

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
(повна назва інституту/факультету)

КАФЕДРА інформатики та програмної інженерії
(повна назва кафедри)

КУРСОВА РОБОТА

з дисципліни «Бази даних»
(назва дисципліни)

на тему: База даних для кулінарної рецептури

Студента (ки) 2 курсу ІП-22 групи
спеціальності 121 «Інженерія програмного
забезпечення»

Андрєвої Уляни Андріївни
(прізвище та ініціали)

Керівник старший викладач, Марченко О.І.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала _____

Кількість балів: _____ Оцінка ECTS _____

Члени комісії

_____	_____
(підпис)	(вчене звання, науковий ступінь, прізвище та ініціали)
_____	_____
(підпис)	(вчене звання, науковий ступінь, прізвище та ініціали)
_____	_____
(підпис)	(вчене звання, науковий ступінь, прізвище та ініціали)

Київ – 2023 рік

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет Інформатики та обчислювальної техніки
(повна назва)

Кафедра Інформатики та програмної інженерії
(повна назва)

Дисципліна Бази даних

Курс 2 Група ІІІ-22 Семестр 3

**З А В Д А Н Н Я
НА КУРСОВУ РОБОТУ СТУДЕНТУ**

Андреевій Уляні Андріївні

(прізвище, ім'я, по батькові)

1. Тема роботи База даних для кулінарної рецептури

керівник роботи старший викладач, Марченко О.І.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

2. Строк подання студентом роботи 28.12.2023

3. Вихідні дані до роботи
завдання на розробку бази даних кулінарної рецептури

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1) Аналіз предметного середовища

2) Побудова ER-моделі

3) Побудова реляційної схеми з ER-моделі

4) Створення бази даних, у форматі обраної системи управління базою даних

5) Створення користувачів бази даних

6) Імпорт даних з використанням засобів СУБД в створену базу даних

7) Створення мовою SQL запитів

8) Оптимізація роботи запитів

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Дата видачі завдання 04.11.2023

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання курсового проекту	Строк виконання етапів проекту	Примітка
1	Аналіз предметного середовища	12.11.2023	
2	Побудова ER-моделі	02.12.2023	
3	Побудова реляційної схеми з ER-моделі	03.12.2023	
4	Створення бази даних, у форматі обраної системи управління базою даних	15.12.2023	
5	Створення користувачів бази даних	17.12.2023	
6	Імпорт даних з використанням засобів СУБД в створену базу даних	19.12.2023	
7	Створення мовою SQL запитів	22.12.2023	
8	Оптимізація роботи запитів	23.12.2023	
9	Оформлення пояснювальної записки	27.12.2023	
10	Захист курсової роботи	30.12.2023	

Студент

(підпис) Уляна АНДРЕЄВА _____
(прізвище та ініціали)

Керівник роботи

(підпис) Олена МАРЧЕНКО _____
(прізвище та ініціали)

АНОТАЦІЯ

Пояснювальна записка до курсової роботи: 89 сторінок, 64 рисунк, 19 таблиць, 1 посилання.

Об'єкт дослідження: кулінарної рецептури.

Мета роботи: розробка та реалізація бази даних для управління та обміном кулінарними рецептами.

Проведено аналіз предметного середовища, побудована ER-модель та реляційна схема, створена база даних у форматі MySQL. Створені користувачі та ролі. Були імпортовані дані в створену базу даних. Створені функції, процедури, тригери, представлення та SQL запити для роботи з базою даних. Також була проведена оптимізація роботи запитів.

Виконана реалізація бази даних кулінарними рецептами у форматі MySQL згідно з варіантом завдання.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОГО СЕРЕДОВИЩА	8
1.1 Опис предметного середовища	8
1.2 Аналіз існуючих програмних продуктів	11
2 ПОСТАНОВКА ЗАВДАННЯ.....	13
3 ПОБУДОВА ER-МОДЕЛІ	15
3.1 Бізнес-правила	15
3.2 Вибір сутностей.....	15
3.3 Опис сутностей.....	16
4 РЕАЛЯЦІЙНА МОДЕЛЬ БАЗИ ДАНИХ	21
5 РЕАЛІЗАЦІЯ БАЗИ ДАНИХ.....	28
5.1 Створення бази даних	28
5.2 Імпортування даних в таблиці	33
6 СТВОРЕННЯ КОРИСТУВАЧІВ БАЗИ ДАНИХ	34
6.1 Кулінар – аматор	34
6.2 Професійний кухар.....	34
6.3 Дієтичний консультант	34
6.4 Блогер кулінарного каналу	34
6.5 Користувач	34
7 РОБОТА З БАЗОЮ ДАНИХ	36
7.1 Тексти генераторів	36
7.2 Тексти збережених процедур та функцій.	37
7.3 Тригери.....	46
7.4 Представлення	57
7.5 SQL-запити.....	64

7.6 Індеси та результати оптимізації	81
ВИСНОВКИ	88
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	89

ВСТУП

У наш час, коли величезну роль у повсякденному житті відіграють інформаційні технології, значимість ефективного зберігання та аналізу даних стає все більш важливою у різних аспектах нашого життя. Особливо це стосується галузі кулінарії, де велика кількість рецептів, інгредієнтів та кулінарних технік вимагають організованого підходу до зберігання та доступу до інформації.

Актуальність створення бази даних для кулінарної рецептури обумовлюється необхідністю забезпечити легкий доступ до рецептів, інгредієнтів, інструкцій приготування та іншої важливої інформації. У сучасному світі, де швидкість та ефективність обробки інформації мають велике значення, така база даних може стати незамінним помічником для професіоналів та любителів кулінарії.

Ця курсова робота має на меті розробити та проаналізувати структуру бази даних для кулінарної рецептури, використовуючи систему управління базами даних. Основні завдання включають детальний аналіз функціональних вимог до бази даних, розробку ефективної структури для зберігання та обробки кулінарної інформації, а також впровадження методів для оптимізації пошуку та доступу до даних.

Метою створення такої бази даних є полегшення процесу пошуку та використання кулінарних рецептів, зменшення часових витрат на їх пошук, та забезпечення надійного та цілісного зберігання інформації. Сфера застосування охоплює різноманітні аспекти кулінарного мистецтва, від домашнього приготування їжі до професійного кулінарного виробництва.

Основними перевагами використання бази даних є можливість зберігати велику кількість рецептів, легко знаходити потрібну інформацію за різними критеріями (тип страви, інгредієнти, складність приготування тощо) та забезпечення високої якості та зручності користування.

Таким чином, ця курсова робота є спробою створити інформаційну систему, яка відповідатиме сучасним вимогам у галузі кулінарії та допоможе покращити якість та ефективність кулінарного процесу.

1 АНАЛІЗ ПРЕДМЕТНОГО СЕРЕДОВИЩА

1.1 Опис предметного середовища

База даних для кулінарної рецептури розроблена з метою забезпечення зручного та ефективного управління інформацією, пов'язаною з кулінарними рецептами та процесом приготування їжі. Це включає в себе великий спектр інформації, починаючи від інгредієнтів та закінчуючи детальними інструкціями приготування.

Основні компоненти системи:

- **Інгредієнти:** Вся інформація про інгредієнти, їх доступність, альтернативи, сезонність, харчова цінність та потенційні алергени.
- **Рецепти:** Різноманітні рецепти, включаючи методи приготування, час готування, рівень складності та поради щодо подачі страв.
- **Користувацькі налаштування:** Можливість користувачам створювати власні колекції рецептів, відзначати улюблені страви та ділитися ними з іншими.
- **Інтерактивні функції:** Відгуки користувачів, рейтинги рецептів, форуми для обговорення кулінарних питань та обміну досвідом.

Аналіз потреб користувачів:

- **Аматори кулінарії:** Пошук легких та інноваційних рецептів, можливість поділитися власними створіннями.
- **Професійні шеф-кухарі:** Доступ до складних та ексклюзивних рецептів, можливість обміну професійним досвідом.
- **Дієтичні консультанти:** Рецепти з контролем калорійності та харчових складових, що дозволяє рекомендувати страви згідно з дієтичними потребами клієнтів.

- **Кулінарні блогери:** Можливість публікації унікальних рецептів, залучення аудиторії та взаємодії з підписниками.

Технологічні аспекти:

- **Безпека даних:** Захист персональних даних користувачів та їх кулінарних творінь від несанкціонованого доступу.
- **Інтеграція з соціальними мережами:** Дозволяє користувачам ділитися своїми кулінарними досягненнями на інших платформах.
- **Адаптивність:** Система повинна бути зручною для використання на різних пристроях, включаючи мобільні телефони, планшети та комп'ютери.
- **Штучний інтелект:** Аналіз популярності рецептів, автоматизовані рекомендації, персоналізовані пошукові запити.

Майбутні розширення:

- **Віртуальні кулінарні майстер-класи:** Інтеграція з платформами для проведення онлайн-курсів.
- **Розширений аналітичний інструментарій:** Для шеф-кухарів та рестораторів, що дозволяє аналізувати кулінарні тренди та переваги споживачів.
- **Інтеграція з постачальниками продуктів:** Спрощення процесу закупівлі інгредієнтів для користувачів.

Виклики та перспективи:

- **Впровадження нових технологій:** Інтеграція з AR/VR для створення інтерактивного кулінарного досвіду.
- **Підтримка багатомовності:** Доступність рецептів на різних мовах, що робить базу даних доступною для міжнародної аудиторії.

- **Залучення спільноти:** Створення мотиваційних програм для активних користувачів та професіоналів.

Вхідними даними для системи є рецепти, їх інгредієнти, методи приготування, інформація про калорійність, а також можливі варіації та адаптації рецептів. Це включає в себе інформацію про тип кухні, рівень складності приготування, час приготування, а також специфічні інструкції та поради.

Основними завданнями системи є надання можливості швидкого пошуку рецептів за різними критеріями, такими як інгредієнти, тип страви, час приготування тощо. Система також має забезпечити зберігання та систематизацію великої кількості рецептів, що дозволяє користувачам легко знаходити потрібні їм рецепти.

Вихідними даними системи є відібрані та сортовані списки рецептів, що відповідають запитам користувачів. Система може надавати рекомендації на основі популярності рецептів, сезонності інгредієнтів, харчових переваг користувачів або їхніх алергій.

Задачі системи включають:

- а) Розробку структури бази даних, яка відповідає потребам зберігання рецептів та їх характеристик.
- б) Автоматизацію процесів пошуку та фільтрації рецептів на основі заданих користувачами критеріїв.
- в) Забезпечення моніторингу та аналізу користувальницької активності для покращення системи рекомендацій.
- г) Розробку інтерфейсу, що є зручним та інтуїтивно зрозумілим для користувачів.

Ключовим аспектом системи є забезпечення зручності та ефективності доступу до кулінарної інформації. Система повинна бути гнучкою, дозволяючи користувачам легко додавати та редагувати рецепти, а також ділитися ними з іншими.

Очікується, що розроблена база даних стане важливим інструментом для широкого кола користувачів - від аматорів кулінарії до професійних шеф-кухарів. Це дозволить оптимізувати процес пошуку та обміну кулінарними знаннями, підвищуючи ефективність та якість кулінарного досвіду.[3]

1.2 Аналіз існуючих програмних продуктів

На ринку існують кілька програмних продуктів та платформ, призначених для кулінарних ентузіастів, ресторанів і кулінарних експертів. Ось деякі з них:

- AllRecipes: Веб-сайт і додаток, які містять велику кількість рецептів і можливість пошуку за інгредієнтами.
- Food Network: Ресурс Food Network також має велику кількість рецептів та інші корисні ресурси для кулінарів.
- Yummly: Додаток та веб-сайт, які надають персоналізовані рецепти на основі ваших вподобань та інгредієнтів, які ви маєте під рукою.
- ChefTap: Додаток для зберігання та організації ваших улюблених рецептів з різних джерел.
- Paprika: Додаток для кулінарів, який допомагає зберігати та організовувати рецепти, планувати приготування їжі та створювати списки покупок.
- BigOven: Велика база даних з рецептами та функціоналом для планування страв та списків покупок.
- Cook'n: Програма для організації рецептів, створення меню та планування страв.

- Tasty: Додаток від BuzzFeed з короткими відеорецептами.

Ці програмні продукти можуть бути використані як ресурси для кулінарів та ресторанів для автоматизації управління рецептами, меню, списками покупок та іншими аспектами гастрономічного дослідження та приготування страв.

2 ПОСТАНОВКА ЗАВДАННЯ

Метою даної курсової роботи є розробка та реалізація бази даних для управління процесом приготування страв згідно кулінарної рецептури. Основними напрямками роботи є автоматизація зберігання та оброблення рецептів, інгредієнтів, даних про страви та контролю над ними.

Задачі роботи:

- 1) аналіз предметного середовища;
- 2) побудова ER-моделі, що відповідає потребам кулінарних рецептів;
- 3) побудова реляційної схеми на підставі розробленої ER-моделі;
- 4) створення бази даних, що була спроектована, з використання СУБД MySQL;
- 5) імпорт даних з використанням засобів СУБД в створену базу даних;
- 6) з використанням мови SQL автоматизація ключових бізнес-процесів, таких як збереження, визначення страв, інгредієнтів та їх властивостей, а також забезпечення можливості моніторингу та аналізу отриманих даних для користувачів[5];
- 7) оптимізація роботи запитів;
- 8) підтримка багатокористувальницького доступу.

База даних повинна відповідати стандартам ефективного управління кулінарною рецептурою та забезпечувати зручний доступ до інформації. Дані повинні зберігатися відповідно до вимог щодо конфіденційності та захисту особистих даних.

База даних буде включати в себе різні таблиці, що описують різні аспекти кулінарної рецептури. Ключові сутності включають рецепти, інгредієнти, дані про страви та контроль над ними. Додатковою важливою сутністю є обліковий запис користувача, через який користувачі матимуть доступ до системи та зможуть взаємодіяти з рецептами та інгредієнтами.

Розробка та впровадження цієї бази даних для управління кулінарною рецептурою сприятиме поліпшенню ефективності та доступності у готуванні різноманітних страв. Очікується, що ця база даних стане ефективним

інструментом для кулінарів та любителів готування, сприяючи зручній та смачній кулінарній творчості. Концептуальна модель відображає важливі аспекти управління та взаємодії з інформацією, які сприятимуть оптимізації та автоматизації процесів у галузі кулінарії.

3 ПОБУДОВА ER-МОДЕЛІ

3.1 Бізнес-правила

До основних бізнес-правил належить:

- 1) Користувач може додати до 5 улюблених рецептів.
- 2) Користувач може завантажити до 20 рецептів.
- 3) Користувач не може додати рецепт з вже використаною назвою.
- 4) Рейтинг рецепту не може бути нижчим за мінімальний поріг, встановлений для даної категорії страв.
- 5) Рейтинг рецепту повинен знаходитися в межах від 1 до 5.
- 6) Користувач не може додати однаковий тег до рецепту більше одного разу.
- 7) Перед завантаженням рецепту, користувач повинен вказати назву страви та інгредієнти.
- 8) Мімальна кількість інгредієнтів для рецепту повинна бути більшою за один.
- 9) Максимальна кількість тегів для рецепту не повинна перевищувати загальну кількість дозволених тегів.
- 10) Додавання рецепту здійснюється лише після проходження модерації.

3.2 Вибір сутностей

Відповідно до поставленого завдання та предметного середовища було виділено наступні сутності:

1. users
2. recipes
3. ingredients
4. instructions
5. dish_types
6. recipe_sources
7. recipe_servings

- 8. recipe_tags
- 9. recipe_notes
- 10.recipe_dietary_labels
- 11.user_profiles
- 12.user_favorites
- 13.user_ratings
- 14.user_dietary_preferences
- 15.recipe_comments
- 16.recipe_videos
- 17.recipe_instructions
- 18.recipe_prep_time

3.3 Опис сутностей

Після детального аналізу опису предметного середовища були виділені наступні атрибути сутностей (таблиця 3.1).

Таблиця 3.1 – Опис сутностей

Назва сутності	Атрибути	Опис
users	user_id	Унікальний ідентифікатор користувача
	username	Ім'я користувача у системі
	password	Пароль для доступу до акаунту
user_profiles	user_id	Вказує на конкретного користувача
	full_name	Повне ім'я користувача
	email	Електронна адреса користувача
	profile_picture_url	URL профільного зображення користувача
	bio	Коротка біографія користувача системи
user_ratings	user_id	Вказує на конкретного користувача , який виставив рейтинг
	recipe_id	Унікальний ідентифікатор рецепту, якому було виставлено рейтинг
	rating	Рейтинг, який користувач виставив рецепту

Продовження таблиці 3.1

Назва сутності	Атрибути	Опис
user_dietary_preferences	user_id	Унікальний ідентифікатор користувача, який має певні дієтичні уподобання
	dietary_preferences	Опис дієтичних уподобань користувача
recipes	recipe_id	Унікальний ідентифікатор рецепту
	recipe_name	Назва рецепту
	dish_type_id	Ідентифікатор типу страви
	calories	Кількість калорій на порцію
	avg_rating	Середній рейтинг рецепту
	comment_count	Кількість коментарів
recipe_videos	recipe_id	Унікальний ідентифікатор рецепту
	video_url	URL відео
recipe_tags	recipe_id	Унікальний ідентифікатор рецепту
	tag_name	Назва тегу
recipe_sources	recipe_id	Унікальний ідентифікатор рецепту
	source_name	Назва джерела
	source_url	URL джерела
recipe_servings	recipe_id	Унікальний ідентифікатор спеціальності
	servings	Кількість порцій
	servings_size	Розмір однієї порції
recipe_prep_time	recipe_id	Унікальний ідентифікатор рецепту
	prep_time	Час підготовки
	cook_time	Час приготування
	total_time	Загальний час приготування.
recipe_notes	recipe_id	Унікальний ідентифікатор рецепту
	note_id	Унікальний ідентифікатор примітки
	note_text	Текст примітки
recipe_ingredients	recipe_id	Унікальний ідентифікатор рецепту
	ingredient_id	Унікальний ідентифікатор певного інгредієнту
	quantity	Певна к-сть інгредієнту, яка потрібна для конкретного рецепту

Продовження таблиці 3.1

Назва сутності	Атрибути	Опис
user_favorites	user_id	Вказує на конкретного користувача
	recipe_id	Унікальний ідентифікатор рецепту, який є улюбленим для певного користувача
recipe_dietary_labels	recipe_id	Унікальний ідентифікатор рецепту
	dietary_label	Дієтичний ярлик
recipe_comments	comment_id	Унікальний ідентифікатор коментаря
	recipe_id	Унікальний ідентифікатор рецепту
	user_id	Унікальний ідентифікатор користувача, який залишив коментар
	comment_text	Текст коментаря
	comment_date	Дата коментаря
instructions	instruction_id	Унікальний ідентифікатор інструкції
	recipe_id	Унікальний ідентифікатор рецепту
	step_number	Номер кроку
	instruction_text	Текст інструкції
ingredients	ingredient_id	Унікальний ідентифікатор інгредієнта
	ingredient_name	Ідентифікатор абітурієнта
dish_types	dish_type_id	Унікальний ідентифікатор типу страви
	dish_type_name	Назва типу страви

Опис зв'язків між сутностями:

1. **Users – Recipes:** один до багатьох (1:N) – кожен користувач може створити багато рецептів, але кожен рецепт належить лише одному користувачу.
2. **Dish Types – Recipes:** один до багатьох (1:N) – кожен тип страви може бути асоційований з багатьма рецептами, але кожен рецепт відноситься лише до одного типу страви.
3. **Recipes – Recipe Ingredients:** один до багатьох (1:N) – кожен рецепт може містити багато інгредієнтів, але кожен інгредієнт в контексті рецепту є унікальним.

4. **Ingredients – Recipe Ingredients:** один до багатьох (1:N) – кожен інгредієнт може бути використаний у багатьох рецептах, але у рамках одного рецепту кожна інгредієнтна позиція є унікальною.
5. **Users – Recipe Comments:** один до багатьох (1:N) – кожен користувач може залишити багато коментарів (по одному на рецепт), але кожен коментар належить лише одному користувачу.
6. **Recipes – Recipe Comments:** один до багатьох (1:N) – кожен рецепт може мати багато коментарів, але кожен коментар належить до одного рецепту.
7. **Users – User Profiles:** один до одного (1:1) – кожен користувач має один профіль, і кожен профіль належить одному користувачу.
8. **Users – User Dietary Preferences:** один до багатьох (1:N) – кожен користувач може мати багато дієтичних переваг, але кожна перевага належить лише одному користувачу.
9. **Recipes – Recipe Tags:** багато до багатьох (M:N) – кожен рецепт може мати багато тегів, і кожен тег може бути присвоєний багатьом рецептам. Зв'язок реалізований через проміжну сутність (не показану на діаграмі).
10. **Recipes – Recipe Dietary Labels:** багато до багатьох (M:N) – кожен рецепт може мати багато дієтичних позначок, і кожна дієтична позначка може бути використана в багатьох рецептах. Зв'язок реалізований через проміжну сутність (не показану на діаграмі).
11. **Recipes – Recipe Sources:** один до багатьох (1:N) – кожен рецепт може мати багато джерел, але кожне джерело вказує на один рецепт.
12. **Recipes – Recipe Notes:** один до багатьох (1:N) – кожен рецепт може мати багато приміток, але кожна примітка належить до одного рецепту.

Побудована ER-модель у відомій нотації ІЕ (Crow's Foot) [6] зображена на рис. 3.1. Кожна таблиця має свої ключові атрибути та відносини з іншими таблицями, створюючи добре структуровану систему для зберігання та управління інформацією. Зовнішні ключі використовуються для забезпечення

4 РЕАЛЯЦІЙНА МОДЕЛЬ БАЗИ ДАНИХ

Для виконання даної роботи було обрано базу даних MySQL з ряду ключових причин, які враховують вимоги та особливості системи кулінарної рецептури. По-перше, MySQL є однією з найпоширеніших та високопродуктивних реляційних систем управління базами даних, що забезпечує надійність та стабільність роботи системи при великому об'ємі інформації. Вона дозволяє ефективно зберігати та оптимізовувати доступ до великої кількості даних, що є важливим у кулінарії, де обробка великої кількості рецептів та їхніх документів є загальним завданням.

По-друге, MySQL має відкритий вихідний код, що дозволяє безкоштовно використовувати та модифікувати систему відповідно до конкретних потреб проекту. Це важливо для забезпечення гнучкості та можливості розширення системи в майбутньому. Відкритість дозволяє також використовувати широкий спектр інструментів та ресурсів спільноти для підтримки та оптимізації роботи бази даних.

Крім того, MySQL володіє великою кількістю документації та інструментів для розробників, що сприяє прискоренню процесу розробки та підтримки системи. Його широке поширення у сфері програмування та веб-розробки також робить його оптимальним вибором для створення системи кулінарної рецептури, яка може взаємодіяти з іншими інформаційними системами ресторанів. Узагальнюючи, обрання MySQL для цієї роботи ґрунтується на його надійності, відкритості та широкому застосуванні, що відповідає вимогам та цілям проекту.

