Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

«Застосування та оцінювання продуктивності сервісів хмарних платформ для обробки великих даних»
(тема)

Виконав:

студент 2 курсу, групи ІТПм-22-2
Башкіров М. О.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні технології проектування
(повна назва освітньої програми)

Керівник проф. Мінухін С. В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Гребеннік І.В
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Системотехніки

Рівень вищої освіти другий (магістерський)

Спеціальність 122-Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні технології проектування
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)
«____»_____20____р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Башкірову Мирославу Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Застосування та оцінювання продуктивності сервісів хмарних платформ для обробки великих даних»
затверджена наказом університету від 20 листопада 2023 р. № 1373 Ст
2. Термін подання студентом роботи до екзаменаційної комісії 12 січня 2024 р.
3. Вихідні дані до роботи Дані статей та публікацій, книг та електронних ресурсів за темою роботи.
4. Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ, 4.2 Аналіз предметної області, вибір та обґрунтування хмарних платформ, сервісів щодо оброблення великих даних, 4.2.1 Зростання обсягів даних та проблеми їх обробки, 4.2.2 Технології Big Data, 4.2.3 Хмарні сервіси для обробки великих даних, 4.2.4 Огляд провідних хмарних платформ та сервісів, 4.2.4.1 Платформа Amazon Web Services, 4.2.4.2 Платформа Microsoft Azure, 4.2.4.3 Платформа Google Cloud Platform, 4.2.4.4 Оцінка та порівняльний аналіз провідних хмарних платформ, 4.2.5 Обґрунтування вибору хмарної платформи на основі аналізу сервісів реляційних баз даних, 4.2.6 Загальні переваги та недоліки хмарних технологій, 4.2.7 Опис предметної області дослідження, 4.2.7.1 Актуальність дослідження, 4.2.7.2 Обґрунтування актуальності застосування хмарних технологій на торгівельних платформах, 4.2.8 Постановка завдань дослідження, 4.3 Розроблення моделей даних та запитів, тестування на локальному ресурсі, 4.3.1 Опис предметної області об'єкту управління прототипу торгівельної компанії, 4.3.1.1 Аналіз реальної моделі торгівельної компанії, 4.3.1.2 Обґрунтування моделі даних, 4.3.1.3 Обґрунтування обсягів даних, 4.3.2 Проєктування та розроблення БД, 4.3.2.1 Вибір системи управління базами даних, 4.3.2.2 Проєктування моделей БД, 4.3.3 Генерація тестових даних, 4.3.3.1 Розробка скриптів генерації, 4.3.3.2 Аналіз та візуалізація згенерованих даних, 4.3.4 Розроблення та опис запитів до БД, 4.3.5 Моделювання виконання запитів на локальному сервері бази даних, 4.4 Експериментальне дослідження продуктивності хмарної бази даних Azure, 4.4.1 Огляд рівнів моделі DTU сервісу БД Azure SQL Database, 4.4.2 Метрики продуктивності, 4.4.2.1 Час відгуку запитів, 4.4.2.2 Використання CPU, 4.4.2.3 Використання DTU, 4.4.3 Порівняльний аналіз часу генерації даних на локальному ресурсі та Azure SQL Database, 4.4.3.1 Опис характеристик ресурсів, 4.4.3.2 Методологія порівняння, 4.4.3.3 Порівняльний аналіз показників генерації даних, 4.4.4 Розроблення режимів оброблення запитів, 4.4.4.1 Інтерактивний режим виконання запитів, 4.4.4.2 Пакетні режими, 4.4.5 Обробка та аналіз отриманих результатів, 4.4.5.1 Аналіз результатів застосування інтерактивного режиму, 4.4.5.2 Аналіз результатів застосування пакетних режимів, 4.5 Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) ER-діаграма за нотацією Чена, Концептуальна модель даних, Логічна модель бази даних, Фізична модель бази даних, Розподіл згенерованих даних по сутностях, %, Залежність об'єму пам'яті від кількості записів у таблицях, Залежність часу генерації від кількості записів у таблицях, Графік часу виконання запитів, Графік часу виконання запитів та кількість повернутих записів за запитами, Графік залежності часу генерації від кількості DTU для рівнів моделі, Графік вартості та часу генерації даних для різних рівнів моделі, Графік комбінованого аналізу використання ресурсів та часу при генерації даних, Порівняльний аналіз часу генерації даних по таблицях БД для різних рівнів моделі, Графік динаміки основних метрик продуктивності у інтерактивному режимі на рівні S0 для різних запитів, Графік динаміки основних метрик продуктивності по запитах в інтерактивному режимі на рівні S1, Графік динаміки основних метрик продуктивності по запитах в інтерактивному режимі на рівні S3, Графік динаміки основних метрик продуктивності по запитах в інтерактивному режимі на рівні S7, Графік динаміки основних метрик продуктивності по запитах в інтерактивному режимі на рівні S12,Графік динаміки використання CPU в інтерактивному режимі на різних рівнях моделі, Графік динаміки використання DTU в інтерактивному режимі на різних рівнях моделі, Графік динаміки часу виконання запитів в інтерактивному режимі на різних рівнях моделі, Графік сумарного часу виконання запитів в інтерактивному режимі на різних рівнях моделі, Графік залежності вартості виконання запитів та часу для різних рівнів моделі в інтерактивному режимі, Графік основних метрик продуктивності від рівня моделі сервісу (Довільний порядок), Графік основних метрик продуктивності для різних рівнів моделі (Порядок за зростанням часу), Графік основних метрик продуктивності для різних рівнів моделі (Порядок за спаданням часу)

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основний розділ	проф. Мінухін. С. В.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на виконання роботи	16.10.2023	
2	Огляд критеріїв для розробки методу порівняння	17-23.11.2023	
3	Огляд критеріїв порівняльного аналізу	24-31.11.2023	
4	Огляд літератури	01-17.11.2023	
5	Опис математичної моделі порівняльного аналізуі	18-30.11.2023	
6	Розробка експериментальних систем	01-15.12.2023	
7	Проведення експерименту	10-15.12.2023	
8	Збор та аналіз даних експерименту	15-20.12.2023	
9	Оформлення матеріалів кваліфікаційної роботи	20-31.12.2023	
10	Подання кваліфікаційної роботи керівникові та її попередній захист	11.01.2024	
11	Подання на рецензування	12.01.2024	

Дата видачі завдання 16 жовтня 2023 р.

Студент _____ Башкіров М.О.
(підпис)

Керівник роботи _____ проф. Мінухін. С. В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до магістерської кваліфікаційної роботи містить: 165 с., 51 рис., 24 табл., 54 джерел, 2 додатка.

БАЗИ ДАНИХ, БІЗНЕС-ПРОЦЕСИ, ВЕЛИКІ ДАНІ, ЕЛЕКТРОННА КОМЕРЦІЯ, ПРОДУКТИВНІСТЬ, ТОРГІВЕЛЬНІ ПЛАТФОРМИ, МОДЕЛЬ DTU, AZURE SQL DATABASE, CPU, MICROSOFT AZURE, MS SQL SERVER, SQL-ЗАПИТИ.

Об'єктом дослідження є процеси створення, оброблення та зберігання великих обсягів даних на хмарній платформі з використанням сервісу щодо реляційних баз даних для потреб торгівельних платформ.

Предметом дослідження є методи, засоби та технології зберігання і обробки великих обсягів даних на базі хмарних сервісів Microsoft Azure для оптимізації вирішення завдань торгівельних платформ.

Метою дослідження є підвищення ефективності обробки великих обсягів даних на прикладі сфер функціонування сучасних торгівельних платформ (майданчиків) шляхом використання хмарного сервісу роботи з реляційними базами даних Azure SQL Database та на основі впливу рівнів моделі сервісу при реалізації SQL-запитів різної складності.

Методи дослідження: системний аналіз – аналіз науково-технічної літератури; моделювання – процеси обробки реляційних баз даних з використанням обчислювальних пристроїв з різною продуктивністю, зокрема, віртуальних середовищ; статистичний аналіз – оцінювання продуктивності сервісів хмарної платформи Azure та статистичний аналіз отриманих результатів.

Наукова новизна роботи полягає у дослідженні та порівнянні ефективності використання різних конфігурацій хмарного сервісу Azure SQL Database для оптимізації обробки великих даних на прикладі торгівельних платформ.

Галузь застосування – торгівельні платформи, що використовують хмарні технології для обробки великих даних про клієнтів, товари, замовлення з метою оптимізації бізнес-процесів компанії.

ABSTRACT

The explanatory note to the master's qualification thesis contains: 165 p., 51 figures, 24 tables, 54 sources, 2 applications.

AZURE SQL DATABASE, BIG DATA, BUSINESS PROCESSES, CPU, DATABASES, DTU MODEL, E-COMMERCE, MICROSOFT AZURE, MS SQL SERVER, PRODUCTIVITY, SQL QUERIES, TRADING PLATFORMS

The object of study is the processes of creating, processing and storing large amounts of data on a cloud platform using a service for relational databases for the needs of trading platforms.

The subject of the research is methods, tools and technologies for storing and processing large amounts of data based on Microsoft Azure cloud services to optimize solving problems of trading platforms.

The aim of the research is to increase the efficiency of processing large amounts of data on the example of the functioning spheres of modern trading platforms by using the Azure SQL Database cloud service for working with relational databases and based on the influence of service model levels when implementing SQL queries of varying complexity.

Research methods: system analysis – analysis of scientific and technical literature; modeling – processes of processing relational databases using computing devices with different performance, in particular, virtual environments; statistical analysis – evaluation of the performance of Azure cloud platform services and statistical analysis of the results obtained.

The scientific novelty of the work lies in the study and comparison of the effectiveness of using different configurations of the Azure SQL Database cloud service to optimize big data processing on the example of trading platforms.

Field of application – trading platforms that use cloud technologies to process large amounts of data about customers, goods, orders in order to optimize the company's business processes.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТЕРМІНІВ

БД – база даних;

ПО – предметна область;

СУБД – Система Управління Базою Даних;

AWS – Amazon Web Services – Веб-сервіси Amazon;

Azure – хмарна платформа від компанії Microsoft;

Azure SQL Database – хмарний сервіс для роботи з реляційними базами даних на платформі Azure;

Big Data – великі дані;

CPU – Central Processing Unit – центральний процесор;

CRM – Customer Relationship Management – Управління відносинами з клієнтами;

DTU – Database Transaction Unit – одиниця виміру продуктивності бази даних;

GCP – Google Cloud Platform – хмарна платформа від компанії Google;

IaaS – Infrastructure as a Service – Інфраструктура як сервіс;

IDC – International Data Corporation – Міжнародна корпорація з обробки даних;

IoT – Internet of Things – Інтернет речей;

Microsoft SQL Server – реляційна система управління базами даних, розроблена корпорацією Microsoft;

OLTP – Online Transaction Processing – онлайн-транзакційна обробка даних, технологія обробки транзакцій в режимі реального часу;

PaaS – Platform as a Service – Платформа як сервіс;

SaaS – Software as a Service – Програмне забезпечення як сервіс;

SQL – Structured Query Language – Мова Структурованих Запитів;

T-SQL – Transact-SQL – реалізація мови SQL корпорацією Microsoft, що використовується для програмування та адміністрування в Microsoft SQL Server.

ВСТУП

За прогнозами IDC, до 2025 року загальний обсяг даних, що генеруються і споживаються в усьому світі, досягне 175 зеттабайт [1]. Це величезні обсяги інформації, які потребують ефективних IT-рішень для збору, зберігання та аналізу.

Застосування та оцінювання продуктивності сервісів хмарних платформ для обробки великих даних є критичними завданнями у сучасному бізнесі, особливо для компаній, які мають справу з великими обсягами інформації, такими, як торгівельні платформ. Збільшення обсягу генерованих даних в вимагає ефективних рішень для їхньої обробки та аналізу.

Великі компанії, такі як Amazon, Rozetka, Walmart та інші, щоденно опрацьовують великі обсяги транзакцій та мають величезну клієнтську базу. Для них хмарні технології стають ключовими інструментами для забезпечення ефективної обробки великих даних. Завдяки хмарним сервісам вони можуть масштабувати свої обчислювальні ресурси, отримувати доступ до передових інструментів аналізу даних і знижувати витрати на IT-інфраструктуру.

Хмарні платформи, такі як Microsoft Azure, Google Cloud Platform, Amazon Web Services та інші, надають багато можливостей для розв'язання завдань обробки великих даних.

Крім того, важливо зазначити, що в галузі обробки великих даних постійно з'являються нові технології та інструменти, які можуть покращити продуктивність та ефективність обробки даних в хмарних платформах. Такі інновації можуть включати в себе вдосконалені алгоритми обробки даних, розширені можливості інтеграції зі сторонніми системами та інші покращення. Тому компанії мають постійно відстежувати ринок і оновлювати свої стратегії для обробки великих даних, щоб залишатися конкурентоспроможними.

У світі, де дані є валютою майбутнього, ефективне використання хмарних платформ для обробки великих даних стає ключовим фактором успіху для багатьох компаній. Ті, хто зможе правильно вибрати та налаштувати сервіси хмарних платформ, а також систематично оцінювати їх продуктивність, матимуть перевагу на ринку та зможуть досягти кращих результатів у своїй діяльності. Однак вибір оптимальної хмарної платформи та оцінка продуктивності її сервісів для

конкретних завдань обробки великих даних залишається актуальною задачею, що потребує дослідження. Існують різні методи та підходи до оцінки продуктивності хмарних сервісів, зокрема тестування продуктивності окремих компонентів, моделювання робочих навантажень, аналіз витрат тощо.

Об'єктом дослідження є процеси створення, оброблення та зберігання великих обсягів даних у хмарному середовищі з використанням реляційних систем управління базами даних для потреб торгівельних платформ.

Мета роботи спрямована на використання методів, інструментів і технологій опрацювання та оптимізації обробки великих обсягів даних на основі хмарних сервісів, а також оцінювання їх ефективності у сфері роботи торгівельних платформ (майданчиків). Вивчити можливості та розробити рекомендації для ефективного використання хмарних платформ разом із реляційними СУБД для оптимізації обробки великих обсягів даних у сфері провідних торгівельних платформ.

Методи дослідження включають аналіз науково-технічної літератури для ознайомлення з існуючими підходами, моделювання процесів оброблення даних у хмарному середовищі, експериментальну оцінку продуктивності різних конфігурацій хмарної платформи, а також статистичний аналіз отриманих результатів.

Наукова новизна роботи полягає у дослідженні та порівнянні ефективності використання різних конфігурацій хмарних сервісів Azure для обробки великих даних на прикладі торгівельних платформ (майданчиків).

Результати дослідження можуть бути використані торгівельними платформами для налаштування оптимальної конфігурації хмарних сервісів Azure задля підвищення ефективності їх роботи.

За темою роботи були опубліковані тези на Міжнародній науково-технічній конференції «Інформаційні системи та технології» (ICT-2023) [2].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ, ОГЛЯД ТА ОБҐРУНТУВАННЯ ХМАРНИХ ПЛАТФОРМ, СЕРВІСІВ ЩОДО ОБРОБЛЕННЯ ВЕЛИКИХ ДАНИХ

Торгівельні платформи стали невід'ємною частиною сучасної економіки та споживчого ринку. Ця форма торгівлі проникає в різні сфери бізнесу та взаємодіє з різними видами клієнтів.

Торгівельні платформи дозволяють здійснювати купівлю, продаж, обмін товарів, послуг та інформації за допомогою комп’ютерних мереж, зокрема Інтернету. Торгівельні платформи мають багато переваг, таких як зручність, швидкість, вибір, ціна тощо.

З розвитком торгівельних платформ зростають і обсяги даних, які генеруються в процесі онлайн-торгівлі. Це дані про користувачів, їх переваги та поведінку, інформація про товари, ціни, залишки, замовлення тощо. Ефективна обробка та аналіз цих даних дозволяє компаніям краще розуміти своїх клієнтів, оптимізувати асортимент і ціни, покращити процеси доставки та логістики.

Для обробки великих обсягів даних на торгівельних платформах все частіше застосовуються технології Big Data та хмарні сервіси. Вони надають можливість масштабувати ресурси під зростаючі потреби в обробці даних, використовувати розподілені бази даних та потужності для проведення аналізу.

Великі дані (Big Data) – це термін, який описує великі, складні та швидко змінювані набори даних, які вимагають нових методів аналізу та обробки [3]. Великі дані стали актуальними для багатьох сфер діяльності, особливо для компаній, які працюють з великими обсягами інформації. Застосування технологій обробки великих даних дозволяє компаніям використовувати великі дані для покращення своєї комерційної діяльності, аналізуючи поведінку покупців, та для покращення якості обслуговування клієнтів.

Торгівельна діяльність з Big Data передбачає збір, збереження, обробку та аналіз великих обсягів даних, що стосуються товарів, покупців, філіалів, складів, робітників, категорій товарів, брендів, замовлень тощо. Це дозволяє компаніям отримувати цінну інформацію про свою діяльність та ринок, що може бути використана для прийняття рішень та планування стратегії.

Хмарні сервіси (Cloud Services) – це послуги, що надаються через Інтернет на вимогу користувача хмарними провайдерами, такими як AWS, Microsoft Azure, Google Cloud Platform та інші [4]. Хмарні сервіси дозволяють компаніям отримувати доступ до різноманітних IT-ресурсів онлайн, без необхідності будувати і підтримувати власну IT-інфраструктуру.

Хмарні сервіси класифікуються за моделями обслуговування: IaaS (інфраструктура як сервіс), PaaS (платформа як сервіс) та SaaS (програмне забезпечення як сервіс) [5]. Хмарні сервіси надають широкий спектр послуг і ресурсів, включаючи зберігання даних, обчислення, програмне забезпечення, бази даних, аналітику, мережеві послуги та інше.

Переваги використання хмарних сервісів включають високу доступність, масштабованість, можливість резервного копіювання даних, зручність доступу до ресурсів з будь-якого місця, а також можливість платити тільки за те, що ви використовуєте. Це робить хмарні сервіси популярними серед підприємств та індивідуальних користувачів для забезпечення обчислювальних та інформаційних потреб.

Все більше торгівельних майданчиків використовують переваги хмарних сервісів та технологій Big Data для ефективної обробки зростаючих обсягів даних, покращення бізнес-аналітики та прийняття рішень.

1.1. Зростання обсягів даних та проблеми їх обробки

Сучасний бізнес стикається з різноманітними джерелами і обсягами даних, що вимагають ефективної обробки та аналізу. Подібно до торгівельних компаній, яка є однією з галузей, що відзначається найбільшим зростанням обсягів даних, інші сектори також стикаються із схожими викликами.

Зростання обсягів даних виникає не лише завдяки розвитку конкретних галузей, але й завдяки загальному зростанню цифрової активності в сучасному світі. Дані включають в себе інформацію про клієнтів, продукти, послуги, транзакції, історії взаємодії та багато іншого. Інтенсивний обмін даними створює потужний інформаційний ресурс для оптимізації бізнес-процесів та поліпшення обслуговування клієнтів.

Обробка таких великих обсягів даних стає все складнішою та вимагає спеціальних рішень. Потрібне належне обладнання та програмне забезпечення для забезпечення безпеки, масштабованості та ефективності зберігання цих даних. Виникають проблеми зі збереженням, масштабуванням та швидкодією аналізу інформації.

Також важливо зазначити, що інформація може бути розпорошеною по різних джерелах, включаючи бази даних, системи управління відносинами з клієнтами (CRM), соціальні мережі, інтернет-магазини, програми лояльності клієнтів та багато інших. Це ускладнює аналіз і використання цих даних для прийняття стратегічних рішень та покращення обслуговування клієнтів.

Для вирішення цих викликів потрібні нові технології та архітектури, які здатні ефективно працювати з великими обсягами даних, забезпечувати їх доступність, масштабованість та швидкий аналіз. В цьому контексті концепції Big Data та хмарні сервіси та технології грають ключову роль.

Отже, обробка і аналіз зростаючих обсягів даних є актуальним питанням для сучасного бізнесу незалежно від галузі. Комплексне використання сучасних підходів та технологій дозволить ефективно розв'язати ці проблеми і отримати цінну аналітику для бізнес-розвитку.

1.2. Технології Big Data

Технології Big Data – це надзвичайно розширений і складний комплекс методів і інструментів, призначених для фундаментальної перетворення обробки даних у великому масштабі [6]. Ці технології не лише сприяють збору, зберіганню, обробці та аналізу великих обсягів даних, але й відкривають перед собою безмежні можливості для розвитку різних галузей.

Основні характеристики Big Data включають:

1. Великий обсяг (Volume): величезні кількості даних, що генеруються щодня, вимагають ефективних методів збору та зберігання. Викинути або не обробляти ці дані більше не варіант, і технології Big Data допомагають вирішити цю проблему [6].

2. Висока швидкість (Velocity): споживачі і компанії вимагають негайного доступу до даних та їх обробки в реальному часі. Big Data технології дозволяють реалізувати цю потребу завдяки потужним обчислювальним ресурсам та алгоритмам [6].

3. Різноманітність (Variety): дані можуть бути представлені у різних форматах, включаючи тексти, числа, зображення, відео та багато іншого. Big Data технології надають змогу ефективно опрацьовувати цю різноманітність та витягати корисну інформацію з різних джерел [6].

Технології Big Data стають надзвичайно важливими для сучасного бізнесу, наукових досліджень та громадянського суспільства в цілому. Вони дозволяють приймати більш обґрунтовані рішення, виявляти приховані залежності та тенденції, а також впливати на подальший розвиток технологій та соціокультурних процесів.

1.3 Хмарні сервіси для обробки великих даних

Хмарні сервіси стають ключовим інструментом для компаній, що працюють з Big Data. Вони надають гнучку IT-інфраструктуру та обчислювальні ресурси для роботи з великими обсягами даних, і дозволяють зберігати та обробляти дані на віддалених серверах, вибравши їх власні фізичні ресурси.

Однією з ключових переваг використання хмарних сервісів для обробки великих даних є можливість швидкого масштабування ресурсів відповідно до потреб бізнесу. Хмарні провайдери, такі як AWS, Azure та Google Cloud, надають широкий спектр сервісів для автоматичного масштабування обчислювальних ресурсів, сховищ даних та інших ресурсів. Це означає, що компанії можуть ефективно реагувати на зміни обсягів даних та обсяг завдань.

Крім того, важливою є можливість використання широкого спектру спеціалізованих сервісів хмарних провайдерів для аналізу даних, машинного навчання та бізнес-аналітики на основі хмарних платформ. Це робить розгортання рішень Big Data більш ефективним і економічним, оскільки компанії можуть використовувати готові інструменти і сервіси замість розробки їх власних рішень з нуля.

Поєднання технологій Big Data та хмарних сервісів дозволяє компаніям ефективно працювати з величезними обсягами даних, отримувати цінну аналітику, оптимізувати бізнес-процеси та приймати обґрунтовані рішення на основі даних. Це стає критично важливим для успіху в сучасному динамічному цифровому середовищі.

Переваги поєднання технологій Big Data та хмарних сервісів для торгівельних компаній включають:

– зменшення витрат. Використання технологій Big Data та хмарних сервісів дозволяє компаніям зменшити витрати на IT-інфраструктуру, обслуговування, оновлення та безпеку. Компанії платять тільки за те, що вони використовують, і можуть масштабувати свої ресурси в залежності від потреб;

– підвищення продуктивності. Використання технологій Big Data та хмарних сервісів дозволяє компаніям автоматизувати та оптимізувати свої бізнес-процеси, такі як управління запасами, логістика, маркетинг, обслуговування клієнтів тощо. Це підвищує продуктивність та ефективність компанії.

Таким чином, технології Big Data та хмарні сервіси є потужними інструментами для торгівельних компаній, які допомагають їм отримувати конкурентні переваги, оптимізувати свої показники та адаптуватися до змін ринків.

1.4 Огляд провідних хмарних платформ та сервісів

Сучасний світ інформаційних технологій переживає величезні зміни, і однією з ключових інноваційних областей є використання хмарних технологій. Хмарні платформи та сервіси вже давно перетворили спосіб, яким бізнеси зберігають, обробляють та використовують дані.

Хмарні платформи – це інфраструктура та сервіси, що надаються через Інтернет для зберігання даних, розгортання і виконання додатків [7]. Вони дозволяють розробникам створювати та розгортати програми, які працюють у хмарі, а користувачам - отримувати доступ до даних, сервісів і додатків з будь-якого пристрою, підключеного до Інтернету [7]. Хмарні платформи забезпечують гнучкість та масштабованість обчислювальних ресурсів. Ілюстративне зображення хмарних

технологій представлено на рисунку 1.1.

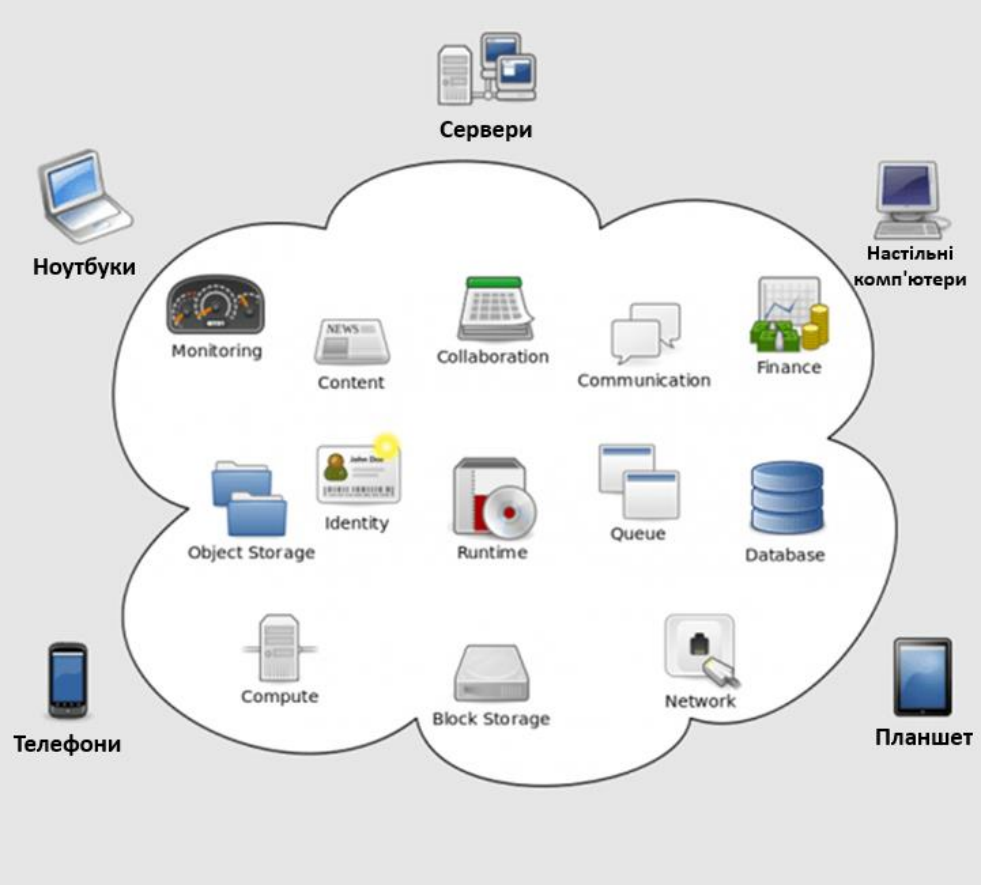

Рисунок 1.1 – Хмарні технології

Використання хмарних платформ надає ряд переваг для організацій, що використовують їх для розгортання IT-інфраструктури та додатків:

– зниження витрат: на рахунок відсутності необхідності в придбанні власного обладнання, програмного забезпечення та ліцензій, а також утримання власних серверів та іншої IT-інфраструктури, використання хмарних платформ дозволяє компаніям знизити капітальні та операційні витрати [7];

– підвищення продуктивності: відсутність необхідності в підтримці власної IT-інфраструктури дозволяє компаніям зосередити IT-персонал на більш стратегічних завданнях, а не на рутинних операціях з обслуговування техніки [7];

– підвищення доступності: Завдяки використанню хмарної інфраструктури користувачі можуть отримати доступ до додатків, сервісів та даних з будь-якого пристрою та місця зі встановленим підключенням до Інтернету [7];

– масштабованість: хмарні платформи дозволяють швидко та еластично

нарощувати обчислювальні потужності та інші ресурси відповідно до зростаючих потреб бізнесу [7];

– гнучка модель оплати: компанії сплачують лише за фактично використані ресурси хмарної платформи, без необхідності платити за простій чи надлишкові потужності [7];

– мобільність: хмарні платформи полегшують міграцію між постачальниками послуг без втрати даних та простою в роботі [7].

Хмарні обчислення – це доступ на вимогу, через Інтернет, до обчислювальних ресурсів – додатків, серверів (фізичних серверів та віртуальних серверів), зберігання даних, інструментів розробки, мережевих можливостей та багато іншого – розміщених в віддаленому дата-центрі, яким керує провайдер хмарних сервісів [7].

Хмарні обчислення розподіляються на три основних типи послуг, кожен з яких має свої унікальні особливості:

– інфраструктура як послуга (IaaS): цей вид послуги надає користувачам доступ до інфраструктури, включаючи сервери, операційні системи, віртуальні машини, мережі та сховища на основі оренди. Популярними представниками IaaS є Amazon Web Services та Microsoft Azure [8];

– платформа як послуга (PaaS): PaaS використовується для розробки, тестування та обслуговування програмного забезпечення. Він подібний до IaaS, але також надає додаткові інструменти, такі як системи управління базами даних та бізнес-інтелект. Прикладами PaaS є Apprenda та Red Hat OpenShift [9];

– програмне забезпечення як послуга (SaaS): цей вид послуги передбачає, що користувачі отримують доступ до програм через Інтернет за підпискою. Серед популярних прикладів SaaS можна відзначити програми Google та Salesforce [10].

Ці різні типи хмарних обчислень надають користувачам широкий вибір можливостей для використання обчислювальних ресурсів та програмного забезпечення в залежності від їхніх потреб і завдань.

Серед провідних постачальників хмарних сервісів, які надають платформи та інструменти для роботи з великими даними, можна виділити AWS, Microsoft Azure та Google Cloud Platform (рисунок 1.2).

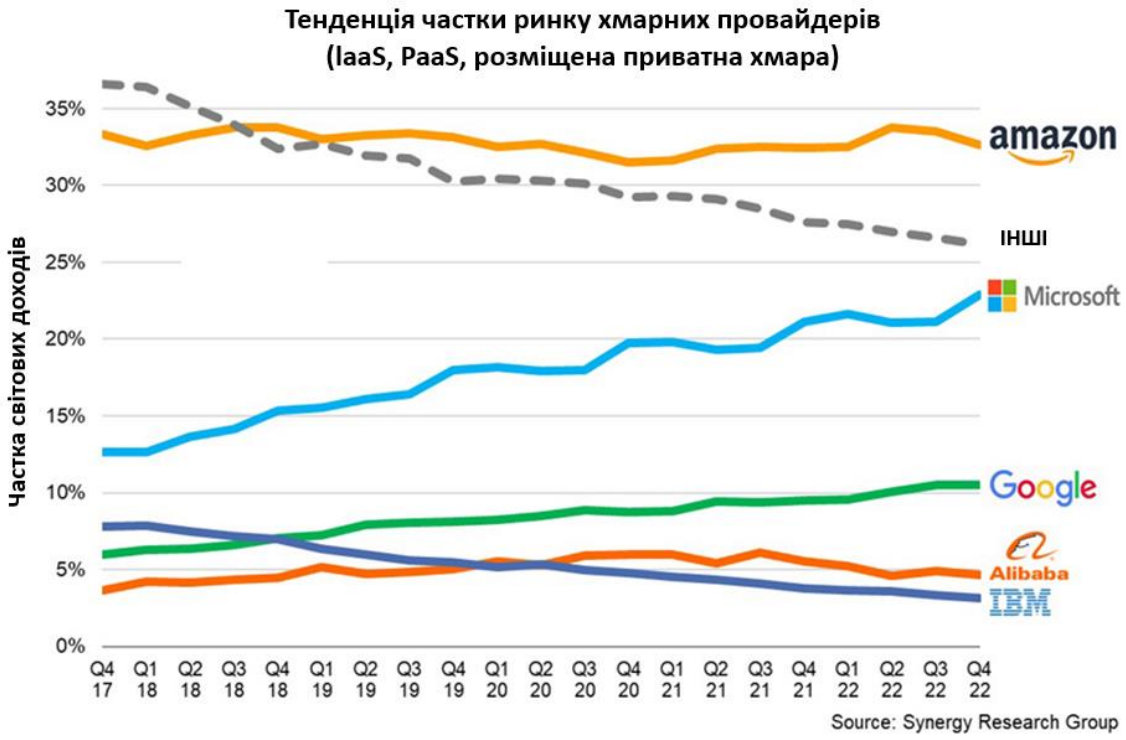

Рисунок 1.2 – Тенденція частки ринку хмарних провайдерів

Розглянемо можливості цих хмарних платформ більш детально.

1.4.1 Платформа Amazon Web Services

Amazon Web Services (AWS) – це один з провідних хмарних сервісів, що надає безліч інфраструктурних та послуг в області обчислення, зберігання, мереж та багато інших [11]. AWS відомий своєю гнучкістю, масштабованістю та надійністю, що робить його важливим гравцем для багатьох компаній та розробників у всьому світі.

Основні характеристики Amazon Web Services включають:

– великий спектр обчислювальних ресурсів, включаючи віртуальні сервери (EC2), контейнери (ECS), та інші;

– широкий вибір засобів для зберігання даних, таких як Amazon S3 (Simple Storage Service) та Amazon EBS (Elastic Block Store);

– мережеві послуги, включаючи Amazon VPC (Virtual Private Cloud) для створення власних віртуальних мереж та Amazon CloudFront для доставки

контенту;

– послуги аналізу даних та машинного навчання, такі як Amazon EMR та Amazon SageMaker.

AWS також надає можливість автоматизації та управління інфраструктурою за допомогою інструментів, таких як AWS CloudFormation і AWS Elastic Beanstalk.

1.4.2 Платформа Microsoft Azure

Microsoft Azure – це хмарна платформа від Microsoft, яка надає широкий спектр сервісів для розгортання та управління додатками і службами через глобальну мережу дата-центрів Microsoft [11].

Основні можливості та переваги Microsoft Azure включають:

– широкий вибір сервісів: IaaS, PaaS, SaaS, сервіси для розробників, бази даних, аналіз даних, Інтернет речей тощо;

– глобальна інфраструктура дата-центрів для забезпечення високої доступності;

– інтеграція з іншими сервісами Microsoft, такими як Office 365, Dynamics 365 тощо;

– гнучкі інструменти для управління та автоматизації розгортання ресурсів, такі як Azure Resource Manager;

– багатий функціонал для безпеки, включаючи шифрування даних, мережевий захист, управління доступом тощо;

– підтримка гібридних сценаріїв використання – поєднання хмарних та локальних сервісі;

– гнучка модель оплати по факту використання ресурсів та сервісів.

Azure широко використовується для створення хмарних додатків, аналізу даних, машинного навчання та багатьох інших сценаріїв.

1.4.3 Платформа Google Cloud Platform

Google Cloud Platform (GCP) – це інша велика гравців на ринку хмарних

послуг, надаючи широкий спектр інфраструктури та сервісів для розробників та підприємств [11]. GCP володіє вражаючими можливостями в області обчислення, аналітики даних, штучного інтелекту, машинного навчання та багатьох інших областях.

Основні характеристики Google Cloud Platform включають:

– широкий вибір обчислювальних ресурсів, включаючи віртуальні машини (VM) та контейнери через Google Kubernetes Engine (GKE);

– потужні сервіси для аналізу та обробки даних, такі як BigQuery та Dataflow;

– інструменти для розробки та тренування моделей штучного інтелекту та машинного навчання за допомогою TensorFlow та AutoML;

– мережеві рішення, такі як Virtual Private Cloud (VPC) та Content Delivery Network (CDN).

Google Cloud також славиться своєю інноваційністю і активним внеском у розвиток відкритих стандартів та ініціатив у сфері хмарних технологій. Компанія постійно впроваджує нові функції та послуги для задоволення різних потреб клієнтів.

Отже, провідні хмарні провайдери надають потужні платформи та інструменти для реалізації рішень Big Data, дозволяючи компаніям обирати оптимальну конфігурацію сервісів для їхніх потреб.

1.4.4 Оцінка та порівняльний аналіз провідних хмарних платформ

Порівняльний аналіз провідних хмарних платформ, таких як Amazon Web Services (AWS), Microsoft Azure і Google Cloud Platform (GCP), є важливим завданням для компаній та підприємств, які шукають ідеальний хмарну платформу для своїх проектів. Всі ці платформи надають широкий спектр можливостей для розгортання IT-інфраструктури, програмних додатків та сервісів в хмарі. Разом з тим, вони мають певні відмінності, які варто враховувати при виборі хмарної платформи. Порівняння сервісів наведено у таблиці 1.1.

Таблиця 1.1 – Аналіз провідних хмарних платформ

Платформа	Опис	Переваги	Недоліки
Amazon Web Services (AWS)	Найбільший постачальник хмарних сервісів, що пропонує широкий вибір інфраструктурних, мережевих, сервісів баз даних, аналітики та інших послуг	Найбільший вибір сервісів. Глобальна інфраструктура дата-центрів. Гнучкі інструменти управління та автоматизації.	Відносно висока вартість. Складність налаштування для новачків.
Microsoft Azure	Другий за величиною постачальник хмарних сервісів з широким спектром можливостей	Інтеграція з продуктами Microsoft Підтримка гібридних сценаріїв Зручні інструменти для .NET розробників	Деякі складнощі перенесення з інших хмар.
Google Cloud Platform (GCP)	Пропонує передові можливості з аналізу даних, машинного навчання та штучного інтелекту	Потужні сервіси аналітики та AI Інноваційні технології Доступна цінова політика для стартапів	Обмежена інтеграція з іншими сервісами. Менша функціональність порівняно з AWS та Azure.

З огляду на потужні хмарні сервіси, які пропонують ці три платформи, Microsoft Azure може бути найкращим вибором для багатьох організацій через його сильну підтримку підприємств, гібридні можливості та глибоку інтеграцію з іншими продуктами Microsoft. Microsoft Azure пропонує гнучкі та надійні сервіси для міграції SQL Server у хмару, оптимальні конфігурації віртуальних машин та засоби автоматизації для розгортання. Також доступна тісна інтеграція з такими сервісами як Azure Synapse Analytics, Azure Data Lake, Azure Machine Learning та Power BI для аналізу великих даних.

1.5 Обґрунтування вибору хмарної платформи на основі аналізу сервісів реляційних баз даних

Для порівняння обрано хмарні платформи AWS та Azure, так як вони є

гігантами в хмарній індустрії. Хмарна платформа Google Cloud на даному етапі не розглядається, оскільки вона має меншу частку ринку хмарних сервісів порівняно з AWS та Azure.

Для реалізації проекту з обробки та аналізу великих даних потрібно обрати оптимальну хмарну платформу та сервіс реляційних баз даних. Розглянуто два основних варіанти – AWS з сервісом RDS та Azure з сервісом SQL Database.

Порівняння їх можливостей представлені у таблиці 1.2

Таблиця 1.2 – Порівняння можливостей Azure SQL Database та AWS RDS

Критерій	Azure SQL Database	AWS RDS
Аналітичні функції	Має широкий спектр можливостей для машинного навчання та аналізу даних безпосередньо в базі даних [12]	Підтримує аналітичні функції, але в меншому обсязі [13]
Інтеграція	Тісно інтегрується з іншими хмарними сервісами Azure [12]	Підтримує інтеграцію, але може бути ускладнена через різницю технологічних стеків [13]
Надійність	Має вбудовані механізми високої доступності та аварійного відновлення [12]	Може вимагати більше зусиль для налаштування [13]
Оновлення версій	Автоматично оновлюється до нових версій [12]	Оновлення потребує ручної ініціації [13]
Масштабування ресурсів	Пропонує простіші механізми для масштабування [12]	Може вимагати додаткових кроків адміністрування [14]
Тарифні плани	Включає повнофункціональний безкоштовний варіант [12]	Безкоштовний тариф може бути обмежений [15]

Отже, з огляду на переваги Azure SQL Database з точки зору вбудованої аналітики, простоти використання та інтеграції з іншими сервісами Azure, саме ця платформа обрана для подальшої розробки та досліджень. Вона найкраще відповідає вимогам проєкту щодо функціональності, продуктивності та

масштабованості при роботі з великими даними. Використання хмарної SQL Database дозволить реалізувати заплановані задачі аналізу даних з мінімальними витратами на налаштування та адміністрування.

1.6 Загальні переваги та недоліки хмарних технологій

Використання хмарних платформ для розгортання та обслуговування додатків та послуг має свої переваги та недоліки, і важливо розглянути їх при виборі оптимального хмарного рішення для бізнесу чи проєкту. Переваги використання хмарних платформ наведено у таблиці 1.3.

Таблиця 1.3 – Загальні переваги та недоліки хмарних технологій

Переваги	Недоліки
Необмежений обсяг пам’яті: ви можете придбати все сховище, яке вам коли-небудь знадобиться.	Залежність від постачальника: ви обмінюєте контроль над інфраструктурою на доступність та надійність постачальника.
Гнучкість та масштабованість: хмарні рішення дозволяють легко масштабувати вашу інфраструктуру в залежності від потреб.	Приватність та безпека даних: зберігання та обробка даних у хмарі може викликати питання щодо приватності та безпеки інформації.
Автоматичне резервне копіювання/відновлення файлів і даних: постачальники хмарних послуг роблять це стандартно 24/7, і користувачам не потрібно про це думати.	Вартість на довгострокову перспективу: хмарні послуги можуть виявитися дорожчими, особливо при інтенсивному використанні ресурсів.
Економія витрат: хмарні платформи дозволяють знизити витрати на IT-інфраструктуру.	Обмеження мережевого доступу: залежно від мережевої доступності, хмарні послуги можуть бути не такими ефективними у віддалених або обмежених регіонах.
Швидкість розгортання: розгортання додатків у хмарі може бути значно швидшим.	

Таким чином, приймаючи рішення про використання хмарної платформи, кожна організація повинна зважити переваги та можливі ризики, виходячи зі специфіки своєї IT-інфраструктури, вимог щодо безпеки даних, наявних навичок персоналу та інших факторів. Грамотне поєднання хмарних та локальних IT-ресурсів дозволяє мінімізувати ризики і максимально використати переваги хмарних обчислень.

1.7 Опис предметної області дослідження

Предметною областю дослідження є процеси зберігання та обробки великих обсягів даних на торгівельних платформах з використанням хмарних технологій і сервісів. Зокрема, проводиться порівняльний аналіз ефективності використання хмарного сервісу бази даних для зберігання і обробки даних з акцентом на аналізі рівнів масштабування.

Торгівельні платформи в сучасних умовах генерують величезні масиви даних в результаті повсякденної діяльності. Це дані про товари, клієнтів, замовлення, логістику, маркетинг та багато іншого. Ефективне зберігання та обробка таких даних з використанням сучасних хмарних технологій дозволяє оптимізувати процеси, підвищити прибутковість та конкурентоспроможність торгівельних платформ.

Однак традиційні підходи до обробки даних не завжди можуть забезпечити необхідну швидкодію, масштабованість та функціональність для вирішення цих завдань. Саме тому актуальним є використання сучасних хмарних технологій та сервісів, які надають гнучку інфраструктуру, високу продуктивність, можливості масштабування та спеціалізовані інструменти для аналізу великих даних.

У рамках дослідження необхідно проаналізувати концепції, принципи та технології, які пов’язані зі зберіганням та обробкою великих обсягів даних на торгівельних платформах. Основний акцент буде зроблено на можливостях використання хмарних сервісів для оптимізації цих процесів, а також на визначенні ефективних підходів до впровадження таких технологій у бізнес-середовище.

Метою дослідження є підвищення ефективності обробки великих обсягів даних на прикладі сфер функціонування сучасних торгівельних платформ (майданчиків) шляхом використання хмарного сервісу роботи з реляційними базами даних Azure SQL Database та на основі впливу рівнів моделі сервісу при реалізації SQL-запитів різної складності.

В контексті дослідження важливо врахувати різноманіття вхідних даних, які включають:

- дані про товари;
- дані замовлень;
- дані філіалів, робітників філіалів та складів;
- дані про клієнтів.

Основною метою є порівняння ефективності зберігання та обробки цих даних з використанням хмарних технологій, зокрема, оцінка продуктивності та масштабованості.

Результати дослідження можуть бути використані для прийняття обґрунтованих управлінських рішень щодо вибору оптимальних рішень для покращення ефективності бізнес-процесів торгівельної платформи.

1.7.1 Актуальність дослідження

У сучасному бізнес-середовищі торгівлі, де конкуренція постійно зростає, ефективне використання технологій для зберігання та обробки великих обсягів даних є надзвичайно важливим для успішної діяльності торгівельних компаній.

Предметом дослідження є процеси зберігання та обробки великих обсягів даних на торгівельних платформах, зокрема порівняння продуктивності та масштабованості хмарних технологій і сервісів у порівнянні з використанням локальних ресурсів.

Електронна торгівля в Україні продовжує набирати обертів, очікуючи зростання вартості продажів до кінця 2023 року (рисунок 1.3) [16]. За прогнозами, ці цифри перевищать рівні пандемійного періоду та підкреслюють стратегічну важливість ефективного аналізу та обробки великих обсягів даних в торгівельному

секторі.

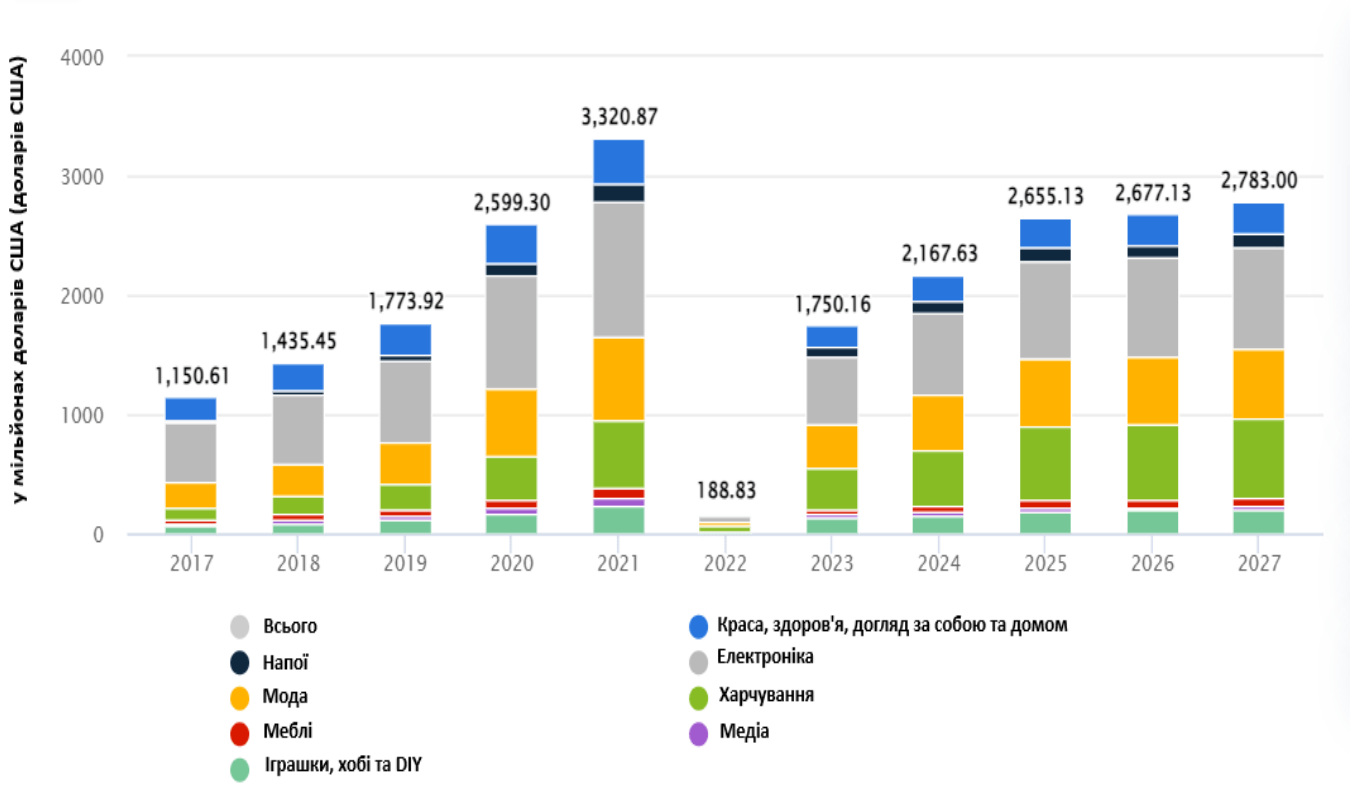

Рисунок 1.3 – Електронна торгівля в Україні по категоріям

Вирішальними факторами успіху в електронній комерції стають швидкість та точність, а здатність аналізувати великі обсяги даних визначає конкурентоспроможність та ефективність бізнесу.

Торгівельні компанії в сучасному віртуальному просторі виконують не лише функцію постачання товарів, а й активно займаються аналізом та реагуванням на споживчі тенденції. Зокрема, дослідження покликане визначити, як технології хмарного зберігання та аналізу великих даних можуть сприяти оптимізації бізнес-процесів.

У відповідь на ці виклики, торгівельні компанії повинні активно впроваджувати інноваційні рішення для ефективної обробки та аналізу великих обсягів даних з метою підтримки своєї конкурентоспроможності на ринку.

