УДК 004.658: 652.3

[0000-0001-5913-6859] **М. Л. Дворецький,** *ст. викладач,*
e-mail: m.dvoretskiy@gmail.com
С. В. Дворецька, *ст. викладач,*
e-mail: svetag603@gmail.com
[0000-0002-0547-3689] **Є. О. Давиденко**, *к.т.н.*
e-mail: davydenko@chmnu.edu.ua
Чорноморський національний університет імені Петра Могили
вул. 68 Десантників, 10, м. Миколаїв, 54003, Україна

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ВИЗНАЧЕННЯ КОРИСНИХ ДАНИХ ПРИ ОПТИМІЗАЦІЇ СТРУКТУРИ ТА МІНІМІЗАЦІЇ ОБСЯГІВ ВУЗЛА РОЗПОДІЛЕНОЇ БД

Дослідження ставить на меті підвищення рівня загальної доступності даних в окремому вузлі розподіленої бази даних та ефективності використання програмних систем по роботі з даними за рахунок зменшення кількості розподілених запитів. Мета досягається шляхом оптимізації структури вузла розподіленої бази даних та мінімізації обсягів даних, що зберігаються у ньому.

Для досягнення мети було створено підсистему обліку користувацьких запитів, граматику мови T-SQL та виконано парсинг коду SQL-запитів. В результаті запити класифікуються за списком таблиць бази даних, які трапляються у запиті, а також, після виконання більш детального аналізу, за списком атрибутів та кортежів відношення. Останнє досягається за рахунок виконання набору запитів зі зверненням до первинного ключа кожного відношення, що входить до складу запиту.

Виконання повного аналізу оцінки корисності атрибутів та кортежів таблиць бази даних є досить ресурсоємною операцією, тому не може виконуватися при кожній зміні даних. У рамках дослідження запропоновано реалізувати вирішення задачі класифікації нових даних із використанням нейронної мережі прямого поширення, що навчається на базі оцінених попередньо даних на базі парсингу SQL-запитів. Враховуючи необхідність виконання аналізу накопичених даних з точки зору множинності вимірів, а також, ймовірно, великі їх обсяги, було виконано представлення даних, необхідних для аналізу, у вигляді багатовимірної моделі.

Ключові слова: *розподілена транзакція, система керування базами даних, розподілена база даних, розподілений SQL-запит, реплікація даних, парсинг тексту, дерево парсингу, профайлінг, ANTLR, OLAP, задача класифікації, нейронна мережа, інтелектуальний аналіз даних.*

Вступ. Створюючи базу даних, користувач прагне впорядкувати інформацію за різними ознаками для швидкого отримання потрібних відомостей з довільним сполученням критеріїв пошуку. Необхідність автоматизації різних типів обліку (складський, бухгалтерський, облік кадрів тощо) у межах підприємства в деяких випадках веде до використання «універсальних» облікових систем. Цей підхід має ряд недоліків, серед яких перевантаженість центральної бази даних (БД), низька відмовостійкість, вразливість системи та недостатньо розвинені механізми обліку більшості напрямів автоматизації [1-2].

Аналіз публікацій та останніх досягнень. Використання окремих спеціалізованих рішень може вирішити наведені недоліки [3-4], але веде до появи різних платформ систем керування базами даних (СКБД), що потребують подальшої синхронізації. Тому, у розвитку сучасних інформаційних систем простежується тенденція переходу від локальних БД до розподілених. Основною задачею розподіленої СКБД є забезпечення управління доступом до даних багатьох споживачів, цілісності й узгодженості даних в умовах використання мережі. Тобто основна функція таких СКБД – це координування спільної роботи багатьох користувачів з розподіленою інформацією [5-7].

Розв’язання проблеми автономності роботи користувачів розподіленої системи створює багато специфічних проблем в організації даних. Комбінована стратегія розподілу даних

DOI: 10.24025/2306-4412.4.2019.184808

поєднує два підходи, пов'язані з розподілом без дублювання та з дублюванням даних, з метою використання їх переваг [8]. Але при її використанні, окрім задачі синхронізації дубльованої інформації, актуальною постає задача оптимального проектування структури БД з точки зору належності даних до категорії того чи іншого вузла розподіленої БД. Крім того, продуктивність системи напряму буде залежати від прийняття рішення щодо необхідності часткового або повного дублювання даних.

Огляд інформаційних систем при вирішенні задач автоматизації різних типів обліку у межах однієї організації та недоліків використання універсальних облікових систем [1-4] дав можливість обґрунтувати необхідність оптимізації структури віддалених вузлів розподіленої бази даних (РБД) шляхом визначення корисних даних та мінімізації обсягів її вузла [8].

Метою роботи є підвищення рівня загальної доступності даних в окремому вузлі РБД та ефективності використання програмних систем по роботі з даними БД за рахунок зменшення кількості розподілених запитів. Мета досягається шляхом оптимізації структури вузла РБД та мінімізації обсягів даних, що зберігаються у ньому.

