Team : Maria Jose Siles Navarro, Carl Tondo

Executive Summary:

The objective of this project is to gather datasets of user interactions and recipe data from Food.com (made up with 18 years worth of data). Using the datasets of recipes and user interaction, we are able to develop queries that show different user and recipe interactions. As an example, we are able to display the highest rated recipe, or the recipe with the most user interaction (rating/review).

Data Sources :

We gathered our data from Kaggle.com, specifically data sources made by Bodhisattwa Prasad Majumder and Shuyang Li. Using these datasets, they provided us with 18 years worth of data collected from Food.com. However, we didn't need to analyze that huge amount of data, so we cleaned the data on Python using the pandas package. We cleaned the data by only receiving rows that fit within a three-month range that we chose to analyze.

Food.com Data Source Link

Data Dictionary:

Field Name	Data Type	Description	Example
user_id	bigint	The user's number ID	2002372706
recipe_id	bigint	The recipe's number ID	63786
date	date	The date of user interaction with recipes.	2018-12-20
rating	int	The user rating of a	5

		recipe.	
review	text	A user review of a recipe.	Finally, I found…
name	varchar	name of the recipe	baked shrimp
minutes	int	how long a recipe takes to make	45
contributor_id	bigint	ID of the user that posted the recipe	33186
submitted	date	date of when recipe was submitted	2018-11-12
tags	varchar	recipe tags to improve search and filters	’60-min-or-less’}
nutrition	varchar	basic nutrition facts of recipe	[52.8, 60.9, …]
n_steps	int	amount of steps a recipe needs to be made	21
steps	varchar	Actual descriptions of steps that a person needs to follow	[‘1’ preheat, …]
description	varchar	quick description of what the recipe is	delicious
ingredients	varchar	the ingredients needed	[‘pillsbury sugar’, …]

		for the recipe	
n_ingredients	int	The amount of ingredients needed for a recipe	5

Data Cleaning:

```
In [32]: mask = (df_interaction['date'] >= '9/20/2018') & (df_interaction['date'] <= '12/20/2018')
         df_daterange = df_interaction.loc[mask]
         df_daterange.head()
```

Out[32]:

	user_id	recipe_id	date	rating	review
982325	2002372706	63786	2018-12-20	5	Finally I found a no nonsense fajita recipe
584485	2001402443	271337	2018-12-19	3	These took quite a long time to make and thoug...
953118	2002371627	153647	2018-12-19	0	Best thing about this recipe? I didn't have to...
480595	2002371755	393600	2018-12-19	5	Healthiest and tastiest by far. I do a few twe...
809003	2002328086	28148	2018-12-19	4	These were good but I tweaked them to give the...

This is the first part of our data cleaning where we needed to create a date range so we didn't need to handle an incredible amount of data. This allows us to analyze data within a three-month period for user interactions.

```
In [44]: mask = (df_recipes['submitted'] >= '9/20/2018') & (df_recipes['submitted'] <= '12/20/2018')
         df_daterangerecipe = df_recipes.loc[mask]
         df_daterangerecipe.head(100)
```

Out[44]:

	name	id	minutes	contributor_id	submitted	...	n_steps	steps	description	ingredients	n_ingredients
1547	5 ingredient salted caramel crumble bars	537485	45	2000378667	2018-11-12	...	21	['1', 'heat oven to 350f spray 8-inch square p...	delicious	['pillsbury sugar cookie dough', 'caramel topp...	5
13104	bailey s chocotini	537459	10	400708	2018-11-10	...	5	['to layer: add chocolate liqueur to glass', '...	a recipe that recipe complements the cocoa in ...	['baileys irish cream', 'chocolate liqueur', '...	3
15158	baked shrimp and orzo with chickpeas lemon a...	537076	15	2002285039	2018-10-02	...	18	['preheat oven to 450 degrees', 'dry shrimp wi...	shrimp and orzo make for a simple and flavorfu...	['jumbo shrimp', 'salt & freshly ground black ...	15
34404	campbell s mini green bean casseroles	537323	40	33186	2018-10-25	...	11	['heat the oven to 375&deg', 'f spray 16 muffi...	recipe courtesy of campbell's: "here's a fabul...	['cut green beans', "campbell's cream of mushr...	6

