ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Інженерії програмного забезпечення

Пояснювальна записка

до магістерської роботи
на ступень вищої освіти магістр

на тему: **«ВДОСКОНАЛЕННЯ МЕТОДИКИ ОБРОБКИ ДАНИХ КЛІЄНТІВ КОМПАНІЇ ТОРГОВОГО ТИПУ В ХМАРНОМУ СХОВИЩІ НА ОСНОВІ ТЕХНОЛОГІЙ BUSINESS INTELIGENCE»**

Виконав студент: 6 курсу групи ПДМ-61

спеціальності

121 Інженерія програмного забезпечення

Бабак Максим Олексійович

Керівник Садовенко В.С

Рецензент

КИЇВ – 2022

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти «Магістр»

Спеціальність підготовки 121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри
Інженерії програмного
забезпечення
_____ О.В. Негоденко
«___»_____2022 року

З А В Д А Н Н Я

Н А М А Г І С Т Е Р С Ь К У Р О Б О Т У С Т У Д Е Н Т У

Бабак Максим Олексійович

1. Тема Роботи Вдосконалення методики обробки даних клієнтів компанії торгового типу в хмарному сховищі на основі технологій Business Intelligence

Керівник роботи

затверджені наказом вищого начальника закладу від «12» жовтня 2022 року №122

2. Строк подання студентом роботи 31.12.2022

3. Вихідні дані до роботи: матеріали практики, методи моделювання хмарних технологій, документація про архітектуру хмарних рішень

4. Зміст розрахунково-пояснювальної записки (перелік питань які потрібно розробити)

4.1 Огляд предметної області

4.2 Визначити основні характеристики аналітики в хмарах

4.3 Визначити архітектуру аналітичної системи в хмарному середовищі

4.5 Побудувати систему обробки даних клієнтів компанії трогового типу в хмарному середовищі

6. Дата видачі завдання: 14 жовтня 2022 року

КАЛЕНДАРРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Отримання завдання на магістерську роботу	14.10.2022	
2	Вивчення матеріалу	17.10.2022	
3	Аналіз предметної області	20.11.2022	
4	Огляд технологій для побудови системи	22.11.2022	
5	Підготовка матеріалів третього розділу дипломного проєкту	24.11.2022	
6	Розробка системи хмарного аналізу за методикою	26.11.2022	
7	Аналіз розробленого рішення	04.11.2022	
8	Розробка графічних та презентаційних матеріалів	11.12.2022	
9	Захист дипломного проєкту	17.01.2023	

Студент Бабак М.О.

Керівник Садовенко В.С

РЕФЕРАТ

Текстова частина магістерської роботи: 64с., 25 рис., 5 дод., 15 джерел

ВДОСКОНАЛЕННЯ МЕТОДИКИ ОБРОБКИ ДАНИХ КЛІЄНТІВ КОМПАНІЇ ТОРГОВОГО ТИПУ В ХМАРНОМУ СХОВИЩІ НА ОСНОВІ ТЕХНОЛОГІЙ BUSINESS INTELLIGENCE

Об’єкт дослідження – аналітика в компаніях торгово типу.

Предмет дослідження – обробка, зберігання та аналіз даних в хмарному середовищі

Мета роботи – спрощення процесу отримання та аналізу ключових показників та метрик клієнтів для компаній торгового типу.

У даній магістерській роботі проведено аналіз будови сучасних хмарних рішень для аналітики. Були визначені основні переваги використання хмарної інфраструктури. Усі ці характеристики показали актуальність систем на сьогодні, особливо у парі з використанням технологій Business intelligence, які є надзвичайно корисними для бізнесу. У роботі було проведено дослідження побудови типової хмарної аналітики з використанням технологій Business intelligence. У наслідок чого були визначені основні компоненти для аналітики.

У результаті було побудовано аналітичну систему обробки даних клієнтів компанії торгового типу в хмарному сховищі на основі Business Intelligance, що дозволяє працювати як з даними, що були щойно додані, так і з обробленими даними раніше.

Проведено аналіз отриманої системи та методики обробки даних та виявлено декілька недоліків. Було висунуті ідеії для покращення системи з метою усунути ці недоліки.

Використання такої методики, дозволяє спростити керування даними, здійснювати їх аналіз, а також додатково підвищує відмовостійкість та забезпечує легку масштабованість.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ 5
ВСТУП 6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 8
1.1 Хмарні рішення для аналітики 8
1.2 Переваги хмарних технологій 9
1.2.1 Ефективне управління даними 9
1.2.2 Захист від втрати даних 10
1.2.3 Оптимізація витрат 10
1.2.4 Підвищена гнучкість 10
1.2.5 Безпека 11
1.2.6 Мобільність 11
1.2.7 Прозорість 11
1.2.8 Автоматичне оновлювання системи 11
1.3 Моделі хмар 12
1.3.1 Публічна хмара 12
1.3.2 Приватна хмара 12
1.3.3 Гібридна хмара 13
Висновки до розділу 1 13
2 ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ ПОБУДОВИ СИСТЕМИ 15
2.1 Типові рішення реалізації хмарної аналітики 15
2.1.1 Рівень інфраструктури 17
2.1.2 Рівень управління даними 17
2.1.3 Рівень аналітики 18
2.1.4 Рівень візуалізації даних 18
2.2 Огляд сервісів Amazon Web Services 19
2.2.1 Amazon Simple Storage Service 19
2.2.2 Amazon Relational Database Service 21

2.2.3 Amazon Athena 22
2.2.4 Amazon QuickSight 24
2.2.5 Amazon Web Services Glue 25
2.2.6 Amazon Web Services Lake Formation 27
2.3 Infrastructure as code 28
2.3.1 Декларативний підхід 29
2.3.2 Імперативний підхід 30
2.3.3 Недоліки IaC 30
2.3.1 Переваги IaC 30
2.3.2 Amazon Web Services CloudFormation 31
Висновки до розділу 2 33
3 РОЗРОБКА СИСТЕМИ ХМАРНОГО АНАЛІЗУ 35
3.1 Процес розгортання інфраструктури 35
3.2 Сховища даних 36
3.2.1 Локальне сховище бази даних 36
3.2.2 Віддалене сховище 37
3.3 Процес міграції даних. 38
3.3.1 Міграція даних до Amazon RDS з локального сховища 38
3.3.2 Процес реплікації даних 39
3.4 Обробка даних для аналітики 40
3.4.1 Встановлення типів даних 40
3.4.2 Нові набори даних для аналітики 42
3.5 Мета-сховища 45
3.6 Візуалізація даних 47
3.7 Методика обробки даних та архітектура 49
Висновки до розділу 3 50
4 АНАЛІЗ РОЗРОБЛЕНОГО РІШЕННЯ 53
4.1 Перевірка коректності передачі даних 53

4.1.1 Схема Application 53
4.1.2 Схема Sales 54
4.1.3 Схема Warehouse 55
4.1.4 Схема Purchasing 56
4.2 Результати перевірки даних 57
4.3 Недоліки розробленого аналітичного рішення 57
4.3.1 Налаштування взаємодії між службами 57
4.3.2 Довга перерва в оновленні даних 58
Висновки до розділу 4 59
ВИСНОВКИ 60
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 63
Додаток А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація) 65
Додаток Б. СТРУКТУРНА СХЕМА СИСТЕМИ 71
Додаток В. ФУНКЦІОНАЛЬНА СХЕМА (діаграма даних) 72
Додаток Г. ПРИЦНИПОВА СХЕМА (алгоритм роботи програми) 73

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

BI – Business intelligence (бізнес-аналітика)

AI – Artificial intelligence (штучний інтелект)

IaaS – Infrastructure as a Service (інфраструктура як сервіс)

PaaS – Platform as a Service (платформа як сервіс)

SaaS – Software as a Service (програмне забезпечення як сервіс)

IoT – Internet of Things (Інтернет речей)

BigData – Великі дані, набір структурованих і неструктурованих даних

data lakes – озера даних

AWS – Amazon Web Services

API – Application Programming Interface (прикладний програмний інтерфейс)

БД – база даних

ВСТУП

Інформатизація суспільства призводить до появи нових завдань, стосовно збору, зберігання, передачі, захисту та аналізу даних, та спонукає до пошуку способів оптимізації цих процесів.

Без спеціальних інструментів машинної обробки чи додаткових обчислювальних ресурсів неможливо опрацювати великі обсяги інформації. Збільшення обсягу даних робить систему складнішою та, таким чином, підтримування такої системи також ускладнюється. Це призводить до потреби в збільшенні кількості IT-спеціалістів, що потребує більше електроенергії та призводить до збільшення витрат на обслуговування. Гнучкість процесів знижується, а масштабованість бізнесу значно скорочується при зберіганні даних компанії на внутрішніх серверах.

Компанії торгового типу завжди страждали від недостатньої інформації про своїх клієнтів і відсутності швидкого доступу до неї. Це призводило до неефективності роботи і втрати потенційних клієнтів. Одним з рішень цього проблеми є використання технологій business intelligence, які дозволяють збирати, обробляти і аналізувати дані клієнтів у хмарному сховищі.

Об'єкт дослідження – аналітика в компаніях торгово типу

Предмет дослідження - обробка, зберігання та аналіз даних в хмарному середовищі

Актуальність теми дипломної роботи полягає в тому, що ефективність швидкого аналізу ключових показників бізнесу допомагає компаніям формувати статистику або аналітику в потрібний час. Усі ці показники можуть швидко сповістити про якісь негаразди або покращення чи тенденції в бізнесі. Ще зручніше коли інструмент для такого аналізу та зберігання даних є в безпечному місці, яким можна скористатись в будь-який час.

Мета роботи – спрощення процесу отримання та аналізу ключових показників та метрик кілєнтів для компаній торгового типу.

Для досягнення цієї мети бу було проведено дослідження ринку технологій business intelligence і вибрано підходящу платформу для обробки даних. Було створено модель обробки даних, яка включає збір інформації про клієнтів, та збереження у хмарному сховищі. Також було розроблено систему візуалізації даних, яка дозволяє отримувати інформацію у зручному для аналізу вигляді.

Для перевірки ефективності розробленої методики було проведено тестування на наборі даних клієнтів компанії. Результати тестування показали, що використання розробленої методики дозволяє збільшити ефективність роботи компанії та задовольнити потреби клієнтів швидше та краще.

На основі результатів дослідження можна сказати, що розроблена методика обробки даних клієнтів компанії торгового типу в хмарному сховищі на основі технологій business intelligence є швидкою, ефективною та зручною у використанні. Вона дозволяє компанії отримувати достовірну інформацію про своїх клієнтів і швидко реагувати на їхні потреби, що призводить до покращення сервісу і збільшення кількості покупок.

В майбутньому рекомендується використовувати розроблену методику у системі обробки даних компанії та постійно розвивати її у напрямку підвищення ефективності та зручності. Також слід запропонувати розроблену методику іншим компаніям торгового типу як спосіб покращення роботи та задоволення потреб клієнтів.

Також слід продовжувати моніторинг ринку технологій business intelligence і вносити зміни в розроблену методику, якщо є нові технології, які можуть бути корисними для обробки даних клієнтів компанії. Також слід проводити періодичні тестування розробленої методики, щоб виявити її слабкі місця та зробити необхідні корективи.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Хмарні рішення для аналітики

У наш час центрами обробки даних виступають серверні приміщення, що відповідають розміщення IT-інфраструктури. Це допомагає компаніям зберігати та обробляти великі обсяги даних, збільшуючи продуктивність, та спрощує роботу з отриманням більш точних результатів. Саме використання таких центрів обробки даних є більш вигідним для багатьох підприємств, адже обробка даних та її аналіз є більш доступною.

Усі хмарні рішення для аналітики мають основні етапи роботи з даними

– збір даних – це прийом структурованих та неструктурованих даних у різних форматах з різних джерел;

– аналіз якості даних – процес обробки,структуризації даних та підготовка інформації для аналітичних алгоритмів;

– робота з програмним забезпечення для аналітики – процес оброки вже підготовлених даних з отриманням результатів по заданим скриптам;

– налаштування відображення даних – процес візуалізації готових даних, що допомагає бізнесу слідкувати за основними метриками, ключовими показниками ефективності та інше.

Хмарна аналітика – це хмарне рішення, яке дозволяє організаціям виконувати бізнес-аналіз чи аналітичні процедури. Ці ріщення та послуги надаються через хмарні моделі, такі як розміщені сховища даних, SaaS (Software as a Service) Business Intelligence (BI) і аналітичні продукти для соціальних мереж на основі хмари. Сервіси хмарної аналітики працюють як хмарні служби аналізу даних, надаючи аналогічні функції та можливості [1].

Хмарне сховище є еластичним, воно дає можливість вертикального масштабування залежно від попиту за затрат бізнесу на ті ресурси – які використовуються. З його допомогою організації безпечно зберігають дані і інтернеті. Дані доступні для користувачів з будь-який час та з будь-якого місця.

Використання хмарних технологій та великих даних набуває все більше популярності у різних сферах: фінанси, реклама, послуги, виробництво та інші. Ці технології підвищують якість послуг, що надаються, допомагають організаціям скоротити затрати, сприяють росту ефективності їх діяльності.

Компанії майже кожного дня збирають, різні за обсягом та швидкістю структуровані та неструктуровані дані. Аби отримати справжню вигоду від великих даних їх потрібно проаналізувати, а з ними зручно працювати в хмарі: для цього не потрібні величезні приміщення або кімнати для серверів, що зберігають та обробляють дані.

Зараз можна знайти спеціальне програмне забезпечення для будь-яких потреб. Примітивні функції можуть виконуватись базовими інструментами, а рішення AI та машинного навчання заберуть на себе найскладнішу роботу, щоб допомогти бізнесу.

Щоб зрозуміти переваги використання хмарної аналітики, потрібно їх детальніше розглянути

1.2 Переваги хмарних технологій

1.2.1 Ефективне управління даними

Це є основною перевагою. Хмарна аналітика допомагає зробити процес прийняття рішень швидким та ефективнішим і все це дає результати у короткий термін. Якщо IT-відділ займається контролем власних даних їм може бути важко приймати швидкі та обґрунтовані рішення. А використовуючи хмарну аналітику спеціалісти цього відділу можуть зосередити увагу на логіці, відкинувши усі проблеми з інфраструктурою. Усе це допомагає швидше досягати поставлених цілей завдяки своєчасно отриманій інформації: залучення нових клієнтів, збільшення продажів та активності на продукті, що тягне на собою підвищення доходів компанії.

1.2.2 Захист від втрати даних

У наш час всі розуміють як важливо захистити себе від втрати даних. Саме тому, люди можуть створювати безліч резервних копій файлів на різних девайсах або в хмарі.

Така ж ситуація зараз і в бізнесі. В любий момент можна втратити всі історичні дані. Якщо зберігати інформацію, документи або важливі файли лише на одному девайсі, диску тощо, одна катастрофа може знищити всю кампанію.

Хмарне сховище дає змогу користувачеві зберігати дані різного обсягу далеко від офісу чи дому та в різних місцях. При втраті даних сервіс може ввімкнути протокол відновлення та повернутися до початкової стадії. Для запобігання таких ситуацій, постачальники хмарних послуг тиражують дані на кількох серверах. Це і є гарантією захисту від втрати даних.

1.2.3 Оптимізація витрат

Хмарні рішення нерідко є дешевшими, за допомогою використання логіки Pay as you go, сервіс може надавати тільки ті інструменти – які компанія потребує. Хмарні сервіси надають можливість додавати або прибирати ресурси хмарного сховища по запиту клієнта. Перекидання робочих навантажень сховища з локального середовища в хмару дає можливість усунути надмірне виділення ресурсів та може позбавити від затрат на використання інфраструктури сховища.

1.2.4 Підвищена гнучкість

Використовуючи хмарні рішення клієнт позбавляє себе необхідності інтегрувати додаткові фізичні ресурси, якщо потрібно збільшити масштаб.

Хмарні сервіси інтегруються з широким асортиментом аналітичних інструментів, саме тому в будь-який час можна отримати доступ до більшої кількості ресурсів, а коли виникне необхідність – можна від них відмовитись задля заощадження коштів.

1.2.5 Безпека

Хмарне сховище дає можливість клієнтам контролювати де зберігаються їх дані, хто може отримувати до них доступ и які ресурси використовуватиме ваша організація в будь-який момент часу. Дозволи та контроль доступу повинні працювати в хмарі так само, як і в локальних сховищах. Також хмарні сервіси обладнані шифруванням даних у місці зберігання або під час передачі даних

1.2.6 Мобільність

У наш час віддалена робота набула неабиякої популярності. Особливо коли працівники однієї компанії можуть бути розкинуті не тільки по різним областям однієї країни, а по куточкам усього світу. У таких випадках хмарні рішення можуть бути дуже корисними. Співробітнику компанії достатньо просто мати підключення до інтернету. Йому потрібно увійти в обліковий запис щоб отримати доступ до потрібних даних.

1.2.7 Прозорість

Хмарні рішення є доволі інтуїтивно зрозумілими у використанні. Такі сервіси об’єднують в собі всю інформацію компанії, що дає змогу отримати повне уявлення про бізнес-процеси організації. Усі співробітники мають змогу, незалежно від їх фізичного місцезнаходження, легко отримати доступ до даних, обмінюватись даними та мати інформацію про основні напрямки компанії.

1.2.8 Автоматичне оновлювання системи

Як вже згадувалось вище, хмарні сервіси є простими у використанні, ними можна користуватись без попередньої підготовки. А ще однією перевагою таких сервісів є автоматичне оновлення зі сторони сервісу. Для цього не потрібно задіювати IT-персонал, що може зекономити затрати часу та коштів.

1.3 Моделі хмар

Всього існує три види хмар: публічні, приватні та гібридні. Тип хмари обирається залежно від потреб клієнта, адже певний з видів може використовуватись краще для конкретних завдань.

