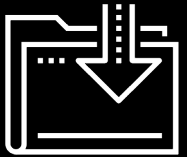




Advanced Usage of the SQLAlchemy ORM

Data Boot Camp
Lesson 10.2



Class Objectives

By the end of today's class, you will be able to:



Use the SQLAlchemy ORM to create classes that model tables.



Perform database CRUD operations by using the SQLAlchemy ORM.



Reflect existing databases.



Use the SQLAlchemy Inspector to view table names in the database.



Plot query results from the ORM.



Run a t-test to validate differences in means.

The background is a dark charcoal gray with a series of parallel diagonal lines running from the top-left to the bottom-right. Overlaid on this are several teal-colored geometric shapes: a large central triangle pointing right, a smaller triangle to its left, and a square to its right. Scattered around these shapes are various white line-art symbols, including a plus sign, a minus sign, a circle with a dot, a circle with a horizontal line, a circle with a vertical line, a circle with a diagonal line, a circle with a cross, a circle with a dot, a circle with a horizontal line, a circle with a vertical line, a circle with a diagonal line, a circle with a cross, a circle with a dot, a circle with a horizontal line, a circle with a vertical line, a circle with a diagonal line, and a circle with a cross.

WELCOME



Queries



Crafting SQLAlchemy queries is much easier than we might first expect.

Welcome & SQLAlchemy Queries

To prove this point, we will work with more realistic datasets today. Our first database contains over 1,000 rows of data that can be searched through.

BaseballPlayers								
name_first	name_last	name_given	weight	height	bats	throws	debut	final_game
David	Aardsma	David Allan	220	75	R	R	2004-04-06	2015-08-23
Hank	Aaron	Henry Louis	180	72	R	R	1954-04-13	1976-10-03
Tommie	Aaron	Tommie Lee	190	75	R	R	1962-04-10	1971-09-26
Don	Aase	Donald William	190	75	R	R	1977-07-26	1990-10-03
Andy	Abad	Fausto Andres	184	73	L	L	2001-09-10	2006-04-13
Fernando	Abad	Fernando Antonio	220	73	L	L	2010-07-28	2015-10-03
John	Abadie	John W.	192	72	R	R	1875-04-26	1875-06-10
Ed	Abbatichio	Edward James	170	70	R	R	1897-09-04	1910-09-15
Bert	Abbey	Bert Wood	175	71	R	R	1892-06-14	1896-09-23
Charlie	Abbey	Charles S.	169	68	L	L	1893-08-16	1897-08-19

**We will demonstrate how SQLAlchemy
can be used in conjunction with SciPy
to perform analysis on a dataset.**

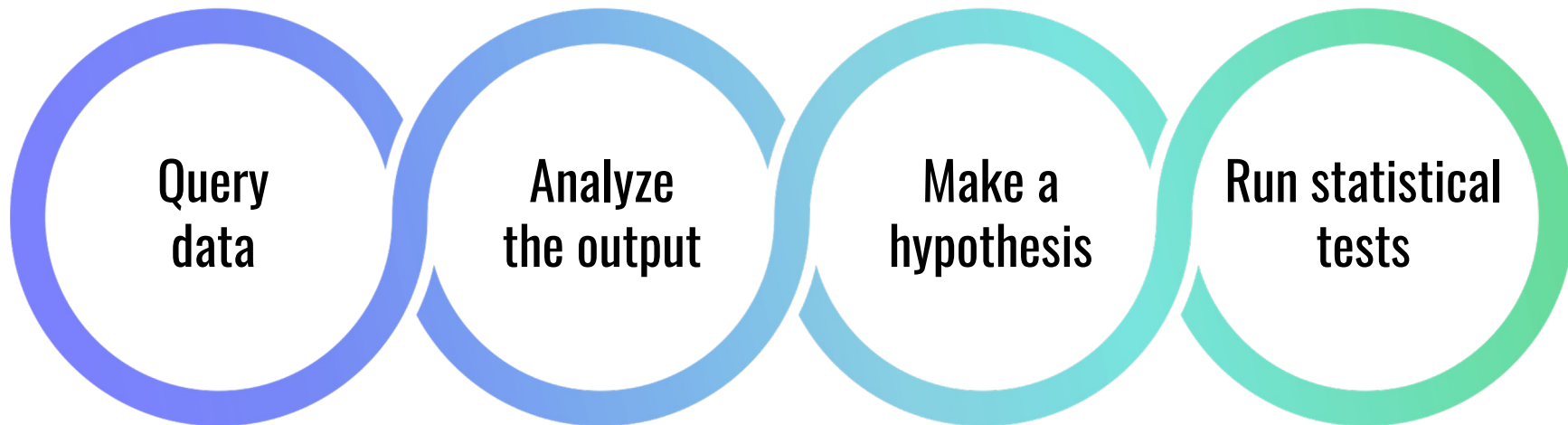
SQLAlchemy

+



SciPy

SQLAlchemy Queries in Action



For students who **are** pursuing
a career in data science:

