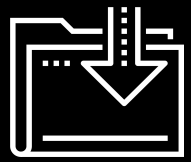
Advanced Use of the SQLAlchemy ORM

Data Boot Camp

Lesson 10.2



Today's Goals

By the end of this class, you will be able to:

- Use SQLAlchemy ORM to model tables.
- Perform CRUD with SQLAlchemy.
- Reflect existing databases with SQLAlchemy.
- Plot query results from SQLAlchemy ORM.
- Run a t-test to validate differences in means.

SQLAlchemy Queries

How can you query a database using SQLAlchemy?

SQLAlchemy Queries

There are two basic ways to query a database in SQLAlchemy:

01 SQL statements

```
data = engine.execute("SELECT * FROM BaseballPlayer")
```

02 Python objects

```
players = session.query(BaseballPlayer)
for player in players:
    print(player.name_given)
```

SQLAlchemy Queries

it is preferred to use Python objects for interacting with a database in SQLAlchemy.

To query a database for all of the records in a specific table, use `session.query()` and pass the SQLAlchemy class that is associated with the table through as a parameter.

```
# Print all of the player names in the database
players = session.query(BaseballPlayer)
for player in players:
  print(player.name_given)
```

What is a t-test, and what is it used for?

T-test

A t-test is used to test the difference between means!

One-sample t-test

One group is compared against a standard value.

Example: Comparing gasoline octane level to an octane level

Independent t-test

Groups come from two distinct populations.

Example: Different countries, different species

Instructor Demonstration

SQLAlchemy Queries in Action

Questions?

Shark Search

Activity: Shark Search

In this activity, you will create a Python script that can search through the SQL file of shark attacks provided.

Suggested Time:

20 minutes

Activity: Shark Search

Instructions

Within a Python script, create a Sharks class that will be able to read in all of the columns from the table you created.

Using SQLAlchemy, perform the following queries:

- Print all locations of shark attacks.
- Find the number of provoked attacks.
- Find the number of attacks in the USA.
- Find the number of attacks in 2017.
- Find the number of attacks while surfing.
- Find the number of fatal attacks.
- Find the number of fatal attacks while surfing.
- Find the number of fatal attacks in Mozambique while spearfishing

Time's Up! Let's Review.

What a Cruddy Database

Using CRUD

We have only looked at one-half of CRUD:

Create	Create data in a table with the `INSERT` statement.
Read	Read data with `SELECT`.
Update	Updated a table's data with `UPDATE`.
Delete	Delete data with `DELETE`.

Instructor Demonstration

Updating and Deleting Rows

Questions?

Partner Activity: What a Cruddy Database

In this activity, you and a partner will create a new SQLite database for a garbage collection company.

Suggested Time:

20 minutes

Partner Activity: What a Cruddy Database

Instructions	
	Within the unsolved Python file, create a new SQLAlchemy class called Garbage that holds the values outlined in the Readme.md
	Create a connection and session before adding a few items into the SQLite database.
	Update the values within at least two of the rows added to the table.
	Delete the row with the lowest weight.
	Print out all of the data within the database.

Time’s Up! Let’s Review.

