

Week 11 Intro to SQLAlchemy: Day 1

What you will learn

- Connect to a SQL database using SQLAlchemy
- Perform basic SQL queries using engine.execute()
- Create Python classes and objects
- Create, read, update, and delete data from a SQL database using SQLAlchemy's ORM

SQLAlchemy

1. What is object-relational mapper (ORM)¹
2. Benefits to using an ORM²
3. Disadvantages to using an ORM³

PREREQ'S

conda install pymysql conda install sqlite

SQLAlchemy Connection

Connection String: <Dialect>://<Username>:<Password>@<Host Address>:<Port>/<Database>

Example: mysql://k5xunpkmojyzse51:ifagg1gp7e2xyapi@ffn96u87j5ogvehy.cbetxkdyhwsb.us-east-1.rds.amazonaws.com:3306/tq6h098h0ym00zp6

Students Activity: Ice Cream Connection

Instructions:

- Connect to your localhost database in MySQL Workbench.
- Create a table of 5 or more ice cream flavors, each of which should have a rating out of 100, and price (make sure that a few cost \$1.25).

¹<http://www.sqlalchemy.org/>

²<https://blogs.msdn.microsoft.com/gblock/2006/10/26/ten-advantages-of-an-orm-object-relational-mapper/>

³<https://stackoverflow.com/questions/7334287/disadvantages-of-object-relational-mapping/7334519>

- Query your table in MySQL Workbench for every flavor of ice cream.
- Query your table in MySQL for all flavors that cost 1.25 or more.
- Create a Python script that uses SQLAlchemy to accomplish all of the above queries.

Instructor Activity: SQLAlchemy and Pandas

```
# Pandas
import pandas as pd
# SQL Alchemy
from sqlalchemy import create_engine
# PyMySQL
import pymysql
engine.connect()
```

Students Activity: Read All the SQL

Instructions:

- Using the following connection string, complete the following tasks within a Jupyter Notebook.
 - `mysql://k5xunpkmojyzse51:ifagg1gp7e2xyapi@ffn96u87j5ogvehy.cbetxkdyhwsb.us-east-1.rds.amazonaws.com:3306/tq6h098h0ym00zp6`
- Query all the data from the `Census_Data` table and load into pandas.
- Query all the data from the `Zip_Census` table and load into pandas.
- Show the `.head()` of your newly imported dataframes and then come up with the best way in which to merge the data together.

BREAK

Instructor Activity: A Schooling on Classes

Object oriented programming (OOP) is a style of coding based around the concept of "objects" Python is a class-based programming language:

```
◦ def __init__(self):
```

Students Activity: Surfer Class

Instructions:

- Create a class `Surfer` that takes in a name, hometown, and rank.
- Create an instance of a surfer and then print the following from your surfer object: name, hometown and rank.

Bonus:

- Create a while loop that will allow you keep taking input that allows you to create new instances of surfers.
- Keep the loop going until the user specifies otherwise.

Instructor Activity: A Method to The Classes

Calling Methods in Class

- `ClassName.MethodName()`

Example

- `dog.bark()`

Students Activity: Surfer Class Extended

Instructions:

- Create a surfer class that has name, hometown, rank, wipeouts, and surfer count.
- Create a method called speak that prints "Hangs loose, bruh!"
- Create a method called biography that prints the surfer's name and hometown.
- Create a method called cheer that will print "I totally rock man, no wipeouts!" if the surfer has no wipeouts. Otherwise, it prints 'Bummer bruh, keep on keeping on!'.
- Create two surfers that print out all their info and run all the methods.

Bonus:

- Add a method to your class that prints out how many surfers are currently shredding.

Hint:

- When dealing with the bonus, make sure to set a variable to keep track of surfers before you initialize but after you declare the class.

Everyone: Back to the SQL

From sqlalchemy.ext.declarative import declarative_base⁴

- declarative base allows SQLAlchemy to convert the classes created in Python to SQL tables

From sqlalchemy.orm import Session⁵

⁴<http://docs.sqlalchemy.org/en/latest/orm/extensions/declarative/api.html>

⁵http://docs.sqlalchemy.org/en/latest/orm/session_basics.html

- *A temporary binding to our DB*
- Session(bind=engine)

Students Activity: Surfing SQL

Instructions:

- Modify the Surfer class created during the previous activity so that it will function with SQLAlchemy.
- __tablename__ should be "surfers"
- Surfer_id should be an integer and the primary keyname should be a string capable of holding 255 characters
- Hometown should be a string capable of holding 255 characters
- Rank should be an integer

Create a new class called Board which will function with SQLAlchemy and has the following parameters...

- __tablename__ should be "surfboards"
- id should be an integer and the primary key
- surfer_id should be an integer that references a surfer_id in the "surfers" column
- board_name should be a string capable of holding 255 characters
- color should be a string capable of holding 255 characters
- length should be an integer
- Pull a list of all of the surfers and surfboards already inside the database
- Push a new surfer and surfboard to the tables on the database

Homework Cheat Sheet

Review these libraries import datetime import numpy import pandas import sqlalchemy from sqlalchemy.ext.automap import automap_base from sqlalchemy.orm import Session from sqlalchemy import create_engine, func from flask import Flask, jsonify Base = automap_base() # Save references to each table using Base Python Flask⁶ **Day 3 (From StudentGuide.md import Flask Mega-Tutorial Video⁷)** app = Flask(__name__)

⁶<http://flask.pocoo.org/>

⁷<https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world>