1. Data Integrity and Validation

Each table in the database was designed with integrity constraints to enforce data accuracy and consistency throughout the population process.

Primary key constraints were applied to every table to uniquely identify each record. Tables such as Users, Recipes, Ingredients, Tags, Favorites, Inventory, and Grocery_List each use a single auto-incrementing integer as the primary key. Junction tables, including Recipe_Ingredients and Recipe_Tags, use composite primary keys composed of two foreign key columns, which prevent duplicate pairings between recipes and ingredients or recipes and tags.

Not null constraints were applied to attributes that are essential for a record to be meaningful. For example, recipe_name in the Recipes table, ingredient_name in the Ingredients table, entity_type in the Nutritional_Values table, is_purchased in the Grocery_List table, and both name and password in the Users table are all required fields. Without these values, a record would be incomplete and unusable by the application.

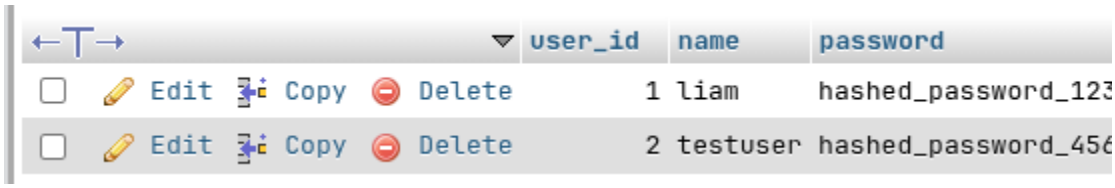
Unique constraints were added to prevent duplicate values that would cause logical errors. The ingredient_name column in the Ingredients table is marked as unique so each ingredient cannot be entered twice under different IDs. The recipe_id column in the Favorites table is also marked as unique because each recipe should be favorited only once. The name column in the Users table is marked as unique to ensure no two users share the same username.

Check constraints were used to restrict the range of acceptable values for numeric fields. The quantity column in Recipe_Ingredients and Grocery_List requires values greater than zero. The quantity column in Inventory requires values greater than or equal to zero to allow for items that have been fully used but not yet removed. The rating column in Favorites is restricted to integers between 1 and 5. All nutritional decimal columns require non-negative values.

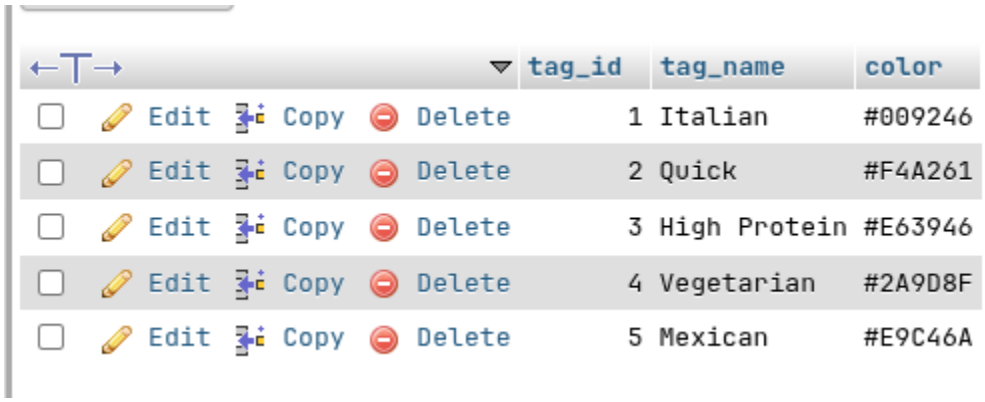
Foreign key constraints were applied to maintain referential integrity across related tables. Recipe_Ingredients references both Recipes and Ingredients. Favorites and Recipe_Tags both reference Recipes. Recipe_Tags also references Tags. Inventory and Grocery_List both reference Ingredients. Nutritional_Values references either Ingredients or Recipes, depending on the entity type. Favorites, Inventory, and Grocery_List each reference Users through a user_id foreign key, ensuring that all personal data belongs to a valid user account. These constraints ensure that no record can reference a nonexistent parent record.