В умовах динамічного розвитку технологій, хмарні платформи виявляються важливим інструментом для забезпечення масштабованої та ефективної обробки даних. Застосування хмарних технологій дозволяє торгівельним компаніям значно

зменшити витрати на обладнання та інфраструктуру, забезпечуючи при цьому високу швидкість обробки та доступ до ресурсів в режимі реального часу.

1.7.2 Обґрунтування актуальності застосування хмарних технологій на торгівельних платформах

Використання хмарних технологій на торгівельних платформах стає все більш актуальним, що підтверджується статистикою. Зокрема, 96,9% роздрібних компаній, 94,9% компаній у сфері медіа та розваг та 92,8% компаній у сфері фінансів та банківської справи вже використовують хмарні технології (рисунок 1.4) [17].

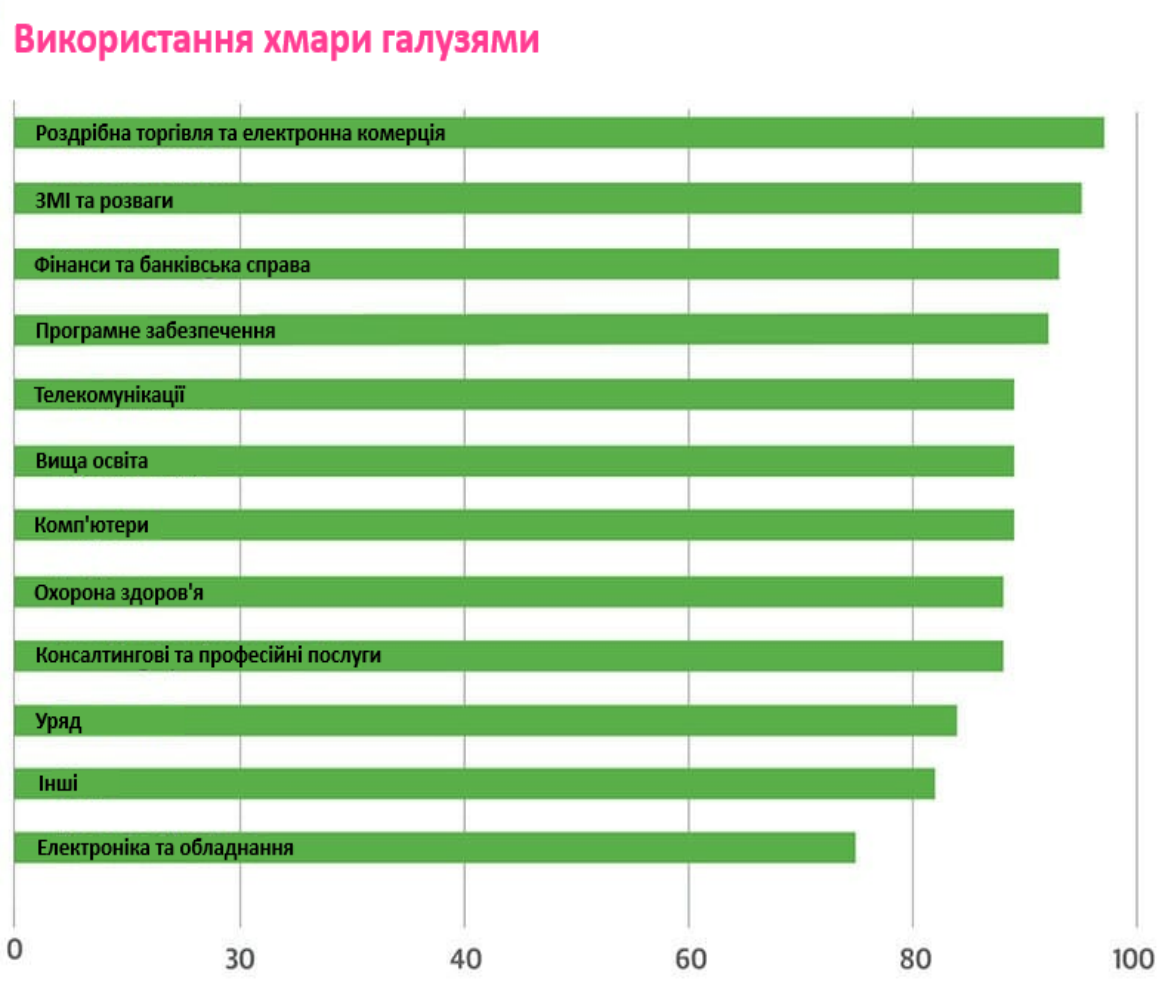

Рисунок 1.4 – Використання хмарних технологій на торгівельних платформах

Важливо взяти до уваги динаміку розвитку ринку в Європі. За останній рік показники свідчать про вражаючий зріст обсягів у сфері роздрібної торгівлі через

хмарні рішення. На 2021 рік обсяг ринку хмари для роздрібної торгівлі в Європі склав вже вражаючі 63 мільярди євро. Прогнози дозволяють очікувати подальший ріст, припускаючи, що цей показник збільшиться майже в 9 разів, до 560 мільярдів євро (рисунок 1.5) [18]. Це відображає не лише експоненційний зріст зацікавленості та використання хмарних технологій у роздрібній торгівлі, але й свідчить про важливість цього тренду, який визначатиме стратегії та конкурентоспроможність торгівельних платформ у найближчому майбутньому.

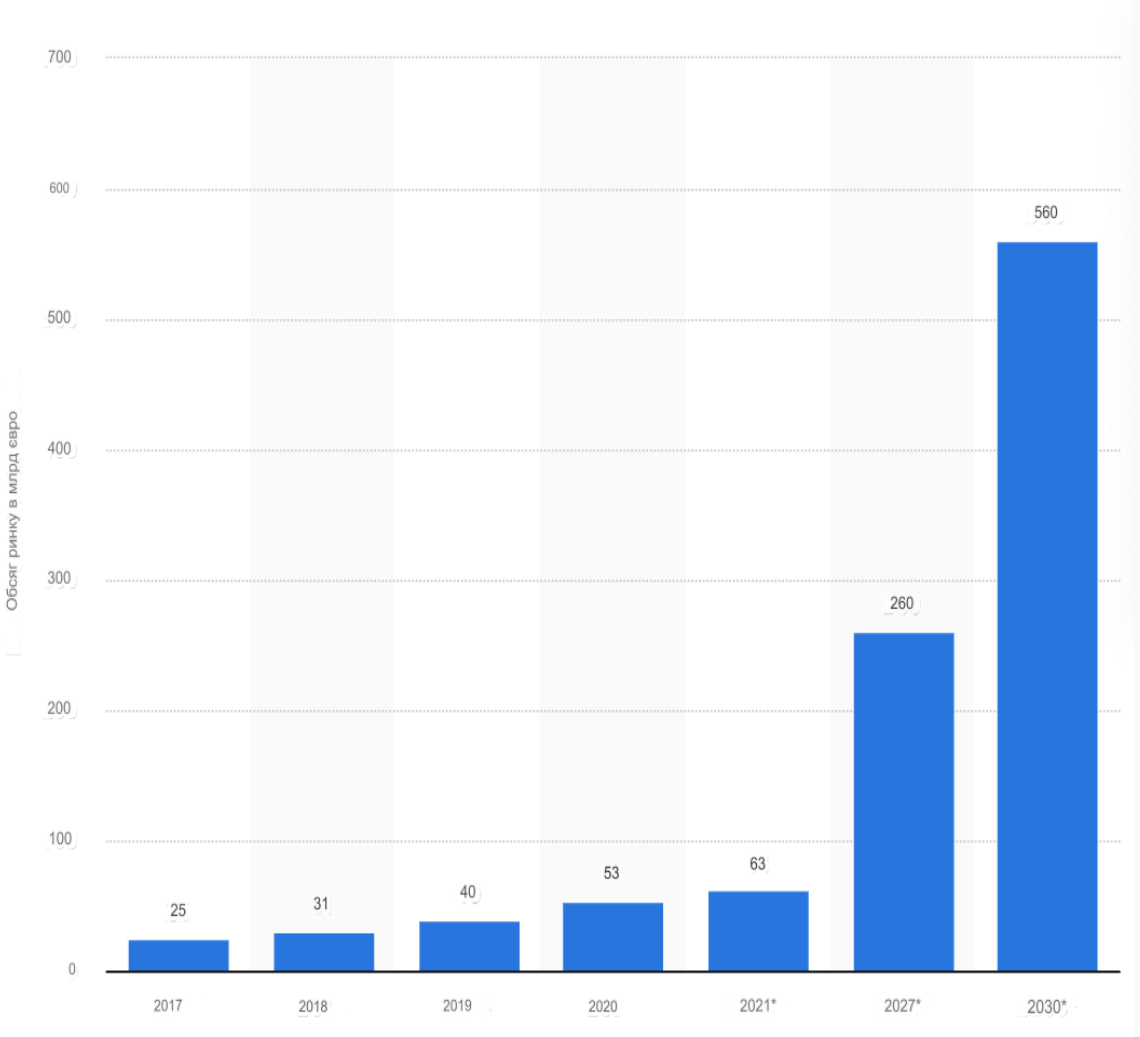

Рисунок 1.5 – Прогноз показнику ринку хмари у Європі

Треба відзначити, що застосування хмарних технологій на торгівельних платформах не є привілегією лише великих корпорацій. Згідно з останніми статистичними даними, 76% великих компаній, 73% середніх та 72% малих підприємств уже використовують хмарні послуги [19]. Це свідчить не лише про широкий розповсюдження використання хмарних технологій, але й підкреслює їхню

доступність та ефективність для компаній різного розміру. Враховуючи, що навіть невеликі та середні підприємства активно впроваджують хмарні рішення, це стає підтвердженням того, що ці технології стають необхідним інструментом для підтримки та розвитку різноманітних торгівельних платформ.

Таким чином, виходячи із цієї статистики, можна зробити висновок, що хмарні технології вже відіграють ключову роль в торгівлі, і ця роль буде тільки посилюватися у майбутньому.

Хмарні технології пропонують низку переваг для торгівельних компаній, таких як зменшення витрат на IT та зберігання, надання оперативного та складського доступу до даних у реальному часі, швидка введення на ринок та гнучкість для масштабування вгору та вниз за необхідності.

Ці переваги роблять хмарні технології ключовим інструментом для роздрібних компаній, які прагнуть оптимізувати свої операції та поліпшити обслуговування клієнтів.

Отже, можна визначити, що хмарні технології вже відіграють важливу роль на торгівельних платформах, і ця роль, ймовірно, буде лише зростати в майбутньому. Це робить вивчення та розуміння хмарних технологій надзвичайно важливим для будь-якої компанії, яка діє в сфері торгівлі.

1.8 Постановка завдань дослідження

Основними завданнями дослідження є:

– провести аналіз сучасного стану роботи з великими даними, зокрема, у сферах діяльності торгівельних платформ. Провести огляд ринкових результатів функціонування сучасних торгівельних платформ, спрямованості їхніх бізнес-процесів та довести доцільність використання технологій щодо обробки великих даних. Визначити основні характеристики щодо потреб в обробки великих обсягів даних. Обґрунтувати вибір хмарної платформи та сервісу для обробки даних;

– розробити моделі даних для обраної предметної області з урахуванням обґрунтованого рівня масштабування щодо обсягів реальних даних; провести тестування бази даних на локальному ресурсі для визначення особливостей

результатів виконання запитів до БД для їх подальшого використання на хмарних сервісах;

– дослідити технології хмарного сервісу щодо зберігання та обробки реляційних баз даних. Розрахувати та проаналізувати метрики продуктивності сервісу Azure SQL Database для різних рівнів обраної моделі та різних режимів оброблення запитів. Визначити їхні переваги, обмеження та пропозиції щодо оптимізації бізнес-процесів;

– розробити стратегії впровадження. Розробити стратегії впровадження технологій хмарного зберігання та обробки даних. Розробити та обґрунтувати ефективні рішення щодо покращення продуктивності системних метрик, а також масштабованості та вартості для підвищення ефективності прийняття управлінських аналітичних рішень;

– практичне впровадження пропонованих рішень**.** Застосувати розроблені стратегії на практиці, впроваджуючи технології сервісу щодо хмарного зберігання та обробки даних. Оцінити та довести їх ефективність при реалізації запитів різної складності для забезпечення розв’язку завдань аналітичного характеру;

– проаналізувати отримані результати та сформулювати рекомендації**.** Здійснити оцінку отриманих результатів щодо впровадження технологій хмарного зберігання та обробки даних. Сформулювати рекомендації для підприємств торгівельних платформ щодо оптимізації бізнес-процесів шляхом інтеграції розроблених методів, моделей та сервісів у практичну діяльність компаній.

Ці завдання спрямовані на створення практичних та ефективних рішень для оптимізації бізнес-процесів та покращення управлінських рішень на торгівельних платформах за допомогою сервісів хмарних технологій.

2 РОЗРОБЛЕННЯ МОДЕЛЕЙ ДАНИХ ТА ЗАПИТІВ, ТЕСТУВАННЯ НА ЛОКАЛЬНОМУ РЕСУРСІ

2.1 Опис предметної області об’єкту управління прототипу торгівельної компанії

Предметна область дослідження охоплює сферу роздрібної торгівлі, а саме – торгівельні компанії. Торгівельні компанії відіграють важливу роль в економіці, забезпечуючи потік товарів від виробників до споживачів. Вони можуть займатися різними видами діяльності, включаючи оптову та роздрібну торгівлю, електронну комерцію.

Торгівельні компанії займаються продажем товарів безпосередньо кінцевим споживачам шляхом продажів або самообслуговування у магазинах. Основні функції торгівельних компаній включають закупівлю товарів від виробників, їх зберігання, продаж та доставку до кінцевих споживачів.

Управління торгівельною компанією вимагає ефективного планування, координації та контролю різних бізнес-процесів, включаючи логістику, маркетинг, продажі, обслуговування клієнтів та інше. Важливим аспектом цього є використання різних інструментів та технологій для збору та обробки великих обсягів даних

Сучасні торгівельні компанії активно впроваджують інформаційні технології для автоматизації своєї діяльності. Оптимізація використання хмарних платформ є ключовим елементом стратегії управління сучасною торгівельною компанією. Це включає в себе вибір найбільш відповідних хмарних сервісів, їх налаштування та моніторинг для забезпечення максимальної продуктивності та ефективності. Таким чином, використання хмарних платформ для обробки великих даних стає важливим елементом стратегії управління сучасною торгівельною компанією.

2.1.1 Аналіз реальної моделі торгівельної компанії

Сучасний ринок електронної комерції насичений різноманітними

торгівельними компаніями, кожна з яких пропонує свої унікальні товари та послуги. Відомі глобальні гравці, такі як Amazon, eBay, Alibaba, Walmart та інші, встановлюють стандарти для індустрії та служать відмінними прикладами успішного використання хмарних платформ для обробки великих даних.

Відомою торгівельною компанією, що активно використовує хмарні технології, є Walmart. Walmart є найбільшим ритейлером у світі за доходом і кількістю працівників [20]. Компанія використовує хмарні сервіси як основу своєї IT-інфраструктури для підтримки глобальних операцій та збору і аналізу даних про покупців [21].

Walmart Inc. є американською транснаціональною корпорацією, що керує мережею гіпермаркетів, великих магазинів і мережею інтернет-магазинів.

Бізнес-модель Walmart заснована на наступних принципах:

– широкий вибір товарів за низькими цінами [22];

– ефективна логістика та управління ланцюгом поставок [23]. Компанія має мережу з 197 дистрибуційних центрів для швидкої доставки товарів в магазини;

– висока автоматизація бізнес-процесів на базі хмарних технологій [24]. Walmart витрачає мільярди доларів на розвиток хмарної IT-інфраструктури для обробки даних;

– багатоканальна комерція, що поєднує офлайн та онлайн канали продажів [25].

Walmart використовує такі категорії:

– продукти: Walmart продає приблизно 160 мільйонів продуктів. Середній магазин Walmart має близько 120 000 товарів різних категорій [26];

– бренди: Walmart пропонує приватні бренди та товари загального бренду [27];

– категорії: Walmart пропонує товари в різних категоріях, включаючи продукти харчування, одяг, взуття, красу, ювелірні вироби, аксесуари, меблі, декор, постіль, ванні кімнати, електроніку, побутову техніку, товари для дому, іграшки, ігри, книги, фільми, музичні інструменти, товари для домашніх тварин, товари для дітей, гігієнічні товари, товари для здоров’я, шкільні та офісні товари,

інструменти, подарунки, садовий центр, аптеку, фотоцентр, спортивні товари та автоцентр [28];

– клієнти: Walmart має близько 265 мільйонів клієнтів щотижня по всьому світу [29];

– склади: Walmart має приблизно 210 складів по всьому світу. Вони також відкрили нові центри виконання наступного покоління [30];

– філії та офіси: Walmart має більше 10 500 магазинів та офісів у 19 країнах та електронні торгові веб-сайти [31];

– працівники: на кінець фінансового року 2023 року в Walmart працювало приблизно 2,1 мільйона співробітників по всьому світу, з яких приблизно 1,6 мільйона співробітників – у США [32].

Walmart є однією з провідних світових компаній в сфері роздрібної торгівлі. Walmart демонструє активне використання передових технологій, включаючи хмарні платформи, для підтримки своїх глобальних операцій.

Ще одним світовим лідером в сфері онлайн торгівлі та використання хмарних сервісів є Amazon.

Amazon є американською багатонаціональною технологічною компанією, яка зосереджується на електронній комерції, хмарних обчисленнях, онлайн-рекламі, цифровому потоці та штучному інтелекті [33]. Amazon намагається бути найбільш орієнтованою на клієнтів компанією на Землі, найкращим роботодавцем та найбезпечнішим місцем роботи на Землі [34].

Amazon, як світовий лідер у сфері електронної комерції, пропонує широкий асортимент товарів і послуг [34]. Amazon має глобальну мережу філіалів та складів, що забезпечують швидку та надійну доставку товарів клієнтам у всьому світі [34]. Ця масштабність та ефективність виконання замовлень демонструють необхідність використання хмарних платформ для обробки великих даних в контексті електронної комерції.

Бізнес-модель Amazon заснована на наступних принципах:

– широкий асортимент товарів та послуг: Amazon пропонує своїм клієнтам широкий вибір товарів і послуг, включаючи товари для дому, електроніку, одяг, книги, музику, фільми та інші 30 категорій [35];

– конкурентоспроможність цін: Amazon намагається забезпечити своїм клієнтам найкращі ціни на ринку, що робить їх пропозиції привабливими для широкого спектра покупців [35];

– ефективність логістики: завдяки своїй глобальній мережі складів, Amazon швидко та надійно доставляє товари своїм клієнтам по всьому світі, що підтверджує їх здатність ефективно виконувати великий обсяг замовлень [36].

Amazon є одним з найбільших гравців в індустрії електронної комерції і має складну структуру, яка включає в себе різні сутності та зв'язки між ними. Amazon включає в себе ряд основних компонентів:

– продукти. Amazon пропонує понад 350 мільйонів продуктових позицій від сотень тисяч брендів [36];

– бренди. Amazon працює з тисячами брендів та виробників, чиї товари представлені на маркетплейсі. По інформації з 2019 року Amazon працює з 158 тисячами брендів [37];

– категорії. Інтернет-магазин Amazon налічує понад 30 основних категорій товарів [38] таких як електроніка, одяг, товари для дому, книги, спорт і т.д.;

– клієнти. Продажі через Amazon в понад 100 країнах світу генерують понад 200 мільйонів активних клієнтських профілів [39]. В системі фіксується історія замовлень, персональні дані, контакти, адреси та рейтинги кожного окремого клієнта;

– замовлення. Завдяки зручності та швидкості оформлення покупок, Amazon обробляє понад 1,6 млн замовлень щодня по всьому світу [40];

– склади. Щоб забезпечити швидку доставку замовлень, Amazon володіє мережею з понад 175 логістичних центрів по всьому світу [41];

– філії та офіси. Для підтримки глобального масштабу діяльності Amazon має понад 100 відділень у різних країнах світу [42];

– працівники. Загальна кількість персоналу Amazon перевищує 1,5 млн осіб [42] – від технічних спеціалістів до адміністративних та логістичних працівників.

Ці сутності та зв'язки між ними формують основу системи Amazon. Наприклад, замовлення, яке робить клієнт, включає в себе вибір продуктів з певних

категорій. Це замовлення потім обробляється працівниками Amazon і відправляється з одного зі складів компанії. Всі ці елементи взаємодіють між собою, щоб забезпечити ефективне обслуговування клієнтів.

З розглянутих компаній, Amazon обрано як прототип для подальшого моделювання та аналізу. Це пов'язано з тим, що Amazon має:

– масштабність бізнесу та обсяг клієнтів. Amazon обслуговує мільйони клієнтів по всьому світу, що підкреслює масштабність його бізнесу. Цей великий обсяг клієнтів генерує величезну кількість даних, які можуть бути оброблені та аналізовані за допомогою хмарних платформ [42];

– різноманітність та кількість товарів. Великий вибір товарів, доступних на Amazon, підкреслює важливість використання хмарних платформ для обробки великих даних [42];

– висока інтенсивність обробки замовлень. Неперервна робота Amazon 24/7 вимагає високої продуктивності всіх компонентів системи, що відповідає вимогам до хмарних платформ для обробки великих даних.

Таким чином, саме Amazon найбільш повно відображає тенденції та можливості сучасних торгівельних компаній у сфері використання хмарних платформ для обробки великих даних. Обраний прототип дозволить провести найбільш детальний та репрезентативний аналіз

2.1.2 Обґрунтування моделі даних

На основі дослідження бізнес-моделі та архітектури системи компанії Amazon як провідного гравця ринку електронної комерції, було розроблено структуру моделі даних типової торгівельної компанії. Ця модель відображає ключові сутності та зв'язки між ними, що є характерними для подібного типу бізнесу. Вона слугуватиме основою для подальшого моделювання та аналізу процесів зберігання та обробки даних з використанням хмарних технологій.

Структура моделі даних торгівельної компанії включає:

1. Продукти – це основна сутність інформаційної системи. Продукт може бути фізичним товаром, цифровим продуктом або послугою. Продукти класифікуються за категоріями, брендами та іншими характеристиками.

2. Категорії – це логічні групи товарів, що мають спільні характеристики. Наприклад, категорія "Електроніка" може включати в себе товари такі як смартфони, ноутбуки, телевізори та іншу електроніку.

3. Клієнти – це фізичні або юридичні особи, які купують товари на торгівельній компанії. Клієнти можуть бути зареєстрованими або не зареєстрованими.

4. Замовлення – це заявка на покупку товарів на торгівельній компанії. Замовлення може бути оформлено клієнтом самостійно через веб-сайт або мобільний додаток торгівельної компанії, а також може бути створено автоматично, наприклад, при підписці на послугу преміум-доставки.

5. Бренди – це виробники або продавці товарів. Бренд може бути представлений одним або декількома продуктами.

6. Склади – це місця зберігання товарів торгівельної компанії. Склади забезпечують швидку та надійну доставку товарів клієнтам.

7. Філіали – це точки продажів торгівельної компанії. Філіали надають клієнтам можливість безпосередньо придбати товари торгівельної компанії.

8. Працівники – це особи, які працюють на торгівельній компанії. Працівники відповідають за різні аспекти роботи компанії, включаючи управління продуктами, логістику, обслуговування клієнтів та інші.

Ці основні компоненти мають такі зв’язки:

– продукти класифікуються за різними категоріями в залежності від їх функціонального призначення, цільової аудиторії тощо, випускаються під унікальним брендом кожної торгової марки і зберігаються на складах до моменту відвантаження клієнтам;

– клієнти є ініціаторами замовлень. Один клієнт може зробити декілька замовлень;

– працівники відповідають за операції з продуктами, замовленнями, логістику та обслуговування клієнтів;

– склади зберігають продукти, які потім відправляються клієнтам відповідно до замовлень;

– замовлення обробляються та виконуються працівниками, вони ініційовані клієнтами для покупки продуктів і виконуються за допомогою продуктів, відібраних на складах;

– у кожному філіалі працює певна кількість працівників, і вони мають свої склади, де зберігаються товари компанії.

ER-діаграма нотації Чена дозволяє візуалізувати зв’язки між компонентами системи, які були вище описані. ER-діаграма, або діаграма сутність-зв’язок – це графічне представлення структури даних для конкретної бази даних. Вона була розроблена Пітером Ченом у 1970-х роках і досі широко застосовується.

ER-діаграма розроблена на основі сутностей торгівельного підприємства, з урахуванням усіх основних зв’язків. На рисунку 2.1 представлено ER-діаграму предметної області даних – прототипу торгівельної компанії, – побудовану за нотацією Чена.

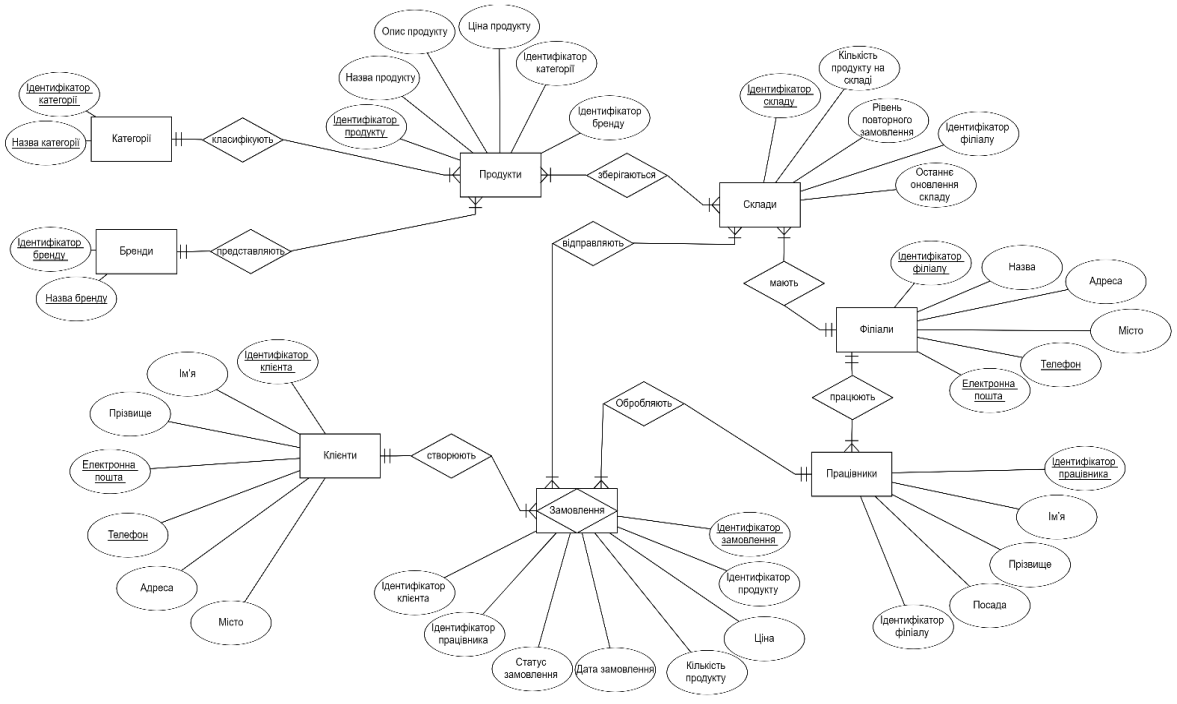

Рисунок 2.1 – ER-діаграма за нотацією Чена

Отже, запропонована структура даних дозволяє адекватно відобразити ключові процеси та взаємозв’язки у торгівельній компанії.

2.1.3 Обґрунтування обсягів даних

Для тестування та оцінки продуктивності системи важливо мати репрезентативні дані, які відображають реальні масштаби існуючого бізнесу. Оскільки за основу взято бізнес-модель компанії Amazon, то для оцінки обсягів даних доцільно проаналізувати публічну інформацію про масштаби діяльності Amazon.

Згідно зі звітністю Amazon за 2022 рік, компанія має понад 200 млн продуктів, понад 200 млн активних користувачів, тисячі постачальників та виробників, та сотні логістичних центрів по всьому світу [43].

Хоча бізнес-модель базується на Amazon, проте на цьому етапі неможливо та недоцільно використовувати дані у таких самих масштабах, як у реальному бізнесі. Це пов'язано з наступними факторами:

– обмеженнями на технічні ресурси. Ресурси хмарної платформи для тестування прототипу можуть бути значно меншими для забезпечення економічної ефективності;

– цілями дослідження. Метою є оцінити можливості масштабування реальних даних, а не максимальна продуктивність. Для цього визначено достатнім використання 1-10% від реальних об’ємів даних;

– оптимізацією часу впровадження розробки. Підготовка та завантаження даних у повному обсязі (наприклад, Amazon) зайняла б дуже багато часу.

Виходячи з масштабів продажів Amazon, для прототипу запропоновано наступні обсяги даних для обраних категорій: Продукти – 2 000 000; Категорії – 36; Бренди – 200 000; Клієнти – 1 000 000; Замовлення – 7 500 000; Склади – 1000; Філіали – 100; Працівники – 250 000.

Таким чином, обрані обсяги даних є компромісом між репрезентативністю, наявними ресурсами та часовими рамками дослідження. Вони дозволять оцінити основні характеристики системи та провести експерименти та зробити аналіз.

Наступним кроком дослідження є проєктування бази даних на основі описаної структури даних та обсягів даних щодо діяльності прототипу торгівельної компанії.

2.2 Проєктування та розроблення БД

Проєктування та розробка бази даних торгівельного підприємства відбувається відповідно до вимог, визначених у попередньому аналізі. Головна мета полягає у створенні структури бази даних та реалізації механізмів для генерування тестових даних заданого обсягу.

База даних (БД) є структурованим набором даних, який зберігається та керується за допомогою програмного забезпечення [44]. БД використовуються для зберігання різноманітних типів даних, таких як текст, числа, зображення та відео.

2.2.1. Вибір системи управління базами даних

В контексті аналізу та оцінювання продуктивності сервісів хмарних платформ для обробки великих даних вибір системи управління базами даних (СУБД) є ключовим етапом у реалізації успішного дослідження.

СУБД – це комплекс програмних засобів, необхідних для створення структури нової бази, її наповнення, редагування вмісту і відображення інформації [45]. Найбільш поширеними СУБД є MySQL, PostgreSQL, Oracle, Microsoft SQL Server.

У рамках цього дослідження використовується Microsoft SQL Server (SQL Server). SQL Server – це система управління реляційними базами даних, розроблена корпорацією Microsoft [46].

SQL Server обрано з таких причин:

– інтеграція з хмарним сервісом Azure: SQL Server є найбільш оптимальним вибором для інтеграції з хмарними платформами, зокрема Azure;

– масштабованість та продуктивність: SQL Server надає широкий спектр можливостей для оптимізації продуктивності та масштабування бази даних;

– безпека та забезпечення відповідності: SQL Server відомий своєю високою рівнем безпеки та можливістю відповідати стандартам забезпечення та відповідності;

– інструменти моніторингу та аналізу продуктивності SQL Server має вбудовані інструменти моніторингу та аналізу продуктивності, які дозволяють вчасно виявляти та вирішувати проблеми з продуктивністю.

Завдяки своїй гнучкості та масштабованості, SQL Server може легко адаптуватися до змінних вимог та обсягів даних, що робить його ідеальним вибором для будь-якого проекту з великими даними. Загалом, вибір SQL Server як СУБД для вашого проекту забезпечить вам надійну, масштабовану та високопродуктивну платформу для обробки ваших великих даних.

2.2.2 Проєктування моделей БД

Проєктування оптимальної схеми бази даних є важливим етапом при розробці для забезпечення ефективного зберігання та швидкого доступу до даних. Від правильно спроектованої структури таблиць та зв’язків між ними безпосередньо залежитиме продуктивність та масштабованість системи.

Метою даного етапу є побудова логічної та фізичної моделей даних БД. Для цього сутності та зв’язки між ними (див. рис. 2.1) перетворюються відповідно до правил побудови реляційних БД.

Зокрема, кожна бізнес-сутність представляється у окрему таблицю з унікальним первинним ключем. Атрибути цих сутностей стають колонками в таблицях. Зв'язки між сутностями реалізуються за допомогою зовнішніх ключів.

Виходячи з попереднього аналізу предметної області, було виділено такі ключові сутності для моделювання структури бази даних: Продукти, Категорії продуктів, Бренди, Клієнти, Замовлення, Склади, Філіали та Працівники. Ці сутності найбільш повно описують основні об'єкти та процеси, задіяні у діяльності торгівельної компанії.

Для підвищення гнучкості моделі та оптимізації обробки замовлень було прийнято рішення додати ще одну сутність – «Деталі замовлень». Ця сутність дозволить зберігати інформацію саме про кожну позицію в замовленні окремо, що спростить обробку та аналіз даних.

Решта сутностей найбільш повно описують основні об'єкти та процеси, задіяні у діяльності торгівельної компанії. Для побудови оптимальної логічної моделі

даних треба більш детальніше описати вищенаведені сутності, включаючи їх ключові атрибути та характеристики.

Опис сутностей та їх характеристики наведені у таблиці 2.1.

Таблиця 2.1 – Типи сутностей

Ім'я сутності	Опис	Характеристика
Brands (Бренди)	Представляють різні марки товарів, що продаються.	Пов'язує товари з їх виробниками. Дозволяє аналізувати популярність окремих брендів.
Categories (Категорії)	Класифікують товари за різними групами.	Групує товари за певними ознаками. Надає змогу аналізу продажів у розрізі категорій.
Products (Продукти)	Різні товари, що продаються. Вони можуть належати до різних брендів та категорій.	Зберігає дані про асортимент товарів.
Customers (Клієнти)	Особи або організації, які купують товари.	Накопичує дані клієнтської бази для проведення аналізу та маркетингу.
Branches (Філіали)	Місця, де знаходяться магазини або офіси компанії.	Фіксує розташування точок продаж для оцінки діяльності.
Employees (Працівники)	Люди, які працюють в компанії. Вони можуть мати різні ролі та обов’язки.	Накопичує інформацію про кадровий склад для аналізу та HR-звітності.
Warehouses (Склади)	Місця, де зберігаються товари до їх продажу.	Фіксує поточний стан запасів для оптимізації логістики та постачання.
Orders (Замовлення)	Замовлення від клієнтів на покупку товарів.	Накопичує дані для аналізу попиту та продажів.
Orders Details (Деталі замовлень)	Специфікації окремих замовлень, що включають товари та їх кількість.	Дає детальну інформацію замовлень для поглибленого аналізу.

Розглянемо атрибути цих сутностей. Атрибути представляють конкретні властивості або характеристики сутності, які підлягають збереженню у відповідній таблиці бази даних. Отже, маємо такі атрибути для сутностей:

- Бренди – Ідентифікатор бренду; Назва бренду;
- Категорії – Ідентифікатор категорії Назва категорії;

– Продукти – Ідентифікатор продукту Назва продукту Опис продукту Ціна продукту Ідентифікатор категорії Ідентифікатор бренду;

– Клієнти –Ідентифікатор клієнта; Ім'я; Прізвище; Електронна пошта; Телефон; Місто; Адреса;

– Філії – Ідентифікатор філіалу; Назва філії; Адреса філії; Місто філії; Телефон філії; Електронна пошта філії;

– Працівники – Ідентифікатор працівника; Ім'я; Прізвище; Посада; Ідентифікатор філіалу;

– Склади – Ідентифікатор складу; Ідентифікатор продукту; Кількість продукту; Рівень поповнення запасу; Дата останнього поповнення запасу; Ідентифікатор філіалу;

– Замовлення – Ідентифікатор замовлення; Ціна за замовлення; Статус замовлення; Дата замовлення; Дата виконання замовлення; Ідентифікатор клієнта; Ідентифікатор працівника;

– Деталі – Ідентифікатор замовлення замовлень; Ідентифікатор продукту; Кількість; Ціна.

Отже, на основі визначених сутностей та їх атрибутів можна встановити логічні зв’язки між ними в моделі даних:

– продукти пов'язані з категоріями – кожен продукт належить до певної категорії. Цей зв’язок необхідний, оскільки кожен продукт має належати до певної категорії. Це дозволяє організувати продукти за групами та полегшує їх пошук та аналіз;

– продукти пов'язані з брендами – кожен продукт випускається під певним брендом. Цей зв’язок важливий, оскільки кожен продукт випускається під певним брендом. Це дозволяє відслідковувати, які продукти випускає кожен бренд;

– замовлення пов'язані з клієнтами – кожне замовлення створюється конкретним клієнтом. Цей зв’язок необхідний, оскільки кожне замовлення робить певний клієнт. Це дозволяє відслідковувати, хто зробив замовлення;

– замовлення пов'язані з працівниками – кожне замовлення обробляється певним працівником. Цей зв’язок важливий, оскільки кожне замовлення обробляє певний працівник. Це дозволяє відслідковувати, хто обробляє замовлення;

– замовлення пов'язані з деталями замовлень – одне замовлення може містити багато деталей замовлення. Цей зв’язок необхідний, оскільки одне замовлення може містити багато деталей замовлення. Це дозволяє відслідковувати, які продукти входять до певного замовлення;

– деталі замовлень пов'язані з продуктами на складах – кожна деталь замовлення стосується конкретного продукту, який наявний на певному складі. Цей зв’язок важливий, оскільки кожна деталь замовлення стосується конкретного продукту, який наявний на певному складі. Це дозволяє відслідковувати, які продукти на складі входять до певного замовлення;

– працівники пов’язані з філіями – кожен працівник працює в певному філіалі. Цей зв’язок необхідний, оскільки кожен працівник працює в певному філіалі. Це дозволяє відслідковувати, в якому філіалі працює певний працівник;

– склади пов’язані з філіями – кожен склад належить конкретній філії. Цей зв’язок важливий, оскільки кожен склад належить до певної філії. Це дозволяє відслідковувати, до якої філії належить певний склад;

– склади поставляють з продуктами – на кожному складі зберігається інформація про наявні продукти. Цей зв’язок необхідний, оскільки на кожному складі зберігається інформація про наявні продукти. Це дозволяє відслідковувати, які продукти зберігаються на певному складі.

У процесі створення бази даних одним з ключових етапів є формалізація логічної моделі даних. Цей процес включає в себе визначення основних сутностей та зав’язків між ними. Кожен тип зв’язку між сутностями потребує детального опису, що допомагає у подальшій реалізації цих зав’язків у структурі бази даних.

Важливим аспектом є коротке та зрозуміле формулювання суті кожного зв’язку. Також необхідно визначити кардинальність зв’язку, яка може бути один-до-одного, один-до-багатьох тощо. Це дозволяє чітко розуміти, як дані пов’язані між собою.

У таблиці 2.2 представлено перелік зав’язків між основними сутностями моделі даних із зазначенням типу зв'язку та його короткого текстового опису.

Така формалізація допоможе краще зрозуміти взаємозв'язки між даними, що у свою чергу спростить подальше проєктування бази даних.

Таблиця 2.2 – Типи зв'язків

Зв'язок	Опис зв'язку	Тип зв'язку
Продукти – Категорії	Категорії класифікують продукти	Багато-до-одного
Продукти – Бренди	Бренди випускають продукти	Багато-до-одного
Замовлення – Клієнти	Клієнти роблять замовлення	Багато-до-одного
Замовлення – Працівники	Працівники обробляють замовлення	Багато-до-одного
Замовлення – Деталі замовлень	Замовлення складаються з деталей замовлень	Один-до-багатьох
Деталі замовлень – Продукти на складах	Деталі замовлень поставляють продукти з складів	Багато-до-багатьох
Працівники – Філіали	Працівники працюють у філіях	Багато-до-одного
Склади – Філіали	Склади належать філіям	Багато-до-одного
Склади – Продукти	На складах зберігаються продукти	Багато-до-багатьох

На основі проведеного аналізу та визначення сутностей, їх атрибутів і зв'язків можна побудувати концептуальну модель даних (КМД) для торгівельного підприємства.

Концептуальна модель призначена для формалізованого подання структури даних на концептуальному рівні, незалежно від подальшої реалізації бази даних. Вона дозволяє абстрагуватися від технічних деталей і зосередитись на змістовному представленні предметної області за допомогою ключових сутностей, атрибутів і зв'язків.

Побудова КМД на цьому етапі проєктування бази даних є важливою з наступних причин:

– надає загальне уявлення про структуру даних та взаємозв'язки між ними;

– дозволяє перевірити адекватність та повноту відображення предметної області;

– спрощує подальше проєктування логічної та фізичної моделей даних;

– є документацією та дозволяє оцінити якість розробленої моделі.

Концептуальна модель даних представлена на рисунку 2.2.

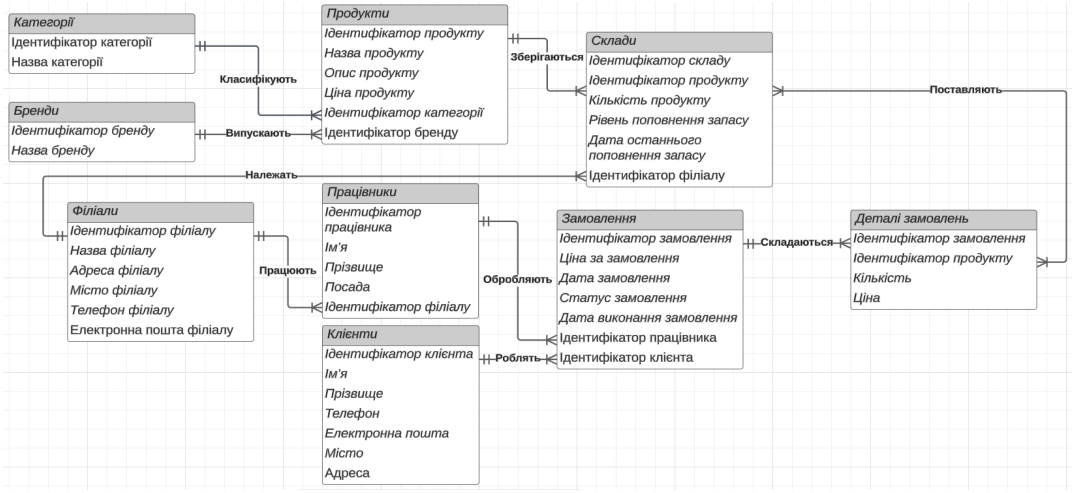

Рисунок 2.2 – Концептуальна модель даних

Після побудови концептуальної моделі даних наступним кроком є трансформація її в логічну та фізичну моделі бази даних.

Логічна модель даних базується на концептуальній, але вже містить деякі технічні деталі реалізації бази даних. Зокрема, в ній описуються сутності у вигляді таблиць з атрибутами-колонками та визначаються первинні і зовнішні ключі. Проте ця модель залишається абстрактною і не прив'язана до конкретної СУБД.

Графічне зображення логічної моделі бази даних представлено на рисунку 2.3. Модель візуалізує всі сутності та зв'язки між ними у вигляді таблиць та ключів.

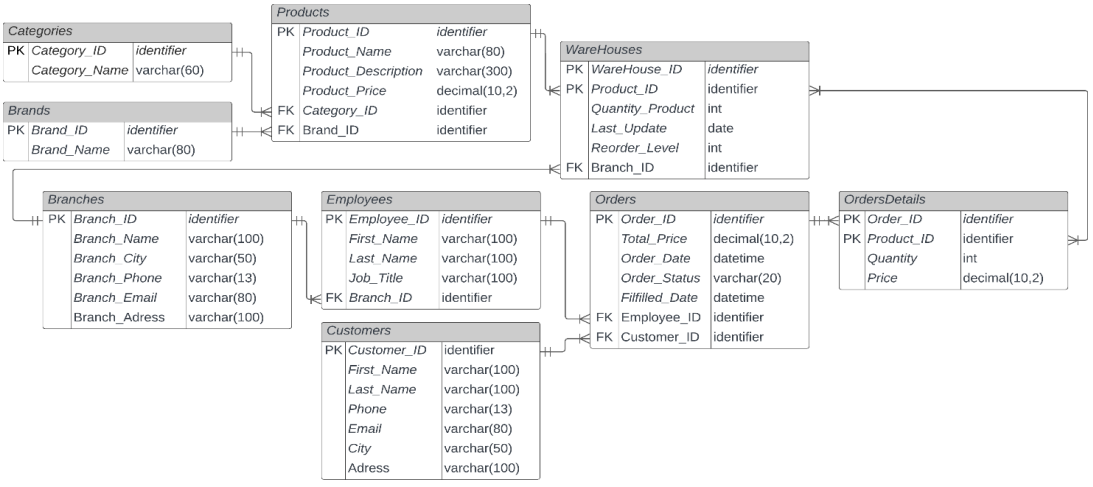

Рисунок 2.3 – Логічна модель бази даних

Метою фізичного проєктування є створення бази даних в обраній системі управління базами даних на основі логічної схеми. В якості СУБД було обрано Microsoft SQL Server зважаючи на його можливості для ефективної роботи з великими даними.

Оскільки в якості СУБД було обрано Microsoft SQL Server, то фізична реалізація схеми бази даних матиме деякі відмінності порівняно з логічною моделлю.

Зокрема, SQL Server має такі особливості щодо типів даних, які необхідно врахувати:

1. Для текстових стовпців замість універсального типу varchar краще використовувати nvarchar. Цей тип даних дозволяє зберігати текст як унікод, в тому числі дані кирилицею, на відміну від звичайного varchar. Наприклад: Product_Name nvarchar(100).

2. Для грошових значень замість універсального decimal краще використати вбудований тип money. Він оптимізований в SQL Server для зберігання грошових сум. Наприклад: Product_Price money.

3. Замість універсального логічного типу Identifier для первинних ключів слід використовувати вбудований тип uniqueidentifier. Він генерує випадкові унікальні ідентифікатори для кожного запису. Наприклад: Product_ID uniqueidentifier PRIMARY KEY.

Окрім явних типів даних в SQL Server також є можливість задати обмеження цілісності (NULL/NOT NULL) для контролю введення даних в стовпці.

Отже, хоча логічна та фізична моделі є дуже схожими за структурою, на етапі фізичної реалізації схеми в конкретній СУБД необхідно врахувати особливості обраної платформи та задіяти специфічні вбудовані механізми для ефективної роботи з даними. Фізична модель бази даних представлена на рисунку 2.4, усі обмеження, описані у таблиці 2.4, залишаються незмінними.

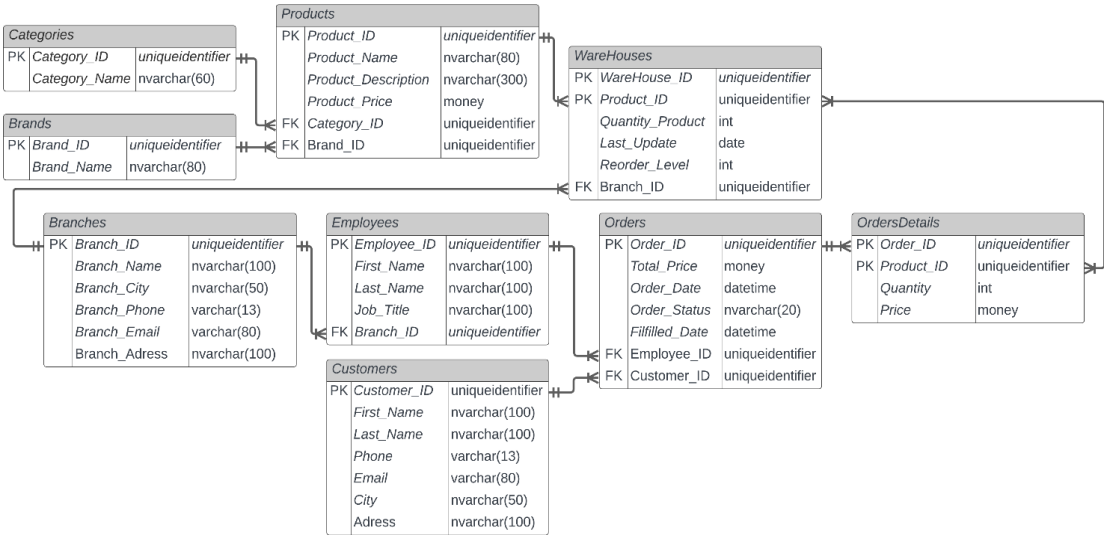

Рисунок 2.4 – Фізична модель бази даних

Таким чином, спроєктовано оптимізовану схему бази даних. Запропонована структура дозволить ефективно зберігати і обробляти дані про асортимент, клієнтів, замовлення тощо для цілей автоматизації та отримання аналітичних рішень для управління торгівельною компанією.

Практична реалізація схеми в Microsoft SQL Server з використанням специфічних механізмів СУБД забезпечить готовність бази даних до швидкого наповнення тестовими та реальними даними.

Наступним логічним кроком є генерування тестових даних за допомогою SQL-скриптів для перевірки працездатності спроєктованої бази на практиці та її подальше наповнення та використання.

2.3 Генерація тестових даних

Для практичної реалізації запропонованого проєкту та подальшого тестування під час застосування хмарних платформ необхідним є наповнення її реалістичними тестовими даними та й у достатньо великому обсязі.

Генерація таких даних може бути виконана за допомогою SQL-скриптів – наборів команд мовою SQL, котрі дозволяють автоматизувати масове створення та модифікацію записів у таблицях. З їх допомогою можна сформувати дані по ключовим сутностям: товари, клієнти, замовлення тощо.

Такий підхід дасть змогу всебічно протестувати спроектовану базу даних, а також проаналізувати ефективність хмарних платформ на реальних сценаріях аналітичної обробки даних.

2.3.1 Розробка скриптів генерації даних

Для генерації тестових даних було розроблено низку SQL-скриптів для ключових сутностей моделі даних: бренди, категорії товарів, продукти, клієнти, філії, працівники, склади, замовлення та деталі замовлень.