Таблиця “recipe_tags” містить назви тегів. Структура таблиці наступна:

Ім'я поля	Тип даних	Розмір	Ключ	Опис
recipe_id	INT		FK	Ідентифікатор рецепту
tag_name	VARCHAR	50		Назва тегу, асоційованого з рецептом

Таблиця “recipe_notes” містить текстовий опис нотаток. Структура таблиці наступна:

Ім'я поля	Тип даних	Розмір	Ключ	Опис
recipe_id	INT		FK	Ідентифікатор рецепту
note_id	INT		PK	Унікальний ідентифікатор нотатки
note_text	text			Текст нотатки

Таблиця “recipe_sources” містить назви джерел та посилання, які були узяті для рецептів. Структура таблиці наступна:

Ім'я поля	Тип даних	Розмір	Ключ	Опис
recipe_id	INT		FK	Ідентифікатор рецепту
source_name	VARCHAR	255		Назва джерела рецепту
source_url	VARCHAR	255		URL-адреса джерела рецепту

Таблиця “recipe_videos” містить посилання на відео з рецептами. Структура таблиці наступна:

Ім'я поля	Тип даних	Розмір	Ключ	Опис
recipe_id	INT		FK	Ідентифікатор рецепту
video_url	VARCHAR	255		URL-адреса відео з рецептом

Таблиця “recipe_dietary_labels” містить позначки, які означають, що рецепт є дієтичним. Структура таблиці наступна:

Ім'я поля	Тип даних	Розмір	Ключ	Опис
recipe_id	INT		FK	Ідентифікатор рецепту
dietary_label	VARCHAR	50		Мітка дієтичного характеру рецепту

Таблиця “dish_types” містить інформацію про тип страви. Структура таблиці наступна:

Ім'я поля	Тип даних	Розмір	Ключ	Опис
dish_type_id	INT		PK	Унікальний ідентифікатор рецепту
dish_type_name	VARCHAR	80		Назва типу страви

Таблиця “recipes” містить інформацію про назву рецепту, калорії страви. Структура таблиці наступна:

Ім'я поля	Тип даних	Розмір	Ключ	Опис
recipe_id	INT		PK	Унікальний ідентифікатор рецепту
recipe_name	VARCHAR	80		Назва рецепту
dish_type_id	INT		FK	Ідентифікатор типу страви
calories	INT			Кількість калорій у страві

Таблиця “recipe_servings” містить інформацію про к-сть та розмір порцій. Структура таблиці наступна:

Ім'я поля	Тип даних	Розмір	Ключ	Опис
recipe_id	INT		FK	Ідентифікатор рецепту
servings	INT			Кількість порцій
serving_size	VARCHAR	50		Розмір однієї порції

Таблиця “recipe_comments” містить інформацію про коментар. Структура таблиці наступна:

Ім'я поля	Тип даних	Розмір	Ключ	Опис
comment_id	INT		PK	Унікальний ідентифікатор коментаря
recipe_id	INT		FK	Ідентифікатор рецепту
user_id	INT		FK	Ідентифікатор користувача
comment_text	text			Текст коментаря
comment_date	timestamp			Дата та час коментування

Таблиця “ingredients” містить інформацію про інгредієнти. Структура таблиці наступна:

Ім'я поля	Тип даних	Розмір	Ключ	Опис
ingredient_id	INT		PK	Унікальний ідентифікатор інгредієнта
ingredient_name	VARCHAR	80		Назва інгредієнта

Таблиця “recipe_prep_times” містить інформацію про час процесів приготування. Структура таблиці наступна:

Ім'я поля	Тип даних	Розмір	Ключ	Опис
recipe_id	INT		FK	Ідентифікатор рецепту
prep_time	INT			Час підготовки
cook_time	INT			Час приготування
total_time	INT			Загальний час приготування

Таблиця “instructions” містить інформацію про кроки під час приготування рецепту. Структура таблиці наступна:

Ім'я поля	Тип даних	Розмір	Ключ	Опис
instruction_id	INT		PK	Унікальний ідентифікатор інструкції
recipe_id	INT		FK	Ідентифікатор рецепту
step_number	INT			Номер кроку в інструкції
instruction_text	text			Текст інструкції

Таблиця “recipe_ingredients” містить інформацію про інгредієнти для рецепту. Структура таблиці наступна:

Ім'я поля	Тип даних	Розмір	Ключ	Опис
recipe_id	INT		FK	Ідентифікатор рецепту
ingredient_id	INT		FK	Ідентифікатор інгредієнта
quantity	VARCHAR	50		Кількість інгредієнта

Таблиця “users” містить інформацію про користувачів системи. Структура таблиці наступна:

Ім'я поля	Тип даних	Розмір	Ключ	Опис
user_id	INT		PK	Ідентифікатор користувача
username	VARCHAR	50		Логін користувача
password	VARCHAR	255		Пароль користувача

Таблиця “user_favorites” містить інформацію про додані користувачами рецепти до обраних. Структура таблиці наступна:

Ім'я поля	Тип даних	Розмір	Ключ	Опис
user_id	INT		FK	Ідентифікатор користувача. Вказує на користувача, який додав рецепт до обраного
recipe_id	INT		FK	Ідентифікатор рецепту з таблиці "recipes". Вказує на рецепт, який був доданий до обраного

Таблиця “user_ratings” містить інформацію про додані користувачами оцінки. Структура таблиці наступна:

Ім'я поля	Тип даних	Розмір	Ключ	Опис
user_id	INT		FK	Ідентифікатор користувача з таблиці "users". Вказує на користувача, який залишив рейтинг
recipe_id	INT		FK	Ідентифікатор рецепту з таблиці "recipes". Вказує на рецепт, якому було присвоєно рейтинг
rating	INT			Рейтинг рецепту, присвоєний користувачем

Таблиця “user_dietary_preferences” містить інформацію про дієтичні переваги користувачів. Структура таблиці наступна:

Ім'я поля	Тип даних	Розмір	Ключ	Опис
user_id	INT		FK	Ідентифікатор користувача з таблиці "users". Встановлює зв'язок між таблицею переваг та користувачем
dietary_preference	VARCHAR	50		Текстове поле, що описує конкретну дієтичну перевагу користувача, наприклад "веган" або "без глютену"

Таблиця “user_profiles” містить інформацію про профіль користувача. Структура таблиці наступна:

Ім'я поля	Тип даних	Розмір	Ключ	Опис
user_id	INT		FK	Ідентифікатор користувача з таблиці "users". Використовується для зв'язування профільної інформації з конкретним користувачем
full_name	VARCHAR		100	Повне ім'я користувача
email	VARCHAR		100	Електронна адреса користувача
profile_picture_url	VARCHAR		255	URL до зображення профілю користувача.
bio	text			Біографічна інформація або короткий опис профілю користувача.

5 РЕАЛІЗАЦІЯ БАЗИ ДАНИХ

5.1 Створення бази даних

Результатом проектування бази даних є сформований SQL-скрипт, який використовується для створення об'єктів, які були наведені в ER-моделі та табличних описах:

Створення БД

```
CREATE DATABASE culinary_recipe;
```

Створення таблиць

```
USE culinary_recipe;
```

```
CREATE TABLE dish_types (
    dish_type_id INT AUTO_INCREMENT PRIMARY KEY,
    dish_type_name VARCHAR(255) NOT NULL
);
```

```
CREATE TABLE ingredients (
    ingredient_id INT AUTO_INCREMENT PRIMARY KEY,
    ingredient_name VARCHAR(255) NOT NULL
);
```

```
CREATE TABLE recipes (
    recipe_id INT AUTO_INCREMENT PRIMARY KEY,
    recipe_name VARCHAR(255) NOT NULL,
    dish_type_id INT,
    calories INT,
    FOREIGN KEY (dish_type_id) REFERENCES dish_types(dish_type_id)
);
```

```
CREATE TABLE recipe_ingredients (
    recipe_id INT,
    ingredient_id INT,
    quantity VARCHAR(50) NOT NULL,
    FOREIGN KEY (recipe_id) REFERENCES recipes(recipe_id),
    FOREIGN KEY (ingredient_id) REFERENCES ingredients(ingredient_id)
);
```

```
CREATE TABLE instructions (
    instruction_id INT AUTO_INCREMENT PRIMARY KEY,
```

```

    recipe_id INT,
    step_number INT NOT NULL,
    instruction_text TEXT NOT NULL,
    FOREIGN KEY (recipe_id) REFERENCES recipes(recipe_id)
);

```

```

CREATE TABLE users (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL,
    password VARCHAR(255) NOT NULL
);

```

```

CREATE TABLE user_favorites (
    user_id INT,
    recipe_id INT,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (recipe_id) REFERENCES recipes(recipe_id)
);

```

```

CREATE TABLE user_ratings (
    user_id INT,
    recipe_id INT,
    rating INT NOT NULL,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (recipe_id) REFERENCES recipes(recipe_id)
);

```

```

CREATE TABLE recipe_prep_times (
    recipe_id INT,
    prep_time INT, -- in minutes
    cook_time INT, -- in minutes
    total_time INT, -- in minutes
    FOREIGN KEY (recipe_id) REFERENCES recipes(recipe_id)
);

```

-- Table to store recipe serving information

```

CREATE TABLE recipe_servings (
    recipe_id INT,
    servings INT,
    serving_size VARCHAR(50),
    FOREIGN KEY (recipe_id) REFERENCES recipes(recipe_id)
);

```

```
);
```

```
-- Table to store recipe dietary restrictions or labels
```

```
CREATE TABLE recipe_dietary_labels (
    recipe_id INT,
    dietary_label VARCHAR(50),
    FOREIGN KEY (recipe_id) REFERENCES recipes(recipe_id)
);
```

```
-- Table to store recipe video URLs or links
```

```
CREATE TABLE recipe_videos (
    recipe_id INT,
    video_url VARCHAR(255),
    FOREIGN KEY (recipe_id) REFERENCES recipes(recipe_id)
);
```

```
-- Table to store recipe source or author information
```

```
CREATE TABLE recipe_sources (
    recipe_id INT,
    source_name VARCHAR(255),
    source_url VARCHAR(255),
    FOREIGN KEY (recipe_id) REFERENCES recipes(recipe_id)
);
```

```
-- Table to store recipe comments or notes
```

```
CREATE TABLE recipe_notes (
    recipe_id INT,
    note_id INT AUTO_INCREMENT PRIMARY KEY,
    note_text TEXT,
    FOREIGN KEY (recipe_id) REFERENCES recipes(recipe_id)
);
```

```
-- Table to store user dietary preferences or restrictions
```

```
CREATE TABLE user_dietary_preferences (
    user_id INT,
    dietary_preference VARCHAR(50),
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

```
-- Table to store user profile information
```

```
CREATE TABLE user_profiles (
```

```

user_id INT,
full_name VARCHAR(100),
email VARCHAR(100),
profile_picture_url VARCHAR(255),
bio TEXT,
FOREIGN KEY (user_id) REFERENCES users(user_id)
);

-- Table to store recipe comments or notes
CREATE TABLE recipe_comments (
    comment_id INT AUTO_INCREMENT PRIMARY KEY,
    recipe_id INT,
    user_id INT,
    comment_text TEXT,
    comment_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (recipe_id) REFERENCES recipes(recipe_id),
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);

-- Table to store recipe tags or labels
CREATE TABLE recipe_tags (
    recipe_id INT,
    tag_name VARCHAR(50),
    FOREIGN KEY (recipe_id) REFERENCES recipes(recipe_id)
);

```

Обмеження:

- ALTER TABLE ingredients ADD CONSTRAINT unique_ingredient_name
UNIQUE (ingredient_name);
- ALTER TABLE recipes ADD CONSTRAINT check_calories CHECK
(calories > 0);
- ALTER TABLE recipes ADD CONSTRAINT unique_recipe_name
UNIQUE (recipe_name);
- ALTER TABLE recipe_ingredients ADD CONSTRAINT
fk_recipe_ingredients_recipe_id FOREIGN KEY (recipe_id) REFERENCES
recipes(recipe_id) ON DELETE CASCADE;


```
— ALTER      TABLE      recipe_comments      ADD      CONSTRAINT
  check_comment_length CHECK (CHAR_LENGTH(comment_text) <=
  500);

— ALTER TABLE recipe_servings ADD CONSTRAINT check_min_servings
  CHECK (servings > 0);

— ALTER TABLE users ADD CONSTRAINT unique_username UNIQUE
  (username);

— ALTER TABLE user_ratings ADD CONSTRAINT check_rating_range
  CHECK (rating >= 1 AND rating <= 5);

— ALTER TABLE recipe_videos ADD CONSTRAINT unique_video_url
  UNIQUE (video_url);

— ALTER TABLE user_profiles ADD CONSTRAINT unique_email UNIQUE
  (email);

— ALTER      TABLE      recipe_prep_times      ADD      CONSTRAINT
  check_prep_time_positive CHECK (prep_time >= 0);

— ALTER      TABLE      recipe_dietary_labels      ADD      CONSTRAINT
  unique_dietary_label UNIQUE (dietary_label, recipe_id);

— ALTER      TABLE      recipe_servings      ADD      CONSTRAINT
  check_positive_servings CHECK (servings > 0);
```

5.2 Імпортування даних в таблиці

Для імпортування згенерованих даних можна використати команду Insert в MySQL:

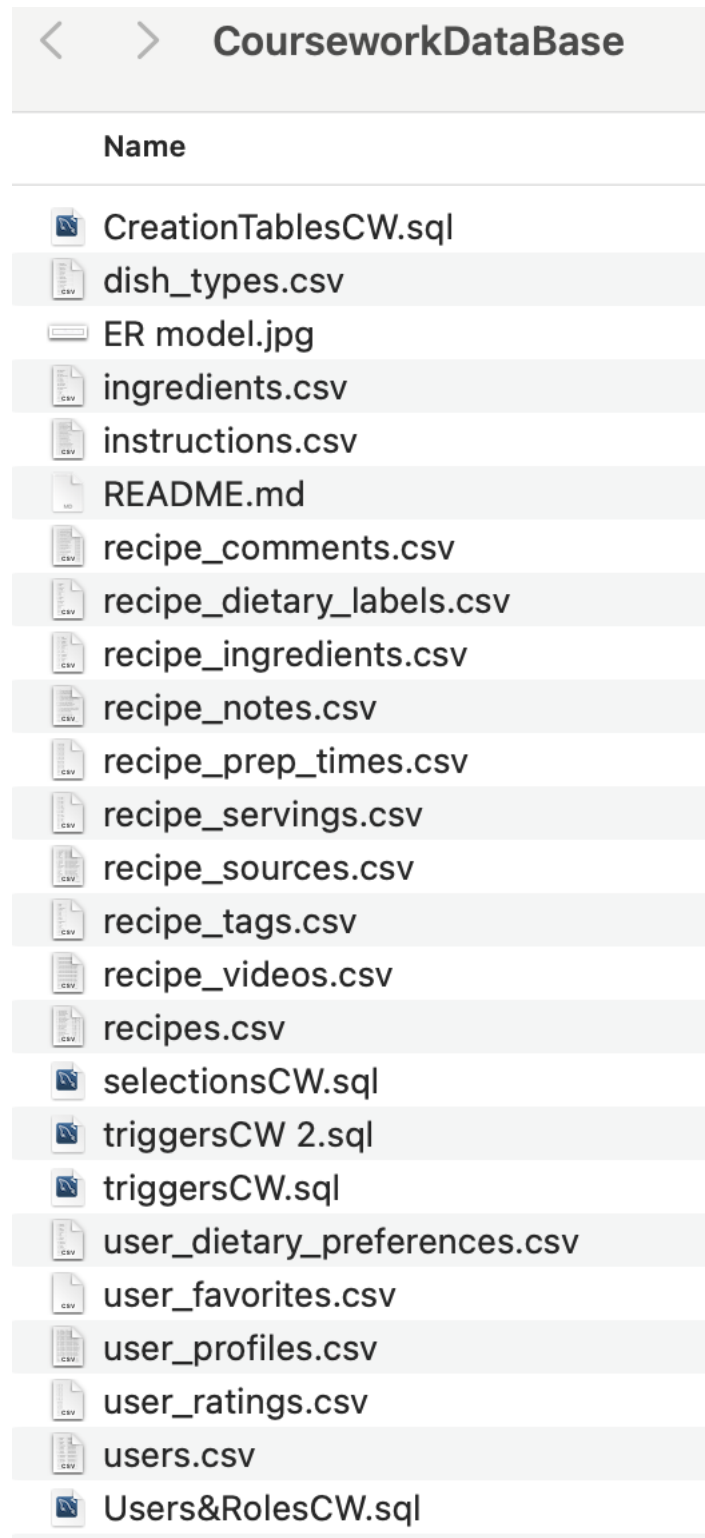


Рис. 5.2 – файли внесених даних у форматі csv

6 СТВОРЕННЯ КОРИСТУВАЧІВ БАЗИ ДАНИХ

6.1 Кулінар – аматор

```
CREATE USER 'amateur_cook'@'localhost' IDENTIFIED BY 'Basic';  
  
GRANT SELECT, INSERT, UPDATE, DELETE ON culinary_recipe.recipes TO  
'amateur_cook'@'localhost';  
  
GRANT SELECT ON culinary_recipe.dish_types TO 'amateur_cook'@'localhost';
```

6.2 Професійний кухар

```
CREATE USER 'professional_chef'@'localhost' IDENTIFIED BY 'Advanced';  
  
GRANT ALL PRIVILEGES ON culinary_recipe.* TO  
'professional_chef'@'localhost';
```

6.3 Дієтичний консультант

```
CREATE USER 'dietary_consultant'@'localhost' IDENTIFIED BY 'Diet';  
  
GRANT SELECT ON culinary_recipe.recipes TO 'dietary_consultant'@'localhost';
```

6.4 Блогер кулінарного каналу

```
CREATE USER 'food_blogger'@'localhost' IDENTIFIED BY 'Blogger';  
  
GRANT ALL PRIVILEGES ON culinary_recipe.* TO 'food_blogger'@'localhost';
```

6.5 Користувач

```
CREATE USER 'user'@'localhost' IDENTIFIED BY 'User';
```

```
CREATE ROLE professional_cooks;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON culinary_recipe.recipes TO  
professional_cooks;
```

```
GRANT SELECT ON culinary_recipe.dish_types TO professional_cooks;
```

```
GRANT professional_cooks TO 'user'@'localhost';
```

Результат виконання команд для створення ролей та користувачів командами SQL на рис. 6.1.

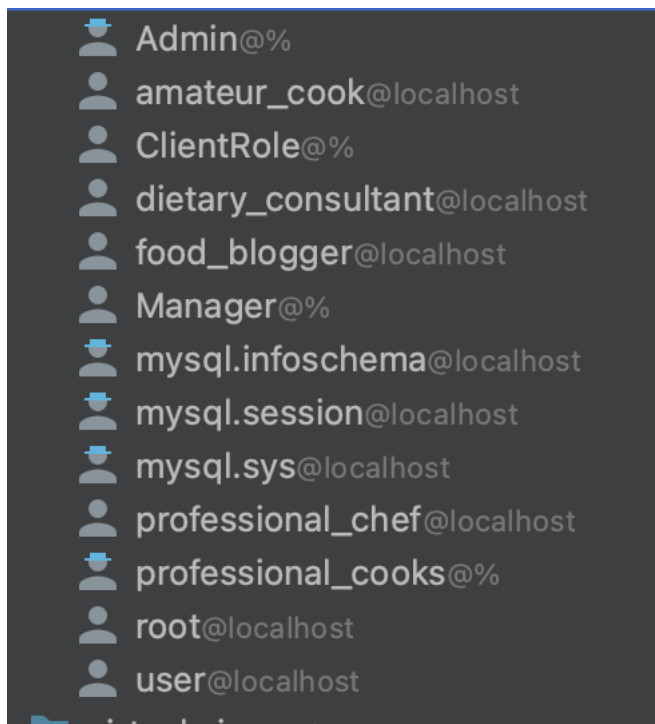


Рис. 6.1 – Створені ролі та користувачі в базі даних

7 РОБОТА З БАЗОЮ ДАНИХ

7.1 Тексти генераторів

Було встановлено значення AUTO_INCREMENT необхідним полям виконанням наступних SQL команд:

```
ALTER TABLE dish_types MODIFY COLUMN dish_type_id INT  
AUTO_INCREMENT PRIMARY KEY;
```

```
ALTER TABLE ingredients MODIFY COLUMN ingredient_id INT  
AUTO_INCREMENT PRIMARY KEY;
```

```
ALTER TABLE recipes MODIFY COLUMN recipe_id INT AUTO_INCREMENT  
PRIMARY KEY;
```

```
ALTER TABLE instructions MODIFY COLUMN instruction_id INT  
AUTO_INCREMENT PRIMARY KEY;
```

```
ALTER TABLE users MODIFY COLUMN user_id INT AUTO_INCREMENT  
PRIMARY KEY;
```

```
ALTER TABLE recipe_notes MODIFY COLUMN note_id INT  
AUTO_INCREMENT PRIMARY KEY;
```

```
ALTER TABLE recipe_comments MODIFY COLUMN comment_id INT  
AUTO_INCREMENT PRIMARY KEY;
```

7.2 Тексти збережених процедур та функцій.

Відповідно до мого предметного середовища були розроблені наступні функції.

7.2.1 Функція CalculateAverageCalories

Ця функція, названа CalculateAverageCalories, розраховує середню кількість калорій для рецептів певного типу страви в базі даних. Результат виконання функції на рис. 7.1.

2nd function

DELIMITER //

```
CREATE FUNCTION CalculateAverageCalories(dishTypeID INT)
```

```
RETURNS DECIMAL(10,2)
```

```
READS SQL DATA
```

```
BEGIN
```

```
    DECLARE avgCalories DECIMAL(10,2);
```

```
    SELECT AVG(calories) INTO avgCalories FROM recipes WHERE  
dish_type_id = dishTypeID;
```

```
    RETURN avgCalories;
```

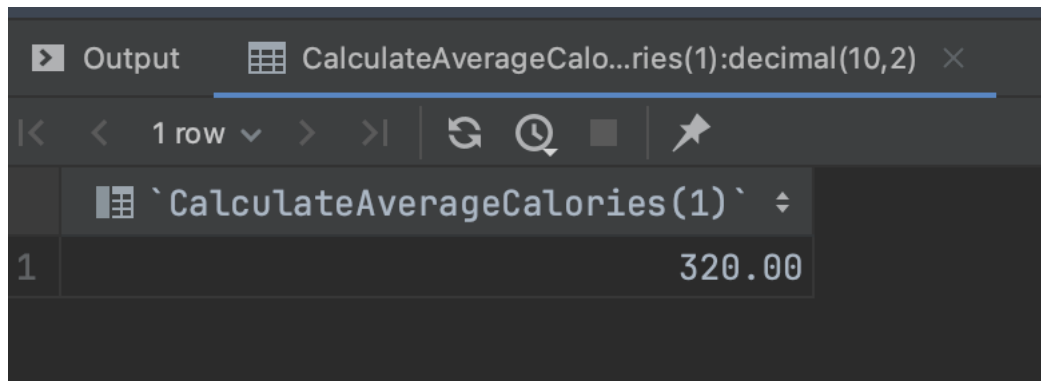
```
END;
```

```
//
```

```
DELIMITER ;
```

```
SET GLOBAL log_bin_trust_function_creators = 1;
```

```
SELECT CalculateAverageCalories(1);
```



The screenshot shows a database output window with a tab titled 'CalculateAverageCalo...ries(1):decimal(10,2)'. Below the tab, there is a table with one row. The first column is labeled '1' and the second column contains the value '320.00'. The table is titled 'CalculateAverageCalories(1)'.

1	320.00

Рис. 7.1 – Результат виконання функції CalculateAverageCalories

7.2.2 Функція GetTotalRecipesByDishType

Ця функція, названа GetTotalRecipesByDishType, призначена для підрахунку загальної кількості рецептів певного типу страви в таблиці recipes бази даних. Результати виконання функції на рис. 7.2 та рис. 7.3.

DELIMITER //

```
CREATE FUNCTION GetTotalRecipesByDishType(dishTypeID INT)
RETURNS INT
```

```
BEGIN
```

```
    DECLARE totalRecipes INT;
```

```
    SELECT COUNT(*) INTO totalRecipes FROM recipes WHERE dish_type_id
= dishTypeID;
```

```
    RETURN totalRecipes;
```

```
END;
```

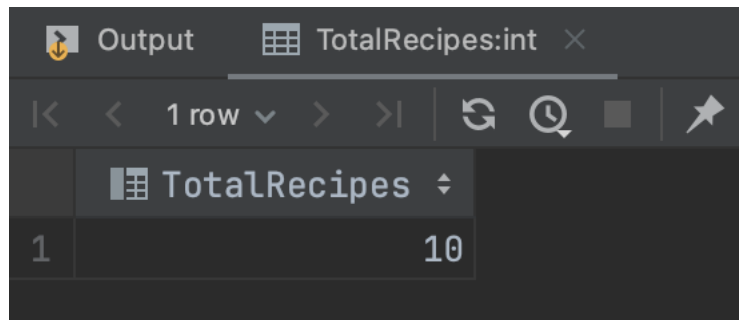
```
//
```

```
DELIMITER ;
```

```
INSERT INTO dish_types (dish_type_name) VALUES ('Dessert'), ('Main
Course');
```

```
INSERT INTO recipes (recipe_name, dish_type_id, calories) VALUES
('Chocolate Cake', 1, 500), ('Pasta', 2, 700);
```

```
SELECT GetTotalRecipesByDishType(1) AS TotalRecipes;
```



TotalRecipes
10

Рис. 7.2 – Результат виконання функції GetTotalRecipesByDishType

7.2.3 Процедура AddNewRecipe

Ця процедура, під назвою AddNewRecipe, створена для додавання нових рецептів до таблиці recipes у базі даних. Результат виконання процедури на рис. 7.3.

1st procedure

DELIMITER //

```
CREATE PROCEDURE AddNewRecipe(IN recipeName VARCHAR(255), IN
dishTypeID INT, IN calories INT)
```

```
BEGIN
```

```
    INSERT INTO recipes (recipe_name, dish_type_id, calories) VALUES
(recipeName, dishTypeID, calories);
```

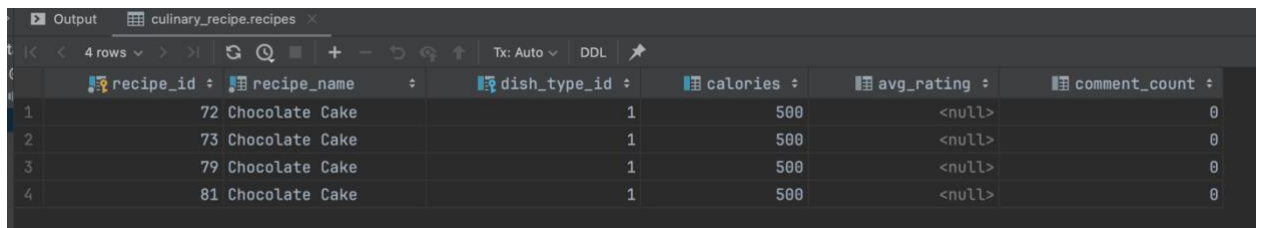
```
END;
```

