

Step 1: First, we must have the normalized tables put into csv files. We can do this by adding this to the end of the ipynb file. For example, df_provider is one of the normalized tables I have so this will be a new csv file.

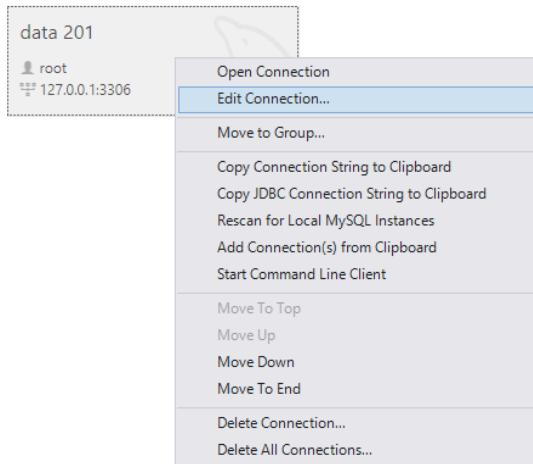
```
# Export each DataFrame to CSV
df_provider.to_csv('provider.csv', index=False)
df_address.to_csv('provider_address.csv', index=False)
df_classification.to_csv('provider_classification.csv', index=False)
df_service_totals.to_csv('service_totals.csv', index=False)
df_drug_services.to_csv('drug_services.csv', index=False)
df_medical_services.to_csv('medical_services.csv', index=False)
df_bene_demo.to_csv('beneficiary_demographics.csv', index=False)
df_bene_cc.to_csv('beneficiary_conditions.csv', index=False)

print("All normalized tables exported successfully!")
```

This will take a bit to run but afterwards these csv files will be in the directory your ipynb file is.

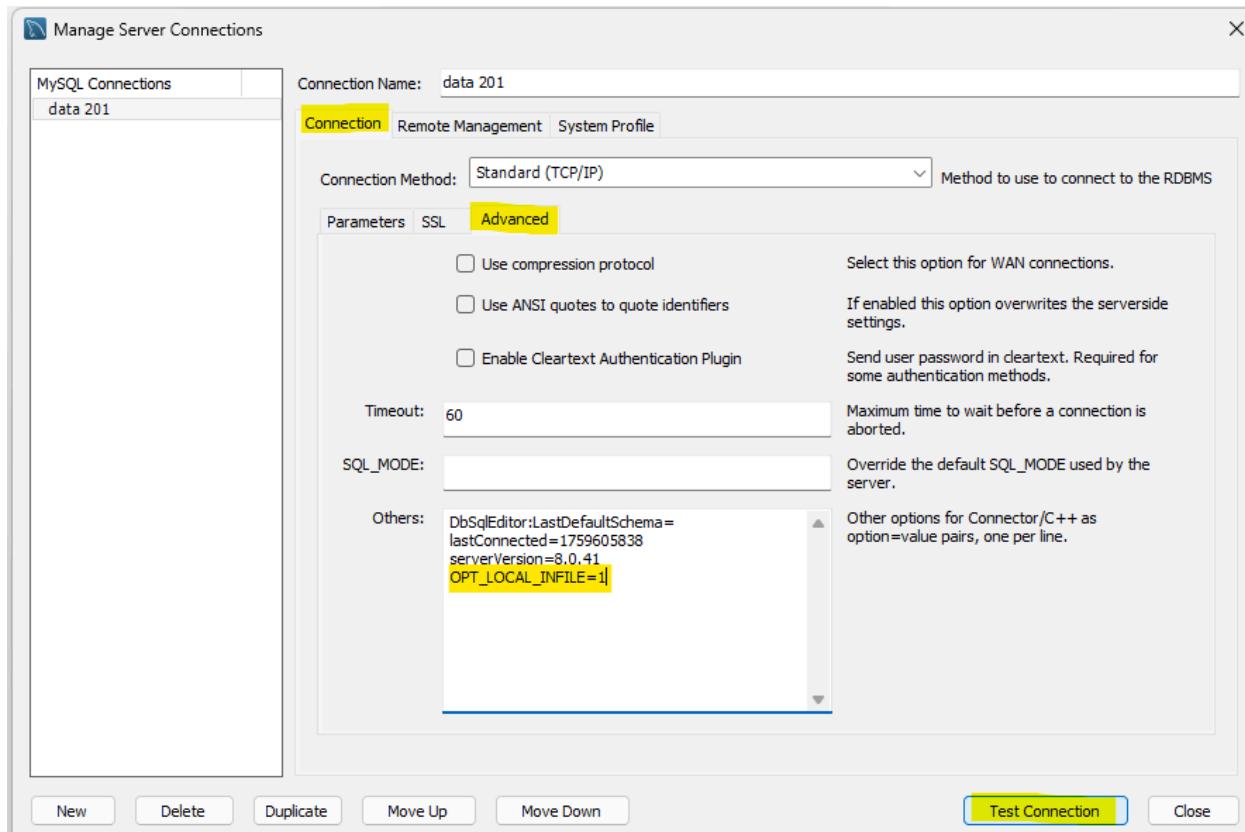
Step 2: Open MySQL and edit the connection you are working in to allow the ability to load in csv files. Right click on the connection you are using, and click on the edit connection option.