This is the second part of our data cleaning where we do the same thing of creating a date range, but for a different table that includes more descriptions about the recipe.

```
In [22]: ttl_recipe_interact = df_daterange.loc[:,"recipe_id"].value_counts()
         ttl_recipe_interact
```

```
Out[22]: 2886      41
         80156     22
         60350     15
         101954    13
         339453    13
         38298     13
         39087     12
         99476     12
         22782     10
         69173     10
```

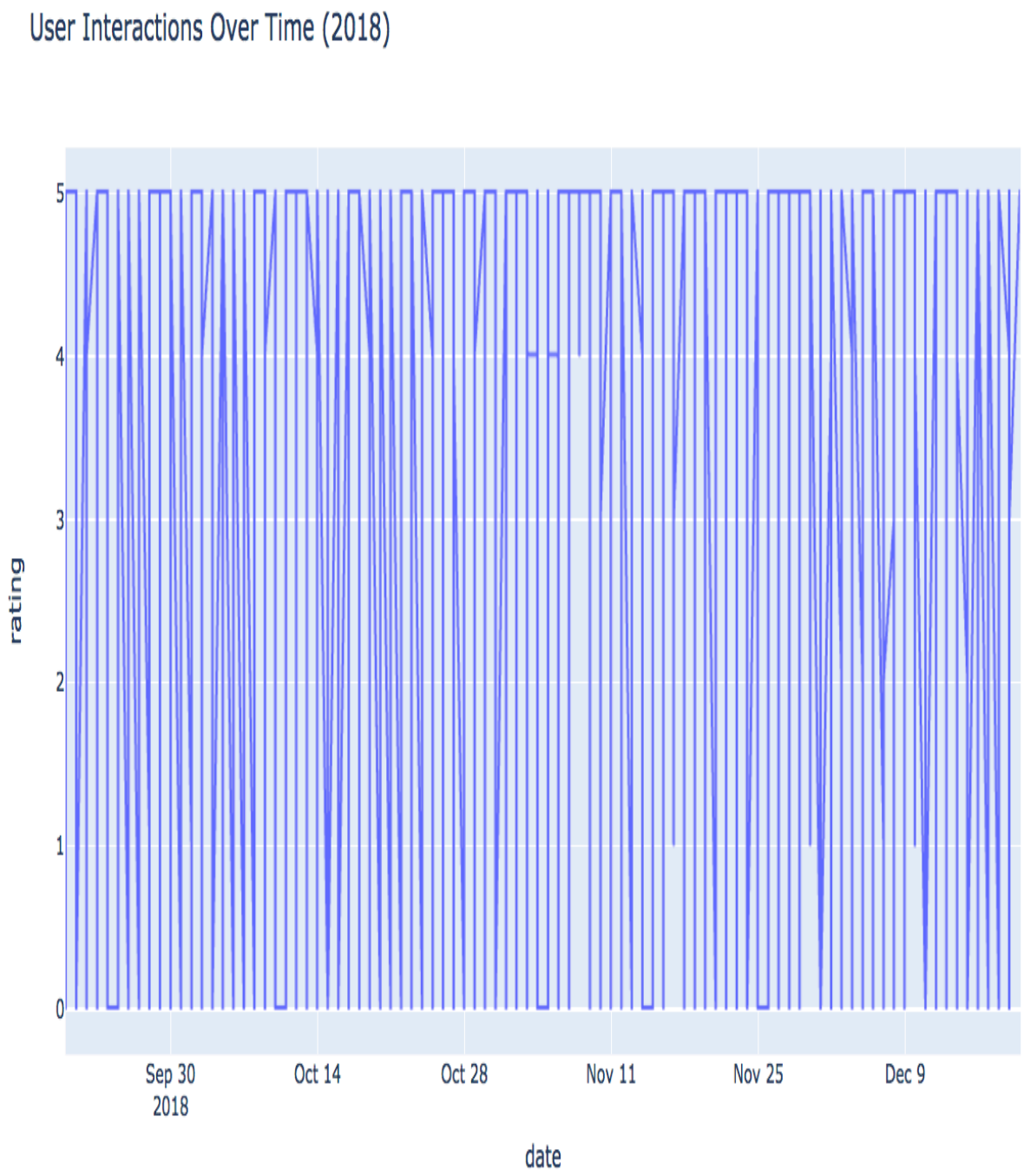
This python script gives us a general overview of recipes with the most user interactions within that three-month period.

```
In [24]: ttl_interact_perday = df_daterange.loc[:,"date"].value_counts()
         ttl_interact_perday
```

```
Out[24]: 2018-10-28    76
         2018-09-23    64
         2018-10-21    63
         2018-09-22    62
         2018-10-23    60
         2018-10-27    58
         2018-10-01    57
         2018-09-30    56
         2018-10-08    55
         2018-10-22    53
```

