МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних технологій

Кваліфікаційна робота бакалавра

на тему: «Програмна система оцінки впливу забруднюючих речовин на навколишнє середовище»

Виконав студент групи К-41
Спеціальності 122 Комп'ютерні науки,
Румелець Денис Юрійович

Керівник доктор філософії, асистент
Бучинська Ірина Вікторівна

Рецензент д.ф.-м.н., професор
Ковальчук Володимир Володимирович

Одеса 2021

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних;

ІТ – інформаційні технології;

ПП – програмний продукт;

СУБД – системою управління базами даних;

JSON –JavaScript Object Notation, – запис об'єктів JavaScript;

OLTP – Online Transaction Processing;

SAP HANA – Systems, Applications, and Products High Performance Analytics Appliance;

SQL – Structured Query Language;

ВСТУП

Розвиток сучасних інформаційних технологій призвів до зростання потреб у нових високопродуктивних методах для ефективної обробки даних. В наш час багато розрахункових методик, різних направлень необхідно автоматизувати, що дозволить зменшити час, затрачений на виконання даних розрахунків та зменшити помилки, при їх виконанні.

У наш час майже все автоматизовано, більшість ІТ-організацій використовують кілька технологій автоматизації. Більшість організацій перебувають на дуже ранніх термінах реалізації таких рішень. Ідея, яка лежить в основі потреби штучного інтелекту та супутніх технологіях, полягає в тому, що за багато рішень все ще відповідають розробники у сферах, які можуть бути ефективно вирішені шляхом адекватного навчання комп'ютерних систем. Наприклад, розробник вирішує, що потрібно виконати, але визначення найкращої системи для виконання процесів може здійснюватися програмним забезпеченням за допомогою аналітики всередині системи.

Одним із прикладів, є автоматизоване тестування: сценарії тестування вже широко поширені, але незабаром автоматизовані процеси тестування можуть з більшою ймовірністю навчитися в дорозі та розвивати, наприклад, більш широке визнання того, як зміниться новий код або зміни вплив виробничих середовищ.

Цифрова трансформація, яка продовжує змінювати ділову та галузеву практику, призвела до експоненціального збільшення обсягу цифрових даних, якими потрібно керувати. Зокрема, ІТ зазнали значних змін і призвели до критичної потреби в автоматизації. ІТ-автоматизація стосується видалення ІТ-фахівцями ручного виконання різних завдань, пов’язаних з ІТ, і заміни цих процесів складними програмними засобами та комплектами, які можуть виконувати завдання автоматично. Цю автоматизацію зазвичай здійснювали

за допомогою програмного інтерфейсу, і сьогодні все частіше вона просякнута штучним інтелектом та алгоритмами машинного навчання.

Автоматизація ІТ-завдань має багато переваг, якщо це робиться належним чином. Повторювані робочі процеси, добре задокументовані та суворо дотримувані бізнес-процесів, менша кількість помилок та більше часу для кваліфікованих ІТ-спеціалістів зосередитись на більш складних завданнях, які безпосередньо сприяють підвищенню ефективності та продуктивності процесу.

Автоматизована система складається з елементів, призначених для виконання комплексу завдань, які були запрограмовані. Оперативні та повторювані завдання стають менш важкими і роблять ваше життя простішим та легшим.

Багато методик, які в своїй роботі використовують інженера-екологи, складні за своїми розрахунками, та потребують часу на їх виконання.

В даній роботі, показано як можливо автоматизувати один з методів розрахунку, та наскільки це допоможе прискорити роботу виконавцю.

Мета роботи розробити систему оцінки впливу забруднюючих речовин на навколишнє середовище для підвищенні продуктивності, зменшення витрат, зменшення формальностей та наданні командам часу, щоб зосередитись на таких завданнях, як додаток до вартості, таких як стосунки з клієнтами або подальші дії у складних ситуаціях.

Основні завдання для виконання автоматизації діяльності підприємства наступні:

– дослідити предметну область;

– спроектувати базу даних;

– створити форму для роботи з базою;

– організовано для користувача меню, яке повинно мати простий та зручний інтерфейс.

- збір, обробка, зберігання та надання даних про діяльність організації та зовнішньому середовищі у вигляді, зручному для аналізу і використання при прийнятті управлінських рішень;
- автоматизація виконання бізнес-операцій (технологічних операцій), що становлять цільову діяльність організації;
- автоматизація процесів, що забезпечують виконання основної діяльності.
- повна (комплексна) або часткова автоматизація діяльності підприємства.

Бакалаврська робота складається з вступу, 4 розділів, висновків, 9 посилань, 9 таблиць, 19 рисунків.

1 ОПИС ВИКОРИСТАННЯ ЗАСОБІВ АВТОМАТИЗАЦІЇ

Автоматизоване управління системами − це єдиний спосіб, яким сьогодні компанії, можуть успішно забезпечити роботу своїх комп’ютерних систем на їх оптимальному рівні, щоб задовольнити запити бізнесу. Найважливішим активом будь-якої компанії є її дані. Надійне управління системами, заблоковане в електронній формі, і залежить від безперебійної роботи IT-інфраструктури компанії та комп’ютерних систем, є необхідністю для забезпечення постійної доступності ключових даних.

Оскільки комп’ютери та додатки поширюються у більшості організацій, на системних менеджерів постійно тиснуть, щоб забезпечити високий рівень обслуговування та надійності. У міру зростання складності наших мереж, завдяки великій взаємозв'язку та глобальним операціям, проблема ускладнювалася. Дуже часто системні менеджери все ще працюють у темряві і не можуть скласти цілісний погляд на мережу та комп'ютерні системи, які вони контролюють. Щоб забезпечити максимальну цінність, IT-менеджерам та компаніям потрібне рішення, яке може бути швидко розгорнуте у всій організації, і таке, що може забезпечити більшу продуктивність для користувачів, забезпечуючи доступність даних, ключових бізнес-додатків та процесів у 100% часу та ризик позапланових простоїв значно знижується.

Ефективне управління системами полягає також у прийнятті значних рівнів автоматизації (включаючи регулярні домашні справи) та налаштуванні систем раннього попередження, щоб вас повідомляли про серйозні події. За умови постійного тиску на сучасний IT-персонал, деякі звичайні, повторювані, але важливі завдання дуже легко пропустити. Завдяки автоматизації можна легко досягти постійного моніторингу та управління дисковим простором, продуктивності, системних повідомлень, системних подій та черг завдань, і це простий та економічний спосіб звільнити час персоналу.

Мало того, але автоматизація багатьох повторюваних завдань системного адміністрування означає, що дорогі та дефіцитні IT-ресурси можна кра-

ще використовувати в інших сферах; такі як дослідження нових технологій, вивчення шляхів вдосконалення бізнес-процесів та робочого процесу та планування майбутніх ІТ-потреб організації.

Проникливі ІТ-директори використовують автоматизовані управлінські рішення як частину своїх повсякденних систем та проводять політику управління за винятком, запроваджуючи «розумний моніторинг». Здебільшого 80% усіх комп'ютерних систем будуть функціонувати належним чином; «Інтелектуальний моніторинг» визначає інші 20% та дає попереднє попередження системним адміністраторам, якщо стан зміниться або стани не очікувані. Багато керованих сервісних компаній успішно використовують інтелектуальний моніторинг, щоб почати більше бізнесу, не збільшуючи штат, і, крім того, зменшують загальні витрати на управління своїми системами.

У сучасному конкурентному середовищі, що знаходиться під тиском, автоматизоване управління системами вже не є розкішшю − це необхідність для мінімізації ризику та забезпечення постійної життєздатності будь-якого бізнесу, незалежно від його розміру.

Більшість організацій перебувають на дуже ранніх термінах реалізації таких рішень. Ідея, яка лежить в основі потреби штучного інтелекту та супутніх технологіях, полягає в тому, що за багато рішень все ще відповідають розробники у сферах, які можуть бути ефективно вирішені шляхом адекватного навчання комп'ютерних систем. Наприклад, розробник вирішує, що потрібно виконати, але визначення найкращої системи для виконання процесів може здійснюватися програмним забезпеченням за допомогою аналітики всередині системи.

Одним із прикладів, який відразу спадає на думку, є автоматизоване тестування: сценарії тестування вже широко поширені, але незабаром автоматизовані процеси тестування можуть з більшою ймовірністю навчитися в дорозі та розвивати, наприклад, більш широке визнання того, як зміниться новий код або зміни вплив виробничих середовищ.

У той же час, завжди рекомендується розглядати як ІТ, так і роботизовану автоматизацію процесів як стратегію, а не як спеціальне рішення. Розглядається як континуум, навіть початковий етап автоматизації простих, повторюваних і чітко визначених завдань врешті-решт призведе до переходу компанії на віртуальну роботу.

Реальність полягає в тому, що діловий світ працює швидше за допомогою технологій, а відмова від використання нових систем означає, що конкуренти на крок попереду. Автоматизація завдань полегшує час, тому співробітники можуть сконцентруватися на інших завданнях, які програмне забезпечення не здатне управляти. Можна заощадити години, делегуючи програмні завдання, такі як управління рахунками-фактурами та контрактами або управління запасами.

Звичайно, для деяких завдань потрібен людський дотик, але автоматизовані інструменти роблять вас ефективнішими та дають більше часу для зосередження на більш важливих завданнях та стратегічному напрямку.

1.1 Основні переваги систем автоматизації

Використання таких систем завжди слід починати з урахування потреб та вибору технології, яка найкраще відповідає цілі. Впровадження розумної автоматизації, як правило, починається з визнання та встановлення пріоритетів цілей − способу мислення, властивого концепції інтелектуальної автоматизації процесів.

Після автоматизації процесу можна розраховувати на багато переваг:

– підвищення продуктивності, зумовлене розширеним доступом; хмарні засоби автоматизації бізнес-процесів зберігають дані у центральній базі даних; це допомагає отримати доступ до даних із будь-якого місця чи пристрою, коли вам це потрібно.
– бізнес-процеси стануть набагато прозорішими (можна відстежувати та контролювати процеси під час їх роботи, що може покращити підзвітність та видимість).

- можливість відстежувати процеси не на робочому місці, також допоможе стежити за помилками, виправляючи їх у міру їх виникнення, звіти про ефективність озброять статистичними даними, щоб була можливість вживати профілактичних заходів проти повторюваних помилок.
- з довгострокової точки зору можна бути помічати більш швидкі терміни виконання та зменшення витрат за рахунок меншої кількості втручань вручну.
- також можна покращити розподіл робочої сили, оскільки програма буде обробляти всі повсякденні регулярні завдання (таким чином можна перенаправити співробітників на завдання, які вимагають людських зусиль та суджень).

Система автоматизації бізнес-процесів в кінцевому рахунку забезпечить зростання ефективності бізнесу. Оскільки воно засноване на понятті постійного вдосконалення процесу, рівень ефективності буде постійно зростати у відповідь.

1.2 Цілі, завдання та принципи автоматизації

В різних компанії, які займаються системною інтеграцією, доводиться багато разів читати і доповнювати документи типу ТКП, ТЗ по ГОСТу, плани приймання та інше. Проекти по автоматизації підприємств переслідували схожі цілі і мали приблизно однаковий формат. Одні і ті ж фрази повторювалися з документа в документ.

Аналітика, статистика, звіти. Все, що дозволяє дані збирати, обробляти і виводити в зручній для сприйняття формі.

Прискорення інформаційних потоків може бути досягнуто як збільшенням продуктивності за рахунок збільшення ресурсів, за рахунок змін архітектурно-технічних рішень, так і за рахунок скорочення числа логічних ланок ланцюжка передачі даних (автоматична обробка замість ручної на якомусь етапі).

В даний час нормативно-методична база з охорони атмосферного повітря продовжує розвиватися на основі науково-дослідної та методичної діяльності. Це стосується широкого кола питань: процедури інвентаризації викидів шкідливих речовин в атмосферне повітря з використанням як інструментальних, так і розрахункових методів, організації та проведення розрахунків забруднення атмосфери, формування пропозицій по нормативам, а також визначенню періодичності виробничого контролю за дотриманням встановлених нормативів викидів і обсягів регулювання викидів в різні періоди.

