

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КІЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

Рецепти

КУРСОВА РОБОТА

з дисципліни «Бази даних»

Керівник:

д.т.н., доц. Завгородній В.В.

«Допущено до захисту»

(особистий підпис керівника)

«____» _____ 20__ р.

Захищено з оцінкою

_____ (оцінка)

Виконавець:

Ликова К.О.
студент II курсу
групи IA-311



(особистий підпис виконавця)

«17» січня 2023 р.

Члени комісії:

_____ (особистий підпис)

_____ (розшифровка підпису)

_____ (особистий підпис)

_____ (розшифровка підпису)

Київ 2023

**Міністерство освіти та науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Кафедра інформаційних систем та технологій

Дисципліна Бази даних

Спеціальність Інформаційні системи та технології

Курс II Група ІА-з11 Семестр 3

**ЗАВДАННЯ
на курсову роботу студента**

Ликова Катерина Олександрівна
(прізвище, ім'я, по батькові)

1. Тема роботи рецепти

2. Термін здачі студентом закінченої роботи « 17 » січня 2022 р.

3. Вхідні дані до роботи система управління базами даних MS SQL Server

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

зміст, вступ, теоретична частина (аналіз та опис предметної області, проектування бази даних), практична частина (створення бази даних, організація вибірки інформації з бази даних), висновки, список використаної літератури, додатки (заповнення таблиць реляційної бази даних)

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

6. Дата видачі завдання « 14 » вересня 2022 р.

КАЛЕНДАРНИЙ ПЛАН-ГРАФІК

виконання курсової роботи

студентом Ликова К.О.
(прізвище, ініціали)

№ з/п	Назва етапів роботи та питань, які мають бути розвроблені відповідно до завдання	Термін виконання	Примітки керівника про виконання завдань
1	Видача завдання	14.09.22	
2	Аналіз та опис предметної області	06.10.22	
3	Проектування бази даних	27.10.22	
4	Створення бази даних	17.11.22	
5	Організація вибірки інформації з бази даних	08.12.22	
6	Оформлення пояснівальної записки	15.12.22	
7	Здача курсової роботи	17.12.22	

Дата видачі завдання «14» вересня 2022 р.

Завдання прийняв до виконання Ликова
(підпись)

К.О. Ликова

Керівник Б.В. Завгородній
(підпись)

Б.В. Завгородній

ЗМІСТ

ВСТУП.....	6
1 ТЕОРЕТИЧНА ЧАСТИНА	8
1.1 Аналіз предметної області.....	8
1.1.1 Опис предметної області	8
1.1.2 Опис вхідних даних	9
1.1.3 Опис вихідних даних	10
1.2 Проектування бази даних	10
1.2.1 Інфологічна модель бази даних	10
1.2.1.1 Опис сутностей.	11
1.2.1.2 Опис атрибутів	12
1.2.1.3 Опис зв’язків	14
1.2.1.4 Діаграма «сущність-зв’язок»	16
1.2.2 Нормалізація таблиць при проектуванні бази даних	18
1.2.3 Даталогічна модель бази даних	19
2 ПРАКТИЧНА ЧАСТИНА	23
2.1 Створення бази даних за допомогою MS SQL Server	23
2.1.1 Створення бази даних.....	23
2.1.2. Створення таблиць бази даних	25
2.1.3 Створення діаграми бази даних.....	27
2.1.4 Заповнення таблиць бази даних.....	28
2.2 Організація вибірки інформації з бази даних.....	30
2.2.1 Проста вибірка даних	31
2.2.2 Вибірка обчислюваних значень.....	33
2.2.3 Вибірка значень з діапазону	35
2.2.4 Вибірка з використанням шаблонів	35
2.2.5 Угруповання даних при організації запитів	36
2.2.6 Об’єднання таблиць	37

ВІСНОВКИ.....	39
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	40
ДОДАТКИ	42
Додаток А. Заповнення таблиць реляційної бази даних	42

ВСТУП

Останніми роками були досліджені та розвинуті системи бази даних (БД). Зараз все частіше вони змінюють роботу більшості організацій та відіграють одну з основних ролей в усіх інформаційних системах. Результатами невпинного розвитку баз даних стали потужні та зручні у використанні системи, які отримали більше поширення серед менш досвідчених користувачів. Врешті решт, люди почали власноруч створювати бази даних та додатки, але через оманливу простоту їх ефективності та якість були зовсім незадовільні. Розчарування користувачів такими результатами викликало явище «кризи програмного забезпечення», яке потребує вирішення і сьогодні.

База даних – це інформаційна модель, що уможливлює впорядковане збереження даних про групу об'єктів, яким притаманні однакові набори властивостей. Дуже розповсюденою практикою застосовувати таблиці для збереження даних у базах даних. На сьогоднішній час найпоширенішими є реляційні бази даних. У реляційних базах даних відбувається збереження інформації в одній або декількох таблицях, що зв'язані одним або декількома співпадаючими полями.

Реляційна база даних – це сукупність відношень, що складаються з інформації, яка потребує зберігання у БД. Для користувальника сприйняття така база даних являє собою сукупність пов'язаних між собою таблиць.

Ще одним напрямком у просторі досліджень та розробок систем управління базами даних (СУБД) є розподілені БД. На даний момент більшість розподілених СУБД базується на реляційній моделі даних і розрахована на використання в локальних обчислювальних мережах. У таких системах можуть виникати специфічні проблеми, які суттєво впливають на якість систем БД.

Головними завданнями системи *SQL Server* є:

- організація для великої кількості користувачів одночасного доступу до даних;
- маніпуляція інформацією, яка збережена у БД.

SQL Server підтримує реляційну модель даних та надає функціонал для створення об'єктів БД (таблиць, індексів, представлень), перевіряє цілісність БД та несе відповідальність за безпеку інформації, збереженої у системі.

Для доступу до даних зазвичай необхідний комп'ютер робочої станції. До того ж дозволяють виконувати операції над даними відповідні додатки.

Система *SQL Server* зберігає створені об'єкти у відповідні файли на дисках комп'ютера сервера. При цьому БД потребує створення спеціальних таблиць, в яких зберігається інформація про різні елементи бази даних: індекси, таблиці, користувачі, тощо.

В даній курсовій роботі була розроблена реляційна база даних «Рецепти» за допомогою *Microsoft SQL Server 2013*, в якій необхідно було вирішити наступні завдання:

- провести аналіз предметної області з описом вхідних та вихідних даних;
- побудувати інфологічну модель бази даних (опис сутностей, атрибутів та зв'язків між сутностями);
- провести нормалізацію таблиць при проектуванні бази даних;
- побудувати даталогічну модель бази даних;
- за допомогою СУБД *Microsoft SQL Server* створити базу даних, таблиці, діаграму бази даних;
- заповнити таблиці необхідними даними;
- виконати вибірку інформації з бази даних (просту, вибірку обчислювальних значень, з діапазону, з використанням шаблонів, при об'єднанні таблиць).

1 ТЕОРЕТИЧНА ЧАСТИНА

1.1 Аналіз предметної області

1.1.1 Опис предметної області

Основне призначення інформаційної системи (ІС) полягає у забезпеченні оперативною інформацією про зовнішній світ користувача шляхом реалізації питально-відповідного відношення.

Питально-відповідні відношення дозволяють виділити для ІС певний її фрагмент – *предметну область (ПрО)*, - який буде втілений в автоматизованій ІС.

Передача в ІС інформації про зовнішній світ відбувається у формі даних, що накладає обмеження на можливості змістової інтерпретації інформації й конкретизує семантику її подання в ІС. Інформаційна та функціональна моделі предметної області утворюються з сукупності виділених для ІС даних, зв'язків між ними та операцій над ними, та описують її стан з певною точністю. Інформаційна та функціональна моделі ПрО є вхідними даними для процесу проектування БД.

Своєчасний доступ до інформації є нагальною потребою будь-якої організації. На сьогоднішній день цінність інформації дуже висока. Зараз частіше за все роль розпорядників інформації відіграють бази даних. Бази даних забезпечують надійне зберігання інформації у структурованому вигляді та своєчасний доступ до неї. Через необхідність управління та адміністрування даних неможливо уявити жодне підприємство без застосування інформаційних систем.

При розробці реляційної бази даних «Рецепти» було проведено дослідження предметної області. Основними операціями в досліджуваній предметній області є складання рецепту [1-3].

Рецепти складаються з різних інгредієнтів, потребують певне обладнання, час, вони розраховані на різну кількість порцій та в результаті отримана страва має деякий смак. Велика кількість рецептів змушує довго шукати для знаходження саме того, що задовольнить всі бажання користувача. Саме тому основною метою даної курсової роботи є автоматизація цього процесу задля скорочення часу оператора на обробку даних.

1.1.2 Опис вхідних даних

Для кожного виду вхідних даних потрібно вказати форму подання, періодичність надходження, джерело виникнення. Якщо вхідні дані зафіксовані у первинних документах, то вказується найменування документа, періодичність його надходження на обробку. Якщо вхідні дані є результатом розв'язання інших задач і зберігаються на машинних носіях, то вони описуються із зазначенням найменування задачі (підпрограми), під час вирішення якої були одержані.

При розробці реляційної бази даних «Рецепти» були виділені наступні вхідні дані:

- інформація про м'ясо;
- інформація про овочі;
- інформація про спеції;
- інформація про інгредієнти;
- інформація про обладнання;
- інформація про процеси;
- інформація про рецепти.

1.1.3 Опис вихідних даних

Для кожного виду вихідних повідомлень необхідно вказувати форму подання, періодичність видання, користувача інформації. Вихідні повідомлення залежно від конкретної задачі можуть бути подані у вигляді набору даних на машинному носії – як проміжні дані для нагромадження, зберігання та використання під час роботи інших програм або розв'язання інших задач. У цьому випадку вони описуються із зазначенням найменування задачі (підпрограми), під час розв'язання якої будуть використовуватися.

Для вихідної інформації, що подається у формі документа, необхідно привести зразок форми документа та його опис, охарактеризувати порядок його оформлення.

Вихідними даними для бази даних «Рецепти» є вихідні запити.

1.2 Проектування бази даних

1.2.1 Інфологічна модель бази даних

Мета інфологічного проектування полягає у забезпеченні найбільш природних для людини способів збору і представлення тієї інформації, яку передбачено зберігати у створеній БД. Саме через це основою побудови інфологічної моделі є природня мова. Інфологічні моделі складаються з таких основних конструктивних елементів: сутності, зв'язки між ними та їх властивості [4, 7-9].

Створення інфологічної моделі відбувається за результатами проведених досліджень ПрО.

Інфологічна модель являє собою опис майбутньої бази даних, подача якого здійснюється за допомогою природної мови, формул, графіків, діаграм, таблиць та інших засобів, зрозумілих як розробникам БД, так і звичайним користувачам.

Призначенням цієї моделі є адекватний опис процесів, інформаційних потоків, функцій системи за допомогою загальнодоступної й зрозумілої мови, що уможливлює залучення експертів предметної області, консультантів, а також користувачів для обговорення моделі та її коригування.

Інфологічна модель є природнім продовженням досліджень предметної області, відрізняється тим, що представляє БД з точки зору проектувальника (розробника). Таке представлення моделі надає необхідну наочність для оцінки її точності та внесення виправлень експертами ПрО.

Важливо звернути увагу на те, що створювана на цьому етапі модель повністю незалежна від фізичної реалізації майбутньої системи. Таке моделювання структур даних називається *семантичним* та виходить з їх смислового призначення.

На даний момент найбільшою поширеністю володіє спосіб опису інфологічної моделі, заснований на застосуванні діаграм «сущність-зв'язок». Інфологічна модель «сущність-зв'язок» (entity relationship model; ER-model) П. Чена – це описова (неформальна) модель предметної області, що семантично визначає в ній сутності та зв'язки.

Такій моделі притаманна відносна простота та наочність опису предметної області, що робить можливим її використання під час діалогу з потенційним користувачем із самого початку інфологічного проектування. Процес побудови інфологічної моделі П. Чена, так само як і інших моделей, є творчим, тому немає обмежень у методиці її створення. Водночас у будь-якому разі під час побудови моделі потрібно реалізувати три головних конструктивні елементи: сутність, атрибут та зв'язок.

1.2.1.1 Опис сутностей.

Базовими елементами моделі «сущність-зв'язок» є сутності. *Сутність (Entity)* представляє із себе множину екземплярів реальних або абстрактних об'єктів, яким притаманні загальні атрибути або характеристики. Кожна сутність повинна бути

унікально ідентифікована та кожен об'єкт системи може бути представлений тільки однією сутністю. При цьому назва сутності має відображати клас або тип об'єкта замість конкретного екземпляру.

У відповідності з описом предметної області були отримані такі сутності:

- «М'ясо» - зберігається інформація про різні види м'яса;
- «Овочі» - зберігається інформація про різні овочі;
- «Спеції» - зберігається інформація про різні спеції;
- «Інгредієнти» - зберігається інформація про набір інгредієнтів, необхідних для рецепту;
- «Обладнання» - зберігається інформація про обладнання необхідне для приготування страви за рецептами;
- «Процес» - зберігається інформація про процес приготування;
- «Рецепти» - зберігається інформація про рецепти.

1.2.1.2 Опис атрибутів

Атрибут (Attribute) – це будь-яка характеристика сутності, що має значення для заданої предметної області та призначення якої полягає у кваліфікації, ідентифікації, класифікації, кількісній характеристиці або вираженні стану сутності.

Екземпляр атрибута – це певна характеристика окремого елемента множини. Екземпляр атрибута складається з типу характеристики та її значення, тобто приймає так зване значення атрибута.

Реляційні таблиці надають можливість реалізації зв'язків між ними задля отримання даних з декількох таблиць одночасно та зменшення обсягу БД. Зв'язокожної пари таблиць реалізується за допомогою полів, що мають одинаковий формат і назву.

