1. What is an Object Relational Mapper and what are the advantages if using one?

An ORM is a tool used to convert database tables into Python classes and objects. This both makes is possible to interact with databases without the use of SQL syntax, and provides a simpler, more straightforward way of interact with databases, especially for editing specific values.

2. By this point, you've finished creating your Recipe app. How did it go? What's something in the app that you did well with? If you were to start over, what's something about your app that you would change or improve?

I felt like things weren't well overall, and I was able to follow the instruction provided in the tasks/exercises well enough. That being said, there are still a number of things beyond the base requirements of the task that I feel could be improved in the future. Examples includes the output displays for the user, and even more robust error handling to account for, and provide specific messaging for more unique error situations. I also feel that there are certain quality of life features that could be implemented for user, such as the ability to return to the main menu and cancel an input from any stage in the process.

3. Imagine you're at a job interview. You're asked what experience you have creating an app using Python. Taking your work for this Achievement as an example, draft how you would respond to this question.

I recently created a command line based Recipe management app using Python that allows users to create, view, edit, or delete recipes. Each recipe includes a number of values, including a unique ID, name, cooking time, and list of ingredients, plus a generated difficulty level based off the number of ingredients a user enters and the cooking time. There are two versions of the app, both which use MySQL to save the recipes to a SQL database. The first version used SQL syntax with MySQL's mysql-connector to interact with the database, while the second version used SQLAlchemy's ORM for this purpose. This project required a firm grasp of different uses for Python's different data types, how to handle errors in order to not cause the script to crash, how Python's scripts can interact with SQL databases in a variety of ways, and how to apply OOP principles in a Python application.

4. You're finished Achievement 1! Before moving on to Achievement 2, take a moment to reflect your learning in the course so far:

a) What went well in this achievement?

I thought I was able to quickly grasp and effectively implement most of the new skills introduced in this Achievement.

b) What's something you're proud of?

In addition to baseline requirements, my mentor mentioned ways I could improve my project beyond what was required and sometimes what was covered in the exercises. I thought I did a good job of figuring out how to implement these things in the project as well, even if they weren't covered directly.

c) What was the most challenging aspect of this Achievement?

I thought considering all of the different error situations that could come up throughout the script, either from user entry or from another source, was somewhat challenging.

d) Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Python skills?

I thought this was a good introduction into some Python concepts and the syntax, but I am hoping to learn more about how to utilize Python outside of scripts designed to be run in the terminal.

e) What's something you want to keep in mind to help you do your best in Achievement 2?

I don't really have anything especially specific in mind, just to keep improving my understanding and implementation of the concepts I learned in this achievement.