How to Run and Analyze SQL Queries with Pandas in Jupyter

Steps:

1. Installation and Setup
2. Connecting to a Database
3. Executing SQL Queries
4. Constructing Pandas Dataframes

Installation & Setup

Step 1:
First, install ipython-sql to get the %sql or %%sql magic command.

pip install ipython-sql

Step 2:

pip install SQLAlchemy

Step 3:

Third, install a DBAPI (Python Database API Specification) driver for whichever dialect you wish to use. SQLAlchemy supports many dialects and drivers;

Here are a few popular ones to get you started:

Dialect	Driver	Conda Install Driver
MySQL	PyMySQL	conda install -c anaconda pymysql
PostgreSQL	Pyscopg2	conda install -c conda-forge psycopg2
SQLite	Pysqlite	conda install -c prometeia pysqlite
Oracle	Cx Oracle	conda install -c anaconda cx_oracle
Microsoft SQL Server	Pyodbc	conda install -c conda-forge pyodbc

Full list of dialects can be seen in
https://docs.sqlalchemy.org/en/13/dialects/index.html

Connecting to a Database

The **Engine** is the starting point for any SQLAlchemy application. The typical form of a database URL using SQLAlchemy is:

dialect+driver://username:password@host:port/database

For secure access, you can dynamically access your credentials to avoid storing your password in the notebook itself.

```
# load the ipython-sql extension
%load_ext sql

# Option1: Show credentials
%sql mysql+pymysql://user:pass@localhost:3306/employees

# Option2: Hide password using getpass
import getpass
user = 'miguel'
password = getpass.getpass()
connection_string =
f'mysql+pymysql://{user}:{password}@localhost:3306/employees'
%sql $connection_string
```

Executing SQL Queries

```
# single-line query
%sql SELECT salary FROM salaries ORDER BY salary DESC LIMIT 10

# multi-line query (use double %%)
%%sql
SELECT salary
FROM salaries
ORDER BY salary DESC
LIMIT 10
```

Constructing Pandas DataFrames

```
# assign to variable (single-line)
top_10_salaries = %sql SELECT salary FROM salaries ORDER BY
salary DESC LIMIT 10

# assign to variable (multi-line)
%%sql top_10_salaries <<
SELECT salary
FROM salaries
ORDER BY salary DESC
LIMIT 10

# construct Pandas DataFrame
df = top_10_salaries.DataFrame()
```

Tips

- Executing SQL queries via iPython is *not* as fast as using the command line or a SQL editor, but you can limit the query results (recommended)
- You can automatically return Pandas DataFrames

```
# limits displayed results; entire set is still pulled into memory
%config SqlMagic.displaylimit=<int>

# limits the result set (good for large sets that can slow down your
browser)
%config SqlMagic.autolimit=<int>

# return Pandas DataFrames instead of regular result sets
%config SqlMagic.autopandas=True
```

for more details

https://nbviewer.jupyter.org/github/corralm/medium-posts/blob/master/SQL_Jupyter_Pandas.ipynb