

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ

Пояснювальна записка

до дипломної роботи

МАГІСТРА

на тему: "МОДЕЛЮВАННЯ ТА МОНІТОРИНГ РОБОТИ З ВЕЛИКИМИ
ДАНИМИ НА ХМАРНІЙ ПЛАТФОРМІ AZURE"

Виконав:

студент 2 року навчання

групи 8.04.122.010.20.1

спеціальності 122 "Комп'ютерні науки"

Лістєв Захар Сергійович

Керівник:

д.т.н. проф. Мінухін Сергій

Володимирович

Харків – 2021 рік

РЕФЕРАТ

Пояснювальна записка до магістерської дипломної роботи містить: 65 стор., 24 рис., 16 табл., 1 додатка (ів), 24 джерел.

Метою роботи є дослідження моделей придбання для роботи з великими даними на платформі Azure, для виявлення оптимального рішення з економічної та продуктивної точки зору.

Об'єктом дослідження є хмарна платформа Microsoft Azure.

Предметом дослідження є проведення експерименту для дослідження найбільш задовільного рішення для роботи з великими даними на хмарній платформі Microsoft Azure.

Результатами дослідження є методи, алгоритми та програмне забезпечення... Алгоритми реалізовані на основі програмного забезпечення, розробленого в середовищі Microsoft Visual Studio мовою C#.

Отримані результати можуть бути впроваджені на підприємствах які зберігають та обробляють велику кількість даних.

ABSTRACT

Explanatory note to the master's thesis contains: 65 p., 24 fig., 16 table., 1 app., 24 sources.

The aim work is investigation of acquisition models for working with big data on the Azure platform, to identify the optimal solution from an economic and productive point of view

The object of research is Microsoft Azure cloud platform.

The subject of research is conducting an experiment to explore the most satisfactory solution for working with big data on the Microsoft Azure cloud platform.

The results of the research are methods, algorithms and software for... The algorithms are implemented on the basis of software developed in the Microsoft Visual Studio environment in C #.

The obtained results can be implemented in enterprises that store and process large amounts of data.

ЗМІСТ

| | |
|--------------------------------------------------------------------------------------------------------|----|
| ВСТУП | 7 |
| РОЗДІЛ 1. АНАЛІЗ МОДЕЛЕЙ ПРИДБАННЯ ДЛЯ ОБРОБКИ РЕЛЯЦІЙНИХ БАЗ ДАНИХ НА ХМАРНИХ ПЛАТФОРМАХ | 9 |
| 1.1 Аналіз та характеристика сервісів хмарних платформ | 9 |
| 1.2 Аналіз моделей придбання хмарної платформи Azure | 14 |
| 1.3 Характеристики моделі придбання DTU | 19 |
| 1.4 Засоби генерації великих даних | 26 |
| РОЗДІЛ 2. МОДЕЛЮВАННЯ МОДЕЛІ ПРИДБАННЯ AZURE | 31 |
| 2.1. Генерація тестового набору даних | 31 |
| 2.2. Моніторинг і налаштування продуктивності в базі даних Azure SQL і керованому екземплярі Azure SQL | 33 |
| РОЗДІЛ 3. ПРОВЕДЕННЯ ТА АНАЛІЗ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ | 45 |
| 3.1 Проектування та опис тестової бази даних | 45 |
| 3.2. Генерація тестового набору даних | 46 |
| 3.3. Аналіз ефективності використання моделей придбання для роботи з реляційними БД | 46 |
| 3.4 Аналіз завантаженості моделей придбання Azure | 54 |
| 3.4. Візуалізація результатів експериментальних досліджень засобами Microsoft Azure | 59 |
| 3.4.1. Огляд засобів візуалізації Microsoft Azure | 59 |
| ВИСНОВКИ | 60 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 61 |
| ДОДАТКИ | 64 |

ВСТУП

У сучасному світі існує велика кількість бізнесів, яким потрібно зберігати значний обсяг даних в структурованому вигляді і при цьому їх частини можуть знаходитися в різних місцях.

Цю проблему можна вирішити за допомогою Microsoft Azure SQL Database.

Microsoft Azure (Windows Azure) - назва хмарної платформи Microsoft. Надає можливість розробки і виконання програм і зберігання даних на серверах, розташованих в розподілених дата-центрах. Спочатку називалася Windows Azure. У 2014 році платформа була перейменована в Microsoft Azure.

Microsoft Azure реалізує хмарні моделі платформи як сервісу (PaaS) та інфраструктури як сервісу (IaaS). Можливе використання як сторонніх, так і сервісів Microsoft в якості моделі (SaaS). Працездатність платформи Microsoft Azure забезпечує мережу глобальних дата-центрів Microsoft.

Microsoft Azure реалізує такий тип сервісу, як - інфраструктура як сервіс. Модель надання інфраструктури (апаратних ресурсів) реалізує можливість оренди таких ресурсів, як сервери, пристрої зберігання даних та мережеве обладнання. Управління всією інфраструктурою здійснюється постачальником, споживач управляє тільки операційною системою і встановленими додатками.

Microsoft Azure складається з:

Compute - компонент, який реалізує обчислення на платформі Windows Azure.

Storage - компонент сховища, який надає можливість масштабуємості сховища. Сховище не має можливості використовувати реляційну модель і є альтернативною, «хмарної» версією SQL Server.

Fabric - Microsoft Azure Fabric за своїм призначенням є «контролером» і ядром платформи, виконуючи функції моніторингу в реальному часі, забезпечення відмово-стійкості, виділення потужностей, розгортання серверів, віртуальних машин і додатків, балансування навантаження і управління обладнанням.

Практично всі сервіси Microsoft Azure мають інтерфейс взаємодії API, побудований на основі обмежень для розподілених гіпер-систем REST, що дозволяє розробникам використовувати «хмарні» сервіси з будь-якою операційною системою, пристроя і платформи.

Крім того, користувачі можуть створювати і редагувати сервіси, користуючись візуальним веб інтерфейсом порталу Azure. Портал дозволяє

управляти налаштуваннями сервісів, редагувати права доступу, моніторити стан ресурсів і управляти білінгом.

За допомогою цього сервісу виникає можливість зберігати і оброблювати реляційні дані, особливо робота з Big Data, а також виконувати генерацію звітності.

Хмарна платформа надає змогу покупцям обирати і платити за ті гігабайти інформації, які будуть зберігатися і за час, який він витратив на роботу з Azure. У клієнта Azure є можливість протестувати цю платформу з малими витратами, а потім прийняти рішення — продовжувати її використання, чи ні. Ще однією перевагою служби є те, що відбувається автоматичне резервне копіювання. Також, не потрібно укладати жодних договорів, завдяки чому економляться час і гроші власника бізнесу.

Використання Microsoft Azure SQL Database — актуальне для вирішення багатьох бізнес-задач. Використання моделі на базі DTU (одиниці виміру продуктивності рівня обслуговування) завжди було способом визначення розміру і налаштування бази даних SQL Azure, яка до недавнього часу розробляла модель на основі vCore для визначення розміру хмарних баз даних.

Мета – дослідження моделей придбання для роботи з великими даними на платформі Azure, для виявлення оптимального рішення з економічної та продуктивної точки зору.

За для досягнення поставленої мети необхідно вирішити наступні задачі:

Аналіз існуючих рішень для роботи з великими даними;

проаналізувати літературні джерела;

розробити моделі, алгоритми для проведення експерименту;

провести експеримент для отримання результатів дослідження;

візуалізувати отримані результати дослідження;

провести оцінку та аналіз результатів дослідження;

1. АНАЛІЗ МОДЕЛЕЙ ПРИДБАННЯ ДЛЯ ОБРОБКИ РЕЛЯЦІЙНИХ БАЗ ДАНИХ НА ХМАРНИХ ПЛАТФОРМАХ

1.1 Аналіз та характеристика сервісів хмарних платформ.

Починаючи з минулого століття, можна помітити, значний розвиток інформаційних технологій. Багато сучасних підприємств, для того, аби займати провідні місця на ринку, використовують сучасні технології. Оскільки вони дозволяють зменшити витрати на виробництво, оптимізувати багато процесів та задовольняти потреби клієнтів.

Кожний бізнес, у своїй структурі, має багато бізнес процесів. Для правильного та ефективного функціонування підприємства дані процеси оптимізують. Аби досягти найбільш точного результату, використовують сучасні технології. Багато великих, середніх та маленьких підприємств, які постійно вдосконалюються. Наприклад, ІТ компанії «EPAM», «SoftServe», «GlobalLogic», «NIX», «DataArt» мають зберігати, оброблювати та аналізувати інформацію не тільки з одного міста, а й з усього світу.

Існує багато мультинаціональних компаній з філіалами в різних країнах світу. Як наслідок, великим корпораціям, які займають значну долю ринку потрібно зберігати та мати доступ до необмеженої кількості інформації. Середнім та маленьким компаніям, в яких не настільки великі об'єми інформації, також мають потреби, у використанні сучасних технологій та хмарних платформ, оскільки компанії розвиваються та є учасниками ринку.

Одним підприємствам потрібно зберігати терабайти інформації, іншим декілька гігабайтів. Дані об'єми інформації потребують певних характеристик апаратного забезпечення. Зростає попит на використання сучасних технологій для зменшення витрат. У сьогоднішніх умовах ринку, вирішення даного питання не є проблемою, оскільки існують технології, які забезпечують зберігання інформації на хмарі. Таким чином поширюється попит на використання хмарних платформ. Оскільки доступ до платформи надається з будь-якого куточку світу знаючи параметри для входу в обліковий запис. Таким чином вирішується багато питань, які задовольняють потреби клієнтів та оптимізують витрати підприємства.

Хмарні технології стали новим транспортним засобом для зберігання та використання ресурсів. Замість того, щоб бути новою технологією самі по собі хмарна платформа - це нова бізнес-модель, обгорнута навколо нових технологій, такі як віртуалізація серверів перевага економії від масштабу та багаторазової

експлуатації для зменшення витрат на використання ресурсів інформаційних технологій. Швидкий розвиток, особливо в галузі інформаційних технологій та зростання хмарних обчислень різних типів широко дозволяє обмінюватися даними в більш зручний спосіб для потреб клієнтів у різних галузях, в якому використовуються однакові методи роботи.

Замість того щоб мати власну обчислювальну інфраструктуру або центри обробки даних, компанії можуть орендувати доступ до чого завгодно, від додатків до сховища, у постачальника хмарних послуг.

Однією з переваг використання послуг хмарних обчислень є те, що компанії можуть уникнути початкових витрат і складності, пов'язаних з володінням і обслуговуванням своєї власної ІТ-інфраструктури, і замість цього просто платити за те, що і коли вони використовують. У свою чергу, постачальники послуг хмарних обчислень можуть отримати вигоду рахунок масштабу, надаючи одні й ті ж послуги широкому колу клієнтів.

Хмарні обчислення можна розбити на три моделі хмарних обчислень. Інфраструктура як послуга (IaaS) відноситься до фундаментальних будівельним блокам обчислень, які можна орендувати: фізичним або віртуальним серверам, сховищам і мережних технологій. Це привабливо для компаній, які хочуть створювати додатки з самого початку і хочуть самі контролювати практично всі елементи. Дослідження, проведене Oracle, показало, що дві третини користувачів IaaS заявили, що використання онлайн-інфраструктури полегшує інновації, скорочує їх час на розгортання нових додатків і послуг і значно скорочує поточні витрати на обслуговування. Проте, половина вважає, що IaaS недостатньо безпечний для більшості важливих даних.

Платформа як послуга (PaaS) - це наступний рівень, а також базові сховища, мережі та віртуальні сервери, які також будуть включати інструменти та програмне забезпечення, які необхідні розробникам для створення додатків поверх: які можуть включати проміжне програмне забезпечення, управління базами даних, операційні системи і засоби розробки.

Програмне забезпечення як послуга (SaaS) - це поставка додатків як послуга, ймовірно, версія хмарних обчислень, до якої звикла більшість людей в повсякденному житті. Базове устаткування і операційна система не мають відношення до кінцевого користувача, який буде отримувати доступ до сервісу через веб-браузер або додаток.

Публічне хмара - це класична модель хмарних обчислень, де користувачі можуть отримати доступ до великого обсягу обчислювальної потужності через Інтернет (будь то IaaS, PaaS або SaaS). Одним з істотних переваг тут є можливість

швидкого масштабування послуги. Постачальники хмарних обчислень володіють величезними обчислювальними потужностями. Їх величезний масштаб означає, достатність резервних потужностей, з якими вони можуть легко впоратися, якщо конкретному клієнтові потрібно більше ресурсів, тому його часто використовують для менш чутливих програм, які вимагають різної кількості ресурсів.

Приватна хмара дозволяє організаціям скористатися деякими перевагами загальнодоступної хмари, без турбот про передачу контролю над даними і послугами, оскільки воно приховано за корпоративним брендмауером. Компанії можуть точно контролювати, де зберігаються їхні дані, і можуть будувати інфраструктуру так, як вони хочуть - в основному для проектів IaaS або PaaS - щоб надати розробникам доступ до пулу обчислювальної потужності, який масштабується на вимогу, не піддаючи ризику безпеку, тому така додаткова безпека обходиться дорого. Лише деякі компанії мають масштабування, наприклад компанія Microsoft. Якщо компаніям потрібна додаткова безпека, приватна хмара може бути корисною.

Гібридне хмара зберігає деякі дані в публічній хмарі, деякі проекти в приватній. Згідно з дослідженням TechRepublic, основними причинами вибору гібридного хмари є планування аварійного відновлення і бажання уникнути витрат на обладнання при розширенні існуючого центру обробки даних.

Хмара допомагає підприємствам мати динамічно масштабовану обстраговану обчислювальну інфраструктуру, доступну за запитом і на основі плати за використання. Ця модель не тільки рятує ІТ-команди від значних інвестицій в інфраструктурі, а й захищає їх від складнощів, пов'язаних з налаштуванням і управлінням інфраструктурою.

Аналізуючи попит, який в подальшому буде тільки зростати. Найбільш підходящою для мого дослідження стала платформа Microsoft Azure SQL Databases [1]. В даному сервісі є можливість зберігання та обробки реляційних даних, а також виконання запитів до бази даних, створення звітів, для подальшого аналізу доцільності використання певного рівня та для оптимізації витрат на зберігання інформації.

Дана хмарна платформа Microsoft Azure [1] надає право вибору рівнів та характеристик в залежності від потреб клієнтів. Для того щоб не витратити зайві кошти, у клієнтів сервісу є можливість платити тільки за ті гігабайти інформації, що зберігаються на хмарі, а також за час роботи проведеної в Azure. Варто зазначити, що на хмарній платформі Microsoft Azure [1] є пробний період для ознайомлення із самою платформою. Після реєстрації на платформі, Azure надає

ліміт в двісті віртуальних доларів на місяць, раніше надавалось сто. Цими віртуальними грошами можна користуватися протягом 30 днів, також є можливість перейти на інший тип підписки до завершення даного періоду та віртуальні гроші перенесуться із безкоштовної підписки на витрати по мірі використання.

У користувача є можливість протестувати дану платформу з малими витратами, якщо не вистачить безкоштовно наданих можливостей. Після пробного періоду користувач вирішує чи потрібно продовжувати користуватися даною платформою. Ще однією перевагою служби є те, що відбувається автоматичне резервне копіювання даних. Дана функція дозволяє клієнтам не сумніватися у надійності зберігання даних. Інші переваги хмарної платформи:

- захищеність даних;
- можливість спільної роботи;
- мінімізація обслуговування технічних засобів;
- гнучкість і масштабованість інфраструктури;
- безперервний доступ до даних з будь-якої точки світу;
- зниження витрат на ІТ-ресурси.

Так само, користувачам даної платформи, не потрібно укладати жодних договорів, як наслідок не потрібно наймати юриста, що також економить бюджет компанії. Проте, потрібно тільки надати згоду на умови користування даним сервісом при реєстрації. Це стандартна процедура, яка регламентується законодавством для усіх типів послуг.

Варто зазначити, що в Microsoft Azure [1] існує багато рівнів для зберігання інформації, та не кожен користувач може одразу зрозуміти, який саме йому потрібен. Проте користувач завжди може ознайомитися із офіційною документацією в мережі, та з легкістю розпочати користуватися хмарною платформою. Таким чином і відбувалося проведення експериментів, а саме міграції баз даних та виконання запитів на платформі Microsoft Azure [1].

Microsoft SQL Database [1] надає можливість підтримки гібридної служби Stretch Database, яка виконує перенесення даних з локальної машини на хмарний сервер. Використання хмарних технологій - є актуальною темою, оскільки дані зберігаються на хмарі, та в користувача є можливість в будь-який час мати доступ до своїх даних. Варто зазначити що дані транзакції мають високий рівень захисту. Хмарні платформи можуть знизити кількість використання програмних продуктів, а отже, і витрати самого підприємства.

У службі додатків Azure є все необхідне для створення веб-сайтів, серверів для мобільних додатків і веб-API для будь-якої платформи або пристрою. У

планах "Безкоштовний" і "Загальний" (попередня версія) надаються різні варіанти тестування додатків в межах бюджету. Плани "Базовий", "Стандартний" і "Преміум" призначені для виробничих навантажень і виконуються на виділених примірниках віртуальних машин. Кожен екземпляр підтримує безліч додатків і доменів. В рамках плану "Ізольований" додатка розміщуються в приватній виділеній середовищі Azure. Це ідеальний варіант, якщо потрібно безпечне підключення до локальної мережі, підвищена продуктивність і додаткові можливості масштабування. Плата за план служби додатків нараховується посекундно.

Плани обслуговування "Безкоштовний" і "Загальний" (попередня версія) - це плани базового рівня, які виконуються на тих же віртуальних машинах Azure, що і інші додатки. Деякі додатки можуть належати іншим клієнтам. Ці рівні призначені тільки для розробки і тестування. Для планів обслуговування "Безкоштовний" і "Загальний" угоду про рівень обслуговування не надається. Тарифікація за планами обслуговування "Безкоштовний" і "Загальний" проводиться на основі кожної програми.

Ціни для плану Shared (попередня версія) включають знижку попередньою версією.

План обслуговування "Базовий" призначений для додатків з більш низькими вимогами до трафіку, коли немає необхідності в розширених можливостях автомасштабування і управління трафіком. Вартість залежить від розміру і кількості працюючих екземплярів. Завдяки вбудованій підтримці балансування мережного навантаження трафік автоматично розподіляється між екземплярами. План обслуговування "Базовий" із середовищами виконання Linux підтримує функцію Веб-додаток для контейнерів.

План обслуговування "Стандартний" призначений для виконання робочих навантажень. Вартість залежить від розміру і кількості працюючих екземплярів. Завдяки вбудованій підтримці балансування мережного навантаження трафік автоматично розподіляється між екземплярами. План "Стандартний" включає функцію автоматичного масштабування, яка може автоматично регулювати число працюючих екземплярів віртуальних машин відповідно до вимог трафіку. План обслуговування "Стандартний" із середовищами виконання Linux підтримує функцію Веб-додаток для контейнерів.

План обслуговування "Преміум" призначений для підвищення продуктивності для робочих додатків. В оновленому плані "Преміум", "Преміум V2" реалізовані віртуальні машини серії Dv2 з більш швидкими процесорами, сховищами на базі SSD і подвійним співвідношенням ресурсів пам'яті і ядра в

порівнянні з планом "Стандартний". Новий план "Преміум" також відрізняється більш високою масштабованістю завдяки збільшеному числу екземплярів, надаючи при цьому всі розширені функції, доступні в рамках плану "Стандартний". Наявні клієнти все ще можуть скористатися першим поколінням плану "Преміум" для задоволення потреб в масштабуванні.

План служб контейнерів категорії "Преміум" (Windows) призначений для підвищення продуктивності тільки робочих додатків, розгорнутих в якості контейнерів Windows. У плані контейнерів категорії "Преміум" реалізовані віртуальні машини серії Dv3 з більш швидкими процесорами, сховищами на базі SSD і більш оптимальним співвідношенням ресурсів пам'яті і ядра в порівнянні з планом категорії "Стандартний". Новий план "Преміум" також відрізняється більш високою масштабованістю завдяки збільшеному числу екземплярів, надаючи при цьому всі розширені функції, доступні в рамках плану "Стандартний".

План обслуговування "Ізольований" призначений для виконання критично важливих робочих навантажень, необхідних для роботи у віртуальній мережі. Цей план дозволяє користувачам працювати з додатками в спеціалізованій закритому середовищі центру обробки даних Azure за допомогою віртуальних машин серії Dv2 з більш швидкими процесорами, сховищем на основі SSD і подвоєним співвідношенням ресурсів пам'яті і ядра в порівнянні з планом "Стандартний". Закрите середовище, використовувана з планом "Ізольований", називається середовищем служби додатків. План підтримує масштабування до 100 примірників (щонайбільше на вимогу). Крім тарифікації за кожен екземпляр плану "Ізольований", також стягується фіксована плата за кожен екземпляр середовища служби додатків.

Модель DTU пропонує попередньо сконфігурованих і заздалегідь визначений обсяг обчислювальних ресурсів.

1.2 Аналіз моделей придбання хмарної платформи Azure

Модель закупівель на основі віртуального ядра vCore [26]. Ця модель придбання забезпечує вибір між наданим обчислювальним рівнем і безсерверним обчислювальним рівнем. За допомогою наданого рівня обчислень ви вибираєте точну кількість обчислювальних ресурсів, які завжди надаються для вашого робочого навантаження. З обчислювальним рівнем без сервера ви вказуєте автомасштабування обчислювальних ресурсів у настроюваному діапазоні обчислень. За допомогою цього обчислювального рівня ви також

можете автоматично призупиняти та відновлювати базу даних на основі робочого навантаження. Ціна одиниці vCore за одиницю часу нижча на наданому обчислювальному рівні, ніж на безсерверному обчислювальному рівні.

vCore - це незалежна масштабованість, в якій можна розглянути конкретну область, таку як кількість ядер ЦП і ресурси пам'яті, то, що не має можливості контролювати на тому ж рівні деталізації при використанні моделі на основі DTU. vCore є більш простою та зрозумілою для користувача Microsoft Azure. Тобто її можна використовувати для більш широкого спектру додатків баз даних з гнучкістю, контролем і прозорістю, що забезпечуються конфігураціями на основі vCore.

Термін vCore відноситься до віртуального ядра. У цій моделі придбання бази даних SQL Azure ви можете вибрати один із наданого рівня обчислень і рівня обчислень без сервера, а саме:

Забезпечений обчислювальний рівень: є можливість обрати точні обчислювальні ресурси для робочого навантаження.

Рівень обчислень без сервера: Azure автоматично призупиняє та відновлює базу даних на основі робочого навантаження на безсерверному рівні. Під час періоду паузи Azure не стягує плату за обчислювальні ресурси.

Загальна мета — стандартний рівень обслуговування для загального робочого навантаження. Він пропонує 99,99% SLA та затримку зберігання від 5 до 10 мілісекунд.

Рівень обслуговування загального призначення має два підрівня.

Рівень обчислень без стану, який запускає процес sqlserver.exe. Він містить перехідні дані та дані кешу, такі як кеш плану, пул буферів, пул сховища стовпців. Azure Service Fabric керує цим рівнем із збереженням стану та включає в себе такі завдання, як ініціалізація, контроль працездатності обчислювальних вузлів та перехід на збій, коли це потрібно.

Рівень обчислень із збереженням стану зберігає файли бази даних (MDF, LDF) у сховищі BLOB-об'єктів Azure. Сховище Azure має вбудовані функції для доступності та резервування даних. Він підтримує базу даних SQL від 5 ГБ до 4 ТБ у наданих і від 5 ГБ до 3 ТБ у безсерверних обчисленнях.

Критично важливий рівень для бізнесу

Критично важливий для бізнесу рівень підходить для критичних додатків, які вимагають найвищої стійкості та найвищої продуктивності введення-виведення бази даних на репліку. Він пропонує 99,995% доступності в конфігурації Multi-AZ і низьку затримку зберігання в 1-2 мілісекунди.

Рівень послуг, які важливі для бізнесу (рис. 1.1), також пропонує гарантований RPO 5 секунд і RTO 30 секунд за допомогою геореплікації. Крім того, можна використовувати сховище від 5 ГБ до 4 ТБ на критично важливому рівні для бізнесу.

