МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ВНУТРІШНІХ СПРАВ
КАФЕДРА КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: **МОДЕЛЮВАННЯ ТА РОЗРОБКА ПІДСИСТЕМИ РОЗМЕЖУВАННЯ ДОСТУПУ ДО БАЗ ДАНИХ**

Виконав:
здобувач 2-го курсу магістратури
факультету № 2 ННІПК
Ростомов Артем Артурович

Науковий керівник:
викладач кафедри кібербезпеки
та інформаційного забезпечення
кандидат юридичних наук
Олена МЕЛЬНИКОВА

Рецензент:
доктор юридичних наук, доцент
Володимир ПЯДИШЕВ

Кваліфікаційна робота допущена до захисту
«__» _____ 2022 р., протокол № __.
Завідувач кафедри кібербезпеки та інформаційного забезпечення,
доктор юридичних наук, професор

Андрій БАБЕНКО

Одеса – 2022

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД – база даних

СКДБ – Система керування базами даних

IL – Intermediate Language

JIT – Just in time, тип компілятора

SQL – Structured query language, мова структурованих запитів

CLR – Common Language Runtime, віртуальна машина для виконання коду на мовах

NET CLI – Common Language Infrastructure, специфікація загальномовної інфраструктури

ОС – операційна система

IDE – Integrated Development Environment, середовище розробки

ВСТУП

Актуальність теми дослідження. Питання безпеки – невід'ємна частина концепції впровадження нових інформаційних технологій у всій сфері життя суспільства. Широкомасштабне використання обчислювальної техніки і телекомунікаційних систем у межах територіально-розподіленої мережі, збільшення обсягів інформації, яка обробляється, і розширення кола користувачів приводять до якісно нових можливостей несанкціонованого доступу до ресурсів і даних інформаційної системи.

На сьогодні кожна без виключення організація не може обійтися без використання баз даних. База даних (англ. database) – сукупність даних, організованих відповідно до концепції, яка описує характеристику цих даних і взаємозв'язки між їх елементами; ця сукупність підтримує щонайменше одну з областей застосування (за стандартом ISO/IEC 2382:2017 [1]). В загальному випадку база даних містить схеми, таблиці, подання, збережені процедури та інші об'єкти. Дані у базі організовують відповідно до моделі організації даних. Таким чином, сучасна база даних, крім саме даних, містить їх опис та може містити засоби для їх обробки [2].

Бази даних дуже важливий і цінний актив для будь-якої компанії. Оскільки в базі даних можуть зберігатися персональні або конфіденційні дані, варто дуже відповідально віднестися до їх захисту. Будь-які несправності в роботі СКБД і баз даних спричинити катастрофічні наслідки. Згідно з результатами сучасних досліджень, грошові збитки від порушення одного з властивостей «конфіденційність-цілісність-доступність» запису бази даних складає від $ 100 до $ 240. Видатки виділяються на відновлення втрачених даних, розслідування факту втрати даних, відновлення та ліквідацію збитків іміджу компанії і т.д. Тому проблема забезпечення безпеки баз даних є дуже актуальною [3].

Зловмисники зазвичай прагнуть отримати доступ до таких видів інформації, як внутрішня корпоративна інформація, персональні дані

співробітників, фінансова інформація, інформація про замовників/клієнтів, інтелектуальна власність, дослідження ринку/аналіз діяльності конкурентів, банківська та транзакційна інформація [4]. Ця інформація зазвичай зберігається в корпоративних сховищах фірм і базах даних різного обсягу [5].

Важливою складовою успіху діяльності будь-якої організації є цілісність, достовірність і доступність інформації. Ефективність механізмів захисту інформації в значній мірі залежить від реалізації ряду принципів. По-перше, механізми захисту доцільно проектувати одночасно з розробкою інформаційної системи, що дозволяє забезпечити їхню безконфліктність, своєчасну інтеграцію в обчислювальне середовище і скорочення витрат. По-друге, питання захисту варто розглядати комплексно в рамках єдиної системи захисту інформації. Системний підхід забезпечує адекватний багаторівневий захист інформації, що розглядається як комплекс організаційно-правових і технічних заходів. Крім того, при реалізації механізмів захисту повинні використовуватися передові, науково обґрунтовані технології захисту, що забезпечують необхідний рівень безпеки прийнятність для користувачів і можливість нарощування систем захисту інформації.

Управління безпекою баз даних охоплює широке коло питань, у число яких входять: забезпечення цілісності, конфіденційності і автентичності інформації, розмежування прав користувачів щодо до доступу до ресурсів бази даних, захист баз даних і її елементів від несанкціонованого доступу. Інформація, як сукупність знань про фактичні дані і залежності між ними, стала стратегічним ресурсом, вона – основа для прийняття будь-якого рішення. Тому, захист інформації як складна проблема в умовах впровадження сучасних інформаційних технологій. В базах даних, які створюються в органах державної влади і у комерційних структурах, циркулює інформація, що містить секретні відомості. Проблема безпеки інформації з погляду державних інтересів останнім часом набула особливої

актуальності і розглядається як одна із пріоритетних державних задач, як важливий елемент національної безпеки.

Зв'язок роботи з науковими програмами, планами, темами.

Кваліфікаційна робота виконувалась по кафедрі кібербезпеки та інформаційного забезпечення та науково-дослідній лабораторії з проблемних питань кримінального аналізу Одеського державного університету внутрішніх справ і, відповідно, до затвердженого напрямку наукових досліджень, а також до основних положень Стратегії сталого розвитку «Україна–2020», затвердженої Указом Президента України від 12.01.2015 № 5/2015 та Концепції розвитку сектору безпеки і оборони, затвердженої Указом Президента України від 14.03.2016 р.

Метою даної наукової роботи є модулювання та розробка підсистеми розмежування доступу до баз даних.

Об'єктом дослідження виступають процеси побудови інформаційних систем розмежування доступу до баз даних.

Предметом дослідження виступають методи та технології побудови інформаційних підсистем розмежування доступу баз даних MS Access.

Для досягнення вказаної мети необхідним є вирішення наступних взаємозалежних **задач**:

– провести аналіз моделей, методів та засобів побудови баз даних в інформаційно-комунікаційних системах;

– розглянути загальні підходи до побудови систем розмежування доступу до інформаційних ресурсів інформаційно-комунікаційної системи;

– розробити функціональну структуру інформаційної підсистеми;

– провести моделювання та проектування інформаційної підсистеми розмежування доступу до баз даних;

– вибрати технологію реалізації спроектованої інформаційної підсистеми;

– провести реалізацію спроектованої підсистеми розмежування доступу до баз даних в інформаційно-комунікаційній системі та навести тестовий приклад реалізації підсистеми;

– розглянити інструменти розробки системи підтримки прийняття рішень при аналізі проблеми інформаційного забезпечення безпеки даних.

Методи дослідження. Методологічна база роботи ґрунтується на комплексі сучасних методів дослідження. Застосовувалися такі методи наукового пізнання: історико-правовий, діалектичний, порівняльно-правовий, формально-логічний, системно-структурного аналізу та синтезу. Дослідження та обґрунтування основних понять, що використані в роботі, виконані за допомогою діалектичного методу пізнання, дедуктивного методу (аналізу) та індуктивного методу (синтезу) вказаних визначень. Історико-правовий метод застосовувався при узагальненні наукових досліджень і аналізу наукової літератури з обраної проблематики. Використання порівняльно-правового методу дозволило дослідити досвід в інших країнах.

Структура роботи. Робота складається зі вступу; переліку скорочень, умовних позначень; 4-х розділів, висновків та списку використаних джерел (34 найменування). Загальний обсяг дипломної роботи 79 сторінок.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ПОБУДОВИ СИСТЕМ РОЗМЕЖУВАННЯ ДОСТУПУ ДО БАЗ ДАНИХ

1.1 Аналіз моделей, методів та засобів побудови баз даних в інформаційно-комунікаційних системах

Інформатизація суспільства призвела до широкого використання інформаційних технологій та спричинило появу великих сховищ даних. З часом бази даних стали невід'ємною частиною інформаційних систем, що використовуються на підприємствах.

Розвиток суспільства та бажання швидкого отримання прибутків призвело до виникнення недобросовісної конкуренції. Основною метою якої є отримання інформації суперника та використання інформаційних ресурсів в своїх цілях шляхом несанкціонованого доступу до інформації. Таким чином виникла проблема захисту інформації в інформаційно-комунікаційних системах від недобросовісних конкурентів.

Аналіз інформаційних систем, що функціонують на підприємствах виявив, що останні використовують як сховище даних засіб Microsoft Office Access.

Microsoft Office Access – система управління базами даних від компанії Майкрософт, за стосунок, що входить до складу пакету офісних програм Microsoft Office. Має широкий спектр функцій, включаючи зв'язані запити, сортування по різних полях, зв'язок із зовнішніми таблицями і базами даних. Основні компоненти Microsoft Access: конструктор таблиць, конструктор екранних форм, конструктор SQL-запитів, конструктор звітів, що виводяться на друк. У системі Access є різні способи управління даними, а саме: система меню, панель інструментів, контекстне меню, укажчик миші, комбінації клавіш [6].

Система управління базами даних – це комплекс програмних засобів, призначений для інтегрованого зберігання та обробки даних. Система

управління базами даних Microsoft Access відноситься до реляційних баз даних.

Перевагою системи управління базами даних Access є те, що вона має прості та зручні засоби обробки кількох таблиць у одній базі даних. Таблиця є основним об'єктом бази даних. У одній базі даних зберігається кілька таблиць та засоби зв'язування таблиць. Система управління базами даних має значну кількість спеціальних програм - “майстрів”. Є майстер таблиць, майстер кнопок, майстер форм. Майстри здійснюють діалог з користувачем, у процесі якого визначаються дані, необхідні для розв'язування відповідної задачі. Для зручності роботи кожен майстер має певні етапи (кроки). Будь-який етап можна пропустити або звернутись до попередніх.

Користувач може управляти формою видачі даних на екран. Важливо правильно конструювати форми, оскільки саме з ними працює користувач при введенні і редагуванні записів бази даних. Крім того, форми можна використовувати для збирання та виведення інформації. Форми - основний засіб для організації інтерфейса користувача в додатках Microsoft Access.

Більшу частину роботи в Access можна виконати користуючись введенням і переглядом даних в окремих формах. Проте є можливість за допомогою командних кнопок зв'язати форми між собою щоб створити додаток бази даних.

Основна перевага командних кнопок - вони дозволяють дуже просто запускати макрос. Цей макрос може не робити нічого, крім відкриття іншої форми, друку звіту або виконання запиту - дії, яка поновлює більшість записів в базі даних. За допомогою макросів Access можна виконати понад 40 різних команд. Використовуючи форми, звіти, макроси та найпростіші процедури Access Basic можна досить просто побудувати складний додаток Access.

База даних – це інтегроване сховище взаємопов’язаних даних конкретної предметної області. Логічна структура баз даних складається з

таких об’єктів: таблиць, запитів, форм, звітів, макросів та модулів. Доступ до цих об’єктів відбувається за допомогою відповідних вкладинок вікна Access.

«Таблиці» – основний об’єкт бази даних, оскільки в ньому зберігається вся інформація. «Запит» - один з потужних об’єктів Microsoft Access, який дозволяє ефективно представити інформацію, що містять в таблиці, з певними властивостями, тобто він служить для фільтрації або селекції даних. Дає змогу вибрати із бази даних необхідну інформацію, яка відповідатиме певним критеріям і далі використовуватиметься для розв’язання конкретного завдання. «Форма» - це певний бланк для заповнення його даними або маска, що ніби накладається на базу даних. Головне призначення форми – спростити процес заповнення бази даних. «Звіти» - це певним чином оформлена програма, що дає змогу видавати для друкування або на екран інформацію з бази даних. «Макроси» використовують для автоматизації процесу виконання операцій, які часто повторюються.

«Модулі» – це програмні модулі,написані мовою VBA для розв’язання складних завдань.

Також в Access можна аналізувати записи бази даних за допомогою графіків, кругових, лінійних та інших діаграм. Діаграми допомагають ефективно донести до користувачів відомості, які містяться у базі даних. Корисним засобом Access є майстер діаграм, який допомагає створити та вставити в форму діаграму.

Існує декілька засобів для побудови баз даних. Наприклад, Oracle, Foxpro SQL Server, Paradox.

База даних Oracle (Oracle DB) - це система управління реляційними базами даних (RDBMS) від корпорації Oracle. Спочатку розроблений у 1977 році Лоуренсом Еллісоном та іншими розробниками, Oracle DB є одним з найбільш надійних і широко використовуваних реляційних двигунів баз даних. Система побудована на базі реляційних баз даних, в яких користувачі (або передній кінець програми) можуть безпосередньо отримувати доступ до об'єктів даних через структуровану мову запитів (SQL). Oracle - це повністю

масштабована архітектура реляційних баз даних і часто використовується глобальними підприємствами, які керують та обробляють дані в широких і локальних мережах. База даних Oracle має власний мережевий компонент, який дозволяє здійснювати зв'язок по мережах [7].

Система управління базами даних Oracle – об'єктно-реляційна система управління базами даних компанії Oracle. Дозволяє підтримувати зв'язок не тільки між клієнтом і сервером, але і між серверами. Побудова баз даних відкриває можливості для рішення цілого комплексу задач: зібрати в одне ціле дані, що зберігаються в різних місцях, збільшити серверну потужність системи, зосередити дані в безпосередній близькості від їх споживачів, зберігаючи при цьому цілісність системи. Концепція побудови баз даних в Oracle заснована на децентралізованій їх організації. Сервери взаємодіють один з одним за допомогою протоколу SQL*Net. Посилання один на одного – так звані канали зв'язку бази даних – сервери зберігають їх якості об'єктів бази даних. В свою чергу повне ім'я об'єкту може включати в себе канал зв'язку, це, безумовно, вимагає забезпеченості унікальності імен серверів в мережі, що досягається за допомогою ієрархічної організації доменів, подібної існуючої в Internet. Оскілки замість «справжніх» імен об'єктів можна використовувати їх синоніми, то додатки клієнта може не знати, чи є даний об'єкт локальним для сервера, з яким встановлений зв'язок, чи ні. Звичайно, механізм каналів зв'язку та синонімів – лише зовнішня сторона тієї системи, яка дозволяє зробити структуру бази даних Oracle абсолютно прозорої для додатків незалежно від розміщення даних і режиму взаємодії серверів [8].

SQL Server – це система управління базами даних, яка повинна працювати на сервері мережі, отримуючи підключення від видалених користувачів та додатків. Можна підключитися до SQL Server локально, з того ж комп'ютера, на якому виконується SQL Server, але у виробничих системах баз даних, як правило, ця можливість не використовується. Отже,

дуже важливо правильно конфігурувати SQL Server для отримання захищених підключень від видалених комп'ютерів.

При установці SQL Server із параметрами за замовчуванням багато функцій відключені, щоб зменшити вразливість системи баз даних проти атак. Сервери баз даних повинні бути добре захищені від несанкціонованого зовнішнього доступу. Якщо необхідно надати доступ до SQL Server через інтернет користувачам або додаткам, слід гарантувати, що мережеве оточення має адекватні механізми захисту, такі, як між мережевий екран або система виявлення вторгнень. В режимі перевірки справжності SQL Server можна створювати імена входу SQL Server та управляти ними. При створенні імені входу SQL Server необхідно задати для цього імені входу пароль. Користувачі повинні вказувати пароль при з'єднанні із екземпляром SQL Server.

Нижче наведені кроки, які ми повинні виконати, щоб встановити SQL Server в операційну систему.

Крок 1. Перейдіть до https://www.microsoft.com/en-gb/sql-server/sql-server-downloads.

Крок 2. Завантажте версію розробника.

Крок 3. Після завантаження налаштування відкрийте його, натиснувши двічі та виберіть Завантажити медіа.

Крок 4. Виберіть файл ISO, виберіть шлях та натисніть Завантажити.

Крок 5. Витягніть завантажений файл і відкрийте папку.

Крок 6. Клацніть на setup.exe і натисніть на перше посилання.

Крок 7. Вам потрібно буде кілька разів натиснути наступну кнопку.

Крок 8. Нарешті, прийняти цей термін і вибрати встановити єдиний варіант.

Крок 9. Почекайте приблизно 10-15 хвилин, щоб встановити його.

Крок 10. Перейдіть на https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017.

Крок 11. Завантажте студію управління SQL-сервером та запустіть налаштування.

Крок 12. Після його встановлення перезавантажте систему [9].

За допомогою цих простих кроків ми можемо завантажити та налаштувати SQL в нашій операційній системі. Після його встановлення ми зможемо використовувати його для зберігання та обробки даних. У наступному розділі ми побачимо, як ми можемо працювати з цим після його успішного встановлення.

Visual FoxPro – об’єктно-орієнтована, візуально-програмована мова, керована по подіям, яка в повній мірі відповідає новим вимогам, що пред'являються до сучасних засобів проектування і реалізації програмного забезпечення. Visual FoxPro можуть застосовувати користувачі різних рівнів підготовки. Якщо ви тільки починаєте освоювати Visual FoxPro і вам необхідно в найкоротші терміни розробити простий додаток, вам допоможуть майстра створення баз даних, таблиць, програм, форм, звітів і багато інших. Для розробки великих і складних проектів служать відповідні конструктори і об'єктно-орієнтована мова, який дозволяє не тільки використовувати базові класи Visual FoxPro і зовнішні бібліотеки, але і створювати користувальницькі класи.

Paradox – реляційна система управління базами даних, розроблена компанією Corel. Входить в пакет WordPerfect Office. Серед численних особливостей Paradox виділяють унікальне поєднання надзвичайної простоти з величезними можливостями функціонально завершеної системи управління даними(в цьому і є парадокс). Як результат парадоксального сполучення – найпотужніша система управління базами даних підчиняється не тільки професійному програмісту, але і користувачеві, який не має ні найменшого уявлення про програмування обробки інформації на комп’ютері. Paradox надає широкий вибір способів зберігання, відображення і подання даних. Компоненти, які використовуються для зберігання та надання даних,

називаються об’єктами. В системі Paradox існують наступні об’єкти: таблиця, форма, звіт, запит, програма, бібліотека програм [10].

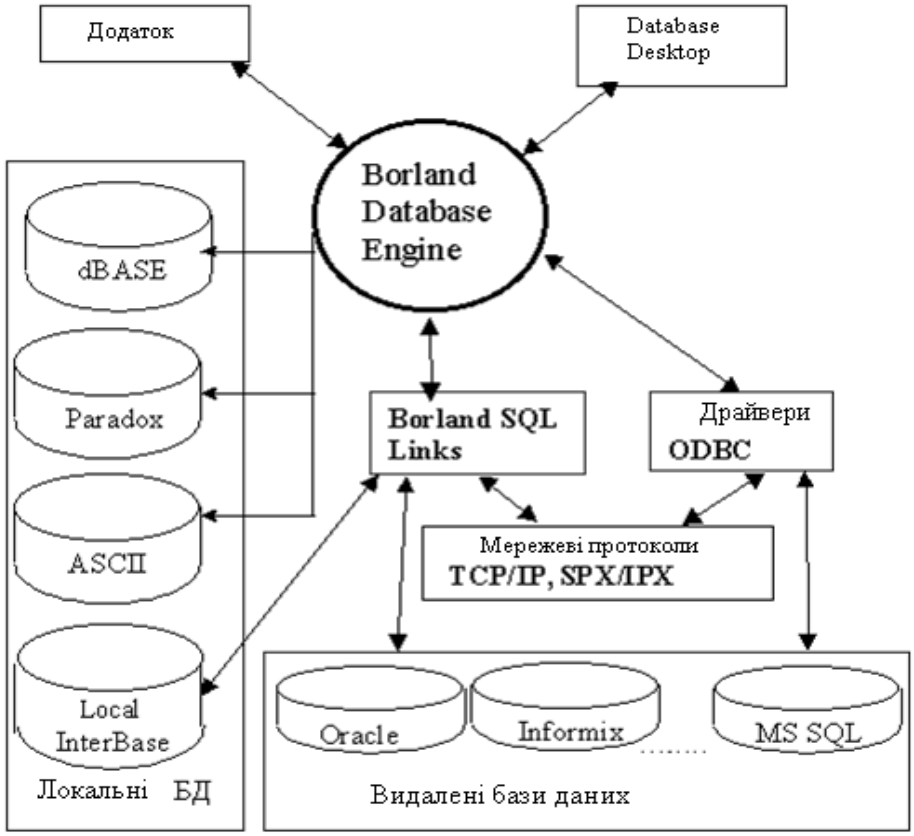

Рисунок 1.1. Механізм організації доступу до баз даних

На рисунку 1.1 зображений механізм організації доступу до баз даних за допомогою середи візуального програмування Delphi. Delphi є візуальним інтегрованим середовищем розробки компонентно-заснованих додатків, що забезпечують швидку розробку Microsoft Windows додатків з мінімум кодування. Бібліотека класів Delphi багато в чому ховає складність Windows програмування і робить можливою розробку простих додатків навіть новачкам і любителям. Однак Delphi це і дуже потужний інструмент професійного програмування, що дозволяє створювати складні додатки, як колективу розробників, так і незалежного розробнику. Робота з даними Delphi здійснюється через BDE , яка забезпечує безпосередній зв'язок з локальними базами даних і використовується при організації доступу до віддалених серверів. Використання BDE дозволяє додатку здійснювати доступ до даних не тільки локальних (PARADOX і dBASE) [11], але і

віддалених баз даних, розташованих на SQL - серверах (InterBase, Sybase, Oracle, MS SQL Server) , а також у будь-яких форматах, доступних через драйвери ODBC [12].

