Lab 04 Managing MySQL: User, Data, & Quality Check

Hide Instruction

Assignment Instruction

To successfully complete this assignment, you are advised to pay attention to the following:

1. **Follow instructions**: Read, interpret, and follow instructions very carefully. Finish reading the whole question and note the requirements **before** you start working on the question.
2. **Use resources**: You may use any resources available to you via the web. You are encouraged to discuss the problems with others before you sit down and attempt to answer the questions.
3. **Work independently**: While you are completing the questions, you are expected to work on your own without direct help from other people (non-human resources such as search engines are still allowed).
4. **Work together**: Come to the office hours to work as a group on the interpretation of and solutions to questions. When using text messaging, attach screenshots.
5. For introduction to basic Linux and shell commands and special characters, see:
    - Linux & shell basic commands
    - Linux & shell special characters

Getting Started:

1. **class** directory: To begin this assignment, make sure you navigate into the class directory under your **user home** directory.
2. **scp** the tar file: Use scp to copy the tar file a04.tar.gz from datacenter@tychen.us to your `~/class` directory. The command is: `scp datacenter@tychen.us:~/class/labs/a04.tar.gz .` (### don't forget the "`.`" at the end of the command) and the password is "InCharge56!"
3. **unzip** the tar file: Extract the contents of the tar file:`tar -xf a04.tar.gz`. Once the files are extracted, you should see the `a04` folder (a directory containing several directories and files) in your class directory. These are the files you need to complete this assignment.
4. **a04**: Unless otherwise specified in the question, all file names are given relative to the directory ~/class/a04.
5. **Passwords**: All the account names and the corresponding passwords are recorded in Canvas Assignment a00, where you keep them updated.

When You Are Done:

When you have completed the assignment,

1. **done.sh**: Run the script **done.sh** in ~/class as follows: `./done.sh firstname lastname a04` (The done.sh script is part of Assignment h01.)
2. **lab04.txt**: The done.sh script will check your answers and create a new .txt called lab04 .txt in the a04 directory that you will need to submit to Canvas for assignment a04.
3. **Download** lab04.txt to submit to Canvas: From your local computer terminal (cmd or PowerShell for Windows and Terminal for macOS), change into a folder (e.g., `cd`

`scp YOUR_USERNAME@YOUR_VM_IP_ADDRESS:~/class/a04/lab04.txt .`

4. Note: You can run the done.sh script at any point when you are working on the assignment and the content of the generated lab04.txt may be helpful in addition to the check scripts in each question. However, you must run it just prior to submitting the generated lab04.txt file in your class/a04 directory to Canvas to make sure you are submitting the most recent file.
5. Note: To see your VM's IP address, run `ip addr` or `ifconfig` (installation of netools required).

Part 1. MySQL Processes And Root Password

- This process must be completed as a Linux system administrator, so you will have to use your **administrative** user account to complete this part of the lab.
- You will reset the MySQL root password on your system and make sure it works properly.

1. Determine if mysql is running on your computer.

   Use the command `ps -aef` to view all of the processes running on your computer (there are a lot of them).

   Determine if mysql is running using the following command:

   ```
   ps -aef | grep mysql
   ```

   You will see the results as below (the first line is the grep line you just issued, the second line is the mysql service and its binary is mysql`d`; the `d` stands for daemon, meaning a background process):

   ```
   chen_ad+ 31011 30856  0 15:57 pts/0    00:00:00 grep --color=auto mysql
   mysql    32970     1  0 Feb02 ?        00:13:45 /usr/sbin/mysqld --daemonize --pid-file=/run
   ```

2. Use the sudo command to stop the mysql service as follows (note the `service` command syntax) (now that you need to run sudo, if you are not a sudoer now, you can use `su` *admin_account* to switch to it):

   ```
   sudo service mysql stop
   ```

   Check to make sure mysql has stopped running using `ps -aef | grep mysql`. Now you should only see the grep line:

   ```
   chen_ad+ 32077 30856  0 16:00 pts/0    00:00:00 grep --color=auto mysql
   ```

3. Next, **create and modify** a security directory:

   ```
   sudo mkdir /var/run/mysqld                  ### create directory
   sudo chown mysql /var/run/mysqld            ### change directory owner to mysql
   ```

4. Force mysql into **safe mode**:

   ```
   sudo mysqld_safe --skip-grant-tables &      ### mysql safe startup script; no password requ
   ```

   When you run mysqld_safe, you will see results like below.