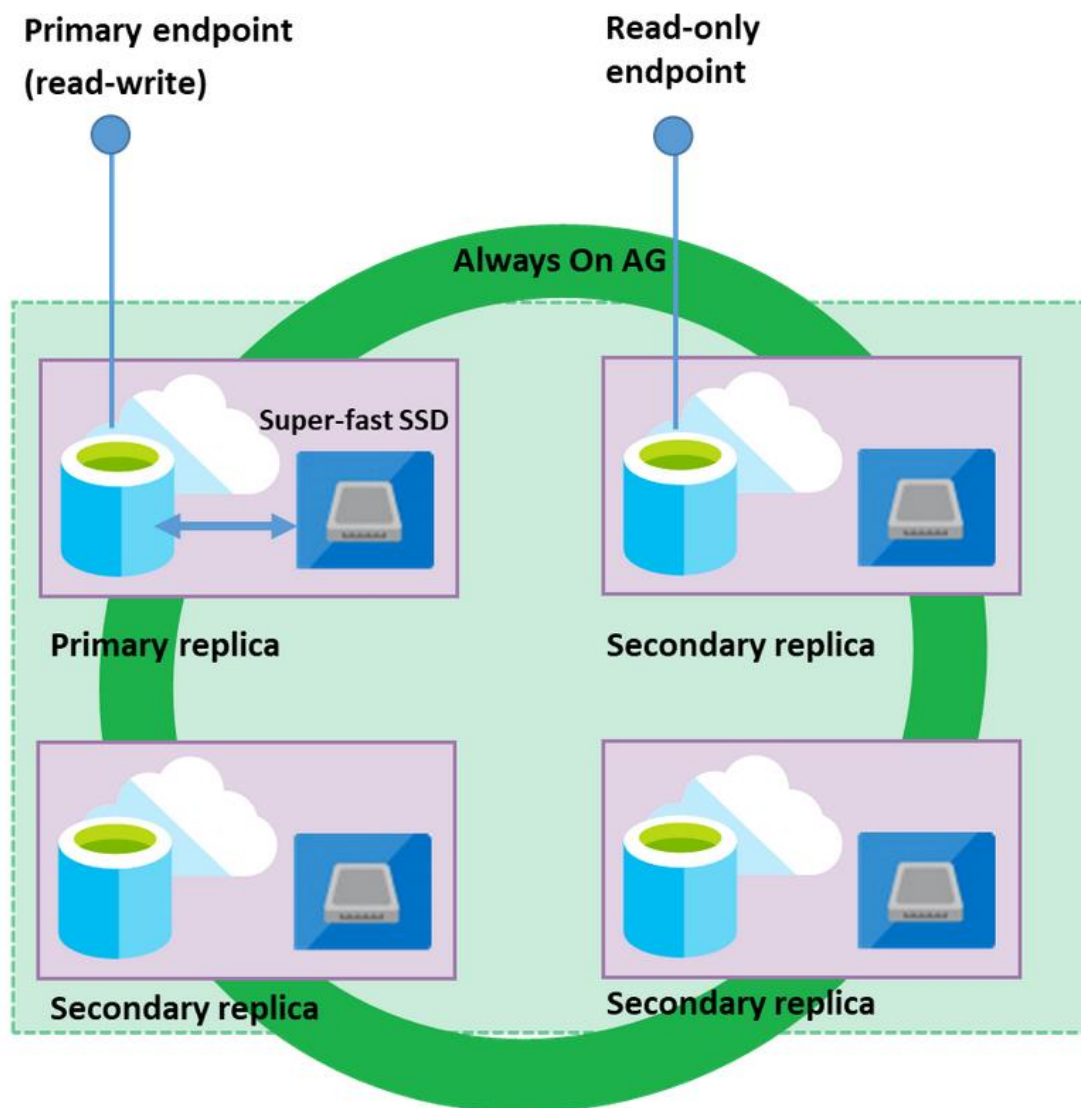


Рис. 1.1. Модель критично важливого рівня для бізнесу

У цій моделі як процес SQL Server (sqlserver.exe), так і базові файли бази даних (MDF, LDF) існують на одному вузлі. Він використовує локально підключений надшвидкий SSD для дуже низької затримки робочого навантаження бази даних. Він також використовує Always On Availability Groups для підтримки кластера вузлів бази даних. Три вторинні репліки отримують дані від первинного вузла. У разі відмови основного вузла вторинна репліка бере на себе роль первинної репліки і починає відповідати на запити бази даних. Основна кінцева точка завжди вказує на поточну активну репліку.

Рівень обслуговування критичного для бізнесу також використовує функцію масштабування читання для розвантаження робочих навантажень (рис. 1.2) лише для читання до однієї з вторинних реплік. Це може підвищити продуктивність основної бази даних, оскільки ви можете розвантажити свою роботу, наприклад звітувати про запити до вторинної. Ця функція доступна в критично важливому для бізнесу рівні послуг без будь-яких додаткових витрат.

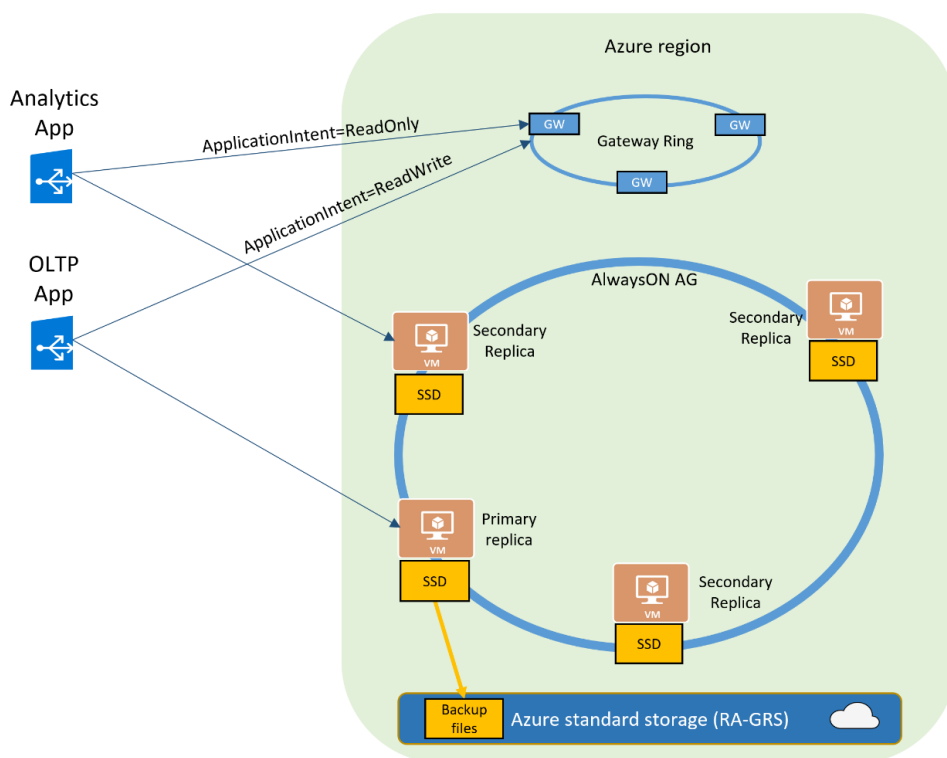


Рис. 1.2. Модель масштабування читання для розвантаження робочих навантажень

Гіпермасштабування

Рівень служби HiperScale підходить для високомасштабованих вимог до зберігання. Він пропонує високу продуктивність і масштабованість з автоматичним масштабуванням до 100 ТБ для бази даних SQL Azure.

База даних HiperScale (рис. 1.3) містить такі компоненти:

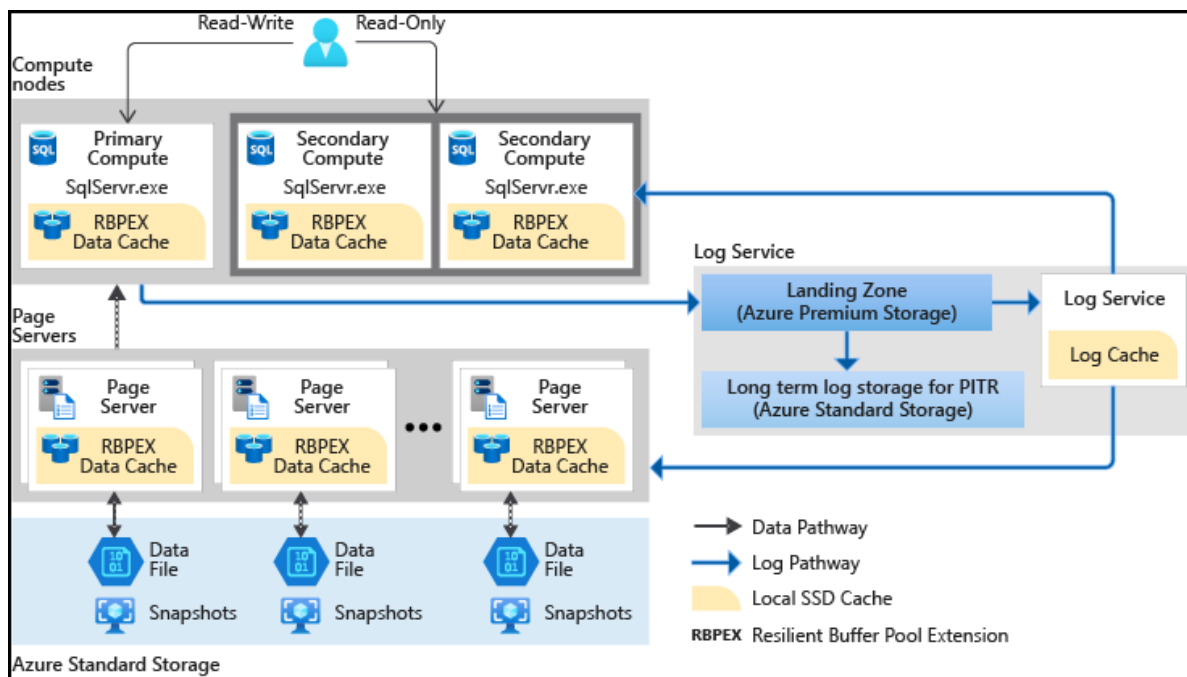


Рис 1.3. Компоненти бази даних Hyperscale

Обчислювальний вузол: обчислювальний вузол використовує кешування на основі SSD, яке називається Resilient Buffer Pool Extension (RBPEX). Він забезпечує мінімальні маршрути мережі для отримання сторінки даних. Основний обчислювальний вузол обслуговує всі робочі навантаження читання-запису. Додаткові вузли працюють як вузли гарячого резервування, і ви можете розвантажити робочі навантаження читання, якщо потрібно

Сервер сторінок: сервери сторінок відповідають за підмножину сторінок у гіпермасштабованій базі даних. Кожен сервер сторінки контролює до 1 ТБ даних. Завдання сервера сторінок полягає в тому, щоб обслуговувати сторінки бази даних обчислювальному вузлу, коли це потрібно. Він також оновлює сторінки, оскільки транзакція оновлює свої дані. Він використовує службу журналів, щоб сторінки оновлювалися. Як показано на діаграмі архітектури, ці сервери сторінок також використовують кеш даних RBPEX для швидкої реакції

Служба журналів: служба журналів отримує записи журналу первинної репліки обчислень, зберігає їх у довготривалому кеші та передає до вторинних реплік. Він також надсилає ці журнали на відповідний сервер сторінок. Тому всі вторинні репліки обчислень і відповідні сервери сторінок отримують зміни даних. Він також переміщує записи журналу в стандартне сховище Azure для довготривалого зберігання. Таким чином, виключається необхідність частого обрізання журналу

Сховище Azure: сховище Azure зберігає всі файли даних гіпермасштабованої бази даних. Сервери сторінок оновлюють ці файли даних і

оновлюють їх. Крім того, Azure використовує моментальний знімок файлів даних для резервного копіювання та відновлення

Ключовою відмінністю між стандартним і преміальним рівнями, про який варто згадати, є продуктивність введення-виведення. Якщо робоча навантаження вимагає інтенсивного введення-виведення, тоді, швидше за все, потрібно застосовувати рівень Premium, де він пропонує більшу кількість операцій введення-виведення і меншу затримку. Рівень Premium швидше виконає міграції та операції запитів, економить час користувачів. Різні рівні потрібні для різних типів рішення бізнес задач.

Якщо було допущено помилку в конфігурації DTU, то змінити рівень обслуговування та / або рівень продуктивності бази даних досить просто. Під покровом Microsoft створює репліку вихідної бази даних на новому рівні продуктивності, а потім перемикає з'єднання на репліку. Під час цього процесу дані не втрачаються, але протягом короткого часу, коли відбувається перемикаєння на репліку, з'єднання буде розірвано. Природно, час для цього залежить від розміру вашої бази даних і від того, наскільки зайнята система, коли був ініційований запит на масштабування.

Модель на основі vCore є новою і пропонує зовсім інший підхід до визначення розміру вашої бази даних. Легше перетворити локальні робочі навантаження в модель на основі vCore, тому що компоненти - це те, що теж використовується.

Існує можливість вибрати один з двох рівнів: «Універсальний» і «Бізнес-критичний». Найбільша відмінність між двома рівнями полягає в можливості зберігання (з точки зору типу і пропускну здатності) і доступності. При критичному підході до бізнесу може бути три репліки, можливість масштабування і резервування зони. Універсальний рівень є більш обмеженим.

1.3 Характеристики моделі придбання DTU

DTU являє собою змішану міру ЦП, пам'яті, операцій зчитування і запису [25]. Основна ідея цієї моделі полягає в її простоті використання попередньо сконфігурованого набору обчислювальних ресурсів за фіксованою ціною з легко масштабованими параметрами.

Усередині цих рівнів є рівні продуктивності. Чим вище рівень продуктивності, тим вище число DTU, призначених даному рівню. Число, присвоєне рівнем продуктивності DTU, є відносним, наприклад, база даних Premium P11 з 1750 DTU забезпечує в 175 разів більшу обчислювальну

потужність DTU, ніж база даних Standard S0 з 10 DTU. Проте за швидшу роботу та більші об'єми ми повинні більше платити.

Отже, якщо в даний час використовується модель на основі DTU, а моніторинг продуктивності показує, що у вас немає проблем з продуктивністю, то краще зупинитися на моделі DTU. Якщо налаштовується нове робоче навантаження для переміщення в базу даних SQL Azure, тоді необхідно вирішити, який маршрут обрати. Якщо переважає плата фіксованої суми щомісяця за попередньо сконфігуровані обчислювальні ресурси, тоді може стати в нагоді модель на основі DTU. Однак, якщо потрібно заглибитися в базові ресурси і масштабувати їх незалежно для досягнення оптимальної продуктивності, тоді vCore може бути найкращим варіантом, особливо якщо активувати гібридну перевагу Azure для SQL Server.

Недоліком хмарних платформ, інколи може стати те, що інформація зберігається у провайдера, тобто не у вашій організації. Отже стовідсоткової гарантії захисту ваших даних ніхто не може надати. Незважаючи на це, витік інформації може бути і у вас, оскільки можуть здійснюватися хакерські атаки. Проте використання хмарних технологій є більш надійним, оскільки компанія розроблює декілька рівнів захисту інформації на платформі. Також є можливість зробити резервне копіювання даних на хмарі.

Варто уважно поставитися до вибору провайдера, для уникнення витоку інформації. При виборі провайдера слід вивчити відгуки та статистику за наданими ресурсами. Послуги надійних провайдерів можуть бути дорожчими, проте рівень захисту ваших даних буде більш надійним.

Розглядаючи технологію Microsoft Azure SQL Databases [1], слід звернути увагу на те, що в користувача є вибір поміж трьох рівнів: базовий, стандартний і преміум. Дані рівні призначені для різних обсягів зберігання даних.

Базовий рівень підходить для компаній з маленькою активністю, тобто малим трафіком відвідувачів та з малою кількістю запитів на сервер.

Стандартний рівень підтримує як базовий рівень, так і може працювати на середній потужності. Стандартний рівень здатний вирішити більш складні задачі. В даному випадку, можна виконувати більшу кількість запитів, порівняно із базовим, але при цьому, час очікування при виконанні запитів не збільшується.

Преміум рівень використовують для роботи з високопродуктивними потоками введення-виведення, може паралельно виконувати багато запитів. Преміум рівень призначений для вирішення великих задач, найчастіше такий рівень використовують великі корпорації, які мають цілу мережу філіалів в

країні, чи по всьому світу. Таким компаніям потрібно на серверах зберігати та обробляти терабайти даних.

Для того щоб зрозуміти який об'єм пам'яті краще зберігати на якому рівні, далі приведено більш точні об'єми для зберігання реляційних баз даних. Для базового рівня включений та максимальний розмір бази даних становить 2 ГБ. Стандартний рівень на перших трьох рівнях S0-S2 має також однаковий включений та максимальний розмір 250 ГБ, починаючи від рівня S3 до рівня S12, включений такий самий, а максимальний 1 ТБ. Для рівня Premium: P1,P2,P4,P6 – включений 500 ГБ, максимальний – 1 ТБ, P11,P15 – 4 ТБ [28].

Також слід звернути увагу, на те, що в Microsoft Azure [1] існує одиниця DTU. Дана одиниця призначена для вимірювання CPU, пам'яті, введення-виведення даних і введення-виведення журналу транзакцій. При збільшенні DTU з'являється більше обчислювальних потужностей. Отже при виборі рівня, слід звертати увагу не тільки на розміри баз даних, а також і на інші характеристики [29], які приведено у таблиці 1.3.

Таблиця 1.1

Ліміти ресурсів для рівнів моделі DTU

| Рівень | | DTU | Включений обсяг сховища, ГБ | Макс. обсяг сховища, ГБ | Макс. об'єм OLTP в пам'яті, ГБ | Макс. кількість запитів, які виконуються одночасно, сек. | Макс. кількість одночасних сеансів |
|-------------|-----|------|-----------------------------|-------------------------|--------------------------------|----------------------------------------------------------|------------------------------------|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| Базовий | B | 5 | 2 | 2 | N/A | 30 | 300 |
| Стандартний | S0 | 10 | 250 | 250 | N/A | 60 | 600 |
| | S1 | 20 | 250 | 250 | N/A | 90 | 900 |
| | S2 | 50 | 250 | 250 | N/A | 120 | 1200 |
| | S3 | 100 | 250 | 250, 500, 750, 1024 | N/A | 200 | 2400 |
| | S4 | 200 | 250 | 250, 500, 750, 1024 | N/A | 400 | 4800 |
| | S6 | 400 | 250 | 250, 500, 750, 1024 | N/A | 800 | 9600 |
| | S7 | 800 | 250 | 250, 500, 750, 1024 | N/A | 1600 | 19200 |
| | S9 | 1600 | 250 | 250, 500, 750, 1024 | N/A | 3200 | 30000 |
| | S12 | 3000 | 250 | 250, 500, 750, 1024 | N/A | 6000 | 30000 |
| Преміум | P1 | 125 | 500 | 500, 750, 1024 | 1 | 200 | 30000 |
| | P2 | 250 | 500 | 500, 750, 1024 | 2 | 400 | 30000 |
| | P4 | 500 | 500 | 500, 750, | 4 | 800 | 30000 |

| | | | | | | | |
|--|-----|------|------------------------------------------------------------------------------|------------------------------------------------------------------------------|----|------|-------|
| | | | | 1024 | | | |
| | P6 | 1000 | 500 | 500, 750, 1024 | 8 | 1600 | 30000 |
| | P11 | 1750 | 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 2584, 3840, 4096 | 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 2584, 3840, 4096 | 14 | 2400 | 30000 |
| | P15 | 4000 | 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 2584, 3840, 4096 | 1024, 1280, 1536, 1792, 2048, 2304, 2560, 2816, 3072, 3328, 2584, 3840, 4096 | 32 | 6400 | 30000 |

База даних Azure SQL і керований екземпляр Azure SQL дозволяють легко придбати повністю керовану платформу як службу бази даних (PaaS), яка відповідає вашим потребам у продуктивності та витратах. Залежно від моделі розгортання, яку ви вибрали для бази даних SQL Azure, ви можете вибрати модель закупівлі, яка вам підходить:

Різні типи сховища оплачуються по-різному. За зберігання даних з вас стягується плата за надане сховище на основі вибраного максимального розміру бази даних або пулу. Вартість не зміниться, якщо ви не зменшите або не збільшите цей максимум. Сховище резервних копій пов'язане з автоматичним резервним копіюванням вашого екземпляра і виділяється динамічно. Збільшення періоду зберігання резервних копій збільшує обсяг резервної копії, яку використовує ваш екземпляр.

За замовчуванням сім днів автоматичного резервного копіювання ваших баз даних копіюються до стандартного облікового запису зберігання BLOB-об'єктів із доступом до читання (RA-GRS). Це сховище використовується для щотижневих повних резервних копій, щоденних диференціальних резервних

копій і резервних копій журналу транзакцій, які копіюються кожні п'ять хвилин. Розмір журналів транзакцій залежить від швидкості зміни бази даних. Мінімальний обсяг зберігання, що дорівнює 100% розміру бази даних, надається без додаткової плати. Додаткове споживання резервного сховища стягується в ГБ на місяць.

Якщо база даних споживає більше 300 DTU, перетворення на модель закупівель на основі vCore може зменшити витрати. Існує можливість конвертації використовуючи вибраний API або за допомогою порталу Azure, без простою. Однак перетворення не потрібне і не виконується автоматично. Якщо модель придбання на основі DTU відповідає вимогам продуктивності та бізнесу, слід продовжувати її використовувати.

Одиниця транзакцій бази даних (DTU) являє собою змішану міру ЦП, пам'яті, читання та запису. Модель закупівель на основі DTU пропонує набір попередньо налаштованих пакетів обчислювальних ресурсів і вбудованого сховища для забезпечення різних рівнів продуктивності програми. Якщо надається перевага простоті попередньо налаштованого пакету та фіксованих платежів щомісяця, модель на основі DTU може бути більш задовільною.

У моделі закупівель на основі DTU є вибір між базовим, стандартним і преміальним рівнем обслуговування для бази даних SQL Azure. Модель закупівлі на основі DTU недоступна для керованого екземпляра Azure SQL.

Одиниці транзакцій бази даних (DTU)

Для однієї бази даних із певним розміром обчислень на рівні служби Azure гарантує певний рівень ресурсів для цієї бази даних (незалежно від будь-якої іншої бази даних у хмарі Azure). Ця гарантія забезпечує передбачуваний рівень продуктивності. Обсяг ресурсів, виділених для бази даних, розраховується як кількість DTU і є сукупною мірою обчислень, ресурсів зберігання та ресурсів введення-виводу.

Співвідношення між цими ресурсами спочатку визначається робочим навантаженням для тестування онлайн-обробки транзакцій (OLTP), розробленим таким чином, щоб бути типовим для реальних робочих навантажень OLTP. Коли ваше робоче навантаження перевищує кількість будь-якого з цих ресурсів, ваша пропускна здатність зменшується, що призводить до уповільнення продуктивності та часу очікування.

Ресурси, які використовуються вашим робочим навантаженням, не впливають на ресурси, доступні для інших баз даних у хмарі Azure. Аналогічно, ресурси, які використовуються іншими робочими навантаженнями, не впливають на ресурси, доступні для бази даних.

DTU найбільш корисні для розуміння відносних ресурсів, які виділяються для баз даних з різними розмірами обчислень і рівнями обслуговування. Наприклад:

Подвоєння DTU за рахунок збільшення обчислювального розміру бази даних дорівнює подвоєнню набору ресурсів, доступних для цієї бази даних.

База даних P11 рівня обслуговування преміум-класу з 1750 DTU забезпечує в 350 разів більше обчислювальної потужності DTU, ніж база даних базового рівня обслуговування з 5 DTU.

Визначення кількості DTU, необхідних для робочого навантаження.

Якщо існує потреба перенести існуюче локальне робоче навантаження або робоче навантаження віртуальної машини SQL Server у базу даних SQL, існує можливість використання калькулятора DTU [30], щоб приблизно визначити кількість необхідних DTU. Для існуючого робочого навантаження бази даних SQL можна проаналізувати статистику продуктивності запитів, щоб зрозуміти споживання ресурсів бази даних (DTU) і отримати більш глибокі уявлення про оптимізацію робочого навантаження. Sys.dm_db_resource_stats вид управління динамічним (DMV) дозволяє переглядати споживання ресурсів в протягом останньої години. У представленні каталогу sys.resource_stats відображається споживання ресурсів за останні 14 днів, але з нижчою точністю – середні значення за п'ять хвилин.

