

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

Магазин мобільних телефонів

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:

Кудінова А.О
студент II курсу
групи ІК-11
залікова книжка
№ ІК-1116



_____ (особистий підпис виконавця)

«22» грудня 2022 р.

Члени комісії:

_____ (особистий підпис)

_____ (розшифровка підпису)

_____ (особистий підпис)

_____ (розшифровка підпису)

Київ 2022
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 ІК-11 Семестр 3

ЗАВДАННЯ
на курсову роботу студента

Кудінова Анастасія Олексіївна

(прізвище, ім'я, по батькові)

1. Тема роботи

Магазин мобільних телефонів

2. Термін здачі студентом закінченої роботи «22 » грудня 2022 р.

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

загальна інформація про замовлення, інформація про замовлення, інформація про клієнта, інформація про продукти, інформація про модель, інформація про параметри, інформація про виробника.

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, створення збережених процедур, створення тригерів, створення представень)

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

Рисунок (1.1), таблиці (1.1-1.11) та рисунки(2.1- 2.18)

6. Дата видачі завдання « 13 » жовтня 2022 р.

КАЛЕНДАРНИЙ ПЛАН-ГРАФІК

виконання курсової роботи

студентом Кудінова Анастасія Олексіївна
(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 | 13.10.22 | |
| 2 | Аналіз та опис предметної області | 20.10.22 | |
| 3 | Проектування бази даних | 27.10.22 | |
| 4 | Створення бази даних | 10.11.22 | |
| 5 | Організація вибірки інформації з бази даних | 24.11.22 | |
| 6 | Оформлення пояснівальної записки | 15.12.22 | |
| 7 | Здача курсової роботи | 22.12.22 | |

Дата видачі завдання «13» жовтня 2022 р.

Завдання прийняв до виконання


(підпись)

A.O. Кудінова

Керівник

(підпись)

B.B. Завгородній

ЗМІСТ

| | |
|--|----|
| ВСТУП | 6 |
| 1 ТЕОРЕТИЧНА ЧАСТИНА | 8 |
| 1.1 Аналіз та опис предметної області | 8 |
| 1.1.1 Опис предметної області | 8 |
| 1.1.2 Опис вхідних даних | 8 |
| 1.1.3 Опис вихідних даних | 9 |
| 1.2 Проектування бази даних | 9 |
| 1.2.1 Інфологічна модель бази даних | |
| 1.2.1.1 Опис сутностей | 10 |
| 1.2.1.2. Опис атрибутів | 11 |
| 1.2.1.3 Опис зв'язків | 13 |
| 1.2.1.4 Діаграма "сущність-зв'язок" | 15 |
| 1.2.2 Нормалізація таблиць при проектуванні бази даних | 16 |
| 1.2.3 Даталогічна модель бази даних | 18 |
| 2 ПРАКТИЧНА ЧАСТИНА | 24 |
| 2.1 Створення бази даних за допомогою MS SQL Server | 24 |
| 2.1.1 Створення бази даних | 24 |
| 2.1.2 Створення таблиць бази даних | 25 |
| 2.1.3 Створення діаграми бази даних | 28 |
| 2.1.4 Заповнення таблиць бази даних даними | 30 |
| 2.1.5 Створення збережених процедур | 31 |
| 2.1.6 Створення тригерів | 33 |
| 2.1.7 Створення представлень | 34 |
| 2.2 Організація вибірки інформації з бази даних | 36 |
| 2.2.1 Проста вибірка даних | 36 |
| 2.2.2 Вибірка обчислюваних значень | 37 |
| 2.2.3 Вибірка значень з діапазону | 37 |
| 2.2.4 Вибірка з використанням шаблонів | 38 |

| | |
|--|-----------|
| 2.2.5 Угруповання даних при організації запитів | 39 |
| 2.2.6 Об'єднання таблиць | 40 |
| ВИСНОВКИ | 41 |
| СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ | 42 |
| ДОДАТКИ | 44 |
| Додаток А. Заповнення таблиць реляційної бази даних | 44 |
| Додаток Б. Створення збережених процедур | 54 |
| Додаток В. Створення тригерів | 59 |
| Додаток Г. Створення представлень | 65 |
| Додаток І. Проста вибірка даних | 68 |
| Додаток Д. Вибірка обчислюваних значень | 70 |
| Додаток Е. Вибірка значень з діапазону | 72 |
| Додаток Є. Вибірка з використанням шаблонів | 73 |
| Додаток Ж. Угруповання даних при організації запитів | 75 |
| Додаток З. Об'єднання таблиць | 78 |

ВСТУП

Створення бази даних – це великий прогрес людства, завдяки якому ми можемо зберігати, обробляти, аналізувати дані. Тому зараз вона використовується в усіх інформаційних технологіях та системах. Винайдення цієї методики значно спростило керування великими потоками інформації.

Термін «база даних» можна визначити як сукупність даних, упорядкованих для спрощення та швидкості пошуку та вилучення за допомогою комп'ютера. Як правило, інформація всередині бази даних зберігається у вигляді набору таблиць. Кожна таблиця містить певні набори даних, які можуть бути взаємопов'язані з іншими даними інших таблиць і посилатися ними.

Роль бази даних – це спростити взаємодію, організацію та аналіз пов'язаних даних з різних джерел. Дані зберігаються у взаємопов'язаних таблицях, що дозволяє маніпулювати ними.

Отже, база даних — це велика сукупність даних, підпорядкованих додатковим правилам. Такі правила залежать від виду бази даних, а за їх дотримання відповідає система управління базами даних (СУБД).

На даний момент більша увага приділяється реляційній базі даних. *Реляційна база даних* - це такий тип бази даних, що дозволяє нам сприймати інформацію у вигляді нормалізованих таблиці та зв'язків між даними таблицями. Кожна таблиця у реляційній базі даних має містити первинний ключ. *Первинний ключ* – це унікальний ідентифікатор запису в таблиці. *Зовнішній ключ* — це поле в таблиці, значення якого відповідає первинному ключу в іншій таблиці.

Система управління базою даних (СУБД) – це певне програмне забезпечення за допомогою якого можна проектувати, налаштовувати та адмініструвати базу даних. СУБД дає доступ до бази даних за допомогою системи забезпечення безпеки, підтримки цілісності бази даних, управління паралельною роботою додатків, відновлення.[5-7]

Microsoft SQL Server - система управління базами даних, яка розробляється корпорацією Microsoft та є представником реляційного типу використання. Мова,

що використовується для запитів — Transact-SQL. Вона є ключем до використання SQL Server. Всі програми, що взаємодіють з екземпляром SQL Server, незалежно від їх реалізації та інтерфейсу користувача, відправляють серверу інструкції Transact-SQL.

SQL є інструментом, призначеним для організації, управління, вибірки та обробки інформації, що міститься в базі даних. Данна мова працює тільки з базами даних певного типу, які називаються реляційними базами даних, які являють собою основний спосіб організації даних широкому діапазоні додатків. SQL в його вихідному вигляді є інформаційно-логічною мовою, а не мовою програмування, але разом SQL передбачає можливість його процедурних розширень, з урахуванням яких мова вже цілком може розглядатися як програмування.[9-10]

В даній курсовій роботі була розроблена реляційна база даних «Магазин мобільних телефонів» за допомогою Microsoft SQL Server 2019, в якій необхідно було вирішити наступні завдання:

- провести аналіз предметної області з описом вхідних та вихідних даних;
- побудувати інфологічну модель бази даних (опис сутностей, атрибути 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, збережені процедури, тригери та представлення;
- 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

Як вже було сказано вище, що бази даних – це дуже зручна інформаційна система. Вона може використовуватися в багатьох сферах життєдіяльності людини, починаючи від маленької қрамнички і до банківських систем. Предметною областю називають сферу застосування конкретної бази даних. В обраній області всі дані між собою пов'язані і мають певну залежність, в залежності від цього вони мають різний розмір, зв'язки та модель.

Сукупність атрибутів об'єкта та їх значень є даними конкретної бази даних, які мають певну логічну структуру, тобто описуються деякою інформаційною моделлю даних.

Моделлю даних називають систему взаємопов'язаних понять та правил, призначену для опису структур та властивостей даних, що використовуються (зберігаються та оброблювані) в інформаційній системі.

При розробці реляційної бази даних «Магазин мобільних телефонів» було проведено дослідження предметної області. Основними операціями в досліджуваній предметній області є отримання детальної інформації про замовлення для ефективного керування ними. [4-6]

1.1.2 Опис вхідних даних

Під вхідними даними розуміється інформація, яка є основною і використовується для подальшої обробки та аналізу, тобто та інформація яку додає адміністратор(користувач). За допомогою них ми можемо здійснювати різноманітні операції

При розробці реляційної бази даних « Магазин мобільних телефонів » були виділені наступні вхідні дані:

- загальна інформація про замовлення,
- інформація про замовлення,
- інформація про клієнта,
- інформація про продукти,
- інформація про модель,
- інформація про параметри,
- інформація про виробника.

1.1.3 Опис вихідних даних

Вихідні дані - повідомлення і результати, що видаються системою. Вони, як правило, беруться з постійних даних, але їх не можна розглядати як частину бази даних. Вихідні дані залежно від попередніх умов подаються у вигляді набору даних.

Вихідними даними для бази даних «Магазин мобільних телефонів» є вихідні запити.

1.2 Проектування бази даних

Проектування інформаційних систем і баз даних, що лежать в їх основі є досить складним завданням. Опис усіх використовуваних у системі або взаємодіючих з нею типів об'єктів реального світу може налічувати сотні і тисячі одиниць, співвідношення та взаємозв'язки між ними часто виявляються нетривіальними і вимагають глибоких знань процесів, що відбуваються в реальному світі.

Інфологічний (концептуальний) рівень ставить свою метою створення концептуальної моделі, що відображає основні сутності предметної області, їх атрибути та зв'язки між сутностями. Як предметну область цілком, так і безпосередньо дані на цьому рівні можна описувати різними моделями (наприклад, семантичної, графової, «сущність-зв'язок»). Як спосіб представлення

моделі найчастіше використовується UML або простий словесний опис (що особливо зручно для обговорення моделі з представником замовника, що не є ІТ-фахівцем). На цьому рівні ми оперуємо сутностями, атрибутами та деякими зв'язками.

Завдяки цій моделі ми зможемо зробити певний аналіз та виявити некоректні зв'язки. Також вона легка у розуміння для людей, які не є спеціалістами у цій справі.

Модель "сущність-зв'язок" використовують при концептуальному моделюванні для отримання концептуальної моделі, яку потім транслюють у логічні моделі, зазвичай реляційні або об'єктно-орієнтовані. Модель "сущність-зв'язок" запропонував П. Чен із метою впорядкування задачі проектування моделей, її проект був опублікований у 1976 р.. На сьогодні не існує єдиного загальноприйнятого стандарту для моделі "сущність-зв'язок", але є набір загальних конструкцій, що лежать в основі більшості її варіантів. Така ситуація виникла через те, що різні автори пропонують свої елементи моделі й відповідну термінологію. [11]

Дану роботу корисно опрацювати для того, щоб ознайомитися з типами невідповідностей між проектом (певною предметною областю) і його концептуальною моделлю, наприклад:

- пропущені елементи проекту в моделі;
- у моделі присутні елементи, що не мають аналогів у проекті;
- використання менш підходящих понять моделі "сущність-зв'язок" для відображення елементів проекту (часто ця невідповідність має суб'єктивний характер, що зазначалося вище).

1.2.1.1 Опис сущностей

Одним із основних понять моделі є поняття сущності. За визначенням, сущність є описом деякого об'єкта реального світу, який може бути чітко відокремлений від інших об'єктів (можливо, також представлених сущностями в

моделі), і його опис однозначно пов'язаний із цим реальним об'єктом. (Наприклад: Книга, постачальник)

Таке, на вигляд розмите та нечітке, визначення насправді задає важливу властивість моделі: сутності повинні бути різними. Іншими словами, від самого початку в моделі постулюється наявність деякого способу ідентифікації, що дозволяє зіставити об'єкти реального світу з їх уявленнями в базі даних. Тобто передбачається ідентифікація сутностей за їх природними ознаками.[6]

У відповідності з описом предметної області були отримано такі сутності:

- «Інформація про замовлення» - зберігається загальна інформація про замовлення,
- «Замовлення» - зберігається інформація про склад замовлення,
- «Клієнт» - зберігається інформація про клієнта,
- «Продукти» - зберігається інформація про продукти,
- «Модель» - зберігається інформація про модель,
- «Параметри» - зберігається інформація про параметри,
- «Виробник» - зберігається інформація про виробника

1.2.1.2 Опис атрибутів

Відношення - це безліч сутностей, що мають одинаковий набір атрибутів. У контексті реляційних баз даних відношення складається із заголовка (схеми) і тіла (набору кортежів).

Атрибут - іменованна властивість сутності (відносини). В побутовому рівні під атрибутом можна розуміти «стовпець таблиці».

Сутність - це абстракція (інформаційна модель) реального об'єкта предметної області, а атрибут сутності - абстракція однієї з властивостей (характеристик) об'єкта, що моделюється. Розрізняють первинні ключі, що мають властивість унікальності в справах сутності і однозначно ідентифікують кожен її екземпляр, і вторинні ключі – ідентифікатори груп екземплярів сутності.

У кожній сутності має бути визначено як мінімум один первинний ключ, що гарантує відсутність дублікатів серед її екземплярів. Якщо кілька атрибутів сутності об'єктивно мають властивість унікальності, один з них оголошується первинним ключем, а інші набувають статусу «можливих первинних ключів», не втрачаючи при цьому властивості своєї унікальності.

Вторинні ключі забезпечують можливість параметричного пошуку екземплярів сутності, які відповідають заданим значенням її атрибутів. Склад вторинних ключів сутностей визначається користувачами запитами до бази даних, що вимагають уgrуповання екземплярів сутностей певним критеріям, які включають відповідні атрибути.

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

1. Таблиця *Інформація про замовлення* (*Info_order*) містить:

Code_info - унікальний код інформація про замовлення

Code_product - унікальний код продукту

Number – кількість одиниць

Code_order - унікальний код замовлення

Price_order - ціна замовлення

2. Таблиця *Замовлення* (*Order*) містить:

Code_order - унікальний код замовлення

Code_client - унікальний код клієнта

Data – дата продажу

3. Таблиця *Клієнт* (*Client*) містить:

Code_client - унікальний код клієнта

Full_name - ПІБ

Phone – номер телефону

4. Таблиця *Продукти (Product)* містить:

Code_product - унікальний код продукту

Code_model - унікальний код моделі

Price_product – ціна продукту

All_in_stock – кількість в наявності

Name_product – назва продукту

5. Таблиця *Модель (Model)* містить:

Code_model - унікальний код моделі

Name_model – називу моделі

Code_parameter - унікальний код параметру

Code_producer - унікальний код виробника

6. Таблиця *Параметри (Parameters)* містить:

Code_parameter - унікальний код параметру

Memory – пам'ять

Camera – камера

Battery – акамулятор

Operation_system – операційна система

Processor – процесор

7. Таблиця *Виробник (Producer)* містить:

Code_producer - унікальний код виробника

Name_producer – назва виробника

Contacts – номер виробника

1.2.1.3 Опис зв'язків

Зв'язок - асоціація, що об'єднує кілька сутностей.

Зв'язок один до багатьох (one to many correspondence) - асоціація, що об'єднує два відносини таким чином, що одному кортежу батьківського відносини (або навіть нулю таких кортежів) може відповідати довільна кількість кортежів дочірніх відносин.

Зв'язок багато до одного (many to one correspondence) - асоціація, що об'єднує два відносини таким чином, довільна кількість кортежів дочірнього відношення може відповідати одному кортежу батьківського відношення (або навіть нулю таких кортежів).

Спрощено: одного запису в таблиці А може відповідати багато записів у таблиці В; при цьому може бути ситуація, коли деяким записам у таблиці В немає відповідності у таблиці А

Зв'язок багато до багатьох (many to many correspondence) — асоціація, що об'єднує два відносини таким чином, що одному кортежу будь-якого з об'єднаних відносин може відповідати довільна кількість кортежів другого відношення.

Зв'язок один до одного (one to one correspondence) - асоціація, що об'єднує два відносини таким чином, що одному кортежу батьківського відношення може відповідати не більше одного кортежу дочірнього відношення.[8]

Таблиця 1.1 – Типи зв'язків

| Номер зв'язку | Головна таблиця | Дочірня таблиця | Тип зв'язку |
|---------------|---|-----------------------|-------------|
| 1 | Info_order (Інформація про замовлення) | Order (Замовлення) | Б:1 |
| 2 | Info_order (Інформація про замовлення) | Product (Продукти) | 1:Б |
| 3 | Order (Замовлення) | Client (Клієнт) | Б:1 |

| | | | |
|---|-----------------------|---------------------------|-----|
| 4 | Product (Продукти) | Model (Модель) | Б:1 |
| 5 | Model (Модель) | Producer (Виробник) | 1:1 |
| 6 | Model (Модель) | Parameters (Параметри) | 1:Б |

Перший зв'язок «Info_order (Інформація про замовлення) - Order (Замовлення)» вказує на те, що одне замовлення може містити декілька інформацій про замовлення.

Другий зв'язок «Info_order (Інформація про замовлення) - Product (Продукти)» вказує на те, що одна інформація може містити в собі інформацію про декілька продуктів.

Третій зв'язок «Order (Замовлення) - Client (Клієнт)» вказує на те, що один клієнт може зробити декілька замовлення.

Четвертий зв'язок «Product (Продукти) - Model (Модель)» вказує на те, що декілька продуктів можуть мати одну модель.

П'ятий зв'язок Model (Модель) - Producer (Виробник) вказує на те, що одна модель може мати одного виробника.

Шостий зв'язок «Model (Модель) - Parameters (Параметри)» вказує на те, що, одна модель може мати декілька параметрів.

1.2.1.4 Діаграма "сущність-зв'язок"

На рисунку 1.1 представлена ER-діаграма бази даних «Магазин мобільних телефонів», на якій відображені всі сутності, їх атрибути та зв'язки між сутностями.

Позначення:

Сущності

Зв'язки

Атрибути

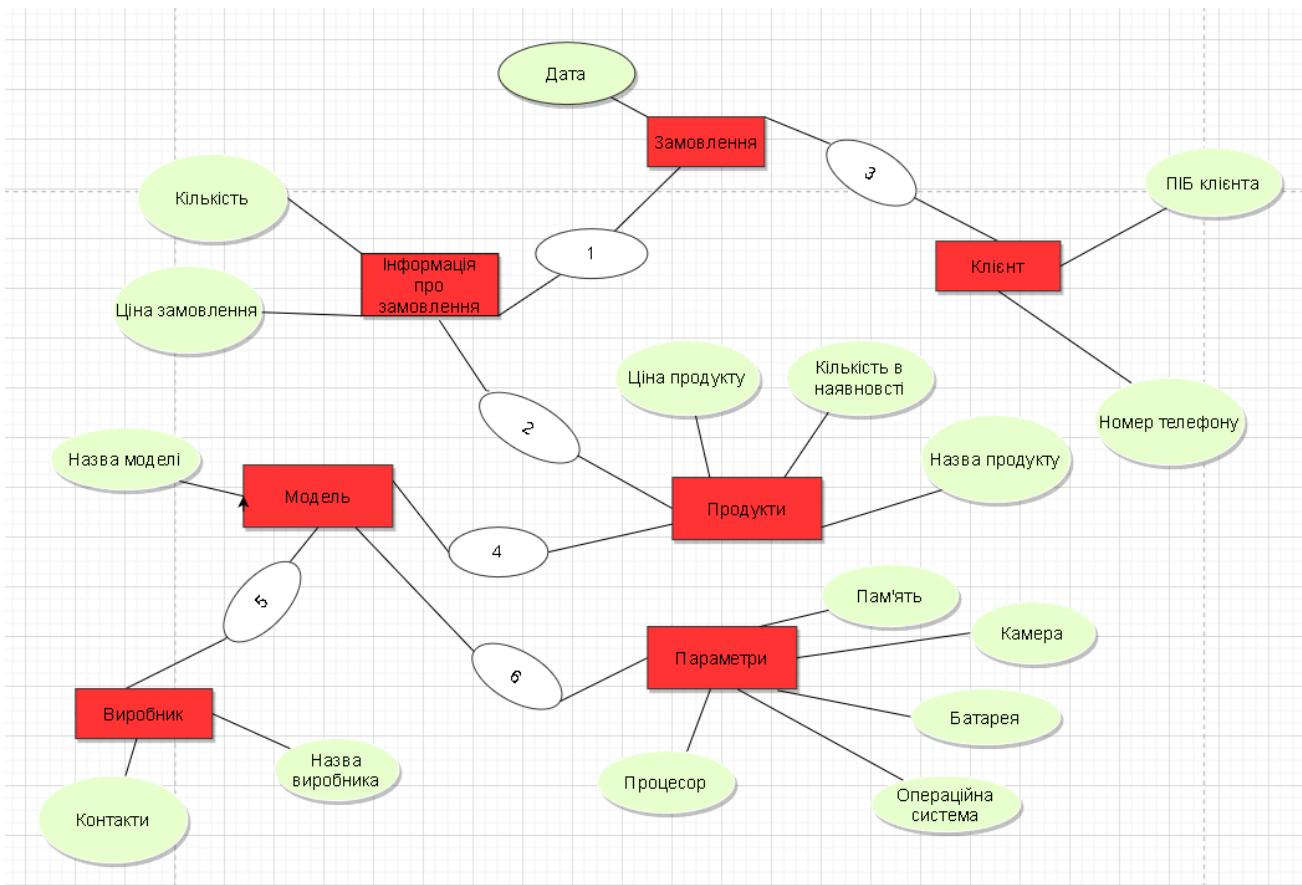


Рисунок 1.1 – ER-діаграма бази даних «Магазин мобільних телефонів»

1.2.2 Нормалізація таблиць при проектуванні бази даних

Нормалізацією реляційної бази даних називається таке інформаційно-еквівалентне перетворення її схеми, в результаті якої відносини, що знаходяться в слабких НФ, декомпозуються на кілька взаємопов'язаних відносин, кожне з яких перебуває в сильнішій НФ, що гарантує відсутність прояву аномалій

Процес нормалізації є невід'ємною частиною проектування баз даних, а тому дозволяє виявити та виправити багато недоліків проектованої схеми — як безпосередньо пов'язані з порушенням нормальних форм, так і багато інших, але не менш небезпечних.

