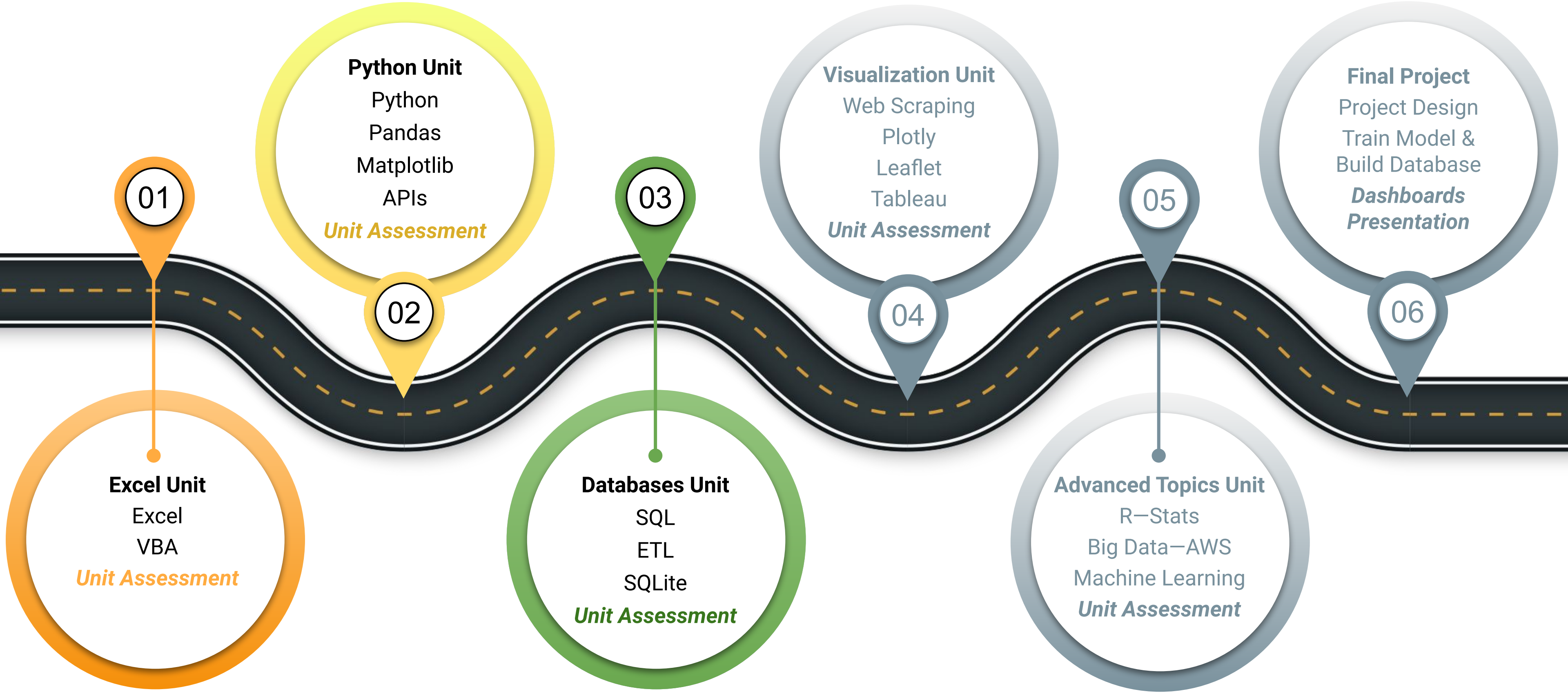
Introduction to SQLAlchemy

Data Boot Camp

Lesson 9.1

The Big Picture



01

Excel Unit
Excel
VBA
Unit Assessment

02

Python Unit
Python
Pandas
Matplotlib
APIs
Unit Assessment

03

Databases Unit
SQL
ETL
SQLite
Unit Assessment

04

Visualization Unit
Web Scraping
Plotly
Leaflet
Tableau
Unit Assessment

05

Advanced Topics Unit
R—Stats
Big Data—AWS
Machine Learning
Unit Assessment

06

Final Project
Project Design
Train Model &
Build Database
Dashboards
Presentation

Quick Tip for Success:

Think through how everything you're learning this week will be applied in the real world to create reusable, clean, and sustainable work.