Актуальність подальшого вдосконалення діяльності обумовлена двома основними причинами:

– обов'язковістю введення в практику повітряохоронного діяльності положень Закону «Про охорону атмосферного повітря» [1], а з 2002 р і Закону «Про охорону навколишнього середовища» [2];

– необхідністю з одного боку більшого обґрунтування вимог, що пред'являються до природокористувачів, і з іншого боку необхідністю спрощення системи нормування викидів.

Ще один важливий аспект розвитку принципів нормування пов'язаний з організацією системи зведених розрахунків забруднення атмосфери в містах і використанням їх результатів при нормуванні викидів. У містах, в яких ці системи функціонують, помітно впорядковується не тільки система нормування викидів, але і підвищується ефективність роботи підрозділів державної експертизи, в тому числі, за рахунок більш оперативного прийняття рішень про можливість розміщення нових виробництв (в тому числі, і за рахунок іноземних інвестицій).

Автоматизація дозволяє підвищити продуктивність праці, поліпшити якість продукції, оптимізувати процеси управління, відсторонити людину від виробництв, небезпечних для здоров'я. Автоматизація, за винятком найпростіших випадків, вимагає комплексного, системного підходу до вирішення завдання. До складу систем автоматизації входять датчики (сенсори), пристрої введення, пристрої керування (контролери), виконавчі пристрої, пристрої ви-

ведення, комп'ютери. Застосовувані методи обчислень іноді копіюють нервові і розумові функції людини. Весь цей комплекс засобів зазвичай називають системами.

2 ХАРАКТЕРИСТИКА БАЗ ДАНИХ

База даних (БД) − це організована колекція структурованої інформації або даних, яка зберігається в електронному вигляді в комп’ютерній системі. База даних зазвичай контролюється системою управління базами даних (СУБД).

Дані в БД, зазвичай моделюються в рядки та стовпці в серії таблиць, щоб зробити обробку та запит даних ефективними. Далі можна легко отримати доступ, керувати ними, модифікувати, оновлювати, контролювати та організовувати. Більшість БД використовують структуровану мову запитів (SQL) для запису та запиту даних.

SQL – це мова програмування, яка використовується майже у всіх реляційних БД для запитів, маніпулювання та визначення даних та забезпечення контролю доступу. Вперше SQL був розроблений в IBM в 1970-х роках, коли основним внеском став Oracle, що призвело до впровадження стандарту SQL ANSI. Хоча SQL все ще широко використовується сьогодні, починають з'являтися нові мови програмування.

2.1 Еволюція бази даних

З часу створення БД різко еволюціонували. Навігаційні БД, такі як ієрархічна БД (яка спиралася на деревоподібну модель і допускала лише взаємозв'язок "один-до-багатьох"), та мережева база даних (більш гнучка модель, яка допускала кілька взаємозв'язків), були оригінальними системами, що використовувались для зберігання і маніпулювати даними. Ці ранні системи, хоча і були простими, були негнучкими. У 1980-х роках популярними стали реляційні БД, а за ними – об'єктно-орієнтовані бази даних. Зовсім недавно бази даних NoSQL виникли як відповідь на зростання Інтернету та потребу в швидшій швидкості та обробці неструктурованих даних. Сьогодні хмарні бази даних та самокеровані бази даних відкривають нові можливості, коли

йдеться про те, як дані збираються, зберігаються, управляються та використовуються.

2.2 Типи баз даних

Існує багато різних типів БД. Найкраща база даних для конкретної організації залежить від того, як організація має намір використовувати дані:

- реляційні БД **(**елементи даної БД організовані як набір таблиць із стовпцями та рядками, ця технологія забезпечує найбільш ефективний та гнучкий спосіб доступу до структурованої інформації).
- об’єктно-орієнтовані бази даних (інформація представлена вигляді об’єктів, як і в об’єктно-орієнтованому програмуванні).
- розподілені бази даних (складається з двох або більше файлів, розташованих на різних сайтах, БД може зберігатися на декількох комп'ютерах, розташованих в одному фізичному місці або розкиданих по різних мережах).
- сховища даних (цей тип БД, спеціально розроблений для швидких запитів та аналізу).
- бази даних NoSQL або нереляційні БД (дозволяє неструктуровані та слабоструктуровані дані зберігати і маніпулювати ними, на відміну від реляційної БД, яка визначає, як всі дані, вставлені в БД повинні бути складені; ці бази стали популярнішими, оскільки веб-програми стали більш поширеними та складнішими).
- графічні бази даних (зберігає дані в межах сутностей та взаємозв’язків між сутностями).
- бази даних OLTP **(**це швидка, аналітична БД, призначена для великої кількості транзакцій, що виконуються декількома користувачами).

Це лише декілька типів БД, що використовуються сьогодні. Інші, менш поширені БД пристосовані до цілком конкретних наукових, фінансових чи інших функцій. На додаток до різних типів БД, зміни у підходах до розробки

технологій та значні досягнення, такі як хмара та автоматизація, рухають бази даних у абсолютно нових напрямках. Деякі з останніх баз даних включають:

- БД з відкритим кодом (це система, вихідний код якої є відкритим).
- хмарні БД **(**являє собою набір даних, або структурованих або неструктурованих, який знаходиться в приватній, громадській, або гібридній платформі хмарних обчислень).
- БД мультимоделей **(**поєднують різні типи моделей баз даних в єдиний інтегрований задній кінець, тобто, вони можуть вміщувати різні типи даних).
- БД документа/JSON (призначені для зберігання, отримання та управління інформацією, орієнтованою на документи, є сучасним способом зберігання даних у форматі JSON, а не рядків і стовпців).
- БД, що самостійно керують (найновіший та найбільш новаторський тип БД, самокеровані БД (також відомі як автономні БД) засновані на хмарі та використовують машинне навчання для автоматизації налаштування, безпеки, резервного копіювання, оновлення та інших рутинних завдань управління, які традиційно виконуються адміністраторами БД).

2.3 Опис програмного забезпечення для БД

Програмне забезпечення БД використовується для створення, редагування та обслуговування файлів і записів БД, що полегшує створення файлів та записів, введення даних, редагування даних, оновлення та звітування. Програмне забезпечення також забезпечує зберігання даних, резервне копіювання та звітування, контроль доступу та безпеку. Сильна безпека БД сьогодні особливо важлива, оскільки крадіжка даних стає все частішою. Програмне забезпечення баз даних іноді також називають СУБД.

Програмне забезпечення для БД спрощує управління даними, дозволяючи користувачам зберігати дані у структурованому вигляді, а потім отриму-

вати до них доступ. Зазвичай він має графічний інтерфейс, який допомагає створювати дані та керувати ними, а в деяких випадках користувачі можуть створювати власні БД за допомогою програмного забезпечення для БД.

Для БД зазвичай потрібна комплексна програма програмного забезпечення для баз даних, відома як СУБД. СУБД служить інтерфейсом між БД та її кінцевими користувачами або програмами, дозволяючи користувачам отримувати, оновлювати та керувати організацією та оптимізацією інформації. СУБД також полегшує нагляд та контроль баз даних, дозволяючи різноманітні адміністративні операції, такі як моніторинг продуктивності, налаштування, резервне копіювання та відновлення.

Деякі приклади популярного програмного забезпечення баз даних або СУБД включають MySQL, Microsoft Access, Microsoft SQL Server, FileMaker Pro, Oracle Database та dBASE.

2.4 Використання баз даних для підвищення ефективності бізнесу та прийняття рішень

Завдяки масовому збору даних з Інтернету речей, що трансформує життя та промисловість по всьому світу, сьогодні компанії мають доступ до великої кількості даних, ніж будь-коли раніше. Прогресивні організації тепер можуть використовувати бази даних, щоб вийти за рамки базового зберігання даних і транзакцій для аналізу величезних обсягів даних з різних систем. Використовуючи БД та інші обчислювальні засоби та інструменти бізнес-аналітики, організації тепер можуть використовувати зібрані дані для ефективнішої роботи, забезпечення кращого прийняття рішень та підвищення гнучкості та масштабності.

БД, що керує автомобілем, готова забезпечити значний приріст цих можливостей. Оскільки БД, що керують собою, автоматизують дорогі, трудомісткі ручні процеси, вони звільняють бізнес-користувачів, щоб вони стали більш активними щодо своїх даних. Маючи безпосередній контроль над мо-

жливістю створювати та використовувати БД, користувачі отримують контроль та автономність, зберігаючи при цьому важливі стандарти безпеки.

2.5 Проблеми з базою даних

Сучасні великі корпоративні БД часто підтримують дуже складні запити, і, як очікується, дають майже миттєві відповіді на ці запити. Як результат, адміністраторів БД постійно закликають застосовувати широкий спектр методів, що сприяють підвищенню продуктивності. Деякі загальні проблеми, з якими вони стикаються, включають:

- поглинання значного збільшення обсягу даних;**.**
- забезпечення безпеки даних;
- оновлення та модернізація;
- управління та підтримка бази даних та інфраструктури;
- зняття обмежень на масштабованість.

Вирішення цих проблем може зайняти багато часу і може перешкодити адміністраторам БД виконувати більш стратегічні функції.

БД, які керують самостійно, – це хвиля майбутнього – і вони пропонують інтригуючу можливість для організацій, та вимагають використовувати найкращі доступні технології БД без проблем при роботі та експлуатації технологій [3]

2.6 Опис найбільш часто використовуваних БД

Нижче представлені найкращі БД, які найбільше використовуються розробниками у всьому світі в 2020 році.

2.6.1 Характеристика MySQL

MySQL – одна з найпопулярніших систем управління базами даних з відкритим кодом (рис. 1). Він був розроблений та оптимізований для вебдодатків і може працювати на будь-якій платформі. Розроблений Oracle, MySQL Database Software – це система клієнт/сервер, яка складається з бага-

топотокового SQL-сервера, який підтримує різні фонові кінці, декількох різних клієнтських програм та бібліотек, адміністративних інструментів та широкого спектру інтерфейсів прикладного програмування (API).

Рисунок 1 − Логотип MySQL

Оскільки він призначений для обробки мільйонів запитів і тисяч транзакцій, MySQL є популярним вибором для підприємств електронної комерції, яким потрібно керувати кількома грошовими переказами. Гнучкість на вимогу – це основна особливість MySQL. MySQL – це СУБД, що стоїть за деякими провідними веб-сайтами та веб-додатками у світі, включаючи Airbnb, Uber, LinkedIn, Facebook, Twitter та YouTube.

2.6.2 Характеристика PostgreSQL

PostgreSQL – це потужна об’єктно-реляційна система БД з відкритим кодом, яка включає деякі ключові функції, такі як надійність, надійність та продуктивність. Він використовує та розширює мову SQL у поєднанні з багатьма функціями, які безпечно зберігають та масштабують найскладніші робочі навантаження даних. PostgreSQL має безліч функцій, спрямованих на допомогу розробникам у створенні додатків. Це дозволяє адміністраторам захищати цілісність даних та створювати відмовостійкі середовища та допомагати керувати даними (рис. 2).

Рисунок 2 − Логотип PostgreSQL

2.6.3 Характеристика Microsoft SQL Server

Microsoft SQL Server – це реляційна СУБД, розроблена корпорацією Microsoft. SQL Server 2019 включає ряд інтуїтивних функцій, таких як отримання уявлення про всі дані шляхом запитів до реляційних, нереляційних, структурованих та неструктурованих даних, гнучкість використання мови та платформи за вибором користувача з підтримкою з відкритим кодом, масштабованість та продуктивність для покращення стабільності та часу відгуку БД тощо.

2.6.4 Характеристика SQLite

SQLite – це технологічна бібліотека, яка реалізує автономний безсерверний механізм бази даних SQL із нульовою конфігурацією. Це вбудований механізм баз даних SQL, і на відміну від більшості інших БД SQL, SQLite не має окремого серверного процесу.

2.6.5 Характеристика MongoDB

MongoDB – це загальнодоступна розподілена база даних на основі документів, створена для сучасних розробників додатків та для епохи хмар. Це одна з популярних БД, яка включає як масштабованість, так і гнучкість. MongoDB – це БД документів, що означає, що вона зберігає дані у JSON-подібних документах.

