Лекція 1.
Джерела великих даних. Інтернет Речей.
Визначення Big Data

План лекції

1.1. Інтернет Речей та зростання даних.
1.2. Платформа Kaggle.
1.3. DrivenData.
1.4. Визначення великих даних.
1.5. Приклади великих даних у реальному світі.
1.6. Відкриті дані.
1.7. Приватність даних.
1.8. Структуровані та неструктуровані дані.
1.9. Хмарні та туманні обчислення.
1.10. Дані в спокої та дані в русі.
1.11. Інфраструктура великих даних.
1.12. Розподілені дані та їх обробка.

1.1. Інтернет Речей та зростання даних

Наш світ є дуже складним. Ця складність зумовлює генерування постійно зростаючого обсягу даних, які потрібно зберігати та аналізувати. Швидкість генерування даних не виявляє ознак уповільнення. Різноманітність даних поширюється на нові області, які ніколи не були доступні для аналізу. Взаємодія між людьми, що використовують медіаплатформи, автоматизація процесів та агрегація даних, що надходять з різних джерел, створюють Інтернет речей. Інтернет речей (Internet of Things, IoT) не тільки приєднує датчики до існуючих речей, він створює ринок нових пов'язаних речей. Усі ці пов'язані речі генерують дані. Це додає немислиму кількість великих даних, що називається Big Data.

Збирати всі наявні дані в рамках проекту чи рішення не завжди можливо. Кількість даних, яку можна зібрати, визначається можливостями датчиків, мережі, комп'ютерів та іншого обладнання. Це також визначається необхідністю, наприклад, для перевірки правильності вирівнювання етикетки кожної пляшки, що рухається високошвидкісною лінією розливу напою У цьому випадку важливі дані кожної пляшки. Для іншого датчика, такого як датчик вологості в кукурудзяному полі, не потрібно повідомляти про вимірювану величину вологості кожну десяту секунду. Кожні п'ять-десять хвилин може бути

достатньо. Не всі зібрані дані можна використовувати як є у первинному вигляді. Можливо, були зібрані сторонні, невірні або неправдиві дані. Щоб зробити ці дані корисними, їх потрібно очистити. Очищення – це видалення небажаних даних, зміни неправильних даних та заповнення відсутніх даних. Для очищення даних прийнято використовувати програмний код. Це досягається шляхом пошуку критеріїв або їх відсутності та оперування даними, поки не буде більше аномалій. Після очищення даних їх можна легше шукати, аналізувати та візуалізувати. За допомогою аналізу даних можна дізнатись цікаві відомості та виявити тенденції. Це часто призводить до нових запитів, які ще не були реалізовані. Якщо не можна виявити додаткову цінність з деякого набору даних, можна експериментувати з тим, як вони організовані та представлені. Наприклад, камера безпеки, яка стежить за парковкою для злочинів, може також використовуватися для повідомлення водіям про кількість та розташування вільних місць. Якщо вважати, що кожне зерно рису еквівалентне одному байту даних, то при послідовному подвоєнні зерен в кожній наступній комірці кількість рисових зерен в останньому квадраті шахової дошки буде еквівалентна дев'яти екзабайт, як показано на рис.1.1. Один екзабайт становить приблизно 1,07 мільярда гігабайт. Дев'ять екзабайтів приблизно еквівалентні обсягу інтернет-трафіку за 2014 рік [1].

