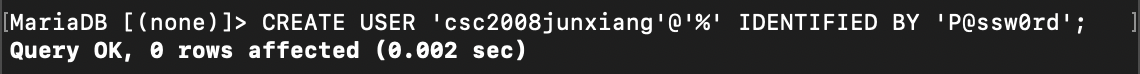
**Question 1: Create new MariaDB user that applies to every IP address with a password**

Command used:

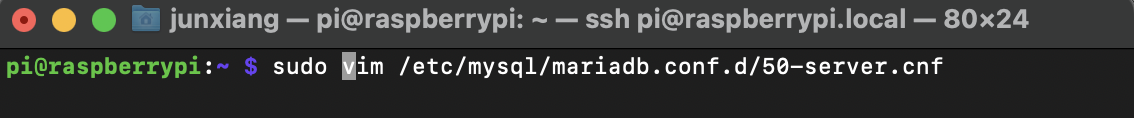
1. CREATE USER ‘csc2008junxiang’@’%’ IDENTIFIED BY ‘P@ssw0rd’  
   

**Question 2: Grant privileges to the user that was just created**

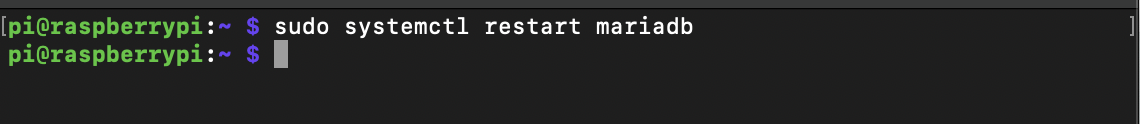
1. GRANT ALL PRIVILEGES ON \*.\* TO ‘csc2008junxiang’@’%’ IDENTIFIED BY ‘P@ssw0rd’;  
   Text

   Description automatically generated

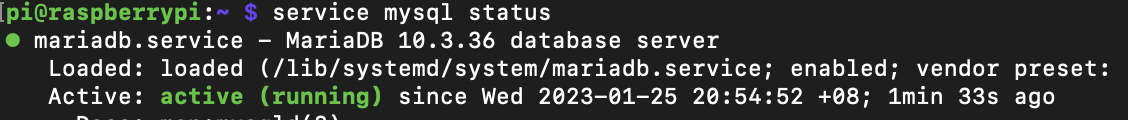
**Question 3: Update range of IP addresses to listen**

1. sudo vim /etc/mysql/mariadb.conf.d/50-server.cnf  
   
2. Vim has opened, scroll down to bind-address  
   Text

   Description automatically generated
3. Press ‘i’ on keyboard to activate the insert mode. Change the bind-address to **0.0.0.0.** Then press ESC on your keyboard, then type colon w q on your keyboard to write and quit.  
     
   :wq  
   Text

   Description automatically generated
4. Restart MariaDB Server for the changes to be applied.  
   sudo systemctl restart mariadb  
   

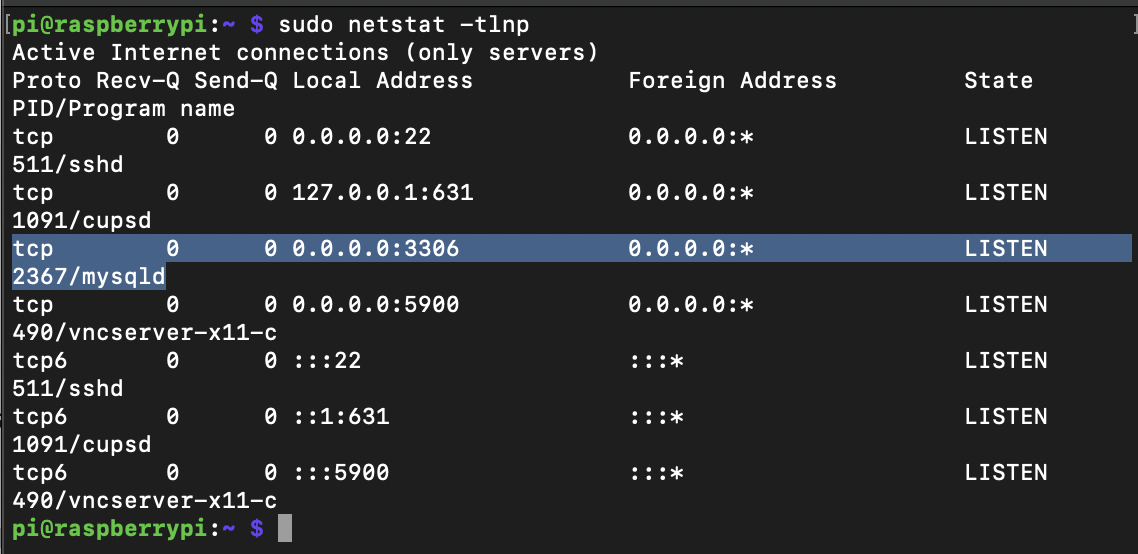
**Question 4: Double check raspberry pi database is running**