2.6.6 Характеристика Redis

Redis – це сховище структур даних з відкритим кодом, яке використовується як база даних, кеш-пам’яті та посередник повідомлень. Він підтримує такі структури даних, як рядки, хеші, списки, набори, відсортовані набори із запитами діапазонів, растрові зображення, гіперлогічні журнали, геопросторові індекси із запитами радіуса та потоками. Redis написаний на ANSI C і працює в більшості систем POSIX, таких як Linux, *BSD, OS X без зовнішніх залежностей.

2.6.7 Характеристика MariaDB

Сервер MariaDB – це один з найпопулярніших серверів БД, який перетворює дані на структуровану інформацію в широкому діапазоні програм, починаючи від банківських і закінчуючи веб-сайтами. Він розроблений як програмне забезпечення з відкритим кодом та як реляційна БД. Він також надає інтерфейс SQL для доступу до даних.

2.6.8 Характеристика Oracle

Oracle Database – це багатомодельна система управління БД для більш безпечного запуску всіх робочих навантажень, як локальних, так і автономних в Oracle Cloud Infrastructure (рис. 3). Існує кілька інтуїтивних функцій, таких як СУБД, яка дозволяє користувачеві вибирати з багатьох варіантів розгортання, таких як локальний, Cloud@ Customer та загальнодоступна хмара.

Це допомагає створювати високомасштабовані програми, підтримуючи всі типи даних, включаючи реляційні, графічні та структуровані та неструктуровані нереляційні дані.

Рисунок 3 – Логотип Oracle Database

2.6.9 Харктеристика Firebase

Розроблений Google, Firebase – це платформа для розробки додатків для створення мобільних та веб-додатків. Він надає розробникам адекватні інструменти для розробки високоякісних додатків, а також для збільшення бази користувачів. Firebase надає різні функції, такі як аналітика, БД, обмін повідомленнями та звіти про збої.

2.6.10 Характеристика Elasticsearch

Elasticsearch – це розподілений механізм пошуку та аналітики з відкритим кодом для всіх типів даних, включаючи текстові, числові, геопросторові, структуровані та неструктуровані дані. Це центральний компонент Elastic Stack, який являє собою набір інструментів з відкритим кодом для прийому, збагачення, зберігання, аналізу та візуалізації даних. Швидкість і масштабованість Elasticsearch можна використовувати для пошуку додатків, пошуку веб-сайтів, реєстрації та аналізу журналів, моніторингу продуктивності додатків, аналітики безпеки тощо. [4]

2.6.11 Характеристика SAP HANA

SAP HANA це скорочення від High Performance Analytics Appliance. Замість традиційної обробки даних, яка вимагає збереження інформації у базі даних, а потім її отримання перед тим, як помістити її в пам’ять, SAP HANA повністю пропускає цей крок [5].

За допомогою SAP HANA всі дані вже зберігаються в пам'яті. Немає тривалого часу обробки великих обсягів даних, оскільки інформація вже є в пам'яті системи. Його можна негайно обробити, замість того, щоб отримувати. Поліпшення загальної продуктивності дозволяє вам закінчувати проекти за менший час, ніж будь-коли раніше.

2.7 Порівняння популярних баз даних

Хоча всі СУБД виконують одне й те саме основне завдання, яке полягає у наданні користувачам можливості створювати, редагувати та отримувати доступ до інформації в БД, але спосіб їх досягнення відрізняється. Крім того, функції, функціональність та підтримка кожної системи управління можуть суттєво відрізнятися.

Порівнюючи різні популярні бази даних, слід врахувати, наскільки зручною та масштабованою є кожна СУБД, а також наскільки вона буде інтегруватися з іншими продуктами, якими необхідно користуватися. Крім того, можна взяти до уваги вартість системи управління та підтримку, доступну до неї [6].

У табл.1 представлені найпопулярніші БД, завдяки тому, що існує ряд чудових безкоштовних опцій, в переліку можна знайти інструмент управління БД, який відповідає низці критеріїв.

Таблиця 1 – Аналіз популярних баз даних

Програми	Переваги	Недоліки
Oracle	–найновіші інновації та функції; –інструменти керування БД неймовірно надійні; –ідеально підходить для великих організацій, які працюють з величезними БД і потребують різноманітних функцій.	–вартість; –встановлена система може зажадати значних ресурсів, тому для впровадження може знадобитися нове обладнання.
MySQL	–доступний безкоштовно; –пропонує багато функціональних можливостей навіть для безкоштовного механізму баз даних; –існує безліч користувальницьких інтерфейсів, які можна реалізувати; –можна змусити працювати з іншими базами даних, включаючи DB2 та Oracle; –ідеально підходить для організацій, яким потрібен надійний інструмент управління БД, але вони не мають бюджет	–можна витратити багато часу та зусиль, щоб змусити MySQL робити те, що інші системи роблять автоматично, (наприклад, створювати додаткові резервні копії); –немає вбудованої підтримки XML або OLAP; –підтримка доступна для безкоштовної версії, але доведеться заплатити за неї.
Microsoft SQL Server	–дуже швидкий і стабільний; –Двигун пропонує можливість регулювання та відстеження рівня –продуктивності, що може зменшити використання ресурсів; –Можна отримати доступ до візуалізації на мобільних пристроях; –дуже добре працює з іншими продуктами Microsoft;	–зависока вартість; –навіть з налаштуванням продуктивності, Microsoft SQL Server може –поглинати ресурси; –виникають проблеми із використанням служб інтеграції SQL Server для імпорту файлів.
Postgresql	–механізм управління БД є масштабованим і може обробляти терабайти даних; –підтримує JSON; –існує безліч заздалегідь визначених функцій; –доступно ряд інтерфейсів; –ідеально підходить для організацій з обмеженим бюджетом, які хочуть вибрати свій інтерфейс та використовувати JSON.	–документація може бути непомітною, тому може опинитися в Інтернеті, намагаючись з’ясувати, як щось зробити; –конфігурація може заплутати. –швидкість може постраждати під час великих масових операцій або читання запитів.
MongoDB	–швидкий і простий у використанні; –механізм підтримує JSON та інші документи NoSQL; –дані будь-якої структури можна швидко і легко зберігати та отримувати до них доступ; –схему можна писати без простоїв.	–SQL не використовується як мова запитів; –доступні інструменти для перекладу SQL на запити MongoDB, але вони додають додатковий крок до використання механізму; –налаштування може бути тривалим процесом; –налаштування за замовчуванням не захищені.

Продовження Таблиці 1

MariaDB	–система швидка та стабільна; –індикатори виконання повідомляють вам про те, як проходить запит; –розширювана архітектура та плагіни дозволяють налаштувати інструмент відповідно до ваших потреб; –шифрування доступне на рівні мережі, сервера та додатків; –ідеально підходить для організацій, які шукають доступну альтернативу MySQL.	–двигун все ще досить новий, тому немає жодних гарантій подальших оновлень та версій; –як і у багатьох інших безкоштовних двигунах БД, вам доведеться платити за підтримку.
DB2	–Blu Acceleration може максимально використати доступні ресурси для величезних БД; –може розміщуватися з хмари, фізичного сервера або обох одночасно; –за допомогою планувальника завдань можна запустити відразу кілька завдань; –коди помилок та коди виходу можуть визначити, які завдання запускаються за допомогою планувальника завдань; –ідеально підходить для великих організацій, яким потрібно максимально використовувати наявні ресурси та обробляти великі бази даних.	–вартість виходить за рамки бюджету багатьох людей та менших організацій; –для роботи кластерів або декількох вторинних вузлів потрібні сторонні інструменти або додаткове програмне забезпечення; –базова підтримка доступна лише протягом трьох років; після цього за це потрібно заплатити.
SAP HANA	–підтримує SQL, OLTP та OLAP. –двигун зменшує вимоги до ресурсів за рахунок стиснення; –дані зберігаються в пам'яті, зменшуючи час доступу, в деяких випадках, значно; –доступні звіти в реальному часі та управління запасами. –взаємодія з іншими програмами; –ідеально підходить для організацій, які отримують дані з додатків і не мають обмеженого бюджету.	–вартість ліцензування висока для SAP HANA навіть для тих, хто звик платити за корпоративне ПЗ; –SAP HANA все ще відносно новачок, і виправлення та оновлення часто трапляються аж до дратування.
Redis	–простий у використанні; –підтримує хешування; –підтримує більшість провідних мов програмування та протоколів, включаючи Python, Java, PHP, Perl, Go, Ruby, C / C # / C ++, JavaScript, Node.js та багато інших.	–немає мови запитів (лише команди) і не підтримується реляційна алгебра; –пропонує лише базову безпеку; –працює лише на одному ядрі процесора в однопотоковому режимі; –є обмеження довжини значення; –вимагає величезної оперативної пам'яті.
Elasticsearch	–сумісний для роботи на будь-якій платформі, оскільки він розроблений на Java; –полегшує масштабування у великій організації, наже легко інтегрувати його до будь-яку велику організацію, масштабуючи його.	–не має багатомовної підтримки для обробки даних запитів та відповідей; –не є хорошим сховищем даних; –гнучка та потужна пошукова система для зберігання даних, але її трохи складно навчитись, особливо з точки зору використання корпоративного пошуку, це не так просто, як нестандартний пошук.

Продовження Таблиці 1

Firebase	–налаштування Firebase відбувається швидко; –синхронізувати та оновлювати дані в режимі реального часу; –інтегрована автентифікація (з простими, безпечними входами в систему) та інтегрований хостинг, HTTPS включено тощо; –моніторинг продуктивності для вимірювання продуктивності.	–обмежена підтримка функцій iOS; –вартість; –проблеми синхронізації в режимі реального часу; –обмежені можливості запитів; –проблеми міграції даних; –не працює в країнах, які не дозволяють Google.

Зміна типів даних, що зберігаються, вимоги до швидкості і продуктивності привели і до триваючого розширення типів баз даних. При цьому кожен з них продовжує бути потрібним у своїй ніші, де взаємозв'язку між даними асоціюються з певною схемою будови бази даних.

3 ОПИС МЕТОДИКИ РОЗРАХУНКУ ТА НЕОБХІДНІСТЬ ЇХ АВТОМАТИЗАЦІЇ

Розвиток сучасних інформаційних технологій призвів до зростання потреб у нових високопродуктивних методах для ефективної обробки даних. В наш час багато розрахункових методик, різних направлень необхідно автоматизувати, що дозволить зменшити час, затрачений на виконання даних розрахунків та зменшити помилки, при їх виконанні.

Багата екологічна база є в ГК «Інтеграл», яка є російським лідером в області розробки програмних продуктів по охороні навколишнього середовища.

В Україні програмних продуктів екологічної сфери нажаль не багато, та багато з них дублюються різними виробниками (табл. 2).

Таблиця 2 – Перелік ПП в галузі охорони атмосферного повітря

Найменування програмного продукту	Версія	Призначення	Чинність
“EOL”	версія 3.5	Програма розрахунку забруднення атмосфери	не обмеженно
EOL + FON	версія 4.3	Програма розрахунку забруднення на ЕОМ + розрахунок фонових концентрацій	не обмеженно
“PLENER”	версія 1.25	Програма розрахунку забруднення атмосфери на ЕОМ	не обмеженно
“EOL +”	версія 5	Програма розрахунку забруднення атмосфери на ЕОМ	не обмеженно
“EOL –2000”	версія 3.1	Програма розрахунку забруднення атмосфери на ЕОМ	не обмеженно
“EOL (ГАЗ)-2000”	версія 3.1	Програма розрахунку забруднення атмосфери на ЕОМ	не обмеженно
“Еколог – Газ”		Програма розрахунку забруднення атмосфери на ЕОМ	до 01.04.2008
“ТАНДЕМ”	версія 1	Експертна система ПГО	не обмеженно
“ЕКСПЕРТ”	версія 1	Ведення банку даних ПГО	не обмеженно
“ІНВЕНТАРИЗАЦІЯ”		Система для обробки даних інвентаризації джерел викидів	не обмеженно
“NEORIST”	версія 1	Розрахунки валових викидів забруднюючих речовин від неорганізованих джерел забруднення атмосфери	не обмеженно
“ІНВЕНТЕР”	версія 1.0	Система для обробки даних інвентаризації джерел викидів на ПК	не обмеженно
"Атмосфера"		Розрахунки викидів забруднюючих речовин в атмосферне повітря, формування таблиць звіту інвентаризації	до 01.01.06

Продовження Таблиці 2

Найменування програмного продукту	Версія	Призначення	Чинність
"Report 1.00"	версія 1	Підготовка форми держстатзвітності №2-ТП(повітря)-квартальна "Звіт про охорону атмосферного повітря"	не обмеженно
ЕОЛ (ГАЗ) - 2000 [h]	версія 4.0	Розрахунки забруднення атмосфері на ЕОМ в приземних і верхніх шарах атмосфері	не обмеженно
ЕОЛ -2000 [h]	версія 4.0	Розрахунки забруднення атмосфери на ЕОМ у приземних та верхніх шарах атмосфери	не обмеженно
Електронні типові форми XML	версія 3.1.1.3.	Експорт (імпорт) електронних копій відповідних документів, з метою їх подальшої обробки та публікаці	не обмеженно
Інтернет додаток "ЕКОЗВІТ"	версія 2	Підготовка в електронній формі документів для отримання дозволу на викиди та статистичної звітності	не обмеженно
"NORMA6XML"	версія 1	Підготовка в електроній формі документів, у яких обгрунтовуються обсяги викидів	не обмеженно

Багато методик, які в своїй роботі використовують інженера-екологи, складні за своїми розрахунками, та потребують часу на їх виконання.

