Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти перший (бакалаврський)

Комп'ютерна система для зберігання та аналізу даних

(тема)

Виконав:
здобувач 4 року навчання,
групи КІУКІ-21-1
Максим ШМАРІН
(власне ім’я, прізвище)

Спеціальність
123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

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

Освітня програма
Комп’ютерна інженерія
(повна назва освітньої програми)

Керівник: ас. Ірина КЛИМОВА
(посада, власне ім’я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (власне ім’я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

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

Освітня програма Комп’ютерна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Шмаріну Максиму Дмитровичу
(прізвище, ім’я, по батькові)

1. Тема роботи Комп'ютерна система для зберігання та аналізу даних

затверджена наказом по університету від “ 26 ” травня 2025 р. № 424 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 17 червня 2025 р.

3. Вхідні дані до роботи
система зберігання та обробки даних
хмарні сервіси
AWS

4. Перелік питань, що потрібно опрацювати у роботі
Огляд існуючих хмарних рішень
Огляд технологій розробки систем
Розробка компонентів комп’ютерної системи
Реалізація системи та проведення експериментів

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 11 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання та аналіз літератури	26.05.2025–30.05.2025	
2	Огляд існуючих рішень та хмар	31.05.2025–03.06.2025	
3	Вибір архітектури системи	04.06.2025–06.06.2025	
4	Вибір програмних засобів	07.06.2025–08.06.2025	
5	Програмна реалізація	09.06.2025–11.06.2025	
6	Аналіз отриманих результатів	12.06.2025–13.06.2025	
7	Оформлення записки	14.06.2025–16.06.2025	

Дата видачі завдання " 26 " травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ ас. Ірина КЛИМОВА
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 47 с., 11 рис., 1 табл., 1 дод., 6 джерел.

КОМП’ЮТЕРНА СИСТЕМА, КОРИСТУВАЧ, ОБРОБКА ДАНИХ, СХОВИЩЕ ДАНИХ, AWS, ЗБЕРІГАННЯ ДАНИХ.

Метою кваліфікаційної роботи є розробка комп'ютерної системи для зберігання та аналізу даних з використанням AWS.

У ході виконання кваліфікаційної роботи реалізовано комп’ютерну систему для зберігання та обробки інформації, яка ґрунтується на використанні хмарних сервісів Amazon та пов’язаних технологічних інструментів. Запропоноване рішення повністю відповідає сучасним технічним стандартам і має потенціал для інтеграції з наявними системами управління даними. Його впровадження сприяє оптимізації процесів керування інформаційними ресурсами, забезпечує надійний аналіз даних, підвищує загальну стійкість до збоїв і створює умови для гнучкого масштабування інфраструктури.

ABSTRACT

Bachelor’s thesis: 47 pages, 11 figures, 1 table, 1 appendices, 6 sources.

COMPUTER SYSTEM, USER, DATA PROCESSING, DATA REPOSITORY, AWS, DATA STORAGE.

The major goal of this thesis is the development of a computer system for data storage and analysis using AWS.

In order to a computer system was developed for storing and processing information, based on the use of Amazon cloud services and related technological tools. The proposed solution fully complies with modern technical standards and has the potential for integration with existing data management systems. Its deployment contributes to the optimization of information resource management processes, ensures reliable data analysis, enhances overall fault tolerance, and enables flexible scalability of the infrastructure.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних

СД – сховище даних

AWS – Amazon Web Services

SQL – structured query language

UML – Unified Modeling Language

ВСТУП

Сучасний етап інформатизації суспільства зумовлює зростання вимог до ефективного управління даними, що охоплює їх збір, завантаження, зберігання, передавання, захист і аналітичну обробку. Зі збільшенням обсягів інформації традиційні засоби опрацювання втрачають ефективність, що обумовлює необхідність у впровадженні високопродуктивних систем машинної обробки даних та додаткових обчислювальних потужностей. Комплексність таких систем, у свою чергу, спричиняє зростання вимог до кадрового забезпечення, енергоспоживання та витрат на обслуговування, водночас знижуючи гнучкість інформаційної інфраструктури та ускладнюючи масштабування бізнесу за умов локального зберігання даних.

У відповідь на ці виклики стрімко зростає популярність хмарних обчислювальних рішень, які забезпечують захищене та стабільне середовище для зберігання, доступу й обробки великих масивів даних. Зокрема, хмарна аналітика, яка у 2022 році зазнала особливо широкого поширення, реалізується переважно в межах моделі «програмне забезпечення як послуга» (SaaS). Вона охоплює інфраструктурні, аналітичні й технологічні компоненти, що забезпечують користувачам можливість оперативного отримання інсайтів із даних, часто – у режимі реального часу.

Аналіз даних у хмарному середовищі виконується поза межами локальних систем, що особливо важливо для організацій із великим обсягом інформації. На відміну від традиційних електронних таблиць, хмарні рішення використовують розподілену інфраструктуру, включаючи сервери, бази даних, мережі, програмне забезпечення, а також інструменти штучного інтелекту та машинного навчання. У деяких випадках такі сервіси надаються в межах платформи як послуги (PaaS), що дозволяє користувачам розробляти власні аналітичні інструменти відповідно до конкретних вимог бізнесу.

Постачальник послуг у межах таких моделей забезпечує підтримку апаратного забезпечення, віртуалізаційних платформ, операційних систем і мережевих компонентів, звільняючи користувачів від необхідності підтримки інфраструктури. Це дозволяє суттєво знизити експлуатаційні витрати, підвищити продуктивність систем і забезпечити доступ до актуальних даних для всіх співробітників, незалежно від їхнього географічного розташування, завдяки централізованому доступу через Інтернет.Метою кваліфікаційної роботи є розробка комп'ютерної системи збору та обробки даних з використанням AWS.

Завданнями:

- аналіз хмарних сховищ даних;
- аналіз хмари Amazon Web Services;
- розробка системи збору та обробки даних;
- розробка архітектури системи;
- вибір програмних засобів для реалізації.

1 ОГЛЯД ІСНУЮЧИХ ХМАРНИХ РІШЕНЬ

1.1 Аналіз хмарних схвоищ даних

Сучасні центри обробки даних (ЦОД) являють собою спеціалізовані серверні комплекси, призначені для створення високопродуктивного інформаційного середовища. Вони забезпечують можливість ефективного опрацювання великих масивів даних, що сприяє підвищенню загальної продуктивності бізнес-процесів і досягненню більш точних, інформативних результатів. Інтеграція ЦОД у корпоративну інфраструктуру є доцільною для більшості підприємств, оскільки забезпечує надійність, масштабованість і доступність обробки даних.

У рамках хмарної аналітики реалізується послідовність ключових етапів. Першим є етап приймання даних, що передбачає збір як структурованих, так і неструктурованих даних із різноманітних джерел – від текстових і табличних файлів до телеметричних потоків з пристроїв Інтернету речей (IoT), аудіо- та відеозаписів. Далі здійснюється попередня обробка: дані структуруються, очищуються та інтегруються для подальшого використання аналітичними модулями. Програмне забезпечення для аналітики дозволяє інтерпретувати ці дані з різною метою, зокрема для аналізу поведінки споживачів чи побудови бізнес-прогнозів.

Кінцевий етап передбачає візуалізацію аналітичних результатів у вигляді таблиць, графіків або інтерактивних панелей, що дозволяє користувачам оперативно відслідковувати ключові показники діяльності та приймати обґрунтовані управлінські рішення в режимі, наближеному до реального часу.

Сучасні ІТ-реалії засвідчують стрімке зростання кількості IoT-пристроїв, які вже перевищують кількість людей у глобальному масштабі. Інтернет речей набуває все ширшого застосування в різних галузях – від

медицини до енергетики та міської інфраструктури. Так, дані з носимих медичних пристроїв використовуються для моніторингу фізичного стану пацієнтів, а сенсори в «розумних» будинках дозволяють підвищити рівень безпеки та комфорту.

Дані, що генеруються в IoT-середовищі, мають гетерогенну природу – вони відрізняються за структурою, обсягом, швидкістю надходження. Ефективна обробка таких потоків потребує гнучких і масштабованих хмарних рішень, що дозволяють уникнути потреби у великих локальних обчислювальних потужностях. Для цього застосовуються методи фільтрації, нормалізації, агрегації, кореляційного аналізу та часової обробки.

Вибір відповідного програмного забезпечення залежить від характеру бізнес-завдань. Якщо базові інструменти забезпечують виконання елементарних функцій, то використання штучного інтелекту та машинного навчання дозволяє розв’язувати складні аналітичні задачі, відкриваючи нові можливості для оптимізації бізнес-процесів.

1.2 Використання хмар

Однією з ключових переваг хмарної аналітики є її здатність істотно пришвидшувати процеси обробки даних, забезпечуючи ефективність і результативність у стислий термін. Це дозволяє організаціям оперативно отримувати цінну інформацію для ухвалення стратегічно обґрунтованих рішень. В умовах, коли традиційний IT-відділ зосереджений на підтримці фізичної інфраструктури, прийняття рішень ускладнюється. Водночас хмарна аналітика звільняє IT-фахівців від задач інфраструктурного рівня, дозволяючи їм зосередитися на розробці логіки аналізу даних.