1. service mysql status  
   
2. mysql -ucsc2008junxiang -pP@ssw0rd  
   Text

   Description automatically generated
3. ifconfig  
   Text

   Description automatically generated

**Question 5: Check port number for the raspberrypi MariaDB Environment**

1. sudo netstat -tlnp  
     
   mysql is using port 3306 as highlighted in the screenshot.  
   
2. SHOW GLOBAL VARIABLES LIKE ‘PORT’;  
     
   MySQL MariaDB is using port 3306.  
   Text

   Description automatically generated

**Question 6: Open MySQL Workbench**

1. MySQL Workbench opened on MacOS  
   Graphical user interface, text

   Description automatically generated

**Question 7: Set up new connection to MySQL Workbench with Raspberry Pi’s information**

1. Connection Name: Raspberry Pi’s MariaDB  
   Hostname: 192.168.0.71  
   Port : 3306  
   A screenshot of a computer

   Description automatically generated

**Question 8: Check if connection has been successfully established**

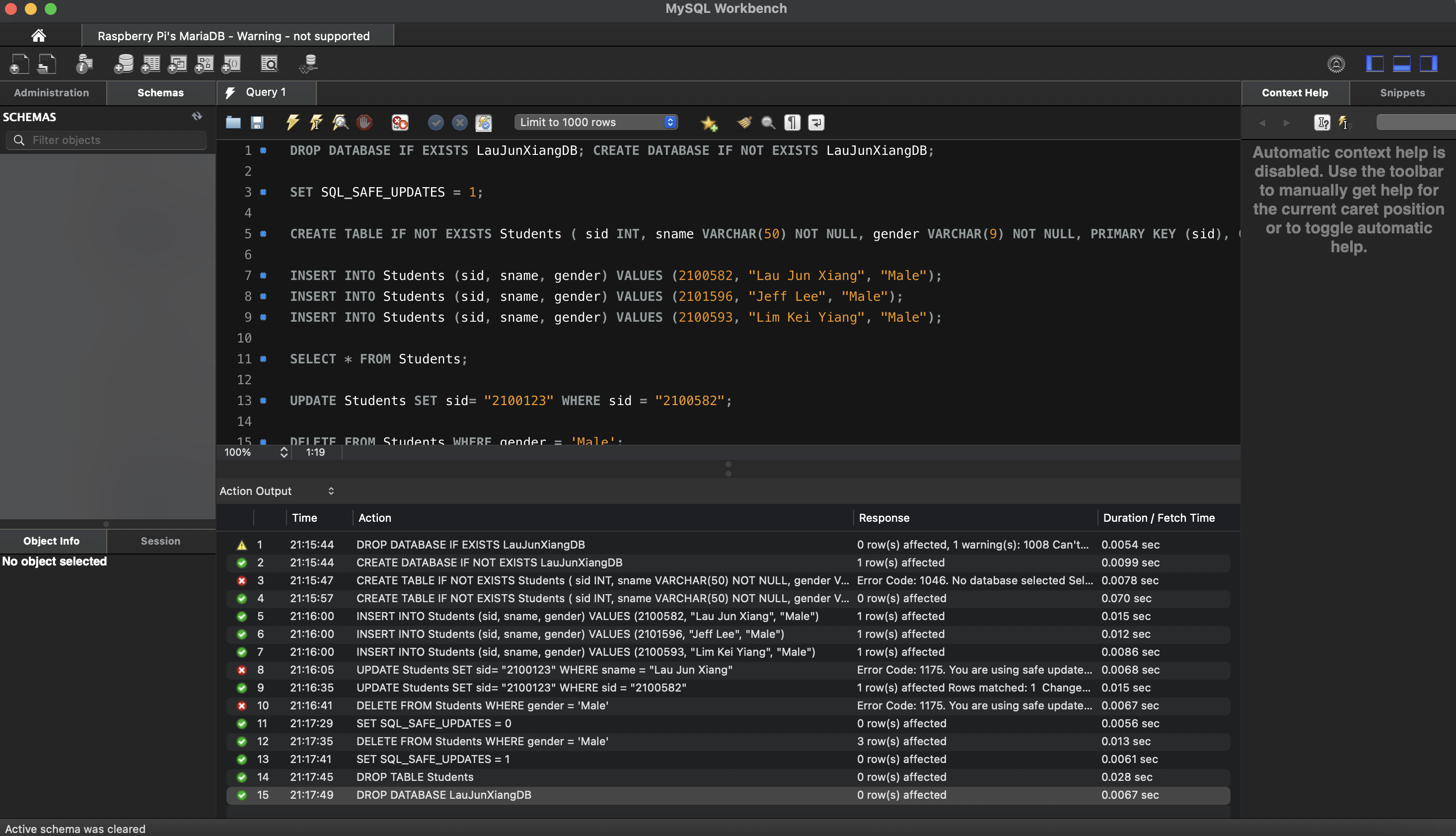
1. MySQL Workbench has been connected to the specified connection for Raspberry Pi.  
   A screenshot of a computer