During data insertion, values were validated against these constraints before being submitted. Mandatory fields were always provided, quantity values were checked to confirm they fell within the allowed range, rating values were kept between 1 and 5, and foreign key values were confirmed to exist in their referenced tables before insertion. Insert statements were also executed in the correct order, starting with the Users table first, followed by parent tables such as Recipes and Ingredients, before populating dependent tables such as Recipe_Ingredients, Favorites, Inventory, and Grocery_List, to avoid foreign key violations.

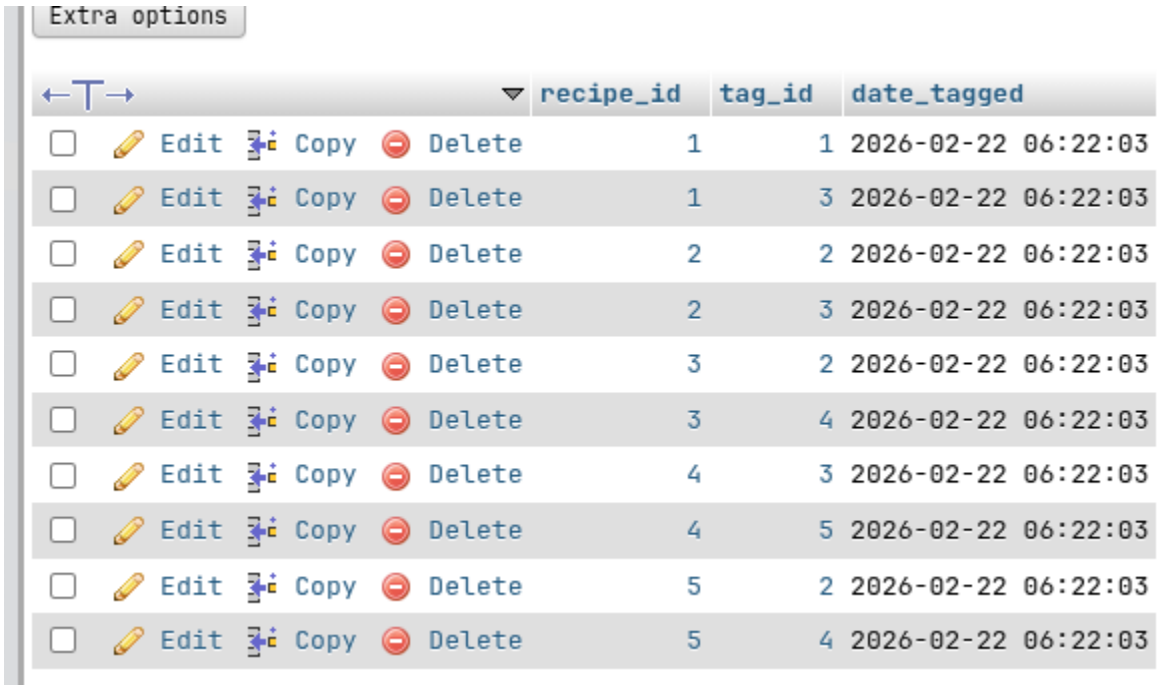
2. Screenshots



	user_id	name	password
Edit Copy Delete	1	liam	hashed_password_123
Edit Copy Delete	2	testuser	hashed_password_456



	tag_id	tag_name	color
Edit Copy Delete	1	Italian	#009246
Edit Copy Delete	2	Quick	#F4A261
Edit Copy Delete	3	High Protein	#E63946
Edit Copy Delete	4	Vegetarian	#2A9D8F
Edit Copy Delete	5	Mexican	#E9C46A