Використання хмарної інфраструктури сприяє досягненню бізнес-цілей завдяки забезпеченню своєчасного доступу до аналітичної інформації, що, своєю чергою, сприяє зростанню прибутків, розширенню клієнтської бази та прискоренню масштабування бізнесу. Незважаючи на те, що витрати на

впровадження хмарних технологій можуть варіюватися залежно від конкретного підприємства, у більшості випадків вони є економічно вигіднішими, ніж традиційні IT-рішення. Завдяки моделі "оплата за фактичне використання" (Pay-as-you-go) організації мають змогу оптимізувати витрати на інфраструктуру.

Економічна ефективність хмарних рішень досягається через універсальність доступу до IT-сервісів, відсутність потреби в капіталовкладеннях у серверне обладнання, можливість замовлення додаткових послуг за потреби та підвищення загальної продуктивності роботи. Крім того, гнучкість хмарної інфраструктури забезпечує легке масштабування ресурсів без необхідності фізичного розширення, що дає змогу оперативно реагувати на зміни у навантаженні.

Значну увагу хмарні сервіси приділяють безпеці. Постачальники впроваджують найсучасніші методи захисту, зокрема шифрування даних під час передавання та зберігання, а також багаторівневу автентифікацію та керування доступом. Водночас користувачі зобов’язані дотримуватись внутрішніх протоколів безпеки, зокрема щодо збереження ключів шифрування, для досягнення максимальної надійності системи.

Розвиток хмарних технологій створив нову норму для організації праці, зокрема у сфері віддаленого доступу. У період пандемії цей підхід став повсюдним, відкривши можливості для залучення висококваліфікованих фахівців незалежно від їхнього географічного розташування. Доступ до даних і ресурсів здійснюється через захищене підключення до Інтернету, що дозволяє виконувати виробничі завдання незалежно від місцезнаходження працівника. Хмарні сервіси також спростили процес спільної роботи з документами, автоматизуючи оновлення файлів у режимі реального часу.

Завдяки об'єднанню всіх джерел даних в єдине інформаційне середовище, хмарна аналітика забезпечує цілісне бачення бізнес-процесів, підвищуючи ефективність комунікацій між працівниками на глобальному рівні. Окрім цього, хмарне зберігання даних відіграє критично важливу роль

у забезпеченні резервного копіювання й аварійного відновлення. Постачальники забезпечують дублювання даних на декількох серверах, що мінімізує ризики втрати інформації у разі збоїв або катастроф.

Ще однією важливою перевагою є автоматичне оновлення програмного забезпечення, яке не вимагає втручання IT-фахівців, що дозволяє заощаджувати ресурси. Більшість сучасних хмарних аналітичних платформ інтуїтивно зрозумілі, що дозволяє використовувати їх без попередньої підготовки.

Таким чином, хмарна інфраструктура є основою цифрової трансформації. З огляду на зростаючі вимоги до ефективності обробки даних, організації повинні обирати модель хмарної аналітики відповідно до власних потреб, враховуючи специфіку своєї діяльності.

1.3 Хмарна інфраструктура

У сучасній цифровій екосистемі виокремлюють три базові типи хмарних інфраструктур – публічну, приватну та гібридну, кожна з яких має свої функціональні переваги та найбільш ефективні сфери застосування.

Публічна хмара репрезентує модель загального доступу до віртуалізованих ресурсів, таких як обчислювальні потужності, сховища даних і прикладне програмне забезпечення, що надаються користувачам за моделлю IaaS (інфраструктура як послуга). Ці ресурси інтегруються в єдину інфраструктуру, розміщену на стороні постачальника, та надаються віддалено. Така система подібна до концепції коворкінгу, де простір спільно використовується різними користувачами без можливості контролю над сусідами. Попри це, публічні хмари характеризуються високою гнучкістю, швидким масштабуванням та невисокими витратами, що робить їх привабливими для багатьох організацій.

Приватна хмара, натомість, створюється виключно для однієї організації і функціонує в межах її захищеної мережевої інфраструктури або

на базі спеціалізованого хостинг-провайдера. Вона забезпечує максимальний рівень ізоляції, конфіденційності та безпеки, зберігаючи при цьому властиві хмарним рішенням переваги у вигляді масштабованості та автоматизованого адміністрування. Водночас, реалізація приватної хмари вимагає значних фінансових і технічних ресурсів для побудови та підтримки інфраструктури.

Гібридна модель поєднує елементи публічної та приватної хмари, формуючи єдине операційне середовище з можливістю безперешкодного обміну даними між ними. Така архітектура дозволяє розміщувати загальнодоступні або менш чутливі дані в публічному сегменті, а критичну інформацію – у приватному. У такий спосіб досягається оптимальний баланс між безпекою, продуктивністю й економічністю, що робить гібридні хмари ефективним інструментом для реалізації комплексних аналітичних сценаріїв.

1.4 Висновки до розділу 1

У першому розділі досліджено ключові аспекти побудови аналітичних рішень із використанням хмарної інфраструктури, зокрема розкрито структурні елементи таких систем та їх функціональну значущість. Хмарні технології демонструють високий потенціал у контексті спрощення управління доступом до інформаційних ресурсів та забезпечення їхньої актуальності для всіх користувачів. Водночас, ці рішення відкривають широкі можливості щодо створення відмовостійких систем завдяки реалізації механізмів реплікації даних у географічно розподілених регіонах.

Зазначені особливості підкреслюють доцільність вибору саме хмарної інфраструктури для побудови системи зберігання та аналітичної обробки даних. У цьому контексті також було здійснено класифікацію хмарних моделей за рівнем доступності: публічні, приватні та гібридні хмари.

Залежно від характеру оброблюваних даних обґрунтовується вибір відповідної моделі. Зокрема, у випадках, коли система не працює з чутливою

або персоналізованою інформацією (наприклад, персональні дані чи медична інформація), найбільш раціональним виглядає використання публічної хмари. Якщо ж обробка стосується конфіденційних даних, доцільно застосовувати приватну або гібридну хмарну модель. У такій конфігурації зберігання може здійснюватися локально на обладнанні організації, а обчислювальні процеси – відбуватися на захищених віддалених серверах із використанням шифрування.

Оскільки набір даних, який передбачається залучити до реалізації аналітичної системи, не включає конфіденційної інформації, оптимальним варіантом виступає публічна хмарна інфраструктура.

2 ОГЛЯД ТЕХНОЛОГІЙ РОЗРОБКИ СИСТЕМ

2.1 Зберігання даних у хмарі

Аналітична система охоплює низку взаємопов’язаних складових, серед яких визначальну роль відіграють джерела надходження інформації, моделі її структурного представлення, логіка обробки, обчислювальні ресурси, аналітичні алгоритми, а також засоби збереження та візуалізації отриманих результатів. Усі ці компоненти забезпечують цілісне функціонування процесу аналізу даних – від моменту збору до прийняття обґрунтованих рішень на основі отриманої інформації.

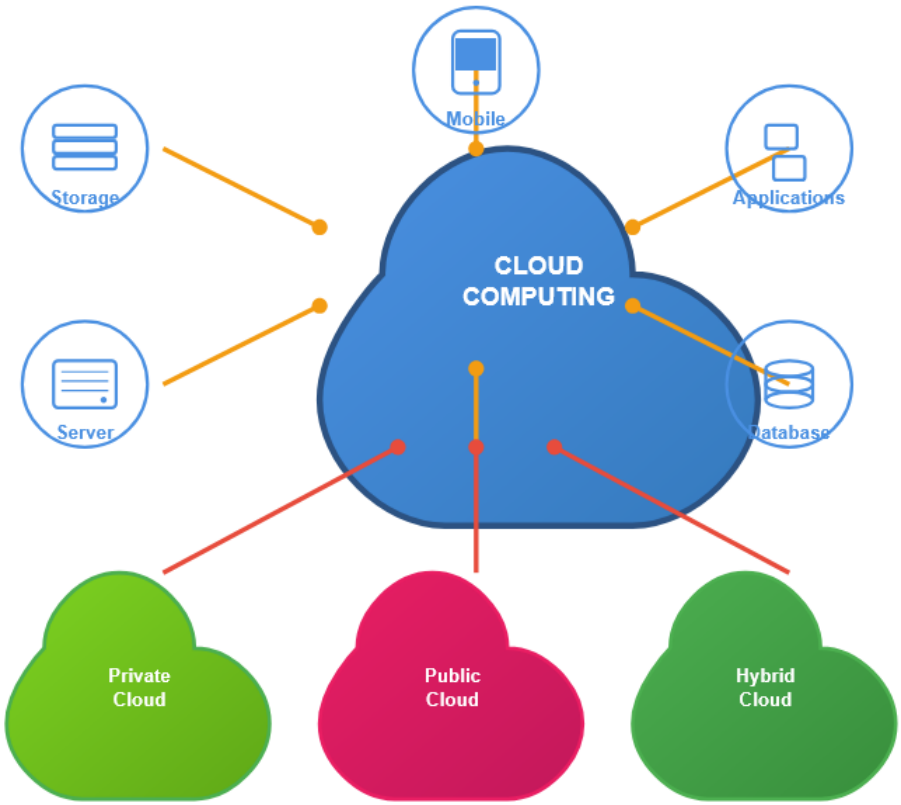


Рисунок 2.1 – Хмарні рішення

Архітектура хмарної аналітики передбачає функціонування системи на основі чотирирівневої моделі, що охоплює інфраструктурний, керуючий,

аналітичний та візуалізаційний рівні. Крім того, на всіх етапах функціонування аналітичної платформи діє наскрізний рівень, який забезпечує відповідність вимогам безпеки, інформаційного управління та нормативного регулювання. Йдеться, зокрема, про механізми ідентифікації, контролю доступу, шифрування даних, ведення політик відповідності національним та міжнародним стандартам (зокрема HIPAA, PCI-DSS тощо).

