Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мережних технологій факультету інформатики

Style Definition: Heading 1

Formatted: Ukrainian

ТЕОРЕТИЧНІ Й ПРИКЛАДНІ АСПЕКТИ АНАЛІЗУ ДАНИХ ВЕЛИКОЇ РОЗМІРНОСТІ

Текстова частина

магістерської роботи

за спеціальністю „Інженерія Програмного Забезпечення” 121

Керівник магістерської роботи
д.т.н., проф. М. М. Глибовець
_____
“____” _____ 2021 р.

Виконав студент О. О. Шаповал
“____” _____ 2021 р.

Київ 2021

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ
Зав.кафедри інформатики, к.ф.-м.н.
_____С. С. Гороховський
„____”_____2021 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на магістерську роботу

Студенту 2 р.н. магістерської програми Комп’ютерні науки
Шаповалу Олександру Олександровичу
Розробити Проаналізувати теоретичні та практичні аспекти роботи з даними великої розмірності, забезпечення якості даних та розробити архітектуру програмного застосунку для оцінки якості даних в системі для обробки та зберігання маркетингових матеріалів

Зміст текстової частини до магістерської роботи:

Зміст
Анотація
Вступ
1 Огляд теоретичних та практичних аспектів даних великої розмірності
2 Аналіз поняття якості даних та сучасних механізмів їх забезпечення
3 Розробка архітектури програмного застосунку для оцінки якості даних в системі для обробки та зберігання маркетингових матеріалів
4 Реалізація програмного застосунку на базі хмарних технологій
Висновки
Список літератури
Додатки

Дата видачі „___” _____ 2021 р.

Керівник
М.М. Глибовець, доктор технічних наук, доцент _____

Завдання отримав
О.О. Шаповал _____

Тема: Теоретичні й прикладні аспекти аналізу даних великої розмірності

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на дипломну роботу	01.11.2020	
2.	Огляд технічної літератури за темою роботи	15.11.2020	
3.	Виконання аналізу сучасних рішень	29.11.2020	
4.	Розробка архітектури питально-відповідальної підсистеми	27.12.2020	
5.	Реалізація програмного застосунку для оцінки якості даних	17.01.2021	
6.	Налаштування програмного застосунку для прикладу обробки та зберігання маркетингових матеріалів	27.03.2021	
7.	Написання пояснювальної записки	24.04.2021	
8.	Створення слайдів для доповіді та написання доповіді	27.04.2021	
9.	Аналіз отриманих результатів з керівником, написання доповіді та попередній захист магістерської роботи	30.04.2021	
10.	Корегування роботи за результатами попереднього захисту	5.05.2021	
11.	Остаточне оформлення пояснювальної записки та слайдів	04.06.2021	
12.	Захист магістерської роботи (проекту)	17.06.2021	

Студент _____

Керівник _____

“____”_____

ЗМІСТ

Field Code Changed
Deleted: 5
Formatted: Do not check spelling or grammar
Formatted: Do not check spelling or grammar
Deleted: 6
Formatted: Do not check spelling or grammar
Deleted: 9
Formatted: Do not check spelling or grammar
Deleted: 9
Formatted: Do not check spelling or grammar
Deleted: 11
Formatted: Do not check spelling or grammar
Deleted: 12
Formatted: Do not check spelling or grammar
Deleted: 13
Formatted: Do not check spelling or grammar
Deleted: 15
Formatted: Do not check spelling or grammar
Deleted: 17
Formatted: Do not check spelling or grammar
Deleted: 22
Formatted: Do not check spelling or grammar
Deleted: 26
Formatted: Do not check spelling or grammar
Deleted: 27
Formatted: Do not check spelling or grammar
Deleted: 28
Formatted: Do not check spelling or grammar
Deleted: 31
Formatted: Do not check spelling or grammar
Deleted: 31
Formatted: Do not check spelling or grammar
Deleted: 33
Formatted: Do not check spelling or grammar
Deleted: 33
Formatted: Do not check spelling or grammar
Deleted: 36
Formatted: Do not check spelling or grammar
Deleted: 38
Formatted: Do not check spelling or grammar
Deleted: 40
Formatted: Do not check spelling or grammar
Deleted: 41

Анотація

В рамках даної роботи проведено огляд теоретичних та практичних аспектів даних великої розмірності, проаналізовано поняття якості даних та приклади сучасних механізмів їх забезпечення розроблена архітектура програмного застосунку для оцінки якості даних в системі для обробки та зберігання маркетингових матеріалів на базі хмарних технологій

Ключові слова: Big Data, Data Quality, AWS, Deequ

Formatted: Ukrainian

ВСТУП

Актуальність. На сьогоднішній день, інформація, без перебільшення, є найціннішим ресурсом суспільства у глобальному сенсі. При цьому, загальний обсяг даних, зібраних людством, росте з кожним роком у геометричній прогресії. Так, у 2010 році загальний обсяг складав 2 цетабайти. У 2021 році загальний обсяг даних, що створюються, фіксуються, копіюються та споживаються у світі досяг 74 цетабайти та продовжує рости [1]. Швидкий розвиток цифровізації сприяє постійно зростаючій глобальній сфері даних, а системи та методи збору на аналізу даних сприяють створенню нових поколінь систем для ефективного та своєчасного прийняття рішень. Для прикладу, аналіз великих даних допомагає організаціям використовувати свої дані та використовувати їх для виявлення нових можливостей. Це, в свою чергу, призводить до розумніших кроків у бізнесі, ефективніших операцій, вищих прибутків та щасливіших клієнтів. У своєму звіті, Big Data in Big Companies [2], директор з досліджень IIA Том Девенпорт взяв інтерв'ю у понад 50 підприємств, щоб зрозуміти, як вони використовували великі дані. Він виявив, що вони отримали цінність такими способами:

- **Зниження витрат**. Такі технології великих даних, як Hadoop та хмарна аналітика, приносять значні переваги у витратах, коли йдеться про зберігання великих обсягів даних. Крім того, вони можуть визначити більш ефективні способи ведення бізнесу.
- **Швидше, якісніше приймати рішення**. Завдяки швидкості та аналітиці, у поєднанні з можливістю аналізу нових джерел даних, компанії можуть миттєво аналізувати інформацію та приймати рішення на основі вивченого.
- **Нові товари та послуги**. Завдяки здатності оцінювати потреби та задоволеність споживачів за допомогою аналітики, ми отримуємо можливість надати клієнтам те, що вони хочуть.

Проте великі дані є корисними лише якщо вони якісні. Погані дані в кращому випадку не мають значення. У найгіршому випадку це може змусити компанії робити дорогі помилки. За підрахунками IBM, погані дані коштують економіці

США 3,1 трлн доларів на рік [3]. Ці витрати походять від часу, коли працівники повинні витратити на виправлення помилкових даних та помилок, які спричиняють помилки у клієнтів.

Саме тому важливими та актуальним є дослідження теоретичних й прикладних аспектів аналізу даних великої розмірності з точки зору оцінки та забезпечення якості на основі принципів та практик інженерії програмного забезпечення з наступною імплементацією досягнень у вигляді прикладних систем контролю якості даних.

Мета дослідження. Проаналізувати теоретичні та практичні аспекти роботи з даними великої розмірності, забезпечення якості даних та розробити архітектуру програмного застосунку для оцінки якості даних.

Deleted: в системі для обробки та зберігання маркетингових матеріалів

Завдання дослідження. Проаналізувати існуючі підходи до розробки архітектури системи зберігання та оброки даних великої розмірності. Вивчити та дослідити програмні продукти, що можуть бути використані як компоненти в архітектурі сучасної системи зберігання та оброки даних великої розмірності. Перевірити можливості інтеграції досліджених компонентів, порівняти компоненти з близькими чи подібними характеристиками, обрати компоненти, що найкраще підходять для вирішення потрібних задач. Обрати підходи та інструменти для практичної реалізації розробленої архітектури.

Об´єкт дослідження. Інтелектуальні системи для обробки та зберігання маркетингових матеріалів, спеціалізовані хмарні рішення для розробки систем обробки потокових мультимедіа файлів, програмні продукти для вирішення задач зберігання та оцінки якості великих об’ємів даних.

Предмет дослідження. Компоненти системи, їх можливості до інтеграції, API, розробка застосунку з інтеграцією та організацією взаємодії досліджуваних компонентів.

Джерела дослідження. Електронні версії друкованої літератури, програмна документація, довідники посилань на API, електронні ресурси, в тому числі спеціалізовані форуми та віртуальні конференції, вихідні коди програм та бібліотек, відео-інструкції.

Наукова новизна одержаних результатів дослідження полягає в створенні сучасної архітектури підсистеми контролю якості великих даних, на прикладі маркетингових матеріалів, впровадження нових підходів у вирішенні проблематики якості вхідних даних.

Практичне значення одержаних результатів. За рахунок використання прогресивних практик при розробці архітектури підсистеми контролю якості, а саме для прикладу маркетингових матеріалів, зменшуються витрати ресурсів на розробку програмного комплексу, що створений на базі розробленої архітектури. Оптимізація відбувається шляхом використання надійних компонентів, що легко конфігуруються, адаптуються та спроектовані з можливістю подальшого незалежного розширення.

Deleted:

РОЗДІЛ 1. Огляд даних великої розмірності та якості даних

1.1. Великі дані

За словами центру досліджень Гартнера, великі дані - це великі обсяги інформації, що забезпечують швидкість та різноманітність, та вимагають економічно вигідних, інноваційних форм обробки інформації для кращого розуміння та прийняття рішень ".[4]

Це визначення чітко відповідає на питання "Що таке великі дані?" - Великі дані стосуються складних та великих наборів даних, які необхідно обробити та проаналізувати, щоб розкрити цінну інформацію, яка може бути корисною для підприємств та організацій.

