УДК 004.65
DOI: https://doi.org/10.26642/ten-2021-2(88)-63-71

О.В. Коротун, к.пед.н., доц.
Г.В. Марчук, ст. викладач
В.В. Мєдвєдєв, студент
Державний університет «Житомирська політехніка»

Проєктування та розробка документоорієнтованої системи керування базами даних

Стаття присвячена проєктуванню та розробці документоорієнтованої системи керування базами даних. Описано компоненти сучасних систем керування базами даних, їх трирівнева архітектура, наведено загальну блок-схему роботи системи. Продемонстровано роботу усіх модулів розробленої системи, зокрема IParseStrategy та IExecuteStrategy. Створено систему парсера, наведено порядок виконання деяких класів, алгоритми роботи компонентів DCL та DDL. Представлено інтерфейс розробленої системи керування базами даних, визначено загальний порядок виконання запиту в ній, показано виконання запитів на створення бази даних та колекцій. Робота з даними в системі керування базами даних дозволяє створювати запити на вставку та редагування даних, їх вибірку та видалення. У системі додатково передбачено можливість виведення даних у файл форматів PDF та Excel. Для операцій, які передбачають поєднання багатьох таблиць, передбачено реалізацію команд Union. Для поліпшення функціоналу до автоматизації в систему введені можливості створення представлень, процедур, що зберігаються, та тригерів на команди. Розроблена система працює з невеликими наборами даних та має низку переваг: невеликі розміри системи, зрозумілий інтерфейс користувача; невеликі розміри коду; утворене абстрактне ядро; обробка регламентних правил через спеціальні об'єкти тощо. Встановлена система підтримки цілісності даних на основі реляційної моделі даних гарантує, що користувач не зможе випадково видалити дані, що використовуються іншими даними. Розроблена система призначена для використання підприємствами малого та мікробізнесу, яким з одного боку потрібна система керування базами даних з підтримкою необхідного мінімуму механізмів контролю цілісності даних, а з іншого – потрібна велика мобільність та зручність використання.

Ключові слова: *бази даних; система керування базами даних; документоорієнтована СКБД; колекція; документ.*

Актуальність теми обумовлена тим, що з розвитком мережевих технологій та технологій збору інформації, системи для організації та управління нею стають основними й найважливішими компонентами усіх підприємств. Навіть маленькі організації наразі вимушені працювати з величезними потоками інформації. Розвиток вимог до надійності та швидкодії призвів до виділення окремого типу програмного забезпечення, що задовольняє ці вимоги й здійснює контроль над великими потоками даних.

Під час проєктування будь-якої системи, що передбачає управління даними, потрібно визначитися з моделлю представлення інформації, на яку буде спиратися система. Тому потрібно означити цілі такого проєктування – створення системи контролю цілісності даних, характерних для реляційної моделі даних, та відносно легка можливість оперувати ними, шо характерно для документоорієнтованих систем керування базами даних (СКБД).

Аналіз останніх досліджень та публікацій, на які спираються автори. Основні поняття теорії баз даних, процес нормалізації БД, методологію їх проєктування , архітектуру БД, сучасні СКБД, мову запитів SQL тощо розглядали такі вітчизняні автори: Л.С. Глоба [7], В.М. Джулій [1], Л.І. Жолубак та Н.Є. Буряк [3], І.С. Зінов'єва та В.О. Артемчук [4], В.В. Осадчий і С.В. Шаров [8] та ін.

Романюк О. та Микитюк І. описали розроблену модель захисту інформації, яка зберігається в базах даних шляхом розмежування доступу користувачів до різних рівнів управління базою даних [6].

Мамалига Н.Є. та Катаєва А.І. розглянули застосування системи керування базою даних, яка буде легкою у використанні і одночасно із цим забезпечувати широкий технічний функціонал, а також якість та надійність зберігання даних у сучасних умовах ІТ-індустрії [5].

Головачов І.А. та Геселева Н.В. описали особливості обробки інформації в сучасному бізнесі, дослідили електронні сховища інформації та системи керування базами даних, розглянули класифікацію баз даних, зокрема реляційні та нереляційні бази даних [1].

Метою статі є проєктування та розробка документноорієнтованої системи керування базами даних.

Викладення основного матеріалу. Як основу для створення СКБД було вирішено використати реляційну модель представлення інформації, яка забезпечує необхідний контроль цілісності даних з введенням додаткової можливості відключити перевірку з переходом до моделі даних, схожої до

© О.В. Коротун, Г.В. Марчук, В.В. Мєдвєдєв, 2021

документної, для збільшення швидкодії на наборах даних, що не вимагають особливого контролю та підтримки.