1.3.1 Публічна хмара

Публічна хмара – це модель, в якій обчислювальні служби та інфраструктура надаються стороннім постачальником та використовуються кількома компаніями через мережу інтернет. Тобто публічна хмара може використовуватись кількома користувачами.

Сервіс може надаватись у вигляді: інфраструктура як послуга (IaaS), платформа як послуга (PaaS) або ПЗ як послуга (SaaS).

Публічна хмара дає можливість швидкого розгортання, та масштабування і є доволі економною у затратах.

Оскільки на серверах публічної хмари одночасно використовуються дані декількох компаній, забезпечення безпеки – є однією з головних проблем. Тут потрібно приділити особливу увагу шифруванню даних.

1.3.2 Приватна хмара

Приватна хмара є вже моделлю хмарних обчислень з виділеною інфраструктурою для одного користувача. Тобто доступ надається тільки тим, хто знаходить всередині приватної мережі.

Плюсом цієї хмари є швидка масштабованість та самообслуговування, а також високий рівень конфіденційності та безпеки. Але, масштабування обмежене ресурсами обладнання на якому розгорнута хмара, тому забезпечити розгортання інфраструктури може бути важче ніж в публічній хмари. Ще одним недоліком можуть бути невиправдані затрати на дорожчі ресурси та обладнання у випадку недостатньо активного використання хмари.

1.3.3 Гібридна хмара

Гібридна хмара – це модель у якій частина інфраструктури розміщується у приватній хмарі, а частина – в публічній. Це надає компаніям переваги обох попередніх підходів та можливість вибору потрібного типу хмари з унікальними вимогами до обробки даних. Наприклад, гібридна хмара пропонує альтернативу для зберігання конфіденційних даних: компанія може надавати послуги через публічну хмару і зберігати конфіденційну інформацію в приватній.

Отже, компанія має усі переваги хмарної аналітики, включаючи безпеку, невисокі затрати та масштабування.

Висновки до розділу 1

У першому розділі магістерської роботи були розглянуті варіанти побудови аналітичних рішень за допомогою хмарної інфраструктури

Було розглянуто основні переваги використання хмарної інфраструктури: ефективне управління даними, захист від втрати даних, оптимізація витрат, підвищена гнучкість, безпека, мобільність, прозорість, автоматичне оновлення системи.

Наведені характеристики доводять актуальність використання хмарних технологій у нас час. Ці сервіси надають можливість компаніям безпечно зберігати дані з можливістю відновлення, отримувати актуальну інформацію віддалено, зменшувати обсяг витрат, відслідковувати інформацію по всім важливим бізнес-метрикам, отримувати звіти та візуалізацію даних. Також сервіси є простими у використанні для клієнта, що значно зменшує витрату часу на підготовку до використання інструменту.

Після формування усіх переваг хмарних технологій були описані типи хмар за їх доступністю: публічні, приватні та гібридні.

Аналітичні рішення, що не взаємодіють з конфіденційними або іншими чутливими даними можуть використовувати публічну хмару через її просто масштабованість та ціну.

Приватна або гібридна хмара є гарним варіантом для аналітичних рішень клієнтів, хто збирається використовувати конфіденційні дані. У такому випадку дані будуть зберігатись на окремих серверах до яких будуть мати доступ тільки ті, хто знаходиться всередині приватної мережі. Гібридна ж хмара дасть можливість розміщати іншу частину інформації на публічних серверах.

Після проведеного аналізу було прийнято рішення про використання публічної хмари, оскільки система аналізу даних не буде містити приватної інформації.

2 ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ ПОБУДОВИ СИСТЕМИ

2.1 Типові рішення реалізації хмарної аналітики

У сьогоденні багато організацій покладаються на необмежену потужність обчислення, зберігання та аналіз даних в хмарних сервісах щоб мати можливість бачити, аналізувати та передбачати поведінку своїх основних показників. Усю інформацію клієнти можуть побачити за допомогою використання технологій business intelligence (BI), що передбачає звіти та візуалізацію, та прогнози що будуються на основі машинного навчання з використанням artificial intelligence (AI) [2].

Дані можна обробляти пакетами або в режимі реального часу, локально або в хмарі, але головною метою будь-якого аналітичного рішення є – використання даних в масштабі. Все більше організацій хочуть створити одне істотне джерело для всіх реляційних та нереляційних даних, які генеруються людьми машинами та інтернетом речей (IoT) [3]. Доволі пошиерним рішенням є використання архітектури BigData або IoT для перетворення необроблених даних в структрований формат, а вже потім переміщати їх в аналітичні сховища даних. Це сховище і є тим самим одним істотним джерелом, яке може використовувати безліч аналітичних рішень.

Важливими компонентами для аналітики даних виступають: джерела даних, логіка обробки даних, обчислювальна потужність системи, аналітична модель, моделі представлення даних, зберігання та відображення результатів.

Всього можна виділити чотири рівні в архітектурі хмарної аналітики, їх також можна побачити на рисунку 2.1:

– рівень інфраструктури (Infrastructure layer);

– рівень управління даними (Data Management layer);

– рівень аналітики (Analytics layer);

– рівень візуалізації (Visualization Layer).

Також присутній спільний рівень який включає в себе безпеку, управління інформацією та відповідність нормативним вимогам. Управління безпекою займається управлінням доступом, аутентифікацією користувача, захистом даних та додатків в архітектурі.

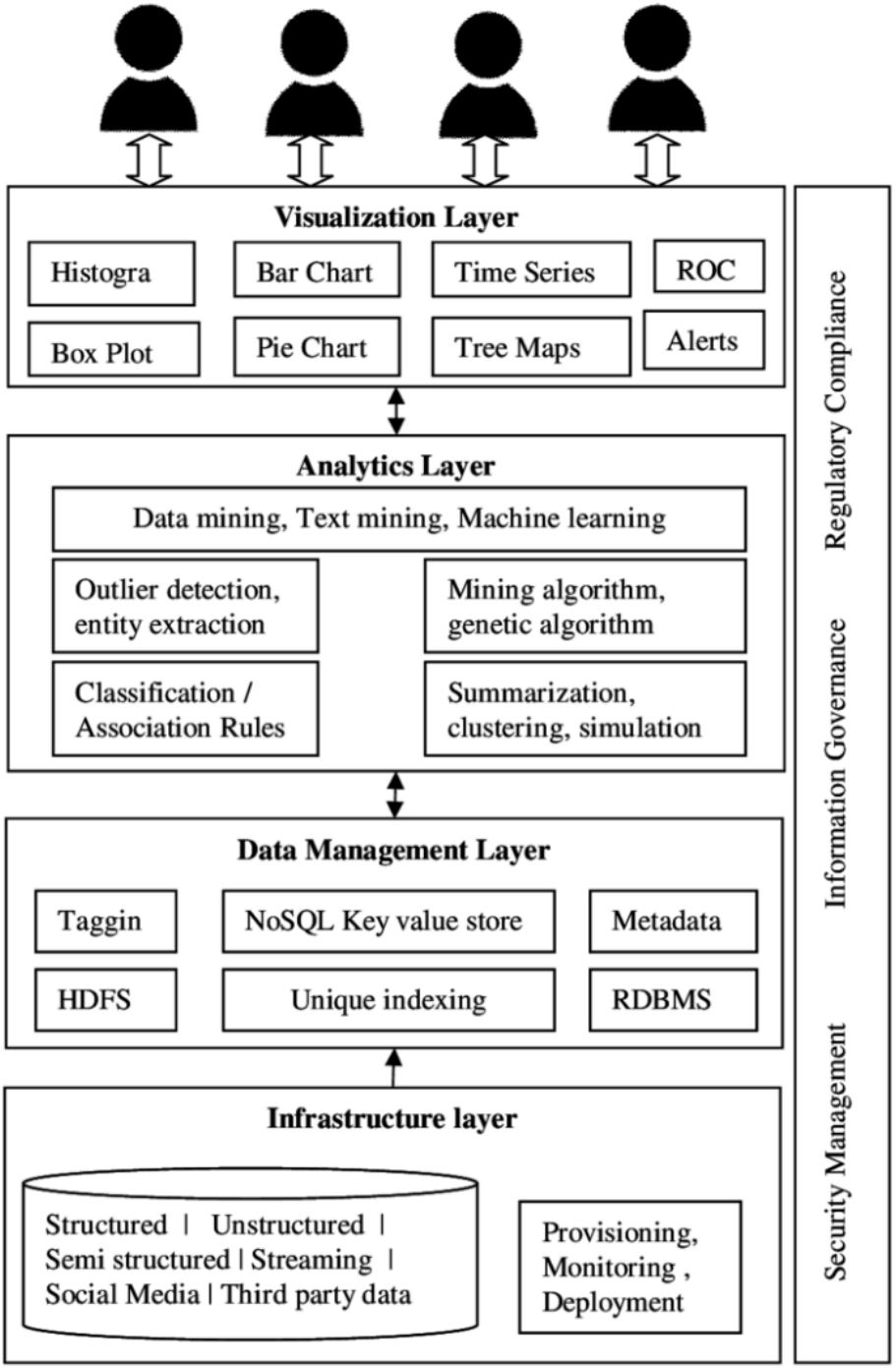


Рисунок 2.1 – архітектура типового хмарного рішення.

2.1.1 Рівень інфраструктури

Рівень інфраструктури відповідає за зберігання та управління даними. Він складається з структурованих та неструктурованих даних, зібраних з різних джерел даних. Як приклад: дані з різних датчиків, дані з мереж, відео, аудіо та інші.

На цьому рівні відбуваються дії керування інфраструктурою: автоматичне розгортання, моніторинг, динамічне надання. Високі навантаження, що тягнуть за собою потреби у швидкому зростанню розміру бази даних або її потужностей обробляються за допомогою можливостей горизонтального і вертикального масштабування хмарної інфраструктури.

Це фундаментальний рівень архітектури хмарної аналітики, який дозволяє організаціям використовувати швидке розгортання інфраструктури.

2.1.2 Рівень управління даними

Цей рівень зберігає в собі дані у структурованому, неструктурованому та напівструктурованому форматах у вигляді flat-файлів. Усі вони зберігаються для аналітичних цілей, схема та формат цих файлів є невизначеними доки не буде сформувати запит. Кожен елемент даних цього шару прив'язаний до унікального ідентифікатора та позначений метаданими Також цей рівень має назву як “озера даних” (data lakes).

Рівень управління даними підтримується з використанням сховища об'єктів, орієнтованого на Hadoop. Основною відмінністю такого виду сховища від корпоративних сховищ даних є використання обробки shema-on-read, коли схема вже визначається тільки під час читання даних. Такий спосіб є більш гнучким. Для збереження відмовостійкості аналітичної хмари використовується автоматична реплікація даних між вузлами хмарного кластеру

2.1.3 Рівень аналітики

Аналітичний рівень архітектури в хмарній аналітиці включає у себе компоненти, які виконують аналіз даних. Це може включати в себе системи збору, зберігання, обробки та виведення даних, а також різноманітні інструменти аналізу, такі як зведення статистики, машинне навчання, бізнес-інтелект та візуалізація. Цей рівень може виконуватися за допомогою віддалених серверів у хмарі, або на місцевих пристроях, які підключені до хмарної системи. Рівень може бути спрямований на задачі різної складності, включаючи звичайне зведення статистики, дослідження даних, створення моделей машинного навчання та побудову прогнозів. Самі різні алгоритми аналізу використовуються на цьому рівні для отримання інсайтів. Такі методи як: класифікація та регресія можуть використовуватись для побудови прогнозних моделей.

Отже, можна сказати, що це один з найважливіших рівнів хмарної аналітики який містить в собі програми бізнес-аналітики.

2.1.4 Рівень візуалізації даних

Цей рівень відповідає за виведення та візуалізацію даних. Його мета - дозволити користувачам зрозуміти та використовувати дані, за допомогою інтуїтивно зрозумілих інструментів та візуалізацій.

Рівень візуалізації може бути реалізований за допомогою різних інструментів та технологій. Наприклад, спеціалізовані інструменти візуалізації, такі як Tableau, Looker, Power BI, QlikView і інші, можуть бути використані для створення графічних представлень даних, таких як діаграми, таблиці, інтерактивні карти та інші. Інші інструменти, такі як бібліотеки JavaScript (D3.js, Highcharts.js), можуть використовуватися для створення візуалізації на веб-сторінках і веб-додатках, даючи користувачам можливість інтерактивно взаємодіяти з даними. Також, використовуються BI-інструменти, такі як

Looker, суттєво полегшують і автоматизують процес аналізу та візуалізації даних в організації.

Рівень візуалізації також може включати в себе функції зворотнього зв'язку, такі як збір даних про користувачів і відгуки, які дозволяють зробити висновки для подальшого аналізу та покращення.

В цілому рівень візуалізації дозволяє користувачам інтерпретувати результати аналізу та отримати значущі інсайти для подальшої діяльності організації.

2.2 Огляд сервісів Amazon Web Services

AWS(Amazon Web Services) надає великий вибір сервісів аналітики, які дозволяють організаціям різного розміру і з будь-якою сферою діяльності переглянути свої бізнес-процеси з допомогою даних. Від переміщення даних, збереження даних, створення озер даних, аналізу великих об'ємів даних, журналів аналітики, стрімового аналізу та машинного навчання, AWS пропонує послуги, спеціально розроблені для отримання максимальної ефективності, продуктивності, масштабованості і найнижчої вартості [4].

2.2.1 Amazon Simple Storage Service

Amazon Simple Storage Service (Amazon S3) є одним із послуг хмарного сховища даних, які надає Amazon Web Services (AWS). Amazon S3 дає користувачам можливість зберігати, передавати і відтворювати дані у хмарному середовищі [5].

Amazon S3 надає широкий вибір можливостей для зберігання даних, включаючи:

– базове зберігання: це найбільш об'єктивне зберігання, яке забезпечує високу доступність, достатню для завантаження та завантаження файлів;

– зберігання із запасними копіями: Це зберігання даних, яке автоматично створює запасні копії даних в інші регіони, щоб забезпечити захист від випадкових втрат даних;

– зберігання статичних веб-ресурсів: для зберігання контенту сайтів такого як зображення, відео, CSS і JavaScript файли;

– зберігання даних з високою доступністю: це зберігання даних, яке забезпечує швидке завантаження даних, що використовується для запуску великих і високопродуктивних завдань;

– зберігання даних для архівації та довгострокового зберігання: це зберігання даних, яке надає низькі витрати на зберігання, але має більш повільну швидкість доступу до даних.

Amazon S3 - це широко використовувана служба хмарного зберігання даних, яка забезпечує високу масштабованість, доступність та безпеку даних, а також високу продуктивність.

Amazon S3 можна використовувати з ціллю:

– створення озер даних;

– збереження критичних даних та резервних копій;

– архівування даних;

– запуск хмарно-орєнтованого додатку.

Amazon S3 надає безліч способів категоризувати та звітувати дані, завдяки унікальним іменам каталогів, префіксам, тегам об'єктів та інвентарю. З можливістю налаштування додаткових функцій S3 для автоматизації дій, S3 Batch Operations значно полегшує керування даними в Amazon S3 будь-якого масштабу. За допомогою пакетних операцій S3 можливо копіювати об'єкти між каталогами, замінювати теги, змінювати контроль доступу та відновлювати архівні об'єкти з класів зберігання S3 Glacier Flexible Retrieval та S3 Glacier Deep Archive за допомогою одного API запиту або кількох кліків у консолі S3.

2.2.2 Amazon Relational Database Service

Amazon Relational Database Service (RDS) є веб-службою, яка дозволяє створювати, конфігурувати та керувати базами даних реляційного типу у веб-середовищі. RDS підтримує бази даних MySQL, MariaDB, PostgreSQL, Oracle, та Microsoft SQL Server, які можна запускати на віртуальних машинах Amazon Elastic Compute Cloud (EC2) [6].

RDS надає велику кількість функцій як бекапи, масштабування, моніторинг та керування базою даних для полегшення адміністрування баз даних. Цей інструмент також може автоматично шардувати базу даних, що дозволяє зберігати дані і підтримувати високу доступність бази даних.

Amazon RDS здійснює основні функції бази даних, такі як моніторинг, автоматичне оновлення, створення резервних копій, відновлення, виявлення та виправлення несправностей.

Amazon RDS пропонує можливість просто створювати реплікації баз даних для збільшення доступності та надійності. Реплікації можуть бути створені у різних регіонах, що дозволяє забезпечити доступність бази даних у разі відмови сервера або збільшення навантаження. Також є можливість використання опції розгортання кількох зон доступності, що дозволяє запускати критично важливі робочі навантаження з високою доступністю та вбудованим автоматичним переходом від основної бази даних до синхронно реплікованої вторинної бази даних. Репліки, що дозволяють тільки читання, також можуть бути використані для вирішення важких робочих навантажень бази даних.

Використання Amazon RDS має кілька переваг порівняно з локально розміщеними базами даних:

– низькі витрати на обслуговування: Ви не повинні відводити ресурси на адміністрування та оновлення баз даних, тому що Amazon RDS автоматично здійснює ці завдання;

– висока доступність: Amazon RDS може забезпечити високу доступність бази даних, використовуючи механізми реплікації та зони доступності;

– швидке масштабування: Amazon RDS може бути легко масштабована за допомогою параметрів та кнопок керування, щоб відповідати зміні навантаження;

– резервне копіювання та відновлення: Amazon RDS автоматично здійснює резервне копіювання баз даних та має можливість швидкого відновлення

– керування безпекою: Amazon RDS надає вбудовані механізми безпеки, такі як контролю доступу та шифрування даних, щоб захистити ваші дані від несанкціонованого доступу;

– надійність: Amazon RDS забезпечує надійну роботу баз даних, і використовує систему автоматичного відновлення для повернення до робочого стану у разі виникнення проблем;

– економія ресурсів: використовуючи Amazon RDS ви можете економити на адмініструванні власної бази даних, маєте можливість масштабувати необхідні ресурси для вашого додатку та знижуєте ризики від відмови послуг з боку власної бази даних;

– підтримка багатьох типів баз даних: Amazon RDS підтримує багато популярних типів баз даних, таких як MySQL, PostgreSQL, SQL Server, Oracle та багато інших.

