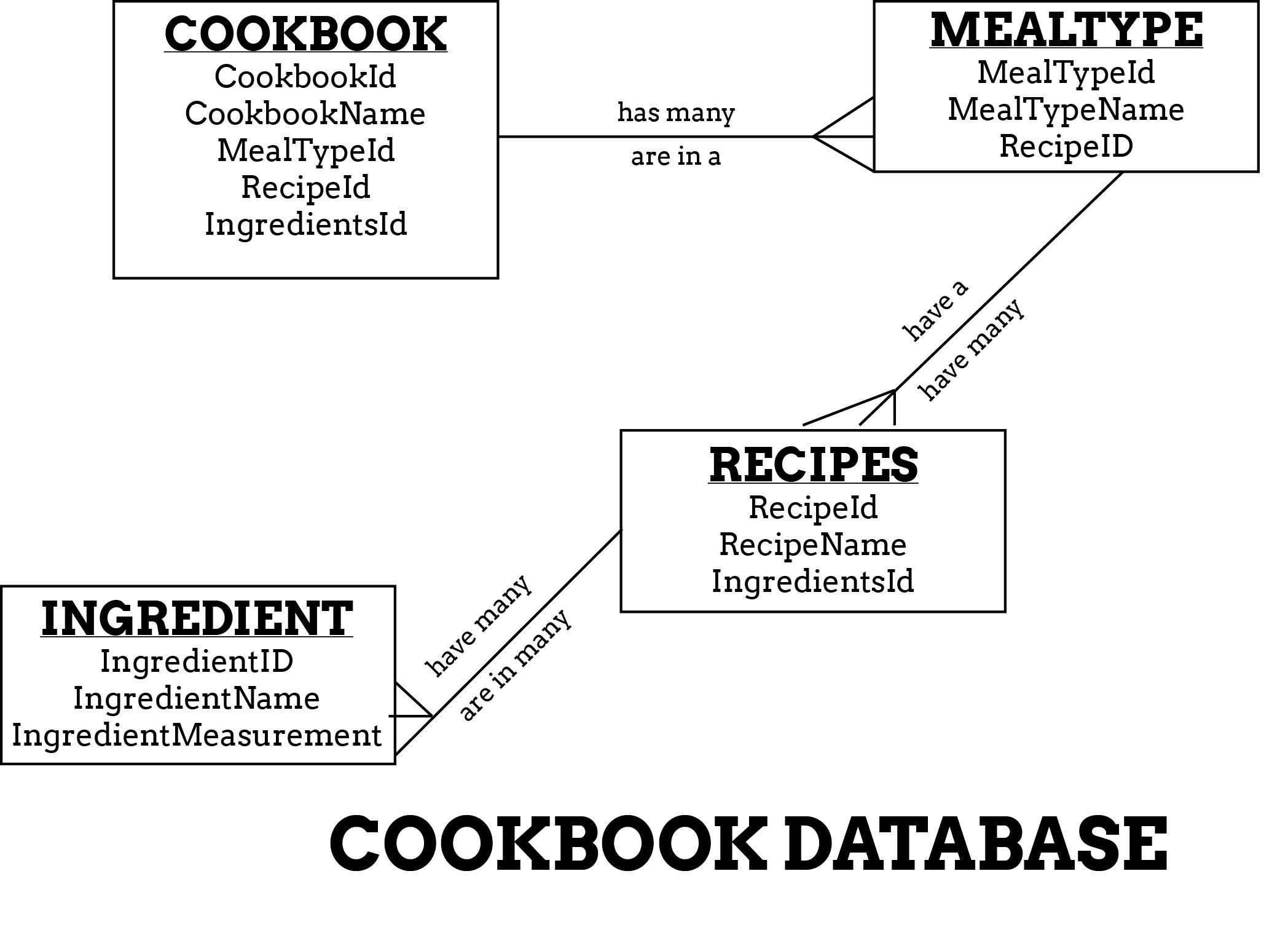
# Ashley Slaughter

# Data Modeling Questions

* Design a database to model a cookbook with meal types, recipes, ingredients.
  1. You can use paper to document your data model.



# SQL Questions

* Given the following SQL SELECT statement what are the highlighted text represent?

SELECT ***CUST***.CUSTOMER AS ***NAME***, ***CUST***.CUSTOMER\_ADDRESS

This line is selecting the

FROM CUSTOMER\_V ***CUST***

WHERE ***NAME*** = ‘Home Furnishings’;

* Given the following Orders and Customers tables:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OrderID** | | **CustomerID** | **OrderDate** | |
| 10308 | | 2 | 1996-09-18 | |
| 10309 | | 37 | 1996-09-19 | |
| 10310 | | 77 | 1996-09-20 | |
| **CustomerID** | **CustomerName** | | | **ContactName** | | **Country** |
| 1 | Alfreds Futterkiste | | | Maria Anders | | Germany |
| 2 | Ana Trujillo Emparedados y helados | | | Ana Trujillo | | Mexico |
| 3 | Antonio Moreno Taquería | | | Antonio Moreno | | Mexico |

What would the result set be given the following SQL SELECT statement?

SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate  
FROM Orders  
INNER JOIN Customers  
ON Orders.CustomerID=Customers.CustomerID;

The results would show the OrderID, CustomerName, and OrderDate for all Customers.

What would the result set be given the following SQL SELECT statement?

SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate  
FROM Orders  
LEFT JOIN Customers  
ON Orders.CustomerID=Customers.CustomerID;

The results would show the OrderID, CustomerName, and OrderDate for all Customers.

# Entity Framework Questions

* What do the following Entity Framework Object Services provide for your application in regards to data from a database?
  1. Materialization – process of changing data in the database and structure into objects
  2. Change Tracking – the layer between the data and the model that tracks changes
  3. Object identities – The code in the classes that make up the database tables
* Design a code first data model which has a Project class that can contain a bunch of tasks.

# Object Orientation Questions

* What are the basic concepts of OOP?
  1. Inheritance
  2. Polymorphism
  3. Encapsulation
  4. Abstraction
* How do you program behavior into your C# class?
  1. Methods
* Explain method overriding.

Method override has to do with the number and types of parameters that can be used in one method name.

* What is Inheritance?

Classes can inherit from one another. You can give you the ancestors of your base class access to the properties and methods of your base class.

* What is abstract class?

An abstract class had properties and methods that can be overridden and used more specifically for what the program needs.

# MVC Questions

What is an example URL that would call the following controller method, assuming the default routes have been configured? –I don’t see the following controller method you are referring? Are you talking about the code under Part II? If so, I’m going to go with Catalog… Maybe Index/Catalog?

Part II: What HTTP Verb is used? ACTIONRESULT

public class CatalogController : Controller

 {

     public ActionResult Specifications(int id)

     {

         var model = new SpecModel(id);

         return View();

     }

}

Examine the following View for an MVC Application:

@model IEnumerable<MVCGuidedLab.Models.Color>

@{

    ViewBag.Title = "Index";

}

<h2>Index</h2>

<p>

    @Html.ActionLink("Create New", "Create")

</p>

<table class="table">

    <tr>

        <th>

            @Html.DisplayNameFor(model => model.Name)

        </th>

        <th>

            @Html.DisplayNameFor(model => model.Value)

        </th>

        <th></th>

    </tr>

@foreach (var item in Model) {Go

    <tr>

        <td>

            @Html.DisplayFor(modelItem => item.Name)

        </td>

        <td>

            @Html.DisplayFor(modelItem => item.Value)

        </td>

        <td>

            @Html.ActionLink("Edit", "Edit", new { id=item.Id }) |

            @Html.ActionLink("Details", "Details", new { id=item.Id }) |

            @Html.ActionLink("Delete", "Delete", new { id=item.Id })

        </td>

    </tr>

}

</table>

1. What type is the Model? – I didn’t know models had types?
2. Is the model a single object, or a sequence? – I have no idea.
3. What properties are on the items in the model? – Name and Value

# Programming Exercises

To turn in this exam, you’ll create a fork of a repository, make modifications to the project in that repository, and submit a pull request with your changes.

We’ll walk you through the github workflow for those pieces. However, the code will be yours.

Go to Github.com and fork the repository <https://github.com/BillWagner/ExperienceITExam>

Then, clone your fork to your desktop. You’ll do that by clicking the “Clone in Desktop” button on the github page.

Now, you’re ready to do your work.

1. Open the program.cs file in the Loops project. The Main method has comments that describe what you should do. Make your changes.
2. Open the program.cs file in the Extension methods project. The Main method has comments that describe the code you should add. Make your changes.

* I don’t understand how a person can be greaterthan, lessthan or equal to another person?

1. Open the program.cs file in the Lazy Evaluation project. The Main method has comments that describe the code you should add. Make those changes as well.
2. Open the program.cs file in the Query Expressions project. As before, the Main method has comments that describe the code you should or change.
3. Make a new ASP.NET MVC project and add it to the solution. Add a controller called DiceRoll controller. Modify the index method and the index view to display all the combinations of results from rolling 2 six-sided dice. (For example, { 1, 1}, {1, 2} etc.

Commit your changes, and then push them to github.

Now, you’re ready to submit a pull request. Navigate to your fork of the repository in github.com in a browser. Below the Code tab on the right side, you’ll see a link that says “Pull Request”. Click that. Once you’re on the Pull Request page, submit a new pull request. Add your name, and any comments you would like on your pull request, and submit the request.

We can now look at your changes, and see how you did.