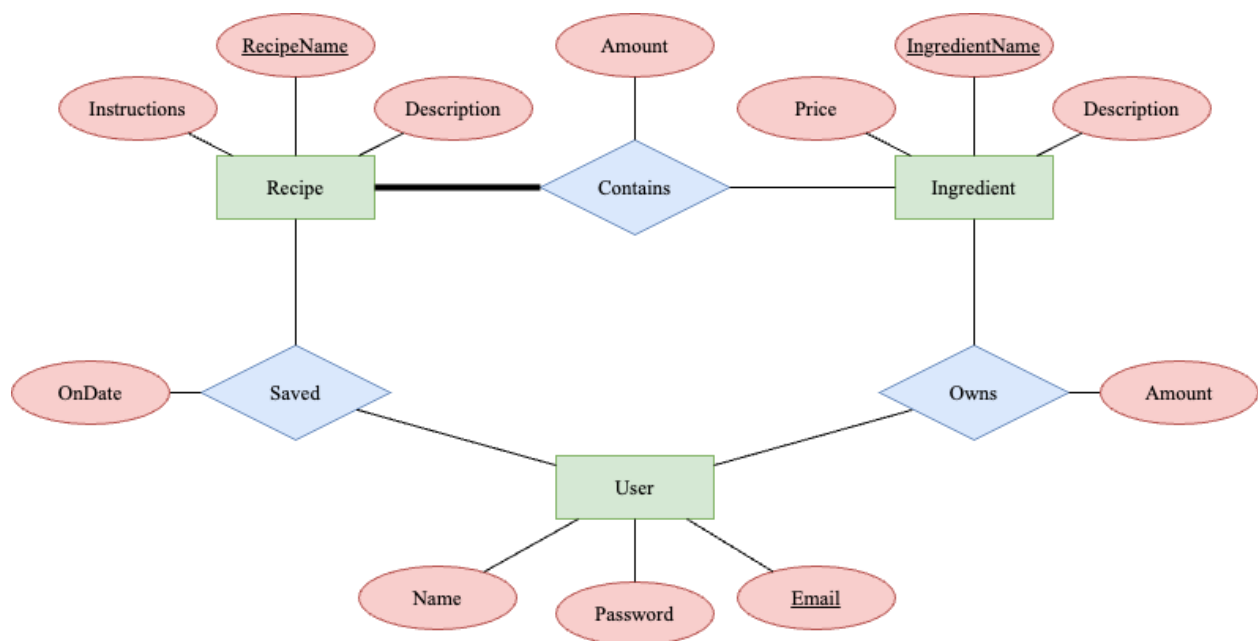


MXLGY Revision

Refined Idea

- The core idea will remain relatively unchanged.
- Our entity-relationship model is fairly stable at this point, but our API will see more iteration.

Revised ER Diagram



- We have reduced the entity-relationship model's dependence on IDs. This will reduce complexity by allowing us to use natural keys, such as the name of a recipe or the name of an ingredient.
- We have decided to use an email address as the primary key for users.

COSC-257 Databases

- We still suspect that the ability to store multiple recipes with the same name may be desired. This could lead us to use an additional attribute to form a composite key.

Role Assignments

Beckett → Backend Development with Express.js, Web Server Setup, API Design

Sawyer → Frontend Development with Next.js and React, HTML and CSS, Data Cleanup

Software

- **Node.js**
 - We have installed the latest stable version of Node.js using the popular Node Version Manager system.
- **Express.js**
 - We have installed Express.js with npm.
- **Next.js**
 - We have installed Next.js with npm.
- **NGINX**
 - We have installed NGINX to a new user called *mxlgy*.
 - We have enabled and started the NGINX service using *systemctl*.
 - We have opened the HTTP and HTTPS ports using *firewalld* and have a test page at <http://cosc-257-node13.cs.amherst.edu/>

COSC-257 Databases

- We intend to use NGINX as a proxy server and use HTTPS with Let's Encrypt.
- **PostgreSQL**
 - We have installed and configured PostgreSQL.
 - We have created a Postgres user called *mxlgy* and a primary database called *mxlgy*.
 - We don't intend yet to expose the SQL database directly to the internet and will instead access the server via our Express.js backend.
 - We have experimented with *psql* commands and have worked with test data.

Open Questions

- How will we handle user accounts?
 - Will we create "dummy" accounts that users can log into?
 - Will we store "dummy" passwords in plaintext or use passwords at all?
 - Will we implement authentication with a system like "Sign-In with Google"?
- Will we allow users to add their own recipes?
 - This could require more relations in our model.

Initial Relational Schema

Entities:

```
CREATE TABLE users (  
    Email VARCHAR(100) PRIMARY KEY,  
    Name VARCHAR(40),  
    Password VARCHAR(30) NOT NULL  
);
```

```
CREATE TABLE recipes (  
    RecipeName VARCHAR(100) PRIMARY KEY,  
    Instructions VARCHAR(2500),  
    Description VARCHAR(2500)  
);
```

```
CREATE TABLE ingredients (  
    IngredientName VARCHAR(100) PRIMARY KEY,  
    Description VARCHAR(2500),  
    Price REAL,  
    Unit VARCHAR(50)  
);
```

COSC-257 Databases

Relations:

```
CREATE TABLE owns (  
    Email VARCHAR(100),  
    IngredientName VARCHAR(100),  
    Amount REAL,  
    PRIMARY KEY (Email, IngredientName),  
    FOREIGN KEY Email REFERENCES users(Email),  
    FOREIGN KEY IngredientName REFERENCES ingredients(IngredientName)  
);
```

```
CREATE TABLE saved (  
    Email VARCHAR(100),  
    RecipeName VARCHAR(100),  
    OnDate TIMESTAMP,  
    PRIMARY KEY (Email, RecipeName),  
    FOREIGN KEY Email REFERENCES users(Email),  
    FOREIGN KEY RecipeName REFERENCES recipes(RecipeName)  
);
```

```
CREATE TABLE contains (  
    RecipeName VARCHAR(100),  
    IngredientName VARCHAR(100),  
    Amount REAL,  
    PRIMARY KEY (RecipeName, IngredientName),  
    FOREIGN KEY RecipeName REFERENCES recipes(RecipeName),  
    FOREIGN KEY IngredientName REFERENCES ingredients(IngredientName)  
);
```