Objectives

After completing this lab, you will be able to use the MySQL command line interface to:

- Create tables using alternative storage engines.
- Query MySQL system tables to retrieve meta data about objects in the database.

theia@theiadocker-rajendraabro:/home/project$ wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0231EN-SkillsNetwork/datasets/World/world_mysql_script.sql

HTTP request sent, awaiting response... 200 OK
Length: 389702 (381K) [application/x-sql]
Saving to: 'world_mysql_script.sql.1'

world_mysql_ 100% 380.57K --.-KB/s   in 0.004s

2025-12-23 16:21:37 (103 MB/s) - 'world_mysql_script.sql.1' saved [389702/389702]

mysql> CREATE DATABASE worldd;
Query OK, 1 row affected (0.02 sec)

mysql> USE world;

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

Execute the world mysql script (world_mysql.sql) to complete the world database

mysql>SOURCE world_mysql_script.sql;

mysql> SHOW TABLES;
+-----------------+
| Tables_in_world |
+-----------------+
| city |
| country |
| countrylanguage |
+-----------------+

To create a new table with a storage engine other than the default InnoDB database, we specify the storage engine we wish to use inside the CREATE TABLE command. Let's create a new table called "test_csv" using the CSV engine by entering the following command into the CLI:3 rows in set (0.00 sec)

```
mysql> SHOW ENGINES;
+--------------------+---------+----------------------------------------------------------------+--------------+------+----------
--+
| Engine             | Support | Comment                                                        | Transactions | XA   |
Savepoints |
+--------------------+---------+----------------------------------------------------------------+--------------+------+----------
--+
| ndbcluster         | NO      | Clustered, fault-tolerant tables                               | NULL         | NULL |
NULL       |
| MEMORY             | YES     | Hash based, stored in memory, useful for temporary tables      | NO
| NO   | NO         |
| InnoDB             | DEFAULT | Supports transactions, row-level locking, and foreign keys     | YES
| YES  | YES        |
| PERFORMANCE_SCHEMA | YES     | Performance Schema                                             | NO
| NO   | NO         |
| MyISAM             | YES     | MyISAM storage engine                                          | NO           | NO   |
NO         |
| FEDERATED          | NO      | Federated MySQL storage engine                                 | NULL         |
NULL | NULL       |
| ndbinfo            | NO      | MySQL Cluster system information storage engine                | NULL         |
NULL | NULL       |
| MRG_MYISAM         | YES     | Collection of identical MyISAM tables                          | NO           |
NO   | NO         |
| BLACKHOLE          | YES     | /dev/null storage engine (anything you write to it disappears) | NO
| NO   | NO         |
| CSV                | YES     | CSV storage engine                                             | NO           | NO   | NO
|
| ARCHIVE            | YES     | Archive storage engine                                         | NO           | NO   | NO
|
+--------------------+---------+----------------------------------------------------------------+--------------+------+----------
--+
11 rows in set (0.00 sec)

mysql> CREATE TABLE csv_test (i INT NOT NULL, c CHAR(10) NOT NULL) ENGINE = CSV;
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> SELECT * FROM csv_test;
Empty set (0.01 sec)

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| world              |
| worldd             |
+--------------------+
6 rows in set (0.01 sec)
```

The user table contains user accounts, global privileges, and other nonprivilege columns.

```
mysql> SELECT User from user;
+------------------+
| User             |
+------------------+
| root             |
| mysql.infoschema |
| mysql.session    |
| mysql.sys        |
| root             |
+------------------+
5 rows in set (0.00 sec)
```

add a new user to the database

```
mysql> CREATE USER test_user;
Query OK, 0 rows affected (0.01 sec)
```

Query the `INFORMATION_SCHEMA` Database Tables

The `information_schema` database is the `TABLES` table which contains meta data about all the tables in the server. One of the columns in this table contains information about a table's storage engine type.

In the CLI to view the storage engine type for the 'country', 'city', 'countrylanguage', and finally the 'csv_test' table you created:

```
mysql> SELECT table_name, engine FROM INFORMATION_SCHEMA.TABLES
    -> WHERE table_name = 'country' OR table_name = 'city'
    -> OR table_name = 'countrylanguage' OR table_name = 'csv_test';
+----------------+--------+
| TABLE_NAME     | ENGINE |
+----------------+--------+
| city           | InnoDB |
| country        | InnoDB |
| countrylanguage | InnoDB |
| csv_test       | CSV    |
+----------------+--------+
4 rows in set (0.00 sec)
```

TABLES table in the information_schema database contains information on the the size of a given table in bytes. This information is stored in two columns: data_length and index_length which stores the size of the data in the table and the size of the index file for that table, respectively. Therefore, the total size of the table is the sum of the values in these two columns. This value would be given in bytes,

```
mysql> SELECT table_name, (data_length + index_length)/1024 FROM
INFORMATION_SCHEMA.TABLES
    -> WHERE table_name = 'country' OR table_name = 'city'
    -> OR table_name = 'countrylanguage' OR table_name = 'csv_test';
+----------------+---------------------------------+
| TABLE_NAME     | (data_length + index_length)/1024 |
+----------------+---------------------------------+
| city           |                       544.0000 |
| country        |                        96.0000 |
| countrylanguage |                       160.0000 |
| csv_test       |                         0.0000 |
+----------------+---------------------------------+
4 rows in set (0.04 sec)
```