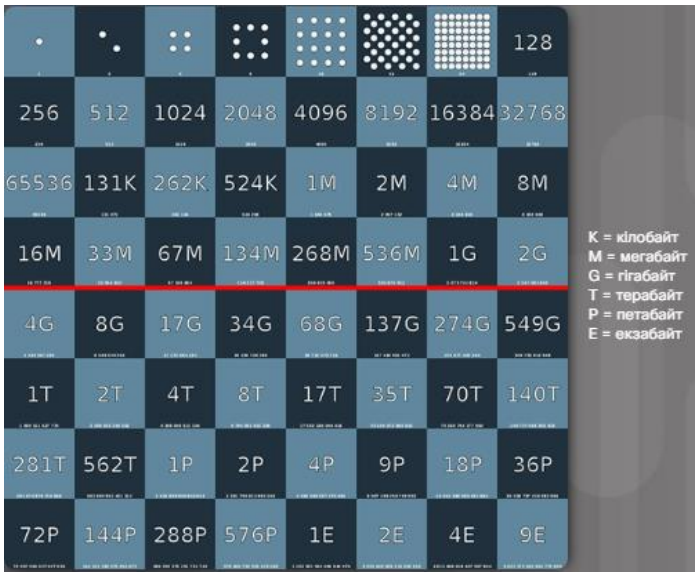
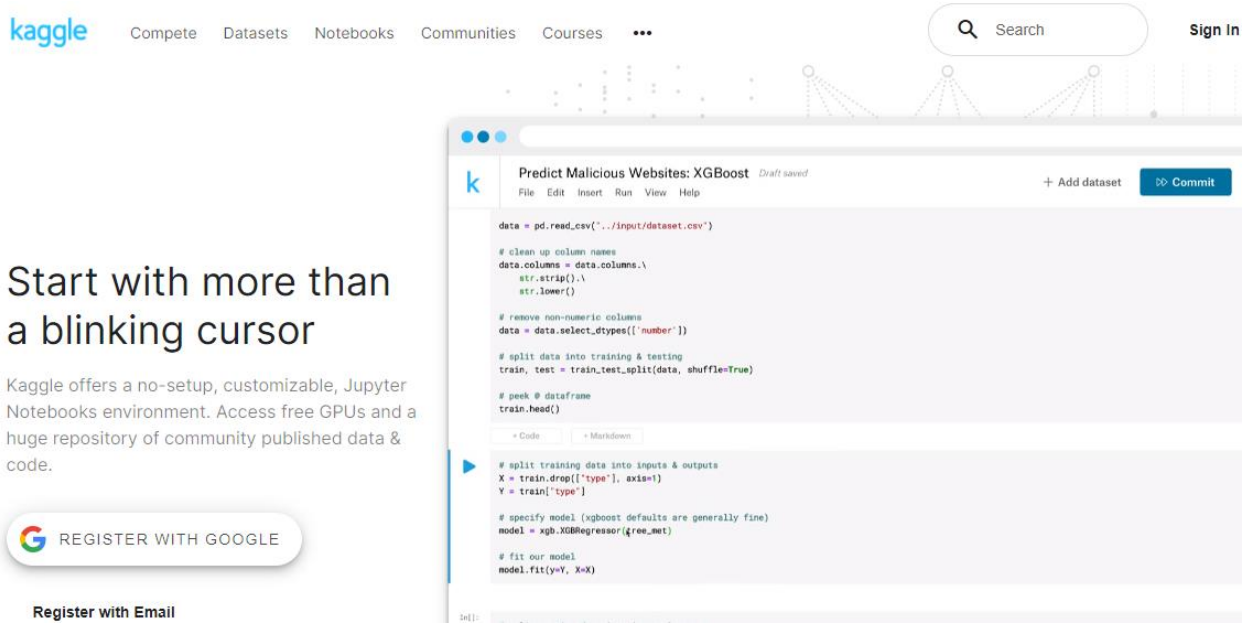


Рис.1.1. Подвоєння кількості байтів в комірках шахової дошки [2]

1.2. Платформа Kaggle

Інновації дозволяють компаніям не зупинятися в розвитку. Сьогодні все більше організацій розміщують датчики у свої продукти. Їх мета – збір та аналіз даних для отримання цінних відомостей. Щоб використовувати можливості IoT, організації потребують кваліфікованих та творчих людей. Інтернет-платформи, такі як Kaggle, дозволяють компаніям знаходити талановитих людей з різних куточків світу.

Kaggle – це платформа, яка об'єднує підприємства та організації, що мають питання щодо своїх даних та людей, які знають, як знайти відповіді на ці питання. Різні організації проводять змагання онлайн для створення кращих світових моделей даних. Учасники змаганнях генерують багато моделей, використовуючи різноманітні прийоми. Гравці з усього світу мають різну освіту та спеціалізації. Вони можуть підключитися до команд або просто допомогти один одному. Переможець або команда-переможець кожного змагання виграє приз. Зазвичай це можуть бути гроші, але іноді це може бути запрошення на роботу у відповідну компанію. У кожному змаганні учасники постійно вдосконалюються, оскільки кожен переможець долає попередній бал. Нові прогнозні моделі даних постійно перевершують існуючі кращі моделі. Клініка Майо, NASA, GE та Deloitte – це лише декілька підприємств та організацій, які приймали змагання з Kaggle [3].