Extra options

	recipe_id	tag_id	date_tagged
Edit Copy Delete	1	1	2026-02-22 06:22:03
Edit Copy Delete	1	3	2026-02-22 06:22:03
Edit Copy Delete	2	2	2026-02-22 06:22:03
Edit Copy Delete	2	3	2026-02-22 06:22:03
Edit Copy Delete	3	2	2026-02-22 06:22:03
Edit Copy Delete	3	4	2026-02-22 06:22:03
Edit Copy Delete	4	3	2026-02-22 06:22:03
Edit Copy Delete	4	5	2026-02-22 06:22:03
Edit Copy Delete	5	2	2026-02-22 06:22:03
Edit Copy Delete	5	4	2026-02-22 06:22:03

	recipe_id	ingredient_id	quantity	unit
Edit Copy Delete	1	1	200.00	grams
Edit Copy Delete	1	2	3.00	count
Edit Copy Delete	1	3	100.00	grams
Edit Copy Delete	1	4	50.00	grams
Edit Copy Delete	2	5	300.00	grams
Edit Copy Delete	2	6	2.00	count
Edit Copy Delete	2	7	30.00	ml
Edit Copy Delete	3	2	2.00	count
Edit Copy Delete	3	8	1.00	count
Edit Copy Delete	3	9	2.00	slices
Edit Copy Delete	4	10	250.00	grams
Edit Copy Delete	4	11	3.00	count
Edit Copy Delete	5	12	150.00	grams
Edit Copy Delete	5	13	60.00	grams
Edit Copy Delete	5	14	20.00	ml

Check all With selected: Edit Copy Delete Export

	recipe_id	recipe_name	instructions	image_url	source_api	last_fetched	is_favorited	cache_priority
Edit Copy Delete	1	Spaghetti Carbonara	Boil pasta. Fry pancetta. Mix eggs and cheese. Com...	https://img.example.com/carbonara.jpg	Spoonacular	2026-02-22 06:22:03	1	favorite
Edit Copy Delete	2	Chicken Stir Fry	Chop vegetables. Cook chicken. Add sauce and stir ...	https://img.example.com/stirfry.jpg	Spoonacular	2026-02-22 06:22:03	0	recently_used
Edit Copy Delete	3	Avocado Toast	Toast bread. Mash avocado. Season and top with egg...	https://img.example.com/avotoast.jpg	Edamam	2026-02-22 06:22:03	0	temporary
Edit Copy Delete	4	Beef Tacos	Brown beef. Season with spices. Fill tortillas wit...	https://img.example.com/tacos.jpg	Spoonacular	2026-02-22 06:22:03	1	favorite
Edit Copy Delete	5	Greek Salad	Chop vegetables. Add olives and feta. Dress with o...	https://img.example.com/greeksalad.jpg	Edamam	2026-02-22 06:22:03	0	recently_used

Check all With selected: Edit Copy Delete Export

	nutrition_id	ingredient_id	recipe_id	entity_type	calories	protein	carbs	sugar	fat
Edit Copy Delete	1	1	NULL	ingredient	3.50	0.13	0.71	0.00	0.01
Edit Copy Delete	2	2	NULL	ingredient	1.55	0.13	0.01	0.00	0.11
Edit Copy Delete	3	4	NULL	ingredient	4.00	0.36	0.00	0.00	0.29
Edit Copy Delete	4	5	NULL	ingredient	1.65	0.31	0.00	0.00	0.04
Edit Copy Delete	5	8	NULL	ingredient	1.60	0.02	0.09	0.00	0.15
Edit Copy Delete	6	10	NULL	ingredient	2.50	0.26	0.00	0.00	0.15
Edit Copy Delete	7	NULL	1	recipe	650.00	35.00	60.00	2.00	25.00
Edit Copy Delete	8	NULL	2	recipe	420.00	48.00	15.00	5.00	12.00
Edit Copy Delete	9	NULL	3	recipe	310.00	14.00	28.00	2.00	18.00
Edit Copy Delete	10	NULL	4	recipe	580.00	38.00	30.00	3.00	28.00
Edit Copy Delete	11	NULL	5	recipe	220.00	8.00	12.00	4.00	16.00

Check all *With selected:* Edit Copy Delete Export

	inventory_id	ingredient_id	quantity	unit	expiration_date	date_opened	user_id
Edit Copy Delete	1	1	500.00	grams	2025-12-01	NULL	1
Edit Copy Delete	2	2	12.00	count	2025-03-15	2025-03-01	1
Edit Copy Delete	3	4	200.00	grams	2025-04-01	2025-03-01	1
Edit Copy Delete	4	14	500.00	ml	2026-01-01	2025-02-01	1
Edit Copy Delete	5	7	300.00	ml	2025-11-01	2025-01-15	1