На рівні інфраструктури реалізуються базові функції приймання, зберігання та масштабування даних. Тут відбувається обробка інформації з гетерогенних джерел: від IoT-пристроїв до соціальних медіа та мультимедійних потоків. Гнучке реагування на пікові навантаження реалізується через горизонтальну та вертикальну масштабованість.

Управління даними здійснюється на рівні сховищ, що оперують концепцією "озер даних", які приймають інформацію у сирому вигляді, без попередньо визначеної схеми. Таке зберігання підтримується механізмами Hadoop, зокрема об’єктно-орієнтованими структурами, що дозволяють забезпечити реплікацію для відмовостійкості та ефективне зчитування завдяки підходу schema-on-read.

На аналітичному рівні відбувається безпосередня обробка та інтерпретація даних. Тут реалізуються алгоритми машинного навчання, методи кластеризації, виявлення закономірностей, прогнозного аналізу. Для цього використовуються як спеціалізовані мови програмування (Python, R, Matlab), так і технології обробки великих даних (Spark, Hive, MapReduce тощо).

Фінальний рівень архітектури – візуалізаційний – забезпечує інтерактивний доступ до результатів аналізу. Інтерфейс користувача дозволяє ефективно працювати зі складною аналітичною інформацією навіть без залучення фахівців з ІТ. Гнучкість цього рівня сприяє ефективному прийняттю управлінських рішень на основі представлених у реальному часі результатів.

2.2 Сервіси у хмарі AWS

AWS надає найширший вибір аналітичних послуг, які дають Сучасні інструменти роботи з даними відкривають перед організаціями широкі можливості трансформації бізнес-процесів незалежно від розміру або галузі. Завдяки хмарним рішенням, зокрема екосистемі Amazon Web Services (AWS), забезпечується ефективна підтримка повного циклу роботи з даними – від їх переміщення, зберігання та створення озер даних до реалізації аналітики великих обсягів інформації, потокового аналізу та застосування алгоритмів машинного навчання.

Amazon S3 є базовою складовою цієї екосистеми. Ця служба об’єктного зберігання відзначається масштабованістю, високим рівнем доступності, безпеки та продуктивності. Вона забезпечує зручне та надійне зберігання даних для широкого спектра задач – від створення озер даних до архівування й резервного копіювання. Гнучка система керування правами доступу, розширена класифікація за допомогою тегів, префіксів та каталогів, а також автоматизовані пакетні операції сприяють зручному керуванню великими обсягами даних.

Крім того, Amazon RDS надає розгорнуту підтримку реляційних баз даних із можливістю використання популярних СКБД, таких як MySQL, PostgreSQL, Oracle та інші. Цей сервіс бере на себе адміністративні задачі – моніторинг, резервне копіювання, автоматичне оновлення, відновлення та усунення збоїв, що істотно знижує навантаження на IT-персонал і сприяє підвищенню загальної надійності IT-інфраструктури.

Служба Amazon RDS забезпечує інструменти для організації високодоступної інфраструктури баз даних шляхом реалізації реплікацій, які суттєво підвищують надійність функціонування інформаційних систем в умовах зростання навантаження або аварійних ситуацій. Використання функціоналу багатозонального розгортання (Multi-AZ) дозволяє підтримувати безперервність критично важливих процесів завдяки

автоматичному перемиканню з основної бази даних на вторинну, що синхронно реплікується.

Таблиця 2.1 – Порівняння локальної бази даних та сервісу від Amazon

	База даних	Amazon RDS
Наявність легкого масштабування згідно навантаження	-	+
Автоматичне встановлення оновлень	-	+
Необхідність проводити обслуговування обладнання	-	-
Штат працівників для розгортання нової інфраструктури	+	-
Форс-мажорні обставини (вразливості катастрофи та ін.)	+	-
Ефективність витрати коштів	-	+

Крім того, можливість створення реплік лише для читання розширює межі обробки даних, зменшуючи навантаження на основний вузол і підвищуючи продуктивність під час виконання складних аналітичних запитів.

Amazon RDS поєднує гнучкість і масштабованість хмарних обчислень із високим рівнем автоматизації, значно перевершуючи традиційні локальні рішення. Ці переваги детально порівнюються в таблиці 2.1.

Доповнюючи інструментарій аналітичної інфраструктури, служба Amazon Athena надає змогу виконувати інтерактивні SQL-запити безпосередньо до даних, збережених в Amazon S3. Цей сервіс усуває потребу у створенні окремих баз даних, дозволяючи аналітику працювати безпосередньо з озером даних, що сприяє оперативному отриманню інсайтів із мінімальними витратами ресурсів.

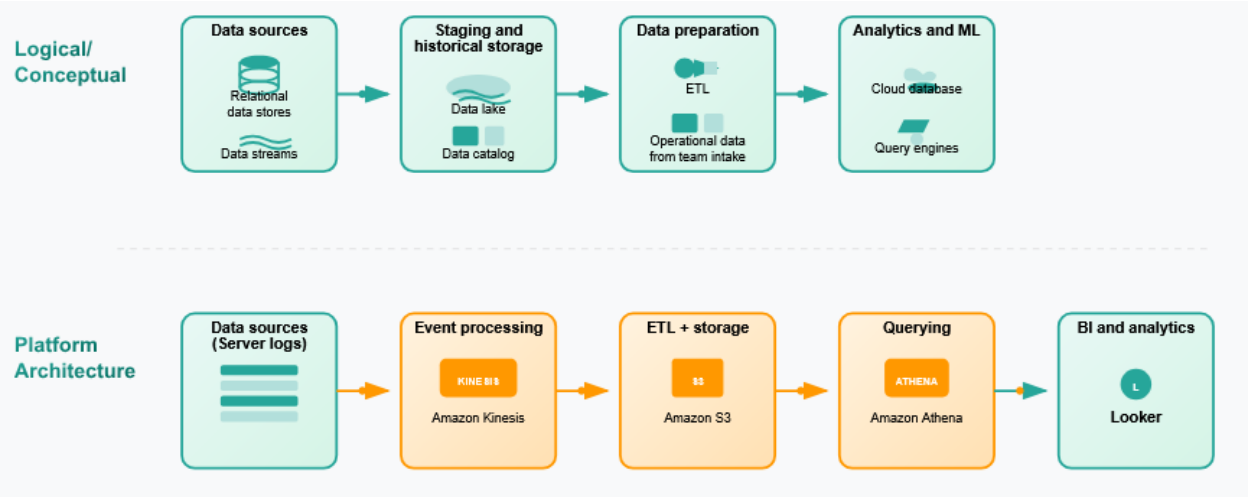


Рисунок 2.2 – Архітектура Pipeline

За допомогою декількох кліків у консолі AWS можна налаштувати службу Athena, вказавши каталог даних на Amazon S3, після чого завдяки SQL-запитам отримувати аналітичні результати за лічені секунди. Athena реалізована за принципом serverless – користувач не опікується інфраструктурою, а оплата нараховується тільки за виконані запити. Сервіс ідеально підходить для роботи з журналами, аналізу необроблених даних та інтерактивних аналітичних сценаріїв. Завдяки автоматичному масштабуванню Athena здатна ефективно обробляти великі обсяги інформації й паралельно виконувати складні запити [5].

На рисунку 2.2 зображено типовий data pipeline: необроблені дані отримуються з різних джерел і зберігаються в S3. Вони можуть бути одразу запитані через Athena без необхідності попереднього налаштування бази даних чи інструментів для обробки. Після аналізу можуть бути застосовані

складніші ETL-трансформації і результати – збережені назад в S3 [6]. При цьому Athena дозволяє надалі запитувати вже оброблені дані, а інструмент бізнес-аналітики Amazon QuickSight підключається до Athena і створює візуалізації на їхній основі.