   ```
   [1] 36875
   chen_admin@vma50:~$ 2021-02-22T21:13:33.044512Z mysqld_safe Logging to syslog.
   2021-02-22T21:13:33.049909Z mysqld_safe Logging to '/var/log/mysql/error.log'.
   2021-02-22T21:13:33.096052Z mysqld_safe Starting mysqld daemon with databases from /var/lib/
   ```

background (in a subshell), we can press Enter to escape back to the Linux CLI.

Now let us make sure mysql safe mode processes are running using `ps -aef | grep mysql`. You should see 4 mysql related processes. Your output will be similar to the following:

```
root     36875 30856  0 16:13 pts/0    00:00:00 sudo mysqld_safe --skip-grant-tables
root     36876 36875  0 16:13 pts/0    00:00:00 /bin/sh /usr/bin/mysqld_safe --skip-grant-ta
mysql    37236 36876  0 16:13 pts/0    00:00:00 /usr/sbin/mysqld --basedir=/usr --datadir=/v
chen_ad+ 38388 30856  0 16:16 pts/0    00:00:00 grep --color=auto mysql
```

This means mysqld (server) is running in safe mode and we now can log in as root without password.

5. Access MySQL as the root user, **without** a password (mysqld is the server, mysql is the command line tool):

```
mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.33-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

6. Once you have logged into the MySQL command line as root without a password, you can reset the MySQL root password by running the following MySQL queries in MySQL command line:

    1. Select the mysql database:

    ```
    mysql> USE mysql;
    Reading table information for completion of table and column names
    You can turn off this feature to get a quicker startup with -A

    Database changed
    ```

    2. Change the password of the root user in the user table (column User):

    ```
    mysql> UPDATE user SET plugin='mysql_native_password' WHERE User='root';
    Query OK, 1 row affected (0.00 sec)
    Rows matched: 1  Changed: 1  Warnings: 0

    mysql> flush privileges;
    Query OK, 0 rows affected (0.00 sec)

    mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'YOUR_PASS_WORD';
    Query OK, 0 rows affected (0.00 sec)
    ```

    3. Reload the grant (permissions) tables to make the changes into effect:

    ```
    mysql> FLUSH PRIVILEGES;
    Query OK, 0 rows affected (0.01 sec)
    ```

restart the mysqld server in normal mode:

1. Exiting mysql CLI using exit or quit:
```
mysql> exit
Bye
chen_admin@vma50:~$
```
Note that after we quit from mysql CLI, we are back to the Linux CLI.

2. Stop mysql safe mode: Now let's stop mysql safe mode by using the command `sudo mysqladmin -u root -pNewPassword shutdown` (### note: no space after the `-p` option):
```
chen_admin@vma50:~$ sudo mysqladmin -u root -pNewPassword shutdown
[sudo] password for chen_admin:
mysqladmin: [Warning] Using a password on the command line interface can be insecure.
2021-02-22T21:35:00.046226Z mysqld_safe mysqld from pid file /var/run/mysqld/mysqld.pid
[1]+  Done                    sudo mysqld_safe --skip-grant-tables
chen_admin@vma50:~$
```
You can use -p with the mysqladmin command like "sudo mysqladmin -u root -p shutdown" to enter the password after hitting enter to avoid the mysqladmin warning message.

3. Restart mysql service: To restart the mysqld service, use the `service` command:
```
sudo service mysql restart
```

4. Test it: To make sure the root password change is successful, log into mysql command line with the new password to make sure it works:
```
mysql -u root -p         ### -u, or --user, option means username; -p means password
```
Note: You will the socket error if you did not restart mysql:
```
administrator@vma43:~/class/a04/q01$ mysql -u root -p
Enter password:
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/run/mysqld/
```

5. If everything goes well, you will get into MySQL command line:
```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```
Typing `exit` or `quit` will get you out of mysql command line.