It's important to practice this workflow on your own.

For students who **are not** pursuing
a career in data science:

*This workflow demonstrates how SQLAlchemy can interface
with another common Python library using queried objects.*



Do you recall how to query a
database using SQLAlchemy?

SQLAlchemy Queries in Action

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

01

with SQL
statements

02

with Python
objects



**In the previous class,
we discussed that Python
objects are preferred
for interacting with a
database in SQLAlchemy.**

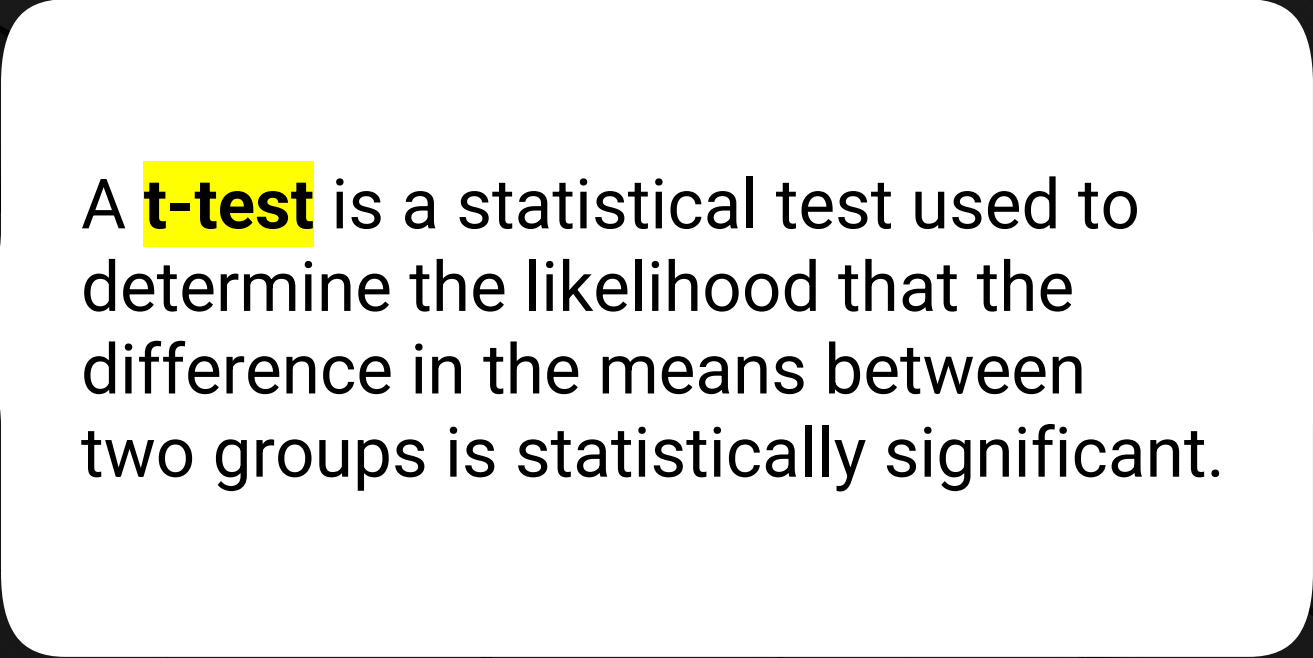
SQLAlchemy Queries in Action

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

```
# 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?



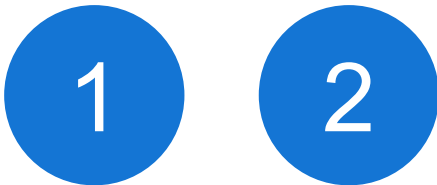
A **t-test** is a statistical test used to determine the likelihood that the difference in the means between two groups is statistically significant.

t-test

There are two types of (two-sample) t-tests:

Paired t-test

Compares the means of the **same group**.



Example:

Mean blood pressure in patients before and after medication.

Unpaired t-test

Compares the means of **two different groups**.



Example:

The mean annual spending on dining out among Minnesotans versus Texans.



Instructor Demonstration

SQLAlchemy Queries in Action

Questions?





Sunny Hours Search