```
//
```

```
DELIMITER ;
```

```
CALL AddNewRecipe('Chocolate Cake', 1, 500);
```

```
SELECT * FROM recipes WHERE recipe_name = 'Chocolate Cake';
```



	recipe_id	recipe_name	dish_type_id	calories	avg_rating	comment_count
1	72	Chocolate Cake	1	500	<null>	0
2	73	Chocolate Cake	1	500	<null>	0
3	79	Chocolate Cake	1	500	<null>	0
4	81	Chocolate Cake	1	500	<null>	0

Рис. 7.3 – Результат виконання процедури AddNewRecipe

7.2.4 Процедура UpdateRecipeCalories

Ця процедура, названа UpdateRecipeCalories, призначена для оновлення кількості калорій у вже існуючому рецепті в таблиці recipes бази даних. Результат виконання процедури, на рис. 7.4.

#3rd procedure

DELIMITER //

```
CREATE PROCEDURE UpdateRecipeCalories(IN recipeID INT, IN newCalories
INT)
```

```
BEGIN
```

```
    UPDATE recipes SET calories = newCalories WHERE recipe_id = recipeID;
```

```
END;
```

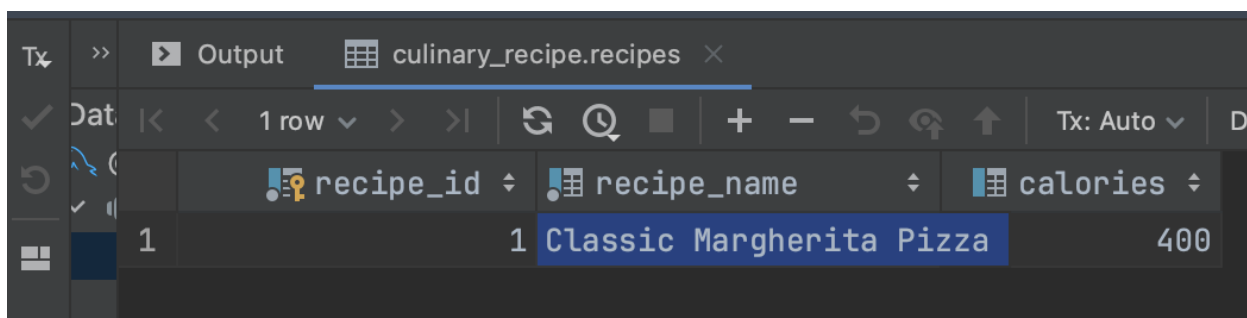
```
//
```

```
DELIMITER ;
```

```
INSERT INTO recipes (recipe_name, dish_type_id, calories) VALUES ('Test
Recipe', 1, 300);
```

```
CALL UpdateRecipeCalories(1, 400);
```

```
SELECT recipe_id, recipe_name, calories FROM recipes WHERE recipe_id = 1;
```



recipe_id	recipe_name	calories
1	Classic Margherita Pizza	400

Рис. 7.4 – Результат виконання процедури UpdateRecipeCalories

7.2.5 Процедура DeleteRecipe

Ця процедура, названа DeleteRecipe, призначена для видалення запису про рецепт із таблиці recipes у базі даних на основі його ідентифікатора. Результат виконання процедури на рис. 7.5.

#4th procedure

DELIMITER //

```
CREATE PROCEDURE DeleteRecipe(IN recipeID INT)
```

```
BEGIN
```

```
    DELETE FROM recipes WHERE recipe_id = recipeID;
```

```
END;
```

```
//
```

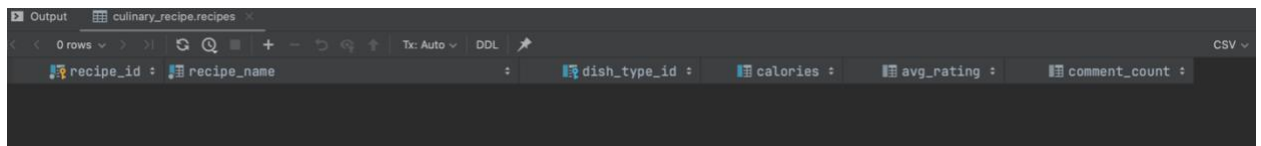
```
DELIMITER ;
```

```
INSERT INTO recipes (recipe_name, dish_type_id, calories) VALUES ('Sample Recipe', 1, 200);
```

```
SELECT recipe_id, recipe_name FROM recipes;
```

```
CALL DeleteRecipe(101);
```

```
SELECT * FROM recipes WHERE recipe_id = 101;
```



recipe_id	recipe_name	dish_type_id	calories	avg_rating	comment_count
0 rows					

Рис. 7.5 – Результати видалення процедури DeleteRecipe

7.2.6 Процедура AddIngredientToRecipe

Ця процедура, названа AddIngredientToRecipe, призначена для додавання інгредієнта до конкретного рецепту в таблиці recipe_ingredients бази даних. Результат виконання функції на рис. 7.6.

DELIMITER //

```
CREATE PROCEDURE AddIngredientToRecipe(IN recipeID INT, IN ingredientID INT, IN quantity VARCHAR(50))
```

```
BEGIN
```

```
INSERT INTO recipe_ingredients (recipe_id, ingredient_id, quantity) VALUES
(recipeID, ingredientID, quantity);

END;
```

```
//
```

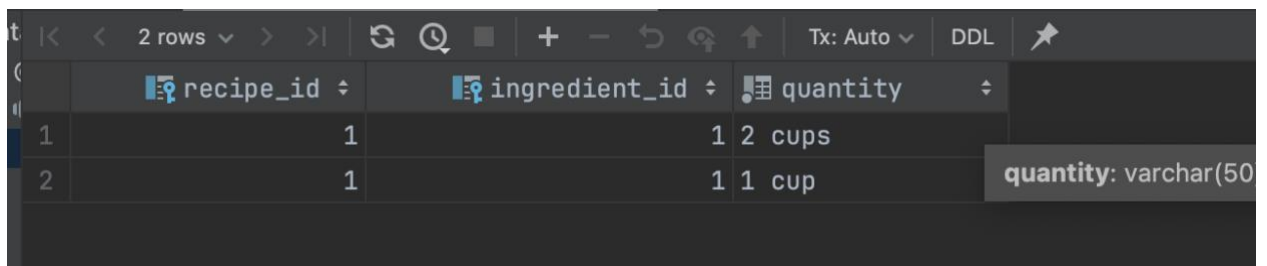
```
DELIMITER ;
```

```
INSERT INTO recipes (recipe_name, dish_type_id, calories) VALUES ('Test
Recipe', 1, 200);
```

```
INSERT INTO ingredients (ingredient_name) VALUES ('Sugar'), ('Salt');
```

```
CALL AddIngredientToRecipe(1, 1, '2 cups');
```

```
SELECT DISTINCT * FROM recipe_ingredients WHERE recipe_id = 1 AND
ingredient_id = 1;
```



	recipe_id	ingredient_id	quantity
1	1	1	2 cups
2	1	1	1 cup

Рис. 7.6 – Результат виконання процедури AddIngredientToRecipe

7.2.7 Процедура RateRecipe

Ця процедура, названа RateRecipe, призначена для додавання оцінки користувачем конкретному рецепту в таблицю user_ratings бази даних. Результат виконання процедури на рис. 7.7.

```
DELIMITER //
```

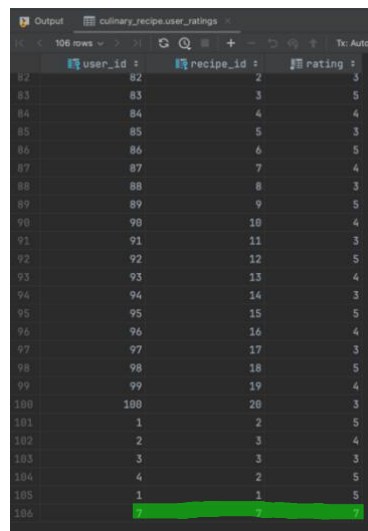
```
CREATE PROCEDURE RateRecipe(IN userID INT, IN recipeID INT, IN rating
INT)
```

```
BEGIN
```

```
INSERT INTO user_ratings (user_id, recipe_id, rating) VALUES (userID,
recipeID, rating);
```

```
END;
```

```
//
DELIMITER ;
INSERT INTO users (username, password) VALUES ('user1', 'password1'),
('user2', 'password2');
INSERT INTO recipes (recipe_name) VALUES ('Recipe A'), ('Recipe B');
CALL RateRecipe(7, 7, 7);
SELECT * FROM user_ratings;
```



	user_id	recipe_id	rating
82	82	2	3
83	83	3	5
84	84	4	4
85	85	5	3
86	86	6	5
87	87	7	4
88	88	8	3
89	89	9	5
90	90	10	4
91	91	11	3
92	92	12	5
93	93	13	4
94	94	14	3
95	95	15	5
96	96	16	4
97	97	17	3
98	98	18	5
99	99	19	4
100	100	20	3
101	1	2	5
102	2	3	4
103	3	3	3
104	4	2	5
105	1	1	5
106	7	7	7

Рис. 7.7 – Результат виконання процедури RateRecipe

7.2.8 Процедура AddNewUser

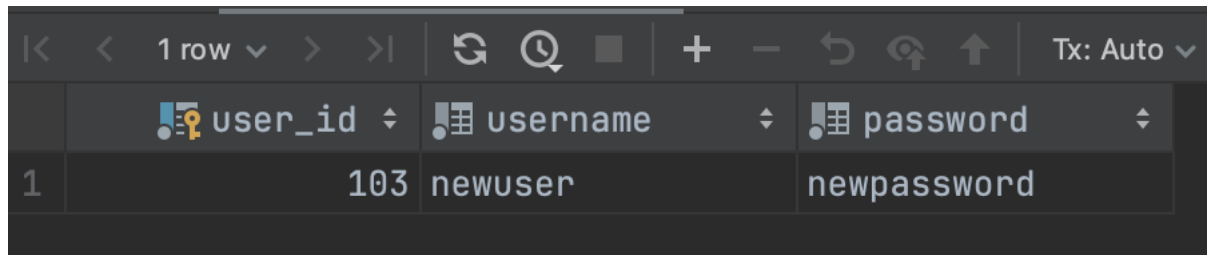
Ця процедура, названа AddNewUser, призначена для додавання нового користувача до таблиці users в базі даних. Результати виконання функції на рис. 7.8

```
DELIMITER //
CREATE PROCEDURE AddNewUser(IN username VARCHAR(50), IN
password VARCHAR(255))
BEGIN
    INSERT INTO users (username, password) VALUES (username, password);
END;
//
```

```
DELIMITER ;
```

```
CALL AddNewUser('newuser', 'newpassword');
```

```
SELECT * FROM users WHERE username = 'newuser';
```



	user_id	username	password
1	103	newuser	newpassword

Рис. 7.8 – Результат виконання процедури AddNewUser

7.2.9 Функція CheckIngredientUsage

Ця функція, названа CheckIngredientUsage, призначена для підрахунку, скільки разів конкретний інгредієнт використовується у рецептах, що зберігаються в таблиці recipe_ingredients бази даних. Результат виконання функції на рис. 7.9.

```
DELIMITER //
```

```
CREATE FUNCTION CheckIngredientUsage(ingredientID INT) RETURNS INT
BEGIN
```

```
    DECLARE usageCount INT;
```

```
    SELECT COUNT(*) INTO usageCount FROM recipe_ingredients WHERE
ingredient_id = ingredientID;
```

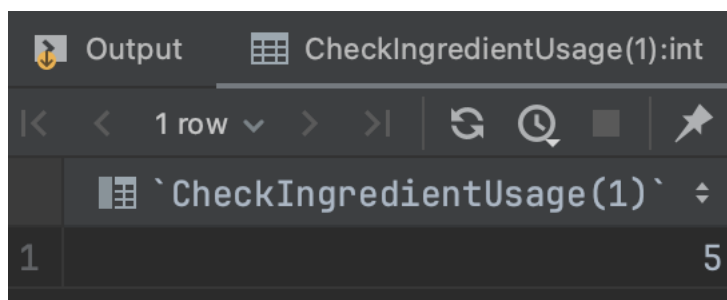
```
    RETURN usageCount;
```

```
END;
```

```
//
```

```
DELIMITER ;
```

```
SELECT CheckIngredientUsage(1);
```



The screenshot shows a database output window with the title 'Output' and a tab labeled 'CheckIngredientUsage(1):int'. Below the title bar is a toolbar with navigation icons. The main area displays a single row of data. The first column contains the value '1' and the second column contains the value '5'.

1	5

Рис. 7.9 – Результат виконання функції CheckIngredientUsage

7.2.10 Процедура UpdateUserProfile

Ця процедура, названа UpdateUserProfile, призначена для оновлення інформації про користувача в таблиці user_profiles бази даних. Результат виконання процедури на рис. 7.10.

DELIMITER //

```
CREATE PROCEDURE UpdateUserProfile(IN userID INT, IN fullName
VARCHAR(100), IN email VARCHAR(100), IN profilePicURL
VARCHAR(255), IN bio TEXT)
```

BEGIN

```
    UPDATE user_profiles SET full_name = fullName, email = email,
    profile_picture_url = profilePicURL, bio = bio WHERE user_id = userID;
```

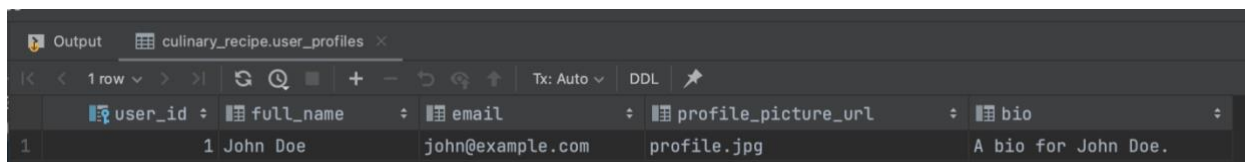
END;

//

DELIMITER ;

```
CALL UpdateUserProfile(1, 'John Doe', 'john@example.com', 'profile.jpg', 'A bio
for John Doe.');
```

```
SELECT * FROM user_profiles WHERE user_id = 1;
```



The screenshot shows a database output window with the title 'Output' and a tab labeled 'culinary_recipe.user_profiles'. Below the title bar is a toolbar with navigation icons. The main area displays a single row of data with five columns: user_id, full_name, email, profile_picture_url, and bio. The values in the row are 1, John Doe, john@example.com, profile.jpg, and A bio for John Doe.

	user_id	full_name	email	profile_picture_url	bio
1	1	John Doe	john@example.com	profile.jpg	A bio for John Doe.

Рис. 7.10 – Результат виконання процедури UpdateUserProfile

7.3 Тригери

7.3.1 Тригер BeforeInsertRecipe

Цей тригер, позначений як BeforeInsertRecipe, виконується перед вставкою нового запису в таблицю recipes. Основна мета цього тригера - перевірити наявність діючого типу страви у таблиці dish_types перед вставкою нового рецепту. Результати вставки нових заяв для різних ситуацій на рис. 7.11, рис. 7.12

1st trigger

```
DROP TRIGGER IF EXISTS BeforeInsertRecipe;
```

```
DELIMITER //
```

```
CREATE TRIGGER BeforeInsertRecipe
```

```
BEFORE INSERT ON recipes
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    DECLARE dishTypeExists INT;
```

```
    SELECT COUNT(*)
```

```
    INTO dishTypeExists
```

```
    FROM dish_types
```

```
    WHERE dish_type_id = NEW.dish_type_id;
```

```
    IF dishTypeExists = 0 THEN
```

```
        SIGNAL SQLSTATE '45000'
```

```
        SET MESSAGE_TEXT = 'Dish type does not exist.';
```

```
    END IF;
```

```
END;
```

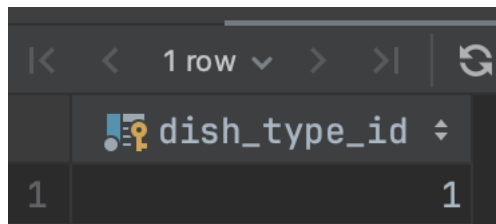
```
//
```

```
DELIMITER ;
```

```
SELECT dish_type_id FROM dish_types WHERE dish_type_name = 'Italian Pasta';
```

```
INSERT INTO recipes (recipe_name, dish_type_id, calories) VALUES ('Test Recipe', 1, 100);
```

```
INSERT INTO recipes (recipe_name, dish_type_id, calories)
VALUES ('Test Recipe', 120, 100);
```



1 row	
	dish_type_id
1	1

Рис. 7.11 – Результат спрацювання BeforeInsertRecipe тригера на перевірку діючого рецепту

```
[2023-12-27 00:23:43] [45000][1644] Dish type does not exist.
[2023-12-27 00:23:43] [HY000][1644] Dish type does not exist.
```

Рис. 7.12 – Тригер BeforeInsertRecipe при перевірці видав текстове повідомлення, що такого типу страви не існує

7.3.2 Тригер UpdateAverageRating

Цей тригер, позначений як UpdateAverageRating, спрямований на підтримку оновлення середнього рейтингу рецепту в таблиці рецептів на основі відгуків користувачів, що були додані у таблицю рейтингів користувачів. Результати виконання тригера в різних ситуаціях на рис. 7.16, рис. 7.13.

2nd trigger

```
DROP TRIGGER IF EXISTS UpdateAverageRating;
```

```
ALTER TABLE recipes
```

```
ADD COLUMN avg_rating DECIMAL(3,2);
```


DELIMITER //

```
CREATE TRIGGER UpdateAverageRating
AFTER INSERT ON user_ratings
FOR EACH ROW
BEGIN
    DECLARE totalRating DECIMAL(5,2);
    DECLARE numberOfRatings INT;

    SELECT SUM(rating), COUNT(*)
    INTO totalRating, numberOfRatings
    FROM user_ratings
    WHERE recipe_id = NEW.recipe_id;

    UPDATE recipes
    SET avg_rating = totalRating / numberOfRatings
    WHERE recipe_id = NEW.recipe_id;
END;
```

//

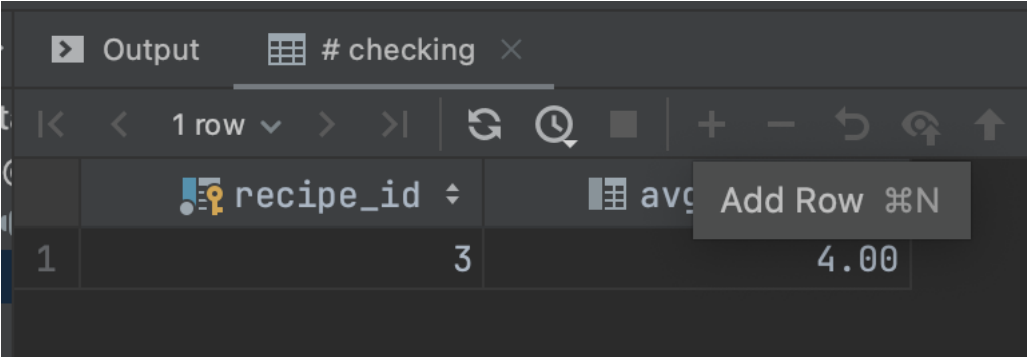
DELIMITER ;

checking

```
SELECT recipe_id, avg_rating FROM recipes WHERE recipe_id = 3;
INSERT INTO user_ratings (user_id, recipe_id, rating) VALUES (1, 2, 5);
INSERT INTO user_ratings (user_id, recipe_id, rating) VALUES (2, 3, 4);
SELECT recipe_id, avg_rating FROM recipes WHERE recipe_id = 3;
INSERT INTO user_ratings (user_id, recipe_id, rating) VALUES (3, 3, 3);
```

```
INSERT INTO user_ratings (user_id, recipe_id, rating) VALUES (4, 2, 5);
```

```
DELIMITER //
```



	recipe_id	avg
1	3	4.00

Рис. 7.13 – Результат тригера UpdateAverageRating, який оновив середній рейтинг рецепту в таблиці

7.3.3 Тригер PreventDeleteRatedRecipe

Цей тригер, позначений як PreventDeleteRatedRecipe, призначений для запобігання видаленню рецепту, якщо на нього є вже існуючі рейтинги користувачів у таблиці user_ratings. Результати виконання тригера на рис. 7.14 та 7.15.

3rd trigger

```
DROP TRIGGER IF EXISTS PreventDeleteRatedRecipe;
```

```
CREATE TRIGGER PreventDeleteRatedRecipe
```

```
BEFORE DELETE ON recipes
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    DECLARE ratingCount INT;
```

```
    SELECT COUNT(*)
```

```
    INTO ratingCount
```

```
    FROM user_ratings
```

```
    WHERE recipe_id = OLD.recipe_id;
```

```

IF ratingCount > 0 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Cannot delete recipe with existing user ratings.';
END IF;
END;

//
DELIMITER ;
# checking
DELETE FROM recipes WHERE recipe_id = 1;

```

```

[2023-12-27 00:34:50] [45000][1644] Cannot delete recipe with existing user ratings.
[2023-12-27 00:34:50] [HY000][1644] Cannot delete recipe with existing user ratings.

```

Рис. 7.14 – Результат тригера PreventDeleteRatedRecipe, який заборонив видаляти рецепт з наявним рейтингом від користувачів

```

culinary_recipe> DELETE FROM recipes WHERE recipe_id = 101
[2023-12-27 00:36:39] completed in 14 ms

```

Рис. 7.15 – Результат тригера PreventDeleteRatedRecipe, який успішно видалив рецепт без наявного рейтингу від користувача

7.3.4 Тригер PreventUpdateDishTypeForFavorites

Цей тригер, позначений як PreventUpdateDishTypeForFavorites, призначений для запобігання оновленню типу страви для рецепту, якщо на цей рецепт є позначки "улюблені" в таблиці user_favorites. Результат виконання тригера на рис. 7.16

4th trigger

```

DROP TRIGGER IF EXISTS PreventUpdateDishTypeForFavorites;
DELIMITER //

```

```

CREATE TRIGGER PreventUpdateDishTypeForFavorites
BEFORE UPDATE ON recipes
FOR EACH ROW
BEGIN
    DECLARE favoriteCount INT;

    SELECT COUNT(*)
    INTO favoriteCount
    FROM user_favorites
    WHERE recipe_id = OLD.recipe_id;

    IF favoriteCount > 0 AND OLD.dish_type_id != NEW.dish_type_id THEN
        SIGNAL SQLSTATE '45000'

        SET MESSAGE_TEXT = 'Cannot update dish type for a recipe that has user
favorites.';
    END IF;
END;
//
DELIMITER ;
INSERT INTO user_favorites (user_id, recipe_id) VALUES (1, 71);
UPDATE recipes SET dish_type_id = 2 WHERE recipe_id = 71;

```

```

culinary_recipe> UPDATE recipes SET dish_type_id = 2 WHERE recipe_id = 71
[2023-12-27 00:41:09] [45000][1644] Cannot update dish type for a recipe that has user favorites.
[2023-12-27 00:41:09] [HY000][1644] Cannot update dish type for a recipe that has user favorites.

```

Рис. 7.16 – Результат тригера PreventUpdateDishTypeForFavorites, який заборонив оновлювати дані рецепту через вподобання користувача

7.3.5 Тригер UpdateCommentCount

Цей тригер, позначений як UpdateCommentCount, призначений для оновлення лічильника кількості коментарів для рецепту в таблиці recipes після вставки нового коментаря в таблицю. Результати виконання тригера на рис.

7.17

5th trigger

```
DROP TRIGGER IF EXISTS UpdateCommentCount;
```

```
ALTER TABLE recipes ADD COLUMN comment_count INT DEFAULT 0;
```

```
DELIMITER //
```

```
CREATE TRIGGER UpdateCommentCount
```

```
AFTER INSERT ON recipe_comments
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    UPDATE recipes
```

```
    SET comment_count = comment_count + 1
```

```
    WHERE recipe_id = NEW.recipe_id;
```

```
END;
```

```
//
```

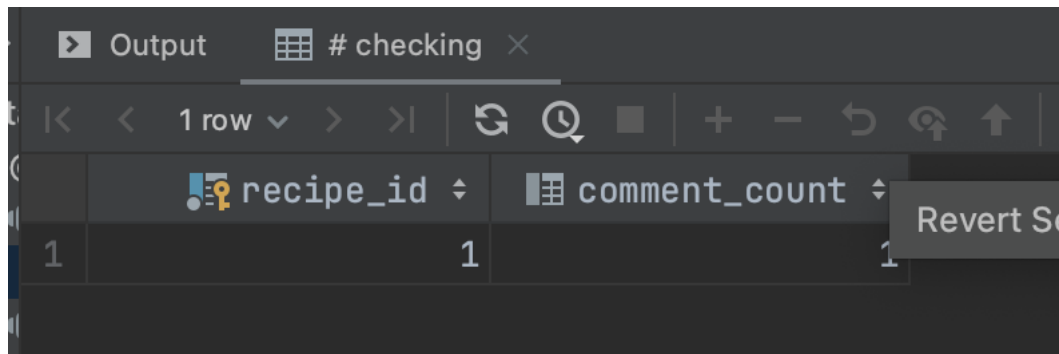
```
DELIMITER ;
```

checking