Module 9

This Week: SQLAlchemy

This Week: SQLAlchemy

By the end of this week, you'll know how to:

- Differentiate between SQLite and PostgreSQL databases
- Connect to a SQLite database using SQLAlchemy
- Use SQLAlchemy to view table names and metadata from a SQLite database
- Use SQLAlchemy to connect to and query a SQLite database
- Convert a SQLAlchemy query into a Pandas DataFrame and plot the results

This Week’s Challenge

Using the skills learned throughout the week, students will query a SQLite database table, retrieve all the temperatures for the months of June and December, create DataFrames from these queries, and then generate summary statistics.

Career Connection

How will you use this module’s content in your career?

Module 9

How to Succeed This Week

Quick Tip for Success:

Take full advantage of office hours and your support network, including tutors and Learning Assistants.

Module 9

Today’s Agenda

Today's Agenda

By completing today's activities, you'll learn the following skills:

01 Connecting to SQL databases

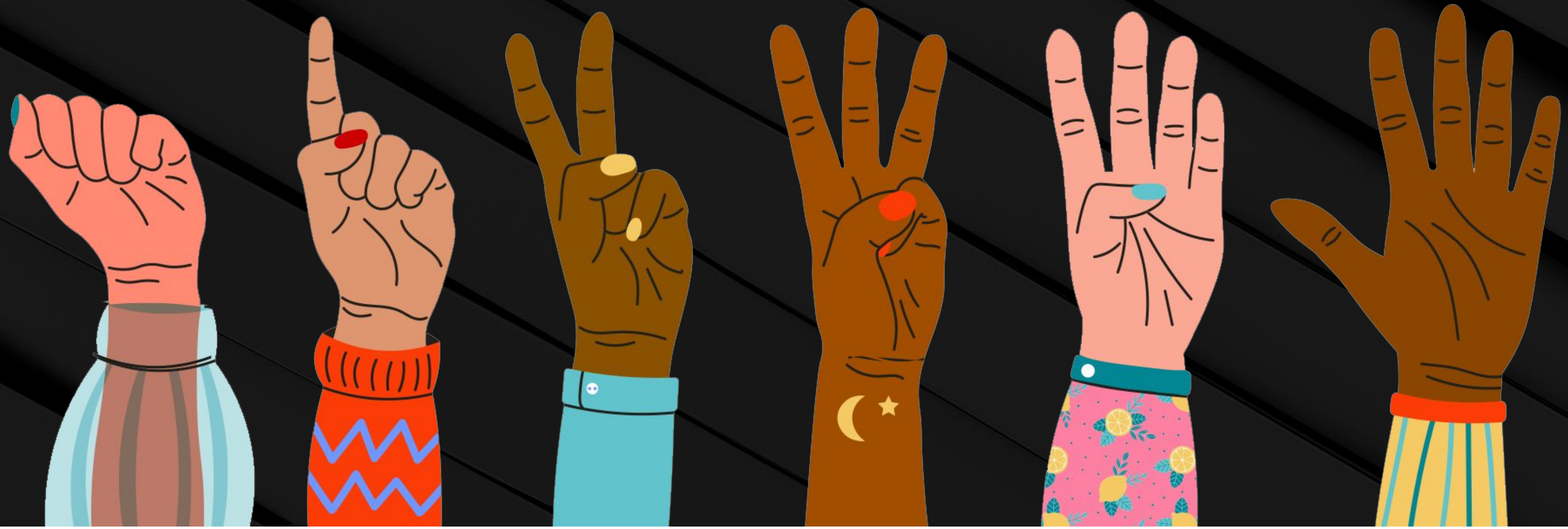
02 Performing basic SQL queries

03 Plotting query results

Make sure you've downloaded any relevant class files!

FIST TO FIVE:

How comfortable do you feel with this topic?



Introduction to SQLAlchemy

SQLAlchemy is a Python library that works across a variety of SQL dialects.

Write the query once, run it anywhere!

SQLAlchemy ORM Is Flexible

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

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

... or more Python!

```
jobs = session.query(Salaries.JobTitle)
for job in jobs:
    print(job)
```

Instructor Demonstration

SQLAlchemy Documentation

Questions?

