

Practical Exercise 10-2: Managing SQL Data

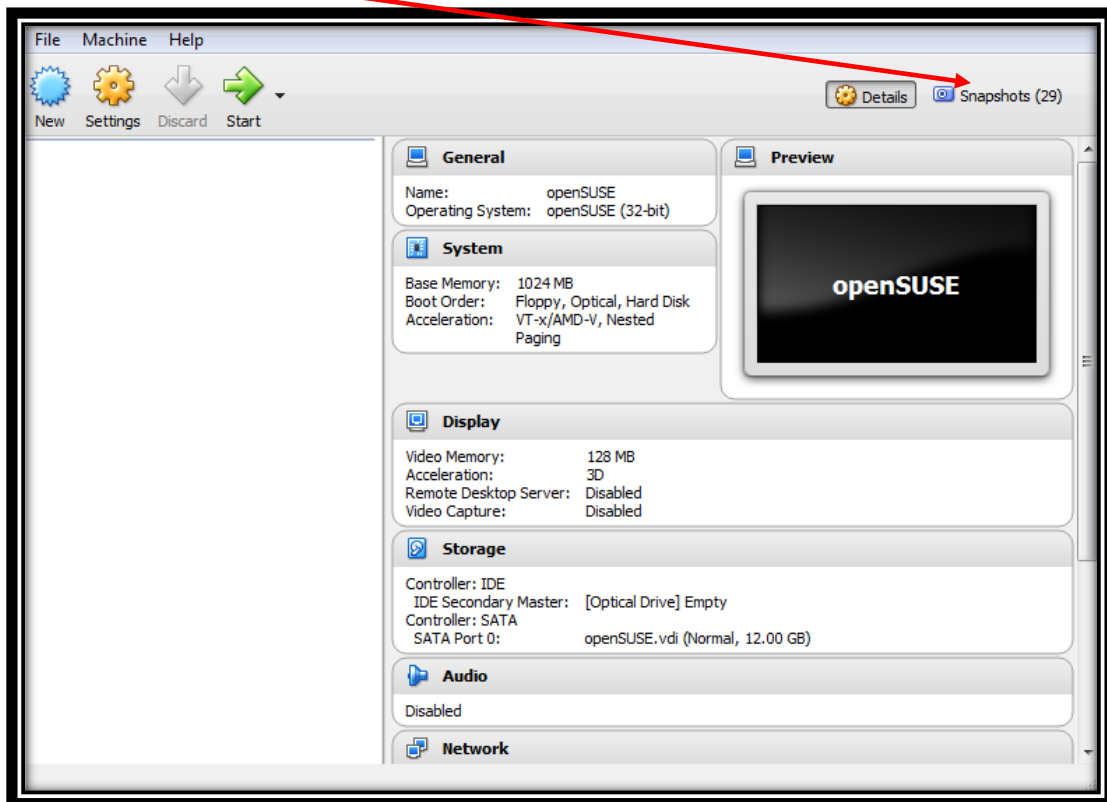
This Practical Exercise will take students through the install of Mariadb which is very similar to MySQL, the creation of a small database called customers, populate it with data and then modify and delete data from the database.

Open VirtualBox and start the openSUSE VM. Run snapshot 16-3 for the correctly configured environment. To run snapshot 16-3:

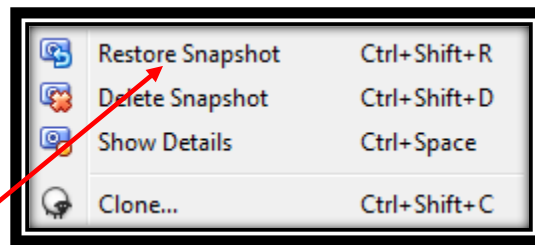
1. Open the Oracle VM VirtualBox manager by double clicking this icon on your desktop:



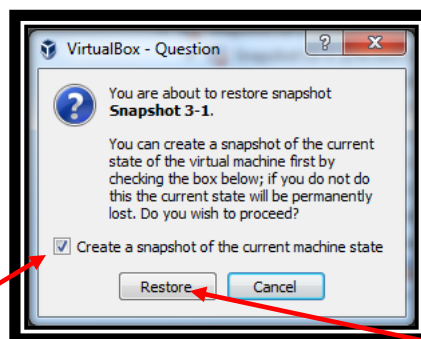
2. Click "Snapshots" in the top right of the Oracle VM Virtualbox Manager.



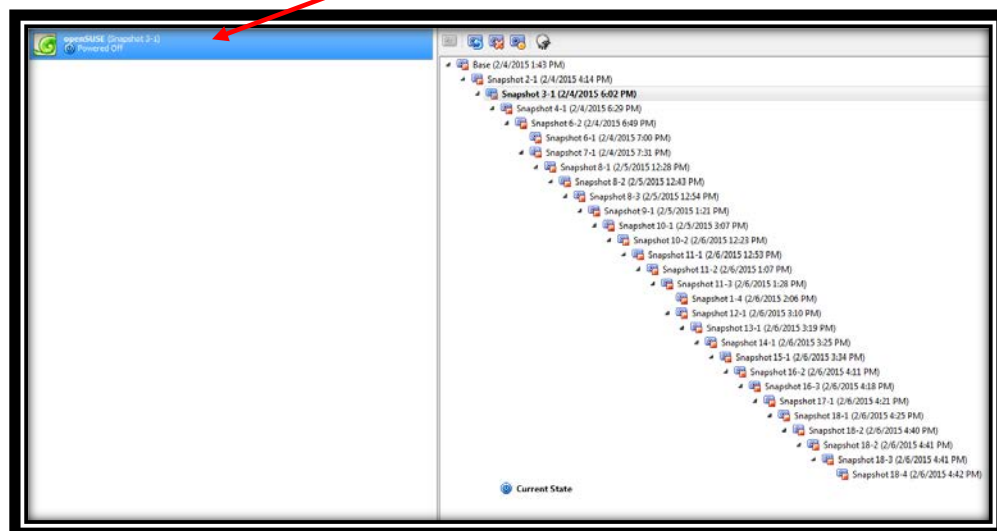
3. In the right side box populated with snapshots scroll up and find the one titled "Snapshot 16-3" and right click on it. The following box should appear:



4. Select "Restore Snapshot" and the following pop-up should appear:



5. Uncheck the "Create a Snapshot of the current machine state" box and then click the "Restore" button. If the pop up box does not have the check box, just click "Restore."
6. You should now see in the left box the openSUSE (Snapshot 16-3) with a status of "Powered Off." Power it on by double clicking it.



7. A separate window should open and you should see the openSUSE Linux OS booting.
8. Press **CTRL+ALT+F1** and log in with username: **root** and password: **student**.
9. Set up your MySQL server's grant tables by changing to the /usr/bin directory (**cd /usr/bin**) and entering **mysql_install_db** command at the shell prompt.
10. There is a bug in this version of MySQL within this snapshot, specifically the **mysql_install_db** script. The bug gives incorrect permissions on the server data directory. If you do not believe me go to step 11 and try to start mysql, I'll wait and see you in a second. Nice to see you again enter **chown -R mysql:mysql /var/lib/mysql**.
11. Start the MySQL service by entering **systemctl start mysql** at the shell prompt.
12. Verify that the server is running by entering the **mysqladmin version** command at the shell prompt.
13. Assign a password to your MySQL root user account by entering **mysqladmin -u root password 'student'** at the shell prompt.
14. Connect to the MySQL server from the command line by entering **mysql -h localhost -u root -p** at the shell prompt. When prompted, enter the root password: **student** you just specified.
15. At the mysql prompt, enter **CREATE DATABASE customers;**. You should be prompted that the database was created.
16. To use the new database, enter **USE customers;** at the mysql prompt. You should see that the none in the brackets changed to customers.
17. To view the tables that were defined by default in the new database, enter **SHOW TABLES;** at the mysql prompt. You should see that no tables are defined. At this point, things get a little more complex. You need to use the **CREATE TABLE** command to create the table, but you will also need to define your table's columns in the command. Therefore, you first need to define the fields you want to include in each record in the table. You also need to decide what kind of data will be stored in each field as well as how long the table will be. In the table we're building here, we'll need four columns:

first
last

phone
lastcontact

We know that the 'first' and 'last' fields will contain alphabetical characters of varying length. The 'phone' field will contain ten digits with two hyphens (area_code-prefix-number). The 'lastcontact' column will contain the date of each customer's last purchase.

18. Create a new table in the database by entering **CREATE TABLE active (first VARCHAR(15), last VARCHAR(15), phone CHAR(12), lastcontact DATE);** at the mysql prompt.
19. Verify that the table and columns were created correctly by entering **DESCRIBE active;** at the mysql prompt. You should see that the active table and its fields were created, as shown here:

```
MariaDB [customers]> DESCRIBE active;
```

Field	Type	Null	Key	Default	Extra
first	varchar(15)	YES		NULL	
last	varchar(15)	YES		NULL	
phone	char(12)	YES		NULL	
lastcontact	date	YES		NULL	

4 rows in set (0.00 sec)

With the table defined, you can now insert data into it.

20. To add a record for a customer named Jon Snow, enter **INSERT INTO active VALUES ('Jon','Snow','801-756-5555','2016-12-09');**
21. To view the data just added to the table, you use the following syntax:

SELECT field(s) FROM table [WHERE conditions] [ORDER BY field]

For example, to view everything in the active table, you would enter **SELECT * FROM active;** at the mysql prompt. You should see the following data:

```

MariaDB [customers]> SELECT*FROM active;
+-----+-----+-----+-----+
| first | last | phone      | lastcontact |
+-----+-----+-----+-----+
| Jon   | Snow | 801-756-5555 | 2016-12-09 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

You can also modify existing data using the UPDATE command. The syntax is as follows:

UPDATE table_name SET column = new_value WHERE condition

You can enter **UPDATE active SET last = 'Stark' WHERE last = 'Snow';**.

Enter **SELECT*FROM active;** and you should see the last name of one Jon Snow change to Jon Stark. See the below example:

```

MariaDB [customers]> UPDATE active SET last = 'Stark' WHERE last = 'Snow';
Query OK, 1 row affected (0.05 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [customers]> SELECT*FROM active;
+-----+-----+-----+-----+
| first | last | phone      | lastcontact |
+-----+-----+-----+-----+
| Jon   | Stark | 801-756-5555 | 2016-12-09 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

You can also delete data from the table. Here is the syntax:

DELETE FROM table WHERE conditions

For example, to delete the Jon Stark record from the table, you could enter **DELETE FROM active WHERE last='Stark';** at the mysql prompt. Before deleting, however, you should use the equivalent SELECT command first to see what exactly you'll be deleting! For example, first run **SELECT FROM active WHERE last='Stark';** and then run **DELETE FROM active WHERE last='Stark';**. Verify one Jon Stark was removed by entering **SELECT*FROM active;**. The table should be empty now like so:

```
MariaDB [customers]> DELETE FROM active WHERE last='Stark';  
Query OK, 1 row affected (0.10 sec)  
  
MariaDB [customers]> SELECT*FROM active;  
Empty set (0.00 sec)
```

--End of Practical Exercise--