

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. ОСНОВНІ ПРОБЛЕМИ РОЗРОБКИ СУЧАСНИХ БАЗ ДАНИХ. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ КУРСОВОЇ РОБОТИ.....	7
1.1 Актуальність проблем розробки баз даних, основні поняття та визначення.....	7
1.2 Загальна схема процесу розробки інформаційної системи з застосуванням концепції БД та опис основних систем проектування СКБД.....	9
1.3 Аналіз даної предметної області задачі проектування.....	12
1.4 Постановка задачі курсової роботи.....	14
РОЗДІЛ 2 МОДЕЛЮВАННЯ ДАНИХ ПРЕДМЕТНОЇ ОБЛАСТІ.....	15
2.1 Розробка концептуальної моделі даних.....	15
2.2 Перетворення концептуальної моделі в логічну модель даних.....	15
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ.....	17
3.1 Мотивований вибір СКБД та інструментальних програмних засобів для реалізації запропонованої системної архітектури.....	17
3.2 Розробка прикладного програмного забезпечення.....	18
3.2.1 Розробка схеми реляційної бази даних «Ресторану».....	20
3.2.2 Створення БД та запити в SQL Server.....	21
3.2.3 Створення збережених процедур.....	24
3.2.4 Створення користувацьких функцій.....	27
3.2.5 Створення тригерів.....	29
3.3 Розробка візуального інтерфейсу користувачів системи.....	31
РОЗДІЛ 4. РЕЗУЛЬТАТИ ЗАСТОСУВАННЯ РОЗРОБЛЕНОЇ ПРОГРАМНОЇ СИСТЕМИ.....	35
4.1 Стислі відомості щодо розгортання системи.....	35
4.2 Основні режими роботи із системою.....	35
ВИСНОВКИ.....	40
СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ.....	41

ВСТУП

Теорія баз даних – порівняно молода область знань. Вік її складає близько 30 років. Проте сучасний світ інформаційних технологій важко уявити собі без використання баз даних. Практично всі системи в тій чи іншій мірі пов'язані з функціями довготривалого зберігання і обробки інформації. Фактично інформація стає чинником, що визначає ефективність будь-якої сфери діяльності. Збільшилися інформаційні потоки і підвищилися вимоги до швидкості обробки даних, і тепер уже більшість операцій не можуть бути виконані вручну, вони вимагають застосування найбільш перспективних комп'ютерних технологій. Будь-які адміністративні рішення вимагають чіткої і точної оцінки поточної ситуації і можливих перспектив її зміни. І, звичайно, обійтися без інформаційної моделі виробництва, що зберігається в базі даних, в цьому випадку неможливо.

Використання баз даних є однією з характерних рис більшості сучасних інформаційних систем. По своїй суті бази даних є тим, навколо чого і будується інформаційна система будь-якого підприємства. Тому теорії створення та практиці використання баз даних приділяється достатня увага протягом періоду функціонування інформаційних систем. Досить тривалий час основним типом були реляційні бази даних, які на сьогодні вже вважаються класичними. Проте розвиток інформаційних систем поставив перед сучасними базами даних завдання, вирішення яких неможливе в межах використання тільки реляційних баз даних. Крім класичних завдань, сучасні бази даних повинні забезпечувати багатомашинну обробку та зберігання великих обсягів інформації, оперативний аналіз даних, інтеграцію із мережею Інтернет, розмежування доступу користувачів до зберіганої інформації, захист інформації під час її передачі по мережі. Хоча на практиці і використовується чимало різноманітних баз даних, але для більшості з них існує велика кількість спільних ознак, як з погляду розробки, так і використання. Це дає можливість вивчати сучасні бази даних і відповідне прикладне та системне програмне забезпечення на прикладах, які,

незважаючи на свою новизну, вже стали класичними. Як такі приклади вибрано загальні питання проектування, розробки та використання бази даних Microsoft SQL Server. Це пояснюється тим, що Microsoft SQL є однією із найпоширеніших і вдосконалених баз даних.

База даних повинна бути зручною у використанні та задовольняти усім стандартам розробки баз даних. Однією з найважливіших функцій бази даних є швидкість обробки запитів від користувача та швидке видання інформації для нього.

База даних має зручний інтерфейс с користувачем. Завдяки цьому користувач може зручно виконувати пошук інформації яка його зацікавить та допоможе обрати більш кращий товар для придбання.

РОЗДІЛ 1 ОСНОВНІ ПРОБЛЕМИ РОЗРОБКИ СУЧАСНИХ БАЗ ДАНИХ, АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ КУРСОВОЇ РОБОТИ

1.1 Актуальність проблем розробки баз даних, основні поняття та визначення

Нині, попри підвищення комп'ютеризації суспільства, у сфері бізнесу і торгівлі досі немає необхідних коштів, щоб у достатній мірі автоматизувати процес ведення документації і звітності. Однією з основних цілей можна трактувати проблему ведення звітності.

Об'єктом до створення бази даних був вибраний ресторан. Про своєчасність та актуальність аналізованої проблеми свідчить те, що більшість свого часу адміністрація ресторану витрачає на оформлення різної документації і звітів.

Ця база даних пропонує ведення звітності, зберігання, введення і коригування даних.

База даних (БД) – впорядкований набір логічно взаємопов'язаних даних, що використовується спільно, та призначений для задоволення інформаційних потреб користувачів. У технічному розумінні включно й система керування БД.

Головним завданням БД є гарантоване збереження значних обсягів інформації (т.зв. записи даних) та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином БД складається з двох частин: збереженої інформації та системи управління нею. З метою забезпечення ефективності доступу записи даних організовують як множину фактів (елемент даних).

Дані – це інформація, відомості, показники, необхідні для ознайомлення з ким-, чим-небудь, для характеристики когось, чогось або для прийняття певних висновків, рішень. В базах даних важливу роль відіграє інформація - Інформація – абстрактне поняття, що має різні значення залежно від контексту.

Кожна створена база даних має свою предметну область. Предметна область – це необхідний для розробки бази даних об'єкт, який має в собі дані, які будуть зберігатися в базі даних.

Модель даних – абстрактне представлення реального світу, що відображає тільки ті об'єкти, що безпосередньо стосуються програми. Це, як правило, визначає специфічну групу об'єктів, їх атрибутивне значення і відношення між ними.

Відомі два підходи до організації інформаційних масивів: файлова організація та організація у вигляді бази даних. Файлова організація передбачає спеціалізацію та збереження інформації, орієнтованої, як правило, на одну прикладну задачу, та забезпечується прикладним програмістом. Така організація дозволяє досягнути високої швидкості обробки інформації, але характеризується рядом недоліків. Характерна риса файлового підходу - вузька спеціалізація як обробних програм, так і файлів даних, що служить причиною великої надлишковості, тому що ті самі елементи даних зберігаються в різних системах. Оскільки керування здійснюється різними особами (групами осіб), відсутня можливість виявити порушення суперечливості збереженої інформації. Розроблені файли для спеціалізованих прикладних програм не можна використовувати для задоволення запитів користувачів, які перекривають дві і більше області. Крім того, файлова організація даних внаслідок відмінностей структури записів і форматів передання даних не забезпечує виконання багатьох інформаційних запитів навіть у тих випадках, коли всі необхідні елементи даних містяться в наявних файлах. Тому виникає необхідність відокремити дані від їхнього опису, визначити таку організацію збереження даних з обліком існуючих зв'язків між ними, яка б дозволила використовувати ці дані одночасно для багатьох застосувань. Вказані причини обумовили появу баз даних. База даних може бути визначена як структурна сукупність даних, що підтримуються в активному стані та відображає властивості об'єктів зовнішнього (реального) світу. В базі даних містяться не

тільки дані, але й описи даних, і тому інформація про форму зберігання вже не схована в сполученні "файл-програма", вона явним чином декларується в базі.

База даних орієнтована на інтегровані запити, а не на одну програму, яку випадку файлового підходу, і використовується для інформаційних потреб багатьох користувачів. В зв'язку з цим бази даних дозволяють в значній мірі скоротити надлишковість інформації. Перехід від структури БД до потрібної структури в програмі користувача відбувається автоматично за допомогою систем управління базами даних (СУБД).

1.2 Загальна схема процесу розробки інформаційної системи з застосуванням концепції БД та опис основних систем проектування СКБД

З розвитком інформаційного забезпечення систем автоматизованої обробки інформації, прагненням забезпечити виконання нових режимів обробки даних у реальному часі і з мультидоступом до схованих даних позначилась нова тенденція до складення інформаційного забезпечення розподілених баз даних. В умовах використання таких баз створюються комплексні масиви нелінійної структури, які мають усі дані про ту чи іншу предметну область або про керований об'єкт як постійного, так і перемінного характеру.

Взагалі база даних це сукупність даних на машинних носіях, які використовуються при функціонуванні системи обробки інформації, організовані по визначеним правилам, які передбачають загальні принципи описування збереження і маніпулювання ними, а також які незалежні від прикладних програм. В основі організації бази даних є модель даних, яка визначає правила, у відповідності з якими структуруються дані. За допомогою моделі представляється велика кількість даних і описуються взаємно зв'язки між ними. Найбільш поширені такі моделі даних: ієрархічна, мережева, реляційна.

В ієрархічній моделі зв'язок даних «один до одного» (1:1) означає, що кожному значенню (екземпляру) елемента даних А відповідає одне і тільки одне значення, пов'язаного з ним елемента В. Наприклад, поміж такими елементами пар даних, як код готової продукції і її найменування є вищезазначений зв'язок, так як тільки кожному коду продукції відповідає одне її найменування.

Зазначимо, що ієрархічна модель даних будується на основі принципу підпорядкованості поміж елементами даних і представляє собою деревоподібну структуру, яка складається із вузлів (сегментів) і дуг (гілок).

Дерево у ієрархічній структурі упорядковане за існуючими правилами розташування його сегментів і гілок: на верхньому рівні знаходиться один, кореневий (вихідний) сегмент, сегмент другого рівня, породжений, залежить від першого, вихідного; доступ до кожного породженого (крім кореневого) здійснюється через його вихідний сегмент; кожний сегмент може мати по декілька екземплярів конкретних значень елементів даних, а кожний елемент породженого сегменту пов'язаний з екземпляром вихідного і створює один логічний запис; екземпляр породженого сегменту не може існувати самостійно, тобто без кореневого сегменту; при вилученні екземпляру кореневого сегмента також вилучаються усі підпорядковані і взаємопов'язані з ним екземпляри породжених сегментів.

В мережевій моделі зв'язок «один до багатьох» (1:В) означає, що значенню елемента А відповідають багато (більше одного) значень, пов'язанню з ним елементів В. Наприклад, поміж елементами даних "код виробу" (елемент А) і "кодом матеріалів" (елементи В) існує такий взаємозв'язок бо на виготовлення одного виробу використовується багато різних матеріалів.

Мережева модель даних представляє собою орієнтований граф з поіменованими вершинами і дугами. Вершини графа - записи, які представляють собою по іменовану сукупність логічних взаємозв'язаних елементів даних або агрегатів даних. Під агрегатом даних розуміють пошановану сукупність елементів даних, які є усередині запису. Для кожного типу записів може бути кілька екземплярів конкретних значень його інформаційних елементів Два записи, взаємозв'язані дугою, створюють набір даних. Запис, з якого виходить дуга, називається власником набору, а запис, до якого вона направлена, - членом набору.

В реляційній моделі зв'язок «багатьох до багатьох» (В:В) указує на те, що декільком значенням елементів даних А відповідає декілька значені елементів даних В. Наприклад, поміж елементами даних «код операції технологічного процесу» і «табельний номер працівника» існує зазначені взаємозв'язок, так як багато операцій технологічного процесу можуть виконувати різні працівники (табельні номери) і навпаки.

Реляційна модель даних являє собою набір двомірних плоских таблиць, що складаються з рядків і стовпців. Первинний документ або лінійний масив являє собою плоску двомірну таблицю. Така таблиця називається відношенням, кожний стовбець-атрибутом, сукупність значень одного типу (стовпця) – доменом, а рядка – кортежем. Таким чином, стовпці таблиці являються традиційними елементами даних, а рядки – записами. Таблиці (відношення) мають імена. Імена також присвоюються і стовпцям таблиці. Кожний кортеж (запис) відношення має ключ. Ключі є прості і складні. Простий ключ - це ключ, який складається з одного атомарного атрибуту, значення якого унікальне (яке не повторюються).Складний ключ складається з двох і більше атрибутів. Для зв'язків відношень друг з другом в базі даних є зовнішні ключі. Атрибут або комбінація атрибута відношення є зовнішнім ключем, якщо він не є основним (первинним) ключем цього відношення, але являється первинним ключем для другого відношення.