Time to Code

SQLAlchemy Connection

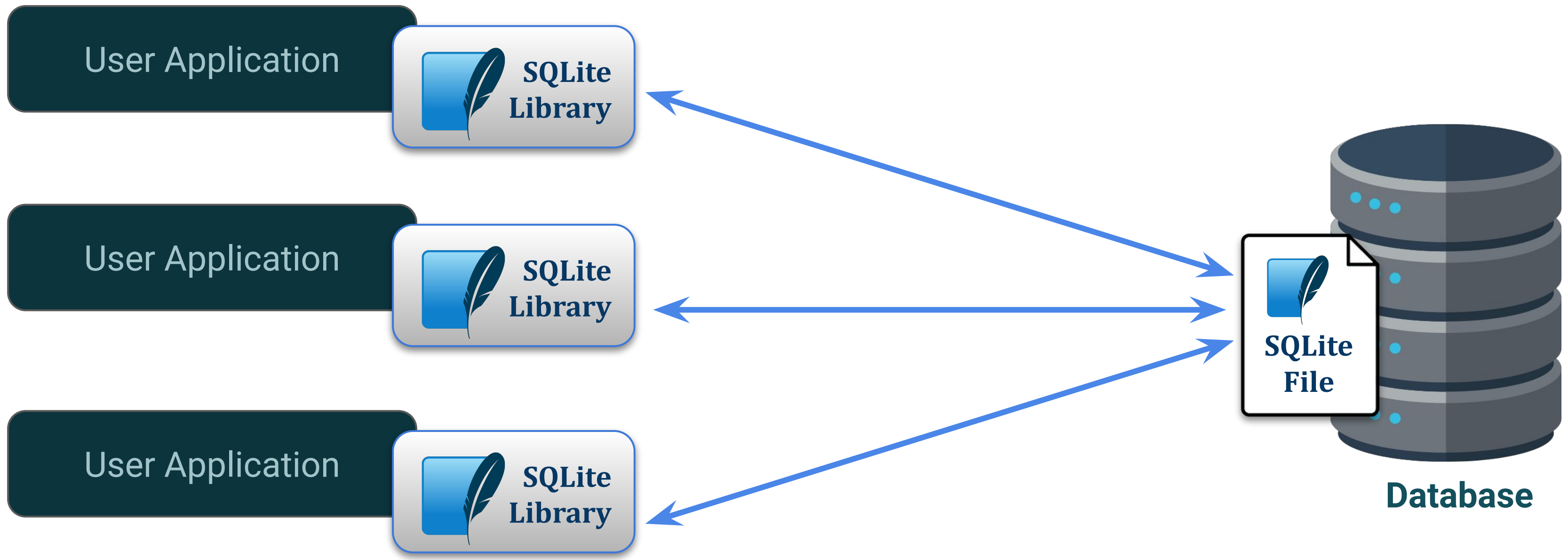
Suggested Time:

10 minutes

Today we will only be working with one SQL dialect—**SQLite**!

SQLite

SQLite is a SQL dialect that shares much of the same syntax as PostgreSQL, but it is entirely serverless. We can read and write to the SQLite file(db) on our computer.