Microsoft Visual Basic – засіб розроблення програмного забезпечення, створений і підтримуваний корпорацією Microsoft, який складається з мови програмування і середовища розроблення. Мова Visual Basic успадкувала дух, стиль і, частково, синтаксис свого предка – мови Бейсік, у якої є чимало діалектів. У той же час Visual Basic поєднує в собі процедури та елементи об'єктно-орієнтованих та компонентно-орієнтованих мов програмування. Середовище розробки Visual Basic містить інструменти для візуального конструювання користувацького інтерфейсу [13], [14].

Visual Basic вважається потужним засобом швидкої розробки прототипів програми, розробки застосунків баз даних і взагалі для компонентного способу створення програм, що працюють під управлінням операційних систем родини Microsoft Windows.

Перше визнання серйозними розробниками Visual Basic отримав після виходу версії 3 – Visual Basic 3. Остаточне визнання як повноцінного засобу програмування для Windows – при виході версії 5 – Visual Basic 5. Версія Visual Basic 6, що входить до складу Microsoft Visual Studio 6.0, стала по-справжньому зрілим і функціонально багатим продуктом. Після цього розробники з Microsoft суттєво змінили напрямок розвитку даної технології.

Visual Basic.NET не дозволяє програмувати по-старому, бо по суті є зовсім іншою мовою, такою ж, як і будь-яка інша мова програмування для платформи .NET. Індивідуальність мови і її переваги (простота, природність створення програм, легкість використання готових компонент) при використанні в середовищі .NET не мають такого значення, як раніше – усе зосереджено на можливостях самої системи .NET, на її бібліотеці класів. Тому сьогодні треба говорити про класичний Visual Basic, його діалекти Visual Basic for Applications і Visual Basic Scripting Edition, і про мову для платформи .NET – Visual Basic.NET[15].

Але, можна сказати, що пакет Microsoft Office Access самий простий засіб для побудови баз даних. Цей пакет дозволяє розв’язувати широке коло завдань користувачів без програмування і доступна для широкого кола непрофесійних користувачів персональних комп’ютерів.

1.2. Загальні підходи до побудови систем розмежування доступу до баз даних

На основі можливостей апаратного забезпечення та фінансового забезпечення, прийнято рішення використовувати спосіб аутентифікації та ідентифікації за допомогою знання претендентом інформації, яку знає тільки легальний користувач – паролю та логіну. Ідентифікація - привласнення суб'єктам або об'єктам доступу ідентифікатора або порівняння пред'явленого ідентифікатора з переліком привласнених ідентифікаторів. Ідентифікація об'єкта – це його впізнання, ототожнення із чим-небудь. Якщо ж говорити про області інформаційних технологій, то даний термін звичайно означає встановлення особистості користувача. Цей процес необхідний для того, щоб система надалі змогла ухвалити рішення щодо видачі людині дозволу для роботи на комп'ютері, доступу до закритої інформації тощо. Таким чином, ідентифікація є одним з основних понять в інформаційній безпеці.

Сьогодні існує декілька способів ідентифікації користувачів. У кожного з них є свої переваги і недоліки, завдяки чому деякі технології підходять для використання в одних системах, інші - в інших.

Ідентифікація – привласнення суб’єктам або об’єктам доступу ідентифікатора або порівняння пред’явленого ідентифікатора з переліком привласнених ідентифікаторів. Ідентифікація об’єкта - це його впізнання, ототожнення із чим-небудь. Якщо ж говорити про області інформаційних технологій, то даний термін звичайно означає встановлення особистості користувача. Цей процес необхідний для того, щоб система надалі змогла ухвалити рішення щодо видачі людині дозволу для роботи на комп’ютері,

доступу до закритої інформації тощо. Таким чином, ідентифікація є одним з основних понять в інформаційній безпеці [16].

Існує три найпоширеніших види ідентифікації:

- парольна ідентифікація. Ще не дуже давно парольна ідентифікація була чи ледве не єдиним способом визначення особистості користувача. Справа в тому, що парольна ідентифікація найбільш проста як у реалізації, так й у використанні. Суть її зводиться до наступного. Кожен зареєстрований користувач системи одержує набір персональних реквізитів (звичайно використаються пари: логін-пароль). Далі при кожній спробі входу людина повинна вказати свою інформацію. Оскільки вона унікальна для кожного користувача, то на підставі її система й робить висновок про особистість та ідентифікує. Недоліком парольної ідентифікації є значна залежність надійності ідентифікації від користувачів, точніше від обраних ними паролів. Справа в тому, що більшість людей використовують ненадійні ключові слова, які легко підбираються;

- апаратна ідентифікація. Цей принцип ідентифікації ґрунтується на визначенні особистості користувача за певним предметом, ключем, що перебуває в його ексклюзивному користуванні. Мова йде про спеціальні електронні ключі. На даний момент найбільше поширення одержали два типи пристроїв. До першого ставляться всілякі карти. їх досить багато, і працюють вони за різними принципами. Так, наприклад, досить зручні у використанні безконтактні карти (їх ще називають проксіміті-карти), які дозволяють користувачам проходити ідентифікацію як у комп'ютерних системах, так й у системах доступу в приміщення. Найбільш надійними вважаються старт-карти – аналоги звичних багатьом людям банківських карт. Іншим типом ключів, які можуть використатися для апаратної ідентифікації, є так звані токени. Ці пристрої мають власну захищену пам'ять і підключаються безпосередньо до одного з портів комп'ютера. Головним достоїнством застосування апаратної ідентифікації є досить висока надійність. У пам'яті токенів можуть зберігатися ключі, підібрати які хакерам

не вдасться. Крім того, у них реалізовано чимало різних захисних механізмів. А вбудований мікропроцесор дозволяє електронному ключу не тільки брати участь у процесі ідентифікації користувача, але й виконувати деякі інші корисні функції.

Недоліком апаратної ідентифікації є висока ціна. Взагалі ж останнім часом вартість як самих електронних ключів, так і програмного забезпечення, що може працювати з ними, помітно знизилася. Проте для введення в експлуатацію системи майнової ідентифікації однаково будуть потрібні деякі вкладення. Все-таки кожного зареєстрованого користувача потрібно забезпечити персональними токенами. Крім того, згодом деякі типи ключів можуть зношуватися або можуть бути загублені користувачами;

- біометрична ідентифікація. Біометрія - це ідентифікація людини за унікальними, властивими тільки їй біологічними ознаками. Сьогодні експлуатується вже більше десятка різних біометричних ознак. Причому для найпоширеніших з них (відбитки пальців і райдужна оболонка ока) існує безліч різних за принципом дії сканерів. Так що користувачам, що вирішили використати біометричну ідентифікацію, є із чого вибрати. Головною перевагою біометричних технологій є найвища надійність. І дійсно, усі знають, що двох людей з однаковими відбитками пальців у природі просто не існує. Основним недоліком біометричної ідентифікації є вартість устаткування. Адже для кожного комп'ютера, що входять до цієї системи, необхідно придбати власний сканер. Варто також відзначити, що подібні дешеві сканери недовговічні. Крім того, у них досить високий відсоток помилок другого роду (відмова в доступі зареєстрованому користувачеві). Тому користувачеві доводиться вибирати, який пристрій придбати - дорожчий й кращий або дешевший й гірший.

Аутентифікацією – називається процедура верифікації належності ідентифікатора суб'єкту. Аутентифікація здійснюється на основі того чи іншого секретного елемента (аутентифікатора), який є у розпорядженні як суб'єкта, так і інформаційної системи. Звичайно, інформаційна система має в

розпорядженні не сам секретний елемент, а деяку інформацію про нього, на основі якої приймається рішення про адекватність суб'єкта ідентифікатору. Наприклад, перед початком інтерактивного сеансу роботи більшість операційних систем запитують у користувача його ім'я та пароль.

Введене ім'я є ідентифікатором користувача, а його пароль – аутентифікатором. Операційна система зазвичай зберігає не сам пароль, а його хеш-суму, що забезпечує складність відновлення пароля. В інформаційних технологіях використовуються такі методи аутентифікації:

- однобічна аутентифікація, коли клієнт системи для доступу до інформації доводить свою аутентичність;

- двобічна аутентифікація, коли, крім клієнта, свою аутентичність повинна підтверджувати і система (наприклад, банк);

- трибічна аутентифікація, коли використовується так звана нотаріальна служба аутентифікації для підтвердження достовірності кожного з партнерів в обміні інформацією.

Загальновизнаними є три завдання аутентифікації:

– аутентифікація користувача (встановлення достовірності користувача, якому потрібний доступ до інформації, що захищається, або який хоче підключитися до мережі);

– аутентифікація даних (перевірка того, що масив даних не був змінений протягом часу, коли він був поза контролем);

– аутентифікація повідомлення (встановлення достовірності повідомлення, яке один абонент посилає іншому по відкритому каналу зв'язку).

Перш ніж дістати доступ до конфіденційного інформаційного ресурсу системи, претендент повинен довести своє право на такий доступ. Зазвичай для цього він спочатку посилає системі свій ідентифікатор (ім'я користувача або реєстраційний номер), а потім посилає доказ свого права на доступ. Таким чином, система перевіряє, чи є претендент тим за кого він себе видає [17].

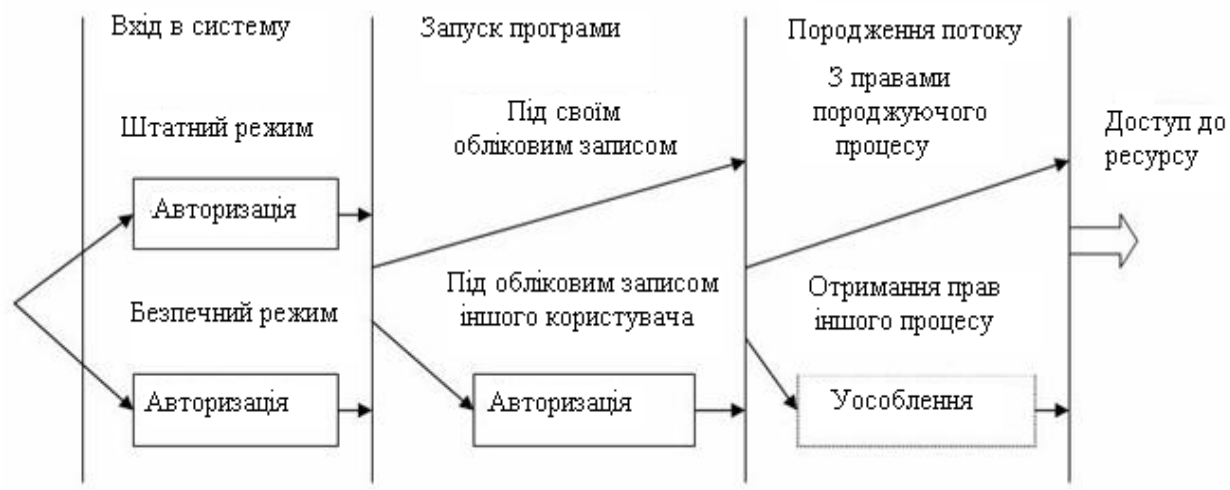

Рисунок 1.2. Етапи ідентифікації та аутентифікації користувача, реалізовані в системі

Перший крок ідентифікації, підтримуваний режимом аутентифікації, реалізується при вході користувача в систему. Тут слід виділити можливість входу в штатному і в безпечному режимі. У порядку зауваження зазначимо, що принциповою відмінністю безпечного режиму є те, що при запуску системи в безпечному режимі можна відключити завантаження сторонніх по відношенню до системи драйверів та програм. Тому, якщо в системі використовується додаткова система захисту інформації від несанкціонованого доступу, можна спробувати завантажити систему в безпечному режимі без компонент системи захисту інформації від несанкціонованого доступу, тобто без засобів захисту. З урахуванням того, що завантажити систему в безпечному режимі може будь-який користувач, то система захисту інформації від несанкціонованого доступу повинна забезпечувати можливість входу в систему в безпечному режимі (після ідентифікації та аутентифікації) тільки під обліковим записом адміністратора.

Другий крок полягає в запуску користувачем процесів, які, у свою чергу, породжують потоки (саме потоки в загальному випадку і здійснюють звернення до ресурсів). Всі працюючі в системі процеси виконуються в контексті захисту того користувача, від імені якого вони так чи інакше були запущені. Для ідентифікації контексту захисту процесу або потоку

використовується об'єкт, званий маркером доступу. В контекст захисту входить інформація, що описує привілеї, облікові записи користувачів і групи, зіставлені з процесом і потоком. При реєстрації користувача в системі створюється початковий тег, який представляє користувача, який входить в систему, і порівнює його з процесом оболонки, що застосовується для реєстрації користувача. Всі програми, що запускаються користувачем, успадковують копію цього маркера.

У загальному вигляді рішення задачі має полягати в наступному. При запиті доступу до ресурсу повинні виявлятися факти події уособлення (відповідно, суб'єктом доступу тут виступає процес, для якого аналізується наявність уособлюючого маркера доступу) і перевірятися їх коректність у відповідності з заданими дозволами (заборонами). Очевидно, що перевірка прав суб'єкта доступу до ресурсу повинна здійснюватися вже після перевірки коректності його ідентифікації.

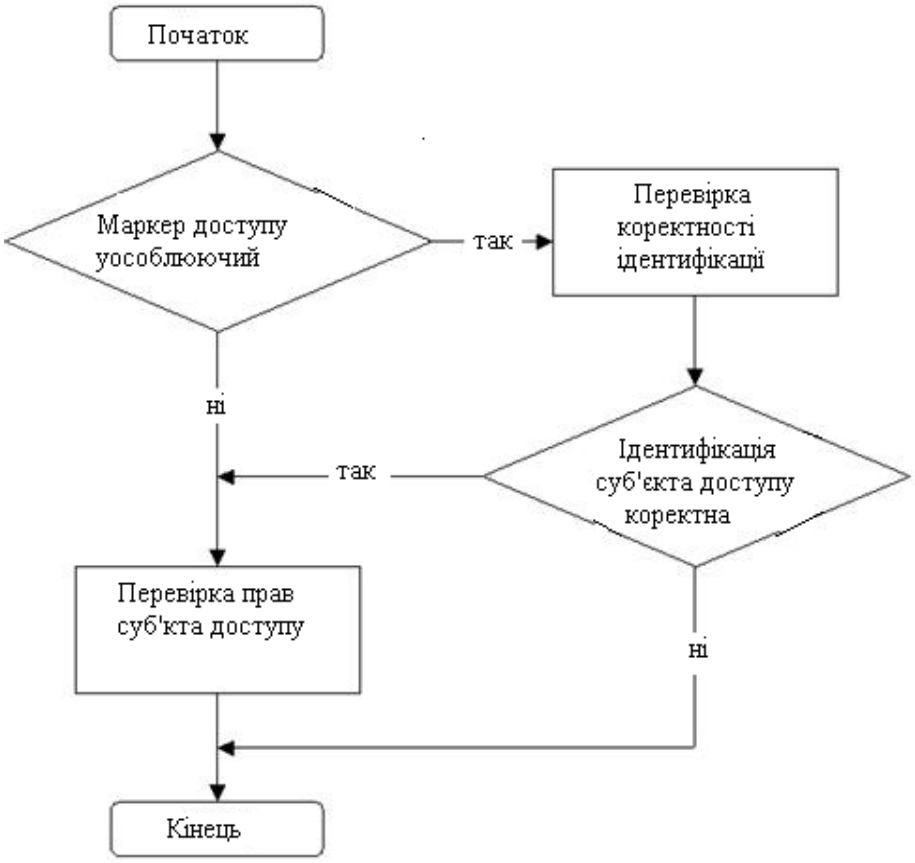

Рисунок 1.3. Укрупнений алгоритм ідентифікації та аутентифікації при запиті доступу до ресурсу

Таким чином, як суб'єкт доступу виступає процес (в тому числі, це обумовлюється тим, що різні процеси (програми) можуть зажадати і різних правил дозволених (заборонених) уособлень, що неможливо забезпечити, якщо в якості суб'єкта доступу прийняти користувача - обліковий запис).

При великій кількості користувачів традиційні підсистеми управління доступом стають вкрай складними для адміністрування. Кількість зв'язків у них пропорційно добутку кількості користувачів на кількість об'єктів. Необхідні рішення в об'єктно-орієнтованому стилі, здатні цю складність зменшити. Таким рішенням є рольове управління доступом. Суть його в тому, що між користувачами та їх привілеями з'являються проміжні суті - ролі. Для кожного користувача одночасно можуть бути активними кілька ролей, кожна з яких дає йому певні права.

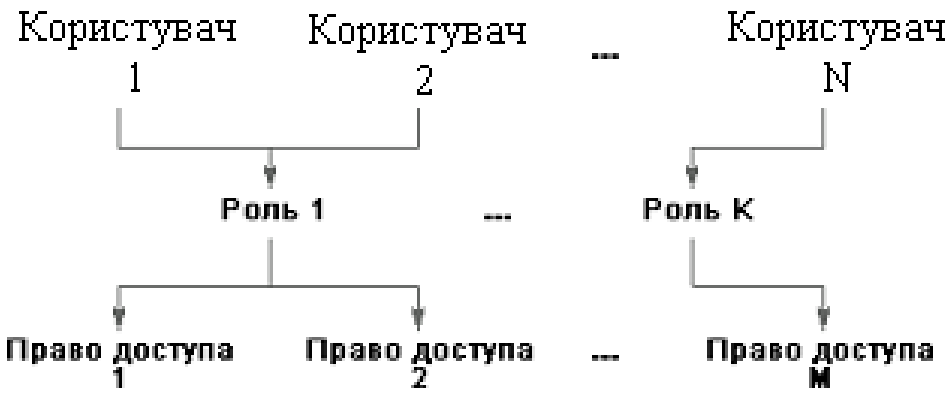

Рисунок 1.4. Користувачі, об’єкти, ролі

Рольовий доступ нейтральний по відношенню до конкретних видів прав і способів їх перевірки; його можна розглядати як об'єктно-орієнтований каркас, який полегшує адміністрування, оскільки він дозволяє зробити підсистему розмежування доступу керованою при як завгодно великому числі користувачів, передусім за рахунок встановлення зв'язків між ролями, аналогічних спадкоємству в об'єктно-орієнтованих системах. Крім того, ролей має бути значно менше, ніж користувачів. В результаті число адміністрованих зв'язків стає пропорційним сумі кількості користувачів і

об'єктів, що по порядку величини зменшити вже неможливо. Рольове управління доступом оперує такими основними поняттями:

- користувач (людина, інтелектуальний автономний агент тощо.);
- сеанс роботи користувача;
- роль (зазвичай визначається згідно з організаційною структурою);
- об'єкт (сутність, доступ до якої розмежовується);
- операція (залежить від об'єкта; для таблиць системи управління базами даних - вставка, видалення);
- право доступу (роздільна здатність виконувати певні операції над певними об’єктами) [18].

Ролям приписуються користувачі та права доступу, можна вважати, що вони (ролі) іменують відносини "багато до багатьох" між користувачами і правами. Ролі можуть бути приписані багатьом користувачам; один користувач може бути приписаний декільком ролям. Під час сеансу користувача активізується підмножина ролей, яким він приписаний, в результаті чого він стає володарем об'єднання прав, приписаних активним ролям. Одночасно користувач може відкрити кілька сеансів.

1.3. Постановка задачі на розробку підсистеми розмежування доступу до баз даних в інформаційно-комунікаційній системі

В даній кваліфікаційній роботі потрібно розробити підсистему розмежування доступу до баз даних. Тобто, потрібно створити декілька груп користувачів із різними правами доступу до цієї бази. Це права на адміністрування, право на управління ресурсами, право на доступ. Деякі користувачі можуть взагалі не мати будь-яких прав, пов'язаних з конкретною базою даних. В розробленій підсистемі кожний користувач повинен мати свій логін і пароль. Кожному користувачеві бази даних присвоюється логін – коротке ім'я, що однозначно визначає користувача для системи управління базами даних. Ці логіни є основою системи безпеки. Від логіна користувача

залежить, чи буде дозволений або заборонений доступ до баз даних. Пароль служить для підтвердження того, що користувач дійсно має право працювати під введеним логіном. Логіни і паролі застосовуються в більшості базах даних. Безпека баз даних є головною проблемою динамічних додатків. Збереження і конфіденційність даних дуже важливі для кожної компанії. Для атаки зловмисник повинен володіти деякою інформацією. Ніколи заздалегідь не відомо, якими шляхами піде дана інформація. Якщо це все-таки станеться, база даних стає незахищеною.

Назва системи – «Розмежування підсистеми розмежування доступу до баз даних».

Призначення системи – підсистема розмежування доступу до баз даних призначена для розмежування доступу різних користувачів до бази даних, створення захисту баз даних від несанкціонованого доступу. Об’єктами бази даних є група користувачів, їх паролі і логіни.