В даній роботі, показано як можливо автоматизувати один з методів розрахунку, та наскільки це допоможе прискорити роботу виконавцю.

3.1 Розрахунок викидів забруднюючих речовин під час процесу перевантаження матеріалу

Розрахунок викидів забруднюючих речовин від процесу перевантаження, зсипання, переміщення матеріалу виконувався згідно «Сборника методик по расчету содержания загрязняющих веществ в выбросах от неорганизованных источников загрязнения атмосферы» м. Донецьк [7].

Загальний об’єм викидів розраховується за формулою:

$$q = A + B = \frac{k_1 \cdot k_2 \cdot k_3 \cdot k_4 \cdot k_5 \cdot k_7 \cdot G \cdot 10^6 \cdot B}{3600} + k_3 \cdot k_4 \cdot k_5 \cdot k_6 \cdot k_7 \cdot q \cdot F \qquad г/с$$

де:

А – викиди при переробці (зсипання, перевантаження, переміщення) матеріалу, г/с;

В – викиди при статистичному зберіганні матеріалу;

k_1 – вагова доля пилової фракції в матеріалі. Визначається методом відмивки та просіювання середньої проби з виділенням фракції пилу розміром 0-200 мкм, (табл. 3);

Таблиця 3 – Значення коефіцієнтів k_1, k_2 для визначення викидів пилу

№	Найменування матеріалу	Щільність матеріалу, г/см^3	Вагова доля пилової фракції k_1 в матеріалі	Доля пилу, який переходить в аерозоль k_2
1	Огарки	3,9	0,04	0,03
2	Клінкер	3,2	0,01	0,003
3	Цемент	3,1	0,04	0,03
4	Вапняк	2,7	0,04	0,02
5	Мергель	2,7	0,05	0,02
6	Вапно комове	2,7	0,07	0,02
7	Вапно мелене	2,7	0,07	0,05
8	Граніт	2,8	0,02	0,04
9	Мрамор	2,8	0,04	0,06
10	Крейда	2,7	0,05	0,07
11	Гіпс комовий	2,6	0,03	0,02
12	Гіпс мелений	2,6	0,08	0,04
13	Доломит	2,7	0,05	0,02
14	Опока	2,65	0,03	0,01
15	Пегматит	2,6	0,04	0,04
16	Гнейс	2,9	0,05	0,02
17	Каолін	2,7	0,06	0,04
18	Нефелін	2,7	0,06	0,02
19	Глина	2,7	0,05	0,02
20	Пісок	2,6	0,05	0,03
21	Пісщанник	2,65	0,04	0,01
22	Слюда	2,8	0,02	0,01
23	Польовий шпат	2,5	0,07	0,01
24	Шлак	2,5-3,0	0,05	0,02
25	Діорит	2,8	0,03	0,06
26	Портфіроїди	2,7	0,03	0,07
27	Графіт	2,2-2,7	0,03	0,04
28	Вуголь	1,3	0,03	0,02

Продовження Таблиці 3

№	Найменування матеріалу	Щільність матеріалу, г/см^3	Вагова доля пилової фракції k_1 в матеріалі	Доля пилу, який переходить в аерозоль k_2
29	Зола	2,5	0,06	0,04
30	Діатоміт	2,3	0,03	0,02
31	Перліт	2,4	0,04	0,06
32	Керамзіт	2,5	0,06	0,02
33	Вермикуліт	2,6	0,06	0,04
34	Аглопорит	2,5	0,06	0,04
35	Туф	2,6	0,03	0,02
36	Пемза	2,5	0,03	0,06
37	Сульфат	2,7	0,05	0,02
38	Шамот	2,6	0,04	0,02
39	Суміш піску та вапна	2,6	0,05	0,01
40	Бій цегли	-	0,05	0,01
41	Мінеральна вата	-	0,05	0,01
42	Щебінь	-	0,04	0,02

k_2 – доля пилу (від усієї маси пилу), яка переходить в аерозоль;

k_3 – коефіцієнт, який враховує місцеві метеорологічні умови, та приймається у відповідності до табл. 4 даної методики;

Таблиця 4 – Значення величини k_3 від швидкості вітру

Швидкість вітру, м/с	k_3
до 2	1
до 5	1,2
до 7	1,4
до 10	1,7
до 12	2
до 14	2,3
до 16	2,6
до 18	2,8
до 20 та більше	3

k_4 – коефіцієнт, який враховує місцеві умови, ступінь захищеності вузла від зовнішніх впливів, умови пилоутворення (табл.5);

k_5 – коефіцієнт, який враховує вологість матеріалу, та приймається у відповідності до даних табл. 6;

Таблиця 5 – Значення величини k4 від місцевих умов

Місцеві умови	k_4
Склади, хранилища відкриті:	
а) з 4-х сторін	1
б) з 3-х сторін	0,5
в) з 2-х сторін повністю та з 2-х частково	0,3
г) з 2-х сторін	0,2
д) з 1-ї сторони	0,1
з) завантажувальний рукав	0,01
ж) закритий з 4-х сторін	0,005

Таблиця 6 – Значення величини k5 від вологості матеріалу

Вологість матерілау, %	k_5
0-0,5	1
до 1	0,9
до 3	0,8
до 5	0,7
до 7	0,6
до 8	0,4
до 9	0,2
до 10	0,1
більше 10	0,01

k_6 – коефіцієнт, який враховує профіль поверхні матеріалу, який складується та визначається як співвідношення $F_{факт}/F$. Значення k_6 знаходиться в границях 1,3-1,6 в залежності від крупності матеріалу та ступені заповнення;

k_7 – коефіцієнт, який враховує крупність матеріалу, та приймається у відповідності до табл. 7;

$F_{факт}$ – фактична поверхня матеріалу з врахуванням рельєфу його перетину (враховувати тільки площу, на якій виконуються завантажувальні-розвантажувальні роботи);

F – поверхня пиління в плані, $м^2$;

q – винесення пилу з одного квадратного метра фактичної поверхні в умовах, коли $k_3 = 1$ та $k_5 = 1$, приймаються у відповідності з даними табл. 8;

G – сумарна кількість матеріалу, який перевантажується, т/год;

Таблиця 7 – Значення величини k7 від крупності матеріалу

Розмір курса, мм	k7
500	0,1
500-100	0,2
100-50	0,4
50-10	0,5
10-5	0,6
5-3	0,7
3-1	0,8
1	1

Таблиця 8 – Значення величини q при умові k3=1 та k5=1

Матеріал, який складується	q г/м^2
Клінкер, шлак	0,002
Щебінь, пісок, кварц	0,002
Мергіль, вапняк, огарки, цемент	0,003
Сухі глинясті матеріали	0,004
Хвости азбестових фабрик, пісчанник, вапняк	0,005
Вугілля, гіпс, крейда	0,005

B – коефіцієнт, який враховує висоту пересипки та приймається у відповідності до табл. 9.

Склади та хранилища приймаються як рівномірно розподілені джерела пиловидалення. Розрахункові коефіцієнти приймається згідно зведених таблиць

Таблиця 9 – Висота падіння матеріалу

Висота падіння матеріалу	B
0,5	0,4
1	0,5
1,5	0,6
2	0,7
4	1
6	1,5
8	2
10	2,5

3.2 Опис бази даних

При розробці системи автоматизації керуються перерахованими нижче загальними принципами (рис. 4):

1. Проектування системи автоматизації починають з вивчення об'єкта автоматизації, яке полягає у визначенні статичних і динамічних характеристик об'єкта автоматизації, вимог до якості регулювання, а також номенклатури параметрів контролю і управління, їх номінальних значень, точності вимірювання параметрів. Науково-технічні рішення, реалізовані в системі автоматизації, повинні відповідати сучасному рівню розвитку автоматизації технологічних процесів галузі.

2. При виборі технічних засобів автоматизації, тобто первинних вимірювальних перетворювачів, вимірювальних приладів, регуляторів і т.д., враховують біохімічні та фізико-хімічні особливості технологічного процесу, номінальні значення і допустимі відхилення технологічних параметрів, діапазон вимірювання параметрів, відстані від місць установки первинних вимірювальних перетворювачів і розміщення виконавчих механізмів до щитів (пультів) управління і контролю, закон регулювання і показники якості регулювання, відомості про умови експлуатації (характер робочої і навколишнього середовища, зокрема пожежно- та вибухонебезпечність, агресивність і токсичність). При виборі комплекту технічних засобів автоматизації перевагу віддають серійно випускається засобів автоматизації [8].

3. Допоміжну енергію (електричну, пневматичну і гідравлічну) вибирають з умов пожежо- та вибухонебезпечності об'єкта автоматизації, агресивності навколишнього середовища, а також з урахуванням довжини ліній зв'язку від місць установки первинних вимірювальних перетворювачів і виконавчих механізмів до щитів і пультів управління.

Рисунок 4 − Алгоритм проектування системи автоматизації процесів

4. Кількість приладів, апаратури управління, контролю і сигналізації, а також їх розташування на щитах і пультах вибираються з урахуванням положень інженерно-психологічного проектування схем автоматизації.

5. При проектуванні систем автоматичного управління передбачають можливість нарощування (в перспективі) комплекту технічних засобів.

Створення бази даних проводиться на етапі інсталяції і виконується автоматичними скриптами розгортання системи. Заповнення системними даними так само покладається на скрипти автоматичного розгортання системи [8].

Наповнення бази даних для користувача даними повинно проводитися через розроблений для користувача інтерфейс (рис 5).

Обслуговування бази даних має зводитися до своєчасного виконання резервного копіювання для мінімізації наслідків можливих збоїв.

Реалізація кошти імпортування масивів інформації в базу даних системи і виконання процедур синхронізації із зовнішніми джерелами не передбачено[10].