Скрипти використовують засоби Transact-SQL (T-SQL), мови програмування і запитів, розробленої Microsoft, такі як цикли, тимчасові таблиці, функції генерації випадкових чисел, об'єднання таблиць для створення унікальних комбінацій даних. T-SQL є розширенням мови SQL і використовується для роботи з базами даних у системах SQL Server [47]. Це дозволяє ефективно генерувати великі обсяги даних та виконувати різноманітні операції обробки даних у базі даних SQL Server.

Далі наведено скрипти, створені для генерування даних по основних сутностях.

Листинг скрипту для таблиці Бренди (Brands) представлений у Додатку Б.

Цей скрипт T-SQL створює тимчасову таблицю в пам’яті з полями Brand_ID та Brand_Name. Brand_ID є унікальним ідентифікатором, а Brand_Name – рядком довжиною 50 символів.

Після цього використовується цикл WHILE, щоб вставити 200 000 записів в таблицю @Brands. Кожен запис складається з унікального ідентифікатора та назви бренду, яка формується шляхом додавання номеру поточного циклу до рядка «Бренд». Після завершення циклу WHILE всі дані з таблиці @Brands вставляються в таблицю Brands.

Скрипт оптимізовано для швидшої генерації даних за рахунок наступних факторів:

– використовується тимчасова таблиця для зберігання даних. Це дозволяє уникнути додаткових операцій зчитування та запису даних у основну таблицю;

– використання циклу WHILE дозволяє генерувати дані у послідовному порядку. Це призводить до більш ефективного використання ресурсів сервера.

Використання циклу WHILE дозволяє генерувати дані у послідовному порядку. Це призводить до більш ефективного використання ресурсів сервера. Якщо дані генерувалися б випадковим чином, то серверу було б потрібно виконувати додаткові операції з пошуку місця для запису даних.

Для таблиці «Categories» використовується пряма вставка значень (див. Додаток Б стор. 206). Цей підхід є простим та ефективним для невеликих наборів даних, як у цьому випадку, де використовується 36 категорій (див. розділ 2.1).

Цей підхід є простим та ефективним для невеликих наборів даних, але може бути неефективним для великих наборів даних через велику кількість операцій вставки. Для великих наборів даних краще використовувати підхід, подібний до того, що було використано для таблиці «Brands» – з використанням тимчасових таблиць та циклів для генерації даних.

Для таблиці «Products» скрипт генерує 2 мільйони записів (див. Додаток Б стор. 206).

Спочатку, у скрипті визначається змінна @MaxProducts для задання максимальної кількості продуктів для генерування. Далі створюються дві тимчасові таблиці з випадковим переставленням ідентифікаторів категорій і брендів для подальшого використання при генеруванні продуктів. Це дозволяє отримати унікальні поєднання категорій і брендів.

Після цього за допомогою конструкції SELECT TOP створюється ще одна тимчасова таблиця з потрібною кількістю рядків, що містять випадкові значення для ціни, посилання на категорію та бренд. Тут використовуються функції генерації випадкових чисел та контрольних сум CHECKSUM і NEWID для отримання унікальних комбінацій.

На останньому кроці виконується INSERT даних у основну таблицю Products з використанням тимчасової таблиці як джерела та генеруванням унікальних ідентифікаторів і рядків з назвою та описом продуктів на базі номера рядка.

Оптимізація для швидкості генерації здійснена за рахунок наступних факторів:

– використання тимчасових таблиць: Зберігає проміжні дані та зменшує завантаження на основну таблицю;

– випадкових індексів замість повного сканування: Отримання потрібних записів з Categories та Brands відбувається швидше за випадковим індексом, ніж за повним скануванням таблиць;

– функція ROW_NUMBER() та TOP для певного порядку: Визначення порядку записів без впливу на випадковість значень;

– функція RAND(CHECKSUM(NEWID())) для більш випадкових чисел: Покращує випадковість порівняно з простим RAND().

Для генерації даних в таблиці «Customers» використовується скрипт, листинг якого представлений у Додатку Б.

Генерація випадкових чисел розпочинається створенням тимчасової таблиці @Numbers для зберігання цілих чисел, яку потім заповнюємо 1 мільйоном послідовно зростаючих значень за допомогою циклу WHILE. Змінна @MaxNumber визначає кількість генерованих чисел. Далі, для виборки випадкових індексів для імен та прізвищ, використовується тимчасова таблиця #TempCustomers з випадковими індексами, використовуючи функцію TOP та ROW_NUMBER() OVER (ORDER BY (SELECT NULL)), щоб випадково вибрати @MaxCustomers записів із @Numbers та отримати послідовні номери запису («RowNum»).

Після цього на етапі підготовки випадкових назв електронної пошти та телефонів створюються віртуальні подання даних UniqueNames та UniqueLastNames, які отримують випадкові номери для імен та прізвищ з ExistingNames за допомогою NEWID(). Основний запит з'єднує #TempCustomers з цими поданнями за відповідними номерами, витягуючи випадкові імена та прізвища. Формуються випадкові адреси електронної пошти та телефонні номери, а також генерується місто проживання.

Сформовані дані вставляються в основну таблицю Customers, присвоюючи значення для CustomerID, FirstName, LastName, Email, Phone та City. На завершальному етапі видаляється тимчасова таблиця #TempCustomers, завершуючи процес генерації та збереження даних клієнтів.

Оптимізація для швидкості генерації за рахунок наступних факторів:

– використання тимчасових таблиць: Зберігає проміжні результати та зменшує навантаження на основну таблицю;

– випадкові індекси для вибору імен та прізвищ: Швидший вибір порівняно із повним скануванням таблиці ExistingNames;

– функція ROW_NUMBER() та TOP для певного порядку: Визначення порядку без впливу на випадковість вибору;

– віртуальні подання даних (With) для повторного використання даних: Повторне залучення випадкових назв та прізвищ без повторного сканування ExistingNames;

– обчислення даних у запиті: Уникає додаткових операцій вставки для електронної пошти, телефону та міста, покращуючи продуктивність.

Скрипт для генерації тестових даних для таблиці «Branches» представлений у Додатку Б. Він генерує 100 записів у таблиці Branches з випадковими назвами, адресами, містами, телефонами та електронною поштою.

Для кожного циклу вставляються дані в таблицю Branches з унікальними значеннями для кожного поля. BranchID генерується за допомогою функції NEWID(). BranchName складається з тексту «Branch » та значення змінної @CounterBranches. BranchAdress формується з тексту «Address » та значення змінної @CounterBranches. BranchCity має формат «Місто » разом із значенням змінної @CityCounterBranches. BranchPhone включає код України «+380», випадкове число від 0 до 999999999 та домен @example.com. BranchEmail створюється із тексту «email», значення змінної @CounterBranches та домену @example.com. Кожен цикл додає унікальний філіал з відповідними параметрами до таблиці Branches.

Оптимізація для швидкості реалізована за допомогою:

– використання циклу WHILE: Цей спосіб виконання дозволяє генерувати дані у послідовному порядку, що покращує продуктивність;

– обчислення даних у запиті: Використання функцій CAST() та RIGHT() для обчислення значень адреси, телефону та електронної пошти покращує продуктивність, оскільки уникає додаткових операцій вставки.

Генерація даних до таблиці «Employees» подана у Додатку Б. Цей скрипт генерує 250 000 записів у таблицю з випадковими іменами, прізвищами, посадами та назвами філій з таблиці «Branches».

Спочатку визначаються змінні для керування процесом: лічильник @i,

максимальна кількість @max, змінні для збереження значень полів таблиці @firstName, @lastName, @jobTitle, @branchID.

Також визначається змінна @nameMax для отримання максимального ідентифікатора з таблиці існуючих імен EmployeesNames.

Створюється тимчасова таблиця @Employees для збереження згенерованих даних співробітників.

Далі йде цикл WHILE, який виконується від 1 до 250 000 ітерацій. На кожній ітерації:

- з таблиці EmployeesNames вибирається випадкове ім'я та прізвище;
- генерується випадкова посада Менеджер/Асистент з ймовірністю 50/50;
- обирається випадковий ідентифікатор філії з таблиці Branches;
- формується запис для вставки у тимчасову таблицю @Employees.

Після циклу дані з @Employees вставляються в основну таблицю Employees.

Для оптимізації для швидкості використовується:

– використання циклу WHILE: Цей спосіб виконання дозволяє генерувати дані у послідовному порядку, що покращує продуктивність;

– використання таблиці EmployeesNames: Це дозволяє генерувати імена та прізвища співробітників швидше, ніж якщо б вони генерувалися випадковим чином;

– випадковий вибір посади: Замість того, щоб використовувати цикл для генерації всіх можливих посад, цей скрипт використовує оператора CASE для випадкового вибору однієї з двох посад. Це дозволяє генерувати дані швидше;

– випадковий вибір номера філіала: Замість того, щоб сканувати всю таблицю Branches, цей скрипт використовує функцію TOP() для вибору одного запису з таблиці. Це дозволяє генерувати дані швидше.

Скрипт генерації тестових даних до таблиці «Warehouses» представлений у Додатку Б.

Спочатку створюються дві тимчасові таблиці – #TempProducts та #TempBranches. В них вставляються всі існуючі записи з таблиць Products та Branches відповідно, з доданим автоінкрементним первинним ключем RowNumber. Це зроблено для оптимізації – замість постійних звернень до вихідних таблиць при

генерації, ми матимемо локальні тимчасові таблиці з індексованим доступом по первинному ключу. Далі починається транзакція для атомарного вставлення даних. Визначаються змінні @MaxProducts та @MaxBranches – кількість унікальних товарів та філій відповідно. Це зроблено також для оптимізації – розраховувати COUNT(*) при кожній ітерації циклу було б затратніше. Починається цикл генерації зі змінною-лічильником @CounterWarehouses та умовою кількості ітерацій. На кожній ітерації генерується новий унікальний ключ складу @WarehouseID = NEWID(). Далі вибирається випадковий існуючий BranchID з тимчасової таблиці за допомогою функції CHECKSUM та оператора % для обмеження максимального значення. Починається вкладений цикл генерації продуктів на складі з лічильником @CounterProducts. Кількість ітерацій тут також випадкова за допомогою CHECKSUM та %. На кожній ітерації обирається випадковий Product_ID з тимчасової таблиці та генеруються інші випадкові атрибути складу за допомогою RAND() та функцій дати/часу. Дані вставляються в таблицю Warehouses. Після завершення вкладених циклів, виконується коміт транзакції та видалення тимчасових таблиць.

Використання транзакцій при масовому вставленні або модифікації даних може суттєво пришвидшити процес, але це залежить від конкретного випадку. Ось декілька з таких причин:

1. Без транзакцій кожна операція вставки/оновлення виконується окремо, що передбачає додаткові накладні витрати на відкриття/закриття з'єднання, логування змін, оновлення індексів тощо. А в транзакції все відбувається в межах однієї сесії.

2. Транзакції дозволяють уникнути часткових оновлень. Якщо в процесі вставки відбувається помилка - відкочуються усі зміни в транзакції.

3. Менша кількість точок блокування даних. Блокується вся транзакція, а не кожна операція окремо.

Однак є і такі недоліки:

– підвищується навантаження на журнал транзакцій;

– довша транзакція – більший ризик deadlock (це ситуація в системах, де кожен із групи процесів очікує на подію, яку може викликати лише інший процес з цієї групи [48]).

Отже, в кожному конкретному випадку потрібно оцінити переваги та недоліки. Наприклад, при вставці в окрему таблицю без складних зв'язків транзакція може бути зайвою, а при масових зв'язаних операціях – критично важливою.

Для швидкості зроблена така оптимізація:

– тимчасові таблиці – зменшують кількість сканувань великих таблиць Products та Branches;

– випадковий вибір із тимчасових таблиць – забезпечує рівномірний вибір продуктів та філіалів без необхідності повного сканування великих таблиць;

– транзакція – забезпечує швидке відновлення у разі помилки, обмежуючи необхідність повторного виконання вставки даних.

Генерація для таблиці «Orders» представлена у додатку Б. Скрипт генерує 3 мільйони (3 000 000) записів у таблиці Orders з випадковими датами замовлення, статусами, клієнтами та співробітниками.

Спочатку створюються дві тимчасові таблиці з існуючими клієнтами та працівниками, оптимізовані для швидкого доступу за допомогою первинного ключа RowNumber. Починається транзакція для атомарного вставлення даних. Визначаються константи – загальна кількість клієнтів та працівників для обмеження максимальних значень при генерації. Це пришвидшує скрипт, уникаючи лішніх підрахунків на кожній ітерації. Запускається цикл генерації замовлень з лічильником @CounterOrders та умовою кількості ітерацій. Далі обираються випадкові існуючі CustomerID та EmployeeID з тимчасових таблиць з використанням функцій CHECKSUM, RAND та оператора %. Також генерується випадковий Order_Status та дата Order_Date у минулому. Після заповнення полів виконується вставка даних у таблицю Orders. Ще окремим запитом ON UPDATE генерується випадкова дата виконання замовлення для всіх «Завершених». Після циклу – коміт транзакції та видалення тимчасових таблиць.

Оптимізація для швидкості:

– тимчасові таблиці – зменшують кількість сканувань великих таблиць Customers та Employees;

– випадковий вибір із тимчасових таблиць – забезпечує рівномірний вибір клієнтів та співробітників без необхідності повного сканування великих таблиць;

– транзакція – забезпечує швидке відновлення у разі помилки, обмежуючи необхідність повторного виконання вставки даних.

Скрипт генерації даних для таблиці «OrdersDetails» представлений у Додатку Б. Цей скрипт генерує деталі замовлень (кількість та ціну продуктів) для існуючих записів таблиці Orders. Він враховує як випадкові замовлення з новими продуктами, так і додаткові деталі для наявних замовлень.

Спочатку створюються дві тимчасові таблиці: #TempProducts та #TempOrders. Вони використовуються для зберігання інформації про продукти та замовлення відповідно. Дані з таблиць Products та Orders копіюються в ці тимчасові таблиці з додаванням номера рядка як первинного ключа. Починається транзакція. Визначаються максимальні значення @MaxProducts та @TotalOrders, які представляють кількість продуктів та замовлень відповідно. Запускається цикл WHILE, який виконується @TotalOrders разів. Кожна ітерація циклу створює новий запис в таблиці OrdersDetails. На кожній ітерації генерується випадкова кількість @RandomQuantity і вставляється новий запис в таблицю OrdersDetails з випадковим продуктом, випадковою кількістю та відповідною ціною. Після цього запускається ще один цикл WHILE, який виконується 1500000 разів. Кожна ітерація цього циклу також вставляє новий запис в таблицю OrdersDetails з випадковим продуктом, випадковою кількістю та відповідною ціною. Після завершення всіх ітерацій обох циклів, в таблиці Orders оновлюється поле Order_Total_Price для кожного замовлення, встановлюючи його значення рівним сумі цін всіх записів в таблиці OrdersDetails для цього замовлення. На завершення, тимчасові таблиці #TempProducts та #TempOrders видаляються, а транзакція фіксується.

Оптимізація для швидкості реалізована у такий спосіб:

– тимчасові таблиці – зменшують кількість сканувань великих таблиць Products та Orders;

– випадковий вибір із тимчасових таблиць – забезпечує рівномірний вибір продуктів та замовлень без необхідності повного сканування великих таблиць;

– обчислення ціни та загальної ціни в запитах – зменшує складність запитів та потенційно покращує продуктивність;

– розділені цикли для нових та наявних замовлень – урізноманітнює дані та покращує продуктивність для обох сценаріїв.

Таким чином, було розроблено комплекс SQL-скриптів для генерації тестових даних по ключових сутностях спроєктованої бази даних торгівельного підприємства. Скрипти реалізують підходи випадкової вибірки, об'єднання таблиць та циклічного формування даних з використанням функцій Transact-SQL для оптимізації швидкодії.

На наступному етапі доцільно виконати аналіз та візуалізацію згенерованих даних для перевірки їх реалістичності та відповідності поставленим вимогам щодо практичного застосування.

2.3.2 Аналіз та візуалізація згенерованих даних

Перед тим, як перейти до аналізу та візуалізації згенерованих даних, важливо визначити ключові вимоги до процесу генерації даних. Ці вимоги допоможуть нам забезпечити, що згенеровані дані відповідають нашим потребам та цілям.

1. Реалістичність: дані, які генеруються, повинні відображати реальні сценарії їх використання. Це означає, що вони повинні відповідати очікуваному розподілу та мають відповідати реальним бізнес-правилам та обмеженням.

2. Спрощення для ефективності: у деяких випадках, для спрощення процесу генерації даних, можна використовувати спрощені або узагальнені дані. Наприклад, назви можуть мати простий формат «Назва {номер}», де {номер} – це порядковий номер. Це допомагає забезпечити простішу генерацію даних, не втрачаючи при цьому важливі аспекти реалістичності.

3. Масштабованість: треба генерувати великі обсяги даних, щоб протестувати та оцінити продуктивність запитів та системи в цілому.

4. Узгодженість: дані між різними сутностями повинні бути узгоджені. Це означає, що вони повинні відповідати відповідним зв’язкам та обмеженням цілісності.

5. Відтворюваність: процес генерації даних повинен бути відтворюваний. Це означає, що ми повинні мати можливість генерувати однаковий набір даних з однаковими вхідними параметрами.

З урахуванням цих вимог можна перейти до аналізу та візуалізації згенерованих даних.

Після успішної генерації тестових даних за допомогою розроблених SQL-скриптів наступним кроком є аналіз та візуалізація отриманих даних.

Метою даного аналізу є:

– перевірка відповідності згенерованих даних;
– контроль реалістичності та узгодженості даних між сутностями;
– візуальне подання прикладів даних по ключових сутностях;
– підтвердження успішності генерації необхідних обсягів даних.

Для аналізу буде використовувати засоби візуалізації та звітності СУБД SQL Server у поєднанні з вибірковими SQL-запитами до згенерованих даних.

Розглянемо результати аналізу даних по ключових сутностях послідовно.

Сутність «Brands» – відповідає за збереження інформації про бренди товарів у базі даних. За допомогою розробленого скрипту було згенеровано 200 000 унікальних брендів, що відповідає визначеним вимогам.

Результати генерації «Brands» у скороченому вигляді представлено в таблиці 2.3.

Таблиця 2.3 – Фрагмент результатів генерації для таблиці «Brands»

Brand_ID	Brand_Name
A45354E4-5DE6-4096-9E14-00000B58BD01	Бренд 1
D8A56A81-003E-4C37-B301-000067816945	Бренд 2
3D680B29-BEB0-4E16-92BE-00014C293227	Бренд 3
…	…
52F59D7B-2C25-4A84-9945-C9BBEB290929	Бренд 200 000

Як видно, назви брендів мають простий формат «Бренд {номер}», де {номер} – це порядковий номер бренду. Такий підхід був обраний для спрощення процесу генерації, оскільки реальні назви брендів потребували б створення окремого довідника на десятки чи сотні тисяч значень. Враховуючи, що бренд не є первинною сутністю та використовується виключно для зв'язку з іншими даними, прості унікальні назви цілком придатні для цілей тестування.

Отже, дані по сутності «Brands» успішно згенеровані та відповідають вимогам для подальшого аналізу та тестування запитів.

Сутність «Categories» – відповідає за класифікацію товарів по категоріях. Згідно з вимогами до прототипу замість генерації була зроблена вставка 36 категорій для покриття типового асортименту товарів в роздрібних магазинах.

Результат вставки категорій «Categories» наведено у таблиці 2.4.

Таблиця 2.4 – Фрагмент результатів вставки для таблиці «Categories»

Category_ID	Category_Name
7343F6EF-A0CB-425A-AFDC-00C10C1674E4	Лаунчпади
430841D0-A204-4D45-85D2-054ADE7391B2	Книги
80E6466F-32CD-4573-83EE-07160BB95E6D	Електронні книги
…	…
675459A5-ED85-401A-8212-EC380F69F662	Взуття та сумки
B1CB2084-E9FF-4E71-9F6E-FA127DFB72D2	CD та вініл
3ACF9EAF-DE64-4544-91BC-FCD3CB88D916	Ювелірні вироби

Перелік усіх категорій, що були вставлені: лаунчпади, книги, електронні книги, пк та відеоігри, пристрої та аксесуари, іграшки та ігри, prime відео, автомобільні товари, сад та природа, освітлення, паперова та офісна продукція, одяг, подарункові картки, dvd та blu-ray, сумки та валізи, дім та кухня, спорт та активний відпочинок, комп'ютери та аксесуари, ручна робота, електроніка та фото, програмне забезпечення, великі побутові прилади, інструменти для ремонту і будівництва, цифрова музика, аудіокниги та оригінали, годинники, додатки та ігри, краса, товари для домашніх тварин, здоров'я та особистий догляд, музичні інструменти та

dj, бакалія, бізнес, промисловість та наука, взуття та сумки, cd та вініл, ювелірні вироби.

Отже, дані сутності «Categories» успішно вставлені та готові для їх подальшого використання.

Ключова сутність «Products» – зберігає дані про товари. Відповідно до вимог було згенеровано 2 млн товарів, що є репрезентативним обсягом для проведення тестування.

Результати генерації «Products» представлені у таблиці 2.5.

Таблиця 2.5 – Фрагмент результатів генерації для таблиці «Products»

Product_ID	Product_Name	Product_Description	Product_Price	Category_ID	Brand_ID
D9455410-2098-46EF	Продукт 1	Опис 1	460,0571	93599BC2-8BB4-49D9	99A04767-2E3F-4289
5112BF17-5032-40AD	Продукт 2	Опис 2	263,6476	6520F664-AB49-4E41	8A570167-F92F-4FE9
CAB2DDD7-2EB5-480A	Продукт 3	Опис 3	308,7715	6667DF97-CCE6-4D26	17957718-0F3C-4BA7
…	…	…	…	…	…
7E0979EB-64FD-41AF	Продукт 2000000	Опис 2 000 000	205,3231	E00FBAA8-15BD-41EF	1B98E596-F572-477D

Отже, кожен товар має:

- унікальний ідентифікатор Product_ID;
- назву у форматі «Продукт {Номер}»;
- опис у форматі «Опис {Номер}»;
- випадкову ціну;
- посилання на випадковий бренд з таблиці «Brands»;
- посилання на випадкову категорію з таблиці «Categories».

При генеруванні даних для сутності «Products» було прийнято рішення використовувати шаблонну структуру записів, а саме:

- назва у форматі «Продукт {Номер}»;
- опис у форматі «Опис {Номер}».

Такий підхід дозволяє ефективно згенерувати потрібну кількість – 2 млн (2 000 000) реалістичних товарів без залучення зовнішніх довідників та словників.

При цьому, головною метою тестових даних є не стільки реалістичність назв та описів кожної конкретної одиниці товару (це потребувало б значних ресурсів), а реалістичність:

– структури та взаємозв'язків даних між сутностями;

– характеру розподілу деяких показників (цін, категорій, брендів тощо);

– можливості розроблення та всебічного тестування аналітичних запитів.

Саме це і забезпечує обраний підхід – шаблонні унікальні назви продуктів задовольняють усі вищенаведеним критеріям.

Тому обраний спосіб генерації оптимально задовольняє вимогам для подальшого тестування продуктивності та аналізу даних у сервісах хмарних середовищ.

Сутність «Customers» – зберігає дані про клієнтів. Відповідно до вимог було згенеровано 1 000 000 клієнтів для забезпечення реалістичної бази покупців.

Результати генерації «Customers» представлені в таблиці 2.6.

Таблиця 2.6 – Фрагмент результатів генерації для таблиці «Customers»

CustomerID	FirstName	LastName	Phone	Email	City
4256B8EE-2ED1	Леонід	Скляренко	+380880907930	лскляренко9@example.com	Місто 30
…	…	…	…	…	…
D24C30FF-265E	Ірина	Чередніченко	+380770946143	ічередніченко4@example.com	Місто 43

Для забезпечення реалістичності та конфіденційності при генерації даних клієнтів було використано такий підхід:

– імена та прізвища генеруються випадково зі списку поширених українських імен та прізвищ;

– email має шаблонний вигляд {ім'я}{порядковий номер}@example.com;

– телефонні номери генеруються реалістично за шаблоном +380XXXXXXXXX.

Такий підхід дозволяє отримати анонімізовані, але реалістичні дані про клієнтів, які можна використовувати для тестування без порушень конфіденційності персональних даних реальних осіб.

Отже, дані сутності «Customers» також успішно згенеровані та готові для подальшого аналізу та тестування SQL-запитів.

Сутність «Branches» – відповідає за збереження інформації про філіали.

Відповідно до вимог прототипу було згенеровано 100 філіалів. Результат генерації «Branches» представлений в таблиці 2.7.

Таблиця 2.7 – Фрагмент результатів генерації для таблиці «Branches»

BranchID	Name	City	Phone	Email	Adress
1793A94D-4AC4-43A5-A548	Філія 1	Місто 15	+380783373984	email15 @example.com	Адреса 15
4475AE6C-A967-4EDF-A4FF	Філія 2	Місто 98	+380485052388	email98 @example.com	Адреса 98
09CC09EC-E8B2-477D-A846	Філія 3	Місто 19	+380669836140	email19 @example.com	Адреса 19
…	…	…	…	…	...
A6999106-D6AB-4AE6-945C	Філія 100	Місто 92	+380800571550	email92 @example.com	Адреса 92

Для забезпечення реалістичного розподілу даних щодо філіалів по регіонах було використано шаблон для міст по типу «Місто {номер}». Також були згенеровані унікальні телефонні номери, email адреси та фактичні адреси розташування для кожного філіалу.

Email адреси мають шаблонний вигляд email{номер}@example.com, де {номер} відповідає порядковому номеру міста, що забезпечує унікальність.

Фактичні адреси також згенеровані за шаблоном "Адреса {номер}", де {номер} визначається відповідним номером міста.

Такий підхід дозволяє отримати реалістичні, але анонімізовані дані, для аналізу та тестування без використання персональної (приватної) інформації.

Отже, дані сутності «Branches» також успішно згенеровані та готові до подальшого використання в аналітичних запитах та тестах продуктивності.

Сутність «Employees» – містить дані про співробітників компанії.

Відповідно до встановлених вимог було згенеровано 250 000 співробітників, що працюють у різних філіалах на позиціях асистента з продаж або менеджера.

Приклад результатів генерації «Employees» представлений у таблиці 2.8.

Таблиця 2.8 – Фрагмент результатів генерації сутності «Employees»

EmployeeID	FirstName	LastName	JobTitle	BranchID
2E08782F-4401-442D-A583	Шолом	Березовський	Асістент по продажам	AA168529-9B8B-4C1B-AFDC
DB1AE83F-1F08-45DC-AB27	Олексій	Лисенко	Асістент по продажам	514B9242-AD29-4463-9C95
…	…	…	…	…
EF1D94BB-0B1C-4A0B-81B0	Валентин	Захаров	Менеджер	A0336CFC-4CAF-4EB3-8EFA

Імена співробітників генерувались випадковим чином із списку поширених українських імен та прізвищ. Позиції призначались як «Асистент з продаж» або «Менеджер» з ймовірністю 70% та 30% відповідно.

Зв'язок із філіалом (Branch_ID) встановлювався випадковим чином серед усіх згенерованих філіалів.

Такий підхід дозволяє отримати реалістичні анонімізовані дані для подальшого аналізу. Дані сутності «Employees» були успішно згенеровані.

Рухаючись далі, розглянемо сутність «Warehouses», яка відповідає за дані про склади та запаси товарів на них.

Згідно початкових вимог було згенеровано 600 складів. Проте, для забезпечення реалістичних даних про запаси, кожен склад був укомплектований різною кількістю продуктів, яка сумарно складає 1 млн записів по всім складам. Продукти обирались випадковим чином з усього каталогу згенерованих товарів.

Фрагмент результатів генерації «Employees» представлений в таблиці 2.9.

Таблиця 2.9 – Фрагмент результатів генерації сутності «Employees»

WarehouseID	ProductID	QuantityProduct	LastUpdate	ReorderLevel	BranchID
5C57A980-9C2D-43D4	C358DAED-7AF6-401E	2807	23.11.2023	18	A4B07467-8259-448B
5C57A980-9C2D-43D4	DB34B10D-A3FA	769	24.10.2023	615	A4B07467-8259-448B
2364AD51-877B-4E7D	FA090BAD-ECE0	3966	27.08.2023	410	90AE9D52-2A92-4593
…	…	…	…	…	…
4F8E1691-F796-4420	029E7C2C-CFF2-461A	3389	02.11.2023	67	C5A133AC-A9B2-434F
4F8E1691-F796-4420	88004E44-F160-473D	3284	25.10.2023	770	C5A133AC-A9B2-434F

Такий підхід дозволяє сформувати реалістичну картину запасів по складах та філіях для проведення подальшого аналізу.

Таким чином, хоча складів було згенеровано у кількості 600, загальна кількість записів в таблиці «Warehouses» склала понад 1 млн за рахунок деталізації запасів.

Сутність «Orders» – містить інформацію про замовлення в інтернет-магазині.

Відповідно до вимог було згенеровано 3 000 000 замовлень за період 3 років – від 2020 р. до 2023 р. Для кожного замовлення випадковим чином вибирався клієнт, співробітник та загальна сума замовлення.

Статус замовлення встановлювався як «Завершене», «Скасоване» або «В процесі». Дата виконання заповнювалась лише для завершених замовлень.

Фрагмент результатів генерації «Orders» представлений у таблиці 2.10.

Таблиця 2.10 – Фрагмент результатів генерації сутності «Orders»

Order_ID	TotalPrice	Date	Status	FilfilledDate	CustomerID	EmployeeID
B87F171E-0EC5-4924	81605,65	11.06.2022 18:30	Завершений	05.07.2022 17:21	E8CC63C2-BC74-4006	EACE9487-5122-4C43
2BF3AFA7-00AE-461B	7010,62	10.05.2022 5:46	Скасований	NULL	64EDC812-2986-4AE3	32890B6B-38AB-4759
AB37BFDD-0D39-488C	8977,54	19.02.2022 3:42	В процесі	NULL	98D2B8D2-6223-421D	384AB8C7-A62E-48EE
…	…	…	…	…	…	…
C4A5F365-8964-4B68	88940,74	02.01.2021 15:17	Завершений	20.01.2021 6:36	2D65AC1D-04C7-4CBB	6F0F2930-4C60-403F

Процес генерації даних був успішно виконаний. Всі сутності, включаючи «Customers», «Branches» та «Orders», були належним чином заповнені псевдо-реалістичними даними, що дозволить у подальшому проводити аналіз та тестування SQL-запитів.

Отже, перейдемо до аналізу генерації для наступної сутності «OrdersDetails», яка містить детальну інформацію про замовлені товари в кожному замовленні.

Фрагмент результатів генерації «OrdersDetails» представлений у таблиці 2.11.

Таблиця 2.11 – Фрагмент результатів генерації сутності «OrdersDetails»

Order_ID	Product_ID	Quantity	Price
B87F171E-0EC5-4924	C99558CE-466D-48E3	80	20070,528
B87F171E-0EC5-5921	55B6FE94-A7D8-4DCF	93	37798,8549
B87F171E-0EC5-4231	306D4311-D655-43DD	54	23736,2724
…	…	…	…
AB37BFDD-0D39-488C	8C01BF13-62D8-4DF5	29	8977,5445

Відповідно до згенерованих даних для кожного замовлення було згенеровано 4.5 мільйони (4 500 000) записів в «OrdersDetails» із зазначенням:

– Order_ID – посилання на відповідний запис в таблиці Orders;

– Product_ID – посилання на конкретний товар в таблиці Products;

– Quantity – кількість замовленого товару;

– Price – ціна одиниці товару на момент замовлення.

Таким чином формується детальна розбивка кожного замовлення на окремі товари та їх кількість.

Отже, процес генерації даних для всіх ключових сутностей бази даних було успішно завершено відповідно до визначених вимог.

Для наочності, у таблиці 2.12 наведено зведену інформацію щодо кількості записів, згенерованих для кожної сутності:

Таблиця 2.12 – Підсумки генерації даних

Назва таблиці	Час генерації, сек.	Кількість записів	Об'єм, МБ
Brands	6	200 000	15
Categories	0.001	36	0.136
Products	26	2 000 000	273
Customers	6	1 000 000	161
Branches	0.001	100	0.136
Warehouses	8	1 000 000	76
Employees	23	250 000	38
Orders	339	3 000 000	453
OrdersDetails	823	4 500 000	415
Усього	1233.002	11 950 136	1431.272

Загальна кількість згенерованих записів складає понад 11 мільйонів (11 000 000), що є цілком достатнім обсягом даних для подальшого всебічного тестування та аналізу реалізації SQL-запитів у сервісах хмарних середовищ.

Графічне зображення розподілу згенерованих даних за основними сутностями представлено на рисунку 2.5.

Рисунок 2.5 – Розподіл згенерованих даних по сутностях, %

На графіку на рисунку 2.5 відображено розподіл згенерованих даних по таблицям бази даних. Найбільший обсяг даних (37,66%) припадає на таблицю «OrdersDetails». Таблиця «Orders» має другий за величиною обсяг – 25,10%. Таблиця «Products» займає третє місце з 16,74% даних. Інші таблиці («Customers», «Warehouses», «Employees», «Branches», «Brands», та «Categories») мають значно менший обсяг даних – від 8,37% до найменшого значення – 0,0003%.

Цей графік допомагає візуалізувати розподіл даних за таблицями бази даних, що може бути корисним для оцінки продуктивності та оптимізації запитів. Зокрема, він може вказувати на таблиці, які можуть вимагати більше ресурсів для обробки через великий обсяг даних при використанні та налаштуванні сервісів хмарних платформ.

Розподіл обсягів даних за сутностями відповідає логіці побудови аналітичної бази даних для торгівельної компанії. Вони дозволяють моделювати реальні процеси та тестувати продуктивність запитів.

Графік залежність об’єму пам’яті від кількості записів у таблицях представлений на рисунку 2.6.

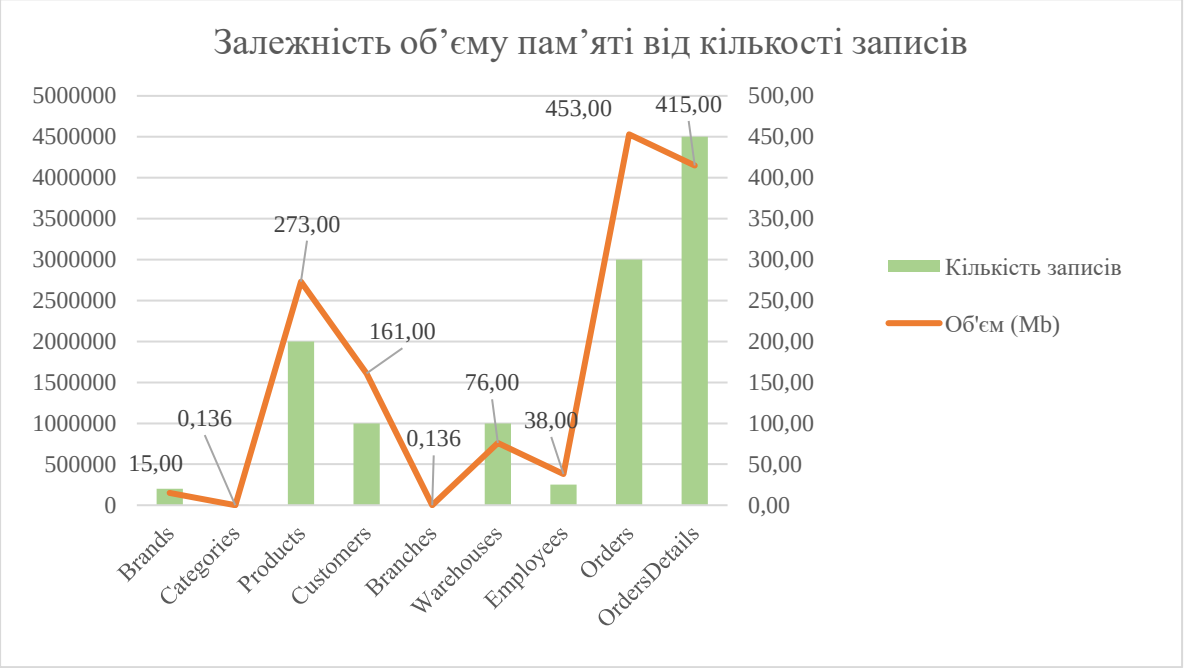

Рисунок 2.6 – Залежність об’єму пам’яті від кількості записів у таблицях

Найбільшу кількість записів містить таблиця «OrderDetails» – приблизно 4,5 млн (4 500 000), що є логічним, – адже вона зберігає деталізовану інформацію про кожне окреме замовлення. Попри таку велику кількість записів, ця таблиця займає 415 МБ пам'яті, що є меншим показником порівняно з таблицею «Orders».

Таблиця «Orders» містить на 1,5 млн (1 500 000) записів менше, проте займає більший об'єм пам'яті – 453 МБ. Це пов'язано з тим, що в таблиці «OrderDetails» зберігаються лише ключові атрибути деталей замовлення, в той час як в «Orders» – ширший список характеристик.

Важливо зазначити, що об’єм пам’яті, який займає таблиця в базі даних, залежить не лише від кількості записів. Інші фактори, які впливають на об’єм пам’яті, включають:

– типи даних: різні типи даних можуть займати різний об’єм пам’яті. Наприклад, числові типи даних, такі як INT або FLOAT, зазвичай займають менше місця, ніж текстові типи даних, такі як VARCHAR або TEXT;

– індекси: індекси, які створюються для покращення продуктивності запитів, також займають пам’ять. Кількість і розмір індексів можуть значно вплинути на об’єм пам’яті, який займає таблиця.

Отже, аналізуючи графік (рис. 2.6), можна зробити висновок, що розмір таблиці загалом зростає майже пропорційною кількістю записів в ній. Це дозволяє оцінити потреби у пам'яті при проєктуванні бази даних.

Після аналізу залежності об’єму пам’яті від кількості записів наступним кроком є дослідження залежності часу генерації від кількості записів у таблицях.

Графік «Залежність часу генерації від кількості записів» представлений на рисунку 2.7.

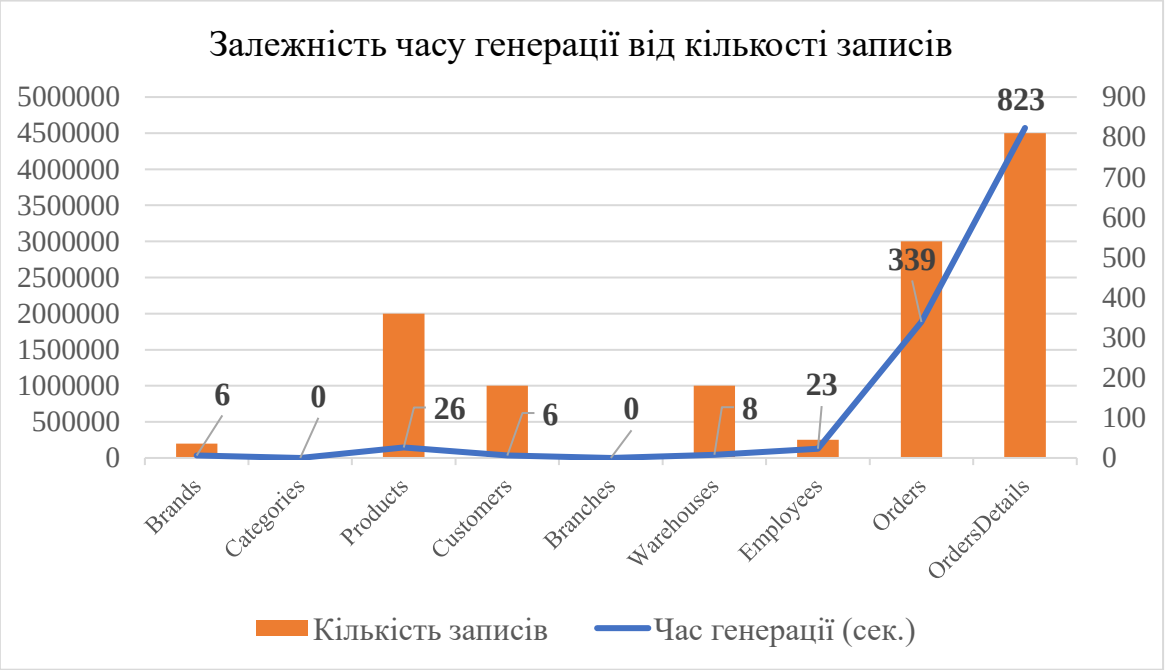

Рисунок 2.7 – Залежність часу генерації від кількості записів у таблицях

Аналіз цих даних свідчить про те, що час генерування записів варіюється в залежності від кількості записів та типу таблиці.

1. Brands: незважаючи на велику кількість записів (200 000), час генерування становить лише 6 сек. Це пов’язано з тим, що дані в цій таблиці мають просту структуру або використовують менше ресурсів.

2. Categories: ця таблиця має найменшу кількість записів (36), і, як очікувалося, має найкоротший час генерування (0.001 сек.).

3. Products: незважаючи на велику кількість записів (2 000 000), час генерування становить лише 26 сек. Це свідчити про ефективність структури даних або процесу генерування.

4. Customers і Warehouses: обидві таблиці мають приблизно однакову кількість записів (1 000 000), але час генерування відрізняється (6 сек. проти 8 сек. відповідно). Це пов’язано з різницею в структурі даних та складністю процесу генерування.

5. Branches: ця таблиця має невелику кількість записів (100), і час генерування становить 0.001 сек., що відповідає очікуванням.

6. Orders: таблиця має велику кількість записів (3 000 000), і час генерування відповідно великий (339 сек.). Це пов’язано зі складністю даних та процесом генерування.

7. Employees: ця таблиця має помірну кількість записів (250 000), і час генерування становить 23 сек.

8. OrderDetails: ця таблиця має найбільшу кількість записів (4 500 000), і час генерування відповідно найбільший (823 сек.). Це очікувано, оскільки ця таблиця містить деталізовану інформацію про кожне замовлення.

Велика кількість записів, а також процес генерування може значно збільшити час генерування, особливо для таблиць зі складною структурою даних.

Таким чином, процес генерації тестових даних для бази даних було успішно реалізовано з використанням SQL-скриптів. Згенеровані дані дозволяють проводити подальше тестування та аналіз.

Підсумовуючи етап генерації даних можна зробити такі висновки:

– структура бази даних відповідає логіці побудови аналітичної системи для торгівельної компанії. Вона включає всі ключові сутності, що дозволяє моделювати реальні бізнес-процеси;

– для генерування даних для ключових сутностей використано раціональний підхід: використання унікальних, але структурованих даних. Це дає можливість ефективно згенерувати великі обсяги за коротший час, зберігаючи при цьому логічну цілісність та узгодженість даних між сутностями;

– кількість згенерованих записів (понад 11 млн (майже 12 000 000)) та обсяг даних (понад 1 ГБ) є достатніми для аналізу продуктивності обробки запитів на всіх етапах дослідження;

– продемонстровано залежності часу генерації та обсягу даних від кількості записів за кожною сутністю. Це дає уявлення про ресурсоємність обробки запитів;

– виконано візуалізацію структури згенерованих даних за допомогою інфографіки. Це полегшує сприйняття та аналіз отриманих результатів генерації даних.

Таким чином, процес генерування тестових даних реалізовано згідно з усіма вимогами та з отриманням достатнього та репрезентативного набору даних для подальшого дослідження. Даний підхід може застосовуватися як шаблон при побудові аналогічних систем тестування продуктивності обробки даних у хмарних середовищах.

2.4 Розроблення та опис запитів до БД

Дані, що містяться в базі даних, можна використовувати для вирішення різних бізнес-задач, таких як аналіз продажів, прогнозування попиту, вивчення поведінки клієнтів тощо. Для цього необхідно розробити аналітичні SQL-запити, які дозволять отримати необхідну інформацію з бази даних.

Однією з ключових задач цього етапу є формулювання типових бізнес-запитів, які дозволять здійснювати комплексний аналіз даних у торгівельній сфері. Важливим аспектом є розробка запитів, які максимально відповідають вимогам та особливостям торгівельної компанії, забезпечуючи ефективність та швидкість обробки великих обсягів інформації.

Розглянемо послідовність розробки типового запиту.

Однією з важливих задач для торгівельної компанії є аналіз популярності різних категорій товарів серед клієнтів. Це дозволяє визначити найбільш затребувані групи товарів і скоригувати асортимент.

Бізнес-завдання: отримати інформацію про продукцію, її кількість на складах, середню ціну, максимальну та мінімальну ціну, а також кількість проданих одиниць для кожного бренду та категорії товарів.

Це бізнес-завдання:

– дає уявлення про популярність різних брендів і категорій товарів - які

продаються краще, а які гірше. Це допомагає вирішити, на які товари зробити ставку, а від яких, можливо, варто відмовитися;

– показує рівень товарних запасів по брендах і категоріях. Це дає можливість оптимізувати логістику і управління складами;

– дозволяє виявити найбільш прибуткові і збиткові товари. Можна скоригувати ціноутворення;

– порівняння обсягів продажів і залишків на складах – сигнал про необхідність стимулювання попиту на деякі товари.

Кроки реалізації запиту:

1. Потрібно об’єднати таблиці Brands, Products, Categories, OrdersDetails та Orders за допомогою операцій LEFT JOIN. Запит повинен включати всі записи з таблиці Brands та відповідні записи з інших таблиць.

2. Слід використати агрегуючі функції для підрахунку необхідних статистичних показників у розрізі кожного поєднання бренд-категорія.

3. Результати групуються за назвою бренду та категорії для формування підсумкових даних.

4. Вихідний результат сортується за спаданням загальної кількості проданих одиниць.

Для вирішення цієї задачі сформовано SQL-запит, представлений у Додатку Б.

Розглянуте бізнес-завдання та відповідний SQL-запит дозволяють комплексно проаналізувати ситуацію з товарами на складах і продажами в розрізі брендів та категорій продукції.

Іншою важливою задачею для торгівельної компанії є визначення найбільш популярних серед клієнтів окремих товарів. Це дозволяє зрозуміти попит, скорегувати асортимент, запланувати закупівлі та вчасно поповнити складські запаси затребуваних позицій.

Бізнес-завдання: отримати інформацію про топ-10 продуктів за обсягом продажів та вартістю, включаючи назву бренду, категорії товару, назву продукту, кількість проданих одиниць, ціну продукту та загальну вартість продажів.

Це бізнес-завдання дозволяє:

- виявити найпопулярніші продукти, на які варто зробити ставку при формуванні асортименту і проведенні маркетингових акцій;
- виявити, продукти яких брендів і категорій користуються найбільшим попитом у покупців. Це дозволить оптимізувати асортимент;
- оцінити затребуваність окремих товарів і спланувати їх закупівлі та поставки;
- проаналізувати цінову політику і прибутковість топ-продуктів.

Кроки реалізації запиту:

1. Потрібно об’єднати таблиці Brands, Products, Categories, Warehouses та OrdersDetails за допомогою операцій JOIN. Запит повинен включати всі таблиці, які мають відповідні записи.
2. Потрібно використати агрегуючі функції для обчислення статистичних даних для кожної комбінації назви бренду, назви категорії, назви продукту та ціни продукту.
3. Результати потрібно згрупувати за назвою бренду, назвою категорії, назвою продукту та ціною продукту.
4. Вихідний результат сортується за загальною вартістю замовлень у порядку спадання.
5. Запит повинен повертати тільки перші 10 записів з відсортованого списку.

Сформовано наступний SQL-запит, представлений у Додатку Б.

Це дає можливість приймати обґрунтовані управлінські рішення стосовно асортименту, ціноутворення та закупівель.

Попередній SQL-запит дозволяє визначити топ-10 продуктів за популярністю серед клієнтів компанії. Проте для ефективного управління асортиментом та планування закупівель не менш важливо розуміти ситуацію з наявними товарними запасами на складах.

Бізнес-завдання: отримати інформацію про товари, кількість яких на складі менше за рівень перезамовлення.

Це бізнес-завдання:

– показує товари, які необхідно терміново до закупити, щоб уникнути їх дефіциту та втрати продажів;

– дозволяє оптимізувати процес закупівель і логістику – замовити необхідну кількість товарів для конкретних складів;

– слугує індикатором попиту на конкретні товари.

Кроки реалізації запиту:

1. Потрібно об’єднати таблиці Products, Brands, Categories, Warehouses та Branches за допомогою операцій JOIN. Це означає, що потрібно включити всі записи, які мають відповідні записи в усіх цих таблицях.

2. Потім потрібно використати умову WHERE для вибору записів, де кількість продуктів у складі (Warehouses.WareHouse_Quantity_Product) менше за рівень повторного замовлення (Warehouses.ReorderLevel). Це допоможе ідентифікувати продукти, які можуть потребувати повторного замовлення.

3. Далі потрібно вибрати для виводу декілька полів з кожної таблиці, включаючи назву філії, ID складу, назву бренду, назву категорії, назву продукту, кількість продуктів у складі, рівень повторного замовлення та дату останнього оновлення складу.

4. Потрібно відсортувати результати за кількістю продуктів на складі за порядком їх зростання.

Запит представлений у Додатку Б.

Такий підхід дозволяє своєчасно ідентифікувати товари, запаси яких потрібно поповнити, а також врахувати специфіку окремих філій компанії при плануванні логістики.

Попередні SQL-запити фокусувалися на аналізі товарів та управлінні асортиментом. Проте не менш важливим завданням для торгівельної компанії є розуміння поведінки клієнтів**.**

Бізнес-завдання: отримати інформацію про клієнтів разом зі статистикою їх замовлень.

