Work with MySQL client
Part II: Working with mysql client

Working with MySQL Client

Changing the password.

Unless you have database administrator's (DBA) privileges, you can only change the password by typing it explicitly as part of a SQL command. The command is

```
mysql> SET PASSWORD = PASSWORD('<your password>');
```

Here, the `SET` command is a `mysql` command that is used to set a variety of operating parameters (more about this command below). You need to replace `<your password>` with the actuall text of the password you want to set. The `PASSWORD()` function converts the plain text password into the password hash that gets stored by `mysql`.

To set your password to `'fooBar'` type the command as follows:

```
mysql> SET PASSWORD = PASSWORD('fooBar');
```

Yes, it is potentially unsafe, as your password is clearly visible. That's mysql for you.

Work with SQL scripts.

Any file containing SQL comments, SQL commands and mysql client commands can be run by the `mysql` client.

There are two ways this can be done: from within the client (interactively) and in a batch mode.

Running SQL scripts interactively. To run SQL scripts interactively, use the source command:

```
mysql> source <filename>;
```

Running SQL scripts in batch mode. From the Linux prompt, enter the following command:

```
$ mysql [options]  < <filename>
```

Here, `options` are the command-line parameters you want `mysql` client to run with.

Comments. Comments in SQL scripts are any lines that start with a double dash '--'. Comments can also be put at the ends of the lines, as shown in the example below.

Consider the following simple SQL script `test.sql`.

```
-- Alex Dekhtyar
-- CSC 365

CREATE TABLE test(
   id int,
   name varchar(20),
   PRIMARY KEY(id)   -- primary key declaration on a separate line
)
;

INSERT INTO test VALUES(1,2),(3,4),(5,6);

select * from test;
```

Running the script interactively yields the following output:

```
mysql> source test01.sql
Query OK, 0 rows affected (0.02 sec)

Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

+----+------+
| id | name |
+----+------+
|  1 | 2    |
|  3 | 4    |
|  5 | 6    |
+----+------+
3 rows in set (0.00 sec)

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

Running the same script in the batch mode results in the following output:

```
 $ mysql <test01.sql
Enter password:
id      name
1       2
3       4
5       6
```

Verbose mode.

`mysql` client can be run in a `verbose` mode, in which all commands sent to the server as echoed on the terminal. To engage the mode, the `mysql` client needs to be started with the `--verbose` or `-v` flag.

For example:

```
 $ mysql -v <test01.sql
Enter password:
--------------
CREATE TABLE test(
   id int,
   name varchar(20),
   PRIMARY KEY(id)
)
--------------

--------------
INSERT INTO test VALUES(1,2),(3,4),(5,6)
--------------

--------------
select * from test
--------------

id      name
1       2
3       4
5       6
```

Log of **mysql** activity

You can create logs of your mysql activity using the `tee` command:

```
mysql> tee <filename>
```

where `<filename>` is the name of the file you want to log your commands to will result in logging of all your `mysql` activity to the named file.

When you want to stop logging you can issue the `notee` command:

```
mysql> notee
```

Here is a simple session:

```
mysql> tee test.out
Logging to file 'test.out'
mysql> source test01.sql
Query OK, 0 rows affected (0.01 sec)

Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

+----+------+
| id | name |
+----+------+
|  1 | 2    |
|  3 | 4    |
|  5 | 6    |
+----+------+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> notee
Outfile disabled.
mysql> exit
Bye
```

```
dekhtyar@londo:~/classes/365/scripts $ cat test.out
mysql> source test01.sql
Query OK, 0 rows affected (0.01 sec)

Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

+----+------+
| id | name |
+----+------+
|  1 | 2    |
|  3 | 4    |
|  5 | 6    |
+----+------+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> notee
```