Визначення використання DTU

Щоб визначити середній відсоток використання DTU/eDTU відносно межі DTU/eDTU бази даних або еластичного пулу, існує така формула:

$$\text{avg_dtu_percent} = \text{MAX}(\text{avg_cpu_percent}, \text{avg_data_io_percent}, \text{avg_log_write_percent})$$

Вхідні значення для цієї формули можна отримати з sys.dm_db_resource_stats, sys.resource_stats і sys.elastic_pool_resource_stats DMV. Щоб визначити відсоток використання DTU/eDTU до межі DTU/eDTU бази даних або еластичного пулу, потрібно обрати найбільше відсоткове значення з наступного:

avg_cpu_percent, avg_data_io_percent, і avg_log_write_percent в певний момент часу.

Покоління обладнання в моделі закупівель на основі DTU

У моделі закупівель на основі DTU клієнти не можуть вибрати генерацію обладнання, яка використовується для їх баз даних. Хоча певна база даних зазвичай залишається на певному поколінні обладнання протягом тривалого часу

(зазвичай протягом кількох місяців), існують певні події, які можуть спричинити переміщення бази даних до іншого покоління обладнання.

Наприклад, базу даних можна перемістити до іншого покоління апаратного забезпечення, якщо вона збільшується або зменшується до іншої мети обслуговування, або якщо поточна інфраструктура в центрі обробки даних наближається до своїх лімітів ємності, або якщо обладнання, що використовується в даний час, виводиться з експлуатації через його кінець життя.

Якщо базу даних перемістити на інше обладнання, продуктивність робочого навантаження може змінитися. Модель DTU гарантує, що пропускна здатність і час відгуку робочого навантаження контрольного тесту DTU залишаться по суті ідентичними, коли база даних переходить до іншого покоління обладнання, доки її ціль обслуговування (кількість DTU) залишається незмінною.

Однак у широкому спектрі робочих навантажень клієнтів, які виконуються в базі даних SQL Azure, вплив використання різного обладнання для однієї цілі служби може бути більш помітним. Різні робочі навантаження виграють від різних конфігурацій обладнання та функцій. Тому для робочих навантажень, відмінних від тесту DTU, можна побачити відмінності в продуктивності, якщо база даних переходить від одного покоління обладнання до іншого.

Наприклад, програма, яка чутлива до мережевої затримки, може бачити кращу продуктивність на апаратному забезпеченні Gen5 порівняно з Gen4 завдяки використанню Accelerated Networking у Gen5, але програма, яка використовує інтенсивне введення-виведення читання, може бачити кращу продуктивність на апаратному забезпеченні Gen4 порівняно з Gen5 через вище співвідношення пам'яті на ядро на Gen4.

Клієнти з робочими навантаженнями, які чутливі до змін обладнання, або клієнти, які хочуть контролювати вибір генерації обладнання для своєї бази даних, можуть використовувати модель vCore, щоб вибрати бажане генерування обладнання під час створення та масштабування бази даних. У моделі vCore задокументовані обмеження ресурсів кожної цілі служби для кожного покоління обладнання як для окремих баз даних, так і для еластичних пулів.

1.4 Засоби генерації великих даних

Microsoft SQL Server Management Studio [2] є безкоштовним та найбільш вживаним інструментом серед користувачів баз даних. Даний інструмент дозволяє створити базу даних на локальному ресурсі, створити таблиці до бази даних, можна заповнити таблиці даними та виконати запити. Підключення можна зробити як до локального ресурсу, так і на саму хмарну платформу, за умови що створений сервер на хмарі. Коли генерується великі об'єми БД, то може виникнути помилка яка пов'язана із нестачею пам'яті у самому інструменті, незважаючи на те, що пам'яті на жорсткому диску вистачає.

Проте, є аналог dbForge Studio 2020 Express for SQL Server [3]. За допомогою якого можна виконати SQL запит. Таким чином згенеровані дані будуть зберігатися на локальному ресурсі. Інструмент dbForge Studio 2020 Express for SQL Server [3] є безкоштовним та має Express версію, тому не всі функції доступні, наприклад, записати можна тільки по 50 рядків. Якщо придбати повну версію, тоді це питання вирішиться та можна одразу перенести усю згенеровану базу даних. Проте в даній кампанії можна знайти цю програму, але в попередній версії. Там не має обмежень щодо кількості строк. Таким чином можна згенерувати таблиці на мільйон записів.

Іншими аналогами для створення БД на локальному пристрої є: MySQL Workbench [4], HeidiSQL [5], Navicat [6], PL/SQL Developer [7], Database .NET [8], Aqua Data Studio [9] та інші. У табл. 1.1 приведено характеристики цих продуктів, визначено розробників, їх версії. Найкраще обирати таку базу, яка задовільнить конкретні цілі користувача.

Таблиця 1.2

Характеристика засобів створення та генерації баз даних

| Фірма-розробник | Назва програмного продукту | Остання версія продукту | Опис |
|-----------------|----------------------------------------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | 2 | 3 | 4 |
| Microsoft | Microsoft SQL Server Management Studio | 18.7.1 | Утиліта з Microsoft SQL Server 2005 і пізніших версій для конфігурації, управління і адміністрування всіх компонентів Microsoft SQL Server. Утиліта включає скриптова редактор і графічну програму, яка працює з об'єктами і настройками сервера |
| Devart | dbForge Studio for SQL Server | 5.8.107 | Система керування базами даних (РСУБД), розроблена корпорацією Microsoft. Основний використовуваний мову запитів - Transact-SQL, створений спільно Microsoft та Sybase. Transact- |

| | | | |
|----------------------|------------------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | | SQL є реалізацією стандарту ANSI/ISO по структурованого мови запитів (SQL) з розширеннями. Використовується для роботи з базами даних розміром від персональних до великих баз даних масштабу підприємства; конкурує з іншими СУБД в цьому сегменті ринку. |
| LGPL | SQuirreL SQL | 4.1.0 | Переносних незалежний клієнт баз даних, написаний на Java, працює з будь-якою системою управління базами даних, що підтримує JDBC 2.0. |
| Allround Automations | PL/SQL Developer | 19.2.1 | Інтегроване середовище розробки для СУБД Oracle, яку створила і поширює фірма Allround Automations. |
| Oracle Corporation | MySQL Workbench | 8.0.122 | Інструмент для візуального проектування баз даних, що інтегрує проектування, моделювання, створення та експлуатацію БД в єдине безшовне оточення для системи базданих MySQL. |

Microsoft Microsoft SQL Server Management Studio 18.7.1 утиліта з Microsoft SQL Server 2005 і пізніших версій для конфігурації, управління і адміністрування всіх компонентів Microsoft SQL Server. Утиліта включає скриптова редактор і графічну програму, яка працює з об'єктами і настройками сервера

Система керування базами даних (PCСУБД) Devart dbForge Studio for SQL Server 5.8.107, розроблена корпорацією Microsoft. Основний використовуваний мову запитів - Transact-SQL, створений спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI/ISO по структурованого мови запитів (SQL) з розширеннями. Використовується для роботи з базами даних розміром від персональних до великих баз даних масштабу підприємства; конкурує з іншими СУБД в цьому сегменті ринку.

LGPL SQuirreL SQL 4.1.0 Переносних незалежний клієнт баз даних, написаний на Java, працює з будь-якою системою управління базами даних, що підтримує JDBC 2.0.

Allround Automations PL/SQL Developer 19.2.1 Інтегроване середовище розробки для СУБД Oracle, яку створила і поширює фірма Allround Automations.

Oracle Corporation MySQL Workbench 8.0.122 Інструмент для візуального проектування баз даних, що інтегрує проектування, моделювання, створення та експлуатацію БД в єдине безшовне оточення для системи баз даних MySQL.

Ansgar Becker HeidiSQL 11.1 Вільний відкритий клієнт, або фронтенд, для управління базами даних, розроблений німецьким програмістом Анзгаром Бекером (Ansgar Becker) та кількома іншими розробниками,

Написаний на Delphi, підтримує з'єднання та роботу з MySQL, їхні форки, таких як MariaDB та Percona, а також Microsoft SQL Server, починаючи з версії 7.0. Щоб управляти базою даних з HeidiSQL, користувач має увійти на локальний або віддалений сервер MySQL з прийнятним паролем, створивши сесію. В рамках цієї сесії користувач може управляти базами даних MySQL на сервері MySQL, і від'єднатися після закінчення роботи. Можливості програми цілком достатні для більшості операцій із загальними та просунутими базами даних, таблицями та записами, але розробка залишається у активному стані, щоб забезпечити повну функціональність, котра очікується від фронтенду MySQL.

PremiumSoft CyberTech Ltd Navicat 15.0.22 Серія програм керування базами даних та розробки програмного забезпечення виробництва ТОВ PremiumSoft CyberTech для MySQL, Oracle, SQLite, PostgreSQL, MariaDB і Microsoft SQL Server. Він має Explorer-подібний графічний користувацький інтерфейс і підтримує кілька з'єднань з базою даних для локальних і віддалених баз. Дизайн програми виконаний з урахуванням потреб найрізноманітніших аудиторій: від адміністраторів баз даних і програмістів до різноманітних підприємств/компаній, які обслуговують клієнтів і обмінюються інформацією з партнерами.

Також відомо, що компанія Devart створила програмний продукт для автоматичної генерації даних dbForge Data Generator for SQL Server [3]. Ця програма також платна та має пробний період. Пробного періоду вистачить для того, щоб автоматично згенерувати дані та зберегти їх на локальному ресурсі.

Аналоги генераторів баз даних великих об'ємів: DTM Data Generator [10], Visual Studio (Premium) Data Generator [11], GEDIS Studio [12], Mockaroo [13], ApexSQL Generate [14], Upscene Advanced Data Generator [15], EMS Data Generator [16], GS Data Generator [17], Spawner Data Generator [18], Generatedata.com [19]. Усі ці інструменти є програмами, тільки Generatedata.com є сайтом, де в онлайн режимі проводиться генерація. В основному у генераторів є пробний період для безкоштовного використання [20].

Microsoft Microsoft SQL Server Management Studio 18.7.1. Утиліта з Microsoft SQL Server 2005 і пізніших версій для конфігурації, управління і адміністрування всіх компонентів Microsoft SQL Server. Утиліта включає скриптова редактор і графічну програму, яка працює з об'єктами і настройками сервера

Devart dbForge Studio for SQL Server 5.8.107 Система керування базами даних (РСУБД), розроблена корпорацією Microsoft. Основний використовуваній мову запитів - Transact-SQL, створений спільно Microsoft та

Sybase. Transact-SQL є реалізацією стандарту ANSI/ISO по структурованого мови запитів (SQL) з розширеннями. Використовується для роботи з базами даних розміром від персональних до великих баз даних масштабу підприємства; конкурує з іншими СУБД в цьому сегменті ринку.

LGPL Squirrel SQL 4.1.0 Переносний незалежний клієнт баз даних, написаний на Java, працює з будь-якою системою управління базами даних, що підтримує JDBC 2.0.

Allround Automations PL/SQL Developer 19.2.1 Інтегроване середовище розробки для СУБД Oracle, яку створила і поширює фірма Allround Automations.

Oracle Corporation MySQL Workbench 8.0.122 Інструмент для візуального проектування баз даних, що інтегрує проектування, моделювання, створення та експлуатацію БД в єдине безшовне оточення для системи баз даних MySQL.

Ansgar Becker HeidiSQL 11.1 Вільний відкритий клієнт, або фронтенд, для управління базами даних, розроблений німецьким програмістом Анзгаром Бекером (Ansgar Becker) та кількома іншими розробниками. Написаний на Delphi, підтримує з'єднання та роботу з MySQL, їхні форки, таких як MariaDB та Percona, а також Microsoft SQL Server, починаючи з версії 7.0. Щоб управляти базою даних з HeidiSQL, користувач має увійти на локальний або віддалений сервер MySQL з прийнятним паролем, створивши сесію. В рамках цієї сесії користувач може управляти базами даних MySQL на сервері MySQL, і від'єднатися після закінчення роботи. Можливості програми цілком достатні для більшості операцій із загальними та просунутими базами даних, таблицями та записами, але розробка залишається у активному стані, щоб забезпечити повну функціональність, котра очікується від фронтенду MySQL.

PremiumSoft CyberTech Ltd Navicat 15.0.22 Серія програм керування базами даних та розробки програмного забезпечення виробництва ТОВ PremiumSoft CyberTech для MySQL, Oracle, SQLite, PostgreSQL, MariaDB і Microsoft SQL Server. Він має Explorer-подібний графічний користувацький інтерфейс і підтримує кілька з'єднань з базою даних для локальних і віддалених баз. Дизайн програми виконаний з урахуванням потреб найрізноманітніших аудиторій: від адміністраторів баз даних і програмістів до різноманітних підприємств/компаній, які обслуговують клієнтів і обмінюються інформацією з партнерами.

Модель придбання на основі vCore дозволяє незалежно масштабувати обчислювальні ресурси і ресурси зберігання, зіставляти локальну

продуктивність і оптимізувати ціну. Це також дозволяє вам вибрати покоління обладнання:

Gen4 - до 24 логічних процесорів на базі процесорів Intel E5-2673 v3 (Haswell) 2,4 ГГц, vCore = 1 PP (фізичне ядро), 7 ГБ на ядро, підключений SSD

Gen5 - до 80 логічних процесорів на базі процесорів Intel E5-2673 v4 (Broadwell) 2,3 ГГц, vCore = 1 LP (гіперпоток), 5,1 ГБ на ядро, швидкий SSD eNVM

Устаткування Gen4 пропонує значно більше пам'яті на vCore. Однак апаратне забезпечення Gen5 дозволяє значно збільшити обчислювальні ресурси.

Модель vCore надає три рівня обслуговування загального призначення, Hyperscale і Business Critical. Рівні обслуговування розрізняються за діапазоном розмірів обчислень, дизайну високої доступності, ізоляції відмов, типам і розміром сховища і діапазону введення-виведення. Ви повинні окремо налаштувати необхідний термін зберігання і термін зберігання резервних копій. На порталі Azure виберіть Сервер (НЕ база даних)> Керовані резервні копії> Налаштування політики> Конфігурація відновлення на певний момент часу> 7-35 днів.

У моделі придбання на основі vCore можливо обміняти існуючі ліцензії на пільгові тарифи в базі даних SQL, використовуючи гібридне перевага Azure для SQL Server. Ця перевага Azure дозволяє вам використовувати ваші локальні ліцензії SQL Server, щоб заощадити до 30% на базі даних SQL Azure, використовуючи ваші локальні ліцензії SQL Server з Software Assurance.

За допомогою гібридної переваги Azure ви можете вибрати оплату тільки базової інфраструктури Azure з використанням існуючої ліцензії SQL Server для самого ядра бази даних SQL (BasePrice) або оплату як базової інфраструктури, так і ліцензії SQL Server (LicenseIncluded). Є можливість вибрати або змінити свою модель ліцензування, використовуючи портал Azure або один з API.

2 МОДЕЛЮВАННЯ МОДЕЛІ ПРИДБАННЯ AZURE

2.1. Генерація тестового набору даних

Для генерації тестового набору даних було розроблено додаток мовою C#, який генерує дані за вказаними параметрами, а саме, кількість даних, підключення до SQL серверу, що дозволяє генерувати дані як на локальному ресурсі так і на Microsoft Azure.

Під час генерації додаток випадковим чином створює дані для таблиць і записує згенеровані дані в відповідну таблицю. Таким чином існує можливість гнучкої конфігурації отриманих даних за рахунок повного доступу до вихідного коду генератора.

Генерація псевдовипадкових даних відбувається за допомогою об'єкту мови програмування C# - System.Random [22] що являє собою генератор псевдовипадкових чисел, який є алгоритмом, який створює послідовність чисел, що відповідає певним статистичним вимогам щодо випадковості [27], а саме:

Розподіл одержуваних простих чисел має бути близьким до рівномірного на множині всіх k-бітних простих чисел. Існує кілька способів забезпечити здійсненність цієї вимоги.

Процес генерації конкретного випадкового простого числа не можна відтворити, навіть знаючи деталі алгоритму та її реалізації. Зазвичай виконання цієї вимоги забезпечується використанням криптистійкого ГПСЧ, проініціалізованого деяким ключем, що отримується ззовні (тобто не є частиною алгоритму або його реалізації). Як ключ може виступати, наприклад, значення криптографічної хеш-функції від секретної фрази, що запитується у користувача.

Псевдовипадкові числа вибираються з рівною ймовірністю із кінцевої множини чисел. Вибрані числа не є повністю випадковими, оскільки для їх вибору використовується математичний алгоритм, але вони є достатньо випадковими для практичних цілей.

Поточна реалізація класу Random базується на модифікованій версії алгоритму віднімання генератора випадкових чисел [21].

Генерація випадкових слів відбувається завдяки масиву літер та цифр «ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789» та тасуванням їх по індексу з вказанням довжини вихідного літеросполучення за допомогою класу Random. Таким чином відбувається генерація випадкових даних рядкового та чисельного типу.

Вихідний код програми наведено у додатку А.

UML діаграма класів додатку наведено на рис. 2.1.

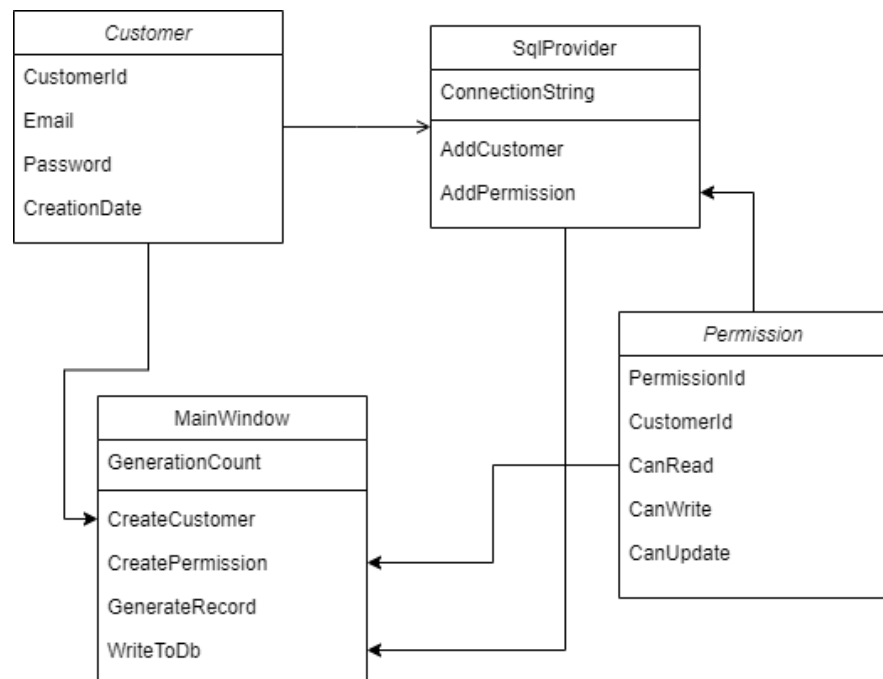


Рис. 2.1. UML діаграма класів

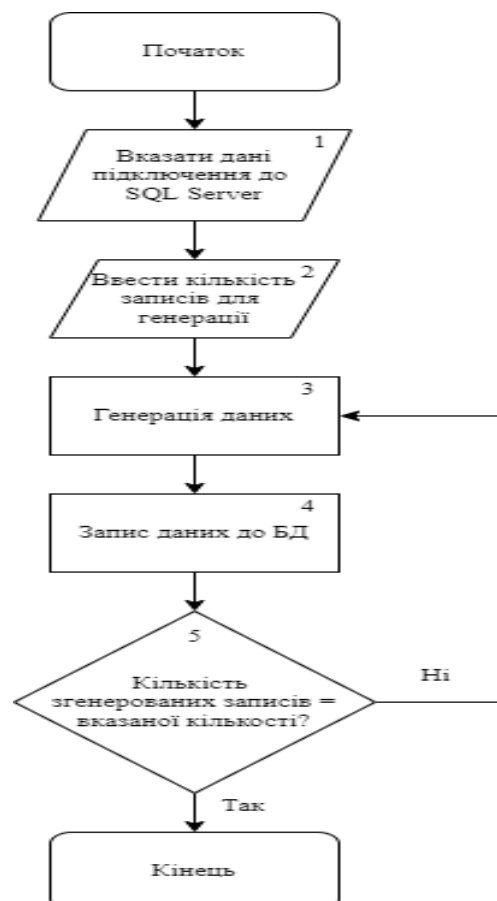


Рис 2.2. Алгоритм використання генератора тестового набору даних.

Для генерації тестового набору даних був використаний такий алгоритм (рис 2.2) використання генератора:

Крок 1. Ввести дані підключення до SQL Server.

Перш за все визначення рядка підключення залежить від типу підключення: підключення по логіну і пароллю, або довірене підключення (trusted connection), де не потрібні логін і пароль (наприклад, при підключенні до локального сервера SQL Server).

Якщо підключення здійснюється за логіном та паролем, то загальний вигляд рядка підключення виглядає наступним чином:

```
Server=адреса_сервера;Database=ім'я_бази_даних;User
Id=логін;Password=пароль;
```

У цьому випадку рядок підключення складається з чотирьох параметрів:

Якщо ми використовуємо так зване довірене підключення (trusted connection) і застосовуємо автентифікацію Windows, наприклад, при підключенні до локального сервера, який запущено на тому ж комп'ютері, то рядок підключення виглядає так:

```
Server=адреса_сервера;Database=назва_бази_даних;Trusted_Connection=Tr
ue;
```

Крок 2. Вказати кількість записів для генерації.

Кількість записів повинна бути вказана у вигляді цілого числа від 0 до 2147483647 (максимальне значення для типу Integer в мові програмування C#)

Крок 3. Генерація одного запису.

Генерація псевдовипадкових значень для кожного поля таблиці.

Крок 4. Запис запису отриманого на кроці 3 до бази даних.

Запис відбувається за використовуючи рядок підключення до бази даних введений на кроці 1.

Крок 5. Якщо кількість записаних даних менше ніж кількість введена на кроці 2, то повернутися на крок 3, інакше завершення роботи.

Ця умова є циклом який використовує кількість повторень вказану на кроці 2.

2.2. Моніторинг і налаштування продуктивності в базі даних Azure SQL і керованому екземплярі Azure SQL

Для відстеження продуктивності бази даних Azure SQL і керованому екземплярі Azure SQL, варто почати з моніторингу ресурсів ЦП та введення-

виведення, які використовуються робочим навантаженням, щодо рівня продуктивності бази даних, який було обрано під час вибору певного рівня служби та рівня продуктивності. Для цього база даних Azure SQL і керований екземпляр Azure SQL видають показники ресурсів, які можна переглянути на порталі Azure або за допомогою одного з цих інструментів керування SQL Server: Azure Data Studio або SQL Server Management Studio (SSMS).

База даних SQL Azure надає ряд порадиноків баз даних, які надають інтелектуальні рекомендації щодо налаштування продуктивності та параметри автоматичного налаштування для підвищення продуктивності. Крім того, Query Performance Insight показує відомості про запити, що відповідають за найбільше використання ЦП та ІО для окремих баз даних і баз даних у пулі.

База даних Azure SQL і керований екземпляр Azure SQL надають розширені можливості моніторингу та налаштування, які підтримуються штучним інтелектом, щоб допомогти у вирішенні несправностей і максимальному підвищенні продуктивності ваших баз даних і рішень. Можна налаштувати потоковий експорт цих Intelligent Insights та інших журналів і показників ресурсів бази даних в одне з кількох місць призначення для споживання та аналізу, зокрема за допомогою SQL Analytics . Azure SQL Analytics — це передове рішення для моніторингу в хмарі для моніторингу продуктивності всіх ваших баз даних у масштабі й у кількох підписках в одному представленні. Щоб отримати список журналів і показників, які можна експортувати, перегляньте діагностичну телеметрію для експорту

SQL Server має власні можливості моніторингу та діагностики, які використовують база даних SQL та керований екземпляр SQL, наприклад, сховище запитів і представлення динамічного керування (DMV) .

Монітор за допомогою аналізу SQL

Azure Monitor SQL Insights [23] — це інструмент для моніторингу керованих екземплярів Azure SQL, баз даних Azure SQL та екземплярів SQL Server у віртуальних машинах Azure SQL. Ця служба використовує віддалений агент для збору даних із представлень динамічного керування (DMV) і направляє дані до Azure Log Analytics, де їх можна відстежувати та аналізувати. Можна переглянути ці дані з Azure Monitor у наданих представленнях або отримати доступ до даних журналу безпосередньо, щоб виконувати запити та аналізувати тенденції.