Це бізнес-завдання дозволяє:

– визначити найцінніших клієнтів і розробити індивідуальні пропозиції для них;

– проаналізувати попит у різних регіонах.

Кроки реалізації запиту:

1. Потрібно об’єднати таблиці Customers, Orders та OrdersDetails за допомогою операцій JOIN. Це означає, що потрібно включити всі записи, які мають відповідні записи в усіх цих таблицях.

2. Потім потрібно вибрати декілька полів з кожної таблиці, включаючи ID клієнта, ім’я, прізвище, електронну пошту, телефон та місто.

3. Потрібно використати агрегуючі функції для обчислення статистичних даних для кожного клієнта.

4. Результати групуються за ID клієнта, ім’ям, прізвищем, електронною поштою, телефоном та містом.

5. Потрібно відсортувати результати за загальною вартістю замовлень у порядку спадання.

Цей SQL-запит представлений у Додатку Б.

Отриманий результат дозволяє виявити топ-клієнтів компанії та приймати обґрунтовані маркетингові рішення щодо утримання та стимулювання лояльності найприбутковіших покупців.

Попередній SQL-запит дозволяє проаналізувати клієнтську базу і виділити найбільш цінних клієнтів компанії. Не менш важливо зрозуміти ефективність роботи власних співробітників, які безпосередньо взаємодіють з клієнтами.

Бізнес-завдання: отримати інформацію про співробітників разом із статистикою їх завершених замовлень.

Це бізнес-завдання:

- показує загальну кількість замовлень та продажів кожного працівника – дозволяє оцінити ефективність роботи персоналу;
- дає інформацію про досвід роботи працівників у компанії в місяцях - корисно для HR;
- визначає топ-продавців компанії за обсягами продажів – можна заохочувати і мотивувати їх;
- показує продуктивність працівників у різних відділеннях - дозволяє порівняти ефективність;
- дає інформацію для аналізу плинності кадрів;
- допомагає оптимізувати розстановку персоналу по відділеннях.

Кроки реалізації запиту:

1. Потрібно об’єднати таблиці Employees, Branches, Orders та OrdersDetails за допомогою операцій JOIN. Це означає, що потрібно включити всі записи, які мають відповідні записи в усіх цих таблицях.

2. Потім потрібно використати умову LEFT JOIN для з’єднання таблиць Employees та Orders, де Employees.EmployeeID = Orders.EmployeeID і Orders.Order_Status = 'Завершений'. Це допоможе включити всі записи з таблиці Employees та відповідні записи з таблиці Orders, де статус замовлення є ‘Завершений’. Якщо відповідних записів немає, він повертає NULL.

3. Потрібно вибрати декілька полів з кожної таблиці, включаючи ім’я, прізвище, посаду співробітника, назву філії.

4. Потрібно використати агрегуючі функції для обчислення статистичних даних для кожного співробітника.

5. Результати групуються за ID співробітника, ім’ям, прізвищем, посадою та назвою філії.

6. Потрібно відсортувати результати за загальною вартістю замовлень у порядку спадання.

Для вирішення цієї задачі розроблено SQL-запит, представлений у Додатку Б.

Цей запит дає чітке уявлення про вклад кожного працівника в загальний результат компанії. Це може бути використано для мотивації та коригування ключового показника ефективності (KPI) персоналу.

Попередній SQL-запит фокусувався на аналізі ефективності роботи окремих співробітників компанії. Проте також важливо проаналізувати топ категорій товарів які продаються, це допоможе зрозуміти попит споживачів і визначити, які категорії є найбільш популярними.

Бізнес-завдання: отримати інформацію про топ-5 категорій товарів разом із статистикою їх продажів.

Це бізнес-завдання дозволяє:

- визначити найбільш популярні і прибуткові категорії товарів;
- проаналізувати попит та скоригувати асортимент;

- спланувати маркетингові заходи та акції для просування топ-категорій;
- прийняти обґрунтовані управлінські рішення щодо асортиментної політики.

Кроки реалізації запиту:

1. Потрібно об’єднати таблиці Categories, Products, OrdersDetails та Orders за допомогою операцій JOIN. Це означає, що потрібно включити всі записи, які мають відповідні записи в усіх цих таблицях.
2. Потім потрібно використати умову WHERE для вибору записів, де статус замовлення (o.Order_Status) є ‘Завершений’. Це допоможе включити лише ті замовлення, які були завершені.
3. Потрібно вибрати декілька полів з кожної таблиці, включаючи ID категорії, назву категорії.
4. Потрібно використати агрегуючі функції для обчислення статистичних даних для кожної категорії.
5. Результати групуються за ID категорії та назвою категорії.
6. Потрібно відсортувати результати за загальною кількістю проданих продуктів у порядку спадання.
7. Запит повинен повертати тільки перші 5 записів з відсортованого списку.

Для вирішення цієї задачі сформовано наступний SQL-запит, представлений у Додатку Б. Таким чином, запит надає топ-5 категорій продукції, які користуються найбільшим попитом у покупців. Це дозволяє аналізувати та коригувати асортиментну політику компанії.

Не менш важливо проаналізувати кількість продажів за категоріями товарів по різних містах та філіях. Це допоможе ідентифікувати, який є попит по категоріям продуктів у різних містах, та проаналізувати продажи по філіям за категоріями товарів.

Бізнес-завдання: проаналізувати, які категорії товарів користуються попитом у клієнтів з різних міст та за філіями.

Це бізнес-завдання дозволяє:

- проаналізувати попит і смаки клієнтів по регіонах;
- оптимізувати асортимент у відділеннях під конкретні ринки;

- спланувати цільові маркетингові заходи;
- порівняти ефективність роботи філій;
- прийняти обґрунтовані управлінські рішення щодо асортименту і ціноутворення.

Кроки реалізації запиту:

1. З'єднати таблиці Orders, Customers, OrdersDetails, Products, Categories, Employees та Branches за відповідними ключами.
2. Фільтрувати результати за завершеними замовленнями.
3. Групувати результати за назвою категорії, містом та філією.
4. Обчислити загальну кількість замовлених одиниць та загальну вартість замовлень для кожної комбінації категорії, міста та філії.
5. Сортувати результати за загальною вартістю замовлень у зворотньому порядку.

Вміст запиту представлений у Додатку Б.

Для отримання повної картини важливо також проаналізувати загальну результативність роботи філій.

Бізнес-завдання: отримати інформацію про кількість та вартість завершених замовлень для кожної філії.

Це бізнес-завдання надає:

- відображає кількість замовлень і загальну суму продажів по кожному відділенню компанії;
- позраховує частку кожного відділення в загальному обсязі продажів компанії у відсотках;
- дозволяє порівняти ефективність роботи різних відділень;
- виявляє найуспішніші відділення з найбільшою кількістю замовлень і оборотом;
- дає можливість проаналізувати причини відставання деяких відділень;
- надає можливість перерозподілити ресурси на користь ефективних відділень.

Кроки реалізації запиту:

1. З'єднати таблиці Branches, Employees, Orders та OrdersDetails за відповідними ключами.

2. Фільтрувати результати за завершеними замовленнями.

3. Групувати результати за назвою філії.

4. Обчислити загальну кількість завершених замовлень, відсоток від загальної кількості завершених замовлень та загальну вартість цих замовлень для кожної філії.

5. Сортувати результати за кількістю завершених замовлень у зворотньому порядку.

Цей запит представлений у Додатку Б.

Останній запит повинен охопити узагальнену аналітичну інформацію про продажі товарів по брендах та категоріях.

Бізнес-завдання: отримати узагальнену аналітичну інформацію про продаж окремих товарів з розбивкою по брендах і категоріях.

Це надає для торгівельної компанії такі можливості:

– оцінити ефективності продажів окремих товарів, брендів і категорій. За допомогою цього можна порівняти показники продажів різних товарів, брендів і категорій, щоб визначити найбільш успішні і менш успішні з них. Це допоможе компанії визначити, на яких товарах варто зосередити увагу, щоб підвищити їх продажі;

– проаналізувати популярність окремих товарів серед клієнтів з розбивкою по брендах і категоріях. Це дає розуміння, які саме продукти користуються попитом, а які – ні;

– оцінити загальні обсяги продажів в натуральних одиницях і грошовому виразі для кожної позиції товару. Це важливо для планування асортименту і закупівель;

– проаналізувати динаміку попиту – дату останнього замовлення кожного продукту. Це дозволяє ідентифікувати товари, на які попит зменшується;

Кроки реалізації запиту:

1. Об’єднати таблиці Products, Brands, Categories, OrdersDetails та Orders за допомогою операцій JOIN. Це означає, що потрібно включити всі записи, які мають відповідні записи в усіх цих таблицях.

2. Вибрати декілька полів з кожної таблиці, включаючи ID продукту, назву продукту, назву бренду, назву категорії.

3. Використати підзапити для обчислення статистичних даних для кожного продукту.

4. Результати групувати за ID продукту, назвою продукту, назвою бренду, назвою категорії.

5. Відсортувати результати за загальною вартістю замовлень у порядку спадання.

Запит представлений у Додатку Б.

Таким чином, цей запит надає комплексне уявлення про результати продажів на рівні окремих товарних позицій, що є критично важливим для ефективного управління асортиментом.

Отже, було розроблено низку аналітичних SQL-запитів для вирішення ключових бізнес-завдань у сфері торгівлі. Запити охоплюють аналіз товарів, клієнтів, персоналу та фінансових результатів. Розроблені запити дозволяють комплексно проаналізувати ситуацію і прийняти обґрунтовані управлінські рішення в ключових напрямках діяльності торгівельного підприємства.

Далі планується змоделювати процес виконання створених SQL-запитів на локальному сервері бази даних. Це дасть можливість оцінити ефективність розроблених запитів, проаналізувати обсяги та зміст даних, які вони генерують, а також час їх виконання.

Отримані результати стануть основою для подальшої оптимізації запитів і порівняння продуктивності з хмарними рішеннями у сфері аналізу великих даних.

2.5. Моделювання виконання запитів на локальному сервері бази даних

Для аналізу ефективності розроблених аналітичних SQL-запитів було проведено моделювання їх виконання в середовищі локального сервера бази даних.

Актуальність цього завдання зумовлена тим, що на етапі проєктування запитів важко в повній мірі оцінити їх ефективність та адекватність для вирішення конкретних бізнес-задач.

Тому метою даного етапу дослідження є:

1. Перевірка працездатності розроблених SQL-запитів в умовах реального середовища бази даних та обсягів інформації.
2. Оцінка швидкодії запитів та часу отримання кінцевих звітних даних.
3. Аналіз отриманих результатів на предмет повноти, корисності та адекватності для вирішення цільових бізнес-завдань.

Тобто дана фаза передбачає комплексне тестування та валідацію розроблених SQL-запитів відносно коректності логіки, швидкодії та бізнес-цінності отриманих даних.

Результати цих тестів є критично важливими для подальшого порівняння з можливостями хмарних платформ.

Для тестування було використано локальний сервер Microsoft SQL Server 2019, на локальному ресурсі з такими характеристиками:

– процесор: 11th Gen Intel(R) Core(TM) i5-11400H, фізичних 6 ядр та 12 логічних ядр; оперативна пам'ять: 16 ГБ; жорсткий диск SSD: 475 ГБ.

На цьому ресурсі було розгорнуто базу даних прототипу торгівельної компанії відповідно до розробленої моделі даних зі згенерованим тестовим набором даних обсягом приблизно 12 мільйонів (12 000 000) записів та 1.4 ГБ пам’яті.

Ефективна робота аналітичних запитів є запорукою швидкого отримання необхідної бізнес-інформації з корпоративних сховищ даних. Саме тому перевірка продуктивності розроблених SQL-запитів в умовах, максимально наближених до реальних, є вкрай важливою.

З огляду на це, треба провести тестування аналітичних запитів на локальному сервері з базою даних, що містить 12 млн (12 000 000) записів псевдореальних даних.

Головною метою тестів є валідація розроблених SQL-запити щодо:

– коректності логіки та повноти даних у результатах;

– часу виконання запитів та їх відповідності вимогам оперативної аналітики.

Результати тестування мають стати об'єктивною основою для порівняння можливостей локальної СУБД та хмарних аналітичних сервісів у наступних розділах дослідження.

Тестування проводилось за таким сценарієм: кожен запит виконувався послідовно в середовищі локальної СУБД, вимірювався час обробки та аналізувались отримані результати.

Перший запит (Q1) для аналізу товарів у розрізі брендів і категорій має на меті отримати аналітичну інформацію про популярність окремих брендів і категорій товарів компанії. Аналітичне завдання:

– проаналізувати популярність окремих брендів і категорій товарів серед клієнтів компанії. Це завдання дозволяє визначити, які бренди і категорії є найбільш затребуваними у клієнтів;

– виявити топ бренди і категорії за кількістю реалізованих одиниць продукції. Це завдання дозволяє визначити, які бренди і категорії є найбільш успішними з точки зору продажів;

– оцінити обсяги продажів в натуральних одиницях по кожному бренду в розрізі категорій. Це завдання дозволяє оцінити масштаб попиту на товари різних брендів і категорій;

– проаналізувати цінову політику і прибутковість різних брендів і категорій на основі даних про середню, мінімальну та максимальну ціну реалізованих продуктів. Це завдання дозволяє оцінити рентабельність продажів різних брендів і категорій.

Запит виконався за 47.18 сек. і повернув 1 745 639 рядків даних. Інформація містила дані про кількість товарів бренду, кількість і вартість реалізованих одиниць для кожного поєднання бренд-категорія, середню, максимальну та мінімальну ціну.

Частина результатів виконання запиту представлена на рисунку 2.8.

	Brand_Name	Category_Name	Product_Count	Average_Price	Max_Price	Min_Price	Sold_Quantity
1	Бренд 33671	Інструменти для ремонту і будівництва	15	201,1904	409,0695	43,6789	934
2	Бренд 93017	Паперова та офісна продукція	15	291,3036	469,9237	163,6452	770
3	Бренд 70107	Музичні інструменти та DJ	12	247,0247	415,3985	54,6061	763
4	Бренд 120352	Дім та кухня	12	261,6751	412,3674	154,0379	759
5	Бренд 150472	Годинники	14	295,9148	403,1311	143,0546	751
6	Бренд 164537	Prime Відео	14	330,9443	496,6929	283,1699	731
7	Бренд 90919	Спорт та активний відпочинок	14	209,067	349,59	21,703	729
8	Бренд 120269	Взуття та сумки	14	339,3353	431,9407	107,8219	719
9	Бренд 66554	Іграшки та ігри	13	221,2188	224,3604	217,5537	719
10	Бренд 26624	Взуття та сумки	11	312,955	409,7398	251,1689	715
11	Бренд 58565	Комп"ютери та аксесуари	15	208,5513	384,1767	43,0052	711
12	Бренд 89913	Лаунчпади	13	438,8346	484,8609	404,6804	706
13	Бренд 13059	Інструменти для ремонту і будівництва	11	203,7701	233,7587	123,8005	697
14	Бренд 195458	ПК та відеоігри	15	169,8414	242,6474	120,2128	687
15	Бренд 41137	DVD та Blu-ray	14	268,0959	473,1556	168,8777	685
16	Бренд 111129	Освітлення	16	307,4113	352,6346	249,2671	676
17	Бренд 128304	Комп"ютери та аксесуари	14	299,5607	463,8585	137,4208	667
18	Бренд 51720	Додатки та ігри	10	332,334	483,3503	46,8966	667
19	Бренд 30077	Ручна робота	14	192,0158	207,4307	166,999	667

Рисунок 2.8 – Результати виконання запиту аналізу товарів у розрізі брендів і категорій

Даний SQL-запит комплексно вирішує завдання аналізу популярності і прибутковості брендів та категорій товарів на основі статистики по фактичним продажам.

Другий SQL-запит (Q2) для виявлення топ-10 найбільш продаваних товарів має на меті отримати аналітичну інформацію про найбільш популярні продукти компанії. Аналітичне завдання:

– виявити топ-10 найбільш продаваних та прибуткових продуктів компанії. Це завдання дозволяє визначити, які продукти є найбільш затребуваними у клієнтів;

– проаналізувати обсяги реалізації цих продуктів як в натуральних одиницях, так і в грошовому виразі. Це завдання дозволяє оцінити масштаб попиту на ці продукти;

– зрозуміти, продукти яких саме брендів і категорій користуються найбільшим попитом у клієнтів. Це завдання дозволяє визначити, в яких сегментах ринку компанія має найбільші конкурентні переваги;

– оцінити середню ціну реалізації кожного з топ-продуктів. Це завдання дозволяє оцінити рентабельність продажів цих продуктів;

– визначити загальну суму виручки від продажів цих 10 найпопулярніших продуктів. Це завдання дозволяє оцінити внесок цих продуктів у загальний фінансовий результат компанії.

Запит виконався за 4.41 сек. і видав 10 рядків даних з інформацією про найпопулярніші серед покупців продукти.

Результати виконання запиту представлені на рисунку 2.9.

	Brand_Name	Category_Name	Product_Name	Order_Count	Product_Price	Total_Price
1	Бренд 72390	Ювелірні вироби	Product 1297863	207860	325,1892	55132065,502
2	Бренд 62830	Комп"ютери та аксесуари	Product 801267	161210	419,6887	50771859,845
3	Бренд 120462	Бакалія	Product 352312	143200	10,1792	48817700,894
4	Бренд 39771	Бізнес, промисловість та наука	Product 1673769	141620	411,4065	48380996,6752
5	Бренд 91795	Автомобільні товари	Product 1826791	154440	455,3051	47202748,515
6	Бренд 105598	Бакалія	Product 530162	164640	172,071	46232928,8008
7	Бренд 177313	Іграшки та ігри	Product 440775	136320	147,4493	44331918,833
8	Бренд 126781	Програмне забезпечення	Product 1568214	147420	483,1995	44270600,556
9	Бренд 44879	Книги	Product 1587940	134082	430,6078	43516468,2302
10	Бренд 11349	Електронні книги	Product 292803	120400	59,6395	41888915,4664

Рисунок 2.9 – Результати виконання запиту для виявлення топ-10 найбільш популярних товарів

Отже, даний запит вирішує завдання виявлення лідерів продажів серед всього асортименту товарів.

Запит на пошук товарів (Q3) з недостатньою кількістю на складах має на меті отримати аналітичну інформацію про запаси товарів на складах компанії. Аналітичне завдання:

– виявити товари на складах різних відділень компанії, запаси яких опустилися нижче встановленого рівня поповнення. Це завдання дозволяє визначити, які товари є найбільш дефіцитними;

– отримати детальну інформацію по кожній такій позиції: назва відділення, ID складу, бренд, категорія, назва товару, поточна кількість, рівень для поповнення. Ця інформація необхідна для того, щоб можна було провести більш детальний аналіз дефіциту товарів;

– відсортувати результати за зростанням фактичної кількості товарів на складі, щоб виявити найбільш критичні ситуації. Це дозволяє визначити, які товари мають найвищий ризик дефіциту;

– проаналізувати дату останнього поповнення складу, щоб зрозуміти наскільки гостро стоїть проблема дефіциту конкретних товарів. Це дозволяє оцінити, наскільки давно товари не поповнювалися на складі;

Запит повернув 106 123 записи за 1.66 сек. Інформація містить перелік продуктів, які потрібно терміново до закупити для конкретних складів компанії.

Частина результатів виконання запиту представлена на рисунку 2.28.

	BranchName	WareHouse_ID	Brand_Name	Category_Name	Product_Name	WareHouse_Quantity_Product	ReorderLevel	WareHouse_Last_Update
1	Branch 65	185C70DF-564D-4BCE-9549-BF157467FBCF	Бренд 110319	Товари для домашніх тварин	Product 1163084	0	670	2023-10-24
2	Branch 17	9F075D93-E144-4763-91CF-B43E464EA547	Бренд 178144	Лаунчпади	Product 1746148	0	560	2023-10-06
3	Branch 30	98A38EA0-6D09-4465-BC2D-BC7B8C2B0A55	Бренд 8252	Автомобільні товари	Product 1221459	0	674	2023-08-26
4	Branch 7	9C206E61-C292-4250-A979-32CA22949984	Бренд 58877	Електроніка та фото	Product 1948259	0	499	2023-07-27
5	Branch 39	15A478E1-F4D0-413C-A19F-5EAA03F8F135	Бренд 11025	Цифрова музика	Product 881410	0	57	2023-03-21
6	Branch 60	AC5CC8C8-E1F4-4FC0-8F17-C41D4F2DA5C5	Бренд 184452	CD та вініл	Product 860916	0	372	2023-03-08
7	Branch 15	EBC2060C-A02D-47DB-BDA9-F7E00EFB5C75	Бренд 118552	Сумки та валізи	Product 1234990	0	443	2023-04-10
8	Branch 45	BA46B0A2-F196-4395-8FF6-E7544595F820	Бренд 121419	Паперова та офісна продукція	Product 503848	0	107	2023-04-29
9	Branch 16	FBD179A0-260D-4B98-AF84-60E305BEAFC5	Бренд 121419	Паперова та офісна продукція	Product 503848	0	550	2023-07-03
10	Branch 9	786BD94F-5CAD-4011-83F9-F2B53F4BAA4E	Бренд 37785	Бакалія	Product 273534	0	503	2023-06-29
11	Branch 29	4F8E1691-F796-4420-9A8D-0AC0D353C637	Бренд 113340	Пристрої та аксесуари	Product 524088	0	498	2023-10-26
12	Branch 21	CB0792C1-F9A5-4EF6-95DD-63B62570B535	Бренд 149169	Ручна робота	Product 1978512	0	601	2023-10-03
13	Branch 30	0C8E81D5-C7CE-4933-A14B-D5D12B7DBF08	Бренд 116148	Іграшки та ігри	Product 69489	0	465	2023-06-04
14	Branch 26	B30ECF9A-54F2-44C5-BA31-BB9DD030ABDC	Бренд 108570	CD та вініл	Product 1733405	0	459	2023-06-04
15	Branch 95	32856F35-0199-4068-94D6-D2B7EA7ADE21	Бренд 86673	Сумки та валізи	Product 1493633	0	203	2023-07-29
16	Branch 59	79552199-E72D-465F-A2B1-10EFDE12362A	Бренд 19000	Здоров"я та особистий догляд	Product 240016	0	413	2023-09-01
17	Branch 7	B1AC0853-3644-4838-A435-BD593E389D3E	Бренд 17528	Одяг	Product 1332269	0	695	2023-05-24
18	Branch 22	E1B42966-A624-4E55-9EA3-FBD45A426B23	Бренд 138435	CD та вініл	Product 401975	0	264	2023-11-24
19	Branch 20	7CAEE0CE-7523-4569-ADBF-783A3093637C	Бренд 176857	Взуття та сумки	Product 1670338	0	595	2023-03-03

Рисунок 2.10 – Результат запиту на пошук товарів з недостатньою кількістю на складах

Таким чином, запит вирішує ключове аналітичне завдання моніторингу складських запасів і своєчасного виявлення товарів, які потрібно терміново докупити на різних філіалах.

Запит для аналізу клієнтської бази і виявлення найцінніших клієнтів (Q4) має на меті отримати аналітичну інформацію про клієнтську базу компанії. Аналітичне завдання:

– проаналізувати клієнтську базу і виявити найприбутковіших клієнтів компанії на основі обсягів та вартості замовлень. Це завдання дозволяє визначити, які клієнти є найбільш цінними для компанії;

– отримати деталізовані відомості про клієнтів: ім'я, контактні дані, місто. Ці дані необхідні для того, щоб можна було провести більш детальний аналіз клієнтської бази;

– розрахувати для кожного клієнта: кількість замовлень за період, загальну вартість цих замовлень, дату останнього замовлення. Ці дані дозволяють оцінити обсяг продажів кожного клієнта, його активність та потенціал;

– відсортувати клієнтів за спаданням загальної суми замовлень для виявлення топ-клієнтів. Це дозволяє виділити найцінніших клієнтів компанії.

Запит виконався за 11.62 сек. та повернув 950 073 рядків даних. Інформація містить дані про кількість, суму та дати замовлень кожного клієнта компанії.

Частина результатів виконання запиту представлена на рисунку 2.11.

	CustomerID	FirstName	LastName	Email	Phone	City	Order_Count	Total_Quantity_Products	Total_Price	Last_Order_Date
1	085553B4-F590-4CFD-9210-5FDBD8A7D950	Тарас	Лукашевич	тлукашевич163860@example.com	+380440163860	Город 60	28	1462	1279909,7788	2023-03-21 11:27:51.727
2	2ACF75F2-AF50-400C-8795-A0C2B4C64608	Ігор	Гаврилюк	ігаврилюк406550@example.com	+380660406550	Город 50	21	1159	1259929,2948	2023-06-14 15:25:17.320
3	2B475F2F-75E2-4313-9593-523FFEF8AF0A	Юлія	Коваленко	юковаленко740068@example.com	+380880740068	Город 68	16	1055	1224004,2524	2023-11-30 03:52:39.480
4	EDD945CE-2FC2-41AB-B78A-FFBDD69557A2	Аврелія	Мельничук	амельничук696164@example.com	+380660696164	Город 64	22	1150	1206513,3675	2023-09-09 07:38:42.273
5	70A92D66-C624-424F-BC33-3EFE92574F4A	Марія	Савченко	мсавченко991583@example.com	+380990991583	Город 83	11	692	1172688,0866	2023-07-18 10:17:32.733
6	FBFF6C5A-AEEC-406C-A6A5-5A03A07ED76D	Максим	Зайцева	мзайцева811618@example.com	+380880811618	Город 18	20	1066	1164260,9995	2023-11-02 08:45:26.877
7	EDD5756D-204E-4446-B81D-48245FC3A854	Автоном	Вербицька	авербицька62417@example.com	+380990062417	Город 17	10	623	1157973,3494	2022-06-16 21:47:10.520
8	EFE6D65D-4494-4A96-8C25-DFD27D757A8D	Орест	Коваленко	оковаленко196474@example.com	+380880196474	Город 74	10	648	1115687,6674	2023-08-28 16:38:01.230
9	ED64B205-4164-4C49-98C5-FDE7F9C1C6C4	Ярослав	Григорович	ягригорович563582@example.com	+380660563582	Город 82	24	1475	1112140,8043	2023-05-07 21:56:31.607
10	0E1F754D-D121-4FEC-ABD6-C6150B9DA45D	Богдан	Клименко	бклименко422292@example.com	+380440422292	Город 92	10	492	1105177,2687	2023-06-19 21:13:43.933
11	046BA478-05B5-4DB7-B4CA-B1CFFCB651EA	Наталка	Клименко	нклименко608931@example.com	+380770608931	Город 31	22	1087	1103440,3078	2023-07-08 21:45:57.720
12	D36BB613-7DF9-48CA-B8F6-742297ED1D60	Олександр	Семенюк	осеменюк546099@example.com	+380770546099	Город 99	22	1143	1099811,2493	2023-12-02 13:30:20.860
13	352E3EB4-6EA2-459F-8056-97138C4EA053	Шолом	Дорошенко	шдорошенко135368@example.com	+380660135368	Город 68	10	577	1095689,1005	2022-07-24 23:59:28.860
14	97ABF955-1719-47C7-8F41-1429B558CCFA	Яків	Андрєєва	яандрєєва21792@example.com	+380440021792	Город 92	14	930	1082586,4706	2023-04-16 01:19:02.107
15	C7B78456-BCFD-4843-9F86-B8F760509D3C	Максим	Кравченко	мкравченко911333@example.com	+380990911333	Город 33	19	906	1080229,6984	2023-11-26 08:38:52.920
16	9A88377D-16AB-4E5F-9FEA-D8756F81A959	Оксана	Ярошенко	оярошенко640997@example.com	+380990640997	Город 97	17	1055	1068235,3485	2023-06-25 08:41:04.430
17	CA07C084-B314-49B5-BAF2-BE02DC62F63F	Фадей	Савчук	фсавчук623091@example.com	+380770623091	Город 91	13	756	1066162,0824	2023-11-18 17:49:49.937
18	DF7BBC96-F32B-49CC-9395-C336E2FC255F	Герасим	Литвиненко	глитвиненко231125@example.com	+380990231125	Город 25	17	949	1063048,6131	2023-11-28 09:11:34.913
19	8A828113-28D5-4E8A-B473-34AE0D41E140	Марія	Савченко	мсавченко629583@example.com	+380770629583	Город 83	10	503	1061377,6237	2023-10-31 02:51:22.037

Рисунок 2.11 – Результат запиту для аналізу клієнтської бази і виявлення найцінніших клієнтів

Таким чином, запит виконує завдання сегментації клієнтів за вартістю їх покупок та виявлення VIP клієнтів компанії для подальших маркетингових дій.

SQL-запит для аналізу продажів співробітників (Q5) має на меті отримати аналітичну інформацію про ефективність роботи співробітників компанії. Аналітичне завдання:

– проаналізувати ефективність роботи співробітників компанії на основі статистики по замовленням, які вони особисто обробляли. Це завдання дозволяє оцінити внесок співробітників у загальний результат діяльності компанії;

– отримати дані про кожного співробітника: ім'я, посада, відділення. Ці дані необхідні для того, щоб можна було провести більш детальний аналіз ефективності роботи співробітників;

– розрахувати для кожного співробітника кількість оброблених замовлень та їх загальну вартість. Ці дані дозволяють оцінити обсяг продажів кожного співробітника;

– відсортувати список за загальним обсягом продажів співробітників у порядку спадання. Це дозволяє виявити найбільш ефективних працівників.

Запит повернув 249 623 рядків за 4.28 сек. У результатах міститься детальна статистика по кількості продажів оброблених працівниками компанії.

Частина результатів виконання запиту представлена на рисунку 2.12.

	FirstName	LastName	JobTitle	BranchName	Order_Count	Total_Sale	First_Order_Date	Last_Order_Date	Experience_in_Months
1	Сергій	Григоренко	Менеджер	Branch 93	26	1478389,5908	2021-01-28 03:05:38.100	2023-11-26 00:40:19.187	34
2	Марина	Полякова	Менеджер	Branch 66	29	1450988,0367	2021-01-12 00:14:56.687	2023-11-27 06:47:12.990	34
3	Аркадій	Савенко	Асістент по продажам	Branch 3	29	1439053,5341	2020-12-07 20:05:28.910	2023-11-04 23:07:09.580	35
4	Анатолій	Романов	Асістент по продажам	Branch 90	26	1430429,423	2021-09-17 10:38:17.783	2023-04-15 01:03:38.040	19
5	Вікторія	Сидоренко	Асістент по продажам	Branch 36	32	1349928,1818	2021-01-05 01:49:17.983	2023-07-08 14:07:51.790	30
6	Аверкій	Прокопенко	Асістент по продажам	Branch 75	25	1338257,687	2021-03-06 04:34:08.530	2023-10-29 04:51:16.960	31
7	Анжела	Ткачук	Менеджер	Branch 44	28	1325551,2836	2020-12-12 08:36:52.460	2023-10-01 11:29:34.987	34
8	Олег	Капуста	Менеджер	Branch 39	24	1321389,7684	2021-02-09 04:26:13.473	2023-05-07 13:02:56.213	27
9	Зеновій	Пащенко	Менеджер	Branch 52	25	1318128,787	2020-12-28 11:52:12.913	2023-09-17 01:14:29.740	33
10	Віктор	Щербак	Менеджер	Branch 58	27	1305136,5137	2021-02-20 04:16:44.063	2023-11-29 20:44:55.523	33
11	Віктор	Бойко	Асістент по продажам	Branch 4	26	1286679,2696	2020-12-10 09:11:12.687	2023-08-18 11:52:18.693	32
12	Владислава	Мужилко	Асістент по продажам	Branch 96	28	1271524,7086	2021-01-17 13:08:27.077	2023-06-23 12:04:19.453	29
13	Єлизавета	Савчук	Менеджер	Branch 53	24	1257642,5525	2021-08-18 17:18:59.070	2023-11-13 13:59:30.670	27
14	Орест	Жуковський	Менеджер	Branch 68	27	1254868,2218	2021-03-12 13:42:05.200	2023-02-24 10:19:50.243	23
15	Ярослав	Мазур	Асістент по продажам	Branch 95	17	1207622,6455	2020-12-25 07:53:09.060	2023-08-05 09:23:06.980	32
16	Євген	Мороз	Менеджер	Branch 65	25	1203725,1348	2021-03-22 06:26:18.110	2023-05-20 10:41:33.100	26
17	Броніслава	Яременко	Менеджер	Branch 24	32	1203110,3918	2021-01-08 17:59:21.047	2023-12-03 13:15:55.230	35
18	Валентин	Захаров	Менеджер	Branch 45	25	1201294,741	2021-01-30 00:18:12.657	2023-10-30 13:40:26.263	33
19	Віталій	Марченко	Асістент по продажам	Branch 88	25	1196806,0304	2020-12-07 23:28:00.627	2023-06-30 03:54:59.880	30

Рисунок 2.12 – Результати запиту для аналізу продажів співробітників

Отже, запит вирішує важливе завдання оцінки працездатності персоналу.

Запит топ-5 категорій (Q6) за обсягом продажів має на меті отримати аналітичну інформацію про структуру попиту на товари та послуги компанії. Аналітичне завдання:

– виявити топ-5 категорій товарів, які мають найбільші обсяги продажів в компанії. Це завдання дозволяє визначити, які товари та послуги є найбільш затребуваними у клієнтів;

– проаналізувати загальну кількість реалізованих одиниць продукції в цих 5 категоріях. Це завдання дозволяє оцінити масштаб попиту на ці товари та послуги;

– розрахувати загальну виручку від продажів товарів в кожній з топ-5 категорій. Це завдання дозволяє оцінити фінансовий потенціал цих товарів та послуг.

Запит виконався за 3.46 сек. та повернув 5 рядків. Дані дають уявлення про структуру попиту та допомагають формувати асортимент.

Результати виконання запиту представлена на рисунку 2.13.

	Category_ID	Category_Name	Product_Count	TotalQuantitySold	TotalSalesValue
1	6975D528-7499-4063-BF17-21E69DBE5C48	Prime Відео	68600	3470658	884558537,9625
2	1CB8776E-C82B-4D17-AC13-28315E6423C0	Освітлення	68543	3454590	878379241,3846
3	89A10A94-27AE-4D97-8CB9-A792129EB7FA	Краса	68465	3449803	873784678,7344
4	3ACF9EAF-DE64-4544-91BC-FCD3CB88D916	Ювелірні вироби	67882	3437950	876722932,3226
5	6520F664-AB49-4E41-B7A1-2E494B3B879E	Одяг	67936	3428614	873441689,1602

Рисунок 2.13 – Результати запиту топ-5 категорій за обсягом продажів

Отже, запит вирішує важливе завдання аналізу попиту в розрізі основних товарних категорій.

Запит аналізу замовлень за категоріями, містами та філіями (Q7) має на меті отримати аналітичну інформацію про регіональну специфіку попиту на товари та послуги компанії. Аналітичне завдання запиту:

– виявити, які категорії товарів мають найбільші обсяги продажів в кожному місті та філіалі. Це завдання дозволяє визначити, які товари та послуги є найбільш затребуваними в різних регіонах;

– проаналізувати, як співвідносяться обсяги продажів в різних категоріях в кожному місті та філіалі. Це завдання дозволяє оцінити, як регіональні фактори впливають на структуру попиту;

– визначити, в яких містах та філіалах найбільші обсяги продажів в найбільш затребуваних категоріях. Це завдання дозволяє визначити, в яких регіонах є найбільший потенціал для зростання продажів.

Запит повернув 359 516 рядків за 12.75 сек. Інформація містить дані про обсяги та вартість реалізованої продукції з розбивкою по категоріях, містах та філіях компанії.

Частина результатів виконання запиту представлена на рисунку 2.14.

	Category_Name	City	BranchName	TotalQuantityOrdered	TotalSalesValue
1	Паперова та офісна продукція	Город 10	Branch 100	1157	380497,8582
2	Prime Відео	Город 41	Branch 14	1045	372303,9432
3	Подарункові картки	Город 30	Branch 100	1105	362033,4676
4	Музичні інструменти та DJ	Город 16	Branch 17	1284	352829,2609
5	Prime Відео	Город 6	Branch 80	946	344389,2181
6	Пристрої та аксесуари	Город 86	Branch 66	1039	340264,9606
7	Лаунчпади	Город 66	Branch 83	1002	337933,3057
8	Музичні інструменти та DJ	Город 69	Branch 75	982	335119,0045
9	Пристрої та аксесуари	Город 96	Branch 93	1010	334882,8853
10	Дім та кухня	Город 53	Branch 63	1020	333019,0663
11	Інструменти для ремонту і будівництва	Город 71	Branch 80	1094	330485,0962
12	Електронні книги	Город 2	Branch 23	1016	329082,9309
13	Ювелірні вироби	Город 73	Branch 44	896	328143,8254
14	Музичні інструменти та DJ	Город 60	Branch 20	874	328040,6641
15	Prime Відео	Город 30	Branch 88	1146	327860,0269
16	Краса	Город 50	Branch 97	985	326702,909
17	Цифрова музика	Город 82	Branch 46	1114	326197,9918
18	Цифрова музика	Город 18	Branch 3	1022	323913,3233
19	ПК та відеоігри	Город 39	Branch 75	1200	323427,8783

Рисунок 2.14 – Результати запиту аналізу замовлень за категоріями, містами та філіями

Це дозволяє зрозуміти, як розподіляються продажі по території, і виявити найбільш перспективні з точки зору продажів міста та філіали.

На основі цих даних можна приймати рішення щодо розширення мережі, перерозподілу ресурсів, а також про проведення маркетингових заходів в конкретних містах та філіалах.

Запит підсумкової статистики виконаних замовлень по філіях (Q8) на меті отримати аналітичну інформацію про ефективність роботи філій компанії.

Аналітичне завдання запиту:

– виявити найбільш продуктивні відділення компанії. Це завдання дозволяє визначити, які відділення компанії працюють найефективніше та мають найбільший потенціал для зростання продажів;

– проаналізувати, як розподіляються продажі по відділеннях. Це завдання дозволяє оцінити, в яких регіонах компанії є найбільший попит на товари та послуги;

– визначити, в яких відділеннях найбільша виручка. Це завдання дозволяє визначити, в яких відділеннях компанії є найбільший прибуток.

Запит виконався за 5.16 секунд та повернув 100 рядків даних. Інформація містить загальну кількість та вартість завершених замовлень у розрізі окремих відділень компанії.

Частина результатів виконання запиту представлена на рисунку 2.15.

	BranchName	Order_Count	Order_Percentage	TotalSalesValue
1	Branch 4	16988	1.05	605384922,0208
2	Branch 35	16906	1.04	609825064,6801
3	Branch 12	16894	1.04	595142369,7185
4	Branch 3	16851	1.04	604302229,7484
5	Branch 48	16823	1.04	592438483,752
6	Branch 66	16814	1.04	598494924,0621
7	Branch 33	16799	1.04	589167084,2488
8	Branch 53	16691	1.03	586411435,3207
9	Branch 80	16650	1.03	583116617,6093
10	Branch 17	16619	1.03	588863220,8037
11	Branch 94	16607	1.02	585384240,5101
12	Branch 88	16605	1.02	571932151,8421
13	Branch 90	16604	1.02	584710860,298
14	Branch 97	16601	1.02	595471294,3696
15	Branch 98	16591	1.02	586535623,6112
16	Branch 62	16566	1.02	599215814,3267
17	Branch 71	16563	1.02	576229797,4365
18	Branch 52	16561	1.02	587930957,1778
19	Branch 45	16534	1.02	587113290,8763

Рисунок 2.15 – Результати запиту підсумкової статистики виконаних замовлень по філіях

На основі цих даних можна приймати рішення про перерозподіл ресурсів, а також про проведення маркетингових заходів в конкретних відділеннях.

Загальний запит продажів із деталізацією по брендах і категоріях (Q9).

Запит має на меті отримати аналітичну інформацію про продажі товарів у компанії. Аналітичне завдання запиту:

– виявити бренди, які мають найбільші обсяги продажів в кожній категорії товарів. Це завдання дозволяє визначити, які бренди є найбільш успішними в кожній категорії товарів;

– проаналізувати, як співвідносяться обсяги продажів різних брендів в кожній категорії товарів. Це завдання дозволяє оцінити конкурентоспроможність різних брендів у кожній категорії товарів;

– визначити, в яких категоріях товарів найбільші обсяги продажів у найбільш успішних брендів. Це завдання дозволяє визначити, в яких категоріях товарів найбільший потенціал для зростання продажів.

Запит обробився за 34.01 сек. та вивів 2 000 000 рядків аналітичної інформації.

Частина результатів виконання запиту представлена на рисунку 2.16.

	Product_ID	Product_Name	Brand_Name	Category_Name	Total_Orders	Total_Quantity	Total_Price	Last_Order_Date
1	90BA268C-34F2-4D48-A2DE-3F521601488A	Product 1975104	Бренд 94962	Дім та кухня	9	693	20048544,8409	2023-10-04 10:49:46.310
2	68733986-73A2-4120-8052-38A66553DB52	Product 1488414	Бренд 114615	Паперова та офісна продукція	10	622	19494126,8471	2023-07-14 04:42:16.430
3	43446DCD-DCC9-48C7-BF0E-E46620A702E2	Product 1067913	Бренд 143863	Аудіокниги та оригінали	7	592	19065042,7172	2023-07-18 05:35:32.283
4	76B9476D-D2BF-4959-8DAB-D361C8CBC1F4	Product 1412354	Бренд 11729	Ручна робота	7	510	18748380,4988	2023-09-16 19:32:58.260
5	B30F22E9-8B59-4AB1-A4CB-B8CD83DA5C8E	Product 347323	Бренд 109719	Одяг	10	679	18424168,2165	2023-03-24 17:03:57.573
6	69A0CE0A-12F9-4AAA-973E-BD1FEE1D5612	Product 938442	Бренд 80050	Сумки та валізи	7	589	18288317,3502	2023-07-23 20:11:57.110
7	F3EE089C-5444-4FC4-832D-DB20440A0365	Product 156445	Бренд 69804	Подарункові картки	8	583	18273231,2169	2023-02-04 04:43:24.767
8	20502B95-2FF7-4C05-9681-3F6DF94BE2AF	Product 627845	Бренд 188486	Ручна робота	6	491	18179549,531	2023-05-12 17:13:32.460
9	789BBF78-519D-4721-9554-D3F4E3776AF5	Product 215547	Бренд 127226	Пристрої та аксесуари	7	540	18167022,7614	2023-09-15 18:55:24.107
10	980D29E6-A8EE-4AA8-9BE7-C8122157F32C	Product 770182	Бренд 147808	Музичні інструменти та DJ	8	591	18052238,998	2023-11-16 19:37:15.257
11	42EB5A41-F613-4ABF-80E6-B8D89A036FB2	Product 1609723	Бренд 116420	Великі побутові прилади	7	573	18006823,0186	2023-12-02 16:02:44.627
12	5DC37B13-3082-4E31-A7CE-E8A9DEB93E7A	Product 923223	Бренд 120150	Дім та кухня	7	612	17975780,807	2023-01-17 13:13:41.590
13	2A2DB5D6-DAA7-4247-946D-5931BE9289F7	Product 972461	Бренд 2175	Подарункові картки	6	489	17972833,9541	2023-08-21 07:52:09.567
14	4286B51B-1FE4-4D2D-8E88-3A9218630C81	Product 514551	Бренд 70150	Здоров"я та особистий догляд	8	590	17851571,6565	2023-05-09 18:09:06.430
15	17F76136-6F22-4D1E-B156-A07C67124A41	Product 213120	Бренд 83357	Годинники	6	539	17847712,1139	2023-11-13 09:53:46.667
16	BBF6C038-DD04-4824-9F2C-C43475D5097F	Product 149626	Бренд 155305	Товари для домашніх тварин	6	498	17791458,5902	2023-12-02 08:28:06.653
17	6FACDF86-5F93-429A-BA14-E9B770DE6947	Product 1076730	Бренд 72144	Сад та природа	12	685	17756415,2567	2023-07-07 04:51:29.650
18	D654FDF4-2E9B-4528-B9F6-018FDCF5C65D	Product 1627935	Бренд 61849	Цифрова музика	8	584	17696182,0874	2023-06-23 15:20:21.427
19	5F7445AF-1F2D-4820-8EAB-4CC1F8F1574A	Product 988623	Бренд 75113	Дім та кухня	9	631	17655980,9429	2023-11-16 21:31:26.530

Рисунок 2.16 – Результати загальних продажів із деталізацією по брендах і категоріях

Це дозволяє зрозуміти, які бренди є найбільш популярними в різних категоріях товарів. На основі цих даних можна приймати рішення про формування асортименту, плануванні маркетингових заходів та коригуванні логістики.

Для наочного узагальнення результатів тестування розроблених аналітичних SQL-запитів бази даних на локальному ресурсі було сформовано таблицю 2.13.

Таблиця 2.13 – Результати тестування аналітичних SQL-запитів

Запит	Опис запиту	Час виконання, сек.	Кількість записів
Q1	Аналіз товарів у розрізі брендів і категорій	47,18	1 745 639
Q2	Виявлення топ-10 найбільш продаваних товарів	4,41	10
Q3	Пошук товарів з недостатньою кількістю на складах	1,66	106 123
Q4	Аналіз клієнтської бази і виявлення найцінніших клієнтів	11,62	950 073
Q5	Аналіз продажів співробітників	4,28	249 623
Q6	Топ-5 категорій за обсягом продажів	3,46	5
Q7	Аналіз замовлень за категоріями, містами та філіями	12,75	359 516
Q8	Підсумкова статистика виконаних замовлень по філіях	5,16	100
Q9	Загальний запит продажів із деталізацією по брендах і категоріях	34,01	2 000 000
Усього		90,52	5 411 089

Таким чином, усі розроблені запити успішно справились зі поставленими аналітичними завданнями та повернули необхідні обсяги бізнес-інформації за прийнятний час.

Для наочного представлення результатів тестування побудовано кругову діаграму часу обробки кожного з аналітичних запитів (рисунок 2.17).

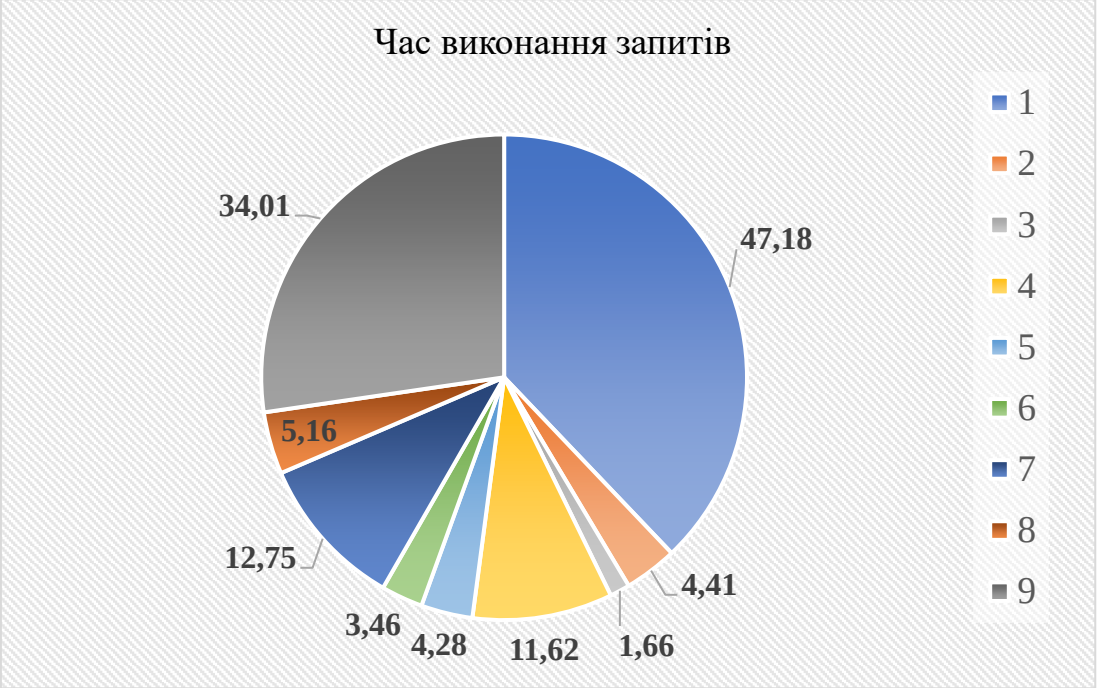

Рисунок 2.17 – Графік часу виконання запитів

Як видно, найбільшу частку за часом обробки займає Запит 1 - аналіз товарів у розрізі брендів і категорій. Його час виконання складає 47,18 сек. або 38% від загального часу обробки всіх запитів.

Це пояснюється тим, що даний запит аналізує увесь масив даних по продажам і повертає один з найбільших обсягів результатів – понад 1,7 млн. (1 700 000) записів.

Другим за тривалістю обробки є Запит 9 – загальний запит продажів з деталізацією по брендах і категоріях. Його час складає 34,01 сек. або 27% загального часу. Це також один з найбільш ресурсноємних запитів за обсягом отриманихданих.

Решта – 7 запитів – займають порівняно невелику частку загального часу – від 1% до 10%.

Отже, графічне зображення добре демонструє, які саме запити є найбільш ресурсновитратними при обробці на локальному сервері СУБД.

Для більш повного аналізу було також побудовано комбіновану діаграму, що відображає час виконання запитів та кількість повернутих ними записів (рисунок 2.18).

