**# SQLAlchemy Homework - Surfs Up!**

**### Before You Begin**

1. Create a new repository for this project called `sqlalchemy-challenge`. \*\*Do not add this homework to an existing repository\*\*.

2. Clone the new repository to your computer.

3. Add your Jupyter notebook and `app.py` to this folder. These will be the main scripts to run for analysis.

4. Push the above changes to GitHub or GitLab.

Congratulations! You've decided to treat yourself to a long holiday vacation in Honolulu, Hawaii! To help with your trip planning, you need to do some climate analysis on the area. The following outlines what you need to do.

**## Step 1 - Climate Analysis and Exploration**

***To begin, use Python and SQLAlchemy*** ***to do basic climate analysis and data exploration of your climate database. All of the following analysis should be completed using SQLAlchemy ORM queries, Pandas, and Matplotlib.***

1. Use the provided [starter notebook](climate\_starter.ipynb) and [hawaii.sqlite](Resources/hawaii.sqlite) files to complete your climate analysis and data exploration.
2. Choose a start date and end date for your trip. Make sure that your vacation range is approximately 3-15 days total.
3. Use SQLAlchemy `create\_engine` to connect to your sqlite database.
4. Use SQLAlchemy `automap\_base()` to reflect your tables into classes and save a reference to those classes called `Station` and `Measurement`.

**### Precipitation Analysis**

1. Design a query to retrieve the last 12 months of precipitation data.
2. Select only the `date` and `prcp` values.
3. Load the query results into a Pandas DataFrame and set the index to the date column.
4. Sort the DataFrame values by `date`.
5. Plot the results using the DataFrame `plot` method.

A screenshot of a cell phone

Description automatically generated

1. Use Pandas to print the summary statistics for the precipitation data.

**### Station Analysis**

1. Design a query to calculate the total number of stations.
2. Design a query to find the most active stations.
3. List the stations and observation counts in descending order.
4. Which station has the highest number of observations?

***Hint: You may need to use functions such as `func.min`, `func.max`, `func.avg`, and `func.count` in your queries.***

1. Design a query to retrieve the last 12 months of temperature observation data (TOBS).
2. Filter by the station with the highest number of observations.
3. Plot the results as a histogram with `bins=12`.

**A screenshot of a cell phone

Description automatically generated**

**## Step 2 - Climate App**

Now that you have completed your initial analysis, design a Flask API based on the queries that you have just developed.

1. Use Flask to create your routes.

**### Routes**

1. `/`
2. Home page.
3. List all routes that are available.
4. `/api/v1.0/precipitation`
5. Convert the query results to a dictionary using `date` as the key and `prcp` as the value.
6. Return the JSON representation of your dictionary.
7. `/api/v1.0/stations`
8. Return a JSON list of stations from the dataset.
9. `/api/v1.0/tobs`
10. Query the dates and temperature observations of the most active station for the last year of data.
11. Return a JSON list of temperature observations (TOBS) for the previous year.
12. `/api/v1.0/<start>` and `/api/v1.0/<start>/<end>`
13. Return a JSON list of the minimum temperature, the average temperature, and the max temperature for a given start or start-end range.
14. When given the start only, calculate `TMIN`, `TAVG`, and `TMAX` for all dates greater than and equal to the start date.
15. When given the start and the end date, calculate the `TMIN`, `TAVG`, and `TMAX` for dates between the start and end date inclusive.

**## Hints**

* You will need to join the station and measurement tables for some of the queries.
* Use Flask `jsonify` to convert your API data into a valid JSON response object.

**## Bonus: Other Recommended Analyses**

The following are optional challenge queries. These are highly recommended to attempt, but not required for the homework.

**### Temperature Analysis I**

* Hawaii is reputed to enjoy mild weather all year. Is there a meaningful difference between the temperature in, for example, June and December?
* You may either use SQLAlchemy or pandas's `read\_csv()` to perform this portion.
* Identify the average temperature in June at all stations across all available years in the dataset. Do the same for December temperature.
* Use the t-test to determine whether the difference in the means, if any, is statistically significant. Will you use a paired t-test, or an unpaired t-test? Why?

**### Temperature Analysis II**

* The starter notebook contains a function called `calc\_temps` that will accept a start date and end date in the format `%Y-%m-%d`. The function will return the minimum, average, and maximum temperatures for that range of dates.
* Use the `calc\_temps` function to calculate the min, avg, and max temperatures for your trip using the matching dates from the previous year (i.e., use "2017-01-01" if your trip start date was "2018-01-01").
* Plot the min, avg, and max temperature from your previous query as a bar chart.
* Use the average temperature as the bar height.
* Use the peak-to-peak (TMAX-TMIN) value as the y error bar (YERR).

**A close up of a clock

Description automatically generated**

**### Daily Rainfall Average**

* Calculate the rainfall per weather station using the previous year's matching dates.
* Calculate the daily normals. Normals are the averages for the min, avg, and max temperatures.
* You are provided with a function called `daily\_normals` that will calculate the daily normals for a specific date. This date string will be in the format `%m-%d`. Be sure to use all historic TOBS that match that date string.
* Create a list of dates for your trip in the format `%m-%d`. Use the `daily\_normals` function to calculate the normals for each date string and append the results to a list.
* Load the list of daily normals into a Pandas DataFrame and set the index equal to the date.
* Use Pandas to plot an area plot (`stacked=False`) for the daily normals.

A screenshot of a cell phone

Description automatically generated

A`### Copyright

Trilogy Education Services © 2019. All Rights Reserved.