Introduction to SQLAlchemy

Data Boot Camp

Lesson 10.1

Class Objectives

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

- Connect to a SQL database using SQLAlchemy
- Learn to 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

Activity: Looking into SQLAlchemy

In this activity, you will break into groups of two or three to research a few questions:

1. What is an ORM?
2. What are the benefits of using an ORM?
3. What are some of the disadvantages of using an ORM?

Suggested Time:

3 minutes

Questions?

Introduction to SQLAlchemy

SQLAlchemy is a Python library designed to work with SQL databases.

Introduction to SQLAlchemy

SQLAlchemy bridges the gaps between the various dialects of SQL.
With SQLAlchemy, a single Python script can perform the same query across the different SQL dialects, including:

- PostgreSQL
- SQLite
- MySQL

SQLAlchemy ORM Is Flexible

It's possible to query a database using more SQL:

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

Or more Python:

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

ORM can also improve security against malicious queries such as SQL injections.

Instructor Demonstration

Building a SQLAlchemy Connection

Ice Cream Connection

Activity: Ice Cream Connection

In this activity, you will create, connect, and insert data into a new database using SQLAlchemy.

Suggested Time:

15 minutes

Activity: Ice Cream Connection

Instructions	Use the database path to create a SQLite engine.
	Use the engine to select all of the rows and columns from the table `icecreamstore.csv`.
	Create a new query that finds the ice cream flavors that cost $1.25 or more.

Activity: Read All the SQL

Instructions	
	Create an engine to connect to the census database.
	Query all the data from the `Census_Data` table, and load into Pandas.
	Create an engine to connect to the zip database.
	Query all the data from the `Zip_Census` table, and load into Pandas.
	Show the `.head()` of your newly imported data.

Time's Up! Let's Review.

Read All the SQL

Break

One of the most impressive aspects of SQLAlchemy is how it integrates with Pandas.

Pandas Integrates with SQLAlchemy

Once we connect to our SQL database using SQLAlchemy ...

```
# Create Engine
engine = create_engine(f"sqlite:///{database_path}")
conn = engine.connect()
```

... we can query directly using Pandas:

```
# Query All Records in the Database
data = pd.read_sql("SELECT * FROM Census_Data", conn)
```

Instructor Demonstration

SQLAlchemy and Pandas

Questions?

Activity: Read All the SQL

In this activity, you will query an external server using Pandas and SQLAlchemy to create new DataFrames based on U.S. census data.

Suggested Time:

10 minutes

Time’s Up! Let’s Review.