AWS Glue – сервіс для інтеграції та підготовки даних, який включає інструменти для побудови ETL-процесів та підтримки метаданих [7]. Його каталог даних надає централізоване зберігання схем, статистики та версій метаданих незалежно від локації даних. Glue автоматизує виявлення форматів даних за допомогою сканерів і класифікаторів, запускає ETL-завдання за розкладом чи за подіями, а також масштабується динамічно – ресурси додаються або знижуються відповідно до обсягу роботи [8].

AWS Lake Formation спрощує створення захищених озер даних, центральних сховищ структурованих і необроблених даних, готових для роботи з аналітикою й ML [9]. Воно автоматизує процес імпорту даних, очистки, класифікації за допомогою ML-алгоритмів, шифрування та визначення політик доступу на рівнях рядків, стовпців і клітинок. Lake Formation формує уніфікований каталог даних, інтегрований із службами аналітики та ML: Redshift, Athena, EMR, QuickSight.

Amazon QuickSight – повністю керована BI-платформа, заснована на машинному навчанні, що дозволяє створювати інтуїтивні dashboards та візуалізації [10]. Вона підтримує підключення до численних джерел даних, включає централізовані інструменти автентифікації та управління доступом, працює з пам’яттю через SPICE для пришвидшених запитів, є економічно ефективною при збільшенні кількості користувачів та не вимагає складної підготовки або розгортання інфраструктури.

2.3 Керування обчислювальними та мережевими ресурсами

Інфраструктура як код (Infrastructure as Code, IaC) є сучасною парадигмою управління обчислювальними та мережевими ресурсами, яка

передбачає їх опис за допомогою програмного коду, що замінює традиційні методи налаштування вручну або через інтерактивні засоби. Такий підхід дає змогу формалізувати конфігурацію інфраструктури, забезпечуючи її відтворюваність і спрощуючи супровід. Використання конфігураційних файлів, які містять усі необхідні параметри інфраструктури, дозволяє централізовано контролювати зміни та мінімізувати ризики, пов’язані з людським фактором.

Завдяки інтеграції систем контролю версій конфігураційні файли стають невід’ємною частиною процесу розробки, що уможливлює гнучке масштабування та повторне використання модульних компонентів інфраструктури. Автоматизоване надання ресурсів сприяє пришвидшенню циклів розгортання та підвищує продуктивність команд розробки, дозволяючи зосередитися на логіці додатків, а не на ручному керуванні серверною частиною.

IaC реалізується за допомогою двох основних підходів: декларативного та імперативного. Перший дозволяє описати цільовий стан системи, який інструменти автоматично реалізують, зберігаючи інформацію про поточну конфігурацію. Імперативна модель передбачає надання чіткого набору команд, які мають бути виконані у певній послідовності. Хоча багато інструментів підтримують обидва підходи, більшість із них орієнтовані саме на декларативну модель через її гнучкість і простоту супроводу.

Історично забезпечення інфраструктури вимагало значних ручних зусиль, але з поширенням хмарних обчислень та контейнеризації виникла потреба в автоматизації масштабування, оновлення та видалення ресурсів. У таких умовах IaC стає необхідним для ефективного керування складною інфраструктурою, зменшуючи кількість помилок та покращуючи узгодженість конфігурацій.

Прикладом реалізації IaC є сервіс AWS CloudFormation, який дозволяє створювати, управляти та підтримувати інфраструктуру за допомогою шаблонів у форматі JSON або YAML. Ці шаблони детально описують

необхідні ресурси та їх взаємозв’язки, що забезпечує автоматизоване розгортання, керування залежностями, а також отримання зворотного зв’язку після завершення розгортання. CloudFormation також пропонує інструменти для візуалізації структури інфраструктури, підтримку повторного використання шаблонів, а також API для автоматизації процесів.

Таким чином, використання IaC, зокрема на прикладі AWS CloudFormation, дозволяє компаніям значно оптимізувати процеси управління інфраструктурою, досягаючи високого рівня гнучкості, масштабованості та надійності, що є критично важливим у контексті сучасної цифрової трансформації.

2.3 Висновки до розділу 2

Основним призначенням відповідних хмарних сервісів є зберігання даних, однак характер і структура цих даних суттєво різняться. Зокрема, Amazon S3 виконує функцію озера даних, що дає змогу зберігати великі обсяги інформації у структурованому, напівструктурованому та неструктурованому вигляді, без потреби попередньої фіксації схеми. Натомість Amazon RDS є керованою службою реляційних баз даних, яка розгортається в інфраструктурі хмарного провайдера й орієнтована на обробку строго структурованих даних відповідно до фіксованих схем.

Узгоджене використання обох сервісів дозволяє формувати комплексне рішення для управління сховищами даних, організації повноцінних автономних процесів їх обробки, а також забезпечує сумісність із сучасними інструментами аналітики. Відповідно, інтеграція S3 і RDS буде важливою складовою побудови системи зберігання й аналізу даних.

Крім того, у межах дослідження було розглянуто концепцію розгортання інфраструктури як коду (Infrastructure as Code, IaC), що забезпечує відтворюваність і консистентність конфігурацій незалежно від фізичного середовища розгортання. У середовищі Amazon Web Services

реалізація цієї концепції здійснюється за допомогою сервісу AWS CloudFormation, який надає змогу описувати інфраструктурні компоненти у форматах JSON або YAML, що суттєво спрощує керування конфігураціями, їх автоматизацію та контроль версій.

Функціональні можливості CloudFormation дозволяють оптимізувати процес розробки аналітичної системи, забезпечити централізований моніторинг, контроль за використанням ресурсів, а також реалізувати швидке розгортання й згортання компонентів. У зв’язку з цим застосування підходу IaC із використанням CloudFormation є доцільним під час побудови хмарної системи зберігання та аналітики даних.

3 РОЗРОБКА КОМПОНЕНТІВ КОМП’ЮТЕРНОЇ СИСТЕМИ

3.1 Аналітична система

Розроблена аналітична система має багаторівневу архітектуру, яка включає локальну базу даних, віддалену реляційну базу даних, об’єктне сховище даних у хмарному середовищі, озеро даних, а також проміжний сервер, який виконує функцію міграції даних між локальним та віддаленим середовищами. Після завершення початкової міграції даних з локальної БД на віддалений сервер здійснюється подальша реплікація до озера даних, що забезпечує узгодженість та збереження повноти інформації.

Усі згадані компоненти були описані за допомогою конфігураційного файлу у форматі YAML, що реалізує підхід «інфраструктура як код». Такий формат забезпечує декларативне описання необхідних ресурсів, їх параметрів, умов створення та вихідних значень. Застосування цього методу дозволяє здійснювати автоматизоване розгортання й згортання інфраструктури в середовищі хмарного провайдера, що суттєво оптимізує витрати, спрощує керування ресурсами та гарантує можливість швидкого відновлення або масштабування системи.

Структура конфігураційного файлу включає кілька ключових розділів: опис загального призначення ресурсу (Description), метадані для динамічного налаштування (Metadata), таблиці відповідностей параметрів і ресурсів (Mappings), параметри з допустимими значеннями (Parameters), умовні конструкції, що регулюють логіку створення ресурсів (Conditions), перелік ресурсів (Resources), а також вихідні значення, які формуються після розгортання (Outputs). Така модульність та формалізований підхід сприяють ефективному управлінню життєвим циклом інфраструктури.

3.2 Розробка сховища для зберігання даних

Процес формування аналітичної системи традиційно розпочинається з проєктування структури зберігання даних і вибору відповідного типу сховища. У межах цього дослідження як первинне сховище було обрано локально розгорнуту реляційну базу даних Microsoft SQL Server, яка використовується для збереження структурованої інформації. Архітектура цієї БД передбачає логічний поділ на чотири схеми: Application, Purchasing, Sales і Warehouse. Перша з них містить загальні конфігураційні параметри, тоді як інші акумулюють інформацію про закупівлі, реалізацію продукції та складські операції відповідно.

Беручи до уваги переваги, що надаються хмарними технологіями, подальше зберігання та обробка даних реалізовано з використанням хмарної інфраструктури Amazon Web Services. Зокрема, дані були перенесені до Amazon RDS – керованого середовища для реляційних баз даних, яке є функціональним аналогом локального рішення. Додатково було створено об’єктне сховище Amazon S3 із двома окремими логічними розділами: landing-bucket для збереження необроблених даних і structured-bucket для очищеної та структурованої інформації, готової до аналітичної обробки.

Процес міграції даних з локальної інфраструктури до хмари здійснювався за допомогою сервісу Amazon Database Migration Service (DMS). Було здійснено попереднє налаштування джерела й приймача даних, після чого визначено необхідні сутності, зокрема: джерело (Source Endpoint), приймач (Target Endpoint), проміжний сервер (Replication Instance) та завдання міграції (Database Migration Task). Після завершення первинного перенесення, дані з Amazon RDS були репліковані до озера даних у S3 за допомогою аналогічного процесу з оновленими параметрами.