У відповідності з описом предметної області були виділені наступні атрибути у кожній сутності:

1. Таблиця *М'ясо (Meat)* містить:

IdMeat – унікальний код м'яса

TypeMeat – тип м'яса

ManufacturerMeat – країна-виробник м'яса

2. Таблиця *Овочі (Vegenables)* містить:

IdVegetables – унікальний код овоча

NameVegetables – назва овоча

ManufacturerVegetables – країна-виробник овоча

3. Таблиця *Спеції (Spices)* містить:

IdSpices – унікальний код спеції

NameSpices – назва спеції

ManufacturerSpices – країна-виробник спеції

4. Таблиця *Інгредієнти (Ingredients)* містить:

IdIngredients – унікальний код набору інгредієнтів

MassGramMeatIngredients – маса у грамах м'яса у наборі інгредієнтів

MassGramVegetablesIngredients – маса у грамах овоча у наборі інгредієнтів

MassGramSpicesIngredients – маса у грамах спеції у наборі інгредієнтів

IdMeat – унікальний код м'яса

IdVegetables – унікальний код овоча

IdSpices – унікальний код спеції

5. Таблиця *Обладнання (Equipment)* містить:

IdEquipment – унікальний код обладнання

TypeEquipment – тип обладнання

NameEquipment – назва моделі обладнання

ManufacturerEquipment – країна-виробник обладнання

6. Таблиця *Процес (Process)* містить:

IdProcess – унікальний код процесу приготування

TypeProcess – тип процесу приготування

TimeMinutesProcess – час у хвилинах процесу приготування

IdEquipment – унікальний код обладнання

7. Таблиця *Рецепти* (*Recipes*) містить:

IdRecipes – унікальний код рецепта

NameRecipes – назва рецепта

TasteRecipes – смак страви рецепта

NumberPortionsRecipes – кількість порцій страви рецепта

IdIngredients – унікальний код набору інгредієнтів

IdProcess – унікальний код процесу приготування

1.2.1.3 Опис зв'язків

Зв'язок – це асоціювання двох і більше сутностей [10,11]. За такої асоціації кожен екземпляр однієї сутності асоційований із довільною (у тому числі нульовою) кількістю екземплярів іншої сутності, і навпаки.

Переважна більшість засобів проектування даних надають інструменти для створення ER-діаграми, де візуально зображені сутності та їхні зв'язки одна з одною. Простота інтерфейсу подібних засобів уможливлює участь користувача, як експерта в предметній області, у логічному проектуванні даних.

Взаємозв'язки між таблицями реляційної бази даних можуть бути типізовані за такими основними видами [5,6]:

- відношення «один до одного» (1:1) означає, що кожному запису однієї таблиці відповідає тільки один запис в іншій таблиці;
- відношення «один до багатьох» (1:Б) виникає, коли один запис взаємопов'язаний з багатьма іншими;
- відношення «багато до одного» (Б:1) означає, що багато записів пов'язані з одним;
- відношення «багато до багатьох» (Б:Б) виникає між двома таблицями в тих випадках, коли:

- один запис із першої таблиці може бути пов'язаний більш ніж з одним записом із другої таблиці;
- один запис з другої таблиці може бути пов'язаний більш ніж з одним записом першої таблиці.

В курсовій роботі були використані наступні типи зв'язків між таблицями реляційної бази даних «Рецепти» (табл. 1.1).

Таблиця 1.1 – Типи зв'язків

Номер зв'язку	Головна таблиця	Дочірня таблиця	Тип зв'язку
1	М'ясо (<i>Meat</i>)	Інгредієнти (<i>Ingredients</i>)	1:Б
2	Овочі (<i>Vegetables</i>)	Інгредієнти (<i>Ingredients</i>)	1:Б
3	Спеції (<i>Spices</i>)	Інгредієнти (<i>Ingredients</i>)	1:Б
4	Обладнання (<i>Equipment</i>)	Процес (<i>Process</i>)	1:1
5	Інгредієнти (<i>Ingredients</i>)	Рецепти (<i>Recipes</i>)	1:Б
6	Процес (<i>Process</i>)	Рецепти (<i>Recipes</i>)	1:Б

Перший зв'язок «М'ясо (*Meat*) - Інгредієнти (*Ingredients*)» вказує на те, що один тип м'яса може використовуватися у багатьох наборах інгредієнтів.

Другий зв'язок «Овочі (*Vegetables*) - Інгредієнти (*Ingredients*)» вказує на те, що один овоч може використовуватися у багатьох наборах інгредієнтів.

Третій зв'язок «Спеції (*Spices*) - Інгредієнти (*Ingredients*)» вказує на те, що одна спеція може використовуватися у багатьох наборах інгредієнтів.

Четвертий зв'язок «Обладнання (*Equipment*) - Процес (*Process*)» вказує на те, що одне обладнання може використовуватися у одному процесі приготування.

П'ятий зв'язок «Інгредієнти (*Ingredients*) - Рецепти (*Recipes*)» вказує на те, що один набір інгредієнтів може використовуватися у багатьох рецептах.

Шостий зв'язок «Процес (*Process*) - Рецепти (*Recipes*)» вказує на те, що один набір процес може використовуватися у багатьох рецептах.

1.2.1.4 Діаграма «сутність-зв'язок»

На рисунку 2.1 представлена ER-діаграма бази даних «Рецепти», на якій відображені всі сущності, їх атрибути та зв'язки між сущностями.

Позначення:



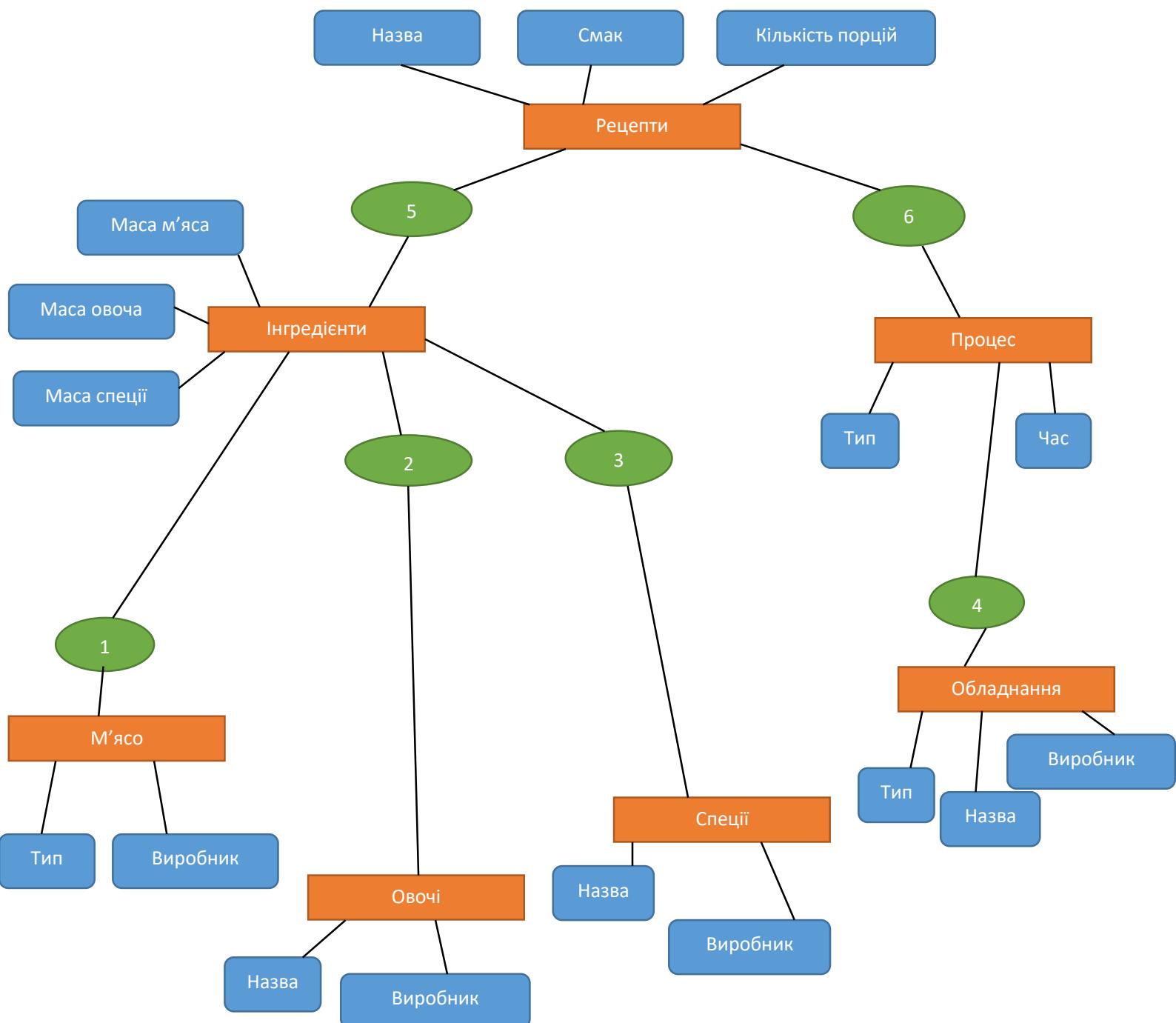


Рисунок 2.1 – ER-діаграма бази даних «Рецепти»

1.2.2 Нормалізація таблиць при проектуванні бази даних

Нормалізація – це розбиття таблиці на дві або більше, що володіють кращими властивостями при включені, зміні і видаленні даних.

У теорії нормалізації існує п'ять нормальних форм (НФ) таблиць. Їхнє призначення полягає у зменшенні надлишкової інформації від першої до п'ятої нормальній форми. Тому кожна наступна НФ повинна задовольняти вимогам попередньої форми і деяким додатковим умовам.

Проведемо нормалізацію наявних сутностей.

Таблиця у першій НФ вимагає, щоб всі значення атрибутів були атомарними, що означає збереження одного-єдиного значення.

Таблиця знаходитьться в другій НФ, якщо вона задовольняє умови першої НФ, і кожний не первинний атрибут повністю функціонально залежить від ключа.

Таблиця знаходитьться в третьій НФ, якщо вона задовольняє умовам другої НФ, і кожний не первинний атрибут не транзитивно залежить від ключа.

Таким чином, щоб привести відношення до 3 НФ дані, що знаходяться в таблиці, повинні мати залежність тільки від ключа.

Процес проектування реляційної бази даних, зазвичай, завершується приведенням до третьої нормальної форми, тому дана модель не потребує подальшого приведення до четвертої форми нормалізації.

Проведемо нормалізацію таблиць:

Дляожної таблиці необхідно обрати первинний ключ:

- для таблиці М'ясо (*Meat*) – поле «унікальний код м'яса – *IdMeat*»;
- для таблиці Овочі (*Vegetables*) – поле «унікальний код овоча – *IdVegetables*»;
- для таблиці Спеції (*Spices*) – поле «унікальний код спеції – *IdSpices*»;
- для таблиці Інгредієнти (*Ingredients*) – поле «унікальний код набору інгредієнтів – *IdIngredients*»;
- для таблиці Обладнання (*Equipment*) – поле «унікальний код обладнання – *IdEquipment*»;

- для таблиці Процес (*Process*) – поле «унікальний код процесу – *IdProcess*»;
- для таблиці Рецепти (*Recipes*) – поле «унікальний код рецепта – *IdRecipes*».

1.2.3 Даталогічна модель бази даних

На етапі даталогічного проектування здійснюється перехід від інфологічної моделі ПрО до логічної моделі, яка підтримується засобами конкретних СУБД, і процес цього переходу називається *відображенням*.

Перш ніж виконати даталогічне проектування, необхідно обрати СУБД. Основні фактори, що впливають на даталогічне проектування з боку СУБД:

- тип логічної моделі, що підтримує обрана СУБД (найпоширеніша реляційна, а також ієрархічна та сіткова);
- особливості фізичної організації даних у середовищі обраної СУБД;
- кількісні обмеження, які накладає СУБД.

Для кожного поля таблиці реляційної бази даних «Рецепти» (табл. 1.2 – 1.8) вказується розмір поля (кількість символів), тип даних та обмеження для кожного поля, якщо необхідно.

Для первинних ключів необхідно ввести заборону невизначених значень. Для інших полів – можливість заборони невизначених значень визначається семантикою предметної області [1].

Таблиця 1.2 – Таблиця М’ясо (*Meat*)

Найменування атрибутів		Тип поля	Розмір поля	Обмеження
Унікальний код м’яса	<i>IdMeat</i>	Int	4	Not null
Тип м’яса	<i>TypeMeat</i>	Varchar	30	
Виробник м’яса	<i>ManufacturerMeat</i>	Varchar	30	

Таблиця 1.3 – Таблиця Овочі (*Vegetables*)

Найменування атрибутів		Тип поля	Розмір поля	Обмеження
Унікальний код овоча	IdVegetables	Int	4	Not null
Назва овоча	NameVegetables	Varchar	30	
Виробник овоча	ManufacturerVegetables	Varchar	30	

Таблиця 1.4 – Таблиця Спеції (*Spices*)

Найменування атрибутів		Тип поля	Розмір поля	Обмеження
Унікальний код спеції	IdSpices	Int	4	Not null
Назва спеції	NameSpices	Varchar	30	
Виробник спеції	ManufacturerSpices	Varchar	30	

Таблиця 1.5 – Таблиця Інгредієнти (*Ingredients*)

Найменування атрибутів		Тип поля	Розмір поля	Обмеження
Унікальний код набору інгредієнтів	IdIngredients	Int	4	Not null
Маса м'яса	MassGramMeatIngredients	Smallint	5	
Маса овоча	MassGramVegetablesIngredients	Smallint	5	
Маса спеції	MassGramSpicesIngredients	Smallint	5	
Унікальний код м'яса	IdMeat	Int	4	Not null
Унікальний код овоча	IdVegetables	Int	4	Not null
Унікальний код спеції	IdSpices	Int	4	Not null

Таблиця 1.6 – Таблиця Обладнання (*Equipment*)

Найменування атрибутів		Тип поля	Розмір поля	Обмеження
Унікальний код обладнання	IdEquipment	Int	4	Not null
Тип обладнання	TypeEquipment	Varchar	30	
Назва обладнання	NameEquipment	Varchar	30	
Виробник обладнання	ManufacturerEquipment	Varchar	30	

Таблиця 1.7 – Таблиця Процес (*Process*)

Найменування атрибутів		Тип поля	Розмір поля	Обмеження
Унікальний код процесу	IdProcess	Int	4	Not null
Тип процесу	TypeProcess	Varchar	30	
Час процесу	TimeMinutesProcess	Smallint	5	
Унікальний код обладнання	IdEquipment	Int	4	Not null

Таблиця 1.8 – Таблиця Рецепти (*Recipes*)