Check all *With selected:* Edit Copy Delete Export

	ingredient_id	ingredient_name	default_unit
Edit Copy Delete	1	Spaghetti	grams
Edit Copy Delete	2	Eggs	count
Edit Copy Delete	3	Pancetta	grams
Edit Copy Delete	4	Parmesan Cheese	grams
Edit Copy Delete	5	Chicken Breast	grams
Edit Copy Delete	6	Bell Pepper	count
Edit Copy Delete	7	Soy Sauce	ml
Edit Copy Delete	8	Avocado	count
Edit Copy Delete	9	Sourdough Bread	slices
Edit Copy Delete	10	Ground Beef	grams
Edit Copy Delete	11	Tortillas	count
Edit Copy Delete	12	Romaine Lettuce	grams
Edit Copy Delete	13	Feta Cheese	grams
Edit Copy Delete	14	Olive Oil	ml

Check all *With selected:* Edit Copy Delete Export

	list_id	ingredient_id	quantity	unit	is_purchased	user_id
Edit Copy Delete	1	5	400.00	grams	0	1
Edit Copy Delete	2	10	500.00	grams	0	1
Edit Copy Delete	3	11	6.00	count	1	1
Edit Copy Delete	4	8	3.00	count	0	1
Edit Copy Delete	5	6	2.00	count	1	1

Check all With selected: Edit Copy Delete Export

Extra options

	favorite_id	recipe_id	date_added	rating	notes	user_id
Edit Copy Delete	1	1	2025-01-10 18:30:00	5	Family favorite, make every Sunday.	1
Edit Copy Delete	2	4	2025-02-14 12:00:00	4	Great for taco night. Add jalapeños next time.	1

Check all With selected: Edit Copy Delete Export

3. Five SQL queries were written to extract meaningful insights from the database. Each query serves a distinct purpose and uses different SQL clauses and operators including:

A query leveraging SELECT, FROM, and WHERE clauses with comparison and logical operators to filter rows based on specific conditions.

```
SELECT r.recipe_name, f.rating, f.notes

FROM Favorites f

JOIN Recipes r ON f.recipe_id = r.recipe_id

WHERE f.rating >= 4 AND r.is_favorited = TRUE;
```

A query employing arithmetic operators (+, -, *, /) to perform calculations within the queries.

```
SELECT r.recipe_name,
    SUM(nv.calories * ri.quantity) AS total_calories,
    SUM(nv.protein * ri.quantity)  AS total_protein_g
FROM Recipes r
JOIN Recipe_Ingredients ri ON r.recipe_id = ri.recipe_id
JOIN Nutritional_Values nv ON ri.ingredient_id = nv.ingredient_id
WHERE nv.entity_type = 'ingredient'
GROUP BY r.recipe_name;
```

A query performing join expressions to combine data from multiple tables.

```
SELECT r.recipe_name, t.tag_name, t.color
FROM Recipes r
INNER JOIN Recipe_Tags rt ON r.recipe_id = rt.recipe_id
INNER JOIN Tags t ON rt.tag_id = t.tag_id
ORDER BY r.recipe_name;
```

A query using the GROUP BY clause and applying aggregate functions to perform group-level calculations.

```
SELECT i.ingredient_name,
    COUNT(ri.recipe_id) AS recipe_count,
    AVG(ri.quantity)    AS avg_quantity_used
FROM Ingredients i
JOIN Recipe_Ingredients ri ON i.ingredient_id = ri.ingredient_id
GROUP BY i.ingredient_name
ORDER BY recipe_count DESC;
```