   Description automatically generated with medium confidence

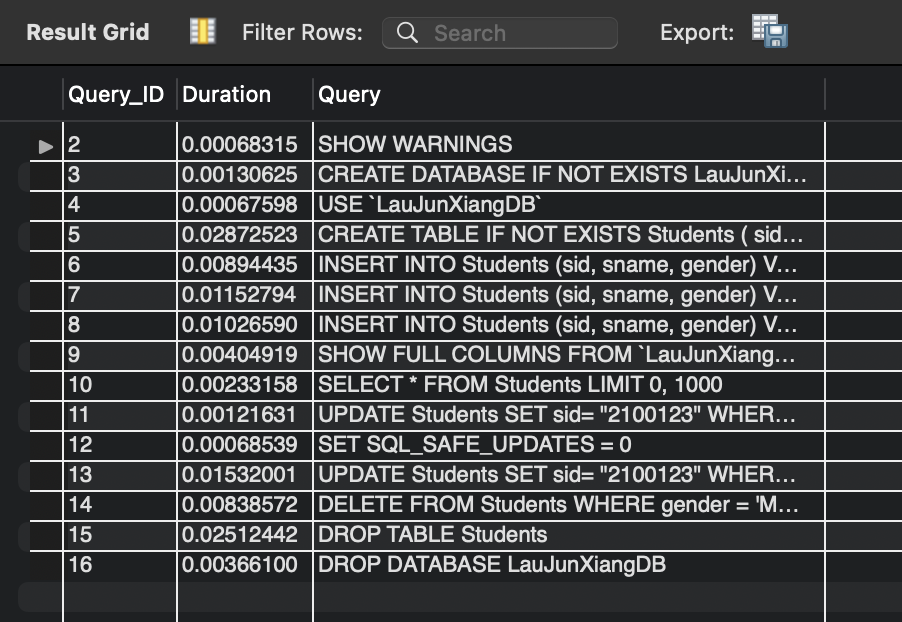
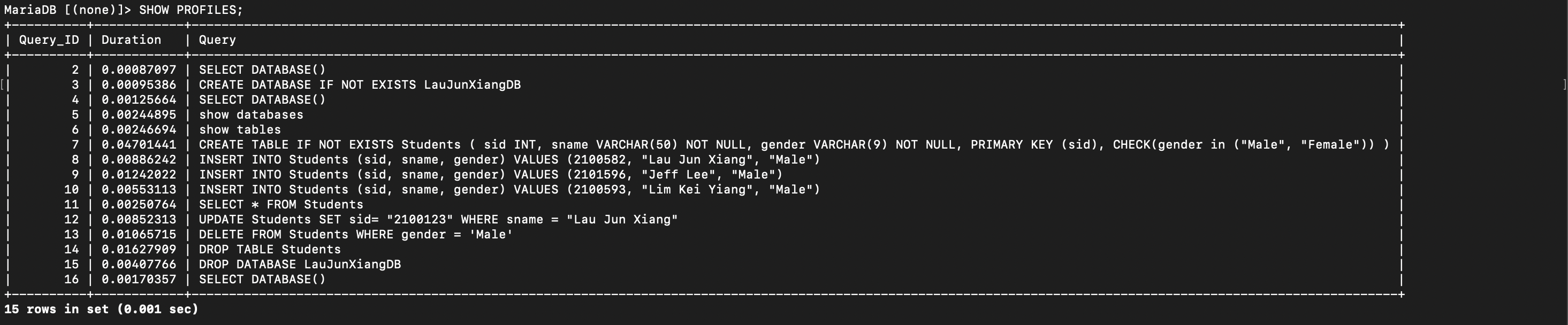
1. More details that can be seen, such as the MariaDB account created in Raspberry Pi.  
   Graphical user interface, application