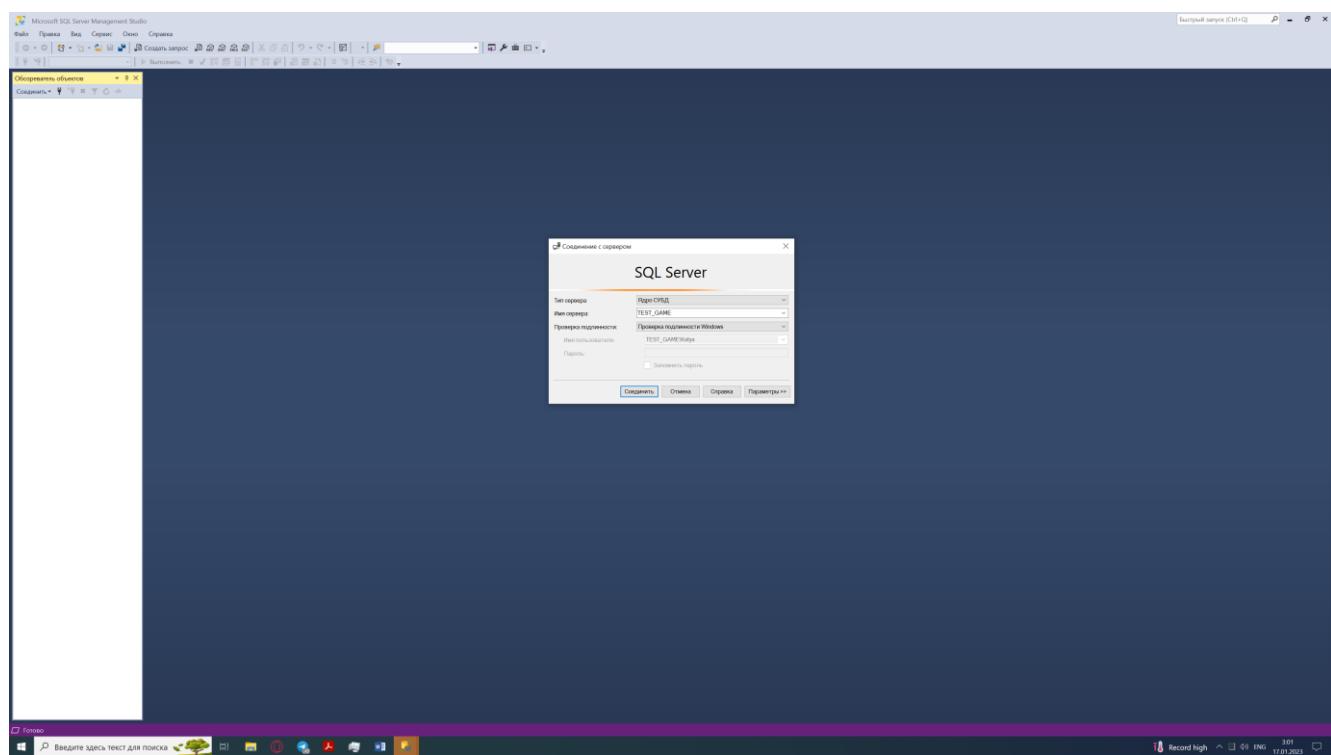
Найменування атрибутів		Тип поля	Розмір поля	Обмеження
Унікальний код рецепту	IdRecipes	Int	4	Not null
Назва рецепту	NameRecipes	Varchar	30	
Сmak	TasteRecipes	Varchar	30	
Кількість порцій	NumberPortionsRecipes	Smallint	5	
Унікальний код інгредієнтів	IdIngredients	Int		Not null
Унікальний код процесу	IdProcess	Int		Not null

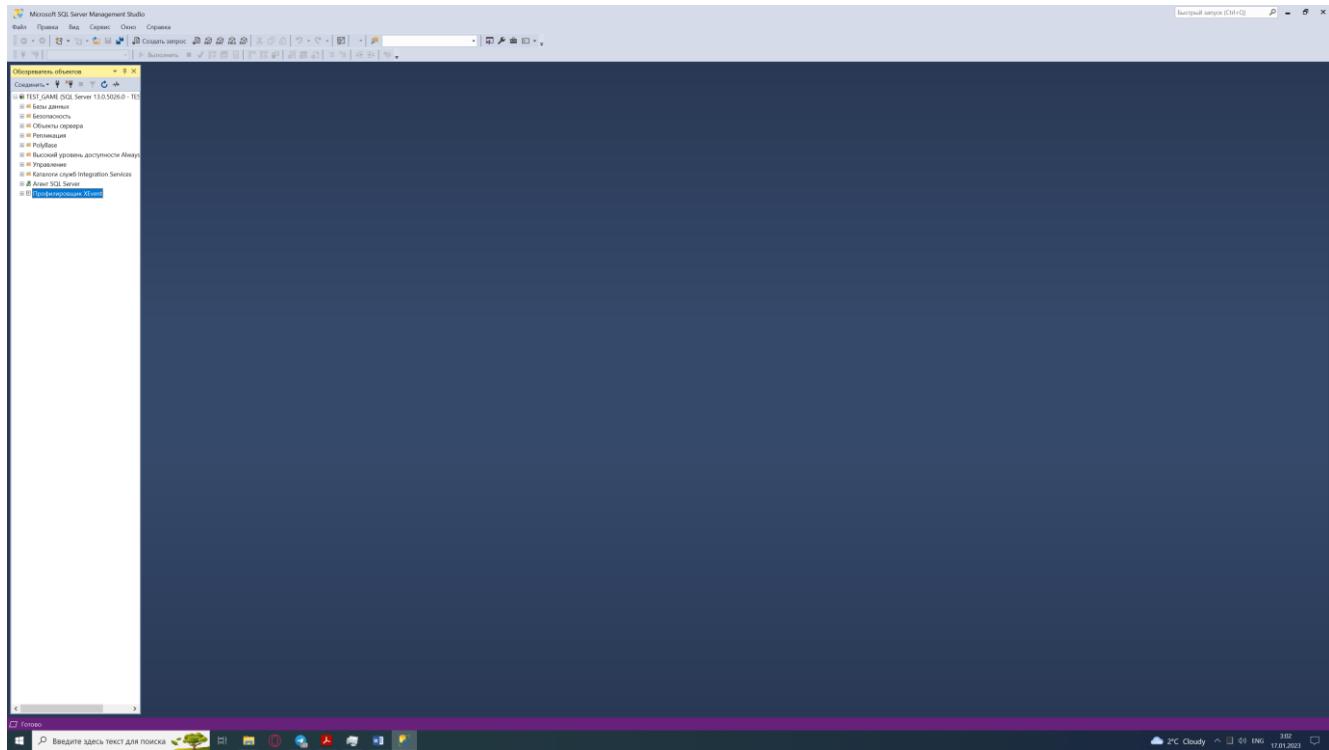
2 ПРАКТИЧНА ЧАСТИНА

2.1 Створення бази даних за допомогою MS SQL Server

2.1.1 Створення бази даних

Для запуску *MS SQL Server 2013* вибираємо утиліту *SQL Server Management Studio* та підключаємося до серверу:





Далі за допомогою *sql*-команд створюємо реляційну базу даних із назвою

Recipes2 [6]:

CREATE DATABASE Recipes2

Результат зображенено на рисунку 2.1.

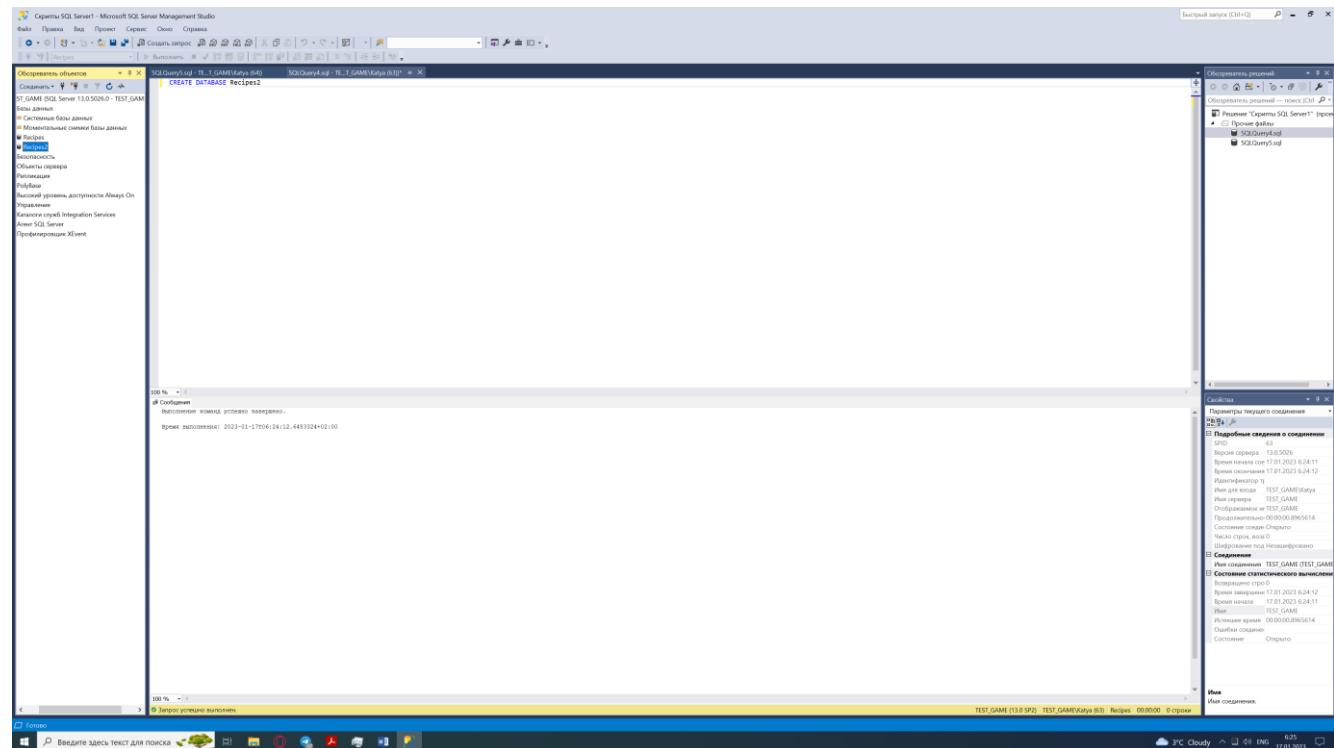


Рисунок 2.1 – Створення реляційної бази даних «Рецепти»

2.1.2. Створення таблиць бази даних

В базі даних *Recipes2* створюємо наступні таблиці:

- м'ясо (*Meat*),
- овочі (*Vegetables*),
- спеції (*Spices*),
- обладнання (*Equipment*),
- інгредієнти (*Ingredients*),
- процес (*Process*),
- рецепти (*Recipes*)

за допомогою наступних команд (рис. 2.2) [12,13]:

--Створення таблиці М'ясо (*Meat*):

```
create table Meat (
IdMeat int primary key,
TypeMeat varchar(30),
ManufacturerMeat varchar(30));
```

--Створення таблиці Овочі (*Vegetables*):

```
create table Vegetables (
IdVegetables int primary key,
NameVegetables varchar(30),
ManufacturerVegetables varchar(30));
```

--Створення таблиці Спеції (*Spices*):

```
create table Spices (
IdSpices int primary key,
NameSpices varchar(30),
ManufacturerSpices varchar(30));
```

--Створення таблиці Обладнання (*Equipment*):

```
create table Equipment (
IdEquipment int primary key,
TypeEquipment varchar(30),
NameEquipment varchar(30),
ManufacturerEquipment varchar(30));
```

--Створення таблиці Інгредієнти (*Ingredients*):

```
create table Ingredients (
```

```

IdIngredients int primary key,
MassGramMeatIngredients smallint,
MassGramVegetablesIngredients smallint,
MassGramSpicesIngredients smallint,
IdMeat int foreign key references Meat(IdMeat),
IdVegetables int foreign key references
Vegetables(IdVegetables),
IdSpices int foreign key references Spices(IdSpices));

--Створення таблиці Процес (Process):
create table Process (
IdProcess int primary key,
TypeProcess varchar(30),
TimeMinutesProcess smallint,
IdEquipment int foreign key references
Equipment(IdEquipment));

--Створення таблиці Рецепти (Recipes):
create table Recipes (
IdRecipes int primary key,
NameRecipes varchar(30),
TasteRecipes varchar(30),
NumberPortionsRecipes smallint,
IdIngredients int foreign key references
Ingredients(IdIngredients),
IdProcess int foreign key references Process(IdProcess));

```

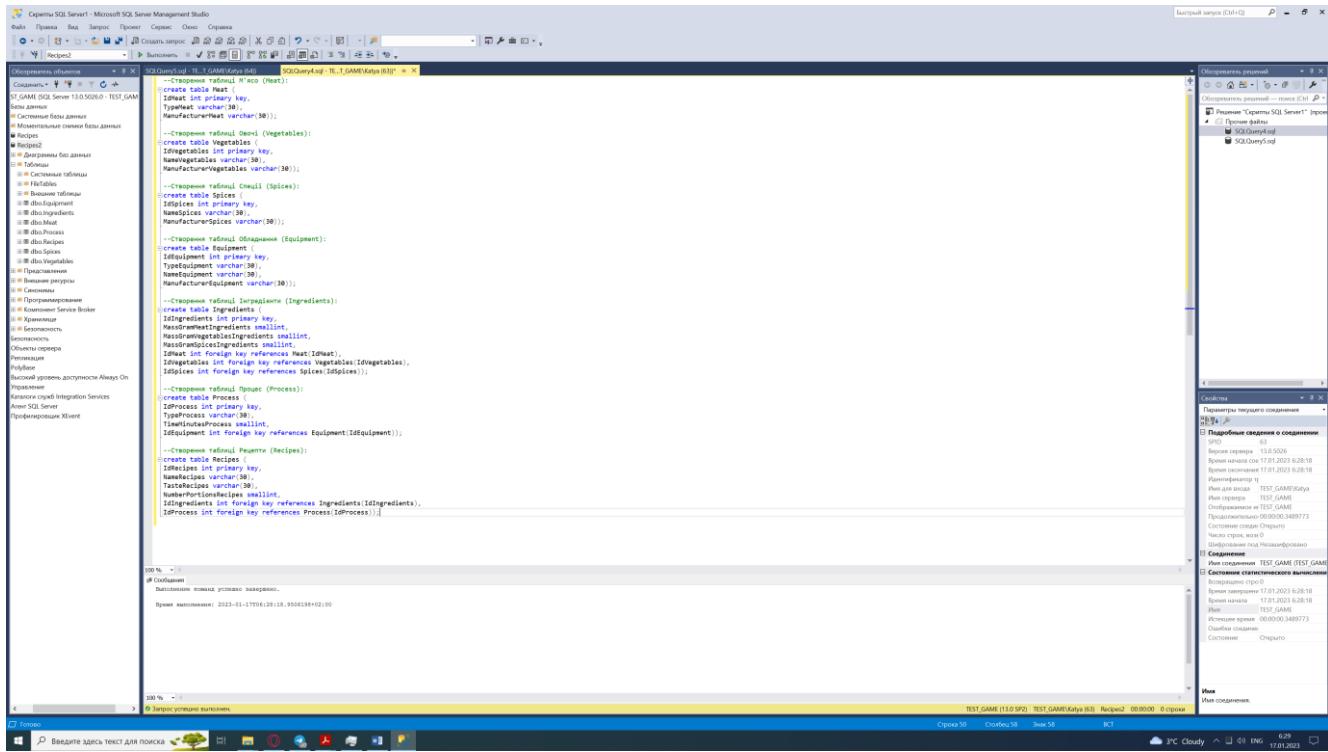


Рисунок 2.2 – Створення таблиць реляційної бази даних «Рецепти»

2.1.3 Створення діаграми бази даних

У розділі діаграм в утиліті *SQL Server Management Studio* створюємо нову діаграму, в яку додаємо зі списку сім таблиць із даної предметної області та перевіряємо зв'язки між таблицями.