- Однак існують певні основні положення великих даних, які спростять відповідь, що таке великі дані:
- Це стосується величезної кількості даних, яка з часом зростає в геометричній прогресії.
- Вони настільки об’ємний, що їх неможливо обробити або проаналізувати за допомогою звичайних методів обробки даних.
- Вони включає видобуток даних, зберігання даних, аналіз даних, обмін даними та візуалізацію даних.
- Цей термін є всеосяжним, що включає дані, структуру даних, а також інструменти та методи, що використовуються для обробки та аналізу даних.

Компанії використовують великі дані в своїх системах для поліпшення операцій, надання кращого обслуговування клієнтів, створення персоналізованих маркетингових кампаній та вжиття інших дій, які, зрештою, можуть збільшити їх прибуток. Підприємства, які ефективно використовують його, мають потенційну конкурентну перевагу перед тими, хто цього не робить, оскільки вони здатні приймати швидші та більш обґрунтовані ділові рішення.

Ось ще кілька прикладів того, як великі дані використовуються організаціями:

Commented [IIW1]: Тут має бути 1. Посилання розставляються по мірі зустрічі в тексті

Commented [AS2R1]: У мене є 3 посилання у вступі. Чи можливий такий варіант, чи вступ повинен бути без цитування?

Formatted: Ukrainian

Deleted: [7

- В енергетичній галузі великі дані допомагають нафтогазовим компаніям визначати потенційні місця буріння та контролювати експлуатацію трубопроводів; так само комунальні служби використовують їх для відстеження електричних мереж.
- Фірми фінансових послуг використовують системи великих даних для управління ризиками та аналізу ринкових даних у реальному часі.
- Виробники та транспортні компанії покладаються на великі дані для управління своїми ланцюгами поставок та оптимізації шляхів доставки.
- Інші урядові заходи включають реагування на надзвичайні ситуації, запобігання злочинності та ініціативи розумних міст.
- Великі дані можуть бути певних типів:
- Структуровані. Під структурованими даними розуміються дані, які можна обробляти, зберігати та отримувати у фіксованому форматі. Вони відноситься до високоорганізованої інформації, яку можна легко та без проблем зберегти та отримати до неї доступ із бази даних за допомогою простих алгоритмів пошукової системи.
- Неструктуровані. Неструктуровані дані відносяться до даних, яким бракує якоїсь конкретної форми чи структури. Це дуже ускладнює обробку та аналіз неструктурованих даних.
- Напівструктуровані. Напівструктуровані дані це ті, що містять обидва формати, згадані вище, тобто структуровані та неструктуровані дані. Якщо бути точнішим, це стосується даних, які хоч і не були класифіковані в певному сховищі (базі даних), але містять важливу інформацію або теги, які відокремлюють окремі елементи в даних.

1.2. Характеристика великих даних

Ще в 2001 році аналітик Gartner Дуг Лейні перелічив 3 «V» великих даних – обсяг (Volume), швидкість (Velocity) та різноманітність (Variety). З часом було додано ще дві характеристики – достовірність (Veracity) і цінність (Value) [2].

Commented [IIW3]: посилання
Commented [AS4R3]: Додав
Formatted: Ukrainian

- **Обсяг** (Volume). Обсяг даних має значення. При великих даних мова йде про обробку великих обсягів неструктурованих даних низької щільності. Це можуть бути дані невідомої цінності, такі як канали даних Twitter, потоки кліків на веб-сторінці або мобільному додатку, або обладнання з підтримкою датчика. Раніше зберігати їх було б проблемою, але нові технології (наприклад, Hadoop) полегшили тягар. Сама назва "Великі дані" пов'язана з величезним розміром. Розмір даних відіграє дуже важливу роль у визначенні цінності даних. Також те, чи можна конкретні дані насправді розглядати як великі дані чи ні, залежить від обсягу даних.
- **Швидкість** (Velocity). Швидкість по суті відноситься до швидкості, з якою дані створюються в режимі реального часу. Наскільки швидко дані генеруються та обробляються для задоволення запитів, визначається реальний потенціал даних. Дані повинні бути доступними в потрібний час для прийняття відповідних бізнес-рішень.
- **Різноманітність** (Variety). Різноманітність відноситься до багатьох доступних типів даних. Традиційні типи даних були структуровані та акуратно вписані в реляційну базу даних. Зі збільшенням великих даних дані надходять у нові неструктуровані типи даних. Неструктуровані та напівструктуровані типи даних, такі як текст, аудіо та відео, вимагають додаткової попередньої обробки для отримання значення та підтримки метаданих. Ця різноманітність неструктурованих даних створює певні проблеми щодо зберігання, видобутку та аналізу даних.
- **Достовірність** (Veracity). Достовірність - це ступінь точності наборів даних та наскільки вони надійні. Сирі дані, зібрані з різних джерел, можуть спричинити проблеми з якістю даних, які важко визначити. Якщо їх не

виправити за допомогою процесів очищення даних, неправильні дані призводять до помилок аналізу, які можуть підірвати цінність ініціатив бізнес-аналітики. Командам з управління даними та аналітики також потрібно забезпечити наявність достатньо точних даних для отримання достовірних результатів.

- **Цінність** (Value). Тільки тому, що ми зібрали багато даних, це не має ніякої цінності, якщо ми не отримаємо з них певної інформації. Це стосується здатності трансформувати цунамі даних у бізнес. Значення означає, наскільки корисними є дані при прийнятті рішень. Насправді це кількість цінних, надійних та надійних даних, які потрібно зберігати, обробляти, аналізувати, щоб знайти ідеї.

1.3. Виклики, що стоять перед великим даним

Проблеми зростання даних

Однією з найбільш актуальних проблем великих даних є правильне зберігання всіх цих величезних наборів даних. Обсяг даних, що зберігаються в центрах обробки даних та базах даних компаній, швидко зростає. Оскільки ці набори даних з часом зростають в геометричній прогресії, вкрай важко впоратися з ними.

Більшість даних є неструктурованими та надходять із документів, відео, аудіо, текстових файлів та інших джерел. Це означає, що їх не можна легко знайти в базах даних.

Інтеграція даних з різних джерел

Дані в організації надходять із різних джерел, таких як сторінки в соціальних мережах, додатки ERP, журнали клієнтів, фінансові звіти, електронні листи, презентації та звіти, створені працівниками. Поєднання всіх цих даних для підготовки звітів є складним завданням.

Складність управління якістю даних

Великі дані не є на 100% точними. Надійність та достовірність зібраних даних слід взяти під контроль. Існує ціла купа методів, присвячених очищенню даних. Ваші великі дані повинні мати відповідну модель. Тільки створивши це, можна приступити до:

- Порівняння даних з єдиною точкою істини (наприклад, порівняйте варіанти адрес із їх написаннями в базі даних поштової системи).
- Збігайте записи та об'єднуйте їх, якщо вони стосуються одного і того ж об'єкта.

Захист даних

Захист цих величезних наборів даних є однією з найстрашніших проблем великих даних. Часто компанії настільки зайняті розумінням, зберіганням та аналізом своїх наборів даних, що вони відкладають безпеку даних на більш пізні етапи. Але це не розумний крок, оскільки незахищені сховища даних можуть стати місцем розмноження зловмисних хакерів. Компанії можуть втратити до 3,7 мільйона доларів за вкрадені записи або порушення даних. [5]

Formatted: Ukrainian

Заплутана різноманітність технологій великих даних

Технологія великих даних змінюється швидкими темпами. Кілька років тому Apache Hadoop була популярною технологією для обробки великих даних. Потім у 2014 році була представлена Apache Spark. Сьогодні поєднання двох фреймворків є найкращим підходом, щоб не відставати від технології великих даних.

Deleted: .

Deleted: Н

1.4. Якість великих даних

Порівняно низька якість великих даних може бути надзвичайно шкідливою. Наприклад, якщо інструмент великих даних аналізує діяльність клієнтів на веб-сайті. У цьому випадку ведення 100% точних записів про

діяльність відвідувачів не було б необхідним лише для того, щоб побачити загальну картину. Насправді це навіть неможливо досягти.

Однак якщо аналітика великих даних відстежує дані в реальному часі від кардіо моніторів у лікарні, похибка в 3% може означати неможливість врятувати чиєсь життя.

Отже, тут все залежить від конкретної справи. А це означає, що перед тим, як поспішати з просуванням даних до найвищого рівня точності, потрібно зрозуміти бізнес-потребу, а потім встановити рівень якості великих даних.

Щоб відрізнити погані та брудні дані від хороших та чистих, слід встановити критерії "хороших даних". Ось 5 обраних критеріїв, що стосуються якості даних загалом [6]:

Formatted: Ukrainian

- **Послідовність** - логічні відносини. У корельованих наборах даних не повинно бути невідповідностей, таких як дублювання, суперечності, прогалини. Наприклад, має бути неможливо мати два однакових посвідчення для двох різних співробітників або посилатися на неіснуючий запис в іншій таблиці.
- **Точність** - реальний стан речей. Дані повинні бути точними, безперервними та відображати реальність. Всі розрахунки на основі таких даних показують справжній результат.
- **Комплектність** - всі необхідні елементи. Дані, ймовірно, складаються з декількох елементів. У цьому випадку потрібно мати усі взаємозалежні елементи, щоб забезпечити правильну інтерпретацію даних. Приклад: безліч даних датчиків, але немає інформації про точне розташування датчиків. Таким чином, буде неможливо зрозуміти, як «поводиться» обладнання вашого заводу і що впливає на цю поведінку.
- **Аудит** - технічне обслуговування та контроль. Самі дані та процес управління даними в цілому повинні бути організовані таким чином, щоб аудит якості даних можна було проводити регулярно або на вимогу. Це допоможе забезпечити більш високий рівень достатності даних.