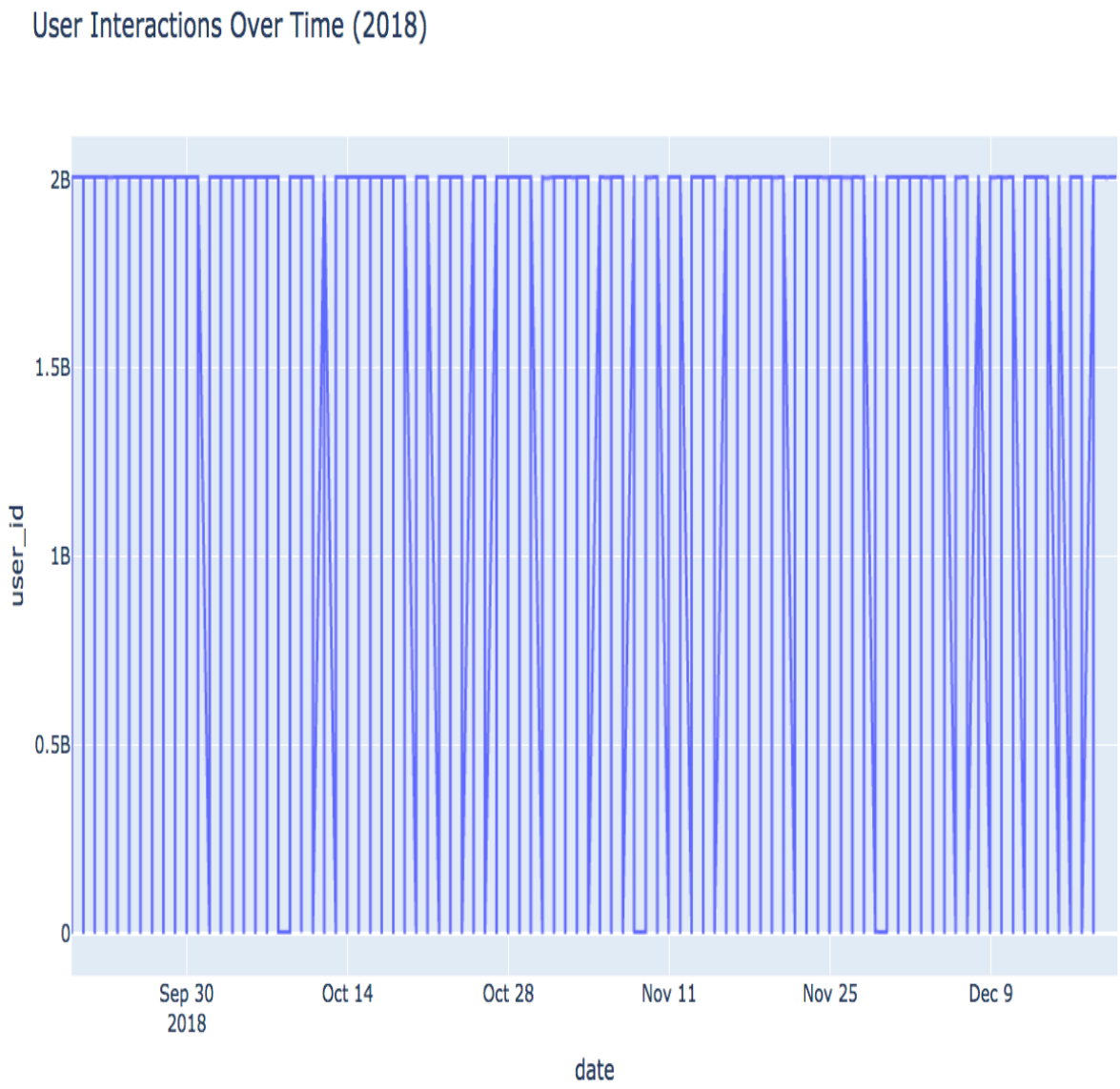
This allows us to see which date within the three-month period has the most user interactions.