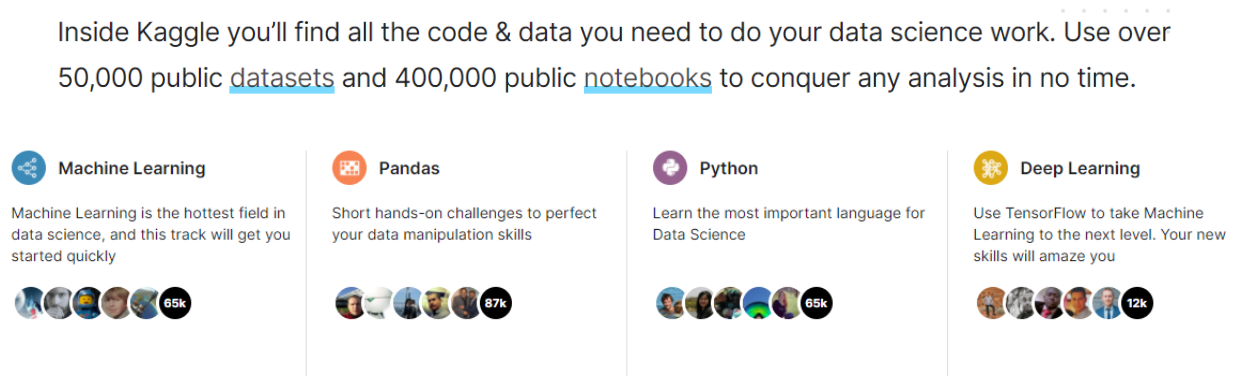


Рис.1.2. Платформа Kaggle [3]

1.3. DrivenData

Технології, що застосовуються в IoT та аналітиці даних, можуть бути використані для вирішення різних соціальних проблем. Зібрані дані можна використовувати для прогнозування різноманітних тенденцій. Наприклад, використовуючи доступні дані, підприємці в країні можуть передбачити, які водяні насоси функціонують, а які потребують ремонту чи не працюють. Завдяки прогнозуванню робота пристроїв обслуговування стає більш ефективною. Змагання, які виконуються для різних соціальних проектів, наприклад, можна знайти на веб-сайті DrivenData [4].

Місією DrivenData є використання передових практик з аналізу даних та краудсорсингу для вирішення глобальних соціальних проблем. Як і Kaggle, вони приймають виклики в Інтернеті, глобальна спільнота науковців змагається за створення найкращої статистичної моделі для складних проблем прогнозування. Ці моделі можуть сприяти позитивним змінам у світі.

DrivenData починається з постановки прогнозного питання, яке може бути вирішене за наявними даними та мати вимірюваний, реальний вплив. Вони співпрацюють з некомерційними організаціями, щоб зрозуміти їхні потреби та виявити продуктивні партнерські стосунки. Далі DrivenData проводить онлайн-конкурс з відкритими інноваціями, де розробники програм та науковці даних подають статистичні моделі. Використовуючи свою конкурентну платформу та механізм оцінювання, моделі класифікуються залежно від того, наскільки добре

вони прогнозують дані конкурентів. І нарешті, вони працюють з організацією, щоб використовувати найкращу модель, новий статистичний підхід або інструмент для аналізу даних. Це дає можливість неприбутковій організації більш ефективно та стабільно виконувати свою місію.

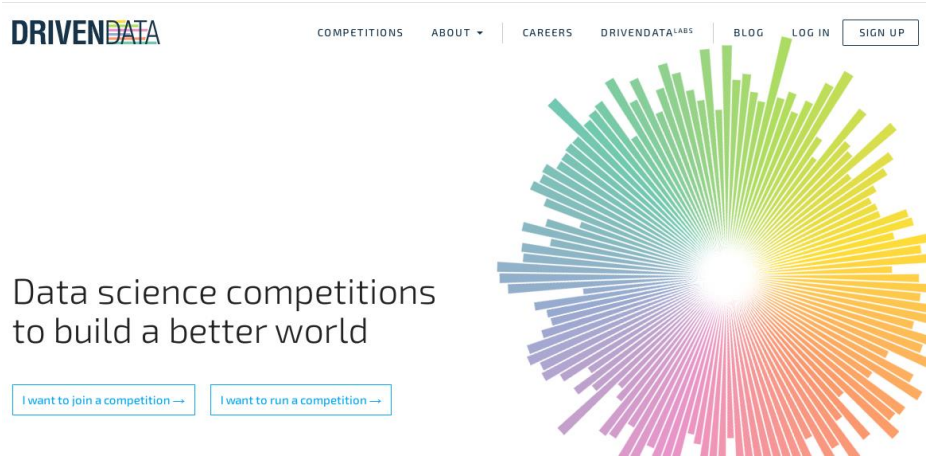


Рис.1.3. Платформа DrivenData [4]