- **Впорядкованість** - структура та формат. Дані повинні бути впорядковані в певному порядку. Він повинен відповідати всім вимогам щодо формату даних, їх структури, діапазону адекватних значень, конкретних бізнес-правил тощо. Наприклад, температуру в духовці потрібно вимірювати за Фаренгейтом і не може становити -14 ° F.

1.5. Поширені проблеми якості даних

У сучасних системах зберігання та обробки інформації виділяють наступні проблеми [6]:

- **Дубльовані дані**. Коли існує безліч систем, які часто використовуються для корпоративних поїздок, дублювання даних стає неминучим. Ту саму поїздку можна забронювати через агентство та одночасно з'явитись у стрічці кредитних карток. Обидві ці системи потрібно поєднати для загальної вартості поїздки - залишаючи нам дублікат записів. Повинен існувати відповідний процес перевірки даних із інструментами дедуплікації даних, щоб перебирати дані та ідентифікувати повторювані записи - навіть якщо запис чи ім'я не зовсім однакові, але мають певну подібність. Оскільки кожен постачальник джерел даних має різний метод написання однієї і тієї ж інформації, наприклад, назву власності готелю, наприклад, переконайтеся, що інструмент дедуплікації даних розпізнає подібні точки даних і може позначити їх для дедуплікації.
- **Не заповнені поля**. Якщо дані збираються вручну шляхом введення людьми, неповні поля є загальною небезпекою. Агенти, що поспішають, іноді не фіксують кожен біт інформації з телефонного бронювання, як і подорожуючі, які подають звіти про витрати, не завжди надають повну інформацію про те, ким був продавець за рахунок. Цю проблему можна подолати за допомогою систем, які не дозволяють подавати бронювання, якщо не заповнені всі поля. Налаштуйте системи управління даними, які автоматично виключають неповні записи - групуючи їх для подальшого аналізу. Якщо використовується стороннє рішення для управління даними,

Formatted: Font: 14 pt, Not Bold

Deleted: ¶

Formatted: Font: 14 pt, Not Bold, Ukrainian

Formatted: Font: 14 pt, Not Bold

Commented [ПW5]: тут би посилання і нижче на пунктах

Commented [AS6R5]: Чи нормально буде показати посилання в першому речені?

переконайтеся, що воно має стосунки з постачальниками даних і може потенційно переслідувати відсутні дані.

- **Невідповідні формати**. Різні формати даних завжди є проблемою для будь-якого аналітика. Наприклад, щось таке просте, як дата, можна ввести різними способами. Це спричиняє проблеми, особливо, оскільки дані переходять з однієї системи в іншу, де неправильно трактовані дати можуть призвести до численних значних помилок даних. Пізніші розробки почали визначати формати, але застарілі протоколи не завжди відповідають цим. При роботі з джерелами даних та постачальниками слід вказати бажані формати даних. Це все одно може залишити проблеми. Переконайтеся, що є автоматизовані процеси перевірки даних на основі AI, щоб уникнути цих проблем раніше, щоб їх можна було виправити.
- **Різні мови та одиниці виміру**. Подібно до різних форматів, кожна країна має свої власні одиниці вимірювань. Якщо скласти дані з двома різними одиницями вимірювання або валютами, це може легко створити неправильні цифри. Що стосується мов, спеціальні символи, такі як висловлення та акценти, можуть спричинити хаос, якщо система не налаштована на їх обробку. Дублікати можна пропустити, а одне і те ж готельне майно можна представити кілька разів. Дані про подорожі, очевидно, стосуються різних мов та вимірювань. Валюти також слід конвертувати в одну перед будь-яким типом розрахунку.
- **Помилка людини**. Найбільшою перешкодою для якості даних є насправді люди. Співробітники та агенти можуть робити помилки, що призводять до проблем із якістю даних, помилок та неправильних наборів даних. Єдиний спосіб обмежити це - максимально зменшити людські зусилля. Ми живемо у світі, де AI з кожним днем робить автоматизацію більш можливою. Замість того, щоб змушувати співробітників заповнювати звіти про витрати, використовуйте віртуальні гаманці, які автоматично реєструють операції з витратами та безпосередні прямі покупки. Крім того, коли дані аналізуються та передаються між системами, використання

систем на основі AI та вдосконалених алгоритмів замість кількох офшорних аналітиків забезпечить якнайменше зменшення помилок людини.

У кожному аналізі завжди будуть проблеми з якістю даних, проте правильна стратегія даних, яка мінімізує можливість помилок від збору даних до обробки та аналізу, може забезпечити вирішення цих питань раніше. Тоді помилки якості даних можуть стати більш керованими.

1.6. ABC Фреймворк

На основі даних, зібраних з різних джерел, бізнес хоче приймати ключові рішення. Для підтримки цих рішень якість даних повинна бути надійною. Якщо в даних є проблеми, бізнес втрачає довіру, а великі дані стають ненадійними. Однією з важливих складових роботи з великими даними є процес ETL (Extract, Transform, Load). ETL - це тип процесу інтеграції даних, який відноситься до трьох різних, але взаємопов'язаних етапів (витяг, перетворення та завантаження) і використовується для багаторазового синтезу даних з декількох джерел для побудови сховища даних, концентратора даних або озера даних.

Три кроки складають процес ETL і дозволяють інтегрувати дані від джерела до пункту призначення [7].

Commented [ПW7]: посилання

Commented [AS8R7]: Додав

Formatted: Ukrainian

Крок 1: Витяг (Extract)

Більшість управляє даними з різних джерел і використовує ряд інструментів аналізу даних для отримання бізнес-аналітики. Щоб скласти таку складну стратегію даних, як ця робота, дані повинні мати можливість вільного переміщення між системами та програмами.

Перш ніж дані можна буде перемістити до нового пункту призначення, їх потрібно спочатку витягти з джерела. На цьому першому кроці процесу ETL структуровані та неструктуровані дані імпортуються та консолідуються в єдине сховище. Сирі дані можна отримати з широкого кола джерел, таких як існуючі

бази даних та застарілі системи, хмарні, гібридні та локальні середовища, програми продажу та маркетингу, CRM-системи, інструменти аналітики тощо.

Незважаючи на те, що це можна зробити вручну, вилучення даних, кодованих вручну, може забирати багато часу та спричиняти помилок. Інструменти ETL автоматизують процес видобутку та створюють більш ефективний та надійний робочий процес.

Крок 2: Перетворення (Transform)

На цьому етапі процесу ETL можуть застосовуватися правила та норми, які забезпечують якість даних та доступність. Процес перетворення даних складається з декількох підпроцесів:

- Очищення - невідповідності та відсутні значення в даних усуваються.
- Стандартизація - до формату даних застосовується правило форматування.
- Дедуплікація - зайві дані виключаються або відкидаються.
- Перевірка - видаляються непридатні дані та позначаються аномалії.
- Сортування - дані впорядковані за типом.
- Інші завдання - для покращення якості даних можна застосовувати будь-які додаткові / необов'язкові правила.

Перетворення даних покращує цілісність даних і допомагає гарантувати, що дані надходять до нового місця призначення повністю сумісними та готовими до використання.

Крок 3: Завантаження (Load)

Завершальним етапом процесу ETL є завантаження нещодавно перетворених даних у нове місце призначення. Дані можна завантажувати відразу (повне завантаження) або через заплановані інтервали (поступове завантаження).

- Повне завантаження. У сценарії повного завантаження ETL все, що надходить із конвеєра трансформації, надходить у нові, унікальні записи у сховищі даних. Хоча бувають випадки, коли це корисно для дослідницьких

цілей, повне завантаження створює набори даних, які зростають в геометричній прогресії і швидко можуть ускладнити підтримку.

- Поступове навантаження. Менш комплексним, але більш керованим підходом є поступове завантаження. Додаткове завантаження порівнює вхідні дані з наявними та створює додаткові записи лише у разі виявлення нової та унікальної інформації. Ця архітектура дозволяє меншим, менш дорогим сховищам даних підтримувати та керувати бізнес-аналітикою.

ETL - це сам по собі складний процес, який найбільш трудомісткий під час побудови сховища даних. Це дозволяє періодично оновлювати дані на складі (щодня, «підтримувати і не втрачати дані щоразу, коли дані завантажуються на склад за допомогою ETL.

Для цього існує система аудиторського балансу та контролю (ABC). Фреймворк ABC (audit, balance, control) для управління якістю даних (DQM) поєднує в собі три процеси, які при впровадженні повинні забезпечувати точність, послідовність, повноту, інтегрованість та своєчасність зберігання даних. Ці процеси зазвичай застосовуються до операцій ETL у сховищі даних.

Аудит (Audit) - це процес виявлення того, що сталося під час операції ETL. Баланс (Balance) - це процес підтвердження того, чи те, що сталося, було правильним чи ні. Контроль (Control) - це процес виявлення та усунення помилок, які могли статися під час процесу ETL. Процес аудиту допомагає інформувати процес балансу, що, в свою чергу, допомагає інформувати процес контролю. Малюнок 1 показує, як процеси аудиту, балансу та контролю працюють разом, щоб забезпечити високоякісні дані.

