**use pandas with SQL**

Can I use pandas with SQL?

Pandas isn't good at handling big data, and **its features can all be done with SQL**. However, Pandas' value comes from its integration with other plotting libraries, machine learning libraries, and the Python language.

Running SQL queries to read data from databases in Python

In addition to what we have already seen so far, there is also an option to write simple SQL queries and execute those against a table in the SQL database. This can be done with the help of the “*read\_sql()*” method that is available in the Pandas dataframe object. This method is useful when you want to have more control over the data that you want to bring into the Python environment. You can also join multiple tables in the SQL statement and filter the query buy adding where clause on it. The final result will thus be executed on the table and data will be fetched into the Python environment.

Let us now see this in action.

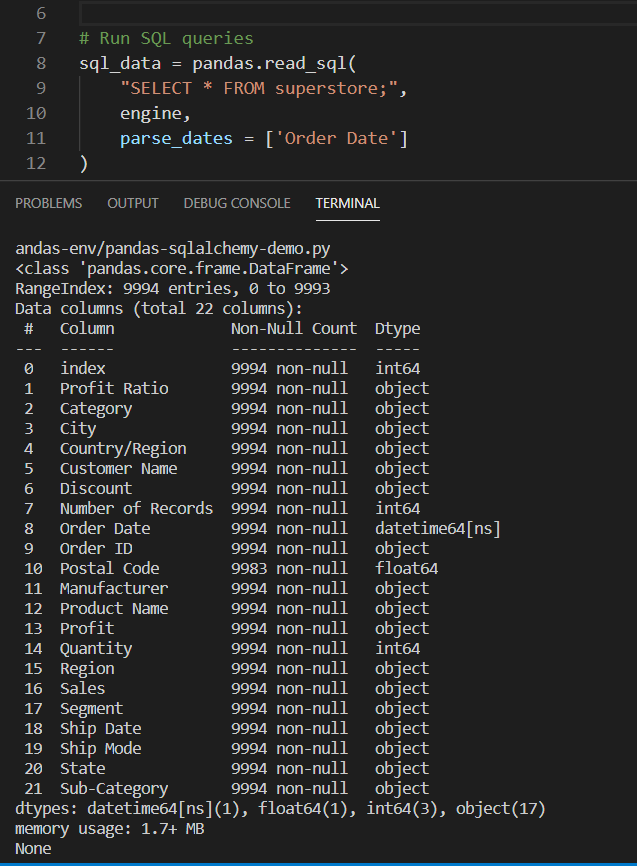
*The CData Python Connector for MySQL enables you use pandas and other modules to analyze and visualize live MySQL data in Python.*

The rich ecosystem of Python modules lets you get to work quickly and integrate your systems more effectively. With the CData Python Connector for MySQL, the pandas & Matplotlib modules, and the SQLAlchemy toolkit, you can build MySQL-connected Python applications and scripts for visualizing MySQL data. This article shows how to use the pandas, SQLAlchemy, and Matplotlib built-in functions to connect to MySQL data, execute queries, and visualize the results.

With built-in optimized data processing, the CData Python Connector offers unmatched performance for interacting with live MySQL data in Python. When you issue complex SQL queries from MySQL, the driver pushes supported SQL operations, like filters and aggregations, directly to MySQL and utilizes the embedded SQL engine to process unsupported operations client-side (often SQL functions and JOIN operations).

Now, connect the sqlite to the database file. And read the SQL query to read the table.

# Read in SQLite databases  
con = sqlite3.connect("database.sqlite")  
#Read the Table  
player = pd.read\_sql\_query("SELECT \* from Player\_Attributes", con)  
#close the connection  
con.close()



*Figure 4 – Running queries to read data from SQL table*

