

<https://doi.org/10.31891/2307-5732-2026-361-59>

УДК 004.91

ОРЛОВСЬКИЙ СЕРГІЙ

Київський національний університет технологій та дизайну

<https://orcid.org/0009-0008-3037-7999>

e-mail: 0734666127o@gmail.com

СТАЦЕНКО ДМИТРО

Київський національний університет технологій та дизайну

<https://orcid.org/0000-0002-3064-3109>

e-mail: statsenko.dv@knutd.edu.ua

БАЗИ ДАНИХ ДЛЯ СИСТЕМИ ВЕДЕННЯ ОБЛІКУ МАЙНА В ЗСУ

Стаття присвячена розробленню та аналізу бази даних для системи ведення обліку військового майна у Збройних Силах України. Запропоновано реляційну модель, що охоплює процеси зберігання, переміщення, інвентаризації та контролю матеріальної відповідальності. Проведено моделювання таблиць, створення SQL-запитів, побудову індексів та експериментальне вимірювання швидкості їх виконання. Результати показали, що оптимальна індексація скорочує час обробки даних у 11–16 разів. Запропоновані рішення можуть бути використані для автоматизації облікових процесів і підвищення ефективності інформаційних систем оборонного призначення.

Ключові слова: база даних, військове майно, індексація, SQL-запити, продуктивність, облік, Збройні Сили України.

ORLOVSKYI SERHII, STATSENKO DMYTRO

Kyiv National University of Technologies and Design

DATABASES FOR THE PROPERTY ACCOUNTING SYSTEM IN THE ARMED FORCES OF UKRAINE

The article is devoted to the development and optimization of a database for an automated system of military property accounting in the Armed Forces of Ukraine. The relevance of the research is determined by the need to increase the efficiency of managing material and technical resources under wartime conditions, when data accuracy, operational speed, and information security are of strategic importance. A relational model consisting of six main entities (Category, Property, Storage, Movement, Inventory, Responsible_Personnel) was developed to ensure data integrity and consistency. Examples of SQL queries are provided to demonstrate data selection, aggregation, analytical operations, and accountability control. Particular attention is paid to the impact of indexing on query execution speed in the PostgreSQL DBMS. An experimental study was conducted to measure the average execution time of queries before and after the creation of indexes. It was found that the use of B-tree and composite indexes accelerates the execution of WHERE, JOIN, and ORDER BY queries by 11–16 times. The influence of data volume, denormalization, and statistics updates on system performance was analyzed. Recommendations are proposed regarding table structuring, transactional integrity maintenance, backup strategies, and the application of optimized indexes in distributed architectures. The obtained results have practical significance for developing reliable military information systems capable of fast data processing, transparent resource tracking, and robust information protection under cyberthreat conditions.

Keywords: database, military property, indexing, SQL queries, performance, accounting, Armed Forces of Ukraine.

Стаття надійшла до редакції / Received 03.12.2025

Прийнята до друку / Accepted 11.01.2026

Опубліковано / Published 29.01.2026



This is an Open Access article distributed under the terms of the [Creative Commons CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

© Орловський Сергій, Стаценко Дмитро

Актуальність проблеми дослідження.

У сучасних умовах ведення оборонних операцій ефективно управління матеріально-технічними ресурсами набуває стратегічного значення. Для Збройних Сил України, які функціонують у складному середовищі воєнного часу, ключовим завданням стає забезпечення оперативного, достовірного та безпечного обліку військового майна. Традиційні методи ведення обліку, засновані на статичних реєстрах або фрагментарних інформаційних системах, не відповідають вимогам сучасного рівня мобільності, масштабованості й надійності даних.

Проблема полягає у відсутності єдиної оптимізованої моделі бази даних, здатної забезпечити швидке опрацювання великого обсягу записів, підтримку багаторівневого доступу користувачів та інтеграцію з іншими військово-логістичними підсистемами. Збільшення кількості транзакцій, складність зв'язків між об'єктами обліку й необхідність одночасної роботи численних користувачів створюють потребу у застосуванні раціональної архітектури бази даних та ефективних механізмів індексації.