1.4. Визначення великих даних

Експонентне зростання даних створило нову сферу інтересів у галузі технологій та бізнесу під назвою "Big Data". Загалом, набір даних або бізнес-проблема належать до класифікації Big Data, коли її дані настільки великі або складні, що їх стає неможливим зберігати, обробляти та аналізувати, використовуючи традиційні підходи до зберігання та аналізу даних.

Скільки даних потрібно, щоб стати Big Data? Чи достатньо 100 терабайт або 1000 петабайт? Обсяг є лише одним із критеріїв, оскільки потреба в обробці даних у режимі реального часу (також це називають даними в русі) або потреба в інтеграції структурованих і неструктурованих даних може кваліфікувати проблему як велику проблему даних. Наприклад, Міжнародна корпорація даних (International Data Corporation, IDC) використовує 100 терабайт як розмір набору даних, який визначається як Big Data. Якщо дані потокові, розмір набору даних може бути меншим, ніж 100 терабайт, але все ще вважається Big Data до тих пір, поки дані, що створюються, збільшуються на понад 60% на рік. Згідно National Institute of Standards and Technology (NIST): "Парадигма великих даних

складається з розподілу систем даних по горизонтально пов'язаних незалежних ресурсах для досягнення масштабованості, необхідної для ефективної обробки великих наборів даних".

Щоб вирізнити дані від великих даних, використовуються чотири Vs:

- **Об'єм (Volume)** - кількість даних, що передаються та зберігаються. Поточним завданням є пошук способів найбільш ефективно обробити зростаючий обсяг даних.
- **Швидкість (Velocity)** - швидкість, з якою формуються дані. Наприклад, дані, згенеровані мільярдом акцій, проданих на Нью-Йоркській фондовій біржі, не можуть бути просто збережені для подальшого аналізу.
- **Різноманітність (Variety)** - тип даних, який рідко знаходиться в стані, який ідеально готовий до обробки та аналізу. Велика частка Big Data – неструктуровані дані, які, за оцінками, становлять від 70 до 90% світових даних.
- **Достовірність (Veracity)** - процес запобігання неточного опису наборів даних. Наприклад, люди можуть створювати онлайн-акаунт та використовувати неправдиву контактну інформацію. Підвищена правдивість у зборі даних зменшує необхідну кількість очищення даних.

Хоча тут перераховано чотири V, більшість дискусій, інструментів та документів стосуються лише перших трьох (об'єм, швидкість, різноманітність).

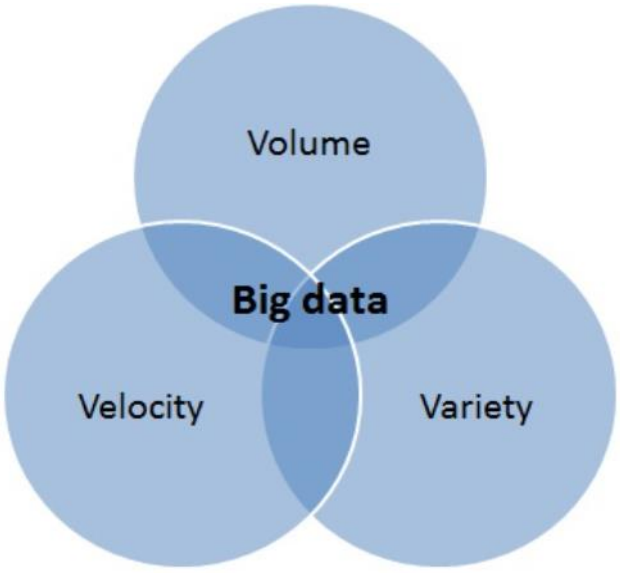


Рис.1.4. Характеристики Big Data [5]

1.5. Приклади великих даних у реальному світі

Розглянемо кілька прикладів у реальному світі генераторів великих даних. Airbus A380 Engine генерує 1 петабайт даних під час рейсу з Лондона в Сінгапур. Великий адронний колайдер (Large Hadron Collider, LHC) генерує 1 гігабайт даних щосекунди. Квадратний кілометровий масив (Square Kilometre Array, SKA) є найбільшим радіотелескопом у світі. Він генерує 20 екзабайтів даних на день. Це еквівалентно 20 мільярдам гігабайт на день [6].