Відношення знаходитьться в 1-й НФ, якщо воно задовольняє базовим обмеженням реляційної моделі даних: всі атрибути відносини непереривні та що відсутні кортежі-дублікати.

Відношення знаходиться у 2-й НФ, якщо воно знаходиться в 1-й НФ і існують повні ФЗ кожного неключового атрибута від усіх можливих ключів цього відношення.

Відношення знаходиться у 3-й НФ, якщо воно знаходиться у 2-й НФ і при цьому щодо відсутні транзитивні ФЗ неключових атрибутів від будь-якого з можливих ключів.

4-та нормальна форма базується на концепції багатозначних залежностей між атрибутами відношення. Тобто, такі відношення знаходяться в 4-й НФ, якщо при існуванні багатозначної залежності існують функціональні залежності решти всіх атрибутів цього відношення від атрибуту А. [10,7]

Дана модель не потребує подальшого приведення до четвертої формами нормалізації. Проведемо нормалізацію таблиць:

1. Поля «ПІБ - Full_name», «Операційна система – Operation_system», «Процесор - Processor» містить інформацію, що ділиться:
 - Поле «ПІБ - Full_name» розб'ємо на «Прізвище – First_Name», «Ім'я – Second_Name», «По батькові - Patronymic»;
 - Поле «Процесор - Processor» розб'ємо на «Версія Процесора - Ver_processor», «Номер процесора - Num_processor»;
 - Поле «Операційна система – Operation_system» розб'ємо на «Версія Операційної системи - Version_operation_system», «Номер операційної системи - Number_Version_operation_system».
2. Для кожної з таблиці необхідно обрати первинний ключ:
 - для таблиці Info_order (*Інформація про замовлення*) поле «унікальний код інформації про замовлення - Code_info»;
 - для таблиці Order (*Замовлення*) поле «унікальний код замовлення – Code_order»;
 - для таблиці Client (*Клієнт*) поле «унікальний код клієнта – Code_client»;

- для таблиці Full_Name (ПІБ) поле «унікальний код ПІБ - id_Full_Name»;
- для таблиці Product (*Продукти*) поле «унікальний код продукту – Code_product»;
- для таблиці Model (*Модель*) поле «унікальний код моделі – Code_model»;
- для таблиці Producer (Виробник) поле «унікальний код виробника – Code_producer»;
- для таблиці Parameters (*Параметри*) поле «унікальний код параметрів – Code_parameter»;
- для таблиці Processor (*Процесор*) поле «унікальний код процесору - id_Processor»;
- для таблиці Operation_system (*Операційна система*) поле «унікальний код - Id_operation_system»;

1.2.3 Даталогічна модель бази даних

Даталогічна модель близька до інфологічної і насамперед відповідає вимогі адекватності предметної області. Фізична модель (не порушуючи вимог адекватності) повинна в тому числі забезпечувати зручність використання, продуктивність, захищеність даних. Даталогічна модель даних встановлює структуру елементів даних і зв'язки між ними. Це не залежить від фізичної бази даних, яка деталізує, як дані будуть реалізовані. Вона служить планом для використаних даних, робить елементи концептуального моделювання даних на крок далі, додаючи до них більше інформації.

З огляду на те, що дані втілюють найважливіший аспект будь-якої програми, програми чи системи, якісні системи обробки та зберігання даних мають бути побудовані на міцній і точній базовій структурі даних. Надійна структура даних дає розробникам додатків свободу створювати найкращий

інтерфейс користувача, систему обробки або налаштування статистичного аналізу та звітності.

Даталогічна модель даних слугує для визначення того, як система повинна бути реалізована незалежно від використованої системи керування базою даних. Розробниками логічної моделі даних зазвичай є архіектори даних і бізнес-аналітики. Метою створення такої моделі даних є розробка високотехнічної карти основних правил і структур даних.

Переваги даталогічної моделі даних:

- Оскільки дані залишаються стабільними з часом, логічна модель даних також є стабільною та дуже сприятливою для повторного використання даних і фізичного обміну даними, що в кінцевому підсумку призводить до зменшення обсягу зберігання надлишкових даних.
- Компоненти логічної моделі даних можна переробляти, повторно використовувати та адаптувати, коли все більше команд зважують свої (часто змінювані) потреби.
- Витрати, пов’язані зі створенням і підтримкою логічної моделі даних, у довгостроковій перспективі компенсиуються перевагами, які вона надає, не в останню чергу шляхом визначення та інтеграції всіх бізнес-потреб і правил на початку.[2-4]

Для кожного поля таблиці реляційної бази даних «Магазин мобільних телефонів» (табл. 1.2 – 1.11) 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.

Таблиця 1.2 – Таблиця Інформація про замовлення (*Info_order*)

| Найменування атрибутів | Тип поля | Розмір поля | Обмеження |
|--|-----------------|--------------------|------------------|
| Унікальний код інформації про замовлення | Code_info | Int | 20 Not null |
| Унікальний код продукту | Code_product | Int | 1000 Not null |
| Кількість одиниць | Number | Int | 50 |
| Унікальний код замовлення | Code_order | Int | 1100 Not null |
| Ціна замовлення | Price_order | Int | |

Таблиця 1.3 – Таблиця Замовлення (*Order*)

| Найменування атрибутів | Тип поля | Розмір поля | Обмеження |
|-------------------------------|-----------------|--------------------|------------------|
| Унікальний код замовлення | Code_order | Int | 1100 Not null |
| Унікальний код клієнта | Code_client | Int | 100 Not null |
| Дата | Data | Data | |

Таблиця 1.4 – Таблиця Клієнт (*Client*)

| Найменування атрибутів | Тип поля | Розмір поля | Обмеження |
|-------------------------------|-----------------|--------------------|------------------|
| Унікальний код клієнта | Code_client | Int | 100 Not null |
| Унікальний код ПІБ | id_Full_name | Int | 25 Not null |
| Номер телефону | Phone | Int | 1000 |

Таблиця 1.5 – Таблиця ПІБ (*Full_Name*)

| Найменування атрибутів | | Тип поля | Розмір поля | Обмеження |
|------------------------|--------------|----------|-------------|-----------|
| Унікальний код ПІБ | id_Full_name | Int | 25 | Not null |
| Прізвище | First_Name | Char | 25 | |
| Ім'я | Second_Name | Char | 25 | |
| Побатькові | Patronymic | Char | 25 | |

Таблиця 1.6 – Таблиця Продукт (*Product*)

| Найменування атрибутів | | Тип поля | Розмір поля | Обмеження |
|-------------------------|---------------|----------|-------------|-----------|
| Унікальний код продукту | Code_product | Int | 1000 | Not null |
| Унікальний код моделі | Code_model | Int | 10120 | Not null |
| Ціна продукту | Price_product | Int | 40000 | |
| Кількість в наявності | All_in_stock | Int | 6 0 | |
| Назва продукту | Name_product | Char | 20 | |

Таблиця 1.7 – Таблиця Модель (*Model*)

| Найменування атрибутів | | Тип поля | Розмір поля | Обмеження |
|--------------------------|----------------|----------|-------------|-----------|
| Унікальний код моделі | Code_model | int | 10120 | Not null |
| Назва моделі | Name_model | char | 25 | |
| Унікальний код параметру | Code_parameter | int | 1300 | Not null |
| Унікальний код виробника | Code_producer | int | 10200 | Not null |

Таблиця 1.8 – Таблиця Виробник (*Producer*)

| Найменування атрибутів | | Тип поля | Розмір поля | Обмеження |
|--------------------------|---------------|----------|-------------|-----------|
| Унікальний код виробника | Code_producer | Int | 10200 | Not null |
| Назва продукту | Name_producer | Char | 35 | |
| Контакти | Contact | Char | 35 | |

Таблиця 1.9 – Таблиця Параметри (*Parameters*)

| Найменування атрибутів | | Тип поля | Розмір поля | Обмеження |
|------------------------------------|---------------------|----------|-------------|-----------|
| Унікальний код параметрів | Code_parameter | Int | | Not null |
| Пам'ять | Memory_Gb | Int | | |
| Камера | Camera_Mp | Int | | |
| Акамулятор | Battery_Ma | Int | | |
| Унікальний код операційної системи | Id_Operation_system | Int | | Not null |
| Унікальний код процесору | Id_Processor | Int | | Not null |

Таблиця 1.10 – Таблиця Операційна система (*Operation_system*)

| Найменування атрибутів | | Тип поля | Розмір поля | Обмеження |
|------------------------------------|---------------------------------|----------|-------------|-----------|
| Унікальний код операційної системи | Id_Operation_system | Int | 40 | Not null |
| Версія операційної системи | Version_operation_system | Char | 20 | |
| Номер операційної системи | Number_Version_operation_system | Int | 20 | |

Таблиця 1.11 – Таблиця Процесор (*Processor*)

| Найменування атрибутів | Тип поля | Розмір поля | Обмеження |
|-------------------------------|-----------------|--------------------|------------------|
| Унікальний код процесору | Id_Processor | Int | 50 Not null |
| Версія процесору | Ver_processor | Char | 20 |
| Номер процесору | Num_processor | Int | 1000 |

2 ПРАКТИЧНА ЧАСТИНА

2.1 Створення бази даних за допомогою MS SQL Server

2.1.1 Створення бази даних

Запускаємо MS SQL Server 2018 та підключаємося до серверу. Результат зображене на рисунок 2.1 та рисунку2.2.

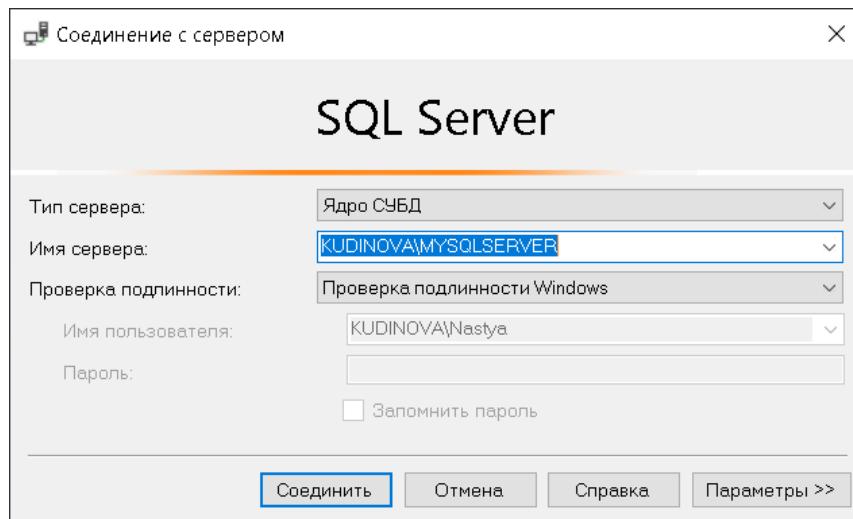


Рисунок 2.1 – Підключення до серверу

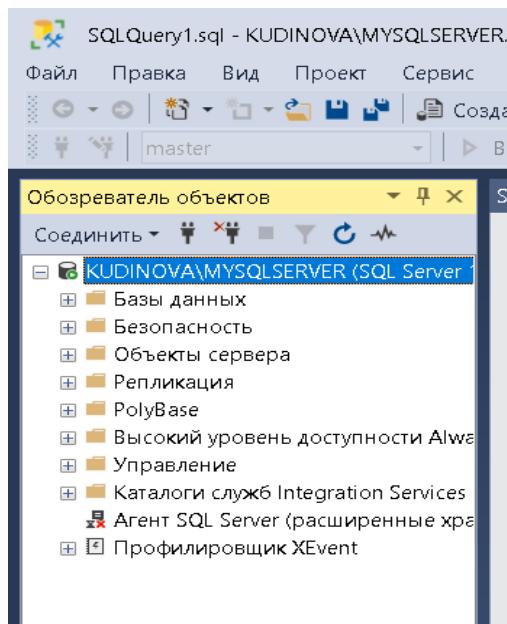


Рисунок 2.2 – Результат підключення до серверу

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

`CREATE DATABASE MobileShop`

Результат зображенено на рисунку 2.3.

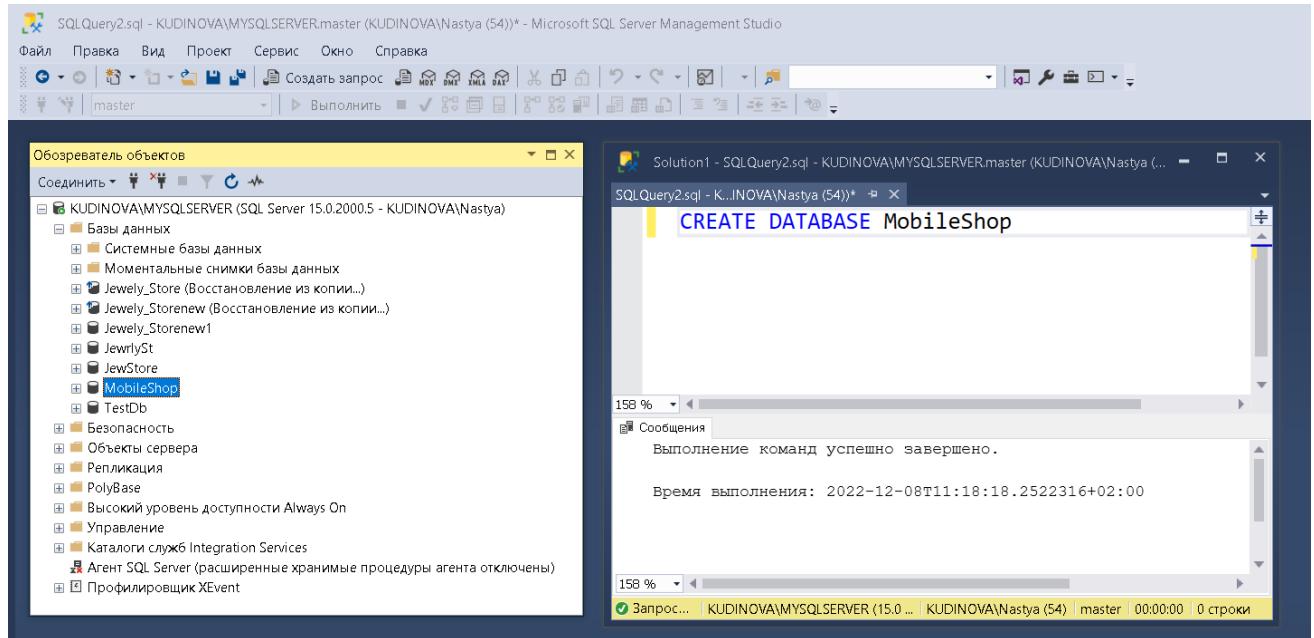


Рисунок 2.3 - Створення реляційної бази даних «Магазин мобільних телефонів»

2.1.2 Створення таблиць бази даних

В базі даних MobileShop створюємо таблиці наступні таблиці:

- інформація про замовлення (*Info_order*),
- замовлення (*Order*),
- клієнт (*Client*),
- ПІБ (*Full_name*),
- продукт (*Product*),
- модель (*Model*),
- виробник (*Producer*),
- параметри (*Parameters*),
- процесор (*Processor*),

- операційна система (Operation_system).

За допомогою наступних команд (рис. 2.4):

-- Створення таблиці ПІБ (Full_Name):

```
create table Full_Name (
id_Full_Name int primary key,
First_Name char(25),
Second_Name char(25),
Patronymic char(25) )
```

-- Створення таблиці Клієнт (Client):

```
create table Client (
Code_client int primary key,
id_Full_Name int foreign key references Full_Name
(id_Full_Name),
Phone int )
```

-- Створення таблиці Замовлення (Orders):

```
create table Orders (
Code_order int primary key,
Code_client int foreign key references Client (Code_client),
Datas date)
```

-- Створення таблиці Виробник (Producer):

```
create table Producer (
Code_producer int primary key,
Name_producer char(35),
Contact char(35))
```

-- Створення таблиці Процесор (Processor):

```
create table Processor(
id_Processor int primary key,
Ver_processor char(20),
Num_processor int)
```

--Створення таблиці Операційна система (Operation_system):

```
create table Operation_system (
Id_Operation_system int primary key,
Version_operation_system char (20),
Number_Version_operation_system int )
```

-- Створення таблиці Параметри (Parameters):

```
create table Parameters (
Code_parameter int primary key,
Memory_Gb int,
```

```
Camera_Mp int,
Battery_Ma int,
Id_Operation_system int foreign key references Operation_system
(Id_Operation_system),
Id_Processor int foreign key references Processor
(Id_Processor))
```

-- Створення таблиці Модель (Model):

```
create table Model (
Code_model int primary key,
Name_model char(25),
Code_parameter int foreign key references Parameters
(Code_parameter),
Code_producer int foreign key references Producer
(Code_producer))
```

-- Створення таблиці Продукт (Product):

```
create table Product (
Code_product int primary key,
Code_model int foreign key references Model (Code_model),
Price_product int,
All_in_stock int,
Name_product char(20))
```

-- Створення таблиці Інформація про замовлення (Info_order):

```
create table Info_order (
Code_info int primary key,
Code_product int foreign key references Product (Code_product),
Number int,
Code_order int foreign key references Orders (Code_order),
Price_order int)
```

```

-- Создание таблицы ИМС (Full_Name):
CREATE TABLE Full_Name (
    id_Full_Name int primary key,
    First_Name char(25),
    Second_Name char(25),
    City char(30)
);

-- Создание таблицы Клиент (Client):
CREATE TABLE Client (
    Code_client int primary key,
    id_Full_Name int foreign key references Full_Name (id_Full_Name),
    Phone int
);

-- Создание таблицы Товары Заказы (Orders):
CREATE TABLE Orders (
    Code_order int primary key,
    Code_client int foreign key references Client (Code_client),
    Data_date
);

-- Создание таблицы Выработчик (Producer):
CREATE TABLE Produces (
    Code_producer int primary key,
    Name_producer varchar(35),
    Contact char(35)
);

-- Создание таблицы Процессор (Processor):
CREATE TABLE Processor (
    id_Processor int primary key,
    Ver_processor char(20),
    Num_processor int
);

-- Создание таблицы Операционная система (Operation_system):
CREATE TABLE Operation_system (
    Id_Operation_system int primary key,
    Version_operation_system char (20),
    Number_Version_operation_system int
);

-- Создание таблицы Параметры (Parameters):
CREATE TABLE Parameters (
    Code_parameter int primary key,
    Memory_Gb int,
    Camera_Mp int,
    Battery_Ma int,
    Id_Operation_system int foreign key references Operation_system (Id_Operation_system),
    Id_Processor int foreign key references Processor (id_Processor)
);

-- Создание таблицы Модель (Model):
CREATE TABLE Model (
    Code_model int primary key,
    Name_model char(25),
    Code_parameter int foreign key references Parameters (Code_parameter),
    Code_producer int foreign key references Producer (Code_producer)
);

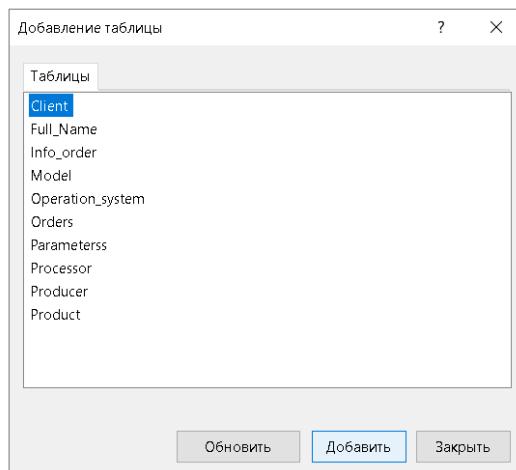
-- Создание таблицы Продукт (Product):
CREATE TABLE Product (
    Code_product int primary key,
    Code_model int foreign key references Model (Code_model),
    Price_product int
);

```

Рисунок 2.4 - Створення таблиць реляційної бази даних «Магазин мобільних телефонів»

2.1.3 Створення діаграми бази даних

У розділі діаграм в утиліті SQL Server Management Studio створюємо нову діаграму, в яку додаємо зі списку десять таблиць із даної предметної області та перевіряємо зв'язки між таблицями. Результат створення діаграми зображенено на рисунку 2.5.