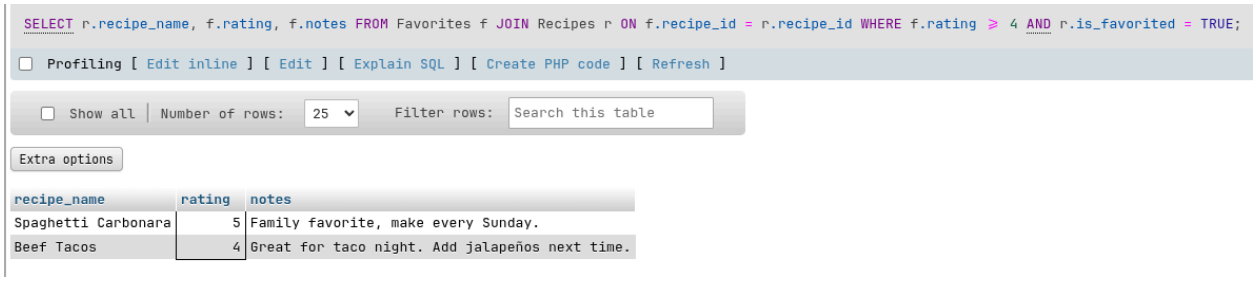
A query using the GROUP BY clause and applying aggregate functions to perform group-level calculations.

```
SELECT i.ingredient_name, COUNT(ri.recipe_id) AS recipe_count
FROM Ingredients i
JOIN Recipe_Ingredients ri ON i.ingredient_id = ri.ingredient_id
GROUP BY i.ingredient_name
HAVING COUNT(ri.recipe_id) > 1;
```

4. The results of each SQL query are included, showcasing the insights extracted from the database. Screenshots of the tables displaying the query results are provided.

Query 1

Query 1 joins recipe names from the Recipes table with ratings and notes from the Favorites table, then filters the results to only show recipes that have a rating of 4 or higher and are marked as a favorite. The purpose of this query is to quickly identify the top-performing recipes in the database.



```
SELECT r.recipe_name, f.rating, f.notes FROM Favorites f JOIN Recipes r ON f.recipe_id = r.recipe_id WHERE f.rating ⩾ 4 AND r.is_favorited = TRUE;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

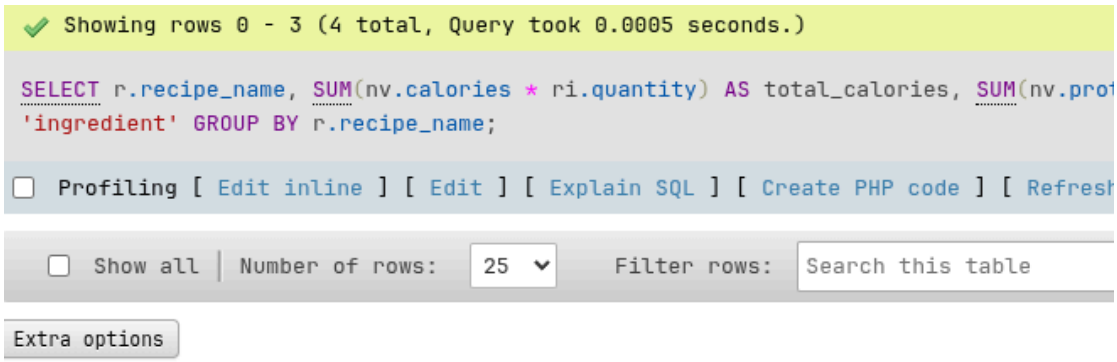
Show all | Number of rows: 25 Filter rows: Search this table

Extra options

recipe_name	rating	notes
Spaghetti Carbonara	5	Family favorite, make every Sunday.
Beef Tacos	4	Great for taco night. Add jalapeños next time.

Query 2

Query 2 joins recipes with ingredients, calories, and protein. Each protein and calorie value is multiplied by the amount of ingredients. The purpose of this query is to know the overall caloric and protein intake of a recipe.



Showing rows 0 - 3 (4 total, Query took 0.0005 seconds.)

