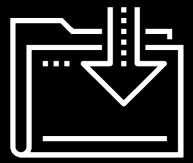
Introduction to SQLAlchemy

Data Boot Camp

Lesson 10.1



Class Objectives

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

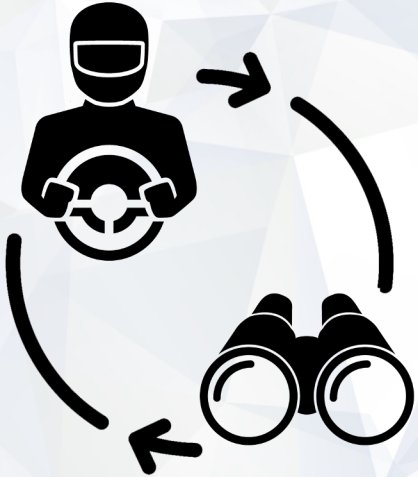
Connect to a SQL database by using SQLAlchemy.

Perform basic SQL queries by using `engine.execute()`.

Create Python classes and objects.

Create, read, update, and delete data from a SQL database by using SQLAlchemy's object-relational mapper (ORM).


WELCOME



Pair Programming Activity:

Looking into SQLAlchemy

In this activity, you'll be working in groups of two or three to research a few questions.

Suggested Time:

5 Minutes

Activity: Looking into SQLAlchemy

Research the following questions:

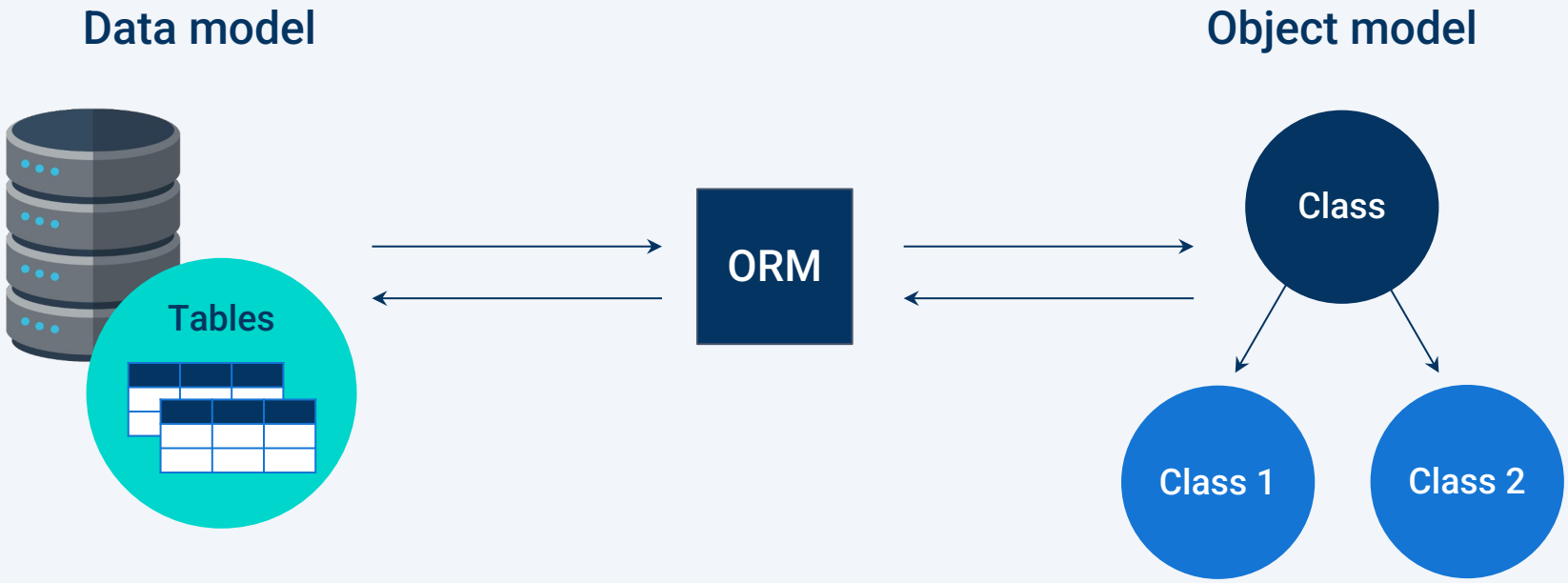
What is an ORM?