Інтернет речей використовує датчики для створення даних. Дані можуть надходити від датчиків температури та вологості, які є у сільському господарстві. Датчики зараз є у всьому, від смартфонів до автомобілів та реактивних двигунів до побутової техніки. Список речей з датчиками зростає з кожним роком, це також сприяє експоненційному зростанню Big Data.

1.6. Відкриті дані

З підвищенням важливості даних для бізнесу та людей виникає багато питань щодо конфіденційності та наявності великих сховищ публічних та приватних даних. Для аналітиків важливо розуміти континуум між відкритими та приватними даними. Прийняття рішень про те, як будуть використовуватися різні типи даних в організації, так само важливі, як і знання про те, як реалізувати розподілене зберігання та оброблення Big Data.

Фонд "Open Knowledge Foundation" [7] визначає відкриті знання як "будь-який вміст, інформацію або дані, які люди можуть використовувати і перерозподіляти без будь-яких юридичних, технологічних чи соціальних обмежень". Відкриті дані є складовими відкритих знань. Відкриті знання – це коли відкриті дані стають корисними та використовуються .

Цінність відкритих даних можна відразу побачити, переглянувши такі сайти, як Портал відкритих даних Нью-Йорка, відкриті дані NYC, де відвідувач може швидко знайти рейтинги ресторанів на основі щорічних перевірок Міністерства охорони здоров'я та психічної гігієни. Портал є посередництвом із понад 1300

наборів даних від міських агентств для сприяння прозорості уряду та громадської активності. Набір даних – це сукупність пов’язаних дискретних записів, які можуть бути доступні для управління окремо або як ціле об'єднання.

Gapminder – некомерційне підприємство, що сприяє сталому глобальному розвитку [8]. На сайті представлена аналітика відкритих наборів даних із уточненням статистичних даних на такі теми, як:

- здоров'я та багатство націй;
- викиди CO_2 з 1820 року;
- дитяча смертність;
- ВІЛ-інфекція.

Портал відкритих даних України [9] містить набори даних за групами Будівництво, Держава, Екологія, Економіка та бізнес, Земля, Молодь і спорт, Освіта і культура, Охорона здоров'я, Податки, Сільське господарство, Соціальний захист, Стандарти, Транспорт, Фінанси, Юстиція у таких форматах, як csv, xls, JSON.

1.7. Приватність даних

По мірі розроблення нових програмних додатків від кінцевого користувача вимагається все більше даних, щоб дати компаніям та рекламодавцям більше інформації для прийняття бізнес-рішень. Використовуючи SafeAnswers, openPDS надає лише відповіді на конкретні запити, а необроблені дані не надсилаються. Розрахунок для відповіді здійснюється в сховищі персональних даних користувача (personal data store, PDS): "Тільки відповіді, узагальнені дані, необхідні додатку, залишають межі PDS користувача (наприклад, експорт GPS даних для додатка, щоб дізнатися, чи користувач активний або, дізнатися про загальну географічну зону, обчислення може бути зроблено в PDS користувача відповідної Q & A модуля". Конфіденційність даних користувачів спеціалісти почали обговорювати в 90-х роках XX століття, сьогодні просувається думка про те, що майбутнє конфіденційності не може бути забезпечене виключно

дотриманням законодавства та нормативно-правової бази; швидше, забезпечення конфіденційності має стати режимом роботи організації за замовчуванням.

1.8. Структуровані та неструктуровані дані

Раніше ми класифікували дані як відкриті або приватні з точки зору їх доступності. Дані також можна класифікувати за способом їх упорядкування, як структуровані або неструктуровані.

Структуровані дані вводяться та підтримуються у фіксованих полях у файлі чи записі. Структуровані дані ми можемо легко вводити, класифікувати, робити запити та аналізувати комп'ютером. Сюди входять дані, знайдені у реляційних базах даних та електронних таблицях.

Якщо набір даних досить малий, для структурованих даних використовують структуризовану мову запитів (Structured Query Language, SQL), мови програмування, створеної для запиту даних у реляційних базах даних. SQL працює лише на структурованих наборах даних. Однак для Big Data структуровані дані можуть бути частиною набору даних, але інструменти Big Data не залежать від цієї структури. Big Data переважно має набори даних, які складаються з неструктурованих даних.

