This tutorial is for the students to get familiar with the basic MySQL relational database functionality.

1) Installing MySQL

- Access RPi's terminal and enter the following command to install MySQL:

  ```
  #sudo apt-get install mysql-server
  ```

- Enter the command below to access MySQL. Press enter when prompted for a password:

  ```
  #sudo mysql –u root -p
  ```

- Enter the commands below in MySQL environment to create the user 'user' with 'pass' as password:

  ```
  >create user  'user'@'localhost'  identified by 'pass';
  >grant all privileges on *.*  to  'user'@'localhost';
  >flush privileges;
  >quit;
  ```

- Enter the command below to access MySQL as user 'user' and enter the password when prompted to do so:

  ```
  #mysql –u user -p
  ```

2) Creating a database and its tables for the following DB:

person (pname, street, city)
vehicle (year, make, model, cost, licplate, pname)
accident (accnum, licplate, accdate, pname)

- Enter the commands below in MySQL to create a database and its tables for the insurance database described above:

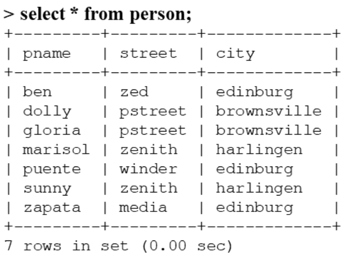
  ```
  >create database insurance;
  > use insurance;

  > create table person (pname VARCHAR(30) NOT NULL, street VARCHAR(50) NOT NULL, city VARCHAR(50) NOT NULL, Primary Key (pname));

  > create table vehicle (year VARCHAR(4) NOT NULL, make VARCHAR(30) NOT NULL, model VARCHAR(30) NOT NULL, cost INT NOT NULL, licplate VARCHAR(7) NOT NULL, pname VARCHAR(30) NOT NULL, Primary Key (licplate,pname));

  > create table accident (accnum INT NOT NULL, licplate VARCHAR(7) NOT NULL, accdate date NOT NULL, pname VARCHAR(30) NOT NULL, Primary Key (accnum,licplate));
  ```

- Using insert command, populate the tables created earlier with the data shown in the tables below (note that multiple records may be entered using one insert command):

  > insert into person values ("marisol", "zenith", "harlingen"), ("dolly", "pstreet", "brownsville");

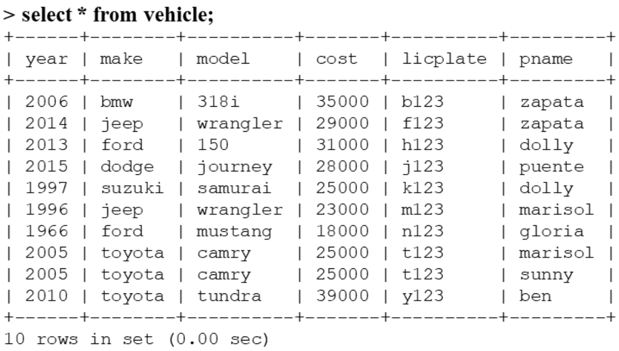
  > insert into vehicle values (2005, 'toyota', 'camry', 25000, 't123', 'marisol');

  > insert into accident values (101, 'n123', '2012/07/15', 'sunny');



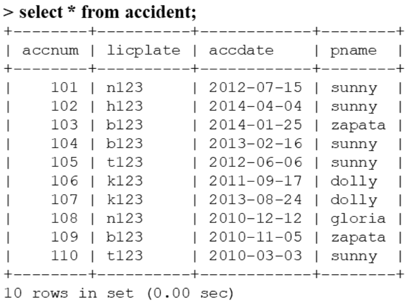
> select * from person;

```
+---------+---------+-------------+
| pname   | street  | city        |
+---------+---------+-------------+
| ben     | zed     | edinburg    |
| dolly   | pstreet | brownsville |
| gloria  | pstreet | brownsville |
| marisol | zenith  | harlingen   |
| puente  | winder  | edinburg    |
| sunny   | zenith  | harlingen   |
| zapata  | media   | edinburg    |
+---------+---------+-------------+
7 rows in set (0.00 sec)
```