Різновидністю баз даних, з точки зору їх зберігання і використання, є

розподіленні бази даних. Ці бази даних широко використовуються при організації комплексів взаємопов'язаних АРМ фахівців, на яких застосовуються ПЕОМ.

Розподілена база даних - це сукупність логічно зв'язаних баз даних або частин однієї бази, які розпаралелені поміж декількома територіально – розподіленими ПЕОМ і забезпечені відповідними можливостями для управління цими базами або їх частинами. Тобто, розподілена база даних реалізується на різних просторово розосереджених обчислювальних засобах, разом з організаційними, технічними і програмними засобами її створення і ведення.

1.3 Аналіз наданої предметної області задачі проектування

Практика розробки та експлуатації баз даних викристалізувала базові вимоги до їх побудови. Це зокрема вимоги:

- цілісності даних (обмеження; правила використання обмежень; правила обробки при порушенні обмежень цілісності; ефективність використання обмежень);
- узгодженості даних;
- відновлюваності даних (при збоях обладнання);
- безпеку (від несанкціонованих дій);
- ефективності, яке визначається такими параметрами, як швидкодія, мінімальний час доступу та мінімальна пам'ять.

З іншої сторони в середовищі прикладних користувачів-розробників були сформовані так звані бізнес – правила для роботи з базами даних. Їх суть полягає в ефективній реалізації роботи з даними, яка гарантує, що дані, які містяться в базах даних, повинні відповідати політиці і стратегії організації, зокрема і її правилам. Наприклад, для реалізації цього принципу потрібно встановити вимоги до кожного стовпця в таблиці баз даних так, щоб він не допускав значень, які суперечили б політиці організації. Фактично бізнес правила вимагають виконання трьох основних

функцій:

- зберігати не бажанні дані поза базами даних;
- однозначно описувати і строго визначати зв'язки між стовпцями і таблицями;
- при необхідності надавати інформацію звідки, коли і ким дані внесені в базу даних.

Глосарій (від лат. *glossarium* - словник глос) – словник до тексту, що пояснює маловідомі або застарілі слова. Глосарій – список понять в специфічній області знання з їх визначеннями. Традиційно, глосарій знаходиться в кінці книги і включає терміни в межах цієї книги, які є або недавно введеними, або як мінімум незвичайними.

Сервер (англ. *server* – «служба») – у комп'ютерній термінології термін може стосуватися окремого комп'ютера чи програми. Головною ознакою в обох випадках є здатність машини чи програми переважну кількість часу працювати автономно, без втручання людини реагуючи на зовнішні події згідно встановленого програмного забезпечення. Втручання людини відбувається під час встановлення серверу і під час його сервісного обслуговування. Часто це роблять окремі адміністратори серверів з вищою кваліфікацією.

Таблиця - це перелік, зведення статистичних даних або інших відомостей, розташованих у певному порядку й за графами.

Атрибут (*attribute*) – невід'ємна, необхідна для забезпечення цілісності об'єкта (предмета) або суб'єкта (людини) властивість, його частина, додаток.

Microsoft SQL Server – комерційна система керування базами даних, що розповсюджується корпорацією Microsoft. Мова, що використовується для запитів – Transact-SQL, створена спільно Microsoft та Sybase.

Агрегатна функція (дослівно – функція складеного значення) – функція, яка повертає одинарне значення з колекції вхідних значень такої як множина (set), мультимножина (multiset) або список (list).

Тип даних – характеристика, яку явно чи неявно надано об'єкту (змінній, функції, полю запису, константі, масиву тощо).

(ГІК, англ. GUI, Graphical user interface) – інтерфейс між комп'ютером і його користувачем, що використовує піктограми, меню, і вказівний засіб для вибору функцій та виконання команд. Зазвичай, можливе відкриття більше, ніж одного вікна на одному екрані.

FOREIGN KEY – в одній таблиці вказує на інший PRIMARY KEY.

PRIMARY KEY – це обмеження дозволяє однозначно ідентифікувати кожний запис у таблиці.

Складовий ключ – це ключ, який складається з двох чи більше атрибутів.

1.4 Постановка задачі курсової роботи

В курсовій роботі за мету взято створення бази даних, яка має надати можливість зберігання інформації, забезпечувати швидкий пошук інформації для клієнта. База даних повинна реалізовувати змогу оновлення та додавання, або видалення даних які в ній зберігаються, а також зручне та надійне користування, та доступ усіх необхідних програм до неї.

При розробленні бази даних необхідно створити таблиці в MicrosoftSQL Server, та заповнити їх необхідними полями(атрибути). Але до реалізації таблиць необхідно розробити модель в model.erwin, в якій буде зображено зв'язки між таблицями та основну схему побудови таблиць.

Після цього треба створити запити завдяки мові mysql, та перевірити їх роботу на видання різної інформації. Запити повинні мати реалізацію у вигляді транзакції, триггеру та ін..

РОЗДІЛ 2 МОДЕЛЮВАННЯ ДАНИХ ПРЕДМЕТНОЇ ОБЛАСТІ

2.1 Розробка фізичної моделі даних

На створеній ER-діаграмі , яка знаходиться в додатку А присутні такі сутності як:

- Співробітники – ця сутність містить дані про співробітників ресторану.
- Посади – тут знаходиться інформація про усі посади ресторану, їх оклад, обов`язки і вимоги до працівників.
- Замовлення – тут знаходяться дані про замовлення, замовника і відмітка про виконання. Зберігає в собі primary key сутностей «Меню» і «Склад».
- Склад - в цій сутності є інформація про товари які є на складі і про їх постачальника.
- Меню – тут знаходиться інформація про страву, її інгредієнти і вартість.

2.2 Перетворення концептуальної моделі в логічну модель даних

На цьому етапі ми продовжимо роботу з локальною концептуальною моделлю даних, створеною на попередньому етапі. Наше завдання полягає в доопрацюванні цієї моделі з метою видалення з них всіх елементів, що ускладнюють реалізацію даної моделі в середовищі реляційної СУБД. В результаті виконання цих дій структура концептуальної моделі даних буде змінена таким чином, щоб повністю відповідати вимогам, що висувуються для реляційної моделі організації баз даних. Тому нову модель більш коректно називати логічною моделлю даних. Далі створена логічна модель даних буде перевірена з використанням правил нормалізації і піддана контролю на можливість виконання всіх необхідних користувачам транзакцій так, як це зазначено у специфікаціях на створюваний додаток. Згодом перевірена логічна моделі даних може використовуватись як прототип, якщо це буде потрібно. На цьому етапі ми займемося перетворенням концептуальної моделі даних з метою видалення з неї всіх структур, реалізація яких у СУБД реляційного типу скрутна.

У локальній концептуальній моделі даних надлишкові зв'язки відсутні, тому ми перетворюємо нашу концептуальну модель даних в логічну модель даних. За допомогою програми Erwin Data Modeler підключаємося до бази даних «Ресторан» створеної раніше в Microsoft SQL Server 2008. В результаті ми отримуємо логічну модель даних:

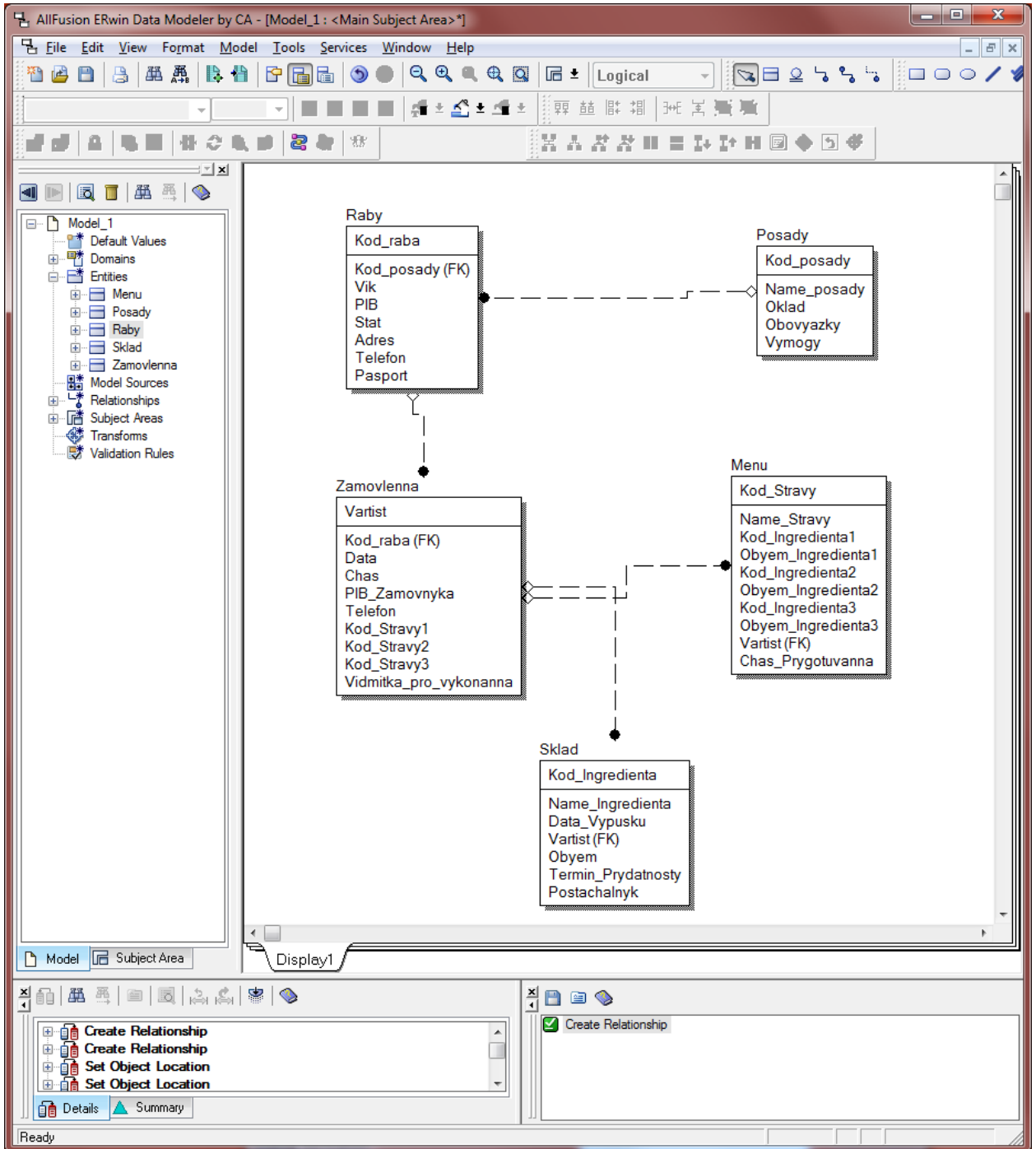


Рисунок 2.1 – Логічна модель даних

РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Мотивований вибір СКБД та інструментальних програмних засобів для реалізації запропонованої системної архітектури

Системи управління базами даних – це програмні засоби, за допомогою яких можна створювати бази даних, заповнювати їх та працювати з ними. У світі існує багато різноманітних систем управління базами даних. Багато з них насправді є не закінченими продуктами, а спеціалізованими мовами програмування, за допомогою яких кожний, хто вивчить дану мову, може сам створювати такі структури, які йому потрібні, і вводити в них необхідні елементи управління. До таких мов відносяться Clipper, Paradox, FoxPro та інші.

Необхідність програмувати завжди утримувала устаткування баз даних в малому бізнесі. Великі підприємства могли дозволити собі зробити наказ на програмування спеціальної системи «під себе». Малим підприємствам звичайно не по силам було не тільки вирішити, але й правильно сформулювати цю задачу своїми силами. Для цього треба володіти основами програмування на мові Basic.

EMS SQL Manager для MySQL Freeware це найпотужніший інструмент для адміністрування баз даних MySQL та розвитку. Вона працює з будь-якими версіями MySQL від 3,23 до новітніх і підтримує всі новітні функції MySQL, включаючи тригери, представлення, збережені процедури і функції, зовнішні ключі InnoDB, Unicode даних і так далі. SQL Manager для MySQL дозволяє створювати і редагувати всі об'єкти баз даних MySQL, проектування баз даних MySQL, запусить SQL скрипти, керувати користувачами і їх привілеями, а також безліч інших корисних інструментів для ефективного адміністрування MySQL. SQL Manager для MySQL має стані арт-графічний користувальницький інтерфейс з добре описано майстер системи, настільки ясно у використанні, що навіть новачкові не слід плутати з ним.

