Data Modelling Tools

AUTM08016

Topic 6
Administrating MariaDB

Dr Diarmuid Ó Briain
Version 1.0 [01 January 2024]

TUS

Ollscoil Teicneolaíochta na Sionainne:
Lár Tíre, An tIarthar Láir

Technological University of the Shannon:
Midlands Midwest

Dr Diarmuid Ó Briain

CEng, FIEI CISSP®

Linux Version

```
~$ lsb_release -a | grep Description
Description:    Ubuntu 22.04.3 LTS
```

Apache2 Version

```
~$ apache2 -v
Server version: Apache/2.4.52 (Ubuntu)
Server built:   2023-10-26T13:44:44
```

MariaDB Version

```
~$ mariadb --version
mariadb  Ver 15.1 Distrib 10.6.12-MariaDB, for debian-linux-gnu
(x86_64) using  EditLine wrapper
```

php Version

```
~$ php --version | head -1
PHP 8.1.2-1ubuntu2.14 (cli) (built: Aug 18 2023 11:41:11) (NTS)
```

perl version

```
~$ perl --version | grep subversion
This is perl 5, version 34, subversion 0 (v5.34.0) built for
x86_64-linux-gnu-thread-multi
```

python version

```
~$ python3 --version
Python 3.10.12
```

Table of Contents

Table of Figures

This page is intentionally blank

1. Introduction

DataBase Administrators (DBA) play a crucial role in ensuring the smooth and efficient operation of MariaDB databases. Their responsibilities encompass a wide range of tasks, from designing and implementing databases to monitoring and troubleshooting performance issues. DBAs are responsible for ensuring that databases are scalable, secure, and reliable, enabling organisations to harness the power of their data to drive business growth and innovation.

1.1 Objectives

By the end of this topic the learner will be able to

- Discuss the MariaDB database
- Administrate a MariaDB database using SQL

1.2 What is SQL

Structured Query Language (SQL), is a database query language that was adopted as an industry standard in 1986. It is the standard language for Relational DataBase Management Systems (RDBMS). SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database.

Some common RDBMS that use SQL are: MySQL, Oracle, and Microsoft SQL Server. Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system. However, the standard SQL commands such as SELECT, INSERT, UPDATE, DELETE, CREATE, and DROP can be used to accomplish almost everything that one needs to do with a database.

1.3 SQL Tables

An RDBMS contains one or more objects called tables. The data or information for the database is stored in these tables. Tables are uniquely identified by their names and are comprised of columns and rows. Columns contain the column name, data type, and any other attributes for the column. Rows contain the records or data for the columns.

2. MariaDB and MySQL Introduction

MariaDB is a free RDBMS that implements the SQL standard. MariaDB is a fork of MySQL and essentially structures MariaDB uses are the same as MySQL as it was designed as a drop-in replacement of MySQL.

MariaDB has added a significant number of new features, which makes MariaDB better in terms of performance and user-orientation than MySQL.

While both databases are free and open-source; however, when Oracle acquired Sun Microsystems there was a potential conflict of interest between MySQL and Oracle's commercial database - Oracle Database Server. As a result, some engineers from the MySQL programme created a MariaDB fork of the MySQL code base.

Both databases can be found on most distributions of GNU/Linux and exists for many other Operating Systems including FreeBSD, Sun Solaris, SCO UNIX, BSD UNIX and Microsoft Windows. MariaDB and MySQL software and documentation can be downloaded from:

MariaDB - `https://mariadb.org`

MySQL Community Server - `https://dev.mysql.com`

Many Graphical tools are available for MariaDB and MySQL. The **phpMyAdmin** project is an excellent example that can be acquired at:

`http://www.phpmyadmin.net`

2.1 GNU/Linux Server Setup

Install GNU/Linux on your server, all distributions are generally based on either Debian or Fedora (Redhat). These notes concentrate on a Debian Server using its **apt** install tool.

