Laboratory 11: basic MariaDB

This exercise is designed to teach MariaDB basics using a single database table. This is an unusually simplistic database design—it is storage (and speed) inefficient and difficult to query. In the next laboratory exercise, you will transform this simple table into a more complex (and realistic) normalized multi–table database. The data that you will analyze were assembled by the Herbier National du Gabon (Obiang et al. 2019) from a series of vegetation plots maintained by Institut de Recherche en Ecologie Tropicale (IRET), Herbier National du Gabon (LBV), and the Smithsonian Institution (US).

Tasks

(1) Install and configure MariaDB.

   (a) Install MariaDB by typing `sudo apt install mariadb-server` in the terminal. Enter your password and agree to the install[1].

   (b) After the MariaDB install is complete, login to the MariaDB client as the root user by typing `sudo mariadb -u root mysql` in the terminal. Enter your password if/when prompted. Answer question (1).

   (c) It is best to avoid issuing commands as root. Rather it is better to use a limited–power user for most tasks (it minimizes the damage one can accidentally do). Create a new general purpose user by typing `CREATE USER IF NOT EXISTS 'working'@'localhost' IDENTIFIED BY 'working';` into the terminal. It is advisable to use a better password than 'working', so replace this password with something of your choosing. Remember a good password will be longer than eight characters, not be a dictionary word, not a known exposed password (https://haveibeenpwned.com/Passwords), and use a mix of uppercase, lowercase, numbers, and symbols. Answer question (2).

   (d) To determine which privileges the 'working' user has, type `SELECT * FROM user WHERE User = 'working';` into the terminal. Answer question (3).

   (e) To modify the 'working' user's privileges type `GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, FILE ON *.* TO 'working'@'localhost';` into the terminal. Answer question (4).

   (f) Once the GRANT statement has been executed without errors, type `FLUSH PRIVILEGES;` into the terminal. Answer question (5).

   (g) Allow MariaDB to load from local data files by typing `SET GLOBAL local_infile = true;` in the terminal.

   (h) Open another terminal window and attempt to login using your newly created 'working' user by typing `mariadb -u working -p` in the terminal. If you are successful, go back to the first terminal (root login) and logout by typing `EXIT` in the terminal.

(2) Create a MariaDB database and table to store the data.

   (a) Make sure that you are using the terminal logged into MariaDB as the 'working' user. Type `CREATE DATABASE lab11;` into the terminal. Make sure there are no error messages.

[1] The default configuration of MariaDB is hobbled by severely restricting hardware usage. We will leave the default in place, but the configuration should be changed for serious work.

(b) Type `USE lab11;` into the terminal. Make sure there are no error messages.

(c) Type the following into the terminal (and then answer question (6)):

```
CREATE TABLE `gabon` (
`gbifID` INT UNSIGNED NOT NULL AUTO_INCREMENT,
`occurrenceID` VARCHAR(16) NOT NULL,
`individualCount` SMALLINT UNSIGNED NOT NULL,
`eventDate` DATE NOT NULL,
`samplingProtocol` VARCHAR(128) NOT NULL,
`decimalLatitude` VARCHAR(16) NOT NULL,
`decimalLongitude` VARCHAR(16) NOT NULL,
`scientificName` VARCHAR(128) NOT NULL,
`family` VARCHAR(32) NOT NULL,
PRIMARY KEY (`gbifID`)
) ENGINE=InnoDB DEFAULT CHARSET=UTF8;
```

(d) Type `SHOW FIELDS FROM gabon;` in the terminal to confirm that the table was created properly.

(e) Type `EXIT` in the terminal.

(3) Download and prepare the data for analysis.

(a) Download the compressed Darwin Core Archive dataset by typing `wget https://api.gbif.org/v1/occurrence/download/request/0132307-230224095556074.zip` in a bash terminal.