– вбудовані моніторинг та аналітика: Amazon RDS містить вбудовані механізми моніторингу та аналітики, що дозволяють вивчати продуктивність бази даних та шукати несправності;

– відсутність необхідності інвестувати в жорсткий диск та контрольну панель: Всі бази даних зберігаються на серверах Amazon та доступні для користування за допомогою мережі Amazon Web Services.

2.2.3 Amazon Athena

Amazon Athena це сервіс на основі платформи Amazon Web Services (AWS), який дозволяє здійснювати аналіз даних у форматі сховища даних Amazon S3, використовуючи мову SQL [7].

Athena не потребує налаштування серверів або адміністрування баз даних, дозволяє запускати запити до даних в режимі реального часу, і повертає результати запитів у форматі CSV або JSON.

Сервіс використовує проектування даних Apache Parquet для збільшення швидкості запитів та розжалювання даних. Вона також підтримує додаткові формати, такі як ORC та Avro.

Athena може використовуватися для розгляду даних різного роду, такі як логи, транзакції, дані IoT, фінансові дані і т.д. Вона може бути використана сумісно з іншими сервісами AWS, такі як Amazon QuickSight для візуалізації даних та Amazon Glue для ETL-обробки.

Athena також має можливість налаштування доступу та контролю доступу через AWS Identity and Access Management (IAM). Це дозволяє контролювати які користувачі та системи мають доступ до даних та можуть виконувати які дії з ними.

Amazon Athena є оплачуваним за використання сервісом, ви оплачуєте тільки за об'єм даних, які ви запитуєте та за обсяг сховища даних, які ви використовуєте.

В цілому Amazon Athena є потужним інструментом для аналізу даних і доступним інструментом для здійснення ад-hoc запитів до великих наборів даних без необхідності створення та керування інфраструктурою.

Рисунок 2.2 показує стандартний конвейер обробки даних, у якому дані отримуються з різних джерел і зберігаються в каталогах S3. Це вихідні дані, що означає, що до них ще не застосовувались жодні перетворення.

На цьому етапі можна використовувати Amazon Athena, щоб підключитися до цих даних в S3 та почати їх аналіз. Сам процес є доволі простим, оскільки не потрібно налаштовувати базу даних або зовнішні інструменти для запиту необроблених даних. Після завершення аналізу та отримання потрібних результатів можна використовувати різні аналітичні інструменти для очищення та обробки необроблених даних зі збереженням їх в Amazon S3.

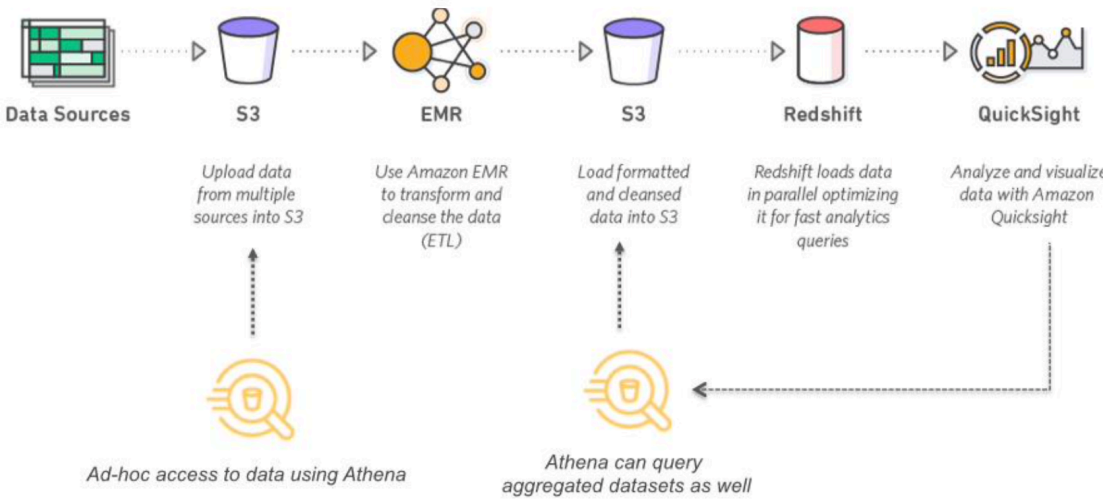


Рисунок 2.2 – типова комунікаційна лінія

2.2.4 Amazon QuickSight

Amazon QuickSight є сервісом візуалізації даних, який надає корпоративним користувачам інструменти для зручного перегляду, аналізу та візуалізації даних. QuickSight може зв'язуватися з багатьма джерелами даних, включаючи Amazon S3, Amazon RDS, Amazon Redshift, та багато інших сервісів та зовнішніх даних. Після з'єднання з даними, користувач може створювати інтерактивні звіти і діаграми, що дозволяє побудувати різноманітні статистики та графіки. QuickSight також містить можливість відображення даних у реальному часі та налаштування розгорнутого доступу до даних іншим користувачам компанії. Платформа QuickSight дозволяє користувачам робити наведення на дані і виконувати аналіз шляхом застосування різних фільтрів та методів візуалізації [8].

QuickSight має можливості спілкування з даними у режимі реального часу, доволі простий інструмент у використанні, має високу продуктивність аналізу, можливість підтримки корпоративних принципів і безпеку. Цей сервіс дає змогу створювати діаграми і звіти з даних, що дозволяє корпоративним користувачам здійснювати більш ефективний аналіз і приймати обґрунтовані рішення.

AWS QuickSight дозволяє легко інтегрувати дані з різноманітних джерел та надає інструменти для створення корпоративної структури контролю та

масштабування кількості користувачів, без необхідності витрачати власні ресурси на інфраструктуру та керування.

Тому можна виділити основні переваги цього сервісу:

– легкість інтеграції: QuickSight може безпосередньо підключатися до багатьох різних джерел даних, таких як Amazon S3, Amazon Redshift, Amazon RDS та бази даних сторонніх розробників, такі як MySQL, Oracle, SQL Server та PostgreSQL;

– швидкість аналізу: QuickSight має вбудовані інструменти для аналізу даних, які дозволяють користувачам швидко та ефективно отримувати інсайти з даних, без використанням візуалізацій та діаграм, які можна швидко створювати без потреби в досвідченому розробнику або аналітик;

– корпоративне керування: QuickSight надає інструменти для створення ролей та контролю доступу для користувачів, які дозволяють керувати переглядом і аналізом даних в корпоративному масштабі;

– масштабованість: QuickSight має масштабовану архітектуру, яка дозволяє користуватися системою з великою кількістю користувачів без необхідності витрачати ресурси на керування інфраструктурою;

2.2.5 Amazon Web Services Glue

Amazon Web Services (AWS) Glue є платформою для етикетування та обробки даних, яка дозволяє користувачам автоматизувати процеси зіставлення даних та екстракції, трансформації та завантаження (ETL) даних. Glue використовує спеціально проектовану модель даних, названу Data Catalog, щоб дозволити користувачам описувати та визначати дані, які вони планують обробляти [9].

Glue може бути використаний для зіставлення та екстракції даних з різних джерел, таких як бази даних, кластери Hadoop, файли в форматі CSV та JSON, а також інші види файлів. Дані можуть бути експортовані до різних місць назначення, таких як Amazon S3, Amazon Redshift, Amazon RDS та інші сервіси AWS.

AWS Glue складається з двох основних компонентів: Glue ETL-інструмент і Glue Data Catalog.

– Glue ETL-інструмент: Це інструмент для створення і виконання скриптів ETL (Extract, Transform, Load). Ви можете створити скрипт на Python або Scala, щоб екстрагувати дані з різних джерел, проводити перетворення та завантажувати дані до місця назначення;

– Glue Data Catalog: це сервіс, який використовується для каталогування даних, які були оброблені Glue. Він містить метадані, такі як опис таблиць, схеми та зв'язки, з кожної бази даних або файлу, який був оброблений Glue. Користувач може використовувати Data Catalog, щоб знайти та доступитися до даних з одного централізованого місця, за допомогою SQL-запитів.

Також Glue може бути використаний для налаштування роботи з даними, такі як створення зв'язків між таблицями, класифікації типів даних та створення робочих джерел для використання з Інсайт, QuickSight та іншими сервісами AWS.

AWS Glue автоматично масштабує ресурси залежно від навантаження, дозволяючи розподілити робоче навантаження ефективно. Ви не повинні турбуватися про надмірну виділені ресурси, витрачати час на оптимізацію кількості працівників або платити за незадіяні ресурси, коли завдання завершуються.

AWS Glue має деякі обмеження та недоліки. Одним з них є те, що він може працювати лише з даними в S3 та базами даних на Amazon RDS або Amazon Redshift. Якщо ваші дані знаходяться в іншому середовищі, ви можете знайти збої в підключенні до них з допомогою Glue.

Інший недолік може бути пов'язаним з можливістю автоматичного визначення схеми даних. В деяких випадках Glue може не коректно визначити схему даних, тому може знадобитися ручне налаштування.

Ще один недолік може бути пов'язаний з можливістю швидко змінювати конвеєри перетворення даних. В деяких випадках перетворення можуть бути

складні і вимагати ручного запису коду, що може збільшувати час і зусилля при внесенні змін.

Але у цього сервісу є і декілька важливих переваг:

– автоматичне масштабування: може динамічно масштабувати ресурси відповідно до робочого навантаження, що дозволяє зберігати високу продуктивність та ефективність;

– сімейство інструментів: надає широкий набір інструментів для керування даними, які дозволяють полегшити міждисциплінарні завдання в обробці даних, наприклад, збереження, перетворення, аналіз та зв'язок даних;

– інтеграція з іншими сервісами AWS: інтегровано з багатьма іншими сервісами AWS, такими як Amazon S3, Amazon RDS, Amazon Redshift, та понад 20 Інші сервіси, що дозволяє легко інтегрувати дані з різних джерел і використовувати їх для аналітики та прийняття рішень;

– простота використання.

2.2.6 Amazon Web Services Lake Formation

Amazon Web Services (AWS) Lake Formation це сервіс, який допомагає з налаштуванням, розгортанням та керуванням централізованою облаштованою корпоративною data lake. Він дозволяє користувачам вводити дані з різних джерел, створювати репліки, захищати дані і автоматизувати процес організації і керування даними. Цей сервіс надає користувачам зручні інструменти для створення та розгортання схеми даних, маппінг даних, індексації та складання зпитів, дозволяє розпочинати аналітику даних з певної точки і оптимізувати роботу з даними для досягнення найкращих результатів [10].

Озеро є централізованим репозиторієм даних, який зберігає всі види даних в їх оригінальному форматі. Це дозволяє користувачам відкривати та аналізувати дані за допомогою будь-яких інструментів, які вони вибирають.

AWS Lake Formation надає простий спосіб для створення та керування вашою озером даних на основі AWS. Він дозволяє зробити ваші дані безпечними, керувати доступом до даних та налаштовувати пароль за

допомогою декількох класів служб, таких як AWS Glue, Amazon S3, Amazon RDS, Amazon Redshift та AWS Identity and Access Management (IAM). Цей сервіс дозволяє користувачам не тільки зберігати дані, але й проводити аналітику над ними в будь-якому масштабі.

Amazon Web Services Lake Formation допомагає зібрати та каталогізувати дані з різних джерел, таких як бази даних та сховища об'єктів, перенести їх в нове Amazon S3 озеро даних, очистити і класифікувати дані за допомогою алгоритмів машинного навчання та забезпечити безпечний доступ до конфіденційних даних з детальним контролем на рівнях стовпця, рядка та комірки. Lake Formation дозволяє користувачам доступати до централізованого каталогу даних, який описує доступні набори даних та їх відповідні контексти використання. Потім, користувачі можуть використовувати ці набори даних для обрання сервісів аналітики і машинного навчання, таких як Amazon Redshift, Amazon Athena, Amazon EMR для Apache Spark та Amazon QuickSight. AWS Glue надає функціональність, яку використовує Lake Formation.

2.3 Infrastructure as code

Infrastructure as Code (IaC) - це практика, яка дозволяє автоматизувати розгортання та управління інфраструктурою за допомогою коду, який можна контролювати та вести версію. Це дозволяє використовувати принципи контролю версій, автоматизації та скриптування для спрощення управління інфраструктурою, зниження часу й затрат на розгортання та оновлення інфраструктури, а також збільшення надійності та відновлюваності систем [11].

IaC можна застосовувати для різних частин інфраструктури, таких як сервери, мережі, хмарні ресурси, бази даних та багато іншого. Зазвичай для цього використовуються спеціальні інструменти такі як Terraform, Ansible, Chef, Puppet.

Іншими словами, IaC – це практика, що використовує код та автоматизовані процеси для керування інфраструктурою вашої системи.

Використовуючи цей метод, можна створювати файли конфігурації, які містять специфікації інфраструктури вашої системи, і використовувати їх для автоматизованого створення, конфігурування, оновлення та керування всіма компонентами.

При використанні IaC, створення та керування інфраструктурою відбувається за допомогою файлів конфігурації, які містять специфікації інфраструктури. Це дозволяє контролювати версії і документувати зміни, а також дозволяє розділяти інфраструктуру на модульні компоненти, які можуть автоматично комбінуватися. Контроль версії та автоматизоване розгортання дозволяють керувати інфраструктурою більш ефективно.

Інфраструктура як код (IaC) дозволяє розробникам автоматизувати надання і керування інфраструктурою, такими як сервери, операційні системи, сховища, за допомогою файлів конфігурації, завдяки чому не потрібно виконувати ці дії вручну під час розробки або розгортання програм.

Є 2 види підходи до IaC – декларативний та імперативний.

2.3.1 Декларативний підхід

Декларативний підхід IaC - це спосіб керування інфраструктурою, при якому спеціалісти визначають бажаний стан інфраструктури за допомогою конфігураційного файлу, уникаючи ручного налаштування. Цей підхід дозволяє автоматизувати процеси створення та налаштування інфраструктури, зменшує кількість помилок та покращує сумісність системи.

Підхід дозволяє легко відслідковувати та контролювати зміни в інфраструктурі, а також легко відновлювати систему в разі помилок. Це дозволяє забезпечити більшу надійність та доступність інфраструктури. Декларативний підхід також покращує співпрацю команд та зменшує залежність від конкретних спеціалістів.

IaC з використанням цього підхду може бути реалізований за допомогою різних інструментів та технологій, таких як Terraform, Ansible, Puppet та інші.

Він дозволяє організаціям створювати та керувати інфраструктурою з максимальною ефективністю та надійністю.

2.3.2 Імперативний підхід

Імперативний підхід IaC - це спосіб керування інфраструктурою, при якому спеціалісти визначають кроки, які потрібно виконати для досягнення бажаного стану інфраструктури. Цей підхід зосереджується на ручному налаштуванні системи, а не на визначенні бажаного стану.

Цей підхід дозволяє більш детально контролювати процес створення та налаштування інфраструктури, але може бути складнішим для автоматизації і відслідковування змін. Імперативний підхід також може збільшувати кількість помилок та покращувати залежність від конкретних спеціалістів. Він може бути реалізований за допомогою інструментів та технологій, таких як Bash, Python та інші.

2.3.3 Недоліки IaC

Infrastructure as Code (IaC) має деякі недоліки, включаючи:

– високі навички: створення і використання IaC вимагає високого рівня навичок у програмуванні та системному адмініструванні, тому може бути складно для новачків;

– конфігурація є постійною: коли ви конфігуруєте систему за допомогою IaC, ви зберігаєте свою конфігурацію в файлах, і це означає, що кожен раз, коли ви хочете зробити зміни, вам потрібно вносити їх в файли;

– проблеми з безпекою: якщо ви не використ овуєте IaC правильно, можуть виникнути проблеми з безпекою, такі як несанкціонований доступ до вашої інфраструктури через небезпечно налаштовані ключі доступу.

2.3.1 Переваги IaC

Раніше забезпечення інфраструктури вважалось трудомістким і доволі дорогим процесом. Використання хмарних обчислень приводить до збільшення

кількості компонентів інфраструктури, з’являється багато додатків, а інфраструктуру часто потрібно масштабувати або вилучати. Пактика використання IaC може допомогти з керуванням масштабу сучасної інфраструктури.

– реплікація інфраструктури: IaC дозволяє легко репліцувати інфраструктуру, що дозволяє легко відновлювати систему в разі виникнення проблем.

– автоматизація: IaC дозволяє автоматизувати процес створення та розгортання інфраструктури, що заощаджує час та зменшує ймовірність помилок.

– контроль версій: IaC дозволяє контролювати версії файлів конфігурації і відслідковувати зміни, що дозволяє легко відновлювати попередні версії в разі потреби.

– зменшення ризиків: IaC дозволяє знизити ризики помилок при ручному наданні і керуванні інфраструктурою.

– зменшення витрат витрат: IaC дозволяє шивдше конфігурувати інфраструктуру і направлений на забезпечення прозорості, щоб допомогти іншим командам з усієї організації працювати швидше та ефективніше.

2.3.2 Amazon Web Services CloudFormation

Amazon Web Services CloudFormation - це сервіс, який дозволяє створювати та керувати інфраструктурою за допомогою декларативного підходу IaC. CloudFormation дозволяє створювати та конфігурувати ресурси AWS за допомогою шаблонів, які містять визначення желаного стану інфраструктури [12].

Шаблони CloudFormation можуть бути створені в різних форматах, включаючи JSON та YAML. Вони містять визначення ресурсів AWS, таких як віртуальні машини, бази даних, служби синхронізації файлів, та інші. Шаблони можуть бути використані для створення нових ресурсів, оновлення існуючих та видалення ресурсів.

