Install Libraries

In [6]:
```
! pip3 install --upgrade --quiet neo4j
```

Open Database Connection

In [7]:
```
from neo4j import GraphDatabase
```

In [8]:
```
def connect_database(URI, AUTH):
    driver = GraphDatabase.driver(URI, auth=AUTH)
    driver.verify_connectivity()  # Verify connection once the driver is creat
ed
    return driver
```

In [9]:
```
# URI = "bolt://zelpa.net:7687"
# AUTH = ("neo4j", "55QvQu95HG")
URI = "bolt://localhost:7687"
AUTH = ("neo4j", "recipeApp")
```

In [10]:
```
driver = connect_database(URI, AUTH)
```

Reset nodes

In [11]:
```
query = "MATCH (n) DETACH DELETE n"
with driver.session() as session:
    result = session.run(query)
    summary = result.consume()  # Get the summary of the execution

# Print the number of nodes deleted
print(f"Deleted {summary.counters.nodes_deleted} nodes from the database.")
```

```
Deleted 30 nodes from the database.
```

User

In [12]:
```
def create_user_node(driver, user_data):
    query = '''
    CREATE (u:User {
        username: $username,
        name: $name,
        email: $email,
        phone: $phone
    })
    '''

    # Run the query with the parameters
    with driver.session() as session:
        session.run(query,
                    username=user_data['username'],
                    name=user_data['name'],
                    email=user_data['email'],
                    phone=user_data['phone'])

    print(f"User {user_data['username']} created!")
```

The data below would be gathered from the website

In [13]:
```
user1_data = {
    'username': 'brody675',
    'name': 'Brody',
    'email': 'brodykerr675@gmail.com',
    'phone': '832-600-8473'
}

user2_data = {
    'username': 'jared123',
    'name': 'Jared',
    'email': 'jared123@gmail.com',
    'phone': '123-456-7890'
}
```

In [14]:
```
create_user_node(driver, user1_data)
create_user_node(driver, user2_data)
```

```
User brody675 created!
User jared123 created!
```

Pantry

Node automatically connected to other nodes

In [15]:
```
def create_pantry_node(driver, user):
    query = '''
    MATCH (user:User{username: $user})
    CREATE (pantry:Pantry {name: $pantryName})
    CREATE (user)-[:OWNS]->(pantry)
    '''
    pantry_name = user + "Pantry"
    with driver.session() as session:
        session.run(query, pantryName=pantry_name, user=user)
    print(f"Pantry node {pantry_name} created!")
```

In [16]:
```
create_pantry_node(driver, user1_data['username'])
create_pantry_node(driver, user2_data['username'])
```

```
Pantry node brody675Pantry created!
Pantry node jared123Pantry created!
```

Cuisine

Shouldn't need to run based off user interaction they should already be in database the user will only be creating the connections to the them

In [17]:
```
def create_cuisine_node(driver, cuisine):
    query = "CREATE (cuisine:Cuisine {name: $cuisineName})"
    with driver.session() as session:
        session.run(query, cuisineName=cuisine)
    print(f"Cuisine node {cuisine} created!")
```

In [18]:
```
create_cuisine_node(driver, "Staple")
```

```
Cuisine node Staple created!
```

Recipe

In [19]:
```
def create_recipe_node(driver, user, recipe_data):
    query = '''
    CREATE (recipe:Recipe {
        name: $recipeName,
        title: $title,
        description: $description
    })
    '''
    recipe_name = user + recipe_data['name']

    # Run the query with the parameters
    with driver.session() as session:
        session.run(query,
                    recipeName=recipe_name,
                    title=recipe_data['title'],
                    description=recipe_data['description'])

    print(f"Recipe {recipe_name} created!")
```

The data below would be gathered from the website

In [20]:
```
recipe1_data = {
    'name': 'BananaBread',
    'title': 'Brodys Favorite Banana Bread',
    'description': 'Has good moistness will save for future'
}
recipe2_data = {
    'name': 'GrilledCheese',
    'title': 'Brodys Favorite Sammich',
    'description': 'Very cheesy and crispy :)'
}
recipe3_data = {
    'name': 'Chili',
    'title': 'Brodys Favorite Chili',
    'description': 'Very spicy'
}
```

In [21]:
```
create_recipe_node(driver, user1_data['username'], recipe1_data)
create_recipe_node(driver, user1_data['username'], recipe2_data)
create_recipe_node(driver, user1_data['username'], recipe3_data)
```

```
Recipe brody675BananaBread created!
Recipe brody675GrilledCheese created!
Recipe brody675Chili created!
```

Ingredient

Shouldn't need to run based off user interaction they should already be in database the user will only be creating the connections to the them

In [22]:
```
def create_ingredient_node(driver, ingredient):
    query = "CREATE (ingredient:Ingredient {name: $ingredientName})"
    with driver.session() as session:
        session.run(query, ingredientName=ingredient)
    print(f"Ingredient node {ingredient} created!")
```

