**Operations with SQLAlchemy**

In this lesson, we performed all of our CRUD operations with SQLAlchemy on an SQLite database. Before we perform any operations, we must first import the necessary libraries, connect to our restaurantMenu.db, and create a session to interface with the database:

from sqlalchemy import create\_engine

from sqlalchemy.orm import sessionmaker

from database\_setup import Base, Restaurant, MenuItem

engine = create\_engine('sqlite:///restaurantMenu.db')

Base.metadata.bind=engine

DBSession = sessionmaker(bind = engine)

session = DBSession()

**CREATE**

We created a new Restaurant and called it Pizza Palace:

myFirstRestaurant = Restaurant(name = "Pizza Palace")

session.add(myFirstRestaurant)

sesssion.commit()

We created a cheese pizza menu item and added it to the Pizza Palace Menu:

cheesepizza = menuItem(name="Cheese Pizza", description = "Made with all natural ingredients and fresh mozzarella", course="Entree", price="$8.99", restaurant=myFirstRestaurant)

session.add(cheesepizza)

session.commit()

# READ We read out information in our database using the query method in SQLAlchemy:

firstResult = session.query(Restaurant).first()

firstResult.name

items = session.query(MenuItem).all()

for item in items:

print item.name

**UPDATE**

In order to update and existing entry in our database, we must execute the following commands:

2. Find Entry
4. Reset value(s)
6. Add to session
8. Execute session.commit()

We found the veggie burger that belonged to the Urban Burger restaurant by executing the following query:

veggieBurgers = session.query(MenuItem).filter\_by(name= 'Veggie Burger')

for veggieBurger in veggieBurgers:

print veggieBurger.id

print veggieBurger.price

print veggieBurger.restaurant.name

print "\n"

Then we updated the price of the veggie burger to $2.99:

UrbanVeggieBurger = session.query(MenuItem).filter\_by(id=8).one()

UrbanVeggieBurger.price = '$2.99'

session.add(UrbanVeggieBurger)

session.commit()

**DELETE**

To delete an item from our database we must follow the following steps:

1. Find the entry
2. Session.delete(Entry)
3. Session.commit()

We deleted spinach Ice Cream from our Menu Items database with the following operations:

spinach = session.query(MenuItem).filter\_by(name = 'Spinach Ice Cream').one()

session.delete(spinach)

session.commit()

Build a calendar app:

date(id(int), date, weekday) - things (time, location, description, last )

The user can add, edit, delete both

Show all dates that have schedules: /date/

Create a new date: /date/new/

Edite a date: date/date\_id/edit/

Delete a date:date/date\_id/delete/

Show schedules in a day: date/date\_id/schedule/

Create a new event in a day: date/date\_id/schedule/new/

Edit an event in a day: date/date\_id/schedule\_id/edit/

Delete an event: date/date\_id/schedule\_id/delete/