



Conversation with:

Ali Hassan
Database Developer

Database Administration Challenge

Chat Transcript

Ali sent:

Hey @playerone, nice to meet you! I'm told you are the new intern.

Welcome to the TryCyber Bathroom Depot tech division 🙌! I'm Ali Hassan, one of the Database Developers that work here, and I'll be your mentor for today.

Ali sent:

I'm going to be showing you how to perform some core **Database Administration** tasks. I usually perform my tasks on a server like the **Ubuntu 22.04 Linux** server you are looking at now.

Ali sent:

All the tasks you'll be performing today will require you to use the **Terminal**. If you're unfamiliar with using a terminal, that's fine! I can bring you up to speed with a brief tutorial on it.

Ali sent:

Shall we get started? 😊

Participant sent:
Terminal tutorial, please!

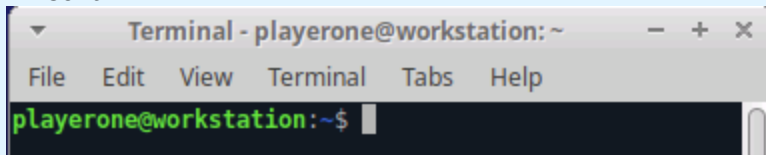
Ali sent:

The **Terminal** is primarily used to access an application called a shell. We interact with shells via a **command-line interface (CLI)** where we input and execute text-based commands on a system.

Ali sent:

Let's open the Terminal on this system now and run a few basic commands so you get the hang of it. You can open the **Terminal** by double-clicking the icon on the desktop named **Terminal Emulator**. It should look like this once you have it open...

Ali sent:



Participant sent:
I've opened the Terminal.

Ali sent:

Great. Basically, you type commands into the shell within the Terminal and then hit **Enter** or **Return** on the keyboard to run the command.

Ali sent:

To properly utilize any CLI command, you must understand its format, options, and other arguments. The **format** defines the structure and order of options and other arguments, the **options** define or modify behavior, and the **arguments** are command defined input types (e.g., files, directories, users, software packages, etc.).

Ali sent:

Additionally, all CLI commands on this system are case-sensitive, so pay close attention to the capitalization of letters.

Ali sent:

The basic command format is...

```
command [OPTIONS] [ARGUMENTS]
```

However, it varies a lot from command to command, and options and other arguments are not always required.

Ali sent:

An example of a command that does not require any options or other arguments is `whoami`. If you type `whoami` into the shell and then hit `Enter` or `Return` on the keyboard, the `whoami` command will output the name of the user running the command into the Terminal. It should look like this if you run that command...

Ali sent:

```
playerone@workstation:~$ whoami  
playerone
```

Participant sent:

What's a more involved example?

Ali sent:

I'll give you a more complex example using the `ls` command.

Ali sent:

In this example, we will use the `ls` command to list detailed information about the entire contents of the `Templates` directory (i.e., folder) in your (playerone's) `home` directory. Our example `ls` command will use the following format...

```
ls [OPTIONS] [DIRECTORY]
```

Note that in this case, the command's argument requires the input to be a directory.

Ali sent:

The actual command we want to run is...

```
ls -a -l /home/playerone/Templates
```

Ali sent:

The `-a` and the `-l` are both options that modify the `ls` command's behavior, and `/home/playerone/Templates` is our argument which is a directory provided in the form of a path.

Ali sent:

For this command's argument, we must provide the **path** to the directory from the root of the file system; otherwise, the system won't know which directory named `Templates` we are referring to.

Ali sent:

And finally, when you run that command, the output should look like this...

Ali sent:

```
playerone@workstation:~$ ls -a -l /home/playerone/Templates/
total 24
drwxr-xr-x  2 playerone playerone 4096 Jul 12 21:36 .
drwxr-x--- 14 playerone playerone 4096 Jul 12 21:36 ..
-rw-r--r--  1 playerone playerone 6859 Apr  9 2022 'OpenDocument Spreadsheet.ods'
-rw-r--r--  1 playerone playerone 7388 Apr  9 2022 'OpenDocument Text.odt'
-rw-r--r--  1 playerone playerone   0 Apr  9 2022 'Plain Text.txt'
```

Participant sent:
Got it. Any extra notes?

Ali sent:

The Terminal and shells are incredibly powerful and versatile tools. Not all commands and programs follow the general structure I've provided you with here. Unfortunately, we only have time to cover the basics, but I do have a few last things I'd like to mention.

Ali sent:

Some commands will not print visual output to the Terminal in normal operation, such as the `cp` command, which is used to copy files and directories.

Ali sent:

Many commands can use multiple arguments, handle multiple input types for arguments (e.g., file and/or directory paths), or have options that will have their own arguments.

Ali sent:

Some options can, or must be, written in a long-form format (e.g., `ls --all [DIRECTORY]` is the same as `ls -a [DIRECTORY]`).

Ali sent:

Options without arguments can often be provided together and in any order (e.g., `ls -la [DIRECTORY]` is the same as `ls -a -l [DIRECTORY]`).

Ali sent:

And last, but certainly not least, you can almost always reference a command's format, options, and other arguments using the command `man [COMMAND]` (e.g., `man cp`) to view the provided command's manual page in the Terminal.

Ali sent:

Hopefully that was not too much information! I know it seems like a lot, but it gets easier the more you use it. For today's tasks, I'll be sure to provide you with more details for any commands and programs you'll need.

Participant sent:

Sounds good! I'm ready to get started!

Ali sent:

Awesome! Today, we'll be working with a **MySQL database**. MySQL is an open-source relational database management system, and a popular one at that.