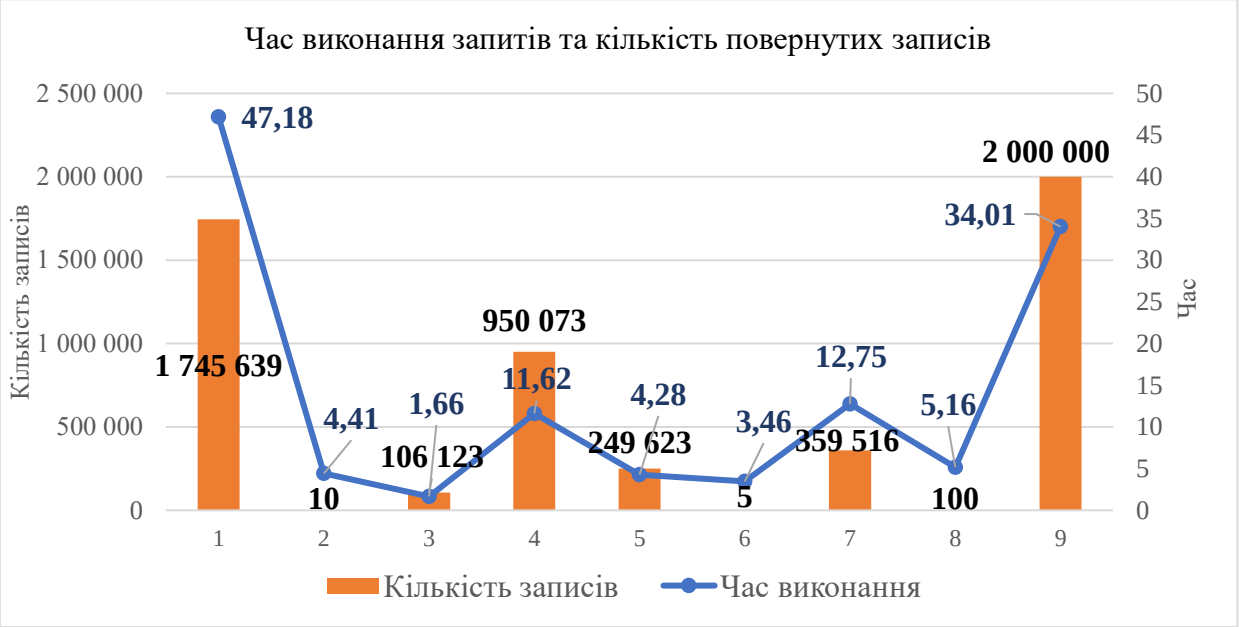

Рисунок 2.18 – Графік часу виконання запитів та кількість повернутих записів за запитами

На цій діаграмі чітко простежується кореляція між зазначеними показниками – зі збільшенням кількості записів у результаті зростає і час обробки запиту. Але в деяких випадках збільшення кількості записів не обов'язково призводить до пропорційного зростання часу обробки.

Як приклад можна навести Запит 3 та Запит 6. Запит 3 повертає 106 123 записів за 1,66 сек., тоді як Запит 6 повертає лише 5 записів але за більший час – 3,46 сек. Така ситуація пояснюється тим, що Запит 3 виконує простий пошук та фільтрацію по умові з нескладними об'єднаннями. В той час як Запит 6 агрегує дані за категоріями з підрахунками та сортуванням, що потребує більше ресурсів для його виконання. Тобто, на швидкодію також, крім кількості записів, впливає складність запиту, наявність агрегатних функцій, кількість з'єднань таблиць.

Найбільше значення по обох вісях мають запити 1 та 9 (рис. 2.36), що інтегрально аналізують увесь масив даних у різних розрізах. Вони ж і займають найбільший час на оброблення. Найменші значення за обома показниками у запитів, орієнтованих на вузькі аналітичні завдання з невеликим обсягом даних -внаслідок чого, їх швидкодія також є найвищою.

Таким чином, комбінована діаграма наглядно демонструє, що саме кількість даних для аналізу є визначальним фактором швидкодії SQL-запитів в даному випадку.

Такі показники свідчать, що розроблена база даних, її структура є ефективними і дозволяють оперативно виконувати складні аналітичні запити на локальному сервері зі значними обсягами тестових даних.

Отже, локальна СУБД цілком відповідає поставленим вимогам щодо швидкодії аналітичних запитів. Це створює базу порівняння для подальшої оцінки можливостей реалізації га хмарних платфорах.

Структура та вміст прототипу відповідають реальним вимогам сучасних торгівельних компаній. Модель охоплює всі ключові сутності та зв'язки між ними, необхідні для комплексного аналізу товарів, клієнтів, фінансів, логістики тощо.

За допомогою скриптів SQL було згенеровано репрезентативний обсяг тестових даних, достатній для моделювання даних щодо реального масштабу бізнесу торгівельної компанії та подальшої оцінки хмарних платформ.

Розроблено набір SQL-запитів для комплексного аналізу згенерованих даних і вирішення типових задач: виявлення топ-продуктів, оцінка ефективності працівників, аналіз попиту по регіонах тощо. Ці запити демонструють високу продуктивність обробки на локальному сервері бази даних з тестовим набором інформації.

Отже, розроблена БД з попередньо згенерованими тестовими даними та набором аналітичних SQL-запитів є основою для подальшого дослідження можливостей масштабування та продуктивності хмарних обчислень при обробці великих даних. Наступним кроком є розгортання БД на хмарних платформах провідних вендорів цих послуг, виконання набору експериментів по обробці даних та порівняльний аналіз отриманих результатів.

3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОДУКТИВНОСТІ ХМАРНОЇ БАЗИ ДАНИХ AZURE

3.1 Огляд рівнів моделі DTU сервісу Azure SQL Database

Azure SQL Database – це повністю керована служба бази даних, яка надає широкі можливості масштабування та високу доступність [49].

Azure SQL Database надає гнучкі можливості налаштування обчислювальних ресурсів і продуктивності відповідно до потреб бізнесу. Це досягається завдяки реалізації кількох рівнів обслуговування з різними характеристиками [50].

Вибір оптимального рівня сервісу є важливим рішенням, адже від нього залежить, наскільки ефективно використовуватимуться ресурси та кошти. Занадто потужний і дорогий рівень для легкого навантаження означитиме переплату, а недостатній рівень – призведе до проблем з продуктивністю.

Azure SQL Database пропонує три основні рівні сервісу: базовий, стандартний та преміум. Кожен рівень сервісу має свої характеристики та сфери застосування.

Базовий рівень (Basic) сервісу надає мінімальну процесорну потужність та обсяг пам’яті, але він ідеально підходить для невеликих застосунків, які не вимагають високої продуктивності. Це можуть бути прості веб-застосунки, малі особисті проекти або застосунки для тестування та розробки. Базовий рівень сервісу також надає обмежену кількість DTU, що робить його більш економічно ефективним для застосунків з низьким навантаженням.

Стандартний рівень (Standart) сервісу надає більше процесорної потужності та обсягу пам’яті, ніж базовий рівень, і він підходить для більшості загальних застосунків. Це можуть бути веб-застосунки середнього розміру, легкі OLTP (online transaction processing) застосунки або застосунки, які вимагають більшої надійності та продуктивності, ніж базовий рівень може надати. Стандартний рівень сервісу також надає більше DTU, що дозволяє обробляти більше транзакцій за одиницю часу.

Рівень Преміум (Premium) сервісу надає найбільшу процесорну потужність та обсяг пам’яті, й отже він підходить для високонавантажених застосунків та баз

даних з великими обсягами даних. Це можуть бути великі веб-застосунки, великі OLTP застосунки, або будь-які інші застосунки, які вимагають високої продуктивності та великого обсягу пам’яті. Преміум рівень сервісу надає найбільше DTU, що дозволяє обробляти найбільшу кількість транзакцій за одиницю часу.

Таким чином, Базовий рівень підходить переважно для невеликих тестових чи особистих проєктів, в той час як Преміум орієнтований на високонавантажені системи зі значними обсягами даних.

Для даного дослідження оптимальним вибором є Стандартний рівень сервісу БД Azure SQL Database. Він забезпечує гарне співвідношення обчислювальних потужностей, продуктивності та вартості для різноманітних робочих навантажень середніх масштабів.

Стандартний рівень можу бути використаний для веб-додатків, комерційних систем з помірним трафіком та навантаженнями. Саме такий сценарій відповідає задачам даного дослідження.

Стандартний рівень включає кілька конфігурацій ресурсів – S0, S1, S2 і т.д. (таблиця 3.1) [50].

Таблиця 3.1 – Характеристики ресурсів для рівнів моделі DTU

Рівень	Кількість DTU	Включений обсяг сховища, ГБ	Макс. обсяг сховища, ГБ	Макс. кількість запитів, які виконуються одночасно, сек.	Макс. кількість одночасних сеансів
S0	10	250	250	60	600
S1	20	250	250	90	900
S2	50	250	250	120	1200
S3	100	250	250, 500, 750, 1024	200	2400
S4	200	250	250, 500, 750, 1024	400	4800
S6	400	250	250, 500, 750, 1024	800	9600
S7	800	250	250, 500, 750, 1024	1600	19200
S9	1600	250	250, 500, 750, 1024	3200	30000
S12	3000	250	250, 500, 750, 1024	6000	30000

Стандартний рівень Azure SQL Database надає найкраще співвідношення можливостей та витрат для побудови аналітичного рішення в торгівельних компаніях середнього та великого бізнесу.

Критерії вибору оптимального рівня сервісу:

– критерії за продуктивністю (DTU). Продуктивність БД визначається кількістю DTU (Database Transaction Units), які виділено для обраного рівня моделі;

– критерії за вартістю. Вартість зростає відповідно до кількості DTU та додаткових опцій;

– критерії за типом навантаження. Якщо переважає читання даних - підійде базовий рівень. Аналітичні системи потребують використання стандартного рівня або вищого, активні транзакційні системи - преміум рівень.

Вибір рівня сервісу є важливим для оптимізації продуктивності та вартості Azure SQL Database. Враховуючи потреби (вимоги) в продуктивності, вартості та навантаженні, треба вибрати такий рівень сервісу, який найкраще відповідає потребам та відповідним вимогам щодо якості обслуговування.

Отже правильний вибір рівня дозволяє оптимізувати витрати і продуктивність під час використання сервісу Azure SQL Database.

3.2 Метрики продуктивності

Ефективність роботи сервісів обробки даних на хмарних платформах, зокрема сервісів роботи з реляційними базами даних, значною мірою визначається їх продуктивністю. Продуктивність даного сервісу можна оцінити за допомогою набору кількісних метрик, що характеризують швидкість обробки запитів, використання обчислювальних ресурсів, можливості масштабування та інші аспекти.

Ці метрики включають час відгуку запитів, використання CPU (Central Processing Unit), використання DTU (Database Transaction Units) та масштабованість. Час відгуку запитів вимірює, як швидко база даних реагує на запити користувачів. Використання CPU та DTU відображає, наскільки ефективно використовуються обчислювальні ресурси. Масштабованість вказує на здатність системи

адаптуватися до збільшення навантаження шляхом додавання додаткових екземплярів БД або збільшення ресурсів (наприклад, кількості DTU, об’єму ОП).

Аналіз метрик дасть уявлення про ефективність використання ресурсів сервісом та його продуктивність при різних режимах роботи, що важливо для подальшого порівняння та вибору оптимальної конфігурації.

Отже, оцінка продуктивності за допомогою ключових метрик є важливим етапом при розгортанні та експлуатації бази даних на хмарних платформах. Розглянемо ці метрики детальніше.

3.2.1 Час відгуку запитів

Час відгуку запитів – це важлива метрика, яка вимірює час, що потрібен базі даних для обробки та відповіді на запит. Він включає в себе час, необхідний для обробки запиту сервером, час передачі даних між сервером та клієнтом, а також час, який потрібен клієнту для обробки відповіді.

Ця метрика важлива, оскільки вона безпосередньо впливає на загальну продуктивність бази даних та користувацький досвід. Чим швидше база даних може обробити та відповісти на запит, тим краще користувачі можуть використовувати додаток. З іншого боку, повільний час відгуку може призвести до зниження продуктивності та незадоволення користувачів.

Оскільки час відгуку впливає на багато аспектів роботи бази даних, він часто використовується як ключовий показник ефективності (KPI) при оцінці продуктивності бази даних. Оптимізація часу відгуку може значно покращити загальну продуктивність бази даних та користувацький досвід. З цієї причини розуміння того, що впливає на час відгуку, та як його можна виміряти та оптимізувати, є важливим для ефективного управління базами даних.

Час відгуку запитів можна розділити на два основні компоненти:

1. Час отримання запиту – це час, який витрачає БД на отримання запиту від клієнта.
2. Час виконання запиту – це час, який витрачає БД на виконання запиту.

Час отримання запиту зазвичай є незначним і не впливає на загальний час відгуку запиту. Однак, в деяких випадках, він може бути значним, наприклад, якщо БД розташована далеко від клієнта.

Час виконання запиту в Azure Database залежить від багатьох факторів, включаючи:

1. Рівень моделі сервісу БД (Basic, Standard, Premium): рівень моделі сервісу бази даних визначає обчислювальні ресурси, які доступні для вашої бази даних. Різні рівні моделі сервісу надають різні обчислювальні можливості, впливаючи на продуктивність та час відгуку. Наприклад, рівень сервісу «Premium» надає більше обчислювальних ресурсів, ніж рівні «Basic» або «Standard», що може привести до швидшого часу відгуку, але в той же час рівень «Premium» коштує набагато більше ніж рівні «Basic» або «Standard».

2. Розмір обчислювальних ресурсів: кількість ядер CPU, ОП: обсяг обчислювальних ресурсів, таких як ядра процесора та оперативна пам’ять, тощо безпосередньо впливає на швидкість виконання запитів та їх обробку. Більше ресурсів може привести до швидшого виконання запитів.

3. Складність запитів: складність SQL-запиту може значно вплинути на час відгуку. Запити з великою кількістю умов, та складних обчислень можуть вимагати більше часу для обробки.

4. Латентність мережі: латентність мережі відноситься до часу, який потрібен для передачі даних між сервером та клієнтом. Затримки через мережу можуть додатково збільшувати час відгуку. Це особливо важливо для розподілених систем, де дані можуть передаватися через декілька мережевих вузлів.

5. Розмір даних: запити, які обробляють великі обсяги даних, зазвичай займають більше часу для виконання, ніж запити, які обробляють невеликі обсяги даних.

Час відгуку запитів є важливою метрикою продуктивності бази даних, оскільки він впливає на ряд ключових аспектів:

– користувачі очікують швидкого відгуку від додатків, і будь-яка затримка може призвести до фрустрації та незадоволення. В кінцевому підсумку, це може вплинути на задоволеність користувачів;

– вартість: якщо база даних витрачає багато часу на виконання запитів, це може призвести до підвищення витрат на хостинг бази даних. Більше часу на обробку запитів означає, що потрібно більше обчислювальних ресурсів, що може збільшити вартість хостингу;

– надійність: якщо база даних не може обробляти запити вчасно, це може призвести до втрати даних або відмови додатків.

Рекомендації щодо оптимізації часу відгуку:

– вибір правильного рівня сервісу БД: рівень сервісу повинен відповідати потребам застосунку;

– масштабування обчислювальних ресурсів: якщо БД стикається з високим навантаженням, потрібно збільшити обчислювальні ресурси;

– оптимізація запитів: треба використовувати оптимізовані запити для зменшення часу відгуку;

– групування запитів у пакети за критеріями (час виконання, використання ресурсів тощо) для оптимізації послідовності їх виконання.

Оптимальні параметри для мінімізації часу відгуку включають правильний вибір рівня сервісу, достатній розмір обчислювальних ресурсів, оптимізацію запитів та групування запитів.

Отже, для часу виконання запитів потрібен комплексний підхід з урахуванням усіх цих факторів.

3.2.2 Використання CPU

В контексті бази даних, центральний процесор (CPU) використовується для обробки запитів, які надходять до бази даних. Кожен запит вимагає певного обсягу обчислень для його виконання – від простих операцій читання даних до більш складних операцій, таких як об’єднання таблиць, агрегація даних або виконання складних алгоритмів.

Використання CPU є важливою метрикою продуктивності бази даних. Високе використання CPU може вказувати на те, що база даних активно обробляє

запити, тоді як низьке використання CPU може вказувати на те, що база даних має вільні ресурси, які можуть бути використані для обробки додаткових запитів.

Оптимізація використання CPU є важливою частиною управління продуктивністю бази даних. Це включає в себе ряд стратегій, включаючи оптимізацію запитів для зменшення обсягу обчислень, балансування навантаження для розподілу обчислень між різними процесорами, або масштабування ресурсів для забезпечення достатньої обчислювальної потужності для обробки запитів.

Чим ефективніше використовується CPU, тим вищою є продуктивність. З іншого боку, тривале перевищення 80-90% використання може призводити до небажаної затримки запитів.

Сервіс Azure SQL Database надає вбудовані можливості моніторингу, які можна використовувати для відстеження рівня використання CPU. Ці можливості включають:

– метрики у порталі Azure – портал Azure надає доступ до таких метрик використання CPU, як середнє використання CPU, максимальне використання CPU і використання CPU за часом. Ці метрики можна використовувати для відстеження загального використання CPU БД, а також для виявлення пікових навантажень. Портал Azure також дозволяє створювати сповіщення, які будуть надсилатися, якщо використання CPU перевищить певний пороговий рівень;

– сторонні інструменти моніторингу – також можна використовувати сторонні інструменти моніторингу для відстеження використання CPU.

Ось деякі популярні сторонні інструменти моніторингу, які можна використовувати для відстеження використання CPU в Azure Database:

– Azure Monitor – це вбудована система моніторингу, яка надає широкий спектр функцій для відстеження продуктивності, доступності та безпеки ваших ресурсів Azure;

– Microsoft SQL Server Management Studio (SSMS) – це потужний інструмент управління, який можна використовувати для відстеження використання CPU, а також інших параметрів продуктивності БД.

Фактори, що впливають на використання CPU:

– кількість віртуальних ядер: віртуальні ядра – це обчислювальні ресурси, які використовуються для обробки запитів в базі даних. Кількість доступних віртуальних ядер впливає на здатність бази даних обробляти запити паралельно. Більша кількість віртуальних ядер може дозволити базі даних обробляти більше запитів одночасно, що може привести до зменшення часу відгуку;

– складність та інтенсивність запитів: запити до бази даних можуть варіюватися від простих операцій читання до складних операцій, які вимагають значних обчислень. Запити, які вимагають більше обчислень, використовують більше CPU. Оптимізація запитів для зменшення їх складності та інтенсивності може допомогти зменшити використання CPU;

– обсяг операцій вводу/виводу: операції вводу/виводу, такі як читання та запис даних, можуть використовувати значні ресурси CPU. Великий обсяг операцій вводу/виводу може призвести до збільшення використання CPU, оскільки CPU використовується для координації цих операцій. Оптимізація операцій вводу/виводу, наприклад, шляхом зменшення обсягу даних, які потрібно читати або записувати, може допомогти зменшити використання CPU;

– масштабування (рівень бази даних): масштабування бази даних, особливо в обчислювальному середовищі хмари, такому як Azure, може мати значний вплив на використання CPU. Вибір рівня бази даних, який відповідає потребам застосунку, може допомогти забезпечити оптимальне використання CPU. Наприклад, вибір вищого рівня бази даних може надати більше обчислювальних ресурсів, що може зменшити використання CPU для кожного запиту. З іншого боку, вибір нижчого рівня бази даних може зменшити вартість, але збільшити використання CPU. Отже, важливо вибрати правильний рівень бази даних для забезпечення балансу між вартістю та продуктивністю.

Використання CPU в базі даних Azure SQL Database є критичним фактором, який впливає на продуктивність та ефективність системи. Воно відображає, наскільки ефективно система використовує свої обчислювальні ресурси для обробки запитів.

Отже, контроль та аналіз використання CPU дозволяє вчасно реагувати для забезпечення максимальної продуктивності Azure SQL Database.

3.2.3 Використання DTU

Одиниця транзакцій бази даних (DTU) – це одиниця виміру, що представляє змішану міру процесора, пам'яті, читання та запису. Фізичні характеристики (процесор, пам'ять, ввід), пов'язані з кожним показником DTU, калібруються за допомогою еталону, який імітує реальне робоче навантаження бази даних [51]. Ця метрика є важливою, оскільки вона допомагає користувачам оцінити продуктивність бази даних та визначити необхідні ресурси.

DTU вимірюється на основі внутрішнього бенчмарку Azure, який включає в себе різні типи операцій бази даних [51]. Цей бенчмарк використовується для визначення «потужності» кожного рівня сервісу Azure SQL Database.

Бенчмарк вимірює продуктивність набору базових операцій бази даних, які найчастіше зустрічаються в роботі з OLTP (online transaction processing) системами [51]. Хоча бенчмарк розроблений з урахуванням хмарних обчислень, схема бази даних, заповнення даними та транзакції розроблені таким чином, щоб бути максимально репрезентативними для базових елементів, які найчастіше використовуються в OLTP системах [51].

Вимірювання DTU базується на максимальній потужності, яку база даних може використовувати: наприклад, база даних з 100 DTU може обробляти більше операцій за одиницю часу, ніж база даних з 50 DTU.

Важливо зазначити, що DTU не є прямим вимірюванням процесора, пам’яті або дискового простору. Він є агрегованою метрикою, яка враховує всі ці ресурси – кількість віртуальних ядер, швидкість операцій введення/виведення та виділений об’єм на жорсткому диску. Тому, якщо база даних використовує 100% DTU, це означає, що вона використовує максимальну доступну потужність на обраному рівні сервісу.

DTU є важливою метрикою для Azure SQL Database з таких причин:

- дозволяє прогнозувати продуктивність своїх БД;
- дозволяє контролювати витрати на свої БД;
- допомагає оптимізувати свої БД для підвищення продуктивності оброблення даних.

Використання DTU може змінюватися в залежності від різних факторів:

– кількість та інтенсивність запитів: більша кількість запитів або більш складні запити можуть використовувати більше DTU. Наприклад, запит, який вимагає складного об’єднання декількох таблиць, може використовувати більше DTU, ніж простий запит на читання;

– розмір бази даних: більші бази даних можуть вимагати більше DTU для обробки запитів. Це означає, що база даних з більшим обсягом даних може використовувати більше DTU, ніж база даних з меншим обсягом даних;

– рівень сервісу бази даних: вищі рівні сервісу надають більше DTU, що може зменшити час відгуку запитів. Наприклад, база даних на рівні «Premium» має більше DTU, ніж база даних на рівні «Standard», тому вона може обробляти більше запитів за одиницю часу.

Azure надає вбудовані інструменти для моніторингу використання DTU. Інструмент Azure Monitor включає метрики використання ресурсів, які можна переглянути в порталі Azure, та сповіщення про використання ресурсів, які можна налаштувати для отримання повідомлень, коли використання DTU перевищує певний поріг.

Оптимізація використання DTU: полягає в наступному:

– вибір правильного рівня сервісу бази даних: рівень сервісу бази даних в Azure визначає кількість доступних DTU. Вибір правильного рівня сервісу є важливим, оскільки він впливає на продуктивність та вартість бази даних. Рівень сервісу повинен відповідати потребам застосунку. Наприклад, якщо застосунок вимагає високої продуктивності та багато транзакцій за секунду, може бути потрібен вищий рівень сервісу з більшою кількістю DTU;

– оптимізація запитів: ефективні запити можуть зменшити використання DTU. Оптимізація запитів може включати в себе різні техніки, такі як використання індексів для швидкого доступу до даних, використання фільтрів для обмеження обсягу даних, які обробляються, або розбиття великих запитів на декілька менших для паралельної обробки;

Використання DTU є важливим аспектом управління продуктивністю Azure SQL Database. Правильне розуміння та управління використанням DTU може допомогти покращити продуктивність бази даних.

3.3 Порівняльний аналіз часу генерації даних на локальному ресурсі та Azure SQL Database

На першому етапі оцінки ефективності використання хмарного сервісу для обробки великих даних, треба порівняти показники генерації даних на локальному ресурсі (сервері) та на хмарному сервісі Azure SQL Database.

Однією з ключових переваг хмарних сервісів є можливість масштабування ресурсів відповідно до поточних потреб. Для того, щоб зрозуміти реальну ефективність та доцільність використання хмари, доцільно порівняти продуктивність хмарних сервісів та локальної IT-інфраструктури на конкретних задачах обробки даних.

Порівнявши ефективність генерації великих обсягів даних на локальному сервері та у хмарному середовищі Azure SQL Database на різних рівнях масштабування – від S0 до S12 – дасть можливість зробити висновки, а саме: підходу використання локального серверу та використання хмарного сервісу Azure SQL Database.

Аналіз результатів дозволить також оцінити доцільність використання хмарної платформи для завдань, пов’язаних з обробкою великих даних залежно від потреб у масштабуванні задіяних ресурсів.

3.3.1 Опис характеристик ресурсів

Ефективність генерації даних значною мірою залежить від обчислювальних можливостей платформи. В Azure SQL Database ці можливості забезпечуються за допомогою віртуальних процесорів та оперативної пам'яті, виділених для бази даних.

На відміну від фізичного обладнання, де ресурси є фіксованими, хмарне середовище дозволяє гнучко масштабувати віртуальні ресурси відповідно до поточних потреб.

Наприклад, на локальному ресурсі використовувалась оперативна пам'ять обсягом 16 ГБ та жорсткий диск SSD обсягом 475 ГБ. Коли у Azure SQL Database обсяг оперативної пам'яті може становити від 6 до 240 ГБ в залежності від обраного рівня моделі сервісу [52]. А загальний обсяг пам'яті може сягати від 250 ГБ до 1024 ГБ також залежно від рівня сервісу [52].

Також масштабування на хмарному сервісі досягається шляхом зміни кількості віртуальних ядер процесора та обсягу оперативної пам'яті. Кількість доступних віртуальних процесорів залежить від рівня сервісу, який ви обрали для вашого екземпляра Azure SQL Database.

На стандартному рівні масштабування, кількість віртуальних процесорів, які можна використовувати, може бути обмежена. Також на стандартному рівні сервісу Azure SQL Database використовується логічний процесор Gen 5 [53]. Цей тип процесора добре підходить для більшості серверів реляційних баз даних [54].

Для більш наглядного порівняння характеристик процесорів, які використовуються на локальному ресурсі та в Azure SQL Database, подано у таблиці 3.2. Це допоможе краще зрозуміти різницю між цими процесорами та їх вплив на продуктивність генерації даних.

Таблиця 3.2 – Порівняльна таблиця характеристик процесорів

Характеристика	11th Gen Intel Core i5 (локальний ПК)	Azure SQL Database Gen5
Модель процесора	Intel Core i5-11400H	Intel Xeon E5-2673 v4, Intel SP-8160, Intel 8272CL, Intel Xeon Platinum 8370C, AMD EPYC 7763v
Архітектура	Rocket Lake	Broadwell, Skylake, Cascade Lake, Ice Lake, Milan (залежно від моделі)
Кількість ядер	6	2-80 віртуальних ядер
Кількість потоків	12	2-80 (з гіперпоточенням)
Тактова частота	2.7 ГГц	Від 2.1 до 3.7 ГГц
Техпроцес	10 нм	14-7 нм
Кеш-пам'ять	12 МБ	33 МБ - 256 МБ

Хмарний сервіс Azure SQL Database має значну гнучкість у плані виділення обчислювальних ресурсів. Кількість віртуальних ядер процесора та обсяг оперативної пам'яті може динамічно змінюватися відповідно до поточних потреб користувачів. На відміну від цього, локальний сервер має фіксовану апаратну конфігурацію, яка не може бути змінена.

Тому для невеликих обсягів даних локальний ПК може забезпечити достатню продуктивність. Проте у разі збільшення потреб у обчислювальних ресурсах, хмарний сервіс буде мати значну перевагу завдяки можливості масштабування віртуальних ресурсів відповідно до вимог продуктивності.

Це дозволить Azure SQL Database краще виконувати завдання, пов'язані з обробкою великих обсягів даних – наприклад, під час масштабування розмірів БД, кількості запитів, кількості одночасних сеансів тощо – на відміну від обмежень локального середовища.

3.3.2 Методологія порівняння

Для того, щоб об'єктивно порівняти ефективність генерації даних на локальному та хмарному ресурсах потрібно визначити чіткі критерії та метрики такого порівняння. Ключовими метриками мають бути показники швидкості та обсягів генерованих даних, а також використання ресурсів системи в процесі генерації.

Далі наведено детальний перелік метрик, які дозволять всебічно оцінити та порівняти ефективність генерації даних на двох платформах.

Порівняння продуктивності генерації даних на локальному ресурсі та Azure SQL DB буде виконано за допомогою кількісної метрики як час. Час, необхідний для створення визначеної кількості даних, є основною метрикою, яка дозволяє оцінити швидкодію та продуктивність кожної платформи. Чим менше часу потрібно для генерації заданого обсягу даних, тим вища продуктивність платформи.

Для комплексного аналізу буде використано показник використання обчислювальних ресурсів (відсоток завантаження CPU та DTU) під час генерації та вартість ресурсів для генерації даних ($). Ці метрики важливі для оцінки ефективності

використання ресурсів та економічної ефективності використання хмарної платформи порівняно з використанням локального ресурсу.

Генерація даних виконуватиметься за допомогою розробленого у другому розділі генератора даних у вигляді SQL-запитів. Для локального та хмарного середовищ будуть використані однакові набори запитів для коректного порівняння. Це включає в себе використання однакових параметрів запиту, таких як розмір вибірки, кількість полів та інші параметри, що впливають на обсяг даних та складність запиту.

3.3.3 Порівняльний аналіз показників генерації даних

Метою даного порівняльного аналізу є всебічна оцінка ефективності генерації даних в хмарному середовищі Azure SQL Database порівняно з локальним сервером, а також між різними рівнями масштабування. Аналізується вплив масштабування ресурсів на продуктивність генерації даних.

Масштабування обчислювальних ресурсів є важливою перевагою хмарних технологій. Однак, на практиці, збільшення виділених ресурсів не завжди призводить до пропорційного прискорення виконання запитів.

Для оцінки масштабованості хмарної платформи Azure SQL Database, важливо проаналізувати, як змінюється продуктивність генерації даних при зміні обсягів виділених ресурсів.

Аналіз ключових показників, таких як обсяги та час генерування даних, дозволить зробити висновок про ефективність використання віртуальних ресурсів в хмарі. Порівняння цих показників з можливостями локального сервера допоможе ілюструвати переваги хмари в контексті гнучкого нарощування потужностей.

Отже, наступним етапом аналізу буде дослідження ефективності використання різних конфігурацій віртуальних ресурсів всередині самої хмарної платформи Azure SQL Database. Тому важливо проаналізувати оптимальне співвідношення витрат на ресурси та отриманого приросту продуктивності.

Отже, на наступному етапі дослідження треба провести порівняльний аналіз показників генерації даних на різних рівнях виділення ресурсів в Azure SQL Database – від мінімального рівня S0 до максимального – S12.

Це дозволить оцінити динаміку змін продуктивності генерації даних залежно від кількості використаних ресурсів хмарної платформи.

Для комплексного аналізу було обрано 5 рівнів моделі Azure SQL Database (Standart): базовий S0, середні S1 і S3, високі S7 і максимальний S12. Такий вибірок дозволить проаналізувати зміну ефективності в широкому діапазоні конфігурацій – від мінімальної до максимальної.

Водночас, повний аналіз абсолютно всіх можливих 13 рівнів потужності призвів би до надмірної деталізації. Адже приріст продуктивності між сусідніми рівнями, особливо в середньому діапазоні, є не дуже великим. Тому обрана вибірка 5 ключових рівнів є оптимальною для отримання загальної картини та проведення аналізу.

Порівняння використання обраних рівнів моделі за ключовими характеристиками представлено у таблиці 3.3.

Таблиця 3.3 – Порівняльний аналіз показників ефективності для різних рівнів моделі при генерації даних БД

Рівень	Час, год.	Середнє DTU, %	Середнє CPU, %
S0	05:29:31	95	89
S1	02:00:01	89	74
S3	00:30:58	61	61
S7	00:14:02	35	21
S12	00:12:44	13	8,3

Як видно, з підвищенням рівня моделі час генерації даних зменшується за рахунок використання більш потужних обчислювальних ресурсів, водночас використання DTU та завантаженість CPU знижуються, що свідчить про збільшення ефективності застосування потужностей на вищих рівнях моделі.

Отже, на основі отриманих результатів можна зробити висновок, що збільшення виділених ресурсів в Azure SQL Database призводить до прискорення генерації даних, але ж при цьому ця залежність не є лінійно пропорційною.

Для наочної демонстрації цієї закономірності на рисунку 3.1 представлено час, необхідний для генерації даних на різних рівнях моделі Azure SQL Database.

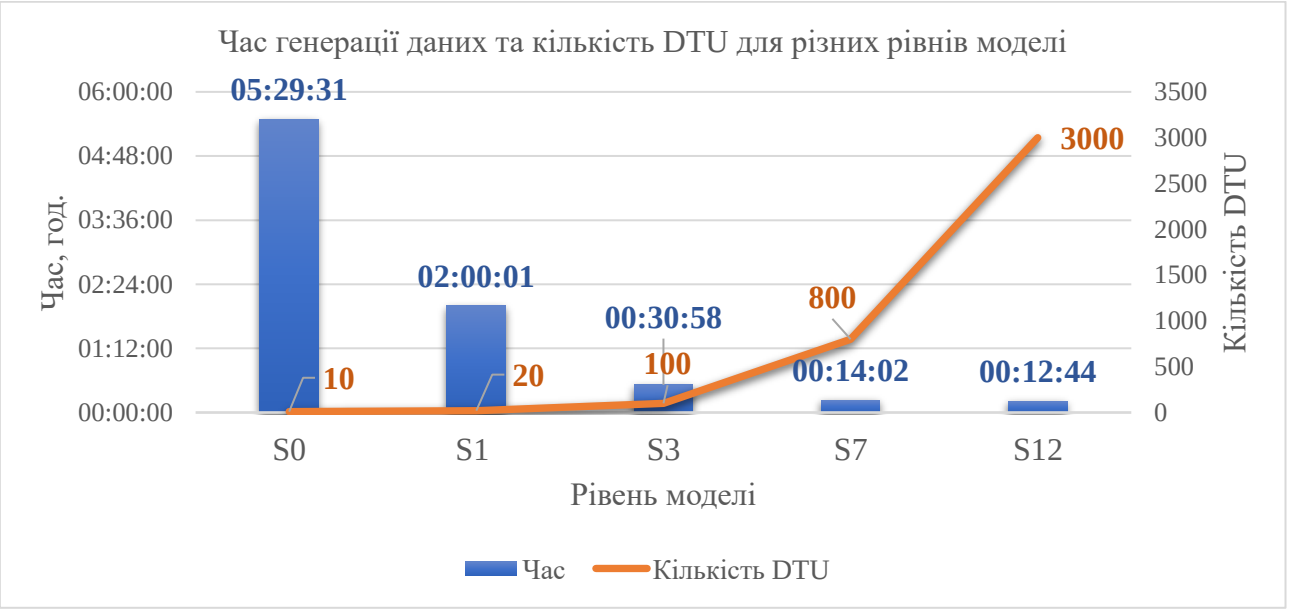

Рисунок 3.1 – Графік залежності часу генерації та кількості DTU для рівнів моделі

Отже, з підвищенням рівня потужності час генерації зменшується. Це свідчить про те, що збільшення обчислювальних ресурсів призводить до прискорення генерації даних.

На нижчому рівні S0 генерація зайняла понад 5 годин через обмежені ресурси. Підвищення рівня до S3 дозволило зменшити час приблизно в 10 разів завдяки більшій кількості DTU.

В подальшому спостерігається уповільнення темпу зростання продуктивності. Зокрема, різниця між часом генерації на рівнях S7 та S12 є не дуже значною (14 хв. проти 12 хв. відповідно). Це можна пояснити тим, що використані SQL-запити для генерації є оптимізованими під конкретні обсяги даних і не потребують задіяння усієї потужності максимального рівня S12 для повної роботи.

Отже, на основі аналізу можна зробити висновок, що масштабування ресурсів в хмарі не завжди є раціональним, якщо навантаження не дуже велике. Існує оптимальний баланс вартості та продуктивності (ефективності), вихід за межі якого

призводить до невиправданих надмірних витрат коштів. Комплексний аналіз має допомогти його визначити.

Висновок про існування оптимального компромісу вартості та ефективності підтверджується даними про вартість використання різних конфігурацій Azure SQL Database за весь час генерації даних.

Загальну вартість роботи кожного рівня потужності з урахуванням часу, протягом якого було виділено відповідні ресурси, подано на рисунку 3.2.

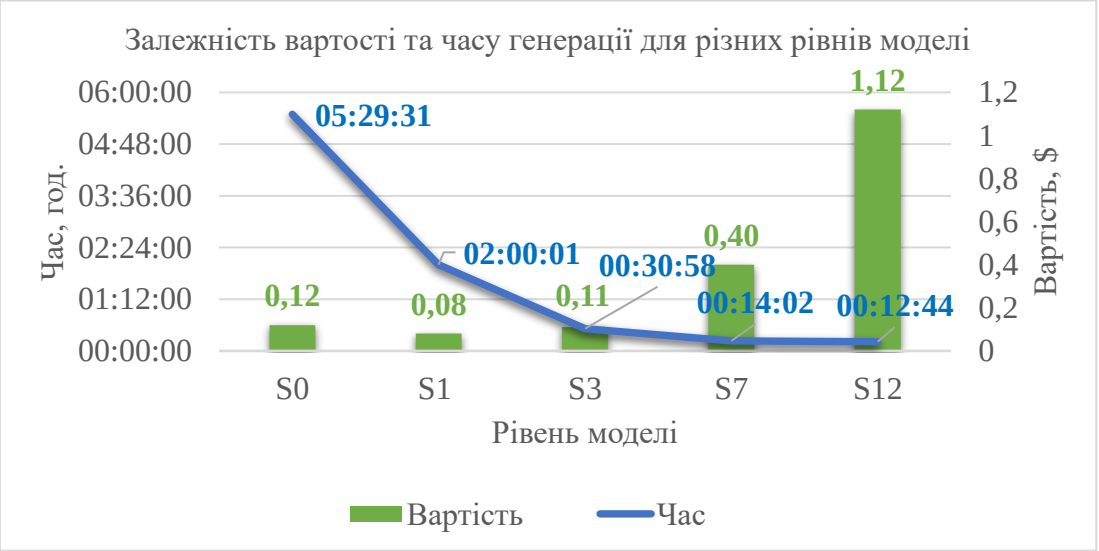

Рисунок 3.2 – Графік вартості та часу генерації даних для різних рівнів моделі

Діаграма витрат за рівнями надає важливий інструмент для прийняття рішень щодо використання обчислювальних ресурсів з урахуванням ефективності та бюджетних обмежень.

Аналізуючи графік, можна зробити висновки.

– рівень S0 забезпечує найнижчу вартість за одну годину (2 цента), але через те що генерація зайняла більше 5 годин уся сума вийшла у 12 центів за весь період генерації. Тому S0 можна вважати неоптимальним порівняно з S1;

– рівень S1 забезпечує суттєве прискорення генерації даних у 2,5 рази (250 %) порівняно з S0, при цьому додаткові витрати склали лише 8 центів за увесь час генерації. Тому рівень S1 є більш ефективним ніж S0;

– рівень S3 показав найкраще співвідношення помірної вартості (11 центів) та швидкої генерації даних (31 хв.). Рівень S3 може бути привабливим для тих, хто шукає баланс між продуктивністю та вартістю;

– рівень S7 відрізняється малим часом генерації (14 хв.), проте вартість за його використання вища (40 центів), порівняно з іншими рівнями. Використання S7 може бути виправданим у випадках, коли пріоритетом є максимальна швидкість генерації даних, і вартість є менш суттєвим фактором. З іншого боку, вартість може бути важливим параметром для проектів з обмеженим бюджетом, і у таких випадках рівень S7 може бути визначений як менш ефективний;

– рівень S12 поруч зі швидкістю генерації у 12 хвилини показує найвищу вартість – $1.12. Це майже у три рази більше ніж S7 при незначному прискоренні роботи. Таким чином, незважаючи на найвищу продуктивність, S12 є занадто витратним рішенням для генерації, оскільки процес не вимагає стільки ресурсів, які надає рівень S12. У випадку ще більшого навантаження, використання рівня S12 могло б бути доцільним.

Отже, для остаточного висновку стосовно вибору оптимального рівня необхідно додатково проаналізувати у порівнянні з локальним ресурсом. Перед цим треба проаналізувати використання системних ресурсів (DTU та CPU) на різних рівнях моделі Azure SQL Database під час генерації даних (рисунок 3.3).

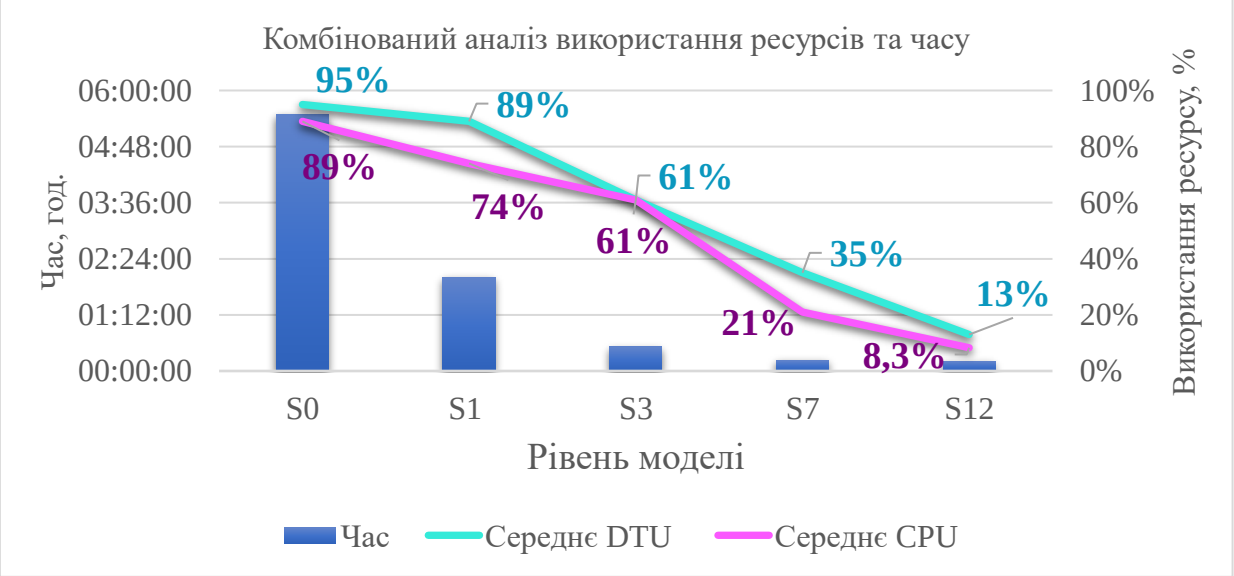

Рисунок 3.3 – Графік комбінованого аналізу використання ресурсів та часу при генерації даних

Представлений графік демонструє середній рівень використання двох ключових системних ресурсів – DTU та CPU на різних рівнях моделі Azure SQL Database під час генерації заданого обсягу даних.

На базовому рівні S0 спостерігається максимальне середнє навантаження як DTU так і CPU – 95% та 89% відповідно. Це свідчить про те, що ресурси S0 працювали на межі своїх можливостей, що призвело до тривалого часу генерації даних (понад 5 год.).

Підвищення рівня використання ресурсів дозволяє знизити середнє навантаження, оскільки завдання має більший запас потужності. На рівні S3 середнє значення використання DTU та CPU практично збігаються на позначці 61%. Це свідчить про стабільну роботу системи та загальне ефективне використання ресурсів.

Подальше підвищення рівня призводить до різкого зниження завантаження ресурсів, оскільки їх виділено набагато більше, ніж потрібно для поточних потреб генерації даних: на рівні S12 середнє значення DTU становить лише 13%, а CPU – 8,3%.

Таким чином, при виборі рівня моделі для генерації даних важливо враховувати не тільки час генерації, але й ефективність використання ресурсів. Оптимальний вибір залежить від конкретних потреб та обмежень.

Для остаточного висновку також важливо порівняти обрану конфігурацію хмарних ресурсів з можливостями локального ресурсу: локальне середовище також має певну продуктивність, яку необхідно врахувати для формування рекомендацій.

Для порівняння були використані рівні S3 та S7, оскільки вони продемонстрували найбільш оптимальне поєднання часу та ефективності використання ресурсів в сервісі Azure SQL Database.

Для наочного порівняння продуктивності обраних конфігурацій хмарних ресурсів (рівнів S3 та S7) та локального ресурсу побудовано діаграму часу, необхідного для генерації даних до основних таблиць бази даних.

На рисунку 3.4 представлено порівняння середнього часу генерації даних до кожної таблиці бази даних (див. розділ 2) на рінях S3 та S7 і на локальному ресурсі.

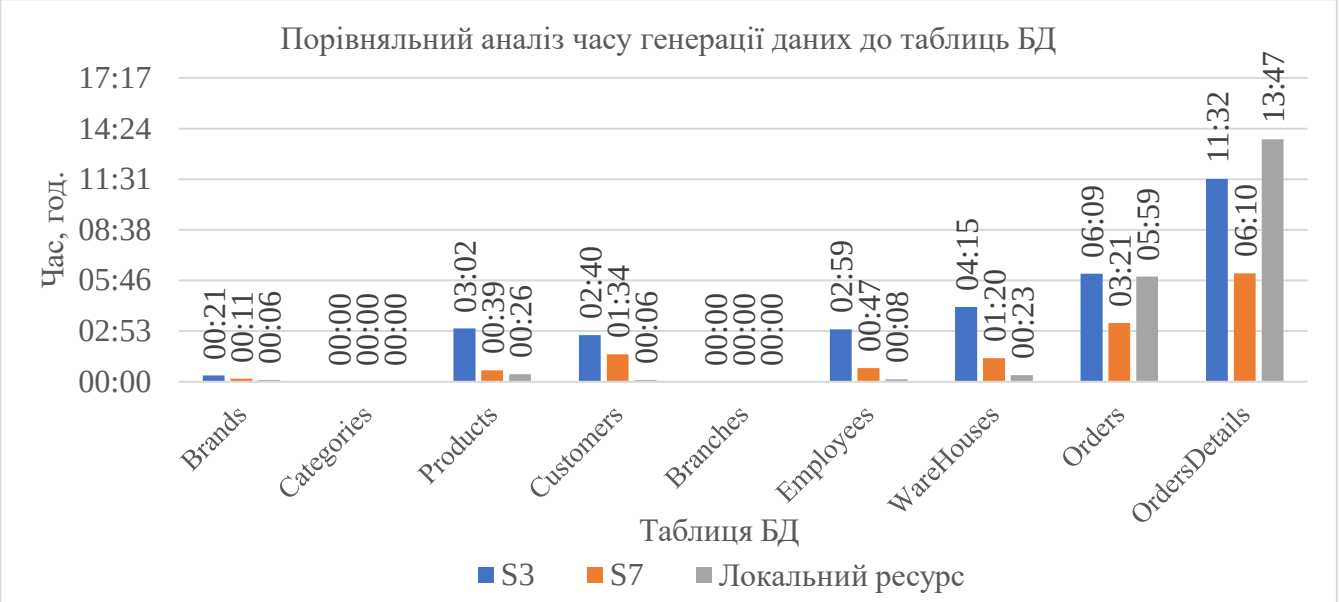

Рисунок 3.4 – Порівняльний аналіз часу генерації даних по таблицях БД на локальному ресурсі та для різних рівнів моделі

Аналізуючи цей графік, можна зробити висновок, що для простих запитів на генерацію даних до невеликих за розміром таблиць (Brands, Categories, Customers, Branches, Employees) локальний ресурс справляється швидше за хмарний сервіс. Це пов’язано з додатковим часом, який потрібен для комунікації між хмарою та локальним ресурсом, а також з особливостями розподіленого обчислення на хмарній платформі.

У той же час, для складніших запитів на генерацію, які потребують значних обчислювальних ресурсів, хмарний сервіс демонструє кращу продуктивність. Це пов’язано з тим, що хмарні ресурси мають більше обчислювальної потужності та можуть масштабуватися для обробки великих обсягів даних, в той час як локальні ресурси обмежені щодо їх доступної потужності.

Хмарні ресурси можуть бути більш ефективними для виконання затратних запитів, ніж локальні ресурси, незважаючи на додатковий час, який потрібен для комунікації між хмарою та локальним ресурсом. Однак, для запитів, які потребують менше ресурсів, локальні ресурси можуть бути більш ефективними.

Таким чином, при виборі між використанням хмарних або локальних ресурсів важливо враховувати специфіку конкретного завдання. Для великих, затратних запитів хмарні ресурси можуть бути більш вигідними, тоді як для менш затратних запитів локальні ресурси можуть бути оптимальним рішенням.

Здійснивши аналіз порівняння генерації даних у локальному середовищі та хмарній платформі Azure SQL Database, можна констатувати наступне.

Хмарна платформа демонструє значні переваги завдяки можливості гнучкого масштабування обчислювальних ресурсів відповідно до поточних потреб. Це надає змогу оптимізувати витрати компанії та підвищити ефективність обробки даних для різноманітних аналітичних завдань.

Експериментальним шляхом встановлено, що продуктивність генерації даних в Azure SQL Database зростає зі збільшенням виділених ресурсів.

Також, при аналізі та порівнянні ефективності платформ важливо не лише зважати на час генерації даних, а й брати до уваги рівень використання системних ресурсів. Адже надлишкове виділення потужностей може виявитися економічно невиправданим.

Беручи до уваги результати порівняльного аналізу, можна сформулювати наступні рекомендації щодо вибору оптимального рівня моделі для генерації даних.