```
SELECT recipe_id, comment_count FROM recipes WHERE recipe_id = 1;
```

```
INSERT INTO recipe_comments (recipe_id, user_id, comment_text) VALUES (1,  
22, 'Test comment');
```

```
SELECT recipe_id, comment_count FROM recipes WHERE recipe_id = 1;
```



	recipe_id	comment_count
1	1	1

Рис. 7.17 – Результат тригера UpdateCommentCount, який оновлює лічильник коментарів

7.3.6 Тригер EnsureUserExistsBeforeComment

Цей тригер, позначений як EnsureUserExistsBeforeComment, призначений для перевірки наявності користувача перед вставкою нового коментаря в таблицю коментарів. Результати виконання тригера на рис. 7.18

```
DROP TRIGGER IF EXISTS EnsureUserExistsBeforeComment;
```

```
DELIMITER //
```

```
CREATE TRIGGER EnsureUserExistsBeforeComment
```

```
BEFORE INSERT ON recipe_comments
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    DECLARE userExists INT;
```

```
    SELECT COUNT(*)
```

```
    INTO userExists
```

```
    FROM users
```

```
    WHERE user_id = NEW.user_id;
```

```
    IF userExists = 0 THEN
```

```
        SIGNAL SQLSTATE '45000'
```

```
        SET MESSAGE_TEXT = 'Cannot add comment: User does not exist.';
```

```

    END IF;

END;

//

DELIMITER ;

INSERT INTO recipe_comments (recipe_id, user_id, comment_text) VALUES (3,
120, 'Sample comment');

INSERT INTO recipe_comments (recipe_id, user_id, comment_text) VALUES (4,
100, 'Sample comment');

```

```

[2023-12-27 00:52:50] [45000][1644] Cannot add comment: User does not exist.
[2023-12-27 00:52:50] [HY000][1644] Cannot add comment: User does not exist.

```

Рис. 7.18 – Результат тригера EnsureUserExistsBeforeComment, який показує, що користувач не існує

7.3.7 Тригер CheckIngredientValidity

Цей тригер, позначений як CheckIngredientValidity, призначений для перевірки наявності інгредієнта перед вставкою нового запису в таблицю recipe_ingredients. Результати виконання тригера на рис. 7.19, 7.20

#7th trigger

```

DROP TRIGGER IF EXISTS CheckIngredientValidity;

DELIMITER //

CREATE TRIGGER CheckIngredientValidity
BEFORE INSERT ON recipe_ingredients
FOR EACH ROW
BEGIN
    DECLARE ingredientExists INT;

    SELECT COUNT(*)
    INTO ingredientExists
    FROM ingredients

```

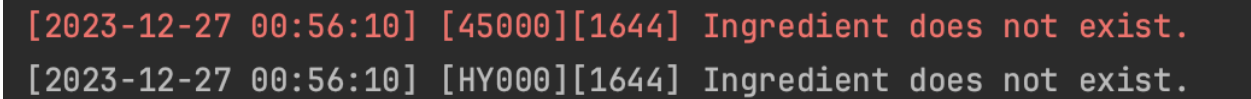
```

WHERE ingredient_id = NEW.ingredient_id;

IF ingredientExists = 0 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Ingredient does not exist.';
END IF;
END;
DELIMITER ;

# checking
SELECT 3 FROM ingredients; -- To find an existing ID.
INSERT INTO recipe_ingredients (recipe_id, ingredient_id, quantity) VALUES
(1, 1, '1 cup');
INSERT INTO recipe_ingredients (recipe_id, ingredient_id, quantity) VALUES
(1, 456, '1 cup');

```

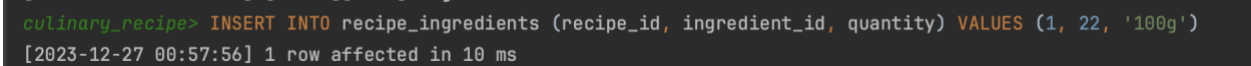


```

[2023-12-27 00:56:10] [45000][1644] Ingredient does not exist.
[2023-12-27 00:56:10] [HY000][1644] Ingredient does not exist.

```

Рис. 7.19 – Результат тригера CheckIngredientValidity, який показує, що інгредієнту не існує



```

culinary_recipe> INSERT INTO recipe_ingredients (recipe_id, ingredient_id, quantity) VALUES (1, 22, '100g')
[2023-12-27 00:57:56] 1 row affected in 10 ms

```

Рис. 7.20 – Результат тригера CheckIngredientValidity, який показує, що при існуванні інгредієнту вставка працює

7.3.8 Тригер PreventDeletionOfIngredientInUse

Цей тригер, позначений як PreventDeletionOfIngredientInUse, створений для запобігання видаленню інгредієнта, який використовується в рецептах

Результати виконання тригера на рис. 7.21, 7.22

#8th trigger

```
DROP TRIGGER IF EXISTS PreventDeletionOfIngredientInUse;
```


DELIMITER //

CREATE TRIGGER PreventDeletionOfIngredientInUse

BEFORE DELETE ON ingredients

FOR EACH ROW

BEGIN

 DECLARE inUseCount INT;

 SELECT COUNT(*)

 INTO inUseCount

 FROM recipe_ingredients

 WHERE ingredient_id = OLD.ingredient_id;

 IF inUseCount > 0 THEN

 SIGNAL SQLSTATE '45000'

 SET MESSAGE_TEXT = 'Cannot delete ingredient: It is used in recipes.';

 END IF;

END;

//

DELIMITER ;

SELECT ingredient_id

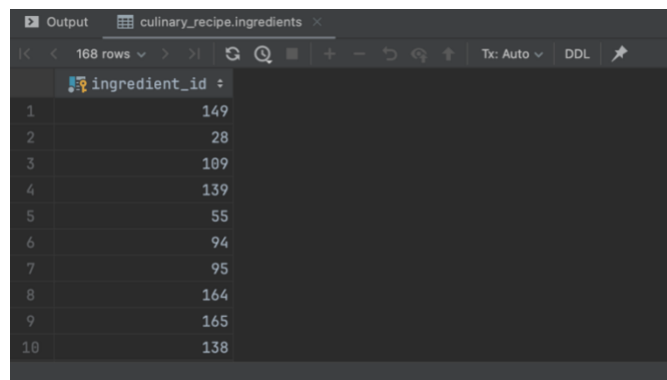
FROM ingredients

WHERE ingredient_id NOT IN (SELECT 1 FROM recipe_ingredients);

```
SELECT ingredient_id FROM recipe_ingredients LIMIT 1;
DELETE FROM ingredients WHERE ingredient_id = 1;
```

```
culinary_recipe> DELETE FROM ingredients WHERE ingredient_id = 1
[2023-12-27 01:01:53] [45000][1644] Cannot delete ingredient: It is used in recipes.
[2023-12-27 01:01:53] [HY000][1644] Cannot delete ingredient: It is used in recipes.
```

Рис. 7.21 – Результат тригера PreventDeletionOfIngredientInUse, який показує, що не можна видалити інгредієнт, який присутній в рецептах



The screenshot shows a database output window titled 'culinary_recipe.ingredients'. It displays a table with two columns: 'ingredient_id' and a list of recipe IDs. The first 10 rows are visible, showing recipe IDs 149, 28, 109, 139, 55, 94, 95, 164, 165, and 138.

	ingredient_id
1	149
2	28
3	109
4	139
5	55
6	94
7	95
8	164
9	165
10	138

Рис. 7.22 – Список рецептів, де присутній цей інгредієнт

7.4 Представлення

7.4.1 Представлення RecipeDetails

В SQL-скрипті представлення RecipeDetails призначене для надання всебічного та структурованого огляду рецептів, їх типів, калорійності та інгредієнтів. Вигляд представлення на рис. 7.23.

1st view

```
DROP VIEW IF EXISTS RecipeDetails;
```

```
CREATE VIEW RecipeDetails AS
```

```
SELECT
```

```
    r.recipe_id,
```

```
    r.recipe_name,
```

```
    dt.dish_type_name,
```

```
    r.calories,
```

```

GROUP_CONCAT(i.ingredient_name ORDER BY i.ingredient_name ASC
SEPARATOR ', ') AS ingredients
FROM
    recipes r
JOIN
    dish_types dt ON r.dish_type_id = dt.dish_type_id
JOIN
    recipe_ingredients ri ON r.recipe_id = ri.recipe_id
JOIN
    ingredients i ON ri.ingredient_id = i.ingredient_id
GROUP BY
    r.recipe_id, r.recipe_name, dt.dish_type_name, r.calories;

SELECT * FROM RecipeDetails;

```

recipe_id	recipe_name	dish_type_name	calories	ingredients
1	Classic Margherita Pizza	Italian Pasta	400	Flour, Flour, Flour, Flour, Honeydew, Salt, Sugar, Thyme
2	Spicy Thai Green Curry	Chinese Dim Sum	400	Butter, Garlic, Olive Oil
3	Traditional Beef Stroganoff	Indian Curry	500	Basil, Onion, Tomatoes
4	Vegetarian Lasagna	Japanese Sushi	350	Black Pepper, Cumin, Parsley
5	Chicken Tikka Masala	Mexican Taco	450	Cinnamon, Ginger, Nutmeg
6	Quinoa and Black Bean Salad	French Pastry	310	Cardamom, Cloves, Turmeric
7	Moroccan Chicken Tagine	Thai Curry	420	Chili Powder, Oregano, Paprika
8	French Onion Soup	Spanish Paella	280	Bay Leaves, Rosemary, Thyme
9	Sushi Platter Assortment	Lebanese Mezze	370	Coriander, Saffron, Vanilla
10	Mexican Chorizo Tacos	American Burger	390	Almonds, Peanuts, Walnuts
11	New York Cheesecake	Vietnamese Pho	450	Beef, Chicken, Pork
12	Korean Bibimbap	Moroccan Tagine	430	Lamb, Salmon, Tuna
13	Spanish Seafood Paella	Greek Gyro	490	Crab, Lobster, Shrimp
14	Vegan Buddha Bowl	Turkish Kebab	360	Cheese, Milk, Yogurt
15	Grilled Salmon with Dill Sauce	Russian Borscht	410	Cream, Eggs, Potatoes
16	Italian Tiramisu	Brazilian Churrasco	420	Broccoli, Carrots, Spinach
17	Indian Vegetable Samosa	Korean BBQ	250	Cabbage, Cucumber, Lettuce
18	BBQ Pulled Pork Sandwich	Ethiopian Injera	520	Bell Pepper, Eggplant, Zucchini
19	Greek Moussaka	Portuguese Bacalhau	480	Apples, Bananas, Oranges
20	Japanese Ramen Bowl	Malaysian Nasi Lemak	550	Grapes, Lemons, Strawberries
21	Chocolate Lava Cake	Indonesian Satay	390	Blueberries, Peaches, Pineapple
22	Smoked Texas Brisket	Australian Meat Pie	580	Mango, Pasta, Rice
23	Turkish Baklava	Canadian Poutine	330	Bread, Oats, Quinoa
24	Cantonese Dim Sum Platter	Argentinian Empanada	380	Chocolate, Honey, Maple Syrup
25	Brazilian Feijoada	Nigerian Jollof Rice	470	Cocoa Powder, Coconut, Lentils
26	Peruvian Ceviche	Israeli Falafel	310	Black Beans, Chickpeas, Kidney Beans
27	American Apple Pie	Belgian Waffle	410	Green Peas, Soybeans, Tofu
28	German Schnitzel	Austrian Schnitzel	490	Barley, Corn, Tempeh
29	Ethiopian Doro Wat	Jamaican Jerk Chicken	440	Buckwheat, Couscous, Rye
30	Lebanese Falafel Pita	Pakistani Biryani	400	Beetroot, Polenta, Sweet Potato

Рис. 7.23 – Вид представления RecipeDetails

7.4.2 Представлення UserFavoriteRecipes

Представлення UserFavoriteRecipes у SQL-скрипті створене для відображення улюблених рецептів кожного користувача. Вигляд представлення на рис. 7.24.

2nd view

```
DROP VIEW IF EXISTS UserFavoriteRecipes;
```

```
CREATE VIEW UserFavoriteRecipes AS
```

```
SELECT
```

```
    u.user_id,
```

```
    u.username,
```

```
    r.recipe_name
```

```
FROM
```

```
    users u
```


```
JOIN
```

```
    user_favorites uf ON u.user_id = uf.user_id
```

```
JOIN
```

```
    recipes r ON uf.recipe_id = r.recipe_id;
```

```
SELECT * FROM UserFavoriteRecipes;
```



	user_id	username	recipe_name
1	1	alexsmith	Classic Margherita Pizza
2	1	alexsmith	Test Recipe
3	1	alexsmith	Test Recipe
4	2	john DOE	Spicy Thai Green Curry
5	3	emilyjones	Traditional Beef Stroganoff
6	4	sarahlee	Vegetarian Lasagna
7	5	michaelbrown	Chicken Tikka Masala
8	6	lauragreen	Quinoa and Black Bean Salad
9	7	davidmiller	Moroccan Chicken Tagine
10	8	chloewhite	French Onion Soup
11	9	danielgarcia	Sushi Platter Assortment
12	10	sophiamartinez	Mexican Chorizo Tacos
13	11	jamesjohnson	New York Cheesecake
14	12	isabelladavis	Korean Bibimbap
15	13	oliverrodriguez	Spanish Seafood Paella
16	14	emmalopez	Vegan Buddha Bowl
17	15	noahhernandez	Grilled Salmon with Dill Sauce
18	16	amandawilson	Italian Tiramisu
19	17	briananderson	Indian Vegetable Samosa
20	18	jessicathomas	BBQ Pulled Pork Sandwich
21	19	kevinjackson	Greek Moussaka
22	20	sarahmoore	Japanese Ramen Bowl
23	21	chriswhite	Chocolate Lava Cake
24	22	megantaylor	Smoked Texas Brisket
25	23	mikeroobinson	Turkish Baklava
26	24	amywalker	Cantonese Dim Sum Platter
27	25	justinlee	Brazilian Feijoada
28	26	caroldavis	Peruvian Ceviche
29	27	garymartinez	American Apple Pie
30	28	lisamartin	German Schnitzel

rows retrieved starting from 1 in 73 ms (execution: 13 ms, fetching: 60 ms)

Рис. 7.24 – Вигляд представлення UserFavoriteRecipes

7.4.3 Представлення RecipeRatingsAndComments

Представлення RecipeRatingsAndComments у SQL-скрипті створене для збору та відображення оцінок та коментарів, пов'язаних з кожним рецептом. Вигляд представлення на рис. 7.25.

3rd view

DROP VIEW IF EXISTS RecipeRatingsAndComments;

CREATE VIEW RecipeRatingsAndComments AS

SELECT

r.recipe_id,

r.recipe_name,

AVG(ur.rating) AS average_rating,

COUNT(DISTINCT rc.comment_id) AS comment_count

FROM

recipes r

LEFT JOIN

user_ratings ur ON r.recipe_id = ur.recipe_id

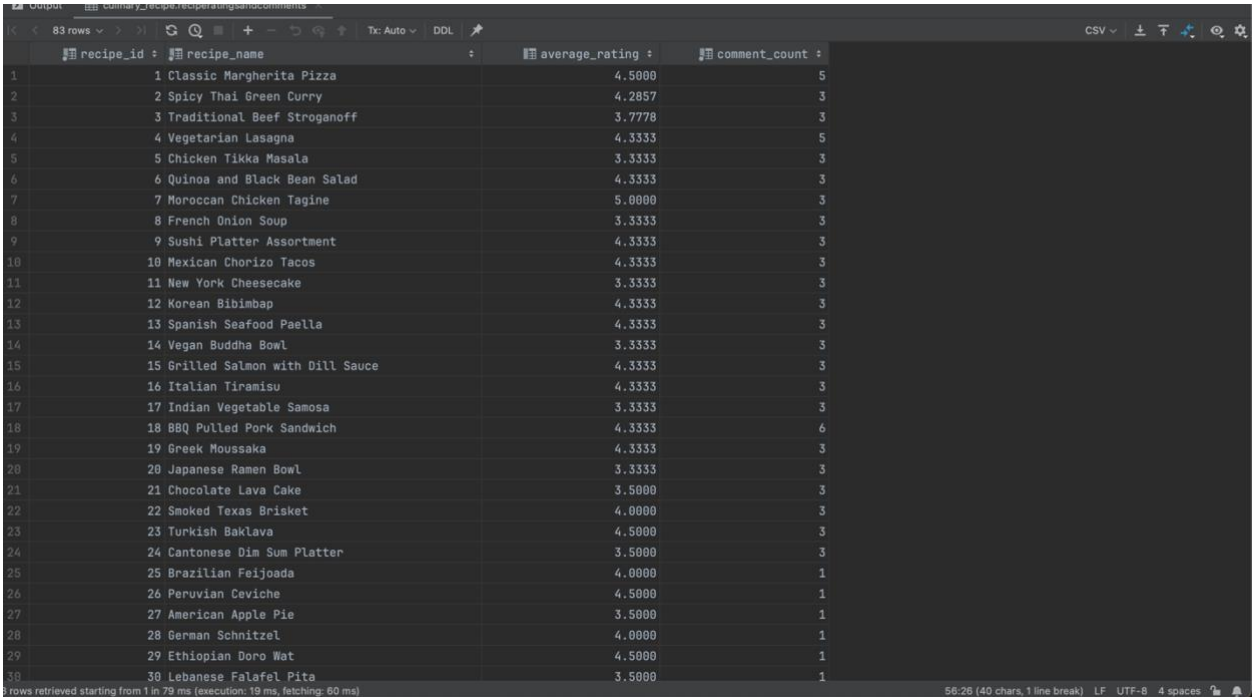
LEFT JOIN

```

recipe_comments rc ON r.recipe_id = rc.recipe_id
GROUP BY
    r.recipe_id, r.recipe_name;

SELECT * FROM RecipeRatingsAndComments;

```



recipe_id	recipe_name	average_rating	comment_count
1	Classic Margherita Pizza	4.5000	5
2	Spicy Thai Green Curry	4.2857	3
3	Traditional Beef Stroganoff	3.7778	3
4	Vegetarian Lasagna	4.3333	5
5	Chicken Tikka Masala	3.3333	3
6	Quinoa and Black Bean Salad	4.3333	3
7	Moroccan Chicken Tagine	5.0000	3
8	French Onion Soup	3.3333	3
9	Sushi Platter Assortment	4.3333	3
10	Mexican Chorizo Tacos	4.3333	3
11	New York Cheesecake	3.3333	3
12	Korean Bibimbap	4.3333	3
13	Spanish Seafood Paella	4.3333	3
14	Vegan Buddha Bowl	3.3333	3
15	Grilled Salmon with Dill Sauce	4.3333	3
16	Italian Tiramisu	4.3333	3
17	Indian Vegetable Samosa	3.3333	3
18	BBQ Pulled Pork Sandwich	4.3333	6
19	Greek Moussaka	4.3333	3
20	Japanese Ramen Bowl	3.3333	3
21	Chocolate Lava Cake	3.5000	3
22	Smoked Texas Brisket	4.0000	3
23	Turkish Baklava	4.5000	3
24	Cantonese Dim Sum Platter	3.5000	3
25	Brazilian Feijoada	4.0000	1
26	Peruvian Ceviche	4.5000	1
27	American Apple Pie	3.5000	1
28	German Schnitzel	4.0000	1
29	Ethiopian Doro Wat	4.5000	1
30	Lebanese Falafel Pita	3.5000	1

Рис. 7.25 – Вигляд представлення RecipeRatingsAndComments

7.4.4 Представлення DetailedRecipeInstructions

Представлення DetailedRecipeInstructions у вашому SQL-скрипті створене для відображення детальних інструкцій по кроках для кожного рецепту. Вигляд представлення на рис. 7.26.

#4th view

```
DROP VIEW IF EXISTS DetailedRecipeInstructions;
```

```
CREATE VIEW DetailedRecipeInstructions AS
```

```
SELECT
```

```
    r.recipe_id,
```

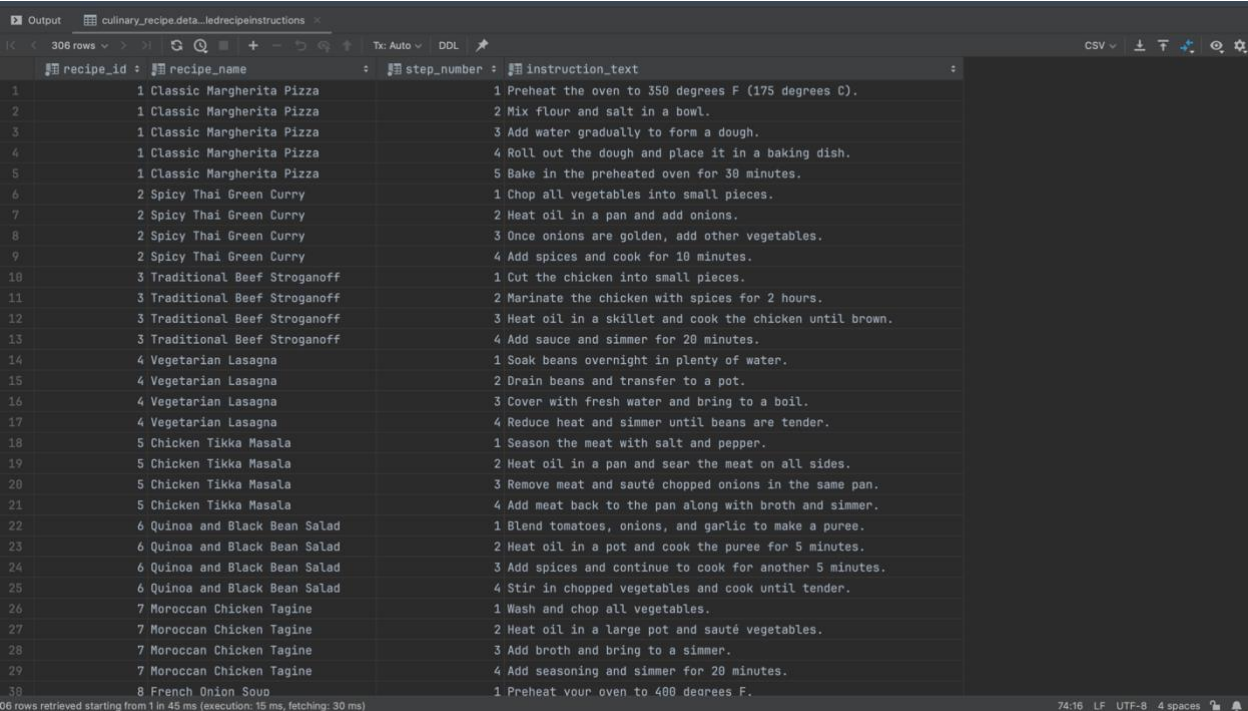
```
    r.recipe_name,
```

```

ins.step_number,
ins.instruction_text
FROM
  recipes r
JOIN
  instructions ins ON r.recipe_id = ins.recipe_id
ORDER BY
  r.recipe_id, ins.step_number;

SELECT * FROM DetailedRecipeInstructions;

```



recipe_id	recipe_name	step_number	instruction_text
1	Classic Margherita Pizza	1	Preheat the oven to 350 degrees F (175 degrees C).
2	Classic Margherita Pizza	2	Mix flour and salt in a bowl.
3	Classic Margherita Pizza	3	Add water gradually to form a dough.
4	Classic Margherita Pizza	4	Roll out the dough and place it in a baking dish.
5	Classic Margherita Pizza	5	Bake in the preheated oven for 30 minutes.
6	Spicy Thai Green Curry	1	Chop all vegetables into small pieces.
7	Spicy Thai Green Curry	2	Heat oil in a pan and add onions.
8	Spicy Thai Green Curry	3	Once onions are golden, add other vegetables.
9	Spicy Thai Green Curry	4	Add spices and cook for 10 minutes.
10	Traditional Beef Stroganoff	1	Cut the chicken into small pieces.
11	Traditional Beef Stroganoff	2	Marinate the chicken with spices for 2 hours.
12	Traditional Beef Stroganoff	3	Heat oil in a skillet and cook the chicken until brown.
13	Traditional Beef Stroganoff	4	Add sauce and simmer for 20 minutes.
14	Vegetarian Lasagna	1	Soak beans overnight in plenty of water.
15	Vegetarian Lasagna	2	Drain beans and transfer to a pot.
16	Vegetarian Lasagna	3	Cover with fresh water and bring to a boil.
17	Vegetarian Lasagna	4	Reduce heat and simmer until beans are tender.
18	Chicken Tikka Masala	1	Season the meat with salt and pepper.
19	Chicken Tikka Masala	2	Heat oil in a pan and sear the meat on all sides.
20	Chicken Tikka Masala	3	Remove meat and sauté chopped onions in the same pan.
21	Chicken Tikka Masala	4	Add meat back to the pan along with broth and simmer.
22	Quinoa and Black Bean Salad	1	Blend tomatoes, onions, and garlic to make a puree.
23	Quinoa and Black Bean Salad	2	Heat oil in a pot and cook the puree for 5 minutes.
24	Quinoa and Black Bean Salad	3	Add spices and continue to cook for another 5 minutes.
25	Quinoa and Black Bean Salad	4	Stir in chopped vegetables and cook until tender.
26	Moroccan Chicken Tagine	1	Wash and chop all vegetables.
27	Moroccan Chicken Tagine	2	Heat oil in a large pot and sauté vegetables.
28	Moroccan Chicken Tagine	3	Add broth and bring to a simmer.
29	Moroccan Chicken Tagine	4	Add seasoning and simmer for 20 minutes.
30	French Onion Soup	1	Preheat your oven to 400 degrees F.

Рис. 7.26 – Вигляд представлення DetailedRecipeInstructions

7.4.5 Представлення RecipeNutritionalInformation

Представлення RecipeNutritionalInformation у вашому SQL-скрипті розроблене для надання інформації про харчову цінність кожного рецепту, включаючи калорії, кількість порцій та розмір порції. Вигляд представлення на рис. 7.27.

#5th view