Результат створення діаграми зображенено на рисунку 2.3.

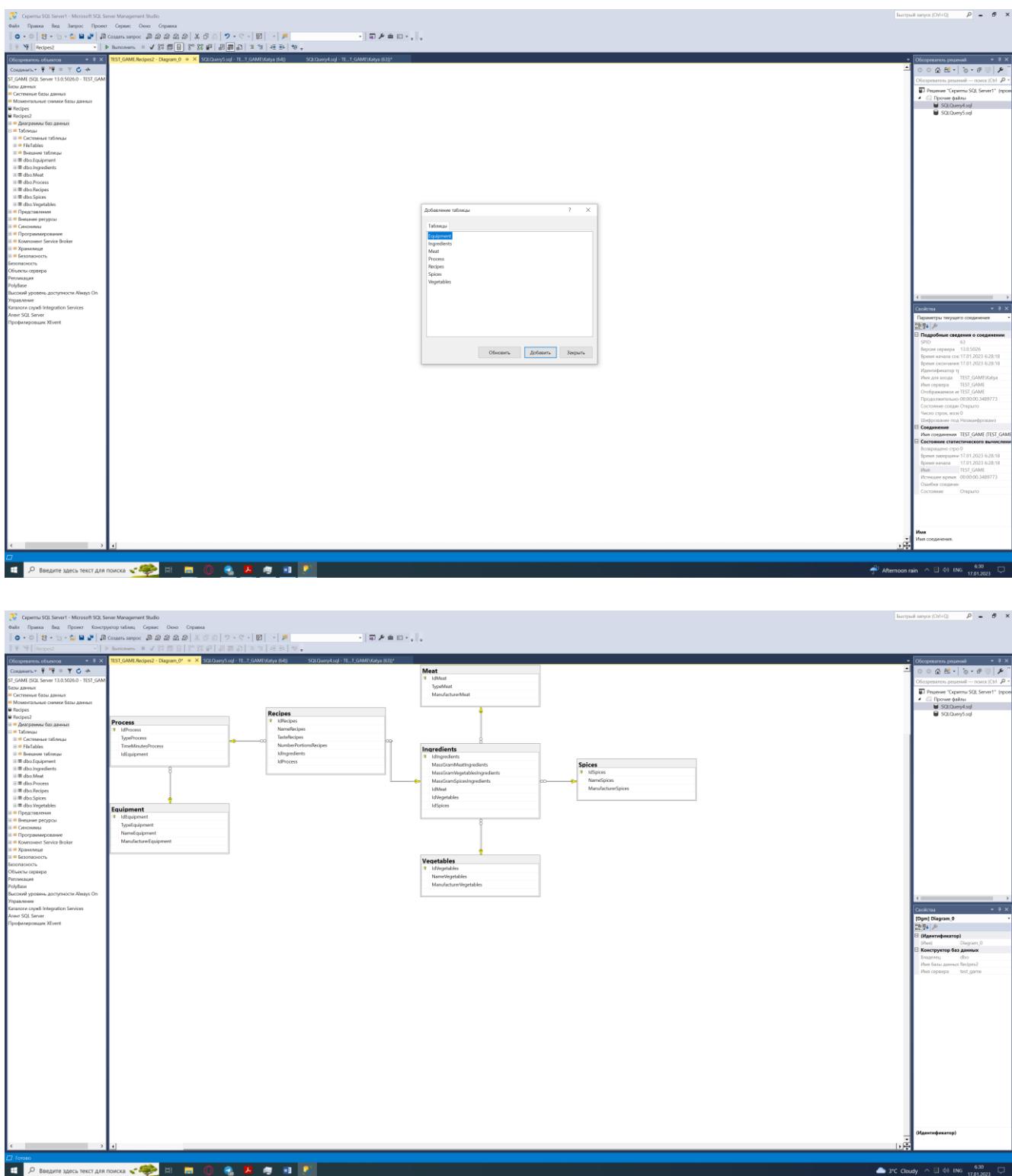


Рисунок 2.3 – Діаграма реляційної бази даних «Рецепти»

2.1.4 Заповнення таблиць бази даних даними

Заповнимо таблицю «М’ясо (Meat)» даними (рис.2.4):

```
use Recipes2
INSERT INTO Meat
VALUES(1, 'Куриця', 'Італія'),
      (2, 'Телятина', 'Україна'),
      (3, 'Свинина', 'Польща'),
      (4, 'Баранина', 'Велика Британія'),
      (5, 'Качка', 'Китай');
```

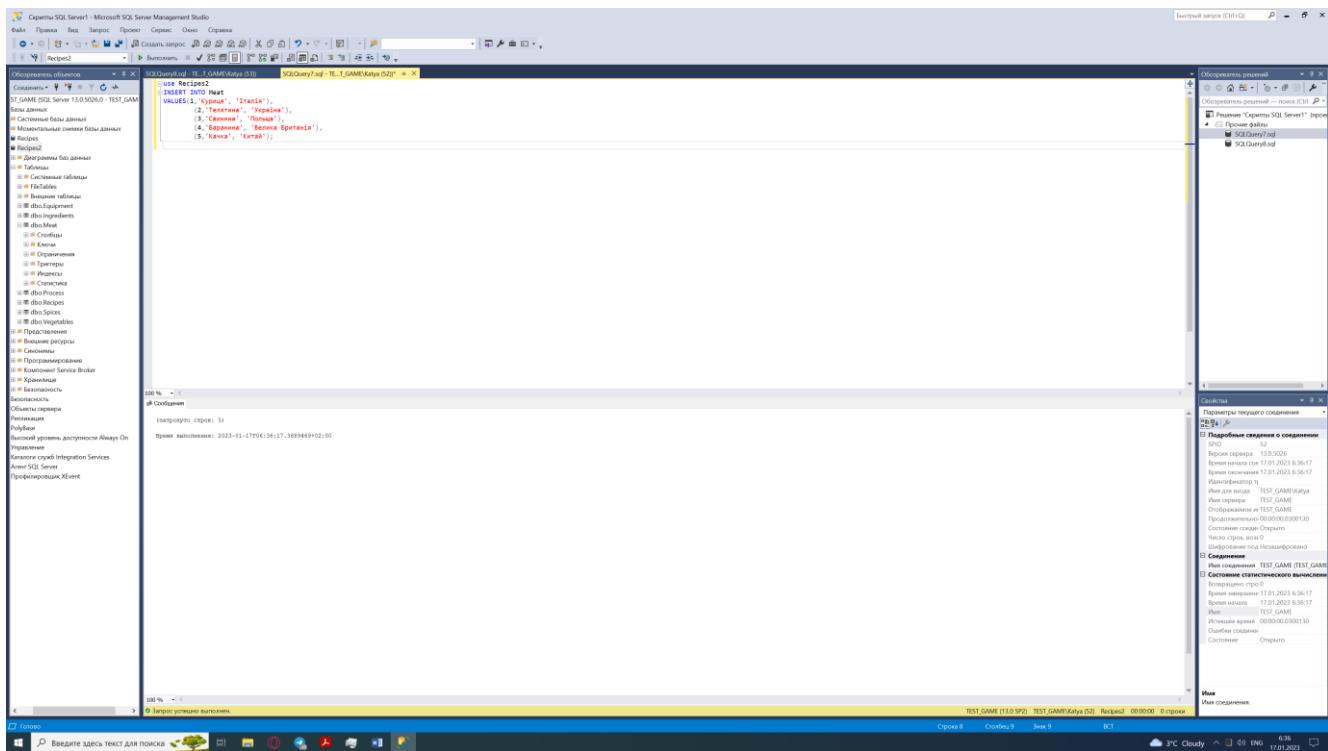
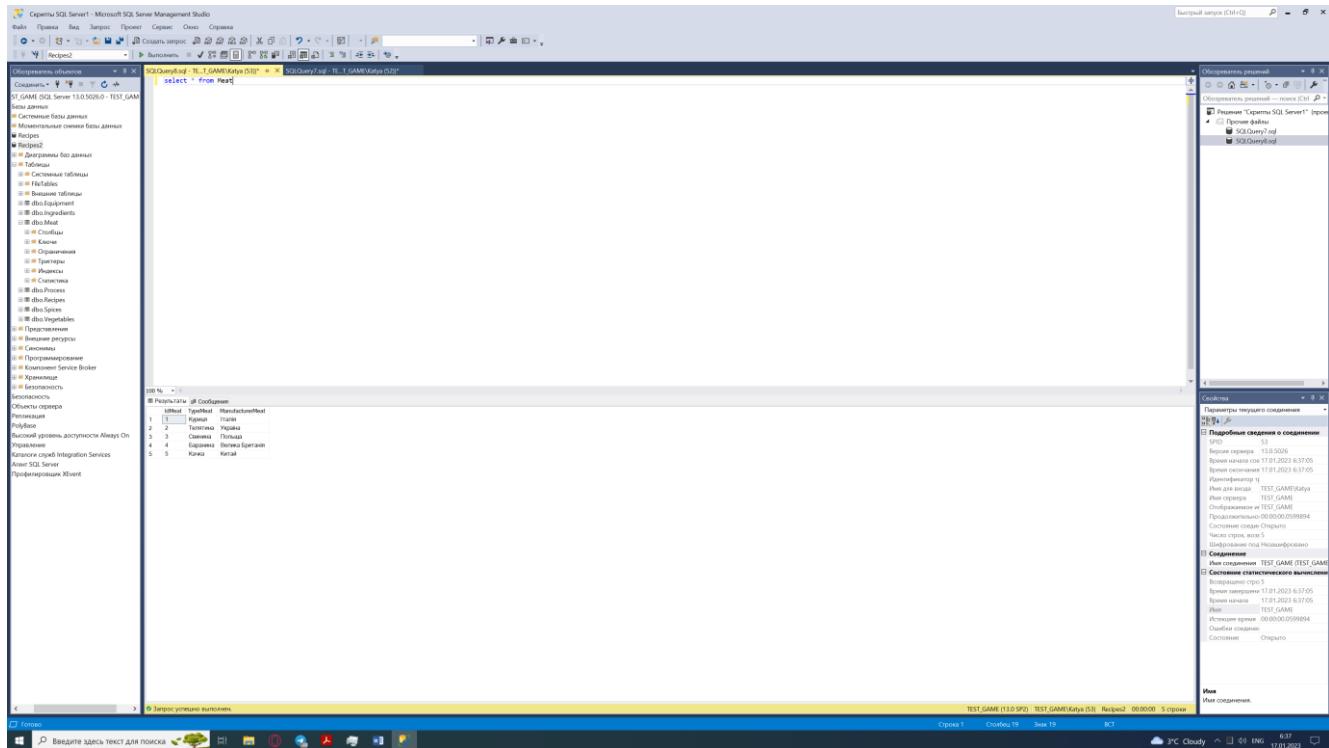


Рисунок 2.4 – Вставка даних для таблиці м’ясо (Meat)

Переглянемо результат:

```
select * from Meat
```



Далі заповнюємо інші таблиці реляційної бази даних «Рецепти». (Додаток А

- Заповнення таблиць реляційної бази даних).

2.2 Організація вибірки інформації з бази даних

Існує багато засобів вибірки даних з таблиць бази даних, серед яких одним з найбільш ефективних та універсальних є використання запитів *SQL*. Їхньою найбільш цінною відмінністю від фільтрів є можливість виконання не тільки інформаційно-довідкової функції з відображенням необхідних даних з таблиць, а й деякого аналізу даних.

Більш того, запити беруть на себе складні операції маніпулювання даними: наприклад, запити на зміну дозволяють видалити відразу кілька записів, що підпадають під певні задані умови, створити нову таблицю за результатами запиту або скопіювати дані з однієї таблиці в іншу.

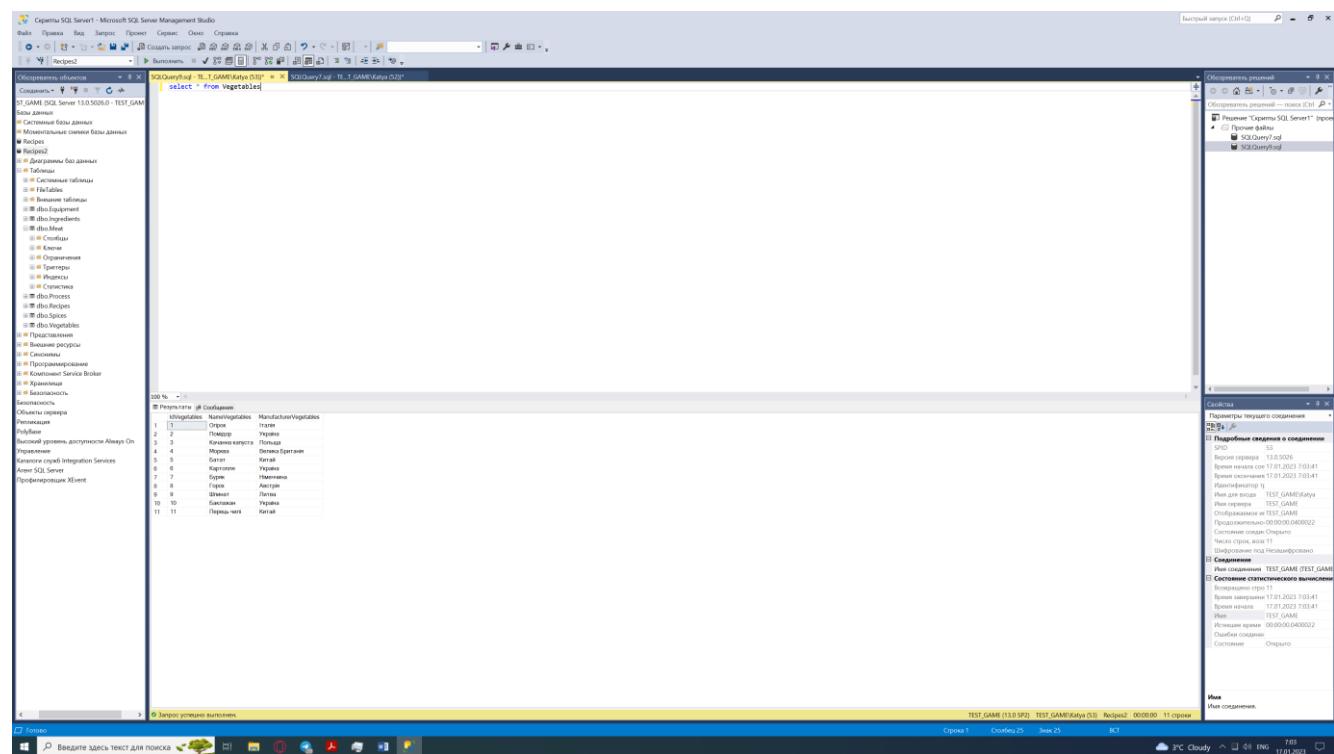
Завдяки запитам є можливим збир інформації воєдино з однієї або декількох таблиць через зв'язки, встановлені між ними.

