

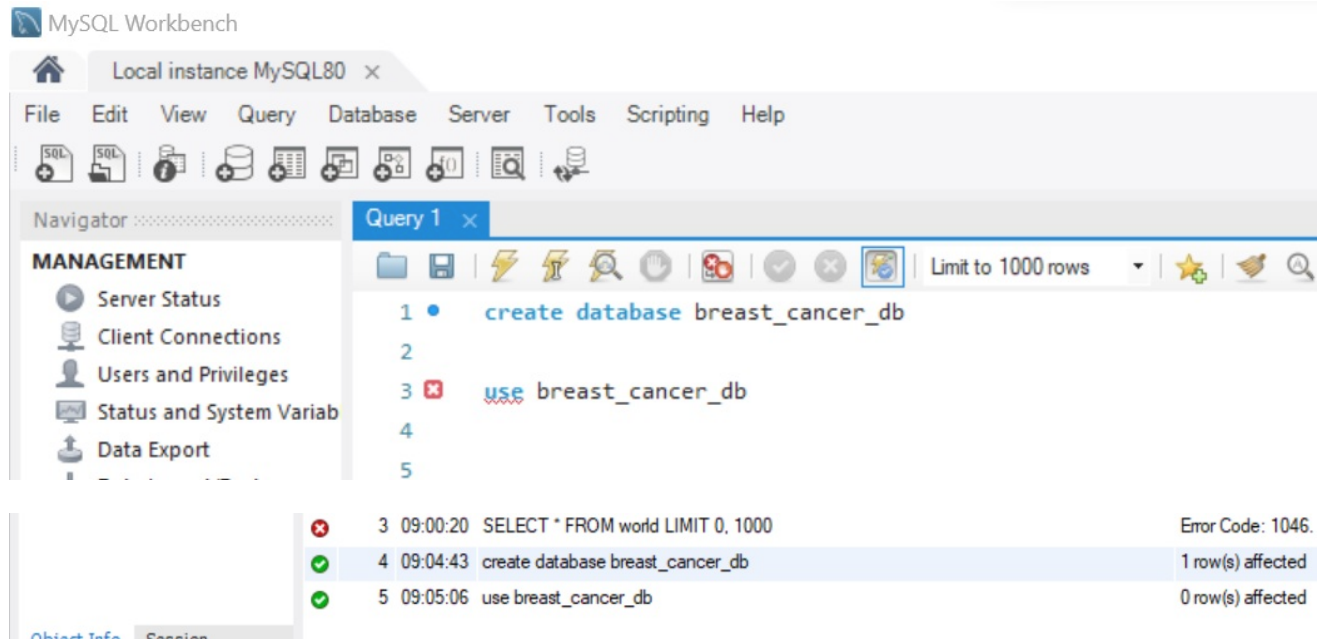
Working with SQL

Python Connector

```
In [1]: !pip install pymysql
```

```
Collecting pymysql
  Obtaining dependency information for pymysql from https://files.pythonhosted.org/packages/e5/30/20467e39523d0cfc2b6227902d3687a16364307260c75e6a1cb4422b0c62/PyMySQL-1.1.0-py3-none-any.whl.metadata
  Downloading PyMySQL-1.1.0-py3-none-any.whl.metadata (4.4 kB)
Downloading PyMySQL-1.1.0-py3-none-any.whl (44 kB)
----- 0.0/44.8 kB ? eta -:-:-
----- 20.5/44.8 kB 640.0 kB/s eta 0:00:01
----- 44.8/44.8 kB 734.6 kB/s eta 0:00:00
Installing collected packages: pymysql
Successfully installed pymysql-1.1.0
```

Create database in MYSQL if we have not already, which we will connect with python



The screenshot shows the MySQL Workbench interface. The 'Query 1' window contains the following SQL commands:

```
1 • create database breast_cancer_db
2
3 ✖ use breast_cancer_db
4
5
```