Наступним кроком є визначення необхідного мінімуму компонентів системи. Зі структурної точки зору СКБД поділяється на такі компоненти [10]:

1) контролер словника забезпечує доступ до системного каталогу й роботу з ним;

2) контролер файлів маніпулює файлами з даними й відповідає за розподіл дискового простору;

3) контролер бази даних взаємодіє із запущеними користувачами програмами й запитами, приймає запити й перевіряє зовнішні й концептуальні схеми для визначення тих концептуальних записів, які необхідні для виконання запиту до БД. Після цього контролер БД викликає контролер файлів для виконання запиту;

4) контролер буфера відповідає за перенос даних між оперативною пам'яттю й жорстким диском;

5) препроцесори та компілятори мов DDL/DML/DCL реалізовані з використанням внутрішніх об'єктів;

6) системний каталог (словник даних) – системний каталог або словник даних є місцем зберігання інформації, що описує дані в БД.

Множина команд DDL відповідає за створення об'єктів бази даних. Виконання множини команд DML пов'язане із взаємодією користувача з даними. Множина команд DCL відповідає за визначення прав користувачів. У межах реалізації трирівневої архітектури ANSI-SPARC виокремлено такі рівні: зовнішній – складається з множини представлень бази даних, у проєкті реалізується правами, що надано відповідним користувачам у логінах для входу; концептуальний – проміжний рівень, що підтримує сутності, їх атрибути та зв'язки, реалізується шляхом використання проміжних об'єктів та метаданих; внутрішній рівень – містить опис структур даних й організації окремих файлів для зберігання інформації на запам'ятовуючих пристроях.

Як основну модель обробки даних використано поєднання класичної системи реляційних СКБД з інтеграцією підходів більш характерних для документоорієнтованих СКБД [11]. Для забезпечення подібної реалізації введено поняття *безпечного* та *небезпечного* режимів, які відповідають за особливості координації взаємодії компонентів системи. В *безпечному* режимі забезпечується підтримка контролю існування даних та підтримка атрибутів даних. В *небезпечному* режимі – дані перевірки вимикаються для поліпшення швидкодії системи.

Під час реалізації систем СКБД потрібно вирішити декілька великих питань, одним з основних є питання безпеки: для користувача головне – це питання авторизації у системі та безпеки й доступності цієї системи. Для безперебійної роботи реалізовано такий протокол дій (рис. 1).

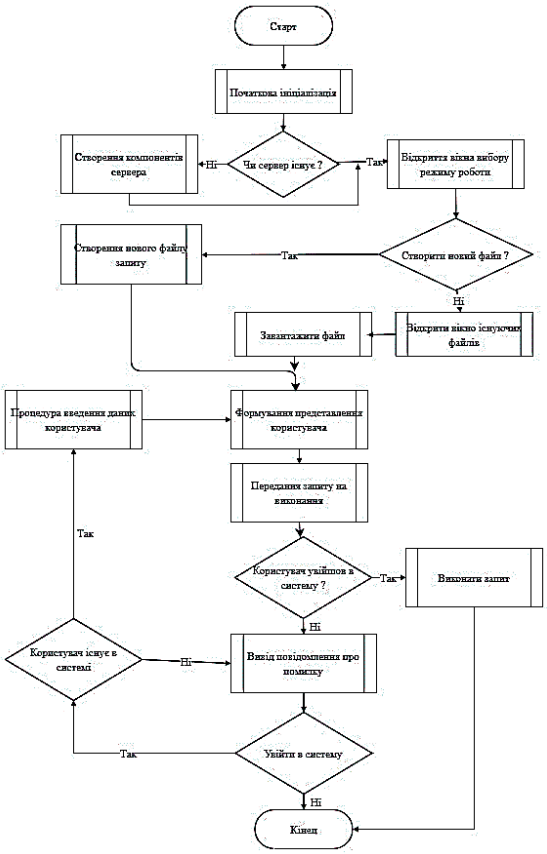

Рис. 1. Загальна блок-схема протоколу дій системи

У розробленій СКБД передбачено, що при виконані запиту код передається в статичний клас Interpret, який використовує методи зі статичних класів Parser та Executer, що координують роботу об'єктів реалізації інтерфейсів IParseStrategy, IExecuteStrategy, IParseSecureStrategy, IExecuteSecureStrategy. Об'єкт класу Executer звертається до статичного класу DBMSIO, використовує об'єкт IIOIDBMSStrategy для визначення стратегії роботи з файловою системою та об'єкт ILoggerStrategy, що відповідає за журналювання. Якщо під час виконання класу Interpret виникає виключна ситуація, керування передається контролюючій конструкції. Якщо при ініціалізації інтерпретатора в ньому було налаштовано об'єкт, що реалізує інтерфейс IInterpretSecureStrategy, відбувається передача обробки виключної ситуації йому, якщо цей об'єкт не передано, відбувається обробка виключення по протоколу за замовчуванням.