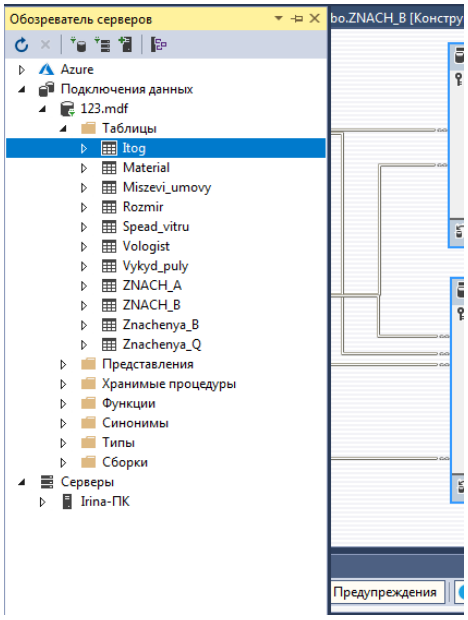

Рисунок 5 – Таблиці БД

На рис. 6 представлена структура БД

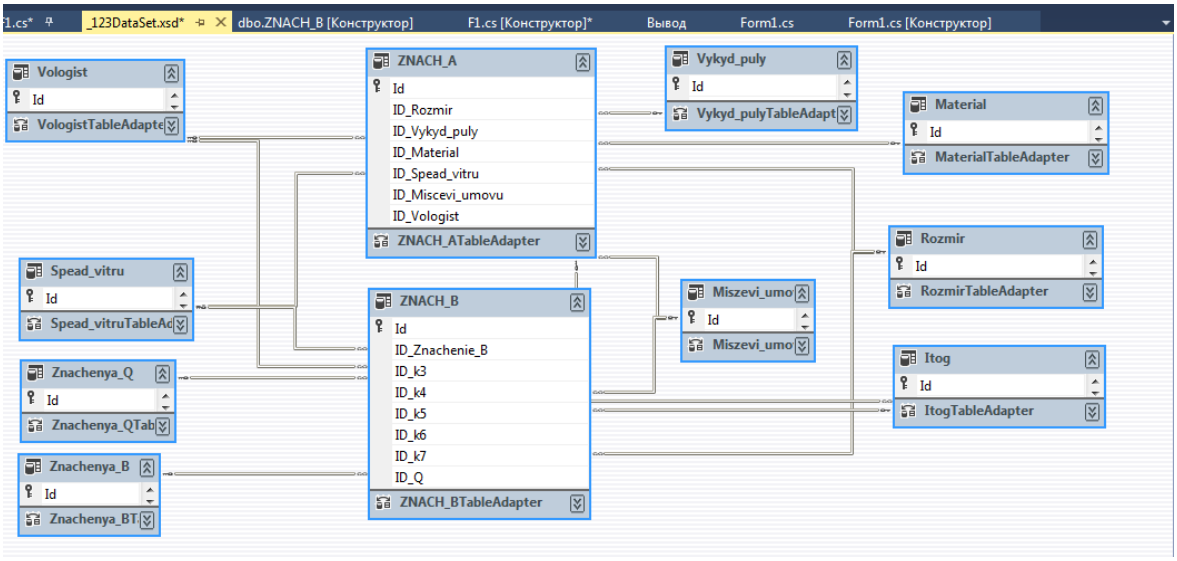

Рисунок 6– Схема бази даних зі зв’язками

В БД присутні наступні таблиці:

– Rozmir − значення величини k_7 від крупності матеріалу;

– Sped_vitru − значення величини k_3 від швидкості вітру ;

- Miszivi umovu − значення величини k_4 від місцевих умов;
- Vyukud_puly − значення коефіцієнтів k_1, k_2 для визначення викидів пилу;
- Vologist – значення величини k_5 від вологості матеріалу;
- Znachenya_B − значення величини B від висоти пересипання;
- Znachenya_Q − значення величини q при умові k3=1 та k5=1;
- ZNACH_A − викиди при переробці (зсипання, перевантаження, переміщення) матеріалу;
- ZNACH_B − викиди при статистичному зберіганні матеріалу;

Програмне забезпечення баз даних використовується для створення, редагування та обслуговування файлів і записів баз даних, що полегшує створення файлів і записів, введення даних, редагування даних, оновлення та звітування. Програмне забезпечення також забезпечує зберігання даних, резервне копіювання та звітування, контроль доступу та безпеку. Сьогодні суттєво важлива безпека баз даних, оскільки крадіжка даних стає все частішою[11].

Програмне забезпечення баз даних спрощує управління даними, дозволяючи користувачам зберігати дані у структурованому вигляді, а потім отримувати до них доступ. Зазвичай він має графічний інтерфейс, який допомагає створювати дані та керувати ними, а в деяких випадках користувачі можуть створювати власні бази даних за допомогою програмного забезпечення для баз даних [3].

4 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМА ОЦІНКИ ВПЛИВУ ЗАБРУДНЮЮЧИХ РЕЧОВИН НА НАВКОЛИШНЄ СЕРЕДОВИЩЕ

Вхід до системи представлений головним вікном, де необхідно ввести логін та пароль. В системі присутні два користувача: Admin та User. При авторизації вони мають різні доступи до інформації. Наприклад при вході під User (рис. 7), користувач не може редагувати будь-яку інформацію.

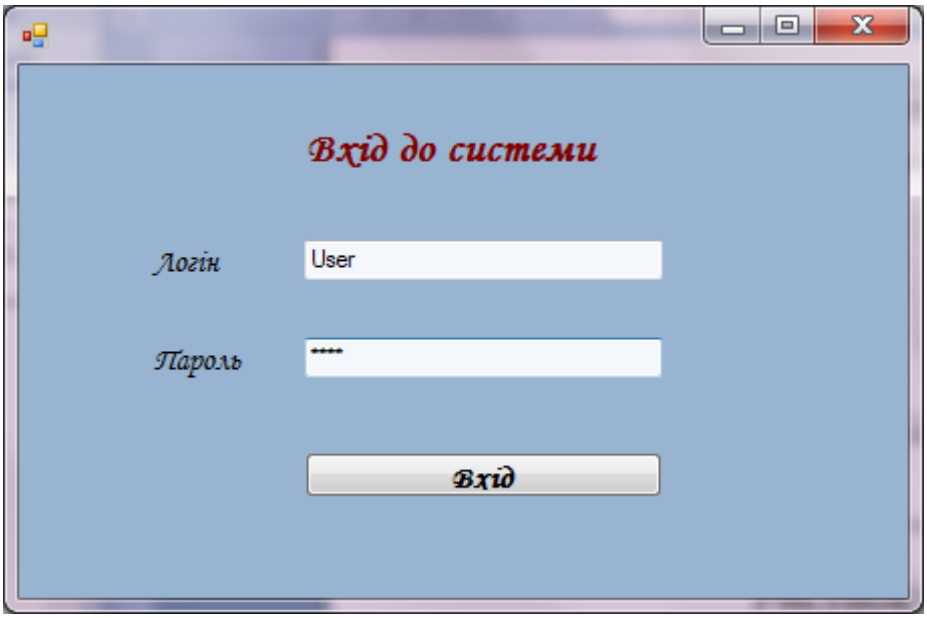

Рисунок 7 – Приклад входу до системи

При вході до ситеми, при вводі неправильного паролю виходить вікно з інформацією про неверно введениф логін або пароль. Також якщо поле залишити пустим, у вікні буде помилка (рис. 8).

Після входу до системи відкривається наступне вікно, де можна побачити основну форму для розрахунку (рис. 9), а також можливість обрати і в майбутньому інші види розрахунків (рис. 10).

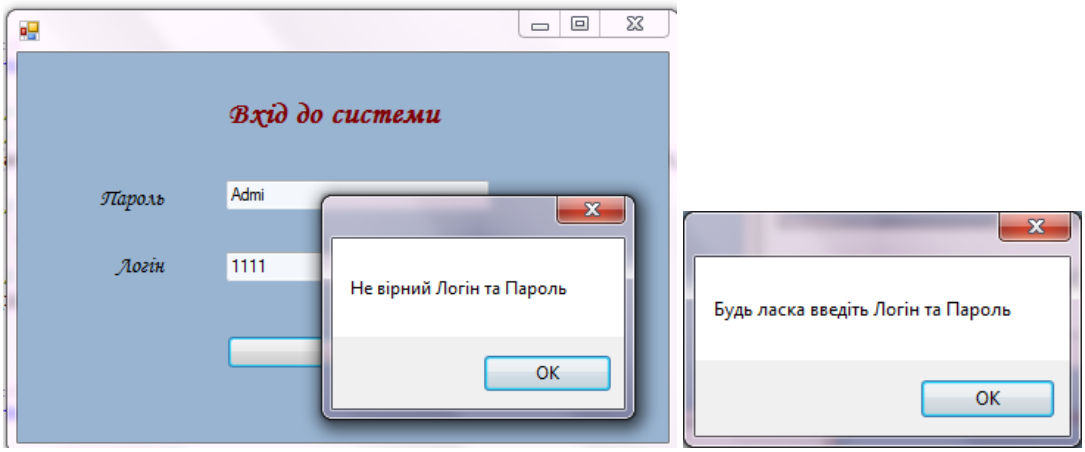

Рисунок 8 – Приклад входу до системи з неправильним логіном/паролем

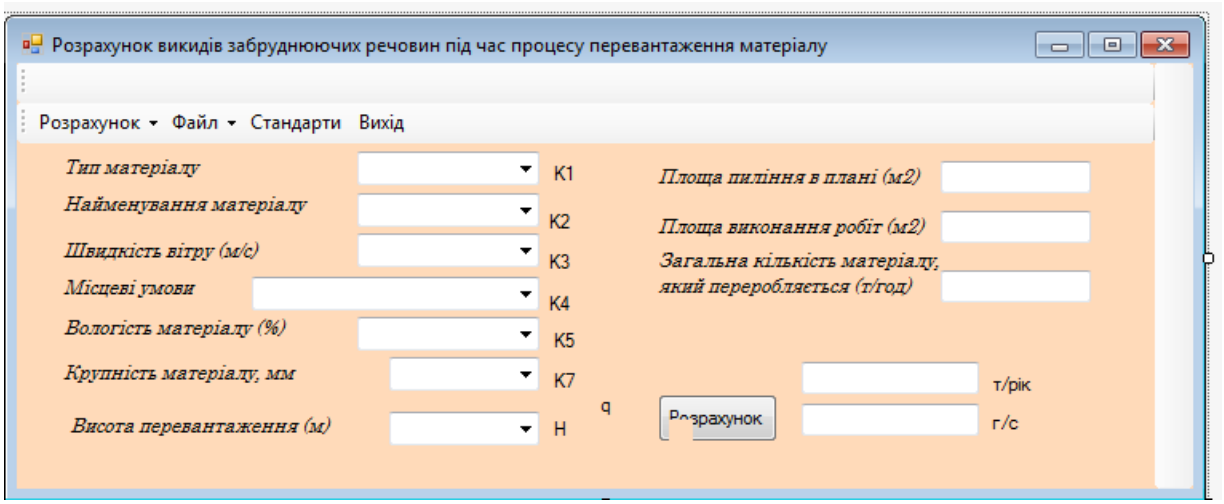

Рисунок 9 – Основна форма для розрахунку забруднюючих речовин

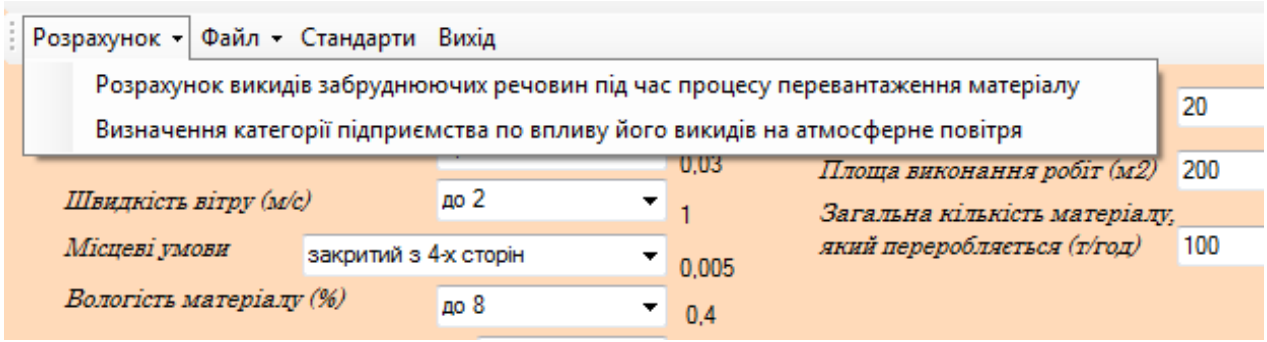

Рисунок 10 – Приклад інших видів розрахунку

Згідно наданих даних, для автоматизації розрахунків приймає форму (рис. 11), для зручності занесення інформації на розрахунок та отримання вихідної інформації.

