Recipe Project

Gregory Billings 300277195
Scott Larsen 300240495
Darren Singh 300268703

Project Detail

On popular recipe websites there is not enough flexibility with queries so we took all the data and put it into a mongoDB format, once we had it in our mongoDB format we were able to write more flexible and extensive queries for the recipe's.
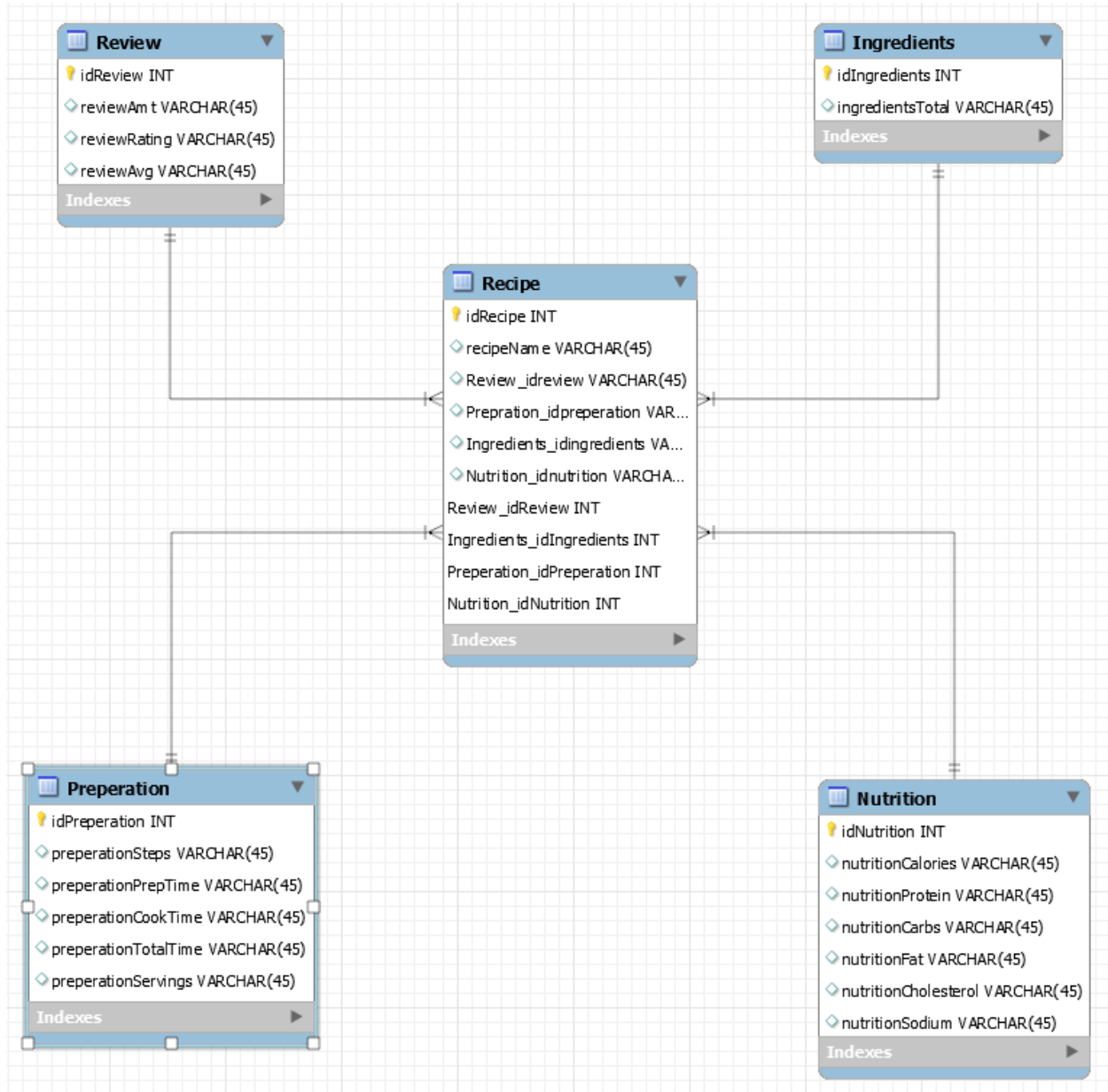
DataSet Generator

In order to generate our dataset we used a web scraper using python modules of requests, pymongo and beautiful soup 4.