При прийнятті рішення варто враховувати такі ключові фактори:

– загальний обсяг даних, який планується згенерувати. Чим більший цей обсяг, тим більш доцільно розглядати використання хмарної платформи Azure SQL Database з її масштабованістю;

– складність SQL-запитів, які планується виконувати на базі згенерованих даних. Чим складніша логіка запитів, тим краще підходить хмарне середовище;

– безпосередньо бюджет компанії на IT та аналітику. Адже використання хмарних ресурсів передбачає додаткові витрати.

Отже, для невеликих обсягів даних та простих запитів оптимальним є локальний ресурс. В той час як для великих масивів і складних запитів рекомендується саме хмарний сервіс Azure SQL Database з огляду на її гнучкі можливості масштабування об’ємів БД, кількості запитів, сеансів тощо під конкретні потреби бізнесу.

3.4 Розроблення режимів оброблення запитів

Виконання запитів до бази даних може відбуватися в інтерактивному чи пакетному режимах. Інтерактивний режим передбачає послідовне виконання окремих

запитів користувачем. Пакетний режим дозволяє виконати групу попередньо підготовлених запитів за певним алгоритмом.

Метою дослідження в цьому розділі є визначення оптимального режиму та послідовності виконання набору тестових запитів для досягнення максимальної продуктивності системи. Результатом має стати рекомендована стратегія організації запитів користувачів в хмарному сервісі Azure SQL Database.

Очікується, що ретельний аналіз режимів оброблення запитів надасть інсайт про те, як краще використовувати потужності Azure SQL Database для обробки великих даних. Визначення оптимальних режимів та їх впливу на продуктивність дозволить розробити рекомендації для ефективного використання хмарних ресурсів у сценаріях обробки великих обсягів даних.

Аналіз отриманих результатів дозволяє зробити висновки щодо оптимальної стратегії організації обробки запитів користувачів в середовищі хмарної платформи Azure.

3.4.1 Інтерактивний режим виконання запитів

Інтерактивний режим є основним режимом виконання запитів до бази даних. Він характеризується послідовним виконанням окремих запитів. Це дозволяє детально проаналізувати поведінку кожного запиту окремо та вивчити його вплив на загальну продуктивність системи.

Використання інтерактивного режиму в середовищі Azure SQL Database відкриває доступ до потужних інструментів моніторингу та діагностики, які надає ця платформа. Це допомагає краще зрозуміти, як запити використовують ресурси, та оптимізувати їх для досягнення максимальної продуктивності.

Важливо зазначити, що інтерактивний режим є особливо корисним для сценаріїв, коли потрібно вивчити вплив окремих запитів на загальну продуктивність системи. Він дозволяє відстежувати виконання кожного запиту в реальному часі та аналізувати його вплив на ресурси системи.

Для експерименту в інтерактивному режимі сформовано набір з 9 аналітичних SQL-запитів (див. розділ 2.4), які охоплюють різні аспекти бізнес-аналітики

компанії, розподілимо їх по номерам та будемо позначати далі як Q1-Q9, відповідно списку:

1. Аналіз продажів та прибутковості по брендах і категоріях продуктів.
2. Топ-10 продуктів за обсягами продажів.
3. Моніторинг залишків на складах для виявлення товарів, що потребують до закупівлі.
4. Аналіз клієнтської бази за обсягами покупок.
5. Оцінка ефективності роботи менеджерів по продажам і філіалам.
6. Топ-5 категорій за обсягом реалізованої продукції.
7. Аналіз продажів за регіонами та містами.
8. Моніторинг завантаженості відділень компанії.
9. Загальна статистика продажів по брендах і категоріями товарів.

Ці запити були розроблені на попередніх етапах дослідження та вже протестовані на локальному ресурсі.

Кожен аналітичний запит виконуватиметься в індивідуальній транзакції з фіксацією часу виконання, максимальної кількості задіяних DTU (%) та максимального використання CPU (%).

Очікується, що тестування в інтерактивному режимі дозволить визначити швидкодію та ефективність виконання аналітичних запитів в реальному часі. Звертається увага на час виконання кожного запиту, обсяг використаних ресурсів та метрики, такі як DTU та CPU, які дозволять зрозуміти вплив інтерактивного режиму на продуктивність бази даних.

Отримані показники порівняють з результатами тестувань на локальному сервері та з результатами для пакетного режиму в Azure SQL Database. Це дозволить зробити обґрунтовані висновки щодо ефективності застосування хмарної платформи для аналітичних запитів в інтерактивному режимі.

Аналіз результатів дозволить оцінити ефективність обробки аналітичних запитів Azure SQL Database в інтерактивному режимі та порівняти з виконанням на локальному сервері БД. Також можна буде виявити найбільш ресурсомісткі аналітичні запити.

3.4.2 Пакетні режими (плани)

На відміну від інтерактивного, пакетний режим передбачає виконання запитів групами (пакетами – планами) за певним алгоритмом для досягнення кращої продуктивності системи.

Для експериментів у пакетному режимі використовуватиметься та сама вибірка з 9 аналітичних запитів, що й в інтерактивному режимі (Q1-Q9). Для зручності посилань, пакети запитів позначатимуться скорочено B1-B8 (Batch1-Batch8) відповідно до порядку їх опису нижче.

Тестуватиметься такі стратегії формування та виконання пакетів (планів) запитів:

1. У довільному порядку (B1). Запити групуються у пакет довільним чином без урахування їхніх характеристик. Це найпростіший підхід, який дозволяє оцінити базову продуктивність у пакетному режимі.
2. За зростанням часу виконання запиту, визначеного в інтерактивному режимі (B2). Запити впорядковуються за зростанням часу їх обробки. Це може покращити загальну продуктивність, оскільки швидші запити виконуються раніше.
3. За спаданням часу виконання (B3). Аналогічно попередньому підходу, але зі спаданням часу виконання. Це корисно для визначення впливу більш тривалих запитів на загальну продуктивність.
4. За зростанням максимальної кількості DTU, задіяних під час виконання запиту (B4). Групування запитів за ресурсоємністю (використання DTU) за зростанням. Це може допомогти визначити, як кількість DTU впливає на продуктивність.
5. За спаданням максимальної кількості DTU (B5). Аналогічно попередньому підходу, але за спаданням ресурсоємності.
6. За зростанням максимального відсотку використання CPU: Групування за навантаженням на хмарний процесор (B6). Це може допомогти визначити, як використання CPU впливає на продуктивність.
7. За спаданням максимального відсотку використання CPU (B7). Те саме, що і попередній підхід, але у зворотньому напрямку.

8. За пріоритетом важливості запитів: Запити виконуються в порядку пріоритету важливості (B8). Це корисно для ситуацій, коли деякі запити важливіші за інші і повинні бути виконані в першу чергу.

Розглянемо детальніше питання визначення пріоритетності аналітичних SQL-запитів для подальшого порівняльного тестування в хмарному середовищі Azure SQL Database. З метою порівняння можливостей хмарної платформи Azure SQL Database щодо пріоритезації та прискореної обробки найбільш важливих аналітичних запитів, було проведено їх ранжування за рівнем значущості в рамках загальної бізнес-логіки.

Розроблені тестові SQL-запити (Q1-Q9) вирішують широке коло задач, тому їх було структуровано на 3 групи за рівнями пріоритетності:

1. Високий пріоритет: запити комплексного аналізу ключових показників бізнесу, критично важливі для прийняття стратегічних рішень - Q1, Q4 та Q9.

2. Середній пріоритет: запити оптимізації основних бізнес-процесів - логістики, HR, маркетингу – Q3, Q5, Q7.

3. Низький пріоритет: спеціалізовані маркетингові та управлінські запити - Q2, Q6, Q8.

Цей план розширить можливості порівняльного аналізу хмарного сервісу при тестуванні SQL-навантаження з урахуванням вагомості бізнес-пріоритетів підприємства (компанії).

Отже, в результаті проведеного ранжування аналітичні SQL-запити які було пронумеровано вище, розподіллено наступним чином:

Ранг 1 (Найвищий пріоритет) – до нього віднесені найбільш важливі запити, що впливають на стратегічні бізнес-рішення:

– запит 1 (Q1) виводить вичерпний аналіз продажів по брендах і категоріям. Це критично важливо для оцінки рентабельності різних продуктів, портфелю брендів, попиту по товарних напрямках;

– запит 4 (Q4) аналізує поведінку та цінність клієнтів. Ці дані є основою для будь-яких маркетингових ініціатив – сегментації, рекламних кампаній, знижок тощо;

– запит 9 (Q9) забезпечує комплексне уявлення про товарні потоки і обсяги продажів в розрізі відділень і напрямків. Це формує загальну картину бізнесу.

Ранг 2 (Середній пріоритет) – запити оптимізації основних бізнес-процесів компанії:

– запит 3 (Q3) дає актуальну інформацію про необхідність поповнення складських запасів;

– запит 5 (Q5) надає оцінку продуктивності персоналу – це необхідно для KPI, мотивації, виявлення потреб у навчанні;

– запит 7 (Q7) дозволяє аналізувати специфіку регіонального попиту для кращого розподілу товару між відділеннями та проведення промо-заходів. Ці запити спрямовані на оптимізацію ключових бізнес-процесів компанії.

Ранг 3 (Низький пріоритет) – спеціалізовані маркетингові та управлінські запити:

– запит 2 (Q2) виявляє найпопулярніші продукти для просування, аналізу поведінки клієнтів щодо них та їхньої рентабельності;

– запит 6 (Q6) відслідковує найбільш перспективні товарні напрямки для оптимізації асортименту під попит;

– запит 8 (Q8) дає загальне уявлення про ефективність філій, але не є критичним для стратегічних рішень.

Цей план розширить можливості порівняльного аналізу хмарних платформ при тестуванні SQL-навантаження з урахуванням його бізнес-пріоритетів.

Отже, результати різних стратегій пакетного виконання будуть порівняні між собою, що дозволить визначити найбільш ефективну стратегію для даного набору запитів. Це допоможе краще зрозуміти вплив режимів обробки запитів та стратегій виконання на продуктивність, і як це можна використати для підвищення ефективності роботи з базою даних.

3.5 Обробка та аналіз отриманих результатів

Ефективна обробка аналітичних запитів є ключовим фактором при виборі рішення для аналізу великих масивів даних. Правильний підбір конфігурації

хмарного середовища дозволяє суттєво прискорити виконання складних SQL-запитів та оптимізувати витрати на хмарні ресурси.

У цьому розділі представлено комплексний аналіз ефективності застосування хмарної платформи Azure SQL Database для аналітичної обробки даних в інтерактивному та пакетному режимах.

Тестування охоплює п’ять рівнів моделі сервісу: S0, S1, S3, S7, S12, що дозволяє дослідити можливості масштабування продуктивності обробки SQL-запитів для широкого спектру аналітичних завдань – від простих до складних.

Аналіз результатів тестувань спрямований на виявлення оптимальної конфігурації Azure SQL Database з точки зору співвідношення витрат на хмарні ресурси та ефективності обробки аналітичного навантаження.

Розглянуті ключові метрики продуктивності, такі як час виконання запитів, використання DTU та процесора дозволяють комплексно оцінити можливості хмарної платформи Azure при масштабуванні для виконання аналітичних задач.

Порівняльний аналіз інтерактивного та пакетного режимів виконання SQL-запитів на різних рівнях сервісу дає змогу визначити стратегію обробки аналітичного навантаження, яка забезпечить найкращу продуктивність та ефективність використання хмарних ресурсів платформи Azure.

Отримані результати та рекомендації допоможуть приймати обґрунтовані рішення щодо архітектури хмарних аналітичних рішень на базі Azure SQL Database з урахуванням бізнес-вимог до швидкодії обробки даних.

Розглянемо результати виконання інтерактивного режиму на різних рівнях сервісу та порівняємо результати між собою.

3.5.1 Аналіз результатів застосування інтерактивного режиму

Ефективність обробки аналітичних запитів у сучасних хмарних середовищах значною мірою визначається оптимальним підбором рівня сервісу бази даних. Адже саме від наявного обсягу обчислювальних ресурсів залежить швидкодія та пропускна здатність системи при виконанні складних аналітичних запитів. Метою є комплексна оцінка можливостей хмарної платформи Azure SQL Database в

інтерактивному режимі обробки аналітичного навантаження на різних рівнях моделі сервісу від S0 до S12. Це дозволить виявити оптимальну з точки зору продуктивності та вартості конфігурацію хмарних ресурсів для задач інтерактивної аналітики.

В якості тестового аналітичного навантаження використовується розроблений набір з 9 SQL-запитів (див. розділ 2.4) на хмарному сховищі даних. Кожен запит виконується в індивідуальній транзакції з фіксацією ключових метрик продуктивності обробки на кожному рівні сервісу.

Аналіз результатів дозволить виявити закономірності впливу обсягу виділених хмарних ресурсів на ефективність інтерактивної обробки аналітичних SQL-запитів та сформувати обґрунтовані рекомендації щодо оптимальної конфігурації середовища Azure SQL Database. Проаналізуємо детальні результати виконання SQL-запитів в інтерактивному режимі на кожному з рівнів сервісу Azure SQL Database - від базового S0 до ресурсоємного S12.

Перейдемо до аналізу результатів на рівні сервісу S0. На цьому мінімальному рівні, який забезпечує мінімальний обсяг обчислювальних ресурсів для оперативної обробки даних, було виконано 9 SQL-запитів. Результати виконання цих запитів представлені у таблиці 3.4.

Таблиця 3.4 – Результати виконання запитів у інтерактивному режимі на рівні S0

Запит	Середнє CPU, %	Середнє DTU,%	Час, хв.
Q1	95	100	7:28
Q2	97	93	0:54
Q3	30	51	0:07
Q4	100	93	1:20
Q5	97	78	0:31
Q6	97	51	0:40
Q7	78	94	4:31
Q8	79	84	0:53
Q9	67	100	20:43

Як видно з таблиці 3.4, на мінімальному рівні сервісу S0 спостерігається досить високе завантаження процесора (CPU) під час обробки аналітичних запитів – в середньому 82%. Це свідчить про те, що обчислювальних ресурсів, які надає рівень S0, недостатньо для швидкої обробки складних аналітичних запитів. Для більш наочного представлення даних про використання CPU, подано графік (рисунок 3.5), який відображає середнє використання CPU для кожного запиту. Цей графік допоможе зрозуміти, як ресурси процесора використовуються протягом виконання кожного запиту.

Найбільше навантаження на CPU створюють запити Q1, Q2, Q4, Q5, Q6 – майже 100% використання процесора. Запити Q7 і Q8 продемонстрували помірне навантаження на CPU на рівні 78-79%. Запит Q9 продемонстрував помірне середнє використання CPU на рівні 67%. Запит Q3 показав найнижче середнє використання CPU – 30%.

Отже, можна зробити висновок, що в інтерактивному режимі на рівні S0 запити в цілому створюють досить високе навантаження на процесор, особливо ресурсоємні запити з групуванням та агрегацією даних.

Проаналізувавши використання процесора запитами, перейдемо до розгляду навантаження на інші ресурси – DTU. Для наочного представлення використання DTU на рівні S0 представлено на рисунку 3.6.

Що стосується використання виділених для сервісу обчислювальних ресурсів (вимірюється в DTU), то тут середнє значення складає 83%. Це високий показник, особливо враховуючи невеликий обсяг ресурсів, які надає рівень S0.

Найбільше навантаження на всі апаратні ресурси сервера створюють запити Q1 і Q9, які показали 100% використання DTU. Високе завантаження ресурсів також демонструють запити Q2, Q4, Q7, Q8 зі значеннями DTU на рівні 84-94%. Помірне навантаження спостерігається у запиті Q5. Найменше навантаження створюють запити Q3 та Q6 з 51% DTU.

Отже, загалом спостерігається пряма залежність між складністю запитів (наявністю групувань, агрегацій, вкладених підзапитів) та їх ресурсоємністю за показником DTU. Для оптимізації рекомендується масштабування ресурсів БД.

Подальший аналіз щодо ефективності запитів доцільно розглянути час їх виконання. На рисунку 3.7 наведено дані щодо тривалості запитів на рівні S0 в інтерактивному режимі.

Найдовше виконувалися запити Q9 (20 хв. 43 сек.), Q1 (07 хв. 28 сек.) та Q7 (04 хв. 31 сек.), що відповідає їх високій ресурсоємності. Натомість найменший час показав простий запит Q3 – 7 с.

Отже, аналіз часу виконання також демонструє залежність виконання запитів від їх складності та обсягу операцій. Спостерігається пряма залежність між складністю запиту та часом його виконання. Для комплексного аналізу всі дані подано на одному графіку (рисунок 3.5)

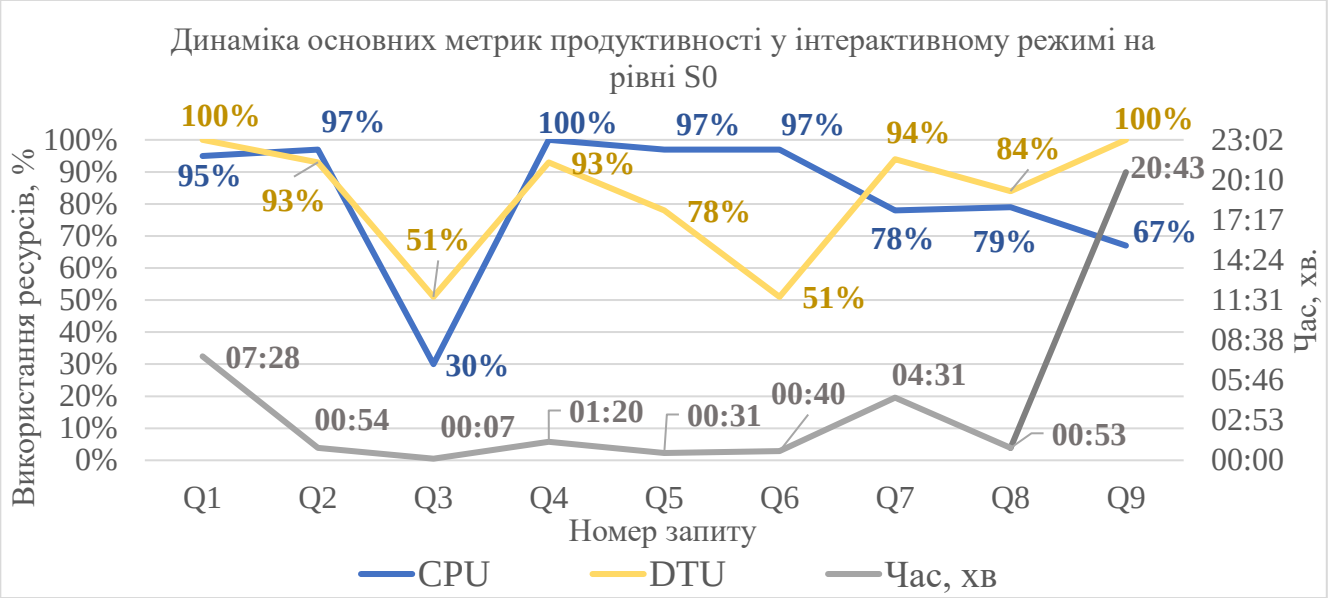

Рисунок 3.5 – Графік динаміки основних метрик продуктивності у інтерактивному режимі на рівні S0 для різних запитів

Як видно з графіку, для запитів Q1, Q2, Q4, Q5, Q7 спостерігаються високі значення використання CPU (95-100%), DTU (93-100%) та тривалий час виконання. Це вказує на їх високу ресурсоємність та низьку ефективність обробки на даному рівні сервісу. Натомість запит Q3 демонструє низькі показники за всіма метриками, що пояснюється його відносною простотою.

Отже, загальний аналіз підтверджує висновок про необхідність оптимізації виконання складних запитів в інтерактивному режимі на рівні S0 шляхом нарощування потужності ресурсів.

У результаті аналізу ключових метрик продуктивності обробки аналітичного навантаження можна констатувати, що мінімальний рівень сервісу S0 неспроможний забезпечити ефективну роботу із запитами в інтерактивному режимі.

Про це свідчать такі фактори:

1. Високе середнє навантаження на процесор (82%) та на виділені обчислювальні ресурси (83% DTU).
2. Значний час виконання складних аналітичних запитів.

Тому рівень S0 не здатний швидко та якісно обробляти аналітичні запити в інтерактивному режимі. Це призводить до значних затримок та гальмування аналітичних робочих процесів.

Отже, для якісної роботи необхідне масштабування середовища Azure SQL Database до рівня з більш потужними обчислювальними можливостями. Це дозволить підвищити ефективність та продуктивність обробки аналітичних запитів в інтерактивному режимі.

Для покращення цих показників доцільно розглянути наступний рівень сервісу – S1, який має вищу пропускну здатність та більший обсяг ресурсів.

Проаналізуємо ефективність запитів на рівні S1 за ключовими метриками використання ресурсів і часу виконання, щоб зрозуміти, чи дає S1 істотне покращення продуктивності порівняно з S0. Результати виконання представленні у таблиці 3.5.

Таблиця 3.5 – Результати виконання запитів у інтерактивному режимі на рівні S1

Запит	CPU	DTU	Час, хв.
Q1	85%	85%	01:48
Q2	42%	46%	00:29
Q3	6%	29%	00:07
Q4	43%	43%	00:52
Q5	29%	29%	00:18
Q6	37%	37%	00:22
Q7	80%	80%	01:16
Q8	46%	46%	00:27
Q9	82%	100%	08:48

Як видно з таблиці 3.5, на рівні обслуговування S1 спостерігається загалом покращення показників ефективності виконання запитів порівняно з мінімальним рівнем S0.

Відбулось істотне зниження навантаження на процесор для більшості запитів: наприклад, для Q2 воно зменшилось з 97% до 42%.

Запити Q3, Q5 та Q6 продемонстрували найбільш істотне покращення - їх навантаження на CPU знизилось у 3-5 разів (300-500%). Це свідчить про ефективність застосування додаткових ресурсів для обробки саме цих запитів.

Водночас складний запит Q7 та Q9 і на рівні S1 показує досить високе завантаження процесора на рівні 80% та 82%. Це значить, що дані запити вимагають виділення більшого обсягу обчислювальних ресурсів для ефективної обробки.

Отже, загалом є суттєве покращення продуктивності по CPU на рівні S1, але для найбільш складних запитів може знадобитися подальше масштабування ресурсів.

Для комплексної оцінки продуктивності варто проаналізувати також показник використання інших апаратних ресурсів – DTU.

Також видно загальну тенденцію до зниження навантаження на ресурси, аналогічно до CPU: наприклад, для запиту Q2 використання DTU зменшилось вдвічі – з 93% на S0 до 46% на S1. Винятком знову є запити Q7 та Q9, які стабільно показують високе значення DTU на рівні 80% та 100%.

Отже, аналіз DTU також демонструє позитивний ефект від збільшення ресурсів на рівні S1 для більшості запитів.

Варто також проаналізувати, як змінився час виконання запитів, що є ключовим показником продуктивності.

Для всіх запитів відбулося істотне прискорення обробки даних. Особливо значне скорочення часу спостерігається для найбільш ресурсоємних запитів Q1 та Q9 – у 7 та у 2,5 разів відповідно в порівнянні з рівнем S0. Це пояснюється тим, що додаткові ресурси на рівні S1 дозволяють швидше виконувати складні обчислення і операції з даними. В результаті навіть запити з високим навантаженням на CPU та DTU обробляються за менший час.

Отже, аналіз часу виконання запитів підтверджує значне підвищення загальної продуктивності та ефективності обробки аналітичного навантаження на рівні S1 порівняно з базовим S0. Це досягається завдяки наявності більших обчислювальних ресурсів.

Для узагальненої оцінки ефективності обробки запитів на рівні S1 доцільно представити графік з усіма ключовими метриками продуктивності. Графік представлений на рисунку 3.6.

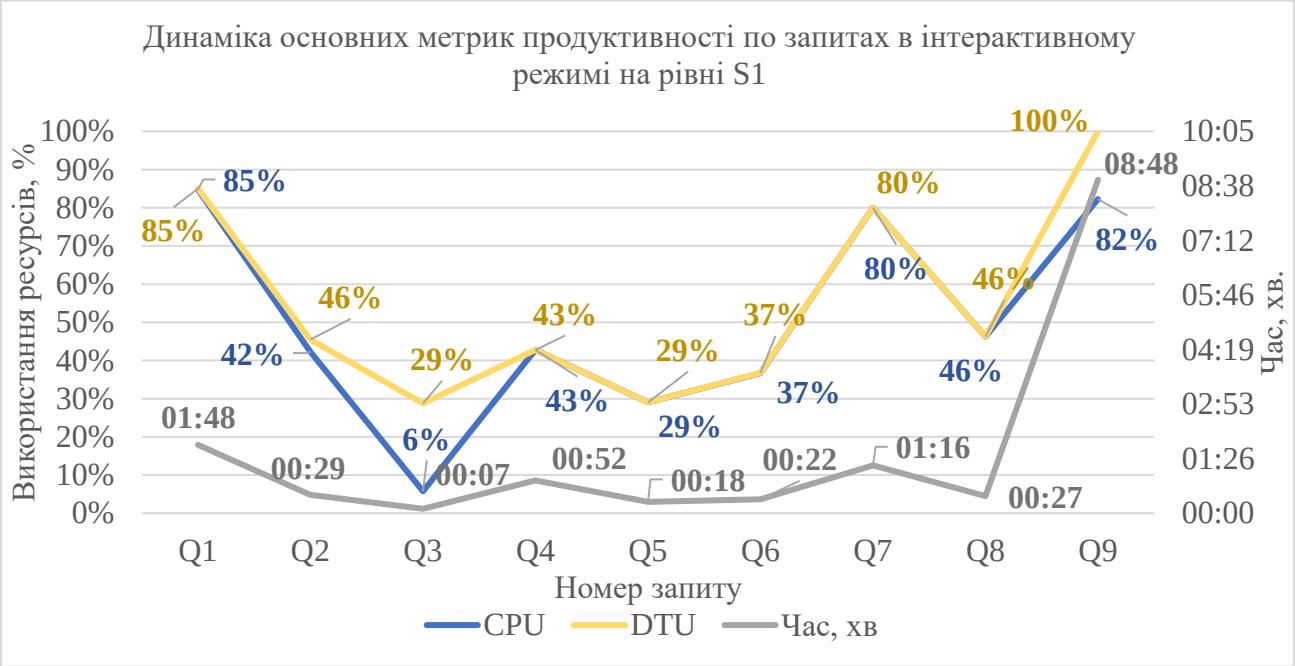

Рисунок 3.6 – Графік динаміки основних метрик продуктивності по запитах в інтерактивному режимі на рівні S1

Аналізуючи цей графік, можна зробити висновок, що перехід до рівня S1 загалом забезпечує підвищення продуктивності обробки аналітичного навантаження за рахунок:

– зменшення середнього навантаження на CPU та DTU;
– зменшення часу виконання запитів, – особливо складних запитів;
– можливості швидше обробляти великі обсяги даних.

Разом з тим, окремі ресурсоємні запити, все ще потребують додаткових ресурсів для ефективнішої обробки.

Схожість показників використання CPU та DTU для більшості запитів на рівні S1 може бути пов'язана з більшим обсягом виділених ресурсів у порівнянні з рівнем S0.

Зокрема, на рівні S1 сервіс Azure SQL Database надає в 2 рази більше обчислювальних ресурсів, ніж на S0. Це включає як збільшення кількості віртуальних ядер процесора, так і розміру ОЗП, пропускної здатності та інших складових.

Таким чином, на рівні S1 запити отримують більші обсяги всіх необхідних ресурсів одночасно – і обчислювальних потужностей CPU, і пам'яті, і пропускної здатності.

Саме тому використання CPU та DTU цими запитами зазвичай є подібним – воно відображає виділений їм у рівних пропорціях збільшений блок ресурсів згідно архітектури та можливостей рівня S1.

Таким чином, збільшення обсягу всіх ресурсів на рівні S1 може призводити до більшої схожості показників використання CPU та DTU порівняно з S0.

Отже, для подальшого підвищення продуктивності варто розглянути наступні рівні сервісу Azure SQL Database, починаючи з S3.

У таблиці 3.6 наведено результати запитів на рівні S3 за ключовими метриками використання CPU, DTU та часу виконання:

Таблиця 3.6 – Результати виконання запитів у інтерактивному режимі на рівні S3

Запит	CPU	DTU	Час, хв.
Q1	45%	72%	01:01
Q2	13%	32%	00:08
Q3	5%	5%	00:02
Q4	37%	37%	00:27
Q5	20%	20%	00:06
Q6	15%	15%	00:03
Q7	39%	39%	00:12
Q8	26%	26%	00:05
Q9	58%	58%	00:58

Аналізуючи ці дані, можна помітити подальше покращення показників порівняно з попереднім рівнем S1. Зокрема, для всіх запитів відбулося зниження середнього використання CPU та DTU.

Можна побачити значне зниження навантаження на процесор для всіх запитів в порівнянні з попередніми рівнями S1 та S0.

Наприклад, для найбільш ресурсоємного запиту Q1 використання CPU зменшилось з 85% на S1 до 45% на рівні S3. Аналогічні тенденції спостерігаються для решти запитів.

Це свідчить про те, що додаткові обчислювальні ресурси на рівні S3 дозволяють ефективніше розподіляти навантаження між віртуальними ядрами процесора. Тобто кожен запит використовує меншу частку CPU для своєї обробки.

Зокрема, найменше навантаження за цим показником створюють нескладні запити Q3 та Q6 (5% та 15% CPU відповідно).

Розглянемо показник завантаження інших апаратних ресурсів – DTU. Можна побачити аналогічну тенденцію зниження навантаження на ресурси порівняно з S1 та S0. Наприклад, для запиту Q1 використання DTU зменшилось з 85% на рівні S1 до 72% на S3. Це свідчить про більш оптимальний розподіл навантаження між всіма апаратними компонентами завдяки підвищеним можливостям рівня S3.

Отже, обидва показники CPU та DTU демонструють значне покращення ефективності використання ресурсів на рівні S3 для обробки аналітичного запиту.

Також видно значне прискорення виконання всіх запитів порівняно з попередніми рівнями. Наприклад, час для запиту Q1 зменшився з 1 хв. 48 с. на S1 до 1 хв. 1 с. на рівні S3.

Візуалізація усіх метрик інтерактивного режиму на рівні S3 представлено на рисунку 3.7.

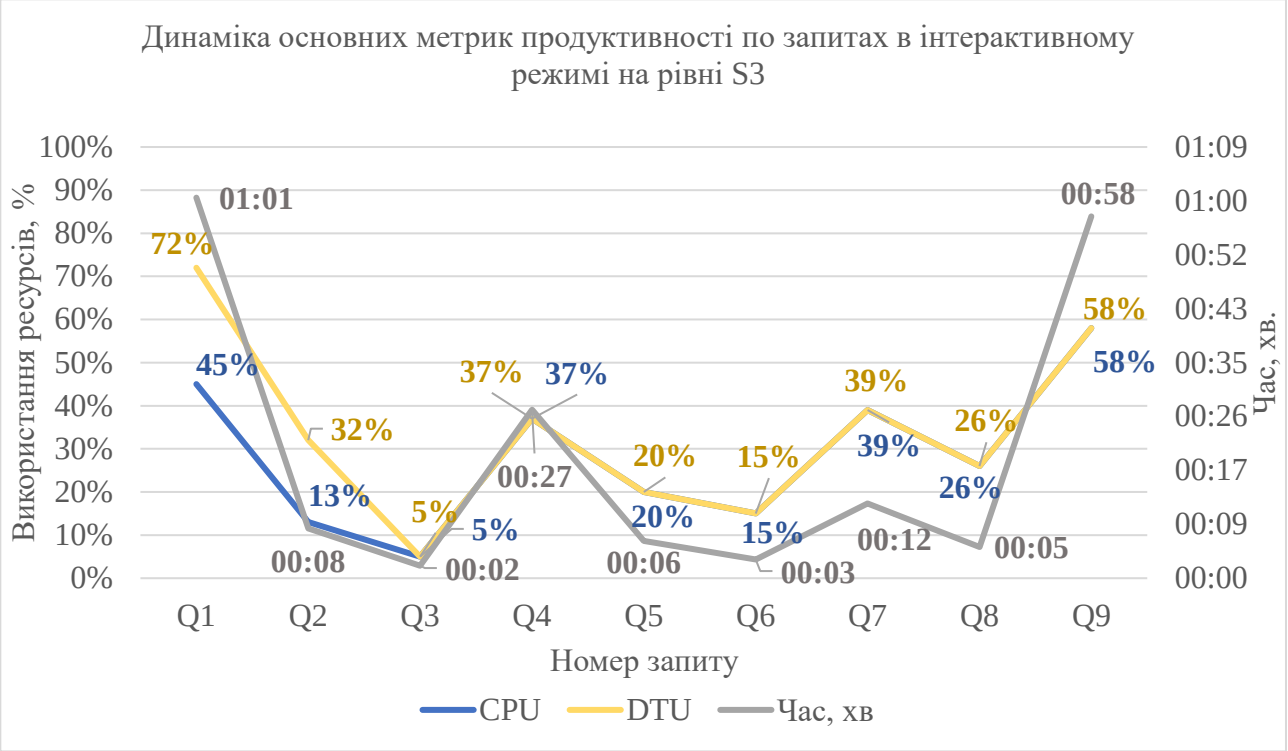

Рисунок 3.7 – Графік динаміки основних метрик продуктивності по запитах в інтерактивному режимі на рівні S3

Аналіз усіх ключових метрик демонструє значне підвищення загальної продуктивності обробки аналітичного навантаження на рівні S3 порівняно з попередніми рівнями.

На рівні S3 бачимо, що для більшості запитів використання CPU та DTU є однаковим або майже однаковим. Це може бути результатом ефективного використання ресурсів в обчислювальному середовищі S3.

Перехід на наступний рівень – S7. Він дозволяє ще більше підвищити продуктивність за рахунок використання потужніших обчислювальних ресурсів. Рівень S7 надає в 8 разів більше граничне значення DTU збільшується зі 100 одиниць до 800, у порівнянні з рівнем S3. Це дозволяє ефективніше масштабувати обробку запитів, особливо для складних аналітичних завдань. Загалом, перехід на S7 суттєво розширює можливості масштабування обчислень і дозволяє досягти ще вищої продуктивності обробки даних в хмарі. Результати виконання запитів в інтерактивному режимі представлені у таблиці 3.7.

Таблиця 3.7 – Результати виконання запитів у інтерактивному режимі на рівні S7

Запит	CPU, %	DTU, %	Час, хв.
Q1	11	11	00:14,0
Q2	2	4	00:01,0
Q3	1	1	00:01,0
Q4	6	6	00:18,0
Q5	3	3	00:03,0
Q6	3	3	00:00,5
Q7	7	7	00:04,0
Q8	8	5	00:00,7
Q9	18	18	00:28,0

Аналіз результатів виконання запитів в інтерактивному режимі на рівні S7 показує подальше підвищення ефективності використання ресурсів порівняно з попереднім рівнем S3. Проаналізуємо використання CPU. Аналіз використання CPU для кожного запиту дозволяє краще оцінити розподіл навантаження на систему.

Отже, найбільше завантаження процесора відбувається для запитів Q1 та Q9, використання ресурсів для яких сягає 11% та 18% відповідно. Інші запити використовують CPU в діапазоні 1-8%.

Такий розподіл навантаження свідчить про ефективне використання обчислювальних ресурсів та їх масштабування відповідно до потреб конкретних запитів. Загалом рівень завантаження CPU є оптимальним для забезпечення високої продуктивності обробки аналітичних запитів в інтерактивному режимі.

Перейдемо до аналізу використання DTU. Видно, що найбільше DTU витрачається на виконання складних запитів Q1 та Q9 – 11% та 18% відповідно. Для інших запитів споживання DTU коливається в діапазоні 1-7%.

Найбільше навантаження на DTU створюють запити Q1 та Q9, що потребують відповідно 11% та 18% цих ресурсів. Це свідчить про те, що саме ці запити є найбільш ресурсоємними.

Для інших запитів використання DTU коливається в діапазоні 1-7%. Тобто ці запити менш складні і вимагають менше ресурсів для їх обробки.

Загалом, рівень використання DTU є оптимальним і дозволяє ефективно виконувати інтерактивну аналітичну обробку великих обсягів даних.

Аналіз часу виконання запитів в інтерактивному режимі на рівні S7 демонструє дуже високу швидкодію обробки. Найбільше часу займає обробка складних запитів Q1, Q4 та Q9 – 14, 18 і 28 сек. відповідно. Для решти запитів час виконання становить від 1 сек. до 7 сек.

Така швидкість обробки є високою і прийнятною для інтерактивного режиму аналітики. Рівень S7 забезпечує більшу швидкодію аналітичних запитів аніж попередні рівні в Azure SQL Database (рисунок 3.8).

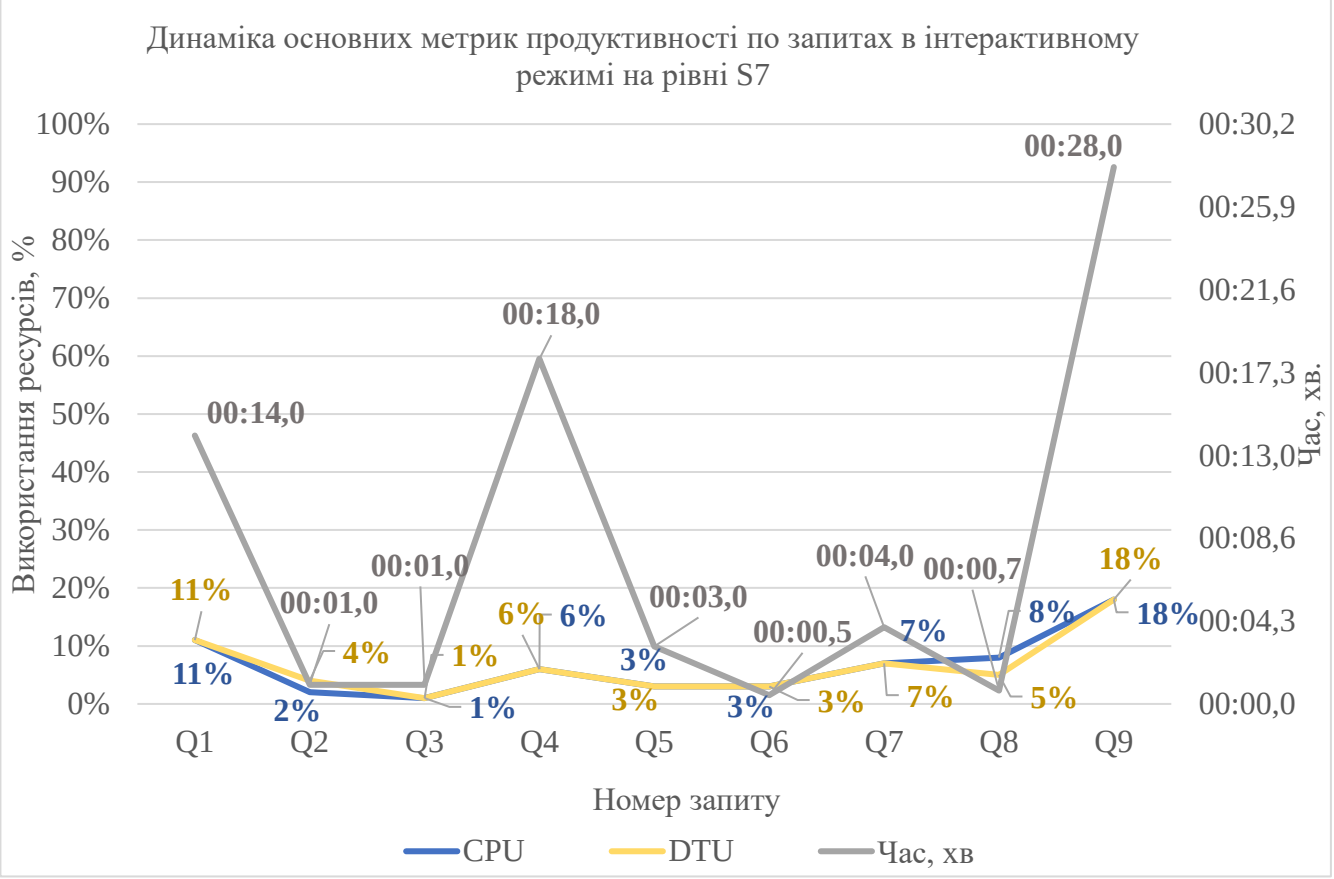

Рисунок 3.8 – Графік динаміки основних метрик продуктивності по запитах в інтерактивному режимі на рівні S7

Отже, найбільш ресурсоємними є запити Q1 та Q9, для яких характерне максимальне завантаження CPU, DTU та часу обробки.

Для інших запитів спостерігається пропорційне використання ресурсів та часу відповідно до складності.

Загалом, рівень S7 демонструє ефективне використання ресурсів та їх оптимальне масштабування для досягнення максимальної продуктивності в інтерактивному режимі.

Для подальшого дослідження підвищення продуктивності треба розглянути перехід на наступний рівень – S12. Він пропонує ще більші обчислювальні ресурси – максимум 3 000 DTU. Результати інтерактивного режиму для рівня S12 представлено у таблиці 3.8.

Таблиця 3.8 – Результати виконання запитів у інтерактивному режимі на рівні S12

Запит	CPU,%	DTU,%	Час, хв.
Q1	4	1	00:13,0
Q2	1	2	00:02,0
Q3	1	1	00:01,0
Q4	2	2	00:14,0
Q5	1	1	00:02,0
Q6	1	1	00:00,4
Q7	3	3	00:03,0
Q8	2	2	00:00,6
Q9	8	8	00:21,0

Аналіз результатів виконання запитів в інтерактивному режимі на рівні S12 показує подальше підвищення ефективності використання ресурсів у порівнянні з рівнем S7.

Спостерігається зниження навантаження на CPU порівняно з попереднім рівнем S7, що свідчить про більш ефективне використання ресурсів. Найбільше CPU витрачає запит Q9 – 8%, що в 2 рази менше ніж на S7. Це демонструє кращу оптимізацію складних обчислень. Для решти запитів використання CPU становить 1-3%, що є досить низьким показником. Така картина свідчить про те, що рівень S12 має значний резерв обчислювальних ресурсів для подальшого масштабування за потреби. Отже, рівень S12 забезпечує ефективне використання CPU.

Також видно, що, використання DTU зменшилося в 2-3 рази (200-300%) порівняно з S7 завдяки більшій кількості доступних одиниць продуктивності на рівні

S12. Найбільш ресурсоємним по DTU залишається запит Q9, проте його частка знизилася з 18% до 8% завдяки кращій оптимізації. Для простіших запитів достатньо 1-3% DTU, тобто спостерігається значна економія ресурсів. Загалом рівень використання DTU демонструє здатність S12 масштабуватися для складних обчислювальних задач аналітики при збереженні високої продуктивності.

Час виконання найскладніших запитів Q1 та Q9 трохи зменшився порівняно з S7 завдяки більш потужним ресурсам S12. Для простіших запитів час виконання залишився практично без змін та становить 1-3 секунди. Загальна швидкість обробки усіх запитів є надзвичайно високою для інтерактивного режиму аналітики. Незначне прискорення виконання складних запитів свідчить про те, що ресурсів S12 достатньо навіть для найвимогливіших аналітичних завдань. Візуально усі метрики подано на рисунку 3.9.

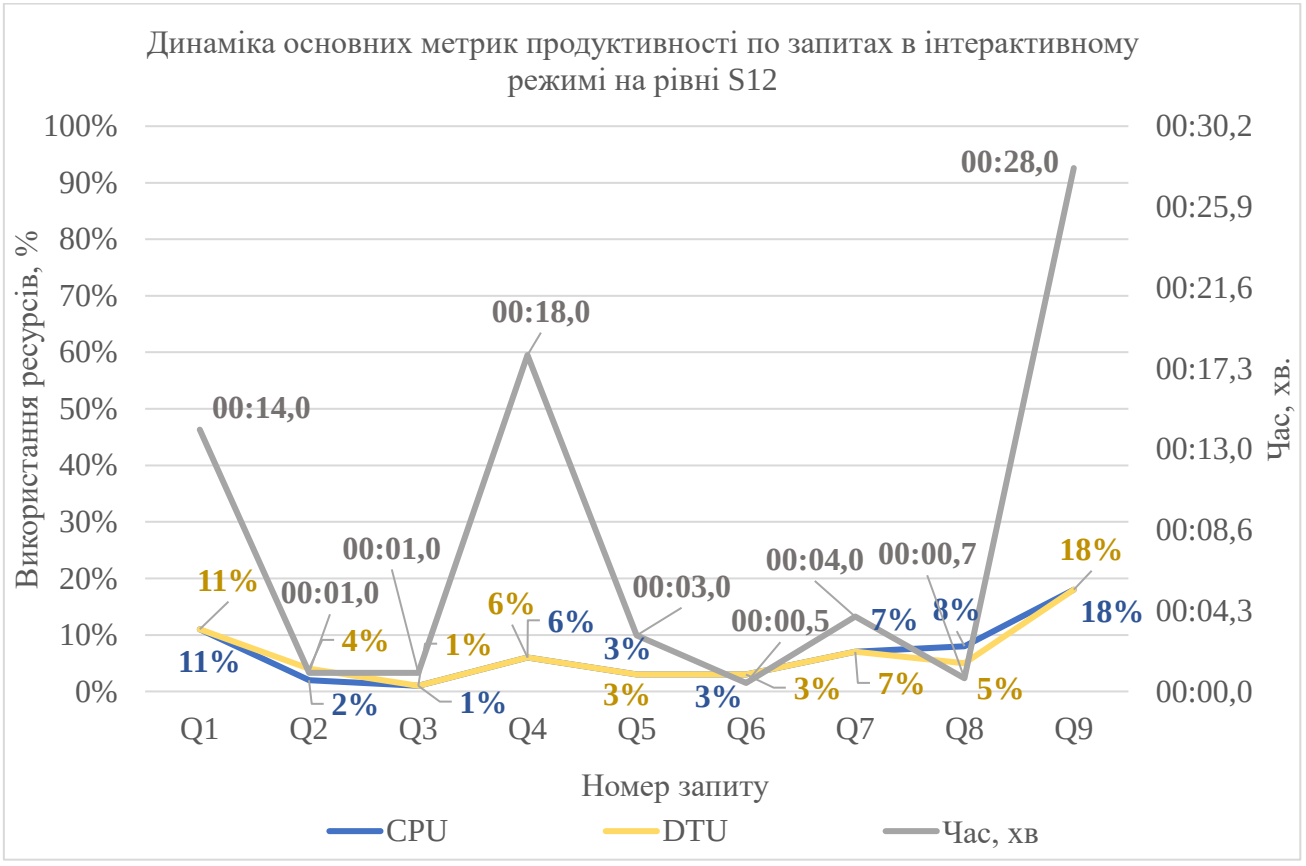

Рисунок 3.9 – Графік динаміки основних метрик продуктивності по запитах в інтерактивному режимі на рівні S12

Використання CPU та DTU для всіх запитів знизилось в 2-3 рази порівняно з рівнем S7. Це свідчить про більшу кількість виділених ресурсів. Найбільш

ресурсоємним лишається запит Q9, але його показники CPU та DTU знизилися вдвічі. Час виконання складних запитів Q1 та Q9 трохи зменшився завдяки наявності потужніших ресурсів.

Отже, рівень S12 дозволяє досягти ще більш ефективного використання ресурсів та незначно підвищити швидкодію обробки складних запитів. Це робить його хорошим вибором у випадку необхідності обробки ресурсоємних запитів та максимальної продуктивності інтерактивної аналітичної обробки масивів даних.

Для наочного порівняння ефективності різних рівнів обслуговування в інтерактивному режимі доцільно побудувати узагальнюючі графіки ключових метрик продуктивності.

Почнемо з представлення графіку використання CPU у інтерактивному режимі на рівнях моделі (рисунок 3.10).