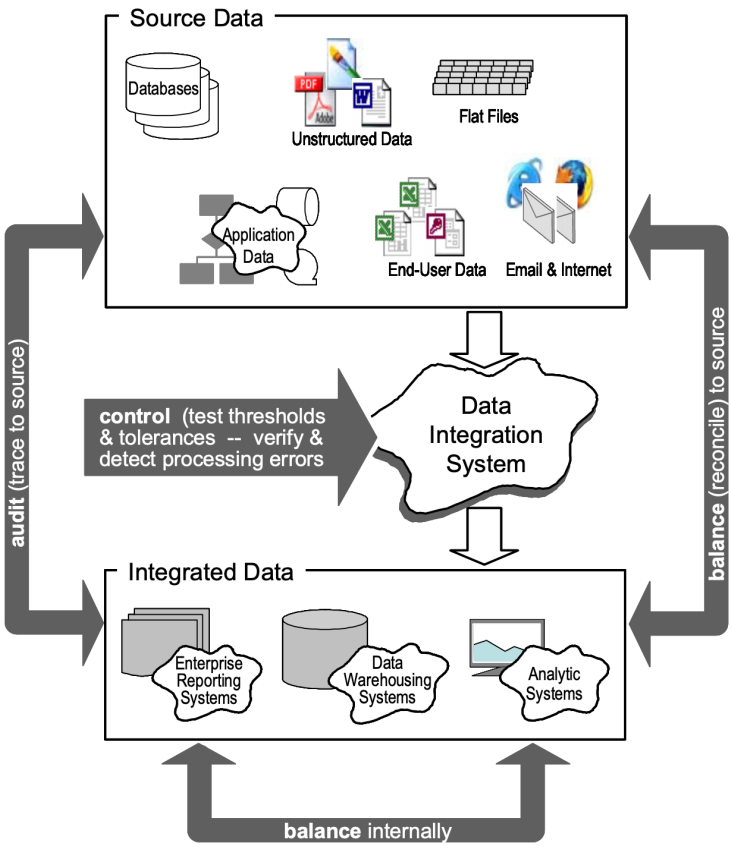

Рис 1.1. ABC Фреймворк [7]

Аудит (Audit)

Збираються різні оперативні дані, включаючи час початку та закінчення завдань ETL / ELT, кроки, кількість оброблених записів, відхилених, тип помилок, попередження тощо. Окрім надання загальних можливостей аудиту процесам, ці дані також допомагають дослідити виробничі проблеми, проаналізувати тенденції продуктивності процесу та забезпечити подальшу автоматизацію. Можливостей безмежно.

Formatted: Ukrainian

Formatted: Ukrainian

Deleted: 4

Баланс (Balance)

Відмінності слід зазначити між об’єктами джерела та цільовими даними, коли дані копіюються, переміщуються та / або перетворюються із джерела на цільове. Це можна зробити різними способами. Це може бути так просто, як порівняння підрахунків рядків або складний процес запуску набору ділових правил для "перевірки якості" цільових даних. Потрібно розробити автоматизовані процеси, і запланований цикл цих процесів повинен створювати звіти про винятки, де це можливо.

Коли дані копіюються або переміщуються з джерела в ціль (без будь-яких структурних змін), можна порівнювати кількість рядків, суму числових стовпців, кількість рядків за ключовими стовпцями тощо. Стовпці дати можна перевірити на наявність форматів.

Коли дані перетворюються з джерела на ціль, процес порівняння джерела та цілі є відносно складним, але не є практичним і не доцільним дублювати весь процес трансформації для перевірки якості даних між джерелом іціллю. На основі вимог та специфікацій картографування потрібно розробити прості процеси для узгодження даних між джерелом та ціллю, щоб мати можливість пояснити відмінності.

Контроль (Control)

Перезавантажуваність: Один із аспектів „управління” полягає у розробці процесів, щоб зробити їх більш гнучкими та „розумними” при перезапуску з останньої точки відмови замість того, щоб перезапустити з самого початку.

Обробка винятків: Процеси інтеграції даних та ETL повинні мати можливість фіксувати, повідомляти та автоматично виправляти (наскільки це можливо) всі типи винятків. Якість даних, що стосуються бізнес-команд, IT-команд, відповідних контактних даних, кодів винятків та описів, відіграє дуже важливу роль у обробці винятків та їх швидкому вирішенні.

Знову ж таки, «контрольний» аспект інтеграції даних та ETL / ELT сильно залежить від якості та деталізації зібраних даних аудиту / журналу процесів.

1.7. Огляд бізнес домену

Для прикладу, імплементація системи контролю якості даних виконана для маркетингової платформи, що працює в 74 країнах і комерціалізує продукцію в категоріях, починаючи від напівпровідників та мобільних телефонів, закінчуючи медичним обладнанням та посудомийними машинами. Клієнт представляв важливу проблему з точки зору розробки економічно ефективного способу створення та управління цими продуктами та багатьох пов'язаних з ними продуктів.

Пропозиція полягала в інтеграції з їх системою управління маркетингових матеріалів та автоматизованому створенні та оновленні цих продуктів та пов'язаних з ними активів у системі. Ця інтеграція повинна брати до уваги складність інформації про товар, яку потрібно поглинати, та великий розмір деяких активів: на продуктивність системи ніколи не слід впливати, оскільки немає можливості простою глобальної системи, яка повинна бути постійно увімкненою. Процес також повинен перевіряти наявність продуктів у системі, щоб уникнути дублювання інформації. Нарешті, результат процесу імпорту повинен повідомляти про його збій (і успіх), щоб адміністратори розуміли, що потрібно виправити, щоб забезпечити успіх майбутніх сесій імпорту.

Зовнішня система автоматизованого процесу імпорту

Служба застосунку системного передавання даних - це лямбда-сервіс AWS, який здійснює доступ до веб-служб зовнішніх систем і виконує наступне завдання:

- Один раз на день активізуються виклик зовнішньої веб-служби, щоб прочитати метадані продукту та інформацію про активи. Отримані метадані від веб-служби зовнішньої системи зберігаються в системі як файли JSON.
- Файли-активи проходять через копіювання з корейського сегмента AWS S3 у сегмент компанії S3, розміщений у Німеччині.

Formatted: Ukrainian
Formatted: Ukrainian
Formatted: Ukrainian
Formatted: Ukrainian
Commented [ПW9]: щось зайве
Commented [AS10R9]: Виправлено
Deleted: в системні
Formatted: Ukrainian
Deleted: json
Deleted: Дані проходить
Deleted: активів
Formatted: Ukrainian
Formatted: Ukrainian

- Згенерований набір файлів json передаються до сегмента компанії S3, розміщеного в Німеччині.
- Створено системний API веб-сервісу, щоб імпортувати json та активи з сегмента компанії, розміщеного в Німеччині, до бази даних системного сховища та сховища вмісту. Використовуючи цю інформацію, яку створює процес (див. Деталі зовнішнього системного API в технічній документації)

Нові продукти в менеджері продуктів

Нові активи в менеджері активів із відповідними тегами, пов'язаними бізнес-одиницями та продуктами. Наразі цей процес може імпортувати 40000 об'єктів за 2,5 години, не впливаючи на доступність та продуктивність системи.

Управління активами

Деякі користувачі відіграють важливу роль у системі, яка дозволяє їм створювати та управляти активами та адаптаціями. В цьому контексті активи - це початкові файли та варіанти їх адаптацій, які ви можете побачити на сайті в різних галереях. Наприклад, початковий файл-актив може бути векторним або растровим зображенням з високою роздільною здатністю, на основі якого система може створити десятки файлів, адаптованих під різні розміри банерів та у оптимальному форматі для веб браузерів. Незалежно від того, якого типу файл чи кінцева мета його використання, усі завантаження в Менеджер активів називаються "Активи".

Галереї активів доступні всім користувачам і дозволяють користувачам ділитися та завантажувати активи. Групи активів доступні лише для редакторів вмісту, шаблонів та адміністраторів і існують для того, щоб містити активи та адаптації та дозволяти ними керувати.

Також не існує особистого листування між галереями активів та групами активів. Галерея активів може отримувати свої активи з декількох груп активів, а активи в групі активів можуть відображатися в декількох галереях активів.

Formatted: Ukrainian

Commented [ПW12]: активи це активи … так не можна визначати поняття

Commented [AS13R12]: Додав виправлення та уточнення

Deleted: активи

Deleted:

Deleted: пристосування

Управління активами. Кожна група активів відображається у своїй власній секції із зображенням або піктограмою, вибраною адміністратором, який створив групу або керує нею (див. Рис. 1.2).

Commented [IIW14]: тут би зображення …

Commented [AS15R14]: Додав публічне зображення системи

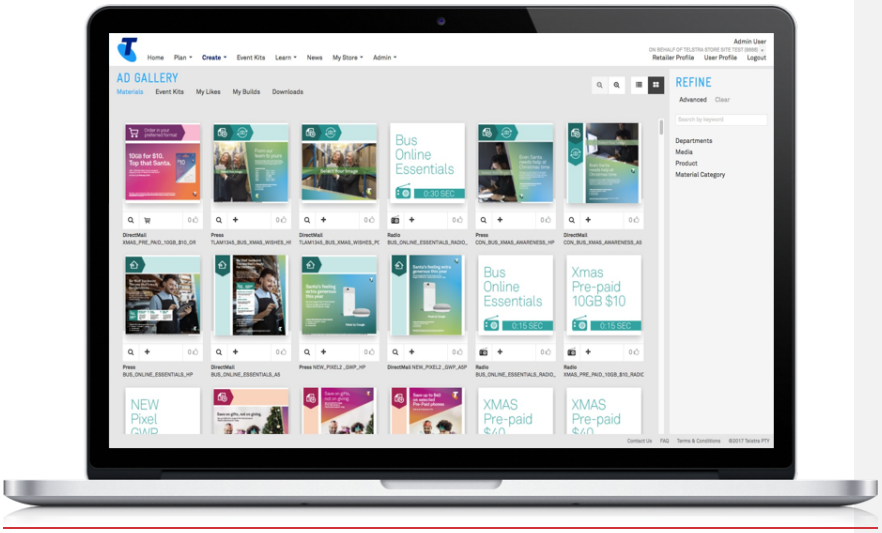

Рис 1.2 Візуальний інтерфейс управління активами та групами

Formatted: Centred

Назва групи знаходиться над зображенням, а кількість об’єктів, які вона містить, відображається під зображенням. Також є кнопка “Перегляд”, яка дозволяє користувачеві переглядати вміст Групи.