Для виконання розрахунку необхідні вихідні дані:

– тип матеріалу, який перевантажується та його назва (перелік матеріалів наведено в табл. 8);
– найменування матеріалу (згідно табл. 3);
– швидкість вітру (згідно табл.4);

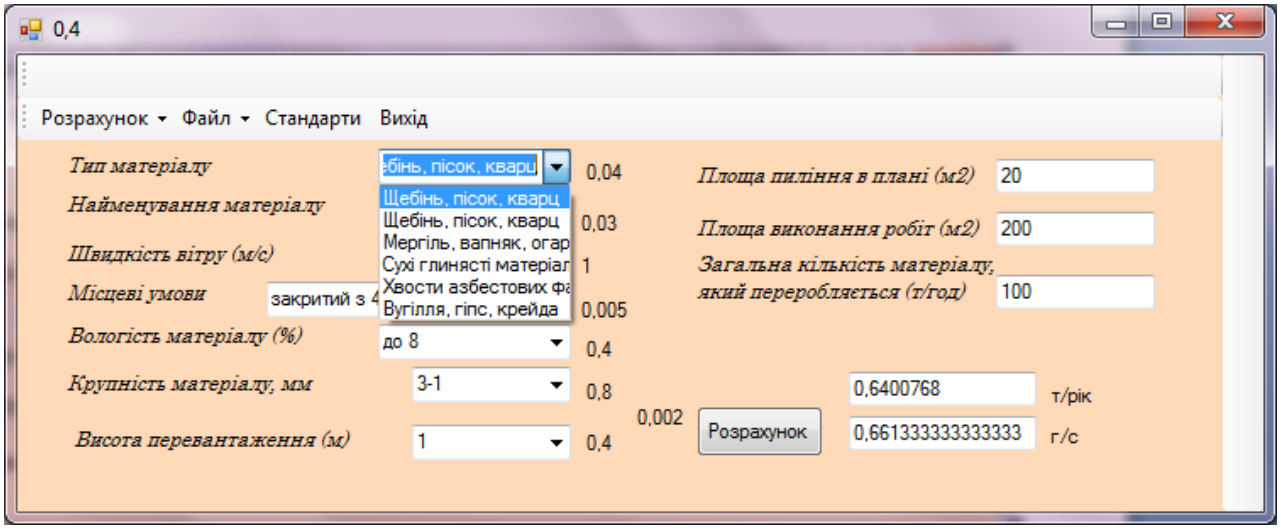

Рисунок 11 – Приклад занесення даних для розрахунку викидів забруднюючих речовин під час перевантаження матеріалу

– місцеві умови (згідно табл. 5);
– вологість матеріалу (згідно табл. 6);
– крупність матеріалу (згідно табл. 7);
– висота перевантаження (згідно табл. 9);
– площа пиління в плані (дані заносяться вручну, згідно вихідних даних);
– площа виконання робіт (дані заносяться вручну, згідно вихідних даних);

– загальна кількість матеріалу, який переробляється за годину (дані заносяться вручну, згідно вихідних даних);

Для зручності опису системи було розділено введення полів на декілька категорій:

– випадаючий список, з якого вибирається необхідні для розрахунку значення (дані відповідають значенням табл. 3-9);
– дані вносяться в ручному режимі, згідно відповідних вихідних даних;
– поля заповнюються автоматично, як результат розрахунку.

Після внесення усіх вихідних даних до полів, автоматично визначаються показники викиду забруднюючих речовин до атмосферного повітря, та відображаються в полі розрахунку.

У вкладці «Стандарти» (рис. 12) присутні всі табличні дані згідно яких ведеться розрахунок. При вході під користувачем User, можна тільки продивлятися інформацію.

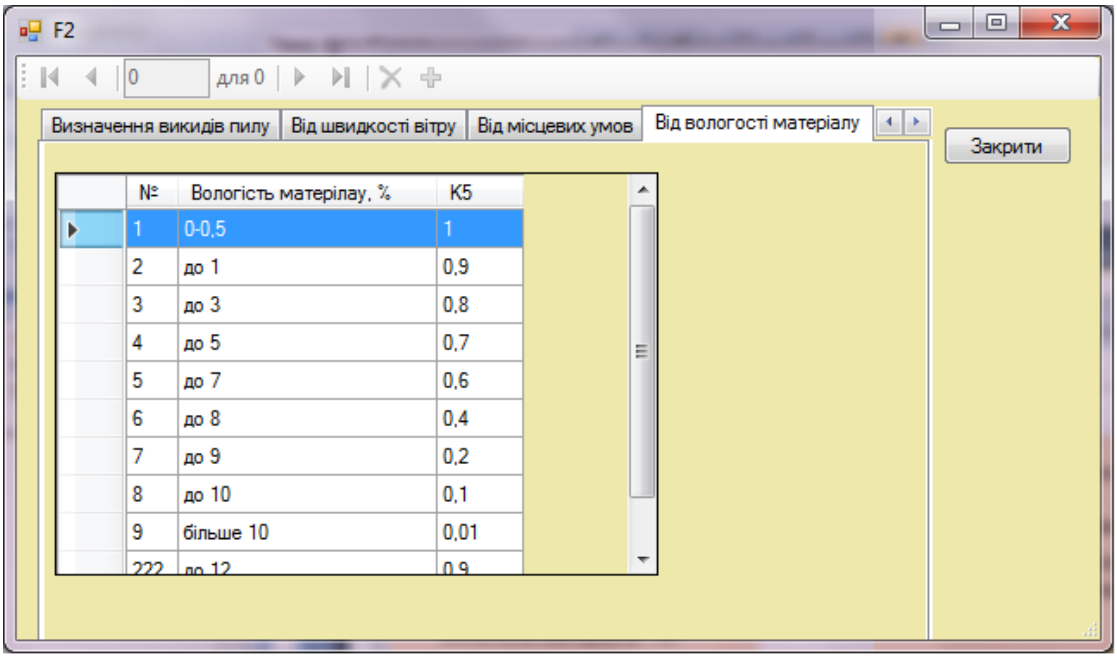

Рисунок 12 – Опис таблиці «Викид пилу»

При вході під користувачем Admin, є можливість редагувати дані. На рис. 13 та 14 показана можливість додавання даних до таблиці.

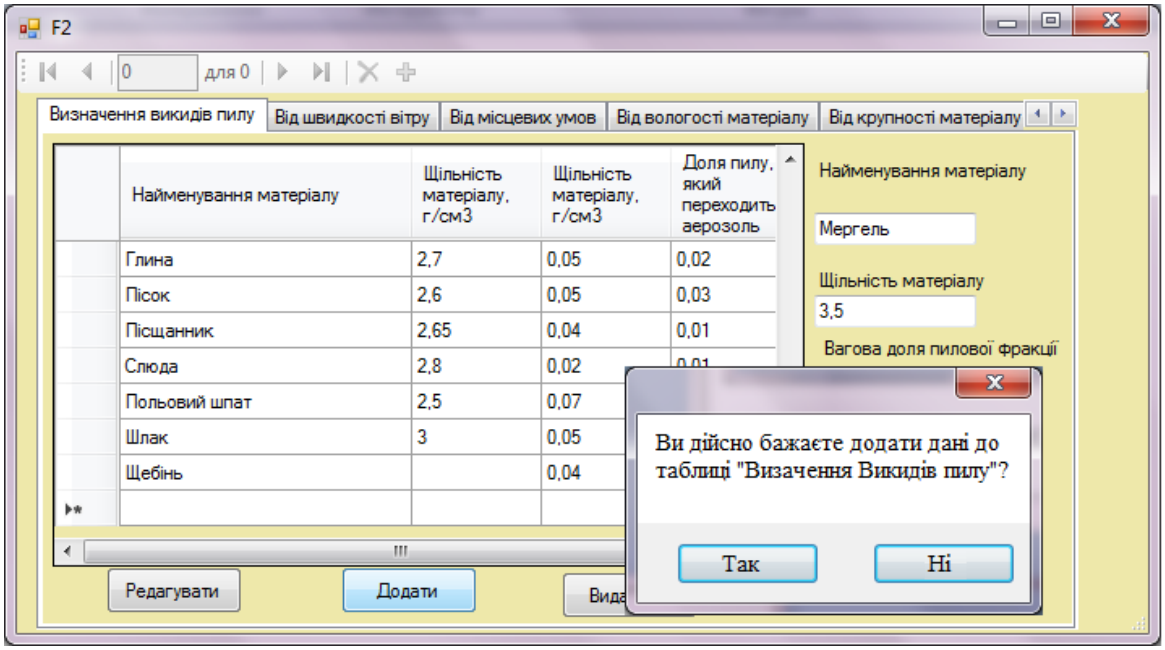

Рисунок 13 – Приклад додавання даних до таблиці «Викид пилу»

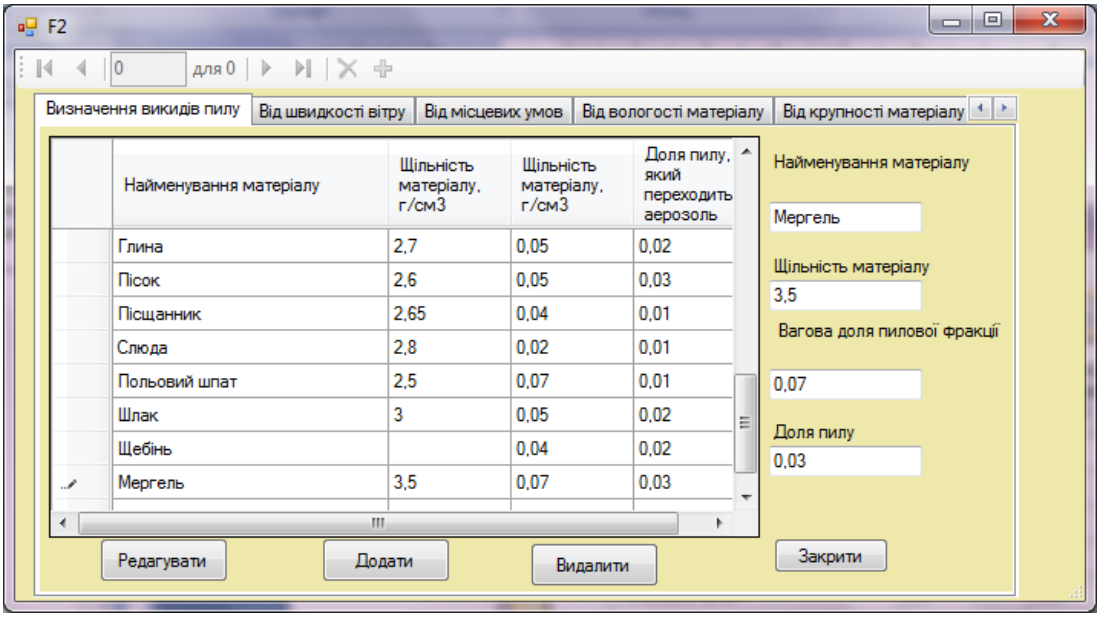

Рисунок 14 – Приклад додавання даних до таблиці «Викид пилу»

У наступному прикладі представлено видалення даних, при якому обов’язково треба виділити всю строку, інакше система видасть помилку (рис. 15,16)

