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’

```
MariaDB [(none)]> CREATE USER 'csc2008junxiang'@'%' IDENTIFIED BY 'P@ssw0rd';
Query OK, 0 rows affected (0.002 sec)
```

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

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

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON *.* TO 'csc2008junxiang'@'%' IDENTIFIE
D BY 'P@ssw0rd';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> flush privileges
    -> ;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]>
```

Question 3: Update range of IP addresses to listen

1. sudo vim /etc/mysql/mariadb.conf.d/50-server.cnf

```
junxiang — pi@raspberrypi: ~ — ssh pi@raspberrypi.local — 80×24
pi@raspberrypi:~ $ sudo vim /etc/mysql/mariadb.conf.d/50-server.cnf
```

2. Vim has opened, scroll down to bind-address

```
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 127.0.0.1
```

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

```
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address            = 127.0.0.1
bind-address = 0.0.0.0
#
# * Fine Tuning
#
#key_buffer_size        = 16M
#max_allowed_packet     = 16M
#thread_stack           = 192K
#thread_cache_size      = 8
# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
:wq
```

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

```
pi@raspberrypi:~ $ sudo systemctl restart mariadb
pi@raspberrypi:~ $
```

Question 4: Double check raspberry pi database is running

1. service mysql status

```
pi@raspberrypi:~ $ service mysql status
● mariadb.service - MariaDB 10.3.36 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset:
   Active: active (running) since Wed 2023-01-25 20:54:52 +08; 1min 33s ago
```

2. mysql -ucsc2008junxiang -pP@ssw0rd

```
pi@raspberrypi:~ $ mysql -ucsc2008junxiang -pP@ssw0rd
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 10
Server version: 10.3.36-MariaDB-0+deb10u2 Raspbian 10

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)]>
```

3. ifconfig

```
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.71  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::9574:8ff3:5ed0:5ece  prefixlen 64  scopeid 0x20<link>
        ether e4:5f:01:02:c4:41  txqueuelen 1000  (Ethernet)
        RX packets 25258  bytes 8888334 (8.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16691  bytes 5790790 (5.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

pi@raspberrypi:~ $
```

Question 5: Check port number for the raspberrypi MariaDB Environment

1. sudo netstat -tlnp

   mysql is using port 3306 as highlighted in the screenshot.