   Description automatically generated

**Question 9: In MySQL Workbench, perform basic SQL Operations tested in previous lab.**

1. Do some basic commands such as  
     
   DROP DATABASE IF EXISTS LauJunXiangDB; CREATE DATABASE IF NOT EXISTS LauJunXiangDB;  
     
   SET SQL\_SAFE\_UPDATES = 1;  
     
   CREATE TABLE IF NOT EXISTS Students ( sid INT, sname VARCHAR(50) NOT NULL, gender VARCHAR(9) NOT NULL, PRIMARY KEY (sid), CHECK(gender in ("Male", "Female")) );  
     
   INSERT INTO Students (sid, sname, gender) VALUES (2100582, "Lau Jun Xiang", "Male"); INSERT INTO Students (sid, sname, gender) VALUES (2101596, "Jeff Lee", "Male"); INSERT INTO Students (sid, sname, gender) VALUES (2100593, "Lim Kei Yiang", "Male");  
     
   UPDATE Students SET sid= "2100123" WHERE sid = "2100582";  
     
   DELETE FROM Students WHERE gender = 'Male';  
     
   DROP TABLE Students;  
     
   DROP DATABASE LauJunXiangDB;  
   

**Question 10: Perform some test and profile analysis**

1. On MySQL Workbench (on My MacOS)  
     
   SET PROFILING = 1;  
     
   DROP DATABASE IF EXISTS LauJunXiangDB; CREATE DATABASE IF NOT EXISTS LauJunXiangDB;  
   SET SQL\_SAFE\_UPDATES = 1;  
     
   CREATE TABLE IF NOT EXISTS Students ( sid INT, sname VARCHAR(50) NOT NULL, gender VARCHAR(9) NOT NULL, PRIMARY KEY (sid), CHECK(gender in ("Male", "Female")) );  
     
   INSERT INTO Students (sid, sname, gender) VALUES (2100582, "Lau Jun Xiang", "Male"); INSERT INTO Students (sid, sname, gender) VALUES (2101596, "Jeff Lee", "Male"); INSERT INTO Students (sid, sname, gender) VALUES (2100593, "Lim Kei Yiang", "Male");  
     
   UPDATE Students SET sid= "2100123" WHERE sname = "Lau Jun Xiang";  
   DELETE FROM Students WHERE gender = 'Male';  
   DROP TABLE Students;  
   DROP DATABASE LauJunXiangDB;  
   SHOW PROFILES;  
   
2. On Raspberry Pi:  
     
   SET PROFILING = 1;  
   DROP DATABASE IF EXISTS LauJunXiangDB; CREATE DATABASE IF NOT EXISTS LauJunXiangDB;  
     
   USE LauJunXiangDB;  
     
   CREATE TABLE IF NOT EXISTS Students ( sid INT, sname VARCHAR(50) NOT NULL, gender VARCHAR(9) NOT NULL, PRIMARY KEY (sid), CHECK(gender in ("Male", "Female")) );  
     
   INSERT INTO Students (sid, sname, gender) VALUES (2100582, "Lau Jun Xiang", "Male"); INSERT INTO Students (sid, sname, gender) VALUES (2101596, "Jeff Lee", "Male"); INSERT INTO Students (sid, sname, gender) VALUES (2100593, "Lim Kei Yiang", "Male");  
     
   UPDATE Students SET sid= "2100123" WHERE sname = "Lau Jun Xiang";  
     
   DELETE FROM Students WHERE gender = 'Male';  
     
   DROP TABLE Students;  
     
   DROP DATABASE LauJunXiangDB;  
     
   SHOW PROFILES;  
   

Conclusion for Profiling:

Some SQL Statements in the Raspberry Pi were faster than in MySQL Workbench. Typically, remote access should be slower.