Рис. 1.5. Структуровані дані [2]

Неструктуровані дані – це необроблені дані, вони не впорядковані заздалегідь і не мають фіксованої схеми, яка б ідентифікувала тип даних. У неструктурованих даних бракує встановленого способу введення або групування даних та їх аналізу. Приклади неструктурованих даних включають вміст фотографій, аудіо,

відео, веб-сторінок, блогів, книг, журналів, дописів, презентацій PowerPoint, статей, електронної пошти, вікі-файлів, текстових текстових документів та тексту в цілому. Прикладом неструктурованих даних є PDF-версія книги. Текст можна шукати, але він не організований у заздалегідь визначеній формі, наприклад, за допомогою полів та записів. Як структуровані, так і неструктуровані дані є цінними для людей, організацій, галузей та урядів. Організаціям важливо прийняти всі форми даних та визначити способи їх форматування, щоб ними можна було керувати та аналізувати.

1.9. Хмарні та туманні обчислення

У минулому набори даних були в основному статичними, розташовувались на одному сервері або в колекції серверів усередині організації та оброблялися з використанням мови програмування бази даних, такої як SQL. Хоча ця модель усе ще існує, зберігання великих наборів даних перемістилося в центри обробки даних (ЦОД). Сьогодні, із зростанням хмарних обчислень, ролі Big Data та необхідністю аналізу даних у режимі реального часу, дані продовжують зберігатись у ЦОД. Дані також повинні бути доступними для аналізу ближче до місця їх створення, і знання, отримані на основі цих даних, можуть мати найбільший вплив. Такий спосіб оброблення даних називається туманним обчисленням (Fog computing).

Туман – це хмара, розташована близько до джерела генерації даних. Туманні обчислення не є заміною хмарних обчислень, скоріше, туманні обчислення дозволяють розробляти нові інструменти. У моделі обчислень туману існує взаємозв'язок між хмарою та туманом, особливо коли мова йде про управління даними та аналітику. Туманні обчислення забезпечують обчислення, зберігання та мережеві послуги між кінцевими пристроями та традиційними ЦОД. Туманні обчислення виробляють величезну кількість даних від різних датчиків і контролерів. При роботі з даними в Інтернеті необхідно враховувати три важливі фактори:

- **Енергія або акумулятор** – кількість енергії, яка використовується датчиком IoT та залежить від швидкості вибірки датчика. Діапазон між пристроями також може впливати на кількість енергії, яка повинна бути використана для передачі даних сенсорів контролерам. Чим далі датчик, тим більше енергії потрібно використовувати для передачі даних.
- **Ширина смуги пропускання** – коли багато датчиків передають дані, може виникнути затримка зв'язку, якщо недостатньо пропускної здатності для підтримки усіх пристроїв. Додатковий аналіз в тумані може допомогти зменшити деякі вимоги до смуги зв'язку.
- **Затримка** – на аналіз даних у режимі реального часу впливає занадто велика затримка в мережі. Дуже важливо, щоб виконувались лише необхідні комунікації з хмарою, і обчислення відбувалося якомога ближче до джерела даних.

