Using MySQL on the Winthrop Linux Systems

by Dr. Kent Foster

adapted for CSCI 297 Scripting Languages by Dr. Dannelly updated Oct 2018

I. Creating your MySQL password:

Your mysql account username has the format "my.yourusername" where "yourusername" is your regular campus username, such as "dannellys2".

To create your MySQL account logon to the web site

https://deltona.birdnest.org/mysql

and enter your Winthrop username and password. Then you get a window that includes your MySQL password in the email, along with other information and instructions. Here is an example of the message that you will receive:

Your MySQL account has been created. Username: my.dannellys2 Password: xxxxxxxx You can change your password by logging into mysql and running: set password=password('new_pass'); Database: my_dannellys2 An e-mail has been sent to you with this information. You may login to the text version of mysql. example: mysql -h deltona.birdnest.org -u my.dannellys2 -p If you have questions please e-mail: helpdesk@winthrop.edu

<u>Remember that password.</u> This is the password that you must use to login to MySQL. If you forget it, then you simply repeat this process, and you will be emailed a new password.

II. Connecting to the MySQL server

To start up MySQL, login to a Linux system and type this command:

```
> mysql -h deltona.birdnest.org -u my.yourusername -p
```

You must enter the MySQL password, not your Winthrop password, that you received in the email from the MySQL administrator. If you successfully have logged into MySQL, you will get a display like this:

```
> mysql -h deltona.birdnest.org -u my.fosterk1 -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 122 to server version: 3.23.49
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

If you did not enter the MySQL password correctly, the display looks like

```
> mysql -h deltona.birdnest.org -u my.fosterk1 -p
Enter password:
ERROR 1045: Access denied for user: 'my.fosterk1@10.3.0.47' (Using password:
YES)
```

III. The "help" Command

You now can type any MySQL command. Start by typing "help" and get the help listing:

```
mysql> help
```

MySQL commands:						
Note that all text commands must be first on line and end with ';'						
help	(\h)	Display this help.				
?	(/?)	Synonym for `help'.				
clear	(\c)	Clear command.				
connect	(\r)	Reconnect to the server. Optional arguments are db and host.				
edit	(\e)	Edit command with \$EDITOR.				
ego		Send command to mysql server, display result vertically.				
exit	(\q)	Exit mysql. Same as quit.				
go	(/g)	Send command to mysql server.				
nopager	(\n)	Disable pager, print to stdout.				
notee	(\t)	Don't write into outfile.				
pager	(\P)	Set PAGER [to_pager]. Print the query results via PAGER.				
print	(\p)	Print current command.				
quit	(\q)	Quit mysql.				
rehash	(\#)	Rebuild completion hash.				
source	$(\setminus .)$	Execute a SQL script file. Takes a file name as an argument.				
status	(\s)	Get status information from the server.				
tee	(\T)	Set outfile [to_outfile]. Append everything into given outfile.				
use	(\u)	Use another database. Takes database name as argument.				

In addition, you can get more help with "help contents":

```
mysql> help contents
You asked for help about help category: "Contents"
For more information, type 'help <item>', where <item> is one of the
following
categories:
```

Account Management Administration Data Definition Data Manipulation Data Types Functions Functions and Modifiers for Use with GROUP BY Geographic Features Language Structure Storage Engines Stored Routines Table Maintenance Transactions Triggers

For example,

```
mysql> help data definition
You asked for help about help category: "Data Definition"
For more information, type 'help <item>', where <item> is one of the
following topics:
   ALTER DATABASE
   ALTER TABLE
   ALTER VIEW
   CONSTRAINT
   CREATE DATABASE
   CREATE INDEX
   CREATE TABLE
   CREATE VIEW
   DROP DATABASE
   DROP INDEX
   DROP TABLE
   DROP VIEW
   MERGE
   RENAME TABLE
```

Try "help create table", but note that the listing is very long.

IV. The "show" Command

Chapter 9 of your textbook (*PHP and MySQL Web Development* by Welling and Thomson) introduces using the MySQL dbms. However, there are some things at the beginning of this chapter that you might not be able to do since your MySQL account has restricted privileges. For example, if you try to create a database, the operation will not succeed. A lot of chapter 9 deals with privileges that you cannot change. Chapter 10 deals with writing SQL commands. W3schools.com also has a good tutorial.

The simplest command is "show". The two arguments for show are "databases" and "tables". MySQL commands end with a semicolon.

mysql> show databases; +-----+ | Database | +----+ | information_schema | | my_dannellys2_default | +-----+ If you try show tables, then you get an error

mysql> **show tables;** ERROR 1046: No Database Selected

This means that you have not selected a database to use with the USE command. Generally, you will only have access to your database, so type the USE command to use your database:

```
mysql> use my_dannellys2_default;
Database changed
mysql> show tables;
Empty set (0.00 sec)
```

The my_dannellys2_default database is initially empty. So there are no tables, yet.

V. Creating Additional Databases

If you know you will have projects with several tables each, it is probably a good idea to group those tables into separate databases. The system creates a default database for you. Databases on Winthrop's system must be named my_user_database, where user is your ACC id and database is the name of the new database.

For example:

```
mysql> show databases;
+----+
Database
+----+
information schema
| my_dannellys2_default |
+----+
2 rows in set (0.00 sec)
mysql> create database my_dannellys2_advising;
Query OK, 1 row affected (0.00 sec)
mysql> show databases;
+----+
Database
+----+
| information_schema |
| my_dannellys2_advising |
| my_dannellys2_default |
+----+
3 rows in set (0.00 sec)
```

VI. Creating Tables and Adding Data

Let's create a couple of tables to use as an example. The first table will be named warehouse and contain information about the id number and location of our warehouses. A second table will contain information about employees and which warehouse an employee works at.

The first step is to create the warehouse table. Notice that MySQL continues the current command until we enter a semicolon. The following long command will create a new table with three fields.

```
mysql> create table warehouse
   -> (
   -> WhNumb Varchar(5) Primary Key,
   -> City Varchar(20),
   -> Floors Int
   -> );
Query OK, 0 rows affected (0.03 sec)
```

Now when we type the "show tables" command we now have something to see.

```
mysql> show tables;
+-----+
| Tables_in_my_dannellys2_default |
+-----+
| warehouse |
+-----+
1 row in set (0.00 sec)
```

The describe command is very useful. It is abbreviated to "desc".

```
mysql> desc warehouse;
```

+	Туре	+ Null	Key	Default	++ Extra ++
WhNumb City Floors	varchar(5) varchar(20) int(11)	NO YES YES	PRI	NULL NULL	++

3 rows in set (0.00 sec)

Now let's create the employee table.

```
mysql> Create Table employee

-> (

-> EmpNumb Varchar(5) Primary Key,

-> WhNumb Varchar(5) References Warehouse(WhNumb),

-> Salary Decimal(8,2),

-> YearHired Int

-> );

Query OK, 0 rows affected (0.00 sec)
```

Now type the INSERT commands to store data in the tables. The following will create two warehouse records and three employee records.

```
mysql> INSERT INTO warehouse (whnumb, city, floors)
    -> VALUES ('WH1','ROCK HILL',3);
Query OK, 1 row affected (0.04 sec)
mysql> INSERT INTO warehouse (whnumb, city, floors)
    -> VALUES ('WH2','CHARLOTTE',2);
Query OK, 1 row affected (0.00 sec)
mysql> INSERT INTO employee (empnumb, whnumb, salary, yearhired)
    -> VALUES ('E1','WH1',20000,1988);
```

```
Query OK, 1 row affected (0.00 sec)
mysql> INSERT INTO employee (empnumb, whnumb, salary, yearhired)
    -> VALUES ('E5','WH2',20000,1988);
Query OK, 1 row affected (0.00 sec)
mysql> INSERT INTO employee (empnumb, whnumb, salary, yearhired)
    -> VALUES ('E8','WH2',22000,1991);
Query OK, 1 row affected (0.00 sec)
```

VII. Queries

Query each table to display the stored data:

```
mysql> Select * From warehouse;
+-----+
| WhNumb | City | Floors |
+----+
| WH1 | ROCK HILL | 3 |
| WH2 | CHARLOTTE | 2 |
+----+
2 rows in set (0.00 sec)
```

```
mysql> select * from employee;
```

+	+	+	++					
EmpNumb	WhNumb	Salary	YearHired					
 E1 E5 E8	 wh1 wh2 wh2	20000.00 20000.00 22000.00	1988 1988 1988 1991					
++								
3 rows in set (0.00 sec)								

If we want to know our total payroll, then we can sum the salary column.