Кожен стек має унікальне ім'я та містить визначення бажаного стану ресурсів, які мають бути створені. Коли виконується команда створення стеку, CloudFormation автоматично створює всі ресурси, визначені у шаблоні, та конфігурує їх за визначеними параметрами.

Change Set (набір змін) – це функція, яка дозволяє переглядати зміни, які будуть здійснені перед виконанням оновлення стеку. Це дозволяє переглянути та перевірити зміни в ресурсах стеку перед тим, як їх застосувати.

Коли ви запускаєте Change Set, CloudFormation аналізує шаблон та порівнює його з поточним станом стеку. Потім він відображає зміни, які будуть здійснені, такі як створення нового ресурсу, оновлення існуючого або видалення ресурсу. Це дозволяє керувати змінами в інфраструктурі та забезпечує більшу контрольність та надійність системи.

Також в CloudFormation можна зустріти таке поняття як: Stack Set (набір стеків) – це функція, яка дозволяє створювати і керувати набором стеків, , які співпадають з одним шаблоном. Це дозволяє застосовувати одні і ті ж налаштування до багатьох стеків, які знаходяться в різних областях та аккаунтах.

Переваги використання CloudFormation:

– автоматизація процесу створення і керування інфраструктурою: використовуючи шаблони CloudFormation, ви можете автоматизувати створення та керування ресурсами AWS за допомогою одного кліку;

– повторне використання шаблонів: ви можете використовувати одні і ті ж шаблони для створення однакової інфраструктури в різних областях або аккаунтах;

– зниження ризику: використовуючи шаблони CloudFormation, ви можете задати завантажені конфігурації та ресурси, що дозволяє вам контролювати зміни та зменшувати ризик помилок;

– керування залежностями: CloudFormation автоматично керує залежностями між ресурсами, що дозволяє вам створювати та змінювати інфраструктуру без безпосереднього керування кожним ресурсом окремо;

– керування версіями: CloudFormation дозволяє керувати версіями шаблонів, що дозволяє вам зберігати історію змін та відкати до попередніх версій, якщо необхідно;

– інтеграція з іншими сервісами AWS: CloudFormation може бути інтегрований з іншими сервісами AWS, такими як AWS Elastic Beanstalk, AWS Elastic Container Service та AWS CodeDeploy, що дозволяє вам автоматизувати створення та керування інфраструктурою на різних рівнях.

Висновки до розділу 2

У даному розділі був проведений огляд побудови типової хмарної аналітики. Були визначені самі основні компоненти для аналітики:

– джерела даних;

– логіка обробки даних;

– обчислювальна потужність системи;

– аналітична модель;

– моделі представлення даних;

– зберігання та відображення результатів.

Також були виділені 4 основні рівні архітектури хмарного рішення для аналітики:

– рівень інфраструктури (Infrastructure layer);

– рівень управління даними (Data Management layer);

– рівень аналітики (Analytics layer);

– рівень візуалізації (Visualization Layer).

Було розглянуто сервіси Amazon Web Services. AWS надає величезний набір сервісів для роботи з даними в хмарі: збереження, створення озер даних, аналіз великих об'ємів даних, журнали аналітики та інше. Сервіс побудований для отримання максимальної ефективності, продуктивності, масштабованості і найнижчої вартості.

AWS має такі сховища даних як: S3 та RDS.

Amazon S3 можна використовувати з ціллю:

– створення озер даних;

– збереження критичних даних та резервних копій;

– архівування даних;

Amazon RDS надає велику кількість функцій як бекапи, масштабування, моніторинг та керування базою даних для полегшення адміністрування баз даних. Цей сервіс може виористовуватись як реляційна база даних, що знаходить на серверах хмари.

Також був розглянутий сервіс Amazon Athena який дозволяє здійснювати аналіз даних у форматі сховища даних Amazon S3, використовуючи мову SQL Amazon QuickSight, що надає можливість візуалізації даних, який надає корпоративним користувачам інструменти для зручного перегляду, аналізу та візуалізації даних. Amazon Web Services Glue автоматизувати процеси зіставлення даних та екстракції, трансформації та завантаження (ETL) даних. Amazon Web Services Lake Formation це сервіс, який допомагає з налаштуванням, розгортанням та керуванням централізованою облаштованою корпоративною data lake.

Після огляду переваг та недоліків цих сервісів, було прийнято рішення, що усі ці служби будуть використовуватись у розробці методики обробки даних компанії торгового типу в хмарному середовищі.

Також був розглянутий підхід IaC (Infrastructure as code) з усіма його недоліками та перевагами. Ця практика дозволяє автоматизувати розгортання та управління інфраструктурою за допомогою коду. Після чого був обріний сервіс AWS CloudFormation, який дозволяє створювати та керувати інфраструктурою за допомогою декларативного підходу IaC. CloudFormation дозволяє створювати та конфігурувати ресурси AWS за допомогою шаблонів,. Даний сервіс допоможе спростити розгортання інфраструктури.

3 РОЗРОБКА СИСТЕМИ ХМАРНОГО АНАЛІЗУ

3.1 Процес розгортання інфраструктури

Розроблена система аналізу обробки даних клієнтів компанії торгово типу в хмарному сховищі на основі технологій Business Intelligence складається з:

– локально розгорнута база даних;

– створене віддалене сховище даних;

– віддалене озера даних (data lake);

– проміжний сервер для міграції даних з локальної бази даних до до віддаленого сховища, а також для реплікації даних з віддаленої бази даних до data lake.

Усі визначені копмоненти системи були описані у конфігураційному файлі з розширенням .yaml. Такий підхід дає можливість швидко розширювати або видаляти ресурси на обладнанні провайдера, а це в свою чергу гарантує швидке відновлення та відтворюваність, а також зупинку стягування коштів провайдером у випадку відключення системи.

Файл конфігурації має такі основні розділи:

– Description: розділ з описом призначення описаних ресурсів;

– Metadata: розділ, що описує набір параметрів, які можна буде редагувати під час розгортання інфраструктури;

– Mappings: розділ з описом сполучень між параметром та ресурсами;

– Parameters: розділ який описує конкретні можливі параметри з допустимими для них значеннями;

– Conditions: розділ з описом можливих умов, які залежать від попередньо визначених мета-даних;

– Resourses: розділ з описом розгорнутих ресурсів;

– Outputs: розділ з описом усіх властивостей, що будуть відображатись після успішного розгортання ресурсів.

3.2 Сховища даних

3.2.1 Локальне сховище бази даних

Побудова аналітичної системи для обробки даних клієнтів компанії торгового типу в хмарному сховищі на основі технологій Business Intelligence починається з вибору сховища даних та схеми зберігання цих даних. В даній магістерській роботі сховищем даних виступає локально розгорнута база даних – Microsoft SQL Server. Було створено 4 схеми з таблицями які зберігає наша база даних: Application, Purchasing, Sales, Warehouse.

Перша схема – Application, вона містить в собі таблицю SystemParameters, в якій записані загальні конфігураційні параметри для бази даних.

Наступна схема – Purchasing, що складається з 3 таблиць які відповідають за покупки постачальника:

– PurchaseOrders: таблиця з деталями про замовлення на закупівлю постачальників;

– PurchaseOrderLines: таблиця з інформацією про кожну позицію в замовленні постачальника;

– PurchaseSupplierTransactions: таблиця з інформацією про усі фінансові операції постачальника.

Схема Sales складається з 6 таблиць, що відповідають за замовлення клієнтів:

– Orders: таблиця з інформацією про замовлення клієнтів компанії;

– OrderLines: таблиця з інформацією про кожну позицію в позиції клієнта;

– Invoices: таблиця з детялями рахунку-фактури, що був внесений замовником;

– InvoiceLines: таблиця з інформацією про кожну позицію в рахунку-фактурі замовника;

– CustomerTransactions: таблиця з інформацією про всі транзакції, що були проведені замовником;

– SpecialDeals: таблиця яка містить інформацію про знижки або спеціальні пропозиції на певні позиції товарів.

У Схемі Warehouse можна побачити 4 таблиці:

– VahicleTemperatures: таблиця, що містить інформацію про зміни температури на складі;

– StockItemHoldings: таблиця з інформацією про наявність товарів в складському приміщенні;

– StockItemStockGroups: таблиця, яка містить інформацію про належність товарів до категорій.

– StockItemTransactions: таблиця з інформацією про всі транзакції, що стосуються товарів розміщених на складі

3.2.2 Віддалене сховище

У другому розділі даної магістерської роботи вже згадувались переваги використання хмарних технологій, тож, було прийняте рішення, що створена аналітична система обробки даних клієнтів компанії торгово типу буде використовувати дані, що будуть знаходитись у віддаленому сховищі, утвореному на базі Amazon RDS. В самій системі процес міграції вже є включеним, тому можна перейти до відтворення віддаленого сховища, яке буде аналогічним за змістом і архітектурою до раніше створеного локального сховища.

Було прийнято рішення використовувати Amazon Simple Storage Service як сховище даних. За допомогою цього сервісу було створено 2 основних та окремих каталоги:

– mill-landing-bucket – це сховище, в якому зберігаються дані без попередньої обробки, тобто ті, що тільки потрапили у систему;

– mill-structured-bucked – це сховище з чітко визначеними типами даних, уже обробленими, що можуть у майбутньому використовуватись для побудови аналітики.

В каталозі mill-landing-bucket дані негайно записуються після того, як вони отримані із Amazon RDS. Цей каталог служить сховищем для усіх даних, які надходять до системи в структурованому, неструктурованому та напівструктурованому вигляді

3.3 Процес міграції даних.

3.3.1 Міграція даних до Amazon RDS з локального сховища

Для міграції даних було використано сервіс Amazon Database Migration Service (Amazon DMS). Цей сервіс може бути використаний для міграції баз даних з інших обладнання та сервісів до AWS, а також для міграції даних між різними сервісами AWS, такими як Amazon RDS та Amazon Aurora. DMS також містить функції резервного копіювання та відновлення даних, що дозволяє користувачам зберігати та відновлювати дані в разі необхідності.

Попередні налаштування були здійснені на локальному та віддаленому сховищах, що, в свою чергу, дозволило успішно встановити з'єднання між цими сховищами. У результаті проведених дій були створені такі сутності Amazon DMS:

– Source Endpoint: сутність, яка описує джерело даних, які мігрують. У нашому випадку це локальна база даних. Кожен Source Endpoint містить інформацію про тип джерела, адресу, аутентифікацію та налаштування для підключення до джерела даних;

– Target Endpoint: сутність, яка описує місце призначення для даних, які мігрують. У магістерській роботі ця сутність описує сховище Amazon RDS. Кожен Target Endpoint містить інформацію про тип місця призначення, адресу, аутентифікацію та налаштування для підключення до сховища даних;

– Replication Instance: це сутність, яка контролює сам процес міграції даних. У даному випадку ця сутність використовується як проміжний сервер для міграції даних. Кожний Replication Instance містить параметри конфігурації

міграції, такі як кількість потоків, стратегія міграції та налаштування безпеки. Він також містить інформацію про підключення до source endpoint та target endpoint, що використовуються для міграції даних;

– Database Migration Task: міграції даних між джерелом та міцем призначення. Він включає в себе створення таблиць та індексів, завантаження даних, перевірку даних та моніторинг прогресу міграції.

3.3.2 Процес реплікації даних

Реплікація даних з Amazon RDS до Amazon S3 включає створення завдання міграції у Amazon Database Migration Service (DMS), яке підключається до інстансу Amazon RDS як джерело та до кошика S3 як місце призначення. У результаті були отримані такі сутності:

– Source Endpoint: це підключення до джерела даних, у даному випадку це буде інстанс Amazon RDS;

– Target Endpoint: це підключення до місця призначення даних, у даному випадку це буде кошик mill-landing-bucket S3;

– Replication Instance: це сутність, яка контролює сам процес міграції даних. Вона містить налаштування міграції такі як кількість потоків, стратегія реплікації і налаштування трансформації даних;

– Database Migration Task: сам процес реплікації, що включає в себе налаштування створення таблиць, індексів, завантаження даних, моніторинг прогресу міграції та відстежування статусу завдання.

У результаті реплікації даних в кошику mill-landing-bucket можна побачити підкаталоги, які носять назви раніше описаних схем. А кожен каталог наповнений директоріями, що носять назву таблиць з схем: Application, Purchasing, Sales, Warehouse, які зберігаються в базі даних. Зараз дані зберігаються у форматі .CSV (Comma-separated values), у такому випадку ми можемо використовувати лише методи обробки тектових даних, адже це розширення файлів не підтримує збереження схеми розмітки даних.

3.4 Обробка даних для аналітики

3.4.1 Встановлення типів даних

Наступним кроком для аналітичного хмарного рішення в магістерській роботі є визначення типів даних з якими буде працювати побудована система. Для цього потрібно звернути увагу з якими типами даних працює AWS Glue, а саме:

– Structured Data: це структуровані дані, які можуть бути відформатовані в табличні форми, такі як CSV, JSON, Avro, Parquet та ORC;

– Semi-Structured Data: це напівструктуровані дані, які можуть містити власні теги, такі як XML і JSON;

– Unstructured Data: це неструктуровані дані, які не мають жодної форми чи схеми. Наприклад: текстові файли, зображення, PDF файли, фотографії та аудіо файли.

AWS Glue дозволяє користувачам завантажувати, трансформувати та зберігати різноманітні типи даних у сховища даних Amazon S3, включаючи можливість зв'язування даних різних типів і застосування правил та процедур для їх обробки.

Цей сервіс має інтерактивне середовище для створення етапів екстракції, перетворення та завантаження (ETL) і містить бібліотеку PySpark, яка дозволяє користувачам писати скрипти для обробки даних на Python.

PySpark є бібліотекою, що використовується для розробки додатків для обробки даних за допомогою мови Python і фреймворку Apache Spark. Він містить API для обробки даних в режимі реального часу та завантаження даних з різних форматів, таких як CSV, JSON, Parquet та Avro. Це дозволяє користувачам написати код для обробки даних на Python, а потім виконати цей код на кластері Apache Spark, що дозволяє обробляти великі об'єми даних швидко і ефективно. Приклад такого почергового запиту читання таблиць з каталогу mill- landing-bucket можна побачити на рисунку 3.1, тут є описаний результат читання таблиці Application.SystemParameters.

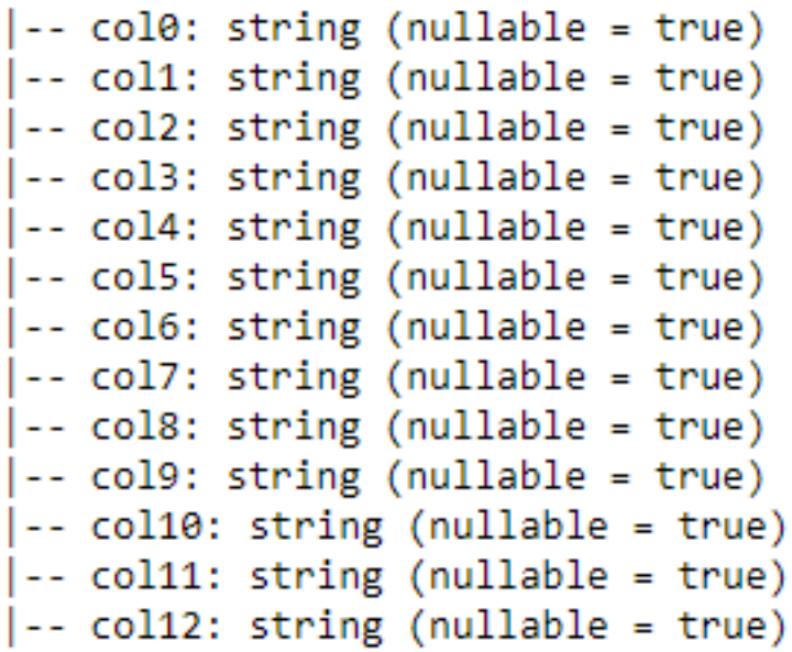


Рисунок 3.1 – результат читання таблиці Application.SystemParams

Після чого відбувається найменування стовпців та визначення типів даних відповідно до нашої схеми. Результат цих дій можна побачити на рисунку 3.2.

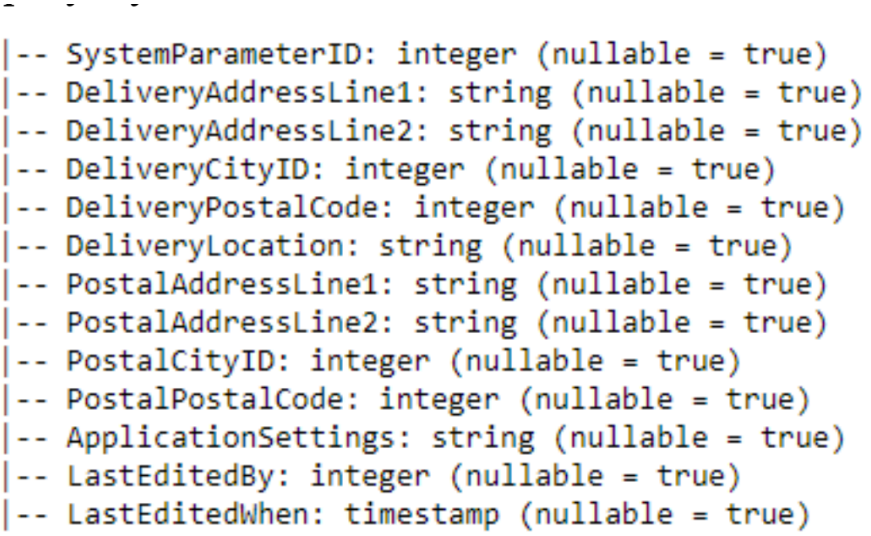


Рисунок 3.2 – Результат визначених типів та названих колонок таблиці Apllication.SystemParams

3.4.2 Нові набори даних для аналітики

Для створення аналітики за показниками було прийнято додати нові набори даних на таблиці Orders з наступною структурою:

– OrdersAddTimeDelivery: набір даних з розширеними даними про час доставки замовлень;

– OrdersTopSalesman: набір даних з інформацію про кількість продаж кожного продався;

– OrdersValuaableCustomer: набір даних з інформацією про самих цінних клієнтів за кількістю замовлень;

– OrdersByWeekdays: набір даних з інформацією про кількість замовлень з навантаженням по дням тижня.