```
DROP VIEW IF EXISTS RecipeNutritionalInformation;
```

```
CREATE VIEW RecipeNutritionalInformation AS
```

```
SELECT
```

```
    r.recipe_id,
```

```
    r.recipe_name,
```

```
    r.calories,
```

```
    rs.servings,
```

```
    rs.serving_size
```

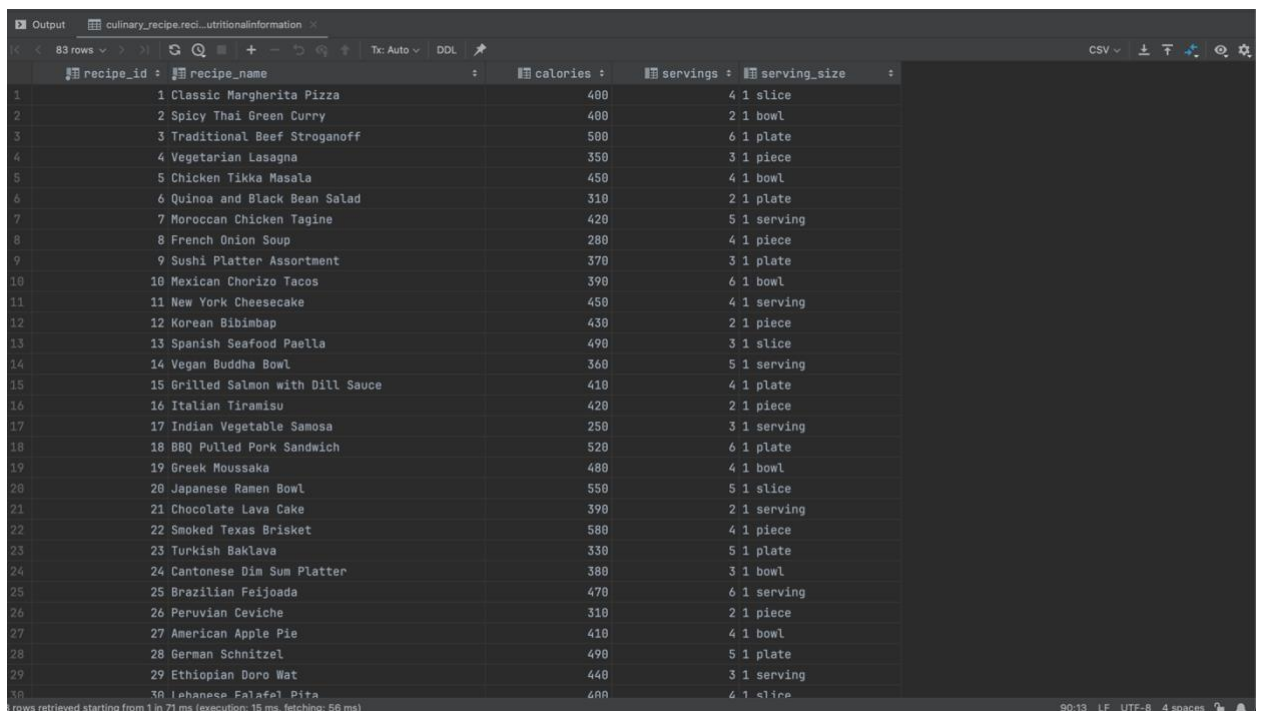
```
FROM
```

```
    recipes r
```

```
LEFT JOIN
```

```
    recipe_servings rs ON r.recipe_id = rs.recipe_id;
```

```
SELECT * FROM RecipeNutritionalInformation;
```



The screenshot shows a database query result with 30 rows. The columns are: recipe_id, recipe_name, calories, servings, and serving_size. The data is as follows:

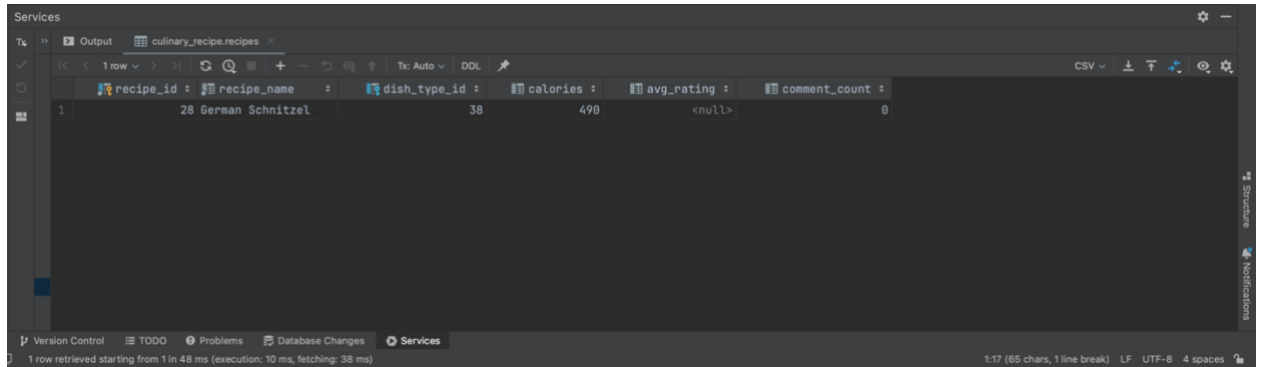
recipe_id	recipe_name	calories	servings	serving_size
1	Classic Margherita Pizza	400	4	1 slice
2	Spicy Thai Green Curry	400	2	1 bowl
3	Traditional Beef Stroganoff	500	6	1 plate
4	Vegetarian Lasagna	350	3	1 piece
5	Chicken Tikka Masala	450	4	1 bowl
6	Quinoa and Black Bean Salad	310	2	1 plate
7	Moroccan Chicken Tagine	420	5	1 serving
8	French Onion Soup	280	4	1 piece
9	Sushi Platter Assortment	370	3	1 plate
10	Mexican Chorizo Tacos	390	6	1 bowl
11	New York Cheesecake	450	4	1 serving
12	Korean Bibimbap	430	2	1 piece
13	Spanish Seafood Paella	490	3	1 slice
14	Vegan Buddha Bowl	360	5	1 serving
15	Grilled Salmon with Dill Sauce	410	4	1 plate
16	Italian Tiramisu	420	2	1 piece
17	Indian Vegetable Samosa	250	3	1 serving
18	BBQ Pulled Pork Sandwich	520	6	1 plate
19	Greek Moussaka	480	4	1 bowl
20	Japanese Ramen Bowl	550	5	1 slice
21	Chocolate Lava Cake	390	2	1 serving
22	Smoked Texas Brisket	580	4	1 piece
23	Turkish Baklava	330	5	1 plate
24	Cantonese Dim Sum Platter	380	3	1 bowl
25	Brazilian Feijoada	470	6	1 serving
26	Peruvian Ceviche	310	2	1 piece
27	American Apple Pie	410	4	1 bowl
28	German Schnitzel	490	5	1 plate
29	Ethiopian Doro Wat	440	3	1 serving
30	Japanese Falafel Pita	400	4	1 slice

Рис. 7.27 – Вигляд представлення RecipeNutritionalInformation

7.5 SQL-запити

1) Запит вибору страв з назвою «German Schnitzel» на початку (результат на рис. 7.28):

```
SELECT * FROM recipes WHERE recipe_name LIKE 'German Schnitzel';
```

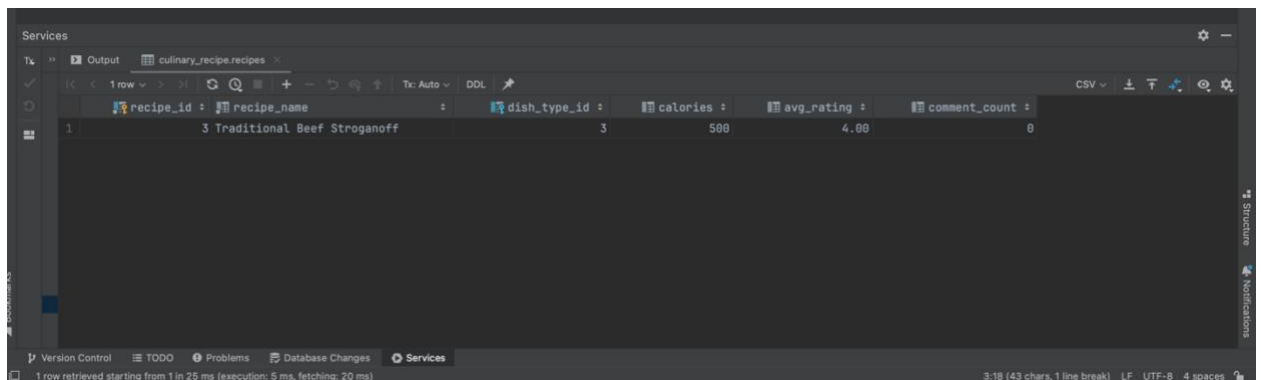


recipe_id	recipe_name	dish_type_id	calories	avg_rating	comment_count
1	28 German Schnitzel	38	490	<null>	0

Рис. 7.28 – Результат виконання запиту 1

2) Запит для вибору рецепту з ідентифікатором 3 (результат на рис. 7.29):

```
SELECT * FROM recipes WHERE recipe_id = 3;
```



recipe_id	recipe_name	dish_type_id	calories	avg_rating	comment_count
1	3 Traditional Beef Stroganoff	3	500	4.00	0

Рис. 7.29 – Результат виконання запиту 2

3) Запит для вибору імен і кількостей інгредієнтів, які використовуються у рецепті з ідентифікатором 57 (результат на рис. 7.30):

```
SELECT ingredient_name, quantity FROM ingredients
JOIN recipe_ingredients ON ingredients.ingredient_id =
recipe_ingredients.ingredient_id
WHERE recipe_ingredients.recipe_id = 57;
```

Services

Output Result 47

3 rows

	ingredient_name	quantity
1	Kiwi	1/2 teaspoon
2	Papaya	3 leaves
3	Cantaloupe	2 tablespoons

Bookmarks

Version Control TODO Problems Database Changes Services

3 rows retrieved starting from 1 in 20 ms (execution: 6 ms, fetching: 14 ms)

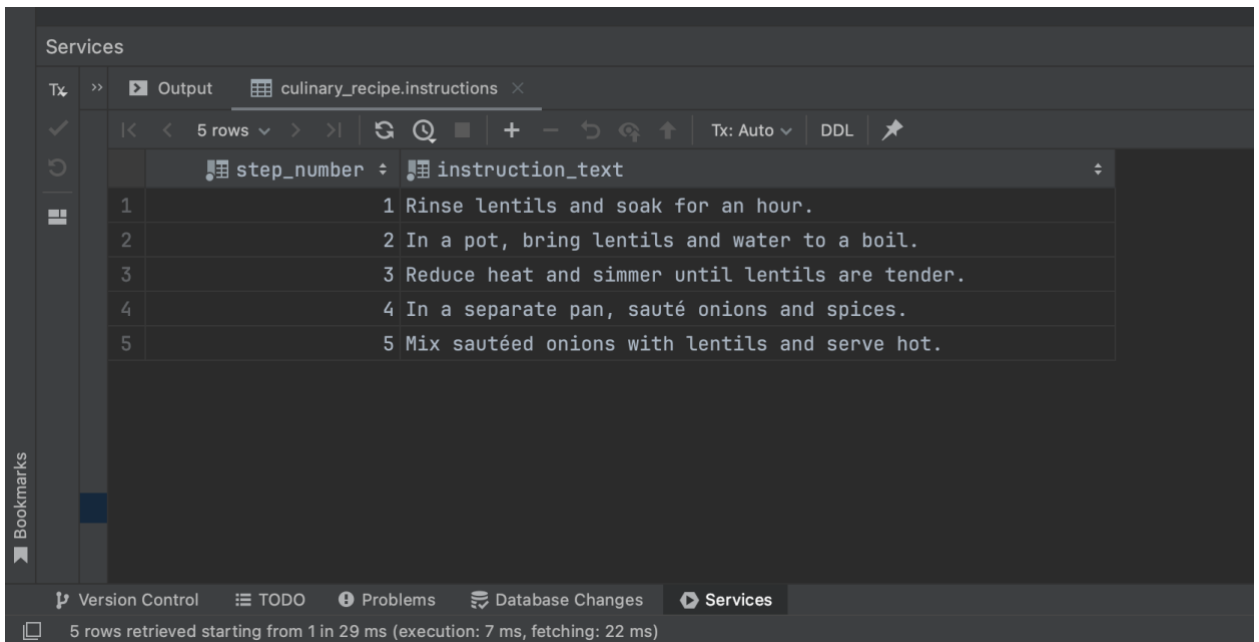
Рис. 7.30 – Результат виконання запиту 3

4) Запит для вибору номерів кроків і тексту інструкцій для рецепту з ідентифікатором 24, впорядкованих за номером кроку (результат на рис. 7.31):

```
SELECT step_number, instruction_text FROM instructions
```

```
WHERE recipe_id = 24
```

```
ORDER BY step_number;
```



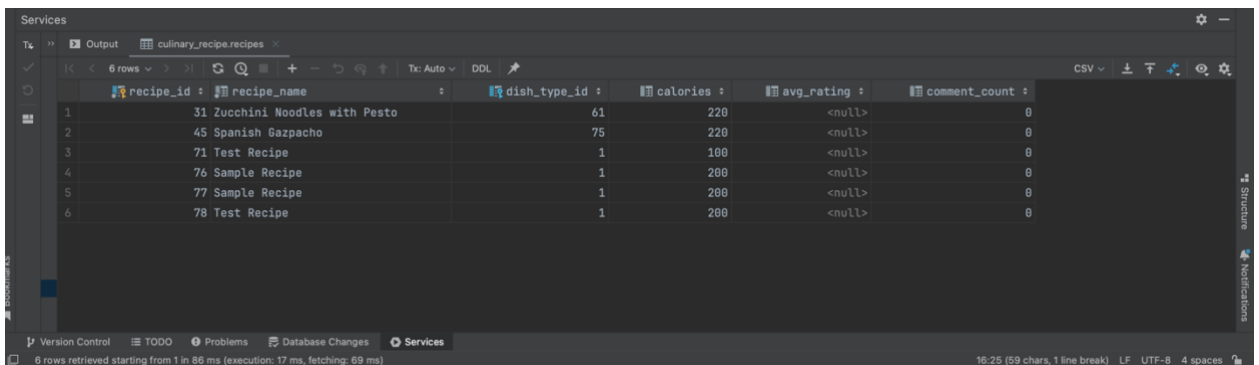
step_number	instruction_text
1	1 Rinse lentils and soak for an hour.
2	2 In a pot, bring lentils and water to a boil.
3	3 Reduce heat and simmer until lentils are tender.
4	4 In a separate pan, sauté onions and spices.
5	5 Mix sautéed onions with lentils and serve hot.

5 rows retrieved starting from 1 in 29 ms (execution: 7 ms, fetching: 22 ms)

Рис. 7.31 – Результат виконання запиту 4

5) Запит для вибору всіх рецептів зі значенням калорій в діапазоні від 100 до 220 (результат на рис. 7.32):

```
SELECT * FROM recipes WHERE calories BETWEEN 100 AND 220;
```



recipe_id	recipe_name	dish_type_id	calories	avg_rating	comment_count
31	Zucchini Noodles with Pesto	61	220	<null>	0
45	Spanish Gazpacho	75	220	<null>	0
71	Test Recipe	1	100	<null>	0
76	Sample Recipe	1	200	<null>	0
77	Sample Recipe	1	200	<null>	0
78	Test Recipe	1	200	<null>	0

6 rows retrieved starting from 1 in 86 ms (execution: 17 ms, fetching: 69 ms)

Рис. 7.32 – Результат виконання запиту 5

6) Запит для отримання унікальних назв страв, які містять інгредієнт "Chocolate" (результат на рис. 7.33):

```
SELECT DISTINCT recipes.recipe_name FROM recipes
JOIN recipe_ingredients ON recipes.recipe_id = recipe_ingredients.recipe_id
JOIN ingredients ON recipe_ingredients.ingredient_id = ingredients.ingredient_id
```

WHERE ingredients.ingredient_name = 'Chocolate';

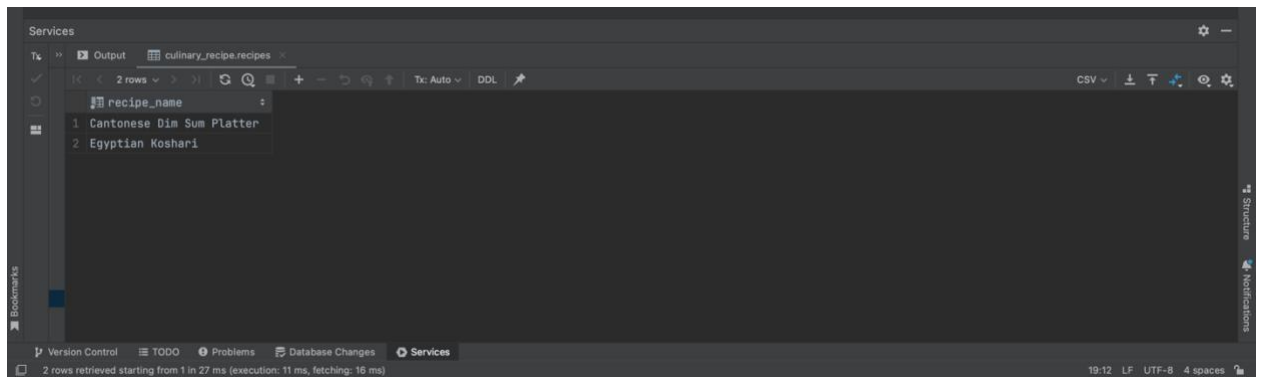


Рис. 7.33 – Результат виконання запиту 6

7) Запит для отримання назв страв та їх середнього рейтингу, відсортованих за спаданням середнього рейтингу та обмежених 10 результатами (результат на рис. 7.34):

```
SELECT recipes.recipe_name, AVG(user_ratings.rating) AS average_rating
FROM recipes
LEFT JOIN user_ratings ON recipes.recipe_id = user_ratings.recipe_id
GROUP BY recipes.recipe_id
ORDER BY average_rating DESC
LIMIT 10;
```

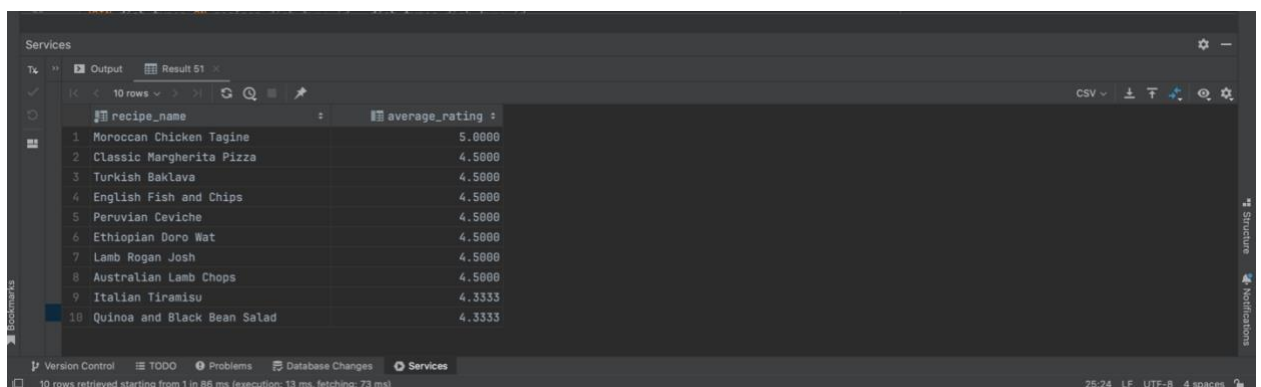


Рис. 7.34 – Результат виконання запиту 7

8) Запит для отримання назв страв та назв типів страв, де тип страви є "Mexican Taco" (результат на рис. 7.35):

```
SELECT recipes.recipe_name, dish_types.dish_type_name
FROM recipes
JOIN dish_types ON recipes.dish_type_id = dish_types.dish_type_id
WHERE dish_types.dish_type_name = 'Mexican Taco';
```

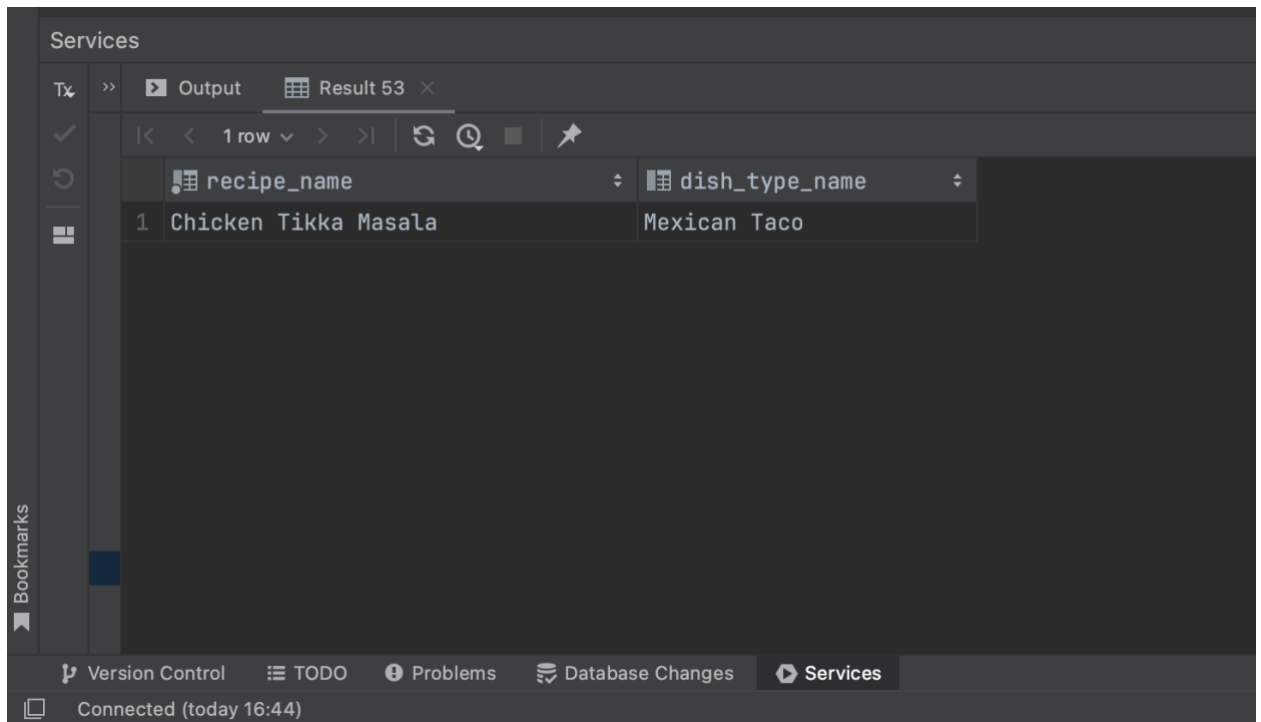
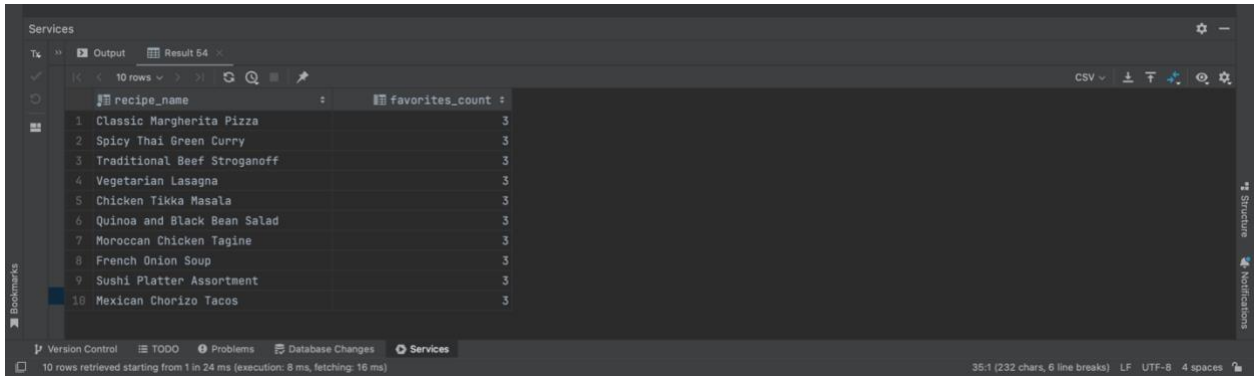


Рис. 7.35 – Результат виконання запиту 8

9) Запит для отримання назв страв та кількості обрань, відсортованих за зменшенням кількості обрань та обмежених 10 результатами (результат на рис. 7.36):