Graphs:

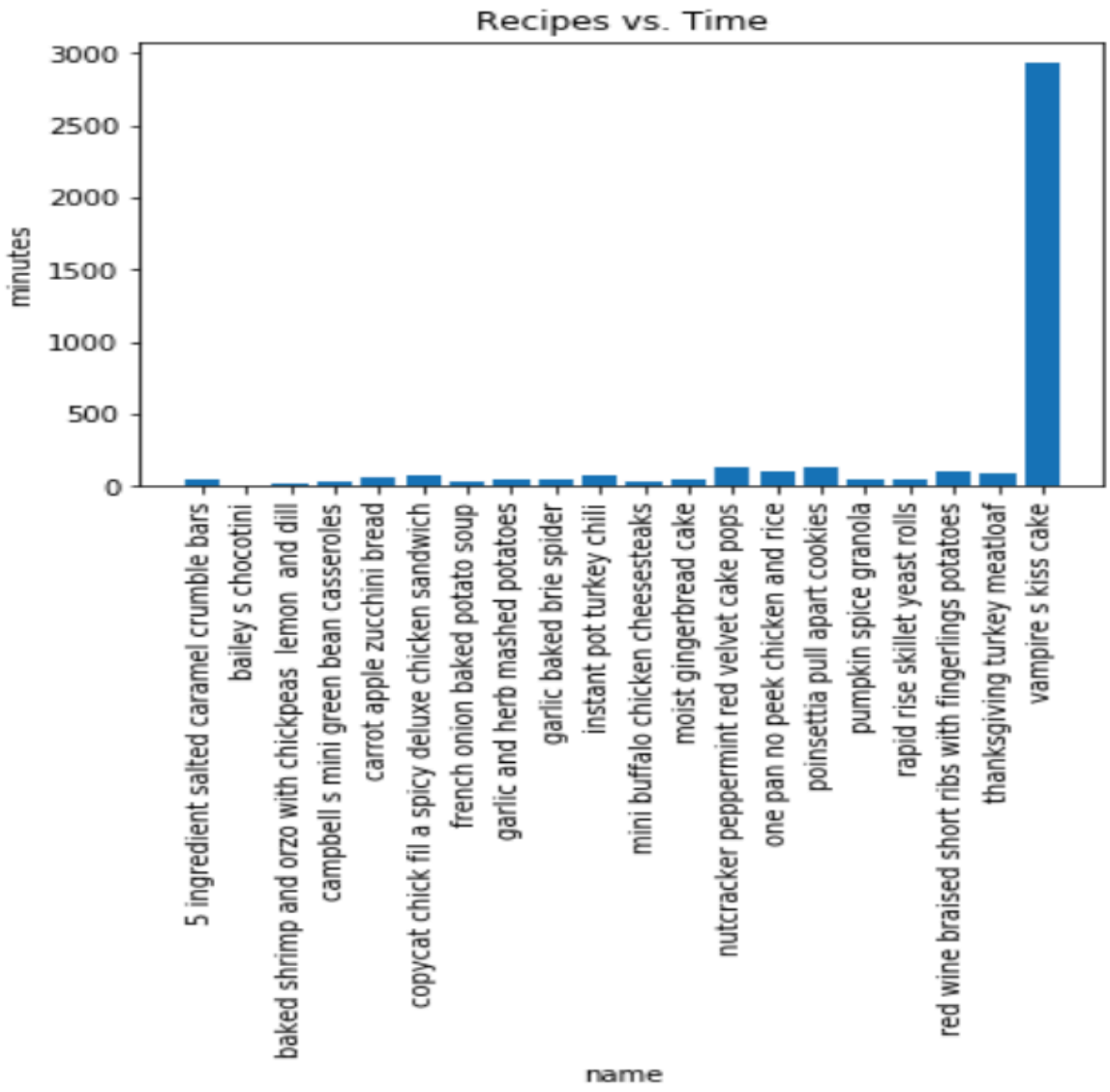


This graph demonstrates the number of ratings submitted by users within a three-month period.