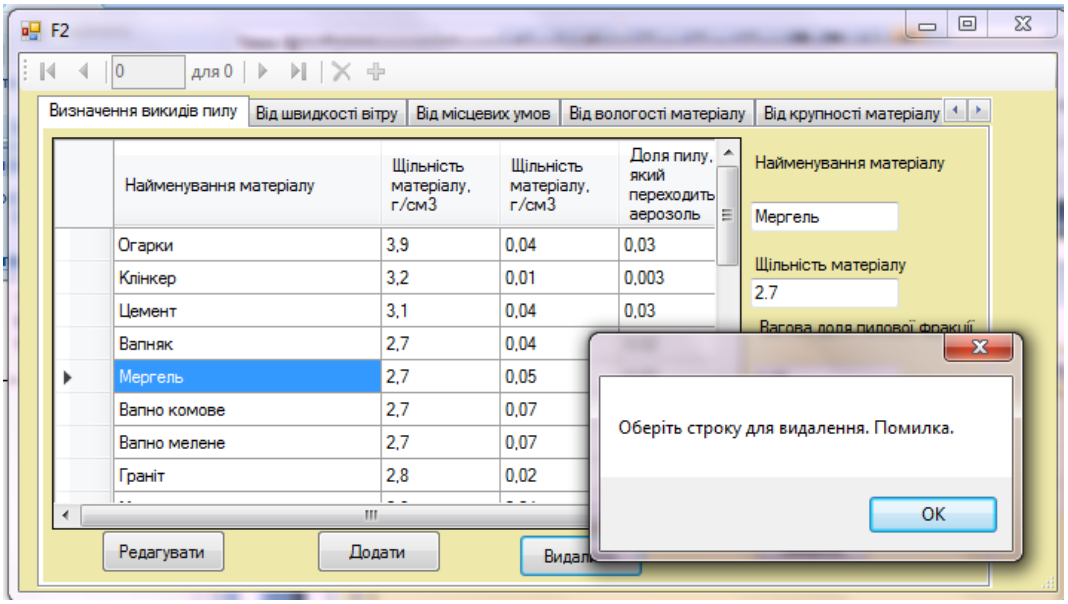

Рисунок 15 – Помилка при видаленні

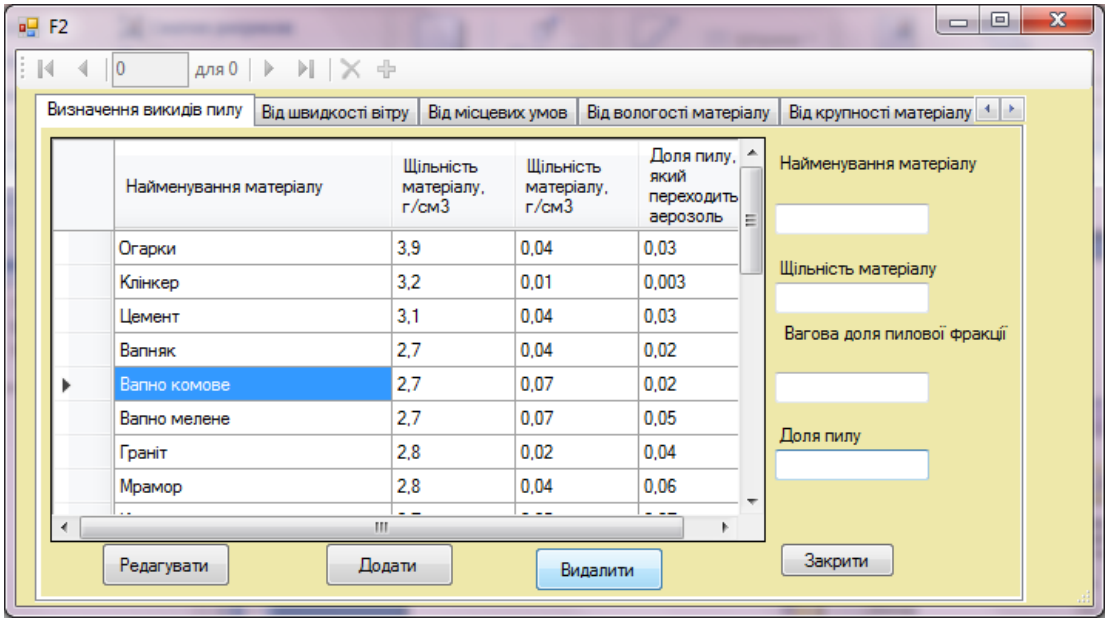

Рисунок 16 – Результат видалення об’єкту

Далі представлено можливість додавання даних (рис. 17)

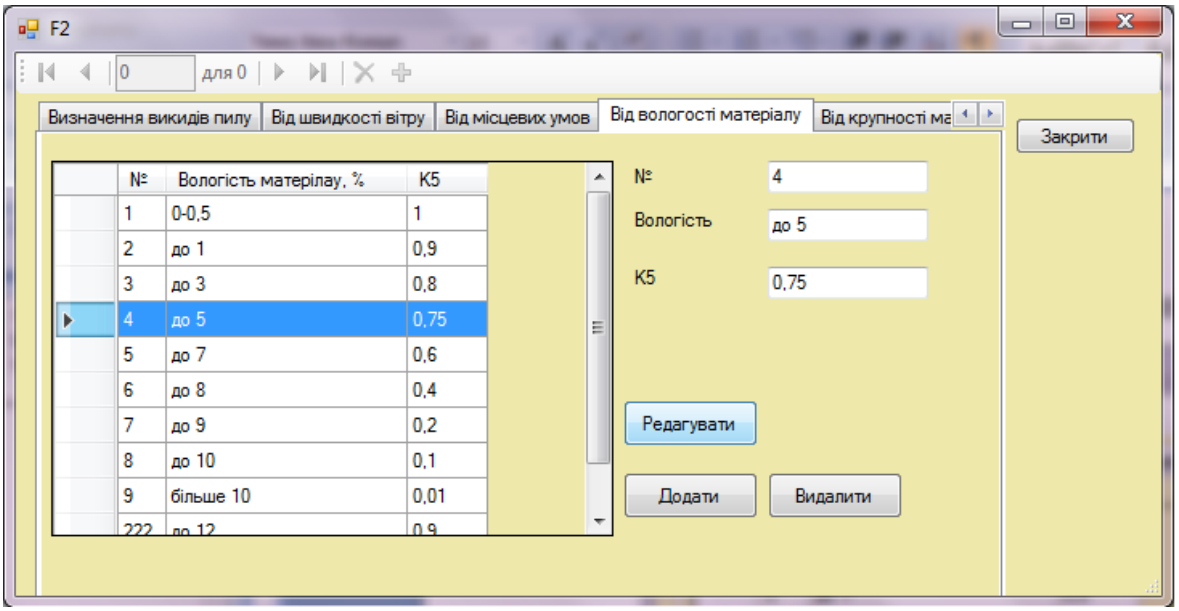

Рисунок 17 – Приклад додавання даних до таблиці «Викид пилу»

Для зручності оформлення розрахунку, є функція ЗБЕРЕЖЕННЯ та ЕКСПОРТУ розрахунку. Експортується розрахунок до документу з розширенням *.doc в такому вигляді (рис. 18,19).

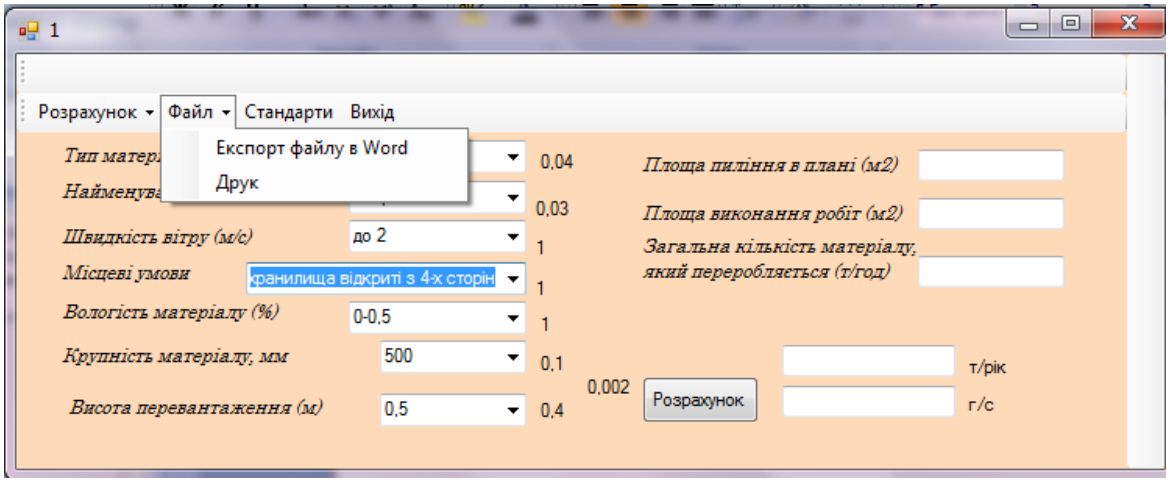

Рисунок 18 – Приклад додавання даних до таблиці «Викид пилу»

Розрахунок викидів забруднюючих речовин від процесу перевантаження, зсипання, переміщення матеріалу виконувався згідно « Сборника методик по расчету содержания загрязняющих веществ в выбросах от неорганизованных источников загрязнения атмосферы» м. Донецьк.

Загальний об'єм викидів розраховується за формулою:

$$q = A + B = \frac{k_1 \cdot k_2 \cdot k_3 \cdot k_4 \cdot k_5 \cdot k_7 \cdot G \cdot 10^6 \cdot B}{3600} + k_3 \cdot k_4 \cdot k_5 \cdot k_6 \cdot k_7 \cdot q \cdot F \qquad г/с$$

де:

A – викиди при переробці (зсипання, перевантаження, переміщення) матеріалу, г/с;

B – викиди при статистичному зберіганні матеріалу;

k_1 – вагова доля пилової фракції в матеріалі. Визначається методом відмивки та просіювання середньої проби з виділенням фракції пилу розміром 0-200 мкм;

k_2 – доля пилу (від усієї маси пилу), яка переходить в аерозоль;

k_3 – коефіцієнт, який враховує місцеві метеорологічні умови, та приймається у відповідності до табл. 42 даної методики;

k_4 – коефіцієнт, який враховує місцеві умови, ступінь захищеності вузла від зовнішніх впливів, умови пилоутворення. Береться за даними табл.5 методики;

k_5 – коефіцієнт, який враховує вологість матеріалу, та приймається у відповідності до даних табл. 6;

k_6 – коефіцієнт, який враховує профіль поверхні матеріалу, який складується та визначається як співвідношення $F_{факт}/F$. Значення k_6 знаходиться в границях 1,3-1,6 в залежності від крупності матеріалу та ступені заповнення;

k_7 – коефіцієнт, який враховує крупність матеріалу, та приймається у відповідності до табл. 7;

$F_{факт}$ – фактична поверхня матеріалу з врахуванням рельєфу його перетину (враховувати тільки площу, на якій виконуються завантажувальні-розвантажувальні роботи);

F – поверхня пилiння в плані, $м^2$;

q – винесення пилу з одного квадратного метра фактичної поверхні в умовах, коли $k_3 = 1$ та $k_5 = 1$, приймаються у відповідності з даними табл. 8;

G – сумарна кількість матеріалу, який перевантажується, т/год;

B – коефіцієнт, який враховує висоту пересипки та приймається у відповідності до табл. 9.

Найменування матеріалу	k_1	k_2	k_3	k_4	k_5	k_7	B	$F_ф$	F	G	Показники викиду забруднюючих речовин	
Пісок	0,05	0,03	1	0,005	0,4	0,8	1	20	200	100	0,0677	г/с
											0,00024	т/рік

Рисунок 19 – Приклад експортованого файлу

ВИСНОВКИ

Налаштування технологій та забезпечення нових співробітників може зажадати багато часу, а отже, переривання інших ваших проектів. Використовуйте IT-автоматизацію, щоб підвищити ефективність використання технологій. Більшість завдань в процесі інкорпорації повторюються і легко перевіряються за допомогою автоматизації.

За допомогою автоматизації IT-процесів ви можете легко створювати вбудовані та автоматизовані засоби контролю відповідності як частину більших робочих процесів і запускати збір даних та звітування щокварталу, щороку або під час транзакційних процесів. IT-автоматизація збирає та структурує дані, щоб отримати розуміння та контроль за внутрішніми процесами протягом року

Було розроблено систему розрахунку для підвищенні продуктивності, зменшення витрат, зменшення формальностей та наданні командам часу, щоб зосередитись на таких завданнях, як додаток до вартості, таких як стосунки з клієнтами або подальші дії у складних ситуаціях.

Для виконання мети, були вирішанні наступні задачі:

– досліджено предметну область;
– спроектовано базу даних;
– створено форми для роботи з базою;
– організовано для користувача меню;
– дана програма повинна має простий і зручний призначений для користувача інтерфейс.
– оптимізація зберігання даних для виключення надмірності;
– створення єдиного інформаційного простору для взаємодії учасників інформаційного обміну;
– надання цілісної картини стану об'єктів інформатизації для полегшення оперативного прийняття управлінських рішень;

- прискорення інформаційних потоків:
- оперативне отримання актуальної інформації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1 Закон «Об охране атмосферного воздуха» М, 1999.

2 Закон «Об охране окружающей среды». М., 2002.

3. What Is a Database? URL: https://www.oracle.com/database/what-is-database/. (Дата звернення 24.03.2021).

4. 10 Most Used Databases By Developers In 2020 URL: https://analyticsindiamag.com/10-most-used-databases-by-developers-in-2020/. (Дата звернення 28.03.2021).

5 SAP HANA - что это такое? Новые возможности для управления бизнесом. URL: https://asapcg.com/press-center/articles/sap-hana/. (Дата звернення 30.03.2021).