> select * from vehicle;

```
+------+--------+----------+-------+----------+---------+
| year | make   | model    | cost  | licplate | pname   |
+------+--------+----------+-------+----------+---------+
| 2006 | bmw    | 318i     | 35000 | b123     | zapata  |
| 2014 | jeep   | wrangler | 29000 | f123     | zapata  |
| 2013 | ford   | 150      | 31000 | h123     | dolly   |
| 2015 | dodge  | journey  | 28000 | j123     | puente  |
| 1997 | suzuki | samurai  | 25000 | k123     | dolly   |
| 1996 | jeep   | wrangler | 23000 | m123     | marisol |
| 1966 | ford   | mustang  | 18000 | n123     | gloria  |
| 2005 | toyota | camry    | 25000 | t123     | marisol |
| 2005 | toyota | camry    | 25000 | t123     | sunny   |
| 2010 | toyota | tundra   | 39000 | y123     | ben     |
+------+--------+----------+-------+----------+---------+
10 rows in set (0.00 sec)
```



> select * from accident;

```
+--------+----------+------------+--------+
| accnum | licplate | accdate    | pname  |
+--------+----------+------------+--------+
|    101 | n123     | 2012-07-15 | sunny  |
|    102 | h123     | 2014-04-04 | sunny  |
|    103 | b123     | 2014-01-25 | zapata |
|    104 | b123     | 2013-02-16 | sunny  |
|    105 | t123     | 2012-06-06 | sunny  |
|    106 | k123     | 2011-09-17 | dolly  |
|    107 | k123     | 2013-08-24 | dolly  |
|    108 | n123     | 2010-12-12 | gloria |
|    109 | b123     | 2010-11-05 | zapata |
|    110 | t123     | 2010-03-03 | sunny  |
+--------+----------+------------+--------+
10 rows in set (0.00 sec)
```

- Use select command to list contents of the tables created:

  > select * from person;

  > select * from vehicle;

  >select * from accident;

3) Quitting, Accessing, Backing up, Restoring and Manipulating MySQL:

- Quitting from MySQL, re-entering MySQL, and listing assets:

```
>quit;                          //to go back to Raspbian environment from MySQL prompt

#mysql –u  user –p              //to access MySQL environment

Enter password: ****

>show databases;                //to list databases in the system

>use insurance;                 //to use an existing database, e.g. insurance

>show tables;                   //to list the table names in the database

>describe person;               //to list the column/field names of table person

>select * from person;          //to list all the rows from a table, e.g. person

>select version();              //to list the version of MySQL
```

- Backing up and restoring a database:

```
#mysqldump -u user -p  --all-databases > backup_Apr09.sql     //Backup (while in Raspbian)

#mysql -u user -p <  backup_Apr09.sql              //Restore (while in Raspbian)

                                                   //in case of errors one may have to

                                                   //delete files from  /var/lib/mysql/
```

- Manipulating tables within a database:

```
>drop table book;                         //to delete a table

>alter table patron add zip INT NOT NULL;         //to add a field to an existing table

>alter table patron drop zip;             //to delete a field from an existing table

>rename table patron to patrons;                  //to rename an existing table
```

- Changing values in a table:

```
>update person set srteet='luna', city='bville' where pname='ben';  //to update existing fields

>delete person where pname='ben';     //to delete rows from an existing table
```

4) SQL as a Query Language:

- Six clauses used in a SQL query:

Select, **From**, **Where** (condition on a row), **Group By**, **Having** (condition a group), **Order By**

a) Find all persons who live in Harlingen and own Toyota:

```
Select person.pname
From person, vehicle
Where person.pname=vehicle.pname and city='harlingen' and make='toyota'
```

b) Find all persons who live in Harlingen and own Toyota (show name and worth of vehicle… descending order by the worth of vehicle):

```
Select person.pname, cost
From person, vehicle
Where person.pname=vehicle.pname and city='harlingen' and make='toyota'
Order by cost desc
```

c) Find all persons who live in Harlingen and own more than $80K worth of vehicles:

```
Select person.pname
From person, vehicle
Where person.pname=vehicle.pname and city='harlingen'
Group by person.pname
Having sum(cost) > 20000
```

d) Find all persons who live in Harlingen and own more than $80K worth of vehicles:

```
Select pname
From person
Where city='harlingen' and
        pname in
        (Select pname
         From vehicle
         Group by pname
         Having sum(cost) > 20000)
```

e) Find all persons who live in Harlingen and on the same street:

```
Select A.pname, B.pname
From person A, person B
Where A.city='harlingen' and A.pname!=B.pname and A.street=B.street
```

f) Find all persons who live in Harlingen and on the same street and own toyota:

```
Select A.pname, C.pname
From person A, vehicle B, person C, vehicle D
Where A.pname=B.pname and A.city='harlingen' and
        C.pname=D.pname and C.city='harlingen' and
        A.pname!=C.pname and A.street=C.street and
        B.make='toyota' and D.make='toyota'
```