# CS - 532: Database Systems Final Project Report

---

**Project Title: Data-Driven Recipe Evaluation, Analysis and Complexity Ranking System**

Team Member(s):
1. Bidwai Sukrut Rahul
2. Jagdale Devang Dilip

---

## I. N+1 NOSQL QUERIES

**Query 1:**

```
 [
  {
    $lookup: {
      from: "reviews",
      localField: "RecipeId",
      foreignField: "RecipeId",
      as: "reviewData",
    },
  },
  {
    $unwind: "$reviewData",
  },
  {
    $match: {
      "reviewData.Review": {
        $regex:
          "\\bsubstitute\\b|\\breplaced\\b
|\\binstead\\b",
        $options: "i",
      },
    },
  },
  {
    $project: {
      reviewText: "$reviewData.Review",
      RecipeIngredientParts: {
        $split: [
          {
            $replaceAll: {
              input:
"$RecipeIngredientParts",
              find: '"',
              replacement: "",
            },
          },
          ", ",
        ],
      },
    },
  },
  {
    $unwind: "$RecipeIngredientParts",
  },
  {
    $project: {
      reviewText: 1,
      ingredient:
"$RecipeIngredientParts",
      reviewWords: {
        $split: ["$reviewText", " "],
      },
    },
  },
  {
    $addFields: {
      substitutions: {
        $filter: {
          input: {
            $map: {
              input: {
                $range: [
                  0,
                  {
                    $subtract: [
                      {
                        $size:
"$reviewWords",
                      },
                      1,
                    ],
                  },
                ],
              },
              as: "idx",
              in: {
                idx: "$$idx",
                word: {
                  $arrayElemAt: [
                    "$reviewWords",
                    "$$idx",
                  ],
                },
              },
            },
          },
          as: "item",
          cond: {
            $in: [
              "$$item.word",
              [
                "substitute",
                "replaced",
                "instead",
              ],
            ],
          },
        },
      },
    },
  },
  {
    $unwind: "$substitutions",
  },
  {
    $addFields: {
      before: {
        $slice: [
          "$reviewWords",
          {
            $subtract:
["$substitutions.idx", 3],
          },
          3,
        ],
      },
      after: {
        $slice: [
          "$reviewWords",
          {
            $add: ["$substitutions.idx",
1],
          },
          3,
        ],
      },
    },
  },
  {
    $lookup: {
      from: "ingredientCounts",
      localField: "before",
      foreignField: "_id",
      as: "ingredientLookupBefore",
    },
  },
  {
    $lookup: {
      from: "ingredientCounts",
      localField: "after",
      foreignField: "_id",
      as: "ingredientLookupAfter",
    },
  },
  {
    $match: {
      $and: [
        {
          ingredientLookupBefore: {
            $ne: [],
          },
        },
        {
          ingredientLookupAfter: {
            $ne: [],
          },
        },
      ],
    },
  },
  {
    $project: {
      _id: 0,
      before: "$before",
      after: "$after",
    },
  },
  {
    $lookup:
      /**
       * from: The target collection.
       * localField: The local join field.
       * foreignField: The target join
field.
       * as: The name for the results.
       * pipeline: Optional pipeline to
run on the foreign collection.
       * let: Optional variables to use in
the pipeline field stages.
       */
      {
        from: "ingredientCounts",
        localField: "before",
        foreignField: "_id",
        as: "ingredientLookupBefore",
      },
  },
  {
    $lookup:
      /**
       * from: The target collection.
       * localField: The local join field.
       * foreignField: The target join
field.
       * as: The name for the results.
       * pipeline: Optional pipeline to
run on the foreign collection.
       * let: Optional variables to use in
the pipeline field stages.
       */
      {
        from: "ingredientCounts",
        localField: "after",
        foreignField: "_id",
        as: "ingredientLookupAfter",
      },
  },
  {
    $match:
      /**
       * query: The query in MQL.
       */
      {
        $and: [
          {
            ingredientLookupBefore: {
              $ne: [],
            },
          },
          {
            ingredientLookupAfter: {
              $ne: [],
            },
          },
        ],
      },
  },
  {
    $project:
      /**
       * specifications: The fields to
       *   include or exclude.
       */
      {
        _id: 0,
        before: "$before",
        after: "$after",
      },
  },
  {
    $addFields:
      /**
       * newField: The new field name.
       * expression: The new field
expression.
       */
      {
        before: {
          $filter: {
            input: "$before",
            as: "item",
            cond: {
              $not: {
                $in: [
                  {
                    $trim: {
                      input: "$$item",
                    },
                  },
```

```
                    [
                      "might",
                      "try",
                      "(sometime",
                      "(chopped)",
                      "the",
                      "will",
                      "which",
                      "for",
                      "our",
                      "it's",
                      "a",
                      "and",
                      "I",
                      "used",
                      "use",
                      "of",
                      "an",
                      "all",
                      "with",
                      "in",
                      ":)",
                      "Tnx",
                      "to",
                      "(but",
                      "so",
                      "because",
                      "sorry",
                      ",",
                      "using",
                      "add",
                      "as",
                      "put",
                      "or",
                      "-",
                      "",
                    ],
                  ],
                },
              },
            },
          },
        },
  },
  {
    $addFields:
      /**
       * newField: The new field name.
       * expression: The new field
expression.
       */
      {
        after: {
          $filter: {
            input: "$after",
            as: "item",
            cond: {
              $not: {
```

```
            $in: [
              {
                $trim: {
                  input: "$$item",
                },
              },
              [
                "might",
                "try",
                "(sometime",
                "(chopped)",
                "the",
                "will",
                "which",
                "for",
                "our",
                "it's",
                "a",
                "and",
                "I",
                "used",
                "use",
                "of",
                "an",
                "all",
                "with",
                "in",
                ":)",
                "Tnx",
                "to",
                "(but",
                "so",
                "because",
                "sorry",
                ",",
                "using",
                "add",
                "as",
                "put",
                "or",
                "-",
                "",
              ],
            ],
          },
        },
      },
    },
  },
},
{
  $match:
    /**
     * query: The query in MQL.
     */
    {
      $and: [
        {
          $or: [
            {
              $expr: {
                $ne: [
                  {
                    $size: "$before",
                  },
                  0,
                ],
              },
            },
            {
              $expr: {
                $ne: [
                  {
                    $size: "$after",
                  },
                  0,
                ],
              },
            },
          ],
        },
      ],
    },
  },
  {
    $group:
      /**
       * specifications: The fields to
       *   include or exclude.
       */
      {
        _id: {
          before: "$before",
          after: "$after",
        },
      },
  },
  {
    $project:
      /**
       * specifications: The fields to
       *   include or exclude.
       */
      {
        _id: 0,
        before: "$_id.before",
        after: "$_id.after",
      },
  },
]
```