MySQL Connections

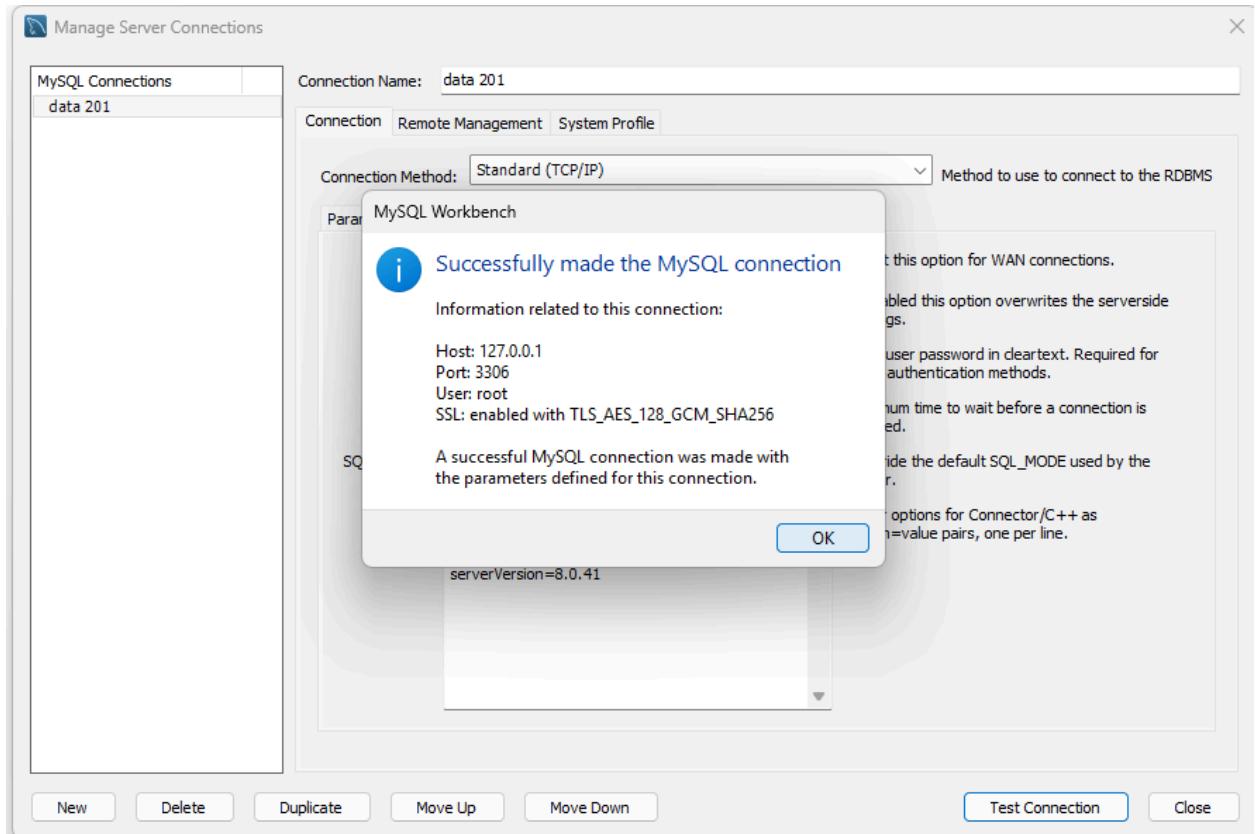


Step 3: Click on the Advanced tab and add the following to the Others section. This allows us to directly import csv file data to our database

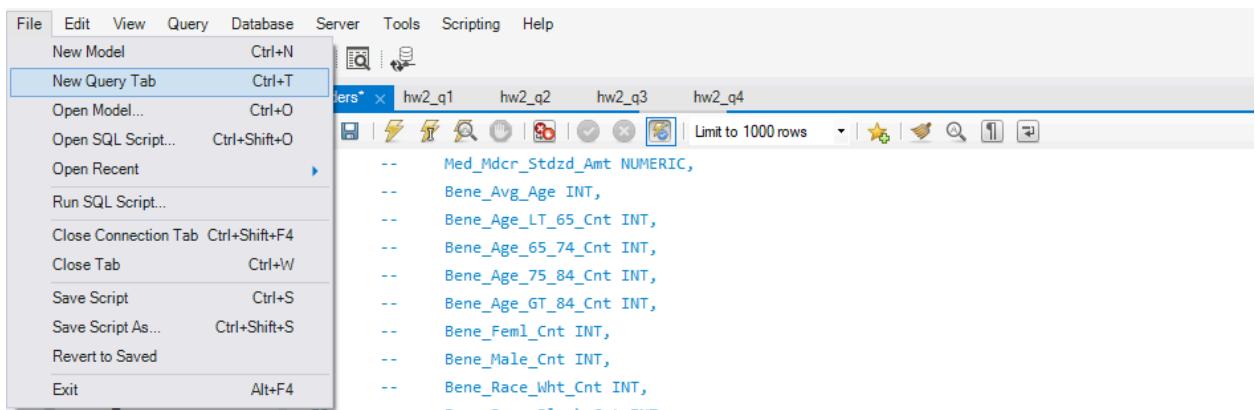
```
OPT_LOCAL_INFILE=1
```



Step 4: After adding that you must click on the Test Connection Button. The following prompt should pop up afterwards and that means it was successful. You can click Okay afterwards and close the window.



Step 5: Open the connection like usual by double clicking on it and open a new Query tab.



Step 6: On this query tab, you will create all the tables and database you will use (we will be using the same database for all tables from your ipynb file and the other datasets)
For example:

The screenshot shows a MySQL Workbench interface with a query editor containing the following SQL code:

```
1 •  DROP DATABASE IF EXISTS cms;
2 •  CREATE DATABASE cms;
3 •  USE cms;
4
5 •  CREATE TABLE provider (
6      Rndrng_NPI INT,
7      Rndrng_Prvdr_Last_Org_Name TEXT,
8      Rndrng_Prvdr_First_Name TEXT,
9      Rndrng_Prvdr_MI TEXT,
10     Rndrng_Prvdr_Crdntls TEXT,
11     Rndrng_Prvdr_Ent_Cd TEXT,
12     Year INT,
13     PRIMARY KEY (Rndrng_NPI, Year)
14 );
15
16 •  CREATE TABLE provider_address (
17     Rndrng_NPI INT,
18     Rndrng_Prvdr_St1 TEXT,
19     Rndrng_Prvdr_St2 TEXT,
20     Rndrng_Prvdr_City TEXT,
21     Rndrng_Prvdr_State_Abrvtn TEXT,
22     Rndrng_Prvdr_State_FIPS TEXT,
23     Rndrng_Prvdr_Zip5 TEXT,
24     Rndrng_Prvdr_Cntry TEXT,
25     Year INT,
26     PRIMARY KEY (Rndrng_NPI, Year)
27 );
```