Activity: Sunny Hours Search

In this activity, you will create a Python script that can search through the provided SQL file of data on hours of sunshine in cities throughout the world.

Suggested Time:

20 Minutes



Time's Up! Let's Review.

Questions?

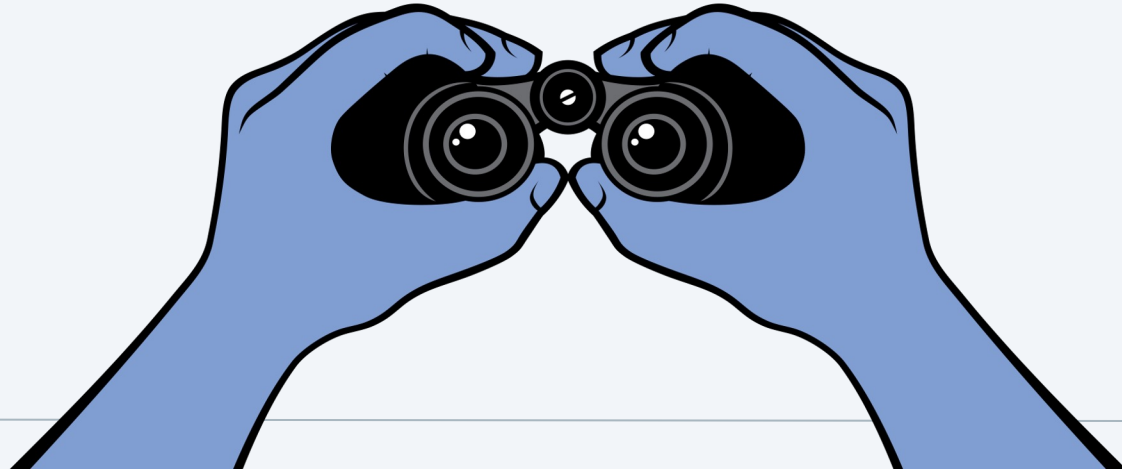




CRUD Database

We have only looked at one-half of CRUD!

C R U D
create read update delete



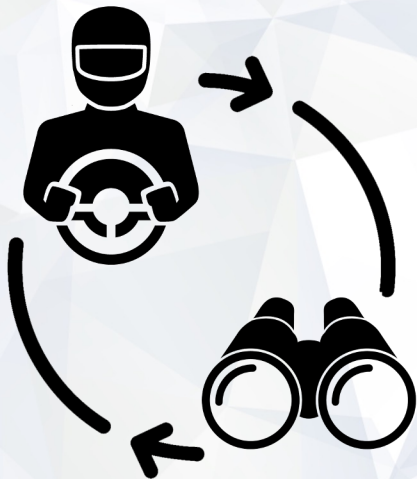


Instructor Demonstration

Updating and Deleting Rows

Questions?





Pair Programming Activity:

CRUD Database

In this activity, you will be tasked with creating a new SQLite database for travel destinations.

You will need to create a table, add rows into the table, update values in rows, and, finally, delete a row from the database.

Suggested Time:

20 Minutes



Time's Up! Let's Review.

A close-up photograph of a computer keyboard. The central focus is a large, white, rectangular key with rounded corners. On this key, there is a dark blue icon of a coffee cup with three wavy lines above it representing steam. Below the icon, the word "Break" is printed in a dark blue, serif font. The key is set against a background of other keyboard keys, which are slightly out of focus. The lighting is soft and even, highlighting the texture of the keys.

Break



Reflecting on SQL

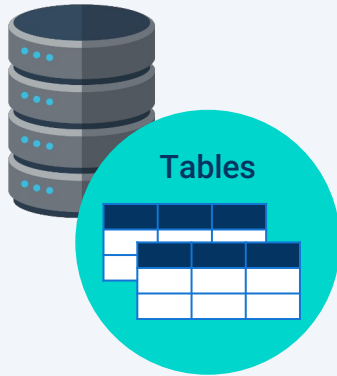