```
pi@raspberrypi:~ $ sudo netstat -tlnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
511/sshd
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
1091/cupsd
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN
2367/mysqld
tcp        0      0 0.0.0.0:5900            0.0.0.0:*               LISTEN
490/vncserver-x11-c
tcp6       0      0 :::22                   :::*                    LISTEN
511/sshd
tcp6       0      0 ::1:631                 :::*                    LISTEN
1091/cupsd
tcp6       0      0 :::5900                 :::*                    LISTEN
490/vncserver-x11-c
pi@raspberrypi:~ $
```

2. SHOW GLOBAL VARIABLES LIKE ‘PORT’;

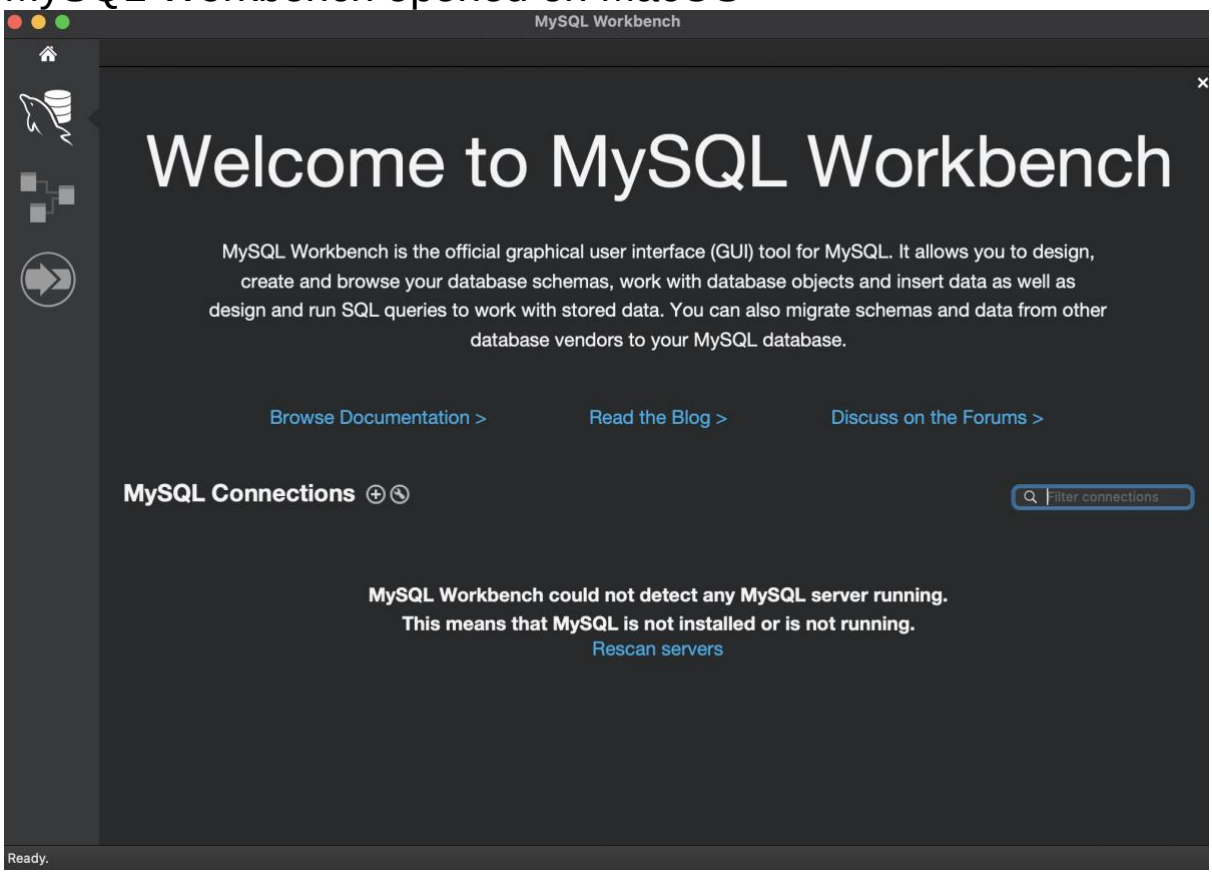
   MySQL MariaDB is using port 3306.

```
MariaDB [(none)]> SHOW GLOBAL VARIABLES LIKE 'PORT';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| port          | 3306  |
+---------------+-------+
1 row in set (0.004 sec)

