DATETIME
    -> );
Query OK, 0 rows affected (0.00 sec)

MariaDB [students]>
```

2. Insert 5 sample rows:

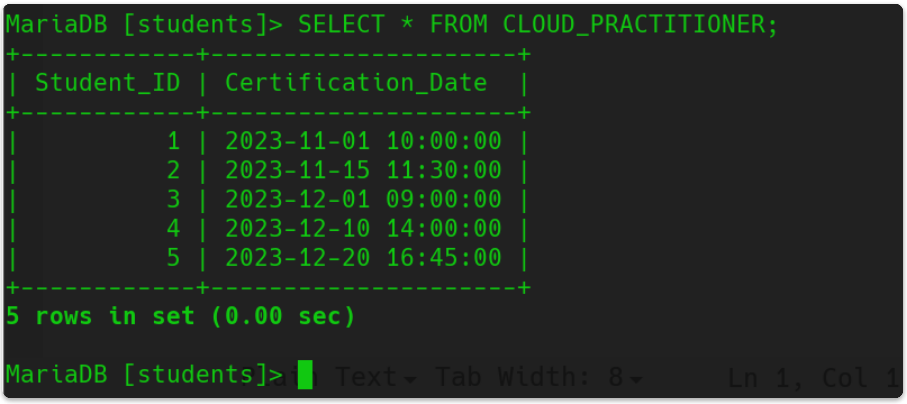
```
INSERT INTO CLOUD_PRACTITIONER (Student_ID, Certification_Date) VALUES
(1, '2024-01-15 10:00:00'),
(2, '2024-01-20 11:00:00'),
(3, '2024-02-01 09:30:00'),
(4, '2024-02-10 14:00:00'),
(5, '2024-02-15 16:00:00');
```



```
MariaDB [students]> INSERT INTO CLOUD_PRACTITIONER (Student_ID, Certific
ation_Date) VALUES
    -> (1, '2023-11-01 10:00:00'),
    -> (2, '2023-11-15 11:30:00'),
    -> (3, '2023-12-01 09:00:00'),
    -> (4, '2023-12-10 14:00:00'),
    -> (5, '2023-12-20 16:45:00');
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [students]>
```

3. Select all rows:

```
SELECT * FROM CLOUD_PRACTITIONER;
```

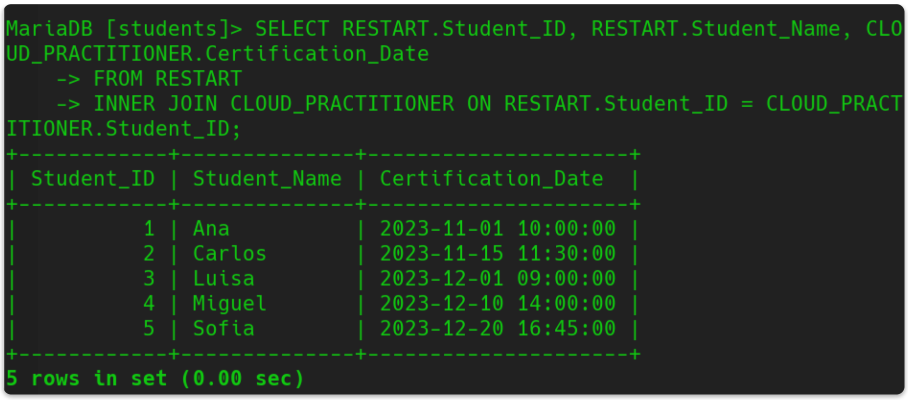


```
MariaDB [students]> SELECT * FROM CLOUD_PRACTITIONER;
+------------+---------------------+
| Student_ID | Certification_Date  |
+------------+---------------------+
|          1 | 2023-11-01 10:00:00 |
|          2 | 2023-11-15 11:30:00 |
|          3 | 2023-12-01 09:00:00 |
|          4 | 2023-12-10 14:00:00 |
|          5 | 2023-12-20 16:45:00 |
+------------+---------------------+
5 rows in set (0.00 sec)

MariaDB [students]>
```

7. Perform an Inner Join Between the Two Tables

1. Execute an INNER JOIN query:

```
SELECT RESTART.Student_ID, RESTART.Student_Name,
CLOUD_PRACTITIONER.Certification_Date
FROM RESTART
INNER JOIN CLOUD_PRACTITIONER ON RESTART.Student_ID =
CLOUD_PRACTITIONER.Student_ID;
```



```
MariaDB [students]> SELECT RESTART.Student_ID, RESTART.Student_Name, CLO
UD_PRACTITIONER.Certification_Date
    -> FROM RESTART
    -> INNER JOIN CLOUD_PRACTITIONER ON RESTART.Student_ID = CLOUD_PRACT
ITIONER.Student_ID;
+------------+--------------+---------------------+
| Student_ID | Student_Name | Certification_Date  |
+------------+--------------+---------------------+
|          1 | Ana          | 2023-11-01 10:00:00 |
|          2 | Carlos       | 2023-11-15 11:30:00 |
|          3 | Luisa        | 2023-12-01 09:00:00 |
|          4 | Miguel       | 2023-12-10 14:00:00 |
|          5 | Sofia        | 2023-12-20 16:45:00 |
+------------+--------------+---------------------+
5 rows in set (0.00 sec)
```

8. Change the Root User Password

1. To set or change the root password:

   ```
   ALTER USER 'root'@'localhost' IDENTIFIED BY 'NewPassword';
   FLUSH PRIVILEGES;
   ```

   Replace `'NewPassword'` with your desired password.
2. If you need to reset the password and cannot access the server:
   1. Stop the MySQL/MariaDB service:

      ```
      sudo systemctl stop mysqld   # For MySQL
      sudo systemctl stop mariadb  # For MariaDB
      ```

   2. Start the service with skip-grant-tables:

      ```
      sudo mysqld_safe --skip-grant-tables &
      ```

   3. Connect to the server:

      ```
      mysql -u root
      ```

   4. Change the root password:

      ```
      ALTER USER 'root'@'localhost' IDENTIFIED BY 'NewPassword';
      FLUSH PRIVILEGES;
      ```

   5. Restart MySQL/MariaDB normally:

```
sudo systemctl stop mysqld   # For MySQL
sudo systemctl stop mariadb  # For MariaDB
sudo systemctl start mysqld  # For MySQL
sudo systemctl start mariadb # For MariaDB
```

Conclusion

Congratulations on successfully setting up your MySQL or MariaDB environment and performing the required operations! This guide has taken you through the essential steps to effectively configure and utilise your database.

For more information and official documentation, the following resources may be helpful:

- MySQL Documentation
- MariaDB Documentation

Thank you for following this guide, and best of luck with your database projects!