Мета. Розробка підсистеми розмежування доступу до бази даних, аутентифікація та ідентифікація користувачів.

Вхідні дані. База даних, користувачі..

Технічні вимоги: апаратне та системне забезпечення: операційна система Windows-98, 2000, NT, XP, Vista, Microsoft Access, Visual Basic.

Функціональні характеристики системи:

- Надати повний доступ до бази даних.
- Надати доступ до бази даних кожному із користувачів.

Перелік операцій обробки даних:

- реєстрація користувачів;
- обробка логінів та паролів;
- введення та збереження даних;
- створення резервних копій даних;
- перевірка та налаштування прав доступу користувачів;
- здійснення операцій узагальнення даних.

Результат науково-дослідної та дослідно-конструкторської роботи – створення підсистеми розмежування прав доступу до бази даних користувачів системи.

Перелік науково-технічної документації, що надається після закінчення робіт:

- пояснювальна записка;
- опис автоматизованих функцій;
- опис організації інформаційної бази даних;
- перелік виконуваних операцій;
- опис алгоритмів;
- опис процесу оброблення даних.

Висновки до першого розділу

1. В першому розділі кваліфікаційної роботи були розглянуті теоретичні основи технології побудови систем розмежування доступу до баз даних в інформаційно-комунікаційних системах.

2. Проаналізовано моделі, методи та засоби побудови баз даних в інформаційно-комунікаційних системах. Аналіз інформаційних систем, що функціонують на підприємствах Житомирщини виявив, що останні використовують як сховище даних засіб Microsoft Office Access. Представлено механізм організації доступу до баз даних.

3. На основі можливостей апаратного забезпечення та фінансового забезпечення, прийнято рішення використовувати спосіб аутентифікації та ідентифікації за допомогою знання претендентом інформації, яку знає тільки легальний користувач – паролю та логіну.

4. Проведений аналіз та виявлені недоліки дозволили сформувати задачі на розробку підсистеми розмежування доступу до баз даних в інформаційно-комунікаційних системах.

РОЗДІЛ 2. МОДЕЛЮВАННЯ ТА РОЗРОБКА ПІДСИСТЕМИ РОЗМЕЖУВАННЯ ДОСТУПУ ДО БАЗ ДАНИХ

2.1. Опис функціональної структури інформаційної підсистеми

Для реалізації функцій інформаційної системи моніторингу земельних ділянок проведемо аналіз представленої до захисту інформаційної системи. Аналіз показав, що база даних інформаційної системи, що потребує захисту складається з наступних таблиць [22]:

- Dani_dilyanka_vod;
- Dilyanka;
- Spoluch;
- Spoluchennya;
- Sub_type_roztash;
- type_energ;
- Type_roztash;
- Type_spoluchennya;
- type_vodov;

Таблиця «Dani_dilyanka_vod», структура якої представлено у таблиці 2.1 призначена для збереження довідникових даних щодо можливих варіантів .водовідведення та їх впливу на вартість земельної ділянки.

Таблиця "Dilyanka" (таблиця 2.2) містить основні характеристики окремих ділянок, що підлягають обліку в земельному кадастрі. Дана таблиця зберігає площу, адресу ділянки, чи є на ділянці будівлі, газифіковано її чи ні та інші характеристики. Таблиця є центральною в базі даних та взаємодіє з іншими таблицями.

Таблиця 2.1. Структура таблиці "type_vodov"

Назва поля	Тип даних	Первинний ключ	Зовнішній ключ	Вміст
id_type	AutoNumber	Так	Ні	Містить тип видовідведення
koefvodov	Single	Ні	Ні	Містить поправковий коефіцієнт типу водовідведення
nazva	Text(20)	Ні	Ні	Містить назву типу водовідведення

Таблиця 2.2. Структура таблиці "Dilyanka"

Назва поля	Тип даних	Первинний ключ	Зовнішній ключ	Вміст
iddil	AutoNumber	Так	Ні	Містить код ділянки
plosha	Text(18)	Ні	Ні	Містить площу ділянки
addr	Text(60)	Ні	Ні	Містить адресу ділянки
prim	Text(18)	Ні	Ні	Містить опис ділянки
gaz	Так/Ні	Ні	Ні	ділянка чи ні
idsubroz	Long Integer	Ні	Так	Містить тип району ділянки
idenerg	Long Integer	Ні	Так	Містить тип енергопостачання ділянки
Дата внесення	Date/Time	Ні	Ні	Містить дату внесення ділянки до реєстру

Таблиці "spoluch" та "Spoluchennya" призначені для збереження довідникових даних щодо можливих варіантів транспортного сполучення для земельної ділянки та їх впливу на вартість земельної ділянки. Структуру даної таблиці представлено у таблиці 2.3

Таблиця 2.3. Структура таблиці "Spoluchennya"

Назва поля	**Тип даних**	**Первинний ключ**	**Зовнішній ключ**	**Вміст**
idsubspol	AutoNumber	Так	Ні	Містить код виду сполучення
koef_sp	Single	Ні	Ні	Містить поправковий коефіцієнт виду сполучення
idspol	Long Integer	Ні	Так	Містить код типу сполучення
nazva	Text(20)	Ні	Ні	Містить назву виду сполучення

Таблиця Type_spoluchennya призначена для збереження типів транспортного сполучення, що визначені у методиці оцінки земельних ділянок несільськогосподарського призначення. Введення такої таблиці дозволяє забезпечити уніфікований облік транспортних засобів, доступних для конкретних земельних об‘єктів.

Таблиця 2.4. Структура таблиці "Type_spoluchennya"

Назва поля	**Тип даних**	**Первинний ключ**	**Зовнішній ключ**	**Вміст**
idspol	AutoNumber	Так	Ні	Містить код виду сполучення
nazva	Text(20)	Ні	Ні	Містиь назву виду сполучення
prim	Text(60)	Ні	Ні	Містить пояснення

				щодо виду сполучення

Таблиці "Sub_type_roztash" та "Type_roztash" призначені для збереження довідкових даних щодо можливих варіантів транспортного сполучення для земельної ділянки та їх впливу на вартість земельної ділянки. Структура даних таблиць представлена у таблицях 2.5-2.6

Таблиця 2.5. Структура таблиці "Sub_type_roztash"

Назва поля	**Тип даних**	**Первинний ключ**	**Зовнішній ключ**	**Вміст**
idsubroz	AutoNumber	Так	Ні	Містить код типу району розташування
koef_roztash	Single	Ні	Ні	Містить поправковий коефіцієнт типу району розташування
nazvasub	Text(20)	Ні	Ні	Містить назву типу району розташування
idroz	Long Integer	Ні	Так	Містить код місцевості

Таблиця 2.6. Структура таблиці "Type_roztash"

Назва поля	**Тип даних**	**Первинний ключ**	**Зовнішній ключ**	**Вміст**

idroz	AutoNumber	Так	Ні	Містить код місцевості
nazva	Text(20)	Ні	Ні	Містить назву місцевості
prim	Text(60)	Ні	Ні	Примітка
vart_metr	Long Integer	Ні	Ні	Містить вартість квадратного метру площі у місцевості

Таблиця "type_energ" призначена для збереження довідникових даних щодо можливих варіантів енергопостачання для земельної ділянки та їх впливу на вартість земельної ділянки. Структура даної таблиці представлена у таблиці 2.7.

Таблиця 2.7. Структура таблиці "type_energ"

Назва поля	**Тип даних**	**Первинний ключ**	**Зовнішній ключ**	**Вміст**
idenerg	AutoNumber	Так	Ні	Містить код типу енергопостачання
nazva	Text(20)	Ні	Ні	Містить назву типу енергопостачання
koefenerg	Single	Ні	Ні	Примітка

Таблиця "spoluch" містить можливі види транспортного сполучення окремих ділянок, що підлягають обліку в земельному кадастрі. Дана таблиця зберігає відповідні коди ділянки та виду сполучення. До видів транспортного сполучення належать всі види міського транспорту – трамвай, тролейбус,

маршрутне таксі та міжміського транспорту – автобусне сполучення, залізничне та інші види.

Таблиця 2.8. Структура таблиці "spoluch"

Назва поля	**Тип даних**	**Первинний ключ**	**Зовнішній ключ**	**Вміст**
idsubspol	Long Integer	Так	Так	Містить код виду транспортного сполучення
iddil	Long Integer	Так	Так	Містить код ділянки
prim	Text(50)	Ні	Ні	Примітка

Таблиця "type_vodov" містить можливі види транспортного сполучення окремих ділянок, що підлягають обліку в земельному кадастрі. Дана таблиця зберігає відповідні коди ділянки та виду сполучення.

Таблиця 2.9. Структура таблиці "dani_dilyanka_vod"

Назва поля	**Тип даних**	**Первинний ключ**	**Зовнішній ключ**	**Вміст домену**
id_type	Long Integer	Так	Так	Містить тип водовідведення
iddil	Long Integer	Так	Так	Містить номер ділянки
prim	Text(60)	Ні	Ні	Примітка

Загальний вигляд схеми бази даних наведено на рис. за допомогою ER діаграми рис. 2.4.

Проаналізована в даному пункті структура бази даних відповідає всім вимогам до оптимальних структур збереження даних, зокрема всі сутності існують у третій нормальній формі. З іншого боку визначені інформаційні

поля забезпечують збереження всієї необхідної інформації про земельну ділянку, що дозволяє реалізувати процедури розрахунку її вартості.

Для забезпечення роботи підсистеми захисту необхідним є розробка таблиці користувачів та таблиці доступу. Розбиваємо користувачів на дві групи: адміністратор та користувачі.

Доступ групі адміністратору надається в повному обсязі, не накладається жодних обмежень щодо використання, налаштування, перегляду інформації, додавання користувачів та інші функції, що передбачені інформаційною системою та БД.

Група користувачів має певні обмеження у роботі. Так, для переважної більшості користувачів системи встановлено перегляд інформації і тільки деякі коригування, які вони можуть проводити в БД.

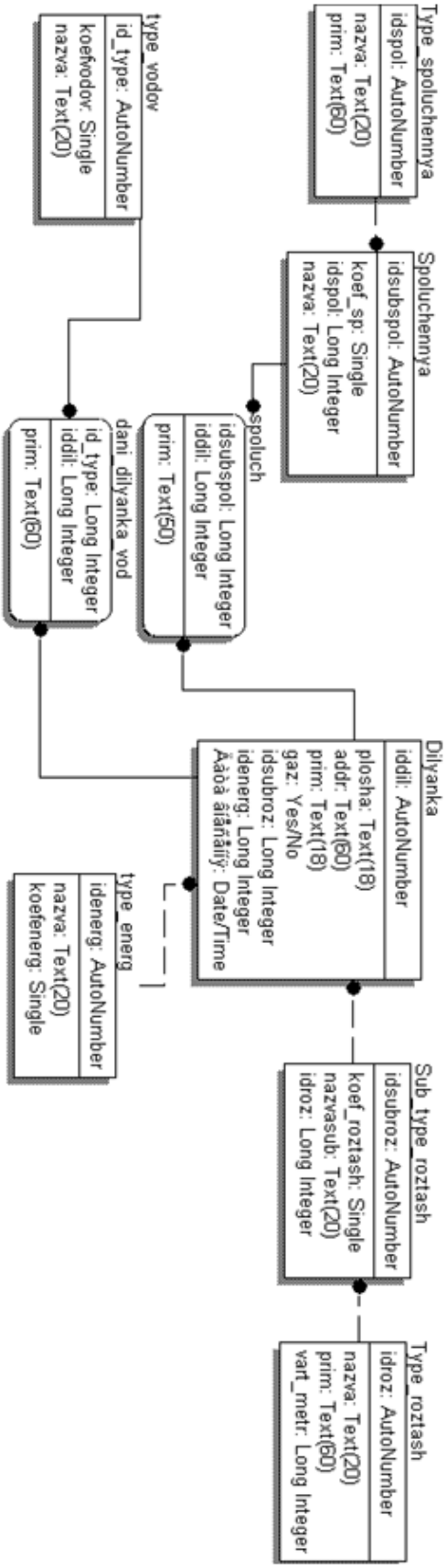

Рисунок 2.1 - Діаграма «сутність-зв‘язок» бази даних

Проведений аналіз інформаційних потреб предметної області дозволив сформувати структурну схему функціональних модулів інформаційної підсистеми розмежування доступу до баз даних, яка представлена на рисунку 2.2.

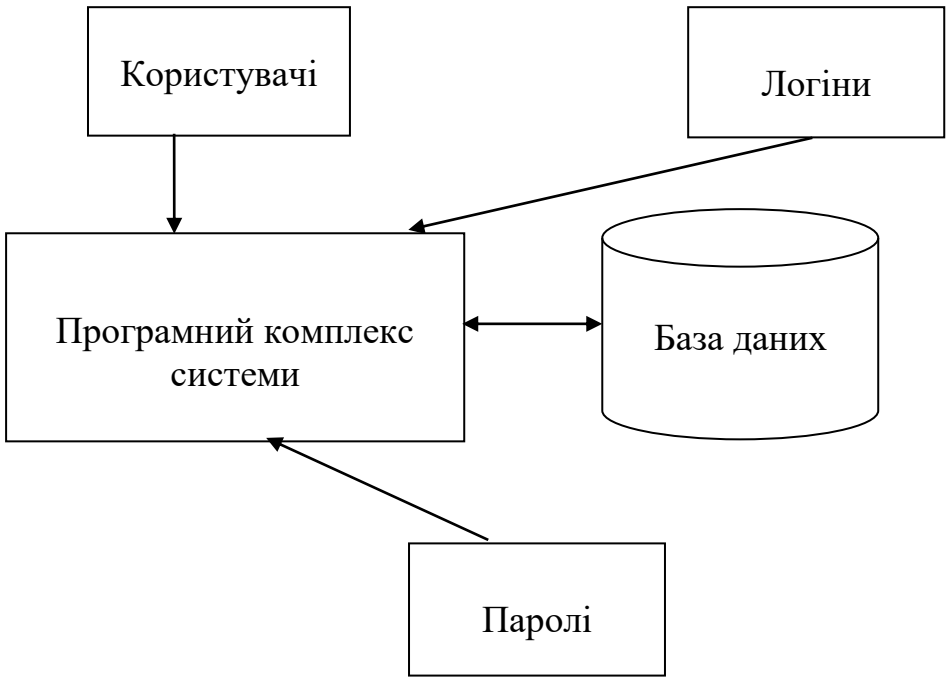

Рисунок 2.2. Структурна схема функціональних модулів інформаційної системи

Для виконання задачі – розмежування доступу до баз даних необхідно, щоб система могла:

- забезпечувати можливість вводу інформації ;
- забезпечувати перегляд інформації;
- контролювати правильність вводу інформації;
- забезпечувати можливість пошуку необхідної інформації;
- забезпечувати можливість сортування інформації.

Є група користувачів із певними правами. Адміністратор має право забезпечувати можливість вводу і перегляд інформації, контролювати правильність вводу інформації, забезпечувати можливість пошуку необхідної інформації, забезпечувати можливість сортування інформації. Оператор має право забезпечувати можливість вводу інформації, переглядати інформацію,

забезпечувати можливість пошуку необхідної інформації, може сортувати інформацію. Менеджер має право забезпечувати можливість вводу інформації, переглядати інформацію, забезпечувати можливість пошуку необхідної інформації. Інспектор з кадрів може переглядати інформацію і вводити необхідну інформацію. Інженер може переглядати необхідну йому інформацію.

Загальні відомості про користувачів можна описати за допомогою таблиці.

Таблиця 1.10. Відомості про користувачів

	Користувачі				
	Адміністратор	Оператор	Менеджер	Інструктор з кадрів	Інженер
логін	admin	operator	meneger	instruktor	ingener
код	1	2	3	4	5

Кожному із користувачів присвоюється логін, пароль і код. При вводі логіну і паролю, користувач буде мати доступ до тих елементів бази даних, до яких має право доступу.

Таблиця 1.11. Логіни

Користувач	Адміністратор	Оператор	Менеджер	Інспектор з кадрів	Інженер
Логін	admin	operator	meneger	instruktor	ingener

Логін – алфавітно-цифровий набір символів, що ідентифікує користувача бази даних і разом із паролем використовується операційною

системою для надання йому дозволу на доступ до бази даних та визначення його прав доступу до ресурсів бази даних. Логін має бути унікальним в межах даної системи.

Таблиця 1.12. Паролі

Користувач	Адміністратор	Оператор	Менеджер	Інспектор з кадрів	Інженер
Пароль	admin12	ghbdtnbr	441875	pfobnf1	user4

Пароль являє собою послідовність знаків, що дозволяє користувачам входити в базу даних, отримувати доступ до інформації в базі даних. Паролі дозволяють бути впевненим у тому, що ніхто не матиме доступу до комп'ютера до тих пір, поки не отримає відповідного дозволу.

Таким чином для виконання завдання дипломного проекту створюємо таблицю tbUsers, структура якої представлена у таблиці 2.13.

Таблиця 2.13. Структура таблиці tbUsers

Назва поля	Тип даних	Первинний ключ	Зовнішній ключ	Вміст домену
sName	Text(50)	Так	Так	Містить логіни користувачів
sPWD	Text(10)	Ні	Ні	Містить пароль користувача

Таким чином визначено структуру бази даних та компоненту захисту. Проведемо моделювання системи розмежування доступу БД

2.2. Моделювання та проектування інформаційної підсистеми

Сучасні технології моделювання не тільки полегшили і прискорили процес побудови та дослідження моделі, але й значно наблизили сприйняття інформації спеціаліста з моделювання систем і спеціаліста, що працює у галузі, яка моделюється. Серед великої кількості методів моделювання, що існують, виділимо такі методи: аналітичне моделювання, математичне моделювання, імітаційне моделювання. Аналіз результатів моделювання складається з оцінки точності результатів моделювання, оцінки стійкості результатів моделювання та оцінки чутливості результатів моделювання. Формування висновків та пропозицій є завершальний етап моделювання, на якому підводяться підсумки та висловлюються думки щодо напрямків подальшого дослідження об’єкта моделювання. Мова GPSS являється мовою загального призначення для моделювання мереж масового обслуговування, що набула широкого розповсюдження завдяки книзі (Шрайбер), в якій розглядається велика кількість прикладів моделювання систем різного призначення мовою GPSS.

Процес моделювання може бути наведено у вигляді UML-діаграм, що розкривають сутність та склад системи [23].

Побудуємо діаграму використання системи, рисунок 2.3.

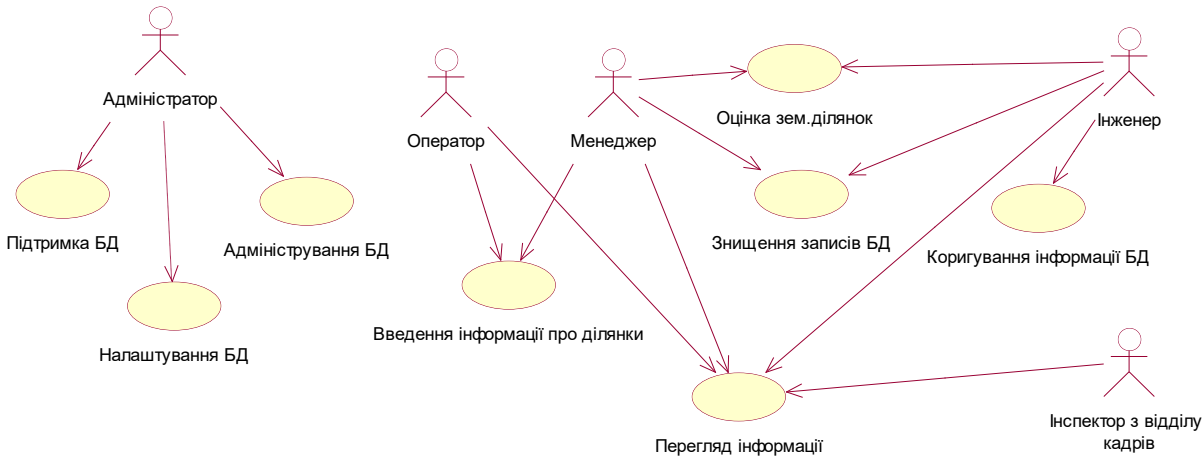

Рисунок 2.3. Діаграма використання інформаційної підсистеми розмежування доступу

Таким чином, як видно з рисунку, адміністратор в системі проводить підтримку БД, її адміністрування та налаштування. Оператор може проводити введення інформації про ділянки та переглядати інформацію, що містить в БД. Менеджер має права на перегляд інформації, введення інформації про ділянки, знищувати записи БД, проводити оцінку земельних ділянок. Інженер може коригувати інформацію в БД, проводити оцінку земельних ділянок, знищувати записи БД, а інспектор з відділу кадрів може лише переглянути інформацію про наявні земельні ділянки в БД та іншу інформацію, що там міститься.

Діаграма активності системи представлена на рисунку 2.4. Як видно з рисунку проводиться запуск БД. Далі проводить процедура аутентифікації шляхом введення логіну користувача та паролю. При неправильному введенні знову повторюєте введення логіну та паролю. При правильному введенні завантажується головне вікно програми. Далі відбувається перегляд даних користувачів та визначається чи дані доступні (БД підключена коректно).