We can see that between October 28 and November 11 has the least amount of user interaction.



This shows us the activity of users, by analyzing when each user interacts with a recipe through either a review or a rating.



This displays the submitted recipes and their respective cooking times. This allows us to see which recipes take the longest or the shortest.

ERD Diagram:

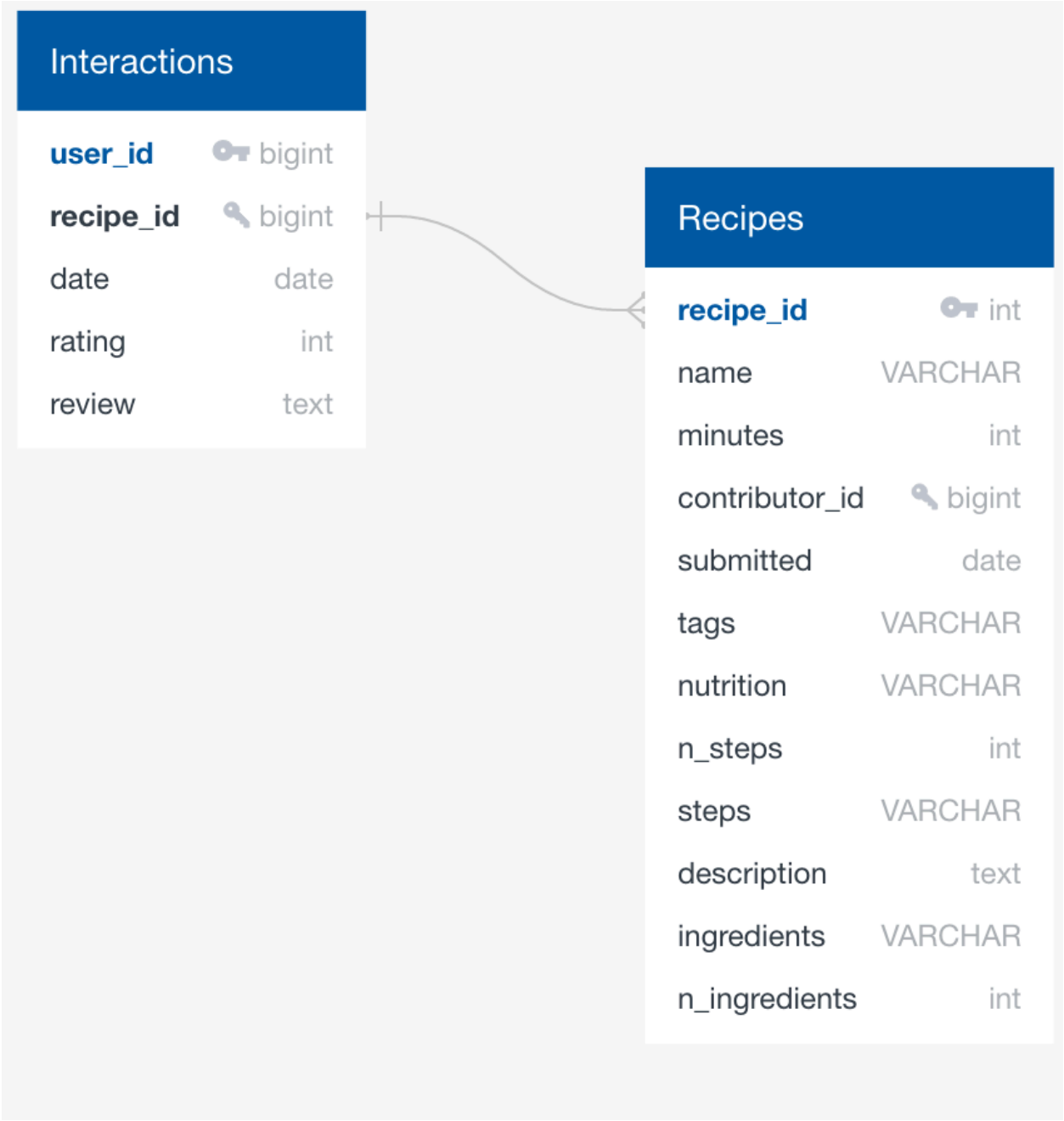

Interactions
user_id bigint
recipe_id bigint
date date
rating int
review text
Recipes
recipe_id int
name VARCHAR
minutes int
contributor_id bigint
submitted date
tags VARCHAR
nutrition VARCHAR
n_steps int
steps VARCHAR
description text
ingredients VARCHAR
n_ingredients int

Table Schema:

```sql
SELECT       recipe_id,
             COUNT(recipe_id) AS value_occurrence
  FROM       interactions
  GROUP BY recipe_id
  ORDER BY value_occurrence DESC
  LIMIT      5;

SELECT       user_id,
             COUNT(user_id) AS value_occurrence
  FROM       interactions
  GROUP BY user_id
  ORDER BY value_occurrence DESC
  LIMIT      5;

SELECT       recipe_id, rating,
            COUNT(rating) AS most_rating
  FROM       interactions
  GROUP BY recipe_id, rating
  ORDER BY most_rating DESC
  LIMIT      5;

SELECT rnames, rid, MIN(minutes)
 FROM recipes
 GROUP BY rnames, rid
 ORDER BY min ASC;

SELECT rnames, rid, MIN(n_steps)
 FROM recipes
 GROUP BY rnames, rid
 ORDER BY min ASC;
```

Queries:

INPUT:

```
SELECT        recipe_id,
              COUNT(recipe_id) AS value_occurrence
    FROM      interactions
    GROUP BY  recipe_id
    ORDER BY  value_occurrence DESC
    LIMIT     5;
```

OUTPUT:

	recipe_id bigint	value_occurrence bigint
1	2886	41
2	80156	22
3	60350	15
4	101954	13
5	38298	13