One of the most impressive aspects of **SQLAlchemy ...**

... is how it integrates with **Pandas!**

Time to <code>

SQLite installation Check

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)
```

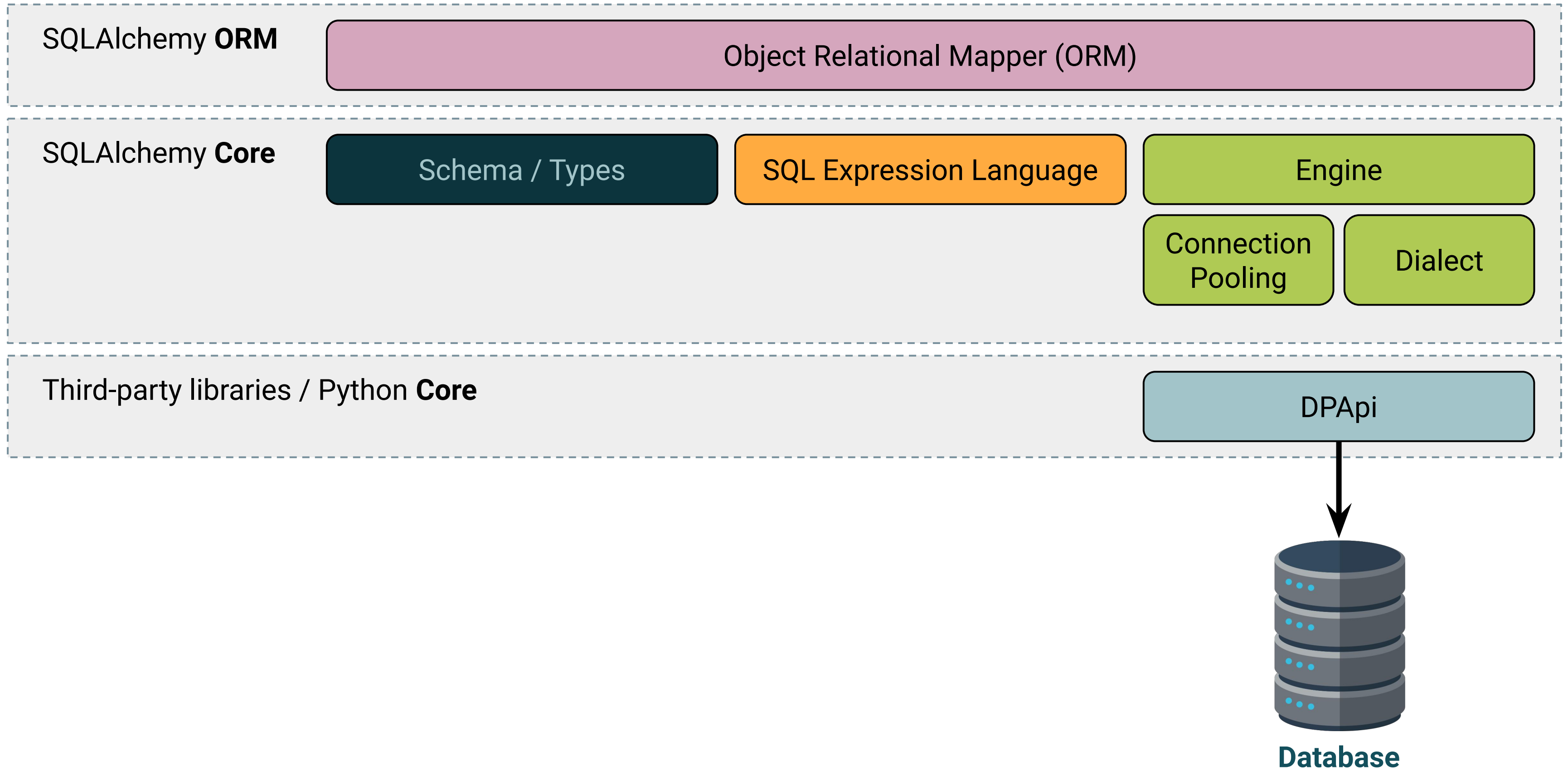
SQLite reads and writes directly to ordinary disk files that can in turn be stored on a computer's hard drive.

This makes it amazingly easy to perform tests with and to share between users.

Reflecting on SQL

As data analysts, developers often need to analyze existing data sources, meaning we would need to create SQLAlchemy classes according to a table's columns by hand—every single time.

SQLAlchemy Consists of the Core and the ORM



Instructor Demonstration

Reflecting an Existing Database

Activity: Reflecting on SQL

In this activity, you will practice reflecting an existing database using SQLAlchemy on a SQLite table that contains demographic data.

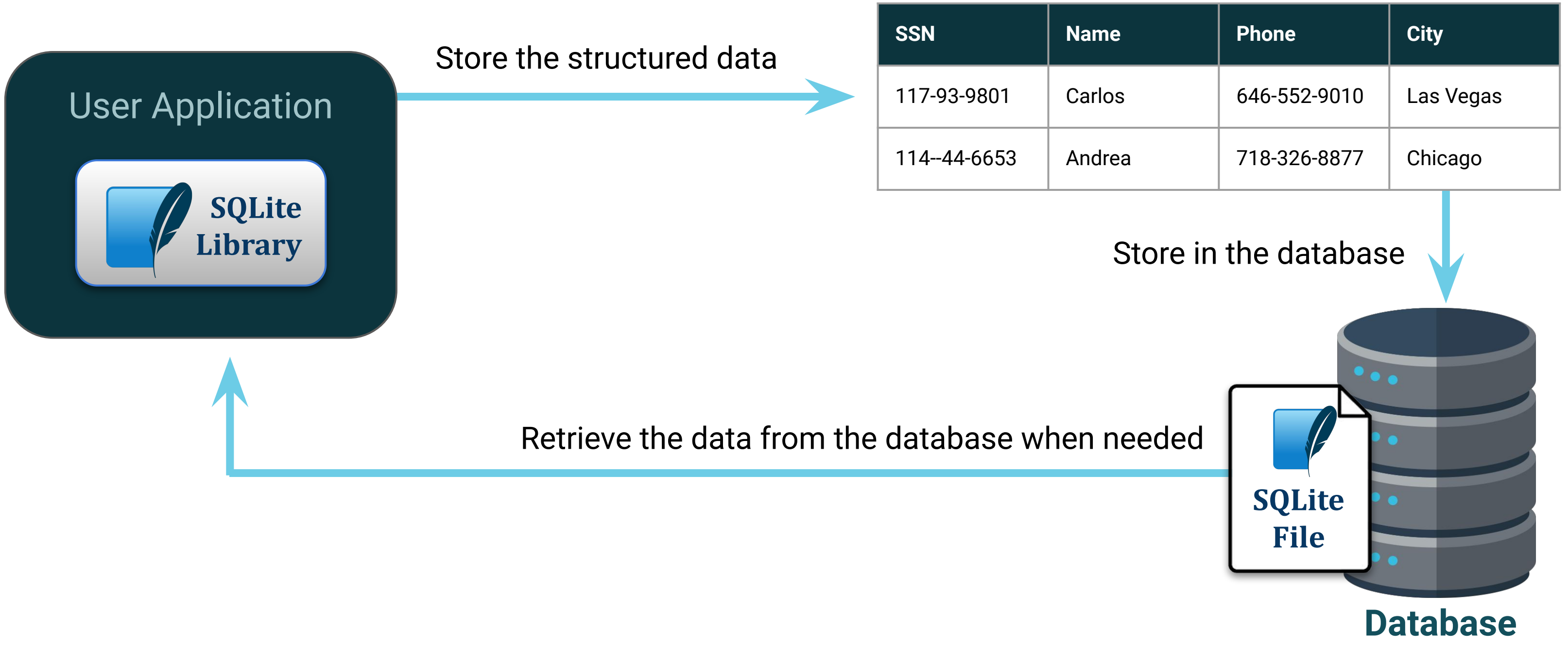
Suggested Time:
15 Minutes

Let's Review

SQLAlchemy Exploration

SQLAlchemy Exploration

When we reflect a database to collect the classes, it would be helpful to know what information is being stored in the table.



User Application
SQLite Library
Store the structured data
Store in the database
Retrieve the data from the database when needed
SQLite File
Database

SSN	Name	Phone	City
117-93-9801	Carlos	646-552-9010	Las Vegas
114--44-6653	Andrea	718-326-8877	Chicago

SQLAlchemy Exploration

`inspect()` function

In the first activity, we used the `inspect()` function to get the table and column schema, which allows us to get the column names and perform queries.