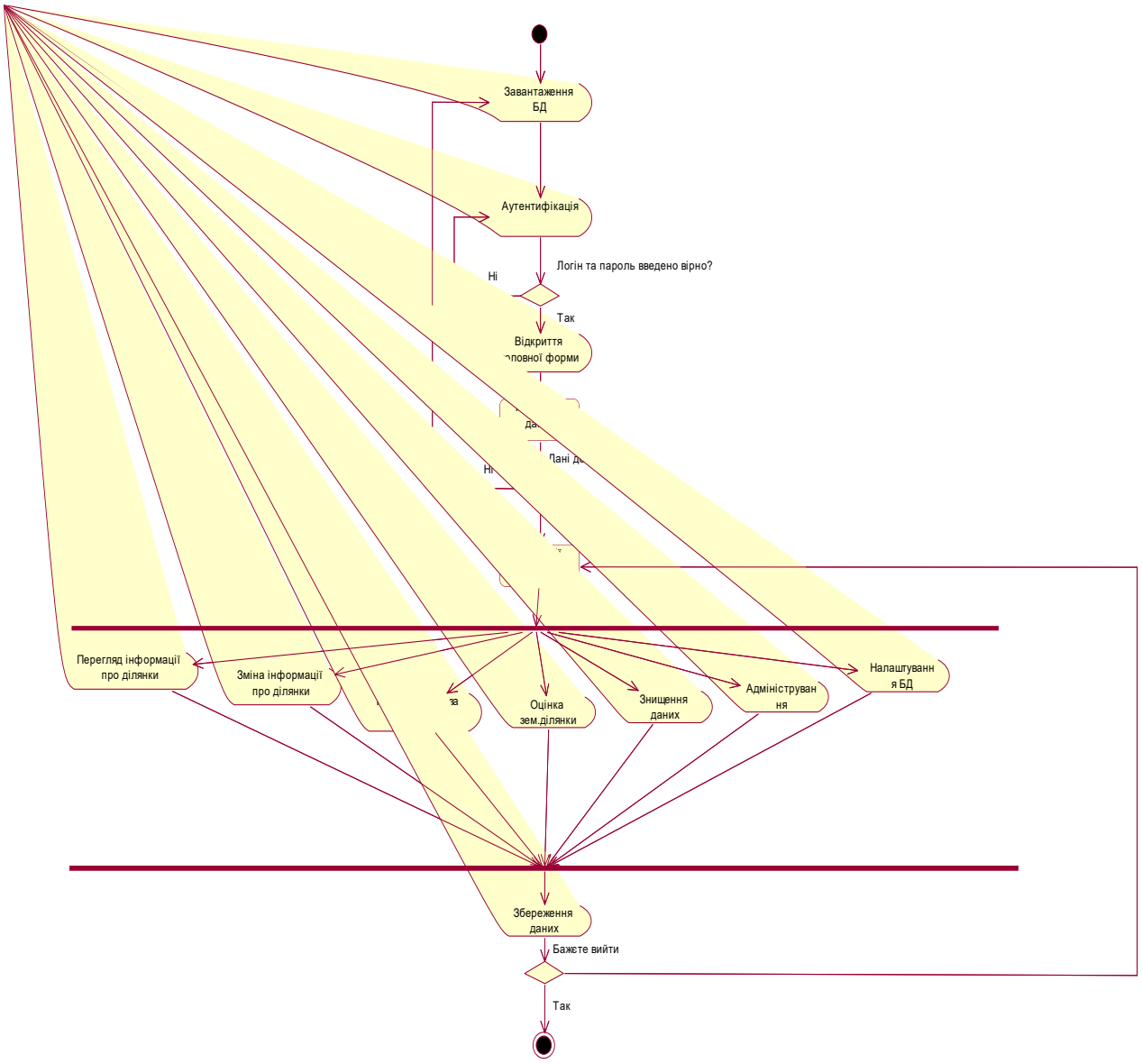

Рисунок 2.3. Діаграма активності системи

Якщо БД не підключилась, то необхідно повторити підключення БД та пройти процедуру аутентифікації ще раз. При успішному підключені БД наступним кроком є вибір дії на головній формі системи:

– перегляд інформації про ділянки;
– зміна інформації про ділянки;
– пошук ділянки за критеріями;
– оцінка земельної ділянки;
– знищення даних;

– адміністрування;

– налаштування БД

В залежності від типу користувача та проведених налаштувань змінюється функціональність запропонованої системи. Після проведених дій з системою відбувається збереження даних. Далі користувач може вийти з системи або продовжити роботу шляхом вибору наступної дії.

Розроблені алгоритми є частиною процесу моделювання інформаційної системи за допомогою мови моделювання UML. Слід зазначити, що система, що змодельована, не орієнтована на використання її декількома користувача одночасно.

2.3. Обґрунтування вибору технології побудови підсистеми розмежування доступу до баз даних

При впровадженні та реалізації інформаційної системи розмежування доступу до баз даних необхідно провести обґрунтований вибір програмних засобів для реалізації логічної та фізичної архітектури. До програмного забезпечення, що дозволяє реалізувати більшість функцій підсистеми розмежування доступу до баз даних, були поставлені такі вимоги [24]:

- ефективність та простота користування;
- сумісність з іншим програмним забезпеченням що використовується для схожих задач.

Розглянемо можливості Access і типи задач які можна вирішувати за допомогою цієї системи управління базами даних. Використовується Access для зберігання і пошуку даних представлення інформації в зручному вигляді і автоматизації виконання задач, що повторюються, розміщення форм у вигляді документів HTML на Web-сторінках обміну даними з вузлами Internet/Intranet. Розробка за допомогою Access простих і зручних форм вводу даних, обробка даних і генерація складних звітів. Підтримка в Access механізму запиту на вибірку, сортування та пошук даних. Перед тим як

почати роботу з будь-яким програмним продуктом важливо зрозуміти його можливості і типи задач для рішення яких він призначений. Microsoft Access об'єднує відомості з різних джерел в одну реляційну базу даних. Створювані форми, запити і звіти дозволяють швидко і ефективно оновлювати дані отримувати відповіді на поставлені питання аналізувати дані друкувати звіти [25].

В Access в повній мірі реалізовано управління реляційними базами даних. Система підтримує первинні та зовнішні ключі і забезпечує цілісність даних на рівні ядра (що попереджує несумісні операції оновлення і видалення даних). Access підтримує всі необхідні типи полів, в тому числі текстовий числовий лічильник грошовий, дата, час логічні. Якщо в процесі спеціальної обробки в полі не має ніяких значень система забезпечує повну підтримку пустих значень. Реляційна обробка даних в Access за рахунок гнучкої архітектури системи здатна задовольнити будь-які потреби. При цьому Access може використовуватися як автономна система управління базами даних в режимі файл-сервера чи клієнтського компонента таких продуктів як SQL Server. Система Access підтримує обробку транзакцій з гарантією їх цілісності. Крім того, передбачений захист на рівні користувача що дозволяє контролювати доступ до даних окремих користувачів і цілих груп.

В базі даних відомості з кожного джерела зберігаються в окремій таблиці. При роботі з даними із декількох таблиць встановлюються зв'язки між таблицями. Для пошуку і відбору даних що задовольняють умови, створюється запит. Запит дозволяє також видалити або обновлювати одночасно декілька записів. Для перегляду, вводу та зміни даних в таблиці використовують форми. Форма дозволяє вибрати дані із однієї чи декількох таблиць і вивести їх на екран використовуючи стандартний або створений користувачем макет. Для аналізу даних або для виведення їх на друк використовується звіт наприклад можна створити і надрукувати звіт що групує дані і вираховує результат. Для автоматичного здійснення деяких

операцій використовують макроси що містять набір із однієї чи більше макрокоманд таких як відкриття форм і друк звітів. Макроси можуть бути корисні для автоматизації часто виконуваних задач. Наприклад при натисканні користувачем кнопки можливо запустити макрос, що роздруковує звіт.

Висновки до другого розділу

У другому розділі розроблено підсистему розмежування доступу до баз даних. Побудовано та описано функціональну структуру інформаційної підсистеми. Визначено, що для розмежування доступу до баз даних необхідно, щоб система могла:

- забезпечувати можливість вводу інформації ;
- забезпечувати перегляд інформації;
- контролювати правильність вводу інформації;
- забезпечувати можливість пошуку необхідної інформації;
- забезпечувати можливість сортування інформації.

Також визначено, що для розмежування доступу до баз даних необхідно створити групу користувачів із логінами, паролями і своїми правами доступу до бази даних. Побудували діаграму активності і використання системи. Обґрунтували вибір технології побудови підсистеми розмежування доступу до баз даних. Підсистему розмежування доступу до баз даних будували за допомогою пакету Microsoft Access і Visual Basic.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЕКТУ РОЗМЕЖУВАННЯ ДОСТУПУ ДО БАЗ ДАНИХ

3.1 Інтерфейс користувача

В даній кваліфікаційній роботі було спроектовано інтерфейс автоматизованої системи розмежування доступу до баз даних, структурна схема якої представлена на рисунку 3.1

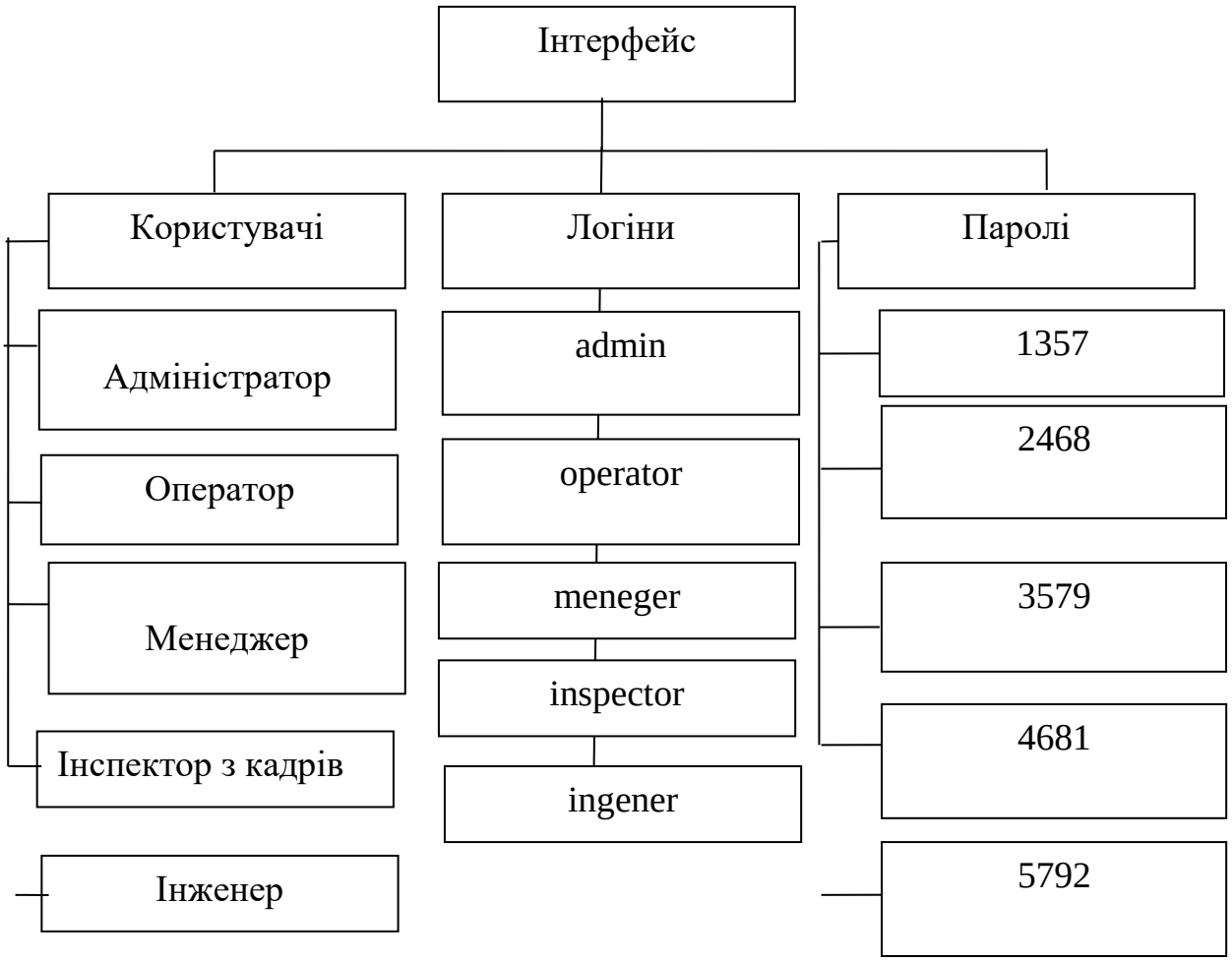

Рисунок 3.1. Структура інтерфейсу підсистеми

До складу розробленої бази даних входять такі елементи інтерфейсу:

1) Форми – дають змогу користувачу переглядати дані кожної таблиці БД окремо, оновлювати дані в таблиці, видаляти або додавати дані у таблиці. Розроблені форми представлені на рисунку 3.2-3.7.

2) Звіти – перегляд інформації що зберігається в БД за потребами користувача в зручному вигляді.

3) Запити – дозволяють користувачеві виводити дані за заданим параметром чи критерієм, виконувати розрахунок, оновлювати дані, здійснювати вставку, видалення даних.

Головна кнопкова форма автоматизованої системи моніторингу земельних ділянок представлена на рисунку 3.2.

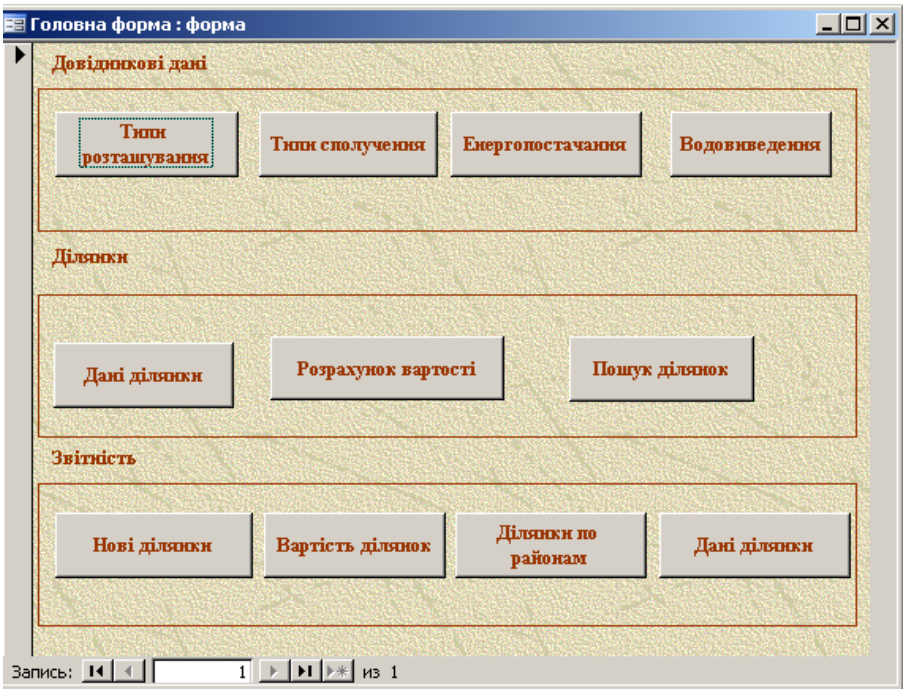

Рисунок 3.2. Вигляд кнопкової форми

Кнопка "Типи сполучення" відкриває форму на рисунку 3.3.

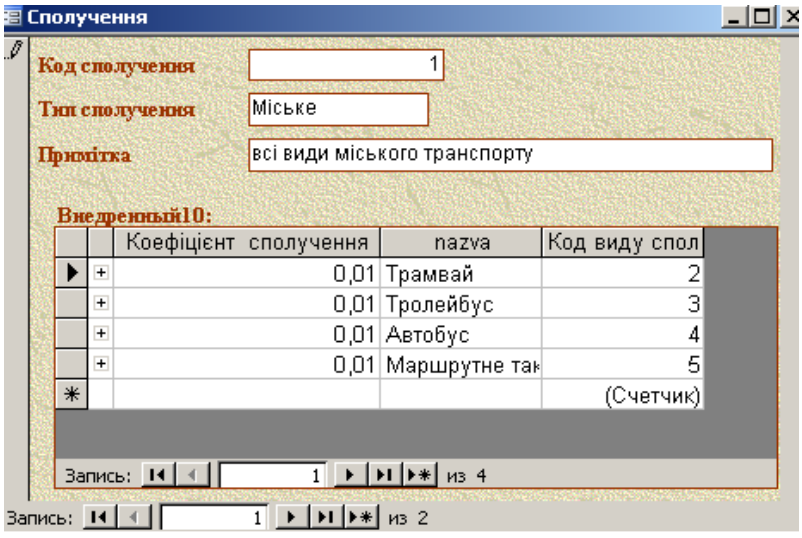

Рисунок 3.3. Вигляд форми «Сполучення»

Форма, що дозволяє додавати, переглядати територіальні райони на рисунку 3.4.

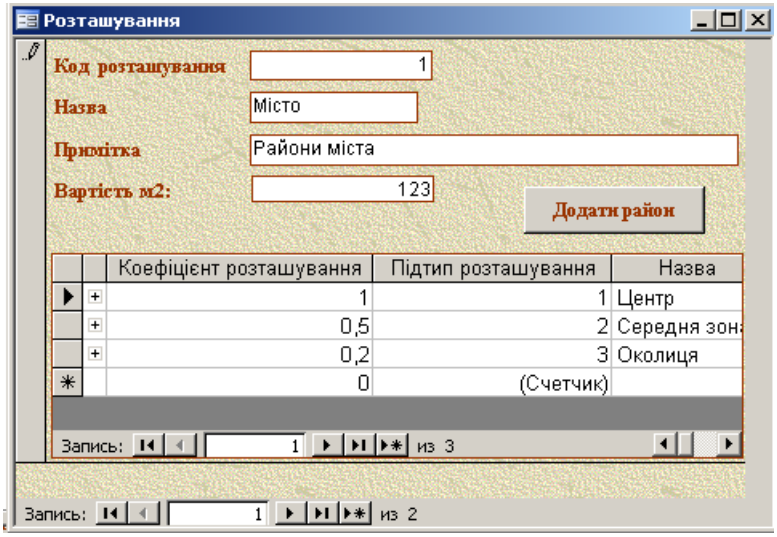

Рисунок 3.4. Вигляд форми

Кнопка "Енергопостачання" відкриває форму на рисунку 3.5.

Рисунок 3.5. Вигляд форми

Ознайомитися з видами водовідведення можна використовуючи форму "Водопостачання" (рисунку 3.6)

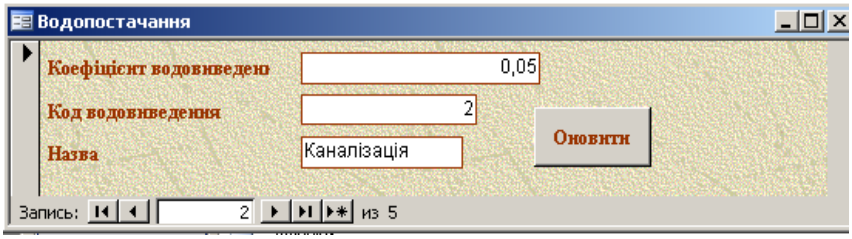

Рисунок 3.6. Вигляд форми

Для реєстрації земельних ділянок та їх характеристик спроектовано складу схему рисунку 3.7

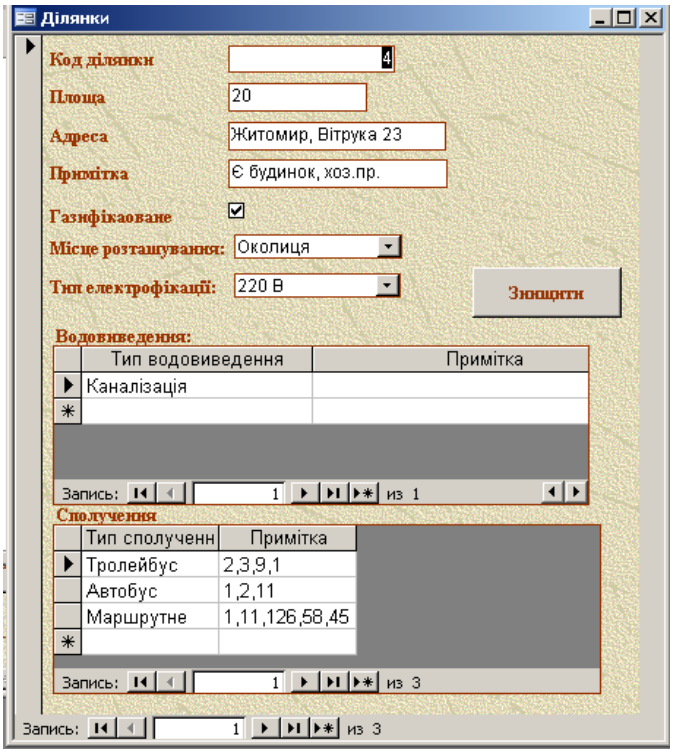

Рисунок 3.7. Вигляд форми

Таким чином, представлено інтерфейс програмного продукту, що дозволяє мати зручний доступ до інформації що зберігається.