```
SELECT r.recipe_name, SUM(nv.calories * ri.quantity) AS total_calories, SUM(nv.prot
'ingredient' GROUP BY r.recipe_name;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh

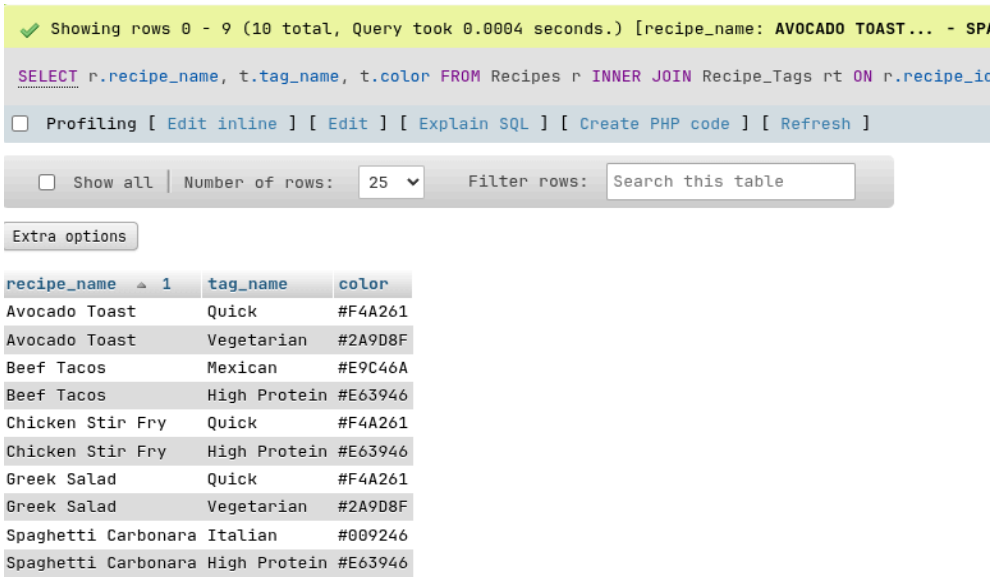
Show all | Number of rows: 25 Filter rows: Search this table

Extra options

recipe_name	total_calories	total_protein_g
Avocado Toast	4.7000	0.2800
Beef Tacos	625.0000	65.0000
Chicken Stir Fry	495.0000	93.0000
Spaghetti Carbonara	904.6500	44.3900

Query 3

Query 3 shows the recipe name from recipes, along with the tag name and color, joins them so the displayed information includes everything requested, and then orders it alphabetically. The purpose of this query is to help categorize recipes and for a new user to understand how the system organizes them.



Showing rows 0 - 9 (10 total, Query took 0.0004 seconds.) [recipe_name: AVOCADO TOAST... - SP

```
SELECT r.recipe_name, t.tag_name, t.color FROM Recipes r INNER JOIN Recipe_Tags rt ON r.recipe_id
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

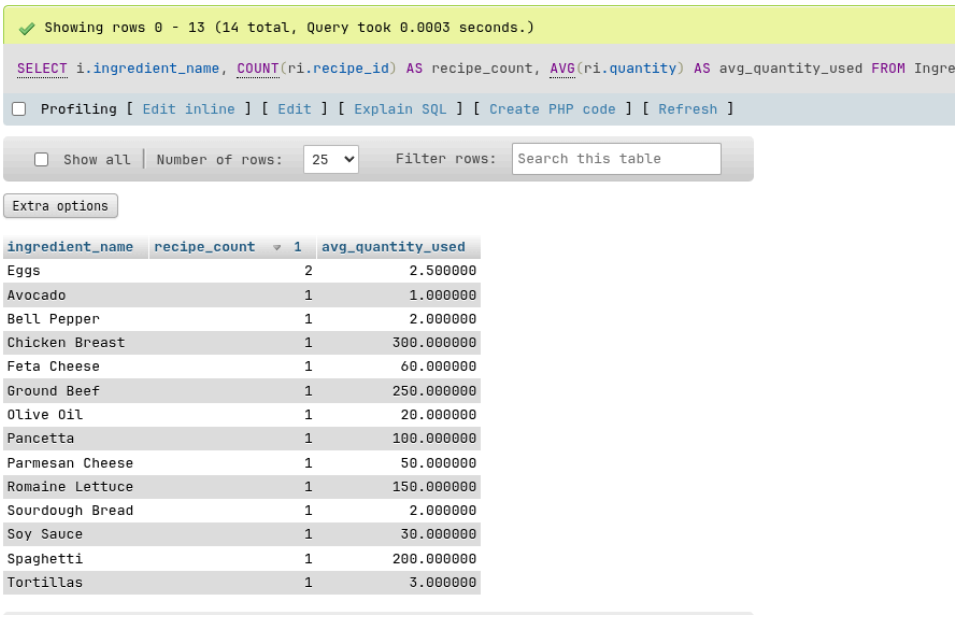
Show all | Number of rows: 25 Filter rows: Search this table

Extra options

recipe_name ▲ 1	tag_name	color
Avocado Toast	Quick	#F4A261
Avocado Toast	Vegetarian	#2A9D8F
Beef Tacos	Mexican	#E9C46A
Beef Tacos	High Protein	#E63946
Chicken Stir Fry	Quick	#F4A261
Chicken Stir Fry	High Protein	#E63946
Greek Salad	Quick	#F4A261
Greek Salad	Vegetarian	#2A9D8F
Spaghetti Carbonara	Italian	#009246
Spaghetti Carbonara	High Protein	#E63946

Query 4

Query 4 selects the ingredient names, counts the recipe IDs, and the average quantity, and then orders the ingredients by the most commonly used in different recipes, by average quantity. The purpose of this query is to learn what the most common ingredient is.



Showing rows 0 - 13 (14 total, Query took 0.0003 seconds.)