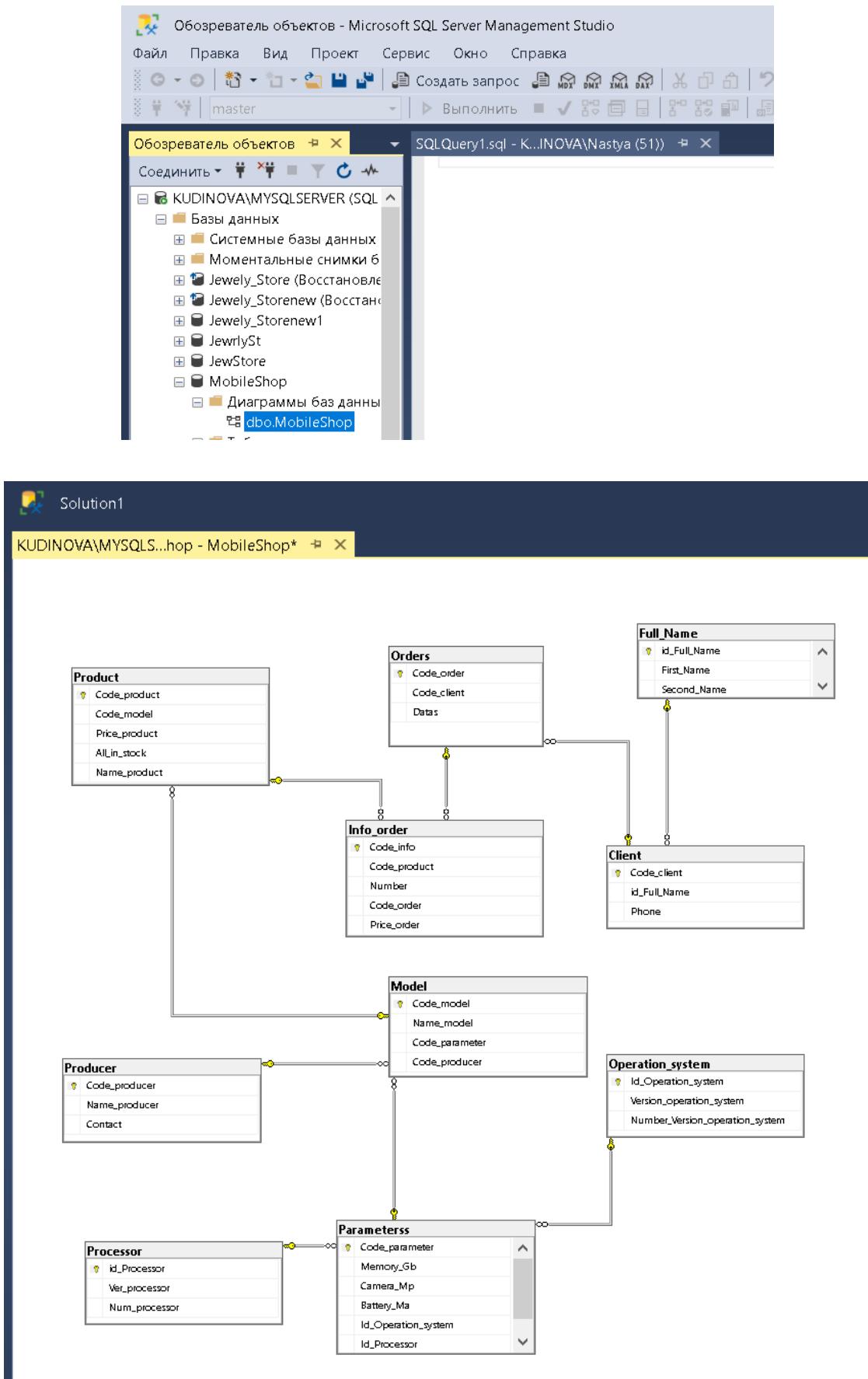


Рисунок 2.5 - Діаграма реляційної бази даних «Магазин мобільних телефонів»

2.1.4 Заповнення таблиць бази даних даними

Заповнимо таблицю «Продукт (*Product*)» даними рисунок 2.6:

```
INSERT INTO Product
```

```
VALUES (111, 1010, 7500, 4, 'Xiaomi'),  
(222, 2020, 16000, 25, 'LG'),  
(333, 3030, 27000, 32, 'Apple'),  
(444, 4040, 38000, 11, 'Samsung'),  
(555, 5050, 9000, 7, 'Lenovo'),  
(666, 6060, 11000, 9, 'Realme'),  
(777, 7070, 8500, 10, 'Nokia'),  
(888, 8080, 11200, 18, 'Meizu'),  
(999, 9090, 10000, 60, 'Motorola'),  
(1000, 10101, 29000, 30, 'Tecno')
```

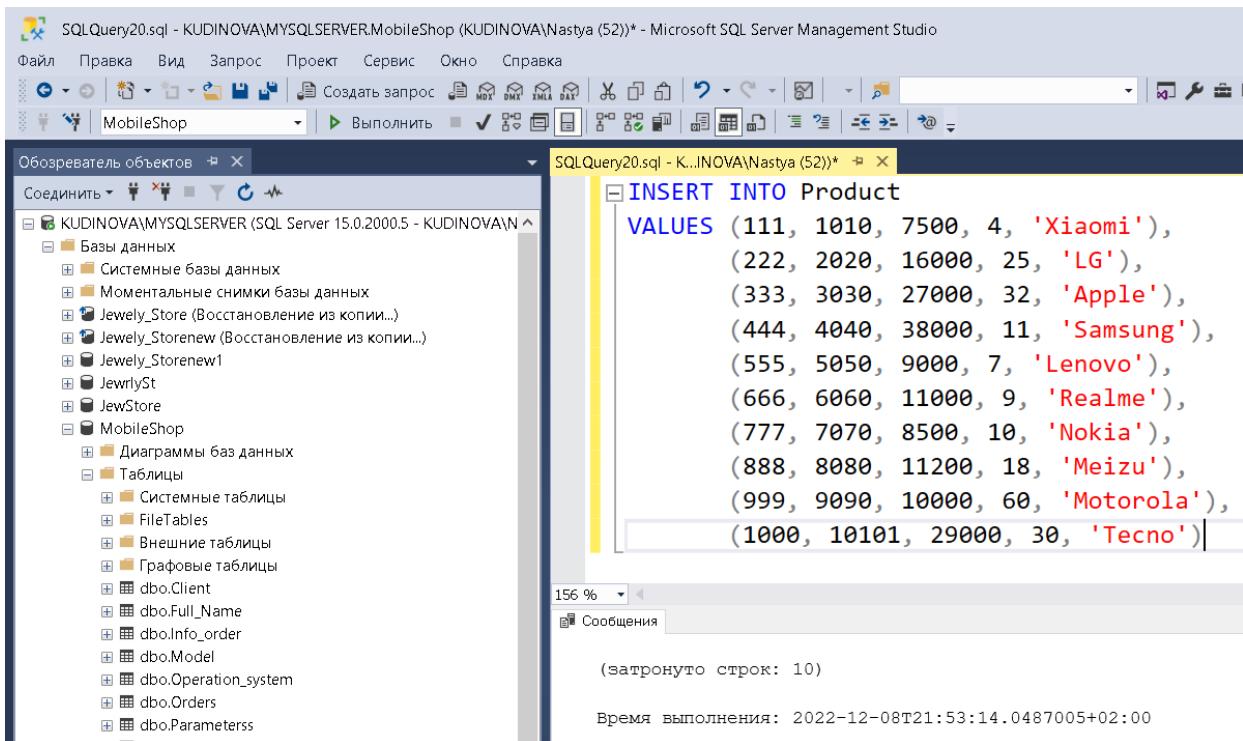


Рисунок 2.6 - вставка даних для таблиці Продукт (*Product*)

Переглянемо результат:

```
select * from Product
```

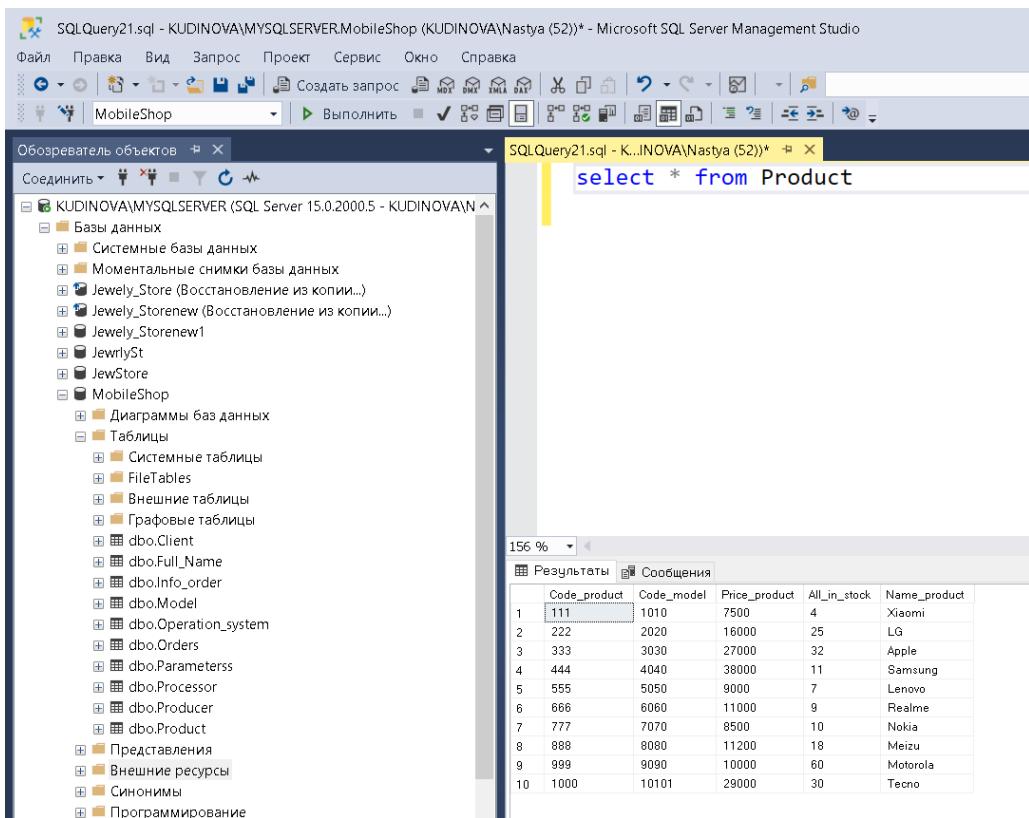


Рисунок 2.7 - результат вставки даних для таблиці Продукт (*Product*)

Далі заповнюємо інші таблиці реляційної бази даних «Магазин мобільних телефонів». (Додаток А – Заповнення таблиць реляційної бази даних).

2.1.5 Створення збережених процедур

Процедура, що зберігається, - це спеціальний тип пакету інструкцій Transact-SQL, створений, використовуючи мову SQL і процедурні розширення. При створенні процедури, що зберігається, можна визначити необов'язковий список параметрів. Таким чином, процедура прийматиме відповідні аргументи при кожному її виклику. Збережені процедури можуть повертати значення, яке містить певну користувачем інформацію або, у разі помилки, відповідне повідомлення про помилку.

Розробимо процедуру для отримання для отримання типу батареї, камери, пам'яті телефону, який відповідає назві моделі. Використаємо також пошук за параметром (рис -2.8):

```

create proc my_proc1
@name char
AS
select Parameterss.Battery_Ma, Parameterss.Camera_Mp,
Parameterss.Memory_Gb
from Parameterss join Model on Parameterss.Code_parameter =
Model.Code_parameter
where Model.Name_model = @name

```

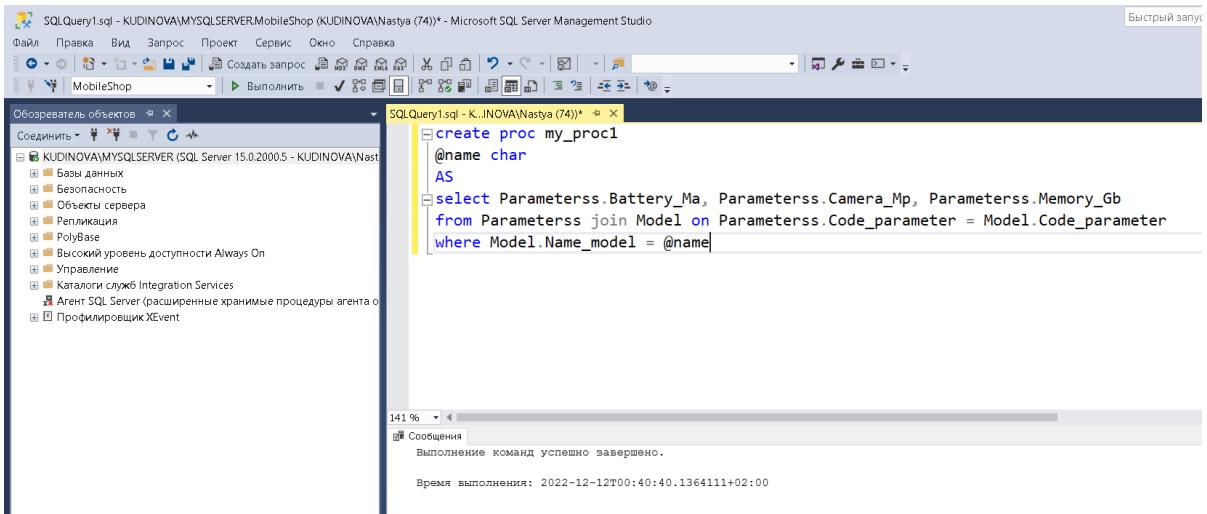


Рисунок 2.8 - Збережена процедура для отримання для отримання типу батареї, камери, пам'яті телефону, який відповідає назві моделі

Для звернення до процедури можна використовувати наступну команду (рис 2.9):

```
exec my_proc1 'Iphone 12'
```

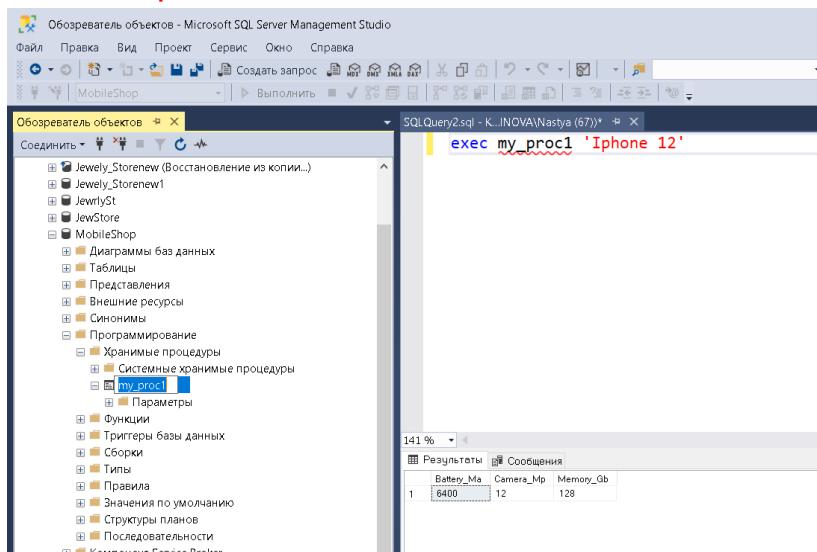


Рисунок 2.9 – Результат збереженої процедури my_proc1

Далі створюємо збережені процедури для інших таблиць реляційної бази даних «Магазин мобільних телефонів» (Додаток Б. Створення збережених процедур).

2.1.6 Створення тригерів

Тригери - спеціальний об'єкт бази даних, що описує перелік дій, які необхідно автоматично виконати при настанні вказаної події. Ім'я тригера може містити щонайбільше 128 символів. Дія тригера може бути або інструкція DML (INSERT, UPDATE або DELETE), або інструкція DDL. Таким чином, існує два типи тригерів: тригери DML та тригери DDL. Виконавча складова тригера зазвичай складається з процедури, що зберігається або пакета.

Створимо для таблиці «Виробник (Producer)» тригер, який забороняємо вставку нових рядків у таблицю, видаючи при цьому повідомлення 'Вставка рядків заборонена' (рис. 2.10):

```
create trigger prod
on Producer
for insert as
print 'Вставка рядків заборонена'
rollback tran
```

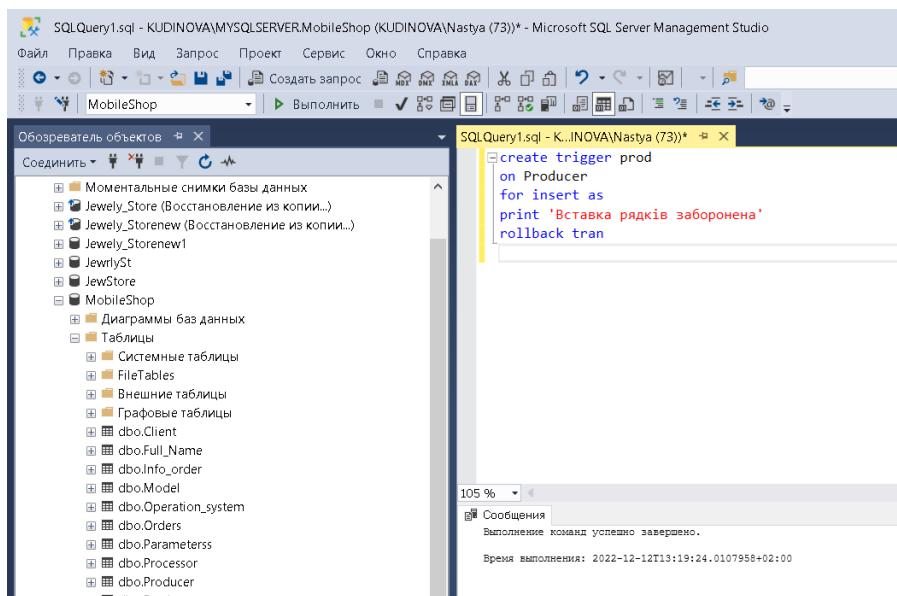


Рисунок 2.10 - тригер «Виробник (Producer)», який забороняємо вставку нових рядків у таблицю

Перевіримо працездатність тригера (рис. – 2.11):

```
insert into Producer  
values (7766, 'Mix', 'mixphone@gmail.com')
```

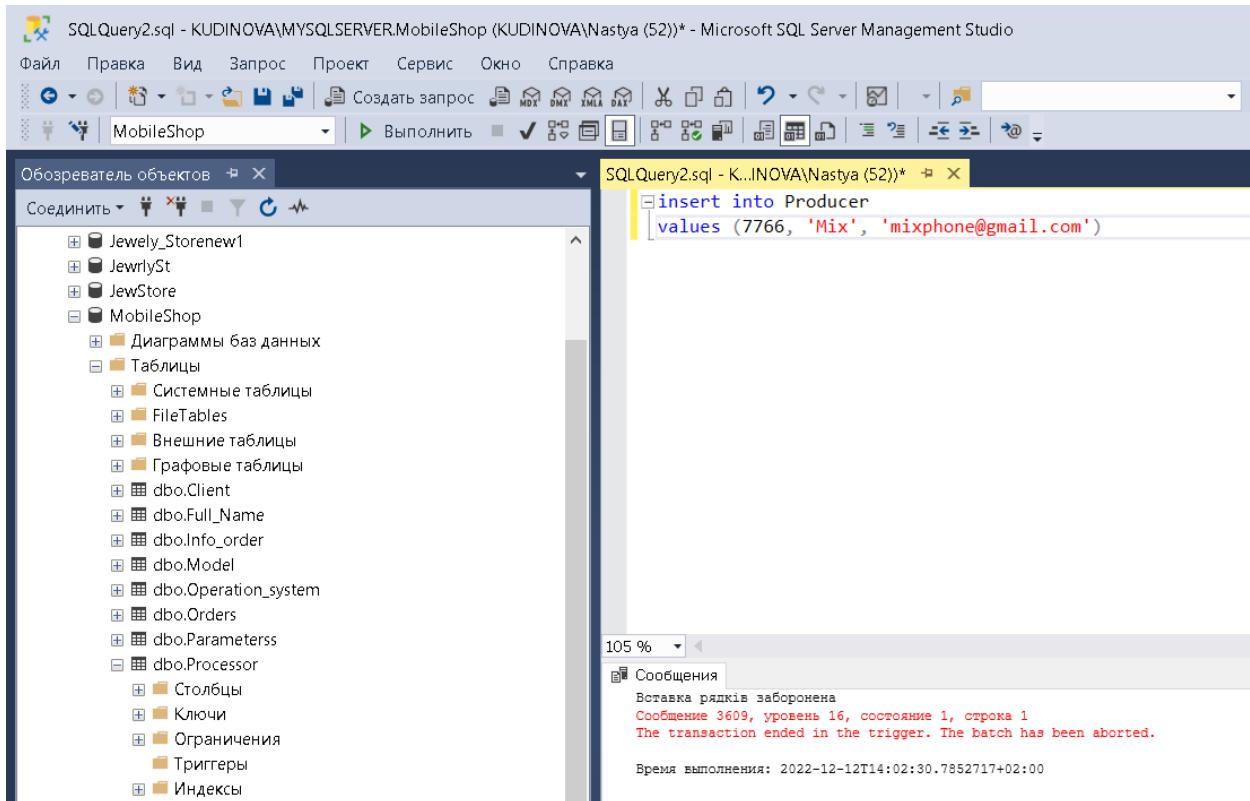


Рисунок 2.11 – Результат роботи тригера який забороняємо вставку нових рядків у таблицю «Виробник (Producer)»

(Додаток В. Створення тригерів).

2.1.7 Створення представлень

Створити представлення на основі обраних даних з трьох таблиць:

Product, Model, Parameters, Operation_system, Processor (рис. 2.12)

```
create view information_view  
as  
select  
Product.Price_product, Product.Name_product,  
Model.Name_model,  
Parameters.Memory_Gb, Parameters.Camera_Mp,  
Parameters.Battery_Ma,
```

```

Operation_system.Version_operation_system,
Operation_system.Number_Version_operation_system,
Processor.Ver_processor, Processor.Num_processor
from Product, Model, Parameterss, Operation_system, Processor
where Product.Code_model = Model.Code_model
AND Model.Code_parameter = Parameterss.Code_parameter
AND Parameterss.Id_Operation_system =
Operation_system.Id_Operation_system
AND Parameterss.Id_Processor = Processor.id_Processor

```

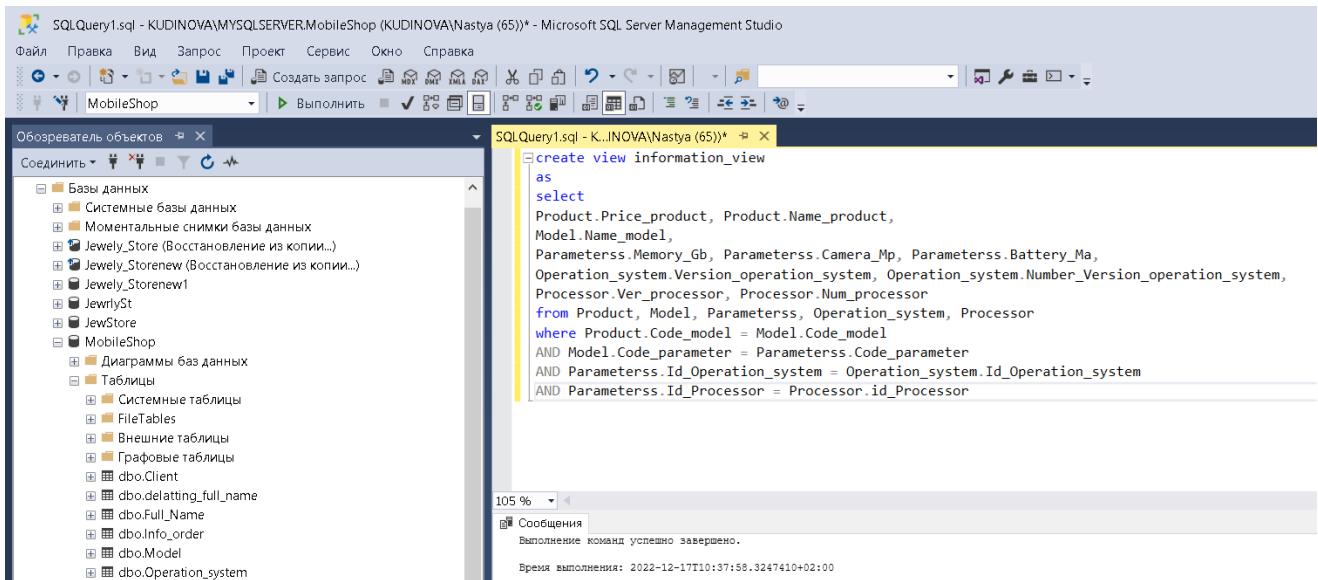


Рисунок 2.12 - представлення на основі обраних даних з троох таблиць:

Product, Model, Parameterss, Operation_system, Processor

Перевіримо результат: (рис. 2.12)

`select * from information_view`

| Price_product | Name_product | Name_model | Memory_Gb | Camera_Mp | Battery_Ma | Version_operation_system | Number_Version_operation_system | Ver_processor | Num_processor |
|---------------|--------------|---------------|-----------|-----------|------------|--------------------------|---------------------------------|---------------|---------------|
| 17900 | Samsung | Galaxy S21 | 8 | 12 | 5000 | 18 | 18 | Adreno | 889 |
| 2 58000 | Tecno | Spark 9C | 32 | 12 | 5000 | 18 | 18 | Adreno | 899 |
| 3 18000 | LG | Q7 ThinQ | 64 | 16 | 3000 | Android | 6 | Adreno | 915 |
| 4 27000 | Apple | iPhone 12 | 128 | 12 | 5400 | 105 | 14 | | Bionic |
| 5 38000 | Samsung | Galaxy Note20 | 256 | 14 | 3700 | Android | 13 | Adreno | 954 |
| 6 8900 | Lenovo | A6 | 16 | 54 | 8000 | Android | 11 | Adreno | 992 |
| 7 11000 | Realme | X Pro | 512 | 54 | 5000 | Android | 12 | Adreno | 993 |
| 8 8500 | Nokia | 810 | 11 | 35 | 7000 | Android | 18 | Adreno | 475 |
| 9 11200 | Motoz | NatPre 11 | 714 | 30 | 8000 | Android | 15 | Adreno | 947 |
| 10 18000 | Motorola | NatPre 11 | 200 | 70 | 9000 | Android | 18 | Adreno | 968 |

Рис 2.12 – Результат роботи представлення

(Додаток Г. Створення збережених процедур).