8. Record your MySQL root password to Assignment a00 on Canvas and log back to your Linux CLI as a regular user.

To check this part of the assignment:

1. Change to your class/a04/q01 directory.
2. Edit the file a01.txt using the command nano a01.txt
3. Use the arrow keys to move the cursor to the line starting with “A.”
4. Enter the word “DONE” on the line starting with “A.”.
5. Save the file (using Ctrl-X)
6. Run the script check01.sh (using ./check01.sh) to check your answer

Part 2. Creating MySQL User Accounts

do the following:

a. Create a database called sales using your root account.
b. Create a mysql user account called salesadmin. This will be your database administrator account.
c. Grant the salesadmin account ALL privileges to all the tables (the "*") in the sales database.
The commands you use will look like this: (## Note that use single quotes around username, localhost, and password may be critical.)

```
mysql> CREATE DATABASE sales;
mysql> CREATE USER 'salesadmin'@'localhost' IDENTIFIED BY 'YOUR_PASSWORD';
mysql> USE sales;
mysql> GRANT ALL ON sales.* TO 'salesadmin'@'localhost';
mysql> FLUSH PRIVILEGES;
```

2. Create a second MySQL user account (salesmgr) that provides access to the data, but doesn't have the authority to change the data in the database sales (i.e., grant this user `SELECT` privileges on all the tables in the database sales by using "GRANT SELECT" instead of "GRANT ALL" after creating the user).
3. Verify that you can log in to both of the new accounts and have access to the sales database. For each account, log into mysql CLI and use `SHOW DATABASES` to show the "information_schema" and "sales" databases (unlike when you are a `root`, where you see all the tables); then use `SHOW GRANTS` to show the appropriate rights for the logged in account. For example, when logged in MySQL command line as salesadmin, we can check the access privileges by:

```
mysql> SHOW GRANTS;
+---------------------------------------------------------------+
| Grants for salesadmin@localhost                               |
+---------------------------------------------------------------+
| GRANT USAGE ON *.* TO 'salesadmin'@'localhost'                |
| GRANT ALL PRIVILEGES ON `sales`.* TO 'salesadmin'@'localhost' |
+---------------------------------------------------------------+
2 rows in set (0.00 sec)
```

When logged in as salesmgr, you will see the privilege SELECT on the second line of the table:

```
mysql> SHOW GRANTS;
+-----------------------------------------------------+
| Grants for salesmgr@localhost                       |
+-----------------------------------------------------+
| GRANT USAGE ON *.* TO 'salesmgr'@'localhost'        |
| GRANT SELECT ON `sales`.* TO 'salesmgr'@'localhost' |
+-----------------------------------------------------+
2 rows in set (0.00 sec)
```

4. Record your mysql account names and passwords to assignment a00 and get back to the Linux CLI as a regular user.

To check this part of the assignment:

1. Change to your class/a04/q02 directory.
2. Complete the checklist below (A-G) in the file a02.txt.
3. Save the file (using Ctrl-X)
4. Run the script check02.sh (using ./check02.sh) to check your answer

Question 2 checklist:

Parts B-F require a one-word answer (YES) when you have completed them.

B. Have you shared your MySQL root password with your lab partner (if you have one)?
C. Have you shared your salesadmin account password with your lab partner (if you have one)?
D. Have you shared salesmgr account password with your lab partner (if you have one)?
E. Does the salesadmin account have ALL privileges?
F. Does the salesmgr account have SELECT privileges?
G. Have you answered the above question truthfully?

Note:

1. When you operate your own virtual machine, answer the lab partner parts as you are your own partner.

Part 3. Inserting Data Into Your Database

We will insert 3 tables into our sales database using tools already made. Our major task will be checking the basic information in the tables.

1. Use your ~/class/a04/q03 directory for this part of the lab.
2. Run the Perl script `agents.pl` provided to read the file agents.txt in the SALESDATA directory. That program will create the file agents.sql.
3. Next, insert this data into the sales database sales using the following command within mysql command line (i.e., you will first need to log back into mysql as user salesadmin in the same directory where you ran the Perl script.):

```
source agents.sql;  ### source executes commands in the current shell.
                    ### We have used it when we worked on environment variables and aliases.
```

You will see something like:

```
...
...
Query OK, 1 row affected (0.01 sec)
Query OK, 1 row affected (0.01 sec)
Query OK, 1 row affected (0.02 sec)
mysql>
```

If you `source` this .sql file for more than one time, you will see:

```
...
...
ERROR 1062 (23000): Duplicate entry 'D39-20-04' for key 'PRIMARY'
ERROR 1062 (23000): Duplicate entry 'D39-20-05' for key 'PRIMARY'
ERROR 1062 (23000): Duplicate entry 'D39-20-06' for key 'PRIMARY'
mysql>
```

4. The Perl script `dealers.pl` will read and translate the data contained in SALESDATA/dealerships.txt into two files in the q03 directory:
   - dealers.sql, and
   - owners.sql.