What are the benefits of using an ORM?

What are some of the disadvantages of using an ORM?

Time's Up! Let's Review.

Object-Relational Mapping (ORM)



Using An ORM

Advantages

- Portability: The ability to work across different SQL DB versions by using the same basic Python query.
- Simplified DB operations : create command line interfaces that allow users to construct SQL queries without needing to know the language.
- Operations Optimization: features like query caching, lazy loading and batching accelerate workflow Ops.

Disadvantages

- ORMs are like a new dialect of a language, so you have to learn how to use them.
- They may reduce control or ability to optimize a query.
- Performance overhead. ORMs introduce an added layer b/w the application code and DB.

Questions?

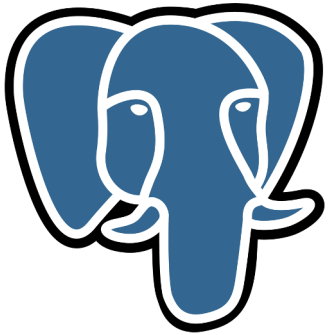
Introduction to

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

Introduction to SQLAlchemy

SQLAlchemy bridges the differences among the various SQL dialects. A single Python script that uses SQLAlchemy can perform the same query across the different SQL dialects, such as:

PostgreSQL



SQLite

MySQL

SQLAlchemy ORM Is Flexible

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

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

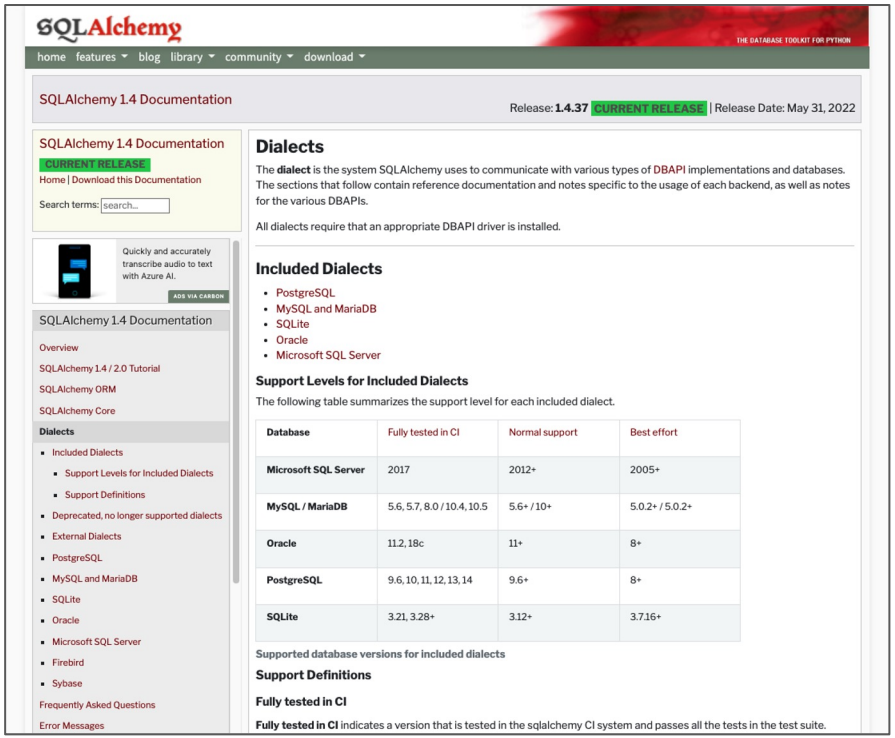
Or more Python:

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

Introduction to SQLAlchemy

The SQLAlchemy documentation lists SQL dialects that are compatible with SQLAlchemy.