Під час процесу відбору даних з таблиці отримуються деякі її поля та їх значення, що задовольняють певним умовам. Внаслідок виконання запиту, на основі якого проводиться вибір даних, отримується *вибірка*. Є можливість обирати дані з однієї або декількох таблиць з допомогою оператора SELECT.

2.2.1 Проста вибірка даних

Написати запит, який би видав усі дані про овочі:

```
select * from Vegetables
```



Написати запит, який би видав список рецептів із зазначенням їх назви, смаку та кількості порцій:

```
select NameRecipes as 'Назва рецепту',
TasteRecipes as 'Смак рецепту',
NumberPortionsRecipes as 'Кількість порцій рецепту'
from Recipes
```

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window displays the following T-SQL code:

```
select NameRecipes as 'Назва рецепту',
       TasteRecipes as 'Відмінність',
       NumberPortionsRecipes as 'Кількість порцій рецепту'
  from Recipes
```

The results grid shows the following data:

	Назва рецепту	Свіжий	Кількість порцій рецепту
1	Помідори	Помідори	1
2	Качана картопля	Салат	2
3	Зелений	Приємно	3
4	Лук	Салат	3
5	Фруктова компотида	Салат	1
6	Сир	Приємно	1
7	Сирний сир	Приємно	1
8	Помаранчевий джем	Приємно	2
9	Гранатовий коктейль	Салат	2
10	Помідори з кетчупом	Салат	4

The status bar at the bottom indicates: Строки 4, Страницы 1, Запись 11, ВСТ, 17.01.2023.

Написати запит, який би видав інформацію про обладнання (тип, назва, виробник) та упорядкувати дані в алфавітному порядку за полем «Тип».

```
select Equipment.TypeEquipment as 'Тип обладнання',
       Equipment.NameEquipment as 'Назва обладнання',
       Equipment.ManufacturerEquipment as 'Виробник обладнання'
  from Equipment
 order by Equipment.TypeEquipment
```

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window displays the following T-SQL code:

```
select Equipment.TypeEquipment as 'Тип обладнання',
       Equipment.NameEquipment as 'Назва обладнання',
       Equipment.ManufacturerEquipment as 'Виробник обладнання'
  from Equipment
 order by Equipment.TypeEquipment
```

The results grid shows the following data:

	Тип обладнання	Назва обладнання	Виробник обладнання
1	Картридж	Буле	Буле
2	Сенсорний дисплей	Фуджі ТМК	Фуджі
3	Сенсорний дисплей	Fujitsu Norden	Фуджі
4	Шайфтушка	BOSCH HBS200YTSQ	Фуджі

The status bar at the bottom indicates: Строки 4, Страницы 1, Запись 33, ВСТ, 17.01.2023.

Написати запит, який би видав список овочів, у яких «Унікальний код овоча» знаходитьться в діапазоні від 4 до 9. Відсортувати запит за назвою овоча.

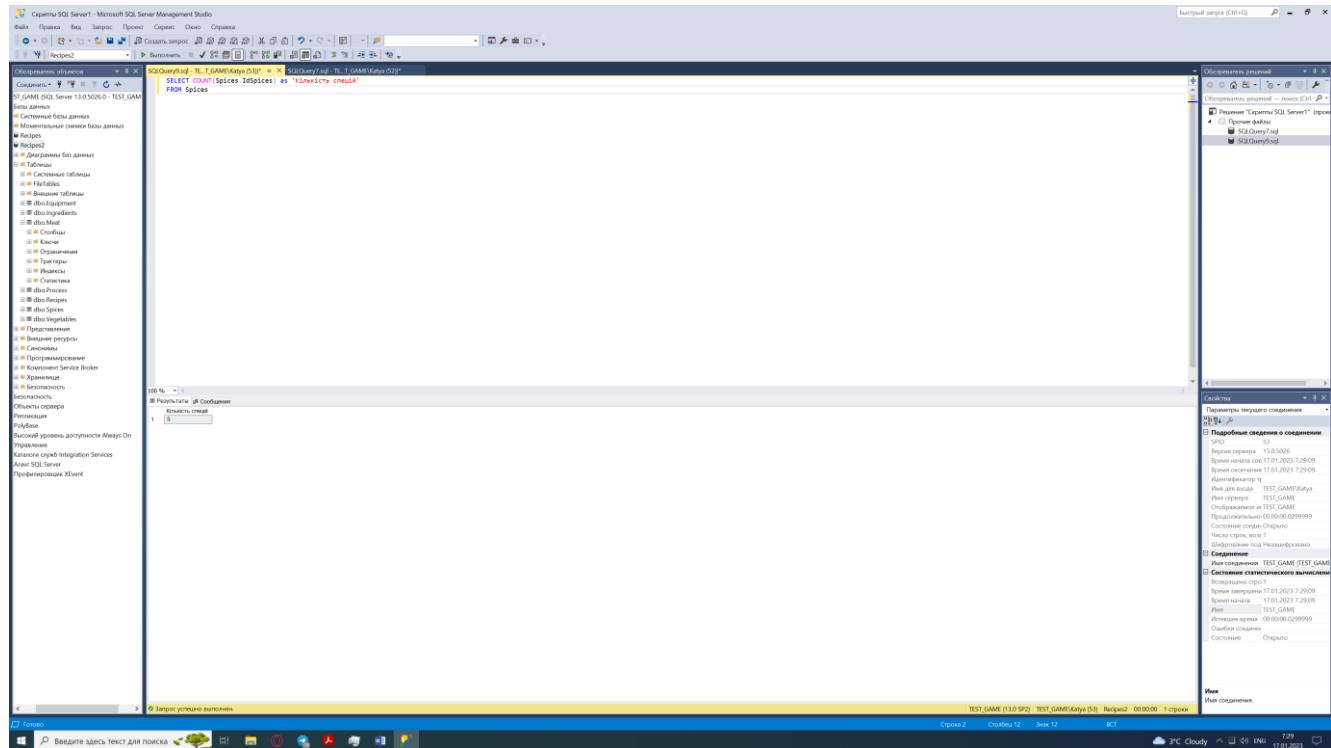
```
select Vegetables.IdVegetables as 'Унікальний код овочів',
Vegetables.NameVegetables as 'Назва овоча'
from Vegetables
where (Vegetables.IdVegetables >= 4) and
(Vegetables.IdVegetables <= 9)
order by Vegetables.NameVegetables
```

IdVegetables	NameVegetables
1	Баклажан
2	Гарбуз
3	Горох
4	Картофель
5	Морква
6	Шпинат

2.2.2 Вибірка обчислюваних значень

Написати запит, який визначив би кількість спецій:

```
SELECT COUNT(Spices.IdSpices) as 'Кількість спецій'
FROM Spices
```

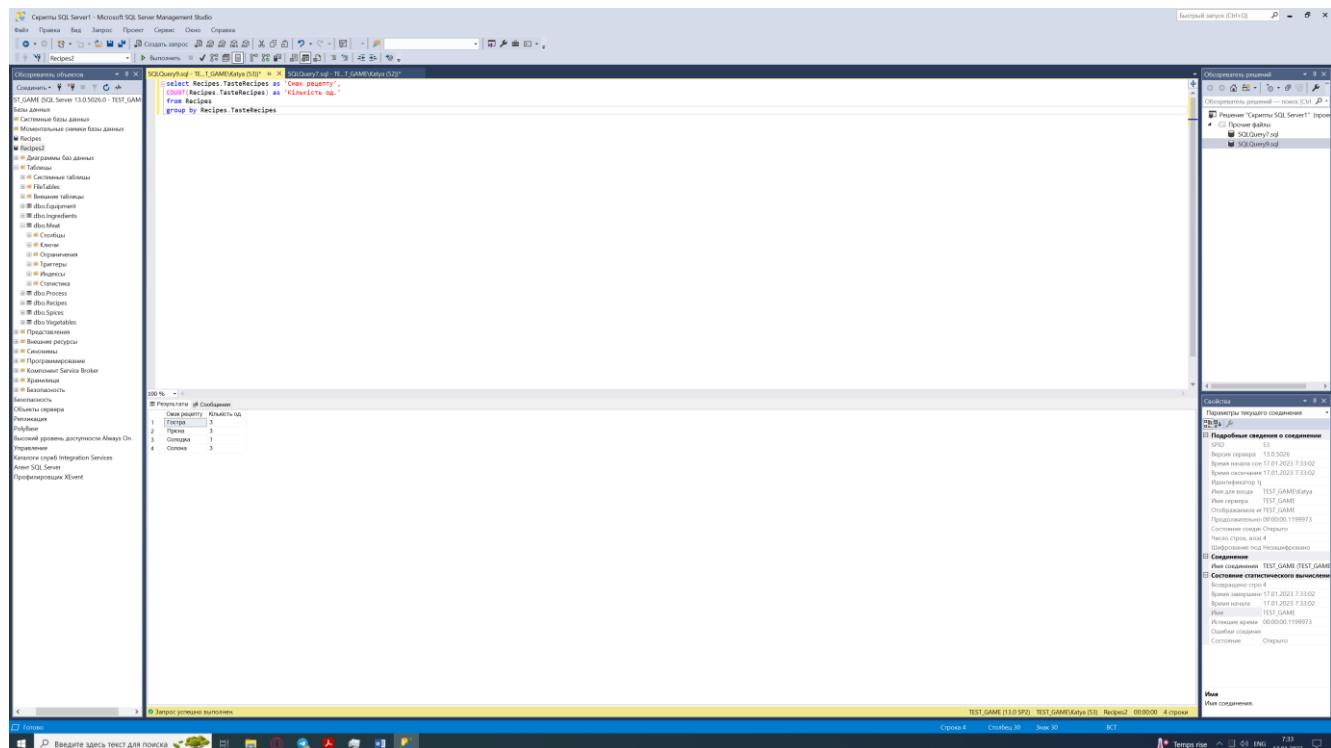


Написати запит, який би визначив, скільки різних смаків рецептів:

```

select Recipes.TasteRecipes as 'Смак рецепту',
COUNT(Recipes.TasteRecipes) as 'Кількість од.'
from Recipes
group by Recipes.TasteRecipes

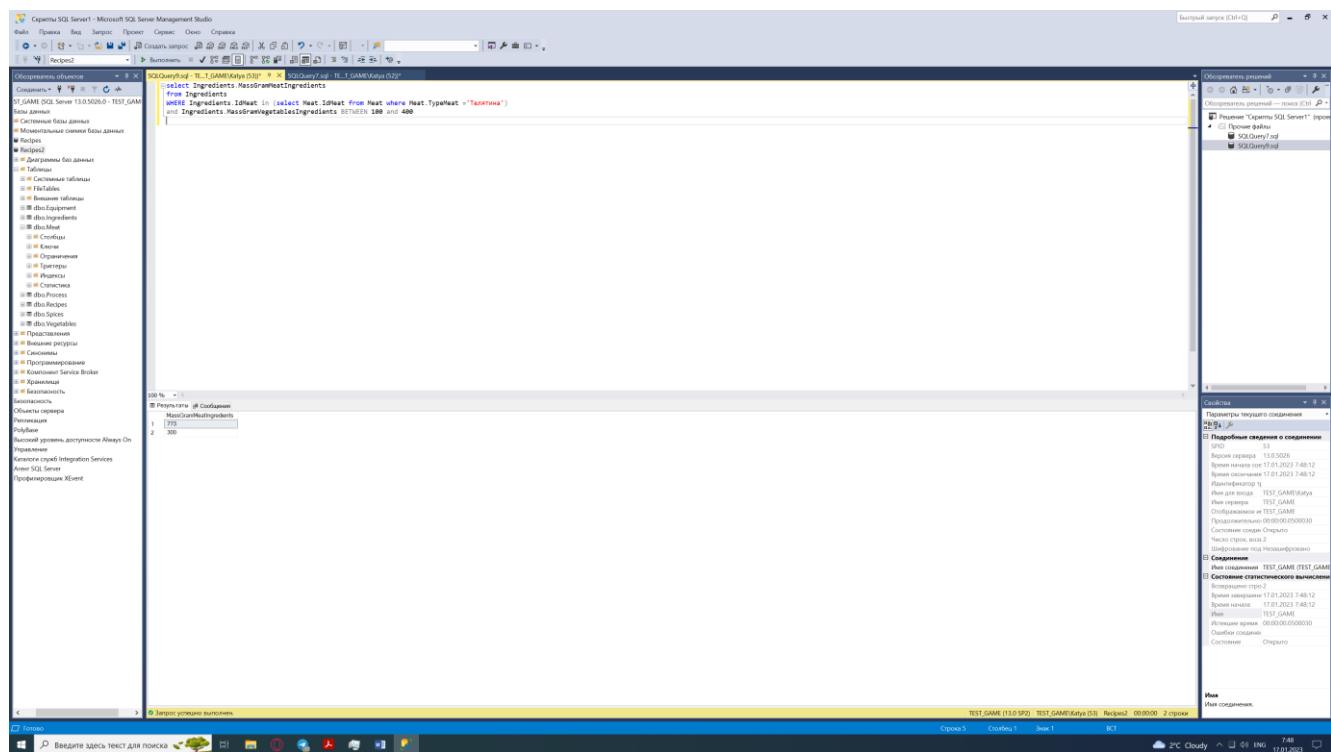
```



2.2.3 Вибірка значень з діапазону

Написати запит, який би визначав яку масу має телятина в інгредієнтах, в яких маса овочів знаходиться в межах від 100 до 400 гр:

