10.2 Lesson Summary - Advanced Usage of the SQLAlchemy ORM

SQLAlchemy provides the ability to perform the critical CRUD (Create, Read, Update, Delete) operations that are present in any serious data storage solution.

Concept: After you have declared a SQLAlchemy model class you can query the database to get the data from the table that corresponds to that model. You can also perform more complex queries, just like with SQL, where you target only the data that's necessary. If you declared Person as a SQLAlchemy model class you could get the number of people over 65 with the following code:

*number\_of\_people\_over\_65= session.query(Person).filter(person.age > 65).count()*

* Activity: 01-Ins\_Basic\_Querying, 02-Stu\_SharkSearch

Concept: **Scipy** can perform statistical analysis of SQLAlchemy query results. For example:

*from scipy import stats*

*from scipy import mean*

*mean(person.age)*

*stats.ttest\_ind(people\_age\_list, people\_height\_list)*

* Activity: 01-Ins\_Basic\_Querying

Concept: Once SQLAlchemy has created an object you can update that object's data and the database. To update the data in the database you must use the *dirty* attribute. For example:

*person = session.query(Person).filter\_by(name="John").first()*

*person.age += 1*

*session.dirty*

*session.commit()*

* Activity: 03-Ins\_Basic\_Updating

Concept: You can delete a row from a SQL database by using the delete method of the SQLAlchemy model object. For example:

*person = session.query(Person).filter\_by(id=4).delete()*

*session.commit()*

* Activity: 03-Ins\_Basic\_Updating

Concept: If you are accessing a SQL database that has already been created you can use reflection to create SQLAlchemy model classes directly from the SQL tables. For example:

*Base = automap\_base()*

*Base.prepare(engine, reflect=True)*

*Person = Base.classes.people*

* Activity: 05-Ins\_Reflection, 06-Stu\_ReflectingOnSQL

Concept: SQLAlchemy can get additional information from your database including table names, field names, and field types. For example:

*from sqlalchemy import inspect*

*inspector = inspect(engine)*

*inspector.get\_table\_names()*

*columns = inspector.get\_columns('people')*

*for column in columns:*

*print(column["name"], column["type"])*

* Activity: 07-Ins\_Exploration, 08-Stu\_SalaryExplore, 09-Par\_EmojiPlotting