Проведемо опис інтерфейсної частини запропонованої підсистеми розмежування доступу до баз даних

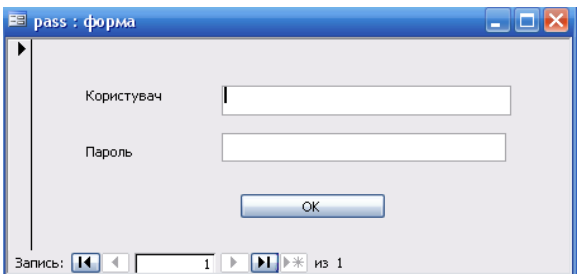

Рисунок 3.8. Форма для проведення процедури аутентифікації

3.2. Адміністрування доступу до баз даних

Адміністрування баз даних це функція управління та підтримки програмного забезпечення систем управління базами даних (СУБД). Основні програмні засоби СУБД, такі як Oracle, IBM DB2 і Microsoft SQL Server,

потребують постійного управління. Таким чином, корпорації, які використовують програмне забезпечення СУБД, часто наймають спеціалістів з інформаційних технологій, які називаються адміністраторами баз даних [26].

Проведемо адміністрування запропонованої підсистеми доступу до бази даних. Для цього заповнимо таблицю tbUsers. В заповненому вигляді представимо на рисунку 3.9.

Наступним етапом є проведення процедури налаштувань прав доступу до бази даних, що представлено на рисунках 3.10-3.20.

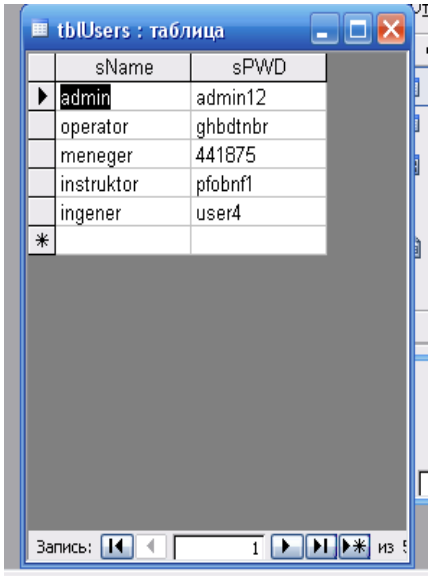

Рисунок 3.9. Вигляд заповненої таблиці

Проводимо спочатку налаштування прав доступу адміністратору інформаційної підсистеми. На рисунку 3.10 представлено перелік дій, які дозволено адміністратору відносно поточної бази.

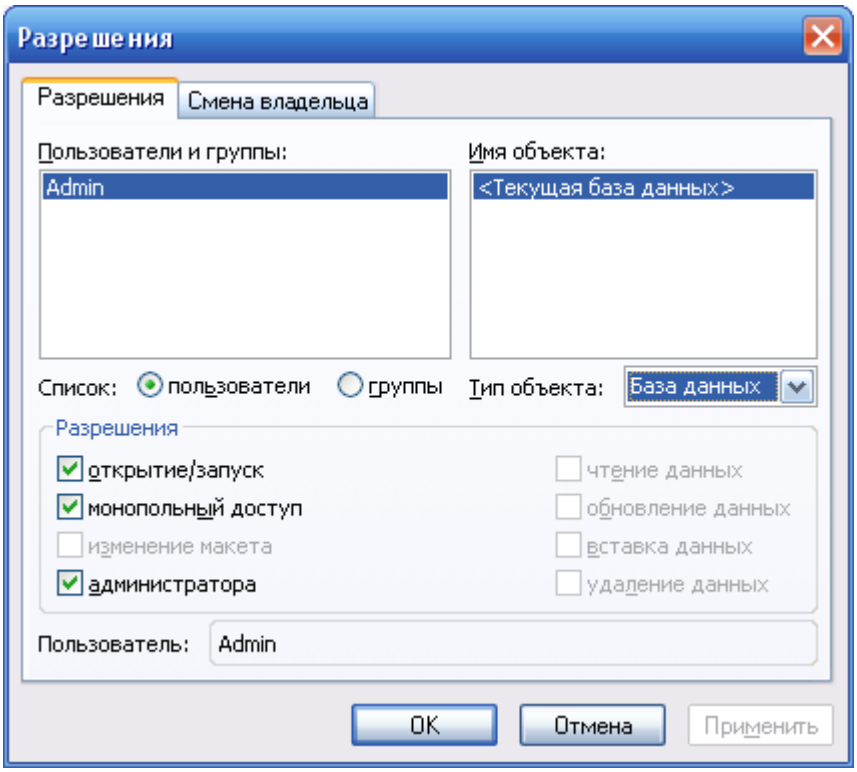

Рисунок 3.10. Налаштування прав доступу та дії до бази даних в цілому

Проведемо наступні налаштування. Дії відносно запитів представлено на рисунку 3.11.

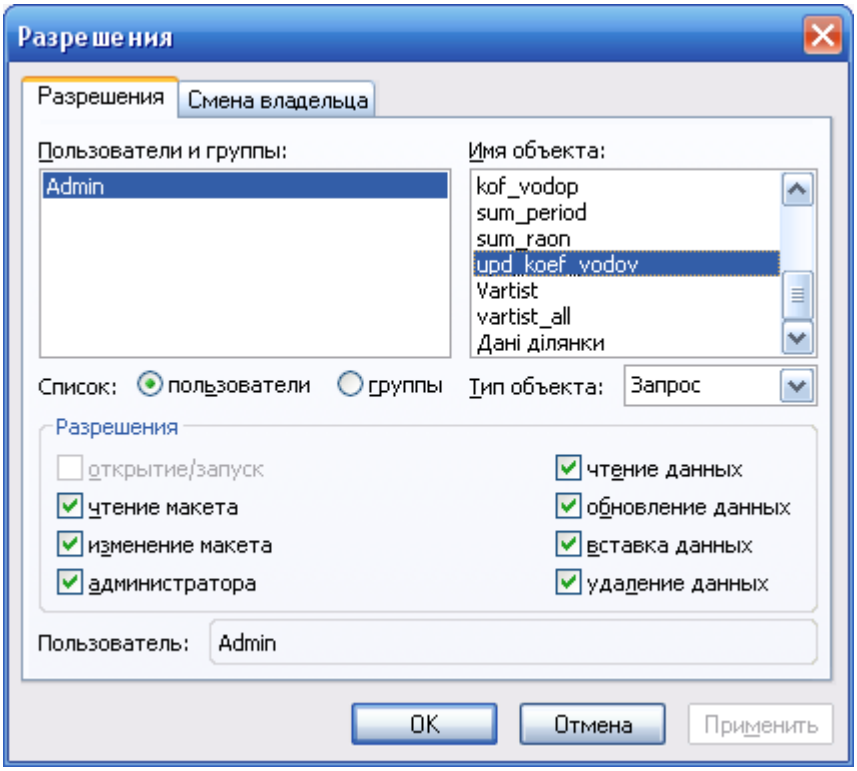

Рисунок 3.11. Налаштування прав доступу до запитів адміністратору

Проведемо налаштування прав доступу на дії, які може виконувати адміністратор відносно таблиць. Результати налаштувань представимо на рисунку 3.12.

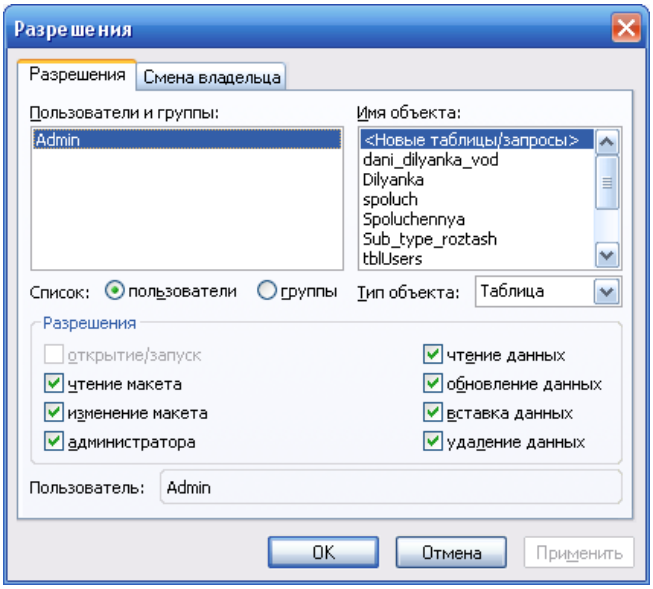

Рисунок 3.12. Налаштування дозволених дій адміністратору відносно таблиць

Проведемо налаштування дій адміністратора в системі відносно форм БД

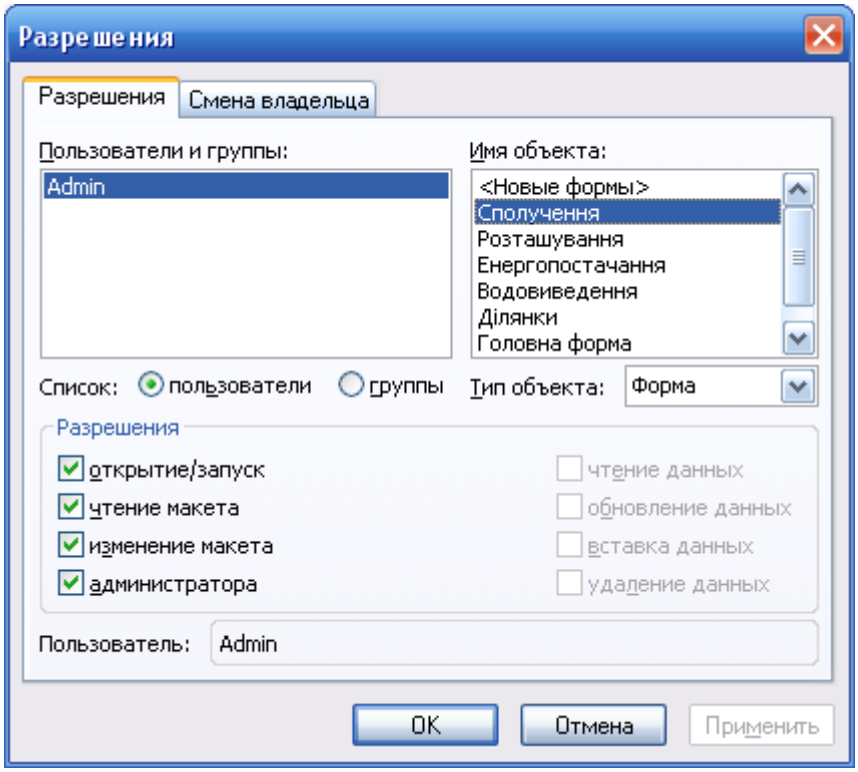

Рисунок 3.13 – Дії адміністратора по відношенню до форм БД

Можливості роботи зі звітами, що формують у БД представлено на рисунку 3.14.

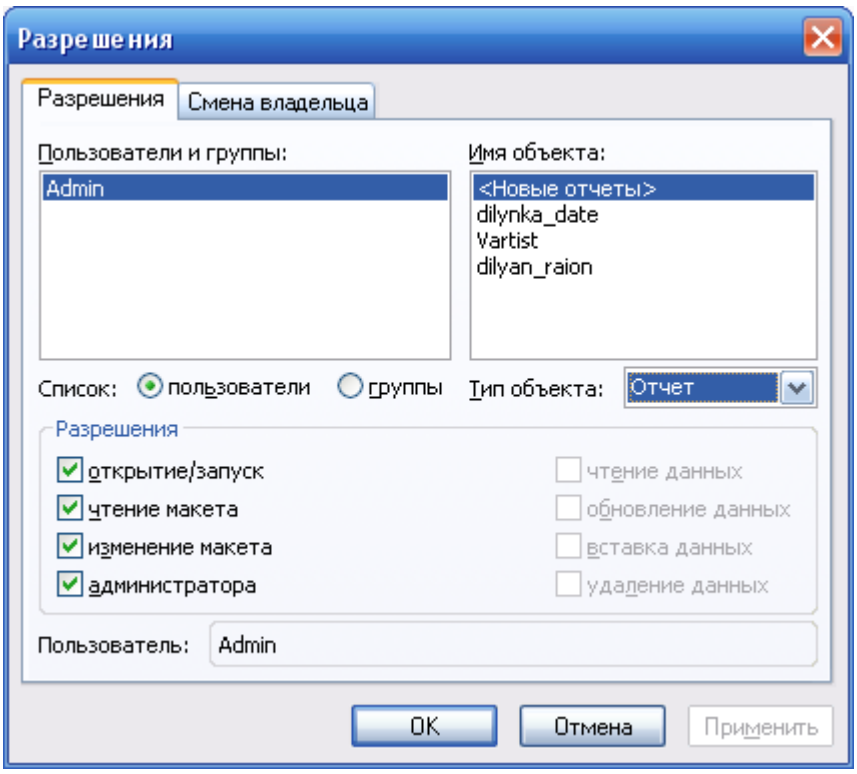

Рисунок 3.14. Дії адміністратора відносно звітів, що формуються БД

Встановлені дії адміністратора відносно макросів представлено на рисунку 3.15.

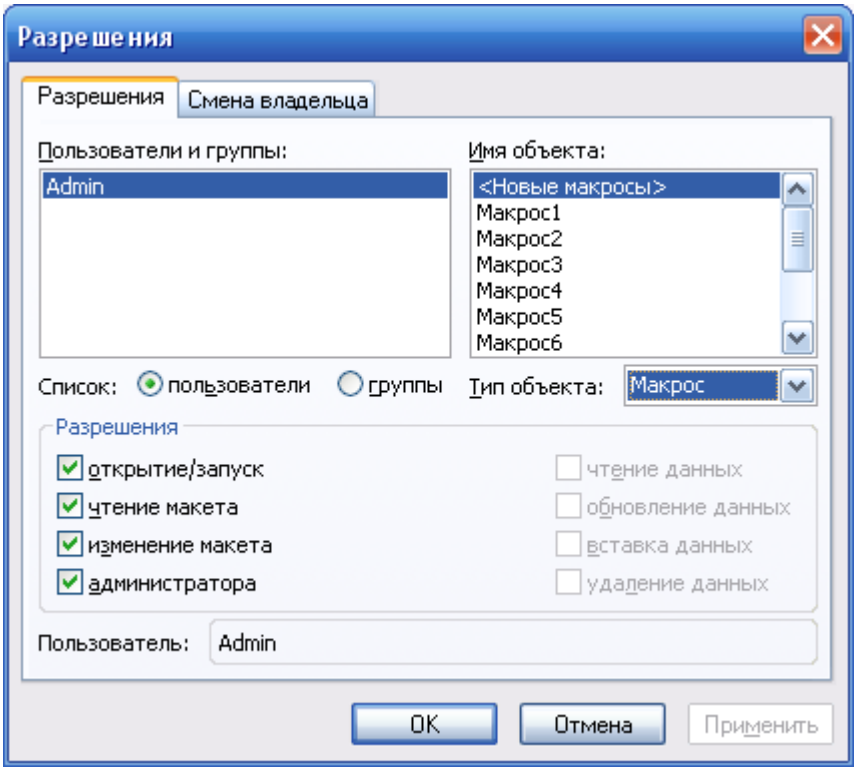

Рисунок 3.15. Встановлені дії адміністратора відносно макросів

Всі права власності на всі типи об’єктів, що знаходяться в БД представлено на рисунку 3.16.

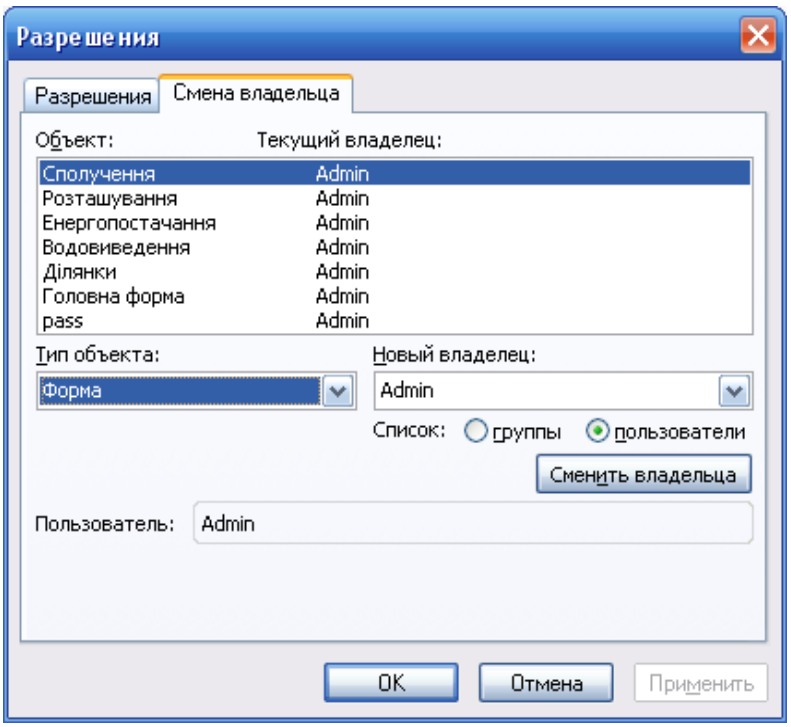

Рисунок 3.16. Вигляд налаштувань прав власності в БД

Проведемо адміністрування робочої групи «Users». Проведемо налаштування дій, які може виконувати дана група в БД. Так, дії користувачів щодо форм представлено на рисунку 3.17, відносно звітів на рисунку 3.18.

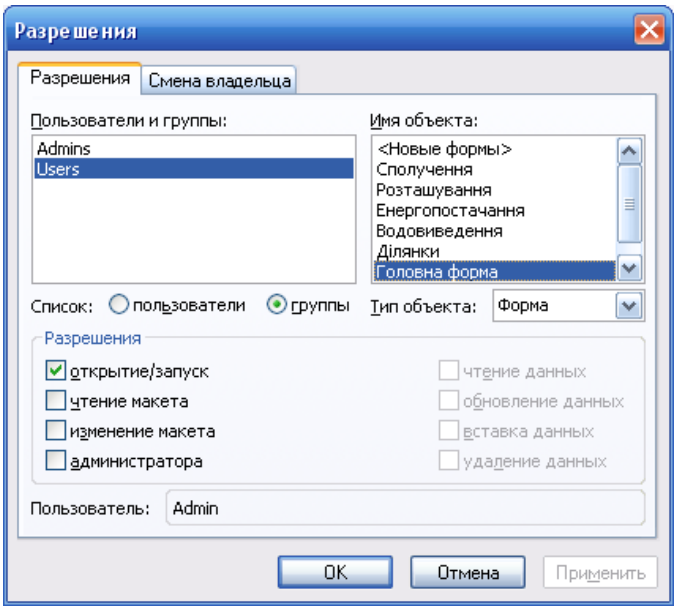

Рисунок 3.17. Дій користувачів щодо форм бази даних

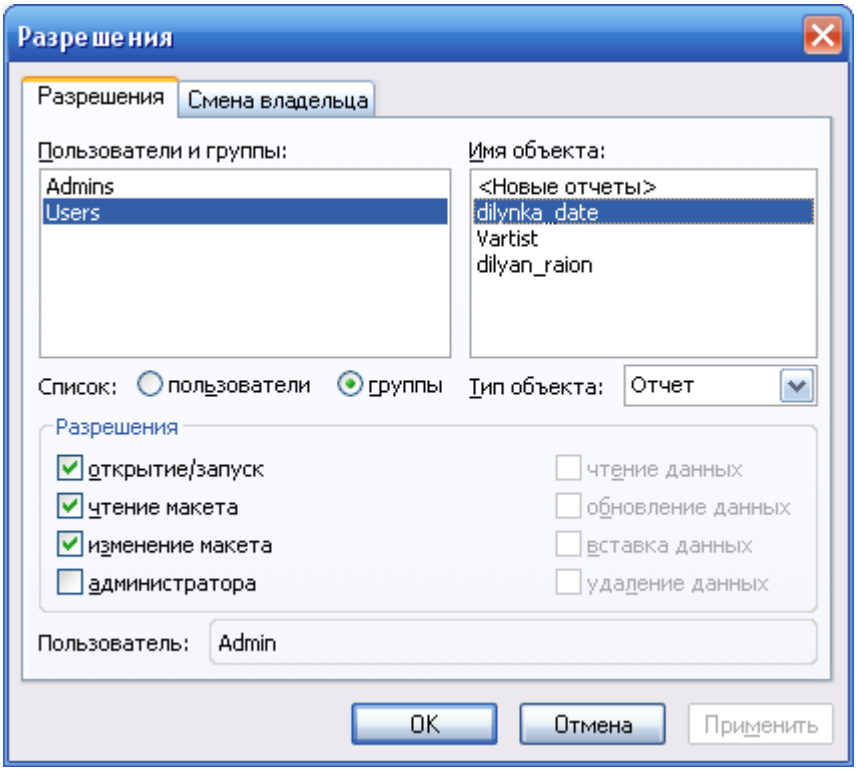

Рисунок 3.18. Налаштування дій користувачів відносно звітів БД

Вигляд форми налаштувань групи користувачів відносно використання та роботи з запитами представлено на рисунку 3.19. Вигляд налаштувань прав дій користувача відносно макросів – рисунок 3.20

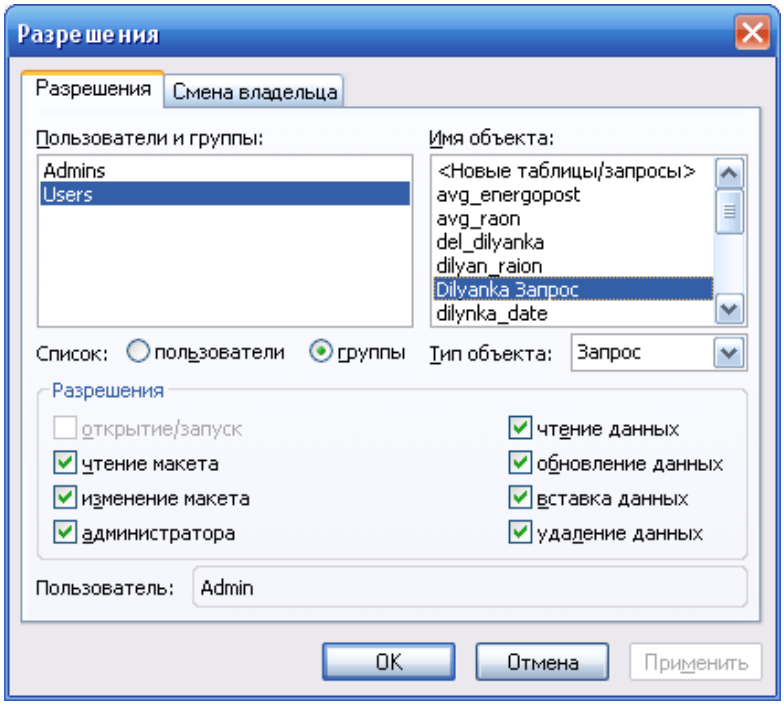

Рисунок 3.19.