Група активів. Система відображає всі групи активів як плитки, а також плитку, яка дозволяє створювати нові групи активів (див. Рис. 1.2). Кожна плитка показує зображення, яке користувач вибрав для представлення Групи, або піктограму, якщо зображення не вибрано. Система дозволяє користувачеві вимкнути групу, якщо вона увімкнена, і ввімкнути її, якщо її вимкнено. Під зображенням плитка відображає кількість активів у групі та кнопки для перегляду та редагування групи. Група активів показує сітку всіх активів, що містяться в цій групі. Кожен актив містить у собі наступну інформацію:

Commented [IIW16]: тут би зображення

Commented [AS17R16]: Додано

Deleted: .

- Назва активу
- Статус активу

- Можливі значення: “Доступно” та “Приховано”.
- Ім'я користувача творця (особа, яка завантажила Актив)
- Дата та час його створення
- Ім'я користувача особи, яка востаннє оновила Актив
- Дата та час оновлення
- Статус затвердження (якщо схвалення включені у вашу конфігурацію)

Метадані об’єкта. У розділі "Метадані" на сторінці відображаються теги, в даний час пов'язані з активом. Для нового активу в списку Тегів за замовчуванням розміщуються теги, пов’язані з Групою активів (див. Рис. 1.3).

Commented [ПW18]: приклад

Commented [AS19R18]: Додав

Formatted: Russian

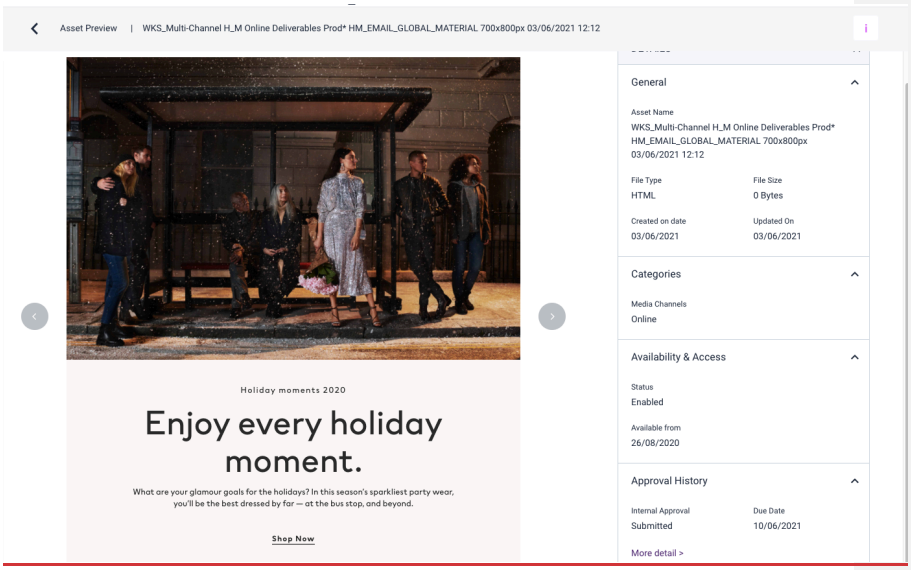

Рис. 1.3. Приклад метаданих активу

Formatted: Font: Not Bold

Не обов’язково пов’язувати теги з активом, але доцільно, оскільки теги використовуються для пошуку активів. Теги можна додавати або видаляти в будь-який час.

Теги існують у категоріях, і категорії можуть містити підкатегорії. Щоб додати тег, почніть із вибору категорії зі спадного меню. Якщо вибрана категорія

має підкатегорії, система відобразить спадне меню для вибору підкатегорії. Не всі категорії мають підкатегорії.

Після вибору категорії та підкатегорії користувач може вибрати тег із доступного спадного меню або ввести новий тег. При введенні нового тегу переконайтесь, що тег відповідає категорії та підкатегорії - це дуже важко видалити теги, які були додані до неправильної категорії та / або підкатегорії, тому ми радимо бути обережними, коли думаємо про додавання нового тегу.

Система відображає теги, які в даний час пов'язані з активом. Щоб видалити одну з тегів, клацніть на “x” у верхньому правому куті тегу. Зауважте, що це призведе до видалення тегу з активу, але не видалення тегу з категорії та / або підкатегорії.

РОЗДІЛ 2. Розробка архітектури

Архітектура програмного забезпечення займається створенням високорівневої структури програмного забезпечення. Це результат складання певної кількості архітектурних елементів у деяких добре підібраних формах для задоволення основних вимог до функціональності та продуктивності системи, а також деяких інших, нефункціональних вимог, таких як надійність, масштабованість, портативність та доступність.

Щоб описати розроблену систему з різних точок, було обрано модель архітектурного вигляду 4+1 [8]. Модель використовується для опису системи з точки зору різних зацікавлених сторін, таких як кінцеві користувачі, розробники та менеджери проектів. Чотири погляди моделі - це логічний, програмний, технологічний та фізичний погляди. Крім того, вибрані випадки використання або сценарії використовуються для ілюстрації архітектури, яка виглядає як "плюс один".

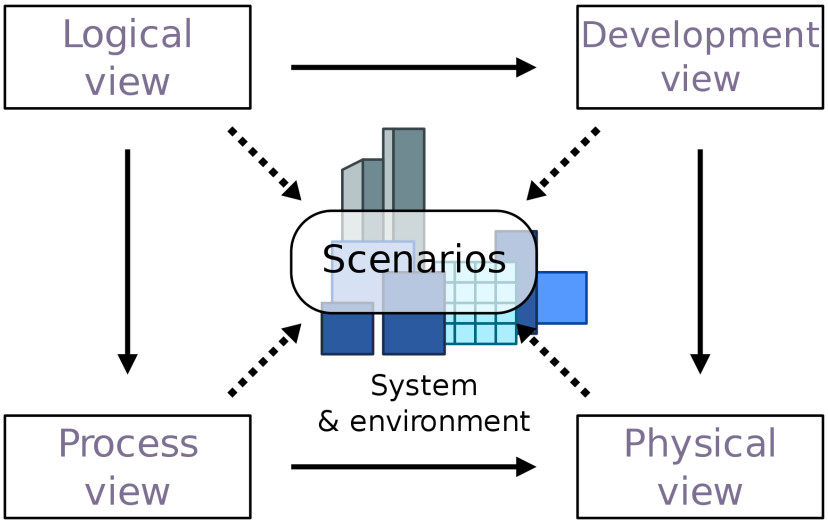

Рис 2.1 Модель архітектурного виду 4+1 [5]

Deleted: ¶
¶
¶
¶

Formatted: Ukrainian

Commented [ПW20]: не може архітектура займатися розробкою

Commented [AS21R20]: Виправлено

Deleted: розробкою

Deleted: та реалізацією

Commented [ПW22]: було обрано

Commented [AS23R22]: Виправлено

Deleted: я обрав

Formatted: Ukrainian

Deleted: 9

Formatted: Ukrainian

2.1. Процесний вигляд

Процесний погляд стосується динамічних аспектів системи, пояснює системні процеси та спосіб їх взаємодії та фокусується на поведінці системи під час виконання. Подання процесу стосується паралельності, розподілу. Архітектура процесу враховує деякі нефункціональні вимоги, такі як продуктивність та доступність. У ньому розглядаються питання паралельності та розподілу, інтеграторів, продуктивності, масштабованості та цілісності системи, відмовостійкості.

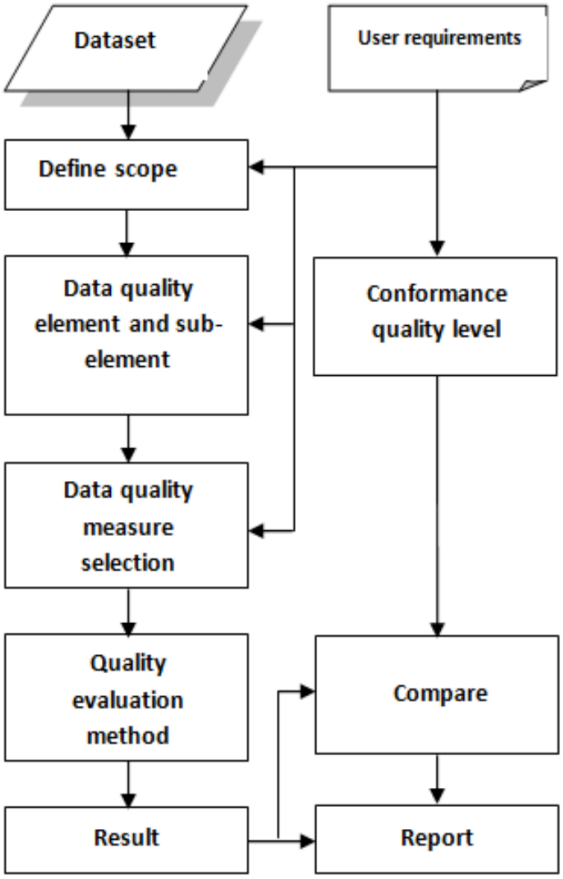

Рис 2.2 Загальний вигляд перевірки якості даних

Formatted: Ukrainian

Formatted: Ukrainian

Діаграми діяльності - це графічні зображення робочих процесів поетапних дій та дій із підтримкою вибору, ітерації та паралельності. В Уніфікованій мові моделювання діаграми діяльності можуть бути використані для опису ділових та операційних покрокових робочих процесів компонентів у системі. Діаграма діяльності показує загальний потік контролю та в основному використовується для представлення подання процесу.