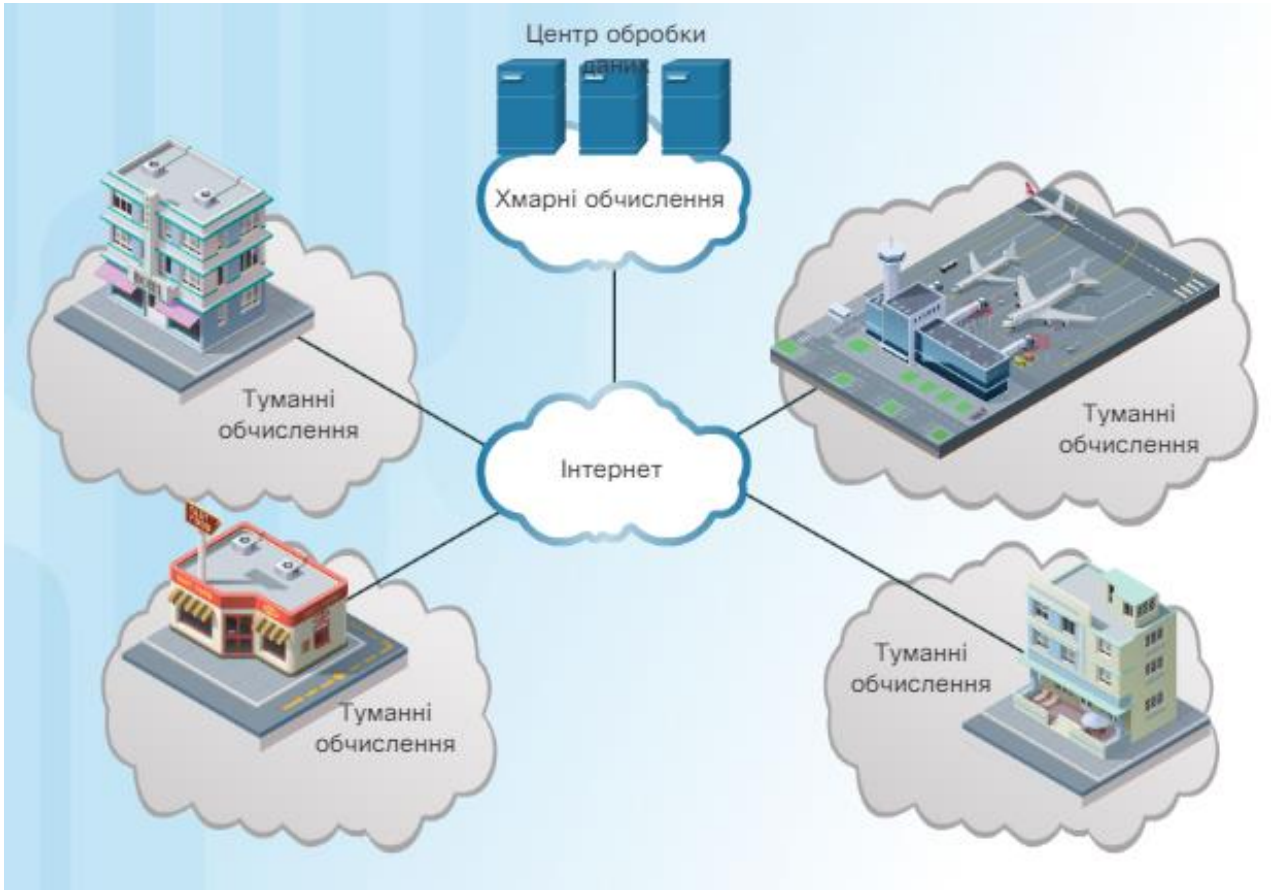


Рис. 1.6. Модель туманних обчислень [2]

1.10. Дані в спокої та дані в русі

Дані в спокої – це статичні дані, які зберігаються у фізичному місці, наприклад, на жорсткому диску на сервері чи в ЦОД. Дані зберігаються в базі даних, а потім аналізуються та інтерпретуються. Ті, хто приймає рішення, отримують повідомлення про те, чи потрібно діяти.

Дані, що перебувають у русі – це динамічні дані, які потребують обробки в режимі реального часу, перш ніж ці дані стають неактуальними або застарілими. Це безперервна взаємодія між людьми, процасами та речами. Пристрої на краю мережі працюють разом, щоб негайно діяти на знаннях, отриманих завдяки динамічному аналізу даних. Порядок аналізу, дії та повідомлення може бути різним. Важлива відмінність даних у стані спокою і даних у русі полягає в тому, що при даних, що перебувають у русі, дія даних на них відбувається до зберігання даних.

Дані, що перебувають у русі, використовуються різними галузями, які покладаються на отримання значень із даних перед їх збереженням. Датчики в полі фермера постійно надсилають дані про температуру, вологість ґрунту та сонячне світло до місцевого контролера, який аналізує дані. Якщо умови не правильні, контролер діє негайно, надсилаючи сигнали виконавчим механізмам у полі, щоб розпочати полив. Потім контролер надсилає повідомлення власнику поля про початок поливу та надсилає дані для зберігання.

Через особливості Big Data неможливо дублювати та зберігати всі ці дані у централізованому сховищі даних. Нові реалізації пристроїв включають велику кількість датчиків, що фіксують та обробляють дані. Рішення та дії потрібно проводити на межі, де і коли створюються дані. Оскільки датчики набирають більше процесорної потужності та стають більш усвідомленими у контексті, тепер можна наблизити інтелектуальні та аналітичні алгоритми до джерела даних. У цьому випадку дані, що знаходяться в русі, залишаються там, де вони створені, і представляють уявлення в реальному часі, спонукаючи до кращих, швидших рішень.