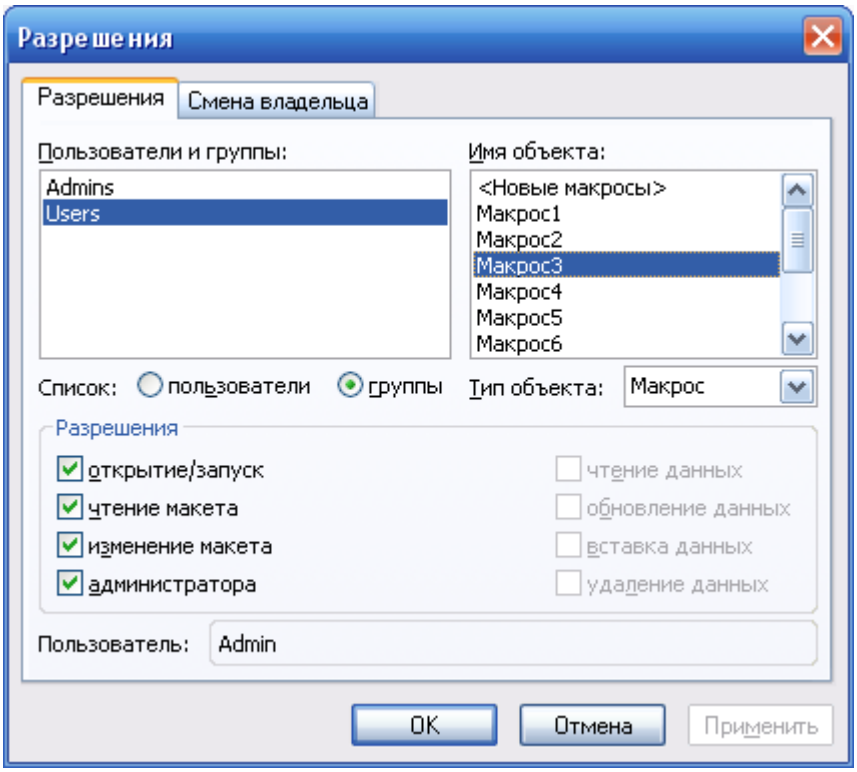

Рисунок 3.20

3.3. Формування запитів та повірка їх працездатності

Форми запитів підсистеми розмежування доступу до баз даних представлено на рисунка 3.21-3.25.

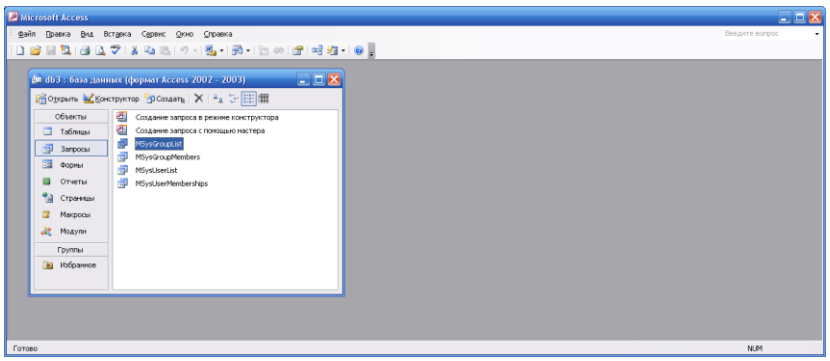

Рисунок 3.21. Перелік запитів

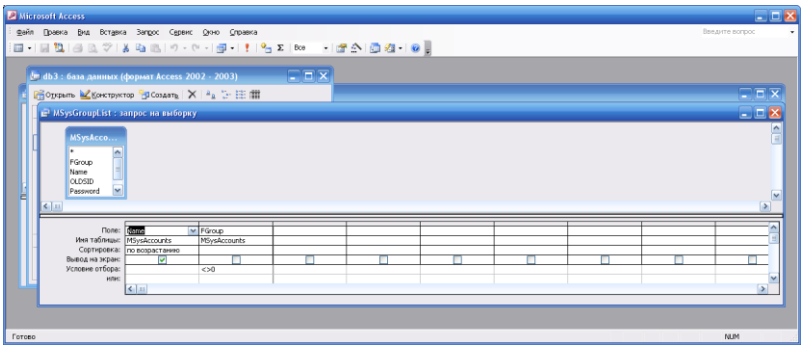

Рисунок 3.22. Вигляд запиту MSysGroupList

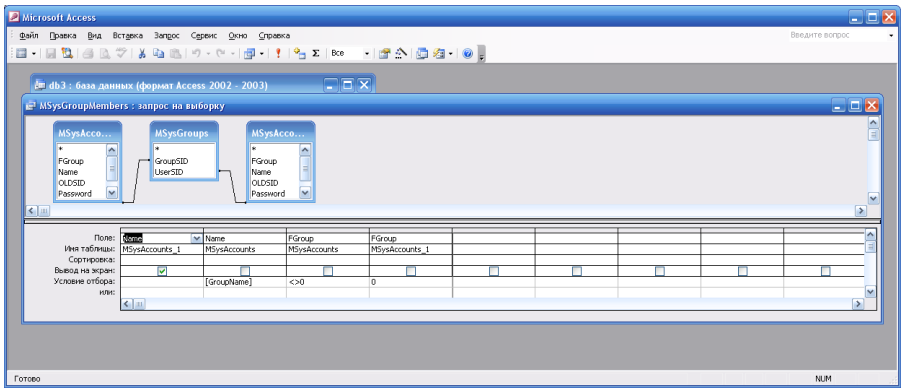

Рисунок 3.23. Вигляд запиту MSysGroupMembers

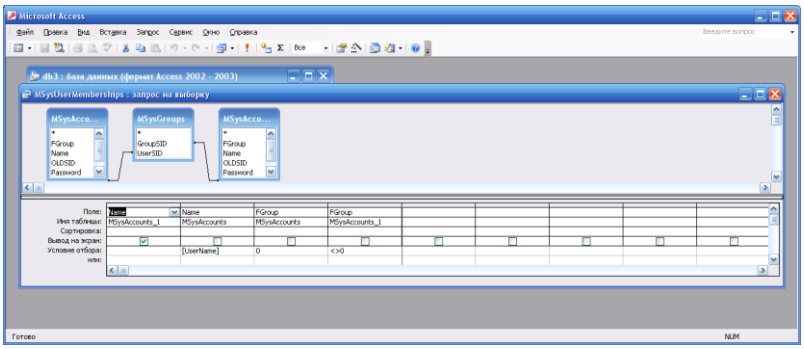

Рисунок 3.24. Вигляд запиту MSysUserMemberships

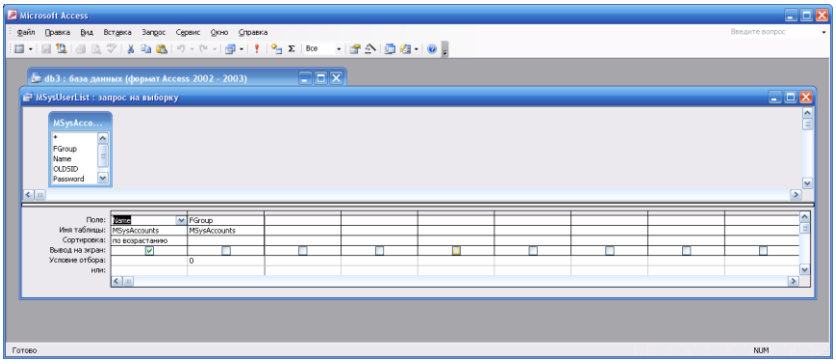

Рисунок 3.25. Вигляд запиту MSysUserList

Проведемо перевірку роботи сформованих запитів. Так, запустимо на виконання перший запит. В результаті отримуємо наступний звіт:

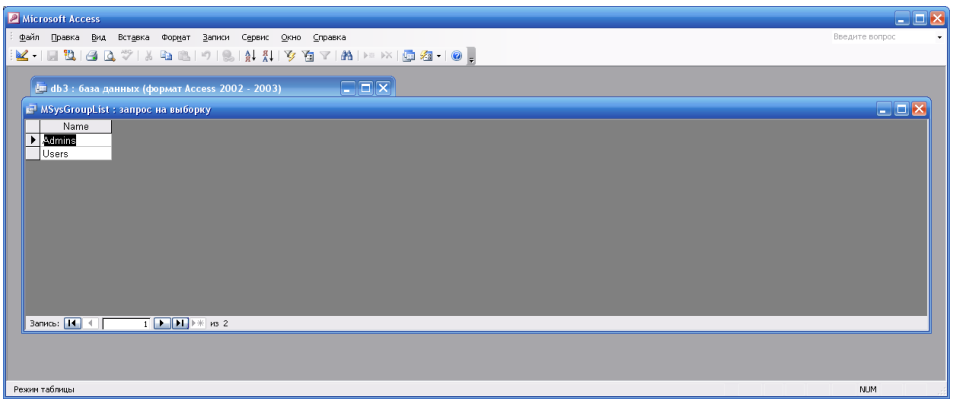

Рисунок 3.26. Вигляд звіти сформованого запитом MSysGroupList

Проведемо перевірку працездатності запиту MSysGroupMembers, рисунок 3.27.

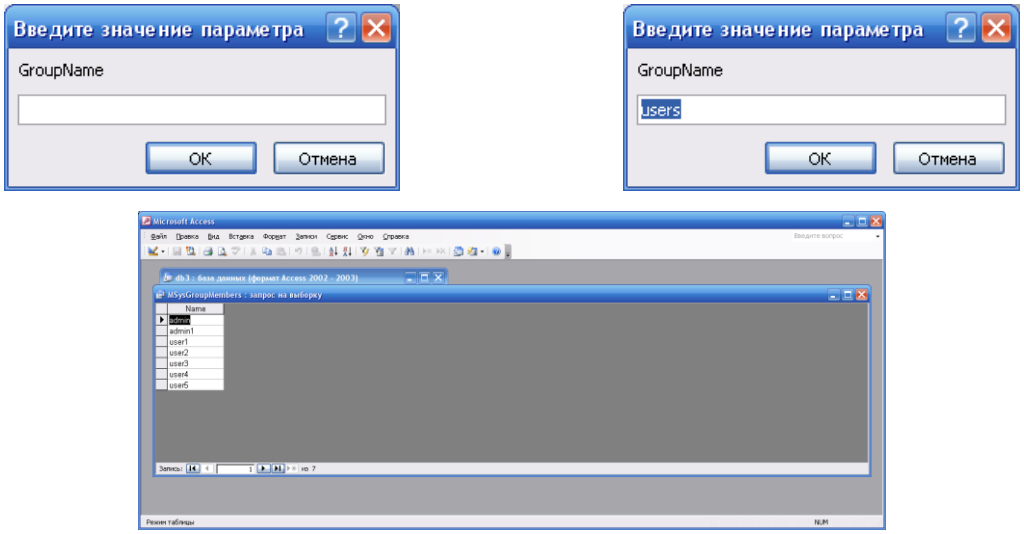

Рисунок 3.27

На рисунку 3.28 Представимо результати роботи запиту MSysUserList

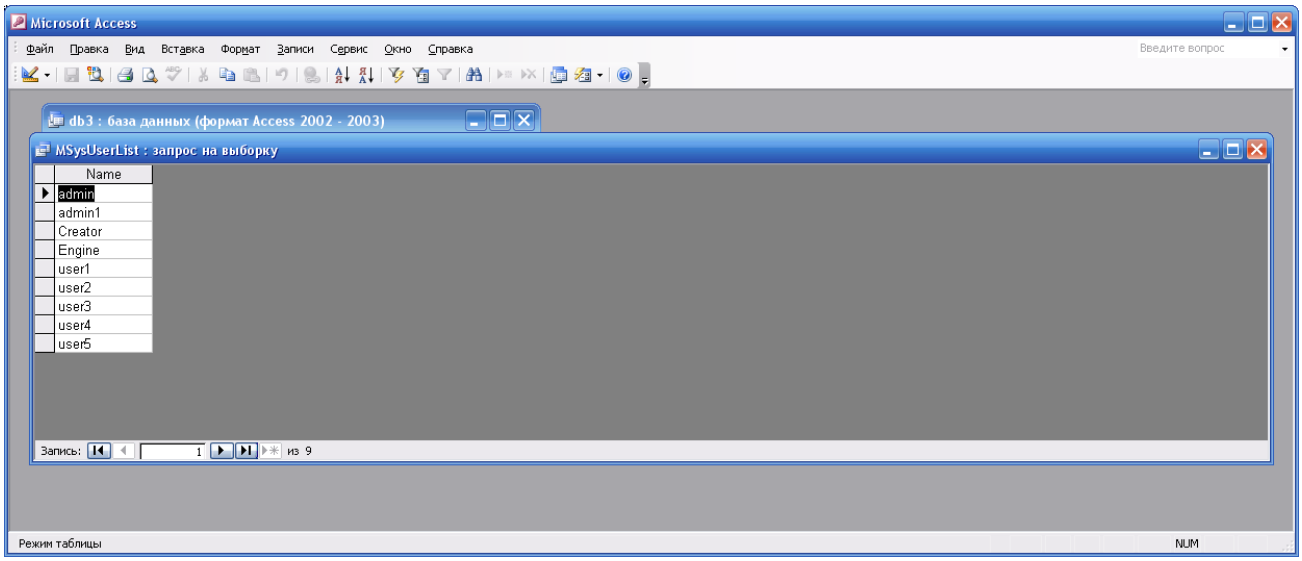

Рисунок 3.28

В результаті проведених досліджень було проведено налаштування підсистеми розмежування прав доступу до БД засобами MS Office Access 2003.

Висновки до третього розділу

В результаті проведених досліджень розроблено та реалізовано інтерфейсну частину підсистеми розмежування доступу до інформаційних ресурсів у вигляді форм для процедури аутентифікації користувачів БД.

Проведено адміністрування БД для встановлених груп користувачів. Проведено чітке розмежування прав доступу до ресурсів бази даних, налаштовані обмеження щодо дій зі складовими БД. Визначено права власності користувачів системи.

Сформовано запити на формування звітів по користувачам БД та проведено перевірку працездатності сформованих звітів.

Розділ 4. ЗАСОБИ РОЗРОБКИ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ АНАЛІЗІ ПРОБЛЕМИ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ДАНИХ

4.1. Основні засоби розробки

У програмного забезпечення, як у живої істоти є свій життєвий цикл. Життєвий цикл ПЗ - стадії, що проходить програмний продукт від появи ідеї до її реалізації в коді, імплементації у бізнес і подальшої підтримки. Моделі життєвого циклу багато в чому зумовлюють і методології розробки ПЗ [27].

На сьогодні до основних засобів розробки необхідно віднести:

- середовище розробки Visual Studio;
- графічний інструмент для роботи з базами даних MySQL Workbench;
- систему керування базами даних MySQL;
- мову програмування C#.

Розглянемо кожен з наведених засобів розробки біль детально.

Середовище розробки Visual Studio. Visual Studio IDE – це повнофункціональна платформа розробки для багатьох операційних систем, а також для Інтернету та хмари. Вона дозволяє користувачам легко користуватися інтерфейсом та можливостями середовища, щоб вони могли швидко та точно писати код.

У комплект входять наступні основні компоненти:

- Visual Basic.NET – для розробки додатків на VisualBasic;
- Visual C ++ – на традиційній мові C ++;
- Visual C # – на мові C # (Microsoft);
- Visual F# – на F# (Microsoft Developer Division).

Функціональна структура середовища включає в себе:

- редактор вихідного коду, який включає в себе безліч додаткових функцій, як автодоповнення IntelliSense, рефракторинг коду тощо;
- налагоджувач коду;

- редактор форм, передбачений для спрощеного конструювання графічних інтерфейсів;

- веб-редактор;

- дизайнер класів;

- дизайнер схем баз даних.

Visual Studio також дозволяє створювати та підключати сторонні доповнення (плагіни) для розширення функціональних можливостей практично на кожному рівні, включаючи додавання систем контролю версій (Subversion і VisualSourceSafe), додавання нових наборів інструментів (для редагування та візуального проектування коду на предметно-орієнтованих мовах програмування або інструментів для інших аспектів процесу розробки програмного забезпечення).

За допомогою Visual Studio IDE розробники також мають доступ до безлічі інструментів налагодження. Вони допомагають їм в аналізуванні помилок та їх швидкій та ефективній діагностиці. Таким чином вони можуть впевнено розгортати свої програми, знаючи, що вони усунули все, що може призвести до помилок у роботі.

Більше того, Visual Studio IDE також є тестовою платформою. Тут розробники можуть імітувати, як їх програми працюватимуть у своїх цільових середовищах, і гарантувати, що вони роблять це безперебійно після їх розгортання.

Середовище пропонує інтерфейс з багатьма можливостями, який при цьому інтуїтивно зрозумілий для користувача (Рисунок 4.1).

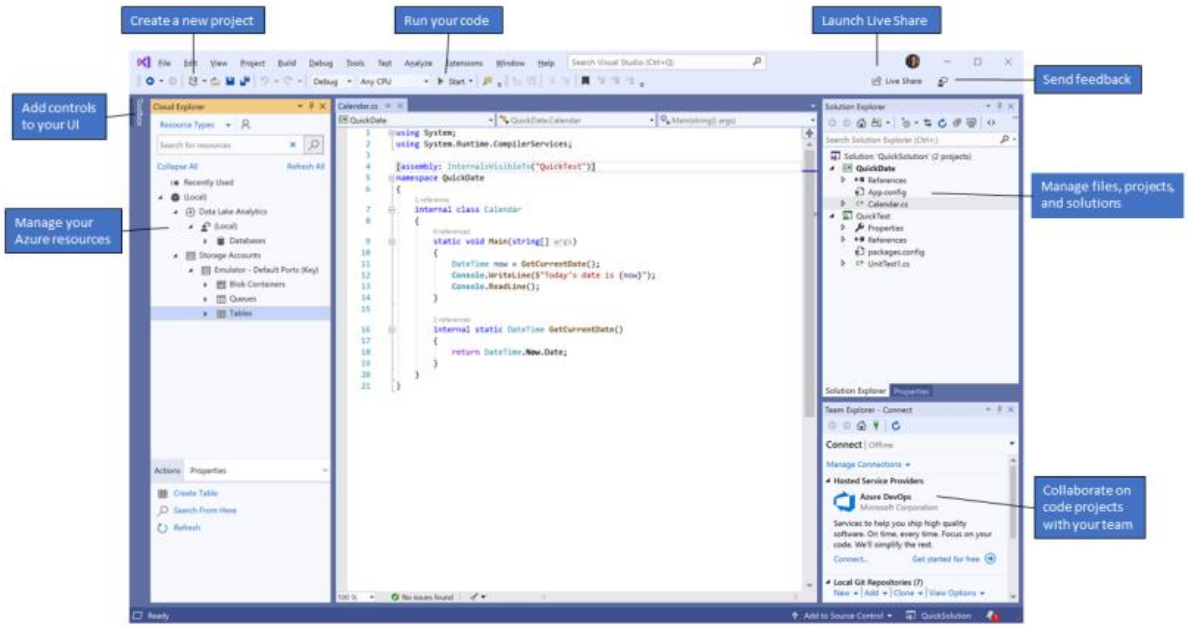

Рисунок 4.1.1. Середовище розробки Visual Studio

За допомогою Visual Studio IDE користувачам надається допомога з написанням коду в реальному часі незалежно від мови програмування, яку вони використовують. Вбудована IntelliSense платформа пропонує підказки та описи API та автоматично доповнює рядки для більшої ефективності в написанні коду.