У результаті реплікації в каталозі landing-bucket було створено ієрархічну структуру, що відповідає логіці організації таблиць у початковій БД: підкаталоги за схемами, які містять директорії, що відображають назви

відповідних таблиць. На цьому етапі дані зберігаються у форматі CSV, що зумовлює необхідність подальшої текстової обробки під час реалізації аналітичних сценаріїв.

3.3 Обробка даних в системі

Подальшим етапом розробки аналітичної системи є ідентифікація типів даних відповідно до форматів, які підтримуються службою AWS Glue. За допомогою цього сервісу, що інтегрується з мовою програмування Python та бібліотекою PySpark, здійснюється поетапне зчитування даних із таблиць, розміщених у каталозі landing-bucket. При цьому кожна таблиця обробляється окремо, а визначення схеми здійснюється динамічно під час завантаження даних. Наприклад, вигляд схеми, сформованої безпосередньо після зчитування таблиці Application.SystemParameters, подано на рисунку 3.1.

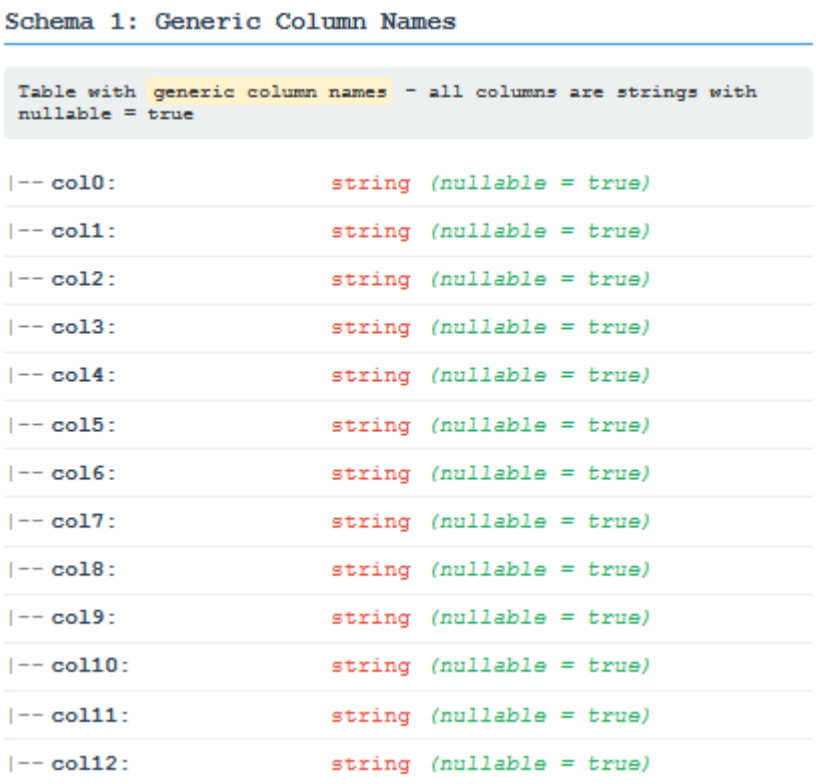


Рисунок 3.1 – Схема таблиці

На наступному етапі обробки здійснюється уточнення структури даних: проводиться перейменування колонок відповідно до прийнятих стандартів і логіки іменування, а також встановлення коректних типів даних для кожного стовпця. Це дозволяє перетворити дані з текстового формату на відповідні числові, логічні або часові типи згідно з початковою схемою бази даних. Кінцевий вигляд структурованої схеми таблиці, сформованої після цього перетворення, представлено на рисунку 3.2.

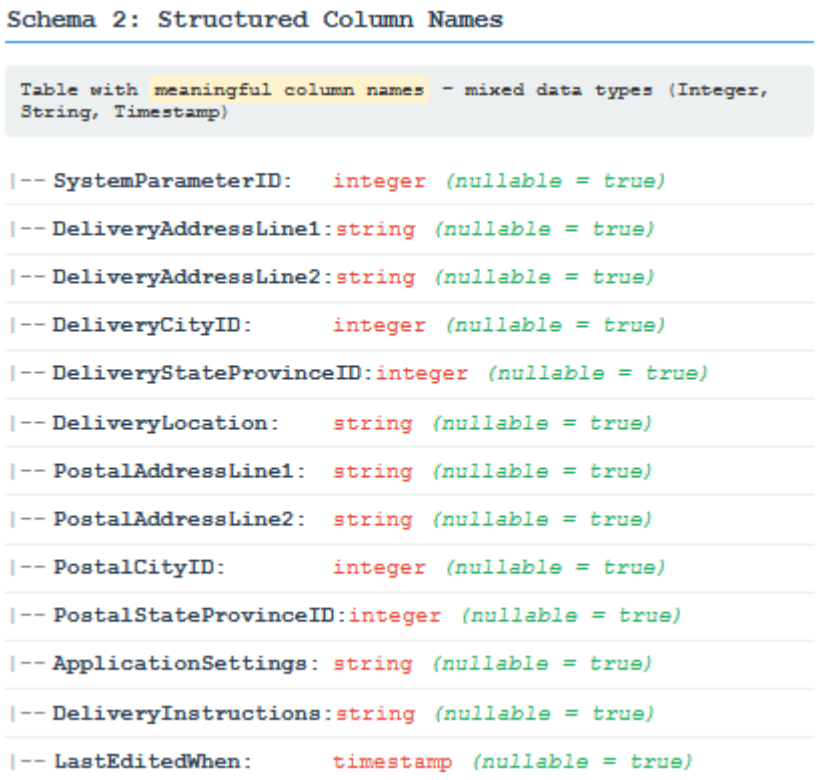
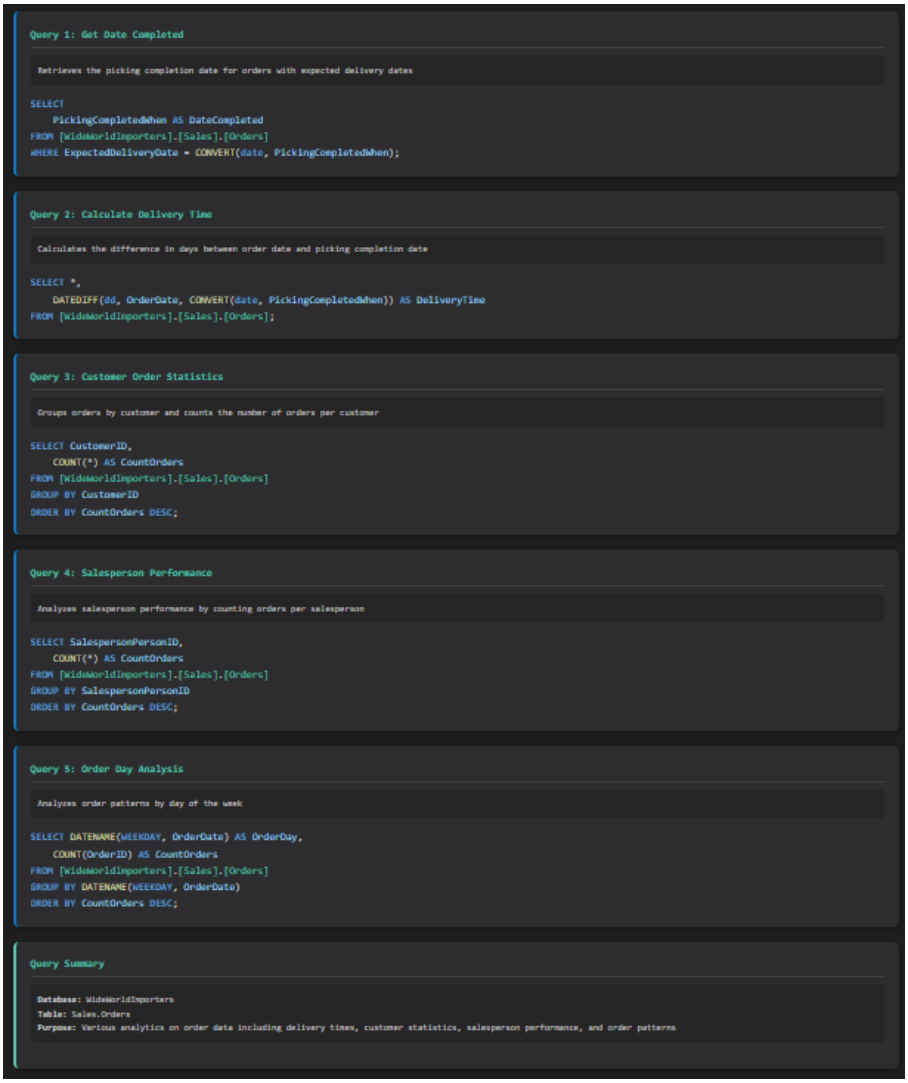


Рисунок 3.2 – Кінцева таблиця

Для створення аналітичних наборів даних на основі таблиці Sales.Orders доцільніше обрати підхід, який передбачає обробку вже реплікованих до озера даних файлів, а не виконання запитів безпосередньо до бази даних. Такий підхід дозволяє зменшити навантаження на реляційну базу та прискорити процес обробки, оскільки дані вже зберігаються у сховищі S3 у вигляді текстових файлів.