2.2. Фізичне представлення

Фізичний вигляд зображує систему з точки зору системного інженера. Це стосується топології програмних компонентів на фізичному рівні, а також фізичних зв'язків між цими компонентами. Цей вигляд також відомий як перегляд розгортання. Фізична архітектура враховує насамперед такі нефункціональні вимоги системи, як доступність, надійність (відмовостійкість), продуктивність (пропускна здатність) та масштабованість. Програмне забезпечення виконується в мережі комп'ютерів або обробних вузлів. Отже, відображення програмного забезпечення до вузлів має бути дуже гнучким і мати мінімальний вплив на сам вихідний код.

Для підготовки фізичного представлення також важливо вибрати набір продуктів та технологій, які будуть використовуватись для побудови системи. Для цього було розглянуто 2 основних провайдера хмарних сервісів: Amazon AWS та Microsoft Azure. Нижче приведена порівняльна таблиця:

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Таблиця 2.1. Порівняння сервісів Amazon AWS та Microsoft Azure

Опис сервісу	**AWS**	**Azure**
Віртуальні сервери дозволяють користувачам розгортати, керувати та підтримувати ОС та серверне програмне забезпечення.	Elastic Compute Cloud (EC2)	Virtual Machines
Платформа керованого хостингу	Elastic Beanstalk	App Service

Хмарний сервіс для навчання, розгортання, автоматизації та управління моделями машинного навчання.	SageMaker Machine Learning	
Хмарне сховище корпоративних даних (EDW), яке використовує масово паралельну обробку (MPP) для швидкого запуску складних запитів через петабайти даних.	Redshift	Synapse Analytics
Повністю керована платформа для аналізу великих даних з низькою затримкою для запуску складних запитів через петабайти даних.	EMR	Azure Data Explorer
Аналітична платформа на основі Apache Spark	EMR	Databricks
Керований сервіс Hadoop. Розгортайте та керуйте кластерами Hadoop в Azure	EMR	HDInsight
Створюйте, плануйте, організовуйте та керуйте конвеєрами даних.	Data Pipeline Glue	Data Factory
Система реєстрації та система виявлення для корпоративних джерел даних	Glue	Data Catalog
Інструменти бізнес-аналітики, які створюють візуалізацію, проводять спеціальний аналіз та розробляють ділові ідеї на основі даних.	QuickSight	Power BI
Інтегруйте системи та запускайте серверні процеси у відповідь на події або розклади без забезпечення серверів та керування ними.	Lambda	Functions
Керована служба реляційних баз даних	RDS	SQL Database

Послуги, що дозволяють масовому поглинанню невеликих входів даних, як правило, від пристроїв та датчиків, обробляти та направляти дані.	Kinesis Streams	Event Hubs
Служба зберігання об’єктів	Simple Storage Services (S3)	Blob storag

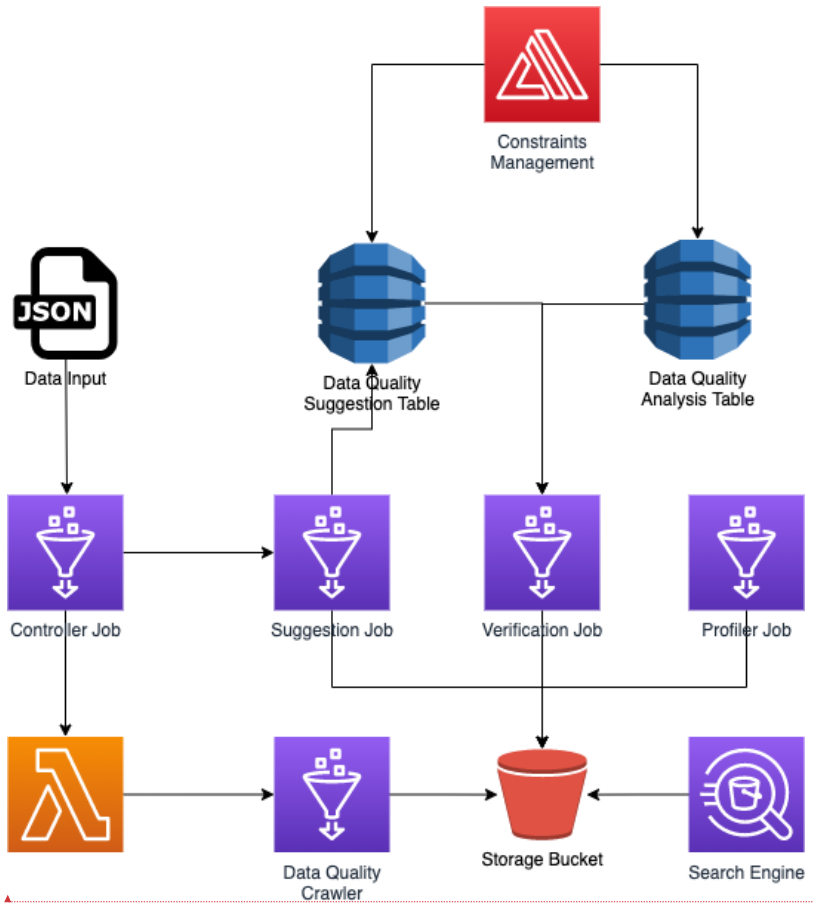

Рис. 2.3. Фізичне представлення системи

Formatted: Ukrainian

Formatted: Ukrainian

2.3. Логічний вигляд

Логічний погляд стосується функціональності, яку система надає кінцевим споживачам. Логічна архітектура в першу чергу підтримує те, що система повинна надавати з точки зору послуг для своїх користувачів. Система розкладається на набір ключових абстракцій, взятих (переважно) із проблемної області, у вигляді об'єктів або класів об'єктів. Це розкладання призначене не лише для функціонального аналізу, але також служить для виявлення загальних механізмів та елементів проектування в різних частинах системи.

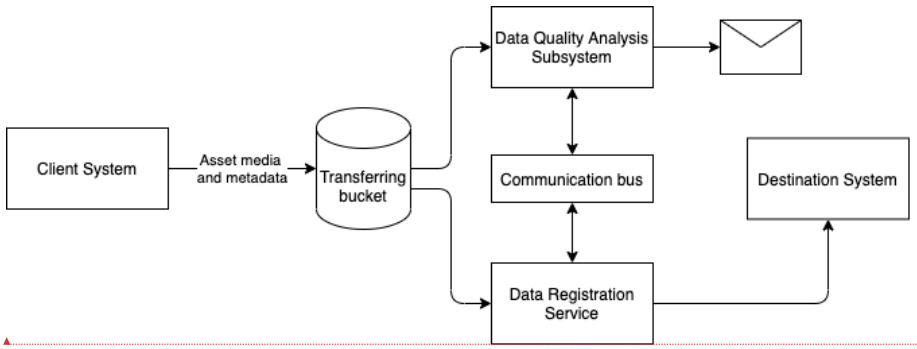

Рис. 2.4. Загальний логічний вигляд архітектури підсистеми перевірки якості даних

2.4. Погляд на розробку

Погляд на розробку ілюструє систему з точки зору програміста і стосується управління програмним забезпеченням. Цей погляд також відомий як подання реалізації. Архітектура розробки фокусується на фактичній організації програмного модуля в середовищі розробки програмного забезпечення. Архітектура розробки системи представлена діаграмами модулів та підсистем, що відображають взаємозв'язки "експорт" та "імпорт". Повну архітектуру розробки можна описати лише тоді, коли всі елементи програмного забезпечення

Formatted: Ukrainian

Formatted: Ukrainian

визначені. Схема пакетів відображає залежності між пакетами, що складають модель.

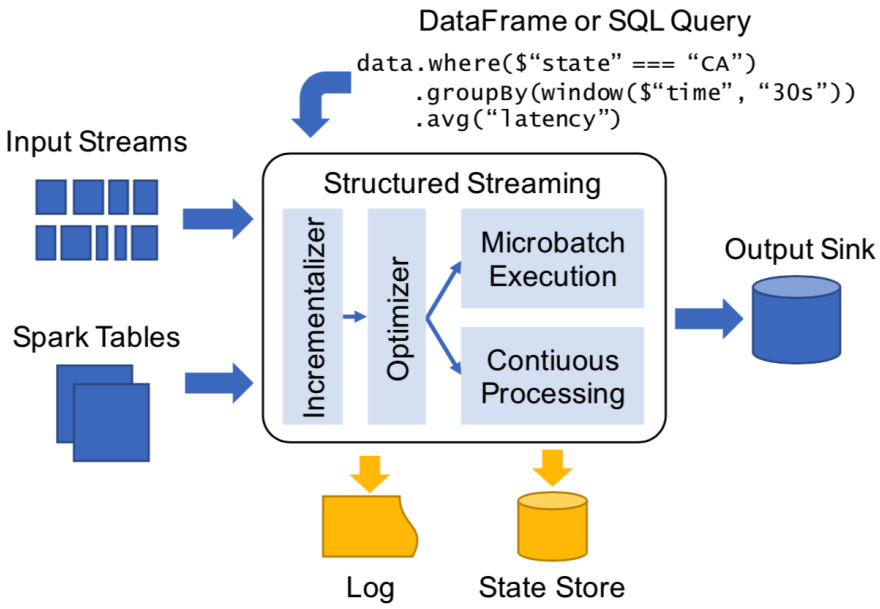

Рис 2.5. Набір компонентів підсистеми аналізу якості даних

При проектуванні системи було обрано її подання з використанням архітектурної моделі 4+1. Ця модель насправді дозволила різним зацікавленим сторонам знайти те, що вони хочуть знати про архітектуру програмного забезпечення. Системні інженери підходять до нього з фізичного погляду, а потім з перегляду процесу. Кінцеві користувачі, клієнти, спеціалісти з обробки даних з логічного подання. Керівники проектів, співробітники конфігурації програмного забезпечення бачать це з точки зору Розробки.