(b) To convert the Darwin Core Archive to a format more ingestible by MariaDB type `unzip -c 0132307-230224095556074.zip occurrence.txt | tail +4 | awk -F'\t' -v OFS='\t' '{if(!length($73)){$73=1};sub(/T00:00:00$/,"",$103);if(length($1)&&length($68)&&length($103)&&length($112)&&length($138)&&length($139)&&length($189)&&length($201)){print $1,$68,$73,$103,$112,$138,$139,$189,$201}}' | mariadb --local-infile -u working -p -e "LOAD DATA LOCAL INFILE '/dev/stdin' INTO TABLE gabon" lab11` into the terminal. Type your password when prompted. Answer question (7).

(4) Login into MariaDB as the 'working' user and query the database.

(a) To check that the records were processed correctly, each column should be compared to the downloaded Darwin Core Archive. For example, type `SELECT COUNT(gbifID) AS count FROM gabon;` into MariaDB, the query should return 16,664 records—the same number of lines output by awk in step (3)(b).

(b) The type of checking that is most appropriate varies by column data type. For example, type `SELECT MIN(individualCount) as min, MAX(individualCount) AS max, AVG(individualCount) as mean, VARIANCE(individualCount) as var FROM gabon;` in MariaDB the query should return 1, 17,108, 13.4536, and 45,276.5931—the same values that datamash computes from the awk output in step (3)(b).

(c) To check a VARCHAR column, one can either evaluate the unique entries individually, or by using summary values. For example, individual entries can be compared for samplingProtocol by typing `SELECT DISTINCT(samplingProtocol) FROM gabon;` into MariaDB, the query should return 16 rows that are identical to the awk output in step (3)(b).

(d) To check summary values of a VARCHAR convert the text to numbers and then process the numbers like a numeric column. For example type `SELECT MIN(LENGTH(family)) as min, MAX(LENGTH(family)) AS max, AVG(LENGTH(family)) as mean, VARIANCE(LENGTH(family)) as var FROM gabon;` into MariaDB. The query should return 7, 17, 10.8191, 5.1849—the same values that datamash can compute from the awk output in step (3)(b). Answer question (8).

(e) To find the number of total records, tree species, and individuals observed in random plots by year over the last 10 years, type `SELECT YEAR(eventDate) as year, COUNT(*) as records, COUNT(DISTINCT(scientificName)) AS species, SUM(individualCount) as individuals FROM gabon WHERE samplingProtocol LIKE '%random plots%' AND (samplingProtocol LIKE '%5 cm%' OR samplingProtocol LIKE '%10 cm%') AND eventDate > (NOW() - INTERVAL 10 YEAR) GROUP BY YEAR(eventDate);` in MariaDB.

(f) To find the five most commonly observed tree species in random plots, type `SELECT family, scientificName, SUM(individualCount) as individuals FROM gabon WHERE samplingProtocol LIKE '%random plots%' AND (samplingProtocol LIKE '%5 cm%' OR samplingProtocol LIKE '%10 cm%') GROUP BY scientificName ORDER BY individuals DESC LIMIT 5;` in MariaDB. Answer question (9).

Questions (https://forms.gle/B8U963mrTy34pPM8A)

(1) For task (1)(b):

(a) What does each of the arguments used to start MariaDB client do?

(b) Why is there no password for the MariaDB root user?

(2) For task (1)(c), what does each part of the create statement do?

(3) For task (1)(d), what privileges does the user 'working' have by default?

(4) For task (1)(e):

(a) What do these privileges do?

(b) Are they necessary for just accessing data?

(5) For task (1)(f), what does the FLUSH command do?

(6) For task (2)(c):

(a) How many records will the table be able to store?

(b) What is the maximum individualCount that can be stored?

(c) How many characters can be stored in scientificName?

(d) What is the maximum scientificName length in the dataset?

(7) For task (3)(b), explain what each step of the text conversion does.

(8) For task (4)(d), write a SQL query to check the data in the occurrenceID field.

(9) For task (4)(f):

(a) What are the top 5 species?

(b) What does the 'ORDER BY' statement do?

(c) What does the 'GROUP BY' statement do?

(d) Would the query work without the 'AS' statement? Please explain.

Literature cited

Obiang, N. E., A. Ngomanda & D. Kenfack. 2019. Vegetation assessment and forest dynamic study of various areas in Gabon from 2000 to 2018. Herbier National du Gabon (https://doi.org/10.15468/i8fwlf).

Due at the start of class April 18.