На першому кроці виконання лістинг запиту передається в об'єкт реалізації IParseStrategy (рис. 2).

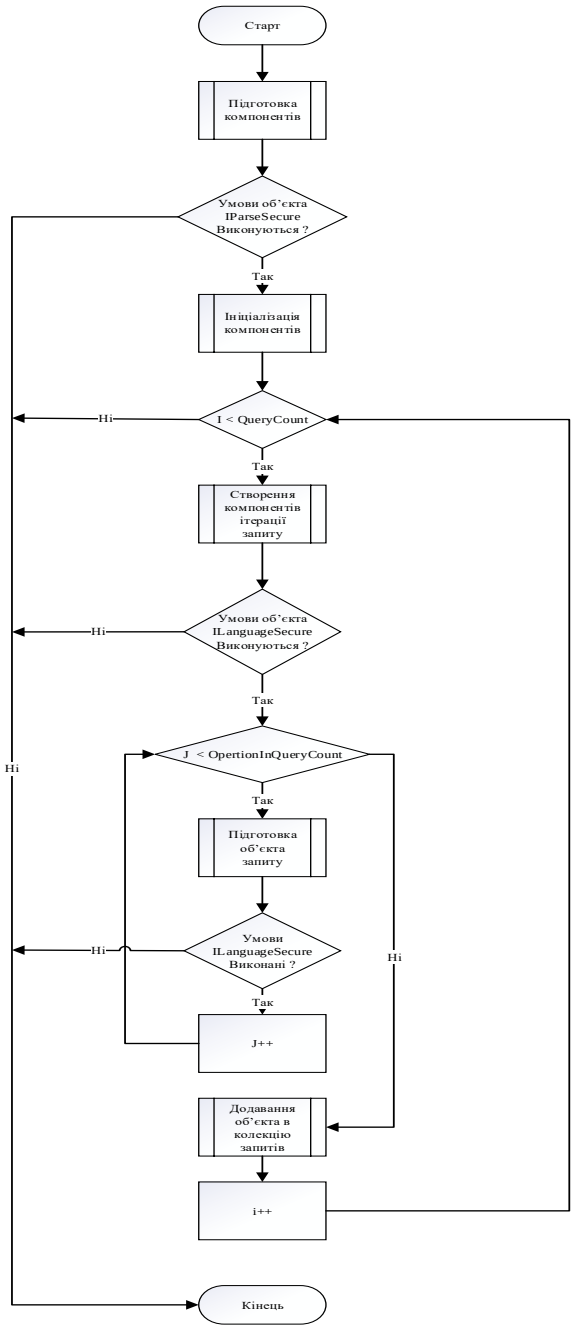

Рис. 2. Робота модуля IParseStartegy

Наступним кроком об'єкт IParseResult передається в метод класу інтерфейсу IExecuteStrategy (рис. 3), який перебирає об'єкти контексту та перевіряє права користувача.