Тепер потрібно визначити спосіб реалізації додавання нових наборів даних.

3.4.2.1 Спосіб 1. Реплікація таблиць до Amazon S3

Перший спосіб полягає в реплікації створених таблиць (за допомогою запитів до бази даних, можна побачити на рисунках 3.3, 3.4, 3.5 та 3.6) з Amazon RDS до Amazon S3.

```
SELECT *,
CONVERT(date, PickingCompleteWhen) as DateCompleted
FROM [WideWorldImporters].[Sales].[Orders]
WHERE ExpectedDeliveryDate < CONVERT(date, PickingCompleteWhen);

SELECT *,
DATEDIFF(dd, OrdeDate, CONVERT(date, PickingCompleteWhen)) as DeliveryTime
From [WideWorldImporters].[Sales].[Orders];
```

Рисунок 3.3 – запит до бази даних за набором даних для отримання інформації про час доставки замовлення

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

Рисунок 3.4 – запит до бази даних за набором даних для отримання інформації про найкращих продавців

```
SELECT CustomerID,
COUNT(*) as CountOrders
From [WideWorldImporters].[Sales].[Orders]
GROUP BY CustomerID
ORDER BY CountOrders DESC;
```

Рисунок 3.5 – запит до бази даних за набором даних для отримання інформації про найцінніших покупців за кількістю покупок

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

Рисунок 3.6 – запит до бази даних за набором даних для отримання інформації про навантаження замовлень за днями тижня

Основним мінусом цього способу є створення додатково навантаження на базу даних, адже нові набори даних будуть використовуватись тільки для аналітичних цілей. Ще одним мінусом буде більша витрата часу на реплікацію до data lake. Тому можна переходити до другого способу.

3.4.2.2 Спосіб 2. Створення нового підкаталогу службою Glue

Як вже було раніше згадано в магістерській роботі за домопогою AWS GLUE можна створити скрипт на Python або Scala, щоб екстрагувати дані з різних джерел, проводити перетворення та завантажувати дані до місця призначення.

Саме тому було прийнято рішення скористатись цим способом, написавши скрипт мовою програмування Python, який допоможе здійснити обробку даних та додати нові набори даних у новий підкаталог mill-structured-bucket.

У результаті проведеної роботи був створений новий підкаталог з назвою OrderAnalytics, в якому, на рисунку 3.7 можна побачити нові утворені набори даних для аналітики: OrdersAddTimeDelivery, OrdersTopSalesman, OrdersValuableCustomer, OrdersByWeekdays.

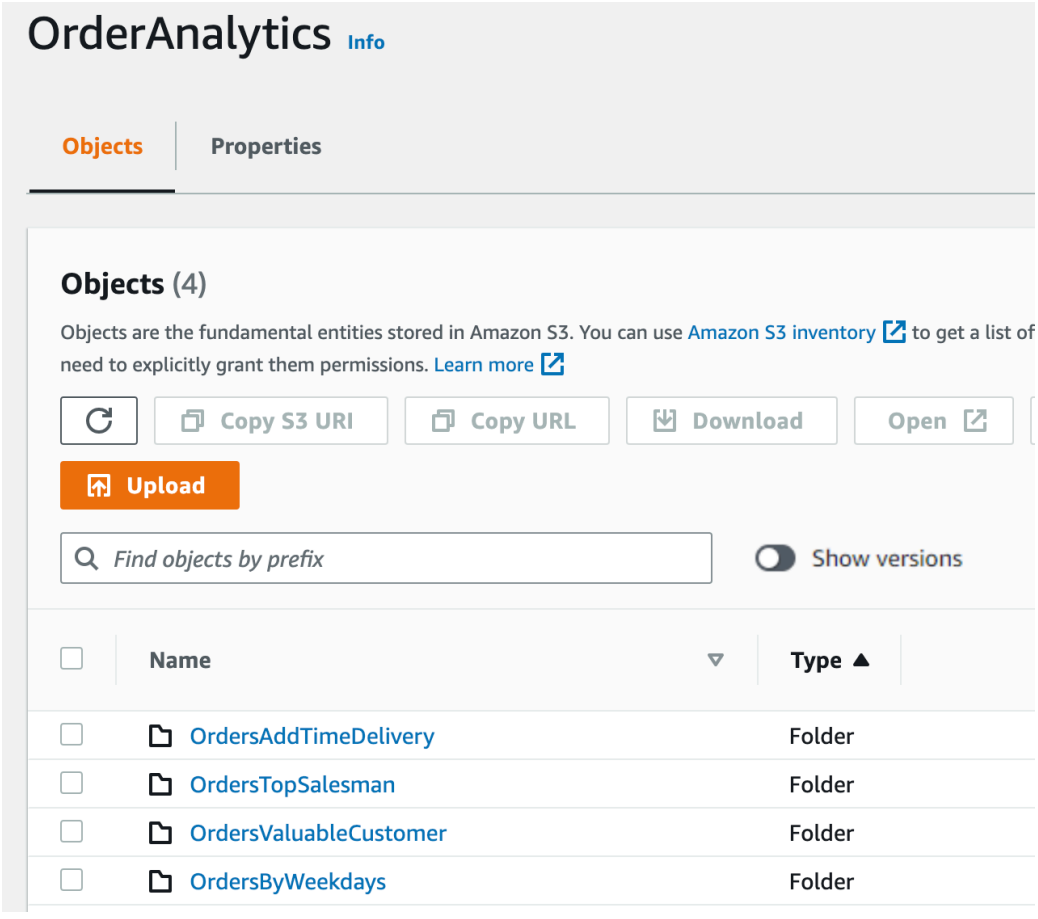


Рисунок 3.7 – демонстрація каталогу OrderAnalytics з новими наборами даних

3.5 Мета-сховища

Мета-сховище в AWS Glue є компонентом, який зберігає метадані про дані, які знаходяться в джерелі даних. Ці метадані можуть включати інформацію про структуру даних, типи даних, назви стовпців і інші атрибути. Вони можуть використовуватися для автоматизації створення каталогу та для створення запитів до даних з програмного коду. Мета-сховище також може бути використане для збереження метаданих про міграцію даних, щоб дозволити користувачам відстежувати переміщення даних між різними системами.

За допомогою служби Glue Crawlers можна автоматизовано створити мета-сховища. Під час виконання магістерської роботи було створено 5 схем в data lake. Відповідно до кожної схеми нам потрібно створити краулер (crawler) [13].

Краулер (crawler) в AWS Glue - це програмний компонент, який сканує джерело даних і зібрання метаданих про них. Краулер може автоматично розпізнавати структуру даних і створювати схему для них. Після створення схеми, краулер може зберігати її в мета-сховищі Glue для подальшого використання. Краулери можуть бути налаштовані для зібрання метаданих з різних типів даних, таких як файли, бази даних, таблиці в облаштуваннях краулера, та можуть додавати ці метадані до існуючого каталогу Glue Data Catalog.

Відповідно до наших 5 схем: application, sales, purchasing, warehouse, analytics, було створено 5 краулерів: applicationCrawler, salesCrawler, puchasingCrawler, analyticCrawler.

У результаті роботи краулерів були отримані усі раніше згадані таблиці у каталозі mill-structured-bucket. Додатково було збережено наші нові таблиці для аналітики.

Після ідентифікації усіх таблиць у каталозі було прийнято рішення перевірити роботу системи спробувавши створити запит до створених даних за допомогою служби Amazon Athena.

Amazon Athena є сервісом аналітики, який дозволяє здійснювати запити SQL до даних записаних в Amazon S3. Він не вимагає налаштування серверів або адміністрування баз даних, дозволяє запитувати дані без затримок і має високу скалярну готовність. Цей сервіс можна використовувати для створення звітів, виконання аналітики і запитів для поновлення даних у реальному часі.

Запит до таблиці та відповідь на запит можна побачити на рисунках 3.8, 3.9

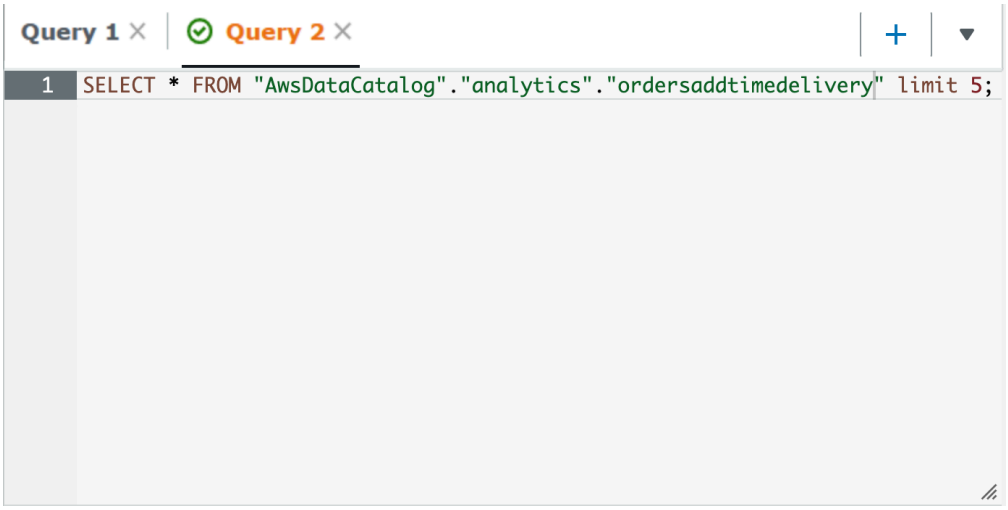
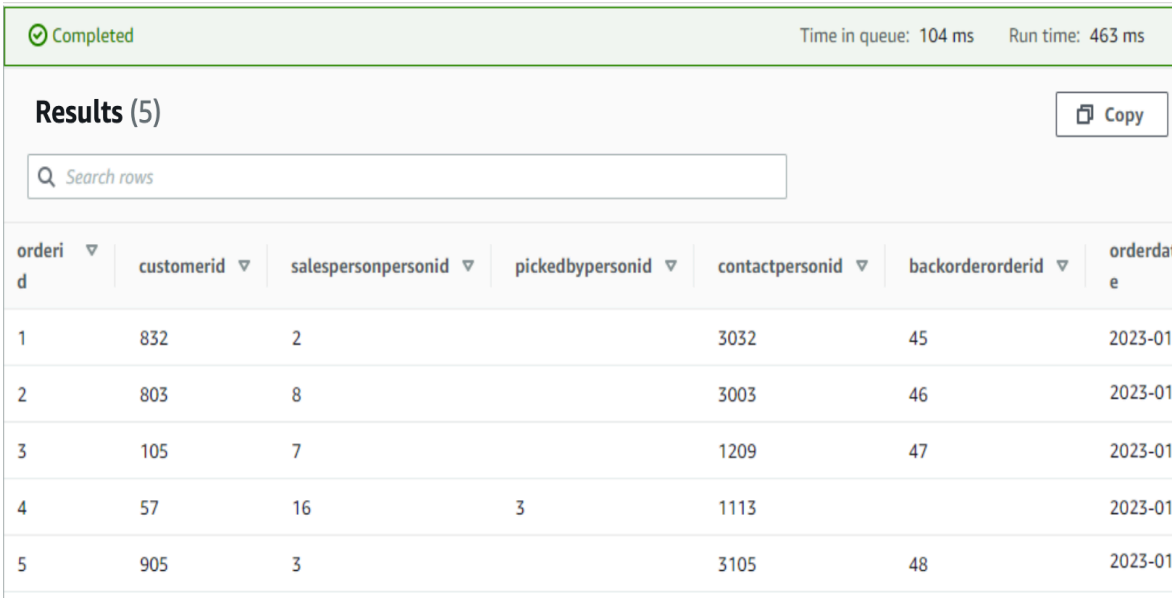


Рисунок 3.8 – запит до OrdersAddTimeDelivery вивести 5 замовлень



orderid	customerid	salespersonpersonid	pickedbypersonid	contactpersonid	backorderorderid	orderdate
1	832	2		3032	45	2023-01
2	803	8		3003	46	2023-01
3	105	7		1209	47	2023-01
4	57	16	3	1113		2023-01
5	905	3		3105	48	2023-01

Рисунок 3.9 – відповідь на запит від Amazon Athena

3.6 Візуалізація даних

У другому розділі магістерської роботи вже згадувався інструмент для аналітики Amazon Quicksight.

Amazon QuickSight є сервісом візуалізації даних, який дозволяє створювати інтерактивні діаграми, графіки та звіти з даних, збережених в різних системах хмарного сховища та базах даних, таких як Amazon S3, RDS, Redshift, і багато інших. QuickSight має інтуїтивний інтерфейс, який дозволяє легко робити запити, змінювати і налаштовувати діаграми і звіти, і дозволяє користувачам спільно працювати з даними в реальному часі.

Було прийнято рішення побудувати аналітичні показники по новоутвореним наборам даних, а саме:

– OrdersAddTimeDelivery: з таблиці можна взяти середній час доставки замовлення (Рисунок 3.10) та відсоткове співвідношення вчасно доставлених товарів з доставками, які запізнились (рисунок 3.11);

– OrdersTopSalesman: топ 5 продавців по кількості виконаних продаж (рисунок 3.12);

– OrdersValuableCustomer: топ 10 замовників по кількості створених замовлень

– OrdersByWeekdays: графік навантаженності доставок по дням тижня (рисунок 3.13).

QuickSight надає можливість створювати різні типи графіків і діаграм, наприклад, лінійні, кругові, стовпчикові, карти та діаграми розсіювання, щоб візуалізувати дані. Також сервіс довзоляє створювати звіти, інтерактивні дашборди і віджети, які можна відображати на сайті або в додатку, і забезпечувати доступ до даних та звітів різним користувачам за допомогою ролей і налаштування доступу. Інструментом можна користуватись не тільки для простого відпбраження даниз а й для сортування та групування даних, а також здайснювати зведення та агрегацію.

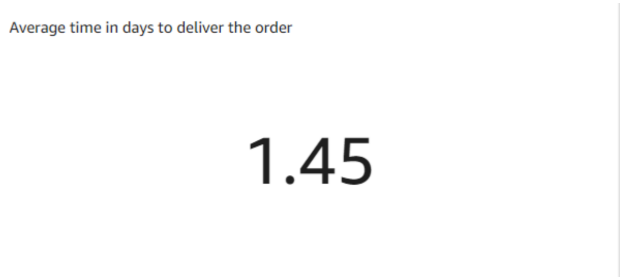


Рисунок 3.10 – середнє значення часу доставки замовлення

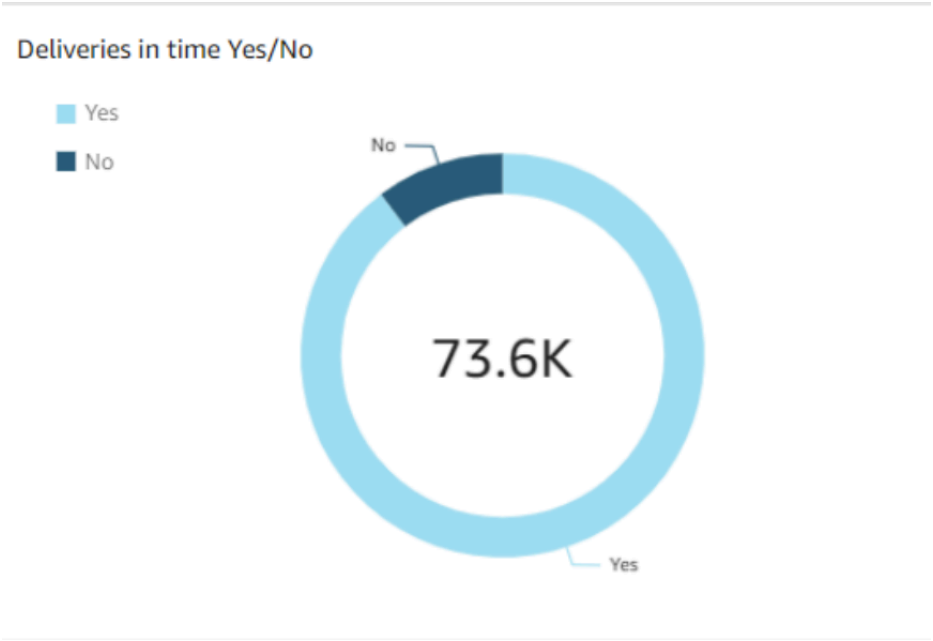


Рисунок 3.11 – Кругова діаграма з співвідношенням вчасно доставлених замовлень до замовлень із запізненням

Top 5 Salesmans

salespersonperso...	count
2	7,474
13	7,400
15	7,371
16	7,532
20	7,387

Рисунок 3.12 – список з топ 5 продавців.

