Creating the `COMPANY` database on Linux

This guide serves as a how-to on getting the COMPANY database running on Linux. It's pretty much identical to the guide provided for Windows, but I want to be thorough here.

1. Ensure MySQL is running on your system.

You can do this with the following command, and expected output:

```
$ systemctl status mysqld

mysqld.service - MySQL 8.0 database server
    Loaded: ...
    Drop-In: ...
    Active: active (running) since ...
    ...
```

The import thing is that it should be **active.** If it isn't, you need to start the `mysqld` service.

2. Connect to the MySQL monitor.

You can do this with the `mysql` command, as shown in the previous guide:

```
$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g
Your MySQL connection id is 10
Server version: 8.0.42 Source distribution

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> _
```

3. Create the `COMPANY` Database

```
mysql> CREATE DATABASE COMPANY;
```

4. Use `COMPANY` as the Implicit Default Database

This allows you to omit `COMPANY` when editing tables within the database, e.g. `EMPLOYEE` instead of `COMPANY.EMPLOYEE`.

```
mysql> USE COMPANY;
```

You can ensure that you selected the right database as follows:

```
mysql> SELECT DATABASE();
+------------+
| DATABASE() |
+------------+
| COMPANY    |
```

```
+------------+
1 row in set (0.00 sec)
```

5. Use the provided MySQL script to create the COMPANY database schema

This assumes you have downloaded a file named `COMPANY_create.sql`. You'll need to know the absolute path to this file. For example, if your user is `johndoe` and you placed it in your downloads, the absolute path will be `/home/johndoe/Downloads/COMPANY_create.sql`. Most file browsers (Dolphin, Nautilus, Nemo, etc.) will allow you to copy the absolute path from the path view at the top.

```
mysql> source /path/to/COMPANY_create.sql
Query OK, 0 rows affected (0.04 sec)

Query OK, 0 rows affected (0.04 sec)

Query OK, 0 rows affected (0.04 sec)

Query OK, 0 rows affected (0.04 sec)

Query OK, 0 rows affected (0.04 sec)

Query OK, 0 rows affected (0.04 sec)
```

You can ensure that the tables were properly created with the following commands:

```
mysql> SHOW TABLES;
+------------------+
| Tables_in_COMPANY |
+------------------+
| DEPARTMENT        |
| DEPENDENT         |
| DEPT_LOCATIONS    |
| EMPLOYEE          |
| PROJECT           |
| WORKS_ON          |
+------------------+
6 rows in set (0.00 sec)

mysql> DESCRIBE WORKS_ON;
+-------+---------------+------+-----+---------+-------+
| Field | Type          | Null | Key | Default | Extra |
+-------+---------------+------+-----+---------+-------+
| Essn  | char(9)       | NO   | PRI | NULL    |       |
| Pno   | int           | NO   | PRI | NULL    |       |
| Hours | decimal(4, 2) | NO   |     | NULL    |       |
+-------+---------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

You're free to `DESCRIBE` any other tables as a sanity check.

6. Load table data into database.

This will only work if you allowed `local_infile` in the MySQL configuration file from the earlier guide. Much like the previous section, you'll need to use the absolute path for each of the files.

```
mysql> LOAD DATA LOCAL INFILE '/path/to/EMPLOYEE.txt' INTO TABLE EMPLOYEE;
Query OK, 40 rows affected, 2 warnings (0.01 sec)
Records 40  Deleted: 0  Skipped: 0  Warnings: 0
```

Running `SELECT * FROM EMPLOYEE;` should print out a 40-row table of various employees. Do this for the remaining five tables (`DEPARTMENT`, `DEPT_LOCATIONS`, `PROJECT`, `WORKS_ON`).

If you received warnings (i.e. if the `Warnings: X` has any number other than 0) on loading in any of the tables, check to see if it was similar to the following:

```
mysql> SHOW WARNINGS;
+-------+------+-----------------------------------------------------+
| Level | Code | Message                                             |
+-------+------+-----------------------------------------------------+
| Note  | 1265 | Data truncated for column '<column_name>' at row X |
+-------+------+-----------------------------------------------------+
```

If so, you can fix it with the following two commands, where `TABLE_NAME` is the table that gave you problems (e.g. `DEPENDENT`):

```
mysql> TRUNCATE TABLE TABLE_NAME;
Query OK, 0 rows affected (0.05 sec)

mysql> LOAD DATA LOCAL INFILE '/path/to/TABLE_NAME.txt' INTO TABLE TABLE_NAME LINES
TERMINATED BY '\r\n';
Query OK, XX rows affected (0.02 sec)
Records: 00  Deleted: 0  Skipped: 0  Warnings: 0
```

Tools used to build this guide

The following are a list of tools I used to build the final PDF form of this guide. Largely meant to be thorough, you're free to use these as jumping-off points to make your own edits:

- Markdown: The markup language I used to create the guide.
- NeoVim: Primary text editor for the original markdown.
- Pandoc: A "compiler" for markdown that makes the output PDF nice and fancy.

Authors

Marcial Machado

License