Такий огляд розробленої системи дав можливість моделювати її поведінку, розуміти її місце в середовищі розробки авіаційного симулятора, визначати її основні компоненти та взаємозв'язки між ними. Все це призводить до полегшення процесу розвитку.

Formatted: Ukrainian

Formatted: Ukrainian

РОЗДІЛ 3. Реалізація програмного застосунку

Для практичної реалізції та з урахуванням вибранного постачальника хмарних рішень, пісистема валідації та верифікації рекламних матеріалів було побудовано на основі стеку Amazon AWS, а саме:

- Amazon S3
- AWS Glue
- AWS Deequ
- AWS DynamoDB

При цьому, основна частина логіки перевірки валідності побудована за допомогою бібліотек Deequ [9]. Нижче приведено характеристику та використані мосливості даного програмного продукту.

3.1. Характеристики Deequ

Deequ внутрішньо використовується в Amazon для перевірки якості багатьох великих виробничих наборів даних. Виробники набору даних можуть додавати та редагувати обмеження якості даних. Система регулярно обчислює показники якості даних (з кожною новою версією набору даних), перевіряє обмеження, визначені виробниками набору даних, і публікує набори даних для споживачів у разі успіху. У випадках помилок публікацію набору даних можна зупинити, і виробники отримують повідомлення про необхідність вжити заходів.

Щоб використовувати Deequ, давайте розглянемо його основні компоненти (також показано на малюнку 3.1.).

Formatted: Ukrainian
Formatted: Ukrainian
Formatted: Ukrainian
Formatted: Ukrainian
Formatted: Ukrainian
Deleted: 10
Formatted: Ukrainian
Formatted: Ukrainian

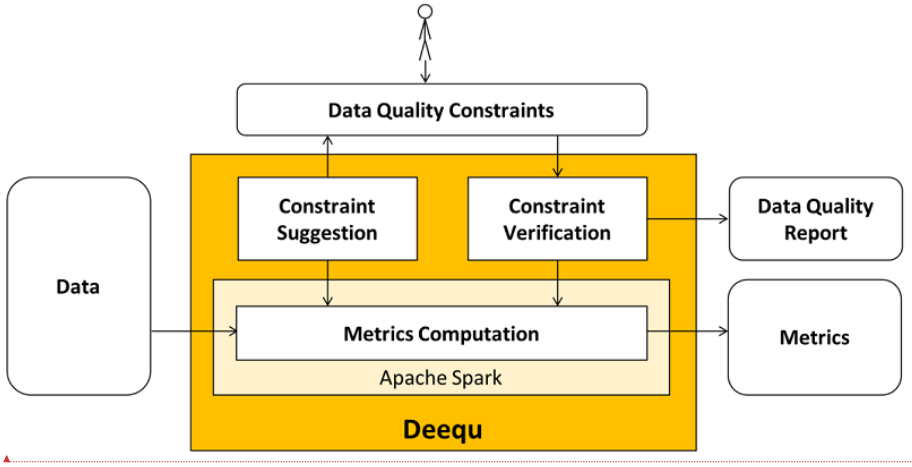

Рис. 3.1 Огляд компонентів Deequ

Обчислення метрик. Deequ обчислює показники якості даних, тобто такі статистичні дані, як повнота, максимум або кореляція. Deequ використовує Spark для читання з таких джерел, як Amazon S3, та для обчислення метрик за допомогою оптимізованого набору запитів агрегації. Також є прямий доступ до необроблених показників, обчислених на основі даних.

Перевірка обмежень. Deeqr має можливість визначити набір обмежень якості даних, які слід перевірити. Deequ піклується про виведення необхідного набору метрик, що обчислюються на даних. Deequ створює звіт про якість даних, який містить результат перевірки обмеження.

Пропозиція обмежень. Також можливо визначити власні обмеження якості даних, або використовувати автоматизовані методи пропозицій обмежень, які профілюють дані, щоб вивести корисні обмеження.

Таблиця 3.1. Показники Deequ

Метрика	Опис
ApproxCountDistinct	Приблизна кількість різного значення, обчислене за допомогою ескізів HyperLogLogPlusPlus.
ApproxQuantile	Приблизний квантил розподілу.

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

ApproxQuantiles	Приблизні квантили розподілу.
Completeness	Частка ненульових значень у стовпці.
Compliance	Частка рядків, які відповідають заданому обмеженню стовпців.
Correlation	Коефіцієнт кореляції Пірсона, вимірює лінійну кореляцію між двома стовпцями. Результат знаходиться в діапазоні [-1, 1], де 1 означає позитивну лінійну кореляцію, -1 означає негативну лінійну кореляцію, а 0 означає відсутність кореляції.
CountDistinct	Кількість різних значень.
DataType	Розподіл типів даних, таких як логічний, дробовий, інтегральний та рядок. Отримана гістограма дозволяє фільтрувати відносні або абсолютні частки.
Distinctness	Частка різних значень стовпця на кількість усіх значень стовпця. Виразні значення трапляються принаймні один раз. Приклад: [a, a, b] містить два різних значення a і b, тому різниця становить 2/3.
Entropy	Ентропія - це міра рівня інформації, що міститься в події (значення в стовпці) при розгляді всіх можливих подій (значення в стовпці). Вимірюється в натах (природних одиницях інформації). Ентропія оцінюється за допомогою підрахунків спостережуваних значень як від'ємної суми (count_count / total_count) * log (value_count / total_count). Приклад: [a, b, b, c, c] має три різні значення з підрахунками [1, 2, 2]. Тоді ентропія (-1 / 5 * log (1/5) -2 / 5 * log (2/5) -2 / 5 * log (2/5)) = 1.055.
Maximum	Максимальне значення.
Mean	Середнє значення; нульові значення виключаються.
Minimum	Мінімальне значення.

MutualInformation	Взаємна інформація описує, скільки інформації про один стовпець (одну випадкову величину) можна вивести з іншого стовпця (інша випадкова величина). Якщо два стовпці незалежні, взаємна інформація дорівнює нулю. Якщо один стовпець є функцією іншого стовпця, взаємна інформація є ентропією стовпця. Взаємна інформація є симетричною та невід'ємною.
PatternMatch	Частка рядків, що відповідає заданому регулярному досліду.
Size	Кількість рядків у DataFrame.
Sum	Сума всіх значень стовпця.
UniqueValueRatio	Частка унікальних значень на кількість усіх різних значень стовпця. Унікальні значення трапляються рівно один раз; різні значення трапляються принаймні один раз. Приклад: [a, a, b] містить одне унікальне значення b та два різних значення a і b, тому унікальне співвідношення значень дорівнює 1/2.
Uniqueness	Частка унікальних значень на кількість усіх значень стовпця. Унікальні значення трапляються рівно один раз. Приклад: [a, a, b] містить одне унікальне значення b, тому унікальність дорівнює 1/3.

Formatted: Ukrainian

3.2. Імплементація перевірок якості даних

У наступному прикладі вікористовується AnalysisRunner для визначення метрик, які нас цікавлять. Можливо запустити наступний код в оболонці Spark, просто вставивши його в оболонку або збереживши в локальному файлі на майстрі і завантажте його в оболонку Spark наступною командою:

```
import com.amazon.deequ.analyzers.runners.{AnalysisRunner, AnalyzerContext}
import com.amazon.deequ.analyzers.runners.AnalyzerContext.successMetricsAsDataFrame
```

```
import com.amazon.deequ.analyzers.{Compliance, Correlation, Size, Completeness,
Mean, ApproxCountDistinct}

val analysisResult: AnalyzerContext = { AnalysisRunner
  // data to run the analysis on
  .onData(dataset)
  // define analyzers that compute metrics
  .addAnalyzer(Size())
  .addAnalyzer(Completeness("review_id"))
  .addAnalyzer(ApproxCountDistinct("review_id"))
  .addAnalyzer(Mean("star_rating"))
  .addAnalyzer(Compliance("top star_rating", "star_rating >= 4.0"))
  .addAnalyzer(Correlation("total_votes", "star_rating"))
  .addAnalyzer(Correlation("total_votes", "helpful_votes"))
  // compute metrics
  .run()
}

// retrieve successfully computed metrics as a Spark data frame
val metrics = successMetricsAsDataFrame(spark, analysisResult)
```

Проаналізувавши та зрозумівши дані, ми хочемо перевірити, чи отримані нами властивості також зберігаються для нових версій набору даних. Визначивши твердження щодо розподілу даних як частини конвеєра даних, ми можемо забезпечити високу якість кожного обробленого набору даних і що будь-яка програма, яка споживає дані, може покладатися на нього.