Detail Structure

```
{
    "_id" : ObjectId("605b8c8902f9c27beb2acd31"),
    "name" : "Burrata Chicken Sliders",
    "ingredients" : [
    "6  thinly sliced boneless chicken breasts (or cutlets)",
    "1 pinch Salt and pepper",
    "1 tablespoon olive oil",
    "12  slider buns, split",
    "1  tomato, thinly sliced",
    "12  fresh basil leaves",
    "1 (8 ounce) package BelGioioso burrata cheese, sliced to include filling"
    ],
    "Reviews" : {
    "Review Average" : 5.0,
    "Total Ratings" : 1
    },
    "Preparation" : {
    "prep" : "20 mins",
    "cook" : "10 mins",
    "total" : "30 mins",
```

```
        "Servings" : 12,
        "Yield" : "12 sliders"
        },
        "nutritionalInfo" : {
        "Calories" : "211.3",
        "protein" : "18.6g",
        "carbohydrates" : "159.6g",
        "dietary fiber" : "0.8g",
        "sugars" : "0.5g",
        "fat" : "8.9g",
        "saturated fat" : "0.6g",
        "cholesterol" : "32.3mg",
        "vitamin a iu" : "2347IU",
        "niacin equivalents" : "8.2mg",
        "vitamin b6" : "0.3mg",
        "vitamin c" : "9.2mg",
        "folate" : "32.2mcg",
        "calcium" : "82mg",
        "iron" : "1.8mg",
        "magnesium" : "39.5mg",
        "potassium" : "251.6mg",
        "sodium" : "100mg"
        },
        "Author" : {
        "Name" : "BelGioioso Cheese"
        }
}
```

Project Schema

Review
- 🔑 idReview INT
- ◇ reviewAmt VARCHAR(45)
- ◇ reviewRating VARCHAR(45)
- ◇ reviewAvg VARCHAR(45)
- Indexes ▶

Ingredients
- 🔑 idIngredients INT
- ◇ ingredientsTotal VARCHAR(45)
- Indexes ▶

Recipe
- 🔑 idRecipe INT
- ◇ recipeName VARCHAR(45)
- ◇ Review_idreview VARCHAR(45)
- ◇ Prepration_idpreperation VAR...
- ◇ Ingredients_idingredients VA...
- ◇ Nutrition_idnutrition VARCHA...
- Review_idReview INT
- Ingredients_idIngredients INT
- Preperation_idPreperation INT
- Nutrition_idNutrition INT
- Indexes ▶

Preperation
- 🔑 idPreperation INT
- ◇ preperationSteps VARCHAR(45)
- ◇ preperationPrepTime VARCHAR(45)
- ◇ preperationCookTime VARCHAR(45)
- ◇ preperationTotalTime VARCHAR(45)
- ◇ preperationServings VARCHAR(45)
- Indexes ▶

Nutrition
- 🔑 idNutrition INT
- ◇ nutritionCalories VARCHAR(45)
- ◇ nutritionProtein VARCHAR(45)
- ◇ nutritionCarbs VARCHAR(45)
- ◇ nutritionFat VARCHAR(45)
- ◇ nutritionCholesterol VARCHAR(45)
- ◇ nutritionSodium VARCHAR(45)
- Indexes ▶

Queries

Query 1

Recipes listed by # of ingredients required, sorted in ascending order

```
db.getCollection('recipe').aggregate([
{$group:{ _id:{
    "id":"$_id",
    "name":"$name"
    },
    "ingredientCount": { "$sum": { "$size": "$ingredients" } }}},
{$sort:{ingredientCount: 1}}
]);
```