Основні характеристики:

- Підтримка даних UTF8

- Швидка навігація і управління базами
- Просте управління всіма об'єктами MySQL
- Потужні інструменти управління даними
- Ефективне управління параметрами безпеки
- Відмінні інструменти для побудови запитів
- Простота у використанні майстра для виконання завдань MySQL
- Підключення до MySQL Server через HTTP

Система управління базами даних (СУБД) поєднує відомості з різних джерел в одній реляційній базі даних. Створювані форми, запити і звіти дозволяють швидко й ефективно обновляти дані, отримувати відповіді на питання, здійснювати пошук потрібних даних, аналізувати дані, друкувати звіти, діаграми і поштові наклейки.

Організація єдиної бази даних стала можливою лише завдяки тому, що були створені спеціальні програмні продукти – системи управління базами даних (СУБД).

Основне призначення СУБД – створення та підтримка в актуальному стані бази даних, а також зв'язок її з програмами розв'язування економічних завдань (прикладні програми користувачів).

База даних – це комп'ютерний термін, який використовується для позначення сукупності інформації з окремої теми або відомостей, пов'язаних з деякою прикладною задачею. Зберігання інформації у вигляді бази даних полегшує доступ до неї, пошук та вилучення потрібних фрагментів.

На магнітному диску база даних може зберігатись у вигляді одного файлу (бази даних MS Access, Informix та ін.) або у вигляді папки з файлами (бази даних Interbase, Paradox та ін.).

3.2 Розробка прикладного програмного забезпечення

Система управління базами даних MySQL дуже часто застосовується для зберігання важливої інформації на веб-сайтах. Якщо це звичайний сайт або форум – в базі можуть зберігатися повідомлення користувачів, дані для

динамічних сторінок, дані про відвідини, якщо це який-небудь інтерактивний сервіс, то окрім даних про доступ (конфіденційних), там зберігається і інша інформація про користувача і його дії. Все це приводить до того, що загальна безпека сайту, вірніше за всю веб-сервера-системи, залежить від того, наскільки захищений саме сервер бази даних.

На звичайних віртуальних хостингах кожен клієнт отримує свій логін і пароль, і йому доступна тільки одна база, в якій він може створювати довільну кількість таблиць. Один і той же фізичний сервер БД використовують різні клієнти, кожен з яких має доступ тільки до однієї певної бази даних. Ситуація, коли у користувача одна база, за володіння якої б'ються і "движок" форуму (якому потрібно створити сотню і більше таблиць), і скрипти списків розсилки, новин, пошуковий скрипт, а якщо ще встановлена система управління контентом (CMS) або електронний магазин – тоді в цій базі виникає така величезна кількість різних таблиць, деколи з дуже дивними і нічого не позначаючими назвами (добре, якщо два скрипти не використовують таблиці з однаковими назвами, але різною структурою). У таких випадках дуже бажано мати можливість створити декілька окремих баз даних, і виділити їх для різних застосувань (наприклад, одна база для форуму, інша для електронного магазину).

Окрім складнощів з управлінням декількома сотнями таблиць в одній базі, ви зіткнетеся з необхідністю обмежувати доступ різних користувачів до таблиць, баз і навіть окремих стовпців конкретної таблиці. Поспішаємо вас заспокоїти – розробники СУБД MYSQL вже поклопоталися про подібну ситуацію – в MYSQL є дуже гнучкий і могутній механізм управління і розмежування доступу користувачів до баз і таблиць.

Працює цей механізм, природно, через службові таблиці. У списку баз даних є одна службова база під назвою «mysql», в якій зберігаються в декількох таблицях всі службові дані, необхідні для роботи сервера. Поки нас цікавить тільки управління правами доступу, за які відповідають наступні таблиці: `columns_priv` – доступ на рівні стовпців таблиці;

db – доступ до окремих баз даних;

host – доступ до баз з конкретних хостов/ір;

tables_priv – доступ до окремих таблиць бази даних;

user – глобальні настройки доступу для конкретних користувачів.

3.2.1 Розробка схеми реляційної бази даних «Ресторану».

Raby	
PK	Kod raba
FK1	Kod posady PIB Vik Stat Adres Telefon Pasport

Posady	
PK	Kod posady
	Name posady Oklad Obovyazky Vymogy

Zamovlenna	
PK	Vartist
FK1	Kod raba Data Chas PIB zamovnyka Telefon Kod stravy 1 Kod stravy 2 Kod stravy 3 Vidmitka pro vykonanna

Menu	
PK	Kod stravy
	Name stravy Kod ingredienta 1 Obyem ingredienta 1 Kod ingredienta 2 Obyem ingredienta 2 Kod ingredienta 3 Obyem ingredienta 3
FK1	Vartist Chas prygotuvanna

Sklad	
PK	Kod ingredienta
FK1	Name ingredient Data vypusku Vartist Obyem Termin prydatnosti Postachalnyk

Для створення діаграми необхідно натиснути правою кнопкою мишки на елементі БД «Діаграми баз даних» і в відкритому контекстному меню вибрати пункт «Создать диаграмму базы данных». В відкритому діалоговому вікні потрібно вибрати таблиці, які будуть додані на діаграму. В результаті отримаємо:

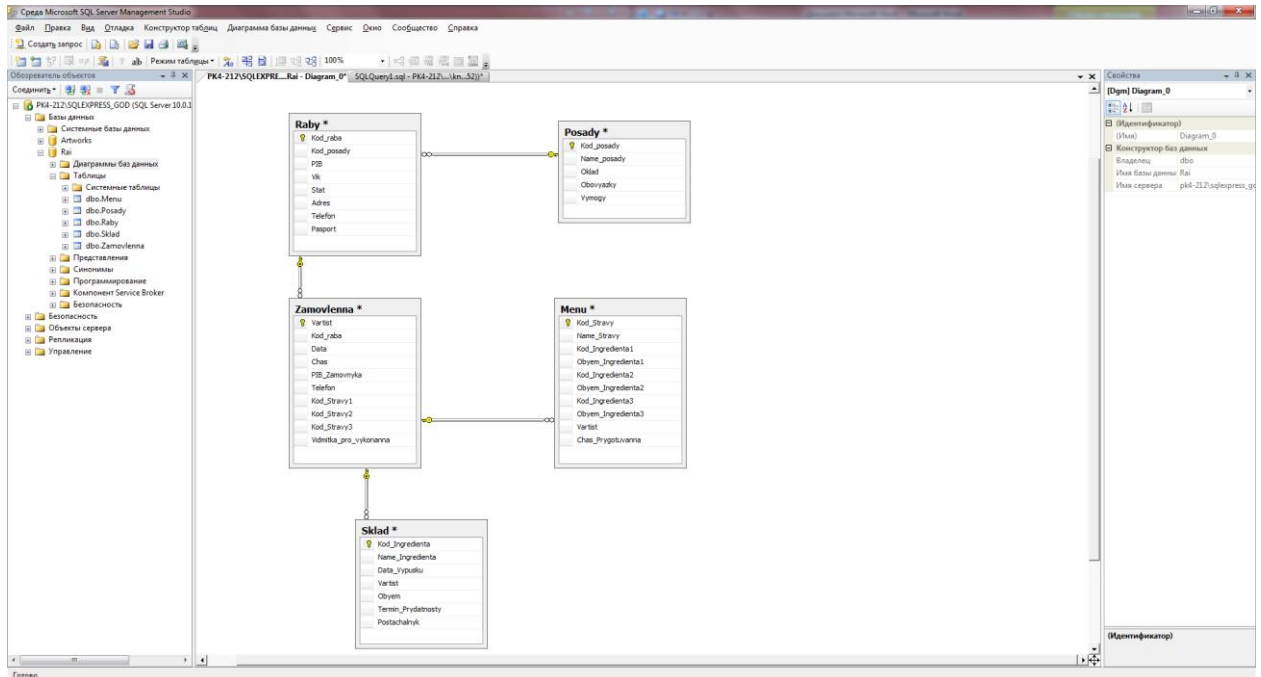


Рисунок 3.1 – Діаграма бази даних

3.2.2 Створення БД та запити в SQL Server

Після запуску середовища розробки з'явиться вікно підключення до сервера. «Соединение с сервером».

В цьому вікні необхідно натиснути кнопку «Соединить».

Після натиснення кнопки «Соединить» з'явиться вікно середовища розробки «SQL Server Management Studio» .

Натиснення кнопки «Создать запрос» здійснить відкриття вікна запитів.

Для виконання запиту, введеного у вікні, необхідно натиснути на кнопку «Выполнить».

Виконаємо створення БД «Radio» в середовищі розробки SQL Server Management Studio. В новому вікні запитів введемо запит на створення БД:

```
IF DB_ID('Rai') IS NULL
CREATE DATABASE Rai;
```

Далі створимо таблиці: «Raby », «Posady», «Zamovlenna», «Menu», «Sklad».

Код запиту на створення таблиці «Raby»:

```
USE Rai;
```

```

IF OBJECT_ID('dbo.Raby', 'U') IS NOT NULL
DROP TABLE dbo.Raby;
CREATE TABLE dbo.Raby
(
Kod_raba BIGINT IDENTITY(1,1) CONSTRAINT
PK_Raby PRIMARY KEY,
Kod_posady BIGINT NULL,
PIB VARCHAR NULL,
Vik VARCHAR NULL,
Kod_raba BIGINT NULL,
Stat VARCHAR NULL,
Adres VARCHAR NULL,
Telefon BIGINT NULL,
Pasport VARCHAR NULL
);

```

Запит для заповнення таблиць:

```

INSERT INTO dbo.[ім'я таблиці] (поля таблиці)
values (дані як потрібно ввести);

```

Код запиту на заповнення таблиці «Raby»:

```

INSERT INTO dbo.Raby (PIB,Vik,Stat,Adres,Telefon,Pasport)
VALUES('Панашук Петро Петрович', 19, 'Ч', 'с.Лище',
380667071803, 'АЮ458710'),
('Свід Олександр Андрійович', 19, 'Ч',
'м.Луцьк', 380997023885, 'АЮ587410'),
('Бобицький Тарас Миколайович', 20, 'Ч', 'сmt.Іваничі',
380985478952, 'АЮ487412'),
('Шульган Андрій Віталійович', 20, 'Ч', 'с.Забужжя',
380965874741, 'АЮ596320'),
('Суша Ангеліна Анатоліївна', 19, 'Ж',
'м.Луцьк', 380985874125, 'АЮ587410'),
('Шевчук Марія Миколаївна', 21, 'Ж',
'м.Київ', 380965236982, 'АЮ587425'),
('Гурко Вікторія Василівна', 20, 'Ж',
'м.Луцьк', 380964585330, 'АЮ485201'),
('Гурко Вікторія Василівна', 20, 'Ж',
'м.Луцьк', 380964585330, 'АЮ485201');

```

Створення запити відділ кадрів:

```

SELECT R.PIB, R.Vik, R.Stat,
P.Name_posady, P.Oklad, P.Obovyazky
FROM dbo.Raby AS R
JOIN dbo.Posady AS P
ON R.Kod_posady = P.Kod_posady;