MariaDB [(none)]>
```

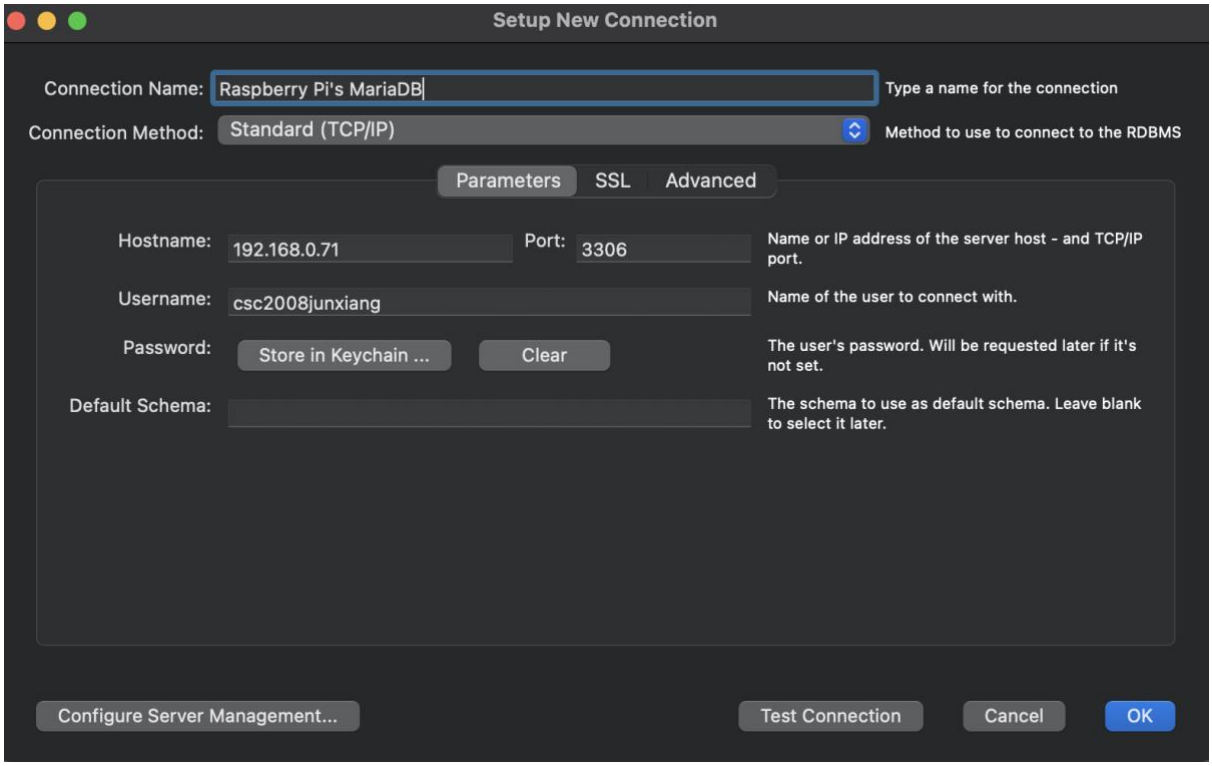
Question 6: Open MySQL Workbench

1. MySQL Workbench opened on MacOS



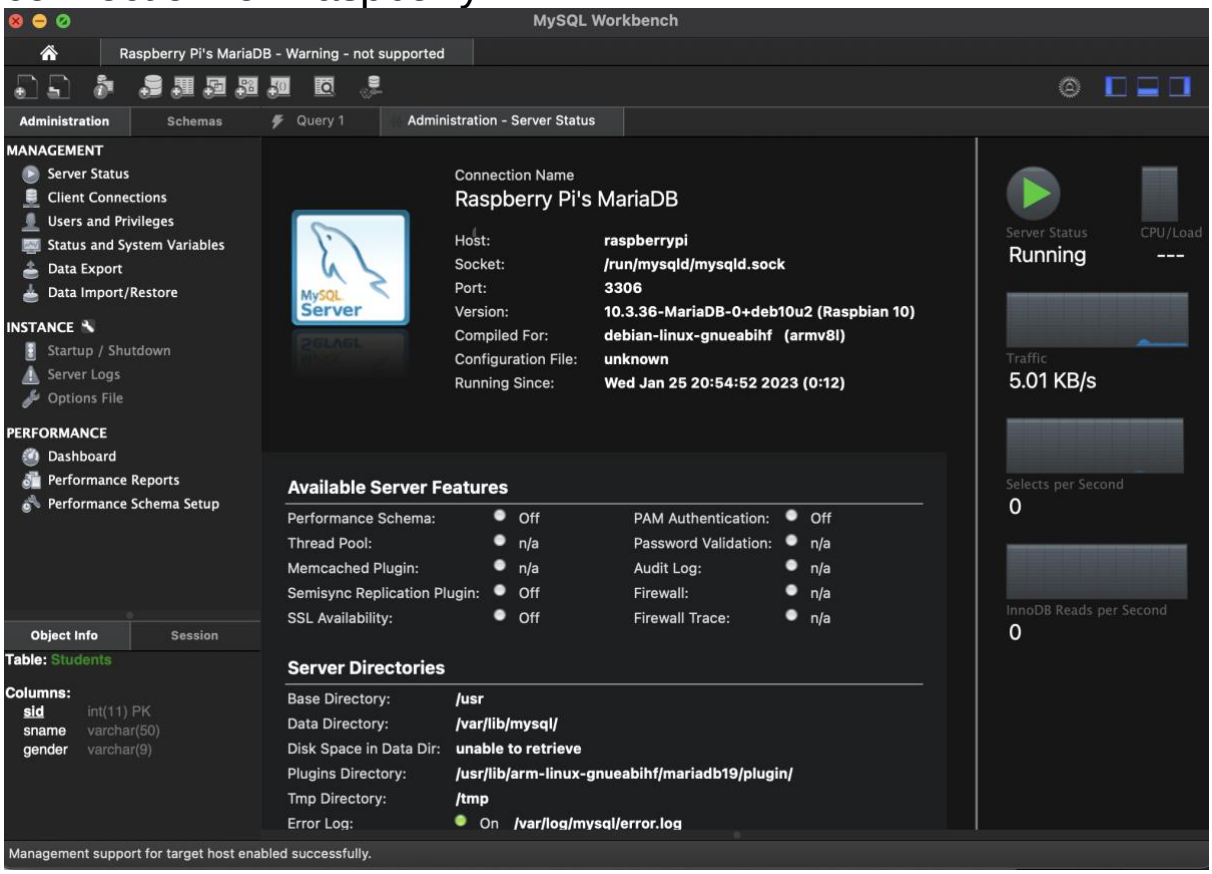
Question 7: Set up new connection to MySQL Workbench with Raspberry Pi's information

1. Connection Name: Raspberry Pi's MariaDB
   Hostname: 192.168.0.71
   Port : 3306