Моніторинг ресурсів бази даних SQL Azure та керованого екземпляра Azure SQL

На порталі Azure у поданні «Метрики» можна швидко відстежувати різноманітні показники ресурсів. Ці показники дозволяють побачити, чи база даних досягає 100% ресурсів процесора, пам'яті або ресурсів в/в. Високий відсоток DTU або процесора, а також високий відсоток ІО вказує на те, що для робочого навантаження може знадобитися більше ресурсів CPU або ІО. Це також може вказувати на запити, які потрібно оптимізувати.

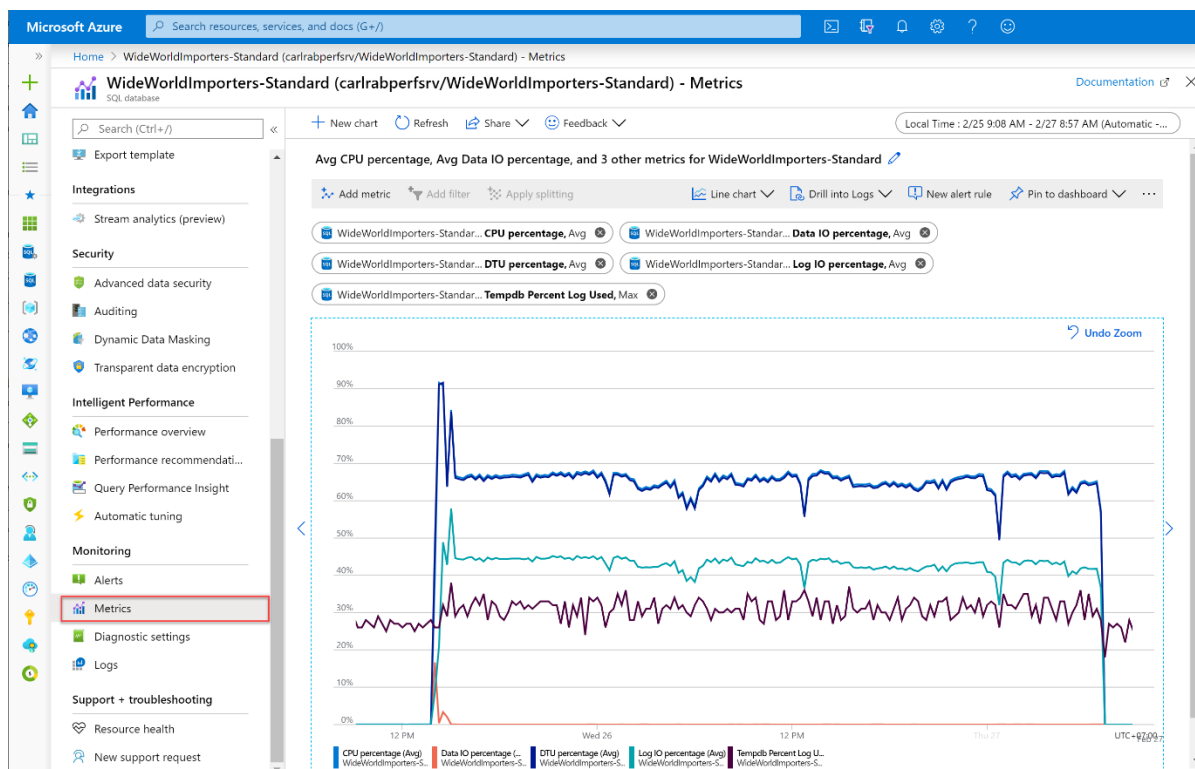


Рис. 2.3. Метрики бази даних SQL Azure

Показники ресурсів

Консультанти баз даних у базі даних SQL Azure

База даних SQL Azure містить поради баз даних, які надають рекомендації щодо налаштування продуктивності для окремих баз даних і баз даних у пулі. Ці рекомендації доступні на порталі Azure, а також за допомогою PowerShell. Можна увімкнути автоматичне налаштування, щоб база даних SQL Azure могла автоматично реалізовувати ці рекомендації щодо налаштування.

Статистика ефективності запитів у базі даних SQL Azure

Query Performance Insight показує продуктивність на порталі Azure найбільш споживаючих і найдовше виконуваних запитів для одиничних і об'єднаних баз даних.

Починаючи з вересня 2020 року, бази даних із надзвичайно низьким використанням можуть відображатися на порталі з меншим, ніж фактичне

використання. Через те, як передається телеметрія під час перетворення подвійного значення до найближчого цілого, певні суми використання менше 0,5 будуть округлені до 0, що спричинить втрату деталізації переданої телеметрії.

Intelligent Insights для бази даних Azure SQL і керованого екземпляра Azure SQL використовує вбудований інтелект для постійного моніторингу використання бази даних за допомогою штучного інтелекту та виявлення подій, які підривають роботу, які спричиняють низьку продуктивність. Intelligent Insights автоматично виявляє проблеми з продуктивністю баз даних на основі часу очікування виконання запиту, помилок або тайм-аутів. Після виявлення виконується детальний аналіз, який створює журнал ресурсів (так званий SQLInsights) з інтелектуальною оцінкою проблем. Ця оцінка складається з аналізу першопричин проблеми продуктивності бази даних і, якщо можливо, рекомендацій щодо підвищення продуктивності.

Intelligent Insights [31] — це унікальна можливість вбудованого інтелекту Azure, яка забезпечує таку цінність:

- Проактивний моніторинг

- Індивідуальна статистика ефективності

- Раннє виявлення зниження продуктивності бази даних

- Аналіз першопричин виявлених проблем

- Рекомендації щодо підвищення продуктивності

- Розширюйте можливості для сотень тисяч баз даних

- Позитивний вплив на ресурси DevOps та загальну вартість володіння

Рішення моніторингу в Azure Monitor [24] забезпечують аналіз операцій певної програми або служби Azure. Ця робота містить огляд рішень моніторингу в Azure і докладні відомості про їх використання та встановлення. У Azure Monitor можемо додати рішення моніторингу для всіх використовуваних додатків і служб. Зазвичай вони надаються безкоштовно, але збір даних може спричинити певну плату за використання ресурсів.

На сторінці «Огляд рішень» в Azure Monitor відображається плитка для кожного рішення, встановленого в log Analytics робочої області. Щоб відкрити цю сторінку, перейдіть до Azure Monitor в портал Azure. В меню Insights (аналітика) оберіть «Додатково», щоб відкрити центр Insights, а потім натисніть «log Analytics» в робочій області (рис. 2.4.).

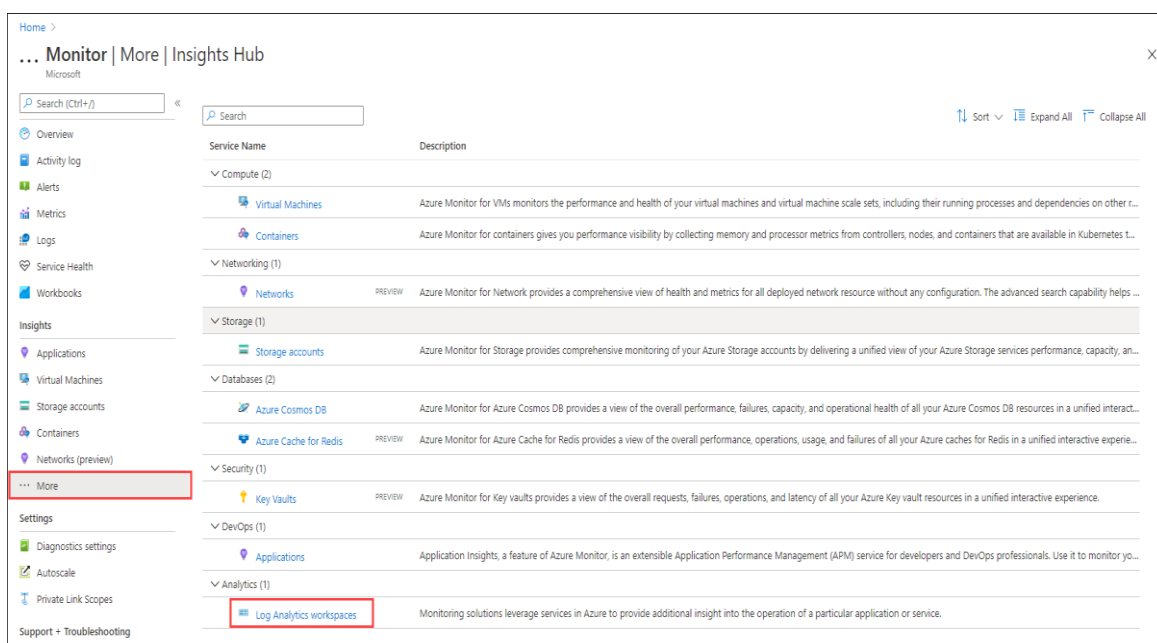


Рис. 2.4. Огляд рішень в Azure Monitor

Використовуйте списки, що розкриваються в верхній частині екрану, щоб міняти робочу область і діапазон часу.

Натисніть розділ «рішення», щоб відкрити його подання з докладним аналізом зібраних даних (рис. 2.7.).

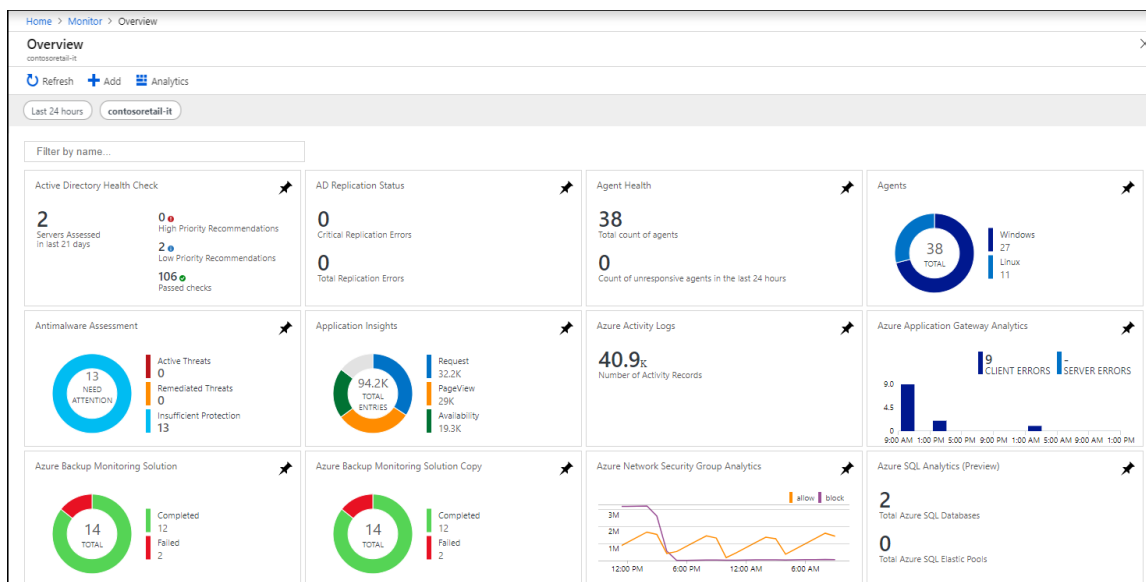


Рис. 2.5. Меню портал Azure з вибраними рішеннями та рішеннями, відображеними в області рішення.

Рішення моніторингу можуть містити кілька типів ресурсів Azure, і всі ресурси в складі рішення можна переглянути точно так же, як будь-який інший ресурс. Наприклад, будь-які запити журналу, включені в рішення, відображаються в розділі запити рішення в браузері запитів. Ці запити можна використовувати при виконанні нерегламентованого аналізу з запитами журналів.

Роздивимось детальніше можливості та засоби Azure Monitor.

Описати ключові можливості інструменту можна наступним чином:

route streaming - передача даних між сервісами і функціонал оповіщення;

store and archive - сховище даних;

query - надання доступу до даних на читання;

visualize - візуалізація, уявлення і аналітика;

automate - автоматизація і тригери процесів (autoscale, email, webhook);

Представлене нижче зображення характерно для обчислюваних (виконуваних) ресурсів і відрізняється від необчислюваних лише наявністю Application Logs and metrics (рис.2.6).

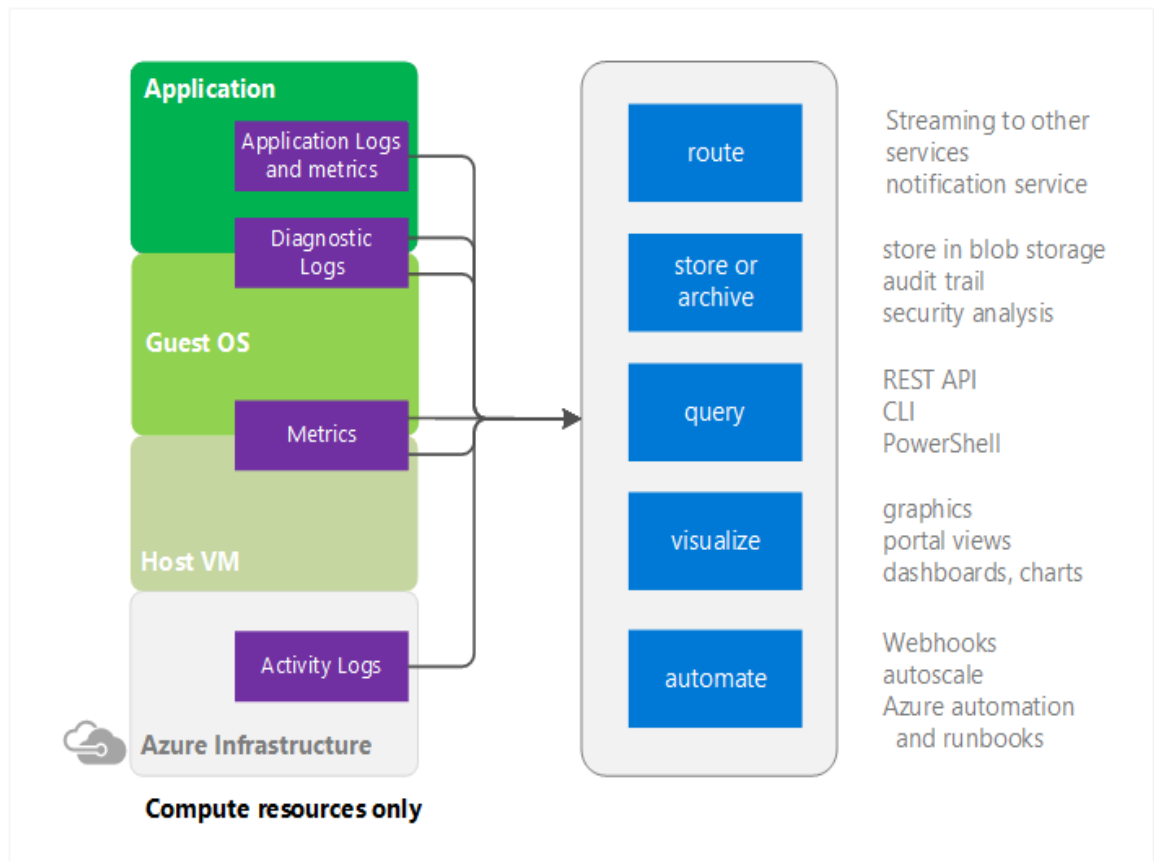


Рис. 2.6. Інфраструктура моніторингу Azure

Activity log [32]

Даний розділ моніторингу містить інформацію про операції, виконані в рамках конкретних компонентів (ресурсів):

Найменування (тип) операції;

Хто її ініціював і коли вона була завершена;

Статус її виконання;

Крім того, надається можливість (рис. 2.7.):

Експорту в Storage Account & Azure Event Hub;

Доступу за допомогою PowerShell & REST API;

Додавання правил оповіщення;

Аналізу в PowerBI;

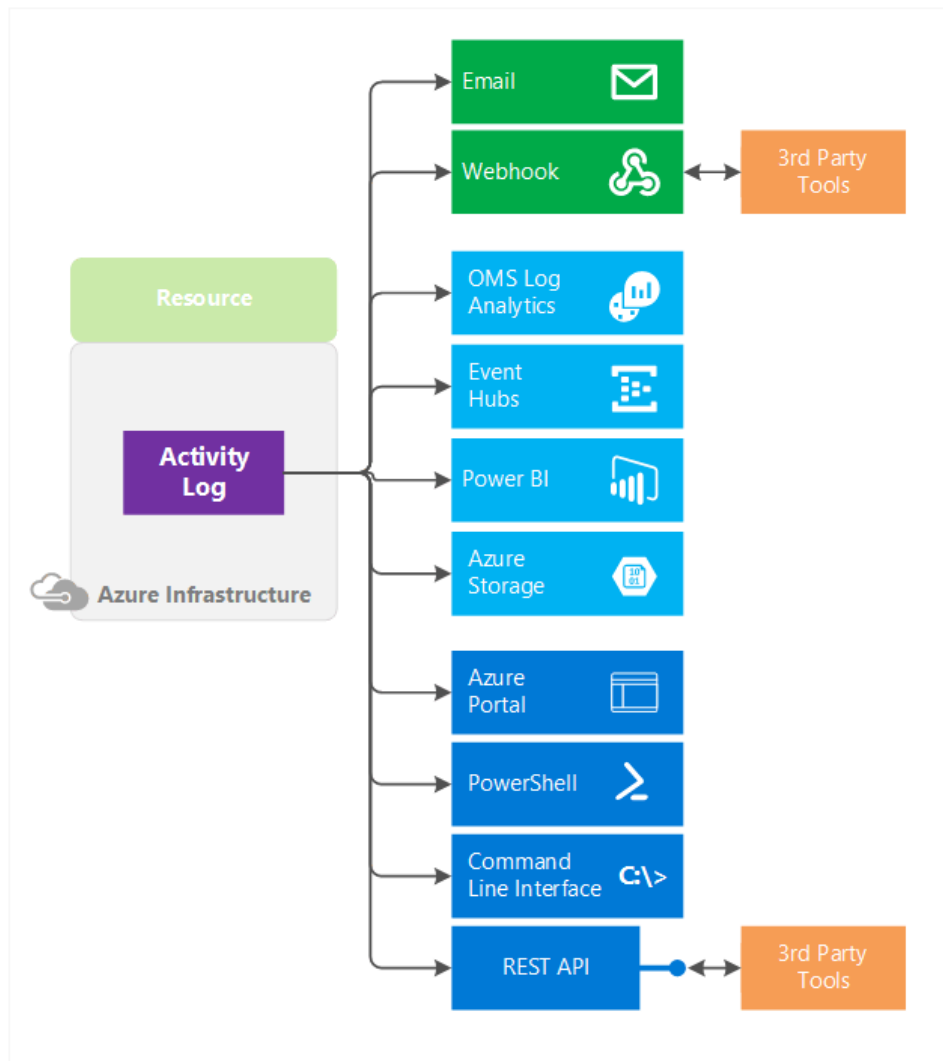


Рис. 2.7. Експорт даних до сховищ

Metrics and Events [33]

Крім можливості відслідковувати інформацію про метрики і події використовуваних компонентів Azure також є додавання власних custom metrics & events.

Якщо ми хочемо зробити дію засновану на значенні метрики, то в цьому нам допоможе система оповіщення і реагування, в арсеналі якої є можливість відправляти email повідомлення, викликати webhook або запускати logic app (за допомогою request trigger) (рис. 2.8).

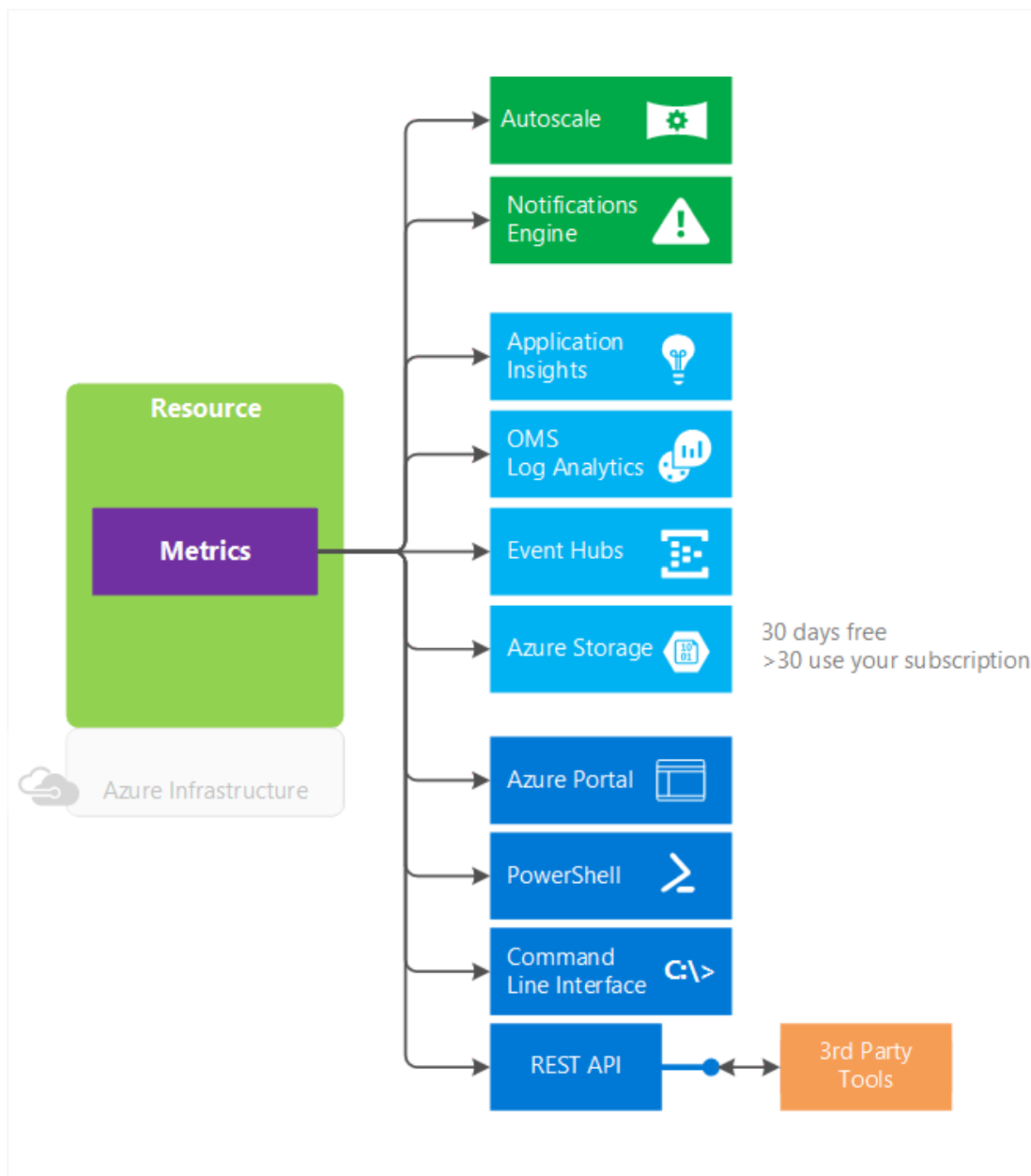


Рис. 2.8. Засоби експорту метрик

Log search

Завдяки log search і спеціалізованій мові запитів доступна можливість виконувати `query/filter/aggregate` може бути застосовано до логів і метрик, а також візуалізувати підсумкові результати в необхідному вигляді (table, list, bar) (рис. 2.9).