Step 7: Create another new Query tab and Copy and paste this line and run it to get the path we will be pasting the csv files with the normalized tables. You will get the following output.

```
SHOW VARIABLES LIKE 'secure_file_priv';
```

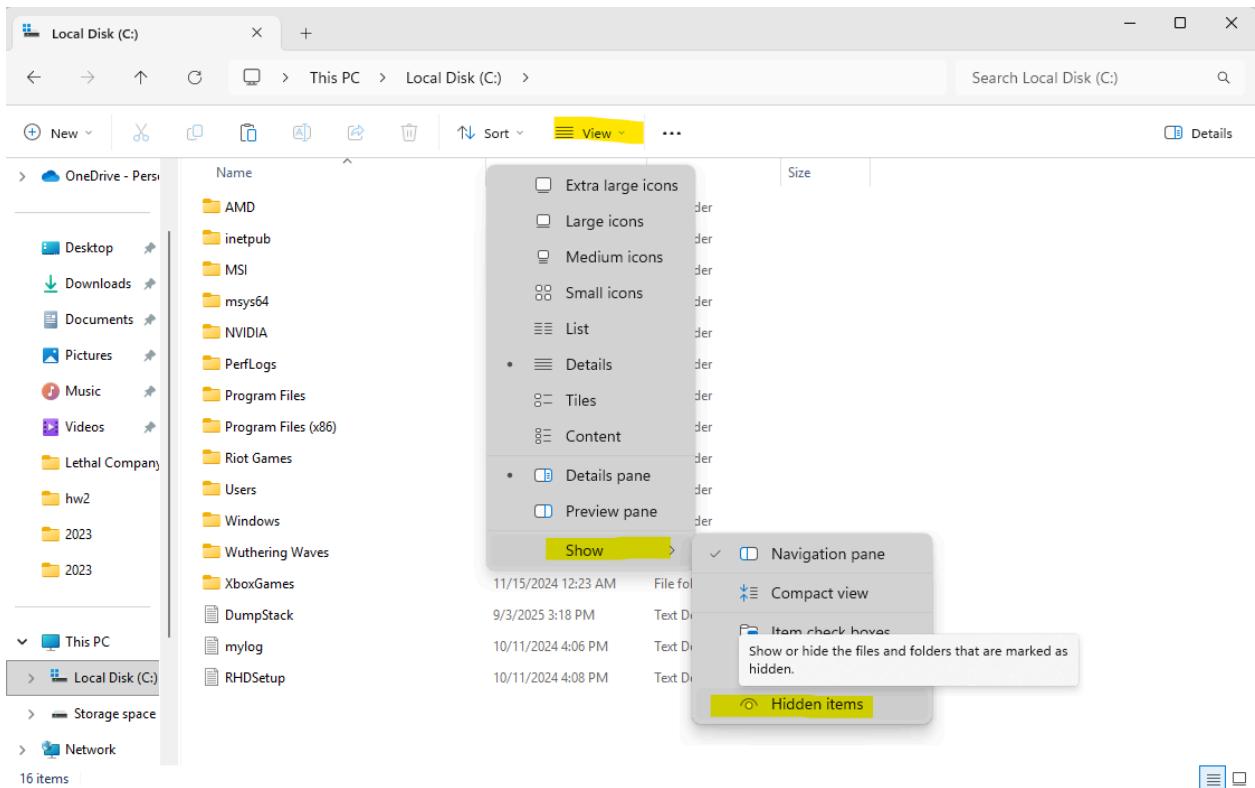
The screenshot shows the MySQL Workbench interface with a query editor at the top containing the SQL command:

```
1 • SHOW VARIABLES LIKE 'secure_file_priv';
```