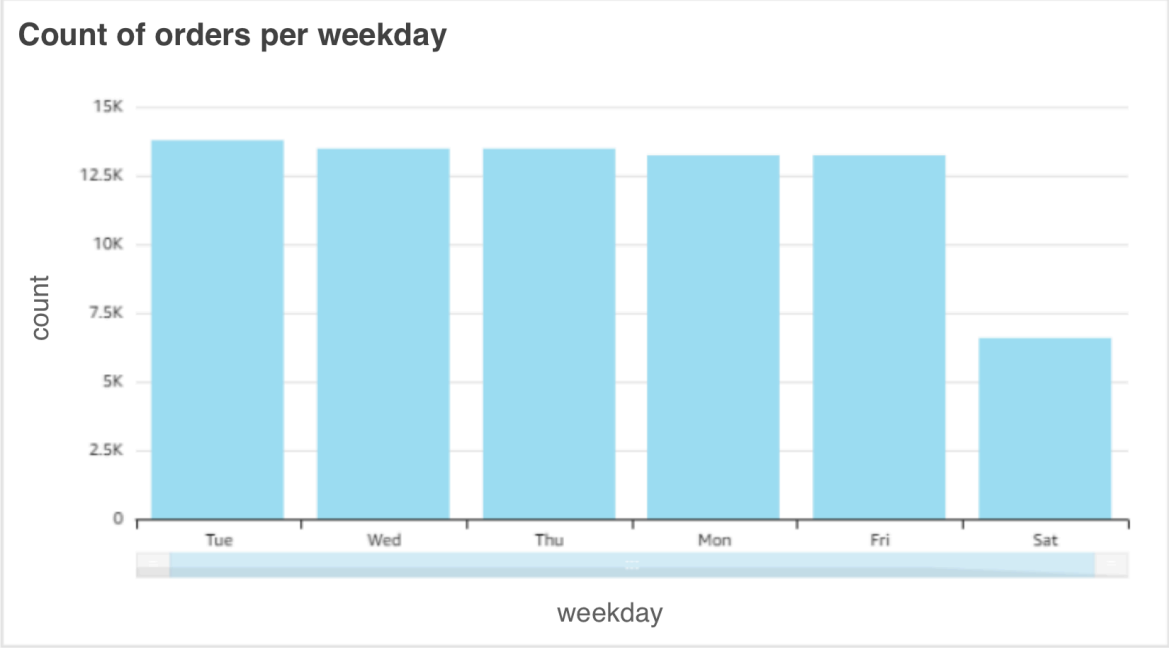


Рисунок 3.13 – діаграма (гістограма) завантаженості днів тижня від кількості замовлень.

3.7 Методика обробки даних та архітектура

На рисунку 3.14 можна побачити архітектуру системи розробленої для магістерської роботи. Система була побудована таким чином:

– було створено локальну базу даних з файлом конфігурації. У локальній базі даних було створено 4 схеми з таблицями;

– створено віддалене сховище Amazon S3;

– за допомогою Amazon RDS було проведено міграцію та реплікацію даних;

– було проведено визначення типів даниз сервісом AWS Glue;

– були додані нові набори даних для аналітики

– були створені мета-сховища за допомогою сервісу AWS Glue

– були прописані запити до даних через сервіс Athena;

– була розроблена візуалізація даних за допомогою сервісу Amazon QuickSight

У результаті отримано аналітичні систему, що дозволяє працювати з новими даними та вже існуючими.

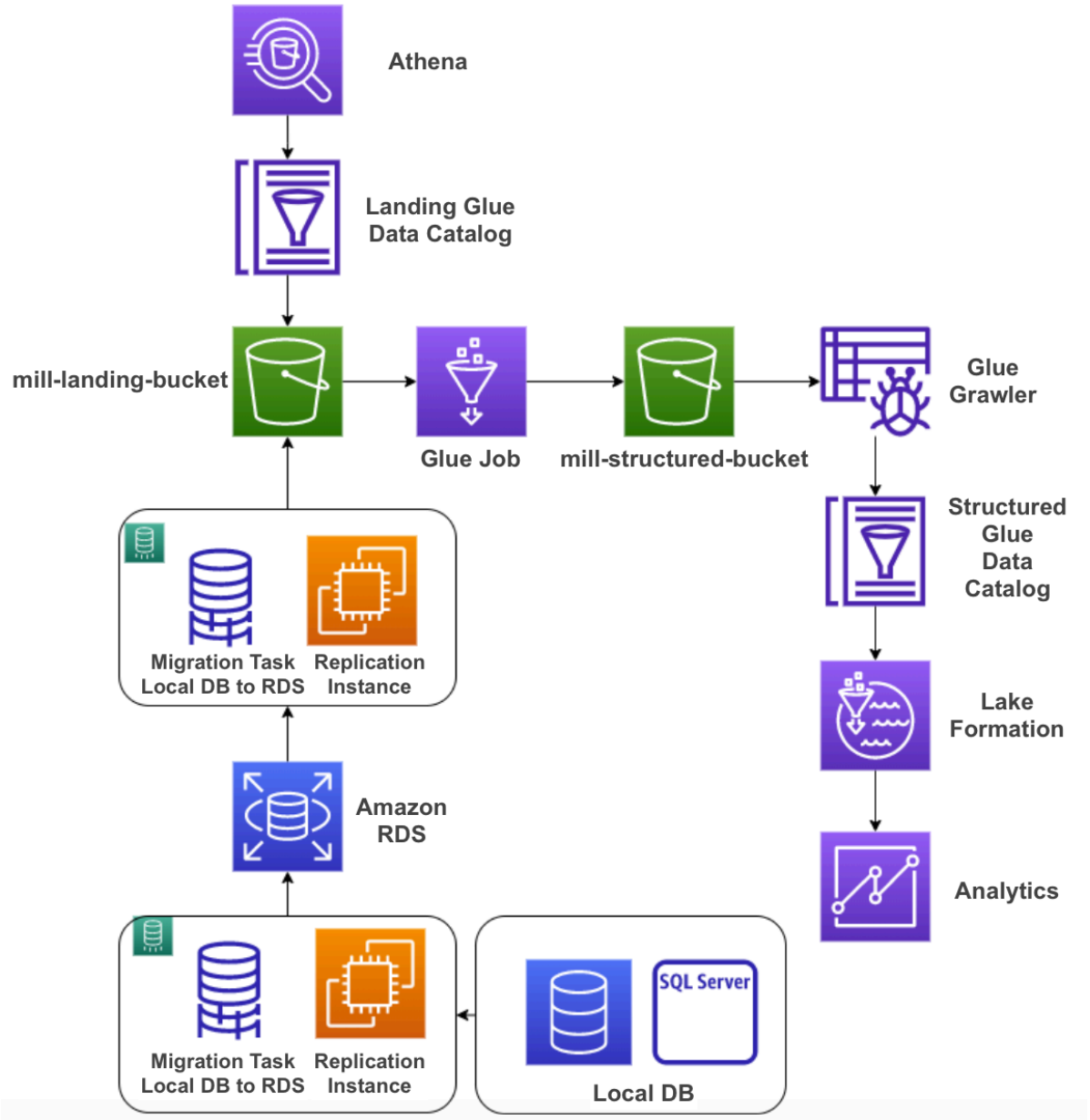


Рисунок 3.14 – архітектура аналітичного інструменту

Висновки до розділу 3

У третьому розділі магістерської роботи було проведене рзгортання інфраструктури системи аналізу обробки даних клієнтів компанії торгово типу в хмарному сховищі на основі технологій Business Intelligence. Система складається з таких компонентів:

– локально розгорнута база даних;

– створене віддалене сховище даних;

– віддалене озеро даних (data lake);

– проміжний сервер для міграції даних з локальної бази даних до до віддаленого схивища, а також для реплікації даних з віддаленої бази даних до data lake.

Усі визначені копмоненти системи були описані у конфігураційному файлі з розширенням .yaml. Такий підхід дає можливість швидко розширювати або видаляти ресурси на обладнанні провайдера, а це в свою чергу гарантує швидке відновлення та відтворюваність, а також зупинку стягування коштів провайдером у випадку відключення системи.

Було побудоване локальне сховище даних - локальна база даних Microsoft SQL Server. Локальна БД містить в собі 4 схеми з таблицям: Application, Purchasing, Sales, Warehouse.

Так як задачею є побудова хмарної аналітики то було створене віддалене сховище даних. Було прийнято рішення про використання Amazon S3. За допомогою цього сервісу було свторено 2 каталоги:

– mill-landing-bucket: сховище з даними без попередньої обробки;

– mill-structured-bucked: сховище з чітко визначеними типами даних.

Після чого був проведений процес міграції даних з локального сховища на Amazon RDS. Для міграції даних було використано сервіс Amazon Database Migration Service (Amazon DMS). Цей сервіс може бути використаний для міграції баз даних з інших обладнання та сервісів до AWS, а також для міграції даних між різними сервісами AWS, такими як Amazon RDS та Amazon Aurora.

Процес реплікації бази даних був теж проведений через AWS DMS.У результаті реплікації даних в кошику mill-landing-bucket можна побачити підкаталоги, які носять назви раніше описаних схем. А кожен каталог наповнений директоріями, що носять назву таблиць з схем: Application, Purchasing, Sales, Warehouse, які зберігаються в базі даних.

За допомогою сервісу AWS Glue для аналітичного хмарного рішення були визначені типи даних з якими буде працювати побудована система. Для

аналітики були введені нові набори даних, був створений новий підкаталог службою Glue.

Також були утворені мета-сховища до 5 створених схем. У результаті роботи краулерів були отримані усі раніше згадані таблиці у каталозі mill-structured-bucket. Додатково було збережено наші нові таблиці для аналітики.

Візуалізація даних була утворена за допомогою сервісу Amazon Quicksight.

Як результат було отримано аналітичне рішення в хмарному середовищі з технологіями Business Intelligence.

4 АНАЛІЗ РОЗРОБЛЕНОГО РІШЕННЯ

4.1 Перевірка коректності передачі даних

Для того, щоб впевнитись в правильній роботі розробленого аналітичного інструменту для обробки даних компанії торгового типу в хмарному сховищі потрібно провести валідацію даних. Для цього достатньо порівняти зміст кожної з таблиць з локальної бази даних з результатами запитів до mill-structured-bucke каталогу через мета-сховища.

Тому було прийнято рішення зробити такі запити почергово для кожної з утворених раніше схем.

4.1.1 Схема Application

Для валідації даних можна створити запит до таблиці.SystemParameters на локальній базі даних (рисунок 4.1) та мета-сховища цієї таблиці (рисунок 4.2). Якщо система працює коректно, то маємо отримати однаковий результат з 2 запитів.

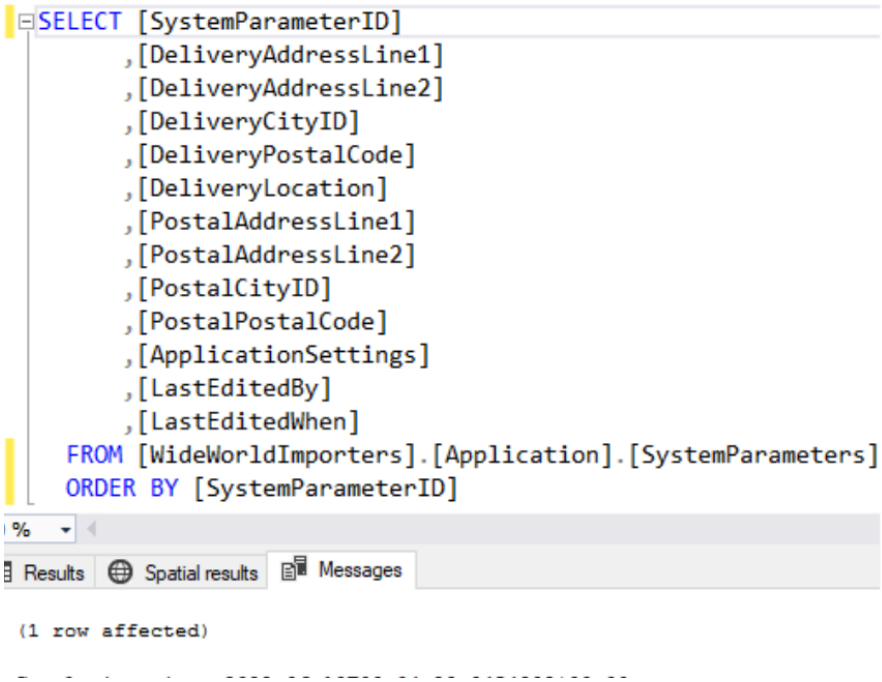


Рисунок 4.1 – запит з результатом до локальної бази даних

Рисунок 4.2 – запит до мета-сховища з результатами

4.1.2 Схема Sales

Для валідації даних можна створити запит до таблиці. SpecialDeals на локальній базі даних (рисунок 4.3) та мета-сховища цієї таблиці (рисунок 4.4). Якщо система працює коректно, то маємо отримати однаковий результат з 2 запитів.

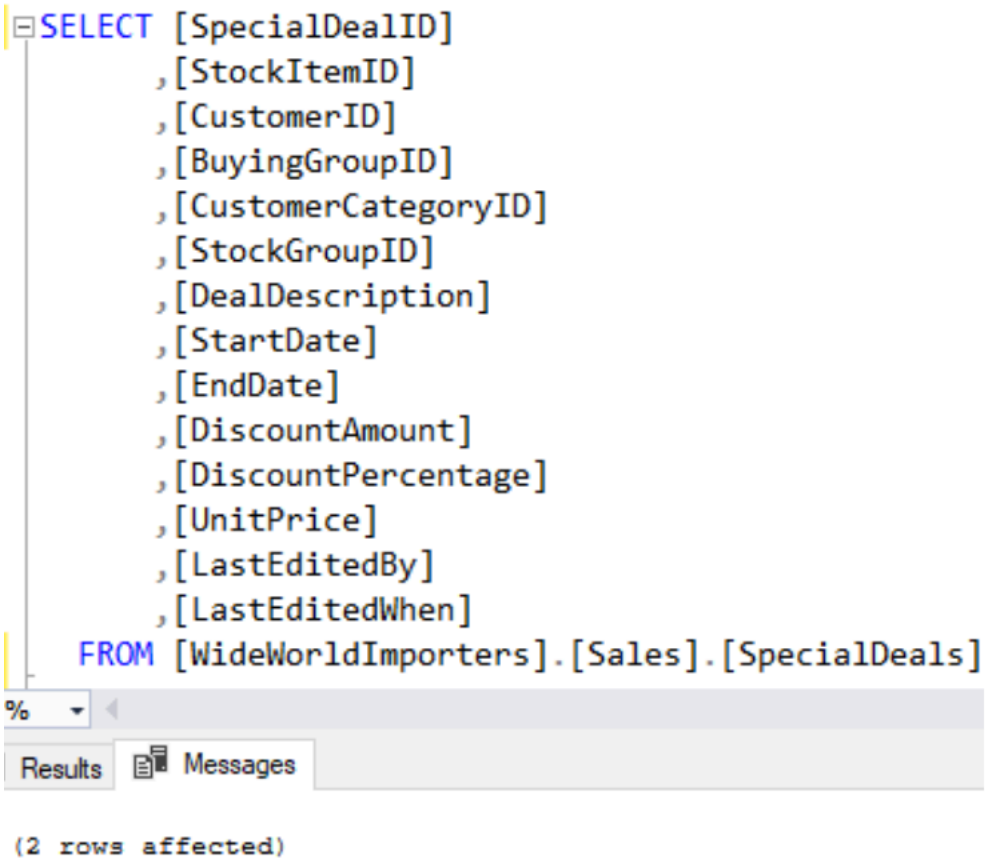


Рисунок 4.3 – запит з результатом до локальної бази даних

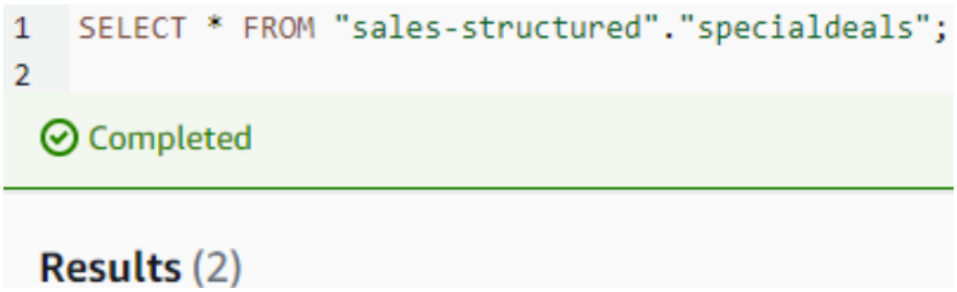


Рисунок 4.4 – запит до мета-сховища з результатами

4.1.3 Схема Warehouse

Для валідації даних можна створити запит до таблиці. StockItemStockGroups на локальній базі даних (рисунок 4.5) та мета-сховища цієї таблиці (рисунок 4.6). Якщо система працює коректно, то маємо отримати однаковий результат з 2 запитів.

