10.1 Lesson Summary - Introduction to SQLAlchemy

When people are learning data analytics, they generally begin by organizing data in a list or a table. For tabular data the columns are the fields where you define the kinds of data that you wish to store. The rows of your table contain the values for these fields. Python and most modern programming languages offer a more powerful way of organizing your data using object-oriented programming. In object-oriented programming the fields of your data are the kinds of properties an object has and the rows are the values of those fields for each data object. Object-oriented programing provides the additional benefit of being able to define methods associated with your data and share this functionality between objects. Object-oriented programing serves as the foundation for most of the tools we have used in this class including Python, Pandas, VBA, matplotlib and more. Object-oriented programming allows us use a tool like SQLAlchemy to simplify the way we interact with a database.

Concept: **SQLAlchemy** is a Python library that allows us to interact with our SQL databases more easily. SQLAlchemy can be used to perform SQL queries, for example:

*from sqlalchemy import create\_engine*

*database\_path = "../Resources/Sample\_Database.sqlite"*

*engine = create\_engine(f"sqlite:///{database\_path}")*

*data = engine.execute("SELECT \* FROM Sample\_Table")*

*for record in data:*

*print(record)*

* Activity: 01-Ins\_BasicSQL\_Connection, 02-Stu\_IceCreamStore

Concept: Pandas also provides the ability to interact with a SQL database. For example:

*my\_dataframe = pd.read\_sql("SELECT \* FROM Sample\_Data", conn)*

* Activity: 03-Ins\_ReadSQL, 04-Stu\_ReadAllTheSQLs

Concept: **Objects** in Python allow you to better organize your data and they allow you to pair methods with your data. In order to create an object, you must specify the properties of the object. In Python you do this by declaring a **class**. Once you have declared a class describing your objects you can then create an object by **instantiating** the class. To create person object, you could use the following code:

*import datetime*

*class Person():*

*def \_\_init\_\_(self, name, dob):*

*self.name = name*

*self.dob = dob*

*example\_person = Person("Susan", datetime.datetime(1990, 3, 14))*

*print(example\_person.name)*

* Activity: 06-Ins\_Classes, 07-Stu\_Surfer\_Class
* Suppl link: <https://www.w3schools.com/python/python_classes.asp>

Concept: Objects can contain data and methods that run code, usually code related to the class. If we wanted to add a method to our previous person object, we could use the following code:

*import datetime*

*class Person():*

*def \_\_init\_\_(self, name, dob):*

*self.name = name*

*self.dob = dob*

*def age(self):*

*return (datetime.datetime.now() - self.dob).days / 365.25*

*example\_person = Person("Susan", datetime.datetime(1990, 3, 14))*

*print(example\_person.age())*

* Activity: 08-Ins\_Classes\_With\_Methods, 09-Stu\_Surfer\_Class\_Extended

Concept: **SQLAlchemy**'s Object Relational Mapping (**ORM**) functionality allows you to use objects to avoid manually defining and maintaining your SQL data. To use SQLAlchemy to create a Person table in SQL with some data you could use the following code:

*import datetime*

*from sqlalchemy import Column, Integer, String*

*from sqlalchemy.ext.declarative import declarative\_base*

*Base = declarative\_base()*

*class Person(Base):*

*\_\_tablename\_\_ = ‘people'*

*id = Column(Integer, primary\_key=True)*

*name = Column(String(255))*

*dob = Column(DateTime)*

*person\_ling = Person(“Ling”, datetime.datetime(1967, 7, 9))*

*person\_ahmad = Person(“Ahmad”, datetime.datetime(1983, 3, 14))*

*from sqlalchemy.orm import Session*

*session = Session(bind=engine)*

*session.add(person\_ling)*

*session.add(person\_ahmad)*

*session.commit()*

*people\_list = session.query(Person)*

*for person in people\_list:*

*print(person.name)*

*print(person.dob)*

* Activity: 10-Ins\_SQL\_Alchemy\_Revisited, 11-Stu\_Surfer\_SQL