

LabEx - Intro to MySQL Labs

Introduction to MySQL:

In this lab, we delve into the foundational configuration and usage of MySQL—one of the most widely adopted open-source relational database management systems (RDBMS). Commonly utilized in web applications and data-driven platforms, MySQL offers powerful tools for managing structured data.

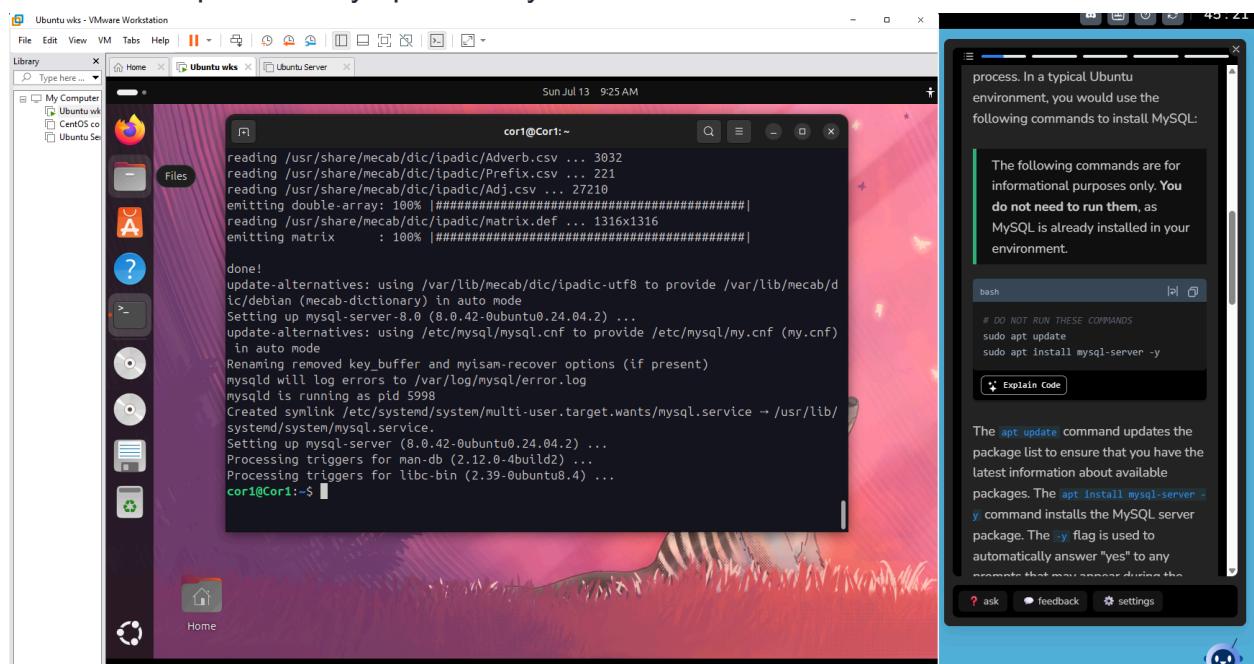
By the end of this lab, you will have hands-on experience with:

- Verifying MySQL installation and securing the initial setup
- Accessing and navigating the MySQL shell
- Viewing system databases and understanding their structure
- Exploring tables within system databases
- Querying data from internal system tables

These skills form a solid groundwork for advancing into real-world database administration, query optimization, and integration tasks.

1. MySQL installation in a typical Ubuntu Environment

- a. sudo apt update
- b. sudo apt install mysql-server -y



LabEx - Intro to MySQL Labs

2. Check the status of MySQL:

a. sudo service mysql status

```
Renaming removed key_buffer and myisam-recover options (if present)
mysqld will log errors to /var/log/mysql/error.log
mysqld is running as pid 5998
Created symlink /etc/systemd/system/multi-user.target.wants/mysql.service → /usr/lib/
systemd/system/mysql.service.
Setting up mysql-server (8.0.42-0ubuntu0.24.04.2) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.4) ...
cor1@Cor1:~$ sudo service mysql status
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-07-13 09:24:55 EDT; 2min 0s ago
       Process: 6267 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited)
      Main PID: 6275 (mysqld)
        Status: "Server is operational"
          Tasks: 37 (limit: 4546)
         Memory: 364.0M (peak: 378.4M)
            CPU: 4.560s
           CGroup: /system.slice/mysql.service
                   └─ 6275 /usr/sbin/mysqld

Jul 13 09:24:53 Cor1 systemd[1]: Starting mysql.service - MySQL Community Server...
Jul 13 09:24:55 Cor1 systemd[1]: Started mysql.service - MySQL Community Server.
[lines 1-14/14 (END)]
```

Once in the terminal, run the following command to check the status of MySQL:

```
bash
sudo service mysql status
Explain Code
```

This command uses `sudo` to run with administrative privileges, which is often required for system-level operations. The `service mysql status` part tasks the system to report on the current state of the MySQL service.