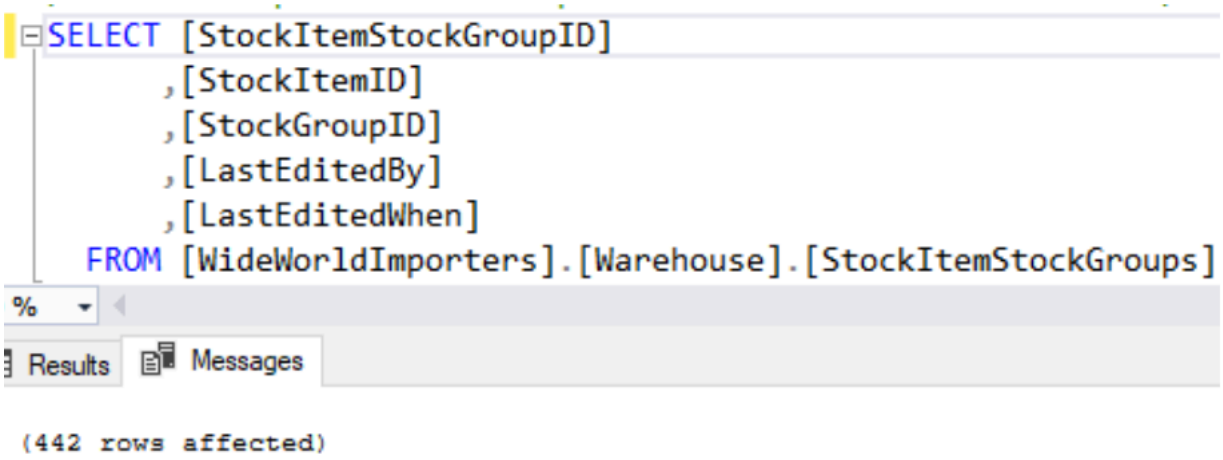


Рисунок 4.5 – запит з результатом до локальної бази даних

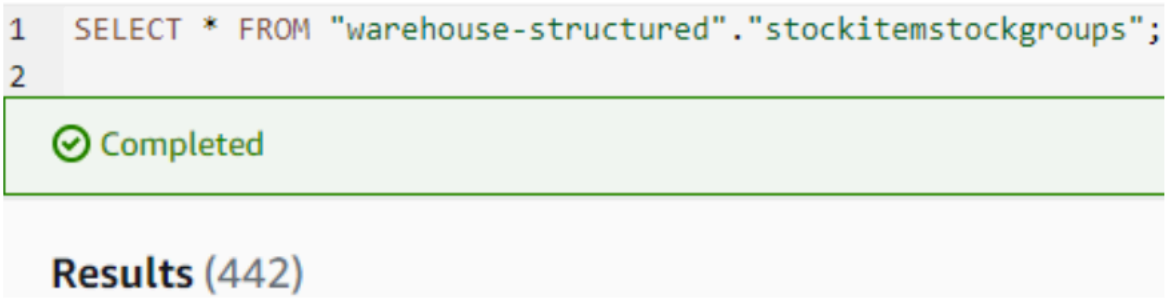


Рисунок 4.6 – запит до мета-сховища з результатами

4.1.4 Схема Purchasing

Для валідації даних можна створити запит до таблиці. SupplierTransactions на локальній базі даних (рисунок 4.7) та мета-сховища цієї таблиці (рисунок 4.8). Якщо система працює коректно, то маємо отримати однаковий результат з 2 запитів.

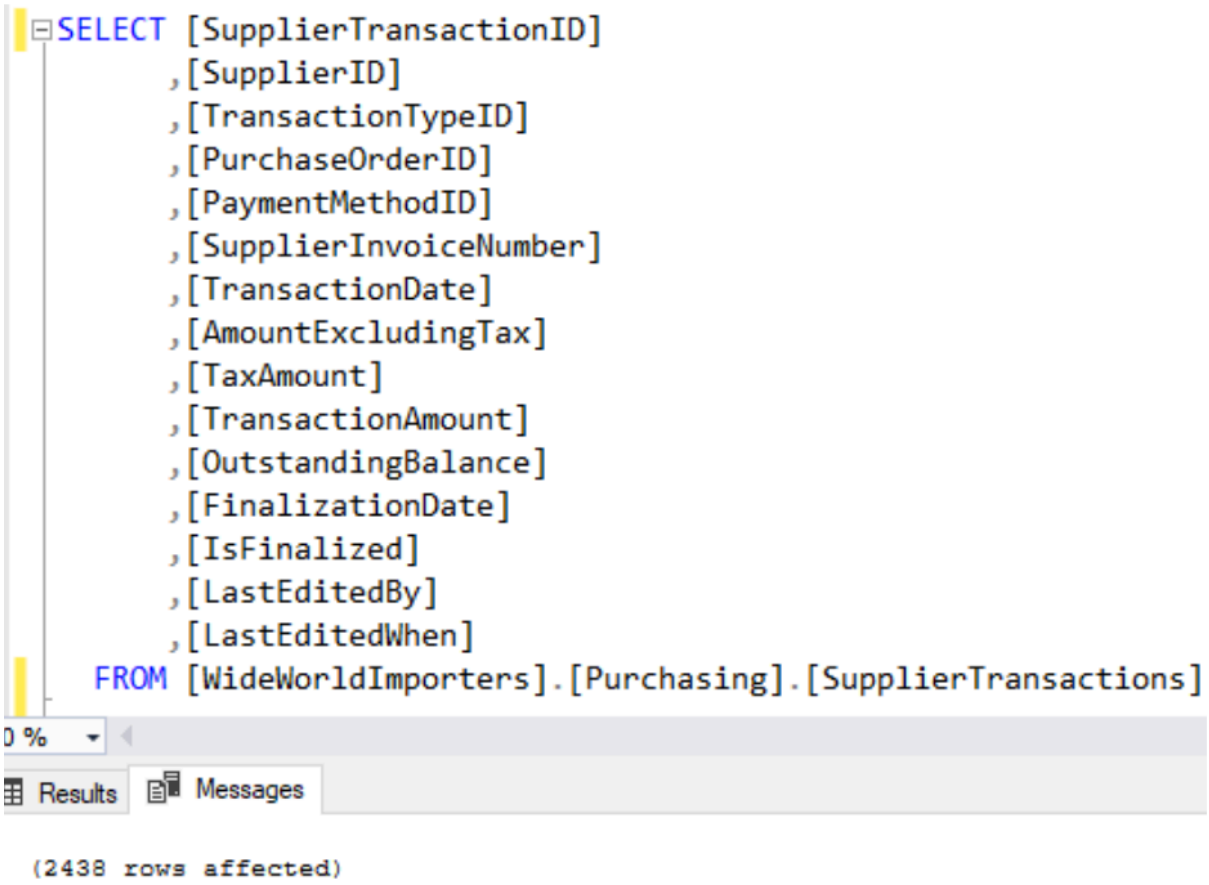


Рисунок 4.7 – запит з результатом до локальної бази даних

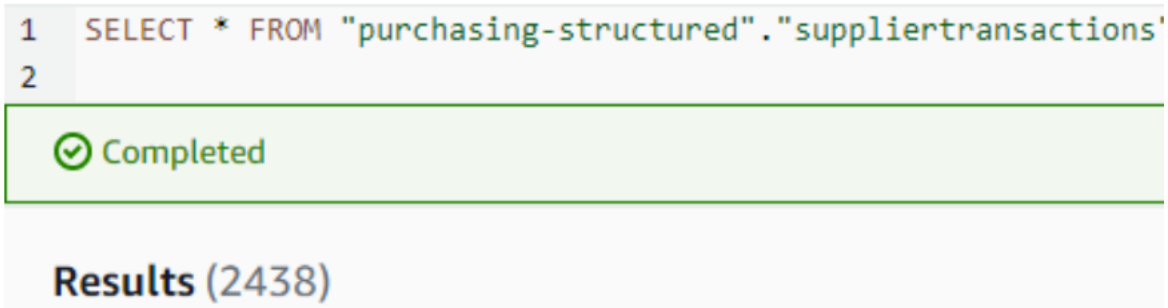


Рисунок 4.8 – запит до мета-сховища з результатами

4.2 Результати перевірки даних

Порівнявши результати запитів до локальної бази даних та мета-сховищ можемо побачити однакові значення. Це означає, що розроблене аналітичне рішення працює коректно та не допускає втрати даних при міграції або реплікації.

4.3 Недоліки розробленого аналітичного рішення

4.3.1 Налаштування взаємодії між службами

Фізична інфраструктура для системи розгортається згідно конфігураційного файлу, що робить цей процес контрольованим та автоматизованим. Це звільняє від необхідності в додаткових зусиллях, таких як побудови додаткової документації для підтримки системи.

Але взаємодія між службами відбувається інакше, вона побудована через інтерфейс Amazon Console. Недоліки налаштування роботи служб через графічний інтерфейс Amazon Console можуть включати в себе необхідність написання інструкцій та збільшення обсягу документації для підтримки системи та її відтворюваності, складність керування доступом до ресурсів та відслідковування змін, що були внесені до системи. Також, відсутність автоматизації та контролю процесу налаштування може призвести до помилок та складностей під час оновлення системи.

Цей недолік можна виправити за допомогою інструменту AWS Cloud Development Kit (AWS CDK).

AWS CDK – це засіб для розробки та налаштування інфраструктури через код. Він дозволяє розробникам використовувати різні мови програмування, такі як JavaScript, TypeScript, C#, Python та Java, для створення та керування інфраструктурою на AWS. Використовуючи CDK, розробники можуть автоматизувати процес налаштування та деплою інфраструктури, використовуючи стандартні практики розробки та керування кодом. Це

дозволяє розробникам заощаджувати час та зменшувати витрати на налаштування інфраструктури [14].

Тобто, AWS CDK є інструментом для створення автоматизованих інфраструктурних рішень у середовищі AWS. Він дозволяє створювати конфігурації інфраструктури в вигляді коду, який може бути контрольованою версіями та опублікований в контролері версій.

Використання такого підходу є доцільним для усунення виявленого недоліку.

4.3.2 Довга перерва в оновленні даних

Розроблена система обробляє дані різ на дому (у визначений час). Усі нові дані одразу з'являються у сховищах після їх завантаження.

Єдиним мінусом є оновлення та обробка даних за розкладом - раз на добу у визначений час. Іноді, оновлення даних раз на добу може призвести до неактуальності даних для деяких аналітичних звітів чи до потреби в додатковій обробці даних для досягнення бажаної рівні свіжості даних.

Цей недолік можна спробувати вирішити за допомогою сервісу AWS Lambda. AWS Lambda є сервісом хмарного обчислення, який дозволяє запускати код без необхідності налаштовувати та запускати сервери. Ви можете завантажувати свій код в Lambda і створювати функції, які виконуються, коли вони викликаються, або коли події відбуваються в інших сервісах AWS, таких як S3, DynamoDB, SNS і т.д. Це дозволяє розробникам фокусуватися на своєму коді, а не на налаштуванні і керуванні інфраструктурою, завдяки чому вони можуть швидше розробляти і виконувати свої додатки [15].

Тобто сервіс Lambda може спрацьовувати по тригеру – завантаженню файлів на Amazon S3 і запускати скрипт по оброці та оновленню даних.

Висновки до розділу 4

У четвертому розділі магістерської роботи були розглянуті недоліки розробленої обробки даних клієнтів компанії торгового типу в хмарному сховищі на основі технологій Business Intelligence. Також було проведено перевірку коректності передачі даних.

Порівнявши результати запитів до локальної бази даних та мета-сховищ таблиць що знаходяться в основних схемах можна зробити висновок, що система працює правильно.

Також було виявлено 2 основних недоліки в розробленому аналітичному рішенні, а саме: довга перерва в оновленні даних та важке налаштування взаємодії між службами. Були запропоновані рішення для допрацювання та усунення недоліків у майбутньому.

Налаштувати автоматизовану спрощену взаємодію інфраструктури може допомогти інструмент AWS Cloud Development Kit. Він дозволяє створювати конфігурації інфраструктури в вигляді коду, та працювати з контролерами версій.

За допомогою сервісу Lambda можна вирішити проблему оновлення даних по розкладу. Дані будуть автоматично оновлюватись при появі нових даних у сховищі.

ВИСНОВКИ

Під час роботи над магістерським проєктом було проведено аналіз існуючих методик обробки даних клієнтів компанії торгового типу. За допомогою дослідження ринку та порівняння з іншими компаніями, було виявлено, що вдосконалення методики обробки даних клієнтів в хмарному сховищі з використанням технологій Business Intelligence може значно покращити ефективність роботи компанії.

У першому розділі магістерської роботи були досліджені варіанти побудови аналітичних рішень за допомогою хмарної інфраструктури.

Були визначені основні переваги використання хмарної інфраструктури:

– ефективне управління даними;

– захист від втрати даних;

– оптимізація витрат;

– підвищена гнучкість;

– безпека; мобільність;

– прозорість;

– автоматичне оновлення системи.

Усі ці характеристики показали актуальність систем на сьогодні, особливо у парі з використанням технологій Business intelligence, які є надзвичайно корисними для бізнесу.

Також були розглянуті типи хмар за їх доступністю: публічні, приватні та гібридні.

Другий розділ магістерського проєкту включає в собі дослідження побудови типової хмарної аналітики з використанням технологій Business intelligence. У наслідок чого були визначені основні компоненти для аналітики: джерела даних, логіка обробки даних, обчислювальна потужність системи, аналітична модель, моделі представлення даних, зберігання та відображення результатів.

Під час роботи над проєктом було виявлено 4 основні рівні архітектури хмарного рішення для аналітики:

– рівень інфраструктури (Infrastructure layer);

– рівень управління даними (Data Management layer);

– рівень аналітики (Analytics layer);

– рівень візуалізації (Visualization Layer).

Були розглянуті служби Amazon Web Services, що надають величезний набір сервісів для роботи з даними в хмарі, а саме:

– Amazon S3;

– Amazon RDS;

– Amazon Athena;

– Amazon QuickSight;

– AWS Glue;

– AWS Lake Formation;

– AWS CloudFormation.

Після оцінки всіх переваг цих сервісів було прийнято рішення використовувати їх надалі в магістерській роботі. Також був розглянутий підхід IaC (Infrastructure as code), ця практика дозволяє автоматизувати розгортання та управління інфраструктурою за допомогою коду. Після чого був обраний сервіс AWS CloudFormation, який дозволяє створювати та керувати інфраструктурою за допомогою декларативного підходу IaC. CloudFormation дозволяє створювати та конфігурувати ресурси AWS за допомогою шаблонів.

У третьому розділі була реалізована методика обробки даних компанії торгового типу в хмарному сховищі з використанням технологій Business intelligence за допомогою розробленого аналітичного інструменту.

Процес розробки системи обробки даних почався з розгортання інфраструктури, усі компоненти системи були описані у конфігураційному файлі з розширенням .yaml. Такий підхід дає можливість швидко розширювати або видаляти ресурси на обладнанні провайдера, а це в свою чергу гарантує

швидке відновлення та відтворюваність, а також зупинку стягування коштів провайдером у випадку відключення системи.

Для початку було розгорнуте лкальне сховище бази даних, що виступає локальною базою даних Microsoft SQL Server.

Далі було створене віддалене сховище даних Amazon Simple Storage Service.

Міграція та реплікація даних відпувалась за допомогою сервісу Amazon RDS. Після визначення типів даних були утворені нові набори даних, які вже безпосередньо будуть використовуватись для аналітики.

Після налаштування мета-сховищ були налаштовані аналітичні показники за допомогою Amazon Quicksight, що використовує технології Busines Intelligence.

У результаті було побудовано аналітичну систему, що дозволяє працювати як з даними, що були щойно додані, так і з обробленими.

Також були виявлені основні недоліки системи: довга перерва в оновленні даних та важке налаштування взаємодії між службами. Для цих недоліків були прописані рекомендації на майбутнє допрацювання.

Використання такої методики, дозволяє спростити керування даними, здійснювати їх аналіз, а також додатково підвищує відмовостійкість та забезпечує легку масштабованість.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Khajeh-Hosseini, A., & Buyya, R. (2011). Cloud Computing: Concepts, Technology & Architecture. Prentice Hall.
2. Investopedia. [Електронний ресурс] – Режим доступу: https://www.investopedia.com/terms/b/business-intelligence-bi.asp (дата звернення 15.11.2022) – Назва з екрану
3. McKenna Consultants. [Електронний ресурс] – Режим доступу: https://www.mckennaconsultants.com/relationship-between-iot-big-data-and-the-cloud/ (дата звернення 17.11.2022) – Назва з екрану
4. AWS. [Електронний ресурс] – Режим доступу: https://aws.amazon.com/big-data/datalakes-and-analytics (дата звернення 20.11.2022) – Назва з екрану
5. Amazon S3. [Електронний ресурс] – Режим доступу: https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html (дата звернення 20.11.2022) – Назва з екрану
6. Amazon RDS. [Електронний ресурс] – Режим доступу: https://aws.amazon.com/rds/ (дата звернення 21.11.2022) – Назва з екрану
7. Amazon Athena. [Електронний ресурс] – Режим доступу: https://docs.aws.amazon.com/athena/latest/ug/what-is.html (дата звернення 21.11.2022) – Назва з екрану
8. Amazon QuickSight. [Електронний ресурс] – Режим доступу: https://docs.aws.amazon.com/quicksight/latest/developerguide/welcome.html (дата звернення 22.11.2022) – Назва з екрану
9. AWS Glue. [Електронний ресурс] – Режим доступу: https://docs.aws.amazon.com/glue/latest/dg/what-is-glue.html (дата звернення 25.11.2022) – Назва з екрану
10. AWS Lake Formation. [Електронний ресурс] – Режим доступу: https://docs.aws.amazon.com/lake-formation/?id=docs_gateway (дата звернення 25.11.2022) – Назва з екрану

11. Microsoft. What is infrastructure as code? [Електронний ресурс] – Режим доступу:https://learn.microsoft.com/en-us/devops/deliver/what-is-infrastructure-as-code (дата звернення 25.11.2022) – Назва з екрану
12. AWS CloudFormation. [Електронний ресурс] – Режим доступу: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/Welcome.html (дата звернення 30.11.2022) – Назва з екрану
13. AWS. How crawlers work. [Електронний ресурс] – Режим доступу: https://docs.aws.amazon.com/glue/latest/dg/crawler-running.html (дата звернення 30.11.2022)
14. AWS. What is the AWS SDK? [Електронний ресурс] – Режим доступу: https://docs.aws.amazon.com/cdk/v2/guide/home.html (дата звернення 02.12.2022)
15. AWS. What is AWS Lambda? ? [Електронний ресурс] – Режим доступу: https://docs.aws.amazon.com/lambda/latest/dg/welcome.html (дата звернення 3.11.2022)

Додаток А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інженерії програмного забезпечення

МАГІСТЕРСЬКА РОБОТА

«Розробка методики обробки даних клієнтів компанії торгового типу в хмарному сховищі на основі технологій Business Inteligence »