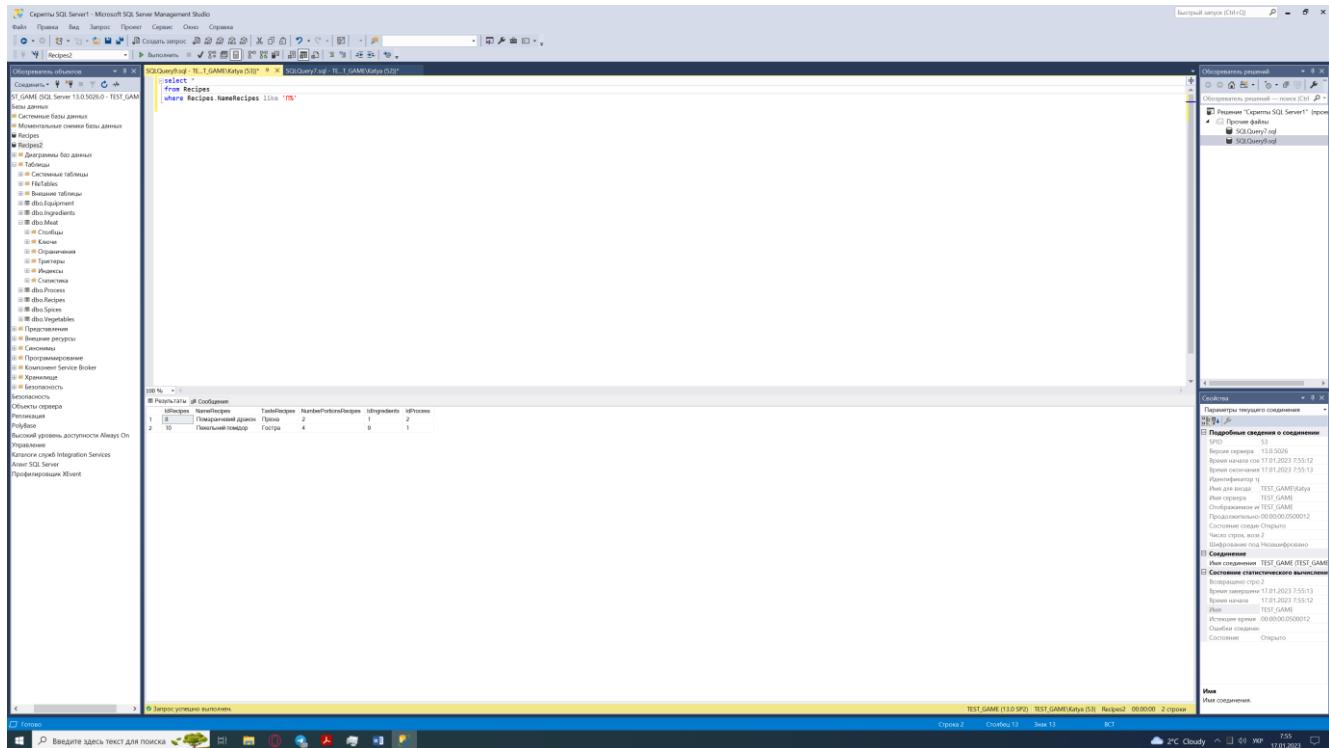
```
select Ingredients.MassGramMeatIngredients
from Ingredients
WHERE Ingredients.IdMeat in (select Meat.IdMeat from Meat
where Meat.TypeMeat = 'Телятина')
and Ingredients.MassGramVegetablesIngredients BETWEEN 100
and 400
```



2.2.4 Вибірка з використанням шаблонів

Написати запит, який би видавав всі дані про рецепти, назви яких починаються на літеру «П»:

```
select *
from Recipes
where Recipes.NameRecipes like 'П%'
```



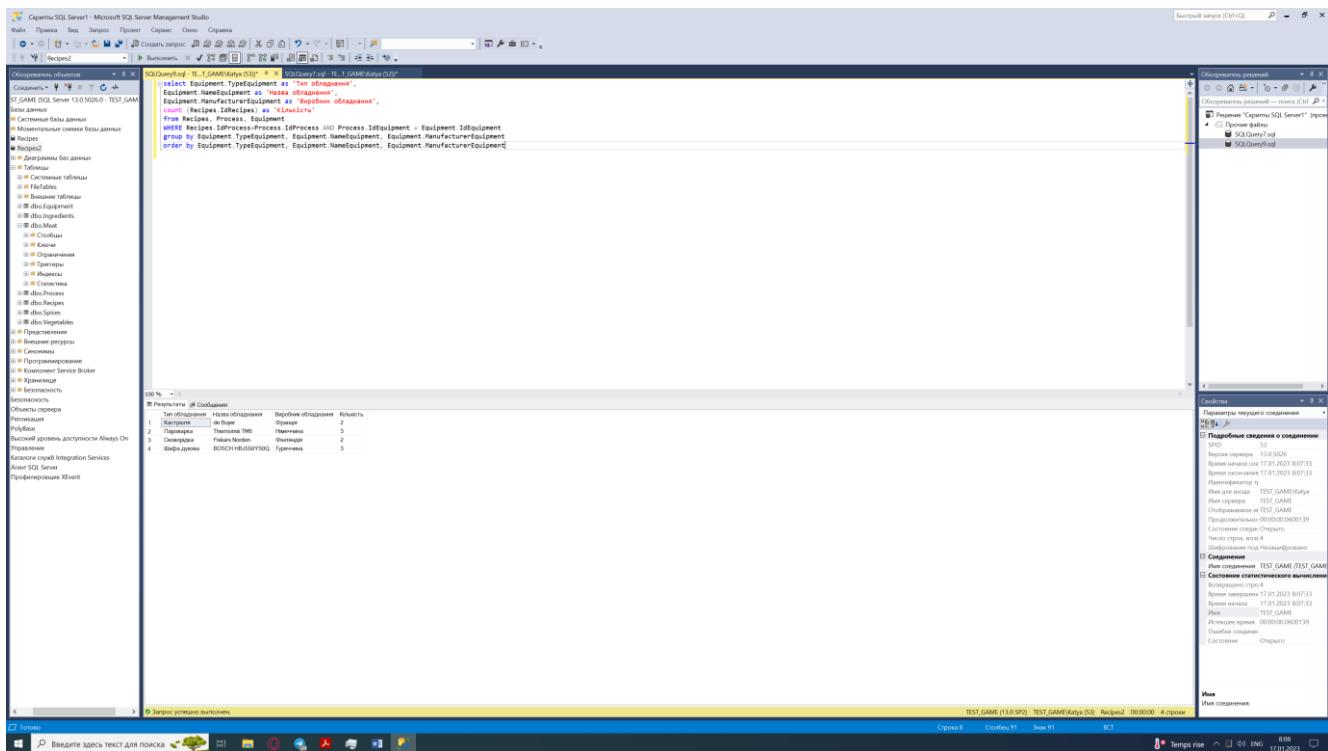
2.2.5 Угруповання даних при організації запитів

Написати запит, який би визначав у скількох рецептах застосовується кожне обладнання та угрупувати ці дані по полям «тип», «назва» та «виробник».

```

select Equipment.TypeEquipment as 'Тип обладнання',
Equipment.NameEquipment as 'Назва обладнання',
Equipment.ManufacturerEquipment as 'Виробник обладнання',
count (Recipes.IdRecipes) as 'Кількість'
from Recipes, Process, Equipment
WHERE Recipes.IdProcess=Process.IdProcess AND
Process.IdEquipment = Equipment.IdEquipment
group by Equipment.TypeEquipment, Equipment.NameEquipment,
Equipment.ManufacturerEquipment
order by Equipment.TypeEquipment, Equipment.NameEquipment,
Equipment.ManufacturerEquipment

```

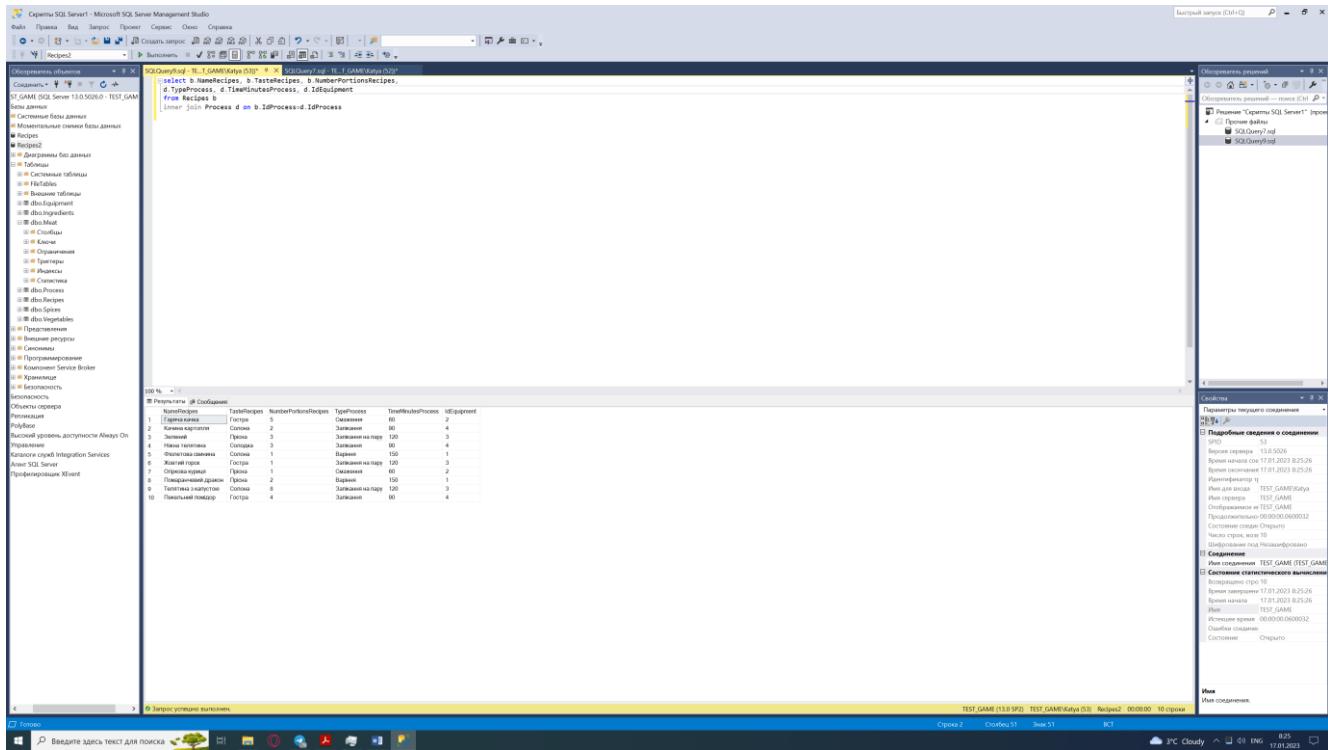


2.2.6 Об'єднання таблиць

Написати запит, який би визначив в яких рецептах використовується певний процес:

```

select b.NameRecipes, b.TasteRecipes,
b.NumberPortionsRecipes,
d.TypeProcess, d.TimeMinutesProcess, d.IdEquipment
from Recipes b
inner join Process d on b.IdProcess=d.IdProcess
  
```



ВИСНОВКИ

Метою курсової роботи було проектування бази даних рецептів. Для виконання курсової роботи були проведені всі необхідні дослідження, що стосуються розробки стратегії автоматизації. Таким чином була надана відповідь на принципові запитання, що стосуються автоматизації вибору потрібного рецепта.

Далі була проведена побудова концептуальної моделі. Для цього була використана мова ER-опису предметної області. Вона працює на концепції того, що інформаційну модель будь-якої предметної області можна описати за допомогою понять сутність, атрибути і зв'язок. Важливою особливістю мови ER-опису є її графічність, що забезпечує наочність представлення концептуальної моделі предметної області. У результаті використання теорії нормалізації було побудовано модель у третій нормальній формі.

Логічне та фізичне проектування бази даних полягало у перетворенні концептуальної моделі предметної області у реляційну модель даних.

Наступним кроком було представлення реляційної бази даних у вигляді команд створення таблиць бази даних у мові SQL. Також у даній мові описані деякі пошукові запити.

Потужність бази даних обумовлена можливістю постійно поновлюватись новими даними у необмеженій кількості інформації, що є дуже дружним для користувача.

Отже, створення бази даних «Рецепти» є достатньо актуальним і корисним.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Атре, Ш. Структурный подход к организации баз данных / Ш. Атре. - М.: Финансы и статистика, 2010. - 317 с.
2. Ахаян Эффективная работа с СУБД / Ахаян и др. - М.: СПб: Питер, 2003. - 704 с.
3. Архипенков, С. Хранилища данных. От концепции до внедрения / С. Архипенков, Д. Голубев, О. Максименко. - М.: Диалог-Мифи, 2002. - 528 с.
4. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Третье издание. Москва-Санкт-Петербург-Киев: Вильямс, 2003 - 1440 с.
5. Крёнке Д. Теория и практика построения баз данных. 8-е изд. – СПб.: Питер, 2003. - 800с.
6. Дунаев В.В. Базы данных. Язык SQL. – СПБ -.: БХВ-Петербург, 2006. – 288с.
7. Марков А.С., Лисовский К.Ю. Базы данных. Введение в теорию и методологию: Учебник. – М.: Финансы и статистика, 2006. – 512 с.
8. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений. – 4-е. изд., доп. и перераб. – СПб.: КОРОНА прнт, 2004. – 736 с.
9. Роб П., Коронел К. Системы баз данных: проектирование, реализация и управление. – 5-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2004. – 1040с.
10. Кузин А.В., Левонисова С.В. Базы данных: учеб. Пособие для студ. высш. учеб. заведений. –2-е изд., стер. –М.: Издательский центр «Академия», 2008. – 320с.
11. Кузнецов С. Д. Основы баз данных: учебное пособие – 2-е изд., испр. – М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. – 484 с.
12. Дунаев В.В. Базы данных. Язык SQL для студента Издательство: БХВ-Петербург, 2007 – 312с.