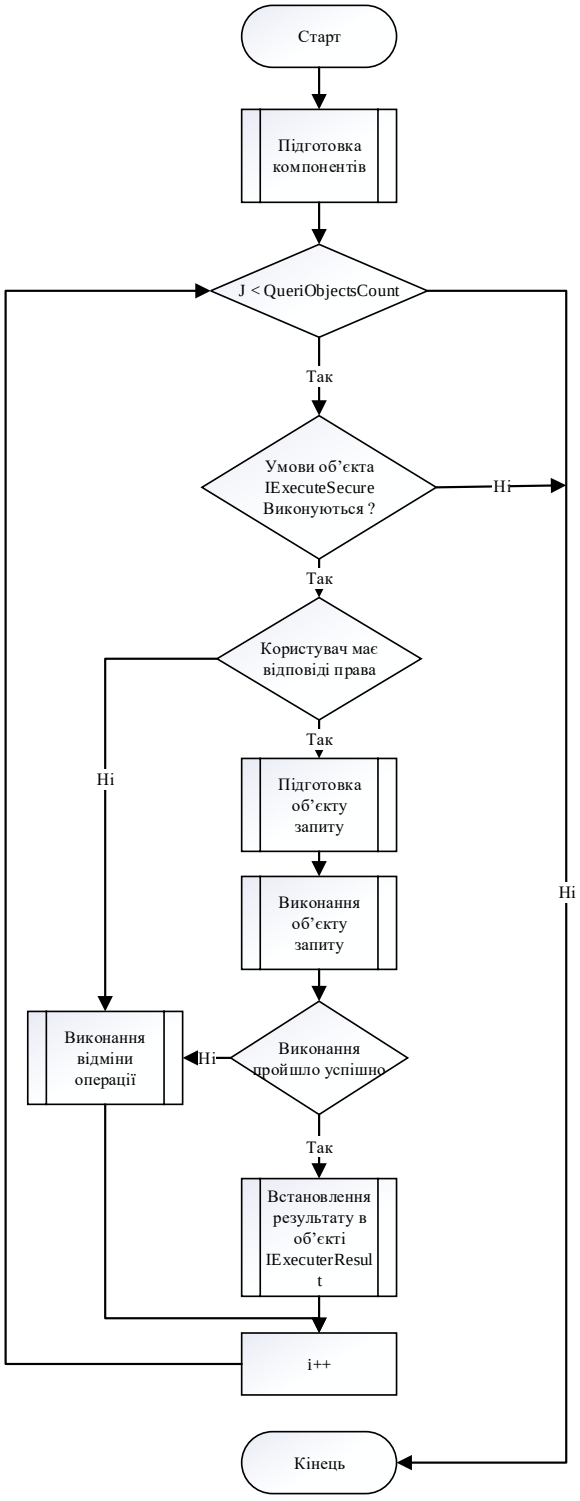

Рис. 3. Робота IExecuteStartegy

Таким чином, пара об'єктів IParseStrategy та IExecuteStategy, окрім саме виконання операції та підтримки системи безпеки, відповідає також за дотримання контролю за семантичною та лексичною коректністю коду запиту. Ця пара об'єктів є об'єктами конфігурації, робота яких координується через клас *Interpret* і на рисунку 4 показано етапи передачі керування в системі парсера.

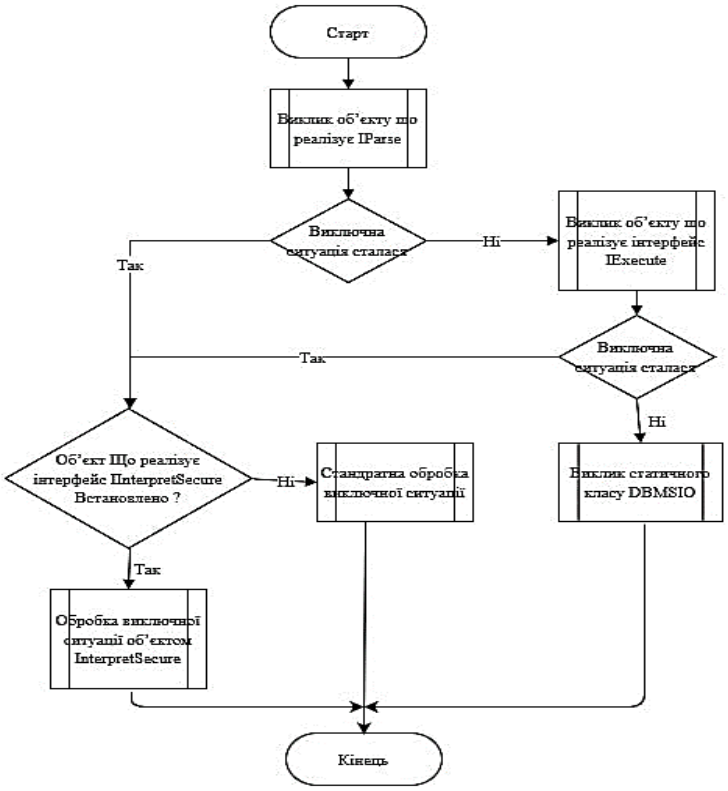

Рис. 4. Етапи передачі керування в системі парсера

Кожен об'єкт виконання, що відповідає команді після ініціалізації, передає в метод класу DBMSIO (ExecuteCommand) специфікатор операції (визначено перерахуванням (enum)) та додаткову службову інформацію, необхідну для виконання команди.

Під час виклику команди з множини *DCL* відбувається вивантаження об'єкта користувача або логіна зі зміною відповідних прав та повторним збереженням.

При виконанні команди з множини *DDL* відбувається обробка переданої інформації, ініціалізація, серіалізація та збереження об'єкта на носії з використанням API операційної системи.

Для виконання команд *DML* відбувається вибірка даних з використанням об'єктів IStorageStrategy, які визначають стратегію зберігання для індексованих, неіндексованих та віртуальних об'єктів, та подальша фільтрація вибірки. Далі під час виконання команди *Select*, якщо умов фільтрації немає, дані передаються у тому вигляді, в якому вони були зчитані зі сховища даних, після чого вони запаковуються у об'єкті, що реалізує інтерфейс IJsonMethodResult, після чого цей об'єкт повертається, передаючи результат та статус виконання операції.

Під час виконання операцій видалення (*Delete*) відбувається вибірка з фільтрацією. Після чого здійснюється перевірка на допустимість видалення, якщо дані можливо видалити, відбувається видалення відповідних записів та збереження оновленої таблиці.

При виконанні команди оновлення (*Update)* відбувається вибірка даних, їх фільтрація та оновлення даних з перевіркою імені поля та сумісності типів з подальшим збереженням оновленої таблиці.