`Metadata()` and `Table()` objects

When we reflect a database, we can also retrieve the table metadata using `Metadata()` and `Table()` objects.

In this demonstration, we'll compare how to get the table column names using the `inspect()` function, and the `Metadata()` and `Table()` objects. Then, we'll query the table using SQL commands and Python.

Instructor Demonstration

SQLAlchemy Exploration

Questions?

Activity: Salary Exploration

In this activity, you will explore a SQLite database of salaries from San Francisco by using the `inspect()` function, `Metadata()` and `Table()` objects, and querying the table using SQL commands and Python.

Suggested Time:
15 Minutes

Let’s Review

Working with Dates

Times and dates have traditionally been trickier to manipulate in programming than integers or decimals.

Python offers libraries that make handling dates easier.

Working with Dates

Python's `datetime` library will parse, convert, compare, and filter by dates in a database.

New Data

Id	datetime	data
3	2020-01-02 01:00:00	xyz
4	2020-01-02 02:00:00	xyz

Insert New Data

Existing Table

Id	datetime	data
1	2020-01-01 01:00:00	xyz
2	2020-01-01 02:00:00	xyz
3	2020-01-03 01:00:00	xyz
4	2020-01-03 02:00:00	xyz

Result

New Table

Id	datetime	data
1	2020-01-01 01:00:00	xyz
2	2020-01-01 02:00:00	xyz
3	2020-01-02 01:00:00	xyz
4	2020-01-02 02:00:00	xyz
5	2020-01-03 01:00:00	xyz
6	2020-01-03 02:00:00	xyz

Instructor Demonstration

Query Dates Using SQLAlchemy

Time to Code

Dates

Suggested Time:

20 minutes

Questions?

The End