```

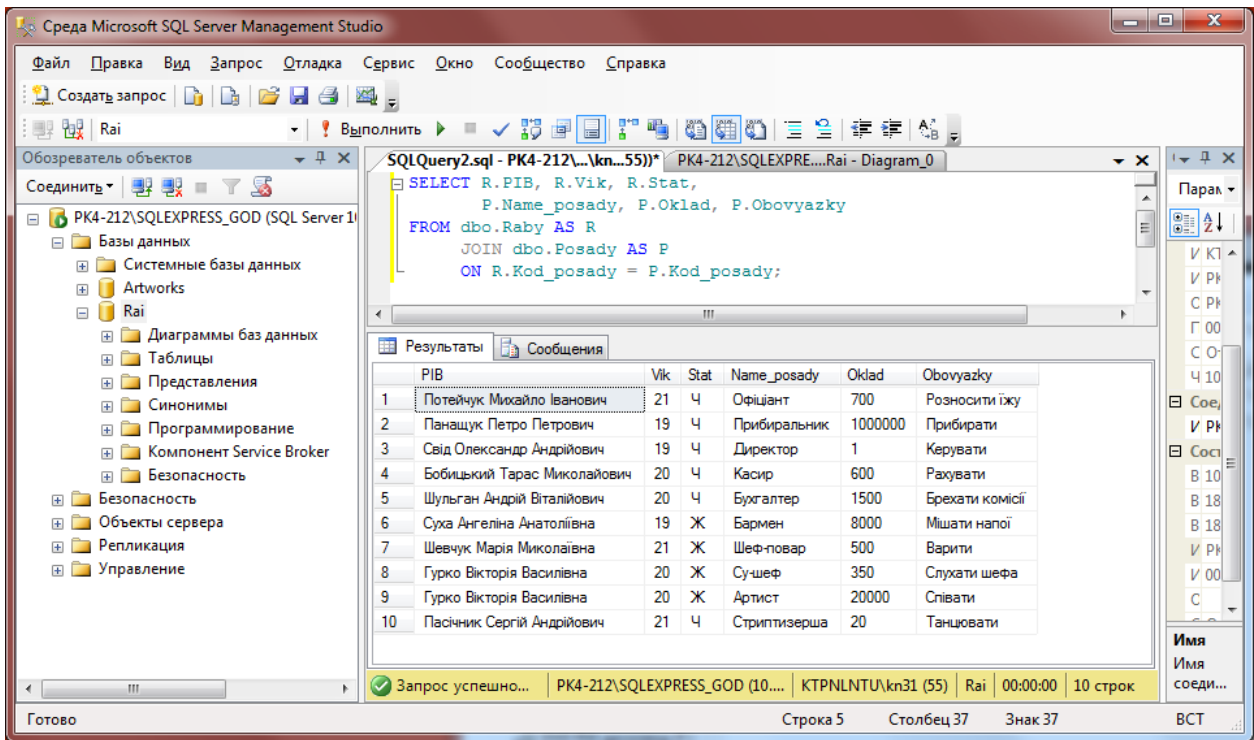


Рисунок 3.2 – Виконання запиту відділ кадрів

Створення запиту меню:

```
SELECT S.Name_Ingredienta, S.Obyem, S.Postachalnyk,
       M.Name_Stravy, M.Vartist, M.Chas_Prygotuvanna
FROM   dbo.Sklad AS S
JOIN   dbo.Menu AS M
ON     S.Kod_Ingredienta = M.Kod_Ingrediental;
```

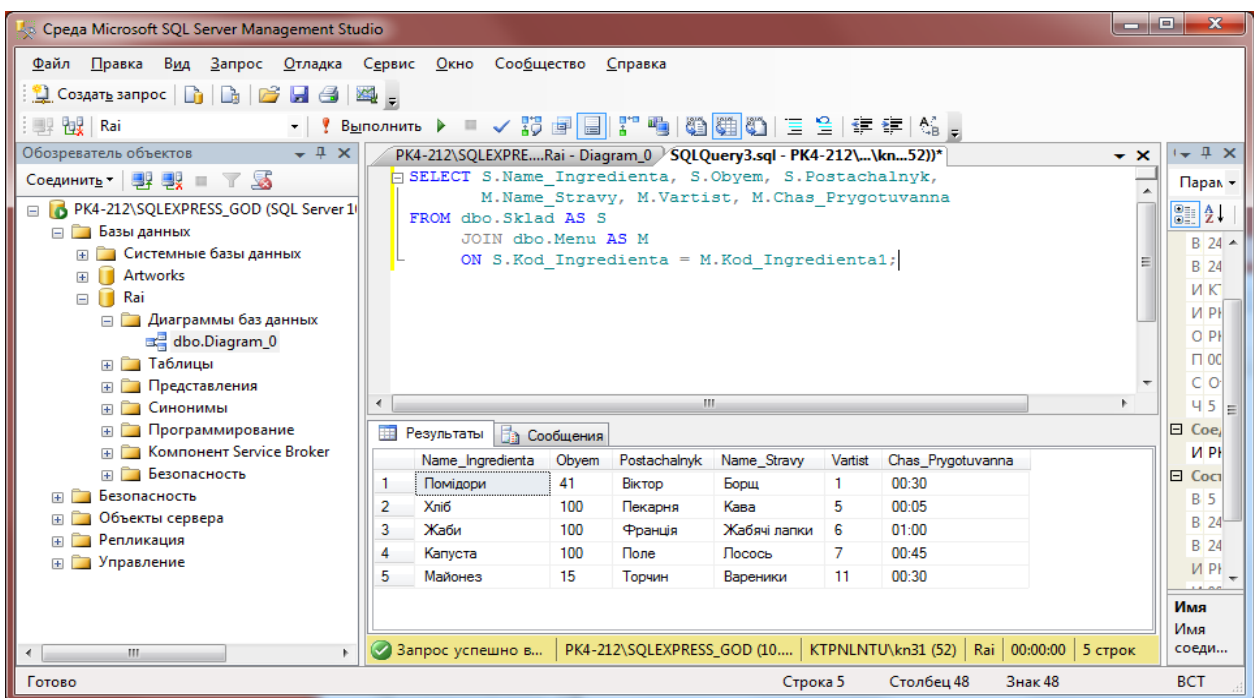


Рисунок 3.3 – Виконання запиту меню

Створення запиту замовлення:

```

SELECT R.PIB, R.Vik, R.Stat,
Z.Data, Z.PIB_Zamovnyka, Z.Vidmitka_pro_vykonanna,
M.Name_Stravy, M.Kod_Ingrediental, M.Chas_Prygotuvanna
FROM dbo.Raby AS R
JOIN dbo.Zamovlenna AS Z
JOIN dbo.Menu AS M
ON M.Vartist = Z.Kod_Stravy1
ON Z.Kod_Stravy1 = R.Kod_raba;

```

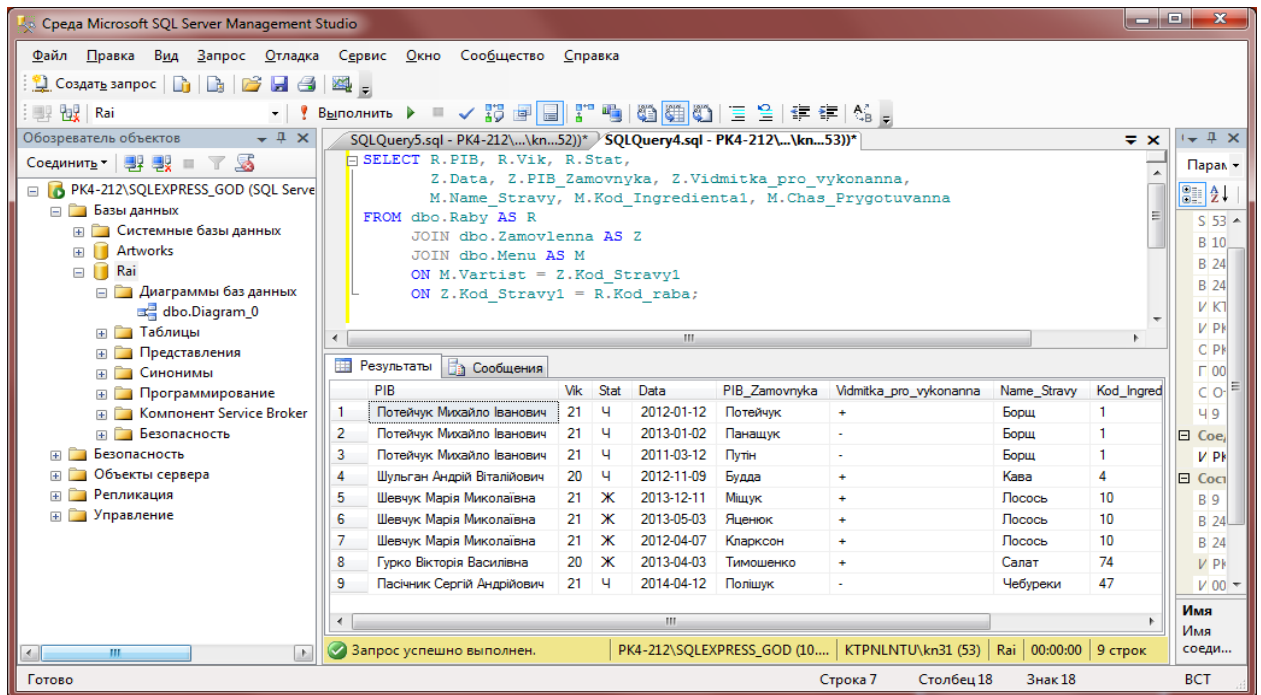


Рисунок 3.4 – Виконання запиту замовлення

3.2.3 Створення збережених процедур

Збережена процедура, для знаходження середнього арифметичного:

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE MeanValue1
@Value1 Real = 0,
@Value2 Real = 0,
@Value3 Real = 0
AS
BEGIN
SET NOCOUNT ON;
SELECT 'MeanValue1'=(@Value1 + @Value2 + @Value3)/3
END
GO

```

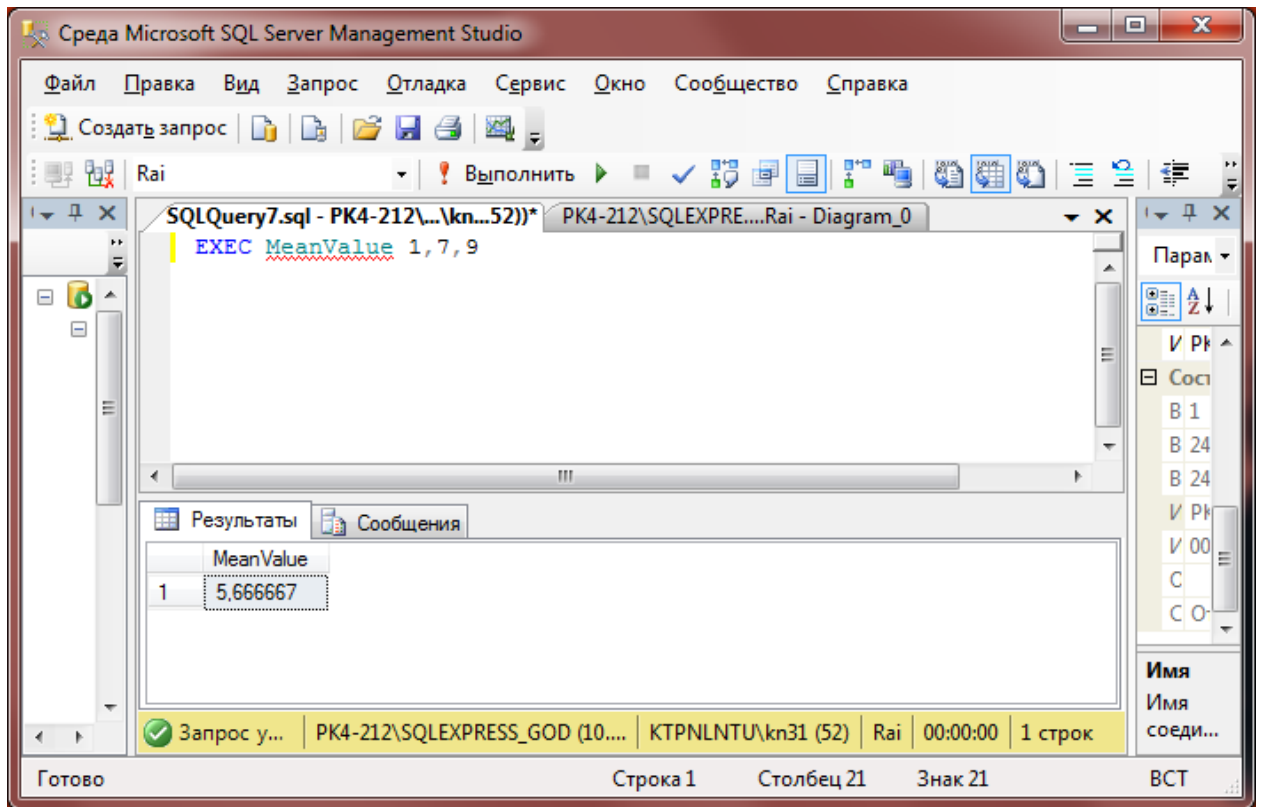


Рисунок 3.5 – Виконання даної процедури

Збережена процедура, для відбору співробітників по їх прізвищам:

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [Rab by Name]
@Name varchar(50)=' '
AS
BEGIN
SET NOCOUNT ON;
SELECT * FROM Raby WHERE PIB = @Name
END
GO