Під час виконання команди вставки (*Insert*) відбувається зчитування таблиці метаінформації та перевірка коректності типів та атрибутів, які закріплені за цим полем. Якщо умови дотримано, відбувається збереження таблиці зі вставленими даними.

За фільтрацію та встановлення результуючої множини об'єктів відповідає об'єкт, що реалізує інтерфейс *IWhere*. Формування вибірки базується на таких твердженнях:

- параметри фільтрації можуть складатися з багатьох умов;
- кожна умова складається з трьох частин:
  - lValue – значення, зазначене по ліву сторону від умови порівняння;
  - constraintOperator – оператор, який розділяє дві частини виразу й визначає характер порівняння;
  - rValue – значення, зазначене по праву сторону від оператора порівняння;
- кожна умова своїм результатом повертає множину значень, що їй задовольняють;
- умови, об'єднані логічним «І», повертають множину, яка складається зі спільних елементів усіх множин елементарних умов (три частини описані раніше);

- умови, об’єднані логічним «АБО», повертають спільну множину;
- умова, яка складається з двох констант, розділених оператором порівняння, не застосовується для фільтрації;
- якщо значення в умові не є строковим літералом і розділене крапкою, значення зліва від крапки приймається за ім’я таблиці, з якої потрібно отримати значення;
- якщо ім’я таблиці не зазначене і значення не є літералом, ім’я відповідної таблиці визначається на основі множини таблиць, з яких відбувається вибірка;
- якщо в декількох таблицях є стовпчик з ім’ям, яке відповідає значенню, що потрібно встановити, такий запит визнається помилковим, оскільки неможливо визначити, який елемент потрібно використати.

Таким чином загальний порядок виконання запиту згори-донизу приймає такий вигляд: код запиту передається в систему Interpret, після чого послідовно викликаються Parse і Execute компоненти, що після обробки запиту звертаються до системи DBMSIO, яка у свою чергу звертається до об’єкта реалізації компонентів для роботи з файлової системи цільової ОС.

Реалізація СКБД

Під час проєктування графічного інтерфейсу потрібно було вирішити питання доступності інтерфейсу для користувача. Першочергово потрібно було визначитися з необхідними елементами (рис. 5).

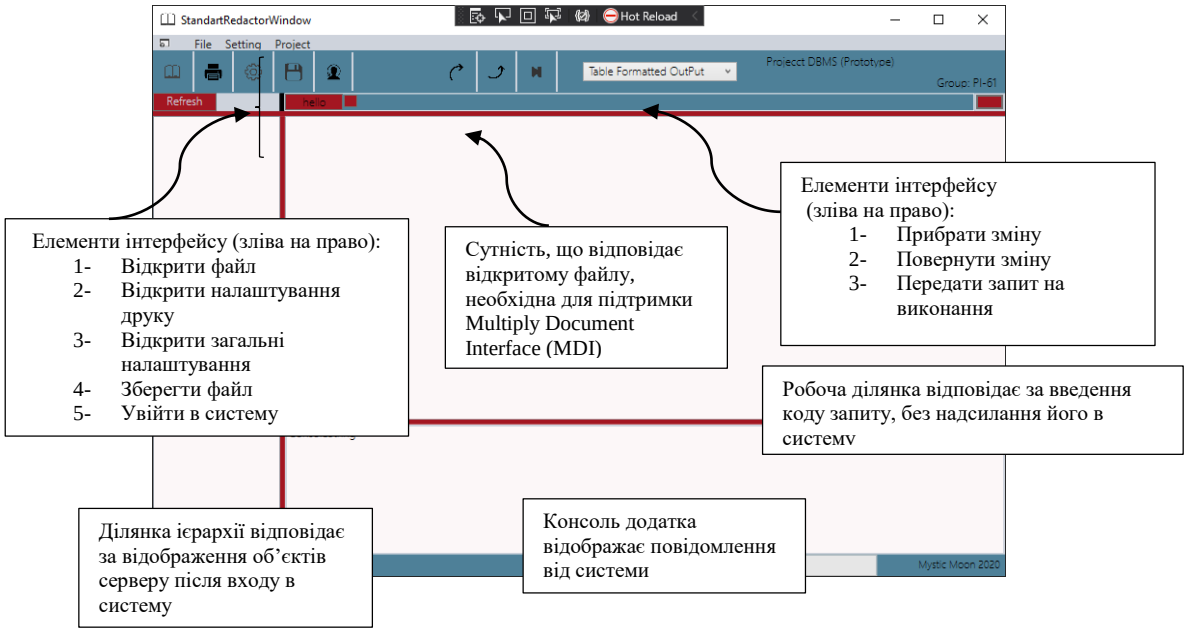

Рис. 5. Інтерфейс користувача СКБД

Для демонстрації роботи системи було створено тестову базу даних BD з декількома колекціями Student, Discipline та проведене тестування виконання операцій на колекції Student. Створення бази даних та її колекцій наведені на рисунку 6.