Output

```
/* 1 */
{
    "_id" : {
        "id" : ObjectId("605ab3a9b62eb3ee60aa017c"),
        "name" : "Edible Red Cabbage Bowl"
    },
    "ingredientCount" : 1
}

/* 2 */
{
    "_id" : {
        "id" : ObjectId("605ab352b62eb3ee60aa013a"),
        "name" : "Sun-dried Tomatoes II"
    },
    "ingredientCount" : 1
}

/* 3 */
{
    "_id" : {
        "id" : ObjectId("605ab3efb62eb3ee60aa01b0"),
        "name" : "Sun-dried Tomatoes II"
    },
    "ingredientCount" : 1
}
```

Query 2

Recipes with 4 or 5 star ratings

```
db.getCollection('recipe').aggregate([
{$match:{
    "Reviews.Review Average": {$in:[5,4]} }},
{$group:{ _id:{
    "id":"$_id",
    "name":"$name",
    "avgRating":"$Reviews.Review Average"
    }}},
{$sort:{avgRating:1}}
]);
```

Output

```
/* 1 */
{
    "_id" : {
        "id" : ObjectId("605ab2c3b62eb3ee60aa00bd"),
        "name" : "Un Monstruo!",
        "avgRating" : 5.0
    }
}

/* 2 */
{
    "_id" : {
        "id" : ObjectId("605ab4c9b62eb3ee60aa0269"),
        "name" : "Avocado Sushi with Brown Rice",
        "avgRating" : 5.0
    }
}

/* 3 */
{
    "_id" : {
        "id" : ObjectId("605ab2e6b62eb3ee60aa00dc"),
        "name" : "Seven-Day Wonders",
        "avgRating" : 5.0
    }
}
```

Query 3

Find all the amount of recipes in which the author uses butter

Output

```
db.recipes.mapReduce(
function(){emit(this.Author.Name, this.name)},
function(author,recipe){return recipe.length},
        {
        query: {ingredients: {$regex: ".*butter.*"}},
        out: "authorRecipes"
        }
)
db.authorRecipes.find()
```

```
* 1 */
{
        "_id" : "LaCuocaEnza",
        "value" : 2.0
}

/* 2 */
{
        "_id" : "Mellan",
        "value" : 2.0
}

/* 3 */
{
        "_id" : "Chantal Perez",
        "value" : 2.0
}

/* 4 */
{
        "_id" : "BETHRECIPES",
        "value" : 1.0
}

/* 5 */
{
        "_id" : "Adrienne Gauthier",
        "value" : 2.0
}

/* 6 */
{
        "_id" : "Marianne",
        "value" : 1.0
}
```

Query 4

Find all the recipes in which the servings are less than 5

```
db.recipes.find({"Preparation.Servings":{$lt:5}})
```

```
/* 1 */
{
    "_id" : ObjectId("605b8c8e02f9c27beb2acd36"),
    "name" : "Christmas Noels",
    "ingredients" : [
    "3 fluid ounces sparkling white wine, chilled",
    "1 fluid ounce peach-flavored vodka, chilled",
    "2 fluid ounces cranberry juice, chilled"
    ],
    "Reviews" : {
    "Review Average" : 5.0,
    "Total Ratings" : 1
    },
    "Preparation" : {
    "prep" : "5 mins",
    "total" : "5 mins",
    "Servings" : 1,
    "Yield" : "1 cocktail"
    },
    "nutritionalInfo" : {
    "Calories" : "169.5",
    "protein" : "0.1g",
    "carbohydrates" : "11g",
    "sugars" : "8.2g",
    "fat" : "0.1g",
    "vitamin a iu" : "5.1IU",
    "niacin equivalents" : "0.2mg",
    "vitamin c" : "26.7mg",
    "folate" : "0.9mcg",
    "calcium" : "9mg",
    "iron" : "0.4mg",
    "magnesium" : "10.4mg",
    "potassium" : "96.5mg",
    "sodium" : "5.7mg"
    },
    "Author" : {
    "Name" : "Bridget Cope Noel"
```

Query 5

Display list of dishes with their names and respective calories sorted in ascending order

```
db.getCollection('recipe').aggregate([
{$group:{ _id:{
    "name":"$name",
    "Calories":"$nutritionalInfo.Calories"
    }}},
{$sort:{nutritionalInfo:1}}
]);
```

```
/* 1 */
{
    "_id" : {
        "name" : "Burrata Chicken Sliders",
        "Calories" : "211.3"
    }
}
```