Install the GNU/Linux server and follow the distribution instructions. Once installed create a user other than the root user that you will use. Login to the server as root and a # symbol is presented on the shell to indicate super-user access.

2.1.1 Check the MariaDB installation

Log into the database with the **`root`** user credentials and add a new **`admin`** administrative user with all privileges. The **`FLUSH PRIVILEGES`** command then loads the grant tables in the mariaDB database enabling the changes to take effect without reloading or restarting **`mariadb`** service.

```
~$ sudo mysql --user=root
MariaDB [(none)]>

> create user 'admin'@'localhost' identified by 'admpass';
Query OK, 0 rows affected (0.002 sec)

> grant all privileges on *.* to 'admin'@'localhost';
Query OK, 0 rows affected (0.002 sec)

> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.002 sec)

> exit;
bye
```

2.2 Accessing the MariaDB Server

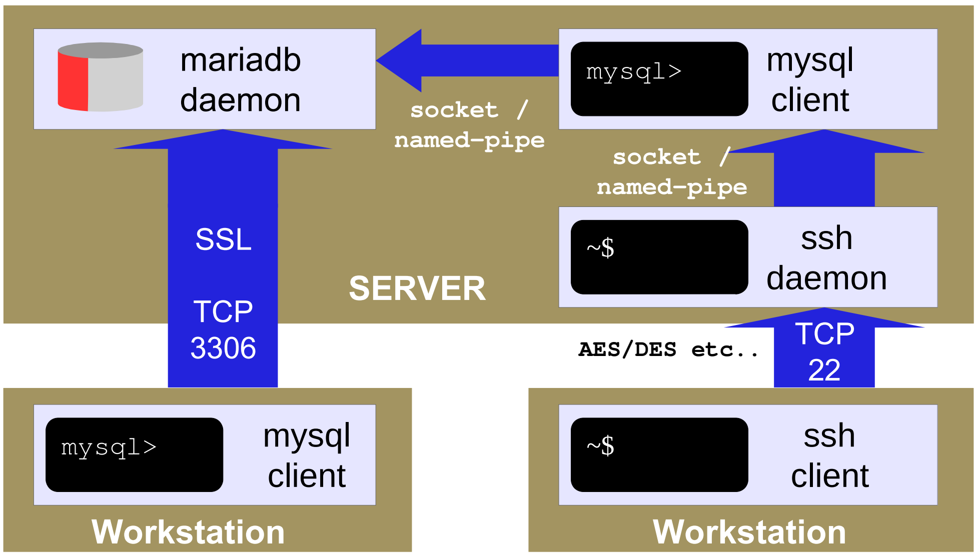


Figure 1: Accessing the MariaDB Database

The MariaDB daemon or service running on the server.

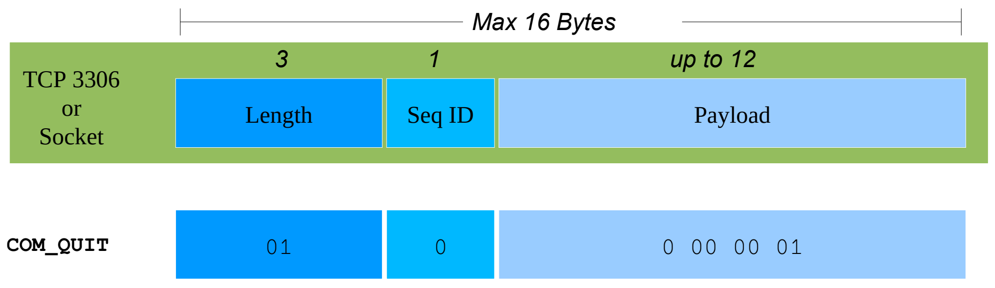
```
~$ systemctl list-units --no-pager | grep mariadb
mariadb.service
loaded active running   MariaDB 10.6.12 database server
```

The service can be accessed either:

- by an MySQL client running on another computer requiring access to the MySQL database. In this case the client will require a TCP connection via port 3306. (It is possible to define a different port). Secure Sockets Layer (SSL) security can be enabled on the server and using the **`-ssl`** switch on the client at the time of connection will impose this security protocol on the link.
- by a MySQL client running on the server. In this case the server must then be accessed via a protocol like Secure Shell (SSH). On GNU/Linux, connections are made using a Unix Socket file to the local MariaDB server daemon. The default socket file name is **`/var/run/mysql/mysqld.sock`**. On Windows, a named pipe can be used if **`-enable-named-pipe`** is configured in the options, for connections to a local MariaDB server process.

The method used is really down to the security policy and where the MySQL client will run.

2.3 MariaDB / MySQL Segment Structure



`length=0x01, sequence_id=0x00, Payload=0x01`

Figure 2: MariaDB / MySQL Segment

Communication between MariaDB Server and the MySQL Client is formatted in 16 byte segments that are carried between them using Transmission Control Protocol (TCP), a Unix Socket or a Microsoft named-pipe. The example COM_QUIT tells the server that the client wants to close the connection.

Communication types are:

`COM_SLEEP`	**`COM_QUIT`**
`COM_INIT_DB`	**`COM_QUERY`**
`COM_FIELD_LIST`	**`COM_CREATE_DB`**
`COM_DROP_DB`	**`COM_REFRESH`**
`COM_SHUTDOWN`	**`COM_STATISTICS`**
`COM_PROCESS_INFO`	**`COM_CONNECT`**
`COM_PROCESS_KILL`	**`COM_DEBUG`**
`COM_PING`	**`COM_TIME`**
`COM_DELAYED_INSERT`	**`COM_CHANGE_USER`**
`COM_DAEMON`	

Responses to communications are via Generic Response Packets:

`OK_Packet`	**`EOF_Packet`**
`ERR_Packet`	

2.4 Managing the MariaDB Service

MariaDB is managed as a service by **`systemd`** and the user is given access via **`systemctl`**, the tool that operates as the system and service manager of **`systemd`**.

2.4.1 Starting the MariaDB Service

The **`mariadb.service`** systemd service can be configured to start at boot by executing the following **`systemctl`** command:

```
~$ sudo systemctl enable mariadb.service
```

The MariaDB daemon **`mariadb.service`** can be started by executing the following **`systemctl`** command:

```
~$ sudo systemctl start mariadb.service
```

2.4.2 Stopping the MariaDB Service

The MariaDB daemon **`mariadb.service`** can be stopped by executing the following **`systemctl`** command:

```
~$ sudo systemctl stop mariadb.service
```

2.4.3 Restarting the MariaDB Service

The MariaDB daemon **`mariadb.service`** can be gracefully stopped and then restarted by executing the following **`systemctl`** command:

```
~$ sudo systemctl restart mariadb.service
```

2.4.4 Status of the MariaDB Service

The MariaDB daemon **`mariadb.service`** can be reviewed by executing the following **`systemctl`** command:

```
~$ sudo systemctl status mariadb.service
● mariadb.service - MariaDB 10.6.12 database server
  Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-12-28 09:21:08 GMT; 7min ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 760 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d
/var/run/mysqld (code=exited, status=0/SUCCESS)
    Process: 783 ExecStartPre=/bin/sh -c systemctl unset-environment
_WSREP_START_POSITION (code=exited, status=0/SUCCESS)
    Process: 822 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= ||
VAR=`cd /usr/bin/..; /usr/bin/galera_recovery`; [ $? -eq 0 ]   && systemctl set-
environment _WSREP_START_POS>
```

```
    Process: 1008 ExecStartPost=/bin/sh -c systemctl unset-environment
_WSREP_START_POSITION (code=exited, status=0/SUCCESS)
    Process: 1010 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
   Main PID: 899 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 9 (limit: 9430)
     Memory: 94.2M
        CPU: 782ms
     CGroup: /system.slice/mariadb.service
             └─899 /usr/sbin/mariadbd
```