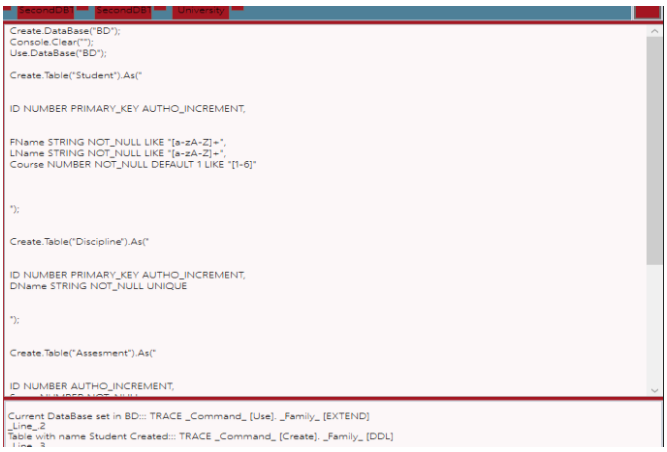

Рис. 6. Створення БД «BD» та колекцій «Student» і «Discipline»

Вставка даних у колекцію продемонстрована на рисунку 7.

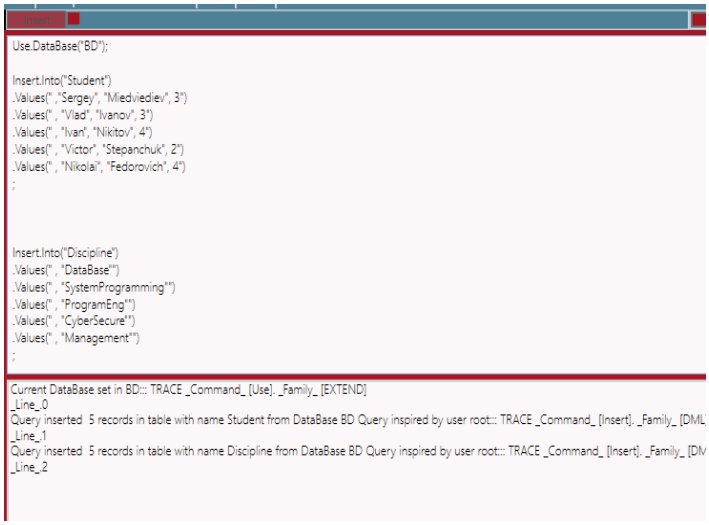

Рис. 7. Вставка даних у колекцію «Student» і «Discipline»

Додатково системою передбачена можливість виведення даних у файл з форматами PDF та Excel. Для операцій, які передбачають поєднання багатьох таблиць, передбачена реалізація команд Union, що поєднує дві колекції в одну, та команда CrossJoin для поєднання декількох колекцій в усіх комбінаціях між собою. З метою поліпшення можливостей до автоматизації в системі передбачено створення представлень, процедур, що зберігаються, та тригерів на команди.

Для адміністрування та налаштування сервера є можливість, використовуючи простір команд Adminer, звертатися до системних представлень та записів налаштувань сервера.

У розроблені СКБД реалізовані агрегатні функції Avg, Count, Sum, Max, Min.

Для отримання інформації про колекції передбачені команди GetDomain, GetConstrain.

У результаті утворена структура здатна виконувати поставлені цілі, реалізуючи мінімально необхідний набір компонентів для роботи з бази даних. Однак поточна реалізація не забезпечує достатньої гнучкості системи типів, яка може знадобитися при розробці складних баз даних або для надання базі даних більшої семантичної виразності, що забезпечується в тому числі через систему типів.

Висновки та перспективи подальших досліджень. Отже, розроблена СКБД призначена для БД з невеликими наборами даних. Утворена система має низку переваг, які роблять її зручною, а саме: невеликий розмір (система займає менше 100 мб), що поліпшує її мобільність; інтерфейс користувача характеризується максимальним спрощенням; невеликі розміри початкового коду; утворене абстрактне ядро; обробка регламентних правил через спеціальні об’єкти; надання переваг чистим функціям, що значно полегшує можливості до супроводження системи й зменшує вартість системи для потенційного користувача. Це дуже важливо для маленьких підприємств та організацій.

Використання зрозумілої організації файлів на базі API операційної системи, можливість автоматичного встановлення сервера при першому старті системи, а також використання загальної мови у вигляді формату JSON значно полегшує експлуатацію та передачу файлів у СКБД.

Відтак розроблена СКБД розглядається як така, що буде використовуватися підприємствами малого та мікробізнесу, яким з одного боку потрібна СКБД з підтримкою необхідного мінімуму механізмів контролю цілісності даних, а з іншого – потрібна велика мобільність та зручність використання.