```
SELECT recipes.recipe_name, COUNT(user_favorites.recipe_id) AS
favorites_count
FROM recipes
LEFT JOIN user_favorites ON recipes.recipe_id = user_favorites.recipe_id
GROUP BY recipes.recipe_id
ORDER BY favorites_count DESC
LIMIT 10;
```



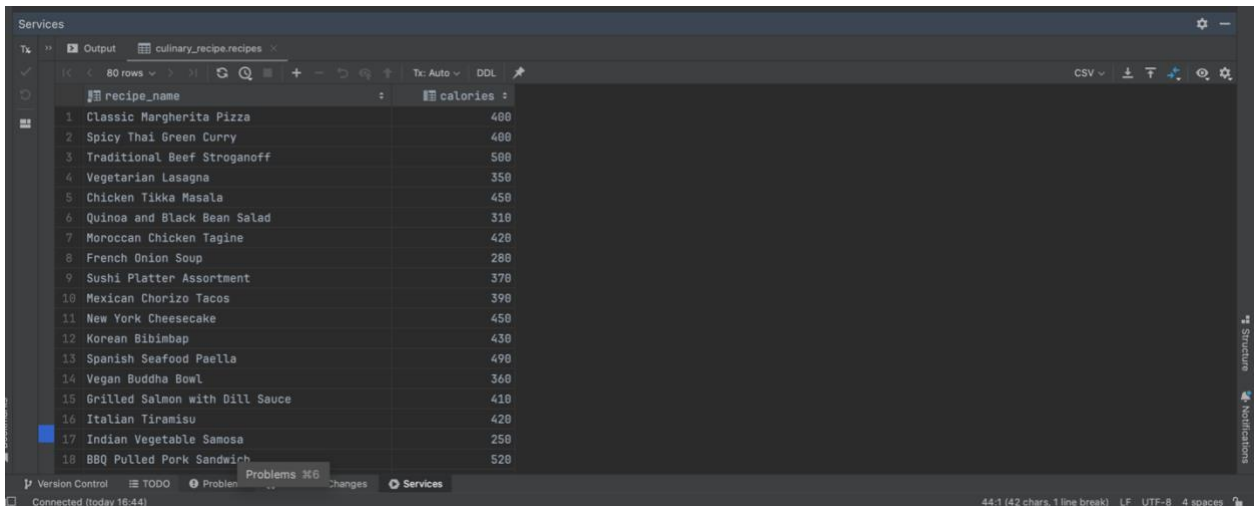
recipe_name	favorites_count
1 Classic Margherita Pizza	3
2 Spicy Thai Green Curry	3
3 Traditional Beef Stroganoff	3
4 Vegetarian Lasagna	3
5 Chicken Tikka Masala	3
6 Quinoa and Black Bean Salad	3
7 Moroccan Chicken Tagine	3
8 French Onion Soup	3
9 Sushi Platter Assortment	3
10 Mexican Chorizo Tacos	3

Рис. 7.36 – Результат виконання запиту 9

10) Запит для вибору назв страв та кількості калорій для всіх рецептів (результат на рис. 7.37):

```
SELECT recipe_name, calories
```

```
FROM recipes;
```



recipe_name	calories
1 Classic Margherita Pizza	400
2 Spicy Thai Green Curry	400
3 Traditional Beef Stroganoff	500
4 Vegetarian Lasagna	350
5 Chicken Tikka Masala	450
6 Quinoa and Black Bean Salad	310
7 Moroccan Chicken Tagine	420
8 French Onion Soup	280
9 Sushi Platter Assortment	370
10 Mexican Chorizo Tacos	390
11 New York Cheesecake	450
12 Korean Bibimbap	430
13 Spanish Seafood Paella	490
14 Vegan Buddha Bowl	360
15 Grilled Salmon with Dill Sauce	410
16 Italian Tiramisu	420
17 Indian Vegetable Samosa	250
18 BBQ Pulled Pork Sandwich	520

Рис. 7.37 – Результат виконання запиту 10

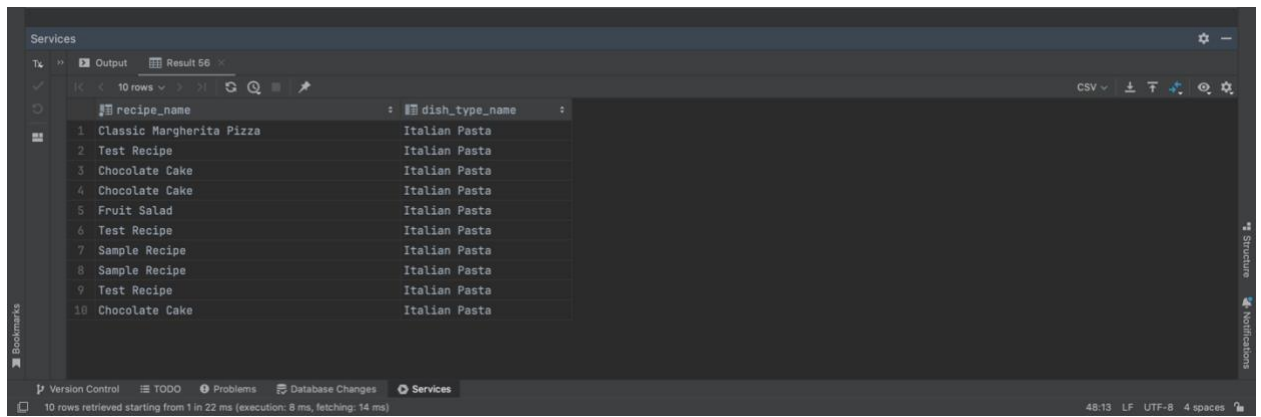
11) Запит для вибору назв страв та їх типу для рецептів, де тип страви є "Italian Pasta" (результат на рис. 7.38):

```
SELECT recipe_name, dish_type_name
```

```
FROM recipes
```

```
JOIN dish_types ON recipes.dish_type_id = dish_types.dish_type_id
```

```
WHERE dish_types.dish_type_name = 'Italian Pasta';
```

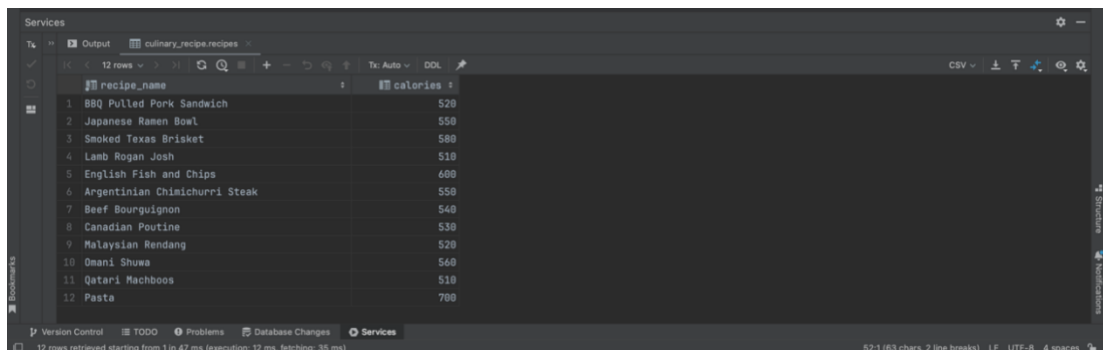


recipe_name	dish_type_name
1 Classic Margherita Pizza	Italian Pasta
2 Test Recipe	Italian Pasta
3 Chocolate Cake	Italian Pasta
4 Chocolate Cake	Italian Pasta
5 Fruit Salad	Italian Pasta
6 Test Recipe	Italian Pasta
7 Sample Recipe	Italian Pasta
8 Sample Recipe	Italian Pasta
9 Test Recipe	Italian Pasta
10 Chocolate Cake	Italian Pasta

Рис. 7.38 – Результат виконання запиту 11

12) Запит для вибору назв страв та кількості калорій для рецептів, де кількість калорій більше 500 (результат на рис. 7.39):

```
SELECT recipe_name, calories
FROM recipes
WHERE calories > 500;
```



recipe_name	calories
1 BBQ Pulled Pork Sandwich	520
2 Japanese Ramen Bowl	550
3 Smoked Texas Brisket	580
4 Lamb Rogan Josh	510
5 English Fish and Chips	680
6 Argentinian Chimichurri Steak	550
7 Beef Bourguignon	540
8 Canadian Poutine	530
9 Malaysian Rendang	520
10 Omani Shuwa	560
11 Qatari Machboos	510
12 Pasta	700

Рис. 7.39 – Результат виконання запиту 12

13) Цей запит вибирає унікальні назви страв, які містять інгредієнт "сьомга" (результат на рис. 7.40):

```
SELECT DISTINCT recipes.recipe_name
FROM recipes
JOIN recipe_ingredients ON recipes.recipe_id = recipe_ingredients.recipe_id
JOIN ingredients ON recipe_ingredients.ingredient_id = ingredients.ingredient_id
WHERE ingredients.ingredient_name = 'salmon';
```

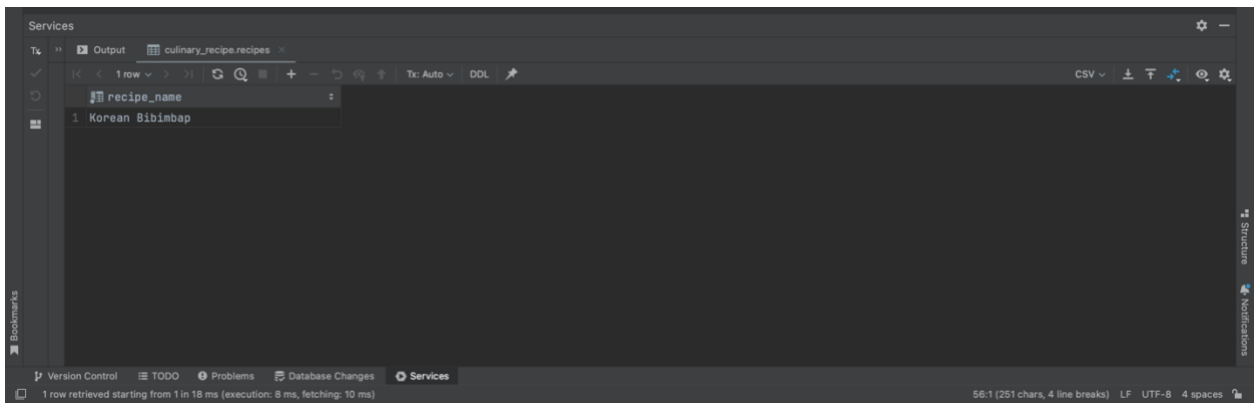


Рис. 7.40 – Результат виконання запиту 13

14) Цей запит повертає назви страв і відповідні відео URL для страв, які мають відео, результат на рис. 7.41:

```
SELECT recipes.recipe_name, recipe_videos.video_url
```

```
FROM recipes
```

```
JOIN recipe_videos ON recipes.recipe_id = recipe_videos.recipe_id;
```

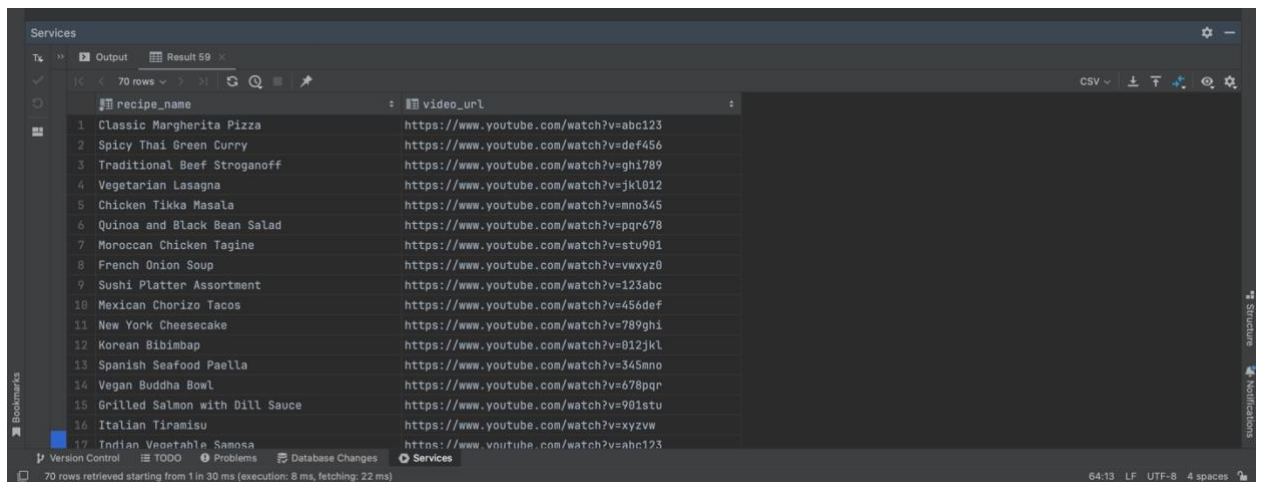


Рис. 7.41 – Результат виконання запиту 14

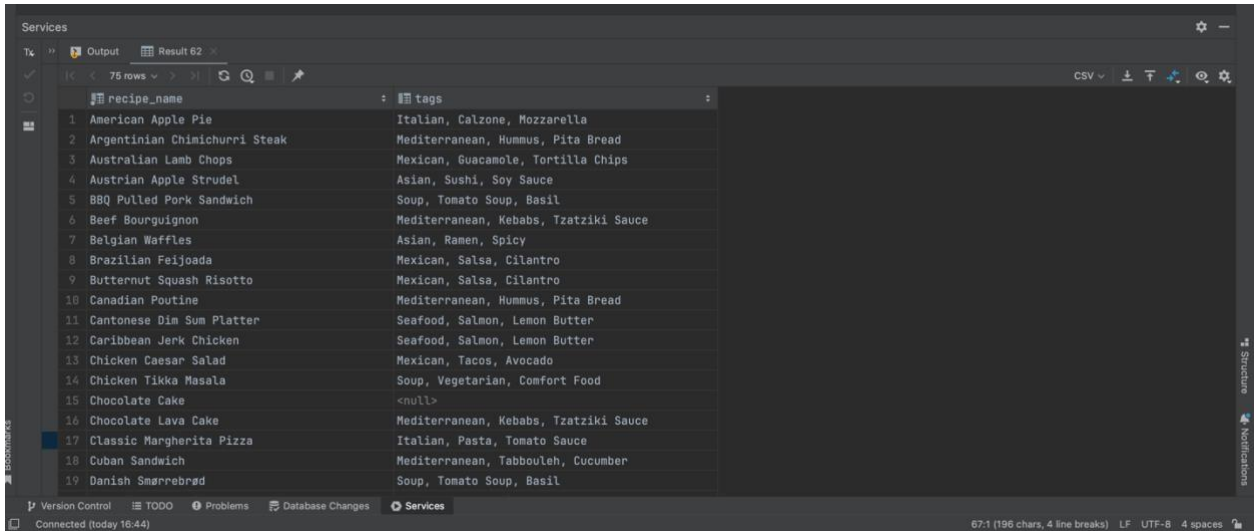
15) Цей запит повертає назви страв і їхні теги, які об'єднані через кому, для кожної страви (результат на рис. 7.42):

```
SELECT recipes.recipe_name, GROUP_CONCAT(recipe_tags.tag_name
SEPARATOR ', ') AS tags
```

```
FROM recipes
```

```
LEFT JOIN recipe_tags ON recipes.recipe_id = recipe_tags.recipe_id
```

```
GROUP BY recipes.recipe_name;
```

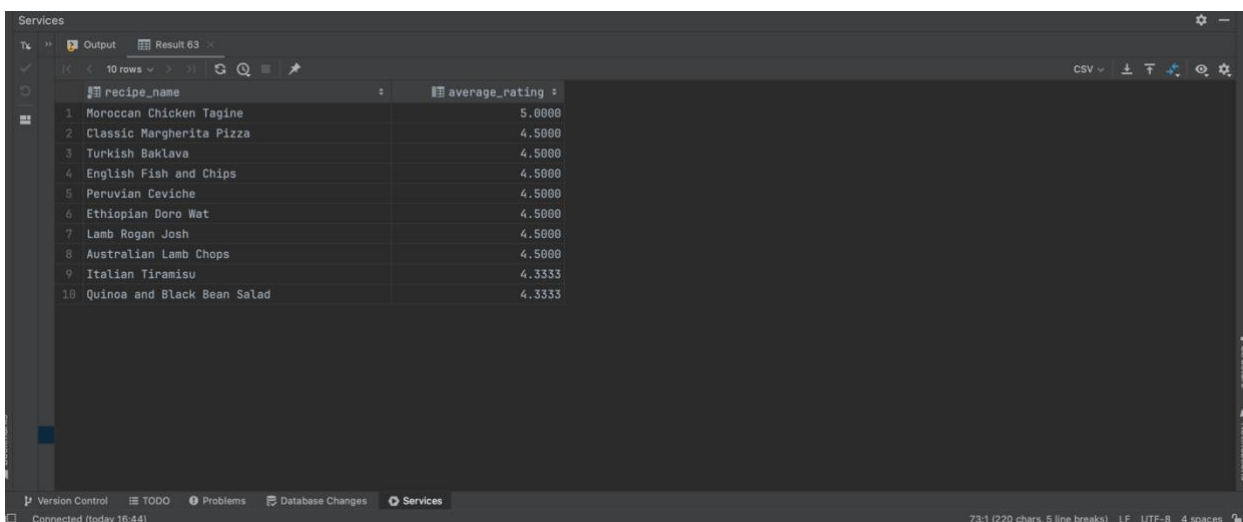



recipe_name	tags
1 American Apple Pie	Italian, Calzone, Mozzarella
2 Argentinian Chimichurri Steak	Mediterranean, Hummus, Pita Bread
3 Australian Lamb Chops	Mexican, Guacamole, Tortilla Chips
4 Austrian Apple Strudel	Asian, Sushi, Soy Sauce
5 BBQ Pulled Pork Sandwich	Soup, Tomato Soup, Basil
6 Beef Bourguignon	Mediterranean, Kebabs, Tzatziki Sauce
7 Belgian Waffles	Asian, Ramen, Spicy
8 Brazilian Feijoada	Mexican, Salsa, Cilantro
9 Butternut Squash Risotto	Mexican, Salsa, Cilantro
10 Canadian Poutine	Mediterranean, Hummus, Pita Bread
11 Cantonese Dim Sum Platter	Seafood, Salmon, Lemon Butter
12 Caribbean Jerk Chicken	Seafood, Salmon, Lemon Butter
13 Chicken Caesar Salad	Mexican, Tacos, Avocado
14 Chicken Tikka Masala	Soup, Vegetarian, Comfort Food
15 Chocolate Cake	<null>
16 Chocolate Lava Cake	Mediterranean, Kebabs, Tzatziki Sauce
17 Classic Margherita Pizza	Italian, Pasta, Tomato Sauce
18 Cuban Sandwich	Mediterranean, Tabbouleh, Cucumber
19 Danish Smørrebrød	Soup, Tomato Soup, Basil

Рис. 7.42 – Результат виконання запиту 15

16) Цей запит обчислює середню оцінку користувачів для кожної страви і повертає назви страв та їхню середню оцінку, впорядковані у спадному порядку за середньою оцінкою, обмежені 10 результатами (результат на рис. 7.43):

```
SELECT recipes.recipe_name, AVG(user_ratings.rating) AS average_rating
FROM recipes
LEFT JOIN user_ratings ON recipes.recipe_id = user_ratings.recipe_id
GROUP BY recipes.recipe_name
ORDER BY average_rating DESC
LIMIT 10;
```



The screenshot shows a database query result with 10 rows. The columns are 'recipe_name' and 'average_rating'. The data is as follows:

recipe_name	average_rating
1 Moroccan Chicken Tagine	5.0000
2 Classic Margherita Pizza	4.5000
3 Turkish Baklava	4.5000
4 English Fish and Chips	4.5000
5 Peruvian Ceviche	4.5000
6 Ethiopian Doro Wat	4.5000
7 Lamb Rogan Josh	4.5000
8 Australian Lamb Chops	4.5000
9 Italian Tiramisu	4.3333
10 Quinoa and Black Bean Salad	4.3333

Рис. 7.43 – Результат виконання запиту 16

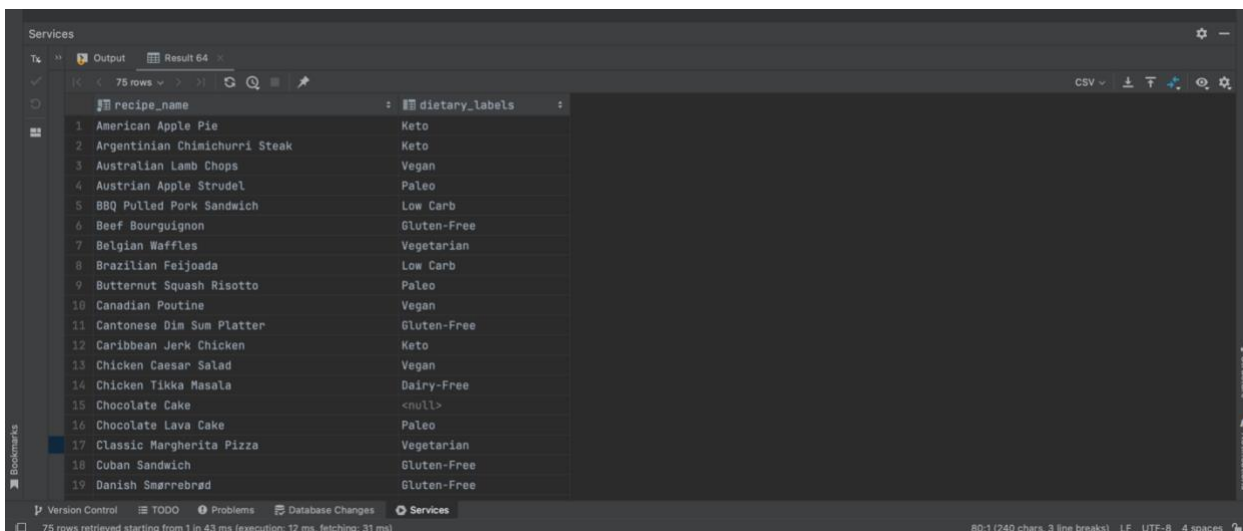
17) Цей запит об'єднує страви з їхніми дієтичними мітками і повертає назви страв разом із згрупованими дієтичними мітками, розділеними комами (результат на рис. 7.44):

```
SELECT recipes.recipe_name,
GROUP_CONCAT(recipe_dietary_labels.dietary_label SEPARATOR ', ') AS
dietary_labels
```

```
FROM recipes
```

```
LEFT JOIN recipe_dietary_labels ON recipes.recipe_id =
recipe_dietary_labels.recipe_id
```

```
GROUP BY recipes.recipe_name;
```



The screenshot shows a database query result with 19 rows. The columns are 'recipe_name' and 'dietary_labels'. The data is as follows:

recipe_name	dietary_labels
1 American Apple Pie	Keto
2 Argentinian Chimichurri Steak	Keto
3 Australian Lamb Chops	Vegan
4 Austrian Apple Strudel	Paleo
5 BBQ Pulled Pork Sandwich	Low Carb
6 Beef Bourguignon	Gluten-Free
7 Belgian Waffles	Vegetarian
8 Brazilian Feijoada	Low Carb
9 Butternut Squash Risotto	Paleo
10 Canadian Poutine	Vegan
11 Cantonese Dim Sum Platter	Gluten-Free
12 Caribbean Jerk Chicken	Keto
13 Chicken Caesar Salad	Vegan
14 Chicken Tikka Masala	Dairy-Free
15 Chocolate Cake	<null>
16 Chocolate Lava Cake	Paleo
17 Classic Margherita Pizza	Vegetarian
18 Cuban Sandwich	Gluten-Free
19 Danish Smørrebrød	Gluten-Free

Рис. 7.44 – Результат виконання запиту 17

18) Цей запит повертає назви страв разом із текстами коментарів та датами цих коментарів, які були залишені для цих страв (результат на рис. 7.45):