Розроблення й дослідження оптимальної структури бази даних для системи обліку майна дає можливість не лише підвищити швидкість виконання запитів, а й забезпечити цілісність, відмовостійкість і масштабованість системи в умовах постійного оновлення інформації. Технічна актуальність роботи полягає у проведенні практичного аналізу продуктивності запитів при застосуванні різних типів індексів, структур даних і рівнів нормалізації таблиць, що дозволяє визначити оптимальні параметри для реальних систем оборонного призначення.

Таким чином, дослідження спрямоване на розроблення ефективного підходу до побудови та оптимізації бази даних для автоматизованої системи обліку військового майна, що забезпечує баланс між швидкодією, безпекою та надійністю зберігання даних у критично важливих інформаційних середовищах.

Метою дослідження є розроблення та оптимізація бази даних для автоматизованої системи ведення обліку військового майна у Збройних Силах України з метою підвищення ефективності управління матеріально-технічними ресурсами, забезпечення оперативного доступу до інформації та збереження її

цілісності. Для досягнення поставленої мети було визначено такі завдання: створити реляційну структуру бази даних, що відображає процеси зберігання, переміщення та інвентаризації майна; реалізувати логічні зв'язки між сутностями з урахуванням вимог інформаційної безпеки; наповнити таблиці тестовими даними та сформулювати приклади SQL-запитів для аналітичної обробки; дослідити вплив індексації на швидкість виконання запитів у середовищі PostgreSQL; провести експериментальні вимірювання продуктивності системи та розробити рекомендації щодо оптимізації бази даних у централізованих і розподілених архітектурах військових інформаційних систем.

Аналіз останніх досліджень і публікацій.

У наукових працях вітчизняних та зарубіжних дослідників питання створення та оптимізації баз даних розглядається з позицій підвищення продуктивності запитів, забезпечення цілісності даних і впровадження сучасних інструментів управління інформаційними ресурсами.

Зокрема, С. Пчелинська [2] досліджує особливості обліку основних засобів в умовах воєнного стану, наголошуючи на потребі цифровізації бухгалтерських процесів та уніфікації систем контролю матеріальних ресурсів. В. Голубінка та А. Худий [3] аналізують методи індексації баз даних як один із ключових чинників підвищення ефективності SQL-запитів, визначаючи оптимальні техніки побудови індексів для зменшення часу обробки інформації.

У працях С. Суліми та О. Єрмолаєва [6] подано системний підхід до оптимізації SQL-запитів у середовищах управління базами даних, зокрема через використання комбінованих індексів і статистичних методів аналізу планів виконання. Подібні ідеї розвивають зарубіжні дослідники J. Nabimana [4], R. Bhajipale та співавт. [5], які узагальнюють практичні прийоми підвищення швидкодії реляційних СУБД за рахунок оптимізації JOIN-операцій, фільтрації та сортування.

Водночас на нормативному рівні ключову роль відіграє Наказ Міністерства оборони України № 279 від 01 травня 2025 року [1], яким затверджено зміни до Інструкції з обліку військового майна у Збройних Силах України. Документ визначає перехід до єдиної електронної моделі управління даними, що забезпечує централізований облік, простежуваність руху майна та автоматизовану верифікацію облікових операцій.

Таким чином, сучасні наукові та нормативно-практичні напрацювання створюють методологічну основу для розроблення оптимізованих баз даних у військовій сфері, де поєднання індексаційних технологій, аналітичних запитів і захисту інформації має визначальне значення для ефективного управління матеріально-технічними ресурсами.

Виклад основного матеріалу.

Формування сучасної системи обліку військового майна вимагає узгодження технічних рішень із чинною нормативно-правовою базою у сфері оборони, інформаційної безпеки та цифрового управління ресурсами. Національне законодавство України визначає чіткі вимоги щодо організації обліку матеріально-технічних засобів, збереження цілісності даних та забезпечення контролю за рухом військового майна в умовах воєнного стану.

Ключовим нормативним актом, який визначає підходи до систематизації, уніфікації та автоматизації процесів обліку, є Наказ Міністерства оборони України № 279 від 01 травня 2025 року «Про затвердження Змін до Інструкції з обліку військового майна у Збройних Силах України». У новій редакції документа передбачено перехід від фрагментарних локальних облікових систем до єдиної електронної моделі управління даними, що забезпечує наскрізну простежуваність руху майна та централізоване ведення облікових записів.