```

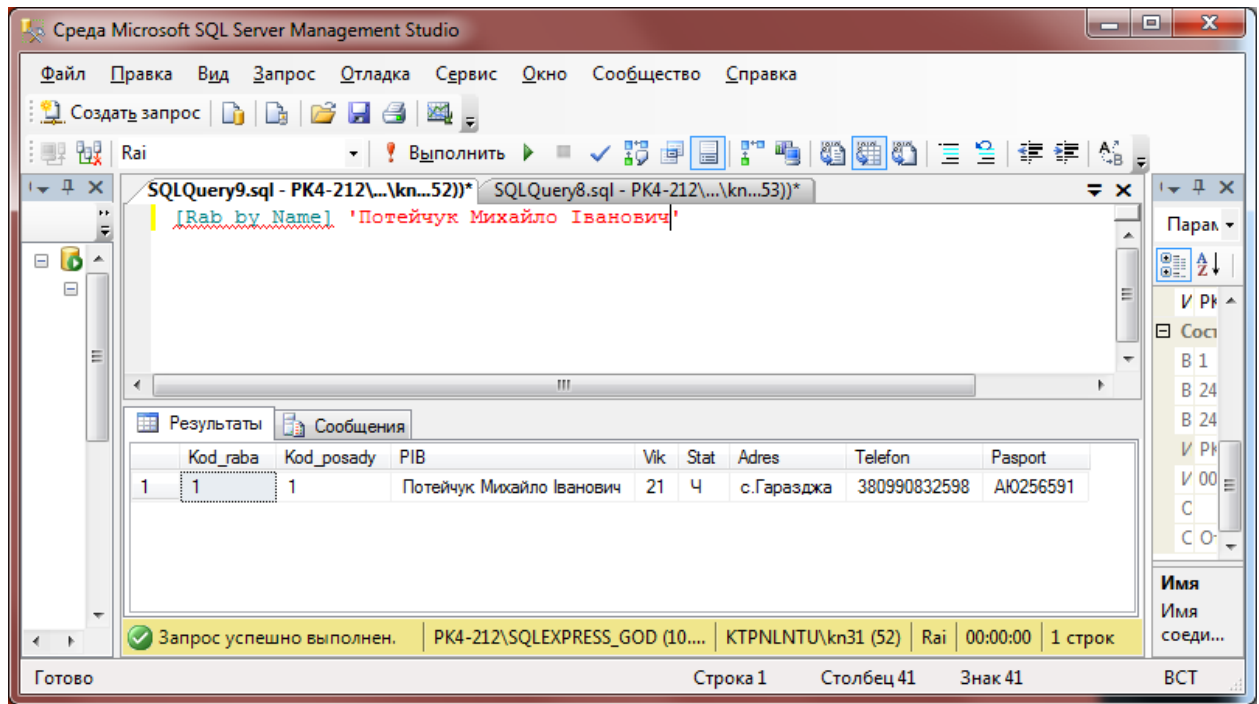


Рисунок 3.6 – Виконання даної процедури

Збережена процедура, для відбору клієнтів які зробили замовлення у 2013 році:

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE Data2013
AS
BEGIN
SET NOCOUNT ON;
SELECT PIB_Zamovnyka FROM Zamovlenna WHERE YEAR(Data) >=
2012 AND YEAR(Data) <= 2014
END
GO

```

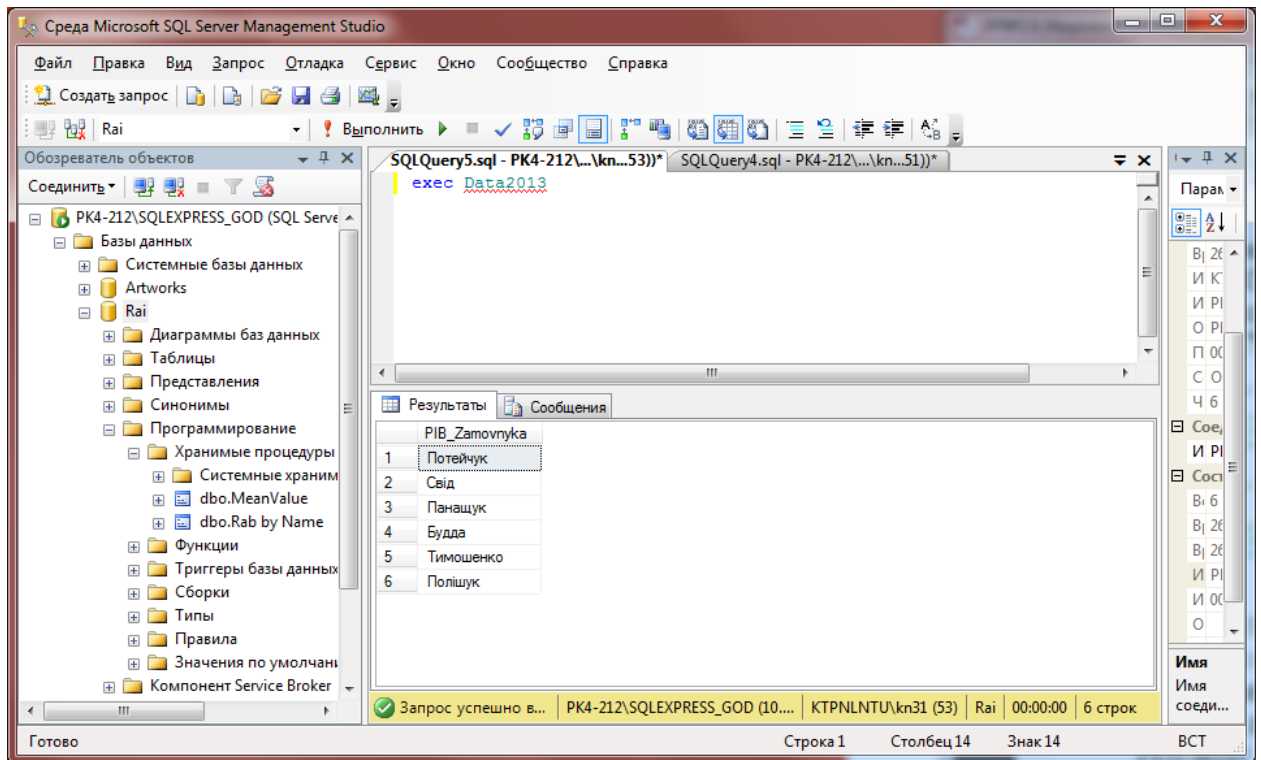


Рисунок 3.7 – Виконання даної процедури

3.2.4 Створення користувацьких функцій

Користувацька функція, для знаходження середнього арифметичного значення:

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION FuncMeanValue
@Value1 int, @Value2 int, @Value3 int
RETURNS real
AS
BEGIN
DECLARE @Result real
SELECT @Result=(@Value1 + @Value2 + @Value3)/3
RETURN @Result
END

```

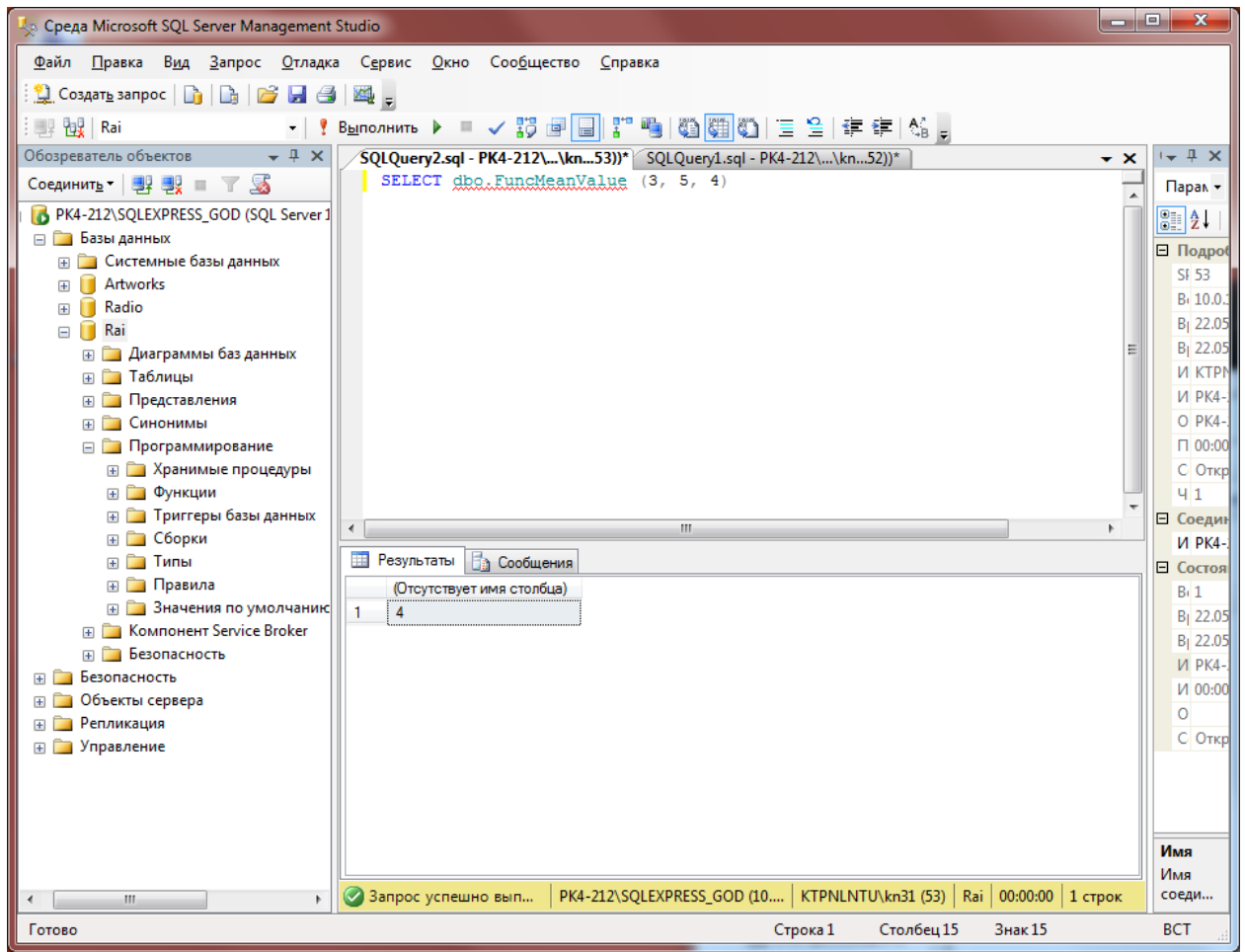



Рисунок 3.8 – Виконання даної функції

Користувацька функція, для пошуку співробітників через таблицю замовлення:

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [GetRab]
(
  @PIB_Zamovnyka varchar(100)
)
RETURNS TABLE
AS
RETURN
(
  SELECT Raby.PIB, Raby.Vik, Raby.Stat
  FROM Raby INNER JOIN
  Zamovlenna ON Raby.Kod_Raba = Zamovlenna.Kod_raba
  WHERE (Zamovlenna.Chas = @PIB_Zamovnyka)
)
GO

```

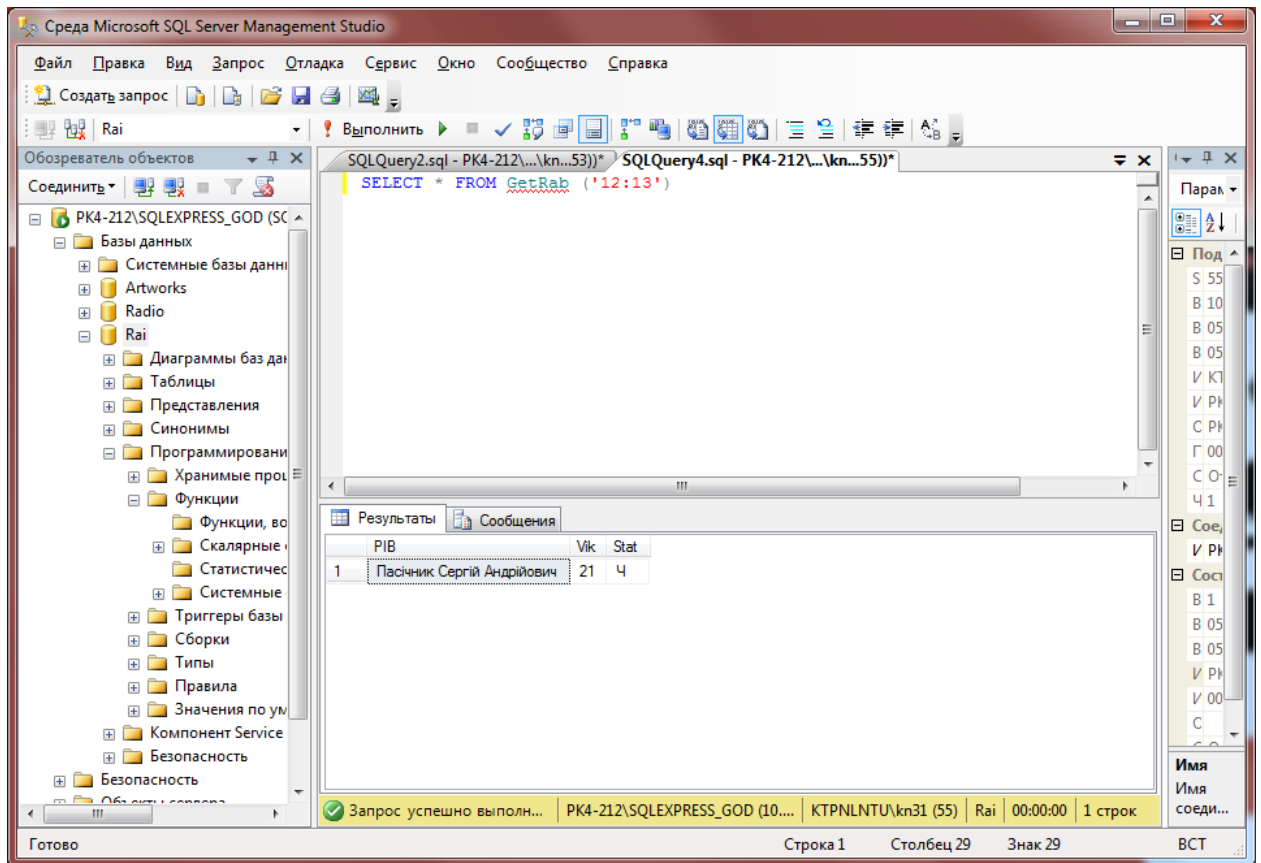


Рисунок 3.9 – Виконання даної функції

3.2.5 Створення тригерів

Тригер, для додавання записів:

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TRIGGER Insert_trigger
ON dbo.Raby
AFTER INSERT
AS
BEGIN
SET NOCOUNT ON;
PRINT ('Запис доданий!');
END
GO