- Complete documentation of the SQLAlchemy library is on the left side of the page.
- Consult this documentation to clarify any questions you may have.
- You should be able to fix most bugs this way.



Ice Cream Connection

Today you will only be working with SQLite databases.

SQLite is a SQL dialect that shares much of the same syntax as PostgreSQL, but it is entirely serverless.

How can a database be serverless?

Building a SQLAlchemy Connection

SQLite reads and writes directly to ordinary disk files, which can in turn be stored on a computer's hard drive. This makes it much easier to use to perform tests and share between users.

If you do not have SQLite installed, run the following code within your terminal/Git Bash:

```
conda install -c anaconda sqlite
```

Instructor Demonstration

Building a SQLAlchemy Connection

Questions?

Activity: Ice Cream Connection

In this activity, you will be creating and connecting to a new database using SQLAlchemy.

Suggested Time:

12 Minutes

Time's Up! Let's Review.

Questions?

Read All the SQL

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

SQLAlchemy and Pandas

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 by using Pandas and SQLAlchemy as you work to create new DataFrames based on U.S. Census data.

Suggested Time:

12 Minutes

Time’s Up! Let’s Review.

Questions?

Preview SQLAlchemy with Classes

Preview 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.

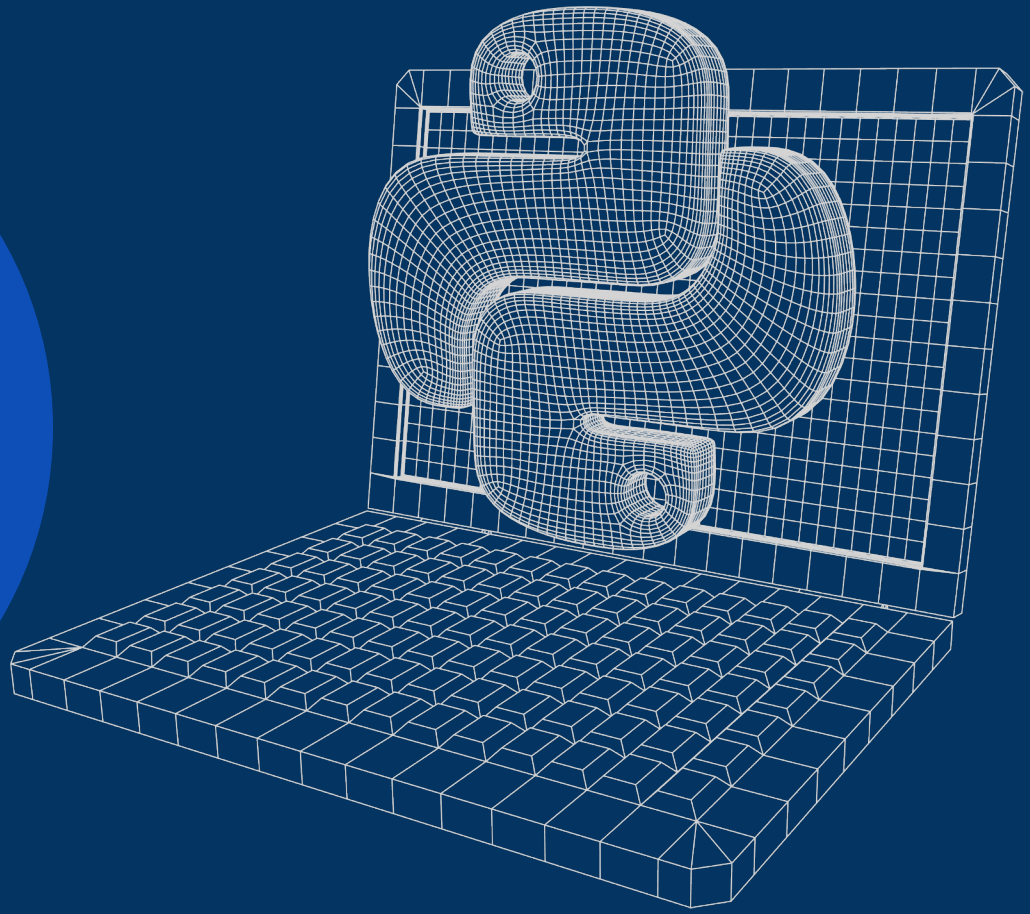
In SQLAlchemy, they are used to define structures.

```
# Sets an object to utilize the default declarative base in SQLAlchemy
Base = declarative_base()

# Creates Classes which will serve as the anchor points for our Tables
class Dog(Base):
    __tablename__ = 'dog'
    id = Column(Integer, primary_key=True)
    name = Column(String(255))
    color = Column(String(255))