Зміни, затверджені наказом, закладають принципи уніфікації ідентифікаційних кодів, номенклатурних позицій, складів і місць зберігання, а також передбачають ведення електронних журналів інвентаризації з автоматичним фіксуванням історії руху об'єктів матеріальної відповідальності. Такі підходи відповідають сучасним тенденціям цифрової трансформації систем військового управління, у яких інформаційна цілісність та оперативна верифікація даних мають стратегічне значення [1].

Крім зазначеного нормативного акту, правове регулювання процесу обліку базується на положеннях Закону України «Про Збройні Сили України», Закону України «Про оборону України», а також Закону України «Про захист інформації в інформаційно-телекомунікаційних системах», що визначають правові засади створення, зберігання та оброблення відомостей, які мають обмежений доступ. Ці нормативні документи формують інституційне підґрунтя для впровадження автоматизованих баз даних у військових частинах, регламентуючи як технічні, так і організаційно-безпекові аспекти цифрового обліку.

Отже, нормативно-правове середовище, що формується на основі зазначених документів, не лише легітимізує перехід до автоматизованих систем обліку, а й визначає вимоги до структури, захисту та аудиту даних, створюючи передумови для інтеграції технологічних рішень у загальну архітектуру інформаційного простору Збройних Сил України [2].

Побудова ефективної системи обліку військового майна у Збройних Силах України передбачає розроблення раціональної структури бази даних, здатної забезпечити повноту, достовірність і простежуваність усіх операцій з матеріальними засобами. З огляду на масштабність об'єктів обліку, необхідність збереження історії переміщень та вимоги інформаційної безпеки, логічна модель бази даних має бути спроектована відповідно до принципів нормалізації, уніфікації записів і підтримки транзакційної цілісності.

Структура бази даних повинна відображати ключові елементи процесу обліку: надходження, зберігання, переміщення, інвентаризацію та закріплення матеріальної відповідальності. Для цього запропоновано реляційну модель, що складається з шести основних сутностей, між якими встановлюються логічні зв'язки:

Category – довідник номенклатурних груп майна (озброєння, техніка, форма, засоби зв'язку, спорядження тощо).

Property – базова таблиця, у якій зберігається інформація про кожну одиницю майна: найменування, інвентарний код, категорія, вартість, дата надходження, поточний склад або підрозділ.

Storage – таблиця місць зберігання (склади, підрозділи, ремонтні бази), що дозволяє відстежувати розташування майна.

Movement – журнал переміщень, у якому фіксуються дати, маршрути та підрозділи, між якими відбувається передача матеріальних цінностей.

Inventory – дані про інвентаризації: дати перевірок, результати, відповідальні особи, виявлені нестачі чи надлишки.

Responsible_Personnel – інформація про матеріально відповідальних осіб, їхні посади, військові звання та контактні дані.

Взаємозв'язки між таблицями забезпечуються за допомогою зовнішніх ключів, що формують логічну узгодженість даних.

Таблиця *Property* має зв'язки багато-до-одного з таблицями *Category* та *Storage*, оскільки кожна одиниця майна належить до певної категорії та зберігається в конкретному місці.

Таблиця *Movement* реалізує зв'язок один-до-багатьох з *Property*, що дозволяє фіксувати кілька записів про переміщення однієї одиниці майна.

Таблиця *Inventory* пов'язана з *Storage* і *Property*, фіксуючи актуальний стан майна під час перевірки.

Таблиця *Responsible_Personnel* забезпечує зв'язок між майном та особами, закріпленими за ним, що є основою контролю матеріальної відповідальності.

У сукупності така архітектура створює єдину логічну модель облікової системи, у межах якої кожен об'єкт має унікальний ідентифікатор, історію переміщень, поточне місце зберігання та відповідальну особу. Впровадження цієї моделі дозволяє автоматизувати основні процеси обліку, підвищити точність звітності та знизити ризики втрати або дублювання інформації.

Після проєктування структури бази даних наступним етапом є її практичне наповнення даними з метою перевірки логічної узгодженості між таблицями, працездатності зв'язків та ефективності виконання SQL-запитів. На цьому етапі база даних переходить із концептуальної моделі у робочий стан, що дозволяє здійснювати реальні операції обліку, контролю та аналізу інформації щодо військового майна [3, с. 70–71].

Для наповнення використовуються оператори INSERT INTO, а також генератори тестових записів (Faker, Mockaroo) з метою моделювання типових ситуацій, що виникають у процесі функціонування системи військового обліку.