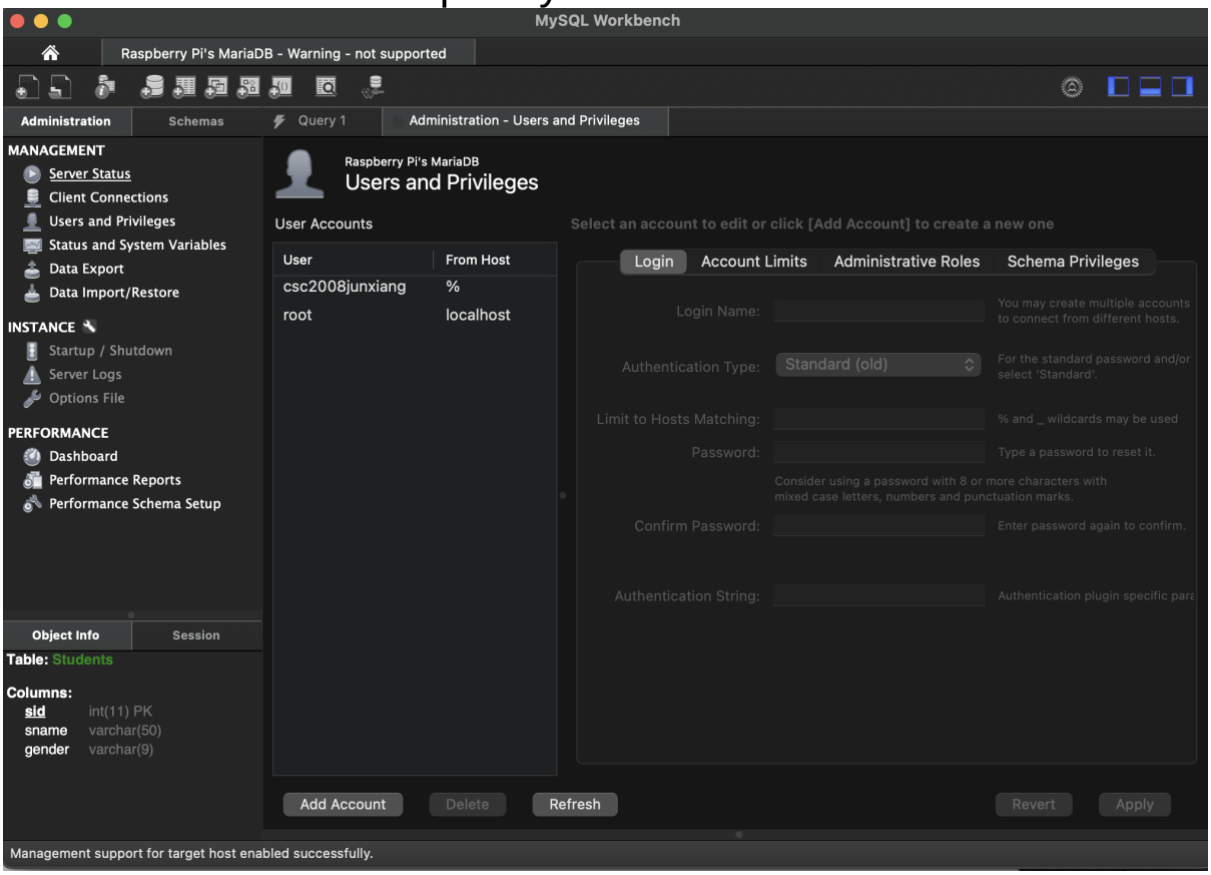


Question 8: Check if connection has been successfully established

1. MySQL Workbench has been connected to the specified connection for Raspberry Pi.



2. More details that can be seen, such as the MariaDB account created in Raspberry Pi.



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")));