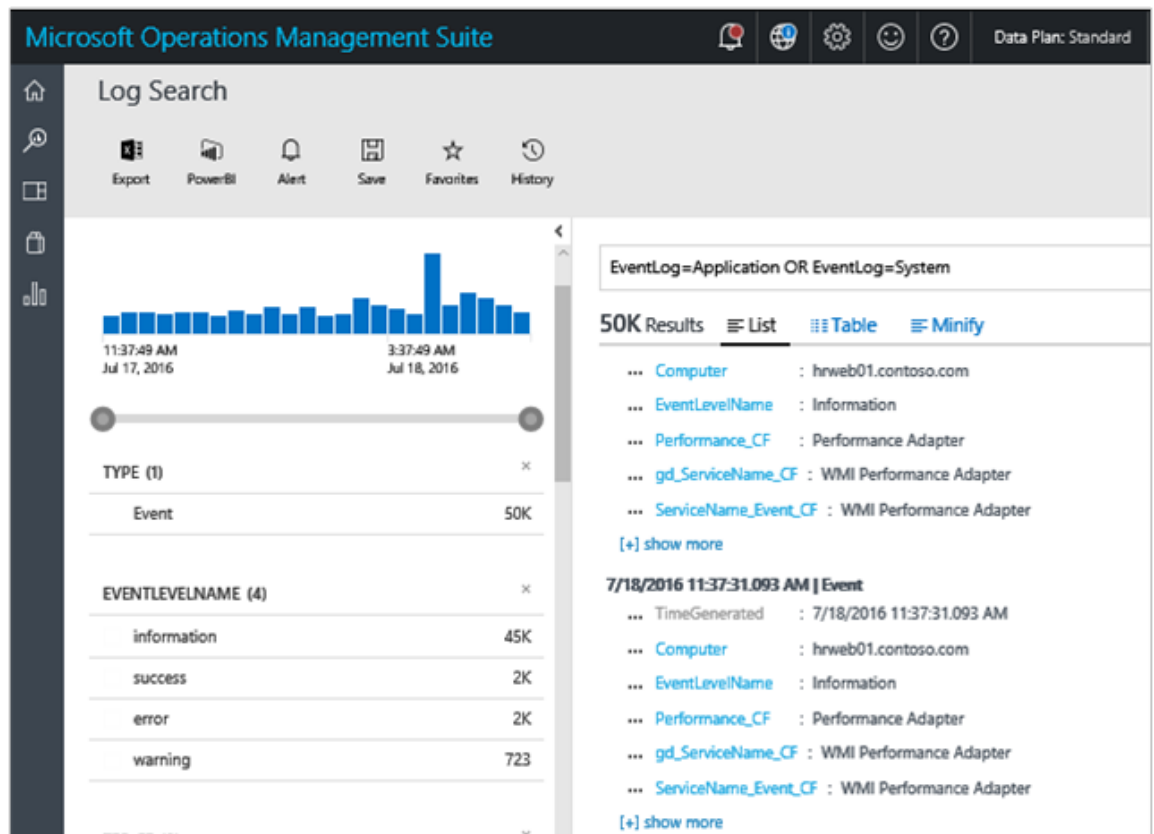


Рис. 2.9. Візуалізація підсумкових результатів

Autoscale [34]

Горизонтальне масштабування засноване на одному з двох видів умов (рис. 2.10):

На підставі значення метрики - наприклад, збільшити кількість запущених екземплярів того чи іншого ресурсу в разі перевищення використання% CPU або завантаження ОЗУ;

На підставі розкладу - наприклад, якщо навантаження в вихідні дні знижується, то можна зменшувати і кількість примірників до необхідного мінімуму;

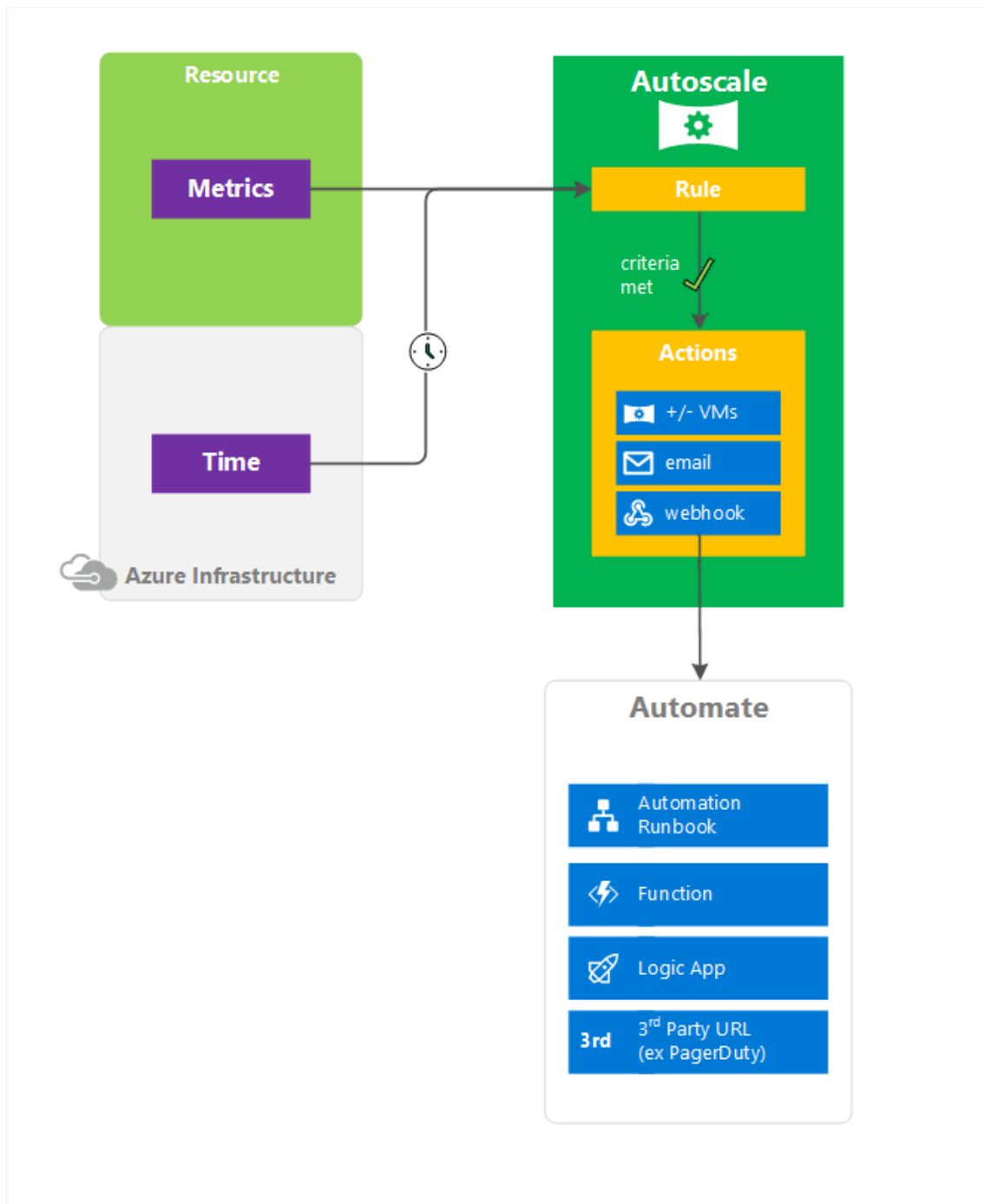


Рис. 2.10. Горизонтальне масштабування

Azure Service Health [35] (рис. 2.11)

Завдяки такому компоненту як Azure Service Health з'являється можливість своєчасно дізнаватися про технічні роботи та збої в інфраструктурі Azure, які можуть торкнутися доступності розгорнутих в хмарних сервісах ресурсів. Як сповіщення можуть виступати повідомлення поштою, телефоном або webhook.

Також з'являється можливість завчасно підготуватися до планового обслуговування Azure.

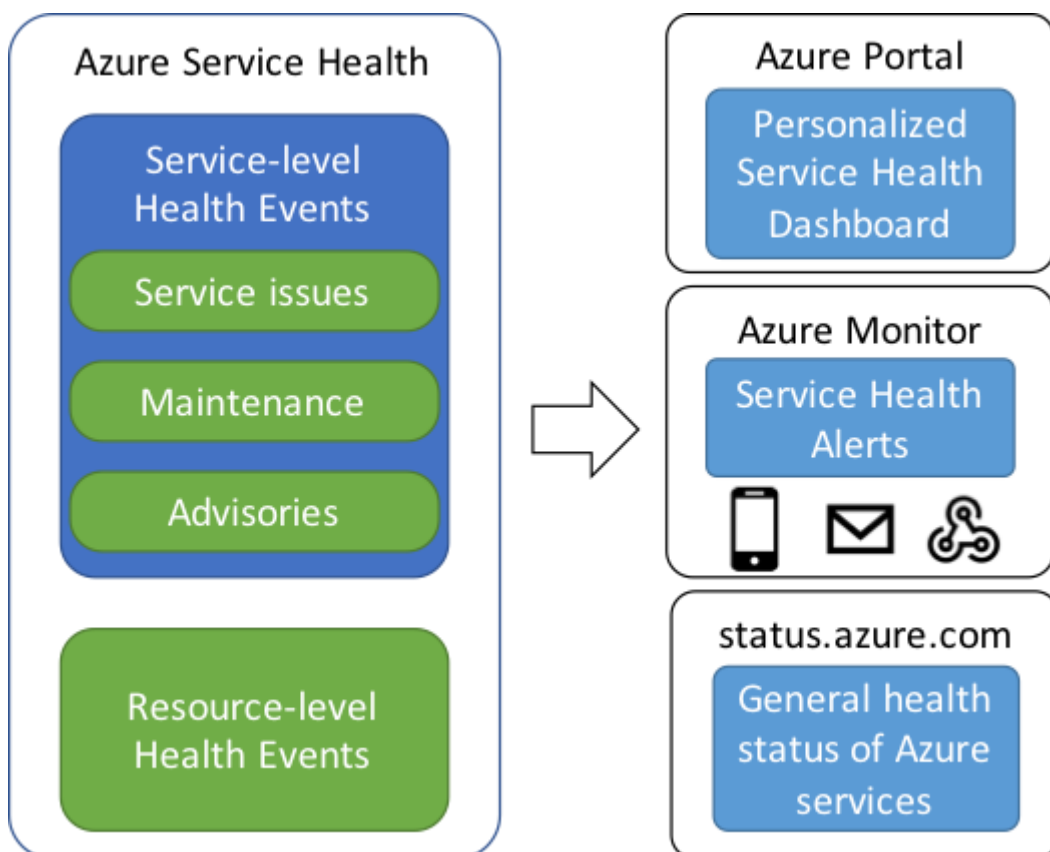


Рис. 2.11. Методи сповіщення

3 ПРОВЕДЕННЯ ТА АНАЛІЗ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

3.1 Проектування та опис тестової бази даних

Загальний обсяг даних в БД становить 10,5 ГБ

Таблиця Customer містить 1001002 записів (табл. 3.1).

Таблиця 3.1

Опис структури таблиці Customer

| Таблиця Customer | |
|------------------|-------------------------------|
| Поле | Тип даних |
| CustomerId | uniqueidentifier, Primary key |
| Email | nvarchar(100) |
| Password | nvarchar(50) |
| CreationDate | datetime |

Таблиця Permission містить 1001002 записів (табл. 3.2).

Таблиця 3.2

Опис структури таблиці Permission

| Таблиця Permission | |
|--------------------|-------------------------------|
| Поле | Тип даних |
| PermissionId | uniqueidentifier, Primary key |
| CustomerId | uniqueidentifier, Foreign key |
| CanRead | bit |
| CanWrite | bit |
| CanUpdate | bit |

Таким чином загальна кількість записів в БД становить 2002004 записів.
ER-модель бази даних приведено на рис. 3.1.

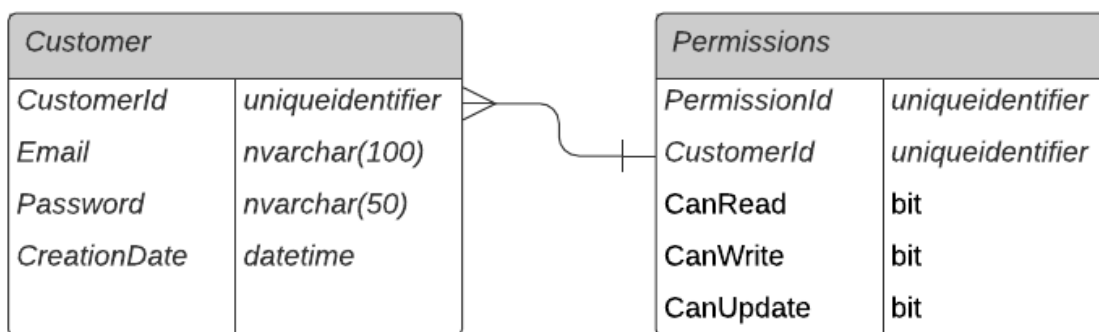


Рис 3.1. ER-модель бази даних

3.2. Генерація тестового набору даних

Для генерації тестового набору даних було розроблено додаток мовою C#, який генерує дані за вказаними параметрами, а саме, кількість даних, підключення до SQL серверу, що дозволяє генерувати дані як на локальному ресурсі так і на Microsoft Azure.

Під час генерації додаток випадковим чином створює дані для таблиць Customer та Permission і записує згенеровані дані в відповідну таблицю.

Вихідний код програми наведено у додатку А.

UML діаграма класів додатку наведено на рис. 3.2.

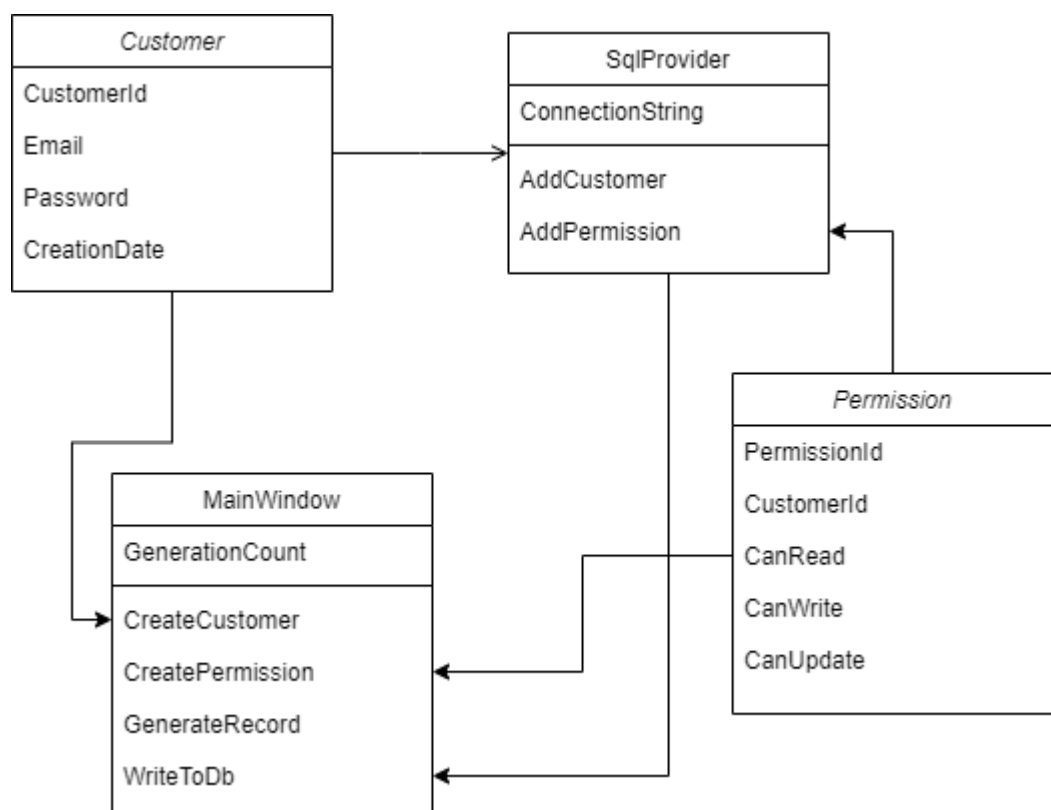


Рис. 3.2. UML діаграма класів

3.3. Аналіз ефективності використання моделей придбання для роботи з реляційними БД

Для проведення дослідження були сформовані наступні типи запитів.

Прості запити (SQ):

1. `SELECT * FROM dbo.Customers` - в результаті виконання запиту буде отримано всі записи з таблиці `dbo.Customer`.

2. `SELECT * FROM dbo.Customers WHERE Email LIKE 'A%'` – в результаті виконання запиту буде отримано всі записи з таблиці `dbo.Customer` в якій значення параметру `Email` починається з літери «А».

3. `SELECT * FROM dbo.Customers WHERE CreationDate < GETDATE()` - в результаті виконання запиту буде отримано всі записи з таблиці `dbo.Customer` в якій дата внесення запису менша ніж дата виконання запиту.

Складні запити (CQ):

1. `SELECT * FROM dbo.Customers AS cus INNER JOIN dbo.Permissions AS per ON cus.CustomerId = per.CustomerId` - в результаті виконання запиту буде отримано всі записи з таблиці `dbo.Customer` у яких унікальний ідентифікатор співпадає з унікальним ідентифікатором в таблиці `dbo.Permissions`

2. `SELECT * FROM dbo.Customers AS cus INNER JOIN dbo.Permissions AS per ON cus.CustomerId = per.CustomerId WHERE cus.Password LIKE 'A%' AND per.CanRead = 1` – в результаті виконання запиту буде отримано всі записи з таблиці `dbo.Customer` у яких у яких унікальний ідентифікатор співпадає з унікальним ідентифікатором в таблиці `dbo.Permissions`, значення параметру `пароль` починається з літери «А» та значення параметру `CanRead` дорівнює одиниці.

3. `SELECT * FROM dbo.Customers AS cus LEFT JOIN dbo.Permissions AS per ON cus.CustomerId = per.CustomerId WHERE cus.CreationDate < GETDATE()` - в результаті виконання запиту буде отримано всі записи з таблиці `dbo.Customer` у яких у яких унікальний ідентифікатор співпадає з унікальним ідентифікатором в таблиці `dbo.Permissions` та дата внесення запису менша ніж дата виконання запиту.

Для проведення експерименту використовується наступний алгоритм.

Крок 1. Обрати середовище (локальне або Microsoft Azure)

Крок 2. Обрати модель придбання БД Azure. Якщо обрано локальне середовище, одразу перейти на крок 3.

Крок 3. Створити базу даних.

Крок 4. Згенерувати достатню кількість даних. Не менше 10 ГБ.

Крок 5. Виконати прості та складні запити.

Крок 6. Провести порівняльний аналіз виконаних запитів.

Схема алгоритму приведена на рисунку 3.3.

Під час проведення експерименту на локальному ресурсі використовується ПК з характеристиками:

Процесор – Intel Core i7-10510U 2.30GHz;

Оперативна пам'ять обсягом 32 ГБ.;

SSD накопичувач обсягом 512 ГБ.

Для відображення потужності БД Azure використовується одиниця пропускної спроможності бази даних (DTU): DTU надають спосіб опису відносної ємності рівня продуктивності Basic, Standard, і Преміум бази даних. DTU засновані на змішаному вимірі ЦП, пам'яті, операцій читання і запису. У міру збільшення DTU потужність, пропонована рівнем продуктивності, збільшується. Наприклад, рівень продуктивності з 5 DTU має в п'ять разів більше потужності, ніж рівень продуктивності з 1 DTU. Максимальна квота DTU застосовується до кожного сервера.

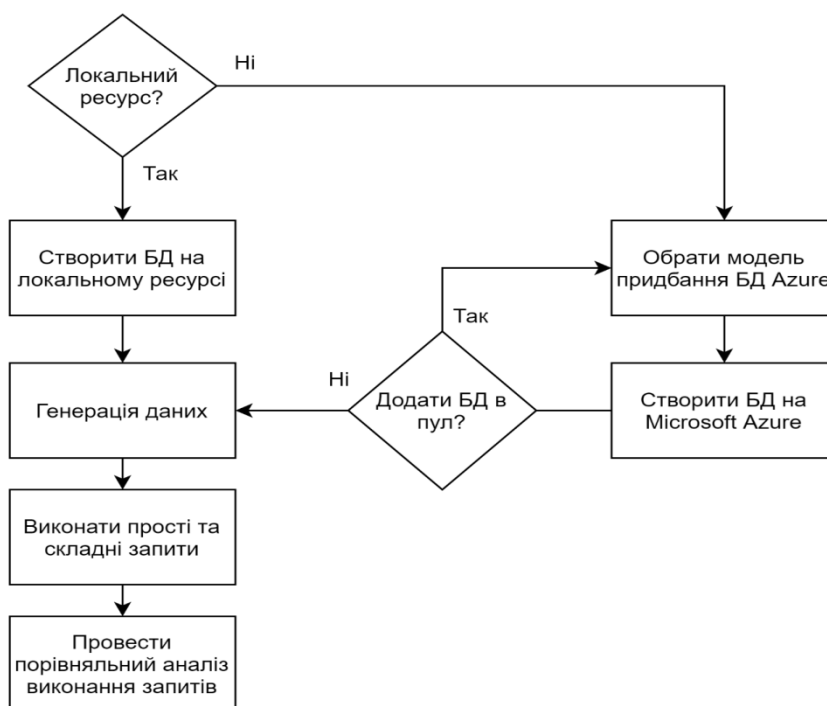


Рис. 3.3. Схема алгоритму проведення експерименту

Результати виконання запитів для моделі придбання S0 наведено у табл. 3.1.

Таблиця 3.3

Результати експерименту виконання запитів у базу даних для моделі придбання S0

| № | SQL запит | Таблиця | Час виконання, сек. | Кількість вибраних записів | Відношення кількості вибраних записів до загальної кількості, % |
|---|-----------|---------|---------------------|----------------------------|-----------------------------------------------------------------|
| | | | | | |

| Прості запити (SQ) | | | | | |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|-----|---------|-------|
| SQ 1 | SELECT * FROM dbo.Customers | dbo.Customer | 79 | 1001002 | 50,03 |
| SQ 2 | SELECT * FROM dbo.Customers WHERE Email LIKE 'A1%' | dbo.Customer | 2 | 706 | 0,04 |
| SQ 3 | SELECT * FROM dbo.Customers WHERE CreationDate < GETDATE() | dbo.Customer | 93 | 201002 | 10,05 |
| Складні запити (CQ) | | | | | |
| CQ 1 | SELECT * FROM dbo.Customers AS cus INNER JOIN dbo.Permissions AS per ON cus.CustomerId = per.CustomerId | dbo.Customer, dbo.Permission | 105 | 1001002 | 50,03 |
| CQ 2 | SELECT * FROM dbo.Customers AS cus INNER JOIN dbo.Permissions AS per ON cus.CustomerId = per.CustomerId WHERE cus.Password LIKE 'A%' AND per.CanRead = 1 | dbo.Customer, dbo.Permission | 54 | 13827 | 0,69 |
| CQ 3 | SELECT * FROM dbo.Customers AS cus LEFT JOIN dbo.Permissions AS per ON cus.CustomerId = per.CustomerId WHERE cus.CreationDate < GETDATE() | dbo.Customer, dbo.Permission | 103 | 201002 | 10,05 |

Результати виконання запитів для моделі придбання S3 наведено у табл. 3.2.

Таблиця 3.4

Результати експерименту виконання запитів у базу даних для моделі придбання S3

| № | SQL запит | Таблиця | Час виконання, сек. | Кількість вибраних записів | Відношення кількості вибраних записів до |
|---|-----------|---------|---------------------|----------------------------|------------------------------------------|
|---|-----------|---------|---------------------|----------------------------|------------------------------------------|

| | | | | | загальної кількості, % |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|----|---------|------------------------|
| Прості запити (SQ) | | | | | |
| SQ 1 | SELECT * FROM dbo.Customers | dbo.Customer | 73 | 1001002 | 50,03 |
| SQ 2 | SELECT * FROM dbo.Customers WHERE Email LIKE 'A1%' | dbo.Customer | 1 | 706 | 0,04 |
| SQ 3 | SELECT * FROM dbo.Customers WHERE CreationDate < GETDATE() | dbo.Customer | 99 | 201002 | 10,05 |
| Складні запити (CQ) | | | | | |
| CQ 1 | SELECT * FROM dbo.Customers AS cus INNER JOIN dbo.Permissions AS per ON cus.CustomerId = per.CustomerId | dbo.Customer, dbo.Permission | 97 | 1001002 | 50,03 |
| CQ 2 | SELECT * FROM dbo.Customers AS cus INNER JOIN dbo.Permissions AS per ON cus.CustomerId = per.CustomerId WHERE cus.Password LIKE 'A%' AND per.CanRead = 1 | dbo.Customer, dbo.Permission | 43 | 13827 | 0,69 |
| CQ 3 | SELECT * FROM dbo.Customers AS cus LEFT JOIN dbo.Permissions AS per ON cus.CustomerId = per.CustomerId WHERE cus.CreationDate < GETDATE() | dbo.Customer, dbo.Permission | 54 | 201002 | 10,05 |

3.3. Результати виконання запитів для моделі придбання S7 наведено у табл.