Таблиця 1

Category – класифікація майна

category_id	category_name	description
1	Озброєння	Зразки стрілецької зброї та боеприпаси
2	Транспорт	Військові автомобілі, бронетехніка
3	Зв'язок	Засоби радіозв'язку, передавачі
4	Спорядження	Уніформа, взуття, засоби індивідуального захисту

Призначення: систематизація майна за функціональними групами для полегшення пошуку та аналітики.

Таблиця 2

Storage – місця зберігання

storage_id	storage_name	location
1	Склад №1	в / ч А1234, м. Київ
2	Склад №2	в / ч А2345, м. Харків
3	Склад №3	в / ч А3456, м. Львів

Призначення: описує фізичні або географічні місця, де зберігається майно.

Таблиця 3

Responsible Personnel – матеріально відповідальні особи

personnel_id	full name	position	rank
1	Капітан Петренко Іван Іванович	Начальник складу	Капітан
2	Сержант Коваленко Олег Петрович	Комірник	Сержант
3	Солдат Бондар Андрій Сергійович	Помічник комірника	Солдат

Призначення: містить інформацію про осіб, відповідальних за облік і збереження майна.

Таблиця 4

Property – військове майно

property_id	property_name	category_id	storage_id	quantity	unit	responsible_id
1	Автомат АК – 74	1	1	50	од.	1
2	Радіостанція Motorola DP4800	3	1	20	од.	2
3	Бронезилет класу 4	4	2	100	од.	3
4	Військовий автомобіль КраЗ – 6322	2	3	5	од.	1

Призначення: є центральною таблицею системи, у якій фіксуються всі одиниці військового майна із зазначенням місця зберігання, категорії та відповідальної особи.

Таблиця 5

Movement – історія переміщень майна

movement id	property id	from storage	to storage	move date
1	1	1	2	2025-05-12
2	2	1	3	2025-06-01
3	3	2	1	2025-06-15

Призначення: фіксує логістику переміщення майна між складами, що важливо для відстеження історії руху та контролю оборотності ресурсів.

Таблиця 6

Inventory – результати інвентаризації

inventory id	property id	storage id	check date	actual quantity
1	1	2	2025-07-10	48
2	2	3	2025-07-10	20
3	3	1	2025-07-10	98

Призначення: відображає фактичну наявність майна під час перевірки, що дозволяє виявляти нестачі або надлишки.

Після розроблення логічної структури бази даних наступним етапом є її практична перевірка шляхом наповнення таблиць тестовими даними та виконанням SQL-запитів. Цей процес дозволяє оцінити коректність зв'язків між таблицями, працездатність системи обліку та адекватність відображення реальних сценаріїв використання бази.

Тестові дані можуть бути створені вручну або згенеровані за допомогою інструментів, таких як Faker, MoscaGo чи власні SQL-скрипти з використанням операторів INSERT INTO. Такий підхід забезпечує імітацію роботи реальної системи Збройних Сил України, де фіксуються процеси надходження, зберігання, інвентаризації та переміщення майна.

Виконання SQL-запитів дає змогу:

- перевірити логіку зв'язків між сутностями;
- виявити можливі помилки у структурі даних;
- сформувані аналітичні звіти для контролю стану майна;
- підтвердити працездатність CRUD-операцій (створення, читання, оновлення, видалення).

Таким чином, наповнення бази даних і проведення запитів виступають ключовим етапом тестування системи, що дозволяє забезпечити її надійність та готовність до практичного використання у військових підрозділах [4, с. 23].

Приклади SQL-запитів для перевірки логіки.

Вибірка даних.

```
SELECT p.name, c.name AS category, s.name AS
storage
FROM Property p
JOIN Category c ON p.category_id = c.id
JOIN Storage s ON p.storage_id = s.id;
```

Результат: відображає перелік усього майна з відповідною категорією та місцем зберігання, що дозволяє перевірити правильність зв'язків між таблицями.

Агрегація даних (звіт за типами).

```
SELECT c.name AS category, COUNT(p.id) AS
total_items, SUM(p.cost) AS total_value
FROM Property p
JOIN Category c ON p.category_id = c.id
GROUP BY c.name;
```

Результат: формує аналітичний звіт за типами майна, показуючи кількість одиниць та їх загальну вартість у межах кожної категорії.