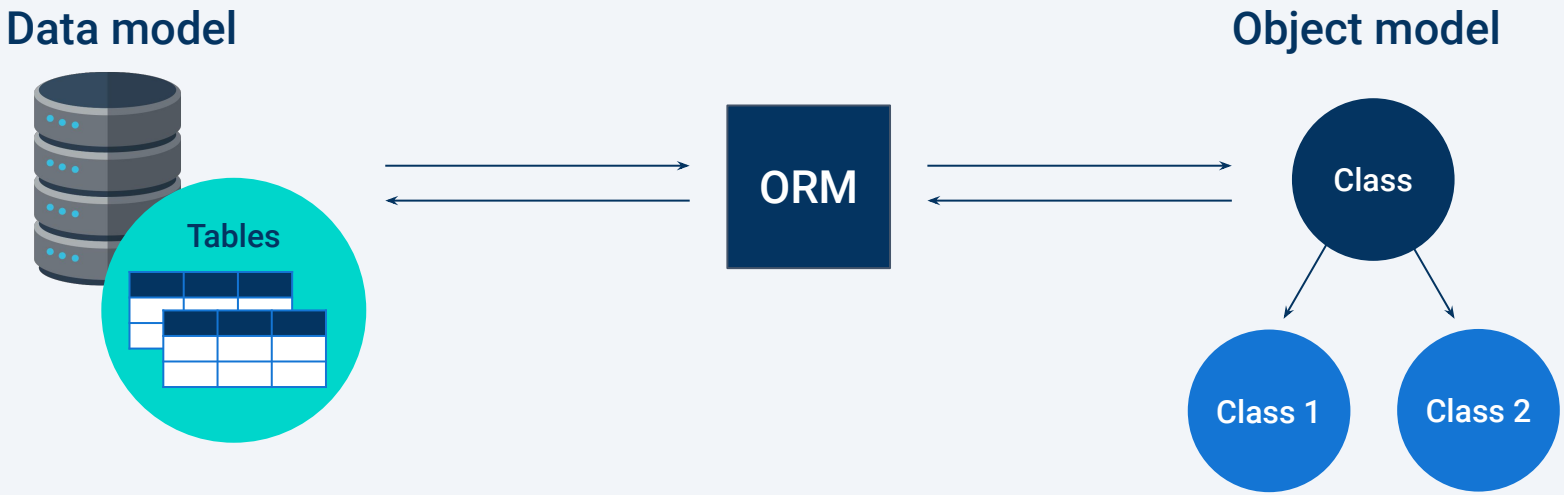

Break

Reflecting on SQL

How can we analyze databases that already exist?

Reflecting on SQL

SQLAlchemy provides tools to create ORM classes for an existing database!



Instructor Demonstration

Reflection

Activity: Reflecting on SQL

In this activity, you will practice your ability to reflect existing databases using SQLAlchemy and a SQLite table with demographic data.

Suggested Time:

15 minutes

Activity: Reflecting on SQL

Instructions	
	Create engine using the demographics.sqlite database file.
	Declare a Base using `automap_base()`, and use this new Base class to reflect the database's tables.
	Assign the demographics table/class to a variable called Demographics.
	Create a session, and use this session to query the Demographics table and display the first five locations.

Time’s Up! Let’s Review.

Salary Exploration

Instructor Demonstration

SQLAlchemy Exploration

Activity: Salary Exploration

In this activity, you will create an inspector and search through a SQLite database of San Francisco salaries.

Suggested Time:

15 minutes

Activity: Salary Exploration

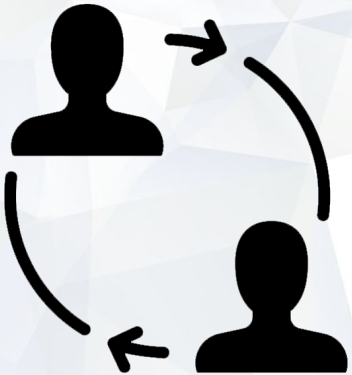
Instructions

Using the SQLite file, use an inspector to collect the following information:

- The names of all of the tables in the database
- The column names and data types for the Salaries table

Time’s Up! Let’s Review.

Emoji Plotting



Group Activity: Emoji Plotting

In this activity, you will will join forces to create a plot based on the data from a SQLite database.

Suggested Time:

15 minutes

Group Activity: Emoji Plotting

Instructions	
	Use the inspector to explore the database and print out the table names stored within it.
	Using the inspector, print out the column names and types for each of the tables contained within the SQLite file.
	Reflect the database into a SQLAlchemy class and start a session that can be used to query the database.
	Using Matplotlib, create a horizontal bar chart and plot the emoji score in descending order. Use `emoji_char` as the y-axis labels and plot only the top 10 emojis ranked by score
	Create the same kind of chart using Pandas to plot the data instead of Matplotlib.

Time’s Up! Let’s Review.