В подальшому необхідно вирішити питання оптимізації запитів. Система типів може бути сформована з використанням загального формату XMLSchema для забезпечення уніфікованості та прозорості системи типів або з використанням JSON, що забезпечить більш природну інтеграцію до кінцевого формату систем, що використовують мову JavaScript. Інформація про зв’язки таблиць бази даних може зберігатись у спеціалізованій структурі-графі для пришвидшення аналізу даних. З метою прискорення операцій DML доцільно ввести систему індексів.

Список використаної літератури:

1. *Головачов І.А.* Реляційні та нереляційні бази даних в бізнесі / *І.А. Головачов, Н.В. Геселева* // Економіка інноваційної діяльності підприємств. – 2018 [Електронний ресурс]. – Режим доступу : https://er.knutd.edu.ua/bitstream/123456789/11102/1/NRMSE2018_V3_P319-320.pdf.

2. *Мозолюк В.О.* Дослідження проблем ідентифікації об'єктів в базах даних / *В.О. Мозолюк, В.М. Джулій* // Інтелектуальний потенціал – 2020 : збірник наукових праць молодих науковців і студ. – Хмельницький : ПВНЗ УЕП, 2020. – Ч. 2. – С. 65–69 [Електронний ресурс]. – Режим доступу : http://elar.khnu.km.ua/jspui/handle/123456789/9641.
3. *Жолубак Л.І.* Правила Кодда в базах даних / *Л.І. Жолубак, Н.Є. Буряк* // Інформаційна безпека та Інформаційні технології : збірник тез доповідей IV Всеукраїнської науково-практичної конференція молодих учених, студентів і курсантів (27 листопада 2020 р.). – Львів : ЛДУ БЖД, 2020. – С. 190–192 [Електронний ресурс]. – Режим доступу : https://sci.ldubgd.edu.ua/bitstream/123456789/7447/1/Zholubak_Burak.pdf.
4. *Зінов'єва І.С.* Сучасні підходи до подальшої еволюції концепції баз даних / *І.С. Зінов'єва, В.О. Артемчук* // Dynamics of the development of world science : abstracts of the 3d International scientific and practical conference. – Vancouver, Canada : Perfect Publishing, 2019 [Електронний ресурс]. – Режим доступу : https://www.researchgate.net/publication/337345210_SUCASNI_PIDHODI_DO_PODALSOI_EVOLUCII_KONCEPCII_BAZ_DANIH.
5. *Мамалига Н.Є.* Система керування базами даних в сучасних умовах ІТ-індустрії / *Н.Є. Мамалига, А.І. Катаєва* ; ред. кол. *Хаджинов І.В.* (гол.) та ін. // Вісник студентського наукового товариства ДонНУ ім. Василя Стуса. – Вінниця : ДонНУ ім. Василя Стуса, 2021. – Вип. 13, Т. 1. – 275 с.
6. *Романюк О.* Розробка моделі розмежування доступу до функцій в сучасних СКБД / *О.Романюк, І.Микитюк.* – 2019 [Електронний ресурс]. – Режим доступу : http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/31263/19_233-238.pdf?sequence=&isAllowed =y.
7. Створення та обробка баз даних : навч. посіб. для студ. техн. спец. ВНЗ / *Л.С. Глоба, М.Ю. Тернова, Р.Л. Новогрудська, О.С. Штогрина*. – Київ : НТУ України «КПІ», 2013. – 477 с.
8. *Шаров С.В.* Бази даних та інформаційні системи : навч. посіб. / *С.В. Шаров, В.В. Осадчий*. – Мелітополь : МДПУ ім. Б.Хмельницького, 2014. – 352 с.
9. *Голден Кришна* Хороший интерфейс – невидимый интерфейс / *Кришна Голден*. – СПб. : Питер, 2016. – 256 с.
10. *Конноли Томас* Базы данных. Проектирование, реализация и сопровождение. Теория и практика / *Томас Конноли, Каролы Бегг*. – 3-е издание ; пер. с англ. – М. : Вильямс, 2003. – 1440 с.
11. *Райордам Р.* Основы реляционных баз данных : пер. с англ. / *Р.Райордам*. – М. : Русская Редакция, 2001. – 384 с.
12. *Уэйншек Сьюзен* 100 новых главных принципов дизайна / *Сьюзен Уэйншек*. – СПб. : Питер, 2016. – 290 с.
13. *Фаулер Мартин* NoSQL: Новая методология разработки нереляционных баз даних : пер с англ. / *Мартин Фаулер, Дж. Садалаж Прамодкумар*. – М. : Вильямс, 2013. – 192 с.