**Query 1 Description:**

The query aims to identify ingredient substitutions mentioned in recipe reviews. It matches reviews containing words like "substitute", "replaced", or "instead", extracts the words before and after these substitution terms, and then checks if these words are ingredients. The results are unique pairs of before and after ingredients.

Columns Used:

- RecipeId
- Review (NS)
- RecipeIngredientParts

------------------------------------------

**Query 2:**
```
db.reciepe.aggregate([
    {
      $project: {
        RecipeId: 1,
        RecipeCategory: 1,
        RecipeInstructions: 1,
        AggregatedRating: 1,
        complexity: {
          $size: {
            $filter: {
              input: { $split:
["$RecipeInstructions", ", "] },
              as: "step",
              cond: { $ne: ["$$step", ""]
}
            }
          }
        },
        uniqueActions: {
          $size: {
            $setUnion: {
              $map: {
                input: { $split:
["$RecipeInstructions", ", "] },
                as: "step",
                in: { $arrayElemAt: [{
$split: ["$$step", " "] }, 0] }
              }
            }
          }
        },
        actionWordsCount: {
          $size: {
            $filter: {
              input: { $split:
["$RecipeInstructions", ", "] },
              as: "step",
              cond: {
                $or: [
                  { $regexMatch: { input:
"$$step", regex:
/peel|add|beat|mash|mix|stir|chop|bake|coo
k|grill|fry|boil|simmer|saute|roast|blend|
whisk|knead|steam|whip|fold|top|drain|spre
ad|baste|marinate|garnish|broil|braise|deg
laze|brush|sprinkle|shake|season|dice|pour
|squeeze|grate|crush|slice|dip|flavor|fros
t|glaze|toast|puree|toss|baste|tenderize|s
hred|soak|cream|julienne|braise|caramelize
|scald|dredge|reduce|scrape|dust|sprinkle|
cure|infuse|smoke|pour|pound|fold|shuck|si
ft|drizzle|butter|coddle|blanch|char|sift|
cure|infuse|simmer|roast|broil|baste|caram
elize|glaze|stew|coddle|blanch|char|poach|
debone|cube/ } }
                ]
              }
            }
          }
        }
      }
    },
    {
      $addFields: {
        complexityScore: {
          $cond: [
            { $gte: ["$complexity", 20]
},
            {
              $min: [
                { $multiply: [
                  { $divide:
["$uniqueActions", 20] },
                  100
                ]
                },
                100
              ]
          },
          {
            $multiply: [
              { $divide:
["$uniqueActions", 20] },
              {
                $multiply:
["$complexity", 5]
              }
            ]
          }
        ]
      }
    }
  },
  {
    $group: {
      _id: "$RecipeCategory",
      totalComplexityScore: { $sum:
"$complexityScore" },
      averageRating: { $avg:
"$AggregatedRating" },
      count: { $sum: 1 }
    }
  },
  {
    $addFields: {
      averageComplexity: {
        $divide:
["$totalComplexityScore", "$count"]
      }
    }
  },
  {
    $sort: { averageComplexity: -1 }
  }
])
```