2.2 Організація вибірки інформації з бази даних

Вибірка – це пошук необхідних даних, а модифікація – додавання, видалення або зміна даних.

Для створення простого запиту на вибірку мовою SQL використовується ключове слово (оператор) SELECT.

Вибірка за допомогою оператора SELECT – найчастіша команда при роботі з реляційною БД. Даний оператор володіє великими можливостями до створення структури вихідної інформації, вказівки джерел вхідної інформації, способу впорядкування вихідної інформації, формуванню нових значень тощо.

Вибірка даних в sql корисна й ефективна річ, за допомогою якої можна робити певний аналіз даних, організовувати вибірку даних за власними вимогами, виконувати обрахунки елементарних речей.

2.2.1 Проста вибірка даних

Надіслати запит, який би видав ініціали клієнтів (рис. 2.13):

```
select * from Full_Name
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer tree view shows the database structure, including the 'MobileShop' database and its tables. In the center, the 'SQLQuery27.sql' query window contains the SQL command: 'select * from Full_Name'. To the right, the 'Results' tab displays the query results in a grid format:

| | id_Full_Name | First_Name | Second_Name | Patronymic |
|----|--------------|--------------|-------------|--------------|
| 1 | 11 | Пашковський | Максим | Ігорович |
| 2 | 12 | Драбовський | Анатолій | Григорович |
| 3 | 13 | Лінкан | Ганна | Василівна |
| 4 | 14 | Грищенко | Тетяна | Миколаївна |
| 5 | 15 | Савчук | Оксана | Василівна |
| 6 | 16 | Киселевський | Дмитро | Давидович |
| 7 | 17 | Васинчук | Іван | Олексійович |
| 8 | 18 | Карпенко | Мартин | Геннадійович |
| 9 | 19 | Іващенко | Вікторія | Сергіївна |
| 10 | 21 | Козаченко | Альона | Вікторівна |

Рисунок 2.13 – результат запиту, який видає інформацію про ініціали клієнтів

(Додаток Г. Проста вибірка даних).

2.2.2 Вибірка обчислюваних значень

Написати запит, який би визначив кількість виробників (рис. 2.14):

```
Select COUNT(Producer.Code_producer) as 'Кількість виробників'  
From Producer
```

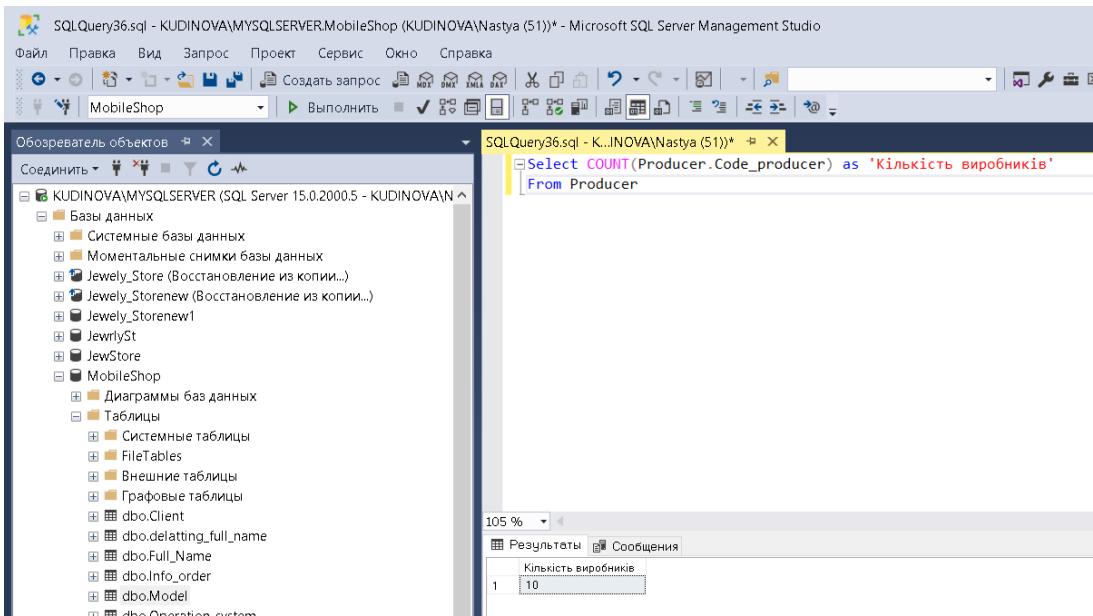


Рисунок 2.14 – Запит, який визначить кількість виробників

(Додаток Д. Вибірка обчислюваних значень).

2.2.3 Вибірка значень з діапазону

Написати запит, який би визначив камеру за назвою моделі, камера в яких знаходить в діапазоні від 10 до 14 (рис 2.15):

```
select Parameterss.Camera_Mp  
from Parameterss  
where Parameterss.Code_parameter in ( select  
Model.Code_parameter  
from Model where Model.Name_model = 'Iphone 12')  
and Parameterss.Camera_Mp BETWEEN 10 AND 14
```

```

SQLQuery40.sql - KUDINOVA\MySQLServer.MobileShop (KUDINOVA\Nastya (51)) - Microsoft SQL Server Management Studio
Файл Правка Вид Запрос Проект Сервис Окно Справка
Соединить > MobileShop Выполнить
Обозреватель объектов > SQLQuery40.sql - KUDINOVA\MySQLServer.MobileShop (KUDINOVA\Nastya (51))
Соединить > MobileShop Выполнить
SQLQuery40.sql - KUDINOVA\MySQLServer (SQL Server 15.0.2000.5 - KUDINOVA\N
Базы данных
Системные базы данных
Моментальные снимки базы данных
Jewely_Store (Восстановление из копии...)
Jewely_Storenew (Восстановление из копии...)
Jewely_Storenew1
JewlySt
JewStore
MobileShop
Диаграммы баз данных
Таблицы
Системные таблицы
FileTables
Внешние таблицы
Графовые таблицы
dbo.Client
dbo.deletting_full_name
dbo.Full_Name
dbo.info_order
dbo.Model
dbo.Operation_system
105 %
Результаты Сообщения
Camera_Mp
1 12

```

Рисунок 2.15 – результат, який би визначив камеру за назвою моделі, камера в яких знаходить в діапазоні від 10 до 14
(Додаток Е. Вибірка значень з діапазону).

2.2.4 Вибірка з використанням шаблонів

Написати запит, який би видав всі дані про клієнтів чиї імена починаються з літеру «M» (рис. 2.16):

```

select *
from Full_Name
where Full_Name.Second_Name like 'M%'

```

```

SQLQuery47.sql - KUDINOVA\MySQLServer.MobileShop (KUDINOVA\Nastya (51)) - Microsoft SQL Server Management Studio
Файл Правка Вид Запрос Проект Сервис Окно Справка
Соединить > MobileShop Выполнить
Обозреватель объектов > SQLQuery47.sql - KUDINOVA\MySQLServer.MobileShop (KUDINOVA\Nastya (51))
Соединить > MobileShop Выполнить
SQLQuery47.sql - KUDINOVA\MySQLServer (SQL Server 15.0.2000.5 - KUDINOVA\N
Базы данных
Системные базы данных
Моментальные снимки базы данных
Jewely_Store (Восстановление из копии...)
Jewely_Storenew (Восстановление из копии...)
Jewely_Storenew1
JewlySt
JewStore
MobileShop
Диаграммы баз данных
Таблицы
Системные таблицы
FileTables
Внешние таблицы
Графовые таблицы
dbo.Client
dbo.deletting_full_name
dbo.Full_Name
dbo.info_order
dbo.Model
dbo.Operation_system
105 %
Результаты Сообщения
id_Full_Name First_Name Second_Name Patronymic
1 11 Пашковський Максим Ігорович
2 18 Карпенко Мартин Генадійович

```

Рисунок 2.16 – Результат роботи запиту який би видав всі дані про клієнтів чиї імена починаються з літеру «M»

(Додаток Є. Вибірка з використанням шаблонів).

2.2.5 Угруповання даних при організації запитів

Написати запит, який би визначив скільки замовлення здійснив кожен клієнт та угрупувати ці данні по полям «прізвище», «ім'я» та «по батькові» (рис. 2.15):

```
select Full_Name.First_Name as 'Прізвище',
Full_Name.Second_Name as 'Імя',
Full_Name.Patronymic as 'Побатькові',
count (Orders.Code_order) as 'Кількість'
from Full_Name, Info_order, Orders, Client
WHERE Info_order.Code_order = Orders.Code_order
AND Orders.Code_client = Client.Code_client
AND Client.id_Full_Name =Full_Name.id_Full_Name
group by Full_Name.First_Name, Full_Name.Second_Name,
Full_Name.Patronymic
order by Full_Name.First_Name, Full_Name.Second_Name,
Full_Name.Patronymic
```

| Прізвище | Імя | Побатькові | Кількість |
|--------------|----------|-------------|-----------|
| Васинук | Іван | Олексійович | 1 |
| Грищенко | Тетяна | Миколаївна | 1 |
| Драбоський | Анатолій | Григорович | 1 |
| Івашенко | Вікторія | Сергіївна | 1 |
| Карпенко | Мартін | Генадійович | 1 |
| Киселевський | Дмитро | Давидович | 1 |
| Козаченко | Альона | Вікторіна | 1 |
| Лічман | Ганна | Василівна | 1 |
| Пашковський | Максим | Ігорович | 1 |
| Савчук | Оксана | Василівна | 1 |

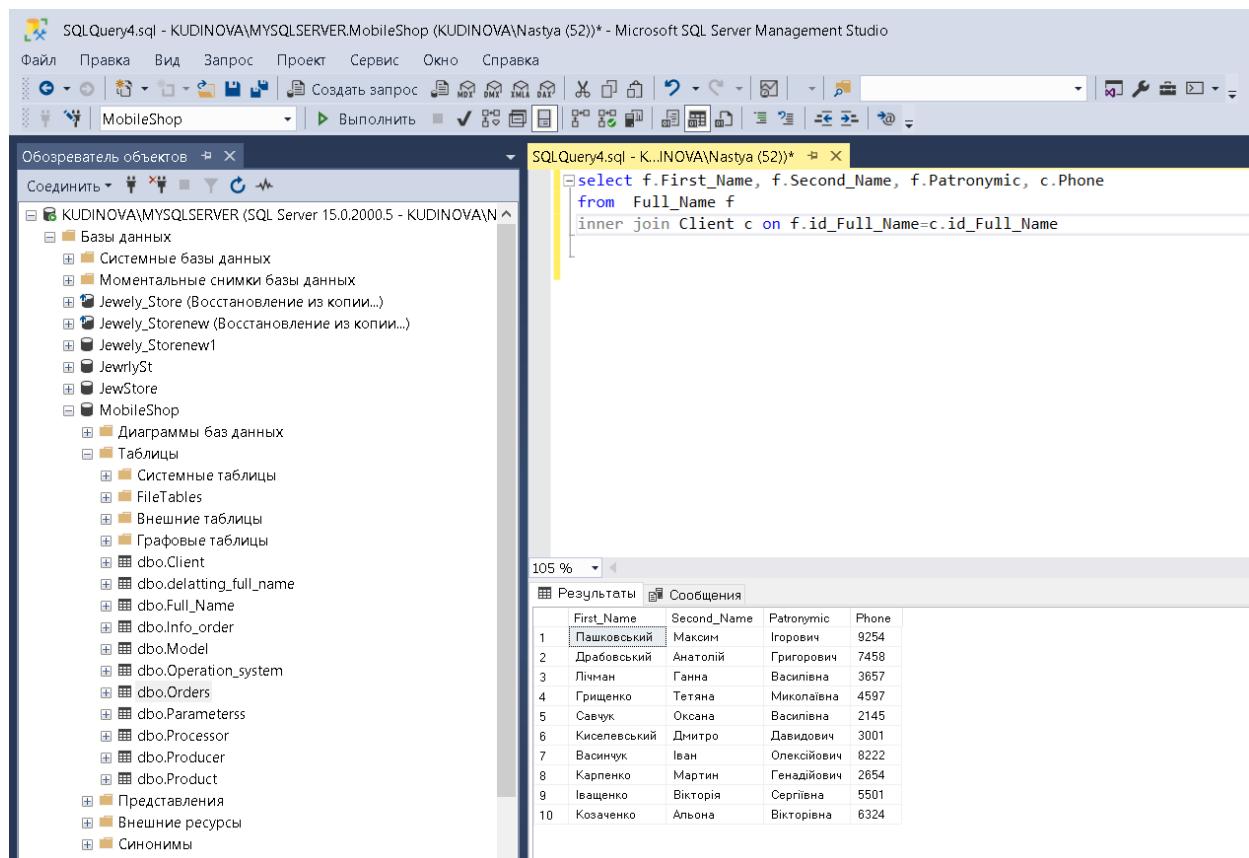
Рисунок 2.17 – Результат запиту

(Додаток Ж. Угруповання даних при організації запитів).

2.2.6 Об'єднання таблиць

Написати запит, який би визначив номер телефону клієнту (рис. 2.18):

```
select f.First_Name, f.Second_Name, f.Patronymic, c.Phone  
from Full_Name f  
inner join Client c on f.id_Full_Name=c.id_Full_Name
```



```
SQLQuery4.sql - KUDINOVA\Nastya (52)* - Microsoft SQL Server Management Studio
```

```
select f.First_Name, f.Second_Name, f.Patronymic, c.Phone  
from Full_Name f  
inner join Client c on f.id_Full_Name=c.id_Full_Name
```

| | First_Name | Second_Name | Patronymic | Phone |
|----|--------------|-------------|---------------|-------|
| 1 | Пашковський | Максим | Ігорович | 9254 |
| 2 | Драбовський | Анатолій | Григорович | 7458 |
| 3 | Лічман | Ганна | Василівна | 3657 |
| 4 | Грищенко | Тетяна | Миколаївна | 4597 |
| 5 | Савчук | Оксана | Василівна | 2145 |
| 6 | Киселевський | Дмитро | Давидович | 3001 |
| 7 | Васинчук | Іван | Олександрович | 8222 |
| 8 | Карпенко | Мартин | Генадійович | 2654 |
| 9 | Іващенко | Вікторія | Сергіївна | 5501 |
| 10 | Козаченко | Альона | Вікторівна | 6324 |

Рисунок 2.18 – Результат запиту, який би визначив номер телефону клієнту

(Додаток З. Об'єднання таблиць).

ВИСНОВКИ

Метою курсової роботи було створення бази даних «Магазин мобільних телефонів». Під час виконання даної роботи ми проектували та проводили аналіз бази даних. В результаті, якої ми отримали реальний спосіб керування даними магазину, який упорядкує всі товари та дає змогу проводити вибірку даних.

Для актуалізації нашої бази даних ми проводили аналіз та опис предметної області, що дало нам розуміння з якою інформацією ми будемо працювати. Наступним кроком було проектування бази даних. Створили інфологічну модель, тобто відобразили у вигляді атрибутів та сущностей - це допомогло нам сприймати інформацію краще та ефективніше. При побудові інфологічної моделі, використали ER - діаграму (сущність-зв'язок) область застосування є усунення несправностей та проблем з логікою. Також було проведено нормалізацію таблиць з використанням нормальній форм та виправлено помилки, які були допущені раніше. Нормально форму визначають як сукупність вимог, якими має задовільнити відношення. Використання складних форм нам не знадобилось, так як у предметній області не було знайдено складних залежностей. Проектування завершилось створенням датологічної моделі.

Наступним кроком було створення бази даних, її таблиці та заповнено даними у середовищі Microsoft Sql Server 2018. Збережені процедури, запити, тригери, представлення реалізовано мовою SQL- Transact.

В даній курсовій роботі була розроблена база даних «Магазин мобільних телефонів».

В заключенні було створено ефективну базу даних, яка забезпечить комфортне використання та управління даними, що дає змогу покращувати якість та швидкість обслуговування клієнтів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Алан Болье O'Reilly Media, Inc. Северный Севастополь, Гравенштайнское шоссе, 1005 Изучение SQL.
2. Бейли Л. Б41 Изучаем SQL. - СПб.: Питер, 2012. - 592 с.
3. Волк В. К. Базы данных. Проектирование, программирование, управление и администрирование: учебник / В. К. Волк. — СанктПетербург: Лань, 2020. — 244 с.
4. Грофф, Джеймс Р., Вайнберг, Пол Н., Оппель, Эндрю Дж. Г89 SQL: полное руководство, 3-е изд: Пер. с англ. - М.: ООО "И.Д. Вильяме", 2015. - 960 с.
5. Дейт, К. Дж. Д27 Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом "Вильяме", 2005. — 1328 с
6. Карвин Б. Программирование баз данных SQL. Типичные ошибки и их устранение / Б. Карвин. - М.: Рид Групп, 2012. -336 с.
7. Кэмпбелл Лейн, Майджорс Черити К98 Базы данных. Инжениринг надежности. — СПб.: Питер, 2020. — 304 с.: ил. — (Серия «Бестселлеры O'Reilly»).
8. Кригель, Алекс, Трухнов, Борис. К82 SQL. Библия пользователя, 2-е издание: Пер. с англ. — М.: ООО “И.Д. Вильямс”, 2010. — 752 с.
9. Куликов, С. С. К90 Реляционные базы данных в примерах: практическое пособие для программистов и тестировщиков / С. С. Куликов. — Минск: Четыре четверти, 2020. — 424 ст.
10. Мартин Грабер. SOL для простых смертных Переводчик В.А. Ястребов Научный редактор П.И. Быстров. Верстка М. Алиевой. Copyright © 1990 SYBEX Inc., 2021 Challenger Drive, Alameda, CA 94501. Перевод © Издательство «ЛОРИ», 2014.
11. Молинаро Э. SQL. Сборник рецептов. – Пер. с англ. – СПб: Символ!Плюс, 2009. – 672 с.

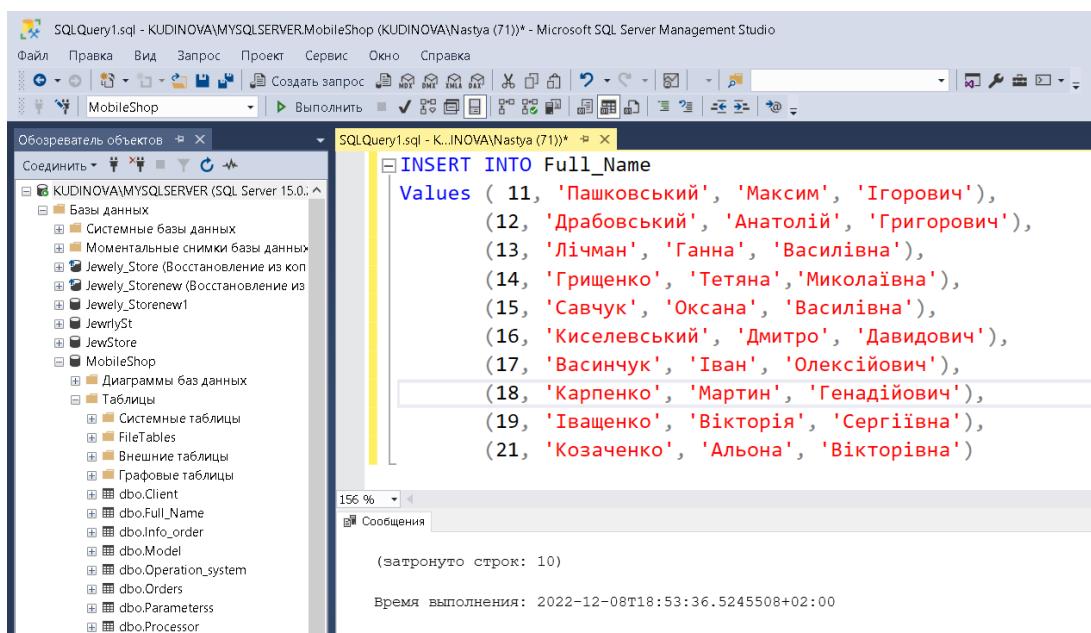
12. Моргунов, Е. П. М79 Язык SQL. Базовый курс: учеб.-практ. пособие / Е. П. Моргунов; под ред. Е. В. Рогова, П. В. Лузанова; Postgres Professional. — М., 2017. — 256 с.
13. Новиков Б. А. Н73 Основы технологий баз данных: учеб. пособие / Б. А. Новиков, Е. А. Горшкова; под ред. Е. В. Рогова. — М.: ДМК Пресс, 2019. — 240 с.
14. Осипов Д. Л. Технологии проектирования баз данных. — М.: ДМК Пресс, 2019. — 498 с.
15. Шилдс Уолтер Ш57 SQL: быстрое погружение. — СПб.: Питер, 2022. — 224 с.: ил. — (Серия «Библиотека программиста»).

ДОДАТКИ

Додаток А. Заповнення таблиць реляційної бази даних

1. Заповнимо таблицю «ПІБ (*Full_Name*)» даними:

```
INSERT INTO Full_Name
Values (11, 'Пашковський', 'Максим', 'Ігорович'),
(12, 'Драбовський', 'Анатолій', 'Григорович'),
(13, 'Лічман', 'Ганна', 'Василівна'),
(14, 'Грищенко', 'Тетяна', 'Миколаївна'),
(15, 'Савчук', 'Оксана', 'Василівна'),
(16, 'Киселевський', 'Дмитро', 'Давидович'),
(17, 'Васинчук', 'Іван', 'Олексійович'),
(18, 'Карпенко', 'Мартин', 'Генадійович'),
(19, 'Іващенко', 'Вікторія', 'Сергіївна'),
(21, 'Козаченко', 'Альона', 'Вікторівна')
```



Переглянемо результат:

```
Select * from Full_Name
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center, there is a query results grid titled 'Select * from Full_Name'. The grid contains 10 rows of data with columns: id_Full_Name, First_Name, Second_Name, and Patronymic. The data is as follows:

| | id_Full_Name | First_Name | Second_Name | Patronymic |
|----|--------------|-------------|-------------|---------------|
| 1 | 11 | Пашковский | Максим | Игоревич |
| 2 | 12 | Драбовский | Анатолий | Григорович |
| 3 | 13 | Личман | Ганна | Василівна |
| 4 | 14 | Грищенко | Тетяна | Миколаївна |
| 5 | 15 | Савчук | Оксана | Василівна |
| 6 | 16 | Киселевский | Дмитро | Давидович |
| 7 | 17 | Васинчук | Іван | Олександрович |
| 8 | 18 | Карпенко | Мартин | Генадійович |
| 9 | 19 | Іващенко | Вікторія | Сергіївна |
| 10 | 21 | Козаченко | Альона | Вікторівна |

2. Заповнимо таблицю «Клієнт (Client)» даними:

```
INSERT INTO Client
VALUES (10, 11, 9254),
       (20, 12, 7458),
       (30, 13, 3657),
       (40, 14, 4597),
       (50, 15, 2145),
       (60, 16, 3001),
       (70, 17, 8222),
       (80, 18, 2654),
       (90, 19, 5501),
       (100, 21, 6324)
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center, there is a query editor window titled 'SQLQuery3.sql - KUDINOVA\Nastya (52)* - Microsoft SQL Server Management Studio'. The query is:

```
INSERT INTO Client
VALUES (10, 11, 9254),
       (20, 12, 7458),
       (30, 13, 3657),
       (40, 14, 4597),
       (50, 15, 2145),
       (60, 16, 3001),
       (70, 17, 8222),
       (80, 18, 2654),
       (90, 19, 5501),
       (100, 21, 6324)
```

Below the query, the status bar indicates: '(затронуто строк: 10)' and 'Время выполнения: 2022-12-08T19:23:32,3901622+02:00'

Переглянемо результат:

Select * from Client

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left is the Object Explorer pane, which lists the database 'KUDINOVA\MySQLServer.MobileShop' with its objects: Базы данных, Таблицы, and other system objects like dbo.Client, dbo.Full_Name, etc. In the center is the Results pane, titled 'SQLQuery4.sql - KUDINOVA\Nastya (62)*'. It contains the query 'Select * from Client'. Below the query is a table with 10 rows of data:

| | Code_client | id_Full_Name | Phone |
|----|-------------|--------------|-------|
| 1 | 10 | 11 | 8254 |
| 2 | 20 | 12 | 7458 |
| 3 | 30 | 13 | 3657 |
| 4 | 40 | 14 | 4597 |
| 5 | 50 | 15 | 2145 |
| 6 | 60 | 16 | 3001 |
| 7 | 70 | 17 | 8222 |
| 8 | 80 | 18 | 2654 |
| 9 | 90 | 19 | 5501 |
| 10 | 100 | 21 | 6324 |

3. Заповнимо таблицю «Замовлення (Orderss)» даними:

```
INSERT INTO Orders
VALUES (101, 10, '20221128'),
       (202, 20, '20221129'),
       (303, 30, '20220902'),
       (404, 40, '20221208'),
       (505, 50, '20221201'),
       (606, 60, '20221021'),
       (707, 70, '20220922'),
       (808, 80, '20220405'),
       (909, 90, '20221117'),
       (1011, 100, '20220710')
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left is the Object Explorer pane, listing the database 'KUDINOVA\MySQLServer.MobileShop' with its objects. In the center is the Results pane, titled 'SQLQuery5.sql - KUDINOVA\Nastya (61)*'. It contains the query 'INSERT INTO Orders'. Below the query is a message indicating that 10 rows were affected. At the bottom of the Results pane, the execution time is shown as 'Время выполнения: 2022-12-08T19:48:10.0870971+02:00'.

Переглянемо результат:

```
select * from Orders
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure for 'KUDINOVA\MySQLServer'. In the center, the 'SQLQuery8.sql' window contains the query: 'select * from Orders'. To the right, the 'Results' pane shows the output of the query:

| Code_order | Code_client | Datas |
|------------|-------------|------------|
| 1 | 101 | 2022-11-28 |
| 2 | 202 | 2022-11-29 |
| 3 | 303 | 2022-09-02 |
| 4 | 404 | 2022-12-08 |
| 5 | 505 | 2022-12-01 |
| 6 | 606 | 2022-10-21 |
| 7 | 707 | 2022-09-22 |
| 8 | 808 | 2022-04-05 |
| 9 | 909 | 2022-11-17 |
| 10 | 1011 | 2022-07-10 |

4. Заповнимо таблицю «Процесор (Processor) » даними:

```
INSERT INTO Processor
VALUES (37, 'Snapdragon', 680),
       (38, 'Snapdragon', 845),
       (39, 'Bionic', 14),
       (41, 'Snapdragon', 8 ),
       (42, 'Snapdragon', 662),
       (43, 'Snapdragon', 695 ),
       (44, 'Snapdragon', 475 ),
       (45, 'Snapdragon', 542),
       (46, 'Snapdragon', 968),
       (47, 'Snapdragon', 123)
```

```

SQLQuery10.sql - KUDINOVA\MySQLServer.MobileShop (KUDINOVA\Nastya (52)) - Microsoft SQL Server Management Studio
Файл Правка Вид Запрос Проект Сервис Окно Справка
Соединить | Выполнить | Создать запрос MDX DML XML DAT
MobileShop | Выполнить | Соединение с базой данных | Таблицы | Внешние таблицы | Графовые таблицы | Системные таблицы | FileTables | Диаграммы баз данных | Таблицы | Базы данных | Системные базы данных | Моментальные снимки базы данных | Jewely_Store (Восстановление из копии...) | Jewely_Stornew (Восстановление из копии...) | Jewely_Stornew1 | JewlySt | JewStore | MobileShop | Processor
Обозреватель объектов | SQLQuery10.sql - KUDINOVA\Nastya (52)*
INSERT INTO Processor
VALUES (37, 'Snapdragon', 680),
       (38, 'Snapdragon', 845),
       (39, 'Bionic', 14),
       (41, 'Snapdragon', 8),
       (42, 'Snapdragon', 662),
       (43, 'Snapdragon', 695),
       (44, 'Snapdragon', 475),
       (45, 'Snapdragon', 542),
       (46, 'Snapdragon', 968),
       (47, 'Snapdragon', 123)

```

(затронуто строк: 10)

Время выполнения: 2022-12-08T20:33:13.6119415+02:00

Переглянемо результат:

```
select * from Processor
```

```

SQLQuery11.sql - KUDINOVA\MySQLServer.MobileShop (KUDINOVA\Nastya (51)) - Microsoft SQL Server Management Studio
Файл Правка Вид Запрос Проект Сервис Окно Справка
Соединить | Выполнить | Создать запрос MDX DML XML DAT
MobileShop | Выполнить | Соединение с базой данных | Таблицы | Внешние таблицы | Графовые таблицы | Системные таблицы | FileTables | Диаграммы баз данных | Таблицы | Базы данных | Системные базы данных | Моментальные снимки базы данных | Jewely_Store (Восстановление из копии...) | Jewely_Stornew (Восстановление из копии...) | Jewely_Stornew1 | JewlySt | JewStore | MobileShop | Processor
Обозреватель объектов | SQLQuery11.sql - KUDINOVA\Nastya (51)*
select * from Processor

```

| | id_Processor | Ver_processor | Num_processor |
|----|--------------|---------------|---------------|
| 1 | 37 | Snapdragon | 680 |
| 2 | 38 | Snapdragon | 845 |
| 3 | 39 | Bionic | 14 |
| 4 | 41 | Snapdragon | 8 |
| 5 | 42 | Snapdragon | 662 |
| 6 | 43 | Snapdragon | 695 |
| 7 | 44 | Snapdragon | 475 |
| 8 | 45 | Snapdragon | 542 |
| 9 | 46 | Snapdragon | 968 |
| 10 | 47 | Snapdragon | 123 |

5. Заповнимо таблицю «Операційна система (*Operation_system*)» даними:

```

INSERT INTO Operation_system
VALUES (25, 'Android', 10),
       (26, 'Android', 8),
       (27, 'IOS', 14),
       (28, 'Android', 13),
       (29, 'Android', 11),
       (31, 'Android', 12),
       (32, 'Android', 19),
       (33, 'Android', 15),

```

```
(34, 'Android', 16),
(35, 'Android', 18)
```

The screenshot shows the SQL Server Management Studio interface. On the left, the Object Explorer displays the database structure under 'KUDINOVA\MySQLServer'. The central pane shows the query window titled 'SQLQuery12.sql' with the following SQL code:

```
INSERT INTO Operation_system
VALUES (25, 'Android', 10),
       (26, 'Android', 8),
       (27, 'IOS', 14),
       (28, 'Android', 13),
       (29, 'Android', 11),
       (31, 'Android', 12),
       (32, 'Android', 19),
       (33, 'Android', 15),
       (34, 'Android', 16),
       (35, 'Android', 18)
```

Below the code, it says '(затронуто строк: 10)' and 'Время выполнения: 2022-12-08T20:48:24.8434808+02:00'.

Переглянемо результат:

```
select * from Operation_system
```

The screenshot shows the results of the previous query. The results pane displays the following data:

| | Id_Operation_system | Version_operation_system | Number_Version_operation_system |
|----|---------------------|--------------------------|---------------------------------|
| 1 | 25 | Android | 10 |
| 2 | 26 | Android | 8 |
| 3 | 27 | IOS | 14 |
| 4 | 28 | Android | 13 |
| 5 | 29 | Android | 11 |
| 6 | 31 | Android | 12 |
| 7 | 32 | Android | 19 |
| 8 | 33 | Android | 15 |
| 9 | 34 | Android | 16 |
| 10 | 35 | Android | 18 |

6. Заповнимо таблицю «Параметри (Parameters)» даними:

```
INSERT INTO Parameters
VALUES (1111, 32, 13, 5000, 25, 37),
       (2222, 64, 16, 3000, 26, 38 ),
       (3333, 128, 12, 6400, 27, 39),
       (4444, 256, 14, 3700, 28, 41),
       (5555, 16, 64, 6000, 29, 42),
       (6666, 512, 54, 5500, 31, 43),
       (7777, 11, 35, 7000, 32, 44),
       (8888, 714, 30, 8000, 33, 45),
```

```
(9999, 200, 70, 9000, 34, 46),
(1212, 8, 109, 9500, 35, 47 )
```

The screenshot shows the SQL Server Management Studio interface. On the left, the Object Explorer displays the database structure for 'KUDINOVA\MySQLServer.MobileShop'. In the center, the 'SQLQuery14.sql' window contains the following SQL code:

```
INSERT INTO Parameters
VALUES (1111, 32, 13, 5000, 25, 37),
(2222, 64, 16, 3000, 26, 38 ),
(3333, 128, 12, 6400, 27, 39),
(4444, 256, 14, 3700, 28, 41),
(5555, 16, 64, 6000, 29, 42),
(6666, 512, 54, 5500, 31, 43),
(7777, 11, 35, 7000, 32, 44),
(8888, 714, 30, 8000, 33, 45),
(9999, 200, 70, 9000, 34, 46),
(1212, 8, 109, 9500, 35, 47 )|
```

Below the code, it says '(затронуто строк: 10)' (10 rows affected) and 'Время выполнения: 2022-12-08T21:07:35.8230960+02:00'.

Переглянемо результат:

```
select * from Parameters
```

The screenshot shows the SQL Server Management Studio interface. On the left, the Object Explorer displays the database structure for 'KUDINOVA\MySQLServer.MobileShop'. In the center, the 'SQLQuery15.sql' window contains the following SQL code:

```
select * from Parameters
```

Below the code, the 'Результаты' (Results) tab shows the following data:

| | Code_parameter | Memory_Gb | Camera_Mp | Battery_Ma | Id_Operation_system | Id_Processor |
|----|----------------|-----------|-----------|------------|---------------------|--------------|
| 1 | 1111 | 32 | 13 | 5000 | 25 | 37 |
| 2 | 1212 | 8 | 109 | 9500 | 35 | 47 |
| 3 | 2222 | 64 | 16 | 3000 | 26 | 38 |
| 4 | 3333 | 128 | 12 | 6400 | 27 | 39 |
| 5 | 4444 | 256 | 14 | 3700 | 28 | 41 |
| 6 | 5555 | 16 | 64 | 6000 | 29 | 42 |
| 7 | 6666 | 512 | 54 | 5500 | 31 | 43 |
| 8 | 7777 | 11 | 35 | 7000 | 32 | 44 |
| 9 | 8888 | 714 | 30 | 8000 | 33 | 45 |
| 10 | 9999 | 200 | 70 | 9000 | 34 | 46 |

7. Заповнимо таблицю «Виробник (Producer)» даними:

```
INSERT INTO Producer
VALUES(1100, 'XiaomiCorporation', 'xiaomiinc24558@gmail.com'),
(2200, 'Lg electronics', 'lggroup@gmail.com'),
(3300, 'Apple Company', 'appl@gmail.com'),
(4400, 'Samsung Group', 'samsunggroup741@gmail.com'),
(5500, 'LenovoGroupLimited', 'lenovolimited1234@gmail.com'),
(6600, 'Realme Inc', 'realmeilc@gmail.com'),
(7700, 'Nokia Com', 'nokia@gmail.com'),
(8800, 'Meizu LL', 'meizu@gmail.ll'),
(9900, 'Motorola Inc', 'motorola@gmail.com'),
```

(5896, 'TechoMix', 'tecnogmail.com')

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

```
INSERT INTO Producer
VALUES (1100, 'Xiaomi Corporation', 'xiaomiinc24558@gmail.com'),
       (2200, 'Lg electronics', 'lggroup@gmail.com'),
       (3300, 'Apple Company', 'appl@gmail.com'),
       (4400, 'Samsung Group', 'samsunggroup741@gmail.com'),
       (5500, 'Lenovo Group Limited', 'lenovolimited1234@gmail.com'),
       (6600, 'Realme Inc', 'realmeilc@gmail.com'),
       (7700, 'Nokia Com', 'nokia@gmail.com'),
       (8800, 'Meizu LL', 'meizu@gmail.ll'),
       (9900, 'Motorola Inc', 'motorola@gmail.com'),
       (5896, 'TechoMix', 'tecnogmail.com')
```

The status bar at the bottom indicates the execution time: 2022-12-08T21:28:52.3328700+02:00.

Переглянемо результат:

```
select * from Producer
```

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

```
select * from Producer
```

The results pane shows the following data:

| Code_producer | Name_producer | Contact |
|---------------|----------------------|-----------------------------|
| 1 | Xiaomi Corporation | xiaomiinc24558@gmail.com |
| 2 | Lg electronics | lggroup@gmail.com |
| 3 | Apple Company | appl@gmail.com |
| 4 | Samsung Group | samsunggroup741@gmail.com |
| 5 | Lenovo Group Limited | lenovolimited1234@gmail.com |
| 6 | Realme Inc | realmeilc@gmail.com |
| 7 | Nokia Com | nokia@gmail.com |
| 8 | Meizu LL | meizu@gmail.ll |
| 9 | Motorola Inc | motorola@gmail.com |
| 10 | TechoMix | tecnogmail.com |

8. Заповнимо таблицю «Модель (Model) » даними:

```
INSERT INTO Model
VALUES (1010, 'Redmi 9C', 1111, 1100),
       (2020, 'G7 ThinQ', 2222, 2200),
       (3030, 'Iphone 12', 3333, 3300),
       (4040, 'Galaxy Flip4', 4444, 4400),
       (5050, 'K12 Pro', 5555, 5500),
       (6060, '9 Pro', 6666, 6600),
       (7070, 'G10', 7777, 7700),
       (8080, 'Net Pro 11', 8888, 8800),
```

```
(9090, 'Net Pro 11', 9999, 9900),  
(10101, 'Spark 8C', 1212, 5896 )
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with the connection path: KUDINOVA\MySQLServer (SQL Server 15.0.2000.5 - KUDINOVA\Nasty). The right pane shows a query window titled "SQLQuery18.sql - KUDINOVA\Nasty (52)*" containing the following SQL code:

```
INSERT INTO Model
VALUES (1010, 'Redmi 9C', 1111, 1100),
(2020, 'G7 ThinQ', 2222, 2200),
(3030, 'Iphone 12', 3333, 3300),
(4040, 'Galaxy Flip4', 4444, 4400),
(5050, 'K12 Pro', 5555, 5500),
(6060, '9 Pro', 6666, 6600),
(7070, 'G10', 7777, 7700),
(8080, 'Net Pro 11', 8888, 8800),
(9090, 'Net Pro 11', 9999, 9900),
(10101, 'Spark 8C', 1212, 5896)
```

Переглянемо результат:

```
select * from Model
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "SQLQuery19.sql - KUDINOVA\MySQLServer.MobileShop (KUDINOVA\Nastya (52))* - Microsoft SQL Server Management Studio". The menu bar includes "Файл", "Правка", "Вид", "Запрос", "Проект", "Сервис", "Окно", and "Справка". The toolbar contains various icons for database management tasks.

The Object Explorer on the left displays the database structure for "MobileShop". It includes nodes for "Базы данных", "Системные базы данных", "Моментальные снимки базы данных", "Jewelry_Store (Восстановление из копии...)", "Jewelry_Storenew (Восстановление из копии...)", "Jewelry_Storenew1", "JewelrySt", "JewStore", and "MobileShop". Under "MobileShop", there are "Диаграммы баз данных", "Таблицы" (with sub-nodes for "Системные таблицы", "FileTables", "Внешние таблицы", "Графовые таблицы", "dbo.Client", "dbo.Full_Name", "dbo.Info_order", "dbo.Model", "dbo.Operation_system", "dbo.Orders", "dbo.Parameters", "dbo.Processor", "dbo.Producer", "dbo.Product", "Представления", "Внешние ресурсы", and "Синонимы").

The main window shows a query results grid titled "SQLQuery19.sql - KUDINOVA\Nastya (52)*". The query is "select * from Model". The results grid has columns: "Code_model", "Name_model", "Code_parameter", and "Code_producer". The data is as follows:

| | Code_model | Name_model | Code_parameter | Code_producer |
|----|------------|--------------|----------------|---------------|
| 1 | 1010 | Redmi 9C | 1111 | 1100 |
| 2 | 2020 | G7 ThinQ | 2222 | 2200 |
| 3 | 3030 | Iphone 12 | 3333 | 3300 |
| 4 | 4040 | Galaxy Flip2 | 4444 | 4400 |
| 5 | 5050 | K12 Pro | 5555 | 5500 |
| 6 | 6060 | 9 Pro | 6666 | 6600 |
| 7 | 7070 | G10 | 7777 | 7700 |
| 8 | 8080 | NetPro 11 | 8888 | 8800 |
| 9 | 9090 | NetPro 11 | 9999 | 9900 |
| 10 | 10101 | Spark 8C | 1212 | 5896 |

9. Заповнимо таблицю «Інформація про замовлення (*Info_order*)» даними:

INSERT INTO Info order

```
VALUES (1, 111, 2, 101, 30000 ),  
       (2, 222, 4, 202, 64000 ),  
       (3, 333, 1, 303, 27000 ),  
       (4, 444, 7, 404, 266000),  
       (5, 555, 3, 505, 27000 ),  
       (6, 666, 5, 606, 55000 ),
```

```
(7, 777, 10, 707, 85000 ),
(8, 888, 16, 808, 179200 ),
(9, 999, 45, 909, 45000),
(10, 1000, 25, 1011, 725000)
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer displays the database structure under 'KUDINOVA\MySQLServer'. In the center, the 'SQLQuery22.sql' window contains the following SQL code:

```
INSERT INTO Info_order
VALUES (1, 111, 2, 101, 30000 ),
       (2, 222, 4, 202, 64000 ),
       (3, 333, 1, 303, 27000 ),
       (4, 444, 7, 404, 266000),
       (5, 555, 3, 505, 27000 ),
       (6, 666, 5, 606, 55000 ),
       (7, 777, 10, 707, 85000 ),
       (8, 888, 16, 808, 179200 ),
       (9, 999, 45, 909, 45000),
       (10, 1000, 25, 1011, 725000)
```

Below the code, the status bar shows: 'затронуто строк: 10' (10 rows affected) and 'Время выполнения: 2022-12-08T22:21:02.6093534+02:00'.