Таблиця 3.5

Результати експерименту виконання запитів у базу даних для моделі придбання S7

| № | SQL запит | Таблиця | Час виконання, сек. | Кількість вибраних записів | Відношення кількості вибраних записів до загальної кількості, % |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|---------------------|----------------------------|-----------------------------------------------------------------|
| Прості запити (SQ) | | | | | |
| SQ 1 | SELECT * FROM dbo.Customers | dbo.Customer | 53 | 1001002 | 50,03 |
| SQ 2 | SELECT * FROM dbo.Customers WHERE Email LIKE 'A1%' | dbo.Customer | 1 | 706 | 0,04 |
| SQ 3 | SELECT * FROM dbo.Customers WHERE CreationDate < GETDATE() | dbo.Customer | 74 | 201002 | 10,05 |
| Складні запити (CQ) | | | | | |
| CQ 1 | SELECT * FROM dbo.Customers AS cus INNER JOIN dbo.Permissions AS per ON cus.CustomerId = per.CustomerId | dbo.Customer, dbo.Permission | 84 | 1001002 | 50,03 |
| CQ 2 | SELECT * FROM dbo.Customers AS cus INNER JOIN dbo.Permissions AS per ON cus.CustomerId = per.CustomerId WHERE cus.Password LIKE 'A%' AND per.CanRead = 1 | dbo.Customer, dbo.Permission | 37 | 13827 | 0,69 |
| CQ 3 | SELECT * FROM dbo.Customers AS cus LEFT JOIN dbo.Permissions AS per ON cus.CustomerId = per.CustomerId WHERE cus.CreationDate < GETDATE() | dbo.Customer, dbo.Permission | 42 | 201002 | 10,05 |

Результати виконання запитів на локальному ресурсі наведено у табл. 3.4.

Таблиця 3.6

Результати експерименту виконання запитів у базу даних на локальному ресурсі

| № | SQL запит | Таблиця | Час виконання, сек. | Кількість вибраних записів | Відношення кількості вибраних записів до загальної кількості, % |
|---|-----------|---------|---------------------|----------------------------|-----------------------------------------------------------------|
|---|-----------|---------|---------------------|----------------------------|-----------------------------------------------------------------|

| Прості запити (SQ) | | | | | |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|----|---------|-------|
| SQ 1 | SELECT * FROM dbo.Customers | dbo.Customer | 15 | 1001002 | 50,03 |
| SQ 2 | SELECT * FROM dbo.Customers WHERE Email LIKE 'A1%' | dbo.Customer | 1 | 706 | 0,04 |
| SQ 3 | SELECT * FROM dbo.Customers WHERE CreationDate < GETDATE() | dbo.Customer | 3 | 201002 | 10,05 |
| Складні запити (CQ) | | | | | |
| CQ 1 | SELECT * FROM dbo.Customers AS cus INNER JOIN dbo.Permissions AS per ON cus.CustomerId = per.CustomerId | dbo.Customer, dbo.Permission | 17 | 1001002 | 50,03 |
| CQ 2 | SELECT * FROM dbo.Customers AS cus INNER JOIN dbo.Permissions AS per ON cus.CustomerId = per.CustomerId WHERE cus.Password LIKE 'A%' AND per.CanRead = 1 | dbo.Customer, dbo.Permission | 1 | 13827 | 0,69 |
| CQ 3 | SELECT * FROM dbo.Customers AS cus LEFT JOIN dbo.Permissions AS per ON cus.CustomerId = per.CustomerId WHERE cus.CreationDate < GETDATE() | dbo.Customer, dbo.Permission | 5 | 201002 | 10,05 |

Відсоток навантаження на ОЗУ та CPU на локальному ресурсі наведено в табл. 3.7.

Навантаження ОЗУ та CPU на локальному ресурсі

| № | SQL запит | Таблиця | Час виконання, сек. | Кількість вибраних записів | Відношення кількості вибраних записів до загальної кількості, % | Об'єм задіяної ОЗУ, ГБ | Завантаженість процесорних ядер, % |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|---------------------|----------------------------|-----------------------------------------------------------------|------------------------|------------------------------------|
| Прості запити (simple request, SR) | | | | | | | |
| SQ 1 | SELECT * FROM dbo.Customers | dbo.Customer | 15 | 1001002 | 50,03 | 1,17 | 23 |
| SQ 2 | SELECT * FROM dbo.Customers WHERE Email LIKE 'A1%' | dbo.Customer | 1 | 706 | 0,04 | 0,9 | 3,8 |
| SQ 3 | SELECT * FROM dbo.Customers WHERE CreationDate < GETDATE() | dbo.Customer | 3 | 201002 | 10,05 | 1,05 | 20,1 |
| Складні запити (complex request, CR) | | | | | | | |
| CQ 1 | SELECT * FROM dbo.Customers AS cus INNER JOIN dbo.Permissions AS per ON cus.CustomerId = per.CustomerId | dbo.Customer, dbo.Permission | 1 | 1001002 | 50,03 | 1,43 | 25 |
| CQ 2 | SELECT * FROM dbo.Customers AS cus INNER JOIN dbo.Permissions AS per ON cus.CustomerId = per.CustomerId WHERE cus.Password LIKE 'A%' AND per.CanRead = 1 | dbo.Customer, dbo.Permission | 1 | 13827 | 0,69 | 1,22 | 4,5 |
| CQ 3 | SELECT * FROM dbo.Customers AS cus LEFT JOIN dbo.Permissions AS per ON cus.CustomerId = per.CustomerId WHERE cus.CreationDate < GETDATE() | dbo.Customer, dbo.Permission | 5 | 201002 | 10,05 | 1,14 | 23 |

Таким чином можна зробити висновок, що час виконання запиту залежить не від його складності, а від кількості записів які будуть обрані в результаті запиту. Графік відношення часу виконання до відсотка від загальної кількості записів наведено на рис. 3.4.

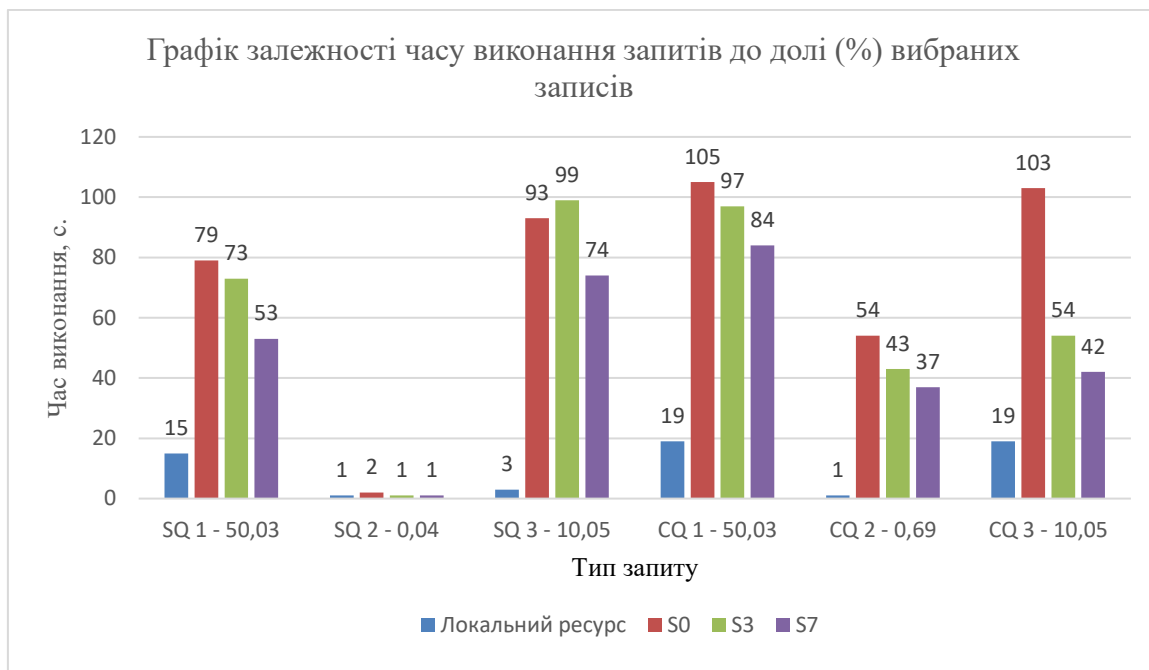


Рис. 3.4. Графік залежності часу виконання запитів до відсотка від загальної кількості записів

3.4. Аналіз завантаженості моделей придбання Azure

Таблиця 3.8

Навантаження моделей придбання при генерації даних

| Модель придбання | Навантаження DTU, % | Навантаження CPU, % |
|------------------|---------------------|---------------------|
| S0 | 100 | 15,57 |
| S3 | 100 | 3,36 |
| S7 | 55 | 0,72 |

Таблиця 3.9

Навантаження моделей придбання при виконанні запитів

| Модель придбання | Навантаження DTU, % | Навантаження CPU, % | Навантаження ОЗУ, Гб. | Кількість операцій за секунду |
|------------------|---------------------|---------------------|-----------------------|-------------------------------|
| Запит SQ1 | | | | |
| S0 | 44 | 4,2 | 1,17 | 12671 |
| S3 | 19 | 1,82 | 1,17 | 13712 |

| | | | | |
|-----------|------|------|------|-------|
| S7 | 1,7 | 0,08 | 1,17 | 18887 |
| Запит SQ2 | | | | |
| S0 | 52 | 3,8 | 0,9 | 353 |
| S3 | 23 | 3,85 | 0,9 | 706 |
| S7 | 2 | 0,31 | 0,9 | 706 |
| Запит SQ3 | | | | |
| S0 | 68 | 5,2 | 1,05 | 2161 |
| S3 | 22 | 1,54 | 1,05 | 2030 |
| S7 | 1,8 | 0,9 | 1,05 | 2716 |
| Запит CQ1 | | | | |
| S0 | 94 | 4,95 | 1,43 | 9533 |
| S3 | 37 | 1,32 | 1,43 | 10320 |
| S7 | 3,7 | 0,35 | 1,43 | 11917 |
| Запит CQ2 | | | | |
| S0 | 83 | 5,74 | 1,22 | 256 |
| S3 | 32 | 3,57 | 1,22 | 322 |
| S7 | 1,4 | 0,21 | 1,22 | 374 |
| Запит CQ3 | | | | |
| S0 | 81 | 4,3 | 1,14 | 1951 |
| S3 | 31,7 | 2,69 | 1,14 | 3722 |
| S7 | 2,1 | 0,31 | 1,14 | 4786 |

Таблиця 3.10

Вартість моделей придбання Azure

| Модель придбання | Кількість DTU | Ціна за 1 DTU за місяць, \$ | Сума за місяць, \$ |
|------------------|---------------|-----------------------------|--------------------|
| S0 | 10 | 1,65 | 16,5 |
| S3 | 100 | 1,65 | 165 |
| S7 | 800 | 1,65 | 1320 |

Таблиця 3.11

Вартість виконання запитів

| Модель придбання | Тип запиту | Ціна за 1 секунду, \$ | Час виконання запиту, сек. | Вартість виконання запиту, \$ |
|------------------|------------|-----------------------|----------------------------|-------------------------------|
| S0 | SQ1 | 0.00038 | 79 | 0.03002 |
| S0 | SQ2 | 0.00038 | 2 | 0.00076 |
| S0 | SQ3 | 0.00038 | 93 | 0.03534 |
| S0 | CQ1 | 0.00038 | 105 | 0.0399 |
| S0 | CQ2 | 0.00038 | 54 | 0.02052 |
| S0 | CQ3 | 0.00038 | 103 | 0.03914 |
| S3 | SQ1 | 0.0038 | 73 | 0.2774 |
| S3 | SQ2 | 0.0038 | 1 | 0.0038 |

| | | | | |
|----|-----|--------|----|--------|
| S3 | SQ3 | 0.0038 | 99 | 0.3762 |
| S3 | CQ1 | 0.0038 | 97 | 0.3686 |
| S3 | CQ2 | 0.0038 | 43 | 0.1634 |
| S3 | CQ3 | 0.0038 | 54 | 0.2052 |
| S7 | SQ1 | 0.03 | 53 | 1.59 |
| S7 | SQ2 | 0.03 | 1 | 0.03 |
| S7 | SQ3 | 0.03 | 74 | 2.22 |
| S7 | CQ1 | 0.03 | 84 | 2.52 |
| S7 | CQ2 | 0.03 | 37 | 1.11 |
| S7 | CQ3 | 0.03 | 42 | 1.26 |



Рис. 3.5. Графік залежності вартості використання моделей придбання від часу виконання простих запитів



Рис. 3.6. Графік залежності вартості використання моделей придбання від часу виконання складних запитів

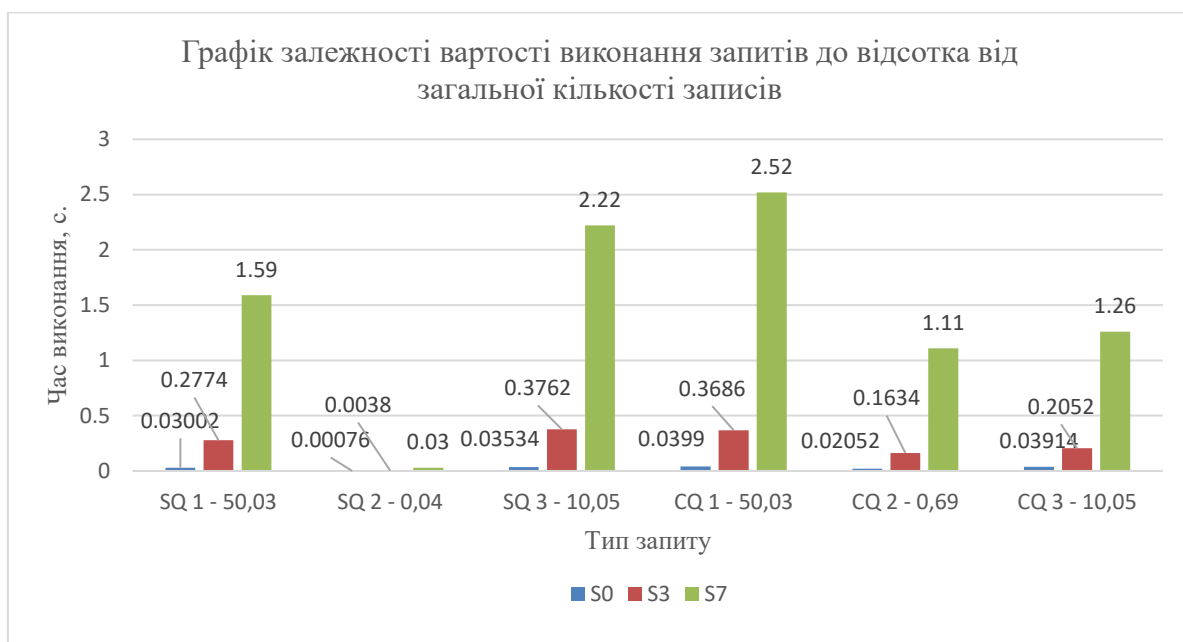


Рис. 3.7. Графік залежності вартості виконання запитів до відсотка від загальної кількості записів

Таблиця 3.12

Швидкість зчитування з диску

| Модель придбання | Час виконання запиту, с. | Розмір отриманих записів, Мб. | Швидкість зчитування, Мб/с. |
|------------------|--------------------------|-------------------------------|-----------------------------|
| | | | |

| | | | |
|------------------|----|-----|-------|
| Локальний ресурс | 15 | 472 | 31.46 |
| S0 | 73 | 472 | 5.97 |
| S3 | 73 | 472 | 6.46 |
| S7 | 53 | 472 | 8.90 |

Таблиця 3.13

Кількість операцій за секунду на локальному ресурсі

| Тип запиту | Кількість вибраних записів | Час виконання, с. | Кількість операцій за секунду |
|------------|----------------------------|-------------------|-------------------------------|
| SQ 1 | 1001002 | 15 | 66733 |
| SQ 2 | 706 | 1 | 706 |
| SQ 3 | 201002 | 3 | 67001 |
| CQ 1 | 1001002 | 17 | 58882 |
| CQ 2 | 13827 | 1 | 13827 |
| CQ 3 | 201002 | 5 | 40200 |

Таблиця 3.14

Кількість операцій за секунду на моделі придбання S0

| Тип запиту | Кількість вибраних записів | Час виконання, с. | Кількість операцій за секунду |
|------------|----------------------------|-------------------|-------------------------------|
| SQ 1 | 1001002 | 79 | 12671 |
| SQ 2 | 706 | 2 | 353 |
| SQ 3 | 201002 | 93 | 2161 |
| CQ 1 | 1001002 | 105 | 9533 |
| CQ 2 | 13827 | 54 | 256 |
| CQ 3 | 201002 | 103 | 1951 |

Таблиця 3.15

Кількість операцій за секунду на моделі придбання S3

| Тип запиту | Кількість вибраних записів | Час виконання, с. | Кількість операцій за секунду |
|------------|----------------------------|-------------------|-------------------------------|
| SQ 1 | 1001002 | 73 | 13712 |
| SQ 2 | 706 | 1 | 706 |
| SQ 3 | 201002 | 99 | 2030 |
| CQ 1 | 1001002 | 97 | 10320 |
| CQ 2 | 13827 | 43 | 322 |
| CQ 3 | 201002 | 54 | 3722 |

Таблиця 3.16

Кількість операцій за секунду на моделі придбання S7

| Тип запиту | Кількість вибраних записів | Час виконання, с. | Кількість операцій за секунду |
|------------|----------------------------|-------------------|-------------------------------|
| SQ 1 | 1001002 | 53 | 18887 |
| SQ 2 | 706 | 1 | 706 |
| SQ 3 | 201002 | 74 | 2716 |
| CQ 1 | 1001002 | 84 | 11917 |
| CQ 2 | 13827 | 37 | 374 |
| CQ 3 | 201002 | 42 | 4786 |

3.5. Візуалізація результатів експериментальних досліджень засобами Microsoft Azure

3.5.1. Огляд засобів візуалізації Microsoft Azure

Azure Monitor показники — це серія вимірних значень і підрахунків, які збираються та зберігаються протягом певного часу. Показники можуть бути стандартними (також званими «платформною») або користувацькими.

Стандартні показники надаються платформною Azure. Вони відображають працездатність і статистику використання ресурсів Azure.

Засіб вибору обсягу ресурсу дозволяє переглядати показники для окремих ресурсів і кількох ресурсів.

Для вибору області візуалізації потрібно обрати ресурси для відображення результатів (рис. 3.7).

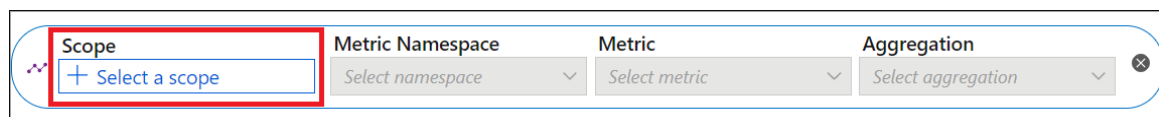


Рис. 3.12. Вибір області візуалізації

Деякі типи ресурсів можуть запитувати показники для кількох ресурсів. Ресурси мають бути в межах однієї підписки та однієї локації.

У провіднику метрик Azure можна створювати діаграми, які відображають кілька ліній метрик або показують кілька діаграм показників одночасно.

Ця функція дозволяє:

- Корелюйте пов'язані показники на одному графіку, щоб побачити, як одне значення пов'язане з іншим.
- Відображати в безпосередній близькості показники, які використовують різні одиниці виміру.
- Візуально об'єднувати та порівнювати показники з кількох ресурсів.

ВИСНОВКИ

На сьогодні хмарні технології одна з найбільш поширених тем і напрямків розвитку в ІТ. Під поняттям хмара, хмарні обчислення та хмарні технології мається на увазі надання користувачу ресурсів та потужностей у вигляді інтернет-сервісу. Актуальність хмарних обчислень пов'язана зі зниженням витрат, масштабованістю і гнучкістю архітектури інформаційних технологій, а також є зручним інструментом для компаній які вирішили оптимізувати свої ресурси шляхом перенесення обчислювальних потужностей в хмарний ЦОД. Перехід від стандартної локальної інфраструктури до хмари значно зменшує капітальні витрати не тільки на обладнання, а й на експлуатаційні витрати зв'язані з технічним обслуговуванням та дає змогу використовувати необхідні ресурси по мірі необхідності. Саме таку платформу надає користувачам Microsoft Azure.

Для визначення найбільш економічно та практично вигідної моделі придбання DTU Azure для роботи з великими даними було проведено експеримент для порівняння рівнів моделей придбання, а саме:

Модель придбання S0;

Модель придбання S3;

Модель придбання S7.

В ході експерименту всі моделі придбання навантажувались однаковими типами запитів, а саме три простих запита та три складних.

В результатів експерименту було отримано порівняння навантаження на CPU, швидкість читання-запису, навантаження на DTU, ціни за виконання запитів.

Відомо, що якщо навантаження на DTU більше ніж 80% це означає що слід перейти на більш високу модель придбання.

В ході експерименту було досліджено, що під час генерації даних модель S0 та S3 показувала навантаження на DTU 100%, при тому модель придбання S7 близько 53%.

Під час виконання запитів тільки модель S0 переходила межу в 80% навантаження на DTU.

Таким чином виходячи з результатів експерименту можемо зробити висновок, що для генерації великої кількості даних рекомендовано використовувати модель придбання не нижче S7. Для виконання запитів до бази даних з великою кількістю запитів рекомендовано використовувати модель придбання не нижче S3.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. База данных SQL Azure [Электронный ресурс]. – Режим доступа: <https://azure.microsoft.com/ru-ru/services/sql-database/>
2. Microsoft SQL Server Management Studio Express [Электронный ресурс]. – Режим доступа: <https://www.microsoft.com/ru-ru/download/details.aspx?id=8961>
3. dbForge Data Generator for SQL Server [Электронный ресурс]. – Режим доступа: <https://www.devarit.com/dbforge/sql/data-generator/>
4. Download MySQL Workbench [Электронный ресурс]. – Режим доступа: <https://dev.mysql.com/downloads/workbench/>
5. Download HeidiSQL 10.1 [Электронный ресурс]. – Режим доступа: <https://www.heidisql.com/download.php>
6. Navicat [Электронный ресурс]. – Режим доступа: <https://www.navicat.com/en/products>
7. PL/SQL Developer [Электронный ресурс]. – Режим доступа: https://www.allroundautomations.com/download.html?gclid=CjwKCAjw27jnBRBuEiwAdjQXDKEev6BxtoUkKR00rcto5UImCkVoPBKxe1v-gaxFnSvItp4ntvLLCRoC85oQAvD_BwE
8. Database .NET [Электронный ресурс]. – Режим доступа: <https://marketplace.visualstudio.com/items?itemName=fishcodelib.DatabaseNET>
9. Aqua Data Studio [Электронный ресурс]. – Режим доступа: https://www.aquafold.com/aquadatastudio_downloads
10. DTM Data Generator [Электронный ресурс]. – Режим доступа: <http://www.sqledit.com/dg/download.html>
11. Visual Studio Data Generator [Электронный ресурс]. – Режим доступа: <https://www.dutchdatadude.com/generating-perfect-test-data-using-visual-studio-and-custom-data-generators/>
12. GEDIS Studio [Электронный ресурс]. – Режим доступа: <https://gedis-studio.download3000.com/>
13. Mockaroo [Электронный ресурс]. – Режим доступа: <https://mockaroo.com/>
14. ApexSQL Generate [Электронный ресурс]. – Режим доступа: <https://www.apexsql.com/sql-tools-generate.aspx>
15. Upscene Advanced Data Generator [Электронный ресурс]. – Режим доступа: <https://www.sqlmanager.net/ru/products/mysql/datagenerator>

16. EMS Data Generator for MySQL [Электронный ресурс]. – Режим доступа: <https://www.sqlmanager.net/ru/products/mysql/datagenerator>
17. GS Data Generator [Электронный ресурс]. – Режим доступа: <https://www.gsapps.com/products/datagenerator/>
18. Spawner Data Generator [Электронный ресурс]. – Режим доступа: <https://sourceforge.net/projects/spawner/>
19. generatedata.com [Электронный ресурс]. – Режим доступа: <https://www.generatedata.com/>
20. 15 Best Test Data Generation Tools In 2019 [Электронный ресурс]. – Режим доступа: <https://www.rankred.com/test-data-generation-tools/>
21. DE Кнuth. Мистецтво комп'ютерного програмування, том 2: напівчислові алгоритми . Аддісон-Веслі, Редінг, Массачусетс, третє видання, 1997.
22. Random Class [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/en-us/dotnet/api/system.random?view=net-6.0>
23. Azure Monitor SQL insights [Электронный ресурс] – Режим доступа: <https://azure.microsoft.com/en-us/updates/azure-monitor-sql-insights-for-azure-sql-in-public-preview/>
24. Azure Monitor [Электронный ресурс] – Режим доступа: <https://azure.microsoft.com/ru-ru/services/monitor/#overview>
25. Choose between the vCore and DTU [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/en-us/azure/azure-sql/database/purchasing-models>
26. DTU and vCore based models for Azure SQL Databases [Электронный ресурс] – Режим доступа: <https://www.sqlshack.com/dtu-and-vcore-based-models-for-azure-sql-databases/>
27. Генератор псевдослучайных чисел [Электронный ресурс] – Режим доступа: [https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_%D0%BF%D1%81%D0%B5%D0%B2%D0%B4%D0%BE%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D1%85_%D1%87%D0%B8%D1%81%D0%B5%D0%BB#:~:text=%D0%93%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80%20%D0%BF%D1%81%D0%B5%D0%B2%D0%B4%D0%BE%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D1%85%20%D1%87%D0%B8%D1%81%D0%B5%D0%BB%20\(%D0%93%D0%9F%D0%A1%D0%A7%2C%20%D0%B0%D0%BD%D0%B3%D0%BB,%D0%B7](https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_%D0%BF%D1%81%D0%B5%D0%B2%D0%B4%D0%BE%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D1%85_%D1%87%D0%B8%D1%81%D0%B5%D0%BB#:~:text=%D0%93%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80%20%D0%BF%D1%81%D0%B5%D0%B2%D0%B4%D0%BE%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D1%85%20%D1%87%D0%B8%D1%81%D0%B5%D0%BB%20(%D0%93%D0%9F%D0%A1%D0%A7%2C%20%D0%B0%D0%BD%D0%B3%D0%BB,%D0%B7)

<https://docs.microsoft.com/ru-ru/azure/sql-database/sql-database-dtu-resource-limits-single-databases>

28. Ограничения ресурсов для одиночных баз данных в модели приобретения на основе единиц DTU [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/azure/sql-database/sql-database-dtu-resource-limits-single-databases>

29. Resource limits for single databases using the DTU-based purchasing model [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-dtu-resource-limits-single-databases>

30. Azure SQL Database DTU Calculator [Электронный ресурс]. – Режим доступа: <http://dtucalculator.azurewebsites.net/>

31. Intelligent Insights using AI to monitor [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/en-us/azure/azure-sql/database/intelligent-insights-overview>

32. Azure Activity Log [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/en-us/azure/azure-monitor/essentials/activity-log>

33. Metrics supported by Azure Event Grid [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/en-us/azure/event-grid/metrics>

34. Автоматичне масштабування Azure [Электронный ресурс] – Режим доступа: <https://azure.microsoft.com/ru-ru/features/autoscale/>

35. Service Health [Электронный ресурс] – Режим доступа: <https://azure.microsoft.com/en-us/features/service-health/>

ДОДАТКИ

Додаток А

Вихідний код програми для генерації псевдовипадкових даних.