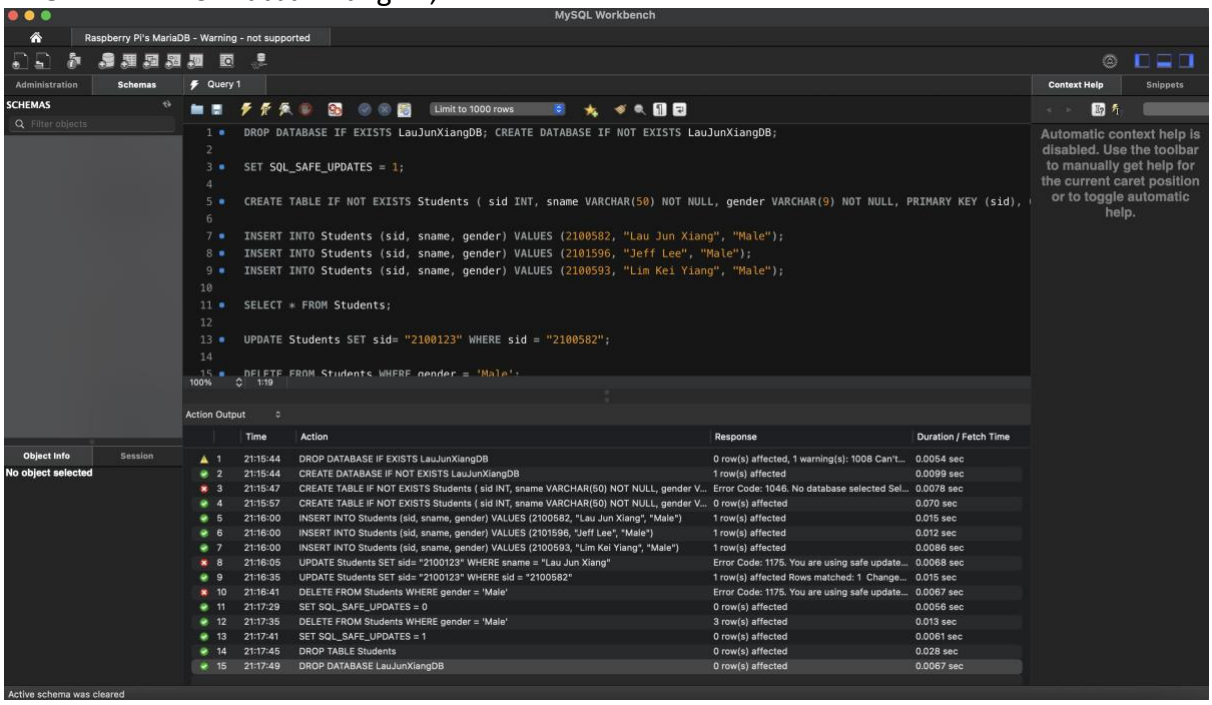
INSERT INTO Students (sid, sname, gender) VALUES (2100582, "Lau Jun Xiang", "Male"); INSERT INTO Students (sid, sname, gender) VALUES (2101596, "Jeff Lee", "Male"); INSERT INTO Students (sid, sname, gender) VALUES (2100593, "Lim Kei Yiang", "Male");

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;
```