```
SELECT recipes.recipe_name, recipe_comments.comment_text,
recipe_comments.comment_date
FROM recipes
JOIN recipe_comments ON recipes.recipe_id = recipe_comments.recipe_id;
```

recipe_name	comment_text	comment_date
1 Classic Margherita Pizza	This recipe is amazing! I tried it last night and it was a hit with my family.	2023-12-18 12:10:42
2 Classic Margherita Pizza	I love this dish! It is become a family favorite.	2023-12-18 12:10:42
3 Classic Margherita Pizza	The flavors are so well-balanced. Great recipe!	2023-12-18 12:10:42
4 Classic Margherita Pizza	I added a little extra spice to mine, and it was perfect!	2023-12-18 12:10:42
5 Classic Margherita Pizza	Test comment	2023-12-24 00:02:14
6 Spicy Thai Green Curry	This pasta is delicious! I make it regularly for dinner.	2023-12-18 12:10:42
7 Spicy Thai Green Curry	Simple yet tasty. My kids love it!	2023-12-18 12:10:42
8 Spicy Thai Green Curry	I like to add some chopped fresh basil for extra flavor.	2023-12-18 12:10:42
9 Traditional Beef Stroganoff	I am a vegetarian, and this recipe is a lifesaver!	2023-12-18 12:10:42
10 Traditional Beef Stroganoff	So much flavor in one dish. I cant get enough of it.	2023-12-18 12:10:42
11 Traditional Beef Stroganoff	Perfect for a quick and healthy lunch.	2023-12-18 12:10:42
12 Vegetarian Lasagna	This recipe is a winner! I served it at a dinner party, and everyone loved it.	2023-12-18 12:10:42
13 Vegetarian Lasagna	The presentation of this dish is as impressive as the taste.	2023-12-18 12:10:42
14 Vegetarian Lasagna	I am a fan of Italian cuisine, and this recipe nails it.	2023-12-18 12:10:42
15 Vegetarian Lasagna	Sample comment	2023-12-24 00:08:32
16 Chicken Tikka Masala	This soup is so comforting, especially on a cold day.	2023-12-18 12:10:42
17 Chicken Tikka Masala	I added some croutons for crunch, and it was perfect!	2023-12-18 12:10:42
18 Chicken Tikka Masala	My family requests this soup all the time.	2023-12-18 12:10:42
19 Quinoa and Black Bean Salad	I love seafood, and this recipe is now one of my favorites.	2023-12-18 12:10:42

Рис. 7.45 – Результат виконання запиту 18

19) Цей запит вибирає назви страв та їх калорійність (калорії), де калорійність менше 300 (результат на рис. 7.46):

```
SELECT recipe_name, calories
FROM recipes
WHERE calories < 300;
```

recipe_name	calories
1 French Onion Soup	280
2 Indian Vegetable Samosa	250
3 Zucchini Noodles with Pesto	220
4 Russian Borscht	280
5 Spanish Gazpacho	220
6 Danish Smørrebrød	290
7 Lebanese Tabouleh	270
8 Test Recipe	100
9 Sample Recipe	200
10 Sample Recipe	200
11 Test Recipe	200

Рис. 7.46 – Результат виконання запиту 19

20) Цей запит вибирає назви страв та їх дієтичні мітки, де принаймні одна з міток містить "Low Carb" (результат на рис. 7.47):

```
SELECT r.recipe_name, GROUP_CONCAT(rd.dietary_label SEPARATOR ', ')
AS dietary_labels
FROM recipes AS r
LEFT JOIN recipe_dietary_labels AS rd ON r.recipe_id = rd.recipe_id
WHERE rd.dietary_label LIKE '%Low Carb%'
GROUP BY r.recipe_name;
```

recipe_name	dietary_labels
1 BBQ Pulled Pork Sandwich	Low Carb
2 Brazilian Feijoada	Low Carb
3 Egyptian Koshari	Low Carb
4 Hungarian Goulash	Low Carb
5 Lebanese Tabouleh	Low Carb
6 New York Cheesecake	Low Carb
7 Pad Thai	Low Carb
8 Vegetarian Lasagna	Low Carb
9 Vietnamese Pho	Low Carb

Рис. 7.47 – Результат виконання запиту 20

21) Цей запит вибирає назви страв разом з переліком інгредієнтів, які включають курку, брокколі і кіноа, і групує їх разом (результат на рис. 7.48):

```
SELECT recipe_name, GROUP_CONCAT(ingredient_name SEPARATOR ', ')
AS ingredients
```

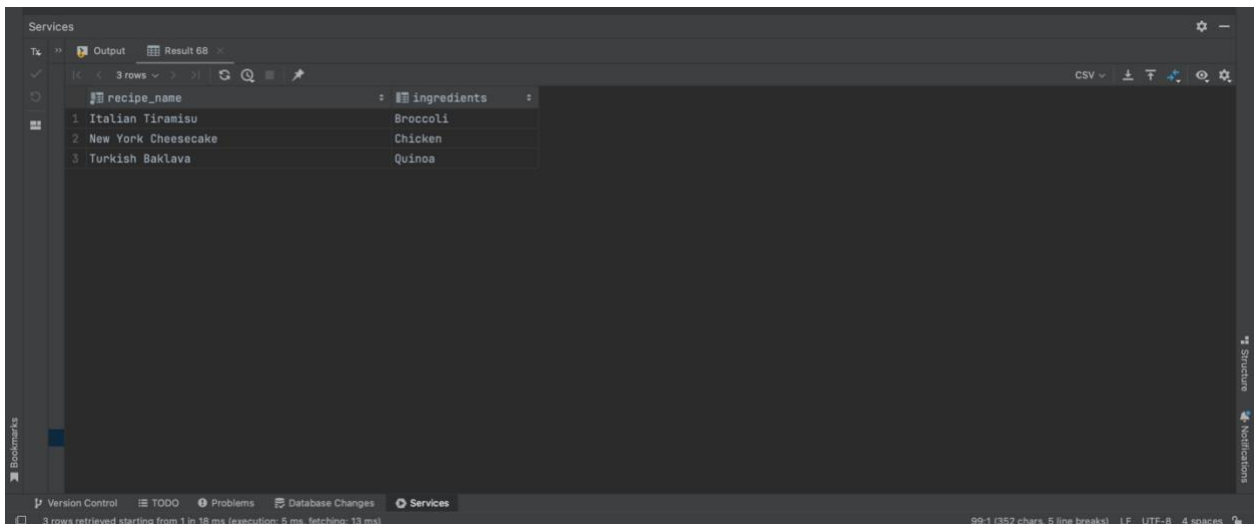
```
FROM recipes
```

```
LEFT JOIN recipe_ingredients ON recipes.recipe_id =
recipe_ingredients.recipe_id
```

```
LEFT JOIN ingredients ON recipe_ingredients.ingredient_id =
ingredients.ingredient_id
```

```
WHERE ingredients.ingredient_name IN ('Chicken', 'Broccoli', 'Quinoa')
```

```
GROUP BY recipe_name;
```



recipe_name	ingredients
1. Italian Tiramisu	Broccoli
2. New York Cheesecake	Chicken
3. Turkish Baklava	Quinoa

Рис. 7.48 – Результат виконання запиту 21

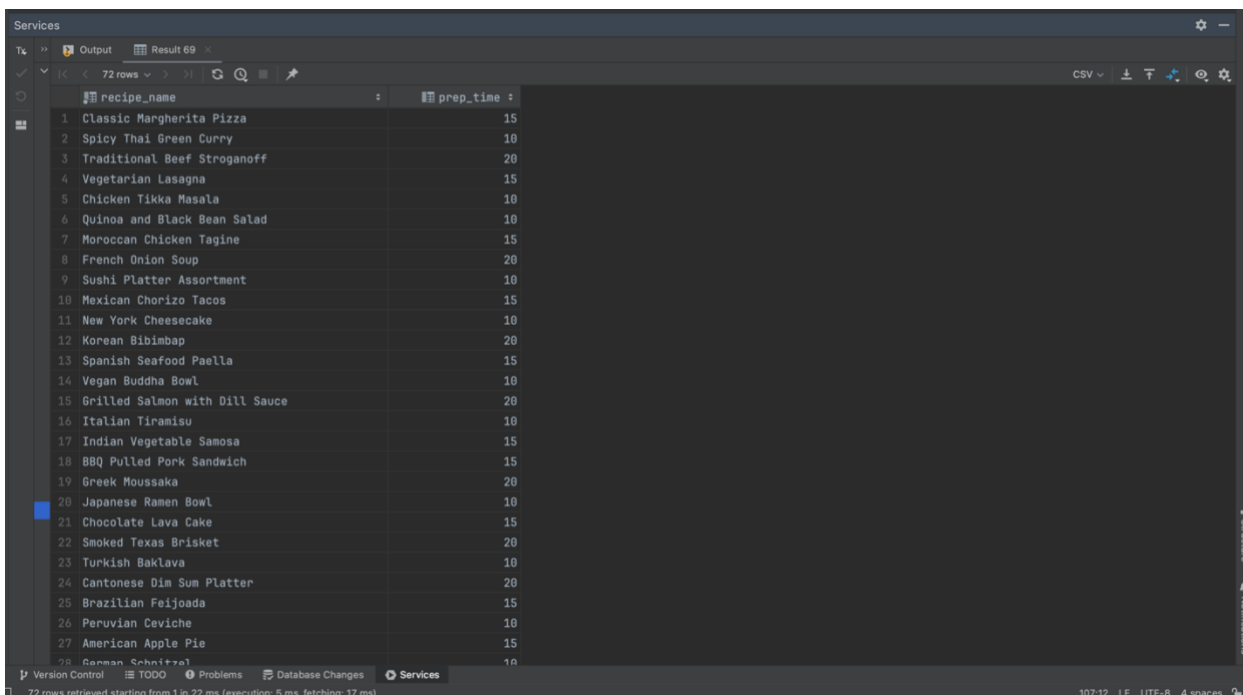
22) Цей запит вибирає назви страв та час підготовки (prep_time) з таблиці рецептів, де час підготовки не перевищує 30 хвилин (результат на рис. 7.49):

```
SELECT r.recipe_name, rt.prep_time
```

```
FROM recipes AS r
```

```
LEFT JOIN recipe_prep_times AS rt ON r.recipe_id = rt.recipe_id
```

```
WHERE rt.prep_time <= 30;
```



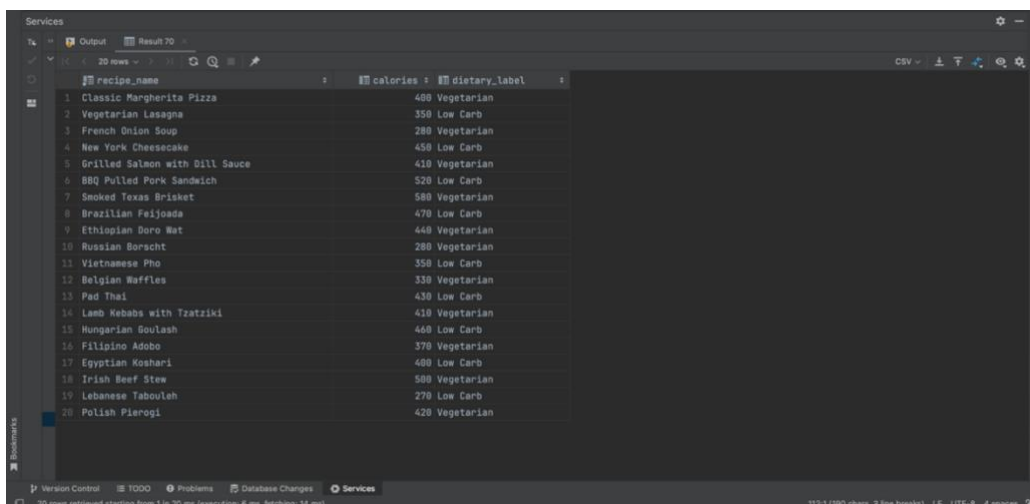
The screenshot shows a database query result with 28 rows. The columns are 'recipe_name' and 'prep_time'. The data is as follows:

recipe_name	prep_time
1 Classic Margherita Pizza	15
2 Spicy Thai Green Curry	10
3 Traditional Beef Stroganoff	20
4 Vegetarian Lasagna	15
5 Chicken Tikka Masala	10
6 Quinoa and Black Bean Salad	10
7 Moroccan Chicken Tagine	15
8 French Onion Soup	20
9 Sushi Platter Assortment	10
10 Mexican Chorizo Tacos	15
11 New York Cheesecake	10
12 Korean Bibimbap	20
13 Spanish Seafood Paella	15
14 Vegan Buddha Bowl	10
15 Grilled Salmon with Dill Sauce	20
16 Italian Tiramisu	10
17 Indian Vegetable Samosa	15
18 BBQ Pulled Pork Sandwich	15
19 Greek Moussaka	20
20 Japanese Ramen Bowl	10
21 Chocolate Lava Cake	15
22 Smoked Texas Brisket	20
23 Turkish Baklava	10
24 Cantonese Dim Sum Platter	20
25 Brazilian Feijoada	15
26 Peruvian Ceviche	10
27 American Apple Pie	15
28 German Schnitzel	10

Рис. 7.49 – Результат виконання запиту 22

23) Цей запит вибирає назви страв, кількість калорій і дієтичні мітки з таблиць рецептів та дієтичних міток, де дієтична мітка є "Low Carb" або "Vegetarian" (результат на рис. 7.50):

```
SELECT r.recipe_name, r.calories, rl.dietary_label
FROM recipes AS r
LEFT JOIN recipe_dietary_labels AS rl ON r.recipe_id = rl.recipe_id
WHERE rl.dietary_label IN ('Low Carb', 'Vegetarian');
```



The screenshot shows a database query result with 20 rows. The columns are 'recipe_name', 'calories', and 'dietary_label'. The data is as follows:

recipe_name	calories	dietary_label
1 Classic Margherita Pizza	480	Vegetarian
2 Vegetarian Lasagna	350	Low Carb
3 French Onion Soup	280	Vegetarian
4 New York Cheesecake	450	Low Carb
5 Grilled Salmon with Dill Sauce	410	Vegetarian
6 BBQ Pulled Pork Sandwich	520	Low Carb
7 Smoked Texas Brisket	580	Vegetarian
8 Brazilian Feijoada	470	Low Carb
9 Ethiopian Doro Wat	440	Vegetarian
10 Russian Borscht	280	Vegetarian
11 Vietnamese Pho	350	Low Carb
12 Belgian Waffles	330	Vegetarian
13 Pad Thai	430	Low Carb
14 Lamb Kebabs with Tzatziki	410	Vegetarian
15 Hungarian Goulash	460	Low Carb
16 Filipino Adobo	370	Vegetarian
17 Egyptian Koshari	480	Low Carb
18 Irish Beef Stew	580	Vegetarian
19 Lebanese Tabouleh	270	Low Carb
20 Polish Pierogi	420	Vegetarian

Рис. 7.50 – Результат виконання запиту 23

24) Цей запит вибирає назву страви, яка має найбільше оцінок користувачів (результат на рис. 7.51):

```
SELECT recipe_name
FROM recipes
WHERE recipe_id = (
    SELECT recipe_id
    FROM user_ratings
    GROUP BY recipe_id
    ORDER BY COUNT(*) DESC
    LIMIT 1
);
```

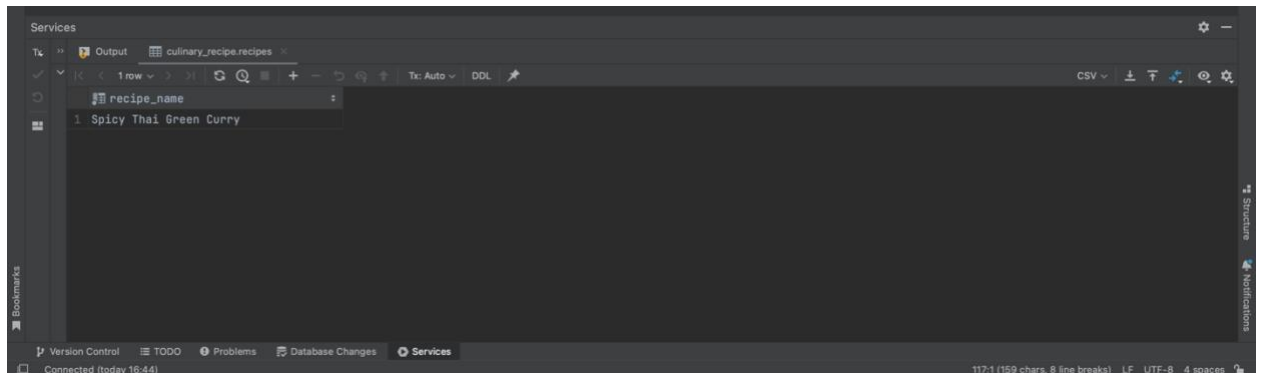


Рис. 7.51 – Результат виконання запиту 24

25) Цей запит вибирає унікальні імена користувачів, які оцінили страву, яка має найбільше оцінок серед користувачів (результат на рис. 7.52):

```
SELECT DISTINCT u.username
FROM users AS u
JOIN user_ratings AS ur ON u.user_id = ur.user_id
WHERE ur.recipe_id = (
    SELECT recipe_id
    FROM user_ratings
    GROUP BY recipe_id
    ORDER BY COUNT(*) DESC
```

LIMIT 1

);

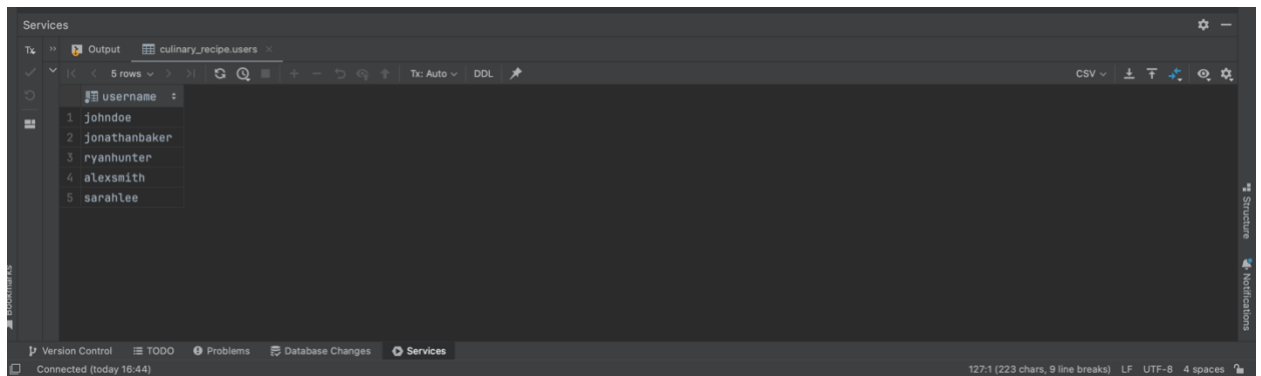


Рис. 7.52 – Результат виконання запиту 25

26) Цей запит вибирає назви страв, які містять інгредієнти зі словом "томат" у їхніх іменах (результат на рис. 7.53):

```
SELECT recipe_name
```

```
FROM recipes
```

```
WHERE recipe_id IN (
```

```
    SELECT DISTINCT ri.recipe_id
```

```
    FROM recipe_ingredients AS ri
```

```
    JOIN ingredients AS i ON ri.ingredient_id = i.ingredient_id
```

```
    WHERE i.ingredient_name LIKE '%tomato%'
```

```
);
```

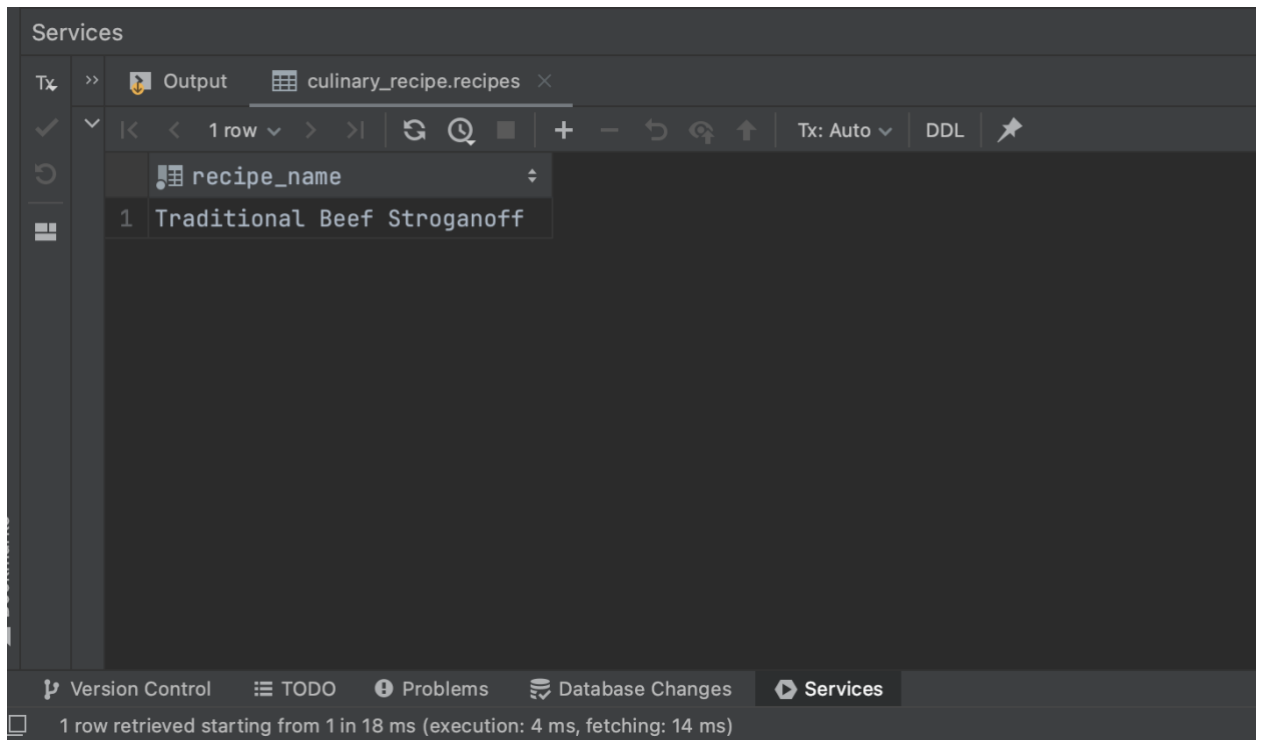



Рис. 7.53 – Результат виконання запиту 26

27) Цей запит обчислює середню кількість калорій для страв типу "Vietnamese Pho". Якщо немає жодної страви цього типу, то повертається значення 0 як середнє значення калорій(результат на рис. 7.54):

```
SELECT COALESCE(AVG(calories), 0) AS average_calories
FROM recipes
WHERE dish_type_id = (
    SELECT dish_type_id
    FROM dish_types
    WHERE dish_type_name = 'Vietnamese Pho'
);
```

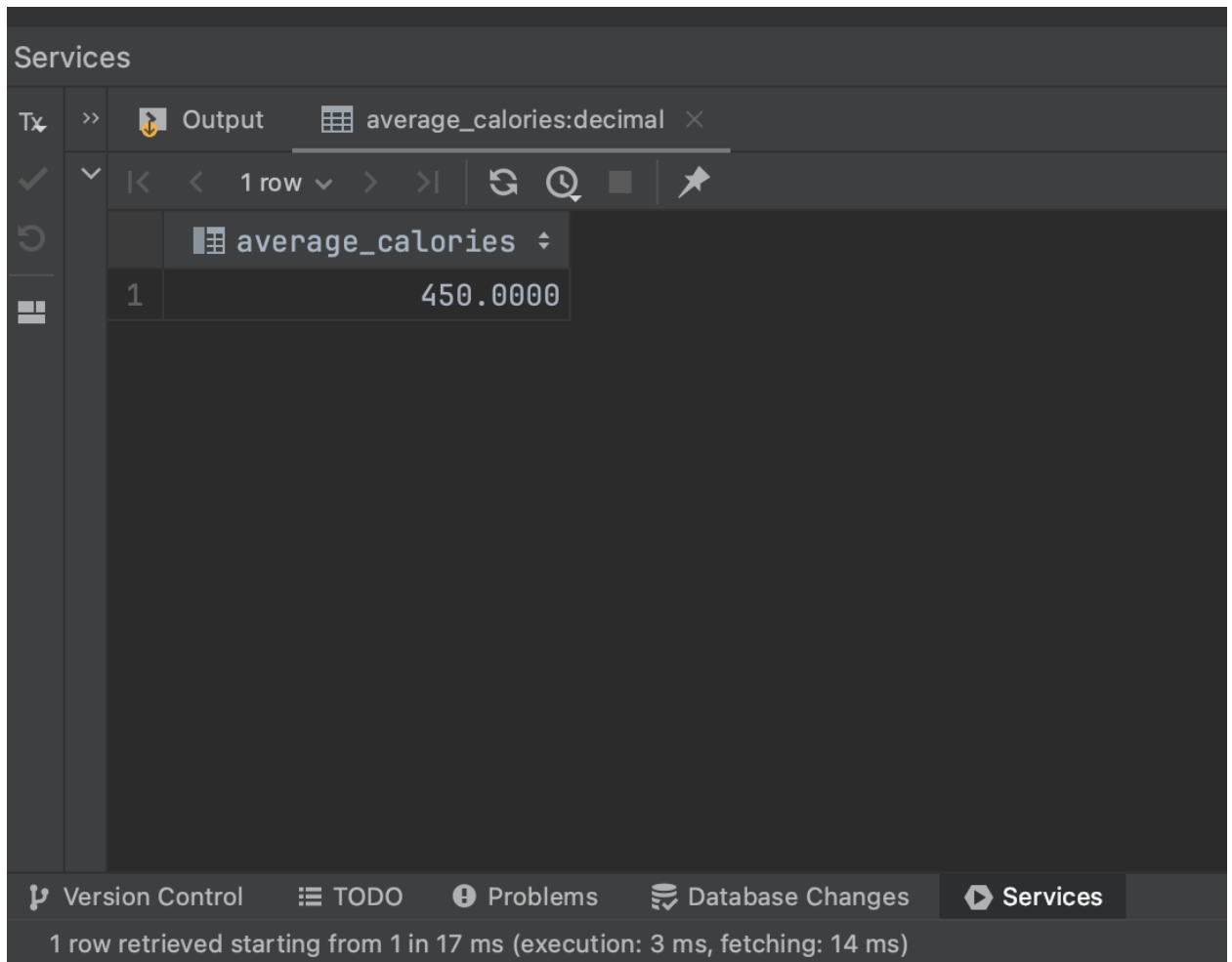


Рис. 7.54 – Результат виконання запиту 27

7.6 Індeksi та результати оптимізації

Для оптимізації роботи бази даних був створений та виконаний наступний скрипт, який робить індексацію даних в таблицях для більш швидкого пошуку по них.