Під час виконання дослідження для досягнення сформульованої мети було вирішено наступні задачі: виконано дослідження методів створення та використання формальних граматик та розроблено підсистему парсингу коду SQL-запитів до центральної БД; розроблено модель та реалізовано підсистему обліку користувацької активності на основі профайлінгу SQL-запитів за різними аналітичними характеристиками; виконано деталізацію звернень користувацьких запитів до відношень БД на рівні множин атрибутів та кортежів; розроблено інформаційну технологію підтримки прийняття рішень при проектуванні структури вузла розподіленої БД та визначенні кількості необхідних даних.

Виклад основного матеріалу. При використанні комбінованого підходу представлення даних в РБД вибір типу реплікації залежить від декількох факторів, зокрема від того, чи є тип запиту віддаленим або розподіленим. У випадку віддаленого запиту, коли взаємодія з іншими вузлами необхідна лише для забезпечення реплікації даних, вважається за доцільне використання асинхронної реплікації [9-10]. Подібне рішення обґрунтовано можливістю виключити із транзакції всі «зайві» вузли, залишивши лише один, на якому, фактично, знаходяться дані (рисунок 1).

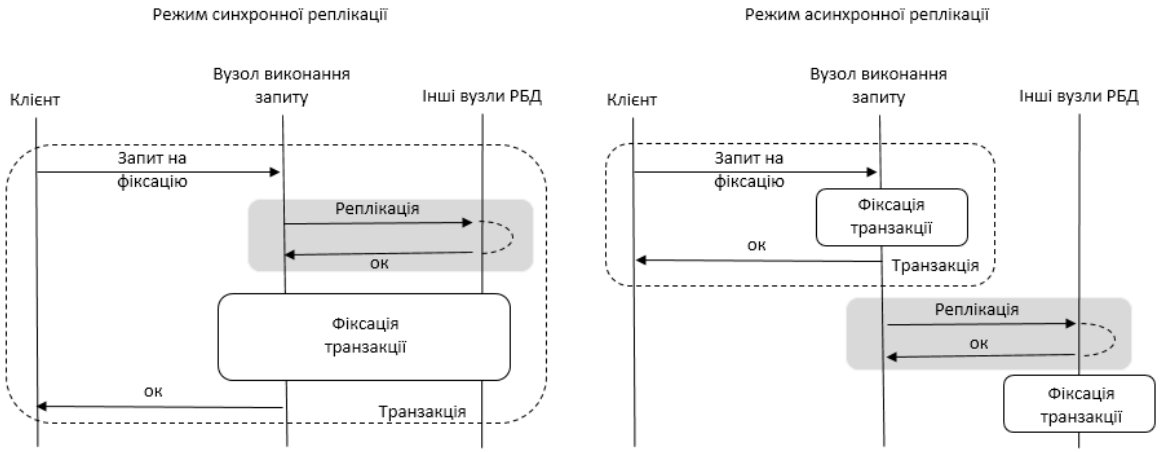

Рисунок 1 – Порівняння режимів реплікації у випадку віддаленого запиту

При синхронізації даних у випадку розподіленого запиту взаємодія із віддаленими вузлами не може бути винесена за межі транзакції. Це обумовлено тим фактом, що до розподіленої транзакції у будь-якому випадку входять фрагменти даних, що зберігаються на інших вузлах та не представлені на вузлі виконання запиту. Зважаючи на цей факт, не вбачається за доцільне винесення процесу реплікації за межі транзакції [9, 11], оскільки зменшення часу виконання транзакції є незначним порівняно з можливими суперечностями в даних, пов'язаними із несинхронністю оновлення (рисунок 2).

DOI: 10.24025/2306-4412.4.2019.184808

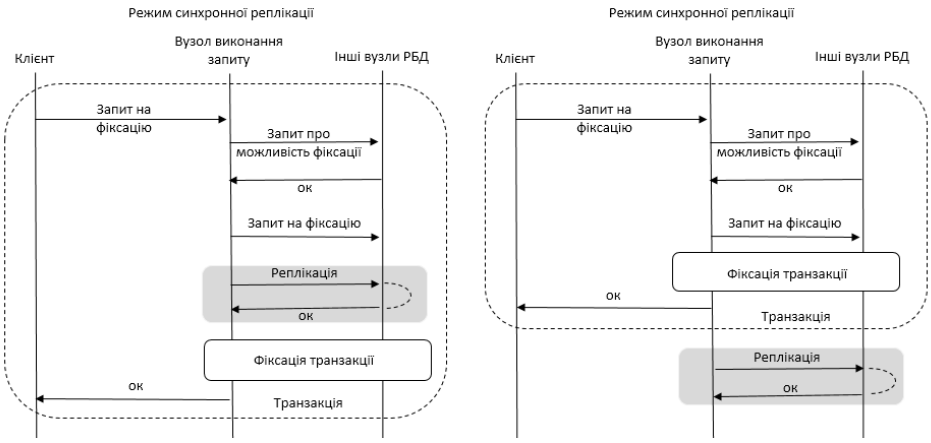

Рисунок 2 – Порівняння режимів реплікації у випадку розподіленого запиту

Виходячи з наведеного, при проектуванні віддаленого вузла розподіленої БД головним завданням на шляху до підвищення рівня загальної доступності та ефективності використання програмних систем по роботі з даними БД є зменшення кількості розподілених запитів, у яких задіяні дані декількох вузлів БД, та заміна їх локальними [12-13].