**How can we analyze databases
that already exist?**

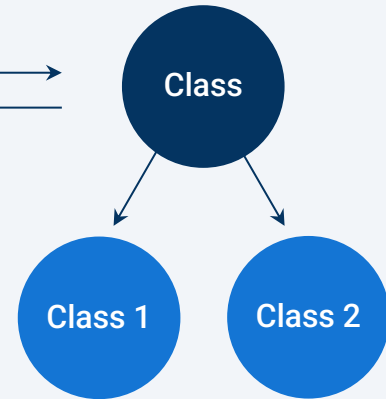
Reflecting on SQL

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

Data model



Object model





Instructor Demonstration

Reflections

Reflections

Reflecting an existing database is a simple four-step process:

01

Import `automap_base` from the SQLAlchemy library.

02

Create an `engine` against the existing database that should be reflected.

03

Create a `Base` by calling `Base = automap_base()`.

04

Call `Base.prepare` with the `engine` from Step 2, and `reflect=True` as its parameters.

Questions?





Activity: Reflecting on SQL

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

Suggested Time:

20 Minutes



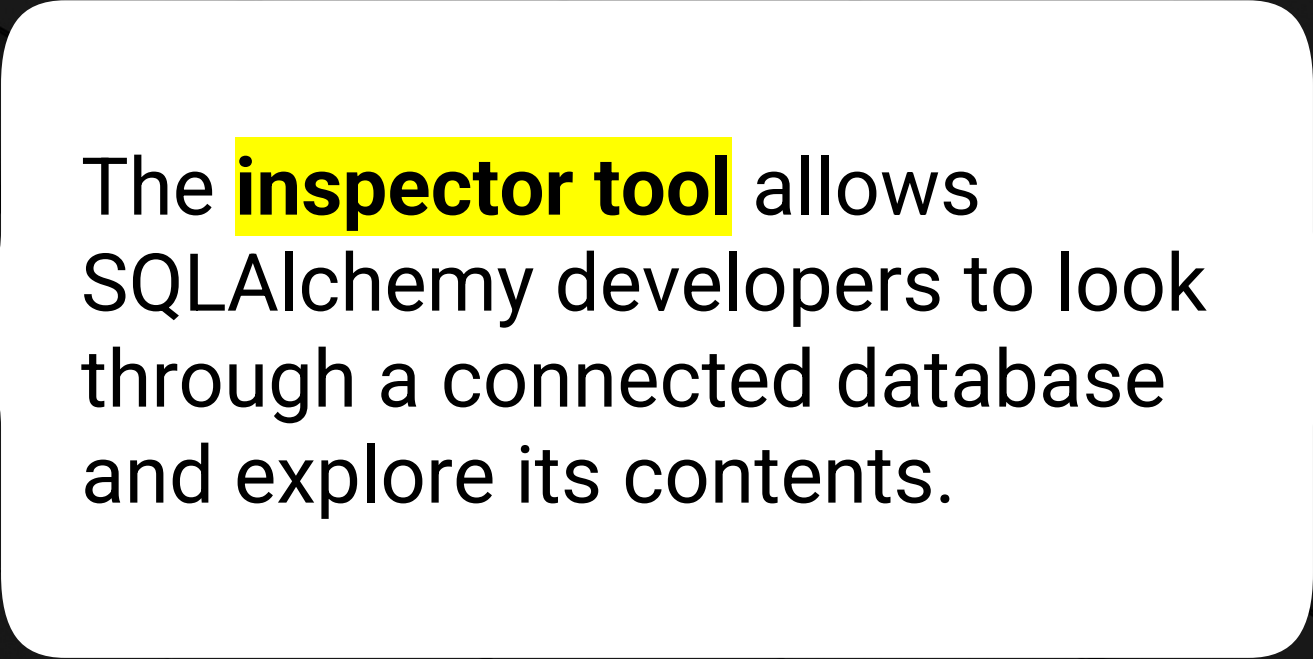
Time's Up! Let's Review.

Questions?





Salary Exploration



The **inspector tool** allows
SQLAlchemy developers to look
through a connected database
and explore its contents.

SQLAlchemy Exploration

The inspector is primarily used to look up:



Tables



Columns



Data types



To look up specific values stored in a table, queries should be used.

Questions?





Instructor Demonstration

SQLAlchemy Inspection



Activity: Salary Exploration

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

Suggested Time:

20 Minutes



Time's Up! Let's Review.

Questions?