```
CREATE INDEX idx_dish_types_dish_type_name ON
dish_types(dish_type_name);
CREATE INDEX idx_ingredients_ingredient_name ON
ingredients(ingredient_name);
CREATE INDEX idx_recipes_dish_type_id ON recipes(dish_type_id);
CREATE INDEX idx_recipes_recipe_name ON recipes(recipe_name);
CREATE INDEX idx_recipe_ingredients_recipe_id ON
recipe_ingredients(recipe_id);
```

```

CREATE INDEX idx_recipe_ingredients_ingredient_id ON
recipe_ingredients(ingredient_id);
CREATE INDEX idx_instructions_step_number ON instructions(step_number);
CREATE INDEX idx_instructions_recipe_id ON instructions(recipe_id);
CREATE INDEX idx_user_favorites_user_id ON user_favorites(user_id);
CREATE INDEX idx_user_favorites_recipe_id ON user_favorites(recipe_id);
CREATE INDEX idx_user_ratings_user_id ON user_ratings(user_id);
CREATE INDEX idx_user_ratings_recipe_id ON user_ratings(recipe_id);
CREATE INDEX idx_recipe_prep_times_recipe_id ON
recipe_prep_times(recipe_id);
CREATE INDEX idx_recipe_servings_recipe_id ON recipe_servings(recipe_id);
CREATE INDEX idx_recipe_dietary_labels_recipe_id ON
recipe_dietary_labels(recipe_id);
CREATE INDEX idx_recipe_videos_recipe_id ON recipe_videos(recipe_id);
CREATE INDEX idx_recipe_sources_recipe_id ON recipe_sources(recipe_id);
CREATE INDEX idx_recipe_notes_recipe_id ON recipe_notes(recipe_id);
CREATE INDEX idx_user_dietary_preferences_user_id ON
user_dietary_preferences(user_id);
CREATE INDEX idx_user_profiles_user_id ON user_profiles(user_id);
CREATE INDEX idx_recipe_comments_recipe_id ON
recipe_comments(recipe_id);
CREATE INDEX idx_recipe_comments_user_id ON recipe_comments(user_id);
CREATE INDEX idx_recipe_tags_recipe_id ON recipe_tags(recipe_id);

```

Для порівняння результатів оптимізації, були створені 3 тестових запити. Була порівняна швидкість їх виконання до і після оптимізації. Результати можна побачити на рисунках 7.55 – 7.60.

```

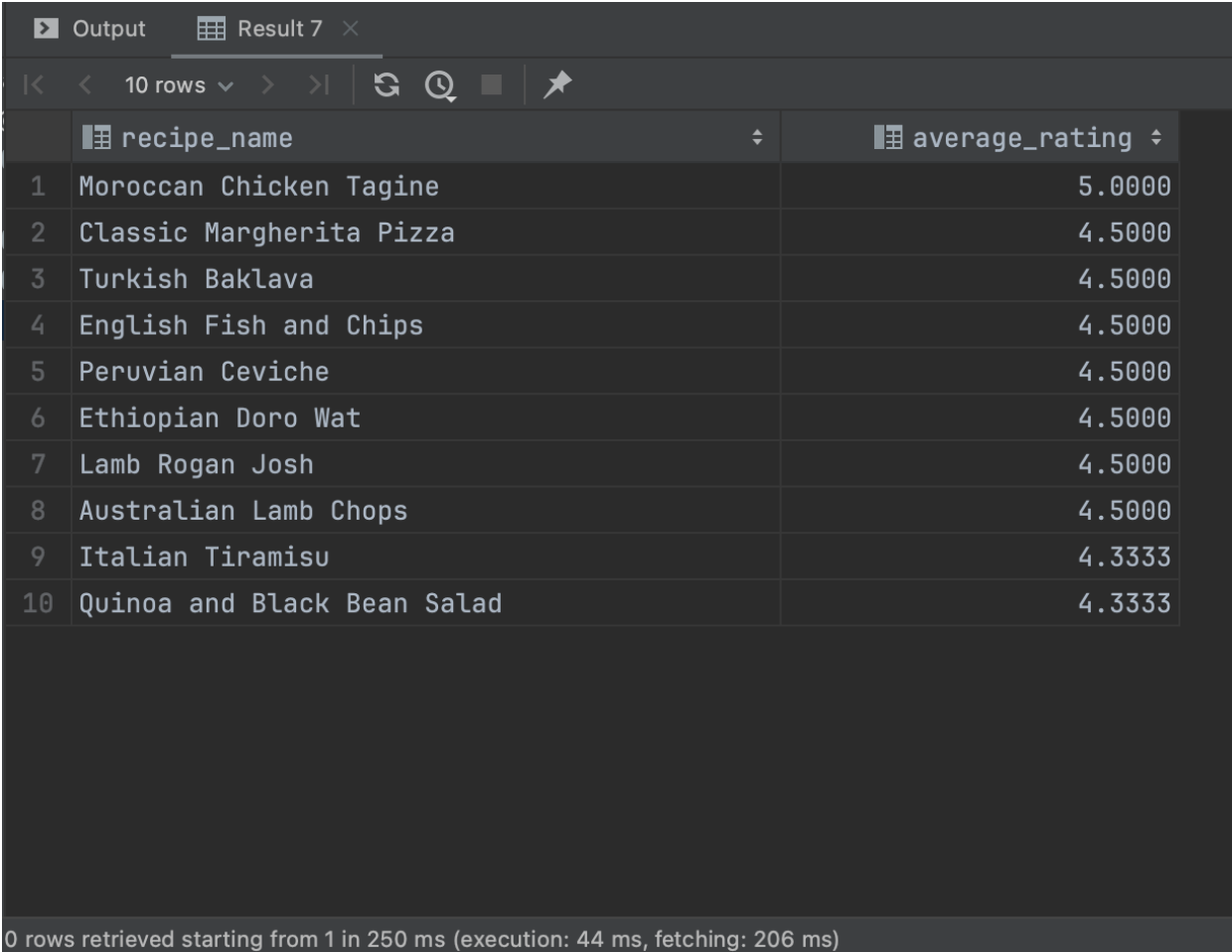
SELECT r.recipe_name, subquery.average_rating
FROM recipes AS r
JOIN (

```

```

SELECT recipe_id, AVG(rating) AS average_rating
FROM user_ratings
GROUP BY recipe_id
ORDER BY average_rating DESC
LIMIT 10
) AS subquery ON r.recipe_id = subquery.recipe_id;

```



	recipe_name	average_rating
1	Moroccan Chicken Tagine	5.0000
2	Classic Margherita Pizza	4.5000
3	Turkish Baklava	4.5000
4	English Fish and Chips	4.5000
5	Peruvian Ceviche	4.5000
6	Ethiopian Doro Wat	4.5000
7	Lamb Rogan Josh	4.5000
8	Australian Lamb Chops	4.5000
9	Italian Tiramisu	4.3333
10	Quinoa and Black Bean Salad	4.3333

0 rows retrieved starting from 1 in 250 ms (execution: 44 ms, fetching: 206 ms)

Рис. 7.55 – Час виконання запиту 1 без оптимізації 250ms

Services

Output Result 13

10 rows

	recipe_name	average_rating
1	Moroccan Chicken Tagine	5.0000
2	Classic Margherita Pizza	4.5000
3	Turkish Baklava	4.5000
4	English Fish and Chips	4.5000
5	Peruvian Ceviche	4.5000
6	Ethiopian Doro Wat	4.5000
7	Lamb Rogan Josh	4.5000
8	Australian Lamb Chops	4.5000
9	Italian Tiramisu	4.3333
10	Quinoa and Black Bean Salad	4.3333

Version Control TODO Problems Database Changes Services

10 rows retrieved starting from 1 in 30 ms (execution: 6 ms, fetching: 24 ms)

Рис. 7.56 – Час виконання запиту 1 з оптимізацією 30ms

```
CREATE INDEX idx_ingredient_name ON ingredients (ingredient_name);
SELECT DISTINCT recipes.recipe_name
FROM recipes
INNER JOIN recipe_ingredients ON recipes.recipe_id =
recipe_ingredients.recipe_id
INNER JOIN ingredients ON recipe_ingredients.ingredient_id =
ingredients.ingredient_id
WHERE ingredients.ingredient_name = 'Chocolate';
```

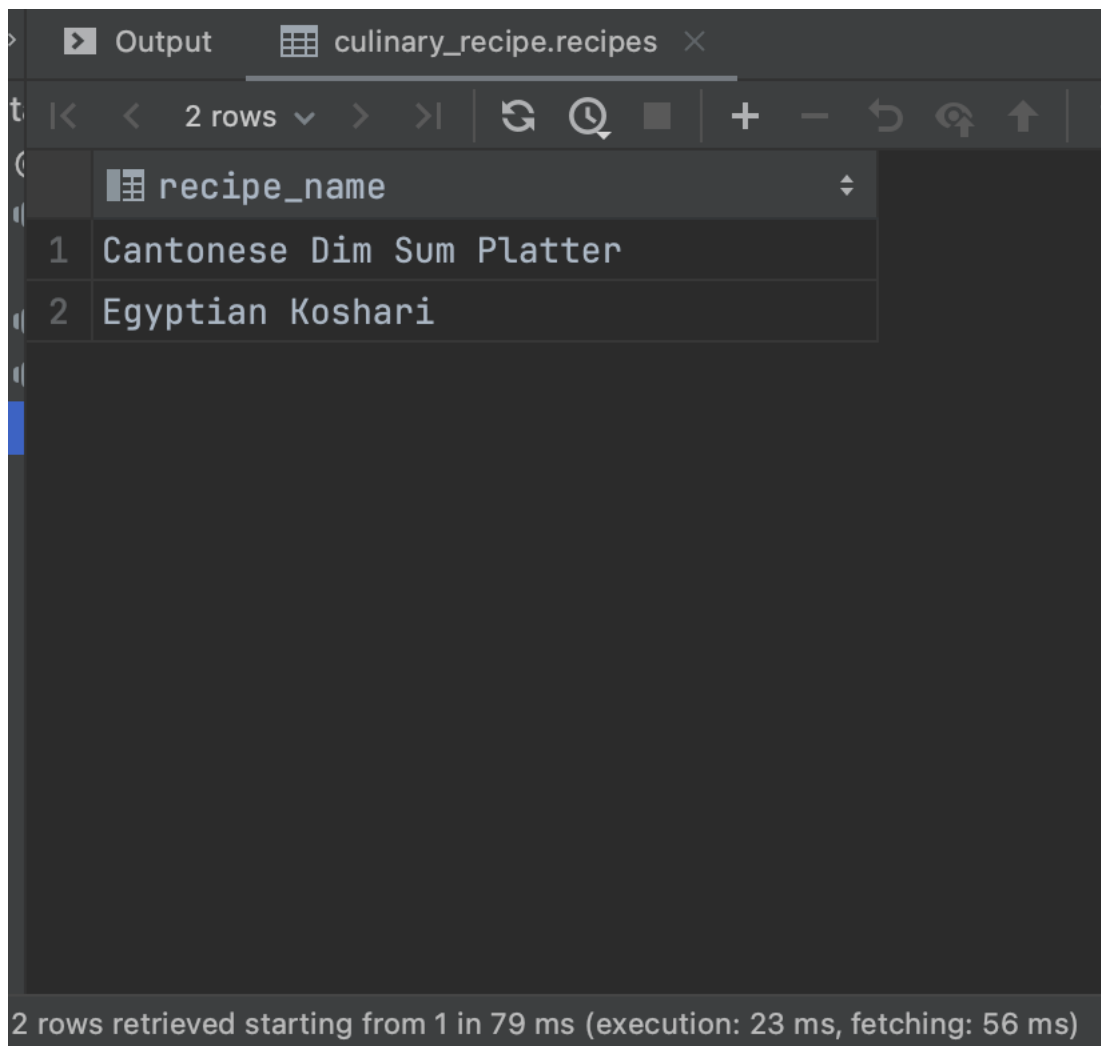


Рис. 7.57 – Час виконання запиту 2 без оптимізації 79ms

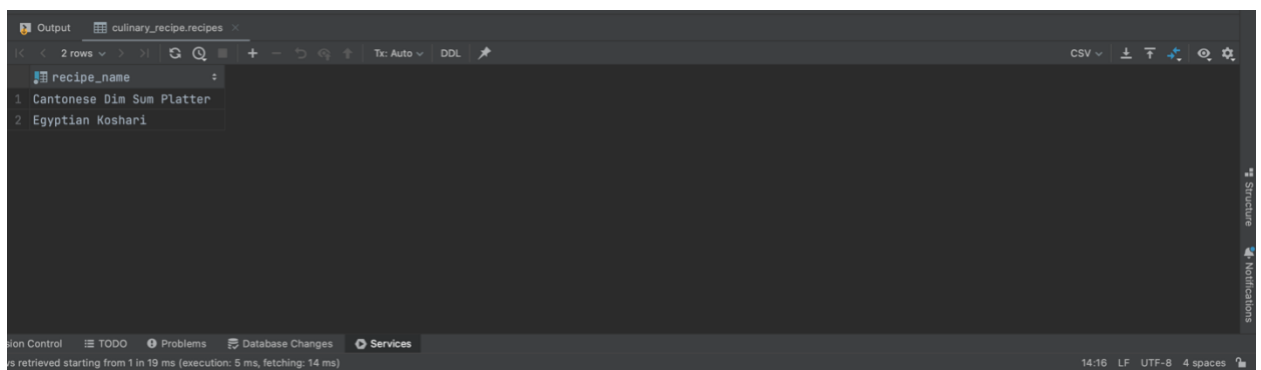
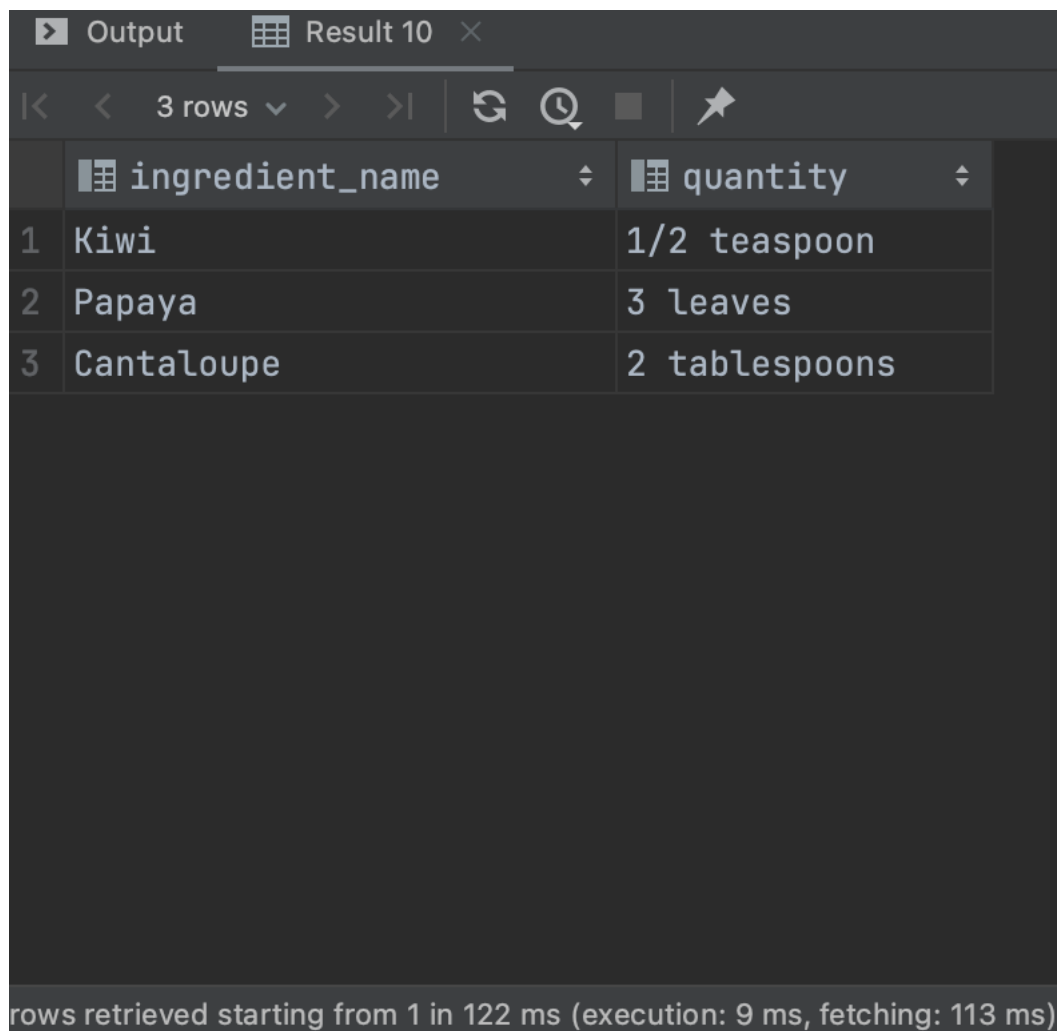


Рис. 7.58 – Час виконання запиту 2 з оптимізацією 19ms

```
CREATE INDEX idx_recipe_id ON recipe_ingredients (recipe_id);
SELECT ingredient_name, quantity
FROM ingredients
```

```
INNER JOIN recipe_ingredients ON ingredients.ingredient_id =
recipe_ingredients.ingredient_id
WHERE recipe_ingredients.recipe_id = 57;
```

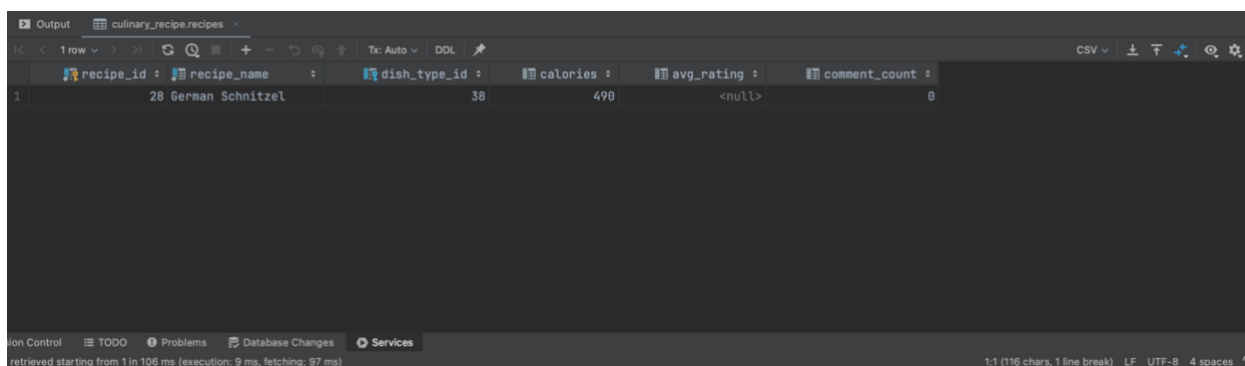


The screenshot shows a database query result window titled 'Output' and 'Result 10'. It displays 3 rows of data. The columns are 'ingredient_name' and 'quantity'. The rows are: 1 Kiwi, 1/2 teaspoon; 2 Papaya, 3 leaves; 3 Cantaloupe, 2 tablespoons. At the bottom, it states '3 rows retrieved starting from 1 in 122 ms (execution: 9 ms, fetching: 113 ms)'.

	ingredient_name	quantity
1	Kiwi	1/2 teaspoon
2	Papaya	3 leaves
3	Cantaloupe	2 tablespoons

3 rows retrieved starting from 1 in 122 ms (execution: 9 ms, fetching: 113 ms)

Рис. 7.59 – Час виконання запиту 3 без оптимізації 122ms



The screenshot shows a database query result window titled 'Output' and 'culinary_recipe.recipes'. It displays 1 row of data. The columns are 'recipe_id', 'recipe_name', 'dish_type_id', 'calories', 'avg_rating', and 'comment_count'. The row is: 1 28 German Schnitzel, 38, 490, <null>, 0. At the bottom, it states '1 row retrieved starting from 1 in 106 ms (execution: 9 ms, fetching: 97 ms)'.

recipe_id	recipe_name	dish_type_id	calories	avg_rating	comment_count
1	28 German Schnitzel	38	490	<null>	0

1 row retrieved starting from 1 in 106 ms (execution: 9 ms, fetching: 97 ms)

Рис. 7.60 – Час виконання запиту 3 з оптимізацією 106ms

Отже, вдалося оптимізувати базу даних використовуючи індексацію полів в таблицях. Це дало невелику перевагу, приблизно від 5 до 40 мілісекунд. В результаті вжитих заходів з оптимізації та удосконалення бази даних вдалося покращити її продуктивність, знизити час відповіді на запити та забезпечити більш ефективне використання ресурсів системи. Якщо таблиця має невеликий розмір, індекси можуть не призвести до значного покращення продуктивності, і їх використання може бути зайвим. якщо запит повертає велику кількість рядків або весь обсяг даних з таблиці, використання індексів може не мати значного впливу, оскільки оптимізатор запитів може вирішити прочитати всі рядки. Отже, не завжди використання індексів може мати позитивний вплив.

У цьому розділі були створені процедури, функції, представлення, SQL запити до бази даних, а також зроблена оптимізація бази даних за допомогою оптимізації пошуку в таблицях завдяки індексації полів.

ВИСНОВКИ

У ході виконання цього проекту було проведено аналіз предметної області та визначено вимоги до створення бази даних для управління кулінарною рецептурою. Проект спрямований на розробку структури бази даних, яка відповідає потребам у зберіганні та управлінні рецептами, інгредієнтами, відгуками користувачів та іншими аспектами кулінарної сфери.

Аналіз бізнес-процесів та вимог до бази даних показав, що створення цієї бази даних є необхідним для підтримки об'єктивного зберігання рецептів, управління інгредієнтами, відстеження відгуків користувачів та забезпечення ефективного доступу до цієї інформації.

Розроблена ER-модель бази даних відображає ключові сутності та їх взаємодії в контексті кулінарної рецептурою. Реалізація бази даних включає створення таблиць, імпорт даних, створення користувачів та розробку функцій для роботи з нею. Забезпечено можливість багатокористувальницького доступу та розроблено різні компоненти, такі як тексти генераторів, збережені процедури, тригери, представлення та SQL-запити, які враховують особливості управління кулінарною інформацією.

Мета проекту – полегшити управління та доступ до кулінарних рецептів та інгредієнтів, забезпечуючи зручний та ефективний інструмент для користувачів. Розроблена база даних має потенціал покращити організацію кулінарних процесів та сприяти обміну рецептами між користувачами. Важливо також забезпечити захист даних та конфіденційність особистої інформації користувачів.

Розроблена база даних відповідає вимогам ефективного управління кулінарними рецептами, забезпечуючи стійкість до навантажень, легку масштабованість та швидкий доступ до інформації. Вона спрощує процес зберігання, пошуку та обміну кулінарними знаннями, роблячи його більш зручним та доступним для всіх.

Отже, розробка та впровадження даної бази даних є важливим кроком у напрямку покращення управління та обміну кулінарними рецептами та інформацією в цій сфері.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- 1) MySQL. URL: <https://www.mysql.com/> (date of access: 24.12.2023).
- 2) MySQL tutorial.
URL: <https://www.w3schools.com/MySQL/default.asp><https://www.w3schools.com/MySQL/default.asp>(date of access: 22.12.2023).
- 3) Ingredients list. URL: <https://foodhero.org/ingredients> (date of access: 26.12.2023).
- 4) Recipes list. URL: <https://recipes.net/articles/20-easy-culinary-school-recipes-to-master/> (date of access: 24.12.2023).
- 5) Trigger syntax. URL: <https://dev.mysql.com/doc/refman/8.0/en/trigger-syntax.html> (date of access: 26.12.2023).