Ali sent:

We are planning on using a MySQL database as part of a new application our division is developing. This application will need to access, store, and update a variety of information associated with our company's retail locations.

Ali sent:

That is where we come in. 😊

Ali sent:

That application needs to query a database for all the information. So, we are going to start setting one up for the application developers in our tech division. That way, they have a database to develop and test their application with.

Ali sent:

Today, I'd like your help with two tasks, each of which involves importing backups of test data I've made into the MySQL database on the Ubuntu Linux server you have in front of you.

Ali sent:

Backing up data from and restoring data into databases are some of the most important Database Administration duties!

Participant sent:

Good to know! Where do we start?

Ali sent:

We won't be starting from the very beginning. I've taken care of a little setup for us already, including installing the database software (MySQL) on the server, provisioning your user account (i.e., playerone) on the server and the database, and creating a database within MySQL to hold the test data named `inventory`.

Ali sent:

The first task I want your help with is **importing a backup** of a test database's tables I made for the application into the `inventory` database within MySQL.

Ali sent:

I've placed a **backup** of the test database's tables in the **Materials** folder on your **Desktop**. The **backup** is named **inventory_tables.sql**. The path to the backup file **inventory_tables.sql** is...

```
/home/playerone/Desktop/Materials/inventory_tables.sql
```

Participant sent:
Noted.

Ali sent:

But, before we do that, or anything else, let's **log in** to the **MySQL Command-Line Client's interactive SQL shell** and set the default database to the **inventory** database, since we will be using both of those for our tasks today. You can do this by running the following **mysql** command in the **Terminal**...

```
mysql -u playerone -p inventory
```

Ali sent:

When you run this command, you will be prompted for your password. Just enter your password into the prompt and hit **Enter** or **Return** on the keyboard when you're done. While typing your password into this prompt, you will NOT see any indication that you are typing in the Terminal; this is normal behavior. (Note: Your password can be found on the Info Tab)

Ali sent:

Once you're logged in, you should see something like this...

Ali sent:

```
Terminal -
File Edit View Terminal Tabs Help
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 155
Server version: 8.0.33-0ubuntu0.22.04.2 (Ubuntu)

Copyright (c) 2000, 2023, 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>
```

Participant sent:
I'm in!

Ali sent:
Great!

Ali sent:

The MySQL Command-Line Client's interactive SQL shell is a simple SQL shell that runs text-based commands not unlike the ones ran in the default Terminal shell.

Ali sent:

Now, before we import that backup of the test database's tables, let's check to make sure the `inventory` database is empty.

Ali sent:

To do that, run the following **SQL command** in the SQL shell to see what's in the `inventory` database...

```
show tables;
```

Ali sent:

The output should look like this if the `inventory` database is empty...

Ali sent:

```
mysql> show tables;
Empty set (0.00 sec)

mysql> |
```

Ali sent:

Now that we've confirmed the database we are working with is empty, I'd say it is about time we **import that backup!**

Ali sent:

While there are multiple ways to import the backup, we want to continue using the **MySQL Command-Line Client's interactive SQL shell**, so we're going to use this SQL shell's `source` command to do the import.

Ali sent:

Are you ready to get our first backup imported?

Participant sent:
I'm ready!

Ali sent:



Ali sent:

Alright! Let's import the test database's tables from the `inventory_tables.sql` backup file into the `inventory` database. To do that, run the following `source` command in your SQL shell...

```
source /home/playerone/Desktop/Materials/inventory_tables.sql;
```

Ali sent:

After you've run that `source` command, you will see a bunch of output stating that queries were run. That is the import being completed.

Ali sent:

Once that has finished, you can verify the import was successful by running the `show` command you previously used to see if the `inventory` database was empty...

```
show tables;
```

Ali sent:

You should see a list of all the imported tables in the `inventory` database, which should look like this...

Ali sent:

```
mysql> show tables;
+-----+
| Tables_in_inventory |
+-----+
| deliveries           |
| employee             |
| product              |
| store                |
| type                 |
+-----+
5 rows in set (0.00 sec)

mysql> |
```

Participant sent:
Done!

Ali sent:

Great, let's move on to the last task on the agenda for today. 🤓

Ali sent:

The last thing I need your help with today is importing the test data into the test database tables you just imported into the `inventory` database in the previous task. Lucky for you, the process to import this data will be nearly identical to the process you used to import the database tables. You can even do it from the SQL shell you have open right now.

Ali sent:

Basically, you just need to run another `source` command, but this time you want it to **import the backup file** `inventory_data.sql`, which is also inside the `Materials` folder on your `Desktop`.

Ali sent:

Once you've correctly imported the data in the `inventory_data.sql` into the tables in the `inventory` database, you can verify the import by running the following `select` command in the SQL shell...

```
select * from type;
```

This command will show you the contents of the table named `type` in the `inventory` database.

Ali sent:

The output should look like this after a successful import...

Ali sent:

```
mysql> select * from type;
+-----+-----+
| type_id | name                |
+-----+-----+
| 1       | Toilet              |
| 2       | Shower              |
| 3       | Bathtub             |
| 4       | Faucet              |
| 5       | Soap Dispenser      |
| 6       | Toilet Paper Holder |
| 7       | Shower Curtain      |
| 8       | Toilet Brush        |
| 9       | Bathroom Scale      |
| 10      | Toiletry Organizer  |
| 11      | Bidet               |
+-----+-----+
```

Ali sent:

Annnnd it looks like I'm being called into a meeting. 🙄

Ali sent:

I'm sure you can get this done without me, though. Thanks for helping me out today!

TRY
CYBER

© 2023 Try Cyber - Sponsored by CISA

TRY
CYBER