Крім того, Visual Studio IDE зберігає місце останньої модифікації коду розробником, якщо той досліджує решту свого коду.

Швидке налагодження. Пошук та діагностика помилок може бути проблемою, однак у Visual Studio IDE є безліч інструментів, які полегшують її. Платформа підтримує налагодження для всіх включених мов. Процес також може здійснюватися як локально, віддалено, так і в середині розробки. Це дозволяє розробникам розгортати програми на робочому столі або емулятори на мобільних пристроях та інші методи налагодження.

Можливості тестування. Visual Studio IDE оснащений платформою для тестування додатків, яка дозволяє розробникам переконатися, що вони готові розгорнути високоякісні продукти. Вони можуть це робити на бажаній мові або фреймворку.

Командна робота. Розробники Visual Studio IDE розуміють, що зазвичай є необхідність для розробки в команді. Ось чому платформа має спільні можливості, що підвищують продуктивність команди. Ці інструменти тісно інтегровані з життєвим циклом розробки проектів.

Крім того, Visual Studio IDE добре працює в режимі спільної роботи незалежно від бажаної платформи кожного учасника.

Параметри налаштування. Visual Studio IDE пропонує налаштування для кожного користувача. Вони можуть розширити функціональні можливості платформи за допомогою розширень та доповнень, доступних у Visual Studio Marketplace. Розробники навіть можуть публікувати власні розширення.

Visual Studio дозволяє отримувати доступ до документації MSDN прямо з середовища IDE. У разі, наприклад, виникнення сумнівів з приводу призначення того 50 чи іншого ключового слова під час роботи з текстовим редактором, можна виділити це ключове слово і натиснути клавішу , в результаті чого Visual Studio автоматично підключиться до MSDN і відобразить відповідні розділи довідки. Аналогічно, якщо потрібно подивитися, що означає та чи інша помилка компіляції, потрібно виділяти повідомлення з помилкою і натиснути.

Також Visual Studio містить графічні редактори і конструктори XML, забезпечує підтримку розробки програм Windows, орієнтованих на мобільні пристрої, підтримку розробки програм Microsoft Office і Windows Workflow Foundation, містить вбудовану підтримку рефакторинга коду і інструменти візуального конструювання класів.

Графічний інструмент для роботи з базами даних MySQL Workbench. MySQL Workbench – це графічний інструмент для роботи з серверами та базами даних MySQL (Рисунок 4.2). MySQL Workbench повністю підтримує версії сервера MySQL 5.6 та новіші. Він також сумісний із старими версіями сервера 5.x MySQL, за винятком певних ситуацій

(наприклад, відображення списку процесів) через змінені системні таблиці. Він не підтримує версії сервера MySQL 4.x.

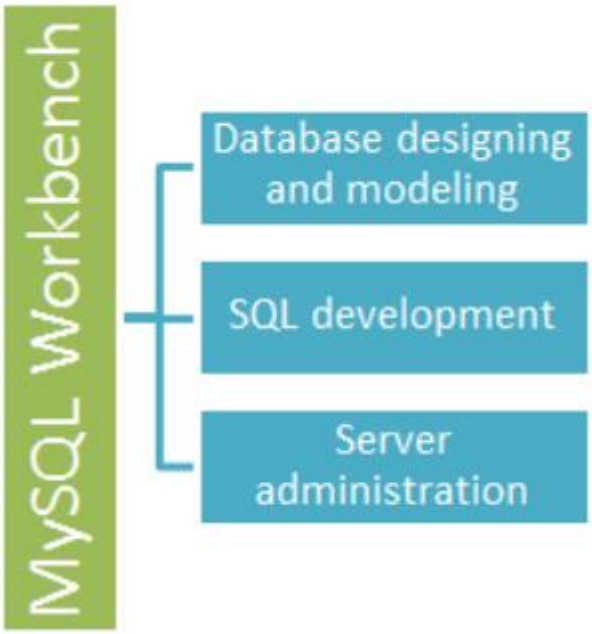

Рисунок 4.1.2. Структура MySQL Workbench

Функціонал MySQL Workbench охоплює п’ять основних тем:

- розробка SQL: дозволяє створювати та керувати з'єднаннями з серверами баз даних. Поряд з можливістю налаштування параметрів з'єднання, MySQL Workbench надає можливість виконувати запити SQL на підключеннях до бази даних за допомогою вбудованого редактора SQL;

- моделювання даних (дизайн): дозволяє створювати моделі схем баз даних графічно, редагувати всі аспекти баз даних за допомогою повноцінного редактора таблиць. Редактор таблиць надає прості у використанні засоби для редагування таблиць, стовпців, покажчиків, тригерів, розділення, параметрів, вкладок та привілеїв, рутинних програм та представлень;

- адміністрування сервера: дає змогу керувати екземплярами сервера MySQL шляхом адміністрування користувачів, виконання резервного

копіювання та відновлення, перевірки даних аудиту, перегляду стану баз даних та контролю за роботою сервера MySQL;

- міграція даних: Дозволяє переходити з Microsoft SQL Server, Microsoft Access, Sybase ASE, SQLite, SQL Anywhere, PostreSQL та інших RDBMS таблиць, об'єктів і даних до MySQL. Міграція також підтримує перехід від попередніх версій MySQL до останніх версій. MySQLworkbench має зрозумілий інтерфейс та інструменти, які дозволяють розробникам та адміністраторам баз даних візуально створювати фізичні моделі дизайну баз даних (Рисунок 4.3), які можна легко перевести в бази даних MySQL.

MySQL Workbench підтримує створення декількох моделей в одному середовищі.

Він підтримує всі об'єкти, такі як таблиці, представлення даних, збережені процедури, тригери тощо, які складають базу даних.

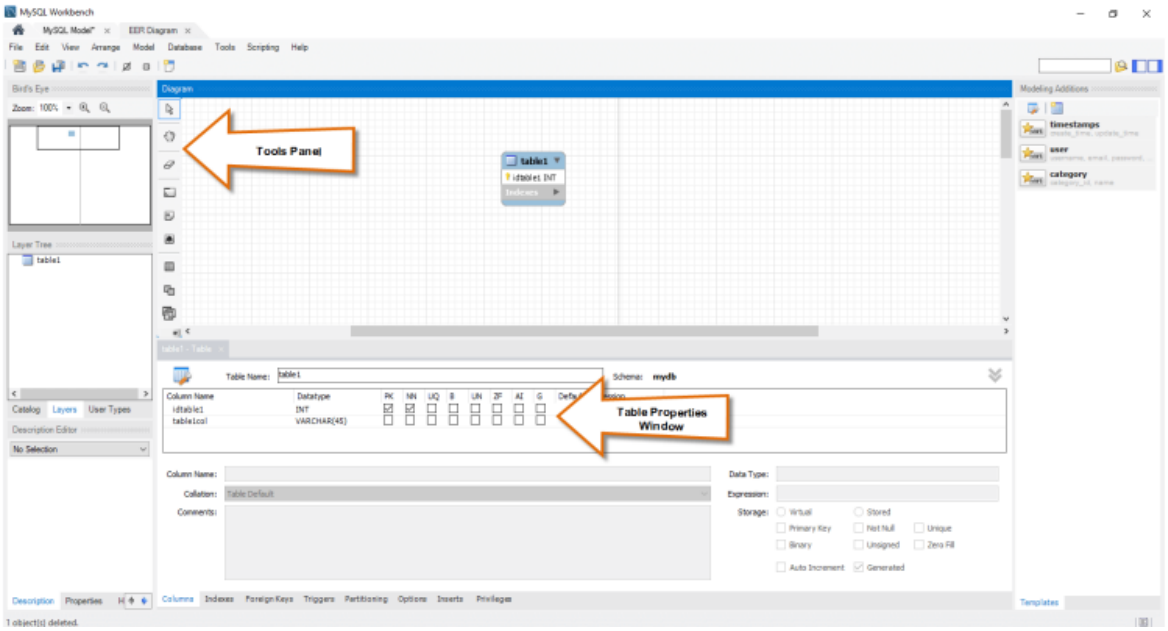

Рисунок 4.1.3. Вікно моделювання в MySQL Workbench

Система керування базами даних MySQL. MySQL і SQL не є однаковими. SQL в MySQL розшифровується як структурована мова запитів. Це стандартна мова, яка використовується для взаємодії з базою даних. MySQL – це система управління реляційними базами даних, яка допомагає

маніпулювати базою даних, що зберігаються в різних таблицях на комп'ютері. Система керування базами даних MySQL – реляційна система керування базами даних (СКБД). Є альтернативою як комерційним СКБД (Oracle Database, Microsoft SQL Server, IBM DB2 та інші), так і СКБД з відкритим кодом (PostgreSQL) [28]. Порівняно з іншими проектами з відкритим кодом, такими як Apache або FreeBSD, MySQL не контролюється якоюсь однією компанією, її розробка можлива завдяки співпраці багатьох людей та компаній, які хочуть використовувати цю СКБД та впроваджувати у неї найновіші досягнення. Сервер MySQL написаний на мові програмування C. Зазвичай розповсюджується у вигляді набору текстових файлів із сирцевим кодом. Для інсталяції необхідно відкомпілювати файли на своєму комп’ютері і скопіювати в деякий каталог. Весь процес детально описаний в документації. В MySQL є підтримка індексів наступних типів: B-дерево, хеш, Rдерево, GiST, GIN. При необхідності можна створити нові типи індексів [29].

MySQL підтримує великий набір вбудованих типів даних:

- числові типи;
- цілі;
- з фіксованою крапкою;
- з нефіксованою крапкою;
- грошовий тип;
- символьні типи довільної довжини;
- двійкові типи (включаючи BLOB);
- типи дата/час;
- булевий тип;
- перерахування;
- геометричні примітиви;
- мережеві типи;
- IP і IPv6-адреси;
- CIDR-формат;

- MAC-адреса;

- UUID-ідентификатор;

- XML-дані;

- JSON-дані;

- масиви;

- OID-типи;

- псевдотипи. Тригери визначаються як функції, що ініціюються DML-операціями. Наприклад, операція INSERT може запускати тригер, перевіряючий доданий запис на відповідності певним умовам. При написанні функцій для тригерів можуть використовуватися різні мови програмування. Тригери асоціюються з таблицями. Множинні тригери виконуються в алфавітному порядку [30].

Мова програмування C#. C# (вимовляється «See Sharp») – це проста, сучасна, об'єктно-орієнтована і строго типізована мова програмування. C# має коріння з сімейства мов C і є знайомою програмістам на C, C ++ та Java. C# стандартизована ECMA International як стандарт ECMA-334 та ISO / IEC як стандарт ISO / IEC 23270. Компілятор Microsoft C# для .NET Framework – це відповідна реалізація обох цих стандартів [31].

C# – об'єктно-орієнтована мова, але C# додатково включає підтримку компонентно-орієнтованого програмування. Сучасний дизайн програмного забезпечення все більше покладається на програмні компоненти у вигляді автономних та самоописуючих пакетів функціональності. Ключовим для таких компонентів є те, що вони представляють модель програмування із властивостями, методами та подіями; вони мають атрибути, які надають декларативну інформацію про компонент; і вони включають власну документацію. C # надає мовні конструкції для повної підтримки цих концепцій, що робить C# дуже природною мовою, для якої можна створювати та використовувати програмні компоненти [32]. Деякі особливості C# (Рисунок 3.4) допомагають створювати надійні та довговічні програми.

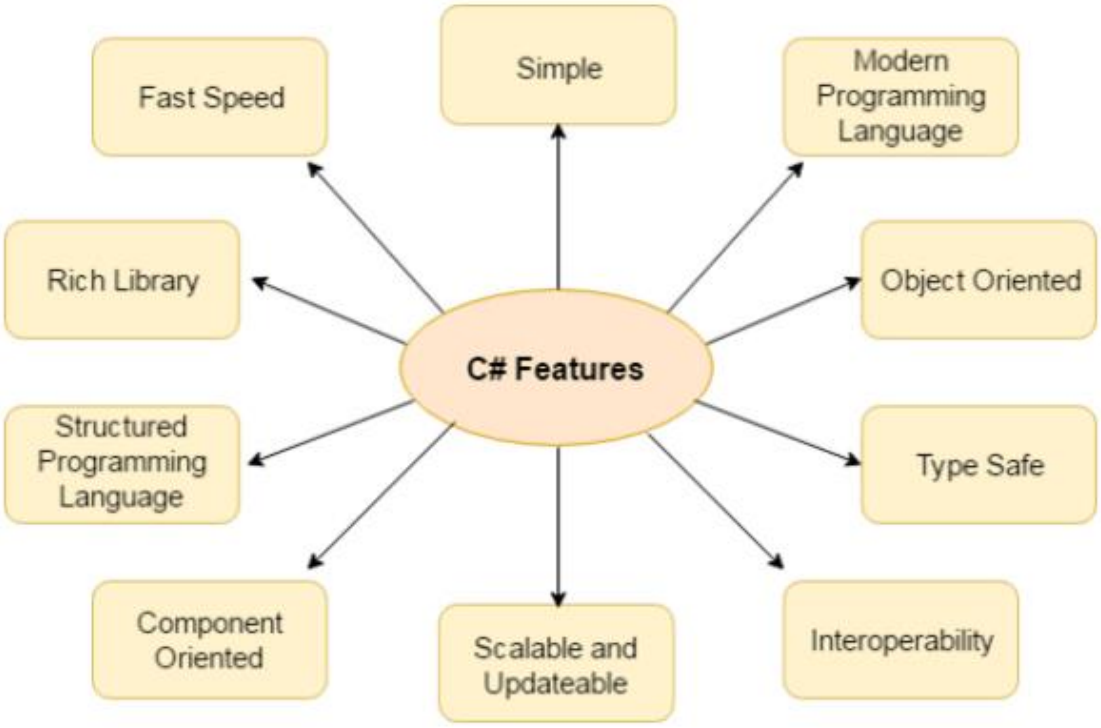

Рисунок 4.1.4. Головні особливості мови C#

Збір сміття автоматично відновлює пам'ять, зайняту невикористаними об'єктами; обробка винятків забезпечує структурований та розширюваний підхід до виявлення та відновлення помилок; безпечна конструкція мови не дає змоги читати з неініціалізованих змінних, індексувати масиви за їх межами тощо.

C# має систему уніфікованих типів. Усі типи даних C#, включаючи примітивні типи, такі як int та double, успадковуються від єдиного типу об'єкту. Таким чином, усі типи розділяють набір загальних операцій, і значення будь-якого типу можуть зберігатися, транспортуватися та керуватися ними послідовно. Крім того, C# підтримує як визначені користувачем типи, що передаються за посиланням, так і типи, що передаються за значенням, що дозволяє динамічно розподілення об'єктів, а також вбудовувано зберігання легких структур [33].

Щоб переконатися, що програми та бібліотеки C # можуть розвиватися з часом, в розробці C# було зроблено великий акцент на версіях. Багато мов

програмування приділяють цьому питанню мало уваги, і, як наслідок, програми, написані цими мовами, ламаються частіше, ніж потрібно, коли впроваджуються новіші версії залежних бібліотек. Аспекти дизайну C#, на які безпосередньо впливали можливі майбутні оновлення версій, включають окремі віртуальні модифікатори та модифікатори, що перевизначаються, правила перевантаження методів та інші особливості. Ключовими поняттями в C# є програми, простори імен, типи, члени та збірки. Програми C# складаються з одного або декількох вихідних файлів. Програми декларують типи, які містять члени. Типи можуть бути організовані в простори імен. Класи та інтерфейси – приклади типів. Поля, методи, властивості та події – приклади членів. Коли програми C# компілюються, вони фізично упаковуються в збірки. Зазвичай збірки мають розширення .exe або .dll, залежно від того, реалізують вони програми чи бібліотеки.

Збірки містять виконуваний код у вигляді інструкцій Intermediate Language (IL) та символьну інформацію у вигляді метаданих. Перед його виконанням код IL в зборці автоматично перетворюється на специфічний для процесора код компілятором Just-In-Time (JIT).

.NET Common Language Runtime. .NET фреймворк відповідає за виконання програми C# в системі. Фреймворк – це поєднання мови програмування, що виконується, та набору бібліотек класів. Впровадження Common Language Infrastructure (CLI) здійснюється за допомогою Common Language Runtime (CLR).

Microsoft .NET Framework складається з таких основних компонентів:

- Common Language Runtime;

- бібліотека базових класів;

- користувацький інтерфейс програми;

- Діаграма архітектури .NET фреймфорку разом із мовою програмування C# наведена нижче (Рисунок 4.5).

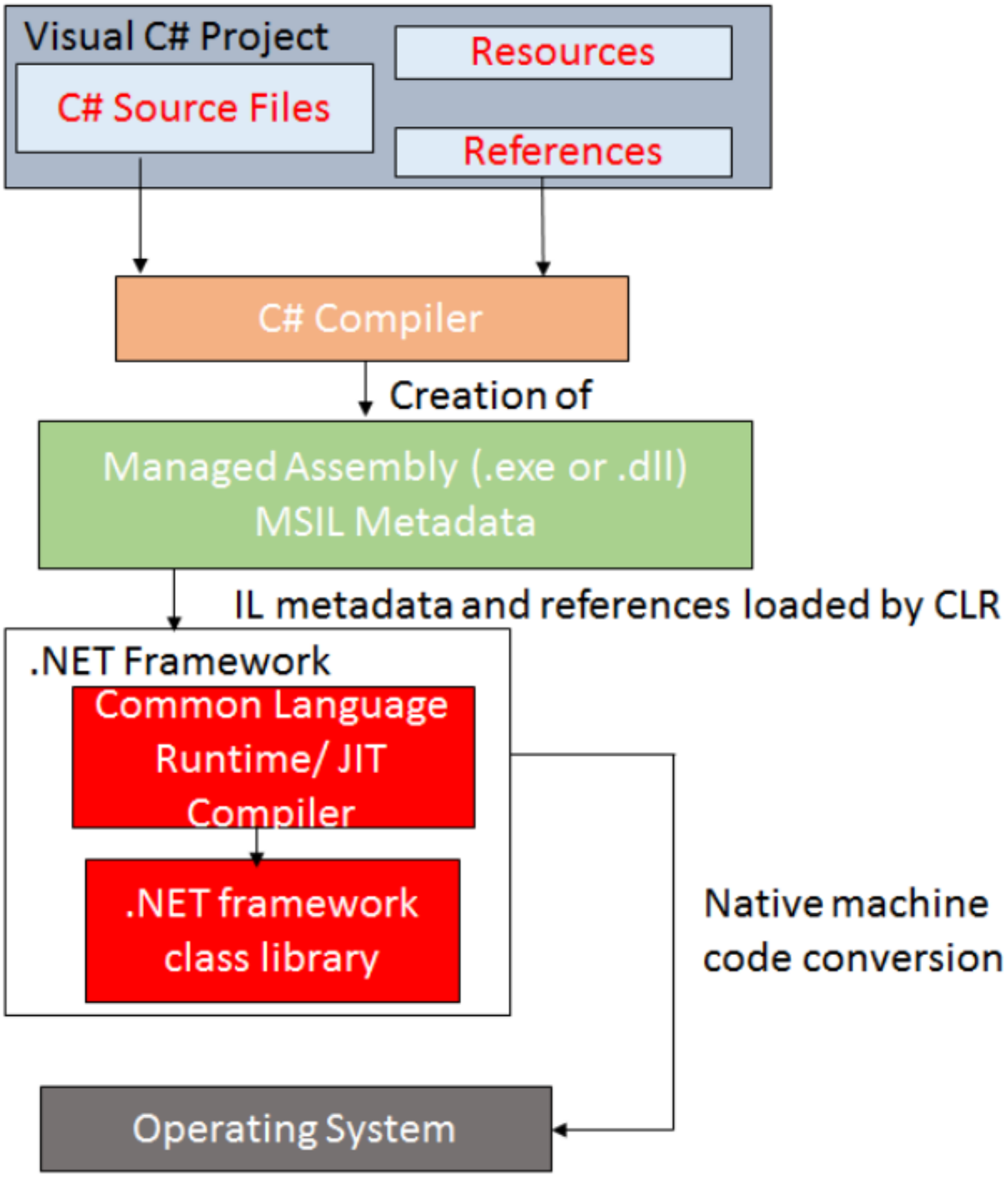

Рисунок 4.1.5. Архітектура .NET фреймфорку

4.2. Архітектура програмної системи

Для зв’язку додатку із базою даних MySQL був використаний MySqlConnector.

MySqlConnector – постачальник даних ADO.NET для MySQL Server, MariaDB, Percona Server, Amazon Aurora, база даних Azure для MySQL, Google Cloud SQL для MySQL тощо. Він забезпечує реалізацію

DbConnection, DbCommand, DbDataReader, DbTransaction – класи, необхідні для запиту та оновлення баз даних з керованого коду [34].

