CS1520 Recitation:

Flask 3: Data Model (part1)

Jeongmin Lee

Data Models

Install SQLAlchemy

- (After turn on virtual env)
- PIP: pip install flask-sqlalchemy
- Then, run python and test: `import sqlalchemy`

First Code

- Import SQLAlchemy class and create db first.
- Create a Flask application object and set URI for the database to be used.

```
from flask import Flask, render_template
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///students.sqlite3'
```

*ORM: object-relational mapping

- Create an object of SQLAlchemy class with application object as the parameter.
- This object contains helper functions for ORM* operations. It provides a parent Model class using which user defined models are declared.

```
db = SQLAlchemy(app)
class Students(db.Model):
    id = db.Column(db.Integer, primary_key = True)
    name = db.Column(db.String(100))
    city = db.Column(db.String(50))
    # init func
    def __init__(self, id, name, city):
        self.id = id
        self.name = name
        self.city = city
```

- Create db!

```
db.create_all()
```

Session Objects

- Session object of SQLAlchemy manages all persistence operations of the data (ORM) objects.
- CRUD (remember!)
  - Create
  - Read
  - Update
  - Delete

Session Objects

- Session object of SQLAlchemy manages all pesistence operations of the data (ORM) objects.
- CRUD (remember!)
  - Create & Update : **db.session.add**(model object)
  - Read : **model.query.all()**
  - Delete : **db.session.delete**(model object)

- Add some students.

```
a = Students(id="0123", name="James Dean", city="Pittsburgh")

b = Students(id="0125", name="Lily Cory", city="Greensburgh")

db.session.add(a)

db.session.add(b)

db.session.commit()
```

- Query student

```
a = Students.query.filter_by(id="0125").first()

a.id

a.name
```

Questions?