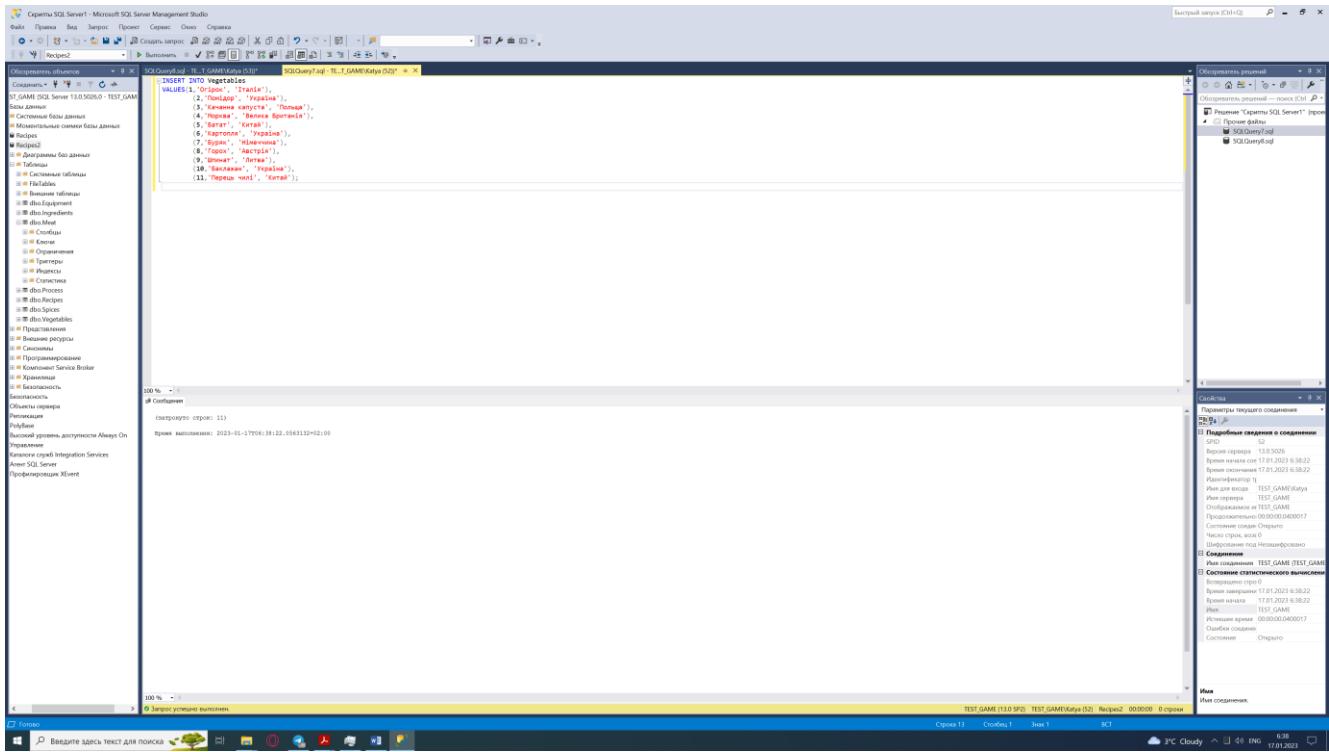
13. Андон Ф., Резниченко В. Язык запросов SQL. Учебный курс. – СПб.: Питер; Киев: Издательская группа BHV, 2006. – 416 с.
14. Белоногов, Г.Г. Автоматизация процессов накопления, поиска и обобщения информации / Г.Г. Белоногов, А.П. Новоселов. - М.: Наука, 2017. - 256 с.
15. Глушаков, С.В. Базы данных / С.В. Глушаков, Д.В. Ломотько. - М.: Харьков: Фолио, 2000. - 504 с.
16. Голицына Базы данных / Голицына, О.Л. и. - М.: Форум; Инфра-М, 2007. - 399 с.
17. Грэй, П. Логика, алгебра и базы данных / П. Грэй. - М.: Машиностроение, 2015. - 368 с.
18. Емельянов, Н.Е. Введение в СУБД ИНЕС / Н.Е. Емельянов. - М.: Наука, 2012. - 256 с.
19. Мюллер, Р.Дж. Базы данных и UML. Проектирование / Р.Дж. Мюллер. - М.: ЛОРИ, 2002. - 420 с.
20. Редько, В.Н. Базы данных и информационные системы / В.Н. Редько, И.А. Бассараб. - М.: Знание, 2004. - 240 с.
21. Ульман, Дж. Базы данных на Паскале / Дж. Ульман. - М.: Машиностроение, 2017. - 368 с.
22. Ульман, Дж. Основы систем баз данных / Дж. Ульман. - М.: Финансы и статистика, 2017. - 292 с.

ДОДАТКИ

Додаток А. Заповнення таблиць реляційної бази даних

Заповнення таблиці «Овочі (Vegetables)»:

```
INSERT INTO Vegetables
VALUES(1, 'Огірок', 'Італія'),
      (2, 'Помідор', 'Україна'),
      (3, 'Качанна капуста', 'Польща'),
      (4, 'Морква', 'Велика Британія'),
      (5, 'Батат', 'Китай'),
      (6, 'Картопля', 'Україна'),
      (7, 'Буряк', 'Німеччина'),
      (8, 'Горох', 'Австрія'),
      (9, 'Шпинат', 'Литва'),
      (10, 'Баклажан', 'Україна'),
      (11, 'Перець чилі', 'Китай');
```



Переглянемо результат:

```
select * from Vegetables
```

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window displays the following SQL code:

```
SELECT * FROM Vegetables
```

The results pane shows the data from the 'Vegetables' table:

ID	Name	Country
1	Помідор	Україна
2	Капуста кочанна	Польща
3	Морква	Італія/Іспанія
4	Баклажан	Італія
5	Картофель	Угорщина
6	Лук	Угорщина
7	Буряк	Угорщина
8	Горіх	Англія
9	Лимон	Іспанія
10	Блекінз	Угорщина
11	Перець чорний	Італія

The status bar at the bottom indicates: Строки 0 Страниц 25 Запрос 25 ВСТ 6:38 17.01.2023

Заповнення таблиці «Спеції (Spices)»:

```
INSERT INTO Spices
VALUES (1, 'Сіль', 'Україна'),
       (2, 'Перець чорний', 'Латвія'),
       (3, 'Гірчиця', 'Південна Корея'),
       (4, 'Майонез', 'Франція'),
       (5, 'Куркума', 'Індія');
```

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window displays the following SQL code:

```
INSERT INTO Spices
VALUES (1, 'Сіль', 'Україна'),
       (2, 'Перець чорний', 'Латвія'),
       (3, 'Гірчиця', 'Південна Корея'),
       (4, 'Майонез', 'Франція'),
       (5, 'Куркума', 'Індія');
```

The results pane shows the data from the 'Spices' table:

ID	Name	Country
1	Сіль	Україна
2	Перець чорний	Латвія
3	Гірчиця	Південна Корея
4	Майонез	Франція
5	Куркума	Індія

The status bar at the bottom indicates: Строки 0 Страниц 1 Запрос 0 ВСТ 6:39 17.01.2023

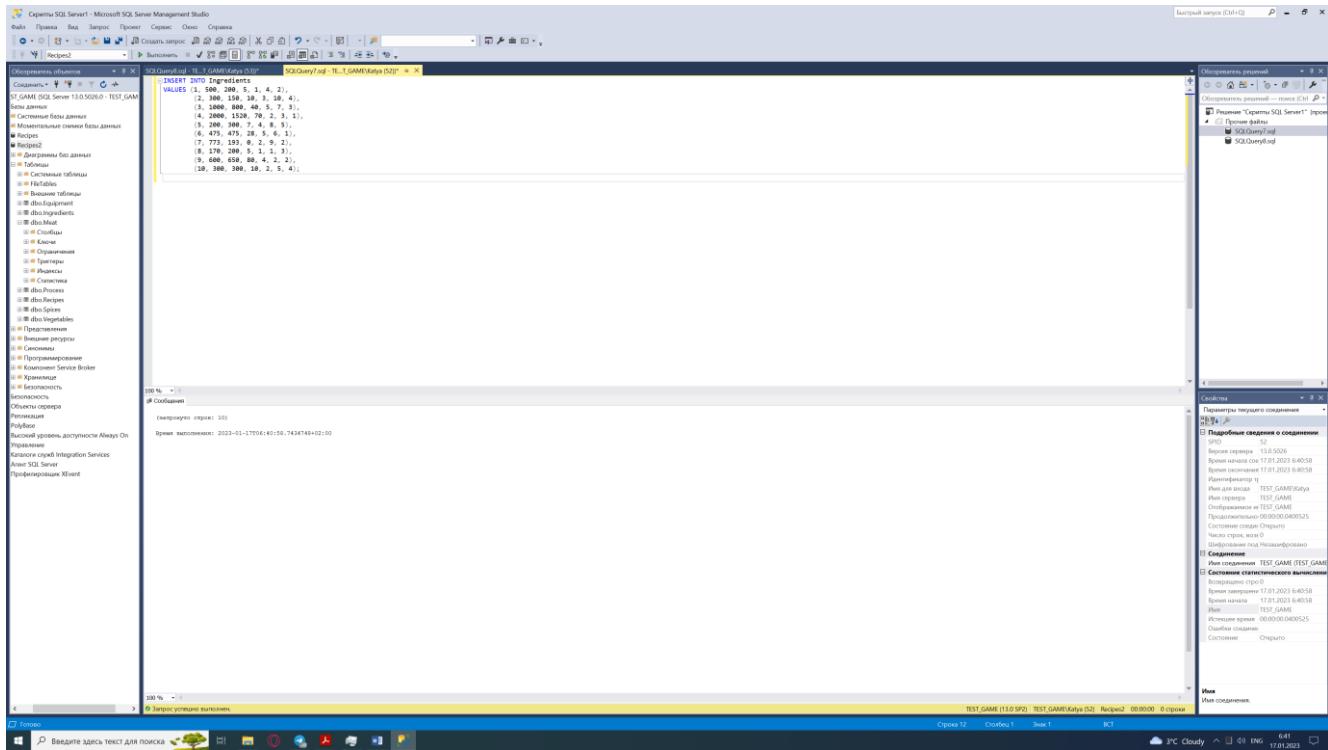
Переглянемо результат:

```
select * from Spices
```

ID	Назва	Тип	Опис	Ціна
1	Спіс	Сушена	Макабасківська	500
2	Паприка чорная	Сушена	Угорська	300
3	Лимон	Сушені фрукти	Помаранчевий	1000
4	Макабасківський корін	Сушені коріння	Французький	2000
5	Корінь	Сушені коріння	Індійський	1520

Заповнення таблиці «Інгредієнти (Ingredients)»:

```
INSERT INTO Ingredients
VALUES (1, 500, 200, 5, 1, 4, 2),
       (2, 300, 150, 10, 3, 10, 4),
       (3, 1000, 800, 40, 5, 7, 3),
       (4, 2000, 1520, 70, 2, 3, 1),
       (5, 200, 300, 7, 4, 8, 5),
       (6, 475, 475, 28, 5, 6, 1),
       (7, 773, 193, 0, 2, 9, 2),
       (8, 170, 200, 5, 1, 1, 3),
       (9, 600, 650, 80, 4, 2, 2),
       (10, 300, 300, 10, 2, 5, 4);
```



Переглянемо результат:

```
select * from Ingredients
```

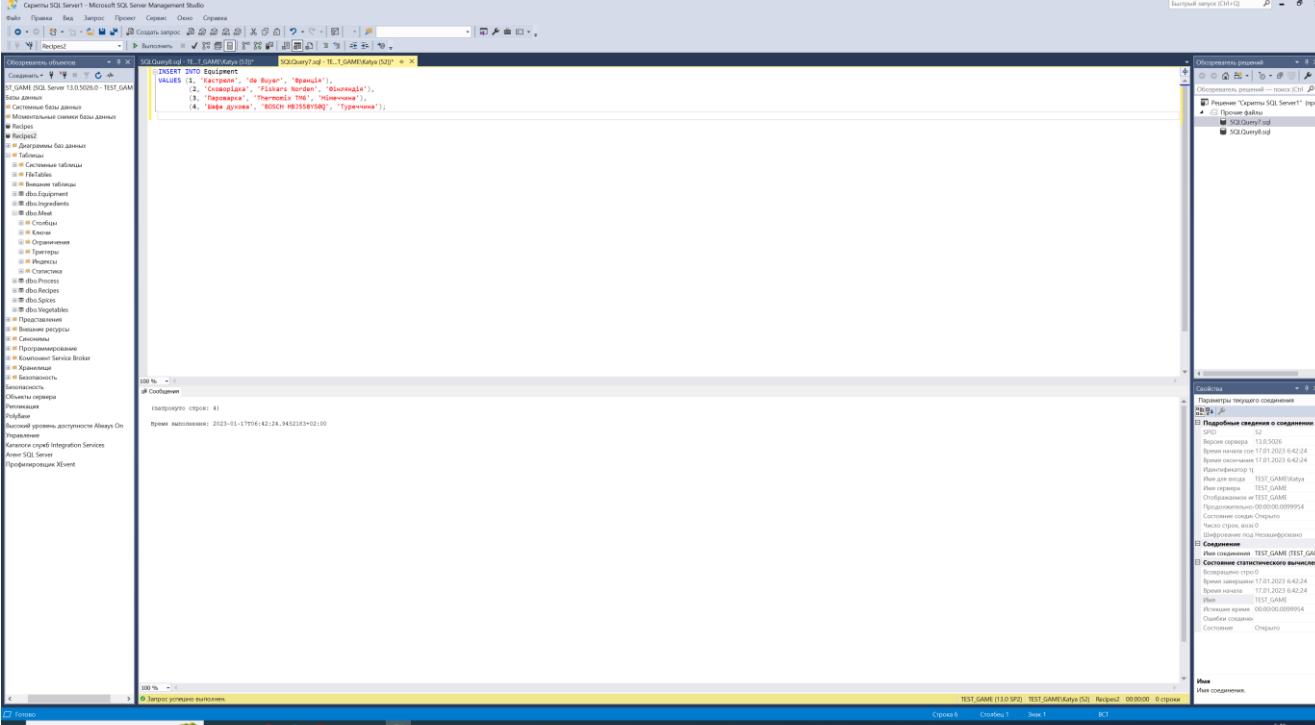
IngredientID	Name	MassGrams	UnitPrice	Unit	UnitBuckets
1	Лимон	200	5	1	4
2	Вода	1000	10	1	4
3	Сахар	500	40	1	4
4	Молочная сыворотка	1500	70	2	3
5	Масло	200	100	1	1
6	Яйца	475	475	28	5, 6, 1
7	Чеснок	150	0	2	9, 2
8	Помидоры	500	100	1	1
9	Грибы	650	650	0	1, 2
10	Соль	300	300	10	2, 5, 4

Заповнення таблиці «Обладнання (Equipment)»:

```

INSERT INTO Equipment
VALUES (1, 'Каструля', 'de Buyer', 'Франція'),
       (2, 'Сковорідка', 'Fiskars Norden', 'Фінляндія'),
  
```

- (3, 'Пароварка', 'Thermomix TM6', 'Німеччина'),
(4, 'Шафа духовка', 'BOSCH HBJ558YSQ', 'Туреччина');