Виявлення нестачі або втрат.

```
SELECT i.inventory_date, s.name, p.name,
i.status
FROM Inventory i
JOIN Storage s ON i.storage_id = s.id
JOIN Property p ON i.property_id = p.id
WHERE i.status = 'Втрачено';
```

Результат: дозволяє виявити випадки втрати майна під час інвентаризації, що є важливою складовою контролю матеріальної відповідальності.

Історія переміщень.

```
SELECT p.name, m.date_moved, m.from_storage,
m.to_storage
FROM Movement m
JOIN Property p ON m.property_id = p.id
ORDER BY m.date_moved DESC;
```

Результат: демонструє історію переміщень кожної одиниці майна між складами чи підрозділами, забезпечуючи прозорість облікових операцій.

У процесі функціонування системи обліку військового майна важливо не лише здійснювати базові операції пошуку, оновлення чи фільтрації даних, але й забезпечити можливість аналітичної обробки інформації для підтримки управлінських рішень.

З цією метою в межах проєкту реалізовано низку складних SQL-запитів, які демонструють потенціал бази даних як інструменту аналітичного рівня.

До таких запитів належать операції умовної класифікації, об'єднання множин даних і фільтрації агрегованих результатів.

1. Класифікація майна за вартістю (оператор CASE).

```
SELECT
p.name,
p.cost,
CASE
WHEN p.cost < 5000 THEN 'Низьковартісне'
WHEN p.cost BETWEEN 5000 AND 20000 THEN 'Середньої
вартості'
ELSE 'Високовартісне'
END AS cost_category
FROM Property p
ORDER BY p.cost DESC;
```

Даний запит здійснює умовну класифікацію об'єктів майна за вартісними групами.

Він може бути використаний для планування ремонтів, списань або формування звітів про структуру активів за фінансовими категоріями.

2. Об'єднання даних про наявне й списане майно (оператор UNION).

```
SELECT p.name, p.inventory_code, 'Наявне' AS status
FROM Property p
WHERE p.is_written_off = FALSE
UNION
SELECT p.name, p.inventory_code, 'Списане' AS status
FROM Property p
WHERE p.is_written_off = TRUE;
```

Запит об'єднує в єдиний звіт записи про наявне та списане майно, що дозволяє створювати повний реєстр матеріальних цінностей із зазначенням їхнього поточного статусу.

Це підвищує зручність контролю та спрощує аудит.

3. Визначення складів із надлишковою кількістю майна (оператор HAVING).

```
SELECT
s.name AS storage,
c.name AS category,
COUNT(p.id) AS total_items
FROM Property p
JOIN Storage s ON p.storage_id = s.id
JOIN Category c ON p.category_id = c.id
```

Цей запит визначає склади або підрозділи, у яких зосереджено понад 20 одиниць певного типу майна.

Результати можуть використовуватися для оптимізації розподілу ресурсів, виявлення перевантажених складів або планування переміщень.

Застосування конструкцій CASE, UNION і HAVING демонструє можливість розширення функціоналу бази даних від простої системи обліку до аналітичного модуля підтримки рішень.

Ці механізми дозволяють здійснювати оперативний аналіз обсягів, вартості та розподілу матеріальних ресурсів у реальному часі, що підвищує ефективність управління військовим майном у Збройних Силах України [5, с. 30].

Ефективність роботи реляційної бази даних у системі обліку військового майна безпосередньо залежить від швидкості доступу до записів.

Оскільки система оперує значними обсягами даних (десятки тисяч одиниць обліку, історія переміщень, інвентаризації, звітні вибірки), одним із ключових інструментів оптимізації виступає *індексація таблиць*.

Індекси дозволяють прискорити пошук, сортування та фільтрацію даних шляхом створення структур (B-tree, Hash, GiST, GIN тощо), які забезпечують прямий доступ до рядків без повного сканування таблиці. Однак надмірна кількість індексів або їх некоректне застосування може призвести до зниження продуктивності при операціях оновлення, вставки або видалення.

У досліджуваній системі застосовано три основні типи індексів:

- Первинний (PRIMARY KEY) – автоматично створюється для унікальної ідентифікації кожного запису.
- Зовнішній (FOREIGN KEY) – забезпечує логічну узгодженість між таблицями (зв'язки «один-до-багатьох»).
- Додаткові (SECONDARY INDEX) – створюються вручну для прискорення аналітичних запитів за полями, що часто використовуються у фільтрації або групуванні (наприклад, category_id, storage_id, cost).