3. Start the MySQL service:

a. sudo service mysql start

4. To access the MySQL shell:

a. sudo mysql -u root

```
Memory: 364.0M (peak: 378.4M)
CPU: 4.560s
CGroup: /system.slice/mysql.service
       └─ 6275 /usr/sbin/mysqld

Jul 13 09:24:53 Cor1 systemd[1]: Starting mysql.service - MySQL Community Server...
Jul 13 09:24:55 Cor1 systemd[1]: Started mysql.service - MySQL Community Server.
cor1@Cor1:~$ sudo service mysql start
cor1@Cor1:~$ SHOW DATABASES;
SHOW: command not found
cor1@Cor1:~$ sudo mysql -u root
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.42-0ubuntu0.24.04.2 (Ubuntu)

Copyright (c) 2000, 2025, 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>
```

Note: Don't forget the semicolon (`;`) at the end of the command. SQL commands must end with a semicolon.

```
mysql> help
+-----+
| Help on the MySQL monitor, and more information about MySQL.|
| You may type '?' or 'help' for help from the command-line.|
| For help, type 'help' or 'man -k help' from the terminal.|
| MySQL comes with ABSOLUTELY NO WARRANTY.|
| This is free software under the GPL.|
| Type 'quit' to exit MySQL.|
+-----+
```

This command lists all the databases present in your MySQL server. You should see output similar to this:

```
plaintext
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
```

LabEx - Intro to MySQL Labs

5. Lists all the databases present in my MySQL server:

- SHOW DATABASES

The screenshot shows a Linux desktop environment with a terminal window open. The terminal displays the MySQL monitor, showing the command 'SHOW DATABASES;' and its output, which lists the databases: information_schema, mysql, performance_schema, and sys. To the right of the terminal is a help panel titled 'In the MySQL shell, run the following command:' with a code editor containing the same SQL command. Below the code editor is a note: 'Note: Don't forget the semicolon (;) at the end of the command. SQL commands must end with a semicolon.' A preview window shows the expected output of the command.

6. The mysql database:

- USE mysql;

The screenshot shows a Linux desktop environment with a terminal window open. The terminal shows the command 'USE mysql;' being run, followed by 'SHOW TABLES;'. The output lists various tables such as columns_priv, db, engine_cost, func, general_log, global_grants, gtid_executed, help_category, help_keyword, help_relation, help_topic, innodb_index_stats, and others. To the right of the terminal is a help panel for the 'performance_schema' database, which provides a way to inspect internal execution of the server at runtime. It notes that the 'sys' database contains objects for DBAs and developers. A section for the 'mysql' database is shown, stating 'For this lab, we'll focus on the mysql database. Let's switch to it:' with a code editor showing 'USE mysql;'. Below this, a note says 'Now, let's see what tables are in this database:' with a code editor showing 'SHOW TABLES;'. A preview window shows the expected output of the command.

LabEx - Intro to MySQL Labs

7. See what tables are in this database:

a. SHOW TABLES;

The screenshot shows a Linux desktop environment with a terminal window open. The terminal displays the output of the MySQL command `SHOW TABLES;`. The output lists numerous system tables starting with `Tables_in_mysql`, `columns_priv`, `component`, `db`, and so on, up to `replication_asynchronous_connection_failover_managed`.

On the right side of the screen, there is a LabEx interface window titled "Performance Schema". It contains a sidebar with information about the `sys` database and a main panel with a MySQL query editor. The query editor has the following code:

```
sql
USE mysql;
Explain Code
```

Below the query editor, a note says: "Now, let's see what tables are in this database:" followed by another query editor with the command `SHOW TABLES;` and an "Explain Code" button.

8. To see the structure of the user table:

a. DESCRIBE user;

The screenshot shows a Linux desktop environment with a terminal window open. The terminal displays the output of the MySQL command `DESCRIBE user;`. The output shows the structure of the `user` table, listing 37 columns with their names, types, and various properties like Nullability, Key status, and Default values.

On the right side of the screen, there is a LabEx interface window titled "Labby". It contains a sidebar with information about the `user` table and a main panel with a MySQL query editor. The query editor has the following code:

```
sql
DESCRIBE user;
Explain Code
```

Below the query editor, a note says: "This will show you all the columns in the `user` table. You'll see a lot of columns, as this table contains detailed information about each MySQL user." followed by another query editor with the command `DESCRIBE user;` and an "Explain Code" button.

LabEx - Intro to MySQL Labs

9. Important columns from the user table: User, Host, Password_expired.

a. SELECT User, Host, Password_expired FROM user;

The screenshot shows a Linux desktop environment with a terminal window and a MySQL Workbench interface. The terminal window displays the output of a MySQL query:

```
cor1@Cor1:~$ mysql -u root -p
Enter password: 
mysql> SELECT User, Host, Password_expired FROM user;
+-----+-----+-----+
| User | Host | Password_expired |
+-----+-----+-----+
| debian-sys-maint | localhost | N
| mysql.infoschema | localhost | N
| mysql.session | localhost | N
| mysql.sys | localhost | N
| root | localhost | N
+-----+-----+-----+
5 rows in set (0.01 sec)

mysql>
```

The MySQL Workbench interface on the right shows a query editor with the same SQL statement and its execution results. A sidebar provides a summary of the columns selected:

Now, let's look at some of the data in this table. We'll focus on a few key columns:

```
sql
Juser, Host, Password_expired FROM user;
Explain Code
```

This query selects four important columns from the `user` table:

- `User`: The username of the MySQL account
- `Host`: The host from which this user is allowed to connect
- `Password_expired`: Whether the password has expired

You should see output similar to this:

```
plaintext
+-----+-----+
| User | Host | password_e |
+-----+-----+
|          |       |          |
|          |       |          |
|          |       |          |
|          |       |          |
|          |       |          |
+-----+-----+
```

10. MySQL with Docker – Lab Summary

This lab focused on mastering essential MySQL database management tasks within a Dockerized environment. Key accomplishments included:

- Verifying and securing MySQL installation
- Checking and managing MySQL service status
- Accessing the MySQL shell interface
- Exploring system databases and internal tables
- Querying metadata from system tables

These foundational skills pave the way for more advanced MySQL operations—such as designing custom databases, writing complex queries, and integrating MySQL with applications. Continued practice will deepen database fluency and enhance system-level troubleshooting capabilities.