In [23]:
```
create_ingredient_node(driver, "Flour")
create_ingredient_node(driver, "Baking Soda")
create_ingredient_node(driver, "Salt")
create_ingredient_node(driver, "Butter")
create_ingredient_node(driver, "Brown Sugar")
create_ingredient_node(driver, "Eggs")
create_ingredient_node(driver, "Bananas")
create_ingredient_node(driver, "White Bread")
create_ingredient_node(driver, "Cheddar Cheese")
create_ingredient_node(driver, "Beef")
create_ingredient_node(driver, "Onion")
create_ingredient_node(driver, "Tomato Sauce")
create_ingredient_node(driver, "Kindey Beans")
create_ingredient_node(driver, "Chili Powder")
create_ingredient_node(driver, "Garlic Powder")
create_ingredient_node(driver, "Black Pepper")
```

```
Ingredient node Flour created!
Ingredient node Baking Soda created!
Ingredient node Salt created!
Ingredient node Butter created!
Ingredient node Brown Sugar created!
Ingredient node Eggs created!
Ingredient node Bananas created!
Ingredient node White Bread created!
Ingredient node Cheddar Cheese created!
Ingredient node Beef created!
Ingredient node Onion created!
Ingredient node Tomato Sauce created!
Ingredient node Kindey Beans created!
Ingredient node Chili Powder created!
Ingredient node Garlic Powder created!
Ingredient node Black Pepper created!
```

Tools

Utensils used in creating recipe

In [ ]:
```
```

Group

In [24]:
```
def create_group_node(driver, group):
    query = "CREATE (group:Group {name: $groupName})"
    with driver.session() as session:
        session.run(query, groupName=group)
    print(f"Group node {group} created!")
```

In [25]:
```
create_group_node(driver, "Senior Project")
```

```
Group node Senior Project created!
```

Shopping List

In [26]:
```
def create_shopping_list_node(driver, user):
    query = "CREATE (shoppinglist:ShopingList {name: $shoppinglistName})"
    shopping_list_name = user + "ShoppingList"
    with driver.session() as session:
        session.run(query, shoppinglistName=shopping_list_name)
    print(f"Shopping List node {shopping_list_name} created!")
```

In [27]:
```
create_shopping_list_node(driver, user1_data['username'])
```

```
Shopping List node brody675ShoppingList created!
```

Meal

In [28]:
```
def create_meal_node(driver, user, meal):
    query = '''
            CREATE (meal:Meal {
                name: $mealName,
                title: $meal
                })
            '''
    meal_name = user + meal
    with driver.session() as session:
        session.run(query,
                    mealName=meal_name,
                    meal=meal)
    print(f"Meal node {meal_name} created!")
```

In [29]:
```
def connect_meal_node(driver, user, meal, recipe):
    query = '''
            MATCH (meal:Meal{name:$mealName})
            MATCH (recipe:Recipe{name:$recipeName})
            CREATE (meal)-[:MADE_WITH]->(recipe)
            '''
    meal_name = user + meal
    recipe_name = user + recipe
    with driver.session() as session:
        session.run(query,
                    mealName=meal_name,
                    recipeName=recipe_name)
    print(f"Meal node {meal_name} connected to {recipe}!")
```

In [30]:
```
meal1_data = ['GrilledCheese', 'Chili']
meal1_title = "Grilled Cheese w/Chili"
```

In [31]:
```
create_meal_node(driver, user1_data['username'], meal1_title)
for recipe in meal1_data:
    connect_meal_node(driver, user1_data['username'], meal1_title, recipe)
```

```
Meal node brody675Grilled Cheese w/Chili created!
Meal node brody675Grilled Cheese w/Chili connected to GrilledCheese!
Meal node brody675Grilled Cheese w/Chili connected to Chili!
```

Meal Plan

In [ ]:
```
```

Step

In [32]:
```
def create_step_node(driver, user, recipe, step_data):
    query = '''
    CREATE (step:Step {
        name: $stepName,
        description: $description,
        order: $order
    })
    '''
    step_name = user + recipe + "Step" + step_data['order']
    with driver.session() as session:
        session.run(query,
                    stepName=step_name,
                    description=step_data['description'],
                    order=step_data['order'])

    print(f"Step {step_name} created!")
```

In [33]:
```
step1_data = {
    'order': '1',
    'description': 'Preheat a nonstick skillet over medium heat. Generously bu
tter one side of a slice of bread. Place bread butter-side down in the hot ski
llet; add 1 slice of cheese. Butter a second slice of bread on one side and pl
ace butter-side up on top of cheese.'
}
step2_data = {
    'order': '2',
    'description': 'Cook until lightly browned on one side; flip over and cont
inue cooking until cheese is melted. Repeat with remaining 2 slices of bread,
butter, and slice of cheese.'
}
```

In [34]:
```
create_step_node(driver, user1_data['username'], recipe2_data['name'], step1_d
ata)
create_step_node(driver, user1_data['username'], recipe2_data['name'], step2_d
ata)
```

```
Step brody675GrilledCheeseStep1 created!
Step brody675GrilledCheeseStep2 created!
```

Tag

Node automatically connected to other nodes

In [35]:
```
def create_tag_node(driver, user, recipe, tag):
    query = '''
    MATCH (recipe:Recipe{name: $recipeName})
    CREATE (tag:Tag {
                    name: $tagName,
                    title: $tag
                    })
    CREATE (tag)-[:DESCRIBES]->(recipe)
    '''
    recipe_name = user + recipe
    tag_name = recipe_name + tag
    with driver.session() as session:
        session.run(query,
                    recipeName=recipe_name,
                    tag=tag,
                    tagName=tag_name)

    print(f"Tag {tag_name} created!")
```

In [36]:
```
create_tag_node(driver, user1_data['username'], recipe3_data['name'], "Spicy")
```

```
Tag brody675ChiliSpicy created!
```

Relationships

In [37]:
```
# query = "MATCH "
# with driver.session() as session:
# result = session.run(query)
# summary = result.consume()  # Get the summary of the execution

# # Print the number of nodes deleted
# print(f"Deleted {summary.counters.nodes_deleted} nodes from the database.")
```

Close Database Connection

In [38]:
```
driver.close()
```