    age = Column(Integer)

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

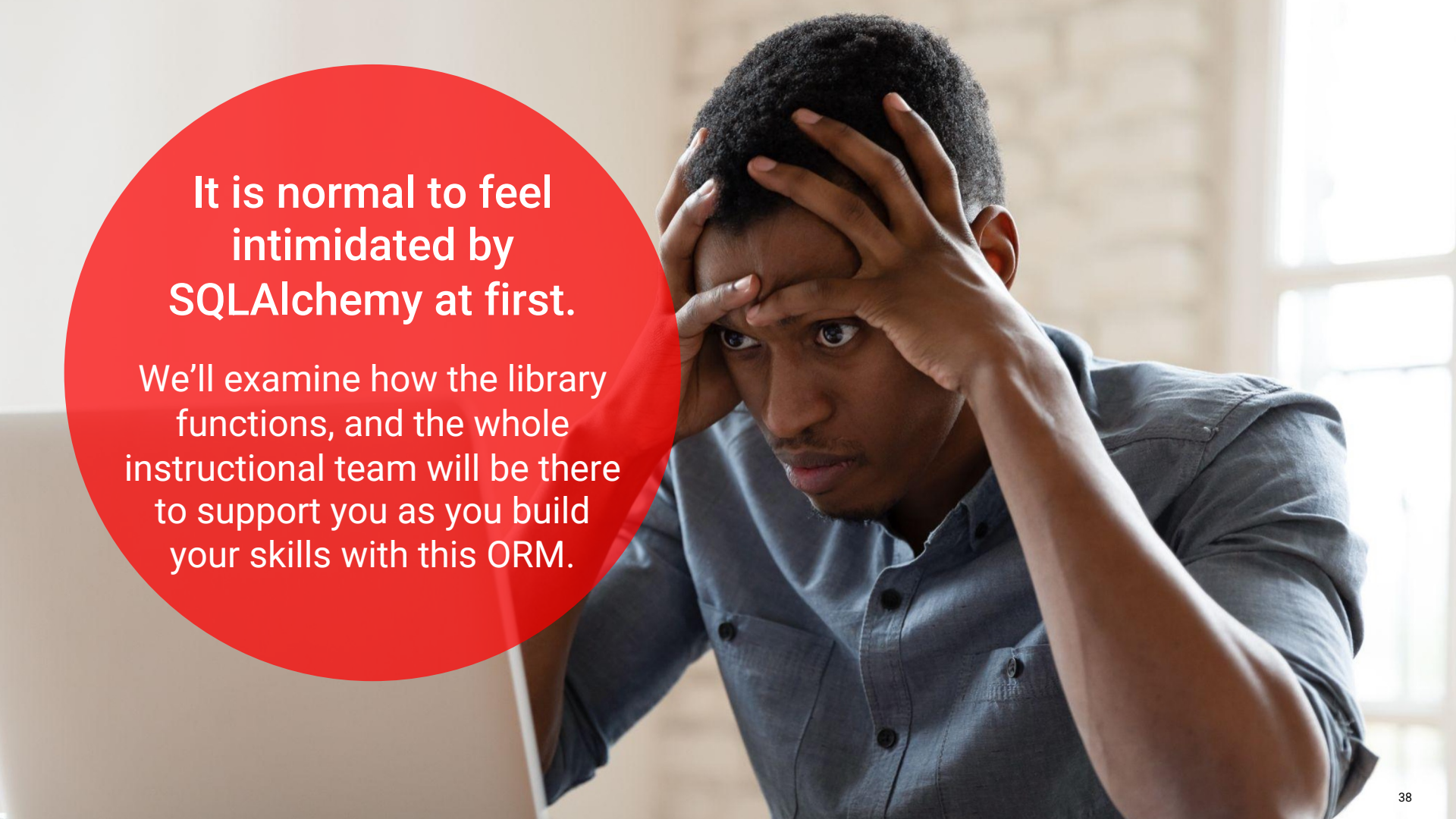
Classes are essentially blueprints for Python objects; they allow developers to create organized variables with keys, values, and methods on the fly.



In the case of SQLAlchemy, we can use classes to make a table blueprint and update the SQL schema.

Instructor Demonstration

Preview SQLAlchemy with Classes


It is normal to feel intimidated by SQLAlchemy at first.
We'll examine how the library functions, and the whole instructional team will be there to support you as you build your skills with this ORM.


Break

Surfer Class

Time for a crash course in object-oriented programming.

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.

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

Questions?

Activity: Surfer Class

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

Suggested Time:

15 Minutes

Time's Up! Let's Review.

Questions?

Surfer Class Extended

A Method to the Classes

Creating and attaching methods to Python classes is also easy to accomplish, allowing developers to attach regularly used functions to objects of similar types.

Add the Method

Adding methods to a class is very similar to the `__init__` method discussed earlier:

- define the function using `def`
- provide it with a name
- pass a list of parameters — including self — into the parentheses that follow.

Run the Method

To run the method in code, use the instance of a created object, and then, using dot notation, reference the method.

For example, `doggy.printHello()` would run the `printHello()` method for the `doggy` object.

A Method to the Classes

The `boast() method` contained within the `Expert` class takes in another object as a parameter and then prints out some statements based on its contents.