Виконав: студент групи ПДМ-61 Бабак Максим Олексійович

Керівник: доцент кафедри ІПЗ Садовенко Володимир Сергійович

Київ - 2022

АКТУАЛЬНІСТЬ ТА ПРАКТИЧНА ЦІННІСТЬ РОБОТИ

Актуальність теми дипломної роботи полягає в тому, що ефективність швидкого аналізу ключових показників бізнесу допомагає компаніям формувати статистику або аналітику в потрібний час. Усі ці показники можуть швидко сповістити про якісь негаразди або покращення чи тенденції в бізнесі. Ще зручніше коли інструмент для такого аналізу та зберігання даних є в безпечному місці, яким можна скористатись в будь-який час

Практична цінність дипломної роботи полягає в тому, що запропоновані рішення допоможуть врахувати особливості систему аналізу даних в хмарному середовищі

МЕТА, ОБ'ЄКТ, ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: спрощення процесу отримання та аналізу ключових показників та метрик кілєнтів для компаній торгового типу

Об'єкт дослідження: аналітика даних компанії торгового типу

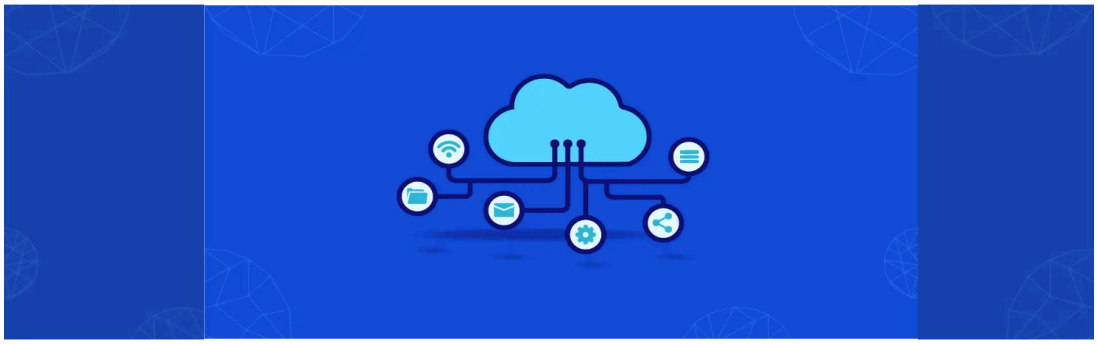
Предмет дослідження: обробка, зберігання та аналіз даних в хмарному середовищі

ЗАДАЧІ ДОСЛІДЖЕННЯ

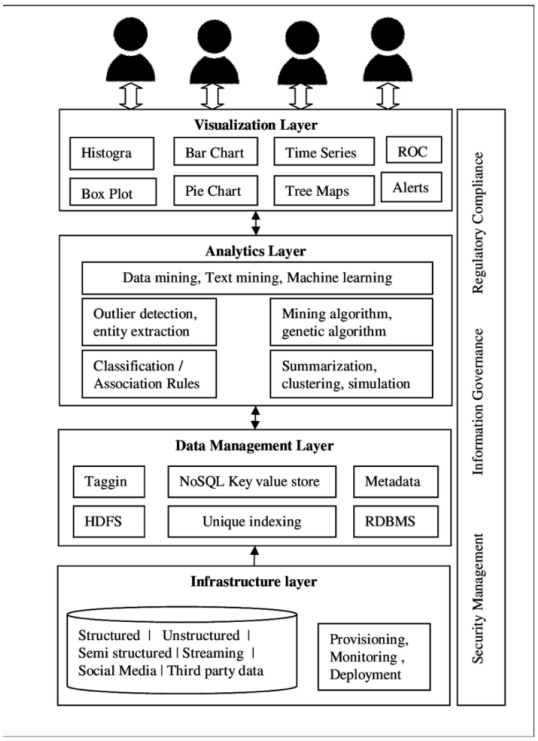
1. Дослідження області аналізу даних та відповідних хмарних рішень для побудови повноцінних аналітичних систем
2. Обґрунтування вибору технологій, підходів та служб.
3. Побудова архітектури аналітичної системи з можливістю зберігання та аналізу даних в хмарному середовищі
4. Оптимізація роботи аналітичної системи для роботи з компанією торгового типу

ПЕРЕВАГИ ВИКОРИСТАННЯ ХМАРНИХ ТЕХНОЛОГІЙ

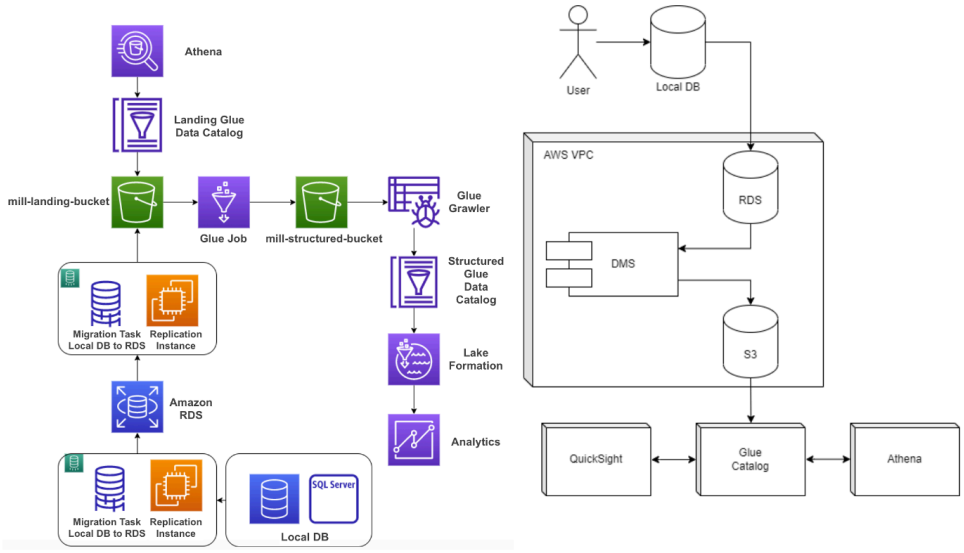
- Зберігання даних у безпечному місці;
- Вартість та оптимізація;
- Гнучкість;
- Підвищення рівня безпеки;
- Мобільність;
- Прозорість;
- Автоматичне оновлення;



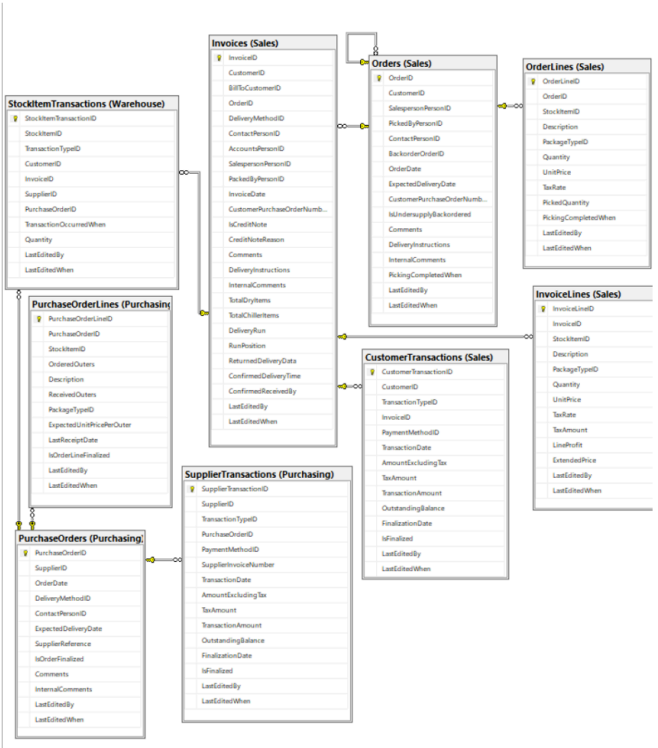
ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ СИСТЕМИ



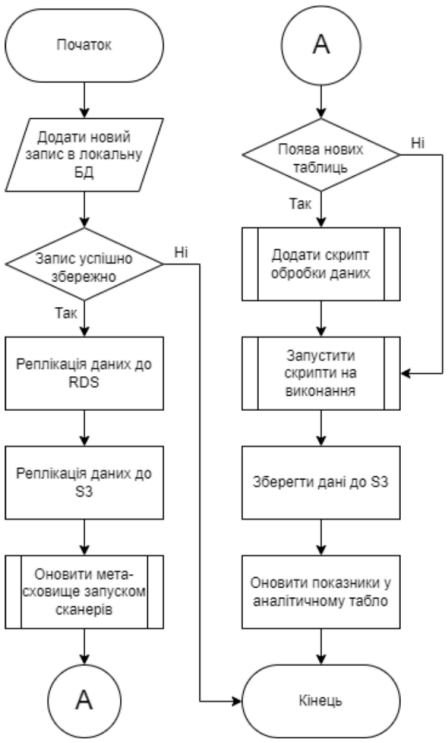
АРХІТЕКТУРА СИСТЕМИ АНАЛІЗУ ДАНИХ В ХМАРІ



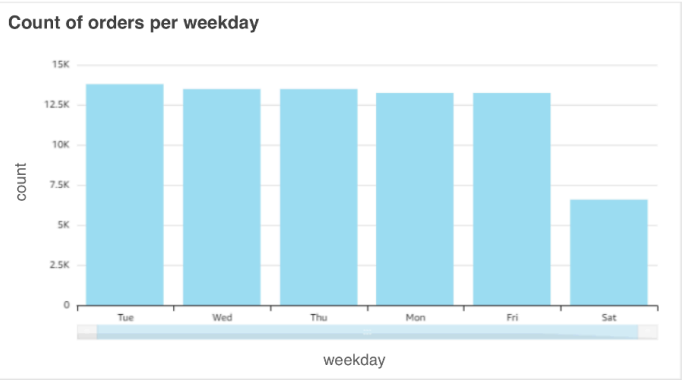
ФУНКЦІОНАЛЬНА СХЕМА (діаграма даних)

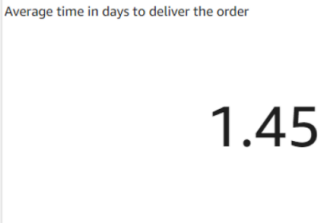


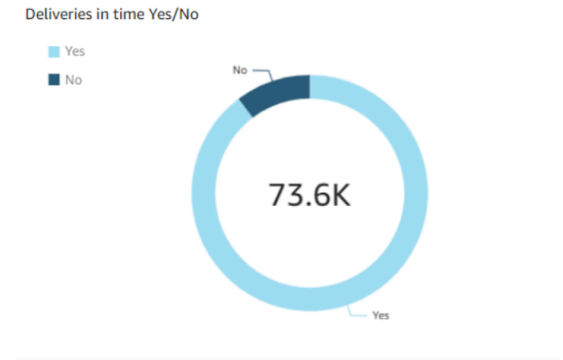
ПРИНЦИПОВА СХЕМА (алгоритм роботи програми)



ПРАКТИЧНИЙ РЕЗУЛЬТАТ



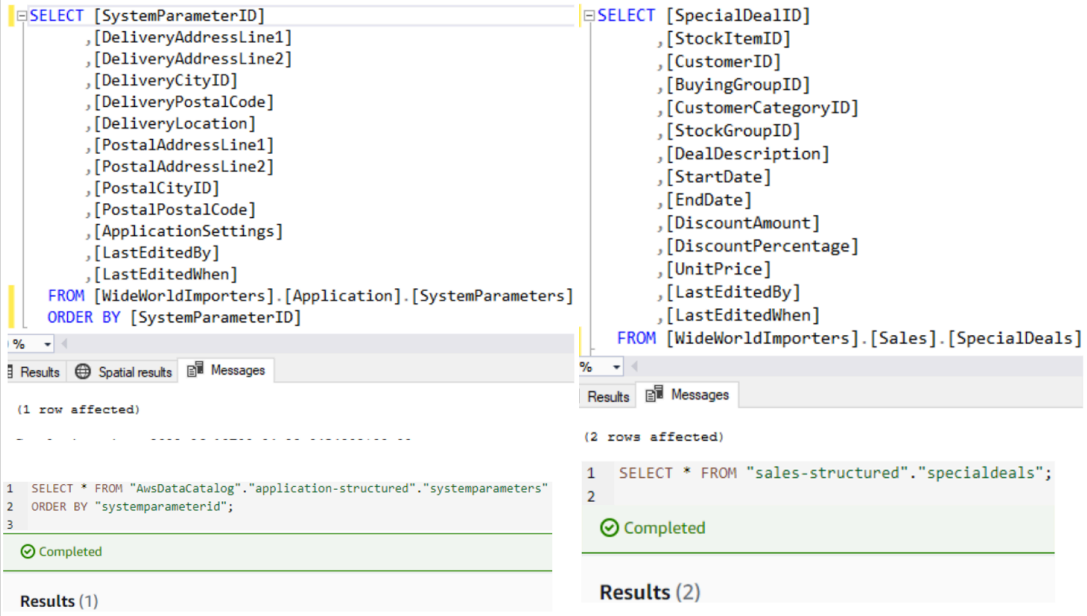




Top 5 Salesmans

salespersonperso...	count
2	7,474
13	7,400
15	7,371
16	7,532
20	7,387

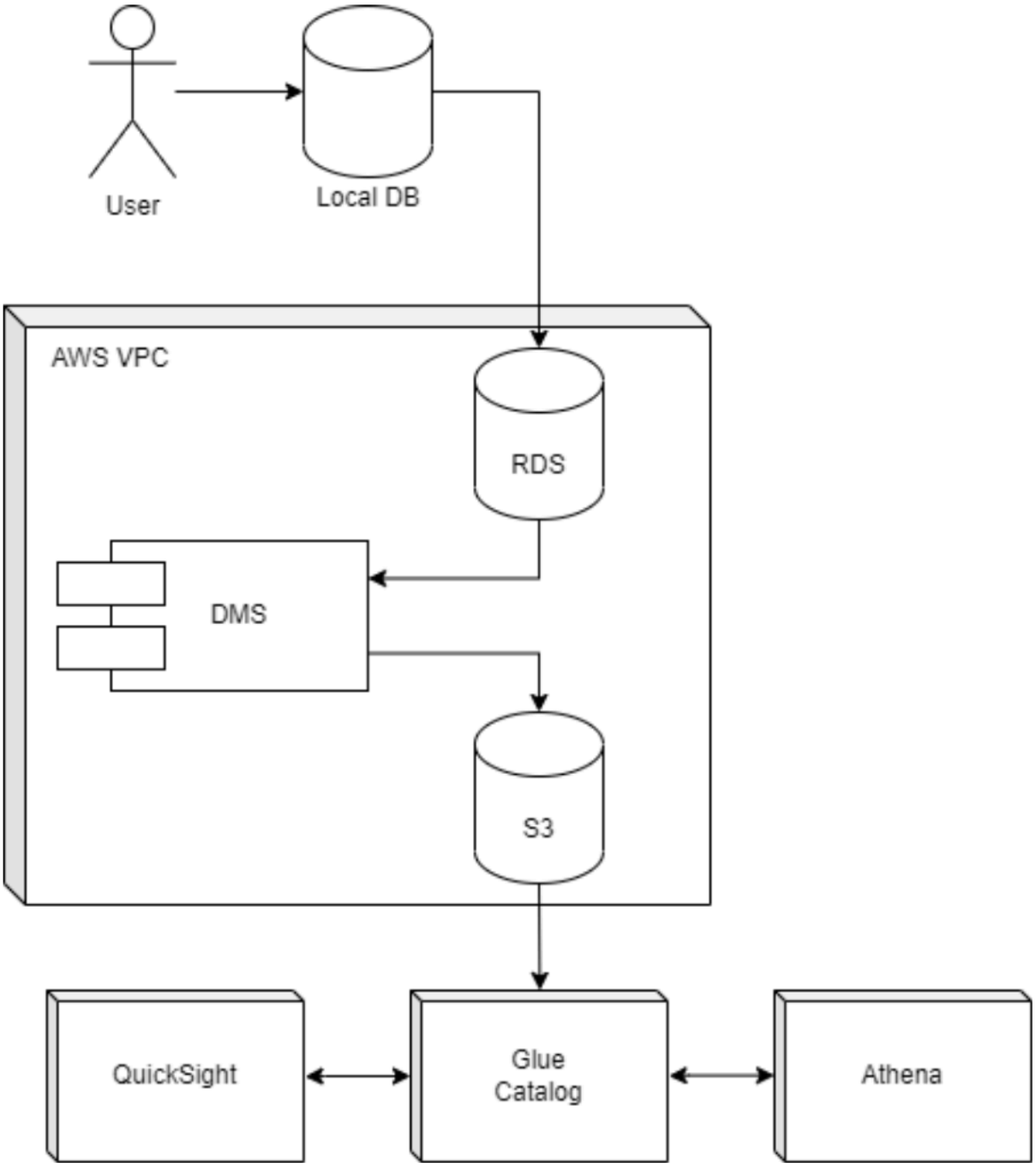
ВАЛІДАЦІЯ ДАНИХ



ВИСНОВКИ

- Під час роботи над дипломним проєктом було проведено дослідження області аналізу даних та відповідних хмарних рішень для побудови повноцінних аналітичних системи. Розглянуто основні підходи побудови аналітичних рішень з використанням хмарних технологій.
- Було визначено схему зберігання даних, розподіл даних по каталогам з урахуванням їх особливостей під час обробки, що дозволило контролювати окремі етапи підготовки даних перед використанням їх аналітичним інструментом
- За допомогою обраних технологій, підходів та служб було спроєктовано архітектуру аналітичної системи з можливістю зберігання та аналізу даних. Архітектура була реалізована засобами Amazon Web Services. Після розгортання інфраструктури та налаштування взаємодії між службами була проведена перевірка коректної роботи аналітичної системи системи за допомогою валідації вхідних та вихідних даних.

Додаток Б. СТРУКТУРНА СХЕМА СИСТЕМИ



Додаток В. ФУНКЦІОНАЛЬНА СХЕМА (діаграма даних)

Додаток Г. ПРИЦНИПОВА СХЕМА (алгоритм роботи програми)