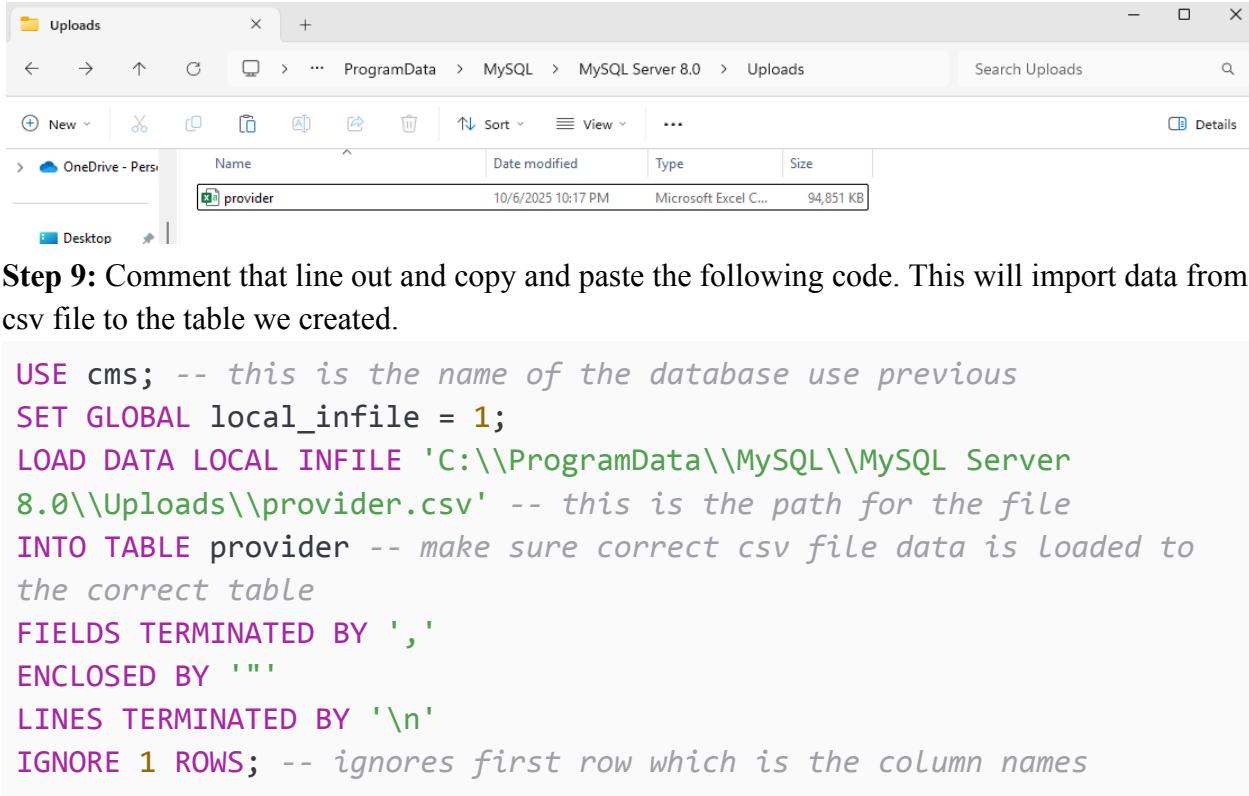
Below the query editor is a results grid titled "Result Grid". The grid has two columns: "Variable_name" and "Value". There is one row returned, showing:

Variable_name	Value
secure_file_priv	C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\

Step 8: Go to file explorer and make sure you can view hidden files by clicking on View→Show→Hidden items. This makes it so you can access the Program Data file if it is not there before.



Step 9: Copy and paste all csv files with normalized tables on the file location path we got before. Program Data→MySQL→MySQL Server 8.0 →Uploads



Step 9: Comment that line out and copy and paste the following code. This will import data from csv file to the table we created.

```
USE cms; -- this is the name of the database use previous
SET GLOBAL local_infile = 1;
LOAD DATA LOCAL INFILE 'C:\\\\ProgramData\\\\MySQL\\\\MySQL Server
8.0\\\\Uploads\\\\provider.csv' -- this is the path for the file
INTO TABLE provider -- make sure correct csv file data is loaded to
the correct table
FIELDS TERMINATED BY ','
ENCLOSED BY ""
LINES TERMINATED BY '\n'
IGNORE 1 ROWS; -- ignores first row which is the column names
```

After running it you should get that it successfully has imported data like the following screenshot. Change the table name and csv name you are importing to repeat process for all tables you have. You can double check that the rows all got imported correctly in the yellow highlighted section.