**Query 2 Description:**

The query calculates a complexity score for recipes based on the number of unique actions and overall complexity of the recipe instructions. Actions are identified using a set of keywords, and the complexity is determined by the number of steps in the recipe. The query then aggregates this information by recipe category, calculates an average complexity score, and sorts the results by this average score.

Columns Used:

- RecipeId
- RecipeCategory
- RecipeInstructions (NS)
- Ratings

-------------------------------------------

**Query 3:**

```
db.CL1.aggregate([
  {
    $project: {
      RecipeId: 1,
      Name: 1,
      RecipeCategory: 1,
      AuthorName: 1,
      ExtractedUtensils: {
        $regexFindAll: {
          input: "$RecipeInstructions",
          regex:
"(knife|spoon|pan|bowl|pot|whisk|ladle|gra
ter|peeler|sieve|tongs)",
          options: "i"
        }
      }
    }
  },
  {
    $unwind: "$ExtractedUtensils"
  },
  {
    $addFields: {
      WashTimeMapping: {
        $switch: {
          branches: [
            { case: { $eq:
["$ExtractedUtensils.match", "knife"] },
then: 30 },
              { case: { $eq:
["$ExtractedUtensils.match", "spoon"] },
then: 20 },
              { case: { $eq:
["$ExtractedUtensils.match", "pan"] },
then: 60 },
              { case: { $eq:
["$ExtractedUtensils.match", "bowl"] },
then: 45 },
              { case: { $eq:
["$ExtractedUtensils.match", "pot"] },
then: 50 },
              { case: { $eq:
["$ExtractedUtensils.match", "whisk"] },
then: 25 },
              { case: { $eq:
["$ExtractedUtensils.match", "ladle"] },
then: 35 },
              { case: { $eq:
["$ExtractedUtensils.match", "grater"] },
then: 40 },
              { case: { $eq:
["$ExtractedUtensils.match", "peeler"] },
then: 15 },
              { case: { $eq:
["$ExtractedUtensils.match", "sieve"] },
then: 30 },
              { case: { $eq:
["$ExtractedUtensils.match", "tongs"] },
then: 30 },
            ],
            default: 25
          }
        }
      }
    },
    {
      $group: {
        _id: { RecipeId: "$RecipeId",
Name: "$Name" },
        RecipeCategory: { $first:
"$RecipeCategory" },
        AuthorName: { $first:
"$AuthorName" },
        TotalWashTime: { $sum:
"$WashTimeMapping" },
        UniqueUtensils: { $addToSet:
"$ExtractedUtensils.match" }
      }
    },
    {
      $addFields: {
        RecipeLevel: {
          $switch: {
            branches: [
              { case: { $gt:
["$TotalWashTime", 100] }, then: "Highly
Non-Trivial" },
              { case: { $gt:
["$TotalWashTime", 80] }, then:
"Moderately Non-Trivial" },
              { case: { $gt:
["$TotalWashTime", 60] }, then: "Slightly
Non-Trivial" },
              { case: { $lte:
["$TotalWashTime", 60] }, then: "Trivial"
}
            ],
            default: "Trivial"
          }
        }
      }
    },
    {
      $group: {
        _id: { RecipeCategory:
"$RecipeCategory", AuthorName:
"$AuthorName" },
        Recipes: {
          $push: {
            RecipeId: "$_id.RecipeId",
            Name: "$_id.Name",
            TotalWashTime:
"$TotalWashTime",
            UniqueUtensils:
"$UniqueUtensils",
            RecipeLevel: "$RecipeLevel"
          }
        },
        MaxWashTime: { $max:
"$TotalWashTime" }
      }
    },
    {
      $project: {
        _id: 0,
        RecipeCategory:
"$_id.RecipeCategory",
        AuthorName: "$_id.AuthorName",
        Recipes: 1,
        MaxWashTime: 1
      }
    }
])
```

**Query 3 Description:**
This aggregation pipeline query processes recipe data. It extracts utensils mentioned in recipe instructions, calculates washing time based on the utensils, groups recipes by category and author, and evaluates the complexity of washing utensils for each recipe. The result provides a breakdown of recipes by category and author, along with washing time information and the highest washing time among recipes.