   These two files contain the mysql commands needed to insert data into a table containing the dealers and owners. Run script `dealers.pl` so it can create the files dealers.sql and owers.sql.
5. Insert these two datasets into the sales database using commands like the one in step 3 above to create the dealers and owners table in your sales database.

```
mysql> USE sales;
Database changed
mysql> SHOW TABLES;
+----------------+
| Tables_in_sales |
+----------------+
| agents          |
| dealers         |
| owners          |
+----------------+
3 rows in set (0.00 sec)

mysql>
```

6. Check that these tables contain the correct number of dealers and owners and then check a couple of the individual entries by hand by comparing the data in the original data with the data in the database.

   Some MySQL keywords such as DESCRIBE, SELECT COUNT(*), and LIMIT can be informative:

   a. DESCRIBE:

   ```
   mysql> DESCRIBE owners;
   +----------+-------------+------+-----+---------+-------+
   | Field    | Type        | Null | Key | Default | Extra |
   +----------+-------------+------+-----+---------+-------+
   | owner_id | int(11)     | NO   | PRI | NULL    |       |
   | fname    | varchar(24) | YES  |     | NULL    |       |
   | lname    | varchar(24) | YES  |     | NULL    |       |
   +----------+-------------+------+-----+---------+-------+
   3 rows in set (0.02 sec)
   ```

   b. SELECT COUNT(*)

   ```
   mysql> SELECT COUNT(*) FROM owners;
   +----------+
   | COUNT(*) |
   +----------+
   |       34 |
   +----------+
   1 row in set (0.00 sec)
   ```

   c. LIMIT

   ```
   mysql> SELECT * FROM owners LIMIT 5;
   +----------+----------+----------+
   | owner_id | fname    | lname    |
   +----------+----------+----------+
   |        1 | Sally    | Walker   |
   |        2 | Ed       | Tipton   |
   |        3 | Kelly    | Jones    |
   |        4 | Patrick  | Wallace  |
   |        5 | Jonathan | Houseman |
   +----------+----------+----------+
   5 rows in set (0.00 sec)
   ```

To check this part of the assignment:

1. Change to your class/a04/q03 directory.
2. Answer the questions below in the file a03.txt.
3. Run the script check03.sh (using ./check03.sh) to check your answer

Questions for Part 3:

B. How many dealers are in the sales database?
C. How many insert statements are in the dealers.sql file?
D. How many lines are in the SALESDATA/dealerships.txt file?

Part 4. Automating The Data Checking Procedure

Because we have a lot of data, we will need a process to check the data insertion process. The process will require checking the tables created within MySQL to make sure that they contain the same information as the original text files. To do that, we need **automated** ways to compare the data in the text files to the data in your MySQL database. Since this is a multistep process, it is helpful to check the data after each step.

A. Use your ~/class/a04/q04 directory for this part of the lab.
B. Create links to enable you to access the SALESDATA directory and the agents.sql, dealers.sql, and owners.sql files your created in Part 3. We create **soft (symbolic) links** to these files to accomplish this. The basic command for linking your agents.sql file in your q03 directory to a file with the same name in your q04 directory is:
`ln –s ~/class/a04/q03/agents.sql agents.sql`

   Once you have entered the above command, you can use the file agents.sql as if it were located in your q04 directory. The only critical difference is that the command `rm agents.sql` removes the link in the q04 directory, but not the file in the q03 directory since it's a soft link to the file. Using the `ls -l` command, you can see this soft link points to the original file in q03:

```
chen_user@vma49:~/class/a04/q04$ ls -l
total 16
-r--r--r-- 1 chen_user chen_user  320 Feb 17  2020 a04.bak
-rw-rw-r-- 1 chen_user chen_user  320 Feb 17  2020 a04.txt
lrwxrwxrwx 1 chen_user chen_user   40 Feb 23 12:20 agents.sql -> /home/chen_user/class/a04/q
-rwxrwxr-x 1 chen_user chen_user 3995 Dec 19  2019 check04.sh
-rwxrwxr-x 1 chen_user chen_user  904 Dec 19  2019 link_data.sh
```

   Create the link above as a practice, remove the link, and then run the script `link_data.sh`. It will create all of the required links for this assignment.