Для мінімізації кількості розподілених транзакцій [14] у межах віддаленого вузла РБД було створено підсистему обліку користувацьких запитів із класифікацією відповідно до належності до того чи іншого автоматизованого робочого місця, географічного розташування, ролі користувача та інших критеріїв, що можливо додати до системи динамічно під час її використання згідно з особливостями тієї чи іншої предметної області. Даталогічну архітектуру розробленої моделі наведено на рисунку 3.

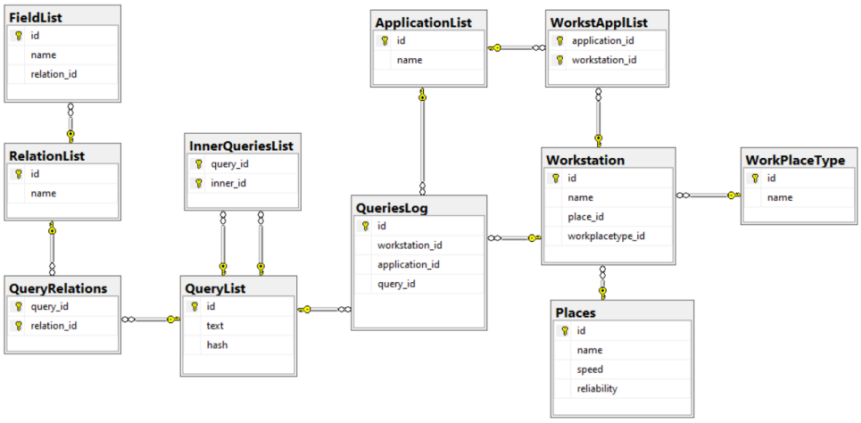

Рисунок 3 – Даталогічна модель підсистеми профайлінгу користувацьких SQL-запитів

Центральне місце займають журнал та довідник користувацьких запитів (таблиці QueriesLog та QueryList). Якщо запит є вкладеним або має декілька рівнів, його деревоподібна структура описується за допомогою таблиці InnerQueriesList. Запити класифікуються за типом програмного забезпечення, робочої станції, користувача та місця розташування (прив'язка до майбутніх вузлів РБД), з якого вони надходять (таблиці Workstation, Places, ApplicatioList, WorksAppList та WorkPlaceType). Виходячи із наведеної структури, також бачимо, що введення до системи додаткових аналітичних характеристик, що можуть знадобитися залежно від тієї чи іншої предметної області, не потребуватиме великих змін у БД. Після виконання парсингу коду SQL-запитів (парсинг – процес перетворення вихідного коду в структурований вигляд), вони також розбиваються за списком таблиць БД, які трапляються у запиті, а також, після виконання більш глибинного аналізу, за списком атрибутів та кортежів відношення (таблиці QueryRelations, RelationList та FieldList).

Наповнення розробленої моделі вхідними даними реалізовано за допомогою меха-

DOI: 10.24025/2306-4412.4.2019.184808

нізмів профайлінгу СКБД. В рамках дослідження було обрано програмний продукт SQL Profiler, що входить до стандартного інсталяційного пакета SQL Server. Цей вибір обумовлено достатнім переліком аналітичних властивостей, що надає функціонал програмного забезпечення (ПЗ), і сумісністю із версією СКБД, що використовується на підприємстві – базі автоматизації. При зміні вхідних умов вибір може бути змінено на користь іншого ПЗ.

Підсистема має користувацький інтерфейс, що складається з форм вводу та редагування даних для основних сутностей. Передбачено механізми імпорту даних із текстових та .csv файлів для можливості взаємодії зі сторонніми програмними продуктами (наприклад при вирішенні задач отримання списків користувачів, програмного забезпечення чи робочих станцій). Деякі сутності також можуть бути наповнені даними на основі результатів статистичної вибірки користувацьких SQL-запитів.

Для виконання подальшого аналітичного аналізу накопичених статистичних даних необхідне виділення конкретних відношень (а також посилань на їх атрибути та кортежі) із тексту користувацьких запитів. Для вирішення цієї задачі використано засоби розбору та лінгвістичного аналізу тексту. SQL, як кожна мова програмування, має правила, які визначають синтаксичну структуру коректних програм. Синтаксис конструкцій мови програмування може бути описаний за допомогою контекстно-вільних граматик або нотації БНФ (Backus-Naur Form, форм Бекуса-Наура).