Приклади створення індексів.

```

Індекс для прискорення вибірок за категорією
CREATE INDEX idx_property_category ON
Property(category_id);

Індекс для пошуку за місцем зберігання
CREATE INDEX idx_property_storage ON Property(storage_id);

Індекс для аналітичних запитів за вартістю
CREATE INDEX idx_property_cost ON Property(cost);
GROUP BY s.name, c.name
    
```

Для порівняння ефективності виконання запитів було проведено експериментальне вимірювання часу виконання вибірок до й після створення індексів.

Тестування проводилось у середовищі PostgreSQL із таблицею Property, що містила 100 000 записів (Таблиця 7).

Таблиця 7

Вимірювання продуктивності запитів

№	Тип запиту	Час без індексу (мс)	Час з індексом (мс)	Прискорення
1	SELECT із фільтрацією за category_id	1180	73	×16,2
2	SELECT із сортуванням за cost	960	81	×11,8
3	JOIN Property + Storage	1340	104	×12,9
4	Агрегація з GROUP BY storage_id	1560	122	×12,8

Проведений аналіз показав, що індексація істотно зменшує час виконання запитів, особливо при частому використанні операцій JOIN, WHERE і ORDER BY.

Найбільший ефект спостерігається при застосуванні індексів до полів, що беруть участь у фільтрації або сортуванні, таких як category_id, storage_id та cost.

Разом з тим, створення надлишкової кількості індексів або дублювання їх на полях із низькою селективністю (наприклад, статуси з 2–3 варіантами значень) не дає відчутного приросту швидкості та збільшує навантаження при оновленнях таблиць.

Отже, максимальна продуктивність досягається за умов:

- збалансованого індексування – створення індексів лише для полів, які реально впливають на частоту звернень;
- регулярного аналізу планів виконання запитів (EXPLAIN ANALYZE) – для виявлення «вузьких місць» у продуктивності;
- оптимізації структури JOIN-зв'язків – уникнення зайвих вкладених підзапитів і неіндексованих зовнішніх ключів;
- контролю оновлень статистики (VACUUM ANALYZE) – підтримання актуальної інформації для оптимізатора запитів.

Таким чином, індексація виступає ключовим механізмом підвищення ефективності обробки даних у системах військового обліку, забезпечуючи швидку аналітичну вибірку без втрати точності або узгодженості інформації [6, с. 154–155].

Для оцінювання ефективності застосованих індексів було проведено експериментальні вимірювання часу виконання SQL-запитів у тестовому середовищі.

Мета – визначити, наскільки індексація впливає на швидкість обробки даних у системі обліку майна Збройних Сил України.

Для дослідження використано систему управління базами даних *PostgreSQL 15*, розгорнуту на сервері з такими характеристиками:

- процесор: *Intel Core i7* (8 ядер, 3.4 GHz);
- оперативна пам'ять: *16 GB*;
- операційна система: *Windows 11 x64*.

Створено модель бази даних, що включає таблиці *Property, Category, Storage, Movement, Inventory, Responsible Personnel*.

У таблицю *Property* було внесено 100 000 записів із реалістичними значеннями вартості, категорій і місць зберігання.

Типи запитів для порівняння:

- фільтрація (*WHERE*) – пошук майна за категорією;
- сортування (*ORDER BY*) – впорядкування за вартістю;
- об'єднання (*JOIN*) – формування звітів із кількох таблиць;
- агрегація (*GROUP BY*) – підрахунок кількості одиниць за місцем зберігання.

Час виконання кожного запиту фіксувався за допомогою вбудованої команди:

```
EXPLAIN (ANALYZE, BUFFERS),
```

яка дозволяє визначити реальний час виконання, кількість зчитаних сторінок і використання індексів.

Для кожного запиту проводилось по три виконання:

- до створення індексів;
- після створення індексів;
- після оновлення статистики (*VACUUM ANALYZE*).

Результати усереднювались для зменшення похибки вимірювань.

Середній час виконання обчислювався за формулою:

$$T_{avg} = \frac{1}{n} \sum_{i=1}^n T_i$$

де: T_{avg} – середній час виконання запиту, мс;

T_i – час виконання i -го запиту;

n – кількість повторів експерименту [7, с. 368].