```
# Define the Expert class
class Expert():

    # A required function to initialize the class object
    def __init__(self, name):

        self.name = name

    # A method that takes another object as its argument
    def boast(self, obj):

        # Print out Expert object's name
        print("Hi. My name is", self.name)

        # Print out the name of the Film class object
        print("I know a lot about", obj.name)
        print("It is", obj.length, "minutes long")
        print("It was released in", obj.release_year)
        print("It is in", obj.language)
```

Instructor Demonstration

A Method to the Classes

Questions?

Activity: Surfer Class Extended

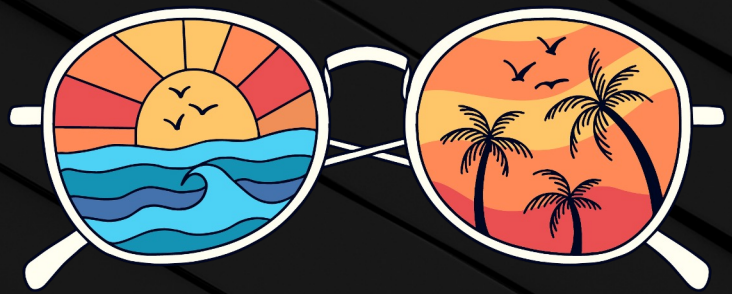
In this activity, you will rework your `Surfer` script as you add in methods to perform specific tasks.

Suggested Time:

10 Minutes

Time’s Up! Let’s Review.

Questions?



Surfing SQL

Time to Code

Back to the SQL

Suggested Time:

20 Minutes

Questions?

Activity: Surfing SQL

In this activity, you will test your SQLAlchemy skills to turn your Python classes into SQL database tables.

Suggested Time:

20 Minutes

Time's Up! Let's Review.

Questions?