C. Create a script called q04.sh that uses some combination of command `grep` and `wc` to:
   a. **Count *.txt**: Count the number of agents, dealers, and owners contained in the initial text data files, i.e., agents.txt and dealerships.txt. Assume each dealership has a single owner and don't worry about one person owning multiple dealerships for now.
   b. **Count *.sql**: Count the number of agents, dealers, and owners contained in the agents.sql, dealers.sql, and owners.sql files
   c. **Count Alabama**: Count the number of dealers in Alabama in your initial text data and sql files.
   d. **Count Walker**: Count the number of dealerships owned by someone named Walker in your initial text data and compare it to the number of owners in owners.sql files who have a last name of Walker.

Once you have figured out which combinations of commands provide the correct output, compile them all into the script q04.sh and produce a summary output.

When you are done, your output should look something like the following:

```
        Dealers:    DATA = 45   SQL = 45
        Owners:     DATA = 45   SQL = 45
        Alabama:    DATA = 11   SQL = 11
        Walker:     DATA = 2    SQL = 4
```

To check part 4 of this assignment:

1. Change to your class/a04/q04 directory.
2. Enter DONE for part A. in the file a04.txt when you are done.
3. Run the script check04.sh (using ./check04.sh) to check your answer.

Note:

- Sometimes when you copy the commands from the webpages or PowerPoint slides, the dash `-` before the option may become a long dash, which is not accepted in bash. In that case, just modify them as short dashes.
- You will not find a file named owners.txt. Instead, the owner information is part of the dealerships.txt. Ideally, to count owners, you may want to count distinct Owner ID's. In this question, if you just count the number of records (disregard that some owners own more than one dealership), you will pass the check script. In this case, the number of Owners counted will be larger than that from the .sql file because of some owners own multiple dealerships.

Part 5. Automating MySQL Checking Procedure

Use nano to create a mysql input file (call it q05.sql). This .sql script will be used to obtain some information from the tables so that we can verify the information in the **sales** database is correct. The process used is as follows:

- Step 1: Create SQL statements

  Create a file called q05.sql that contains all of the sql statements needed to answer questions A-G below. (don't forget the `USE sales;` statement to specify the database to use at the beginning of a .sql file).

- Step 2: Run .sql to generate .txt file

  Run mysql from the Linux command line, use q05.sql as input and write the output into q05.txt. This can be done by issue commands in Linux command line as such:

  `mysql -u mysql_username -pmysql_password < q05.sql > output.txt`

  Note that in the command above, we use the MySQL command line tool, mysql, followed by username and password, then we use I/O redirections to input the .sql file and output the .txt file.

The q05.sql file should contain the SQL statements needed to answer the following questions:

- How many agents are there in total? `### SELECT COUNT`
- How many unique owners are there in total? `### SELECT DISTINCT COUNT( )`
- How many agents have a first name of Bill? `### SELECT ... WHERE ...`
- How many dealerships are in Alabama? `### SELECT ... WHERE ...`
- How many dealers are in Sawyer, TN? `### SELECT ... WHERE ... AND ...`
- How many dealerships are owned by someone whose last name is Walker? `### JOIN`
- How many sales agents work for Maureen Isern? `### JOIN`

To check Part 5 of this assignment:

mysql command line to see the results of your queries.)

2. Run the script check05.sh to check your answer.
3. Don't forget to run Step 2 above to generate the .txt output file. The check script may not inform you if the .txt file is missing.

Questions for Part 5:

A. Enter DONE for part A when you have completed your script. Then answer the rest of the questions in your a05.txt file.
B. How many agents are there in total?
C. How many unique owners are there in total?
D. How many agents have a first name of Bill?
E. How many dealerships are in Alabama?
F. How many dealers are in Sawyer, TN?
G. How many dealerships are owned by someone whose last name is Walker?
H. How many sales agents work for Maureen Isern?

Note:

1. For the sake of passing the check script, please DO NOT use the AS keyword when you are preparing the INNER JOIN statements.
2. A good way to create these SQL statements is to open two SSH terminal windows, run mysql command line in one, and nano editor in the other. Compose and test your SQL statements using mysql, and then cut and paste them into nano when they are working.