2.5 Review the databases in the MariaDB Server

On GNU/Linux the working MySQL files and directories are in **`/var/lib/mysql/`**

```
~$ ls /var/lib/mysql
aria_log.00000001  ibdata1      multi-master.info   phpmyadmin
aria_log_control   ib_logfile0  mysql               tc.log
debian-10.3.flag   ib_logfile1  mysql_upgrade_info
ib_buffer_pool     ibtmp1       performance_schema
```

These directories represent the databases that exist on the server.

2.6 Logging into the Database for the first time

To logon to a database initially use the command:

```
~$ sudo mysql --user root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 34
Server version: 10.6.12-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

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)]>
```

2.6.1 Setup Authentication and Admin user

Authentication since MariaDB 10.4 has changed. The **`mysql_native_password`** authentication plugin is the default authentication plugin used for an account created when no authentication plugin is explicitly mentioned and **`old_passwords=0`** is set. The **`mysql_native_password`** authentication plugin is statically linked into the server, so no installation is necessary.

Check if the variable **`old_passwords`** is set (i.e. **`ON`**).

```
MariaDB [(none)]> show variables like "old_passwords";
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| old_passwords | OFF   |
+---------------+-------+
1 row in set (0.003 sec)
```

If the variable **`old_passwords`** is **`ON`**, then set it to **`0`** as follows:

```
MariaDB [(none)]> show variables like "old_passwords";
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| old_passwords | ON    |
+---------------+-------+
1 row in set (0.003 sec)

MariaDB [(none)]> set old_passwords=0;
Query OK, 0 rows affected (0.000 sec)
```

Confirm the user plugins as follows:

```
MariaDB [(none)]> SELECT User, Host, plugin FROM mysql.user;
+-------------+-----------+-----------------------+
| User        | Host      | plugin                |
+-------------+-----------+-----------------------+
| mariadb.sys | localhost | mysql_native_password |
| root        | localhost | mysql_native_password |
| mysql       | localhost | mysql_native_password |
| admin       | localhost | mysql_native_password |
| phpmyadmin  | localhost | mysql_native_password |
+-------------+-----------+-----------------------+
5 rows in set (0.004 sec)
```

2.7 Database Users

Users are stored in a **`users`** table in a special systems database called **`mysql`**. To see this add a user, list users, remove the user and list the users again.

```
MariaDB [(none)]> CREATE USER 'enguser'@'localhost' IDENTIFIED BY
'engpass';
Query OK, 0 rows affected (0.010 sec)

MariaDB [(none)]> SELECT USER FROM mysql.user;
+-------------+
| User        |
+-------------+
| admin       |
| enguser     |
| mariadb.sys |
| mysql       |
| phpmyadmin  |
| root        |
+-------------+
6 rows in set (0.001 sec)

MariaDB [(none)]> DELETE FROM mysql.user WHERE USER='enguser';
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.002 sec)
```

```
MariaDB [(none)]> SELECT USER FROM mysql.user;
+-------------+
| User        |
+-------------+
| admin       |
| mariadb.sys |
| mysql       |
| phpmyadmin  |
| root        |
+-------------+
5 rows in set (0.004 sec)
```

Now recreate the working user **`enguser`** and login to the server with that user.

```
MariaDB [(none)]> CREATE USER 'enguser'@'localhost' IDENTIFIED BY
'engpass';
```

2.7.1 Change a users password

It may be necessary to change the user password from time to time.

```
MariaDB [(none)]> SET PASSWORD FOR 'enguser'@'localhost' =
PASSWORD('newengpass');
Query OK, 0 rows affected (0.12 sec)
```

2.8 Create a Database

Create a new working database.

```
MariaDB [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| phpmyadmin         |
| sys                |
+--------------------+
5 rows in set (0.010 sec)

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

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

2.8.1 User rights

Grant user rights to the new user to specific databases.