Використаний у рамках дослідження генератор парсерів ANTLR є LL(*), він існує вже більше 20 років, а в 2013 р. вийшла його 4-а версія. Нині його розробка ведеться на GitHub. В даний момент він дає змогу генерувати парсери мовами Java, C#, Python2, Python3, JavaScript. Початковим етапом реалізації підсистеми парсингу T-SQL запитів є створення граматики. Далі, згенеровані класи лексеру та парсеру дають можливість представити код запиту у вигляді деревоподібної моделі, на базі якої відбувається визначення списку відношень та атрибутів, які були використані у запиті. На рисунку 4 зображено дерево ієрархії елементів для запиту «select (select number from groups where id=students.id), name, age from students», що звертається до декількох атрибутів і має у своєму складі один вкладений запит.

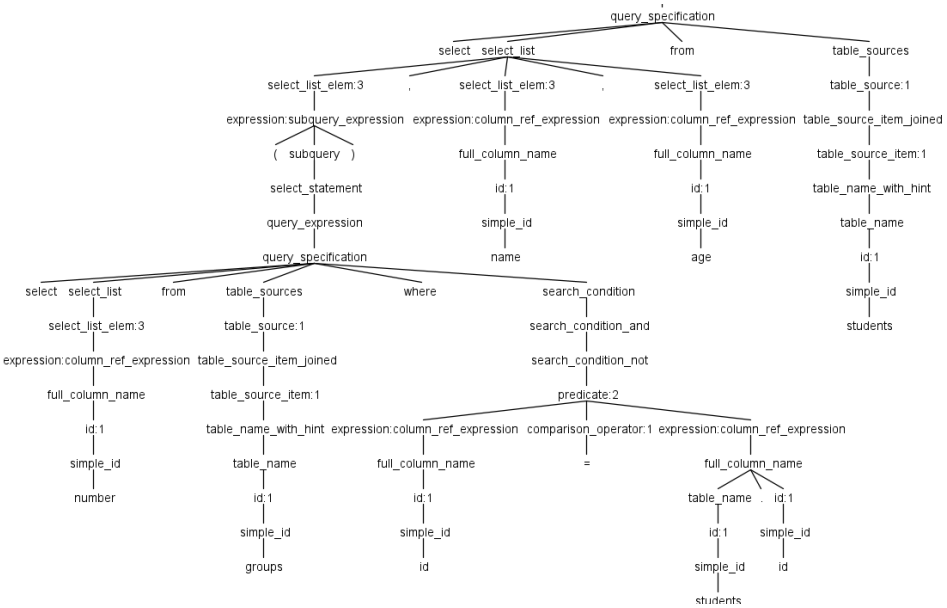

Рисунок 4 – Дерево парсингу запиту T-SQL

Далі кожний запит представляється у вигляді об’єкта, що має такі атрибути: робоча станція, користувач та ПЗ, від яких надійшов запит; сам текст запиту; колекція таблиць, до яких звертається запит; та колекція вкладених запитів, якщо такі мають місце. Сутність «таблиця» (TsqlRelation) у цьому випадку є підмножиною таблиці БД і містить назву таблиці, колекцію атрибутів, які задіяні у запиті, та колекцію значень первинного ключа таблиці згідно з множиною кортежів, що повертаються як результат роботи запиту [15, 16]. Фрагмент діаграми класів інформаційної технології парсеру SQL-запитів зображено на рисунку 5.

DOI: 10.24025/2306-4412.4.2019.184808

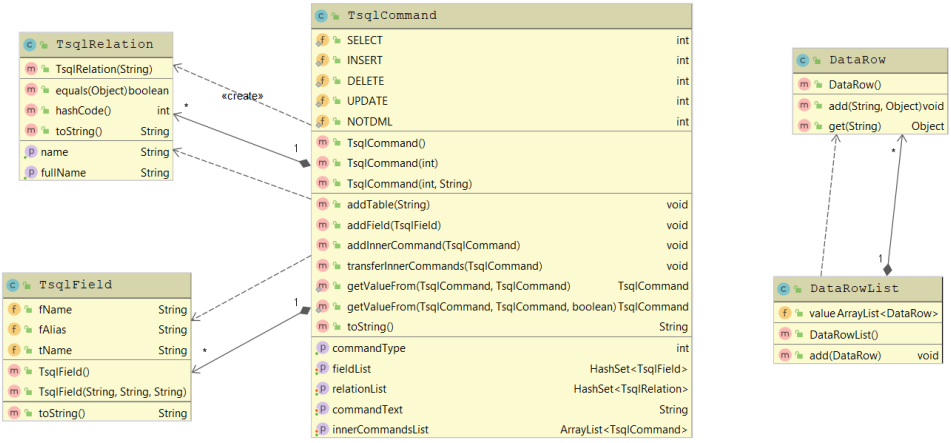

Рисунок 5 – Класи, що відповідають за представлення запитів у моделі

Отже, на цьому етапі ми маємо множину відношень із атрибутами, що використовуються у запиті. Цього достатньо для визначення списку таблиць, представлених у вузлі РБД, та для обмеження загальної кількості даних через виконання операції проекції до них. Однак цього може бути замало, оскільки зазвичай у БД облікових систем існують таблиці з великою кількістю кортежів, що займають багато місця на диску та створюють додаткове навантаження при виконанні запитів до них. При цьому, у тому чи іншому вузлі РБД реально використовується досить невеликий відсоток від загального обсягу даних, що входять до них.