Перевіримо результат:

```
select * from Info_order
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer displays the database structure under 'KUDINOVA\MySQLServer'. In the center, the 'SQLQuery23.sql' window contains the following SQL code:

```
select * from Info_order
```

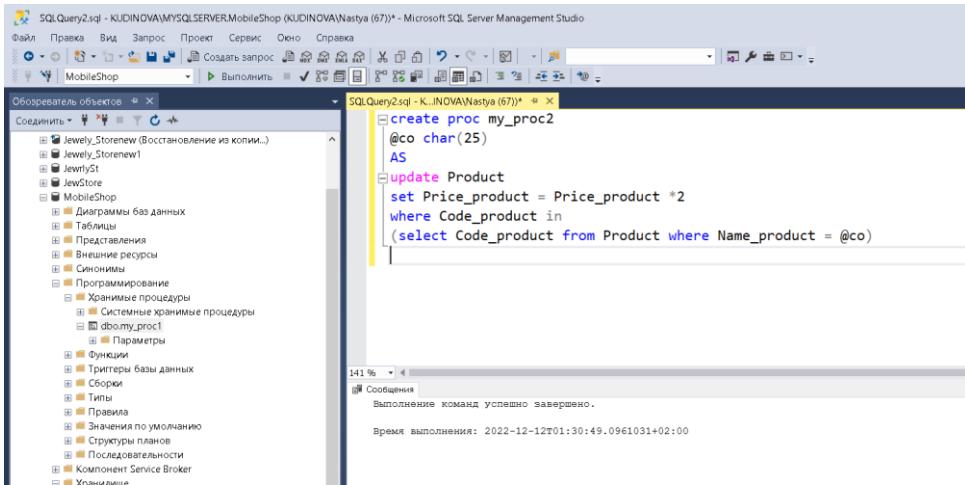
Below the code, the 'Результаты' (Results) tab shows the following data:

| | Code_info | Code_product | Number | Code_order | Price_order |
|----|-----------|--------------|--------|------------|-------------|
| 1 | 111 | 2 | 101 | 30000 | |
| 2 | 222 | 4 | 202 | 64000 | |
| 3 | 333 | 1 | 303 | 27000 | |
| 4 | 444 | 7 | 404 | 266000 | |
| 5 | 555 | 3 | 505 | 27000 | |
| 6 | 666 | 5 | 606 | 55000 | |
| 7 | 777 | 10 | 707 | 85000 | |
| 8 | 888 | 16 | 808 | 179200 | |
| 9 | 999 | 45 | 909 | 45000 | |
| 10 | 1000 | 25 | 1011 | 725000 | |

Додаток Б. Створення збережених процедур

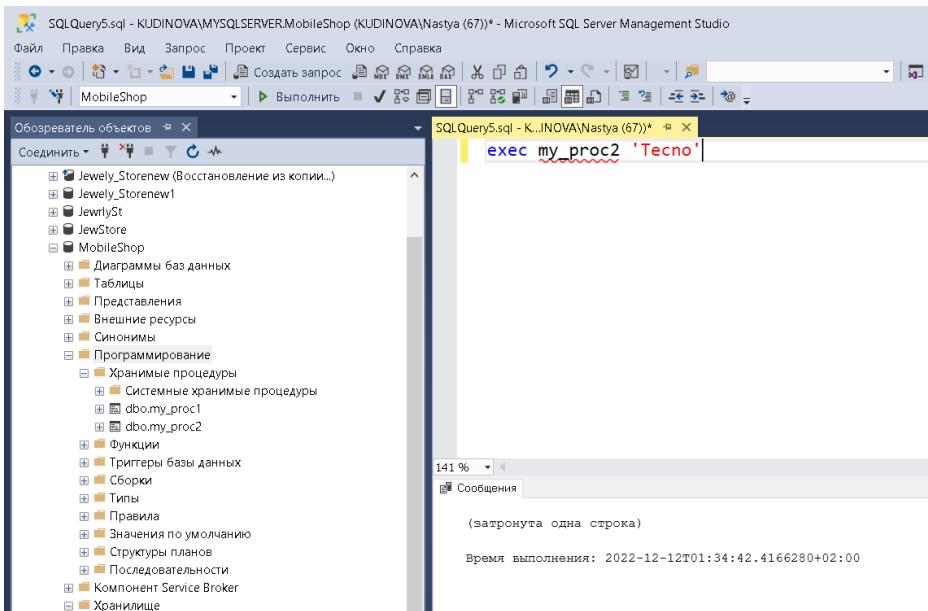
- Створення збереженої процедури, яка збільшить ціну обраного телефону вдвічі

```
create proc my_proc2
@co char(25)
AS
update Product
set Price_product = Price_product *2
where Code_product in
(select Code_product from Product where Name_product = @co)
```



Переглянемо результат на збільшення ціни на телефон Тесно

```
exec my_proc2 'Тесно'
```



`select * from Product`

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays a tree view of database objects under the 'MobileShop' database, including tables, stored procedures, and functions. In the center, the 'SQLQuery6.sql' query window contains the SQL command `select * from Product`. To the right, the 'Результаты' (Results) tab shows the output of the query as a table:

| | Code_product | Code_model | Price_product | All_in_stock | Name_product |
|----|--------------|------------|---------------|--------------|--------------|
| 1 | 111 | 1010 | 7500 | 4 | Xiaomi |
| 2 | 222 | 2020 | 16000 | 25 | LG |
| 3 | 333 | 3030 | 27000 | 32 | Apple |
| 4 | 444 | 4040 | 38000 | 11 | Samsung |
| 5 | 555 | 5050 | 9000 | 7 | Lenovo |
| 6 | 666 | 6060 | 11000 | 9 | Realme |
| 7 | 777 | 7070 | 8500 | 10 | Nokia |
| 8 | 888 | 8080 | 11200 | 18 | Meizu |
| 9 | 999 | 9090 | 10000 | 60 | Motorola |
| 10 | 1000 | 10101 | 58000 | 30 | Tecno |

- Створення збереженої процедури, яка дозволяє отримати назву моделі продукта, яка більша за задану суму

```
create proc my_proc3
@pr int
AS
select Model.Code_model, Model.Name_model
from Model join Product
on Model.Code_model = Product.Code_model and Product.Price_product > @pr
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays a tree view of database objects under the 'MobileShop' database. In the center, the 'SQLQuery7.sql' query window contains the SQL command for creating a stored procedure:

```
create proc my_proc3
@pr int
AS
select Model.Code_model, Model.Name_model
from Model join Product
on Model.Code_model = Product.Code_model and Product.Price_product > @pr
```

To the right, the 'Сообщения' (Messages) tab shows the message: 'Выполнение команд успешно завершено.' (The command execution was successfully completed). Below it, the message 'Время выполнения: 2022-12-12T01:59:09.5748461+02:00' is displayed.

Переглянемо результат за заданим параметром більше 15000

```
exec my_proc3 15000
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure of 'MobileShop' with various objects like tables, stored procedures, and functions. In the center, the 'SQLQuery8.sql - KUDINOVA\Nastya (54)*' window contains the command `exec my_proc3 15000`. To the right, the 'Результаты' (Results) pane shows a table with four rows of data:

| Code_model | Name_model |
|------------|------------|
| 1 | 2020 |
| 2 | 3030 |
| 3 | 4040 |
| 4 | 10101 |

3. Створення збереженої процедури, яка дозволить отримати кількість телефонів за номером версії операційної системи

```
create proc my_proc4
AS
select Version_operation_system as 'Операційна система',
Number_Version_operation_system as 'Номер операційної системи', kol as
'Кількість телефонів'
from Operation_system join
(select Id_Operation_system, count(Id_Operation_system) as kol
from Parameters group by Id_Operation_system) sel
on Operation_system.Id_Operation_system = sel.Id_Operation_system
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure of 'MobileShop'. In the center, the '4sql - KUDINOVA\Nastya (51)*' window contains the creation script for the stored procedure:

```
create proc my_proc4
AS
select Version_operation_system as 'Операційна система',
Number_Version_operation_system as 'Номер операційної системи', kol as 'Кількість телефонів'
from Operation_system join
(select Id_Operation_system, count(Id_Operation_system) as kol
from Parameters group by Id_Operation_system) sel
on Operation_system.Id_Operation_system = sel.Id_Operation_system
```

In the bottom right corner of the results pane, a message indicates: 'Выполнение команд успешно завершено.' (The command execution was successfully completed.)

Переглянемо результат:

```
exec my_proc4
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer displays a tree view of database objects under the 'MobileShop' database, including tables, stored procedures, and functions. In the center, a query window titled 'SQLQuery10.sql' contains the command 'exec my_proc4'. To the right, a results grid titled 'Результаты' (Results) shows a table with data from the procedure execution:

| Операційна система | Номер операційної системи | Кількість телефонів |
|--------------------|---------------------------|---------------------|
| Android | 10 | 1 |
| Android | 8 | 1 |
| iOS | 14 | 1 |
| Android | 13 | 1 |
| Android | 11 | 1 |
| Android | 12 | 1 |
| Android | 19 | 1 |
| Android | 15 | 1 |
| Android | 16 | 1 |
| Android | 18 | 1 |

4. Створення збереженої процедури, для отримання назви виробника та контактних даних, за допомогою параметра

```
create proc my_proc5
@mm char(25)
AS
select Producer.Name_producer, Producer.Contact, Producer.Code_producer,
Model.Name_model
from Producer join Model
on Producer.Code_producer = Model.Code_producer and Model.Name_model = @mm
```

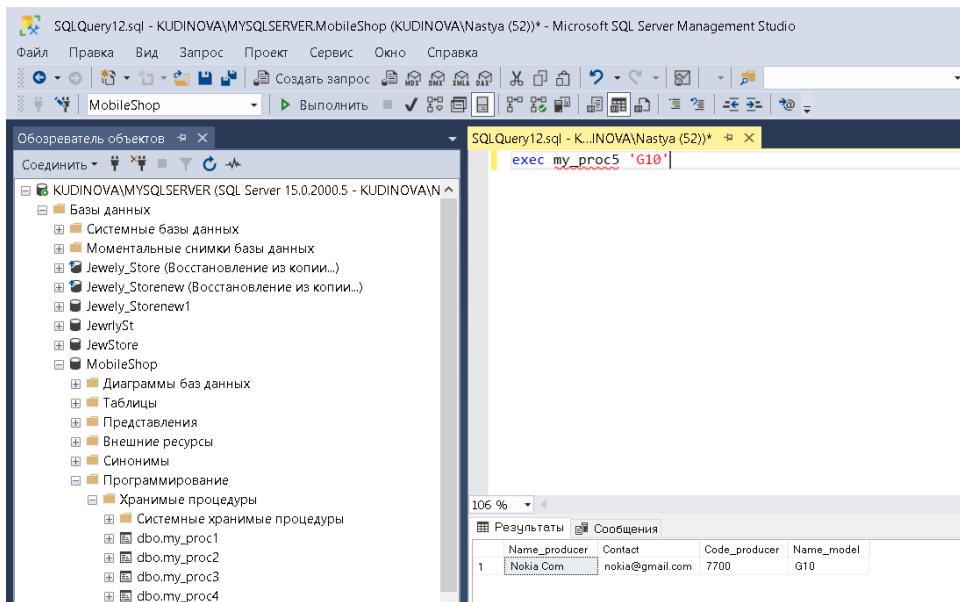
The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer displays a tree view of database objects under the 'KUDINOVA\MySQLServer.MobileShop' database, including tables, stored procedures, and functions. In the center, a query window titled 'SQLQuery11.sql' contains the stored procedure definition:

```
create proc my_proc5
@mm char(25)
AS
select Producer.Name_producer, Producer.Contact, Producer.Code_producer,
Model.Name_model
from Producer join Model
on Producer.Code_producer = Model.Code_producer and Model.Name_model = @mm
```

To the right, a results grid titled 'Сообщения' (Messages) shows the message: 'Выполнение команд успешно завершено.' (Command execution completed successfully.) Below it, the execution time is listed as 'Время выполнения: 2022-12-12T03:00:26.4731403+02:00'.

Переглянемо результат:

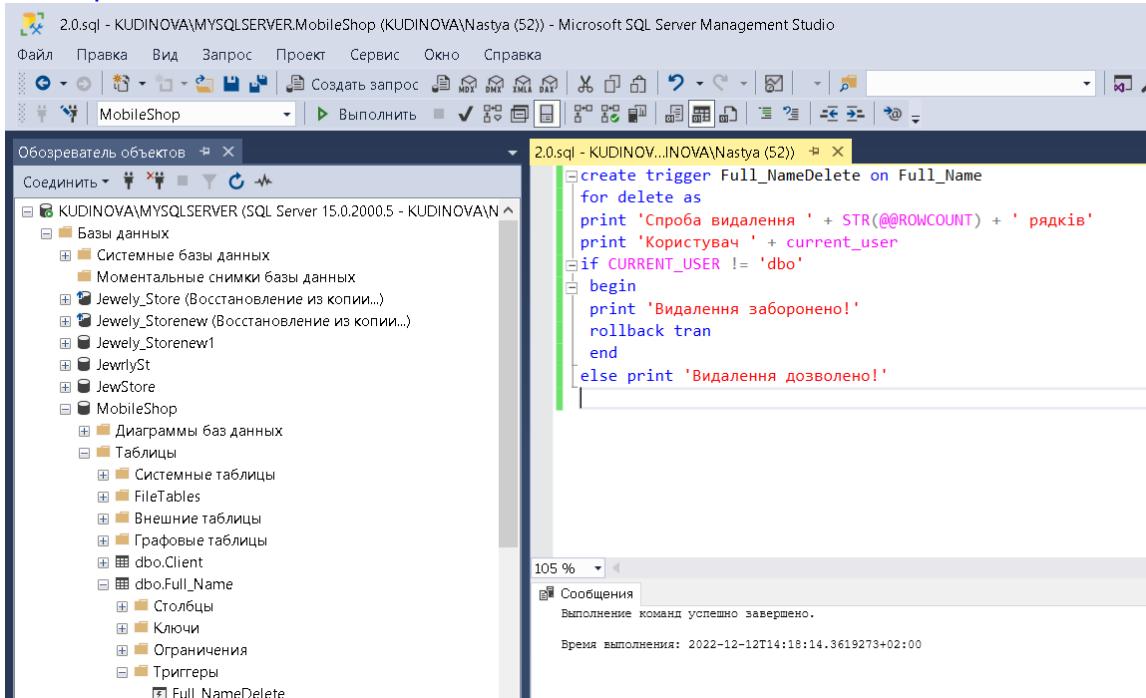
```
exec my_proc5 'G10'
```



Додаток В. Створення тригерів

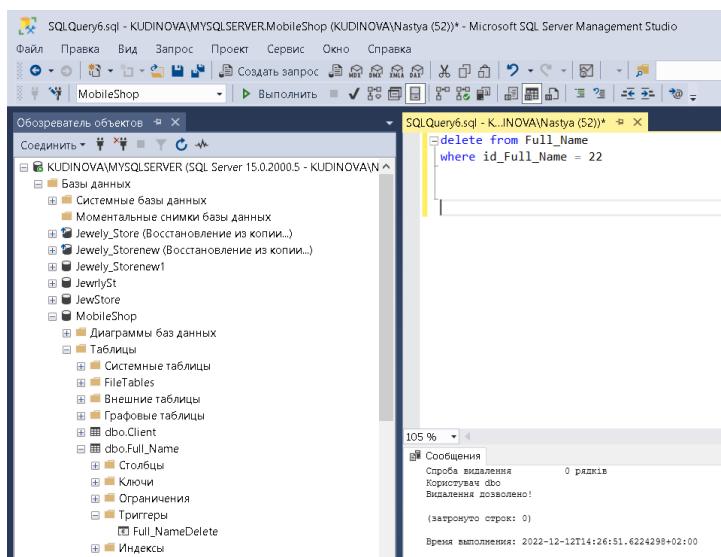
- Створення тригера для таблиці «ПІБ (Full_Name)» типу DELETE, який буде виводити інформацію про спроби видалення та кількість рядків, що підлягають видаленню:

```
create trigger Full_NameDelete on Full_Name
for delete as
print 'Спроба видалення ' + STR(@@ROWCOUNT) + ' рядків'
print 'Користувач ' + current_user
if CURRENT_USER != 'dbo'
begin
print 'Видалення заборонено!'
rollback tran
end
else print 'Видалення дозволено!'
```



Переглянемо результат роботи тригера Full_NameDelete:

```
delete from Full_Name
where id_Full_Name = 22
```



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer on the left, under the 'MobileShop' database, there is a table named 'dbo.Full_Name'. In the center, a query window titled 'SQLQuery8.sql' displays the following SQL command:

```
select * from Full_Name
```

The results grid shows 10 rows of data:

| | id_Full_Name | First_Name | Second_Name | Patronymic |
|----|--------------|--------------|-------------|---------------|
| 1 | 11 | Пашковський | Максим | Ігорович |
| 2 | 12 | Драбовський | Анатолій | Григорович |
| 3 | 13 | Лічинан | Ганна | Василівна |
| 4 | 14 | Грищенко | Тетяна | Миколаївна |
| 5 | 15 | Савчук | Оксана | Василівна |
| 6 | 16 | Киселевський | Дмитро | Давидович |
| 7 | 17 | Васинчук | Іван | Олександрович |
| 8 | 18 | Карпенко | Мартин | Генадійович |
| 9 | 19 | Іващенко | Вікторія | Сергіївна |
| 10 | 21 | Козаченко | Альона | Вікторівна |

- Створення тригера, який запускається при модифікації таблиці Тригер повинен заносити до журналу (допоміжну таблицю) інформацію про час модифікації та імені користувача, що провадив модифікацію таблиці та інформація, що модифікувалася.

Для цього нам необхідно створити в базі даних додаткову таблицю, в якій буде зберігатися список рядків, що видаляю. Виконаємо це за допомогою наступної команди:

```
select top 0 *, suser = suser_sid(), datetime=GetDate()
into delatting_full_name
from Full_Name
```

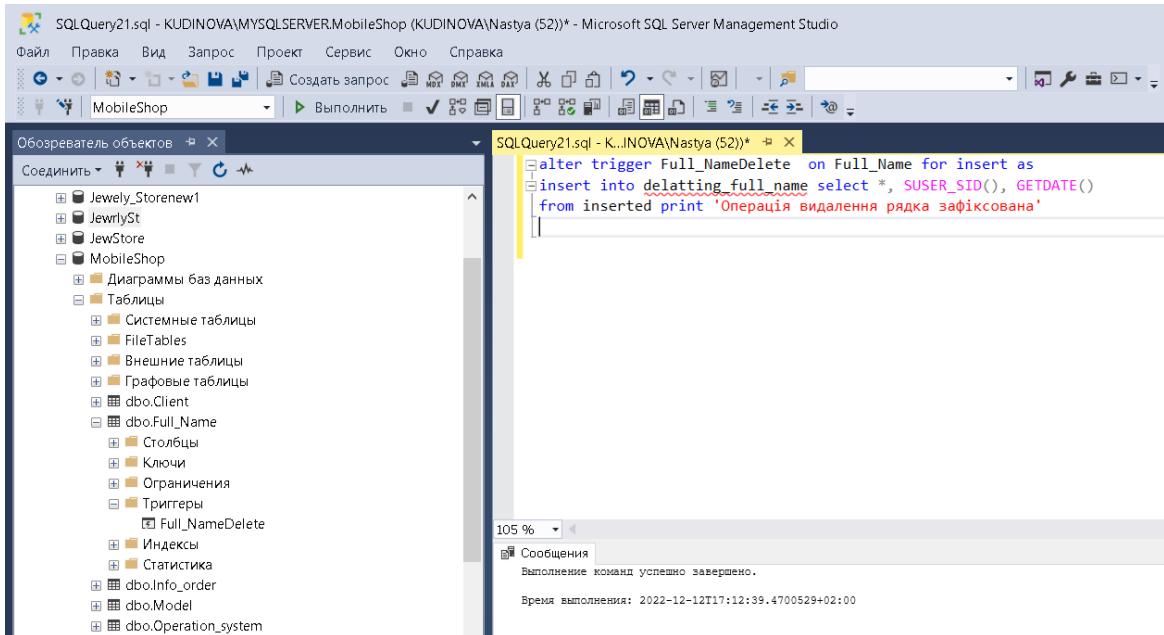
The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer on the left, under the 'MobileShop' database, there is a table named 'dbo.Full_Name'. In the center, a query window titled 'SQLQuery21.sql' displays the following SQL command:

```
select top 0 *, suser = suser_sid(), datetime=GetDate()
into delatting_full_name
from Full_Name
```

The results grid shows the message '(запромото строк: 0)' (0 rows promoted).

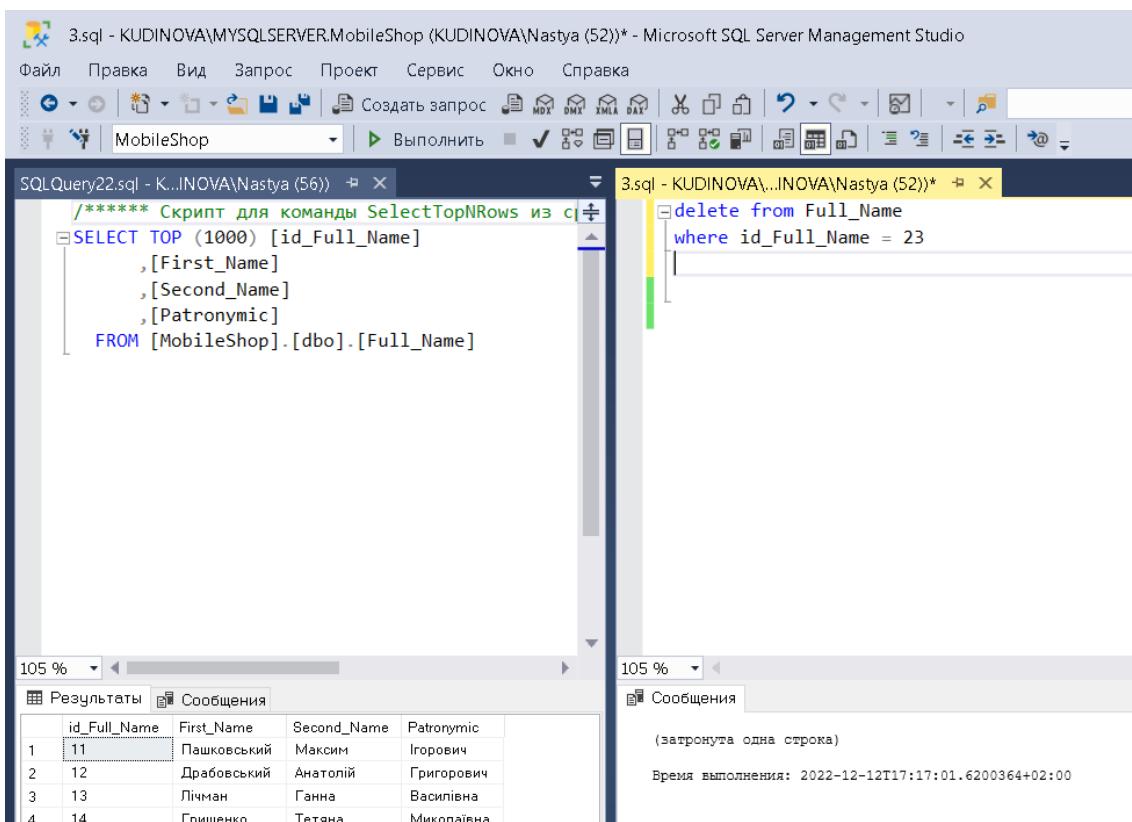
Тепер змінимо тригер:

```
alter trigger Full_NameDelete on Full_Name for insert as
insert into delatting_full_name select *, SUSER_SID(), GETDATE()
from inserted print 'Операція видалення рядка зафіксована'
```



І виконаємо запит:

```
delete from Full_Name
where id_Full_Name = 23
```



Перевірка працездатності тригера:

```
select * from delatting_full_name
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left is the Object Explorer pane, which lists various database objects like Системные таблицы, FileTables, Внешние таблицы, Графовые таблицы, dbo.Client, dbo.Full_Name, dbo.Info_order, dbo.Model, dbo.Operation_system, dbo.Orders, dbo.Parameterss, dbo.Processor, dbo.Producer, dbo.Product, Представления, Внешние ресурсы, Синонимы, Программирование, Компонент Service Broker, Хранилище, Безопасность, Пользователи, and Роли. The main window displays a query results grid for the 'delatting_full_name' table. The grid has columns: id_Full_Name, First_Name, Second_Name, Patronymic, suser, datetime, and a few other columns that are mostly cut off. There is one row with values: id_Full_Name = 23, First_Name = Пономаренко, Second_Name = Віталій, Patronymic = Олегович, suser = 0x0105000000000005150000005F65888B8EDE88BEA90D8A..., and datetime = 2022-12-12 17:16:19.730.