```
MariaDB [(none)]> GRANT ALL ON Eng.* TO 'enguser'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

In another terminal login as the new user and explicitly change to the database.

```
~$ mysql --user enguser --password
Enter password: newengpass
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 38
Server version: 10.6.12-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

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)]> USE Eng;
Database changed

MariaDB [Eng]>
```

2.9 Database Tables

2.9.1 Creating Tables

Create a new table in the **`Eng`** database. Each field (column heading) is given a defined type and whether it is acceptable to leave the field **`NULL`** or not. The field identified as the primary key cannot be **`NULL`**.

```
MariaDB [Eng]> CREATE TABLE EngProject (
    -> Student_no INT NOT NULL,
    -> Username TEXT NULL,
    -> FirstName TEXT NULL,
    -> LastName TEXT NULL,
    -> Email TEXT NULL,
    -> Role TEXT NULL,
    -> PRIMARY KEY (Student_no)
    -> );
Query OK, 0 rows affected (0.058 sec)
```

Here a table was added called **`EngProject`** to the **`Eng`** database and a number of columns were added.

1. Column called **`Student_no`** with a type of **`integer (INT)`** and it is defined as '**`NOT NULL`**' which means that in each row this column must have an entry.
2. The remaining columns are simple **`TEXT`** columns.
3. A **`PRIMARY KEY`** column was also defined. A primary key is a special database table column designated to uniquely identify each table record, hence there must be records in each row and therefore this column must be defined as **`NOT NULL`**. Here it is tied to the **`Student_no`**.

2.9.2 Deleting Tables

This command will delete the table called **`EngProject`**.

```
MariaDB [Eng]> DROP TABLE EngProject;
Query OK, 0 rows affected (0.028 sec)

MariaDB [Eng]> SHOW TABLES;
Empty set (0.002 sec)
```

Recreate the table once more.

```
MariaDB [Eng]> CREATE TABLE EngProject (Student_no INT NOT NULL,
Username TEXT NULL, FirstName TEXT NULL, LastName TEXT NULL, Email
TEXT NULL, Role TEXT NULL, PRIMARY KEY (Student_no));
Query OK, 0 rows affected (0.044 sec)
```

2.9.3 List Tables

```
MariaDB [Eng]> SHOW TABLES;
+----------------+
| Tables_in_Eng |
+----------------+
| EngProject    |
+----------------+
1 row in set (0.002 sec)
```

2.9.4 Adding values to Tables

For the table already created some values are added to the rows. Here are two lines added to the table. Note: It will not accept the command if there are more entries than there are columns defined in the table. If a null entry in a column is required then simply use a double comma.

```
MariaDB [Eng]> INSERT INTO EngProject VALUES (
    -> 000000,
    -> 'alovelace',
    -> 'Ada',
    -> 'Lovelace',
    -> 'ada@lovelace.com',
    -> 'Programmer'
    -> );
Query OK, 1 row affected (0.010 sec)

MariaDB [Eng]> INSERT INTO EngProject VALUES (000001, 'cbabage',
'Charles', 'Babbage', 'charles@babbage.com', 'Hardware');
Query OK, 1 row affected (0.009 sec)
```

2.9.5 View a Table

The following command displays an ASCII graphical representation of the table contents.

```
MariaDB [Eng]> SELECT * FROM EngProject;
+------------+-----------+-----------+----------+---------------------+------------+
| Student_no | Username  | FirstName | LastName | Email               | Role       |
+------------+-----------+-----------+----------+---------------------+------------+
|          0 | alovelace | Ada       | Lovelace | ada@lovelace.com    | Programmer |
|          1 | cbabbage  | Charles   | Babage   | charles@babbage.com | Hardware   |
+------------+-----------+-----------+----------+---------------------+------------+
2 rows in set (0.002 sec)
```