6 The Pros and Cons of 8 Popular Databases. URL: https://www.keycdn.com/blog/popular-databases. (Дата звернення 02.04.2021).

7 Сборника методик по расчету содержания загрязняющих веществ в выбросах от неорганизованных источников загрязнения атмосферы» м. Донецьк. Донецк: ОАО "УкрНТЭК", отдел НТИ, 155 с.

8 Системы управления технологическими процессами и информационные технологии. URL: https://studme.org/293442/tehnika/proektirovanie_sistem_avtomatizatsii. (Дата звернення 09.04.2021).

9 What is a Database? Definition, Meaning, Types, Example. URL: https://www.oracle.com/database/what-is-database/. (Дата звернення 14.03.2021).

10 Transact SQL. URL: https://ru.wikipedia.org/wiki/Transact−SQL (дата звернення 25.04.2020)

11 What is Microsoft SQL Server and what is it used for. URL: https://www.infotectraining.com/blog/what−is−microsoft−sql−server−and−what−is−it−used−for. (дата звернення 25.04.2020)

Додаток

Уривок лістингу програми

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApp1
{
    public partial class User : Form
    {
        public User()
        {
            InitializeComponent();
        }
       string cs = @"Data Source =

(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Irina\Desktop\ДЕН\проба\123.mdf;Integrat
ed Security = True; Connect Timeout = 30";
      private void InitializeComponent();
        {
            this.SuspendLayout();

             User

         this.ClientSize = new System.Drawing.Size(561, 330);
           this.Name = "User";
           this.ResumeLayout(false);

       }

        private void button1_Click(object sender, EventArgs e)
        {
            var Login1 = "Admin";
            var Pass1 = "1111";
            var Login2 = "User";
            var Pass2 = "2222";

            if (UserName.Text == "" || Password.Text == "")
            {
                MessageBox.Show("Будь ласка введіть Логін та Пароль");
                return;
            }
            if ((UserName.Text == Login1) && (Password.Text == Pass1))
            {
                F1 newForm = new F1();
                newForm.MdiParent = this.MdiParent;
                newForm.Show();
                this.Close();
            }
            if ((UserName.Text == Login2) && (Password.Text == Pass2))
            {
                this.Hide();
                F1 f1 = new F1();
```

```
                f1.MdiParent = this.MdiParent;
                f1.Show();
               // this.Close();
            }
            else
            {
                MessageBox.Show("Не вірний Логін та Пароль");
                return;
            }

        }}}


using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;


namespace WindowsFormsApp1
{
    public partial class F1 : Form
    {
        public F1()
        {

            InitializeComponent();
        }


        string s, a, b, c, i, K7, K4, P, K1, K2, K3, K5, H, q;

        private void Form1_Load(object sender, EventArgs e)
        {
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"_123DataSet2.Znachenya_B". При необходимости она может быть перемещена или удалена.
              this.znachenya_BTableAdapter.Fill(this._123DataSet.Znachenya_B);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"_123DataSet1.Spead_vitru". При необходимости она может быть перемещена или удалена.
            this.spead_vitruTableAdapter.Fill(this._123DataSet1.Spead_vitru);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"_123DataSet.Vologist". При необходимости она может быть перемещена или удалена.
            this.vologistTableAdapter.Fill(this._123DataSet.Vologist);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"_123DataSet.Spead_vitru". При необходимости она может быть перемещена или удалена.
            this.spead_vitruTableAdapter.Fill(this._123DataSet.Spead_vitru);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"_123DataSet.Znachenya_Q". При необходимости она может быть перемещена или удалена.
            this.znachenya_QTableAdapter.Fill(this._123DataSet.Znachenya_Q);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"_123DataSet.Vykyd_puly". При необходимости она может быть перемещена или удалена.
            this.vykyd_pulyTableAdapter.Fill(this._123DataSet.Vykyd_puly);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"_123DataSet.Material". При необходимости она может быть перемещена или удалена.
            this.materialTableAdapter.Fill(this._123DataSet.Material);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"_123DataSet.Znachenya_B". При необходимости она может быть перемещена или удалена.
            this.znachenya_BTableAdapter.Fill(this._123DataSet.Znachenya_B);
```

```
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"_123DataSet.Rozmir". При необходимости она может быть перемещена или удалена.
            this.rozmirTableAdapter.Fill(this._123DataSet.Rozmir);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"_123DataSet.Miszevi_umovy". При необходимости она может быть перемещена или удалена.
            this.miszevi_umovyTableAdapter.Fill(this._123DataSet.Miszevi_umovy);

        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {
            s = textBox1.Text;
            a = Convert.ToDouble(s);
        }

        private void textBox2_TextChanged(object sender, EventArgs e)
        {
            s = textBox2.Text;
            b = Convert.ToDouble(s);
        }

        private void textBox4_TextChanged(object sender, EventArgs e)
        {
            s = textBox4.Text;
            P = Convert.ToDouble(s);
        }
        private void labelK5_TextChanged(object sender, EventArgs e)
        {
            s = labelK5.Text;
            K5 = Convert.ToDouble(s);
        }


        private void label20_TextChanged(object sender, EventArgs e)
        {
            s = label20.Text;
            K7 = Convert.ToDouble(s);
        }

        private void labelK2_TextChanged(object sender, EventArgs e)
        {
            s = labelK2.Text;
            K2 = Convert.ToDouble(s);
        }
        private void labelK3_TextChanged(object sender, EventArgs e)
        {
            s = labelK3.Text;
            K3 = Convert.ToDouble(s);
        }

        private void labelK4_TextChanged(object sender, EventArgs e)
        {
            s = labelK4.Text;
            K4 = Convert.ToDouble(s);
        }

        private void labelK1_TextChanged(object sender, EventArgs e)
        {
            s = labelK1.Text;
            K1 = Convert.ToDouble(s);
        }

        private void button2_Click(object sender, EventArgs e)
```

```
        {
            this.Hide();
            F2 f2 = new F2();
            f2.ShowDialog();
        }

        private void q_TextChanged(object sender, EventArgs e)
        {
           s = q.Text;
           q = Convert.ToDouble(s);
        }
        private void label1_TextChanged(object sender, EventArgs e)

        {
            s = label1.Text;
            H = Convert.ToDouble(s);
        }


        private void button1_Click(object sender, EventArgs e)
        {
            c = (K1 * K2 * K3 * K4 * K5 * K7 * P * 1000000 * H) / 3600 + (K3 * K4 * K5 *
(b / P) * K7 * b );
            textBox3.Text = Convert.ToString(c);
            i = (K1 * K2 * K3 * K4 * K5 * K7 * P * H) + (K3 * K4 * K5 * (b / P) * K7 * b
);
            textBox5.Text = Convert.ToString(i);
        }

        private void button3_Click(object sender, EventArgs e)
        {
            this.Hide();
            User f0 = new User();
            f0.MdiParent = this.MdiParent;
            f0.Show();

        }

        private void F1_FormClosing(object sender, FormClosingEventArgs e)
        {
            Application.Exit();
        }

        private void toolStripButton1_Click(object sender, EventArgs e)
        {
            this.Hide();
            User f0 = new User();
            f0.MdiParent = this.MdiParent;
            f0.Show();
        }

        private void toolStripButton2_Click(object sender, EventArgs e)
        {
            this.Hide();
            F2 f2 = new F2();
            f2.ShowDialog();
    }}}


using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
```

```
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApp1
{
    public partial class F2 : Form
    {
        public F2()
        {
            InitializeComponent();
        }

        private void F_2_Load(object sender, EventArgs e)
        {
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"_123DataSet.Znachenya_Q". При необходимости она может быть перемещена или удалена.
            this.znachenya_QTableAdapter.Fill(this._123DataSet.Znachenya_Q);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"_123DataSet.Znachenya_B". При необходимости она может быть перемещена или удалена.
            this.znachenya_BTableAdapter.Fill(this._123DataSet.Znachenya_B);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"_123DataSet.Rozmir". При необходимости она может быть перемещена или удалена.
            this.rozmirTableAdapter.Fill(this._123DataSet.Rozmir);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"_123DataSet.Vologist". При необходимости она может быть перемещена или удалена.
            this.vologistTableAdapter.Fill(this._123DataSet.Vologist);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"_123DataSet.Miszevi_umovy". При необходимости она может быть перемещена или удалена.
            this.miszevi_umovyTableAdapter.Fill(this._123DataSet.Miszevi_umovy);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"_123DataSet.Spead_vitru". При необходимости она может быть перемещена или удалена.
            this.spead_vitruTableAdapter.Fill(this._123DataSet.Spead_vitru);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
"_123DataSet.Vykyd_puly". При необходимости она может быть перемещена или удалена.
            this.vykyd_pulyTableAdapter.Fill(this._123DataSet.Vykyd_puly);

        }

        private void button3_Click(object sender, EventArgs e)
        {
            {
                if (dataGridView1.SelectedRows.Count > 0)
                {
                    dataGridView1.Rows.RemoveAt(dataGridView1.SelectedRows[0].Index);
                }
                else
                {
                    MessageBox.Show("Оберіть строку для видалення. Помилка.");
                }
            }
        }

        private void button4_Click(object sender, EventArgs e)
        {{
                if (dataGridView2.SelectedRows.Count > 0)
                {
                    dataGridView2.Rows.RemoveAt(dataGridView2.SelectedRows[0].Index);
                }
                else
                {
                    MessageBox.Show("Оберіть строку для видалення. Помилка.");
              }
```

```
    }
}

private void button7_Click(object sender, EventArgs e)
{
    {
        if (dataGridView3.SelectedRows.Count > 0)
        {
            dataGridView3.Rows.RemoveAt(dataGridView3.SelectedRows[0].Index);
        }
        else
        {
            MessageBox.Show("Оберіть строку для видалення. Помилка.");
        }
    }
}

private void button10_Click(object sender, EventArgs e)
{
    {
        if (dataGridView4.SelectedRows.Count > 0)
        {
            dataGridView4.Rows.RemoveAt(dataGridView4.SelectedRows[0].Index);
        }
        else
        {
            MessageBox.Show("Оберіть строку для видалення. Помилка.");
        }
    }
}

private void button13_Click(object sender, EventArgs e)
{{
        if (dataGridView5.SelectedRows.Count > 0)
        {
            dataGridView5.Rows.RemoveAt(dataGridView5.SelectedRows[0].Index);
        }
        else
        {
            MessageBox.Show("Оберіть строку для видалення. Помилка.");
        }
    }
}

private void button16_Click(object sender, EventArgs e)
{
    {
        if (dataGridView6.SelectedRows.Count > 0)
        {
            dataGridView6.Rows.RemoveAt(dataGridView6.SelectedRows[0].Index);
        }
        else
        {
            MessageBox.Show("Оберіть строку для видалення. Помилка.");
        }
    }
}

private void button19_Click(object sender, EventArgs e)
{
    {
        if (dataGridView7.SelectedRows.Count > 0)
        {
            dataGridView7.Rows.RemoveAt(dataGridView7.SelectedRows[0].Index);
```

```
                }
                else
                {
                    MessageBox.Show("Оберіть строку для видалення. Помилка.");
                }
            }
        }

        private void button22_Click(object sender, EventArgs e)
        {
           // this.Hide();
            F1 f1 = new F1();
            f1.MdiParent = this.MdiParent;
            f1.Show();
            this.Close();
        }

        private void AddVologist_Click(object sender, EventArgs e)
        {
           Vologist newForm = new Vologist();
           newForm.Show();
          dataGridView4.Rows.Add(new object[] { textBox1.Text, textBox2.Text,
textBox3.Text });

          //  Добавляем созданную строку в коллекцию строк таблицы.
          _123DataSet.Vologist.Rows.Add(row);

            // Сохраняем изменения.
            _123DataSet.Vologist.AcceptChanges();
        }

        private void bindingNavigator2_Click(object sender, EventArgs e)
        {
            this.Validate();
            this.tableBindingSource.EndEdit();
            this.tableAdapterManager1.UpdateAll(this._123DataSet);
        }

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApp1
{
    class WindowsFormsApp1F1
    {
        [STAThread]
        string conn_string = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Irina\Desktop\ДЕН\проба\123.mdf;I
ntegrated Security=True;Connect Timeout=30";
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new User());
        }
    }
}
```