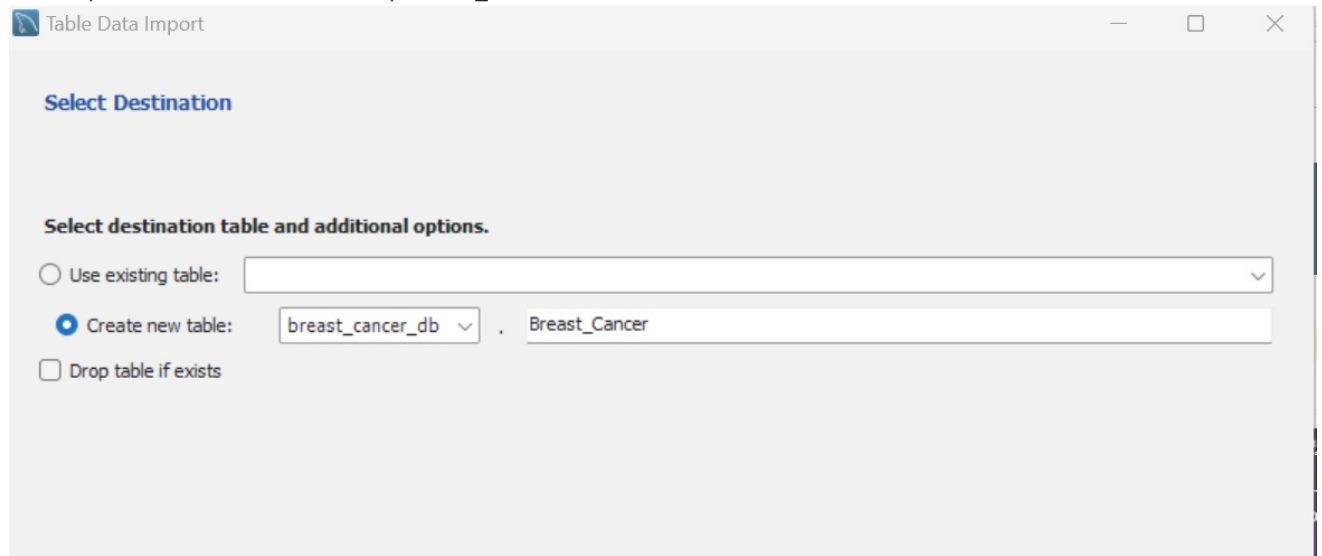
The 'Navigator' pane on the left shows the 'MANAGEMENT' section with options like 'Server Status', 'Client Connections', 'Users and Privileges', 'Status and System Variables', and 'Data Export'.

Below the query window, a log of executed queries is visible:

Query ID	Time	Query	Result
3	09:00:20	SELECT * FROM world LIMIT 0, 1000	Error Code: 1046.
4	09:04:43	create database breast_cancer_db	1 row(s) affected
5	09:05:06	use breast_cancer_db	0 row(s) affected

Importing csv file in mysql from local using Table data import wizard

We can put table name on our own, I kept Breast_Cancer



The screenshot shows the 'Table Data Import' dialog box. The 'Select Destination' section is active, showing the following options:

- ☐ Use existing table: [dropdown menu]
- ☒ Create new table: breast_cancer_db . Breast_Cancer
- ☐ Drop table if exists

Click next with default until data does not get imported

Import Results

File C:\Users\USER\Downloads\Breast_cancer_data.csv was imported in 2.408 s

Table breast_cancer_db.breast_cancer was created

569 records imported

lets check:

```
5 select * from Breast_Cancer;
```

	mean_radius	mean_texture	mean_perimeter	mean_area	mean_smoothness	diagnosis
▶	17.99	10.38	122.8	1001	0.1184	0
	20.57	17.77	132.9	1326	0.08474	0
	19.69	21.25	130	1203	0.1096	0
	11.42	20.38	77.58	386.1	0.1425	0
	20.29	14.34	135.1	1297	0.1003	0

NOW lets look, how we can connect it with Python for future EDA analysis

```
In [2]: import pymysql
```

```
In [3]: #Here connecting jupyter notebook/python with mysql database
db = pymysql.connect(host="localhost",user= "root",password= "",database= "breast_cancer_db")
```

```
In [4]: db
```

```
Out[4]: <pymysql.connections.Connection at 0x1caf23def50>
```

Read SQL query in pandas dataframe

```
In [6]: import pandas as pd
```

```
In [7]: df=pd.read_sql_query("""select * from Breast_Cancer""", db, parse_dates= True)
```

C:\Users\USER\AppData\Local\Temp\ipykernel_15348\189591120.py:1: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.

```
df=pd.read_sql_query("""select * from Breast_Cancer""", db, parse_dates= True)
```

```
In [8]: df.head()
```

```
Out[8]:
```

	mean_radius	mean_texture	mean_perimeter	mean_area	mean_smoothness	diagnosis
0	17.99	10.38	122.80	1001.0	0.11840	0
1	20.57	17.77	132.90	1326.0	0.08474	0
2	19.69	21.25	130.00	1203.0	0.10960	0
3	11.42	20.38	77.58	386.1	0.14250	0
4	20.29	14.34	135.10	1297.0	0.10030	0

```
In [9]: #Now in this way we can work with SQL and Pandas
```

Working with JSON

Importing pandas library and reading json file, which is downloaded in download folder

```
In [10]: df=pd.read_json(r"C:\Users\USER\Downloads\train.json")
```

```
In [11]: df.head(6)
```

Out[11]:

	id	cuisine	ingredients
0	10259	greek	[romaine lettuce, black olives, grape tomatoes...
1	25693	southern_us	[plain flour, ground pepper, salt, tomatoes, g...
2	20130	filipino	[eggs, pepper, salt, mayonaise, cooking oil, g...
3	22213	indian	[water, vegetable oil, wheat, salt]
4	13162	indian	[black pepper, shallots, cornflour, cayenne pe...
5	6602	jamaican	[plain flour, sugar, butter, eggs, fresh ginge...

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js