Наступний крок – визначення множини кортежів, що використовуються у тому чи іншому запиті. Для цього доповнюємо клас TsqlRelation колекцією типу TsqlField, що визначатиме первинний ключ відношення. У випадку, якщо первинний ключ у відношенні відсутній або він складається з великої кількості атрибутів нечислового типу даних, і, як наслідок, має великий об'єм, на рівні БД може бути створене унікальне автоінкрементне поле *not null*, що буде використовуватися надалі для ідентифікації кортежу.

Далі, якщо запит має у своєму складі вкладені запити, кожен із них оброблюється окремо. Для кожної таблиці з колекції відношень запиту виконується додатковий запит до БД, список атрибутів у якому замінюється на первинний ключ відношення. Результат запиту запам'ятовується як множина атрибутів таблиці, що використовується поточним запитом. Так, наприклад, у випадку надходження запиту, що вибирає прізвища студентів із номерами студентських груп із двох таблиць із використанням вкладеного запиту для фільтрації за спеціальністю (рисунок 6), він буде розбитий на наступну множину запитів (рисунок 7).

```
select s.name, g.number
from students s inner join groups g on s.gr_code = g.code
where g.special_code in(
    select code from specialties where name = 'computer science'
)
```

Рисунок 6 – Приклад вхідного запиту

```
select code from specialties where name = 'computer science';

select distinct s.code from students s inner join groups g on s.gr_code = g.code
where g.special_code in(
    select code from specialties where name = 'computer science'
);

select distinct g.code from students s inner join groups g on s.gr_code = g.code
where g.special_code in(
    select code from specialties where name = 'computer science'
);
```

Рисунок 7 – Список запитів для отримання множин атрибутів відношень, що використовуються вхідним запитом

DOI: 10.24025/2306-4412.4.2019.184808

Після виявлення множин атрибутів з'являється можливість визначення корисної для вузла РБД підмножини даних із використанням фільтрації таблиць за значенням первинного ключа, що дає можливість розмістити переважну більшість необхідних вузлу даних локально та, відповідно, знизити кількість розподілених запитів. При цьому об'єм даних локального вузла також є мінімальним, що, у свою чергу, мінімізує потребу у синхронізації різних версій одних і тих же даних.

Однак у процесі експлуатації описаної технології виникає проблема, пов'язана із модифікацією даних БД: коли до таблиць додаються нові дані, існуючі модифікуються або видаляються. У цьому випадку найпростішим варіантом рішення проблеми є реплікація до віддаленого вузла БД усіх нових або змінених даних (оскільки ми не знаємо про ступінь їх корисності для вузла). Коли ж обсяг цих даних досягає якоїсь критичної точки, виходячи, наприклад, із міркувань навантаження при вибірці даних або їх синхронізації, необхідно заново провести аналіз SQL-запитів та виконати оновлення бази використання атрибутів та кортежів таблиць БД віддаленим вузлом.

Слід зауважити, що виконання повного аналізу використовуваності атрибутів та кортежів таблиць БД є досить ресурсоємною операцією та не може виконуватись часто, тому вищенаведений підхід є неприйнятним для великих та часто змінюваних БД. Тому було запропоновано представити проблему із вирішенням корисності нових або змінених даних у вигляді задачі класифікації інтелектуального аналізу даних. На вході у нас є назва таблиці та перелік значень її атрибутів (новий або змінений рядок), а на виході – рішення про його корисність для віддаленого вузла БД. Для вирішення цієї задачі запропоновано використати нейронну мережу прямого поширення [17] із одним прихованим шаром. Навчання мережі відбувається на базі даних, що були отримані в результаті аналізу результатів парсингу SQL-запитів. Далі, після навчання, мережа виконує класифікацію нових та змінених даних.

Враховуючи необхідність виконання аналізу накопичених даних з точки зору множинності вимірів, а також, ймовірно, великі їх обсяги, було виконано представлення даних, необхідних для аналізу, у вигляді багатовимірної моделі [18]. Для цього на рівні РБД інформаційної системи обліку користувацьких запитів вводиться ряд представлень, що реалізує таблицю фактів і таблиці вимірів у вигляді схеми «зірка». Структуру отриманого багатовимірного кубу зображено на рисунку 8.