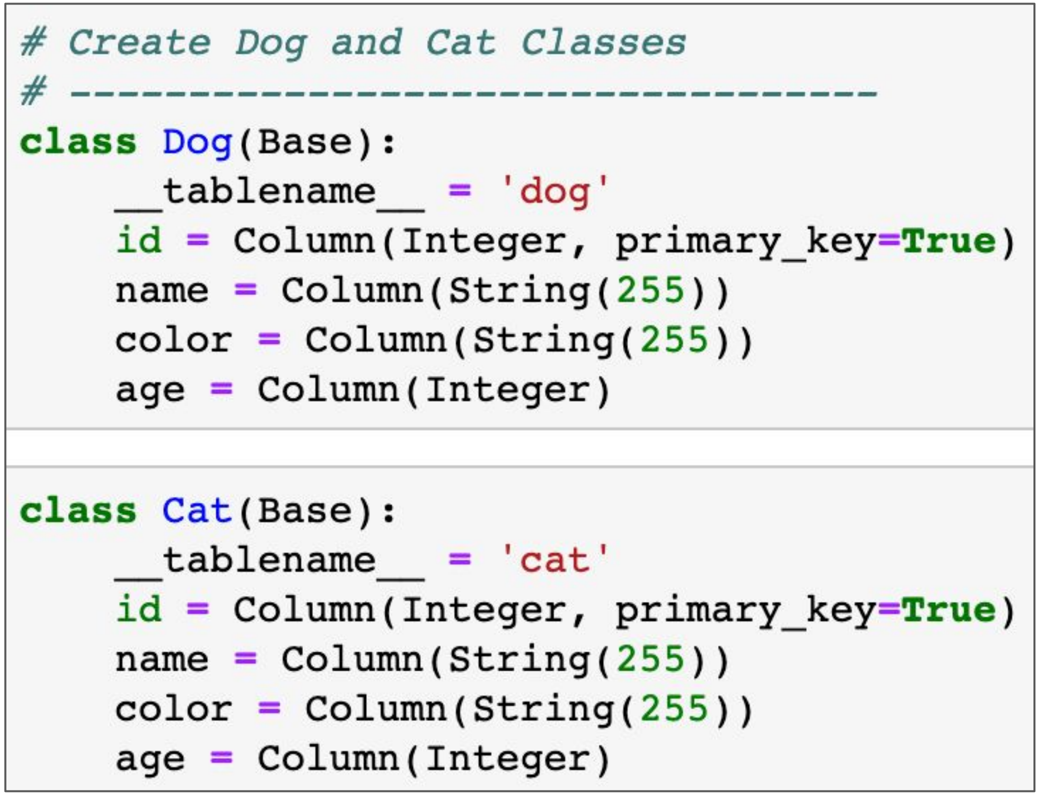
SQLAlchemy with Classes

SQLAlchemy is not just for making SQL queries in Python.

It can also update a SQL database using Python classes.

Python classes are traditionally used to bundle data and functions together.

In SQLAlchemy, they are used to define structures.



```python
# Create Dog and Cat Classes
# ----------------------------------
class Dog(Base):
    __tablename__ = 'dog'
    id = Column(Integer, primary_key=True)
    name = Column(String(255))
    color = Column(String(255))
    age = Column(Integer)
```

```python
class Cat(Base):
    __tablename__ = 'cat'
    id = Column(Integer, primary_key=True)
    name = Column(String(255))
    color = Column(String(255))
    age = Column(Integer)
```

Instructor Demonstration

Preview SQLAlchemy with Classes

Surfer Class

Object-oriented programming (OOP)

Object-oriented programming (OOP) is a style of coding based around the concept of "objects." These objects may contain data, often known as attributes, and functions, often known as methods.

Encapsulation

Object data (and often functions) can be neatly stored (or encapsulated).

Inheritance

New classes can be created based on other classes (the `Person` class is parent to the `Student` and `Teacher` classes).