```

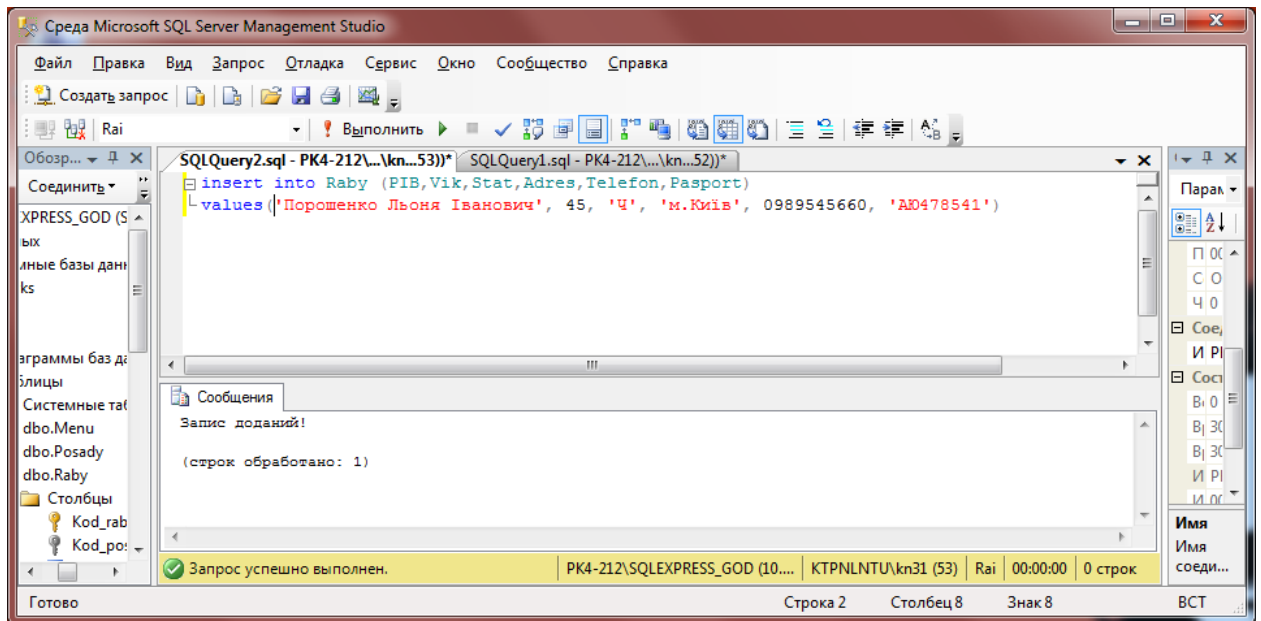


Рисунок 3.10 – Виконання тригера для додавання записів

Тригер, для редагування записів:

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TRIGGER Update_trigger
ON dbo.Raby
AFTER UPDATE
AS
BEGIN
SET NOCOUNT ON;
PRINT('Запис змінений!')
END
GO
```

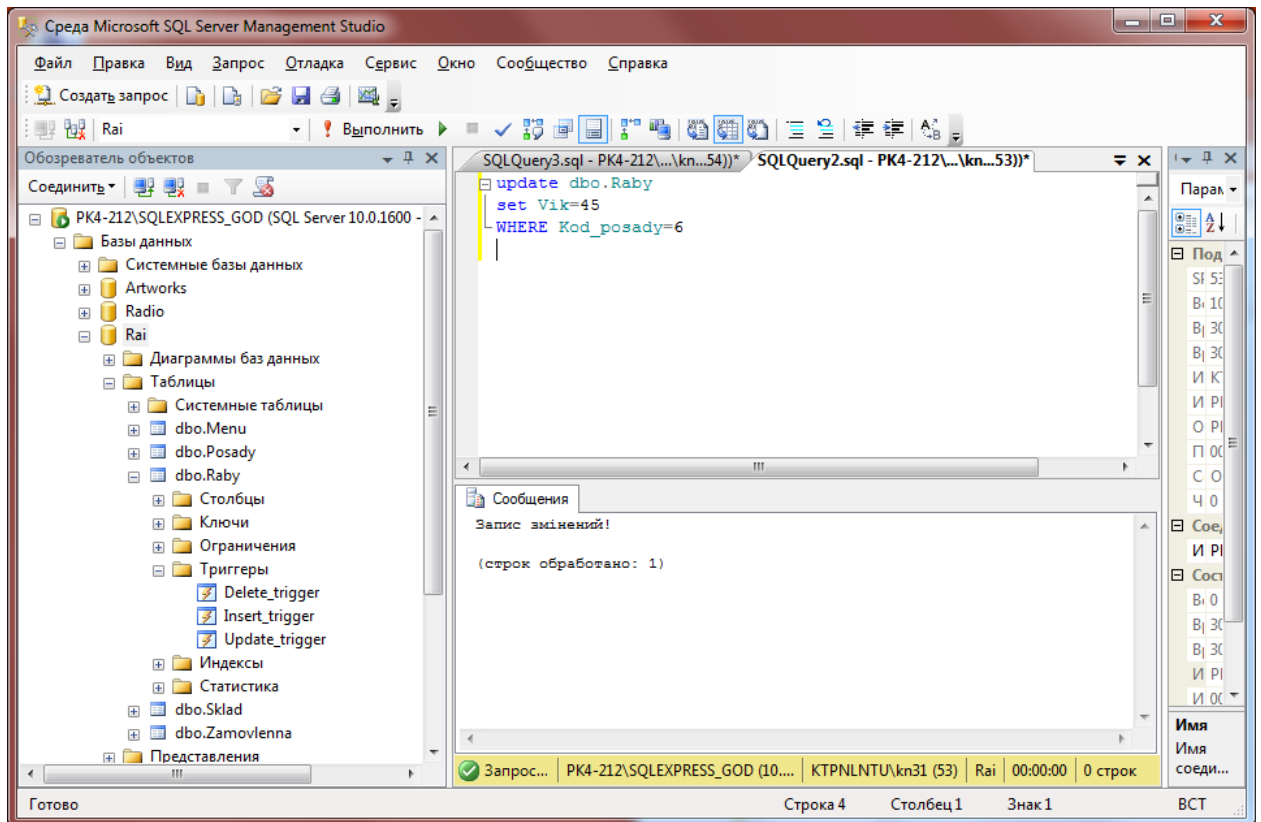


Рисунок 3.11 – Виконання тригера для редагування записів

3.3 Розробка візуального інтерфейсу користувачів системи

Для підвищення якості та швидкості роботи розробленої СУБД розроблено сім звичайних форм в Delphi 7 (кожна форма містить функції швидкого перегляду даних, що підвищує ефективність використання даної СУБД).

Підключення БД в Delphi 7 здійснюється за допомогою:

- 1) Встановлюємо всі компоненти на форму, які зображені на рис.3.12

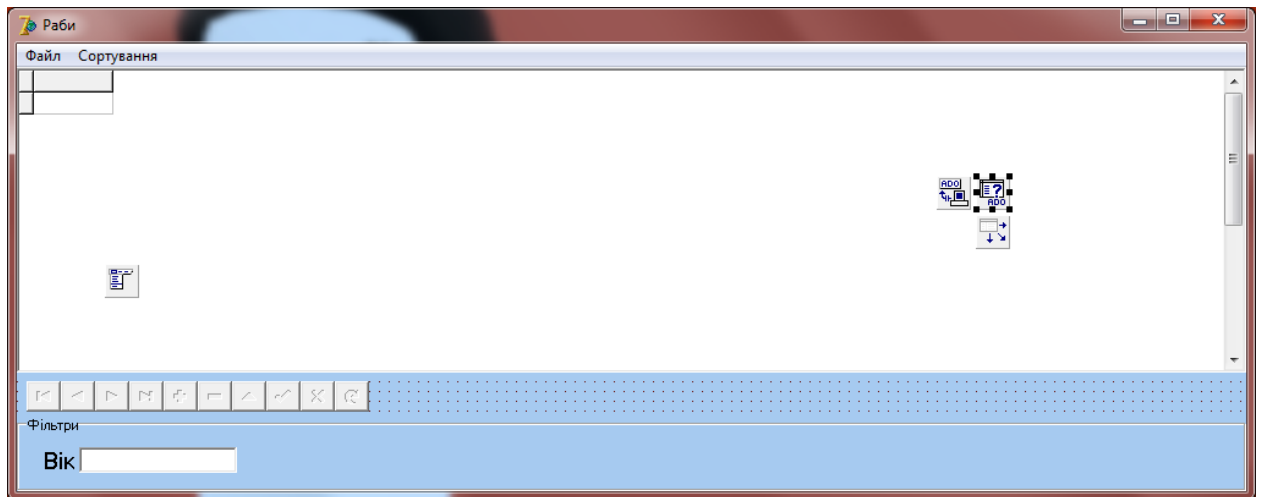


Рисунок 3.12 – Створена форма для таблиці «Раби»

- 2) В ADOConection зайти в властивості і вибрати тип підключення та натиснути далі;

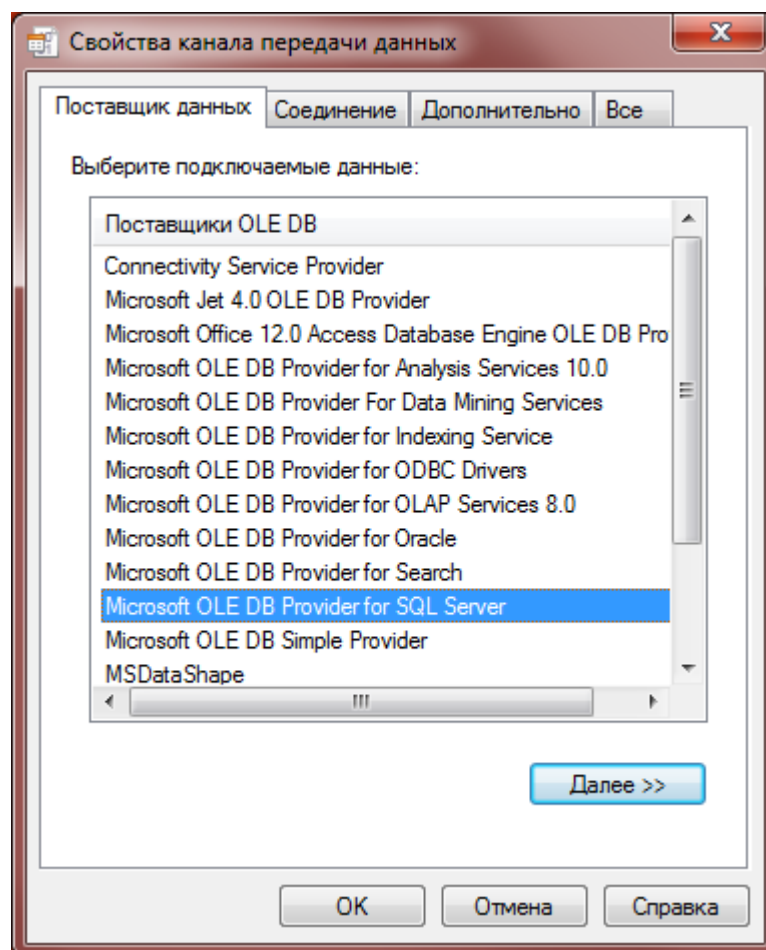


Рисунок 3.13 – Экспорт даних

3) Виберемо ім'я серверу, та базу даних з даного серверу;

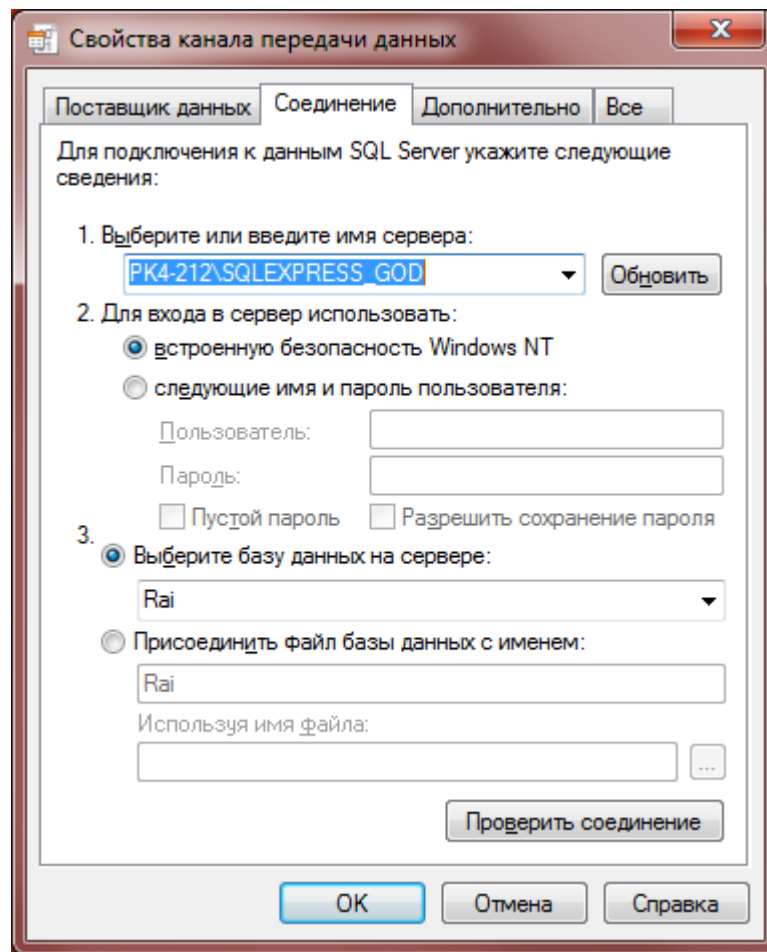


Рисунок 3.14 – Підключення БД

4) Натиснути «Перевірити підключення», якщо вдале, то натиснути «ОК»;

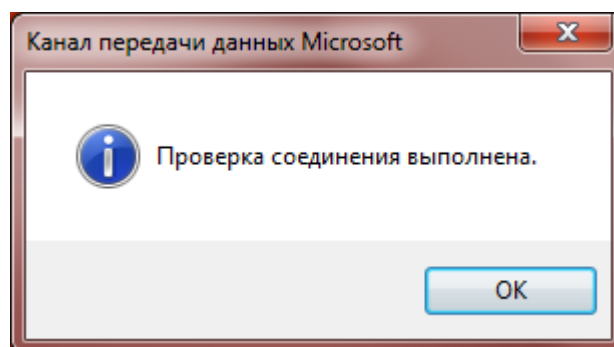


Рисунок 3.15 – Перевірка підключення

5) Зробимо з'єднання трьох компонентів між собою (ADONeconnection, ADOQuery, DataSource);

- 6) В компоненті ADOQuery в властивостях вибрати SQL та прописати запит `SELECT * FROM dbo.Cpivrobotniku;`

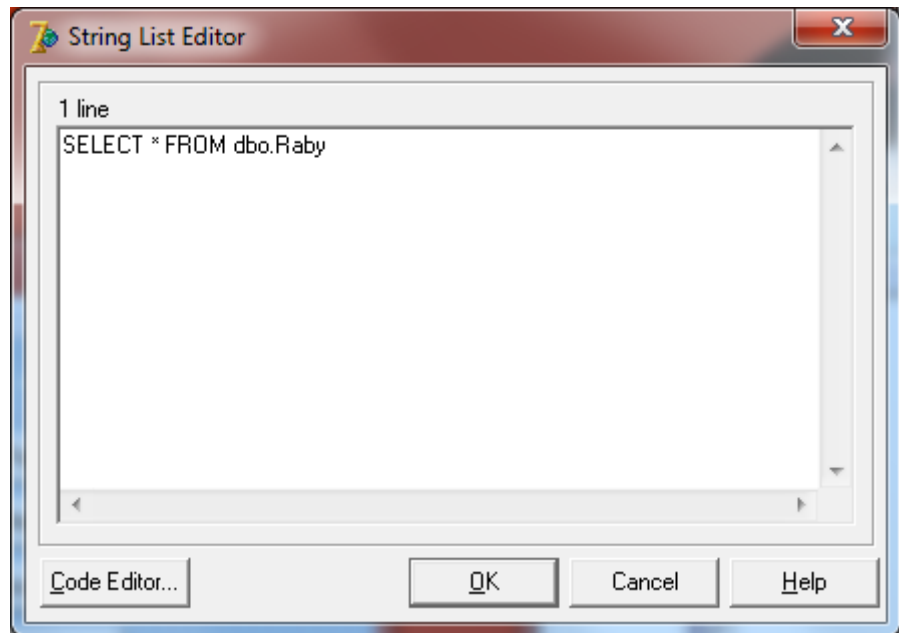


Рисунок 3.16 – Створений запит

- 7) Встановити значення Active в true, і база буде підключена

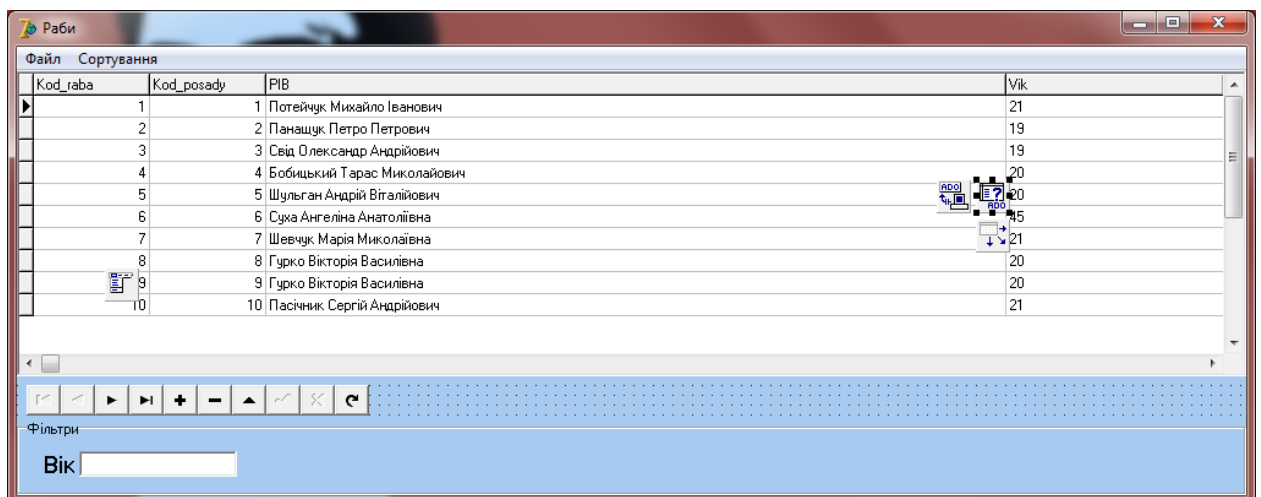


Рисунок 3.17 – Форма з підключеною БД

РОЗДІЛ 4 РЕЗУЛЬТАТИ ЗАСТОСУВАННЯ РОЗРОБЛЕНОЇ ПРОГРАМНОЇ СИСТЕМИ

4.1 Стислі відомості щодо розгортання системи

Для розробки даної бази даних використовувалося операційне середовище Windows 7 та microsoft sql server 2008.

Для створення таблиці виконуються наступні дії.

1. При запуску microsoft sql server 2008 відкривається вікно, в якому необхідно підключитися до сервера.

2. Створити БД і привласнити ім'я і натиснути "Создать".

3. У меню, що з'явилося, вибрати вкладку "Таблицы" і натиснути "Создать".

4. У вікні заповнити необхідні імена полів, встановити їх тип, властивості і опис (не обов'язково).

5. За допомогою кнопки на панелі інструментів або через вікно бази даних перейти в режим таблиць здійснити введення або коректування даних.

Для створення запитів виконується наступне.

1. Вибрати в меню "Запросы" і натиснути "Создать".

2. У вікні, що з'явилося, потрібно ввести SQL-запит.

4.2 Основні режими роботи із системою

Для підтвердження правильності роботи всіх форм нижче наведені рисунки вікон усіх форм.