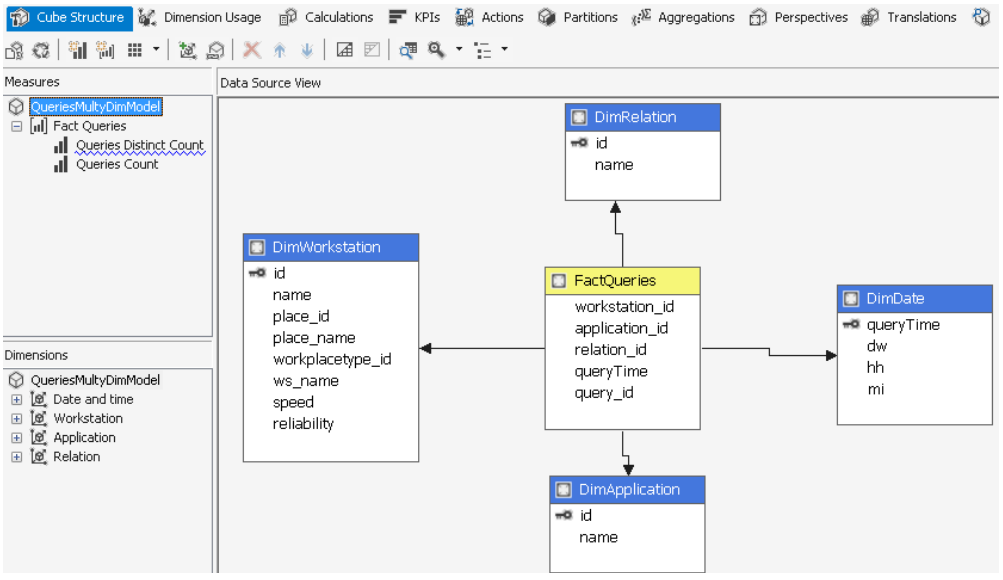

Рисунок 8 – Структура багатовимірного кубу підсистеми оперативно-аналітичного аналізу запитів

Висновки. У ході дослідження вперше запропоновано використання багаторівневих довідників аналітичних характеристик користувацьких SQL-запитів із оцінкою ступеня актуальності даних; розроблено підсистему парсингу SQL-запитів, що дає можливість представити користувацький запит у вигляді множин відношень із деталізацією до викорис-

DOI: 10.24025/2306-4412.4.2019.184808

таних атрибутів та кортежів. Також вперше при аналізі користувацьких SQL-запитів використано багатовимірну модель даних і на її базі реалізовано систему підтримки прийняття рішень при проектуванні структури віддаленого автоматизованого робочого місця, що дає змогу проводити оперативно-аналітичний аналіз SQL-запитів по вимірах, швидко та ефективно виконувати операції консолідації, деталізації, обертання та зрізу. При модифікації даних центральної БД запропоновано використати нейронну мережу для вирішення задачі класифікації нових або змінених даних відповідно до ступеня їх корисності для віддаленого вузла РБД.

Результати дослідження були виробувані при розробці структури БД автоматизованого робочого місця касира супермаркету та при реалізації подальшої синхронізації із центральною БД. Результат виконаного порівняльного аналізу при використанні централізованої та РБД наведено у таблиці 1.

Таблиця 1 – **Порівняльний аналіз використання централізованого та розподіленого підходів до реалізації структури БД**

	Централізована БД	БД незалежного АРМ	Різниця
Об'єм БД, мб	29 025,63	174,88	165 разів
Кількість таблиць	460	34	14 разів
Час резервного копіювання БД, хв	7:58	0:13	37 разів
Середня кількість підключень до БД	59,75	1,02	58 разів
Середня кількість запитів у хв	817	44	19 разів
Середній час обробки, мс	5665	99	57 разів
Час синхронізації, сек	0	18	необхідність синхронізації даних
Незалежність від центрального сервера БД	ні	так	незалежність від центральної БД
Незалежність від комунікаційного обладнання	ні	так	

Отже, результатом є підвищення швидкості роботи інформаційної системи віддаленого автоматизованого робочого місця та розвантаження центральної БД за рахунок оптимізації структури РБД та мінімізації розподілених транзакцій і використання синхронного режиму реплікації даних.

Список літератури

[1] Н. Козлюк, и С. Угримова, *Складской учет на предприятиях торговли*. Феникс, 2005.

[2] Торговля, склад и CRM в облаке. [Электронный ресурс]. Режим доступа: https://www.moysklad.ru

[3] М. Л. Дворецький, С. В. Дворецька та С. Ю. Боровльова, "Web-застосунок складського обліку в неавтоматизованих торгових точках", *Наукові праці Чорномор. нац. ун-ту ім. Петра Могили комплексу «Києво-Могилянська академія». Серія: Комп'ютерні технології:* наук.-метод. журн. Миколаїв: Вид-во ЧНУ ім. П. Могили, вип. 308, т. 320, с. 45-52, 2018.

[4] 1С: Предприятие 8. Управление торговым предприятием для Украины. [Электронный ресурс]. Режим доступа: http://rarus.com.ua/torgovyy-i-skladskoy-uchet/1S_Predpriyatie_8_Upravlenie_torgovym_predpriyatiem_dlya_Ukrainy_/

[5] М. Кузнецов, и И. Симдянов, *MySQL 5*. Санкт-Петербург: БХВ-Петербург, 2010.

[6] Д. Петкович, *Microsoft SQL Server 2008: руководство для начинающих* / пер. с англ. Санкт-Петербург: БХВ-Петербург, 2009.