Для написання тестів на дані ми починаємо з VerificationSuite і додаємо перевірки на атрибути даних. У цьому прикладі ми перевіряємо такі властивості наших даних:

```
import com.amazon.deequ.{VerificationSuite, VerificationResult}
import com.amazon.deequ.VerificationResult.checkResultsAsDataFrame
import com.amazon.deequ.checks.{Check, CheckLevel}

val verificationResult: VerificationResult = { VerificationSuite()
  // data to run the verification on
  .onData(dataset)
  // define a data quality check
  .addCheck(
    Check(CheckLevel.Error, "Review Check")
      .hasSize(_ >= 3000000) // at least 3 million rows
      .hasMin("star_rating", _ == 1.0) // min is 1.0
      .hasMax("star_rating", _ == 5.0) // max is 5.0
      .isComplete("review_id") // should never be NULL
      .isUnique("review_id") // should not contain duplicates
      .isComplete("marketplace") // should never be NULL
      // contains only the listed values
      .isContainedIn("marketplace", Array("US", "UK", "DE", "JP", "FR"))
```

```
      .isNonNegative("year")) // should not contain negative values
  // compute metrics and verify check conditions
  .run()
}

// convert check results to a Spark data frame
val resultDataFrame = checkResultsAsDataFrame(spark, verificationResult)
```

3.3. Розроблені перевірки якості даних

Formatted: Indent: Left: -0.02 cm, Outline numbered + Level: 2 + Numbering Style: 1, 2, 3, … + Start at: 3 + Alignment: Left + Aligned at: 1.9 cm + Indent at: 3.17 cm

Нижче наведено приклад атомарних документів з метаданими до маркетингових файлів-активів, що використовуються в подальшому для формування рекламних матеріалів:

Схема продуктів

```
{
  "ID": "45d7ba5e-764a-4e46-a418-44f5944a3932",
  "Name": "Product 1.1",
  "SKU": "TSH-MED-WHI-COT",
  "Country": "UA"
}
{
  "ID": "8bc6a934-ca68-4d29-a4c1-8274eebb6b05",
  "Name": "Product 1.2",
  "SKU": "TSH-MED-WHI-COT-2",
  "Country": "UA"
}
{
  "ID": "98addacd-75cd-4222-a0cf-7b871c2f66ab",
  "Name": "Product 1.3",
  "SKU": "TSH-MED-WHI-COT-3",
  "Country": "UK"
}
```

Схема метаданих до файлів-активів

```
{
  "Name": "Product 1 in Black",
  "ID": "45d7ba5e-764a-4e46-a418-44f5944a3932",
  "SKUs": ["TSH-MED-WHI-COT", "TSH-MED-WHI-COT-2", "TSH-MED-  WHI-COT-3"],
  "File": "/Product1/BlackImage.png"
}
{
  "Name": "Product 1 in White",
  "ID": "45d7ba5e-764a-4e46-a418-44f5944a3932",
  "SKUs": ["TSH-MED-WHI-COT", "TSH-MED-WHI-COT-3"],
  "File": "/Product1/BlackImage"
}
```

Кожний файл-актив може бути використаний для рекламування декількох продуктів. При цьому, помилки в даній моделі даних можуть бути 2 рівнів:

1. Помилки рівня логіки функціонування системи – ті, що можуть порушити штатний режим роботи та вилитись у винятковій ситуації аж до несподіваного вимкнення
2. Помилки рівня бізнес процесів – це ситуації, коли дані, представленні в систему є логічно правильними, проте не відповідають очікуванням бізнес напрямку

Для перевірки даних на наявність помилок першого рівня було реалізовано наступні валідації:

- JSON файли містять усі необхідні поля
- Усі необхідні поля заповнені даними очікуваних типів (для прикладу, у "Product 1 in White" є обов’язкове поле File, проте воно містить посилання на файл без розширення)

Для верифікації даних з точки зору бізнесу, було розроблено наступні перевірки:

- Файли-активи промарковані продуктами тим країн, в яких є активні маркетингові кампанії
- Файл-актив промаркований усіма необхідними продуктами

Таким чином, реалізована. Функціональність допомагає клієнту провести валідацію та верифікацію якості даних та забезпечити якісний рівень обробки даних для подальшого використання у бізнес-процесах по всьому світу.

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Висновки по роботі та рекомендації для подальших досліджень

В першому розділі було проведено аналіз теоретичних й прикладних аспектів даних великої розмірності та особливості роботи з ними. Виділено характеристики якості даних, а також як один з варіантів досягнення високої продуктивності через об'єднання обробки магістраллю ETL (Extract – Transform - Load) з ABC (Audit – Balance – Control) як двома асинхронними процесами, котрі можуть спілкуватись між собою задля отримання якісних характеристик без втрати швидкості обробки даних. Також, для подальшої розробки прикладної моделі, було приведено приклад особливостей бізнес-домену, з розрахунку якого було в подальшому створено архітектуру системи та реалізовано підсистему існуючого продукту, як доказ концепції.

Другий розділ присвячено аналізу підходів до розробки архітектури системи валідації та верифікації якості даних, було досліджено програмні продукти, що можуть бути використані як компоненти в архітектурі системи. Також обрано програмні компоненти, розроблено та представлено архітектуру підсистеми на основі хмарного рішення Amazon AWS.

В третьому розділі розглянуто практичну реалізацію програмного комплексу, що побудований на базі розробленої архітектури та з використанням Amazon Deequ. Здійснено інтеграцію між різними компонентами системи, та реалізовано можливість перевірки даних на наявність логічних та бізнес порушень.

Отримана в результаті розробки архітектура може бути в подальшому розширена новими компонентами за необхідності. Зокрема, можуть бути додані компоненти для перевірки даних за новими правилами, а також розширена інтеграція с іншими системами через HTTP канали.

В цілому, розроблений на базі архітектури програмний комплекс підтверджує актуальність та ефективність даної архітектури, та може служити прототипом для подальшого розширення та використовуватися в інших бізнес доменах.

Deleted: високого

Commented [IIW24]: високого рівня чого?

Commented [AS25R24]: Виправлено

Deleted: рівня

Deleted:

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Formatted: Ukrainian

Список літератури

1. Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2024. [Електронний ресурс]. Режим доступу: https://www.statista.com/statistics/871513/worldwide-data-created/
2. Big Data in Big Companies. [Електронний ресурс]. Режим доступу: https://docs.media.bitpipe.com/io_10x/io_102267/item_725049/Big-Data-in-Big-Companies.pdf
3. IBM Journey to AI Blog. [Електронний ресурс]. Режим доступу: https://www.ibm.com/blogs/journey-to-ai/
4. TDWI Data Integration Techniques. ETL and Alternatives for Data Consolidation. [Електронний ресурс]. Режим доступу: http://download.101com.com/pub/TDWI/Files/TDWI_Data_Integration_Techniques_(preview)_2008v1.pdf
5. Gartner's Big Data Definition Consists of Three Parts, Not to Be Confused with Three "V"s. [Електронний ресурс]. Режим доступу: https://www.upgrad.com/blog/major-challenges-of-big-data/
6. Dirty, clean or cleanish: what's the quality of your big data? [Електронний ресурс]. Режим доступу: https://www.scnsoft.com/blog/big-data-quality
7. IEEE. Big data: A review. [Електронний ресурс]. Режим доступу: https://ieeexplore.ieee.org/abstract/document/6567202
8. 4+1 architectural view model. [Електронний ресурс]. Режим доступу: https://en.wikipedia.org/wiki/4%2B1_architectural_view_model
9. AWS Big Data Blog. Test data quality at scale with Deequ. [Електронний ресурс]. Режим доступу: https://aws.amazon.com/blogs/big-data/test-data-quality-at-scale-with-deequ/

Deleted: ¶
Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2024. [Електронний ресурс]. Режим доступу: https://www.statista.com/statistics/871513/worldwide-data-created/¶
Big Data in Big Companies. [Електронний ресурс]. Режим доступу: https://docs.media.bitpipe.com/io_10x/io_102267/item_725049/Big-Data-in-Big-Companies.pdf¶
IBM Journey to AI Blog. [Електронний ресурс]. Режим доступу: https://www.ibm.com/blogs/journey-to-ai/¶
. [Електронний ресурс]. Режим доступу: https://www.forbes.com/sites/gartnergroup/2013/03/27/gartners-big-data-definition-consists-of-three-parts-not-to-be-confused-with-three-vs/?sh=7f980c8b42f6 ¶
Gartner's Big Data Definition Consists of Three Parts, Not to Be Confused with Three "V"s. [Електронний ресурс]. Режим доступу: https://www.upgrad.com/blog/major-challenges-of-big-data/ ¶
Dirty, clean or cleanish: what's the quality of your big data? [Електронний ресурс]. Режим доступу: https://www.scnsoft.com/blog/big-data-quality ¶
TDWI Data Integration Techniques. ETL and Alternatives for Data Consolidation. [Електронний ресурс]. Режим доступу: http://download.101com.com/pub/TDWI/Files/TDWI_Data_Integration_Techniques_(preview)_2008v1.pdf ¶
IEEE. Big data: A review. [Електронний ресурс]. Режим доступу: https://ieeexplore.ieee.org/abstract/document/6567202¶
4+1 architectural view model. [Електронний ресурс]. Режим доступу: https://en.wikipedia.org/wiki/4%2B1_architectural_view_model¶
AWS Big Data Blog. Test data quality at scale with Deequ. [Електронний ресурс]. Режим доступу: https://aws.amazon.com/blogs/big-data/test-data-quality-at-scale-with-deequ/¶

Formatted: Heading 1, Indent: Left: 0 cm

Formatted: List Paragraph, Indent: Left: 0 cm, Hanging: 1.25 cm, Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 1.27 cm + Indent at: 1.9 cm

Formatted: Ukrainian

Formatted: Normal