Result Grid Filter Rows: Search Export:

	Query_ID	Duration	Query
▶	2	0.00068315	SHOW WARNINGS
	3	0.00130625	CREATE DATABASE IF NOT EXISTS LauJunXi…
	4	0.00067598	USE `LauJunXiangDB`
	5	0.02872523	CREATE TABLE IF NOT EXISTS Students (sid…
	6	0.00894435	INSERT INTO Students (sid, sname, gender) V…
	7	0.01152794	INSERT INTO Students (sid, sname, gender) V…
	8	0.01026590	INSERT INTO Students (sid, sname, gender) V…
	9	0.00404919	SHOW FULL COLUMNS FROM `LauJunXiang…
	10	0.00233158	SELECT * FROM Students LIMIT 0, 1000
	11	0.00121631	UPDATE Students SET sid= "2100123" WHER…
	12	0.00068539	SET SQL_SAFE_UPDATES = 0
	13	0.01532001	UPDATE Students SET sid= "2100123" WHER…
	14	0.00838572	DELETE FROM Students WHERE gender = 'M…
	15	0.02512442	DROP TABLE Students
	16	0.00366100	DROP DATABASE LauJunXiangDB

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;

```
MariaDB [(none)]> SHOW PROFILES;
+----------+------------+-------------------------------------------------------------
| Query_ID | Duration   | Query
+----------+------------+-------------------------------------------------------------
|        2 | 0.00087097 | SELECT DATABASE()
|        3 | 0.00095386 | CREATE DATABASE IF NOT EXISTS LauJunXiangDB
|        4 | 0.00125664 | SELECT DATABASE()
|        5 | 0.00244895 | show databases
|        6 | 0.00246694 | show tables
|        7 | 0.04701441 | CREATE TABLE IF NOT EXISTS Students ( sid INT, sname VARCHAR(50) NOT NULL, gender VARCHAR(9) NOT NULL, PRIMARY KEY (sid)
|        8 | 0.00886242 | INSERT INTO Students (sid, sname, gender) VALUES (2100582, "Lau Jun Xiang", "Male")
|        9 | 0.01242022 | INSERT INTO Students (sid, sname, gender) VALUES (2101596, "Jeff Lee", "Male")
|       10 | 0.00553113 | INSERT INTO Students (sid, sname, gender) VALUES (2100593, "Lim Kei Yiang", "Male")
|       11 | 0.00250764 | SELECT * FROM Students
|       12 | 0.00852313 | UPDATE Students SET sid= "2100123" WHERE sname = "Lau Jun Xiang"
|       13 | 0.01065715 | DELETE FROM Students WHERE gender = 'Male'
|       14 | 0.01627909 | DROP TABLE Students
|       15 | 0.00407766 | DROP DATABASE LauJunXiangDB
|       16 | 0.00170357 | SELECT DATABASE()
+----------+------------+-------------------------------------------------------------
15 rows in set (0.001 sec)
```

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