The screenshot shows the MySQL Workbench command history and execution results. The command history at the top includes:

```
1 -- SHOW VARIABLES LIKE 'secure_file_priv';
2 USE cms; -- this is the name of the database use previous
3 SET GLOBAL local_infile = 1;
4 DROP DATABASE cms;
5 CREATE DATABASE cms;
6 INTO TABLE provider ... -- this is the path for the file
7 FIELDS TERMINATED BY ','
8 ENCLOSED BY ""
9 LINES TERMINATED BY '\n'
10 IGNORE 1 ROWS; -- ignores first row which is the column names
```

The execution results table shows the following rows:

Action	Time	Action	Message	Duration / Fetch
1	12:10:50	SHOW VARIABLES LIKE 'secure_file_priv';	1 row(s) returned	0.016 sec / 0.000 sec
2	12:21:36	DROP DATABASE IF EXISTS cms;	3 row(s) affected	0.047 sec
3	12:21:36	CREATE DATABASE cms;	0 row(s) affected	0.000 sec
4	12:21:36	USE cms;	0 row(s) affected	0.000 sec
5	12:21:36	CREATE TABLE provider (Rdng_NPI INT, Rdng_Prov_Loc_Org_Name TEXT, Rdng_Prov_Prov_Name TEXT, Rdng_Prov_Prov_Ctrct TEXT, Rdng_Prov_Prov_EstLcd TEXT);	0 row(s) affected	0.033 sec
6	12:21:36	CREATE TABLE provider_address (Rdng_NPI INT, Rdng_Prov_21 TEXT, Rdng_Prov_22 TEXT, Rdng_Prov_City TEXT, Rdng_Prov_State_Abrv TEXT, Rdng_Prov_State_STPS TEXT);	0 row(s) affected	0.015 sec
7	12:21:36	CREATE TABLE provider_classification (Rdng_NPI INT, Rdng_Prov_RUCA_Desc TEXT, Rdng_Prov_Prov_Type TEXT, Rdng_Prov_Prov_Proctc_Ind TEXT, Year INT);	0 row(s) affected	0.031 sec
8	12:21:36	CREATE TABLE provider_benefits (Rdng_NPI INT, Tot_HCPCS_Cds INT, Tot_Benes INT, Tot_Srvcs FLOAT, Tot_Shrtfd_Chp FLOAT, Tot_Mbrs_Nbrd_Amt FLOAT, Tot_Mbrs_Pmt_Amt FLOAT);	0 row(s) affected	0.016 sec
9	12:21:36	CREATE TABLE provider_services (Rdng_NPI INT, Drug_Spcn_Ind TEXT, Drug_Tot_HCPCS_Cds FLOAT, Drug_Tot_Srvcs FLOAT, Drug_Shrtfd_Chp FLOAT, Drug_Mbrs_Amt);	0 row(s) affected	0.016 sec
10	12:21:36	CREATE TABLE provider_medications (Rdng_NPI INT, Med_Spcn_Ind TEXT, Med_Tot_HCPCS_Cds FLOAT, Med_Tot_Benes FLOAT, Med_Shrtfd_Chp FLOAT, Med_Mbrs_Amt);	0 row(s) affected	0.016 sec
11	12:21:36	CREATE TABLE bene_demographics (Rdng_NPI INT, Bene_Avg_Age INT, Bene_Age_LT_50_Crt FLOAT, Bene_Age_50_74_Crt FLOAT, Bene_Age_75_84_Crt FLOAT, Bene_Age_GT_84_Crt FLOAT);	0 row(s) affected	0.016 sec
12	12:21:36	CREATE TABLE bene_clinical_conditions (Rdng_NPI INT, Bene_CC_BH_AdHD_ONCD_V1_Pct FLOAT, Bene_CC_BH_Alcohol_Disg_V1_Pct FLOAT, Bene_CC_BH_Tobacco_V1_Pct FLOAT, Bene_C);	0 row(s) affected	0.031 sec
13	12:21:46	USE cms;	0 row(s) affected	0.000 sec
14	12:21:46	SET GLOBAL local_infile = 1;	0 row(s) affected	0.000 sec
15	12:21:46	LOAD DATA LOCAL INFILE 'C:\\\\ProgramData\\\\MySQL\\\\MySQL Server 8.0\\\\Uploads\\\\provider.csv' -- this is the path for the file INTO TABLE provider -- make sure correct csv file data is loaded to the correct table FIELDS TERMINATED BY ',' ENCLOSED BY "" LINES TERMINATED BY '\n' IGNORE 1 ROWS;	Error Code: 2. File 'C:\\\\ProgramData\\\\MySQL\\\\MySQL Server 8.0\\\\Uploads\\\\provider.csv' not found (OS error 2 - No such file or directory)	0.000 sec
16	12:22:45	USE cms;	0 row(s) affected	0.000 sec
17	12:22:45	SET GLOBAL local_infile = 1;	0 row(s) affected	0.000 sec
18	12:22:45	LOAD DATA LOCAL INFILE 'C:\\\\ProgramData\\\\MySQL\\\\MySQL Server 8.0\\\\Uploads\\\\provider.csv' -- this is the path for the file INTO TABLE provider -- make sure correct csv file data is loaded to the correct table FIELDS TERMINATED BY ',' ENCLOSED BY "" LINES TERMINATED BY '\n' IGNORE 1 ROWS;	2415213 rows affected Records: 2415213 Deleted: 0 Skipped: 0 Warnings: 0	27.032 sec