[7] Т. Коннолли, и К. Бегг, *Базы данных. Проектирование, реализация и сопровождение. Теория и практика*, 3-е изд. / пер. с англ. Москва: Вильямс, 2003.

[8] M. Tamer Özsu, and P. Valduriez, *Principles of distributed database systems.* 3rd ed. Springler, 2011.

DOI: 10.24025/2306-4412.4.2019.184808

[9] Автоматическая синхронизация распределенных баз данных в разделенном режиме. [Электронный ресурс]. Режим доступа: http://stimul.kiev.ua/materialy.htm?a=avtomaticheskaya_sinkhronizatsiya_raspredelennykh_baz_dannykh_v_razdelennom_rezh

[10] Д. Д. Ульман, Д. Уидом и Г. Гарсиа-Молина, *Системы баз данных:* полный курс. Москва: Вильямс, 2004.

[11] Управление транзакциями (компонент Database Engine). [Электронный ресурс]. Режим доступа: https://technet.microsoft.com/ru-ru/library/ms175523(v=sql.105).aspx

[12] Ребекка М. Райордан, *Основы реляционных баз данных.* Москва: Русская Редакция, 2001.

[13] М. Л. Дворецький, "Проектування та оцінка оптимальності структури сховища даних та багатовимірної БД", *Наукові праці Чорномор. нац. ун-ту ім. Петра Могили комплексу «Києво-Могилянська академія». Серія: Комп'ютерні технології:* наук.-метод. журн. Миколаїв: Вид-во МДГУ ім. П. Могили, вип. 77, т. 90, с. 52-60, 2008.

[14] Использование синхронных и асинхронных операций базы данных. [Электронный ресурс]. Режим доступа: http://help.adobe.com/ru_RU/as3/dev/WS5b3ccc516d4fbf351e63e3d118666ade46-7d39.html

[15] М. Л. Дворецький, Є. О. Давиденко, та С. Ю. Боровльова, "Проектування структури розподіленої БД на базі парсингу SQL-запитів". *Наукові праці Чорномор. нац. ун-ту ім. Петра Могили комплексу «Києво-Могилянська академія». Серія: Комп'ютерні технології:* наук.-метод. журн. Миколаїв: Вид-во ЧНУ ім. П. Могили, вип. 275, т. 287, с. 53-61, 2016.

[16] M. Fisun, M. Dvoretskyi, A. Shved, and Y. Davydenko, "Query parsing in order to optimize distributed DB structure," in *9th IEEE Int. Conf. Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Bucharest, 2017, pp. 172-178. doi: 10.1109/IDAACS.2017.8095071

[17] What is an artificial neural network? Here's everything you need to know. [Online]. Available: https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/

[18] М. Т. Фісун, М. Л. Дворецький, та А. В. Юхатов, "Порівняльний аналіз методів побудови olap-систем із використанням засобів MS SQL SERVER та ORACLE", *Наукові праці Чорномор. нац. ун-ту ім. Петра Могили комплексу «Києво-Могилянська академія». Серія: Комп'ютерні технології:* наук.-метод. журн. Миколаїв: Вид-во ЧНУ ім. П. Могили, вип. 271, т. 283, с. 36-42, 2016.

References

[1] N. Kozliuk, and S. Ugrimova, *Warehouse accounting in trade enterprises.* Feniks, 2005 [in Russian].

[2] Trading, warehouse and CRM in the cloud. [Online]. Available: https://www.moysklad.ru

[3] M. L. Dvoretskyi, S. V. Dvoretska, and S. Yu. Borovlova, " Web-based warehouse accounting application in non-automated outlets", *Naukovi pratsi Chornomor. nats. un-tu im. Petra Mohyly kompleksu «Kyievo-Mohylianska akademiia». Seriia: Kompiuterni tekhnolohii:* sci.-method. journ. Mykolaiv: Vyd-vo ChNU im. P. Mohyly, iss. 308, vol. 320, pp. 45-52, 2018 [in Ukrainian].

[4] 1C: Enterprise 8. Management of a trade enterprise for Ukraine. [Online]. Available: http://rarus.com.ua/torgovyy-i-skladskoy-uchet/1S_Predpriyatie_8_Upravlenie_torgovym_predpriyatiem_dlya_Ukrainy_

[5] M. Kuznetsov, and I. Symdianov, *MySQL 5.* St. Petersburg: BKhV-Peterburg, 2010 [in Russian].

[6] D. Petkovych, *Microsoft SQL Server 2008: beginner's guide*, transl. from Engl. St. Petersburg: BKhV-Peterburg, 2009 [in Russian].

[7] T. Konnolli, and K. Begg, *Databases. Design, implementation and maintenance. Theory and practice*, 3rd ed., transl. from Engl. Moscow: Wiliams, 2003 [in Russian]

[8] M. Tamer Özsu, and Patrick Valduriez. *Principles of distributed database systems*, 3rd ed. Springler, 2011.