References:

1. Golovachov, I.A. and Geseleva, N.V. (2018), «Reljacijni ta nereljacijni bazy danyh v biznesi», *Ekonomika innovacijnoi' dijal'nosti pidpryjemstv*, [Online], available at: https://er.knutd.edu.ua/bitstream/123456789/11102/1/NRMSE2018_V3_P319-320.pdf
2. Mozoljuk, V.O. and Dzhulij, V.M. (2020), «Doslidzhennja problem identyfikacii' ob'jektiv v bazah danyh», *Intelektual'nyj potencial – 2020*, zbirnyk naukovyh prac' molodyh naukovciv i stud., PVNZ UEP, Hmel'nyc'kyj, Part 2, pp. 65–69, [Online], available at: http://elar.khnu.km.ua/jspui/handle/123456789/9641
3. Zholubak, L.I. and Burjak, N.Je. (2020), «Pravyla Kodda v bazah danyh», *Informacijna bezpeka ta Informacijni tehnologii'*, zbirnyk tez dopovidej, IV Vseukrai'ns'ka naukovo-praktychna konferencija molodyh uchenyh, studentiv i kursantiv, 27 lystopada 2020 r., LDU BZhD, L'viv, pp. 190–192, [Online], available at: https://sci.ldubgd.edu.ua/bitstream/123456789/7447/1/Zholubak_Burak.pdf
4. Zinov'jeva, I.S. and Artemchuk, V.O. (2019), «Suchasni pidhody do podal'shoi' evoljucii' koncepcii' baz danyh», *Dynamics of the development of world science*, abstracts of the 3d International scientific and practical conference, Perfect Publishing, Canada, Vancouver, [Online], available at: https://www.researchgate.net/publication/337345210_SUCASNI_PIDHODI_DO_PODALSOI_EVOLUCII_KONCEPCII_BAZ_DANIH
5. Mamalyga, N.Je. and Katajeva, A.I. (2021), «Systema keruvannja bazamy danyh v suchasnyh umovah IT-industrii'», in Hadzhynov, I.V. (ed.) et al., *Visnyk students'kogo naukovogo tovarystva DonNU im. Vasylja Stusa*, DonNU im. Vasylja Stusa, Vinnycja, Issue 13, Vol. 1, 275 p.
6. Romanjuk, O. and Mykytjuk, I. (2019), «Rozrobka modeli rozmezhuvannja dostupu do funkcij v suchasnyh SKBD», [Online], available at: http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/31263/19_233-238.pdf?sequence=&isAllowed =y
7. Globa, L.S., Ternova, M.Ju., Novogruds'ka, R.L. and Shtogryna, O.S. (2013), *Stvorennja ta obrobka baz danyh*, navch. posib. dlja stud. tehn. spec. VNZ, NTU Ukrai'ny «KPI», Kyi'v, 477 p.
8. Sharov, S.V. and Osadchyj, V.V. (2014), *Bazy danyh ta informacijni systemy*, navch. posib., MDPU im. Bogdana Hmel'nyc'kogo, Melitopol', 352 p.
9. Golden, Krishna (2016), *Khoroshii interfeis – nevidimyi interfeis*, Piter, SPb, 256 p.
10. Konnoli, Tomas and Karoli, Begg (2003), *Bazy dannykh. Proektirovanie, realizatsiya i soprovozhdenie. Teoriya i praktika*, third ed., translated from Eng.,Vil'yams, M., 1440 p.
11. Raiordam, R. (2001), *Osnovy relyatsionnykh baz dannykh*, translated from Eng., Russkaya Redaktsiya, M., 384 p.
12. Ueinshek, S'yuzen (2016), *100 novykh glavnykh printsipov dizaina*, Piter, SPb., 290 p.
13. Fauler, Martin and Pramodkumar, Dzh. Sadalazh (2013), *NoSQL: Novaya metodologiya razrabotki nerelyatsionnykh baz danikh*, translated from Eng., Vil'yams, M., 192 p.

Коротун Ольга Володимирівна – кандидат педагогічних наук, доцент кафедри комп'ютерних наук Державного університету «Житомирська політехніка».

https://orcid.org/ 0000-0003-2240-7891.

Наукові інтереси:

- бази даних;
- системи керування базами даних;
- чисельні методи;
- хмарні технології.

Марчук Галина Вікторівна – старший викладач кафедри комп'ютерних наук Державного університету «Житомирська політехніка».

https://orcid.org/0000-0003-2954-1057.

Наукові інтереси:

- інтелектуальний аналіз даних;
- комп'ютерна дискретна математика.

Мєдвєдєв Віталій Володимирович – студент третього курсу факультету інформаційно-комп'ютерних технологій Державного університету «Житомирська політехніка», спеціальність «Інженерія програмного забезпечення».

Наукові інтереси:

- дослідження функціонування та взаємодії компонентів у багатокритеріальних системах.

Стаття надійшла до редакції 04.08.2021.