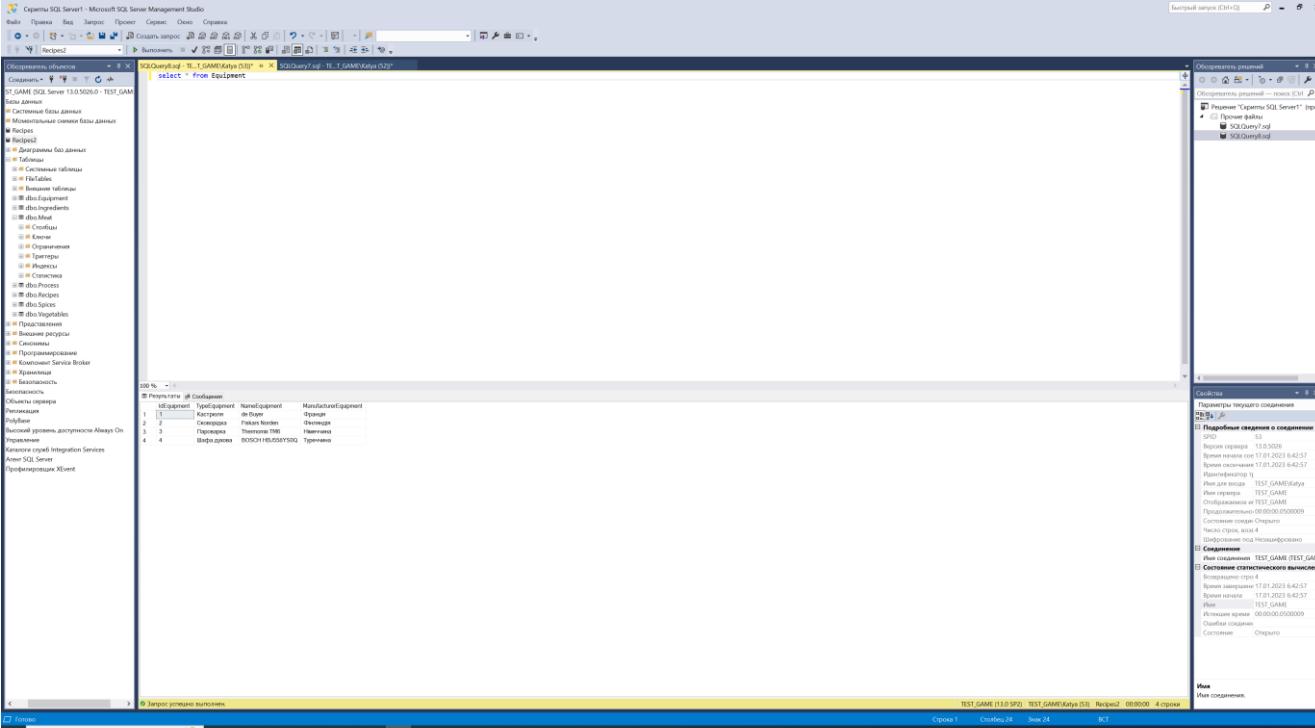
```
INSERT INTO Equipment
VALUES (1, 'Суниця', 'de Beur', 'Франція'),
       (2, 'Сонячні яблуки', 'Fikaera Norden', 'Фінляндія'),
       (3, 'Пароварка', 'Thermomix TM6', 'Німеччина'),
       (4, 'Шафа духовка', 'BOSCH HBJ558YSQ', 'Туреччина');
```

The screenshot shows the SQL Server Management Studio interface with a query window containing the above SQL code. The results pane shows the inserted data:

id	EquipmentName	Manufacturer	Country
1	Суниця	de Beur	Франція
2	Сонячні яблуки	Fikaera Norden	Фінляндія
3	Пароварка	Thermomix TM6	Німеччина
4	Шафа духовка	BOSCH HBJ558YSQ	Туреччина

Переглянемо результат:

`select * from Equipment`



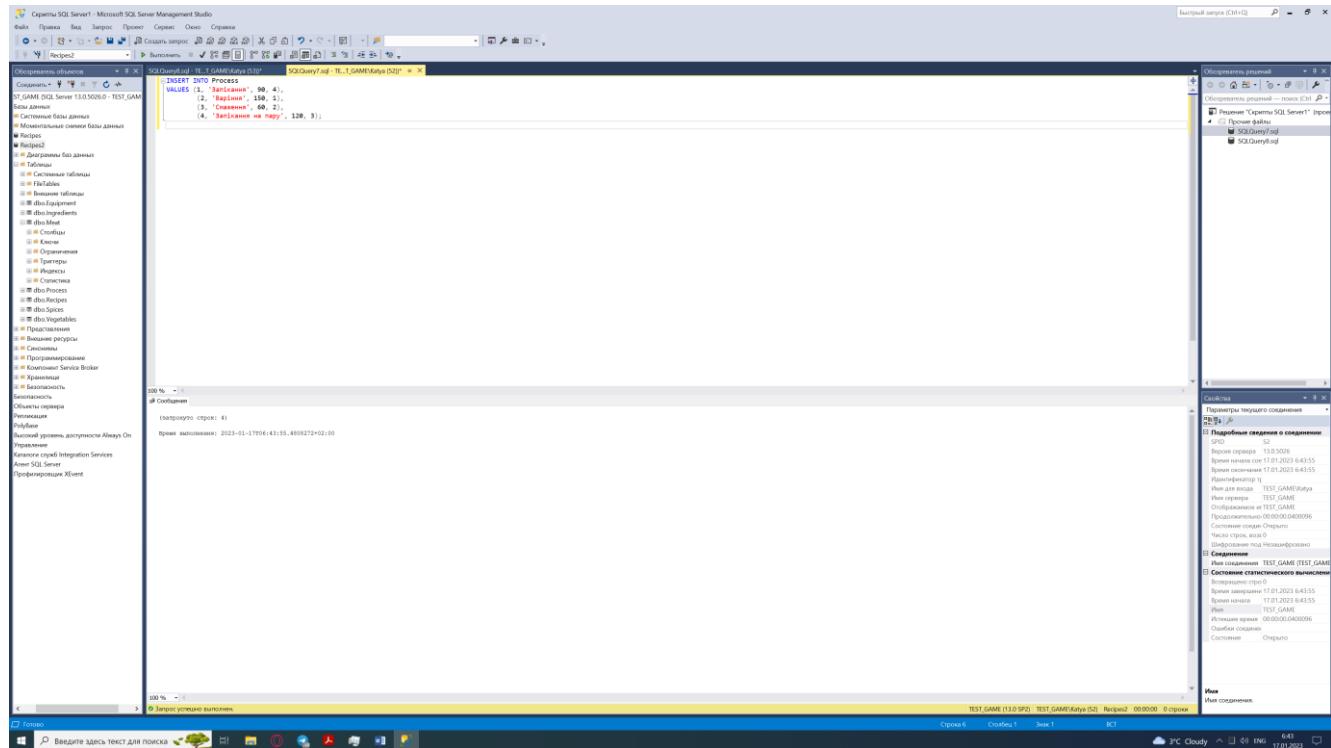
```
select * from Equipment
```

The screenshot shows the SQL Server Management Studio interface with a query window containing the above SQL code. The results pane shows the data from the Equipment table:

id	EquipmentName	TypeOfEquipment	NameOfEquipment	ManufactureEquipment
1	Суниця	de Beur	Франція	
2	Сонячні яблуки	Fikaera Norden	Фінляндія	
3	Пароварка	Thermomix TM6	Німеччина	
4	Шафа духовка	BOSCH HBJ558YSQ	Туреччина	

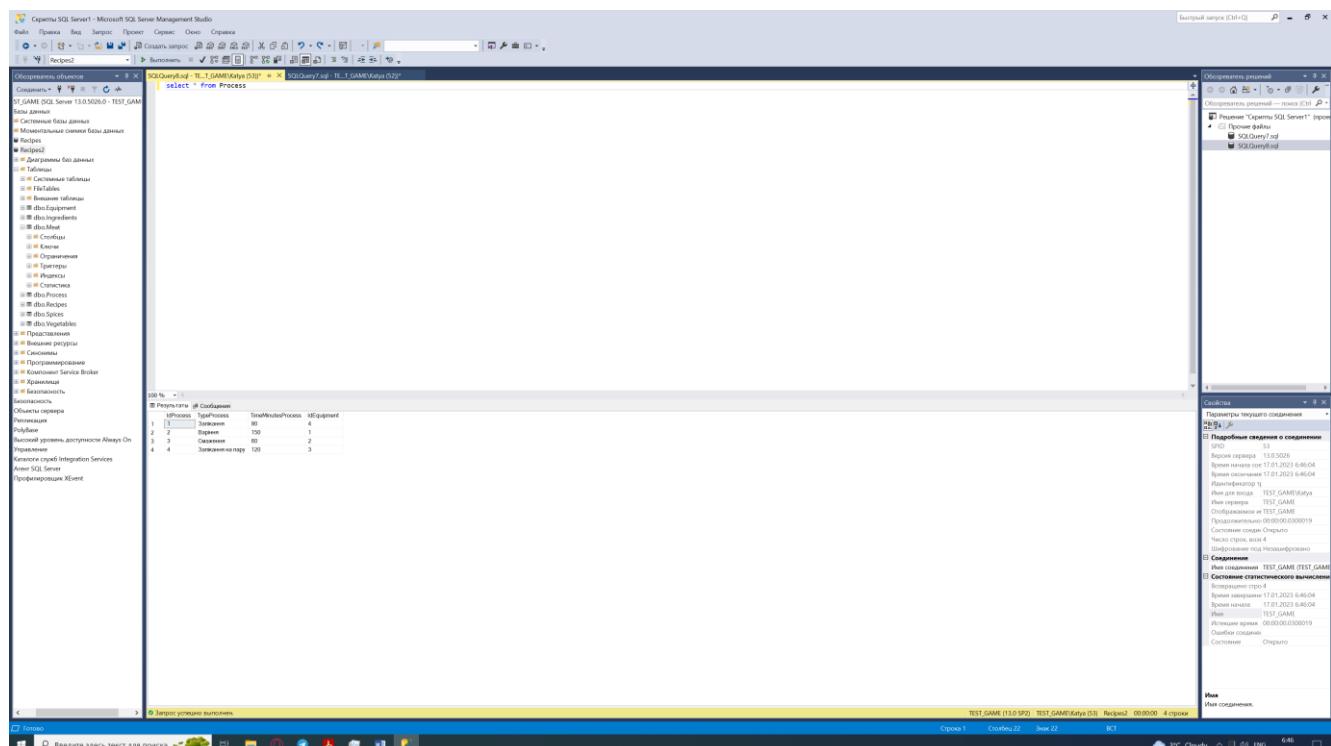
Заповнення таблиці «Процес (Process)»:

```
INSERT INTO Process
VALUES (1, 'Запікання', 90, 4),
       (2, 'Варіння', 150, 1),
       (3, 'Смаження', 60, 2),
       (4, 'Запікання на пару', 120, 3);
```



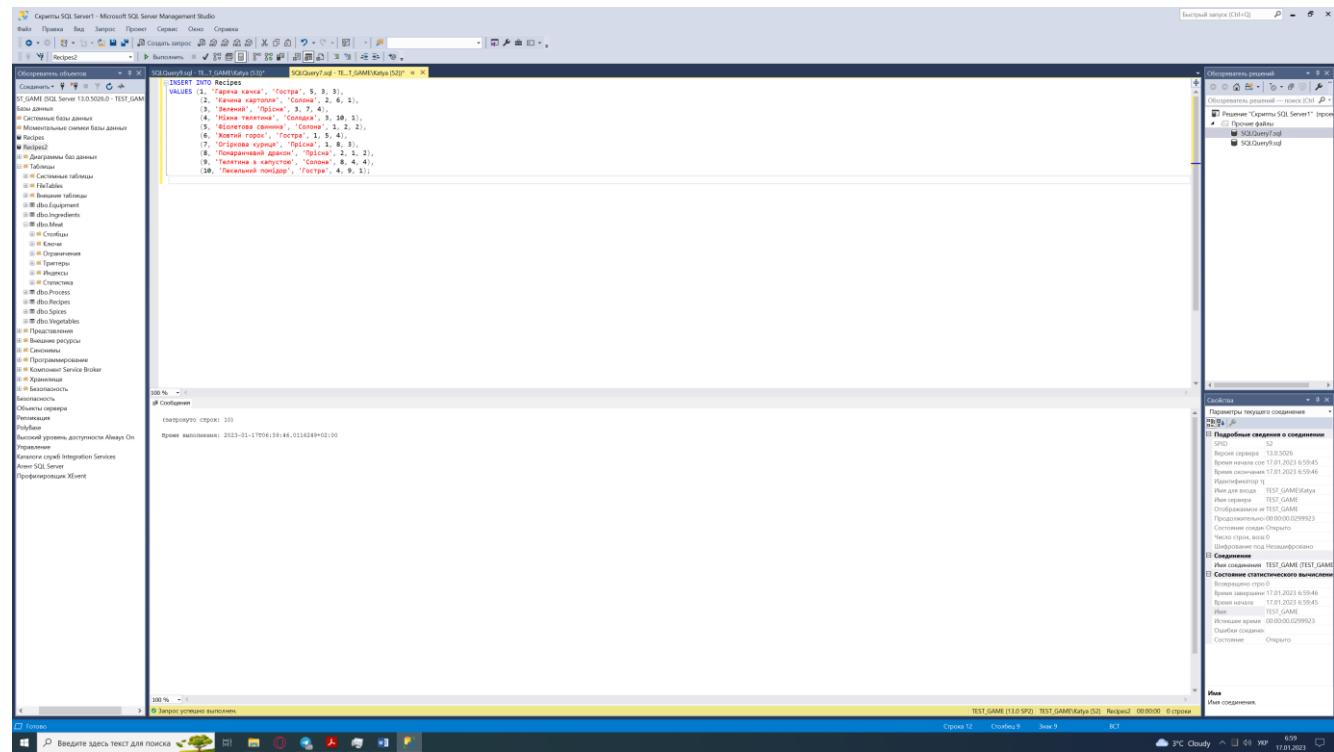
Переглянемо результат:

```
select * from Process
```



Заповнення таблиці «Рецепти (Recipes)»:

```
INSERT INTO Recipes
VALUES (1, 'Гаряча качка', 'Гостра', 5, 3, 3),
(2, 'Качина картопля', 'Солона', 2, 6, 1),
(3, 'Зелений', 'Прісна', 3, 7, 4),
(4, 'Ніжна телятина', 'Солодка', 3, 10, 1),
(5, 'Фіолетова свинина', 'Солона', 1, 2, 2),
(6, 'Жовтий горох', 'Гостра', 1, 5, 4),
(7, 'Огіркова куриця', 'Прісна', 1, 8, 3),
(8, 'Помаранчевий дракон', 'Прісна', 2, 1, 2),
(9, 'Телятина з капустою', 'Солона', 8, 4, 4),
(10, 'Пекельний помідор', 'Гостра', 4, 9, 1);
```



Переглянемо результат:

```
select * from Recipes
```