Columns Used:

- AuthorName
- RecipeCategory
- RecipeInstructions (NS)

## II. NoSQL Database and Dataset

NoSQL Database to be used: MongoDB

Link:
https://www.kaggle.com/datasets/irkaal/foodcom-recipes-and-reviews

**Dataset Description**

- Tabular Data
  - o The recipes dataset contains 522,517 recipes from 312 different
  - o categories. This dataset provides information about each recipe
  - o like cooking times, servings, ingredients, nutrition,
  - o instructions, & more.
  - o • The reviews dataset contains 1,401,982 reviews from 271,907
  - o different users. This dataset provides information about the
  - o author, rating, review text, and more.
  - o Example: RecipeID, Name, AuthorID, Category, Rating, etc.

- **Non-Tabular Data**
  - o Review
  - o RecipeInstructions
  - o RecipeIngredientParts
  - o Description

## III. Project Outcome

Query 1 results:

| | | | |
|---|---|---|---|
| **white** | **wine** | **wine** | **vinegar,** |
| **cream** | **cheese** | **frosting** | **chocolate** |
| **whole** | **milk** | **coffee** | **creamer** |
| **whole** | **eggs** | **egg** | **whites.** |
| **mustard** | **seeds** | **dry** | **mustard** |
| **fresh** | **garlic** | **garlic** | **powder.** |
| **green** | **onion** | **red** | **onion** |
| **old fashioned** | **oats** | **quick** | **oats** |

Fig 1: The above figure describes the best possible substitutions for the primary ingredient where columns 1 and 2 are primary ingredients and column 3,4 are substitution ingredients

Query 2 results:

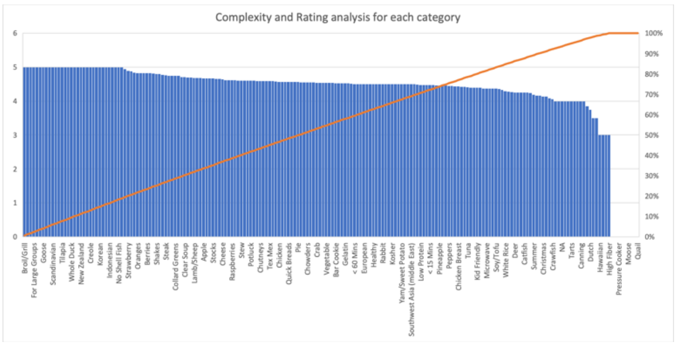
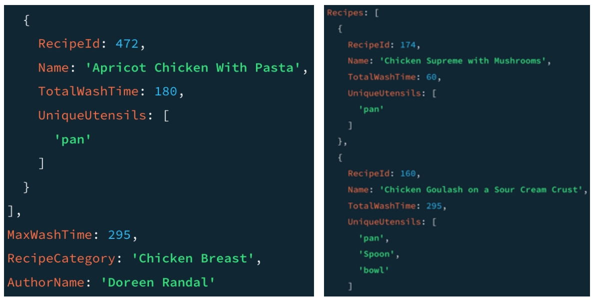


Fig 2: Above graph is a plot for Average complexity and average rating for each of the recipe category. Bar graphs represent the average ratings of each category and orange line graph represents the average complexity. This graph shows as rating decreases complexity increases which is true in most of the cases.

Query 3 results:



```
  {
    RecipeId: 472,
    Name: 'Apricot Chicken With Pasta',
    TotalWashTime: 180,
    UniqueUtensils: [
      'pan'
    ]
  }
],
MaxWashTime: 295,
RecipeCategory: 'Chicken Breast',
AuthorName: 'Doreen Randal'
```

```
Recipes: [
  {
    RecipeId: 174,
    Name: 'Chicken Supreme with Mushrooms',
    TotalWashTime: 60,
    UniqueUtensils: [
      'pan'
    ]
  },
  {
    RecipeId: 160,
    Name: 'Chicken Goulash on a Sour Cream Crust',
    TotalWashTime: 295,
    UniqueUtensils: [
      'pan',
      'Spoon',
      'bowl'
    ]
```

Fig 3:

The results show various recipes categorized by RecipeCategory and authored by different individuals. Each recipe includes its RecipeId, Name, TotalWashTime (time to clean utensils), UniqueUtensils used, and the maximum wash time (MaxWashTime) among all recipes in that category.

More results are provided in the following google docs document for reference. Link[5]

## References

[1] https://www.mongodb.com/docs/compass/current

[2] https://www.mongodb.com/docs/compass/current/documents/

[3] https://www.mongodb.com/docs/compass/current/collections/

[4] https://github.com/mongodb/docs-compass

[5] https://docs.google.com/document/d/17ZrVcvH5Z3Lisv1vyg7-eKW5vxyoGe_LTBGku-yhz50/edit?usp=sharing