Основним критерієм оцінювання виступав середній час виконання запиту (мс).

Додатково аналізувалися такі метрики:

- кількість зчитаних рядків (*rows*);
- кількість звернень до диску (*I/O operations*);
- тип використаного плану (*Seq Scan* або *Index Scan*).

Після виконання серії тестів спостерігалось значне скорочення часу обробки даних у випадках, коли запити використовували фільтрацію або сортування за проіндексованими полями.

Зокрема, середнє прискорення склало від 11 до 16 разів у порівнянні з початковими значеннями без індексів.

Результати експерименту підтвердили, що:

- найбільше зниження часу досягається для запитів типу *JOIN* та *WHERE*;
- ефективність індексації залежить від селективності поля (кількості унікальних значень);
- надмірна кількість індексів негативно впливає на швидкість оновлення даних;
- регулярне оновлення статистики (*VACUUM ANALYZE*) є критично важливим для коректної роботи оптимізатора запитів [8, с. 189].

Висновки

Отже, проведене дослідження підтвердило, що продуктивність системи ведення обліку військового майна у Збройних Силах України суттєво залежить від оптимізації структури бази даних, застосування відповідних типів індексів і коректного вибору умов виконання запитів.

У результаті аналізу було встановлено такі закономірності:

1. *Тип індексації та характер запитів.* Індекси типу *B-tree* показали найвищу ефективність для операцій вибірки з умовами *WHERE* і сортування *ORDER BY*, забезпечивши скорочення часу виконання запитів у середньому в 12–15 разів. Для складних зв'язків між таблицями (*JOIN*) доцільним є використання *composite indexes* (складених індексів), що зменшують кількість звернень до диску. У випадках великого обсягу текстових даних або пошуку за фрагментами рядків доцільно застосовувати *GIN* чи *GiST* індекси.

2. *Вплив обсягу даних і структури зв'язків.* Зі збільшенням кількості записів понад 100 000 спостерігається нелінійне зростання часу обробки запитів у таблицях без індексації, тоді як індексовані структури демонструють стабільну швидкодію. Наявність чітких зв'язків типу «один-до-багатьох» між сутностями (*Property* → *Category*, *Property* → *Storage*) сприяє ефективній оптимізації планів виконання запитів, особливо при попередньому оновленні статистики.

3. *Вплив денормалізації*. Часткова денормалізація (наприклад, дублювання ідентифікаторів складів або категорій у таблиці *Property*) може підвищити швидкість вибірки при великому навантаженні, однак збільшує ризик втрати цілісності даних. Тому рекомендовано дотримуватись *третьої нормальної форми (3NF)* для більшості таблиць і застосовувати денормалізацію лише у звітних підсистемах або аналітичних копіях бази.

4. *Оптимальні умови функціонування*. Для централізованої архітектури (єдиний сервер бази даних військової частини) найбільшу ефективність забезпечує комбінація:

- індексація ключових полів (*category_id, storage_id, property_id*),
- регулярне оновлення статистики командою *VACUUM ANALYZE*,
- використання кешування результатів (*Query Cache*).

У розподілених системах (наприклад, на рівні оперативних командувань) доцільно застосовувати реплікацію даних і горизонтальне розділення таблиць (*sharding*), що дозволяє зберігати високу продуктивність навіть при одночасній роботі кількох підрозділів.

5. Підтримка цілісності та резервування.

Для забезпечення надійності обліку рекомендується:

- використовувати механізми *foreign key constraints* і *ON UPDATE CASCADE*;
- реалізувати щоденне інкрементальне резервне копіювання та щотижневе повне;
- зберігати копії баз даних у географічно рознесених центрах обробки інформації (згідно з вимогами кібербезпеки МОУ);

- контролювати коректність змін через тригери аудиту та журнал транзакцій.

Узагальнюючи результати дослідження, можна стверджувати, що застосування індексів істотно підвищує продуктивність виконання SQL-запитів, особливо під час фільтрації та аналітичних обчислень. Водночас надмірна кількість або неправильно обрані типи індексів можуть знижувати ефективність системи під час масових оновлень даних, створюючи додаткове навантаження на сервер. Тому регулярне обслуговування бази даних, зокрема оновлення статистики, дефрагментація та моніторинг планів виконання запитів, є ключовими чинниками стабільної та надійної роботи облікової системи Збройних Сил України.