3. Створити тригер, який буде запускається при занесенні нового рядка в таблицю та показувати скільки рядків було додано

```
create trigger add_op on Operation_system
for insert as
print 'Спроба додавання ' + str(@@rowcount) + ' рядків у таблиці
Oparation_system'
print 'Користувач ' + current_user
if CURRENT_USER != 'dbo'
begin
print 'Додавання заборонено!' rollback tran
end else
print 'Додавання дозволено!'
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The main window displays the SQL code for creating a trigger:

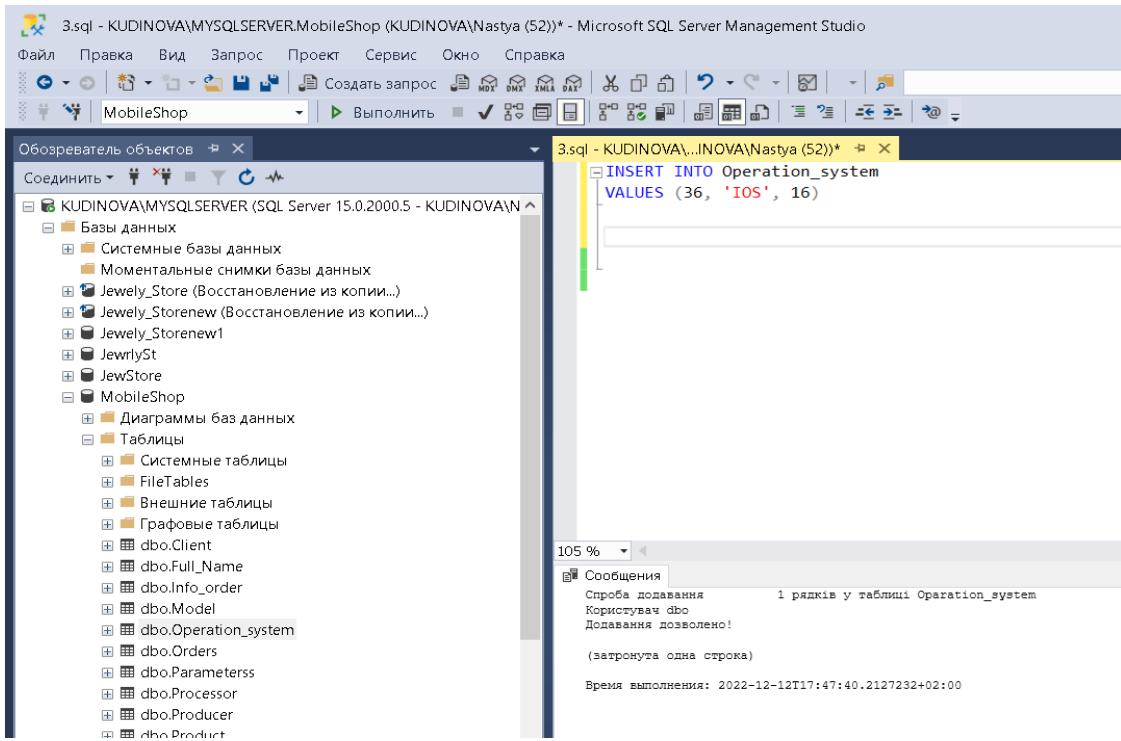
```
create trigger add_op on Operation_system
for insert as
print 'Спроба додавання ' + str(@@rowcount) + ' рядків у таблиці Oparation_system'
print 'Користувач ' + current_user
if CURRENT_USER != 'dbo'
begin
print 'Додавання заборонено!' rollback tran
end else
print 'Додавання дозволено!'
```

Below the code, a message box shows the status: 'Выполнение команды успешно завершено.' (The command execution was successfully completed.) and the execution time: 'Время выполнения: 2022-12-12T17:38:04.0101012+02:00' (Execution time: 2022-12-12T17:38:04.0101012+02:00).

Виконаємо запит:

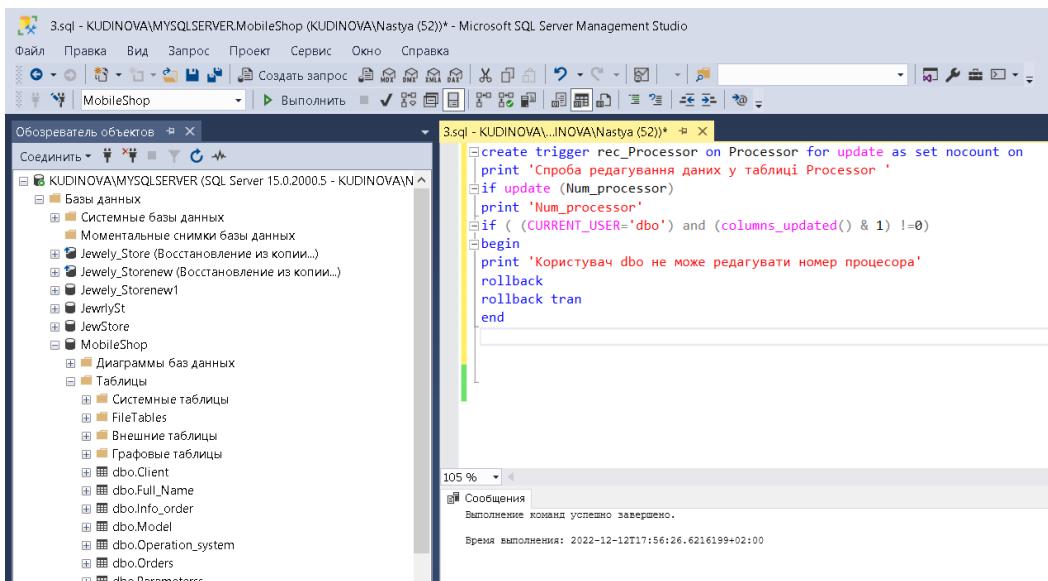
```
INSERT INTO Operation_system
```

VALUES (36, 'IOS', 16)



4. Створимо тригер, який буде дозволяти зміну стовпця всім, крім 'dbo':

```
create trigger rec_Processor on Processor for update as set nocount on
print 'Спроба редагування даних у таблиці Processor '
if update (Num_processor)
print 'Num_processor'
if ( (CURRENT_USER='dbo') and (columns_updated() & 1) !=0)
begin
print 'Користувач dbo не може редагувати номер процесора'
rollback
rollback tran
end
```



Спробуємо виконати зміну номера процесору:

```
update Processor  
set Num_processor =954  
where id_Processor = 41
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer displays the database structure under 'KUDINOVA\MySQLServer'. The 'MobileShop' database is expanded, showing tables like 'Client', 'Full_Name', 'Info_order', 'Model', 'Operation_system', 'Orders', 'Parameters', and 'Processor'. On the right, the 'SQLQuery25.sql - KUDINOVA\Nastya (52)*' window contains the following SQL code:

```
update Processor  
set Num_processor =954  
where id_Processor = 41
```

Below the code, the 'Messages' pane shows the execution results:

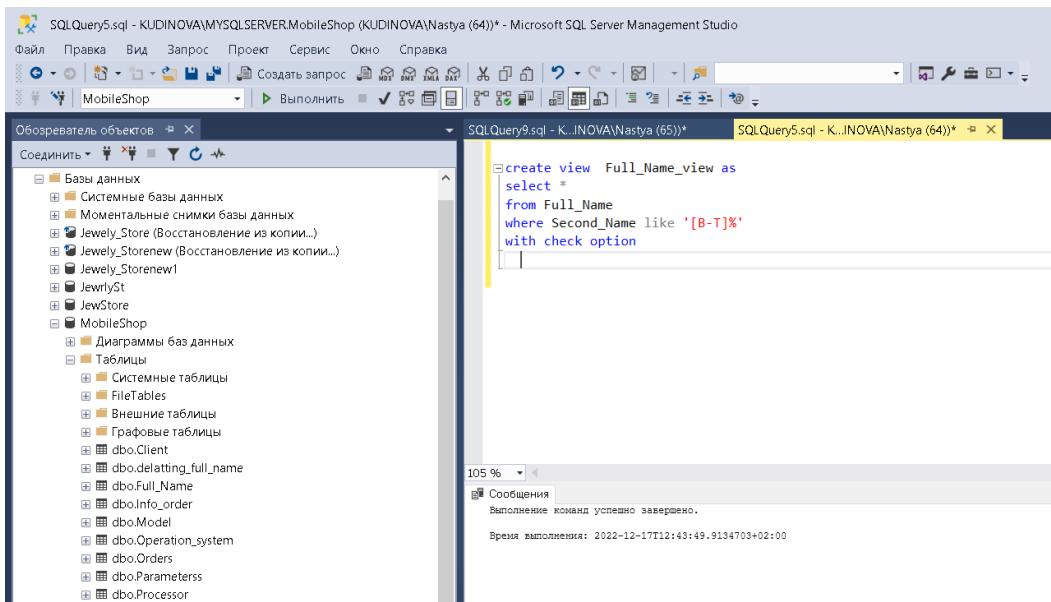
Строка редактирования данных у таблицы Processor
Num_processor
(затронута одна строка)

Время выполнения: 2022-12-12T18:01:07.2599083+02:00

Додаток Г. Створення представлення

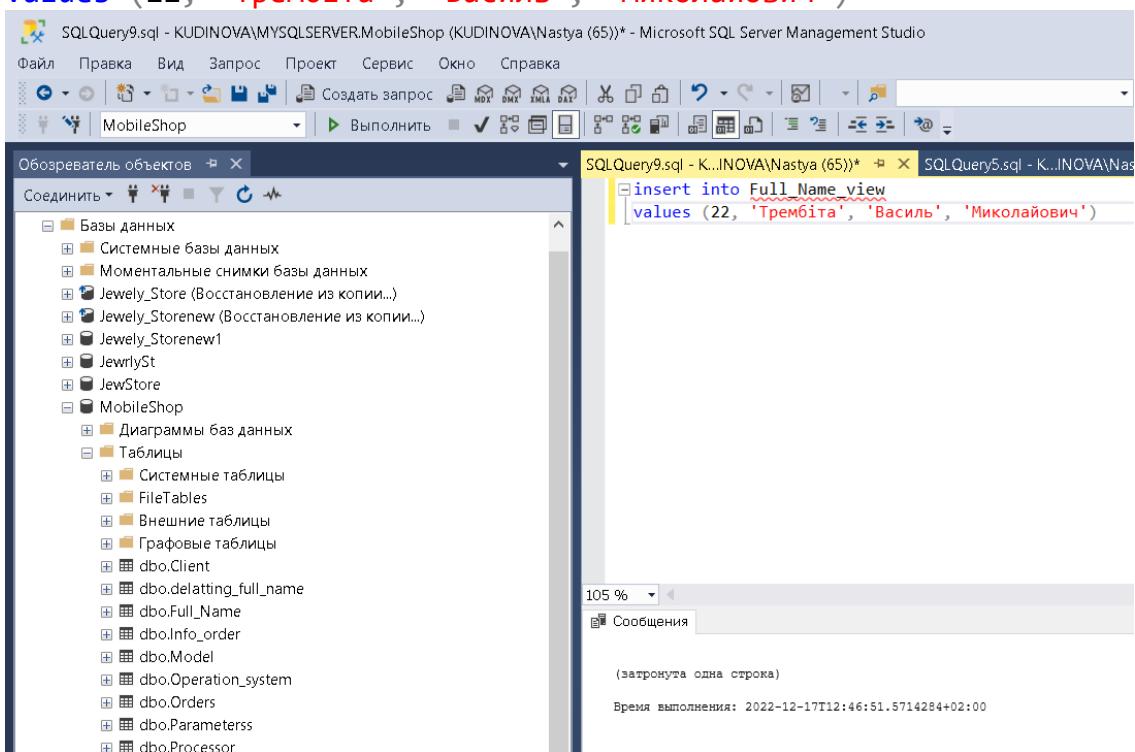
1. Створимо запит, що створює оновлюване представлення на базі таблиці *Full_Name* по всіх її полям, але тільки за умови, що ім'я (*Second_Name*) буде знаходитись в таких межах «В-Т»

```
create view Full_Name_view as
select *
from Full_Name
where Second_Name like '[B-T]%'
with check option
```



Після цього до уявлення можна додати новий запис:

```
insert into Full_Name_view
values (22, 'Трембита', 'Василь', 'Миколайович')
```



Дивитись результат:

```
select * from Full_Name_view
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure of 'MobileShop'. In the center, the Results pane shows the output of the query 'select * from Full_Name_view'. The results are presented in a table with columns: id_Full_Name, First_Name, Second_Name, and Patronymic. The data is as follows:

| | id_Full_Name | First_Name | Second_Name | Patronymic |
|---|--------------|--------------|-------------|-------------|
| 1 | 11 | Пашковський | Максим | Ігорович |
| 2 | 13 | Лічман | Ганна | Василівна |
| 3 | 14 | Грищенко | Тетяна | Миколаївна |
| 4 | 15 | Савчук | Оксана | Василівна |
| 5 | 16 | Киселевський | Дмитро | Давидович |
| 6 | 17 | Васинчук | Іван | Олексійович |
| 7 | 18 | Карпенко | Мартін | Генадійович |
| 8 | 19 | Іващенко | Вікторія | Сергіївна |
| 9 | 22 | Трембіта | Василь | Миколайович |

2. Створити представлення на основі даних з таблиць: Info_order, Orders, Client, Full_Name

```
create view info_order_view as
select
Info_order.Number, Info_order.Price_order,
Orders.Datas,
Client.Phone,
Full_Name.First_Name, Full_Name.Second_Name, Full_Name.Patronymic
from Info_order, Orders, Client, Full_Name
where Info_order.Code_order = Orders.Code_order
AND Orders.Code_client = Client.Code_client
AND Client.id_Full_Name = Full_Name.id_Full_Name
```

```

SQLQuery15.sql - KUDINOVA\MySQLServer.MobileShop (KUDINOVA\Nastya (51)) - Microsoft SQL Server Management Studio
Файл Правка Вид Запрос Проект Сервис Окно Справка
MobileShop Выполнить
Обозреватель объектов
Соединить
create view info_order_view as
select
    Info_order.Number, Info_order.Price_order,
    Orders.Data,
    Client.Phone,
    Full_Name.First_Name, Full_Name.Second_Name, Full_Name.Patronymic
from Info_order, Orders, Client, Full_Name
where Info_order.Code_order = Orders.Code_order
AND Orders.Code_client = Client.Code_client
AND Client.id_Full_Name = Full_Name.id_Full_Name

```

Сообщения
Выполнение команд успешно завершено.
Время выполнения: 2022-12-17T13:05:59.6769187+02:00

Перевіримо запит :

```
select * from Info_order_view
```

```

SQLQuery18.sql - KUDINOVA\MySQLServer.MobileShop (KUDINOVA\Nastya (57)) - Microsoft SQL Server Management Studio
Файл Правка Вид Запрос Проект Сервис Окно Справка
MobileShop Выполнить
Обозреватель объектов
Соединить
select * from Info_order_view

```

Результаты Сообщения

| Number | Price_order | Data | Phone | First_Name | Second_Name | Patronymic |
|--------|-------------|--------|------------|------------|--------------|---------------------|
| 1 | 2 | 30000 | 2022-11-28 | 9254 | Пашковський | Максим Ігорович |
| 2 | 4 | 64000 | 2022-11-29 | 7458 | Драбовський | Анатолій Григорович |
| 3 | 1 | 27000 | 2022-09-02 | 3657 | Лічман | Ганна Василівна |
| 4 | 7 | 266000 | 2022-12-08 | 4597 | Грищенко | Тетяна Миколаївна |
| 5 | 3 | 27000 | 2022-12-01 | 2145 | Савчук | Оксана Василівна |
| 6 | 5 | 55000 | 2022-10-21 | 3001 | Киселевський | Дмитро Давидович |
| 7 | 10 | 85000 | 2022-09-22 | 8222 | Васинчук | Іван Олексійович |
| 8 | 16 | 179200 | 2022-04-05 | 2654 | Карпенко | Мартін Генадійович |
| 9 | 45 | 45000 | 2022-11-17 | 5501 | Іващенко | Вікторія Сергіївна |
| 10 | 25 | 725000 | 2022-07-10 | 6324 | Козаченко | Альона Вікторівна |

Додаток Г. Проста вибірка даних

- Написати запит, який би видав список продуктів із значеннями їх ціни, кількості в наявності та назви.

```
select Price_product as 'Ціна продукту',
All_in_stock as 'Кількість в наявності',
Name_product as 'Назва продукту'
from Product
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer displays the database structure under 'KUDINOVA\MySQLServer'. On the right, the 'SQLQuery28.sql' window contains the following query:

```
select Price_product as 'Ціна продукту',
All_in_stock as 'Кількість в наявності',
Name_product as 'Назва продукту'
from Product
```

The results window below shows the output of the query:

| | Ціна продукту | Кількість в наявності | Назва продукту |
|----|---------------|-----------------------|----------------|
| 1 | 7500 | 4 | Xiaomi |
| 2 | 16000 | 25 | LG |
| 3 | 27000 | 32 | Apple |
| 4 | 38000 | 11 | Samsung |
| 5 | 9000 | 7 | Lenovo |
| 6 | 11000 | 9 | Realme |
| 7 | 8500 | 10 | Nokia |
| 8 | 11200 | 18 | Meizu |
| 9 | 10000 | 60 | Motorola |
| 10 | 58000 | 30 | Tesla |

- Написати запит, який би видав інформацію про ПІБ (Прізвище, Ім'я, По батькові) та упорядкувати дані у алафівітному порядку

```
select First_Name as 'Прізвище',
Second_Name as 'Ім'я',
Patronymic as 'Побатькові'
from Full_Name
order by First_Name
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer displays the database structure under 'KUDINOVA\MySQLServer'. On the right, the 'SQLQuery30.sql' window contains the following query:

```
select First_Name as 'Прізвище',
Second_Name as 'Ім'я',
Patronymic as 'Побатькові'
from Full_Name
order by First_Name
```

The results window below shows the output of the query:

| | Прізвище | Ім'я | Побатькові |
|----|--------------|----------|--------------|
| 1 | Васинчик | Іван | Олексійович |
| 2 | Грищенко | Тетяна | Миколаївна |
| 3 | Драбовський | Анатолій | Григорович |
| 4 | Івашенко | Вікторія | Сергіївна |
| 5 | Карпенко | Мартін | Геннадійович |
| 6 | Кирилівський | Дмитро | Дмитрович |
| 7 | Козаченко | Альона | Вікторіна |
| 8 | Лінчан | Ганна | Василівна |
| 9 | Пашковський | Максим | Ігорович |
| 10 | Салюк | Оксана | Василівна |

3. Написати запит, який би видав список назв моделей, у яких «Унікальний код моделей» знаходиться в діапазоні від 2020 до 7070. Відсортувати запит за назвою моделі.