Note: If there is a yellow warning sign instead of the green check mark, it is also okay as long as you can see the time in seconds on the very left. You can double check if its loaded in by going to step 10.

101	12:21:53	USE cms	0 row(s) affected	0.000 sec
102	12:21:53	SET GLOBAL local_infile = 1	0 row(s) affected	0.000 sec
103	12:21:53	LOAD DATA LOCAL INFILE 'C:\\\\ProgramData\\\\MySQL\\\\MySQL Server 8.0\\\\Uploads\\\\provider.csv' -- this is the path for the file INTO TABLE provider -- make sure correct csv file data is loaded to the correct table FIELDS TERMINATED BY ',' ENCLOSED BY "" LINES TERMINATED BY '\n' IGNORE 1 ROWS;	2415213 rows affected Records: 2415213 Deleted: 0 Skipped: 0 Warnings: 0	27.032 sec

Step 10: You can check everything is done correctly by doing the following.

```
SELECT * FROM provider; -- check the first 1000 rows
SELECT COUNT(*) FROM provider; -- check count of rows
```

12 • SELECT * FROM provider;

Rndrng_NPI	Rndrng_Prvdr_Last_Org_Name	Rndrng_Prvdr_First_Name	Rndrng_Prvdr_MI	Rndrng_Prvdr_Crdnts	Rndrng_Prvdr_Ent_Cd	Year
1003000126	ENKESHAFI	ARDALAN		M.D.	I	2019
1003000126	Enkeshafi	Ardalan		M.D.	I	2023
1003000134	CIBULL	THOMAS	L	M.D.	I	2019
1003000134	Cibull	Thomas	L	M.D.	I	2023
1003000142	KHALIL	RASHID		M.D.	I	2019

12 • SELECT * FROM provider;

13

14 • SELECT COUNT(*) FROM provider;

Rndrng_NPI	Rndrng_Prvdr_Last_Org_Name	Rndrng_Prvdr_First_Name	Rndrng_Prvdr_MI	Rndrng_Prvdr_Crdnts	Rndrng_Prvdr_Ent_Cd	Year
1003000126	ENKESHAFI	ARDALAN		M.D.	I	2019
1003000126	Enkeshafi	Ardalan		M.D.	I	2023
1003000134	CIBULL	THOMAS	L	M.D.	I	2019
1003000134	Cibull	Thomas	L	M.D.	I	2023
1003000142	KHALIL	RASHID		M.D.	I	2019

| Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows: |

| COUNT(*) | 2415213 |