[9] Automatic synchronization of distributed databases in split mode. [Online]. Available: http://stimul.kiev.ua/materialy.htm?a=avtom

DOI: 10.24025/2306-4412.4.2019.184808

ati-cheskaya_sinkhronizatsiya_raspredelennykh_baz_dannykh_v_razdelennom_rezh
[10] D. D. Ulman, D. Uidom, and G. Garsya-Molina. *Database systems:* full course. Moscow: Wiliams, 2004 [in Russian].
[11] Transaction Management (Database Engine component) [Online]. Available: https://technet.microsoft.com/ru-ru/library/ms175523(v=sql.105).aspx
[12] Rebekka M. Raiordan, *Relational database fundamentals.* Moscow: Russkaia Redaktsyia, 2001 [in Russian].
[13] M. L. Dvoretskyi, "Design and evaluation of data warehouse structure and multidimensional database structure", *Naukovi pratsi Chornomor. nats. un-tu im. Petra Mohyly kompleksu «Kyievo-Mohylianska akademiia». Seriia: Kompiuterni tekhnolohii:* sci.-method. journ. Mykolaiv: Vyd-vo MDHU im. P. Mohyly, iss. 77, vol. 90, pp. 52-60, 2008 [in Ukrainian].
[14] The use of synchronous and asynchronous database operations. [Online]. Available: http://help.adobe.com/ru_RU/as3/dev/WS5b3cc c516d4fbf351e63e3d118666ade46-7d39.html
[15] M. L. Dvoretskyi, Ye. O. Davydenko, and S. Yu. Borovlova, "Design of the structure of a distributed database based on SQL query parsing". *Naukovi pratsi Chornomor. nats. un-tu im. Petra Mohyly kompleksu «Kyievo-Mohylianska akademiia». Seriia: Kompiuterni tekhnolohii:* sci.-method. journ. Mykolaiv: Vyd-vo MDHU im. P. Mohyly, iss. 275, vol. 287, pp. 53-61, 2016 [in Ukrainian].
[16] M. Fisun, M. Dvoretskyi, A. Shved, and Y. Davydenko, "Query parsing in order to optimize distributed DB structure," in *9th IEEE Int. Conf. Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Bucharest, 2017, pp. 172-178. doi: 10.1109/IDAACS.2017.8095071
[17] What is an artificial neural network? Here's everything you need to know. [Online]. Available: https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/
[18] M. T. Fisun, M. L. Dvoretskyi, and A. V. Yukhatov, "Comparative analysis of methods for building olap systems using MS SQL SERVER and ORACLE", *Naukovi pratsi Chornomor. nats. un-tu im. Petra Mohyly kompleksu «Kyievo-Mohylianska akademiia». Seriia: Kompiuterni tekhnolohii:* sci.-method. journ. Mykolaiv: Vyd-vo MDHU im. P. Mohyly, iss. 271, vol. 283, pp. 36-42, 2016 [in Ukrainian].

M. L. Dvoretskyi, *senior lecturer,*
e-mail: m.dvoretskiy@gmail.com
S. V. Dvoretskaya, *senior lecturer,*
e-mail: svetag603@gmail.com
Ye. O. Davydenko, *Ph. D. (Eng.)*
e-mail: davydenko@chmnu.edu.ua
Petro Mohyla Black Sea National University
68 Desantnykiv str., 10, Mykolaiv, 54003, Ukraine

INFORMATION TECHNOLOGY FOR DETERMINING USEFUL DATA WHILE OPTIMIZING THE STRUCTURE AND MINIMIZING THE VOLUME OF THE DISTRIBUTED DATABASE NODE

The paper deals with the tendency to move from "universal" accounting systems to specialized solutions usage. This requires the synchronization of distributed database data. It is noted that among the strategies of data distribution between distributed database nodes, the combined one is the most justified, but the main disadvantage consists in the existence of distributed transactions when handling data.

The research aims to improve the general availability of data in the separate node of the distributed database and the efficiency of using software systems to work with database data by reducing

DOI: 10.24025/2306-4412.4.2019.184808

the number of distributed requests. The goal is achieved by optimizing the structure of the distributed database node and minimizing the amount of data stored in it.

To achieve the goal, users' query accounting subsystem and T-SQL grammar have been created, and SQL query code has been parsed. As a result, the queries are classified by the list of database tables that are found in the query, and, after performing more deyailed analysis, by the list of attributes and relation tuples. The last one is achieved by executing a set of queries with getting the primary key of each relation included in the query.

Performing the complete analysis of the database tables attributes and tuples estimation is a very resource-intensive operation, so it cannot be performed every time the database data is changed. The research proposes to solve the problem of classification of new data by using the perceptron, which learns on the basis of pre-evaluated data based on SQL query parsing. Also, according to the need of performing the analysis of received data from the point of view of multiple dimensions, as well as probably their large amount, the data required for the analysis has been presented in the form of a multidimensional model.

Keywords: *distributed transaction, database management system, distributed database, distributed SQL-query, data replication, text parsing, parse tree, profiling, ANTLR, multidimensional analysis, classification task, neural network, data mining.*

Стаття надійшла 13.11.2019
Прийнято 06.12.2019

DOI: 10.24025/2306-4412.4.2019.184808