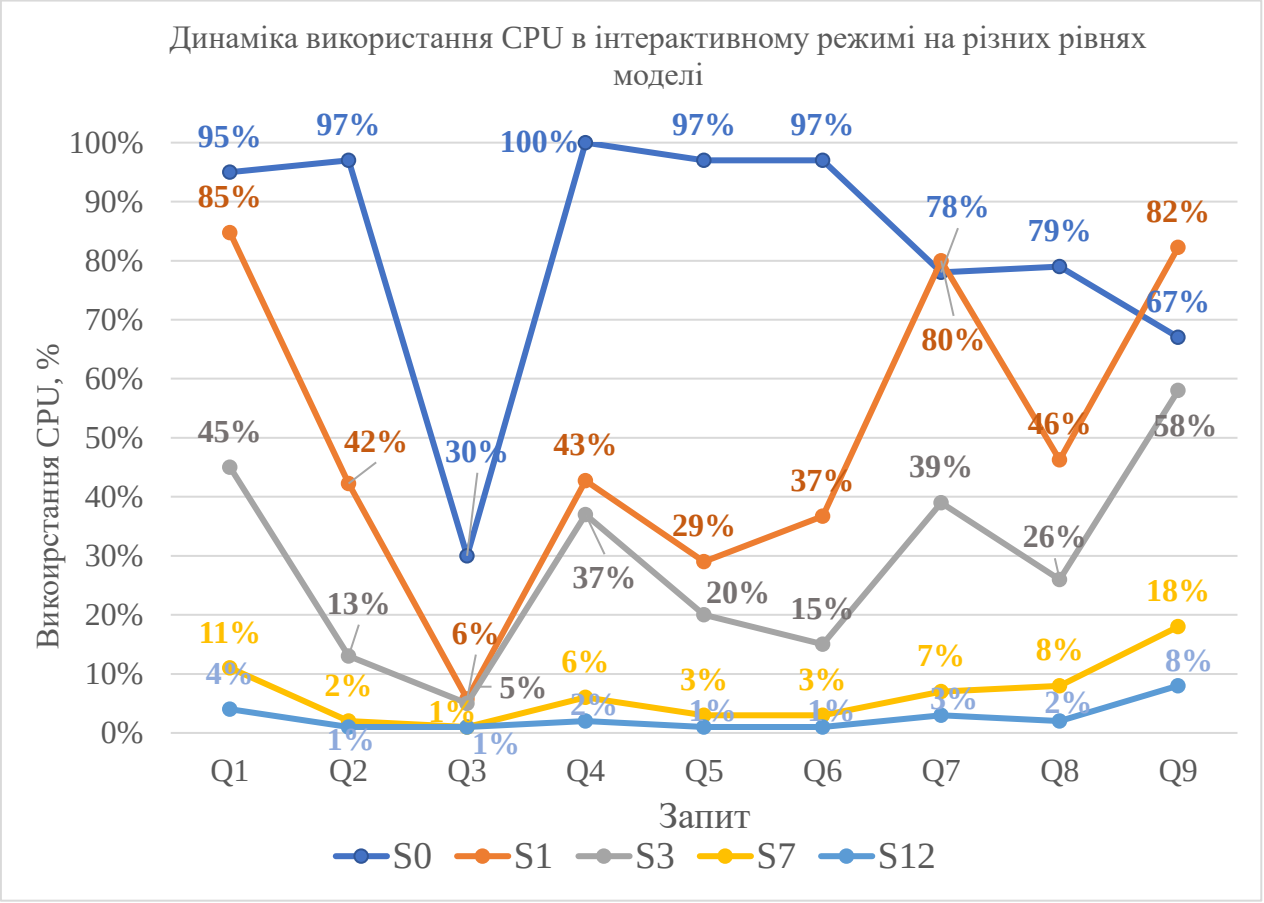

Рисунок 3.10 – Графік динаміки використання CPU в інтерактивному режимі на різних рівнях моделі

Аналізуючи графік використання CPU для запитів на різних рівнях обслуговування в інтерактивному режимі, можна зробити висновок, що, на початкових рівнях S0 і S1 спостерігається критично високе завантаження CPU, що негативно впливає на продуктивність обробки запитів. Це свідчить про нестачу ресурсів для якісної обробки. Перехід на S1 трохи знижує навантаження, але критичні запити все ще використовують понад 80% CPU. Починаючи з рівня S3 навантаження на CPU поступово зменшується завдяки наявності більшого обсягу обчислювальних ресурсів. Оптимальна картина спостерігається на рівнях S7 та S12.

Таким чином, вищі рівні обслуговування забезпечують найбільш ефективне використання CPU для високопродуктивної аналітичної обробки в інтерактивному режимі.

Порівняльний графік використання DTU на різних рівнях моделі сервісу подано на рисунку 3.11.

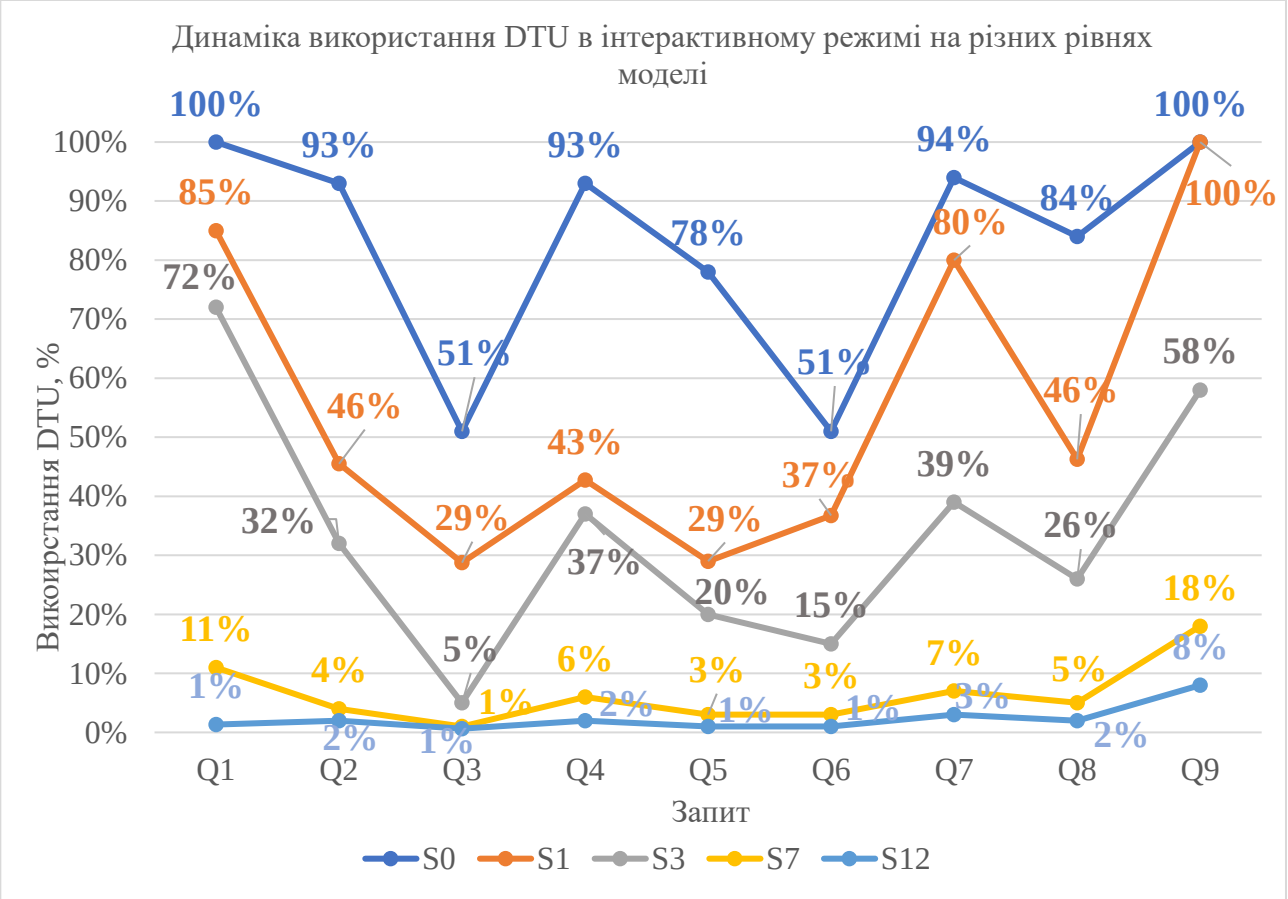

Рисунок 3.11 – Графік динаміки використання DTU в інтерактивному режимі на різних рівнях моделі

На рівні S0 майже для всіх запитів спостерігається критичне використання DTU на рівні 93 – 100%. Це свідчить про гостру нестачу цих ресурсів. Ситуація дещо покращується на S1, але складні запити все ще завантажують 80 – 100% DTU. Помітна оптимізація починається з S3, де використання DTU для найвимогливіших запитів знижується до 58 – 72%. Найбільш ефективно DTU використовуються на рівнях S7 та S12 – всі запити обробляються з завантаженням 1 – 18% цих ресурсів.

Отже, аналіз DTU також демонструє покращення ефективності використання ресурсів на вищих рівнях обслуговування. Зокрема S7 та S12 забезпечує найоптимальніше співвідношення продуктивності та витрат DTU.

Проаналізуємо час виконання запитів на різних рівнях обслуговування в інтерактивному режимі. Графік часу виконання представлено на рисунку 3.12.

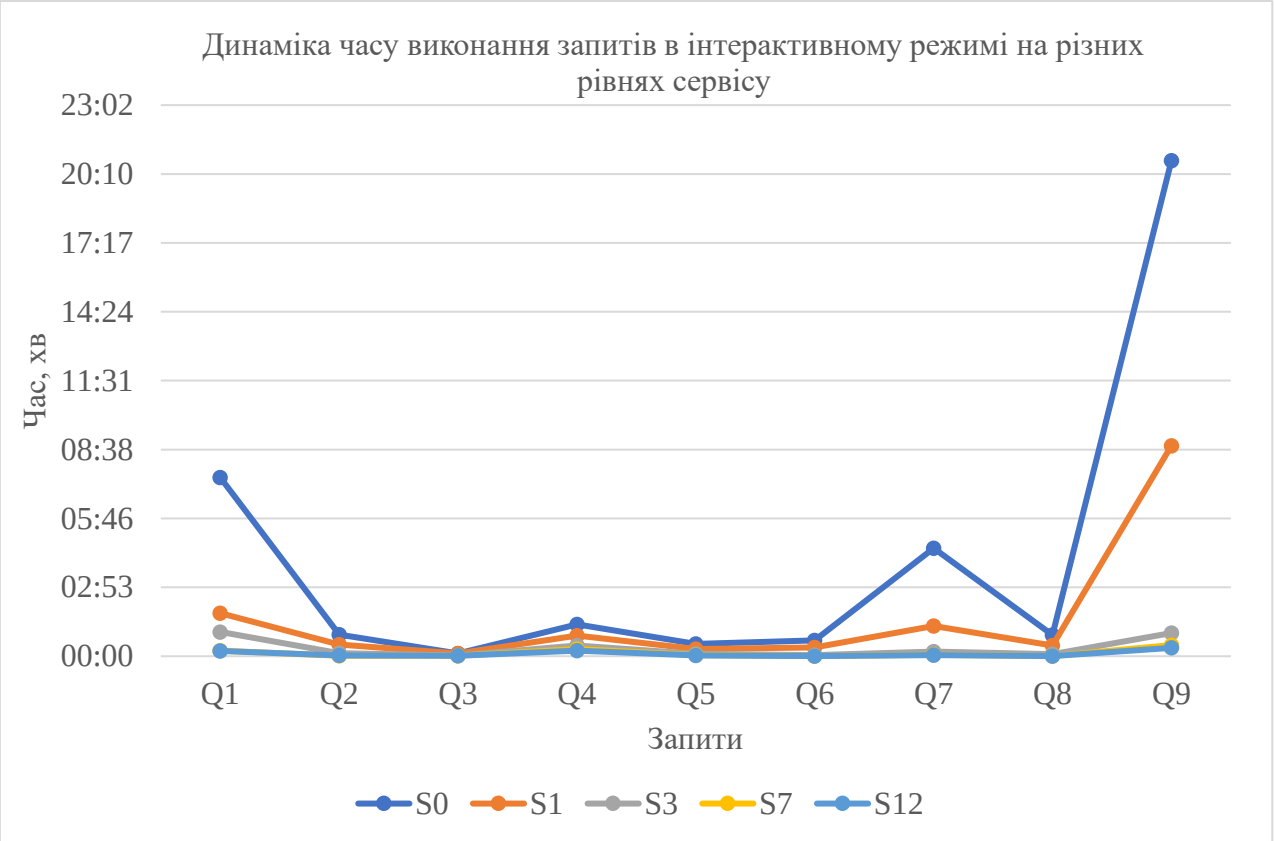

Рисунок 3.12 – Графік динаміки часу виконання запитів в інтерактивному режимі на різних рівнях моделі

Можна побачити, що на рівні S0 час виконання найскладніших запитів Q1 і Q9 перевищує 20 хв., що є неприпустимо довго для інтерактивної обробки. Ситуація покращується на S1, але критичні запити все одно обробляються досить довго

– понад 8 хвилин. Прискорення обробки відбувається на рівнях S3 і вище – час виконання складних запитів зменшується до 1 хв. Найшвидше запити обробляються на S7 та S12 – 13-21 сек. – для найвимогливіших запитів Q1 і Q9.

Для більш детального аналізу проаналізуємо сумарний час виконання запитів на рівнях моделі. Графік сумарного часу виконання представлено на рисунку 3.13.

Рисунок 3.13 – Графік сумарного часу виконання запитів в інтерактивному режимі на різних рівнях моделі

На рівнях S0 і S1 спостерігається дуже тривалий загальний час обробки запитів – понад 14-37 хв. відповідно. Істотне прискорення відбувається на рівні S3 – сумарний час зменшується до 3 хв. На рівнях S7 і S12 час становить 1-1.5 хв. При цьому суттєвої різниці між цими рівнями немає.

Отже, найбільш значне покращення швидкодії обробки спостерігається при переході з S0 на S3. Подальше підвищення рівня призводить до несуттєвого прискорення.

Для визначення оптимального рівня з точки зору співвідношення швидкості та вартості треба також проаналізувати співвідношення вартості обробки запитів на різних рівнях (рисунок 3.14).

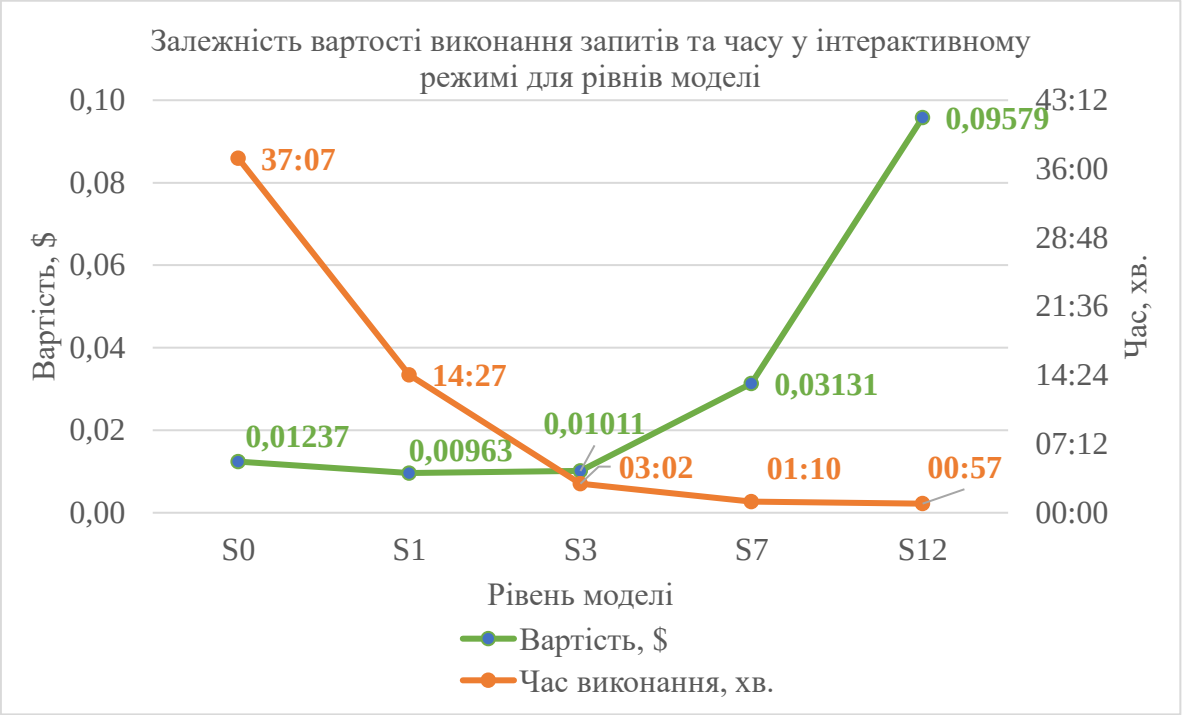

Рисунок 3.14 – Графік залежності вартості та часу виконання запитів для різних рівнів моделі в інтерактивному режимі

Рівень S0 має найнижчу швидкість виконання та високу вартість відносно до рівня S1, який виконується швидше майже у 2 рази (200%).

Рівень S1 пропонує найнижчу вартість обробки запитів – 0,0096 долара. Однак час виконання на цьому рівні є надзвичайно великим – понад 14 хв.

Перехід на рівень S3 дозволяє істотно прискорити обробку до 3 хвилин при незначному збільшенні вартості до 0,01 долара. Цей час цілком прийнятний для інтерактивної аналітики і дозволяє оперативно отримувати результати аналізу даних.

Подальше підвищення рівня до S7 і S12 призводить до стрибка вартості (0,03-0,09 доларів) при прискоренні обробки до 1-1,5 хв. Таке співвідношення є економічно затратним.

Отже, для більшості задач інтерактивного режиму оптимальним є рівень S3, який забезпечує гарне поєднання швидкості та вартості. Якщо потрібна максимально висока швидкість обробки, можна розглянути рівень S7, але при цьому вартість значно зросте.

Для підбиття підсумків аналізу інтерактивного режиму сформуємо такі рекомендації:

– рівень S3 є оптимальним вибором для більшості задач інтерактивної аналітики, оскільки забезпечує найкраще співвідношення вартості та швидкості обробки запитів;

– якщо потрібна максимально висока швидкодія, можна розглянути використання рівнів S7 або S12, але слід враховувати значне зростання вартості;

– для складних аналітичних завдань доцільно використовувати максимально можливий обсяг ресурсів CPU та DTU на обраному рівні;

– рекомендується періодично аналізувати завантаження системи та за потреби коригувати рівень обслуговування для оптимізації використання ресурсів.

Отже, ретельний вибір та аналіз рівня обслуговування є запорукою ефективної та економічно вигідної інтерактивної аналітичної обробки даних на платформі Azure SQL Database.

Далі розглянемо особливості пакетного режиму обробки.

3.5.2 Аналіз результатів застосування пакетних режимів

На відміну від інтерактивного режиму, у пакетному режимі запити виконуються групами згідно певного алгоритму. Метою аналізу пакетного режиму є визначення оптимальної стратегії формування та виконання пакетів SQL-запитів для досягнення максимальної продуктивності на різних рівнях сервісу Azure SQL Database.

Аналіз пакетного режиму є важливим, оскільки він дозволяє оптимізувати використання обчислювальних ресурсів при масовій обробці даних. На відміну від інтерактивного режиму, де важлива швидкість обробки кожного окремого запиту, в пакетному режимі критичним є загальна продуктивність виконання всієї черги запитів.

У підрозділі 3.4.2 було детально розглянуто можливі підходи до формування черги запитів для пакетного режиму обробки в Azure SQL Database. Тепер настав час практично протестувати ці плани та проаналізувати їх ефективність на різних рівнях обслуговування.

Кожна стратегія буде протестована на 5 рівнях сервісу: S0, S1, S3, S7 та S12 для комплексної оцінки можливостей масштабування продуктивності пакетної обробки запитів у хмарі.

Для початку розглянемо результати найпростішої стратегії – формування пакету у довільному порядку, і в подальшому, для зручності, будемо використовувати позначення B1 (Batch 1). Мета – оцінити базову продуктивність пакетного режиму та її динаміку при масштабуванні ресурсів від S0 до S12.

Для наочного відображення отриманих результатів було побудовано графік використання основних метрик продуктивності від рівня сервісу (рисунок 3.15). На графіку представлено відсоток використання CPU, відсоток використання DTU та час виконання пакету запитів (в хвилинах) для кожного з досліджуваних рівнів моделі сервісу – від S0 до S12.

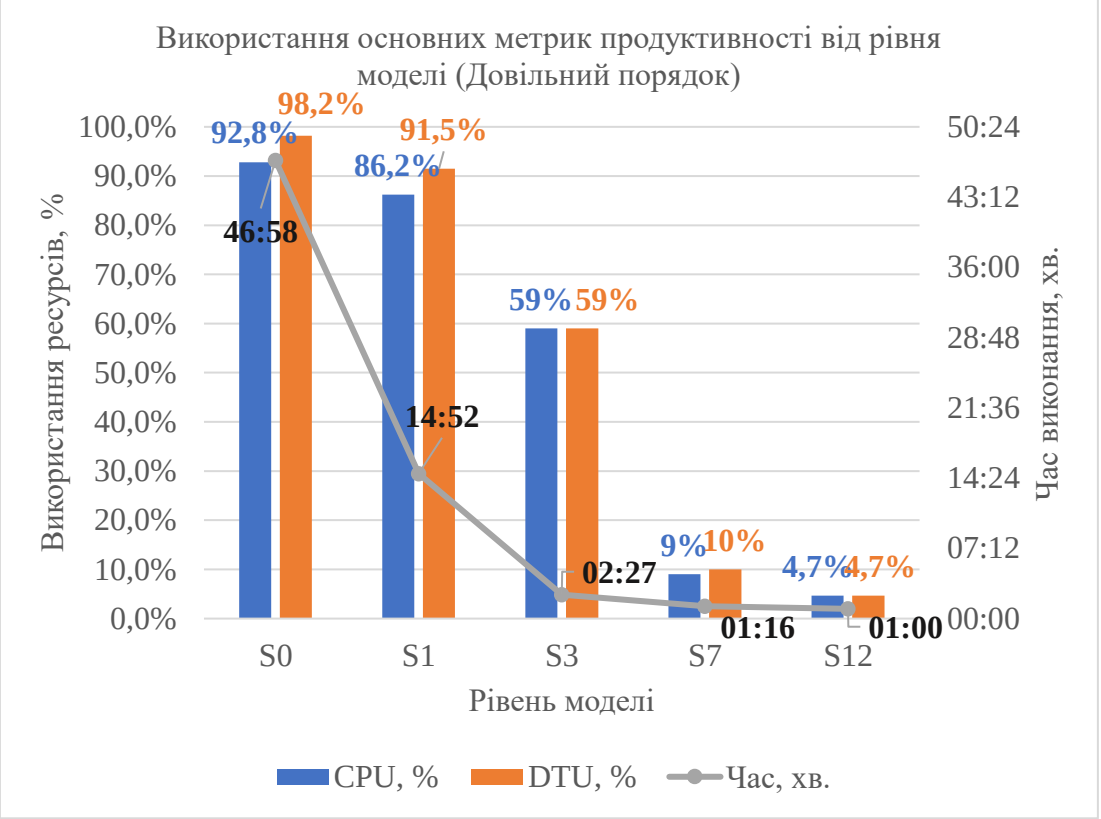

Рисунок 3.15 – Графік основних метрик продуктивності від рівня моделі сервісу (Довільний порядок)

Як видно з графіку, при збільшенні ресурсів спостерігається помітне зниження відсотку використання CPU та DTU, а також скорочення часу виконання пакету запитів.

Найвище навантаження за використанням CPU та DTU спостерігається на рівні S0 – 92,8% та 98,2% відповідно. Це пояснюється тим, що S0 - це базовий рівень з найменшими обчислювальними ресурсами. Виконання пакетного режиму зайняло майже 47 хв.

На рівні S1 навантаження трохи нижче – 86,2% по CPU та 91,5% по DTU. Час виконання скоротився майже в 3 рази – до 14 хв. 52 сек. Це свідчить, що додаткові ресурси рівня S1 дозволяють ефективніше обробляти запити.

Значне покращення спостерігається на рівні S3 – використання CPU зменшилось до 59%, DTU – до 59%, час виконання – до 2 хв. 27 сек. Тобто ресурсів рівня S3 цілком достатньо для обробки даного навантаження.

На вищих рівнях S7 та S12 спостерігається недовикористання ресурсів, про що свідчать показники використання CPU та DTU на рівні 9-10% та 4-5% відповідно. Проте час виконання продовжує зменшуватися з підвищенням рівня.

Перейдемо до наступного підходу – групування за зростанням часу виконання запитів (B2).

Для цього випадку запити були виконані в інтерактивному режимі для визначення часу їх обробки (див. розділ 3.5.1). Потім вони були впорядковані за зростанням цього показника перед включенням до пакету у пакетному режимі.

Така оптимізація дозволяє спочатку обробити більш прості та швидкі запити, а потім перейти до більш складних. Це може покращити загальну продуктивність пакетної обробки.

На рисунку 3.16 наведено результати пакетного режиму з впорядкуванням запитів за зростанням часу їх виконання на різних рівнях моделі сервісу.

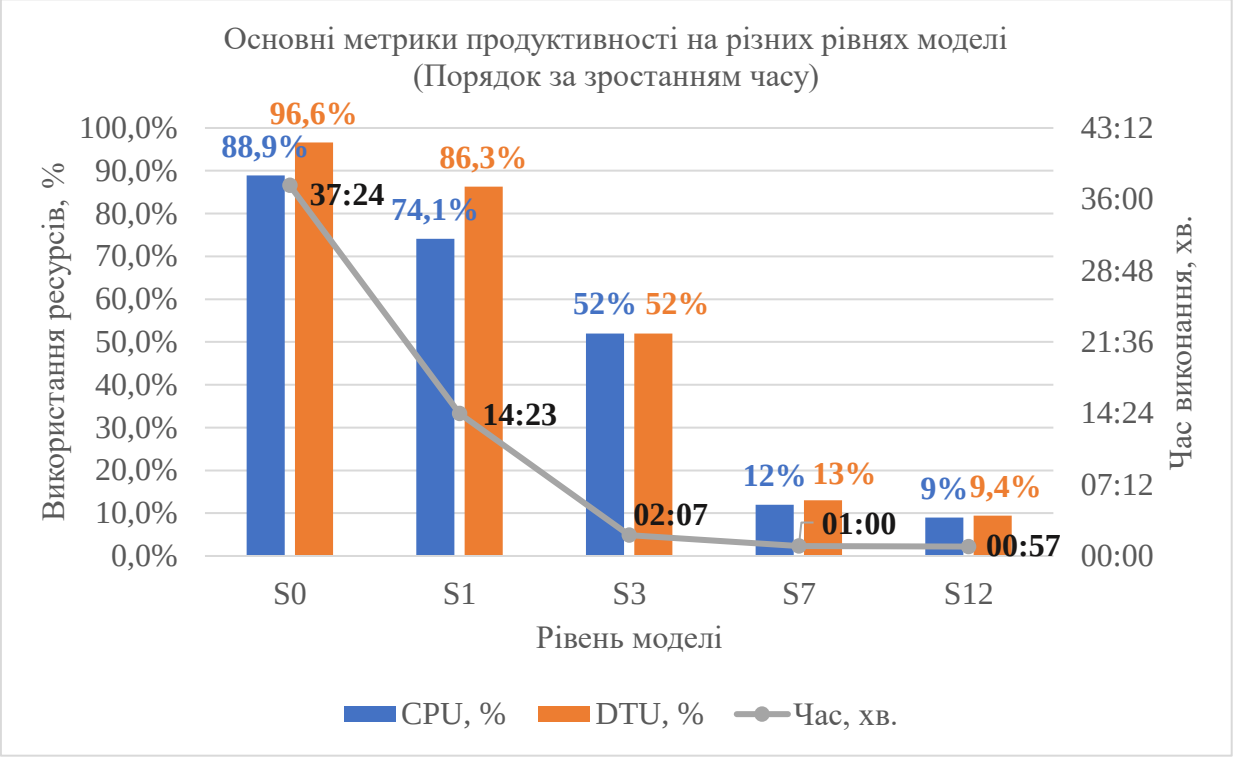

Рисунок 3.16 – Графік основних метрик продуктивності для різних рівнів моделі (Порядок за зростанням часу)

Порівнюючи результати пакетного режиму з групуванням запитів за зростанням часу виконання з попереднім варіантом довільного порядку, можна зробити висновки.

На рівнях S0 та S1 спостерігається покращення показників використання ресурсів (DTU та CPU на 2 і 4 % відповідно нижче на рівні S0, та на 5 і 12 % на S1 нижче в порівнянні з попереднім пакетом) та скорочення загального часу виконання (на приблизно 9 хв. Менше, ніж на рівні S0, та на 49 сек. менше ніж на рівні S1 в порівнянні з попереднім пакетом). Це вказує на позитивний ефект від застосування порядкування у плані виконання запитів.

Починаючи з рівня S3 ефект від порядкування стає менш помітнішим. Ресурси використовуються так само ефективно, а час виконання відрізняється лише на 5-20 сек.

На високих рівнях S7 та S12 результати фактично ідентичні для обох підходів. Це пояснюється надлишком ресурсів для даного навантаження.

Отже, оптимізація порядку запитів у пакеті шляхом їх сортування за зростанням часу виконання має позитивний ефект на продуктивність для рівнів сервісу. Проте вже з середніх рівнів ця оптимізація не сильно впливає на результат.

Для подальшого аналізу розглянемо із зворотнім групуванням запитів – за спаданням часу їх виконання (ВЗ).

Перед формуванням пакету усі запити були відсортовані за спаданням часу виконання – від найбільш тривалих до найбільш швидких. Результати такого підходу представлені на рисунку 3.17.

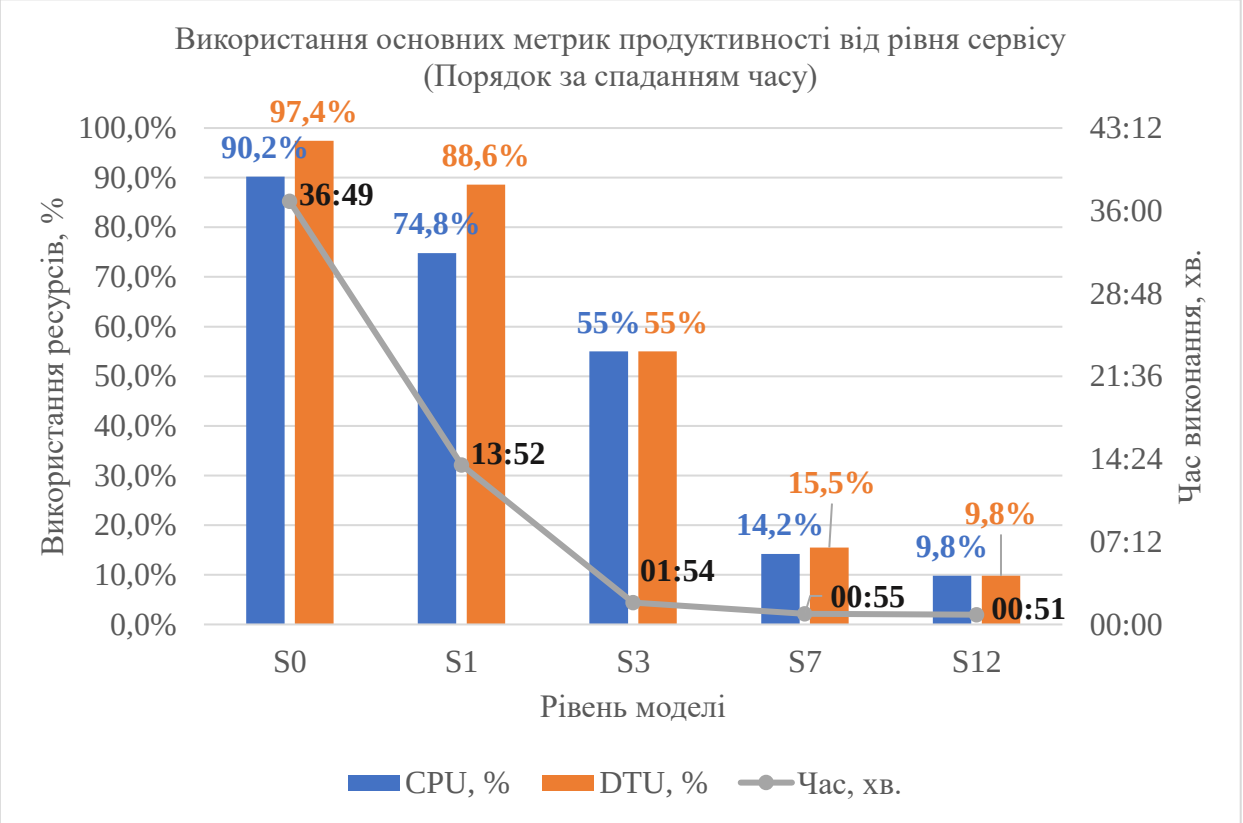

Рисунок 3.17 – Графік основних метрик продуктивності для різних рівнів моделі (Порядок за спаданням часу)

Групування запитів за спаданням часу виконання призводить до невеликого, але все ж покращення загального часу обробки пакету – на кілька секунд або хвилин на різних рівнях моделі.

Це можна пояснити тим, що обробка найбільш складних запитів на початку дозволяє ефективніше розподілити навантаження на ресурси системи і тому подальша обробка простіших запитів відбувається швидше.

Отже, незважаючи на певне пікове завантаження ресурсів при спадному групуванні, загальний час все ж зменшується. Тому, якщо метою є мінімізація часу, а короткочасне високе завантаження є прийнятним, такий підхід може бути виправданим.

Продовжуючи аналіз пакетного режиму, перейдемо до наступного способу групування запитів – за зростанням середньої кількості DTU, які використовувались під час виконання кожного запиту (B4).

При такому групуванні запити в пакеті розташовувалися в порядку зростання середньої кількості DTU, задіяних під час їх обробки: тобто, спочатку йшли менш ресурсоємні, а потім більш вимогливі до системи запити. Результати експерименту представлені на рисунку 3.18.

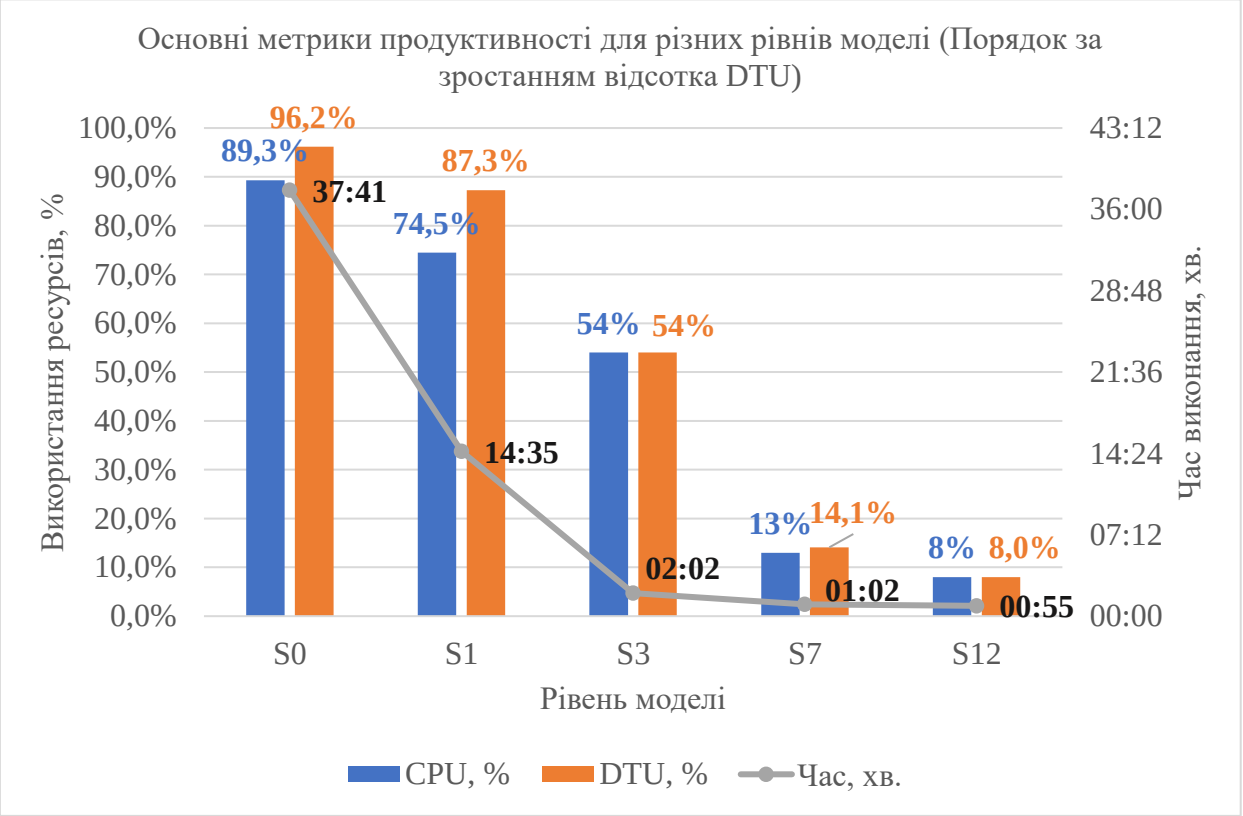

Рисунок 3.18 – Графік основних метрик продуктивності для різних рівнів моделі (Порядок за зростанням відсотка DTU)

Аналізуючи результати пакетного режиму з групуванням запитів за зростанням використання DTU, можна побачити що час виконання на різних рівнях

зменшився в порівнянні з B1 (пакет з довільним порядком групування запитів, який є базовим). Це свідчить про певний позитивний ефект даної оптимізації порядку.

Також можна побачити що в порівнянні з B1, що використання ресурсів як DTU та CPU також знизилось. Це можна пояснити тим, що ресурси у пакетному режимі з групуванням за зростанням DTU краще розподіляються в часі завдяки «плавному» наростанню навантаження – від простих до більш ресурсоємних запитів.

Спочатку задіяні мінімальні ресурси для обробки простих запитів. Потім, по мірі переходу до складніших, поступово підключаються додаткові ресурси. Це дозволяє уникнути пікових навантажень, коли потрібно одразу задіяти максимум ресурсів для обробки складного запиту, і, відповідно, зменшити середнє навантаження.

Таким чином, групування запитів за зростанням ресурсоємності сприяє більш плавному і ефективному розподілу DTU та навантаження CPU у пакетному режимі обробки.

На рівнях S0 та S1 до сих пір спостерігається перенавантаження ресурсів, це призводить до уповільнення швидкодії та збільшення часу відгуку системи, а також неефективного масштабування навантаження в разі збільшення кількості користувачів.

Отже, групування за зростанням використання DTU надає покращення на різних рівнях за рахунок більш плавного зростання навантаження.

Режим B5.

Для подальшого аналізу варто розглянути зворотній підхід – групування за спаданням DTU (B5). При такому групуванні запити в пакеті розташовувалися в порядку зменшення середньої кількості DTU, задіяних під час їх обробки.

Результати використання режиму B5 представлені на рис. 3.19.

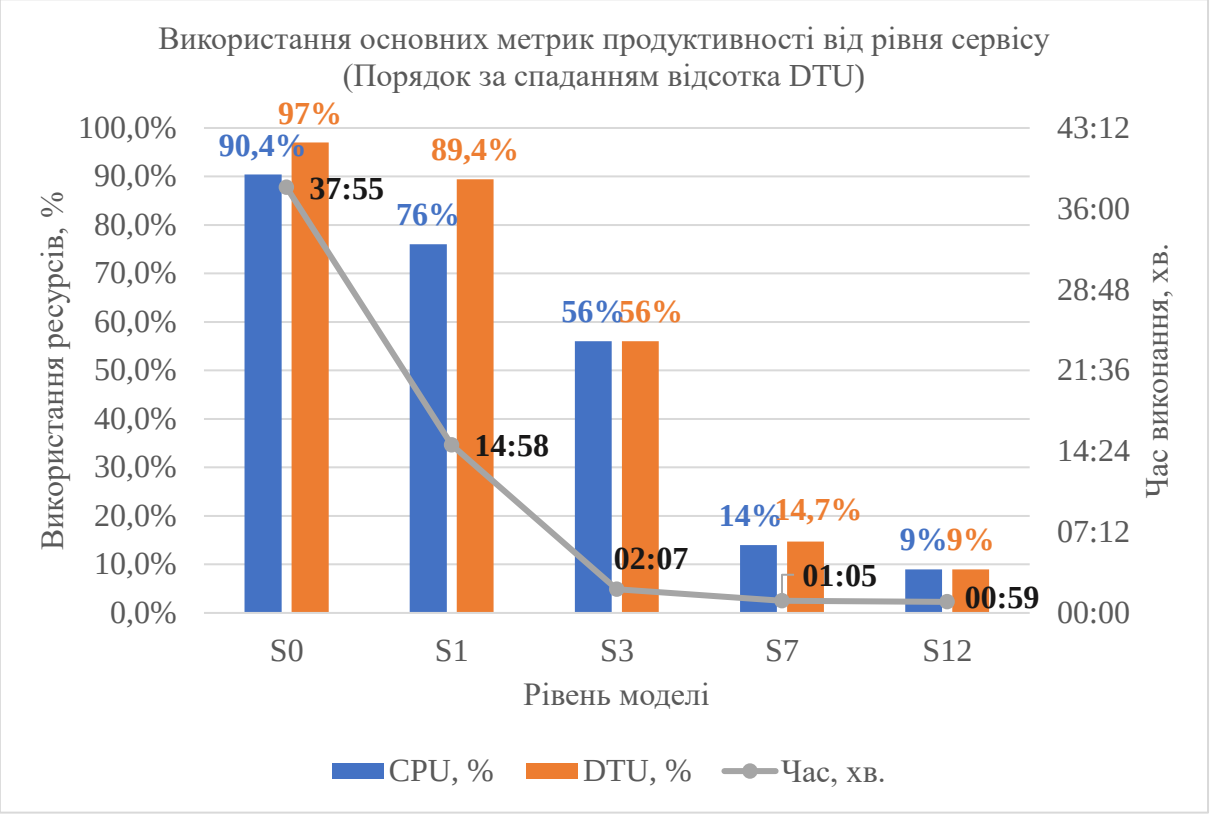

Рисунок 3.19 – Графік основних метрик продуктивності для різних рівнів моделі (Порядок за спаданням відсотка DTU)

Аналізуючи результати плану B5 за спаданням кількості DTU в порівнянні з попереднім планом B4 за зростанням DTU, можна побачити, що час виконання збільшився для усіх рівнів (від 3 сек. до 23 сек. на різних рівнях), це свідчить про те, що планування пакету за спаданням DTU є менш ефективним, ніж за зростанням.

Пояснити це можна тим, що при такому порядку спочатку запускаються найбільш ресурсоємні запити, які можуть перевантажувати систему і уповільнювати обробку наступних запитів у черзі. Натомість планування за зростанням DTU дозволяє спочатку «розігнати» систему на простіших запитах, а потім ефективніше обробляти складніші.

Отже, для оптимізації пакетної обробки краще використовувати планування за зростанням складності запитів, ніж навпаки. Це дає виграш у загальному часі виконання пакету для усіх рівнів моделі.

Для більш глибокого аналізу пакетних режимів обробки запитів також доцільно розглянути групування за показником використання центрального процесора (CPU).

Адже CPU є критичним ресурсом при виконанні обчислювальних та аналітичних операцій в базах даних, тому його завантаженість безпосередньо впливає на продуктивність та швидкодію.

Розглянемо результати пакетів, де запити групувалися спочатку за зростанням (B6), а потім спаданням (B7) максимального відсотку використання CPU під час їх обробки.

Це дозволить проаналізувати, яким чином порядок чергування більш чи менш інтенсивних по навантаженню на процесор запитів впливає на ефективність пакетної обробки в Azure SQL Database.

Результати виконання запитів для режиму B6 представлені на рисунку 3.20.

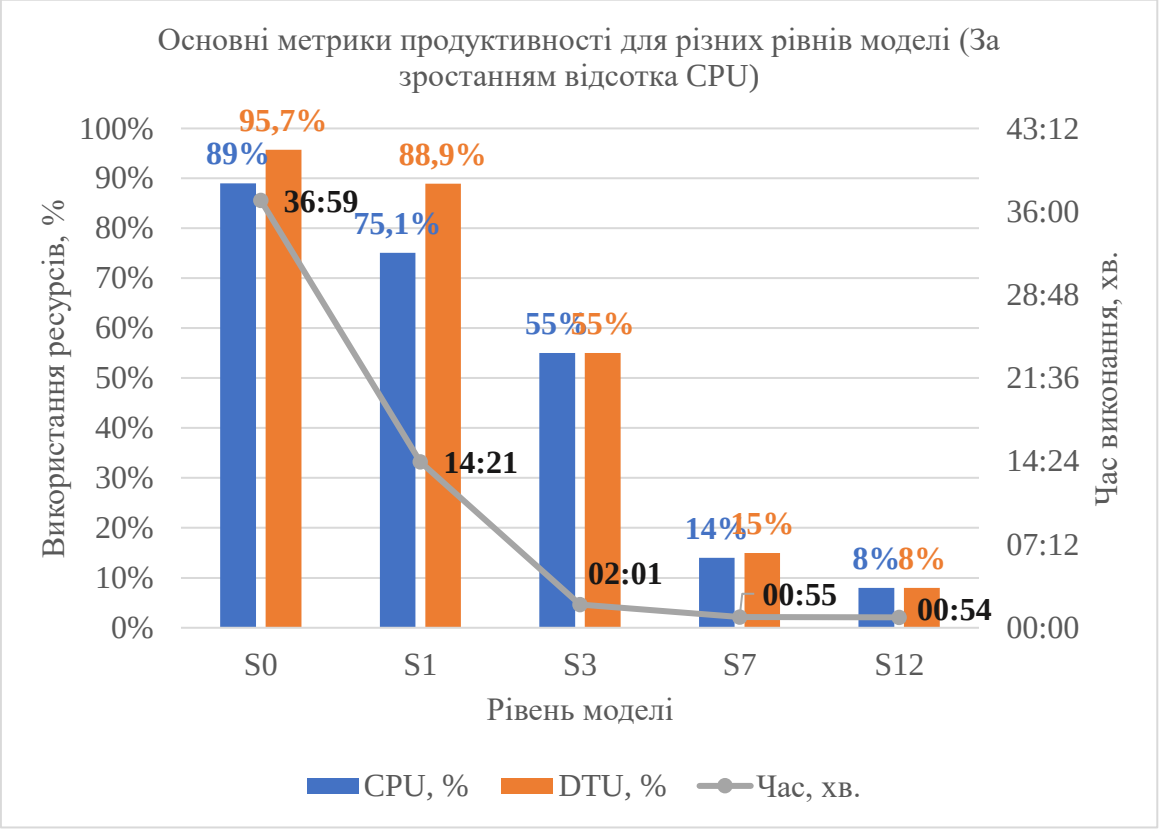

Рисунок 3.20 – Графік основних метрик продуктивності для різних рівнів моделі (За зростанням відсотка CPU)

Порівнюючи результати плану B6 за зростанням використання CPU з базовим планом B1 (довільний порядок), можна побачити, що час виконання на різних рівнях зменшився.

Також на рівнях S0, S1, S3, можна побачити зниження використання ресурсів, що є доволі позитивним ефектом, зниження відбувається через більш рівномірне завантаження системи простішими запитами на початку черги. На рівнях S7 та S12 спостерігається невелике збільшення використання ресурсів у плані P6 порівняно з B1. Це пов'язано з тим, що при плануванні за зростанням навантаження CPU система виділяє максимум потужностей для складних запитів в кінці черги і утримує ці ресурси до завершення їх виконання. Натомість в B1 через довільний порядок, після виділення ресурсів для складного запиту може йти простіший, і система звільняє частину раніше виділених потужностей. Таке неплавне коливання призводить до менш ефективного загального використання ресурсів.

Таким чином, планування пакетів за зростанням використання CPU дозволяє дещо знизити загальне навантаження на систему та прискорити обробку у порівнянні з довільним порядком. Це пояснюється більш плавним наростанням складності запитів в черзі.

Перейдемо до аналізу результатів використання режиму B7, у якому розташування запитів йде у зворотному порядку – спочатку складні, а потім - прості запити. Результати розрахунків метрик продуктивності представлені на рисунку 3.21.

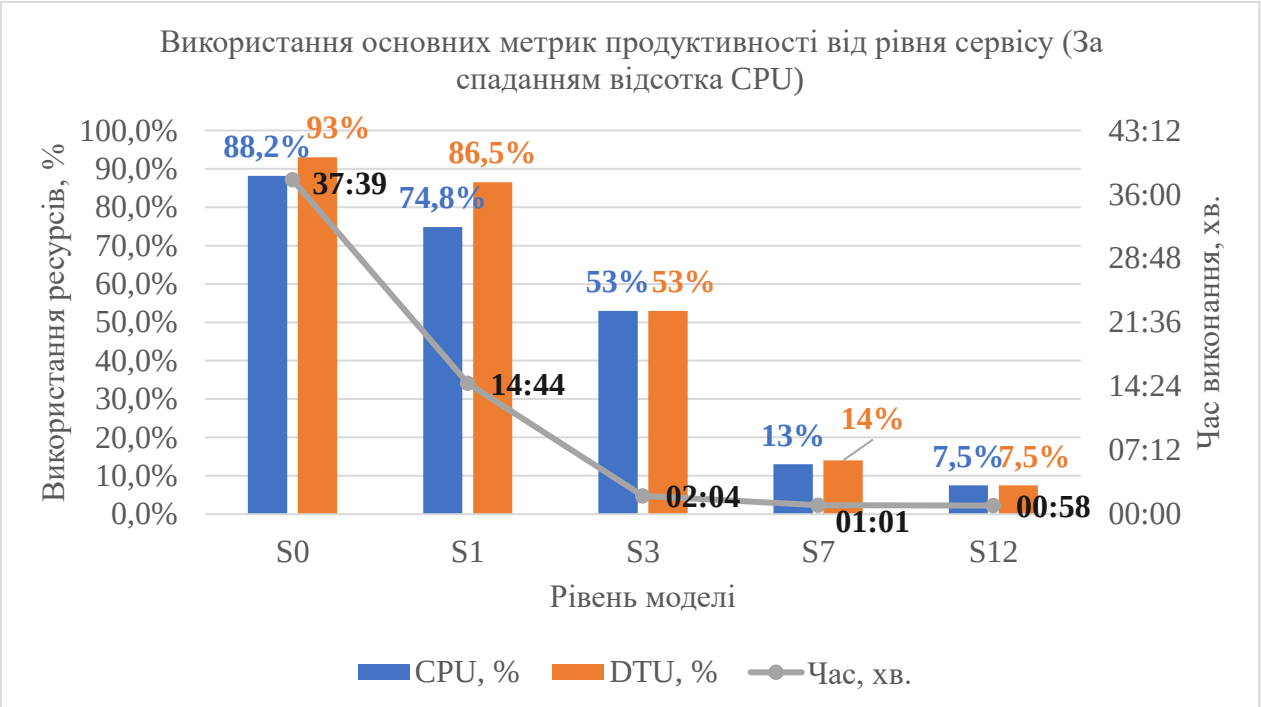

Рисунок 3.21 – Графік основних метрик продуктивності для різних рівнів моделі (За спаданням відсотка CPU)

Порівнюючи результати виконання плану B7 (за спаданням відсотка використання CPU) із попереднім планом B6 (за зростанням відсотка використання CPU), можна побачити, що використання CPU в B7 трохи нижче, ніж в B6 - приблизно на 1-2% на кожному рівні обслуговування. Використання DTU також дещо менше в B7 – різниця становить 1-3% на різних рівнях моделі. Час виконання пакету запитів практично не змінився (різниця складає лише кілька секунд на кожному рівні).