На основі цих файлів можна виконати відповідні трансформації в середовищі AWS Glue або іншим інструментом, що підтримує обробку даних у хмарі. Це дозволяє зберегти ефективність процесів реплікації, забезпечити масштабованість і зменшити витрати на ресурси бази даних.

Для аналітичної обробки використовуватимуться створені на основі Sales.Orders набори: OrdersTopSeller, OrdersTopCustomer, OrdersAddCols і OrdersWeekdays, які міститимуть відповідні агреговані або трансформовані дані, необхідні для подальшого аналізу.



```
Query 1: Get Date Completed

Retrieves the picking completion date for orders with expected delivery dates

SELECT
    PickingCompletedWhen AS DateCompleted
FROM [WideWorldImporters].[Sales].[Orders]
WHERE ExpectedDeliveryDate = CONVERT(date, PickingCompletedWhen);

Query 2: Calculate Delivery Time

Calculates the difference in days between order date and picking completion date

SELECT *,
    DATEDIFF(dd, OrderDate, CONVERT(date, PickingCompletedWhen)) AS DeliveryTime
FROM [WideWorldImporters].[Sales].[Orders];

Query 3: Customer Order Statistics

Groups orders by customer and counts the number of orders per customer

SELECT CustomerID,
    COUNT(*) AS CountOrders
FROM [WideWorldImporters].[Sales].[Orders]
GROUP BY CustomerID
ORDER BY CountOrders DESC;

Query 4: Salesperson Performance

Analyzes salesperson performance by counting orders per salesperson

SELECT SalespersonPersonID,
    COUNT(*) AS CountOrders
FROM [WideWorldImporters].[Sales].[Orders]
GROUP BY SalespersonPersonID
ORDER BY CountOrders DESC;

Query 5: Order Day Analysis

Analyzes order patterns by day of the week

SELECT DATENAME(WEEKDAY, OrderDate) AS OrderDay,
    COUNT(OrderID) AS CountOrders
FROM [WideWorldImporters].[Sales].[Orders]
GROUP BY DATENAME(WEEKDAY, OrderDate)
ORDER BY CountOrders DESC;

Query Summary

Database: WideWorldImporters
Table: Sales.Orders
Purpose: Various analytics on order data including delivery times, customer statistics, salesperson performance, and order patterns
```

Рисунок 3.3 – Скрипт запиту до бази даних

Після виконання всіх етапів попередньої обробки та трансформації даних у середовищі AWS Glue, результати були збережені у новому

підкаталозі AnalyticsOrders, що розміщується в каталозі structured-bucket. У цьому підкаталозі зберігаються вже підготовлені набори даних, сформовані на основі таблиці Sales.Orders, які були використані для побудови аналітики. Такий підхід дозволяє зберігати дані в упорядкованому вигляді та забезпечує їхнє ефективне використання у подальших аналітичних процесах.

Перевагою такого рішення є не лише логічна сегментація даних, а й забезпечення гнучкості та масштабованості при обробці й візуалізації інформації.

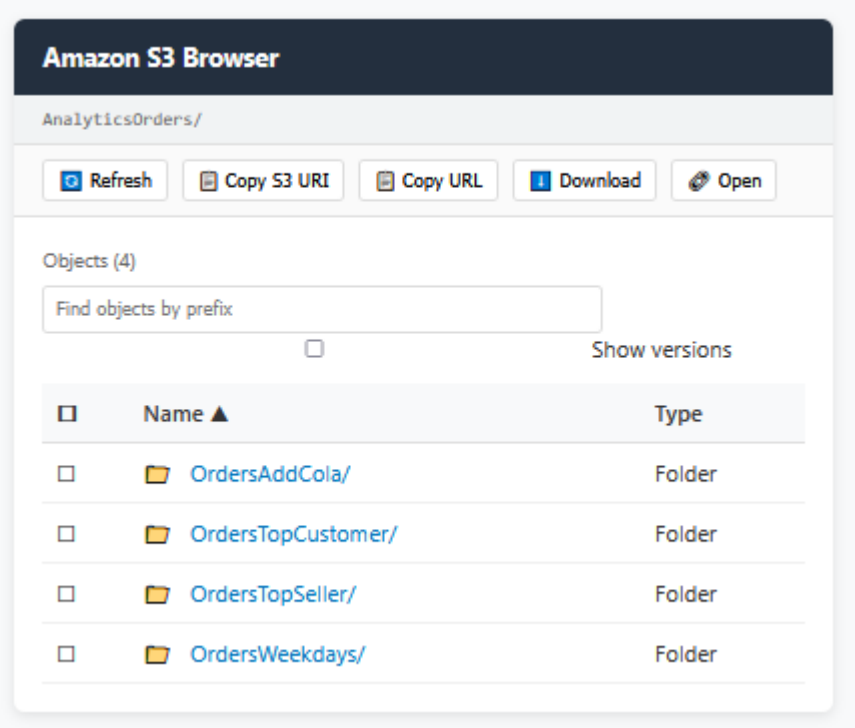


Рисунок 3.4 – AnalyticsOrders

Для забезпечення подальшої роботи з новоствореними наборами даних як із реляційними таблицями необхідно створити відповідні мета-сховища, які забезпечать опис структури даних і дозволять засобам аналітики, зокрема Amazon Athena, ефективно взаємодіяти з ними.

Це завдання реалізується за допомогою служби AWS Glue Crawlers, яка дозволяє автоматично сканувати дані, виявляти їхню структуру та створювати або оновлювати таблиці в каталозі даних AWS Glue. Glue Crawler

аналізує файли, що зберігаються у визначеному каталозі (наприклад, structured-bucket/AnalyticsOrders), розпізнає їхні поля, типи даних та формує відповідні схеми у мета-каталозі. Це значно спрощує подальший доступ до наборів даних для SQL-запитів і візуалізацій.

Використання Glue Crawlers забезпечує масштабованість, автоматизацію та узгодженість метаданих, що критично важливо для побудови надійної аналітичної інфраструктури.

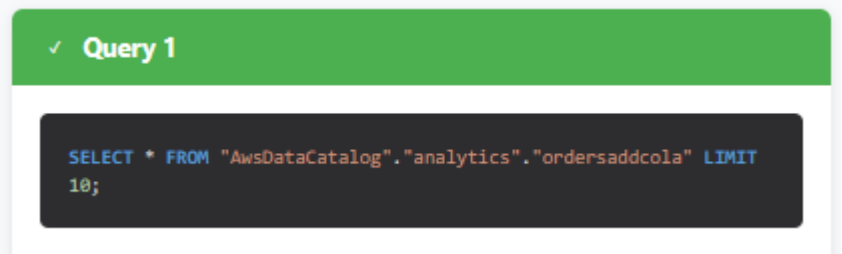


Рисунок 3.5 – Запит службою Athena до таблиці OrderAddColds

Після успішного завершення роботи Glue Crawlers у менеджері озера даних AWS Lake Formation автоматично з’являються нові таблиці, що були розпізнані у структурованому сховищі structured-bucket. Ці таблиці містять усю необхідну метаінформацію, яка дозволяє аналітичним службам швидко орієнтуватися в структурі даних.

На цьому етапі стає можливим виконання SQL-запитів безпосередньо до цих таблиць за допомогою Amazon Athena. Оскільки Athena працює на основі сервісу serverless, це дозволяє запускати запити без необхідності налаштовувати додаткову інфраструктуру. Крім того, виявлені таблиці можуть бути використані як джерело даних для сторонніх аналітичних інструментів, таких як Amazon QuickSight або інші BI-платформи, що підтримують підключення до AWS.

Таким чином, завершення процесу створення мета-сховищ відкриває шлях до повноцінної аналітичної обробки та візуалізації даних у хмарному середовищі.