```
select Model.Code_model as 'Унікальний код моделі',
Model.Name_model as 'Назва моделі'
from Model
where (Model.Code_model >= 2020) and (Model.Code_model <=7070)
order by Model.Name_Model
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer displays the database structure under 'KUDINOVA\MySQLServer'. In the center, the 'SQLQuery33.sql' query window contains the SQL code. Below it, the 'Results' tab shows the execution progress at 105% and the final result set:

| | Унікальний код моделі | Назва моделі |
|---|-----------------------|--------------|
| 1 | 6060 | 9 Pro |
| 2 | 7070 | G10 |
| 3 | 2020 | G7 ThinQ |
| 4 | 4040 | Galaxy Flip4 |
| 5 | 3030 | Iphone 12 |
| 6 | 5050 | K12 Pro |

Додаток Д. Вибірка обчислюваних значень

1. Написати запит, який би визначив скільки різних типів версій процесорів мобільних телефонів

```
select Processor.Ver_processor as 'Версія процесору',
COUNT(Processor.Ver_processor) as 'Кількість телефонів'
from Processor
group by Processor.Ver_processor
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure under 'KUDINOVA\MySQLServer.MobileShop'. In the center, the 'SQLQuery38.sql' query window contains the SQL code provided above. Below it, the 'Results' pane shows the execution output:

| Версія процесору | Кількість телефонів |
|------------------|---------------------|
| Bionic | 1 |
| Snapdragon | 9 |

2. Написати запит, який би визначив кількість клієнтів

```
Select COUNT (Client.Code_client) as 'Кількість клієнтів'
from Client
```

3. Написати запит, який би визначив скільки різних типів версій операційних систем мобільних телефонів

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure under 'KUDINOVA\MySQLServer.MobileShop'. In the center, the 'SQLQuery39.sql' query window contains the SQL code provided above. Below it, the 'Results' pane shows the execution output:

| Кількість клієнтів |
|--------------------|
| 10 |

```

select Operation_system.Version_operation_system as 'Версія операційної системи',
COUNT(Operation_system.Version_operation_system) as 'Кількість телефонів'
from Operation_system group by Operation_system.Version_operation_system

```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer displays the database structure under 'KUDINOVA\MySQLServer'. The 'MobileShop' database is selected, showing its tables: 'Client', 'FileTables', 'Model', 'Operation_system', and 'Orders'. In the center, the 'SQLQuery37.sql' window contains the provided SQL query. Below it, the 'Results' window shows the execution output:

| | Версія операційної системи | Кількість телефонів |
|---|----------------------------|---------------------|
| 1 | Android | 9 |
| 2 | iOS | 2 |

Додаток Е. Вибірка значень з діапазону

- Написати запит, який буде видавати параметри про камеру, пам'ять, код параметрів за визначенім діапазону параметру батареї від 6000 до 8000

```
select Code_parameter AS 'Код параметру',
Memory_Gb AS 'Память',
Camera_Mp AS 'Камера',
Battery_Ma AS 'Батарея'
from Parameterss
WHERE Battery_Ma BETWEEN 6000 AND 8000
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer window displays the database structure under 'KUDINOVA\MySQLServer'. In the center, the 'SQLQuery42.sql' query window contains the provided T-SQL code. Below it, the 'Results' window shows the output of the query:

| Код параметру | Память | Камера | Батарея |
|---------------|--------|--------|---------|
| 3333 | 128 | 12 | 6400 |
| 5555 | 16 | 64 | 6000 |
| 7777 | 11 | 35 | 7000 |
| 8888 | 714 | 30 | 8000 |

- Написати запит, який буде видавати назви продуктів, ціна яких знаходиться в діапазоні від 10000 до 45000

```
select Price_product AS 'Ціна продукту',
All_in_stock AS 'В наявності',
Name_product AS 'Назва продукту'
from Product
WHERE Price_product BETWEEN 10000 AND 45000
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer window displays the database structure under 'MobileShop'. In the center, the 'SQLQuery45.sql' query window contains the provided T-SQL code. Below it, the 'Results' window shows the output of the query:

| Ціна продукту | В наявності | Назва продукту |
|---------------|-------------|----------------|
| 16000 | 25 | LG |
| 27000 | 32 | Apple |
| 38000 | 11 | Samsung |
| 11000 | 9 | Realme |
| 11200 | 18 | Meizu |
| 10000 | 60 | Motorola |

Додаток Є. Вибірка з використанням шаблонів

- За допомогою наступного запиту може бути отриманий список виробників назва, яких починається на літері в діапазоні ‘А’ – ‘Л’

```
select *
from Producer
WHERE Name_producer like '[A-L]%'
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure under 'KUDINOVA\MySQLServer'. In the center, the 'SQLQuery48.sql' window contains the following T-SQL code:

```
select *
from Producer
WHERE Name_producer like '[A-L]%'
```

On the right, the 'Results' tab shows the execution output:

| Code_producer | Name_producer | Contact |
|---------------|----------------------|-----------------------------|
| 2200 | Lg electronics | lgroup@gmail.com |
| 3300 | Apple Company | appl@gmail.com |
| 5500 | Lenovo Group Limited | lenovolimited1234@gmail.com |

- За допомогою наступного запиту може бути номер клієнта, у якого код 30

```
select Code_client AS 'Код', Phone AS 'Номер телефону'
from Client
WHERE Code_client in (30)
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure under 'KUDINOVA\MySQLServer'. In the center, the 'SQLQuery48.sql' window contains the following T-SQL code:

```
select Code_client AS 'Код', Phone AS 'Номер телефону'
from Client
WHERE Code_client in (30)
```

On the right, the 'Results' tab shows the execution output:

| Код | Номер телефону |
|-----|----------------|
| 30 | 3657 |

3. Написати запит, який би видав всі дані про продукти чиї імена починаються з літери «М»

```
select *
from Product
where Product.Name_product like 'M%'
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure under 'KUDINOVA\MySQLServer'. It includes a tree view of databases (Bases de datos), tables (Tablas), and system tables (Tablas de sistema) for the 'MobileShop' database. In the center, the 'SQLQuery53.sql - KUDINOVA\Nastya (51)*' window contains the SQL query: 'select * from Product where Product.Name_product like 'M%''. To the right, the 'Results' tab shows the execution results in a grid:

| | Code_product | Code_model | Price_product | All_in_stock | Name_product |
|---|--------------|------------|---------------|--------------|--------------|
| 1 | 888 | 8080 | 11200 | 18 | Meizu |
| 2 | 999 | 9090 | 10000 | 60 | Motorola |

Додаток Ж. Угруповання даних при організації запитів

1. Написати запит, який би визначив скільки моделей виготовляє кожен виробник та угруппувати ці данні по полям «Назва виробника», «Контакти»

```
select Producer.Name_producer as 'Назва виробника',
Producer.Contact as 'Контакти',
COUNT (Model.Code_model) as 'Кількість'
from Producer, Model, Product
where Product.Code_model = Model.Code_model
AND Model.Code_producer = Producer.Code_producer
group by Producer.Name_producer, Producer.Contact
order by Producer.Name_producer, Producer.Contact
```

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure under 'KUDINOVA\MySQLServer'. It includes sections for 'Базы данных' (Databases), 'Таблицы' (Tables), and 'Процедуры' (Procedures). The 'MobileShop' database is selected. On the right, the 'SQLQuery55.sql - KUDINOVA\Nastya (61)' tab contains the query provided above. Below it, the '14.sql - KUDINOVA\Nastya (61)' tab also contains the same query. At the bottom, the 'Результаты' (Results) pane shows a table with 10 rows of data, mapping each producer to their contact email and the count of models produced.

| Назва виробника | Контакти | Кількість |
|----------------------|-----------------------------|-----------|
| Apple Company | appl@gmail.com | 1 |
| Lenovo Group Limited | lenovolimited1234@gmail.com | 1 |
| Lg electronics | lggroup@gmail.com | 1 |
| Meizu LL | meizu@gmail.ll | 1 |
| Motoln Inc | motorola@gmail.com | 1 |
| Nokia Com | nokia@gmail.com | 1 |
| Realmi Inc | realmelc@gmail.com | 1 |
| Samsung Group | samsunggroup741@gmail.com | 1 |
| TechoMix | tecno@gmail.com | 1 |
| Xiaomi Corporation | xiaomiinc24558@gmail.com | 1 |

2. Написати запит, який би визначив скільки версій процесорів купив кожен клієнт угруппувати ці данні по полям «прізвище», «ім’я», «по батькові», «версія процесору»

```
select Full_Name.First_Name as 'Прізвище',
Full_Name.Second_Name as 'Ім'я',
Full_Name.Patronymic as 'По батькові',
Processor.Ver_processor as 'Версія процесора',
COUNT (Processor.id_Processor ) as 'Кількість куплених процесорів клієнтом'
from Processor,Full_Name, Parameterss, Model,Product, Orders, Info_order,
Client
Where Full_Name.id_Full_Name= Client.id_Full_Name
```

```

AND Client.Code_client = Orders.Code_client
AND Orders.Code_order = Info_order.Code_order
AND Info_order.Code_product = Product.Code_product
AND Product.Code_model = Model.Code_model
AND Model.Code_parameter= Parameterss.Code_parameter
AND Parameterss.Id_Processor = Processor.id_Processor
Group by Full_Name.First_Name,
Full_Name.Second_Name,
Full_Name.Patronymic,
Processor.Ver_processor

```

```

select Full_Name.First_Name as 'Прізвище',
Full_Name.Second_Name as 'Ім'я',
Full_Name.Patronymic as 'Побатькові',
Processor.Ver_processor as 'Версія процесорів',
COUNT (Processor.id_Processor) as 'Кількість куплених процесорів клієнтом'
from Processor,Full_Name,Parameterss,Model,Product,Orders,Info_order,Client
Where Full_Name.id_Full_Name= Client.id_Full_Name
AND Client.Code_client = Orders.Code_client
AND Orders.Code_order = Info_order.Code_order
AND Info_order.Code_product = Product.Code_product
AND Product.Code_model = Model.Code_model
AND Model.Code_parameter= Parameterss.Code_parameter
AND Parameterss.Id_Processor = Processor.id_Processor
Group by Full_Name.First_Name,
Full_Name.Second_Name,
Full_Name.Patronymic,
Processor.Ver_processor

```

| Прізвище | Ім'я | Побатькові | Версія процесора | Кількість куплених процесорів клієнтом |
|--------------|----------|-------------|------------------|--|
| Васинчук | Іван | Олексійович | Snapdragon | 1 |
| Грищенко | Тетяна | Миколаївна | Snapdragon | 1 |
| Драбовський | Анатолій | Григорович | Snapdragon | 1 |
| Іващенко | Вікторія | Сергіївна | Snapdragon | 1 |
| Карпенко | Мартин | Генадійович | Snapdragon | 1 |
| Киселевський | Дмитро | Давидович | Snapdragon | 1 |
| Козаченко | Альона | Вікторівна | Snapdragon | 1 |
| Лічман | Ганна | Василівна | Bionic | 1 |
| Пашковський | Максим | Ігорович | Snapdragon | 1 |
| Савчук | Оксана | Василівна | Snapdragon | 1 |

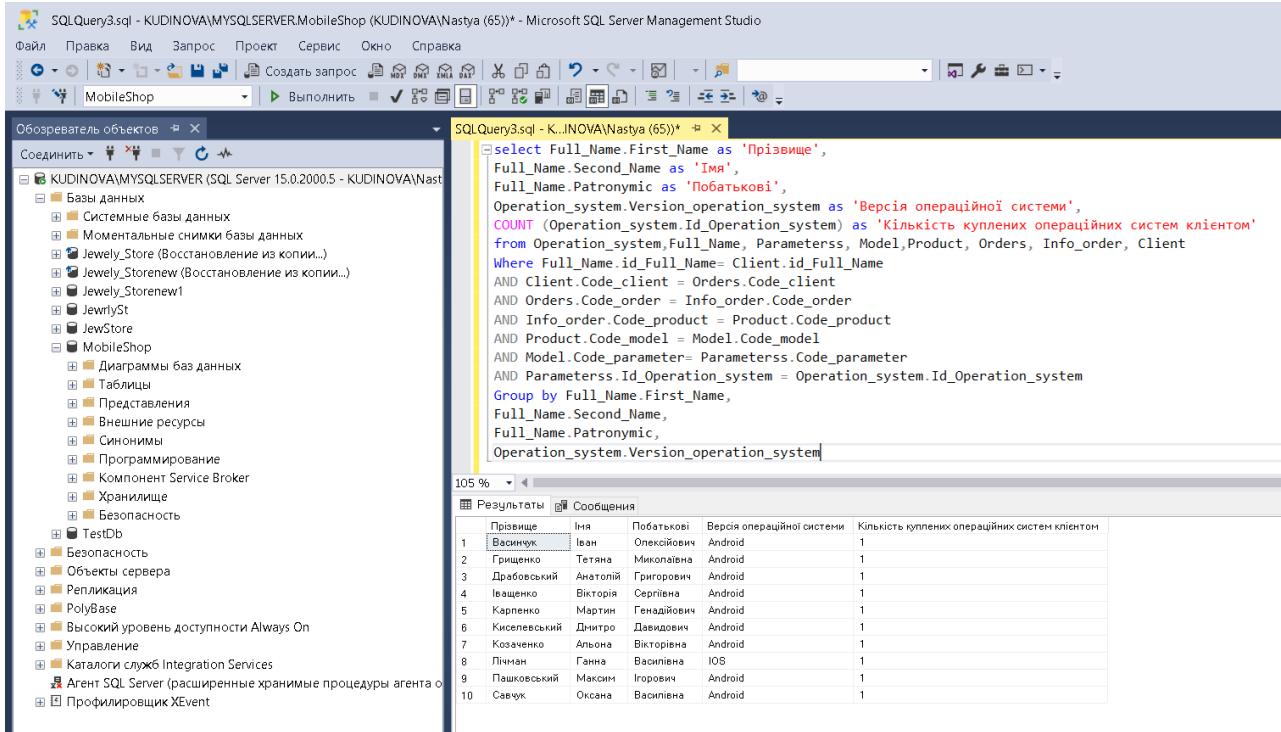
3. Написати запит, який би визначив скільки версій операційної системи купив кожен клієнт угрупувати ці данні по полям «прізвище», «ім'я», «по батькові», «версія операційної системи»

```

select Full_Name.First_Name as 'Прізвище',
Full_Name.Second_Name as 'Ім'я',
Full_Name.Patronymic as 'Побатькові',
Operation_system.Version_operation_system as 'Версія операційної системи',
COUNT (Operation_system.Id_Operation_system) as 'Кількість куплених
операційних систем клієнтом'
from Operation_system,Full_Name, Parameterss, Model,Product, Orders,
Info_order, Client
Where Full_Name.id_Full_Name= Client.id_Full_Name
AND Client.Code_client = Orders.Code_client
AND Orders.Code_order = Info_order.Code_order
AND Info_order.Code_product = Product.Code_product
AND Product.Code_model = Model.Code_model
AND Model.Code_parameter= Parameterss.Code_parameter
AND Parameterss.Id_Operation_system = Operation_system.Id_Operation_system

```

**Group by Full_Name.First_Name,
 Full_Name.Second_Name,
 Full_Name.Patronymic,
 Operation_system.Version_operation_system**



SQLQuery3.sql - KUDINOVA\MySQLServer\MobileShop (KUDINOVA\Nastya (65)) - Microsoft SQL Server Management Studio

Файл Правка Вид Запрос Проект Сервис Окно Справка

Создать запрос МХЛ ОМК ЗМЛД ДАГ Выполнить

Обозреватель объектов SQLQuery3.sql - K...INNOVA\Nastya (65)*

Соединить Базы данных КУДИНОВА\MySQLServer (SQL Server 15.0.2000.5 - KUDINOVA\Nastya)

Базы данных
 Системные базы данных
 Моментальные снимки базы данных
 Jewelry_Store (Восстановление из копии...)
 Jewelry_Storenew (Восстановление из копии...)
 Jewelry_Storenew1
 JewrlySt
 JewStore
 MobileShop
 Диаграммы баз данных
 Таблицы
 Представления
 Внешние ресурсы
 Синонимы
 Программирование
 Компонент Service Broker
 Хранилище
 Безопасность
 TestDb
 Безопасность
 Объекты сервера
 Репликация
 PolyBase
 Высокий уровень доступности Always On
 Управление
 Каталоги служб Integration Services
 Агент SQL Server (расширенные хранимые процедуры агента о...
 Профилировщик XEvent

```

select Full_Name.First_Name as 'Прізвище',
       Full_Name.Second_Name as 'Імя',
       Full_Name.Patronymic as 'Побатькові',
       Operation_system.Version_operation_system as 'Версія операційної системи',
       COUNT(Operation_system.Id_Operation_system) as 'Кількість куплених операційних систем клієнтом'
  from Operation_system,Full_Name, Parameterss, Model,Product, Orders, Info_order, Client
 Where Full_Name.id_Full_Name= Client.id_Full_Name
   AND Client.Code_client = Orders.Code_client
   AND Orders.Code_order = Info_order.Code_order
   AND Info_order.Code_product = Product.Code_product
   AND Product.Code_model = Model.Code_model
   AND Model.Code_parameter= Parameterss.Code_parameter
   AND Parameterss.Id_Operation_system = Operation_system.Id_Operation_system
 Group by Full_Name.First_Name,
          Full_Name.Second_Name,
          Full_Name.Patronymic,
          Operation_system.Version_operation_system
    
```

Результаты Сообщения

| Прізвище | Імя | Побатькові | Версія операційної системи | Кількість куплених операційних систем клієнтом |
|--------------|----------|-------------|----------------------------|--|
| Васинчук | Іван | Побатькові | Android | 1 |
| Грищенко | Тетяна | Миколаївна | Android | 1 |
| Драбовський | Анатолій | Григорович | Android | 1 |
| Іващенко | Вікторія | Сергіївна | Android | 1 |
| Карпенко | Мартин | Генадійович | Android | 1 |
| Киселевський | Дмитро | Давидович | Android | 1 |
| Козаченко | Альона | Вікторівна | Android | 1 |
| Лічман | Ганна | Василівна | iOS | 1 |
| Пашковський | Максим | Ігорович | Android | 1 |
| Савчук | Оксана | Василівна | Android | 1 |

Додаток 3. Об'єднання таблиць

1. Написати запит, який видасть інформацію про параметри моделі «Galaxy Flip4»

```
select p.Memory_Gb, p.Camera_Mp, p.Battery_Ma, m.Name_model
from Parameterss p JOIN Model m ON p.Code_parameter = m.Code_parameter
WHERE m.Name_model = 'Galaxy Flip4'
```

The screenshot shows the SQL Server Management Studio interface. On the left is the Object Explorer pane, which lists the database 'MobileShop' and its objects. In the center is the 'SQLQuery6.sql' query editor window containing the provided SQL code. Below it is the 'Results' pane, which displays the output of the query:

| | Memory_Gb | Camera_Mp | Battery_Ma | Name_model |
|---|-----------|-----------|------------|--------------|
| 1 | 256 | 14 | 3700 | Galaxy Flip4 |

2. Написати запит, який видасть інформацію про дату замовлення, ціну та кількість одиниць

```
Select i.Number, i.Price_order, o.Datas
from Info_order i
RIGHT OUTER JOIN Orders o
ON i.Code_order = o.Code_order
```

The screenshot shows the SQL Server Management Studio interface. On the left is the Object Explorer pane, which lists the database 'MobileShop' and its objects. In the center is the 'SQLQuery9.sql' query editor window containing the provided SQL code. Below it is the 'Results' pane, which displays the output of the query:

| | Number | Price_order | Datas |
|----|--------|-------------|------------|
| 1 | 2 | 30000 | 2022-11-28 |
| 2 | 4 | 64000 | 2022-11-29 |
| 3 | 1 | 27000 | 2022-09-02 |
| 4 | 7 | 266000 | 2022-12-08 |
| 5 | 3 | 27000 | 2022-12-01 |
| 6 | 5 | 55000 | 2022-10-21 |
| 7 | 10 | 85000 | 2022-09-22 |
| 8 | 16 | 179200 | 2022-04-05 |
| 9 | 45 | 45000 | 2022-11-17 |
| 10 | 25 | 725000 | 2022-07-10 |

3. Написати запит, який видасть та об'єднає інформацію з таблиці продукти та модель
- ```
select p.price_product, p.all_in_stock, p.name_product, m.name_model
from Product p
FULL OUTER JOIN Model m
ON p.Code_model=m.Code_model
```

The screenshot shows the SQL Server Management Studio interface. On the left, the Object Explorer displays the database structure for 'KUDINOVA\MySQLServer'. On the right, the Query Editor window contains the following SQL code:

```
SQLQuery11.sql - KUDINOVA\Nastya (52)* - Microsoft SQL Server Management Studio
SELECT p.price_product, p.all_in_stock, p.name_product, m.name_model
FROM Product p
FULL OUTER JOIN Model m
ON p.Code_model=m.Code_model
```

The results pane shows a table with 10 rows of data:

|    | price_product | all_in_stock | name_product | name_model   |
|----|---------------|--------------|--------------|--------------|
| 1  | 7500          | 4            | Xiaomi       | Redmi 9C     |
| 2  | 16000         | 25           | LG           | G7 ThinQ     |
| 3  | 27000         | 32           | Apple        | Iphone 12    |
| 4  | 38000         | 11           | Samsung      | Galaxy Flip4 |
| 5  | 9000          | 7            | Lenovo       | K12 Pro      |
| 6  | 11000         | 9            | Realme       | 9 Pro        |
| 7  | 8500          | 10           | Nokia        | G10          |
| 8  | 11200         | 18           | Meizu        | Net Pro 11   |
| 9  | 10000         | 60           | Motorola     | NetPro 11    |
| 10 | 58000         | 30           | Tecno        | Spark 8C     |

4. Написати запит за допомогою, якого отримуємо інформацію про виробника та модель, яку він виробляє

```
select m.Name_model, p.Name_producer, p.Contact
from Model m
LEFT OUTER JOIN Producer p
ON m.Code_producer = p.Code_producer
```

The screenshot shows the SQL Server Management Studio interface. On the left, the Object Explorer displays the database structure for 'KUDINOVA\MySQLServer'. On the right, the Query Editor window contains the following SQL code:

```
SQLQuery12.sql - KUDINOVA\Nastya (51)* - Microsoft SQL Server Management Studio
SELECT m.Name_model, p.Name_producer, p.Contact
FROM Model m
LEFT OUTER JOIN Producer p
ON m.Code_producer = p.Code_producer
```

The results pane shows a table with 10 rows of data:

|    | Name_model   | Name_producer        | Contact                     |
|----|--------------|----------------------|-----------------------------|
| 1  | Redmi 9C     | Xiaomi Corporation   | xiaomincc24558@gmail.com    |
| 2  | G7 ThinQ     | Lg electronics       | lgroup@gmail.com            |
| 3  | Iphone 12    | Apple Company        | appi@gmail.com              |
| 4  | Galaxy Flip4 | Samsung Group        | samsunggroup741@gmail.com   |
| 5  | K12 Pro      | Lenovo Group Limited | lenovolimited1234@gmail.com |
| 6  | 9 Pro        | Realme Inc           | realmeic@gmail.com          |
| 7  | G10          | Nokia Corp           | nokia@gmail.com             |
| 8  | Net Pro 11   | Meizu LL             | meizu@gmail.ll              |
| 9  | Net Pro 11   | Motoln Inc           | motorola@gmail.com          |
| 10 | Spark 8C     | TecnoMix             | tecnoc@gmail.com            |