Отже, групування плану у порядку за спаданням навантаження на CPU менш ефективне ніж у порядку зростання. А також, що «важкі» запити на початку уповільнюють обробку та погіршують розподіл ресурсів.

Отже, планування пакетів у порядку спадання навантаження на CPU порівняно із зростанням навантаження дає незначний виграш в економії ресурсів при практично однаковій тривалості обробки.

Після детального аналізу впливу різних способів групування запитів на ефективність пакетної обробки, варто розглянути ще один підхід – групування за пріоритетом важливості запитів (B8).

На відміну від попередніх критеріїв упорядкування, які базувались на технічних характеристиках (час виконання, ресурси, навантаження), даний підхід враховує бізнес-логіку та вимоги конкретного застосунку.

Адже на практиці часто трапляються ситуації, коли одні аналітичні запити мають критичне значення для операційної діяльності і повинні оброблятися у першочерговому порядку порівняно зі стандартними періодичними звітами.

Розглянемо гіпотетичні результати для пакету B8, в якому запити виконувались відповідно до заданих пріоритетів важливості для бізнесу, які були розписані у розділі 3.4.2.

Результати використання режиму B8 представлено на рисунку 3.22.

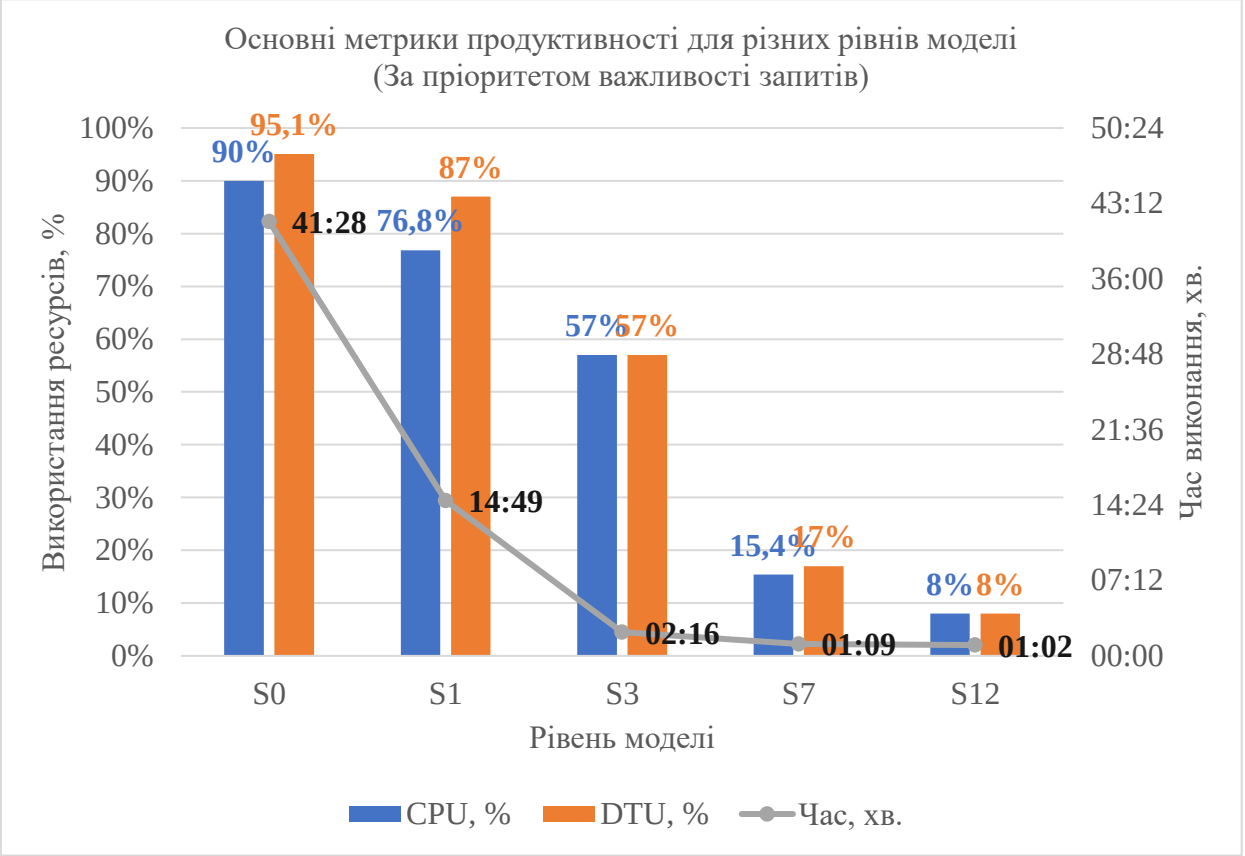

Рисунок 3.22 – Графік основних метрик продуктивності для різних рівнів моделі (За пріоритетом важливості запитів)

Порівнюючи пакети B8 (групування за пріоритетом) та B1 (довільне групування) можна побачити, що на рівнях S0 та S1 в B8 спостерігається скорочення часу обробки порівняно з B1. Використання CPU та DTU також дещо нижче (на 2-10%). На S3 час обробки в B8 менший на 13 секунд при менших об'ємах задіяних ресурсів в порівнянні з B1. Це пояснюється тим, що навантаження на систему при обмежених ресурсах рівня S3 раціональніше розподіляється. Водночас, на вищих рівнях сервісу S7 та S12 спостерігається схожа картина для обох стратегій – низьке навантаження та близькі значення часу виконання.

Отже, застосування пріоритезації не призводить до значущих змін основних результатів у порівнянні з іншими проаналізованими пакетами, окрім B1 (довільний порядок). Це пояснюється тим, що при пріоритезації групування запитів відбувалось не за технічними характеристиками, а лише за заданим рівнем важливості для бізнесу. Отже, вплив на обробку даних є подібним до пакету з довільним порядком, де відбувається випадкове коливання ресурсів під час виконання запитів.

На відміну від інших пакетів, де групування базується на технічних критеріях, пріоритезація не оптимізує розподіл ресурсів системи.

Після детального аналізу ефективності обробки окремих пакетів запитів, важливо провести комплексне порівняння результатів між запропонованими підходами.

Доцільно почати з аналізу та зіставлення показників використання центрального процесору (CPU). Адже цей критичний ресурс безпосередньо впливає на продуктивність аналітичних операцій в базах даних.

Графік зіставлення показників використання центрального процесору (CPU) по пакетах за рівнями моделі представлено на рисунку 3.23.

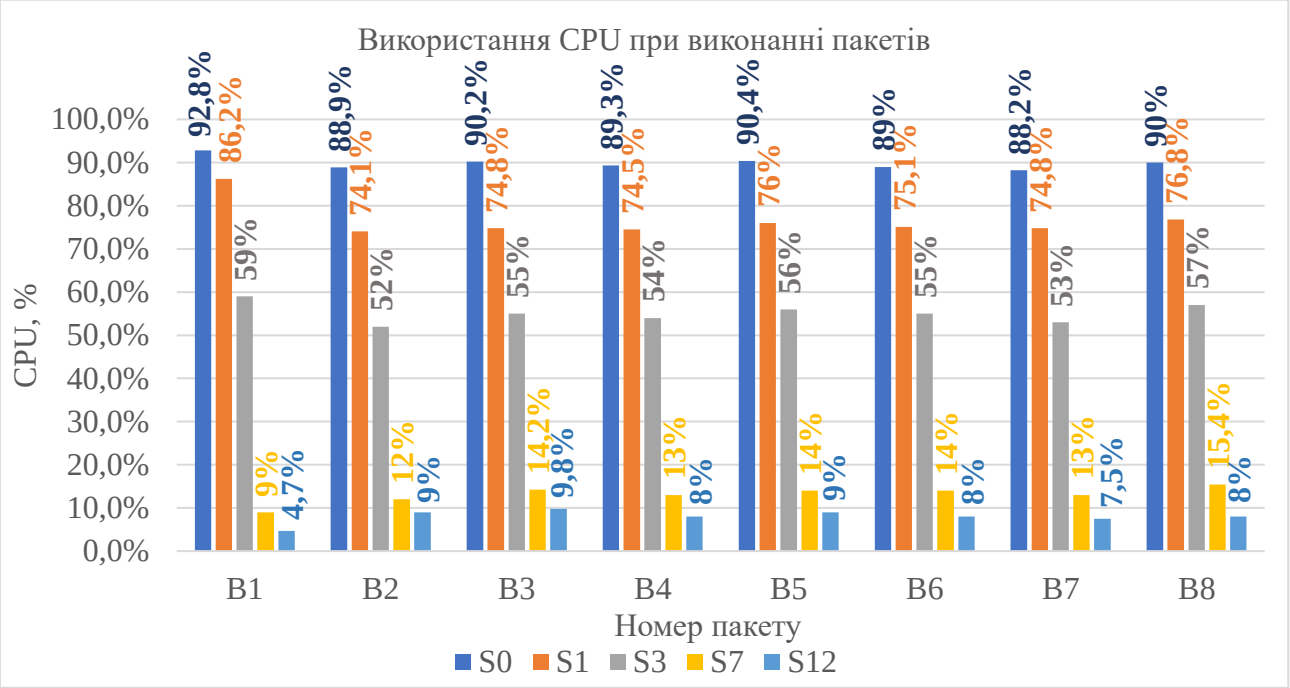

Рисунок 3.23 – Графік використання CPU при виконанні пакетів для різних рівнів моделі

Аналізуючи дані з використання CPU для різних пакетів запитів від B1 до B8, можна сказати, що на рівнях S0 та S1 у всіх пакетів спостерігається високе загальне завантаження CPU, що свідчить про перенавантаження та недостачу ресурсів для оптимального виконання запитів.

Також можна побачити, що в порівнянні з B1 (Довільний порядок запитів), усі інші плани мають зменшення використання CPU, окрім останніх рівнів (S7-

S12), де ресурсів виділяється більше: це пов’язано с тим що ресурси більш якісно розподіляється та використовуються коли є обґрунтоване групування запитів, тому на вищих рівнях зростання використання CPU надає позитивний ефект, тому що не задіяні ресурси більш краще використовуються, що пришвидшує виконання запитів.

Загалом для більш навантажених рівнів моделі (S0-S3) для зменшення навантаження на CPU доцільно використовувати режим B2 (Порядок за зростанням часу виконання запиту) та режим B7 (Порядок за спаданням максимального відсотку використання CPU), – вони оптимально розподіляють навантаження. Для більш високих рівнів (S7-S12) краще за всіх показав режим B3 (Порядок за спаданням часу виконання), при виконанні цього пакету показники використання CPU зросли, що надало більше вільних ресурсів для виконання, які до цього були не задіяні.

Отже, групування запитів у пакети (плани) дозволяє оптимізувати використання ресурсів сервісу.

Далі перейдемо до розгляду використання DTU за пакетами, для того щоб проаналізувати ефективність використання інших ресурсів.

Графік використання DTU по пакетах за рівнями моделі представлено на рисунку 3.24.

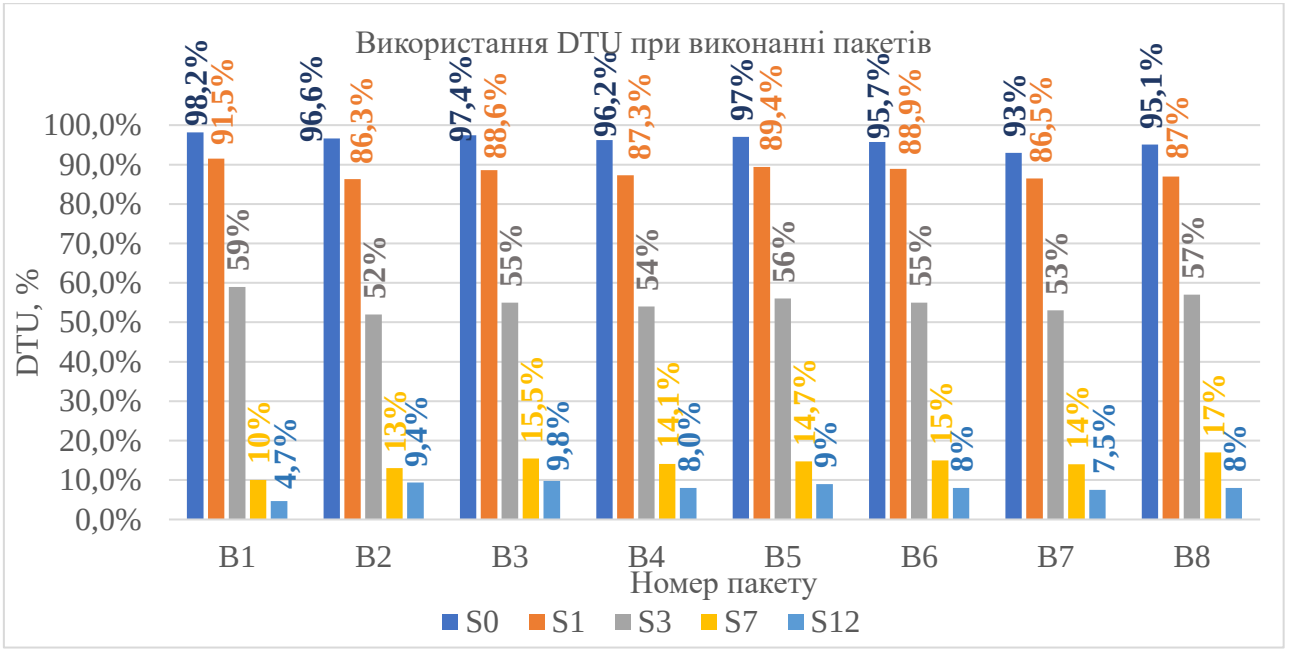

Рисунок 3.24 – Графік використання DTU при виконанні пакетів для різних рівнів моделі

На рівнях S0 та S1 спостерігається високе загальне завантаження DTU (понад 86%) для всіх пакетних режимів, що свідчить про недостатність цих ресурсів та їх перевантаження. В порівнянні з B1 (довільний порядок), всі інші пакети демонструють зниження використання DTU на цих рівнях. Це пояснюється більш оптимальним розподілом навантаження при групуванні запитів.

На вищих рівнях S7-S12 спостерігається загальне низьке завантаження DTU (до 17%), що свідчить про надлишок цих ресурсів. Тут деяке збільшення використання в окремих пакетах (B3, B5, B8) є виправданим, оскільки краще задіяно наявні можливості.

Отже, для оптимізації використання DTU на нижчих перевантажених рівнях найкраще підходять пакети B7 та B4, а на високих рівнях з резервом ресурсів – B3.

Загалом найбільш збалансованим є пакет B4, який ефективно розподіляє навантаження як на нижчих насичених рівнях, так і на верхніх з надлишком ресурсів.

Тепер розглянемо загальний час виконання пакетів для оцінки продуктивності.

Графік часу виконання пакетів для різних рівнів моделі представлено на рисунку 3.25.

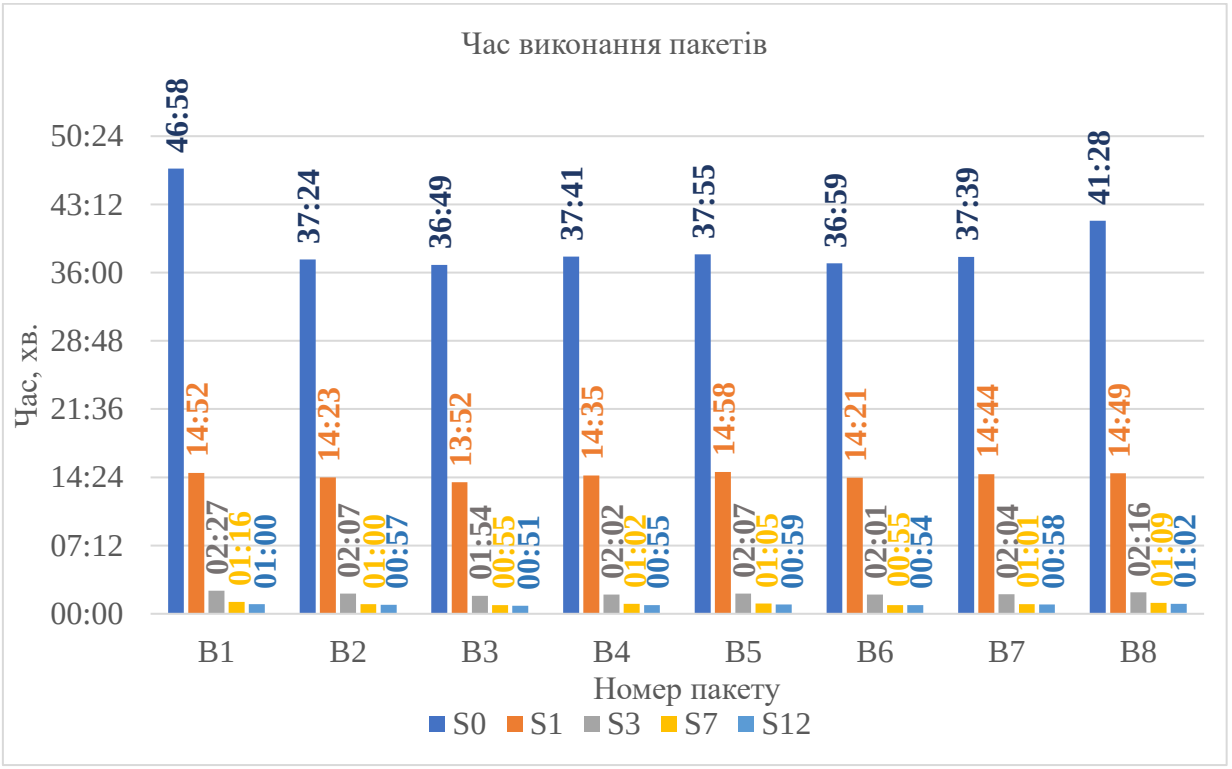

Рисунок 3.25 – Графік часу виконання пакетів для різних рівнів моделі

Отримані результати свідчать, що на рівні S0 спостерігається найтриваліший час обробки пакетів запитів для всіх досліджуваних вибірок (понад 36 хвилин). Це пов'язано з обмеженістю виділених ресурсів. Аналогічна ситуація спостерігається і на рівні S1, де час виконання складає 13-14 хв. для всіх пакетних режімів.

У порівнянні з B1, всі інші пакети демонструють скорочення часу на усіх рівнях завдяки більш ефективному розподілу навантаження при групуванні запитів за певними критеріями (послідовностями складання планів запитів).

На вищих рівнях S7-S12 час виконання є найменшим (1-2 хв.) через те, що вони мають дуже багато ресурсів, яких вистачає для швидкої паралельної обробки запитів у пакеті.

Загалом, найбільше прискорення на всіх рівнях продемонстрував пакет B3 з групуванням за спаданням часу виконання, що дозволило ефективніше розподілити навантаження ресурсів. Для прискорення обробки на нижчих рівнях моделі найкращим є пакети B3 та B6.

Отже, для оптимізації часу обробки доцільно застосовувати B3 або подібні підходи групування з урахуванням тривалості виконання запитів.

Задля порівняння ефективності використання хмарних ресурсів і можливості їх застосування у порівнянні з локальними серверами, було проведено додаткові вимірювання часу виконання базових пакетів запитів – B1, B2 та B3 з використанням локальних обчислювальних потужностей. Зафіксовані показники склали 02:36, 02:34 та 02:37 хв. відповідно.

Порівнюючи отримані показники з результатами для хмарного середовища Azure SQL Database, можна зробити висновок, що починаючи з рівня навантаження S3 і вище, усі досліджувані пакети запитів (B1-B8) продемонстрували менший час виконання, ніж на локальному ресурсі.

Це свідчить про те, що за умов середнього та високого навантаження хмарний сервіс має значну перевагу в продуктивності обробки даних.

Окрім швидкодії, важливо оцінити і економічну ефективність отриманих рішень. Було проаналізовано вартість виконання різних пакетів запитів в Azure залежно від рівня ресурсів.

Результати щодо оцінювання вартості запропонованих режимів наведено на рисунку 3.26.

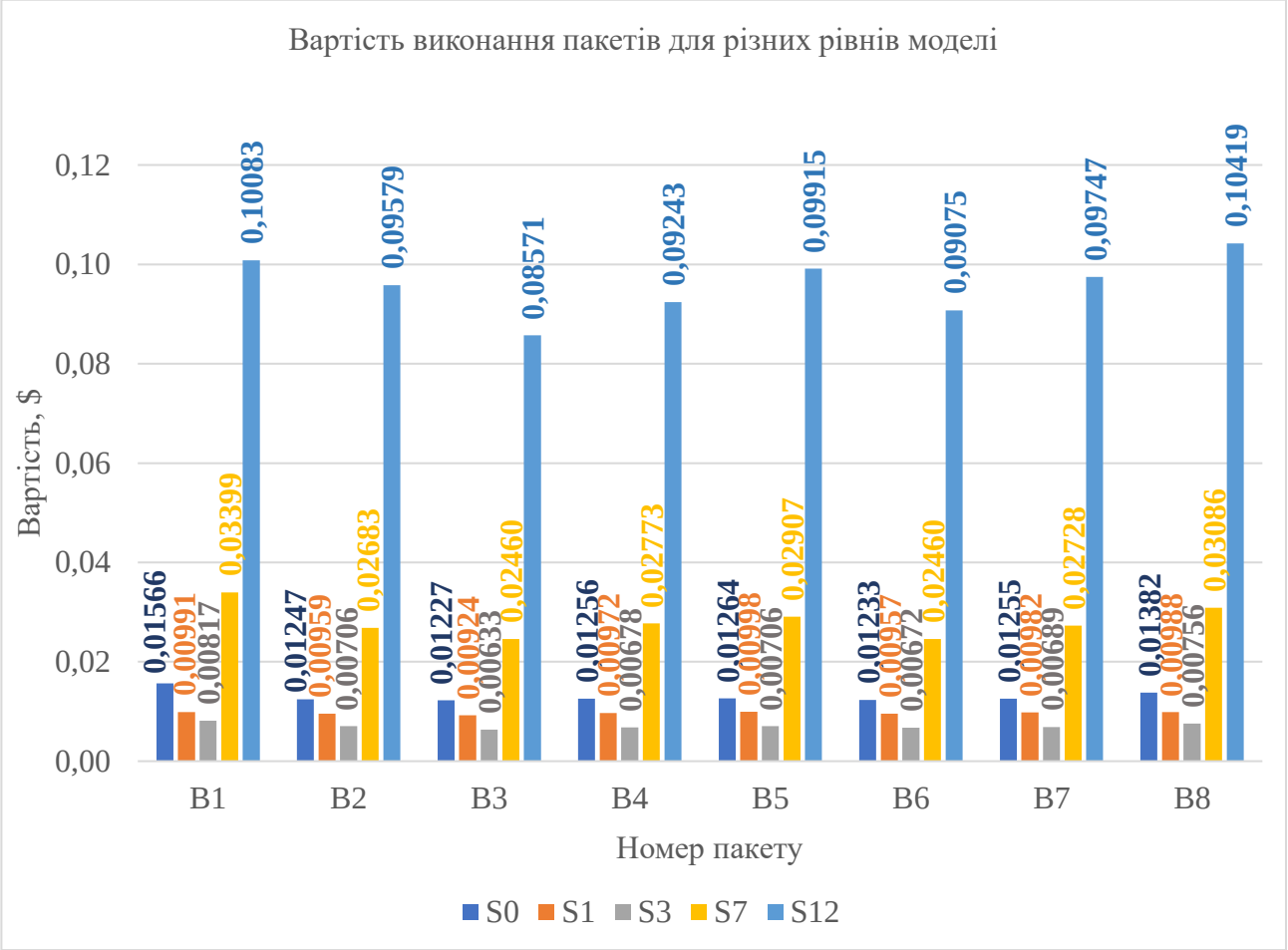

Рисунок 3.26 – Графік вартості виконання пакетів для різних рівнів моделі

Аналізуючи дані щодо вартості виконання різних пакетів запитів в залежності від рівня виділених хмарних ресурсів, можна сказати, що найвищі показники вартості обробки всіх пакетів запитів спостерігаються на рівні обслуговування S12 (він є найдорожчим), далі йде рівень S7.

Найнижчі ціни зафіксовано на рівнях S0-S3. Проте порівнюючи вартість між пакетами, безумовно найнижчі показники на всіх рівнях сервісу демонструє пакет запитів B3, сформований за критерієм спадання часу виконання.

B3 є оптимальним пакетом для мінімізації витрат на обробку даних, що характеризується найнижчою вартістю серед усіх проаналізованих рівнів при достатній кількості ресурсів для вирішення аналітичних задач реального масштабу.

Для мінімізації бюджету рекомендується використовувати саме B3, сформованого за критерієм упорядкування запитів у порядку спадання часу їх обробки. Саме цей підхід забезпечує найнижчу вартість виконання аналітичних задач на всіх досліджуваних рівнях хмарного сервісу Azure SQL Database.

Отже, на основі проведеного дослідження ефективності різних режимів формування та виконання пакетів SQL-запитів в хмарному середовищі Azure SQL Database можна зробити наступні висновки:

1. Щодо ефективності різних режимів пакетів:

– пакет B3, в якому запити групуються за критерієм спадання часу їх виконання, виявився найефективнішим. Цей режим дозволяє спочатку задіяти максимум ресурсів для найбільш тривалих і ресурсоємних запитів, що прискорює їх виконання. Такий підхід дозволяє ефективніше розподілити навантаження на систему та забезпечує найбільше прискорення обробки даних на всіх досліджуваних рівнях моделі в порівнянні з іншими режимами;

– пакет B6, в якому запити групуються за критерієм зростання використання ресурсів CPU, показав гарні результати за ефективністю за часом виконання, це другий пакет по ефективності виконання після B3. Поступове нарощування навантаження від простих до складних запитів в цьому режимі забезпечує більш плавний розподіл ресурсів процесора. Це дає змогу оптимізувати використання ресурсів процесора та оптимальніше розподілити навантаження на систему;

– пакет B7, в якому запити групуються за спаданням максимального відсотку використання CPU, незважаючи на середні показники ефективності на більшості рівнів, все ж продемонстрував кращі результати на нижчих рівнях S0 та S1. Це пов'язано з тим, що на даних рівнях спостерігається найвище перевантаження ресурсів процесора та DTU для всіх пакетів запитів. Пакет B7, завдяки початковому виконанню найбільш ресурсоємних по CPU запитів, дозволяє зменшити загальне перевантаження системи. В результаті на рівнях S0 та S1 цей пакет демонструє найкращі показники ефективності порівняно з іншими режимами формування пакетів SQL-запитів;

– пакет B1, в якому запити групуються в довільному порядку, виявився найменш ефективним. Хаотичне чергування простих і складних запитів призводить

до різких коливань у використанні ресурсів, їх нераціонального розподілу та уповільнення обробки даних. Цей режим продемонстрував найгірші результати через неоптимальний розподіл ресурсів;

– усі інші пакети, пов’язані з групуванням по технічним характеристикам, покращили час виконання в порівнянні з пакетом B1, який групується у довільному порядку. Проте жоден з цих пакетів не показав кращих результатів, ніж B3, B7 та B6. Це свідчить, що найважливішими критеріями оптимізації є час виконання запитів та завантаження процесора. Інші технічні характеристики мають менший вплив на ефективність.

2. Щодо оптимальності рівнів моделі:

– найбільш оптимальним виявився рівень S3 моделі обслуговування Azure SQL Database, який забезпечує найкраще співвідношення продуктивності та вартості обробки даних;

– рівні S0 та S1 з найменшою кількістю ресурсів виявилися перевантаженими та неефективними для даного навантаження;

– вищі рівні S7 та S12 характеризуються надлишком ресурсів, що призводить до невиправданого збільшення витрат на обробку даних.

Для порівняння ефективності різних режимів роботи SQL Database також зіставимо пакетний режим (на прикладі ефективного пакету B3 на рівні S3) з інтерактивним режимом виконання SQL-запитів (на тому ж рівні S3).

Результати представлені на рисунку 3.27.

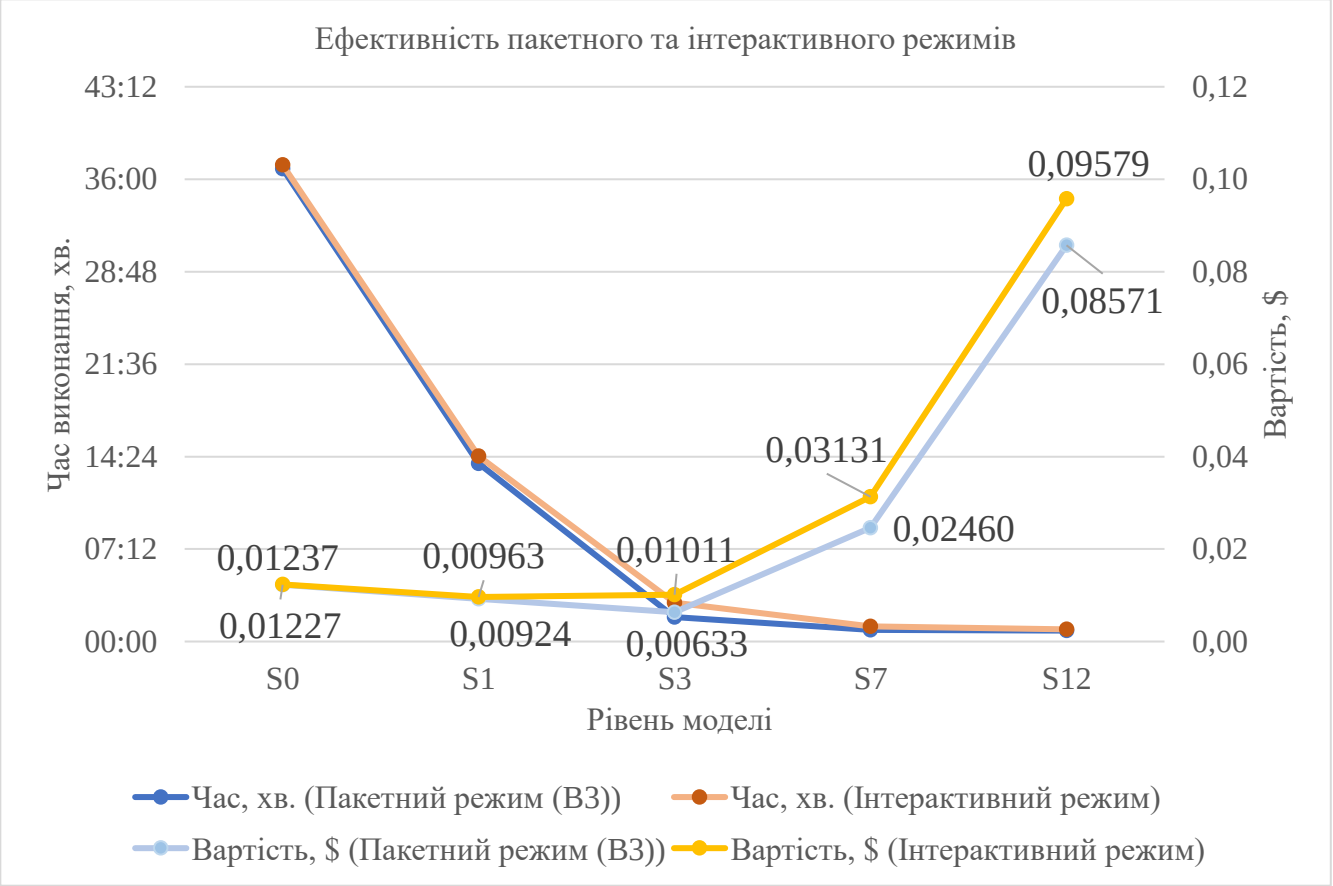

Рисунок 3.27 – Графік порівняння ефективності пакетного (B3) та інтерактивного режимів для різних рівнів моделі

Як видно, пакетний режим перевершує інтерактивний за показниками часу та вартості на всіх рівнях моделі сервісу. Найбільша різниця спостерігається на середніх рівнях S3 та S7, де пакетний режим дозволяє скоротити час в 1,5 рази (150%) та зменшити витрати на 10-37% порівняно з послідовним виконанням запитів. Результати підтверджують, що запропонований пакетний режим дозволяє суттєво прискорити аналітичну обробку даних та зменшити вартість виконання запитів у хмарному сервісі Azure SQL Database.

Задля деталізації дослідження порівняємо пакетний режим (взявши ефективний пакет B3 з рівнем моделі S3) з інтерактивним режимом (також на рівні S3, взявши сумарний час виконання усіх запитів) та на локальному ресурсі за часом виконання усіх запитів. Результати представлені на рисунку 3.28.

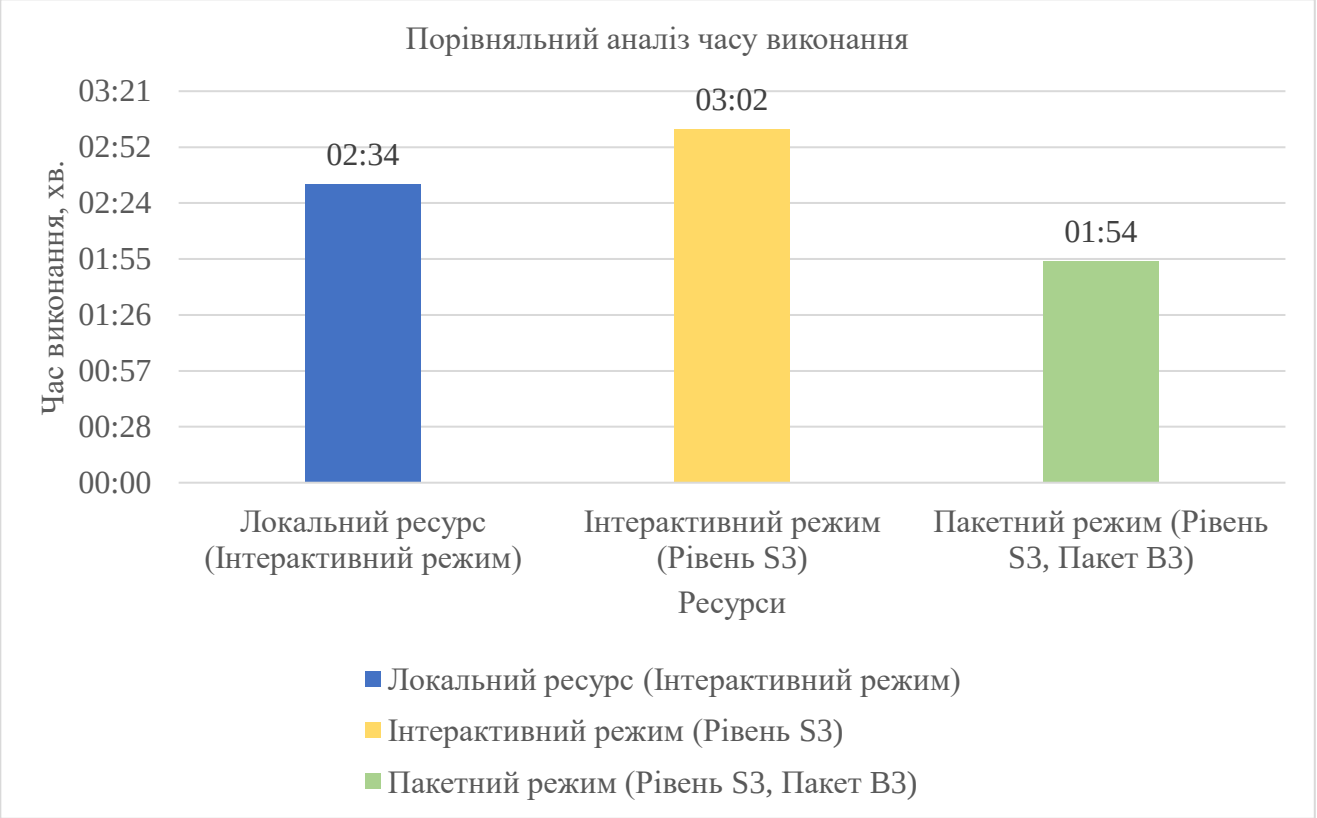

Рисунок 3.28 – Порівняльний аналіз часу виконання для пакетного, інтерактивного режимів (рівень S3) та на локальному ресурсі

Результати свідчать про те, що саме пакетний режим обробки даних дозволяє не просто наздогнати, але й перевершити швидкодію локального ресурсу на 35%. Це пояснюється ефективним розподілом навантаження та гнучким масштабуванням ресурсів хмари.

Таким чином, запропонований підхід до пакетної обробки SQL-запитів демонструє свої переваги не лише в порівнянні з інтерактивним режимом, але й з локальним виконанням. Це підтверджує доцільність використання хмарних технологій для аналітичних систем.

Отже, результати дослідження довели, що вибір оптимальної стратегії формування та виконання пакетів запитів є вкрай важливим для максимізації ефективності аналітичних систем на базі хмарних технологій Azure SQL Database. Це може бути використано при проєктуванні систем обробки даних на базі Azure SQL Database для вибору оптимальної стратегії, що забезпечить максимальну продуктивність їх використання.

ВИСНОВКИ

У даній роботі розглянуто задачу оптимізації процесів обробки великих обсягів даних на торгівельних платформах в умовах хмарного середовища. Було сформовано теоретичну базу щодо технологій Big Data, хмарних сервісів та реляційних баз даних. У ході виконання роботи було досліджено можливості провідних хмарних платформ, проаналізовано їх сервіси та обґрунтовано вибір платформи Microsoft Azure.

Спроєктовано моделі реляційної бази даних на основі аналізу функціонування великих торгівельних компанії та реалізовано фізичну модель БД у SQL Server. За допомогою SQL-скриптів розроблено генератор тестових даних та згенеровано реалістичні тестові дані для таблиць БД. Розроблено SQL-запити для аналізу даних щодо продажів, асортименту, клієнтської бази тощо на локальному ресурсі. Результатом цього етапу стало виявлення факторів, що вплинуть на продуктивність використання хмарної бази даних Azure SQL Database, зокрема, час оброблення даних, завантаженість CPU, об’єм даних, кількість та порядок виконання SQL-запитів.

Для оцінки продуктивності було розгорнуто аналогічну БД на Azure SQL Database і порівняно швидкодію запитів на різних рівнях моделі та локальному ресурсі. Також було проаналізовано використання CPU, DTU та час відгуку при реалізації запитів.

Розроблено підходи до оптимізації запитів та режими їх виконання (засобами групування, – упорядкуванням за часом виконання запитів та використаними ресурсами). У ході тестування було перевірено працездатність запропонованих підходів та обґрунтовано їх ефективність. В результаті було виявлено ключові фактори, що впливають на ефективність роботи хмарного сервісу СУБД, та надано рекомендації щодо оптимальної конфігурації при вибору рівнів моделі сервісу для потреб торгівельної компанії.

Запропоновані рішення може бути використано торгівельними компаніями для оптимізації бізнес-аналітики на основі хмарних технологій з використанням сервісів оброблення великих даних.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Network World. IDC: Expect 175 zettabytes of data worldwide by 2025. URL: https://www.networkworld.com/article/3325397/idc-expect-175-zettabytes-of-data-worldwide-by-2025.html (дата звернення: 10.11.2023)

2. Мінухін С., Башкіров М. Моделювання роботи з базами даних торгівельних компаній на хмарних платформах. с. 45-46 // Інформаційні системи та технології: матеріали 12-ї Міжнародної науково-технічної конференції. Частина 2. Молодіжна секція, м. Харків, 01 грудня 2023 року / наук. ред. В.В. Безкоровайний, L. Petryshyn, З.В. Дудар, Ю.В. Міщеряков. – Х.: ХНУРЕ, 2023. – 80 с. URL: https://istconf.sedep.online/archive/ist_2023_part_2.pdf

3. Termin.in.ua. BIG DATA (Великі дані, Біг Дата) — це що таке, суть і визначення. URL: https://termin.in.ua/big-data-velyki-dani/ (дата звернення: 08.01.2024).

4. Microsoft Azure. What is Cloud Computing? URL: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-cloud-computing/ (дата звернення: 10.11.2023)

5. Amazon Web Services. Types of Cloud Computing. URL: https://aws.amazon.com/ru/types-of-cloud-computing/ (дата звернення: 10.11.2023)

6. Amazon Web Services. What is Big Data? URL: https://aws.amazon.com/ru/what-is/big-data/?nc1=h_ls (дата звернення: 10.11.2023)

7. DevOps School. Top 10 Cloud Platforms. URL: https://www.devopsschool.com/blog/top-10-cloud-platforms/ (дата звернення: 10.11.2023)

8. Digiexe. 7 прикладів програмного забезпечення інфраструктури як послуги 2024. URL: https://digiexe.com/uk/blog/infrastructure-software-as-a-service-examples/ (дата звернення: 08.01.2024)

9. Wezom. Що таке модель PaaS - Платформа як послуга, користь хмарних платформ для бізнесу. URL: https://wezom.com.ua/ua/blog/scho-take-paas-i-komu-vin-potriben (дата звернення: 08.01.2024)

10. Amazon Web Services. Що таке SaaS? - Пояснення ПЗ як послуги. URL:

https://aws.amazon.com/ru/what-is/saas/ (дата звернення: поточна дата)

11. Microsoft Learn. Compare AWS and Azure database technology - Azure Architecture Center. URL: https://learn.microsoft.com/en-us/azure/architecture/aws-professional/databases (дата звернення: 08.01.2024)

12. DZone. Amazon RDS vs Azure SQL — Know the Key Differentia. URL: https://dzone.com/articles/amazon-rds-vs-azure-sql-know-the-key-differentiato (дата звернення: 08.01.2024)

13. Clouduts. AWS vs Azure Database Service Comparison. URL: https://clouduts.com/aws-vs-azure-database-service-comparison/ (дата звернення: 08.01.2024)

14. TrustRadius. Amazon RDS vs Azure SQL Database. URL: https://www.trustradius.com/compare-products/amazon-relational-database-service-vs-sql-azure (дата звернення: 08.01.2024)

15. Global24. Online Sales Ukraine Ecommerce 2023 Analysis. URL: https://global24.com/en/blog/online-sales-ukraine-ecommerce-2023-analysis/ (дата звернення: 10.11.2023)

16. EcommerceDB. Markets UA All. URL: https://ecommercedb.com/markets/ua/all (дата звернення: 10.11.2023)

17. Infovision. 6 Key Benefits of Cloud Computing in Retail. URL: https://www.infovision.com/blog/6-key-benefits-of-cloud-computing-in-retail (дата звернення: 10.11.2023)

18. Statista. Retail turnover of retail enterprises Ukraine. URL: https://www.statista.com/statistics/1033675/retail-turnover-of-retail-enterprises-ukraine/ (дата звернення: 10.11.2023)

19. Statista. Cloud Computing in Europe. URL: https://www.statista.com/topics/8472/cloud-computing-in-europe/#topicOverview (дата звернення: 10.11.2023)

20. HSC. Benefits of Cloud Computing in Retail. URL: https://www.hsc.com/resources/blog/benefits-of-cloud-computing-in-retail/ (дата звернення: 10.11.2023)

21. Вікіпедія. Walmart. URL: https://en.wikipedia.org/wiki/Walmart (дата

звернення: 08.01.2024)

22. The Strategy Watch. Pricing Strategy Walmart. URL: https://www.thestrategywatch.com/pricing-strategy-walmart/ (дата звернення: 08.01.2024)

23. Extensiv. Supply Chain Management Walmart. URL: https://www.extensiv.com/blog/supply-chain-management/walmart (дата звернення: 08.01.2024)

24. CIO Dive. Walmart outlines cloud strategy to investors. URL: https://www.ciodive.com/news/walmart-cloud-strategy-investors/573175/ (дата звернення: 08.01.2024)

25. Supermarket Perimeter. Walmart succeeds as an omnichannel retailer. URL: https://www.supermarketperimeter.com/articles/9582-walmart-succeeds-as-an-omnichannel-retailer (дата звернення: 08.01.2024)

26. Zippia. Walmart Statistics. URL: https://www.zippia.com/advice/walmart-statistics/ (дата звернення: 08.01.2024)

27. Вікіпедія. List of Walmart brands. URL: https://en.wikipedia.org/wiki/List_of_Walmart_brands (дата звернення: 08.01.2024)

28. Walmart. All Departments. URL: https://www.walmart.com/all-departments (дата звернення: 08.01.2024)

29. Capital Counselor. Walmart Statistics. URL: https://capitalcounselor.com/walmart-statistics/ (дата звернення: 08.01.2024)

30. Warehouse Ninja. Walmart Distribution Center Locations. URL: https://warehouse.ninja/walmart-distribution-center-locations/ (дата звернення: 08.01.2024)

31. Walmart Corporate. Location Facts. URL: https://corporate.walmart.com/about/location-facts (дата звернення: 08.01.2024)

32. Walmart Corporate. How many people work at Walmart? URL: https://corporate.walmart.com/askwalmart/how-many-people-work-at-walmart (дата звернення: поточна дата)

33. Wikipedia. Amazon (company). URL: https://en.wikipedia.org/wiki/Amazon_%28company%29 (дата звернення: 08.01.2024).

34. About Amazon. Who we are. URL: https://www.aboutamazon.com/about-us (дата звернення: 08.01.2024).

35. Nuoptima. Find the next best seller for your Amazon store. URL: https://nuoptima.com/amazon-product-categories (дата звернення: 08.01.2024).

36. Business Model Analyst. Amazon Business Model - How Amazon Makes Money. URL: https://businessmodelanalyst.com/amazon-business-model/ (дата звернення: 08.01.2024).

37. Wikipedia. List of Amazon products and services. URL: https://en.wikipedia.org/wiki/List_of_Amazon_products_and_services (дата звернення: 08.01.2024).

38. EcomCrew. In-Depth Study of All 88 Amazon Private Label Brands. URL: https://www.ecomcrew.com/amazons-private-label-brands/ (дата звернення: 08.01.2024).

39. EarthWeb. How Many Products Does Amazon Have in 2024? URL: https://earthweb.com/how-many-products-does-amazon-have/ (дата звернення: 08.01.2024).

40. Serpwatch.io. Amazon Customer Demographics. URL: https://serpwatch.io/blog/amazon-customer-demographics/ (дата звернення: 08.01.2024).

41. Jungle Scout. Amazon Statistics. URL: https://www.junglescout.com/blog/amazon-statistics/ (дата звернення: 08.01.2024).

42. About Amazon. Delivery & Logistics. URL: https://www.aboutamazon.com/what-we-do/delivery-logistics (дата звернення: 08.01.2024).

43. About Amazon. Corporate Offices. URL: https://www.aboutamazon.com/workplace/corporate-offices (дата звернення: 08.01.2024).

44. Statista. Number of Amazon employees. URL: https://www.statista.com/statistics/234488/number-of-amazon-employees/ (дата звернення: 08.01.2024).

45. HOSTiQ.ua. База даних (БД) – Що це таке? Визначення бази даних. URL:

https://hostiq.ua/wiki/ukr/database/ (дата звернення: 08.01.2024)

46. Вікіпедія. Система управління базами даних. URL: https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%96%D0%BD%D0%BD%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85 (дата звернення: 08.01.2024)

47. Вікіпедія. Microsoft SQL Server. URL: https://en.wikipedia.org/wiki/Microsoft_SQL_Server (дата звернення: 08.01.2024)

48. Вікіпедія. Transact-SQL. URL: https://uk.wikipedia.org/wiki/Transact-SQL (дата звернення: 08.01.2024)

49. Microsoft Learn. SQL Server Deadlocks Guide. URL: https://learn.microsoft.com/en-us/sql/relational-databases/sql-server-deadlocks-guide?view=sql-server-ver16 (дата звернення: 08.01.2024)

50. S. Minukhin, Performance Study Of The DTU Model For Relational Databases On The Azure Platform, Innovative Technologies and Scientific Solutions for Industries, 2022, No 1 (19), pp. 27- 39. DOI: https://doi.org/10.30837/ITSSI.2022.19.027

51. Microsoft Azure. Azure SQL Database. URL: https://azure.microsoft.com/en-us/products/azure-sql/database/#Resources (дата звернення: 08.01.2024)

52. Microsoft Learn. Azure SQL Database DTU Benchmark. URL: https://learn.microsoft.com/en-us/azure/azure-sql/database/dtu-benchmark?view=azuresql (дата звернення: 08.01.2024)

53. Microsoft Learn. Azure SQL Database Resource Limits for Single Databases. URL: https://learn.microsoft.com/uk-ua/azure/azure-sql/database/resource-limits-vcore-single-databases?view=azuresql (дата звернення: 08.01.2024)

54. Microsoft Azure. Azure SQL Database Pricing Details. URL: https://azure.microsoft.com/en-us/pricing/details/azure-sql-database/single/ (дата звернення: 08.01.2024)