1. This shows us the top recipes with the most user interactions (that includes both reviews and ratings).

INPUT:

```
SELECT        user_id,
              COUNT(user_id) AS value_occurrence
    FROM      interactions
    GROUP BY  user_id
    ORDER BY  value_occurrence DESC
    LIMIT     5;
```

OUTPUT:

	user_id bigint	value_occurrence bigint
1	2000498330	25
2	198154	21
3	2123645	19
4	400708	19
5	305531	19

2. This shows us the most active users by tracking the users with the most interactions to a recipe.

INPUT:

```
SELECT        recipe_id, rating,
              COUNT(rating) AS most_rating
  FROM        interactions
  GROUP BY    recipe_id, rating
  ORDER BY    most_rating DESC
  LIMIT       5;
```

OUTPUT:

	recipe_id bigint	rating integer	most_rating bigint
1	2886	5	33
2	101954	5	13
3	80156	5	12
4	38298	5	11
5	339453	5	10

3. This shows us the recipes with the most ratings and the highest value ratings from users.

INPUT:

```
SELECT rnames, rid, MIN(minutes
FROM recipes
GROUP BY rnames, rid
ORDER BY min ASC;
```

OUTPUT:

	rnames text	rid bigint	min integer
1	bailey s cho...	537459	10
2	baked shri...	537076	15
3	french onio...	537071	35
4	mini buffalo...	537716	40
5	campbell s ...	537323	40
6	garlic and h...	537458	42
7	5 ingredient...	537485	45

4. This shows us the recipes that can be made at the least amount of time for those that want quick under an hour of cooking.

INPUT:

```
SELECT rnames, rid, MIN(n_steps)
FROM recipes
GROUP BY rnames, rid
ORDER BY min ASC;
```

OUTPUT:

	rnames text	rid bigint	min integer
1	bailey s cho...	537459	5
2	one pan no ...	537351	7
3	moist ginge...	537543	8
4	pumpkin sp...	537319	10
5	thanksgivin...	537039	10
6	campbell s ...	537323	11
7	mini buffalo...	537716	12

5. This shows us recipes that are simple to make for beginner due to their low number of steps.