Клас Customer

```
using System;

namespace DataGenerator.Models
{
    public class Customer
    {
        public Customer(string eMail, string password, DateTime creationDate)
        {
            CustomerId = Guid.NewGuid();
            EMail = eMail;
            Password = password;
            CreationDate = creationDate;
        }

        public Guid CustomerId { get; set; }
        public string EMail { get; set; }
        public string Password { get; set; }
        public DateTime CreationDate { get; set; }
    }
}
```

Клас Permission

```
using System;

namespace DataGenerator.Models
{
    public class Permission
    {
        public Permission(Guid customerId)
        {
            PermissionId = Guid.NewGuid();
            CustomerId = customerId;
        }
        public Guid PermissionId { get; set; }
        public Guid CustomerId { get; set; }
        public bool CanRead { get; set; }
        public bool CanWrite { get; set; }
        public bool CanUpdate { get; set; }
    }
}
```

Клас SqlProvider

```
using DataGenerator.Models;
using System;
using System.Data.SqlClient;
using System.Data.SqlTypes;
using System.Threading.Tasks;

namespace DataGenerator
{
    public class SqlProvider
    {
        private string connectionString;
```

```

public SqlProvider(string connectionString)
{
    this.connectionString = connectionString;
}

public async Task ExecuteQueryAsync(string query)
{
    try
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            connection.Open();
            using (SqlTransaction trans = connection.BeginTransaction())
            using (SqlCommand cmd = connection.CreateCommand())
            {
                cmd.Transaction = trans;
                cmd.CommandText = query;
                await cmd.ExecuteNonQueryAsync();
                trans.Commit();
            }
        }
    }
    catch (Exception ex)
    {
    }
}

public async Task AddCustomerAsync(Customer customer)
{
    try
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            string sqlExpression = $"INSERT INTO Customers (Id, Email, Password,
CreationDate) " +
                $"VALUES ('{customer.CustomerId}', {customer.Email},
{customer.Password}, {new SqlDateTime(customer.CreationDate).ToSqlString()})";
            connection.Open();
            using (SqlTransaction trans = connection.BeginTransaction())
            using (SqlCommand cmd = connection.CreateCommand())
            {
                cmd.Transaction = trans;
                cmd.CommandText = @"INSERT INTO Customers (CustomerId, Email,
Password, CreationDate) VALUES (@CustomerId, @Email, @Password, @CreationDate)";
                cmd.Parameters.AddWithValue("@CustomerId", customer.CustomerId);
                cmd.Parameters.AddWithValue("@Email", customer.Email);
                cmd.Parameters.AddWithValue("@Password", customer.Password);
                cmd.Parameters.AddWithValue("@CreationDate", customer.CreationDate);
                await cmd.ExecuteNonQueryAsync();
                trans.Commit();
            }
        }
    }
    catch (Exception ex)
    {
    }
}

public async Task AddPermissionAsync(Permission permission)
{
    try
    {
        using (SqlConnection connection = new SqlConnection(connectionString))

```



```

    var password = RandomString(14);
    var creationDate = DateTime.Now;
    return new Customer(eMail, password, creationDate);
}

private Permission CreatePemission(Guid customerGuid)
{
    var result = new Permission(customerGuid);
    var randomPermission = random.Next(1, 3);
    switch (randomPermission)
    {
        case 1:
        {
            result.CanUpdate = true;
            break;
        }
        case 2:
        {
            result.CanRead = true;
            break;
        }
        case 3:
        {
            result.CanWrite = true;
            break;
        }
    }
    return result;
}

private string RandomString(int length)
{
    const string chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
    return new string(Enumerable.Repeat(chars, length)
        .Select(s => s[random.Next(s.Length)]).ToArray());
}

private async void GenerateButton_Click(object sender, RoutedEventArgs e)
{
    if (int.TryParse(GenerationCountTextBox.Text, out generationCount))
    {
        var startDate = DateTime.Now;
        IterationCountTextBlock.Text = generationCount.ToString();
        CreateProgress(generationCount);
        await GenerateRecord(generationCount);
        MessageBox.Show("Времени затрачено: " + (DateTime.Now -
startDate).ToString(), "Завершено", MessageBoxButton.OK);
    }
}

private async Task GenerateRecord(int count = 1)
{
    await Task.Run(async () =>
    {
        for (int i = 0; i <= count; i++)
        {
            var customer = CreateCustomer();
            var permission = CreatePemission(customer.CustomerId);
            await WriteAsync(customer, permission);
            UpdateProgress(i);
        }
    });
}

private void CreateProgress(int maximum)

```

```

    {
        GenerationProgressBar.Maximum = maximum;
        GenerationProgressBar.Minimum = 0;
    }

    private void UpdateProgress(int value)
    {
        Dispatcher.Invoke(() =>
        {
            GenerationProgressBar.Value = value;
            CurrentIterationTextBlock.Text = value.ToString();
        });
    }

    private async Task WriteAsync(Customer customer, Permission permission)
    {
        await sqlProvider.AddCustomerAsync(customer);
        await sqlProvider.AddPermissionAsync(permission);
    }

    private async void Button_Click(object sender, RoutedEventArgs e)
    {
        string query = "SELECT * FROM Customers";
        List<Task> taskList = new List<Task>();
        for (int i = 0; i <= 10; i++)
        {
            taskList.Add(sqlProvider.ExecuteQueryAsync(query));
        }
        await Task.WhenAll(taskList);
    }
}
}
}

```

```

<Window x:Class="DataGenerator.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:DataGenerator"
        mc:Ignorable="d"
        Title="MainWindow" Height="450" Width="800">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="50"/>
            <RowDefinition Height="*/>
        </Grid.RowDefinitions>
        <StackPanel Orientation="Horizontal"
            Margin="10">
            <TextBlock Text="Кол-во генераций: "/>
            <TextBox x:Name="GenerationCountTextBox"
                Width="150"
                Height="20"
                VerticalAlignment="Top"/>
            <Button Name="GenerateButton"
                Content="Начать генерацию"
                Height="20"
                VerticalAlignment="Top"
                Margin="5,0,0,0" Click="GenerateButton_Click"/>
            <Button Content="Бомбёжка"
                Click="Button_Click"/>
        </StackPanel>
        <Grid Grid.Row="1"
            Margin="5">
            <Grid.RowDefinitions>
                <RowDefinition Height="20"/>
            </Grid.RowDefinitions>
        </Grid>
    </Grid>

```

```

        <RowDefinition Height="*" />
    </Grid.RowDefinitions>
    <StackPanel x:Name="TextPanel" Orientation="Horizontal">
        <TextBlock Name="CurrentIterationTextBlock" />
        <TextBlock Text=" запись из " />
        <TextBlock Name="IterationCountTextBlock" />
    </StackPanel>

    <ProgressBar Name="GenerationProgressBar"
        Height="25"
        Grid.Row="1"
        VerticalAlignment="Top" />

</Grid>

</Grid>
</Window>

```

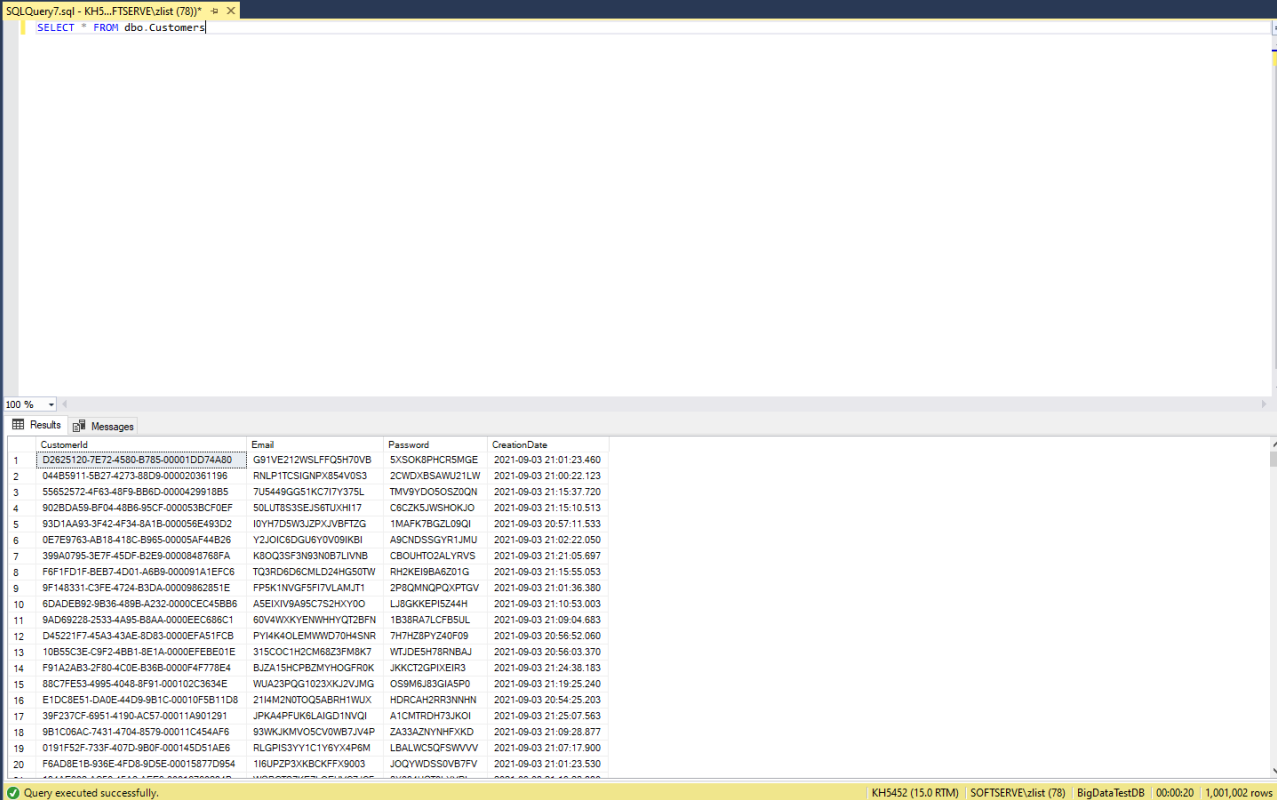
App.Config

```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
</configuration>

```

Додаток Б



| CustomerId | Email | Password | CreationDate |
|--------------------------------------|----------------------|----------------|-------------------------|
| D2625120-7E72-4580-B785-00001DD74A80 | G91VE212W5LFFQSH70VB | 5XSOK8PHCRSMGE | 2021-09-03 21:01:23.460 |
| 044B5911-5B27-4273-8BD9-000020361196 | RNLP1TCSIGNPX854V0S3 | 2CWDXBSAWJ21LW | 2021-09-03 21:00:22.123 |
| 55652572-4F63-48F9-BB6D-0000429918B5 | 7U5449GG51KC7I7Y375L | TMV9YDOSOSZ0QN | 2021-09-03 21:15:37.720 |
| 902BDA59-BF04-48B6-95CF-000053BCF0EF | S0LUT8S3SEJS6TUXHI17 | C6CZK5W5HOKJO | 2021-09-03 21:15:10.513 |
| 93D1AA93-3F42-4F34-8A1B-000056E493D2 | IDYH7D5W3JZPXJVBTZG | 1MAFK78GL09QI | 2021-09-03 20:57:11.533 |
| 0E7E9763-AB18-418C-B965-000056A44B26 | V2J0IC6DGEV0V09IKBI | A9CNDSSGYR1JMU | 2021-09-03 21:02:22.050 |
| 398A0795-3E7F-45DF-B2E9-0000848768FA | K8OQ3SF3N83NOB7LIVNB | CB0UHTO2ALYRVS | 2021-09-03 21:21:05.697 |
| F8F1FD1F-BEB7-4D01-46B9-000091A1EFC6 | TQ3RD6D6CMLD24HG50TW | RH2KEI9BA6Z01G | 2021-09-03 21:15:55.053 |
| 9F148331-33FE-4724-B3DA-00009862851E | FPSK1NVG5F17VLAJTI | 2P8QMNQ9XPTGV | 2021-09-03 21:01:36.380 |
| 6DADEB92-9B36-489B-A232-0000CEC45BB6 | ASEIXIV9A95C752HXKYO | LJ8GKKEPI5244H | 2021-09-03 21:10:53.003 |
| 9AD69228-2533-4A95-B8AA-0000CEC686C1 | 60V4WXKYENWHYQTZBPN | 1B38RA7LCFBSUL | 2021-09-03 21:09:04.683 |
| D45221F7-45A3-43AE-8D83-0000EFA51FCB | PVY4K4OLEMWW070H4SNR | 7H7HZ8PY240F09 | 2021-09-03 20:56:52.060 |
| 10B55C3E-C9F2-48B1-8E1A-0000EFEBE01E | 315COC1H2CM68Z3FM8K7 | WTJDESH78RNBAL | 2021-09-03 20:56:03.370 |
| F91A2AB3-2F80-4C0E-B368-0000F4F778E4 | BJZA15HCPBZMYHOGFR0K | JKKCT2GPXIEIR3 | 2021-09-03 21:24:38.183 |
| 88C7FE53-4995-4048-BF91-000102C3634E | WUA23PQG1023XKJZVJMG | OS9M6J83GIA5P0 | 2021-09-03 21:19:25.240 |
| E1DC8E51-DA0E-44D9-9B1C-00010F5B11D8 | 2144M2NOTOQ5ABRH1WUX | HDRCAH2RR3NHNH | 2021-09-03 20:54:25.203 |
| 39F237CF-6951-4190-AC57-00011A901291 | JKK44PFUK6LAIGD1NVQI | A1CMTRDH73JKOI | 2021-09-03 21:25:07.563 |
| 9B1C06AC-7431-4704-8579-00011C454AF6 | 93WKJKMV05CV0WB7JV4P | ZA33AZNYNHFXKD | 2021-09-03 21:09:28.877 |
| 0191F52F-733F-407D-9B0F-000145D51AE6 | RLGPIS3YV1C1YVY4P6M | LBALWCSQF5WVVV | 2021-09-03 21:07:17.900 |
| F6ADBE1B-936E-4FD8-9D5E-00015877D954 | 16UPZP3XKBCCKFFX9003 | JOQYWDS0VB7PV | 2021-09-03 21:01:23.530 |

Рис. Б.1. Результати виконання запиту SQL на локальному ресурсі

SQLQuery7.sql - KH5...FTSERVE\zlist (78) |

```
SELECT * FROM dbo.Customers WHERE Email LIKE 'A1X*'
```

100 %

Results Messages

| CustomerId | Email | Password | CreationDate |
|------------|---------------------------------------|-----------------------|-----------------|
| 1 | A17A31D4-F510-47F8-8F3F-03238CA2727C | A12N7CS120MKSCWVHVHY | 1G6ASKGZ2E099F |
| 2 | DD0A9F01-6DF9-4171-A4DC-03450A5A7821 | A1E1PPW8B0OPXZ6K7KH | OP3IFGBKGHOYSF |
| 3 | 5B799D5C-B9CC-4AEE-A108-0345773DA431E | A1712EIF3PLWNK91EKOW | WCPHLZVXTJ121M |
| 4 | 38E68A43-02EF-4C12-8363-03A199C26384 | A14ICG2YX9XV68GJDBE5 | IGTP9LTYH8101F |
| 5 | 27C0508D-58C8-4E5F-AD85-040E1ED6D13D | A15RREN5311JWC3W16J | V2BZVTC568KSH |
| 6 | E2BE9001-441D-4D12-ACB5-048076DF77FF | A1NXXKO2L4XCTV4Y1CH | VXIAWJZPW50Y4 |
| 7 | FACC97F5-29A7-4D0E-8CDF-04EB2AA4F3FC | A15D2SSALQEOMIFZTQ4M | RKYWL7ME9FNUE6 |
| 8 | C2AF350B-4EB8-40D6-96A8-05008A295AA | A1NISFVGOWS2NB5YIIF | JMSA1ZORMPHZJU |
| 9 | DACC93E9-2C6A-4260-AD1A-051F15842A51 | A1GYW9PHAKS1MYEG05MN | LV48PUNN83YTS |
| 10 | 002409AE-2A38-4855-A0C9-053A7967FC42 | A10NJA2DUSAFMCRVF46I | 85WHB0XJRPJ96D |
| 11 | 6AD8C609-32CE-4FC4-89A8-05682A7CF3A8 | A1NQXEUDBB2NU21AMQST | JUL13FZV5FPNOH |
| 12 | 5CB4703B-0AC4-4EA2-8DAE-0579A88B0607 | A1EB5KHZZJ9XN88M1AIZ | 7KLRT31BDKH0YF |
| 13 | 97D362D6-D020-423A-9A60-059585DB196A | A1LNMCCQD38C8WRSU80P7 | 2ZGK9GF90R0ITZ |
| 14 | 81AE1AFA-88BE-497F-A72E-059968125CBC | A1YP3KMF9YHBPEDOCF53 | IENKVDZ2R6W53F2 |
| 15 | 3384FC14-C08F-4F46-96DF-268B2206969A | A1XSR31BP2BYZ8ZV1PBF | 41UEWHLI8K5D01 |
| 16 | CB2555DE-6D10-4004-BEE7-2692198BAE1E | A1T2DL7JA74L4GRMV6W | 7L1L0UVWVAPALG |
| 17 | B8C7E699-91E2-43EE-8E2C-26C46C8B3702 | A1B44RNC6GLP5IDUCYMT | J1S3D401LVYL82 |
| 18 | DF197939-256A-486E-8F2C-26DE99C5339F | A1JLEKYWJUW4IHWQ5BO | RK8S9VHT59EG2J |
| 19 | 79A28A3C-F5FE-4A60-9FEB-2753932DAF0C | A1YT2OZYVLLJK7KPA0A | 9NEFXEJZCJ71Z |
| 20 | 83B69CE8-6365-45E8-9712-27A837A23FD2 | A1PLSYNPNPYYSLADHN1SG | W0IE7JGPQLL10S |

Query executed successfully. KH5452 (15.0 RTM) | SOFTSERVE\zlist (78) | BigDataTestDB | 00:00:00 | 706 rows

Рис. Б.2. Результати виконання запиту SQ2 на локальному ресурсі

SQLQuery7.sql - KH5...FTSERVE\zlist (78) |

```
SELECT * FROM dbo.Customers WHERE CreationDate < GETDATE()
```

100 %

Results Messages

| CustomerId | Email | Password | CreationDate |
|------------|--------------------------------------|----------------------|-----------------|
| 1 | D2625120-7E72-4580-B785-00001DD74A80 | G91VE212WSLFFQSH70VB | 5XSOK8PHCRSMGE |
| 2 | 044B5911-5B27-4273-88D9-000020361196 | RNL1PTCSIGNPX854V0S3 | 2CWDXB8AWUZ1LW |
| 3 | 55652572-4F63-48F9-BB6D-0000429918B5 | 7U5449GG51KC717Y375L | TMV9YDOSOS20QN |
| 4 | 902BDA59-8F04-48B6-95CF-000053BCFOEF | 50LUT8S3SEJ58TUHXH17 | C6CZKSJWSHOKJO |
| 5 | 93D1AA93-3F42-4F34-8A1B-000056E493D2 | I0YH7DSW3JZPKVBFZTG | 1MAFK7BQZL09QJ |
| 6 | 0E7E9763-AB18-418C-B965-00005AF44B26 | Y2J0IC8DGU6Y0V09IKBI | ABCNDS5GYR1JMU |
| 7 | 399A0795-3E7F-45DF-B2E9-0000848768FA | K8Q03SF3N93N087LVNB | CBOUHTO2ALYRVS |
| 8 | FF148331-C3FE-4724-B3DA-00009862851E | FP5K1NVGF5F7V1LAMI1 | 2P8QMNQPGQXPTGV |
| 9 | 6DADEB92-9B36-4898-A232-0000CEC45BB6 | A5EIXIV9A95C7S2HXV00 | LJ8GKKEPIS244H |
| 10 | 9AD69228-2533-4A95-B8AA-0000EEC866C1 | 60V4WXYENWHYQY2BFN | 1B38RA7LCF8SUL |
| 11 | D45221F7-45A3-43AE-8D83-0000EFA51FCB | PY14K4OLEMWNW07H4SNR | 7H7H28PV24F09 |
| 12 | 10B5C3E-C9F2-48B1-8E1A-0000EFEBE01E | 31SCOC1H2CM68Z3FM8K7 | W7JDESH78RNB4J |
| 13 | F91A2AB3-2F80-4C0E-B368-0000F4778E4 | BJ2A15HCPB2MYHOGFROK | JKKCTZGPXIEIR3 |
| 14 | 88C7FE53-4995-4048-8F91-000102C3634E | WUA23PGQ1023KJZVJMG | OS9M6J83GIA5P0 |
| 15 | E1DC8E51-DA0E-44D9-9B1C-00010F5B11D8 | 214M2NTOQ5ABRH1WUX | HDRCAH2RR3NNHN |
| 16 | 39F237CF-6951-4190-AC57-00011A901291 | JKK44PFUK6LAIGD1NVQJ | A1CMTRDH73JKOI |
| 17 | 9B1C06AC-7431-4704-8579-00011C454AF6 | 93WKJMKVOSCV0WB7JV4P | ZA33AZNYNHFKKD |
| 18 | 0191F52F-733F-407D-980F-000145D51AE6 | RLG9IS3Y1C1Y8YX4P6M | LBALWCQFSWVVVV |
| 19 | FDAD8E1B-936E-4FD8-9D5E-00015877D954 | 1I6UPZP3XKBCFFX9003 | JQOYWDSS0VB7FV |