Перш ніж починати роботу с MySqlConnector необхідно в проекті додати посилання до нього (Рисунок 4.2.1).

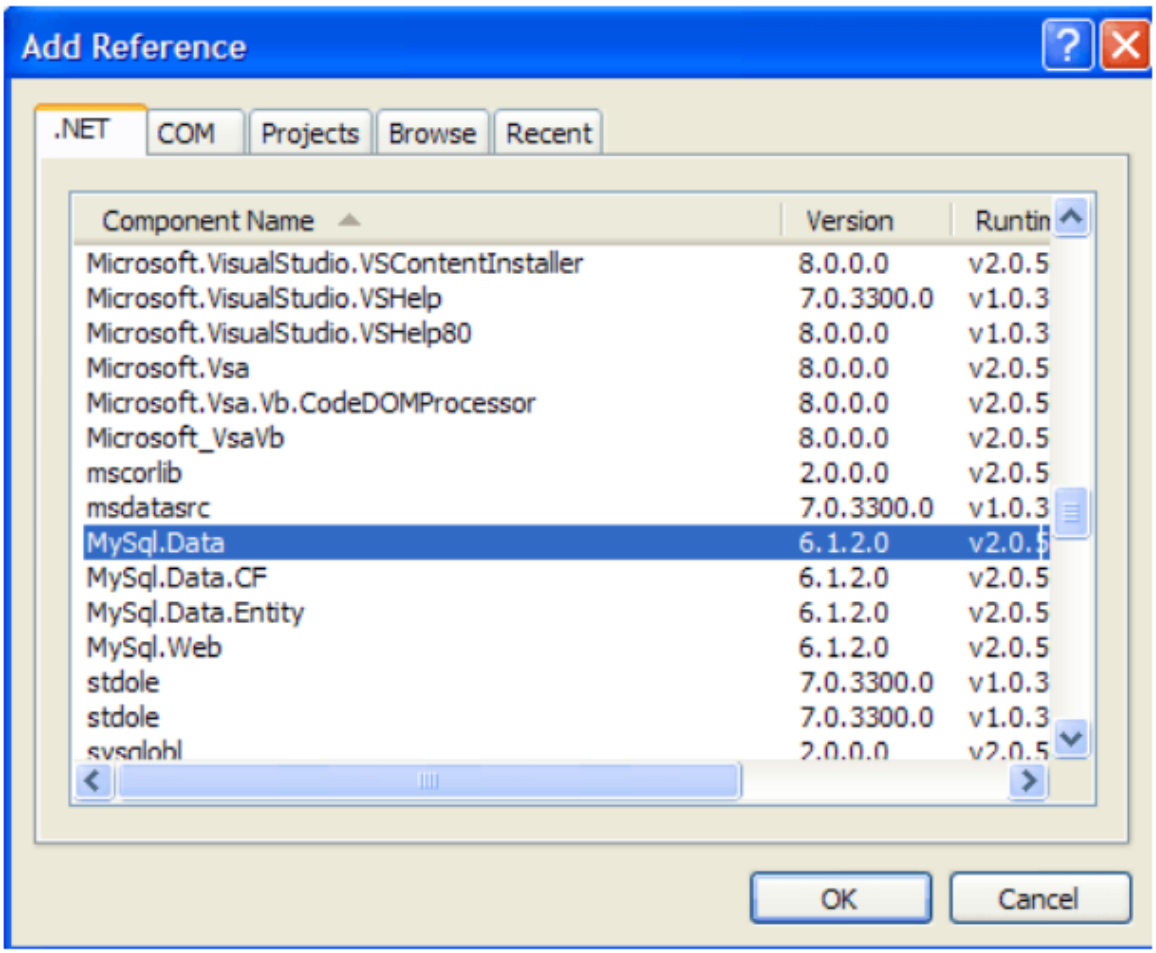

Рисунок 4.2.1. Додавання MySQLConnector до проекту

Для роботи з MySQL необхідно у коді додати бібліотеку MySQL Connector: using MySql.Data.MySqlClient;

Для запитів і здійснення маніпуляцій за базами даних необхідно використовувати об’єкти класу MySqlConnection.

Вся взаємодія між .NET-додатком та сервером MySQL здійснюється шляхом об’єкта MySqlConnection при використанні класичного протоколу MySQL. Перш ніж програма може взаємодіяти з сервером, вона повинна створити об'єкт MySqlConnection, створити налаштування та відкрити MySqlConnection.

Для з’єднання з необхідною базою даних необхідно використати такі змінні:

– connection: буде використовуватися для відкриття з'єднання з базою даних; – server: вказує, де розміщується наш сервер, у нашому випадку - localhost;

– database: це ім'я бази даних, яку ми будемо використовувати; – uid: це наше ім'я користувача MySQL;

– password: це наш MySQL пароль;

– connectionString: містить рядок з'єднання для підключення до бази даних і буде призначений змінній з'єднання. Для виконання оператора Select додаємо ще кілька кроків і використовуємо метод ExecuteReader, який поверне об’єкт dataReader для зчитування та зберігання даних або записів.

– відкриваємо з’єднання із базою даних;

– створюємо SQL команду;

– присвоюємо з’єднання і запит до команди. Це можна зробити за допомогою конструктора або за допомогою методів Connection та CommandText класу MySqlCommand;

– створюємо об’єкт MySqlDataReader для читання вибраних записів / даних; – виконуємо команду; – закриваємо об’єкт dataReader;

– закриваємо з'єднання.

Приклад коду одного зі з’єдань і виконання команди select із базою даних в даному проекті наведено нижче. (Рисунок 4.2.2).

```
string Connect = "Database=diplom;Data Source=localhost;User Id=root;Password=64nitoda";
string sql = "SELECT * FROM users WHERE  username='" + login.Text + "' and password='" + password.Text + "'";
MySqlConnection connection = new MySqlConnection(Connect);
MySqlCommand sqlCom = new MySqlCommand(sql, connection);
connection.Open();
sqlCom.ExecuteNonQuery();
MySqlDataAdapter dataAdapter = new MySqlDataAdapter(sqlCom);
DataTable dt = new DataTable();
dataAdapter.Fill(dt);
var myData = dt.Select();
if (dt.Rows.Count == 1)
```

Рисунок 4.2.2. Код з’єднання з базою даних

Необхідно завжди відкривати з'єднання перед тим, як робити запит до нашої таблиці та закривати його відразу після закінчення роботи, щоб звільнити ресурси та вказати, що це з'єднання більше не потрібно. Відкриття та закриття підключення до бази даних дуже просте, однак, завжди краще використовувати обробку винятків перед відкриттям з'єднання або закриттям, щоб виявити помилки та вирішити їх.

4.3. Опис бази даних

База даних складається з 4 таблиць: «users» (Користувачі), «number_of_attacks» (кількість атак), «db_attacks» (атаки на базу даних), «db_savings» (збереження баз даних). Таблиця «users» призначена для зберігання даних про користувачів системи. В ній зберігається логін, пароль, ім’я користувача, роль (адміністратор, користувач), посада, організація в якій працює користувач (Рисунок 4.3.3).

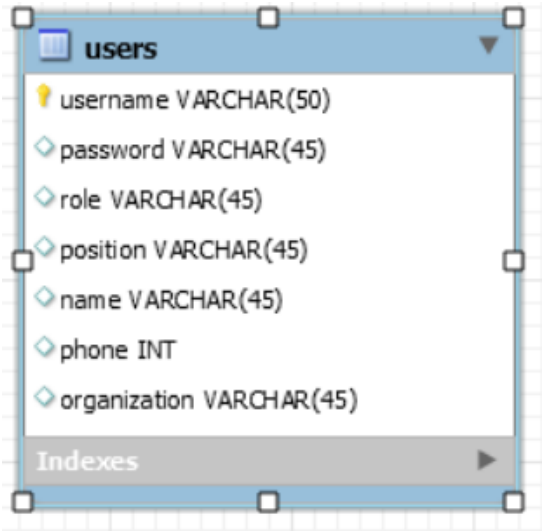

Рисунок 4.3.3. Структура таблиці «users»

Таблиця «number_of_attacks» призначена для зберігання даних про атаки на систему. Зокрема, вона зберігає дані про час атаки, кількість атак, і порти на які було здійснено атаки (Рисунок 4.3.4).

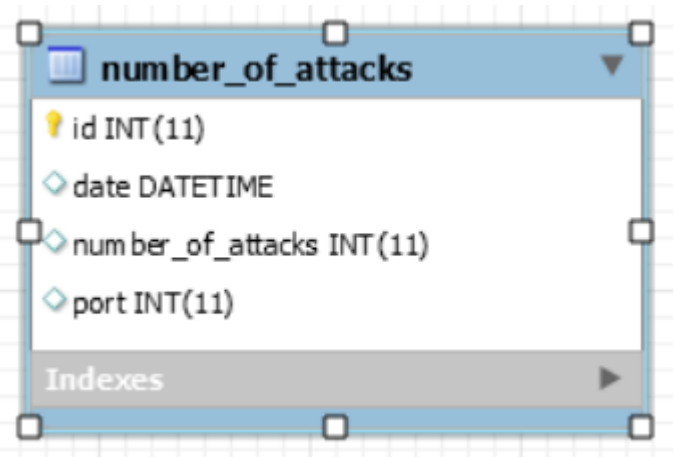

Рисунок 4.3.4. Структура таблиці «number_of_attacks»

Таблиця “db_attacks” призначена для зберігання даних про атаки на бази даних. Зокрема, вона зберігає дані про час атак, їх кількість, а також імена баз даних на які була здійснена та чи інша атака (Рисунок 4.3.5).

Рисунок 4.3.5. Структура таблиці «db_attacks»

Таблиця «db_savings» призначена для зберігання інформації про історію збережень баз даних. Зокрема, про те, яку базу даних було збережено, дату збереження і хто виконував збереження (Рисунок 4.3.6).

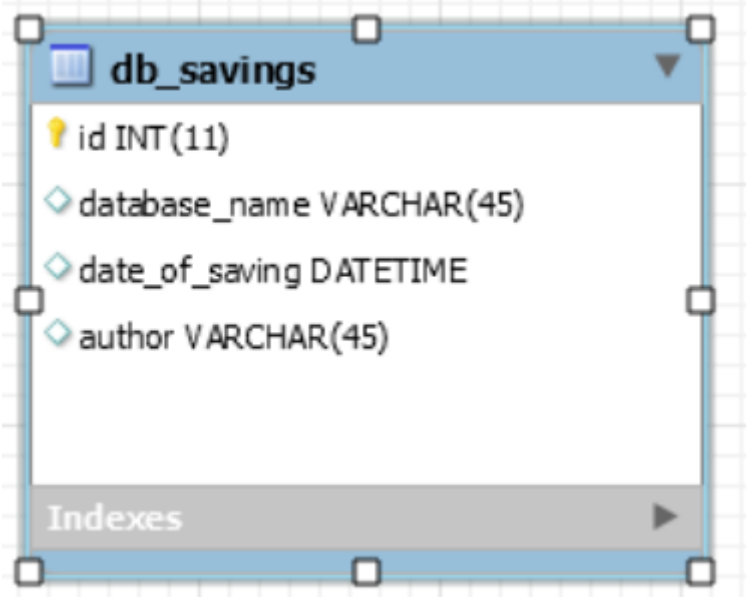

Рисунок 4.3.6. Структура таблиці «db_savings» На рисунку нижче також представлено приклад заповнення однієї з таблиць бази даних. (Рис. 4.3.7).

Result Grid | Filter Rows: | Edit:

id	date	number_of_attacks	port
20	2019-10-07 02:56:38	29	7
21	2019-10-07 07:11:32	18	11
22	2019-10-07 12:50:26	5	7
23	2019-10-07 14:23:07	25	80
24	2019-10-07 14:37:08	28	80
25	2019-10-07 16:51:16	9	53
26	2019-10-07 22:09:00	37	53

Рисунок 4.3.7. Приклад заповнення таблиці «number_of_attacks»

Висновки до четвертого розділу

В даному розділі було розглянуто основні інструменти для розробки системи підтримки прийняття рішень при аналізі проблеми інформаційного забезпечення безпеки даних та систем, до яких необхідно віднести: середовище розробки Visual Studio, графічний інструмент для роботи з базами даних MySQL Workbench, систему керування базами даних MySQL, мову програмування C#.

C# було вибрано основною мовою для розробки системи, оскільки вона має необхідний набір переваг, якими має володіти сучасна мова програмування, і є оптимальним вибором для написання подібних систем підтримки прийняття рішень. Дана мова програмування має безліч переваг, такі як величезний набір бібліотек, які підключаються до проектів, що можуть полегшити розробку систем і також має можливості для розробки та проектування сучасного користувацького інтерфейсу. Було використано бібліотеку для побудови користувацького інтерфейсу ComponentOne. А також було використано бібліотеки для побудови зв’язку з базою даних для проведення маніпуляцій з таблицями.

Для розробки даної системи було необхідним провести дослідження можливих сучасних інструментів та засобів розробки для її створення. Це впливає на ефективність роботи при її розробці, якість кінцевого програмного продукту, та його швидкість у роботі.

В якості СКБД було обрано систему MySQL, яка наразі є найбільш популярною та затребуваною реляційною СКБД. Вона має безліч зручних інструментів для управління та підтримки баз даних, в тому числі і графічний інструмент MySQL Workbench.

При цьому було застосовано найбільш ефективний та швидкий спосіб для налагодження зв’язку між базою даних та додатком.

Для розробки і написання програмного продукту ми використовували інтегроване середовище розробки Visual Studio 2017, тому що воно надає

максимальний набір інструментів для ефективного написання коду. Має можливості для автоматичного виправлення помилок при роботі, а також дає підказки під час написання коду, що дозволяє максимально швидко виконувати роботу. Для програмування на мові C# дане середовище є найоптимальнішим.

ВИСНОВКИ

Кваліфікаційна робота присвячена розробці підсистеми розмежування доступу до баз даних, в ході виконання якої були отримані наступні теоретичні та практичні результати:

1. Проведено аналіз моделей, методів та засобів побудови баз даних в інформаційно-комунікаційних системах. Аналіз інформаційних систем, що функціонують на підприємствах виявив, що останні використовують як сховище даних засіб Microsoft Office Access. Представлено механізм організації доступу до баз даних.

2. На основі можливостей апаратного забезпечення та фінансового забезпечення, прийнято рішення використовувати спосіб аутентифікації та ідентифікації за допомогою знання претендентом інформації, яку знає тільки легальний користувач – паролю та логіну.

3. Розроблено підсистему розмежування доступу до баз даних. Побудовано та описано функціональну структуру інформаційної підсистеми. Визначено, що для розмежування доступу до баз даних необхідно, щоб система могла:

- забезпечувати можливість вводу інформації ;
- забезпечувати перегляд інформації;
- контролювати правильність вводу інформації;
- забезпечувати можливість пошуку необхідної інформації;
- забезпечувати можливість сортування інформації.

4. Також визначено, що для розмежування доступу до баз даних необхідно створити групу користувачів із логінами, паролями і своїми правами доступу до бази даних. Побудовано діаграму активності і використання системи. Обґрунтовано вибір технології побудови підсистеми розмежування доступу до баз даних.

5. В результаті проведених досліджень розроблено та реалізовано інфтерфейсну частину підсистеми розмежування доступу до інформаційних ресурсів у вигляді форм для процедури аутентифікації користувачів БД.

6. Проведено адміністрування БД для встановлених груп користувачів. Проведено чітке розмежування прав доступу до ресурсів бази даних, налаштовані обмеження щодо дій зі складовими БД. Визначено права власності користувачів системи.

7. Сформовано запити на формування звітів по користувачам БД та проведено перевірку працездатності сформованих звітів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ ISO/IEC 2382:2017 Інформаційні технології. Словник термінів (ISO/IEC 2382:2015, IDT). URL: http://online.budstandart.com/ru/catalog/doc-page?id_doc=75076

2. База даних. URL: http://surl.li/bqulg

3. Безпека бази даних. URL: https://php.ru/manual/security.database.html

4. Про інформацію: Закон України від 02.10.1992 №2658-XII у редакції від 01.01.2017. URL: http://zakon2.rada.gov.ua/laws/show/2657-12

5. Основні аспекти безпеки СУБД: що знати. URL: https://tproger.ru/articles/db-security-basics/

6. Microsoft Office Access. URL: https://uk.wikipedia.org/wiki/Microsoft_Access

7. Що таке база даних oracle (oracle db)? - визначення з техопедії. In-The News. 2022. URL: https://uk.theastrologypage.com/oracle-database

8. Що таке база даних? URL: https://www.oracle.com/ru/database/what-is-database

9. Що таке SQL Server? URL: https://uk.education-wiki.com/7023109-what-is-sql-server

10. Берко О.Ю. Системи баз даних та знань. Книга 1. Організація баз даних та знань: підручник. 2-ге вигляд. / А.Ю. Берко, О.М. Верес, В.В. Пасічник. Вид-во: "Магнолія2006", 2015. 440 с.

11. Рад Б.Я., Цехановський В.В., Чортовський В.Д. Бази даних: теорія та практика: підручник для бакалаврів. М: Юрайт, 2013. 463 с.

12. Швець М.Ю., Заруб Д.С., Хохлов Ю.В. Порівняння SQL та NoSQL баз даних. Інформатика, обчислювальна техніка та автоматизація. 2018. Т. 29. №6. С. 21-25.

13. Яка АІБС краща? URL: https://core.ac.uk/download/pdf/73907287.pdf

14. Bringing MySQL to the web: веб-сайт. URL: https://core.ac.uk/download/pdf/73907287.pdf

15. Мова програмування Visual Basic for Applications. URL: https://inlnk.ru/Jj25Qe

16. Основи управління інформаційною безпекою: навч. посібник/О.М. Гребенюк, Л.В. Рибальченко. Дніпро: Дніпроп. держ. унт внутріш. справ, 2020. 144 с.

17. The Most Popular Passwords of 2019. URL: https://nordpass.com/blog/top-worst-passwords-2019

18. Типи ідентифікації користувачів. URL: http://www.infobezpeka.com/publications/?id=92

19. Рібальченко Л.В., Косіченко О.О. Проблеми безпеки персональних даних України. Регіональна економіка. Вінниця. 2019. с. 57-62.

20. Europol. (2018). Public Awareness and Prevention Guides. URL: https://www.europol.europa.eu/activitiesservices/public-awareness-and-preventionguides

21. Про криптографічний та технічний захист інформації. Закон України. URL: https://ips.ligazakon.net/document/NT1819

22. Інформаційні системи та технології у фінансових установах / О.В. Олійник, В.М. Шацька, навчальний посібник. Львів: Новий Світ-2000, 2006. 436 с.

23. Стеценко, І.В. Моделювання систем: навч. посіб. М-во освіти і науки України, Черкаси. держ. технол. ун-т. Черкаси: ЧДТУ, 2010. 399 с.

24. Сидоренко В.В., Константинова Л.В., Смірнов С.А. Організація баз даних: Навчальний посібник. Кропивницький: ЦНТУ, 2018. 274 с.

25. SQL у Access: основні поняття, глосарій та синтаксис. URL: https://inlnk.ru/Bpewyx

26. Адміністрування баз даних та автоматизація. URL: https://inlnk.ru/xveQa2

27. Моделі життєвого циклу, принципи та методології розробки програмного забезпечення (ПЗ). URL: https://evergreens.com.ua/ru/articles/software-development-metodologies.html

28. David Knox Oracle Database 12C Security. McGraw-Hill Education, 2016. 768 pag.

29. MySQL. URL: https://www.mysql.com

30. Ron Ben Natan Implementing Database Security and Auditing. Digital Press, 2005. 432 pag.

31. C# documentation. URL: https://docs.microsoft.com/en-us/dotnet/csharp

32. Ріхтер Д. CLR via C #. Програмування на платформі Microsoft .NET Framework 4.5 мовою C#. 4-те вид. СПб: Пітер, 2018. 896 с.

33. Петцольд Ч. Програмування для Microsoft Windows. СПб, 2014. 1008, с.

34. MySQL: Developer Zone. URL: https://dev.mysql.com

35. Шикула О.М. Система керування базами даних MS Access: Навчальний посібник. Київ. ІПДО, 2017. 177 с.

36. Аносов А. Критерії вибору СКБД при створенні інформаційних систем. URL: http: //easycode.com.ua/2011/02/ kriteri%D1%97-viboru-subd-pri-stvorenni-informacijnix-sistem

37. Організація баз даних : навч. посібник / О. Г. Трофименко, Ю. В. Прокоп, Н. І. Логінова, І. М. Копитчук. 2-ге вид. виправ. і доповн. Одеса : Фенікс, 2019. 246 с.

38. Тарасов О. В., Федько В. В., Лосєв М. Ю. Використання мови SQL для роботи з сучасними системами керування базами даних. Х. : Вид. ХНЕУ, 2013. 348 с.

39. Зарицька О. Л. Бази даних та інформаційні системи : метод. посіб. Житомир : Вид-во ЖДУ ім. І. Франка, 2009. 132 с.

40. Берко А. Ю., Верес О. М., Пасічник В. В. Системи баз даних та знань. Книга 2. Системи управління базами даних та знань : навч. посіб. Львів : "Магнолія-2006", 2012. 584 с.