Література

1. Про затвердження Змін до Інструкції з обліку військового майна у Збройних Силах України: Наказ Міністерства Оборони України від 01.05.2025 № 279. URL: <https://zakon.rada.gov.ua/laws/show/z0688-25#Text> (дата звернення: 04.10.2025).
2. Пчелинська Г. Особливості обліку вибуття основних засобів в умовах воєнного стану в Україні. *Економіка та суспільство*. №51. 2023. DOI: <https://doi.org/10.32782/2524-0072/2023-51-13>
3. Голубінка В. П., Худий А. М. Підвищення продуктивності запитів до баз даних: аналіз технік індексації. *Наукові журнали та конференції*. 2024. Вип. 15. С. 65-73. DOI: <https://doi.org/10.23939/sisn2024.15.065>.
4. Habimana J. Query optimization techniques – tips for writing efficient and faster SQL queries. *International Journal of Scientific & Technology Research*. 2015. №4(10). P. 22–26.
5. Bhajipale R., Bisen P., Meshram A., & Thakur S. S. SQL tuner. *International Journal of Computer Trends and Technology*. 2016. №33(1). Pp. 29–32. DOI: <https://doi.org/10.14445/22312803/IJCTT-V33P106>
6. Суліма С. В., Єрмолаєв О. Д. Метод оптимізації SQL запитів системи управління базами даних. *Системи управління, навігації та зв'язку*. 2023. №2. С. 151-157. DOI: <https://doi.org/10.26906/SUNZ.2023.2.151>.
7. Sahal R., Nihad M., Khafagy M. H., & Omara F. A. iHOME: Index based JOIN query optimization for limited big data storage. *Journal of Grid Computing*. 2018. №16(2). Pp. 345–380.
8. Olszak J., Skublewska-Paszowska M. The impact of relational and non-relational databases on application performance. *Journal of Computer Sciences Institute*. 2025. №35. Pp. 184–190. DOI: <https://doi.org/10.35784/jcsi.7260>

References

1. Law of Ukraine on approval of Amendments to the Instruction on accounting of military property in the Armed Forces of Ukraine № 279. (2025, May 1). *Ministry of Defence of Ukraine*. Retrieved October 4, 2025, from <https://zakon.rada.gov.ua/laws/show/z0688-25#Text> [in Ukrainian].
2. Pchelynska, H. (2023). Osoblyvosti obliku vybuttia osnovnykh zasobiv v umovakh voiennoho stanu v Ukraini [Features of accounting for the disposal of fixed assets under martial law in Ukraine]. *Ekonomika ta suspilstvo*, 51. <https://doi.org/10.32782/2524-0072/2023-51-13> [in Ukrainian].
3. Holubinka, V. P., & Khudyi, A. M. (2024). Pidvyshchennia produktyvnosti zapytiv do baz danykh: analiz tekhnik indeksatsii [Improving database query performance: analysis of indexing techniques]. *Naukovi zhurnaly ta konferentsii*, 15, 65–73. <https://doi.org/10.23939/sisn2024.15.065> [in Ukrainian].
4. Habimana, J. (2015). Query optimization techniques – tips for writing efficient and faster SQL queries. *International Journal of Scientific & Technology Research*, 4(10), 22–26.
5. Bhajipale, R., Bisen, P., Meshram, A., & Thakur, S. S. (2016). SQL tuner. *International Journal of Computer Trends and Technology*, 33(1), 29–32. <https://doi.org/10.14445/22312803/IJCTT-V33P106>
6. Sulima, S. V., & Yermolaiev, O. D. (2023). Metod optymizatsii SQL zapytiv systemy upravlinnia bazamy danykh [Method of SQL query optimization in database management systems]. *Systemy upravlinnia, navihatsii ta zviazku*, 2, 151–157. <https://doi.org/10.26906/SUNZ.2023.2.151> [in Ukrainian].
7. Sahal, R., Nihad, M., Khafagy, M. H., & Omara, F. A. (2018). iHOME: Index based JOIN query optimization for limited big data storage. *Journal of Grid Computing*, 16(2), 345–380.
8. Olszak, J., & Skublewska-Paszowska, M. (2025). The impact of relational and non-relational databases on application performance. *Journal of Computer Sciences Institute*, 35, 184–190. <https://doi.org/10.35784/jcsi.7260>