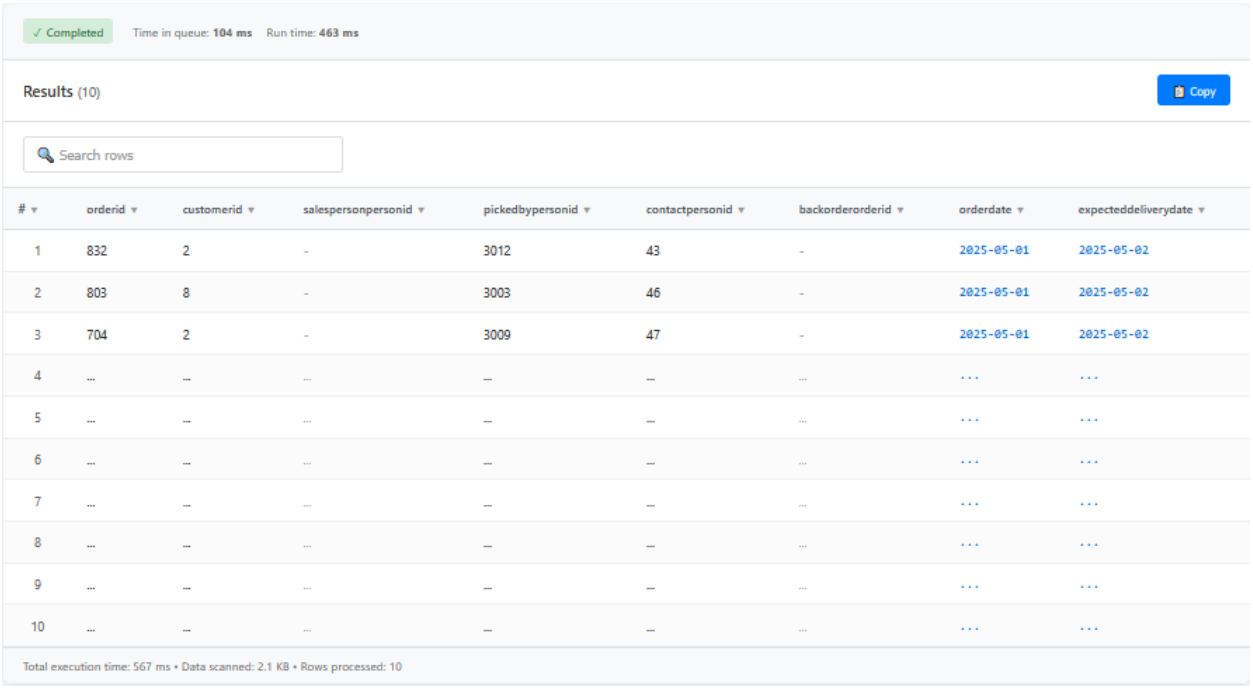


Рисунок 3.6 – Відповідь служби Athena на запит

Як демонструє рисунок 3.5, приклад виконання SQL-запиту в середовищі Amazon Athena дозволяє швидко отримати результати обробки даних, збережених у structured-bucket. На рисунку 3.6 видно, що результат запиту представлено у табличному вигляді з можливістю подальшого аналізу.

Для подальшої обробки результатів запитів, візуалізації та моніторингу ключових бізнес-показників використовується Amazon QuickSight – гнучкий аналітичний інструмент із підтримкою хмарного масштабування. Платформа дозволяє створювати інформаційні панелі, графіки, діаграми та інші інтерактивні елементи, що спрощують аналіз та прийняття рішень.

Завдяки інтеграції з Athena, QuickSight може підключатись безпосередньо до таблиць, створених у Glue/Lake Formation, забезпечуючи актуальну візуалізацію без проміжного експорту даних.

На початковому етапі використання Amazon QuickSight важливо здійснити підключення відповідних джерел даних. Для цього обираються таблиці, згенеровані службами Glue Crawlers та збережені у Lake Formation, які потім інтегруються у середовище QuickSight як джерела даних для

візуалізації.

Рисунок 3.7 – Відображення показників

Після підключення аналітик формує аналітичні вікна – інтерактивні панелі з ключовими показниками. QuickSight забезпечує інтуїтивно зрозумілий інтерфейс, що дозволяє змінювати спосіб відображення даних, групувати їх за категоріями, застосовувати фільтри, а також налаштовувати розташування та формат показників. Аналітичне вікно зберігається автоматично, що дає змогу повторно використовувати його в майбутньому.

Рисунок 3.7 демонструє приклад такого візуального представлення ключових показників, що значно полегшує прийняття рішень на основі даних.

3.4 Архітектура системи

Розгортання аналітичної системи здійснювалося комбінованим підходом, що передбачає поєднання інфраструктури як коду (IaC) та інтерактивного керування через хмарну платформу. Зокрема, фізичні ресурси, необхідні для функціонування системи, були детально описані у конфігураційному файлі у форматі YAML, що забезпечило автоматизоване, відтворюване та контрольоване розгортання базових елементів інфраструктури.

Наступні етапи, зокрема налаштування процесів міграції та реплікації

даних, а також їх подальша обробка та інтеграція з аналітичними сервісами, реалізовувалися за допомогою інструментів, доступних через Amazon Console. Такий підхід дав змогу зберегти гнучкість у керуванні процесами, забезпечити оперативне внесення змін, а також підтримати високий рівень прозорості на кожному етапі розгортання аналітичної системи.

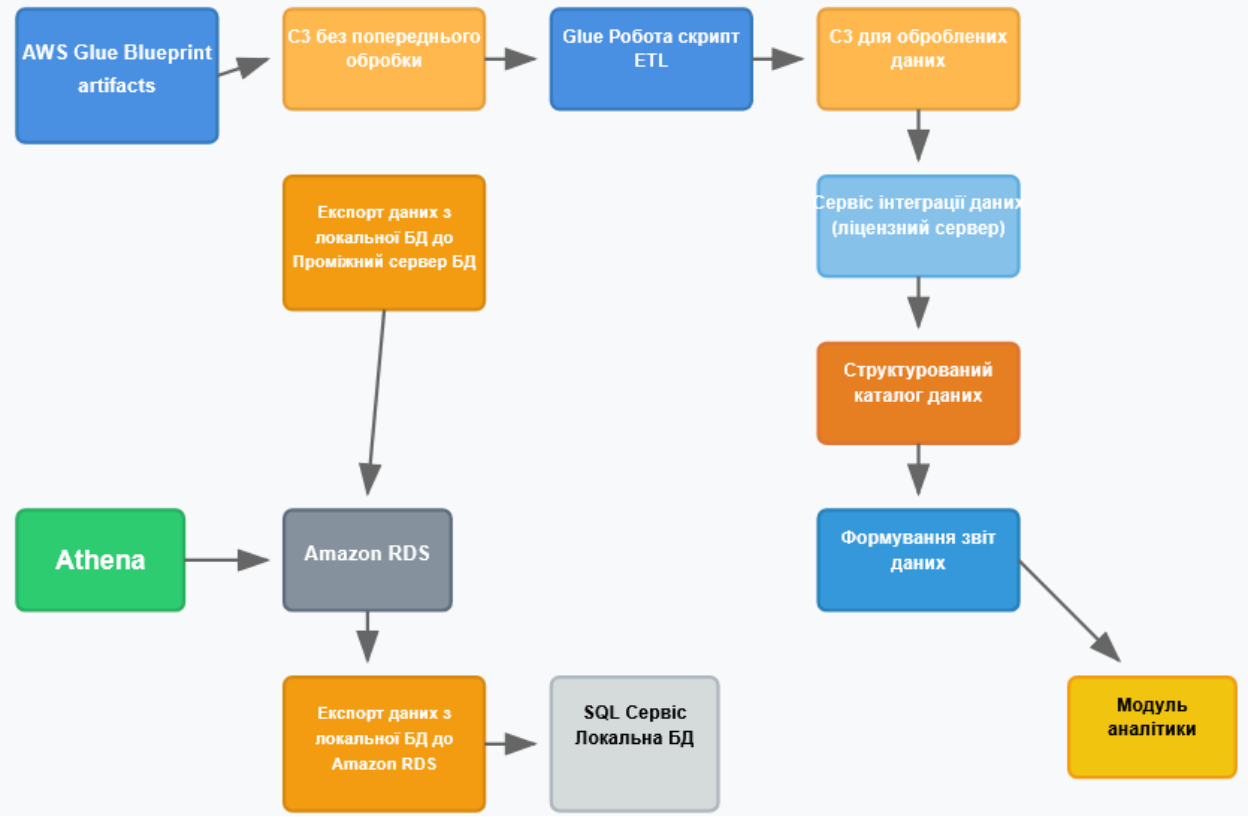


Рисунок 3.8 – Архітектура системи для зберігання та аналізу даних

У результаті підключення розгорнутої інфраструктури до відповідних хмарних служб було сформовано повноцінну аналітичну систему, що підтримує обробку даних на всіх етапах їх життєвого циклу. Така система надає змогу працювати як з новими, щойно доданими даними, так і з попередньо обробленими наборами, забезпечуючи їх доступність для подальшого аналізу.

Реалізована архітектура передбачає масштабованість і витримує навантаження, що виникає як з боку аналітичних інструментів, так і з боку кінцевих користувачів, які взаємодіють із базою даних у режимі запису.

Додатковою перевагою системи є підтримка багатокористувацького доступу з можливістю розподілу ролей, що дозволяє ефективно організувати паралельну роботу працівників в межах єдиного середовища.

Кінцеву архітектуру аналітичної системи представлено на рисунку 3.8.