2.9.6 Deleting a value from a table

To delete a row from a table use the DELETE query.

```
MariaDB [Eng]> DELETE FROM EngProject WHERE Student_no='0';
Query OK, 1 row affected (0.010 sec)
```

Check the Student number **0** is removed by performing a select all and note that only student **1** remains.

```
MariaDB [Eng]> SELECT * FROM EngProject;
+------------+-----------+-----------+----------+----------------------+---------+
| Student_no | Username  | FirstName | LastName | Email                | Role    |
+------------+-----------+-----------+----------+----------------------+---------+
|          1 | cbabbage  | Charles   | Babage   | charles@babbage.com  | Hardware |
+------------+-----------+-----------+----------+----------------------+---------+
1 row in set (0.00 sec)
```

Add the user back into the database before progressing and check the table has both entries.

```
MariaDB [Eng]> INSERT INTO EngProject VALUES (000000, 'alovelace',
'Ada', 'Lovelace', 'ada@lovelace.com', 'Programmer');
Query OK, 1 row affected (0.00 sec)
```

```
MariaDB [Eng]> SELECT * FROM EngProject;
+------------+-----------+-----------+----------+---------------------+------------+
| Student_no | Username  | FirstName | LastName | Email               | Role       |
+------------+-----------+-----------+----------+---------------------+------------+
|          0 | alovelace | Ada       | Lovelace | ada@lovelace.com    | Programmer |
|          1 | cbabbage  | Charles   | Babage   | charles@babbage.com | Hardware   |
+------------+-----------+-----------+----------+---------------------+------------+
2 rows in set (0.002 sec)
```

2.9.7 Ordering SQL query output

In this example a student name list is extracted from the database.

```
MariaDB [Eng]> SELECT FirstName, LastName FROM EngProject;
+-----------+----------+
| FirstName | LastName |
+-----------+----------+
| Ada       | Lovelace |
| Charles   | Babage   |
+-----------+----------+
2 rows in set (0.002 sec)
```

To order the list alphabetically by the **LastName**.

```
MariaDB [Eng]> SELECT FirstName, LastName FROM EngProject ORDER BY
LastName;
+-----------+----------+
| FirstName | LastName |
+-----------+----------+
| Charles   | Babage   |
| Ada       | Lovelace |
+-----------+----------+
2 rows in set (0.002 sec)
```

2.9.8 Altering a value

Note the spelling mistake on Charles Babbage's **LastName**. To fix this use the **UPDATE** query as follows.

```
MariaDB [Eng]> UPDATE EngProject SET LastName='Babbage' WHERE
Student_no='1';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

To check the effect made with the update select everything (*****) from the table where the primary column value is '**1**'.

```
MariaDB [Eng]> SELECT * FROM EngProject;
+------------+-----------+-----------+----------+----------------------+----------+
| Student_no | Username  | FirstName | LastName | Email                | Role     |
+------------+-----------+-----------+----------+----------------------+----------+
|          1 | cbabbage  | Charles   | Babbage  | charles@babbage.com  | Hardware |
+------------+-----------+-----------+----------+----------------------+----------+
1 row in set (0.00 sec)
```

2.9.9 Counting Data

Data in databases can be counted, here are a number of examples, initially counting the number of row in the database, then counting the number of **LastName** entries and finally counting the **LastName** and **Email** columns.

```
MariaDB [Eng]> SELECT COUNT(*) FROM EngProject;
+----------+
| COUNT(*) |
+----------+
|        2 |
+----------+
1 row in set (0.00 sec)
```

```
MariaDB [Eng]> SELECT COUNT(*) FROM EngProject GROUP BY LastName;
+----------+
| COUNT(*) |
+----------+
|        1 |
|        1 |
+----------+
2 rows in set (0.00 sec)