```
mysql> select sum(salary) from employee;
+-----+
| sum(salary) |
+-----+
| 62000.00 |
+----+
1 row in set (0.00 sec)
```

If we want to know just the warehouse number of the warehouse in Rock Hill, we can issue this query.

```
mysql> select whnumb from warehouse where city="Rock Hill";
+-----+
| whnumb |
+-----+
| wh1 |
+----+
1 rows in set (0.00 sec)
```

What if we want to know the yearshired for everyone who works in Rock Hill? The city is stored in the warehouse table, but the yearhired is stored in the employee table. So, what we want is yearhired from employee where those employees match the id number of the warehouse in Rock Hill.

When querying multiple tables we have to indicate each field as "table.field" and all the accessed tables need to be comma separated in the "from" portion of the SQL command.

```
mysql> select employee.yearhired
    -> from employee, warehouse
    -> where employee.whnumb = warehouse.whnumb
    -> and warehouse.city = "Rock Hill";
+-----+
| yearhired |
+-----+
| 1988 |
+-----+
1 row in set (0.01 sec)
```

VIII. Issuing a batch of MySQL statements

To create the database objects in a "batch" operation, create a file of the SQL statements necessary to define the database objects. Then submit this to SQL with a command of the MySQL 'source' command. For example, if you created a file named TESTDB.DEF, then submit it to MySQL with the command, for example,

mysql> source /home/acc.fosterk1/csci555/TESTDB.DEF;

Here, the file TESTDB.DEF contains the following text:

```
--COMMIT;
-- File: TESTDB.DEF
_ _
-- CREATE SCHEMA FILENAME "WAREHOUSES";
_ _
       *** Create tables ***
--
_ _
Create Table warehouse
    (
      WhNumb Varchar(5) Primary Key,
      City Varchar(20),
      Floors Int
     );
-- Insert some rows
INSERT INTO warehouse (WhNumb, City, Floors) Values ('WH1', 'Rock Hill', 1);
INSERT INTO warehouse (WhNumb, City, Floors) Values ('WH2', 'Charlotte', 2);
INSERT INTO warehouse (WhNumb, City, Floors) Values ('WH3', 'Rock Hill', 2);
Create Table employee
     (
       EmpNumb Varchar(5) Primary Key,
       WhNumb Varchar(5) References Warehouse(WhNumb),
       Salary Decimal(8,2),
       YearHired Int
     );
INSERT INTO employee (WhNumb, EmpNumb, Salary, YearHired)
             Values ('WH1', 'E1', 20000, 1998);
INSERT INTO employee (WhNumb, EmpNumb, Salary, YearHired)
             Values ('WH1', 'E4', 21000, 1999);
INSERT INTO employee (WhNumb, EmpNumb, Salary, YearHired)
             Values ('WH2', 'E2', 22000, 1998);
```

```
INSERT INTO employee (WhNumb, EmpNumb, Salary, YearHired)
Values ('WH3', 'E3', 20000, 2000);
INSERT INTO employee (WhNumb, EmpNumb, Salary, YearHired)
Values ('WH1', 'E5', 25000, 2001);
--COMMIT;
Select * from warehouse;
Select * from employee;
Drop table warehouse;
Drop table warehouse;
Show tables;
-- end of source
```

The output from the source file of MySQL commands is below.

```
mysql> source /home/acc.fosterk1/csci555/TESTDB.DEF;
Query OK, 0 rows affected (0.00 sec)
Query OK, 1 row affected (0.00 sec)
Query OK, 1 row affected (0.00 sec)
Query OK, 1 row affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 1 row affected (0.00 sec)
+----+
| WhNumb | City | Floors |
+----+
| WH1 | Rock Hill | 1 |
WH2 Charlotte 2
| WH3 | Rock Hill |
                           2
+----+
3 rows in set (0.00 sec)
+----+
| EmpNumb | WhNumb | Salary | YearHired |
+----+

      E1
      WH1
      20000.00
      1998

      E4
      WH1
      21000.00
      1999

      E2
      WH2
      22000.00
      1998

      E3
      WH3
      20000.00
      2000

      E5
      WH1
      25000.00
      2001

+----+
5 rows in set (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Empty set (0.00 sec)
mysql>
```