3.5 Висновки до розділу 3

У ході реалізації аналітичної системи було визначено логічну схему зберігання даних, а також продемонстровано особливості їхнього розподілу по відповідних каталогах на різних етапах обробки. Такий підхід забезпечує чітке структурування процесів трансформації даних і дозволяє контролювати кожну фазу підготовки – від моменту надходження сирих даних до формування структурованих аналітичних витягів.

На прикладах, наведених у відповідних ілюстраціях, показано як здійснюється розмітка даних за типами, які SQL-запити використовуються для аналізу ключових показників діяльності компанії, а також простежено повний життєвий цикл даних – від первинного запису в базу даних до виведення обробленої інформації у зручному для користувача вигляді в аналітичному інтерфейсі.

4 РЕАЛІЗАЦІЯ СИСТЕМИ ТА ПРОВЕДЕННЯ ЕКСПЕРИМЕНТІВ

У цьому підсумковому етапі перевірки функціональності аналітичної системи було здійснено порівняння результатів вибірки з локальної бази даних і даних, збережених у мета-сховищі Amazon S3. Аналіз показав повну відповідність за кількістю записів, що свідчить про коректність реалізованої міграції та реплікації. Водночас виявлено обмеження хмарних служб щодо централізованого моніторингу передачі даних, що ускладнює контроль за системою в реальному часі.

Особливу увагу було приділено проблемі обробки JSON-об’єктів, які через символи форматування викликають помилки при створенні мета-сховищ. Для вирішення цієї проблеми запропоновано або змінити формат файлів з CSV на більш гнучкий (наприклад, JSON або Parquet), або провести попереднє розкриття об’єктів у нові колонки, що виправдано лише в разі їх подальшої аналітичної цінності.

Інша проблема полягає у фрагментарному підході до управління ресурсами: фізична інфраструктура документується конфігураційно, а логіка служб налаштовується вручну через консоль AWS. Оптимізувати цю частину можливо за допомогою AWS CDK, що дозволить централізовано керувати всіма складовими системи мовою програмування, зменшуючи ризики та трудові витрати.

Щодо автоматизації обробки даних, було проаналізовано альтернативу розкладу – використання AWS Lambda, яка запускається по події завантаження нових даних. Такий підхід ефективно вирішує проблему надмірних витрат і забезпечує близькість до режиму реального часу, відповідаючи вимогам сучасних аналітичних систем.

Розроблена аналітична система включає в себе локально розгорнуту БД, віддалену БД, віддалене сховище даних, віддалене озеро даних, проміжний сервер для здійснення міграції даних з локальної БД до

віддаленої, а також наступну реплікацію даних з віддаленої БД до озера даних. Ці компоненти були чітко визначені та описані у конфігураційному файлі з розширенням .yaml.

Даний підхід дозволяє швидко розгортати та видаляти ресурси на обладнанні провайдера, що гарантує відтворюваність, швидке відновлення, а також припинення стягування коштів, у разі повного видалення системи.

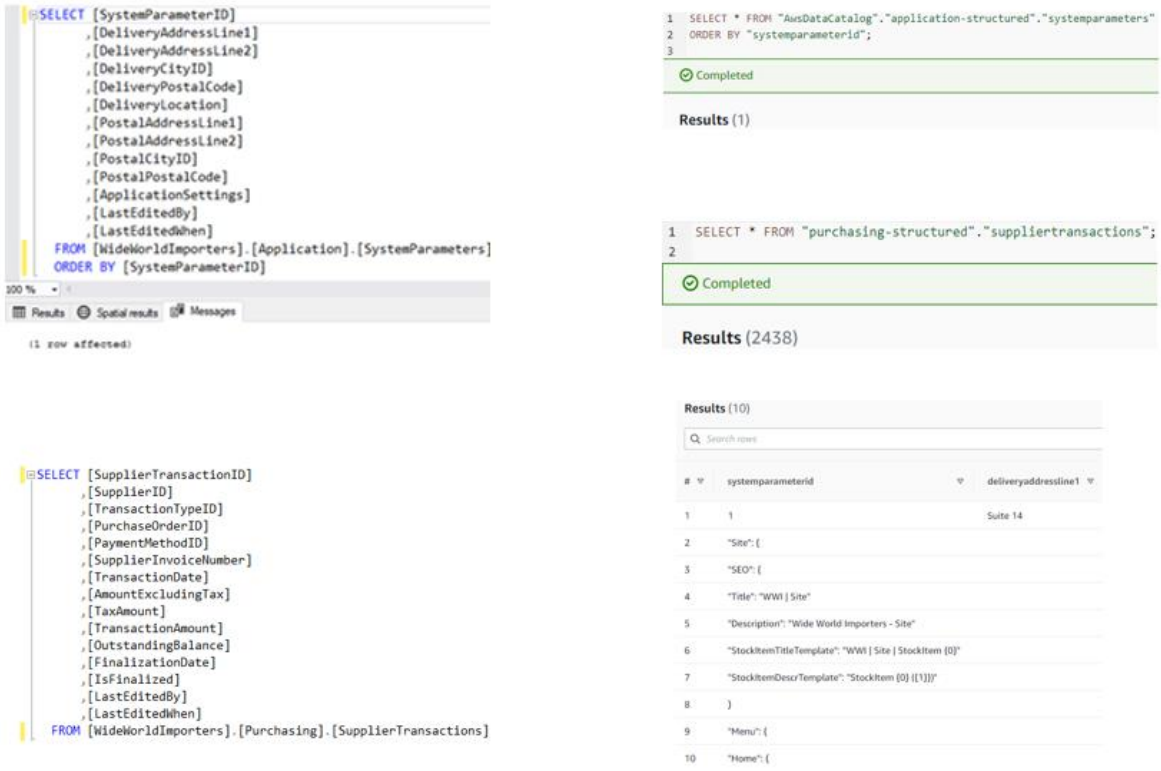


Рисунок 4.1 – Результати роботи

Наступним етапом є визначення типів даних, відповідно до типів, наявних у AWS Glue. За допомогою сервісу Glue, що підтримує мову Python та бібліотеку PySpark здійснюється почергове читання таблиць з каталогу landing-bucket, вигляд схеми одразу після читання таблиці Application.SystemParameters можна спостерігати на слайді. Далі здійснюється іменування колонок та визначення типів даних, відмінних від текстового, відповідно до схеми. Автоматизоване створення мета-сховищ досягається використанням служби Glue Crawlers. Для відображення

результатів обробки та даних та відстеження ключових показників використовується аналітичний інструмент Amazon Quicksight. Для початку потрібно здійснити підключення бажаних джерел даних до аналітичного інструменту.

Аналітична система була розгорнута комбінованим шляхом. Фізичні ресурси були задокументовані у конфігураційному файлі, подальші завдання з міграції, реплікації, обробки даних були виконані з допомогою служб Amazon через взаємодію з Amazon Console. Після того як розгорнута інфраструктура була підключена до відповідних служб маємо завершену аналітичну систему, що дозволяє працювати як з даними, що були щойно додані, так і з обробленими, а також витримує навантаження як від аналітичного інструменту так і від користувачів, що можуть здійснювати записи до БД. Дана аналітична система дозволяє розподіляти ролі між працівниками і забезпечувати їх паралельну роботу в одній системі. Кінцеву архітектуру можна спостерігати на слайді.

Для перевірки коректності роботи спроектованої системи, потрібно порівняти змісти таблиць локальної БД та результати запитів через мета-сховище до structured-bucket каталогу. Порівнюючи результати запитів до локальної БД і мета-сховища, бачимо, що кількість повернутих записів, в обох випадках, однакова, тож втрат даних під час міграції та реплікації не було. Служби Amazon не дозволяють централізовано відслідковувати передачу даних одночасно через декілька сховищ, що є суттєвим недоліком, оскільки потребує від розробників пошуку альтернативних рішень.

ВИСНОВКИ

У ході підготовки кваліфікаційної роботи була розроблена комп'ютерна система збору та обробки та даних з використанням сервісів та інструментів Amazon. Розроблена комп’ютерна система відповідає сучасним вимогам і може бути інтегрована з системами управління даних. Використання системи, дозволяє спростити керування даними, здіснювати їх аналіз, а також додатково підвищує відмовостійкість та забезпечує легку масштабованість.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. https://docs.aws.amazon.com/glue/index.html (дата звертання 25.05.2023)

2. https://aws.amazon.com/ru/blogs/big-data/tag/aws-glue-crawler/ (дата звертання 25.05.2023)

3. M. D. Assunção, R. N. Calheiros, S. Bianchi. “Big Data computing and clouds: Trends and Future Directions”, c. 3-15

4. X. Sun, B. Gao, Y. Zhan, “Towards delivering analytical solutions in cloud: Business models and technical challenges,” - Computer Society, Washington, USA, 2011, c. 347-351

5. D. Talia. “Toward Cloud-based Big-data Analytics,” - IEEE Computer Science, c. 98-101.

6. C. Elena. “Business Intelligence” - Journal of Knowledge Management, Economics and Information Technology, 2011, c. 1-12.