MariaDB [Eng]> SELECT LastName, Email, COUNT(*) FROM EngProject
GROUP BY LastName, Email;
+----------+---------------------+----------+
| LastName | Email               | COUNT(*) |
+----------+---------------------+----------+
| Babbage  | charles@babbage.com |        1 |
| Lovelace | ada@lovelace.com    |        1 |
+----------+---------------------+----------+
2 rows in set (0.00 sec)
```

2.10 Joining data from different tables tables

Generate a new table as follows:

```
MariaDB [Eng]> CREATE TABLE EngHobbies (Student_no INT NOT NULL,
Hobbies TEXT NULL, PRIMARY KEY (Student_no));
Query OK, 0 rows affected (0.053 sec)

MariaDB [Eng]> INSERT INTO EngHobbies VALUES (000000, 'camogie,
cards');
Query OK, 1 row affected (0.017 sec)

MariaDB [Eng]> INSERT INTO EngHobbies VALUES (000001, 'football,
darts');
Query OK, 1 row affected (0.022 sec)

MariaDB [Eng]> SELECT * FROM EngHobbies;
+------------+-----------------+
| Student_no | Hobbies         |
+------------+-----------------+
|          0 | camogie, cards  |
|          1 | football, darts |
+------------+-----------------+
2 rows in set (0.002 sec)
```

Data can be extracted from multiple tables linked by a common field.

```
 PRIMARY KEY                                 EngProject
+------------+-----------+-----------+----------+---------------------+------------+
| Student_no | Username  | FirstName | LastName | Email               | Role       |
+------------+-----------+-----------+----------+---------------------+------------+
|          0 | alovelace | Ada       | Lovelace | ada@lovelace.com    | Programmer |
|          1 | cbabbage  | Charles   | Babbage  | charles@babbage.com | Hardware   |
+------------+-----------+-----------+----------+---------------------+------------+


+------------+-----------------+
| Student_no | Hobbies         |
+------------+-----------------+  EngHobbies
|          0 | camogie, cards  |
|          1 | football, darts |
+------------+-----------------+
```

Figure 3: Linking tables via Primary Key

As displayed in Figure 3 the two tables have a **`Student_no`** column that is common to both. Data can be joined from one field to data in another field within an SQL query. The primary key in the first table is linked as a foreign key in the second table. A foreign key is a column that is used to establish and enforce a link between the data in two tables.

```
MariaDB [Eng]> SELECT a.LastName, b.Hobbies
            -> FROM  EngProject a INNER JOIN EngHobbies b
            -> ON a.Student_no = b.Student_no;
+----------+-----------------+
| LastName | Hobbies         |
+----------+-----------------+
| Lovelace | camogie, cards  |
| Babbage  | football, darts |
+----------+-----------------+
2 rows in set (0.003 sec)

MariaDB [Eng]> SELECT a.LastName, b.Hobbies
            -> FROM  EngProject a INNER JOIN EngHobbies b
            -> ON a.Student_no = b.Student_no
            -> WHERE a.Student_no = 0;
+----------+----------------+
| LastName | Hobbies        |
+----------+----------------+
| Lovelace | camogie, cards |
+----------+----------------+
1 row in set (0.002 sec)

MariaDB [Eng]> SELECT b.Hobbies
            -> FROM  EngProject a INNER JOIN EngHobbies b
            -> ON a.Student_no = b.Student_no
            -> WHERE a.FirstName = 'ada';
+----------------+
| Hobbies        |
+----------------+
| camogie, cards |
+----------------+
1 row in set (0.002 sec)
```

2.11 Saving Databases

To save a database use the **`mysqldump`** command. This produces a dump file that can be stored off system and a new system can be built from the file.

```
~$ mysqldump –u user –p db_name > File_name.sql
Enter password: <password>
```

Here is an example from the topic participants data set. No password is necessary as the command is being ran with root privileges.

```
~$ sudo mysqldump –u root Eng > EngProject.sql

~$ ls
EngProject.sql
```

This produces a dump file that looks like this.