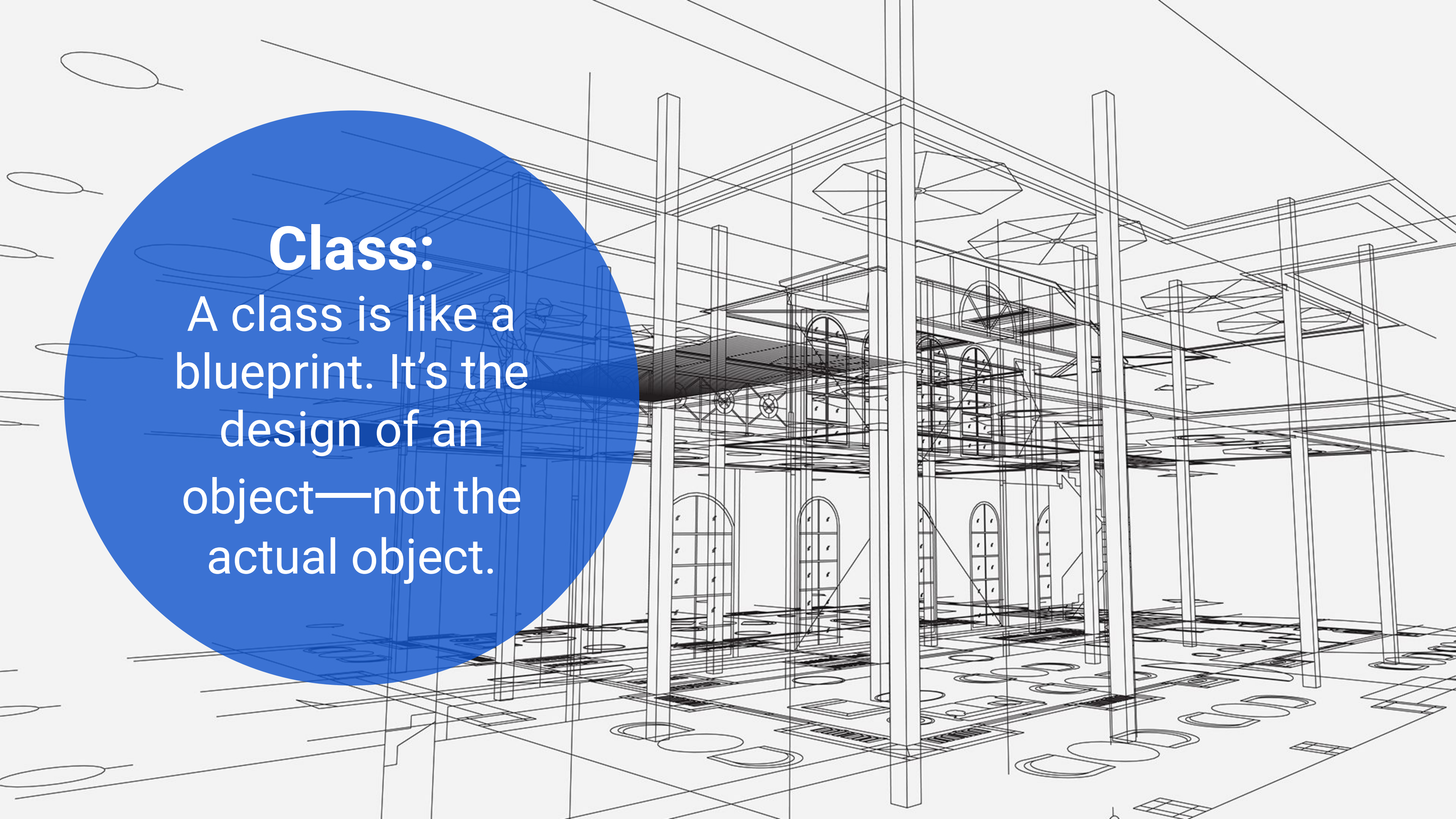
Abstraction

Creating a simple model of something that is complex.

Polymorphism

Multiple object types can implement the same functionality.

Python is a class-based programming language.



Class:

A class is like a blueprint. It’s the design of an object—not the actual object.

Objects can be created according to user-created blueprints, allowing developers to rapidly create objects with a similar structure/purpose—just with different values.

Instructor Demonstration

A Schooling on Classes

Activity: Surfer Class

In this activity, you will work on creating your own classes in Python.

Suggested Time:

15 minutes

Activity: Read All the SQL

Instructions	Create a class Surfer and initialize it with name, hometown, and rank-instance variables.
	Create an instance of a surfer.
	Then print the name, hometown, and rank of your surfer object.
Bonus	Create a `while` loop that will allow you to continuously create new instances of surfers using `input()`.
	Keep the loop going until the user indicates otherwise.

Time’s Up! Let’s Review.

Adding Methods to Python Classes

Adding methods to Python classes is easy as 1-2-3!

01

Define the function using `def`.

02

Provide a name and list of parameters.

03

Use `class.method()` to run the method in your script!

Instructor Demonstration

A Method to the Classes

Activity: Surfer Class Extended

In this activity, you will be reworking your Surfer script from earlier as you add in methods to perform specific tasks.

Suggested Time:

10 minutes

Activity: Surfer Class Extended

Instructions	Create a Surfer class that has name, hometown, rank, and wipeouts instance variables.
	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.

Time’s Up! Let’s Review.

Time to Code

Back to the SQL

Suggested Time:

20 minutes

Questions?

Activity: Surfing SQL

In this final activity, you will test your SQLAlchemy skills and update your Surfer database.

Suggested Time:

20 minutes

Activity: Surfing SQL

Instructions	Modify the Surfer class created during the previous activity so that it will function with SQLAlchemy.
	Create a new class called Board, which will function with SQLAlchemy and has the following parameters: ● 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.

Time's Up! Let's Review.