```
SELECT i.ingredient_name, COUNT(ri.recipe_id) AS recipe_count, AVG(ri.quantity) AS avg_quantity_used FROM Ingre
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

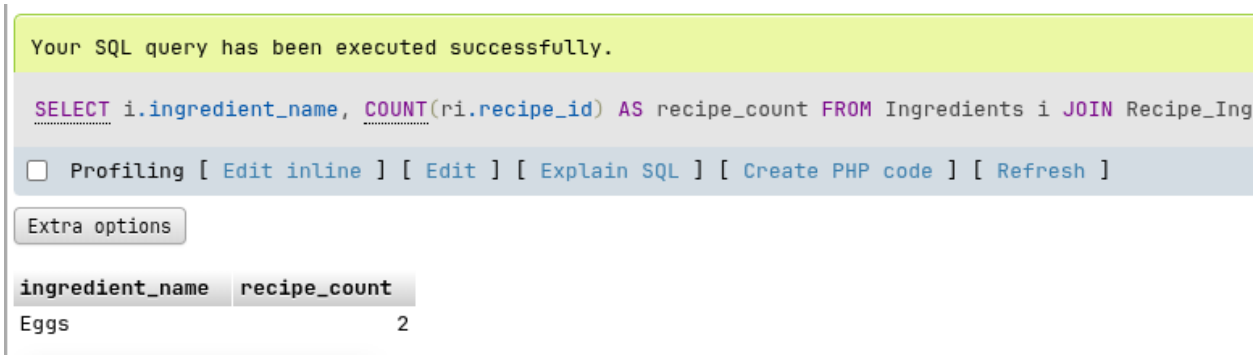
Show all | Number of rows: 25 Filter rows: Search this table

Extra options

ingredient_name	recipe_count ▼ 1	avg_quantity_used
Eggs	2	2.500000
Avocado	1	1.000000
Bell Pepper	1	2.000000
Chicken Breast	1	300.000000
Feta Cheese	1	60.000000
Ground Beef	1	250.000000
Olive Oil	1	20.000000
Pancetta	1	100.000000
Parmesan Cheese	1	50.000000
Romaine Lettuce	1	150.000000
Sourdough Bread	1	2.000000
Soy Sauce	1	30.000000
Spaghetti	1	200.000000
Tortillas	1	3.000000

Query 5

Query 5 counts the number of ingredients used in more than one recipe and displays them. The purpose of this query is to find the most common ingredient among recipes



Your SQL query has been executed successfully.

```
SELECT i.ingredient_name, COUNT(ri.recipe_id) AS recipe_count FROM Ingredients i JOIN Recipe_Ing
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Extra options

ingredient_name	recipe_count
Eggs	2