```
~$ cat EngProject.sql
-- MySQL dump 10.19  Distrib 10.3.29-MariaDB, for debian-linux-gnueabihf
(armv7l)
--
-- Host: localhost    Database: Eng
-- ------------------------------------------------------
-- Server version      10.3.29-MariaDB-0+deb10u1

/*!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'
*/;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `EngHobbies`
--

DROP TABLE IF EXISTS `EngHobbies`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `EngHobbies` (
  `Student_no` int(11) NOT NULL,
  `Hobbies` text DEFAULT NULL,
  PRIMARY KEY (`Student_no`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `EngHobbies`
--

LOCK TABLES `EngHobbies` WRITE;
/*!40000 ALTER TABLE `EngHobbies` DISABLE KEYS */;
INSERT  INTO  `EngHobbies`  VALUES  (0,'camogie,  cards'),(1,'football,
darts');
/*!40000 ALTER TABLE `EngHobbies` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `EngProject`
--

DROP TABLE IF EXISTS `EngProject`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `EngProject` (
  `Student_no` int(11) NOT NULL,
  `Username` text DEFAULT NULL,
  `FirstName` text DEFAULT NULL,
  `LastName` text DEFAULT NULL,
  `Email` text DEFAULT NULL,
  `Role` text DEFAULT NULL,
  PRIMARY KEY (`Student_no`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `EngProject`
--

LOCK TABLES `EngProject` WRITE;
/*!40000 ALTER TABLE `EngProject` DISABLE KEYS */;
INSERT              INTO              `EngProject`              VALUES
(0,'alovelace','Ada','Lovelace','ada@lovelace.com','Programmer'),
(1,'cbabbage','Charles','Babbage','charles@babbage.com','Hardware');
/*!40000 ALTER TABLE `EngProject` 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 */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2021-08-26 13:34:30
```

2.12 Recover Databases

To recover a database or to build a new database server from the dump file use the **`mysql`** command and redirect the dump file into it.

```
~$ sudo mysql –u user –p DB_name < file_name.sql
```

First lets drop the **`Eng`** database and confirm it is gone. Login as the MariaDB **`root`** user and drop the database.

```
~$ sudo mysql -u root

MariaDB [(none)]> DROP DATABASE Eng;
Query OK, 2 rows affected (0.058 sec)

MariaDB [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| phpmyadmin         |
| sys                |
+--------------------+
5 rows in set (0.000 sec)
```

Now rebuild the **`Eng`** database from the dump file.

```
~$ sudo mysql –u root

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

MariaDB [(none)]> exit;
Bye

~$ sudo mysql –u root Eng < EngProject.sql


~$ mysql -u enguser -p Eng
Enter password: newengpass
```

```
MariaDB [Eng]> SELECT * FROM EngProject;
+------------+-----------+-----------+----------+---------------------+------------+
| Student_no | Username  | FirstName | LastName | Email               | Role       |
+------------+-----------+-----------+----------+---------------------+------------+
|          0 | alovelace | Ada       | Lovelace | ada@lovelace.com    | Programmer |
|          1 | cbabbage  | Charles   | Babbage  | charles@babbage.com | Hardware   |
+------------+-----------+-----------+----------+---------------------+------------+
2 rows in set (0.002 sec)
```

The database `Eng` has been recovered from the dump file.

3. Exercise Laboratory #1

3.1 Create a Database

- Create a database using MariaDB with the following data.
- Database name – Counties
- Table 1 – County Capital, County highest mountain, County main river.
  - Populate all County rows with the county names but with blank values in the county capital, mountain and river fields except for Limerick, Clare and Tipperary where the full rows should be filled.
- Table 2 – Sport of prominence in the counties, allow for two sports per county.
  - Again populate all entries with blank values except Limerick, Clare and Tipperary.
- Perform SQL query that returns all counties whether having blank values or not in alphabetical order
- Document each stage.
-

NOTE: Use county name as the Primary Key in each database

Notes:

This page is intentionally blank