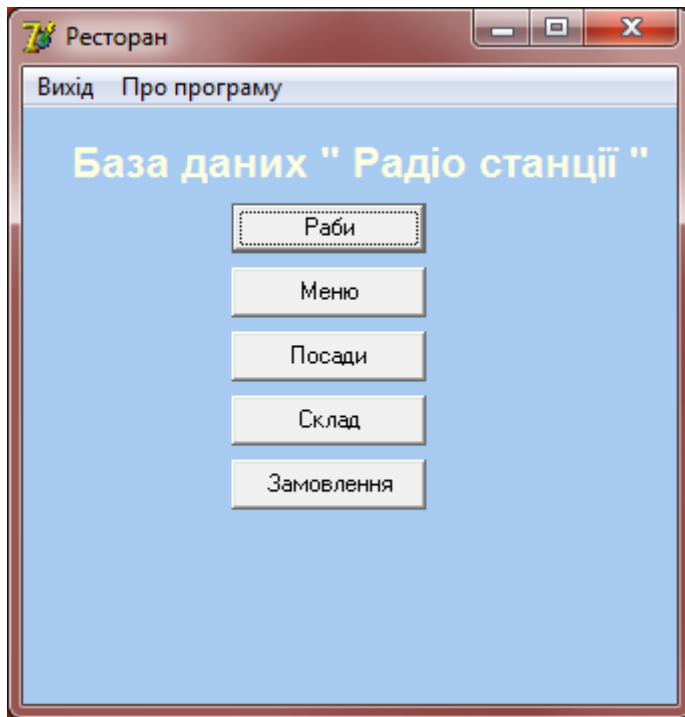


Рисунок 4.1 – Головне вікно програми, яке дозволяє здійснювати перехід між вікнами самих таблиць.

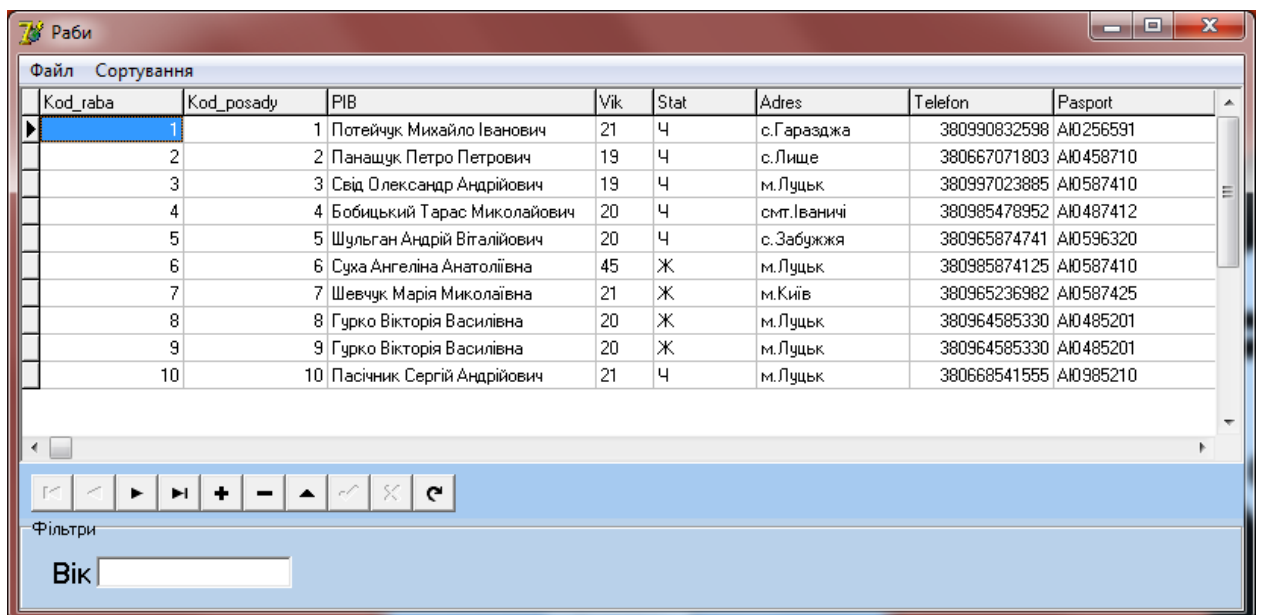


Рисунок 4.2 – Вікно програми «Раби» в якому можна здійснювати: додавання та видалення даних їх сортування та фільтрації.

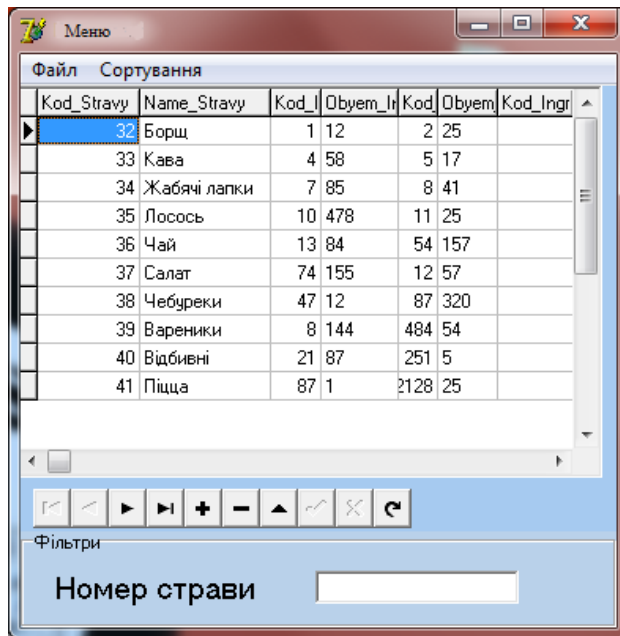


Рисунок 4.3 – Вікно програми «Меню» в якому можна здійснювати: додавання та видалення даних їх сортування та фільтрації.

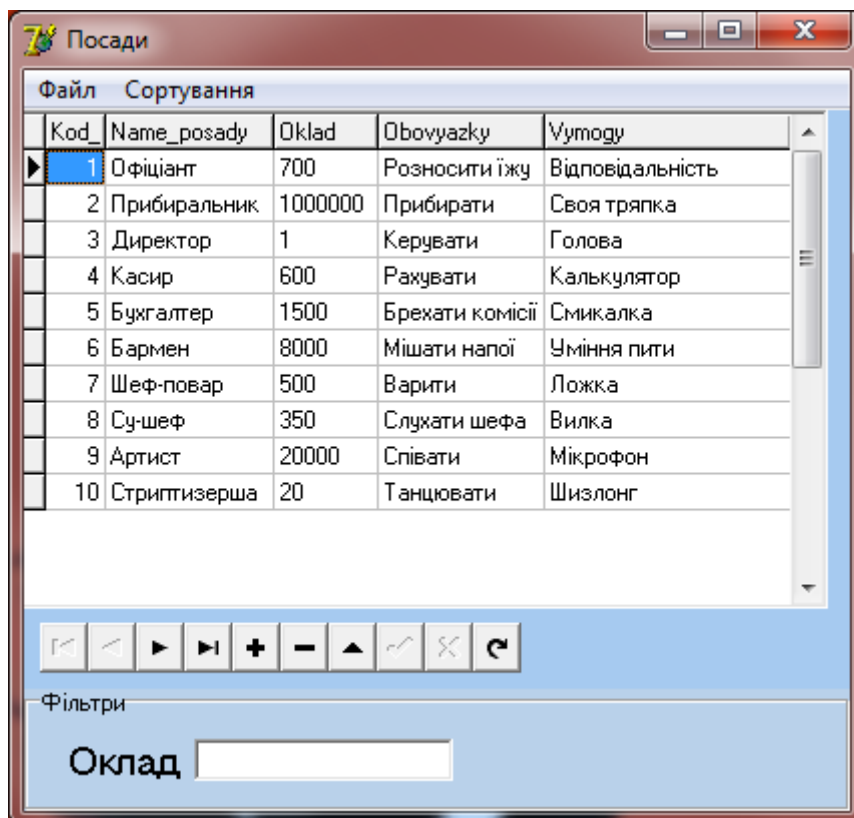


Рисунок 4.4 – Вікно програми «Посади» в якому можна здійснювати: додавання та видалення даних їх сортування та фільтрації.

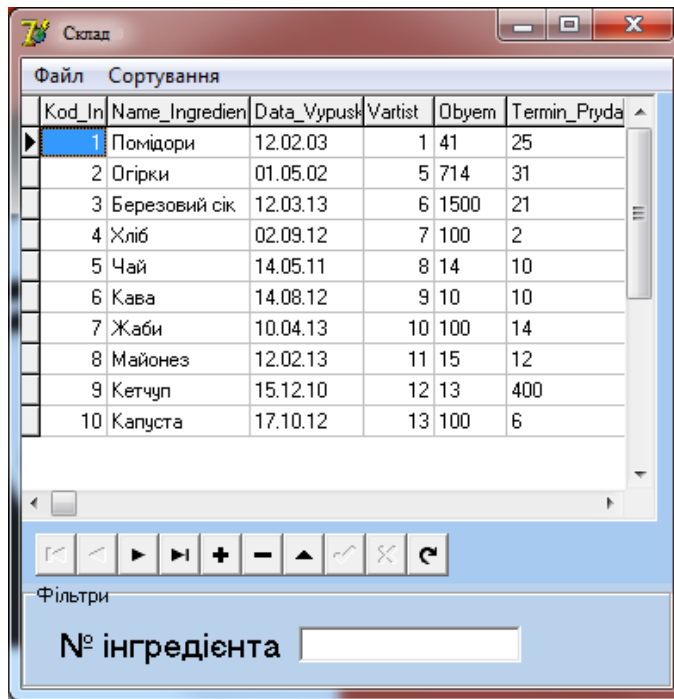


Рисунок 4.5 – Вікно програми «Склад» в якому можна здійснювати: додавання та видалення даних їх сортування та фільтрації.

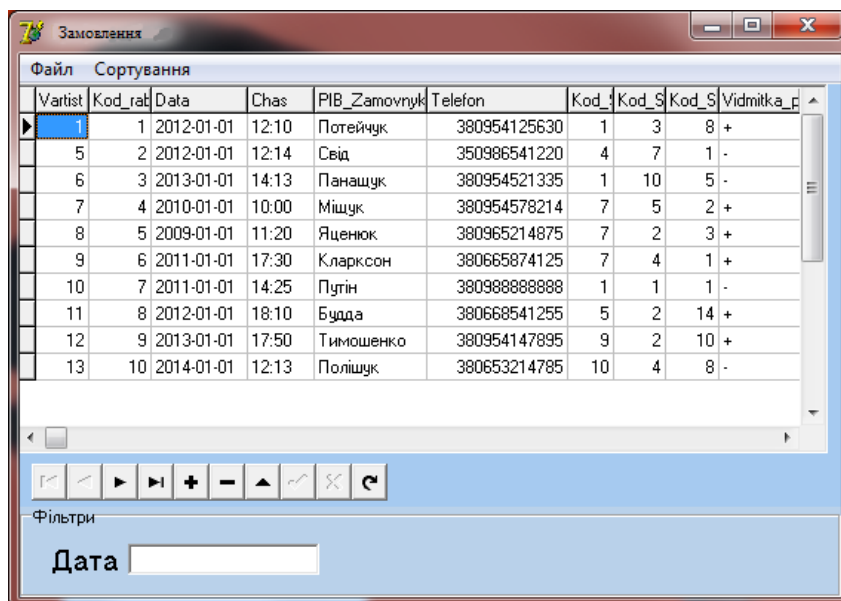


Рисунок 4.5 – Вікно програми «Замовлення» в якому можна здійснювати: додавання та видалення даних їх сортування та фільтрації.

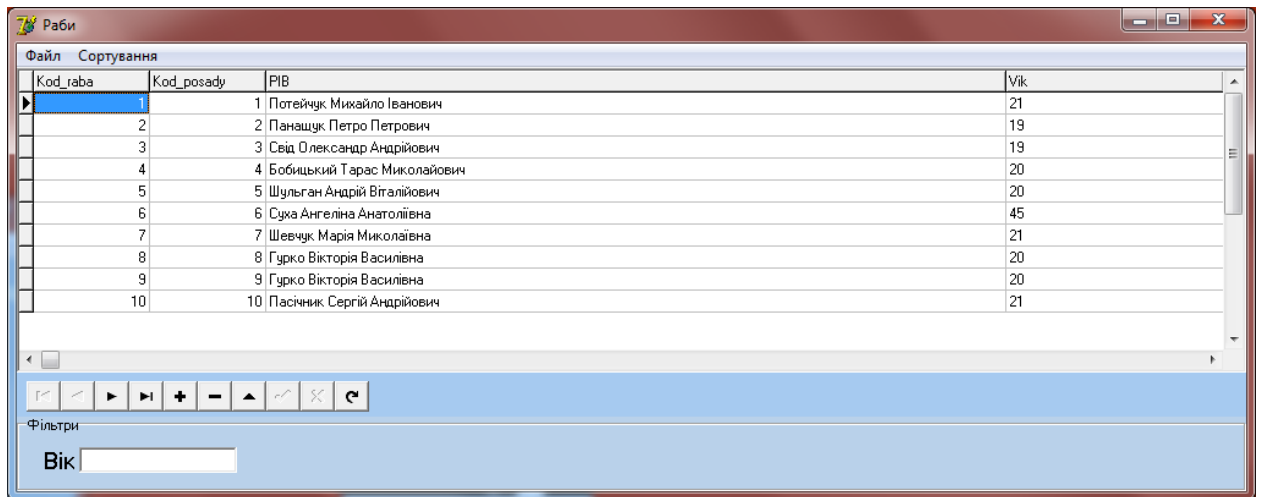


Рисунок 4.6 – Вікно програми «Раби» в якому можна здійснювати: додавання та видалення даних їх сортування та фільтрації.

ВИСНОВОК

Технічні характеристики сучасних засобів керування базами даних постійно зростають. Для забезпечення високих показників роботи необхідно правильно обрати СКБД відповідно задачі. Задачею цього курсового проекту було створення бази даних для Internet магазину Sport-Device. Для створення бази даних було обрано СКБД Microsoft SQL Server 2008 . В базі даних були реалізовані запити на додавання даних у таблицю, оновлення даних, видалення та вибір інформації з таблиць. Всі запити працюють надійно та видають чітку інформацію.

СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. Гарсиа-Молина Г., Ульман Д., Уидом Д. Системы баз данных. Полный курс.: Пер. с англ. - М.: Издательский дом "Вильямс", 2004. - 1088 с.
2. Дейт К. Дж. Введение в системы баз данных. :Пер. с англ. – 6-е изд. – К.: Диалектика, 1998. – 784 с.
3. Калянов Г.Н. CASE-технологии. Консалтинг в автоматизации бизнес-процессов. – 3-е изд. – М.: Горячая линия-Телеком,2002. - 320 с.
4. Карпова Т.С. Базы данных: модели, разработка, реализация. – СПб.: Питер, 2001. – 304 с.
5. Когаловский М.Р. Энциклопедия технологий баз данных. – М.: Финансы и статистика, 2002. – 800 с.
6. Конноли Т., Бегг К., Страчан А. Базы данных: проектирование, реализация и сопровождение. Теория и практика., 2-е изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2001. – 1120 с.
7. http://uk.wikipedia.org/wiki/Microsoft_SQL_Server
8. <http://www.microsoft.com/ru-ru/download/details.aspx?id=30438>
9. <https://www.google.com.ua>