Query executed successfully. KH5452 (15.0 RTM) | SOFTSERVE\zlist (78) | BigDataTestDB | 00:00:15 | 1,001,002 rows

Рис. Б.3. Результати виконання запиту SQ3 на локальному ресурсі

SQLQuery7.sql - KHS...FTSERVE\zlist (78)*

```
SELECT * FROM dbo.Customers AS cus INNER JOIN dbo.Permissions AS per
ON cus.CustomerId = per.CustomerId
```

| CustomerId | Email | Password | CreationDate | PermissionId | CustomerId | CanRead | CanWrite | CanUpdate |
|---------------------------------------|----------------------|-----------------|-------------------------|--------------------------------------|---------------------------------------|---------|----------|-----------|
| 186DFCC0-FB38-445C-9268-157B3A3F6C2C | V5SRFRNMCSN158MYCOA | J8A4Z5P1W9HCG1 | 2021-09-03 21:00:58.920 | 0F734A0E-8F1-4944-B0EB-FE0710F72B89 | 186DFCC0-FB38-445C-9268-157B3A3F6C2C | 0 | 0 | 1 |
| 14D39C74-59E6-4D56-A4FC-157B4C64FCFC | W0H8BJ30AOHWKZT8Z2YK | NIUBIC10L3M8EC | 2021-09-03 21:10:29.873 | 9F22A1F0-63E6-4FBC-86A3-FDC9F3333E24 | 14D39C74-59E6-4D56-A4FC-157B4C64FCFC | 0 | 0 | 1 |
| 22707F42-E3E5-46CF-AA12-157B610838D0 | Q520CCLSOENUK05ERSRW | RZYU4WH1HADVSM | 2021-09-03 21:18:22.917 | 992216D7-887A-4A5E-84F2-19CA4A8FD283 | 22707F42-E3E5-46CF-AA12-157B610838D0 | 1 | 0 | 0 |
| 82D78DEA-AE47-42B1-AB6F-157B670A0FDD | P3750UBW577PV8SHW43 | EWARKLC04C052D | 2021-09-03 21:06:30.210 | 0E7A210E-1C38-4B88-80B1-D68D7C1D5313 | 82D78DEA-AE47-42B1-AB6F-157B670A0FDD | 1 | 0 | 0 |
| E42F8F8E-0C8E-438C-A866-157B66674E31 | L7KLZNR1F333BEAHAM | 4QEMF7J87U98TV | 2021-09-03 21:08:27.057 | DA8897CD-67AF-4A4E-2A4D-9FF59CA39C64 | E42F8F8E-0C8E-438C-A866-157B66674E31 | 1 | 0 | 0 |
| 7443F813-674F-4474-9E92-157B903F8859 | ITPC340EJFEQUBQ823 | W81M077P9LQ9M2 | 2021-09-03 21:25:25.090 | ED95C81F-6F8E-44EC-8D09-107CEB422C7 | 7443F813-674F-4474-9E92-157B903F8859 | 0 | 0 | 1 |
| 82E5199F-58DF-48FD-89C7-157BAE2F7841 | D7574JGMX6L9V9FYMD | ZWK10CNHVU1LGD | 2021-09-03 21:10:38.600 | 32349F05-9D1B-402F-89D1-C077885C8787 | 82E5199F-58DF-48FD-89C7-157BAE2F7841 | 0 | 0 | 1 |
| 4300D3A4-D5AF-4509-8A67-157B3C52C718 | 9Q7A8ESQYQKLQXKSD0N | Z3D5A2W00ESIT | 2021-09-03 21:18:37.960 | F57BF3CE-574F-672C-91D7-F4907291276D | 4300D3A4-D5AF-4509-8A67-157B3C52C718 | 1 | 0 | 0 |
| E3785D90-855E-4891-A61E-157BCED074E0 | PA39J5N3EY9C9R9PSUJ | TAE11WY0MSSKLS5 | 2021-09-03 20:57:19.540 | EC2A1C8E-217A-44DD-3205-C09A8F121ECB | E3785D90-855E-4891-A61E-157BCED074E0 | 1 | 0 | 0 |
| 8410F95E-37CF-4C8D-AC3D-157BDAF68B90 | NS3991P47RNM70JCSV5S | XXC2MTE8E6Z2WV | 2021-09-03 21:16:29.713 | C8CA164-0184-47DF-9EDF-542E1549724C | 8410F95E-37CF-4C8D-AC3D-157BDAF68B90 | 1 | 0 | 0 |
| C0671002-9731-44F9-82A5-157BFCC7703D | N4R10EVM5GZSRWES21C | 0PFFZJ1D1VHT0E | 2021-09-03 20:59:26.457 | 2305D03-347A-4385-4A66-FDFC868D48E | C0671002-9731-44F9-82A5-157BFCC7703D | 1 | 0 | 0 |
| 73FD4EAS-84DE-42AE-8688-157C0019D0505 | K7591JRGUJ1CYP08K3V | 0PFC5K7N2E5EY4 | 2021-09-03 20:55:51.070 | EACFEA4A-8BD4-4235-858C-62B068908616 | 73FD4EAS-84DE-42AE-8688-157C0019D0505 | 0 | 0 | 1 |
| 89F39B7F-A206-4844-A941-157C19F80949 | NDR0K6E90A0MG8E71V8 | SJL8R0Y0WJ30GF | 2021-09-03 20:54:46.993 | D83C1E16-1C5D-486E-9041-9FEA82880671 | 89F39B7F-A206-4844-A941-157C19F80949 | 0 | 0 | 1 |
| A64C7E46-8F4F-4A87-85EA-157C543D3237 | 2ANVYSG0T4EYW45MUFA | 34R8EJLZDU0WB | 2021-09-03 20:58:08.537 | 02838970E1B-487F-8A81-D65207D838F9 | A64C7E46-8F4F-4A87-85EA-157C543D3237 | 1 | 0 | 0 |
| B58CFDC5-22ED-4EF1-878E-157C3058E435 | IZM7DGLTR10WET9Q8A | X0IF5F5NC6W8A1 | 2021-09-03 21:27:37.153 | 4F5D0C5-36C3-41EE-9088-E73E8475A4F6 | B58CFDC5-22ED-4EF1-878E-157C3058E435 | 1 | 0 | 0 |
| 1D8A7D8A-39D7-43C3-A32F-157C32FEF10 | 0GUFY3UBP8DJ8Y39I | A1JFUS4700MDC | 2021-09-03 21:18:13.123 | 81F84E87-0053-4232-9508-8E34EC1E00F9 | 1D8A7D8A-39D7-43C3-A32F-157C32FEF10 | 0 | 0 | 1 |
| 64E85771-22DC-4088-A953-157C38EA7F58 | 9C3N0M62ZZ0IN8TZ255X | Q2HEL3323E6K6W | 2021-09-03 21:20:30.473 | 4725C98E-0881-4CAF-9D03-9425C9D02BDF | 64E85771-22DC-4088-A953-157C38EA7F58 | 1 | 0 | 0 |
| 018A5314-953F-4D4C-8780-157C5239242E | 32VST7C5C3Z0R1CXKO | M8BWS9PFFZTA8K | 2021-09-03 21:03:39.250 | 8489D8F0-4576-11B2-B441-56A000165DEA | 018A5314-953F-4D4C-8780-157C5239242E | 1 | 0 | 0 |
| 7E8E6E2C-4D5D-4131-A072-157C5243CB4E | 67NWVYF9P8FY2B38E2M | 3NEH5TU6V0FCH | 2021-09-03 21:20:20.533 | EB416775-6A04-4288-8489-8FFE21D8625F | 7E8E6E2C-4D5D-4131-A072-157C5243CB4E | 0 | 0 | 1 |
| 2E1C2137-E14D-4171-85DB-157C5F9A7832 | 79KXDSQAV096ELLZD5DU | YEU8VCP08BA5X | 2021-09-03 21:18:36.433 | DEF8989C-C144-9338-9C87-893313A3895E | 2E1C2137-E14D-4171-85DB-157C5F9A7832 | 0 | 0 | 1 |

Query executed successfully. KH5452 (15.0 RTM) SOFTSERVE\zlist (78) BigDataTestDB 00:00:18 1,001,002 rows

Рис. Б.4. Результати виконання запиту CQ1 на локальному ресурсі

SQLQuery7.sql - KHS...FTSERVE\zlist (78)*

```
SELECT * FROM dbo.Customers AS cus INNER JOIN dbo.Permissions AS per
ON cus.CustomerId = per.CustomerId WHERE cus.Password LIKE 'A%' AND per.CanRead = 1
```

| CustomerId | Email | Password | CreationDate | PermissionId | CustomerId | CanRead | CanWrite | CanUpdate |
|--------------------------------------|----------------------|-----------------|-------------------------|--------------------------------------|--------------------------------------|---------|----------|-----------|
| 2A49D908-3452-45CB-A688-0D103218539B | HE1CNS2DOWGNP81BHN3C | AEJQL30WVGHDHP | 2021-09-03 21:29:08.940 | FC431217-F3AF-49D6-ACA9-25BEE0C2002D | 2A49D908-3452-45CB-A688-0D103218539B | 1 | 0 | 0 |
| 7D543674-5703-4D08-1E8F-F4155A8E0526 | UJ0587QVJPD1G8LZHRGU | A50P8WZJVYMKD2 | 2021-09-03 21:28:08.083 | A6604E98-A014-4D08-AF94-25BFAF7B82DE | 7D543674-5703-4D08-1E8F-F4155A8E0526 | 0 | 0 | 0 |
| 78C7908A-2A72-41F8-8270-AF508A54C065 | TX0JZJGYPUTOSV9RE1Z | A1JVMEX53QVYVF | 2021-09-03 21:24:51.743 | 8658E7A0-50EA-467F-BC7C-25C39A186E6A | 78C7908A-2A72-41F8-8270-AF508A54C065 | 1 | 0 | 0 |
| 91B34C1F-EEF8-4D41-8FEA-421A7EC8BCF7 | 5GHNV2GGPAQLO8ZNGO | AU1CLEH2J0T7M | 2021-09-03 20:53:27.873 | E535F8A0-B8C0-4310-ABCD-25CDE7C6678E | 91B34C1F-EEF8-4D41-8FEA-421A7EC8BCF7 | 1 | 0 | 0 |
| B61EA12B-2AE4-4239-A61D-73D91E367D8E | CAHGA1Y8BR8RONK8H098 | AU14NH8DCKPK8LQ | 2021-09-03 21:00:32.463 | 0F989C7F-43E8-4F87-962F-25CF1F629344 | B61EA12B-2AE4-4239-A61D-73D91E367D8E | 1 | 0 | 0 |
| 99E3A8A3-BEB9-45ED-BC68-C457F155DC8F | SEL7JXUPT8KBK8W5VW0B | AWMP1G4N1RTOFK | 2021-09-03 21:03:03.757 | E484DEBB-A47C-4C2A-A025-25D00AD03F80 | 99E3A8A3-BEB9-45ED-BC68-C457F155DC8F | 1 | 0 | 0 |
| 9609547B-A825-4FB2-9727-CA659C673A1C | AH381ALGW5GATPK08PD | AIE40E2Q4ND3CF | 2021-09-03 20:54:23.327 | 59D447F0-2C57-4F8E-A012-25D0AD06D713 | 9609547B-A825-4FB2-9727-CA659C673A1C | 1 | 0 | 0 |
| 654A1305-D875-4118-8354-586488109F55 | 7RSLRBITU7JUJEEY9TX | AYPVO1E3HA4FEC | 2021-09-03 21:18:17.603 | D0DE514D-5F95-410E-AE31-25D3F8965F7 | 654A1305-D875-4118-8354-586488109F55 | 1 | 0 | 0 |
| 9B0484D9-0A8F-4C00-A8E3-299208498471 | 57GCVLA1P2ZAV68F1KZ | ACRFM20CEXTAK | 2021-09-03 21:07:38.173 | 0286D28D-8F4F-4D77-BFF5-25DF59CEDCA5 | 9B0484D9-0A8F-4C00-A8E3-299208498471 | 1 | 0 | 0 |
| A278F16E-147F-4E6D-AFD7-2567D6258BBC | 6JMBWHPZUSI033NR0IS | AEQ2I6S7G58LSF | 2021-09-03 21:06:00.227 | 6FE777E-A055-4F8D-B1-CC-25E375C4742A | A278F16E-147F-4E6D-AFD7-2567D6258BBC | 1 | 0 | 0 |
| AS6C24D-54AE-4CD7-82CE-47F68D7C2CF | 805VFAEY7SAO93IDE4J | AZU177JCOHTFO | 2021-09-03 21:09:31.793 | D762F18E-9182-4B84-95DB-25E66803FEF0 | AS6C24D-54AE-4CD7-82CE-47F68D7C2CF | 1 | 0 | 0 |
| A3AC8B89-5625-40BC-A7BD-7107A55E9FFE | 6J0NMVBOQVAH17RUL2E | A7C2VNC03QWPC | 2021-09-03 21:07:12.473 | 9B078AC-83E3-4314-ADP6-25E8A8413518 | A3AC8B89-5625-40BC-A7BD-7107A55E9FFE | 1 | 0 | 0 |
| 8AE6AF6B-3715-462D-876E-2A6D1178E87D | ENB03QYVWZJJP8IA13Z | A573Q4ZLN1K9DI | 2021-09-03 21:26:31.380 | 5F0623A0-ESA1-4D59-A72C-25ED17D83118 | 8AE6AF6B-3715-462D-876E-2A6D1178E87D | 1 | 0 | 0 |
| A868CF2D-B153-45D6-853E-BD448681E0BD | YJ8IFJCK6G8OR411HT | AC0S0176VQVLU | 2021-09-03 21:26:56.813 | 4480C476-01C4-4C60-894E-25F4752DC429 | A868CF2D-B153-45D6-853E-BD448681E0BD | 1 | 0 | 0 |
| 339FEF89-4AD3-454C-954B-2AC38C71F44D | 1TX3SF8FPXU40LUB897M | AOC770TLJ82Y72 | 2021-09-03 20:57:55.490 | 64945563-32CA-44F2-A9F3-25F8C93425BC | 339FEF89-4AD3-454C-954B-2AC38C71F44D | 1 | 0 | 0 |
| 3C29A53-0F8E-452D-8CDD-1EF873223133 | QBZDKNDU0TSJXJ0ZV1U | A3LW4H7SAFYECZ | 2021-09-03 21:14:51.147 | D51892E9-2BD5-479C-9A0D-25FF0EF584D4 | 3C29A53-0F8E-452D-8CDD-1EF873223133 | 1 | 0 | 0 |
| 39A1A180-875E-4A08-BFDA-1CA6C3826C91 | 9KXTSRS9S3D0H20J7GG | ABUNVOSSCW6P7G | 2021-09-03 20:54:26.567 | 9EF1CD49-FCD3-42C0-9E7D-26011D2114DF | 39A1A180-875E-4A08-BFDA-1CA6C3826C91 | 1 | 0 | 0 |
| 58D084F-AA3C-4B6A-B399-CEB91CD38E99 | VMVZ2KKH8RRFMWTRK30 | AQM0RXZLD38CSB | 2021-09-03 21:02:38.900 | 40DFA1C8-9828-4C2C-842F-26077F7887EB | 58D084F-AA3C-4B6A-B399-CEB91CD38E99 | 1 | 0 | 0 |
| C59594DF-28D0-4209-9530-C9CCCD1455EC | 0919VKSOGFCR2NHA19U | ABJQXKFLMXBOTQ | 2021-09-03 21:05:12.110 | 688BDA68-7698-4B39-81FF-260999908648 | C59594DF-28D0-4209-9530-C9CCCD1455EC | 1 | 0 | 0 |
| 1FC78FA4-36C1-48F4-A21D-6EC94DA955C4 | 1YUWFE831ELMXDYQK8L | AFS2JMVZYBDDOC | 2021-09-03 20:55:05.460 | 69D412E8-6AC7-48EF-B23D-260A06BE9F85 | 1FC78FA4-36C1-48F4-A21D-6EC94DA955C4 | 1 | 0 | 0 |

Query executed successfully. KH5452 (15.0 RTM) SOFTSERVE\zlist (78) BigDataTestDB 00:00:00 13,827 rows

Рис. Б.5. Результати виконання запиту CQ2 на локальному ресурсі

SQLQuery7.sql - KH5...FTSERVER\elist (78) *

```
SELECT * FROM dbo.Customers AS cus LEFT JOIN dbo.Permissions AS per
ON cus.CustomerId = per.CustomerId WHERE cus.CreationDate < GETDATE()
```

100 %

Results Messages

| | CustomerId | Email | Password | CreationDate | PermissionId | CustomerId | CanRead | CanWrite | CanUpdate |
|----|--------------------------------------|-----------------------|----------------|-------------------------|--------------------------------------|--------------------------------------|---------|----------|-----------|
| 1 | D2625120-7E72-4580-8785-00001DD74A80 | G91VEZ12WSLFFQSH70VB | SXSOK8PHCRSMGE | 2021-09-03 21:01:23.460 | C55F1719-8788-4371-AA38-0D38A48F2113 | D2625120-7E72-4580-8785-00001DD74A80 | 0 | 0 | 1 |
| 2 | 04485911-5B27-4273-8B09-000020361196 | RNLPTCSIGNPX854V053 | 2CWDXBSAWU21LW | 2021-09-03 21:00:22.123 | D4020131-0D2F-4A1E-BBFA-6A29012FA240 | 04485911-5B27-4273-8B09-000020361196 | 0 | 0 | 1 |
| 3 | 55652572-4F63-48F9-BB6D-000042991885 | 7U5449GG51KC7I7Y375L | TMV9YDOSOSZQDN | 2021-09-03 21:15:37.720 | 71ABBA1C-D488-456F-BDBB-7EC0174E6B51 | 55652572-4F63-48F9-BB6D-000042991885 | 0 | 0 | 1 |
| 4 | 9028DA59-8F04-4886-95CF-000053BCFOEF | S0LUT853SEJ58TUXH17 | C6CZKSJWSHOKJO | 2021-09-03 21:15:10.513 | 48F05A28-467F-42DC-B1FD-324681F908A3 | 9028DA59-8F04-4886-95CF-000053BCFOEF | 0 | 0 | 1 |
| 5 | 93D1AA93-3F42-4F34-8A1B-000056E493D2 | I0YH7D5W3ZPXJVBP7ZG | 1MAFK7BGZL09QI | 2021-09-03 20:57:11.533 | D9A701DB-560C-45F3-96AE-33076514D0E2 | 93D1AA93-3F42-4F34-8A1B-000056E493D2 | 1 | 0 | 0 |
| 6 | 0E7E9783-AB18-418C-B965-00005AF44B26 | Y2JOIC8DGu8Y0V09IKBI | ABCNDSGGR1JMU | 2021-09-03 21:02:22.050 | DA7C39A5-001E-4488-A142-E398AFB0799F | 0E7E9783-AB18-418C-B965-00005AF44B26 | 1 | 0 | 0 |
| 7 | 399A0795-3E7F-45DF-B2E9-0000848768FA | K8Q3Q3F3N83N0B7LIVNB | CB0UHTO2ALYRVS | 2021-09-03 21:21:05.697 | 1E41BB17-0E1C-48FF-8A94-1787C2B63C28 | 399A0795-3E7F-45DF-B2E9-0000848768FA | 1 | 0 | 0 |
| 8 | F8F1FD1F-BEB7-4D01-A869-000091A1EFC8 | TQ3RD6D6CMLD24HG50TW | RH2KEI9BA6201G | 2021-09-03 21:15:55.053 | 16958EA8-78D6-4E42-AE36-8D3353928649 | F8F1FD1F-BEB7-4D01-A869-000091A1EFC8 | 0 | 0 | 1 |
| 9 | 9F148331-C3FE-4724-83DA-00009862851E | FFSK1NVGFSF7VFLAMJT1 | 2F8QMM9QXPTGV | 2021-09-03 21:01:36.380 | 00F65F50-2356-4DE4-A748-CE4966BF1403 | 9F148331-C3FE-4724-83DA-00009862851E | 1 | 0 | 0 |
| 10 | 6DADEB92-9B36-489B-A232-0000CEC45BB6 | ASEIXIV9A95C7S2HXIY00 | LJ8GKKEPI5244H | 2021-09-03 21:10:53.003 | F7BE0050-4E2D-4079-9D68-CE34B7FE19DE | 6DADEB92-9B36-489B-A232-0000CEC45BB6 | 1 | 0 | 0 |
| 11 | 9AD69228-2533-4A95-88AA-0000EEC686C1 | 60V4WIKYENWHYQIT2BFN | 1B38RA7LCF85UL | 2021-09-03 21:09:04.683 | 878A1EA1-F88D-43AC-92E8-9693E998344 | 9AD69228-2533-4A95-88AA-0000EEC686C1 | 1 | 0 | 0 |
| 12 | D45221F7-45A3-43AE-8D83-0000EFA51FCB | PYH4K4OLEMWW070H4SNR | 7H7HZ8PY240FD9 | 2021-09-03 20:56:52.060 | ZD8F7259-4EE8-4D17-92B0-13BEE1D1918A | D45221F7-45A3-43AE-8D83-0000EFA51FCB | 1 | 0 | 0 |
| 13 | 10B55C3E-C9F2-4BB1-8E1A-0000EFA51FCB | 315COC1H2CM88Z3FM8K7 | WJJDESH78RNBAB | 2021-09-03 20:56:03.370 | E85ADA42-5B19-4D56-942F-09CA1A7D873C | 10B55C3E-C9F2-4BB1-8E1A-0000EFA51FCB | 0 | 0 | 1 |
| 14 | F91A2AB3-2F80-4C0E-B36B-0000F4F778E4 | BJZA19HCPBZMYHOGFRK7 | JKKCTZGPIKEIR3 | 2021-09-03 21:24:38.183 | 5697B0AD-873D-4FC2-A310-9FF468D58E09 | F91A2AB3-2F80-4C0E-B36B-0000F4F778E4 | 0 | 0 | 1 |
| 15 | 88C7FE53-4995-4048-8F91-000102C3634E | WUA23PQG1023KKJZVJMG | OS9M6J83GIA5P0 | 2021-09-03 21:19:25.240 | 1080670C-E659-4052-8E28-08EE163EB026 | 88C7FE53-4995-4048-8F91-000102C3634E | 0 | 0 | 1 |
| 16 | E1DC8E51-DA0E-44D9-9B1C-00010F5B11D8 | Z1H4M2N70GSABRH1WUX | HDRCAH2RR3NHNH | 2021-09-03 20:54:25.203 | EAS9C771-3781-4927-8AF6-C3A85C0D781F | E1DC8E51-DA0E-44D9-9B1C-00010F5B11D8 | 1 | 0 | 0 |
| 17 | 39F237CF-6951-4190-AC57-00011A901291 | JPKA4PFUK8LAGD1NVQI | A1CMTDRH73JKOI | 2021-09-03 21:25:07.563 | DAD06FCE-E601-4F87-9DA1-0A9C08B258C6 | 39F237CF-6951-4190-AC57-00011A901291 | 0 | 0 | 1 |
| 18 | 9B1C06AC-7431-4704-8579-00011C454AF8 | 93WKJMMVOSCV0WB7JV4P | ZA33AZNYHFKD | 2021-09-03 21:09:28.877 | 165C5438-8872-411F-8C9F-962188EB4C4A | 9B1C06AC-7431-4704-8579-00011C454AF8 | 0 | 0 | 1 |
| 19 | 0191F52F-733F-407D-980F-00014SD51AE8 | RLGPIS3Y1C1Y8YX4P8M | LBALWCSQFSWVVV | 2021-09-03 21:07:17.900 | 6688D156-5BDD-4FA2-9017-3843CAFE1784 | 0191F52F-733F-407D-980F-00014SD51AE8 | 1 | 0 | 0 |
| 20 | F6AD8E1B-939E-4FD8-9D5E-00015877D954 | 18UPZP3XKBCKFFX9003 | JOQYWDSS0VB7FV | 2021-09-03 21:01:23.530 | A2719D08-718E-43CD-BA0B-4B79E8A7A118 | F6AD8E1B-939E-4FD8-9D5E-00015877D954 | 0 | 0 | 1 |

Query executed successfully.

KH5452 (15.0 RTM) | SOFTSERVER\elist (78) | BigDataTestDB | 00:00:17 | 1,001,002 rows

Рис. Б.6. Результати виконання запиту CQ3 на локальному ресурсі