1.11. Інфраструктура великих даних

Багато компаній розуміють, що є сенс інвестувати в технології Big Data, щоб залишатися конкурентоспроможними на своєму ринку. На даний час їх інфраструктура даних може виглядати приблизно так, як рис.1.7, із серверами баз даних та традиційними засобами обробки даних. Зазвичай доступ до даних обмежений кількома відповідальними особами в організації. Компанії швидко рухаються до використання технологій Big Data для керування бізнес-аналітикою. За даними National Institute of Standards and Technology (NIST), парадигма Big Data складається з розподілу систем даних по горизонтально пов'язаних незалежних ресурсах для досягнення масштабованості, необхідної для ефективної обробки великих наборів даних. Це горизонтальна масштабованість. Він відрізняється від вертикальної масштабованості тим, що не намагається додати більше процесорної потужності та пам’яті існуючим машинам. Ці інфраструктури дозволяють багатьом користувачам одночасно безперешкодно та безпечно отримувати доступ до даних. Одним із таких прикладів є тисячі інтернет-покупців або мобільних геймерів.

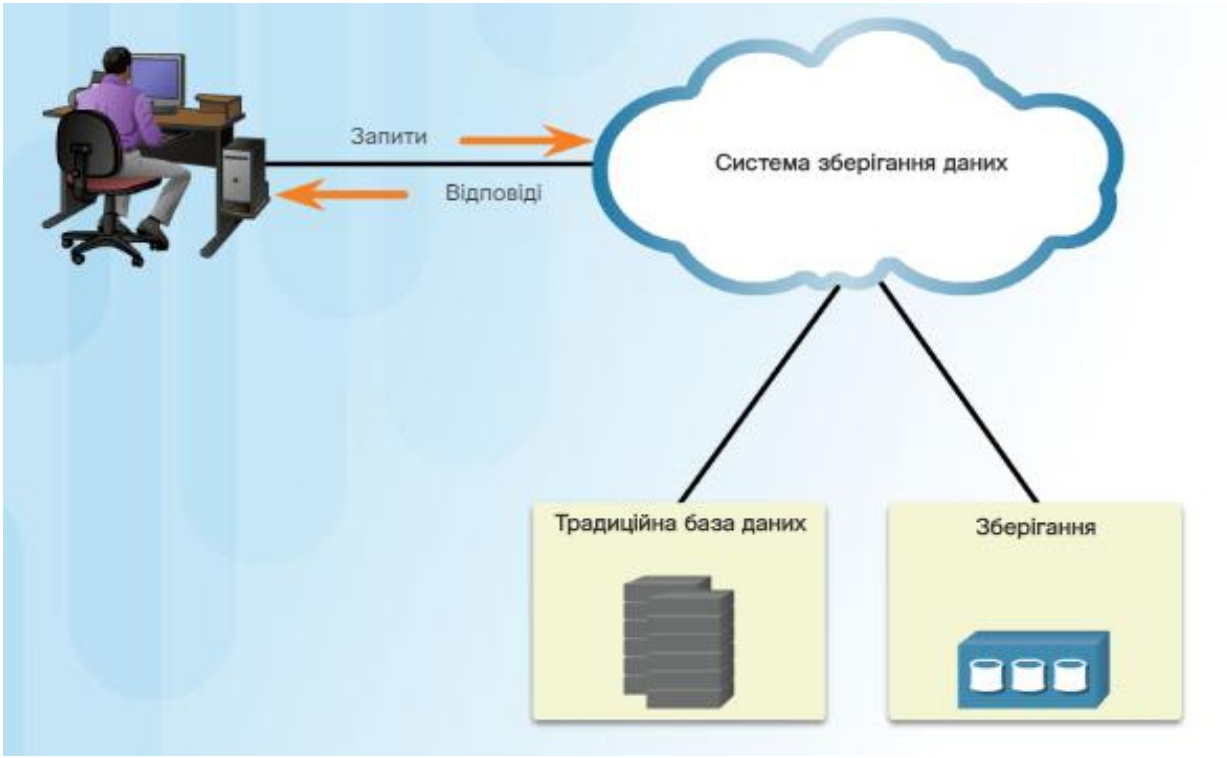


Рис. 1.7. Традиційна система управління базами даних [2]

На рис. 1.8 представлено пристрої в інфраструктурі великих даних організації, де інфраструктура бізнесу потребує прийняття рішень на місці, використовуючи хмарні обчислення.

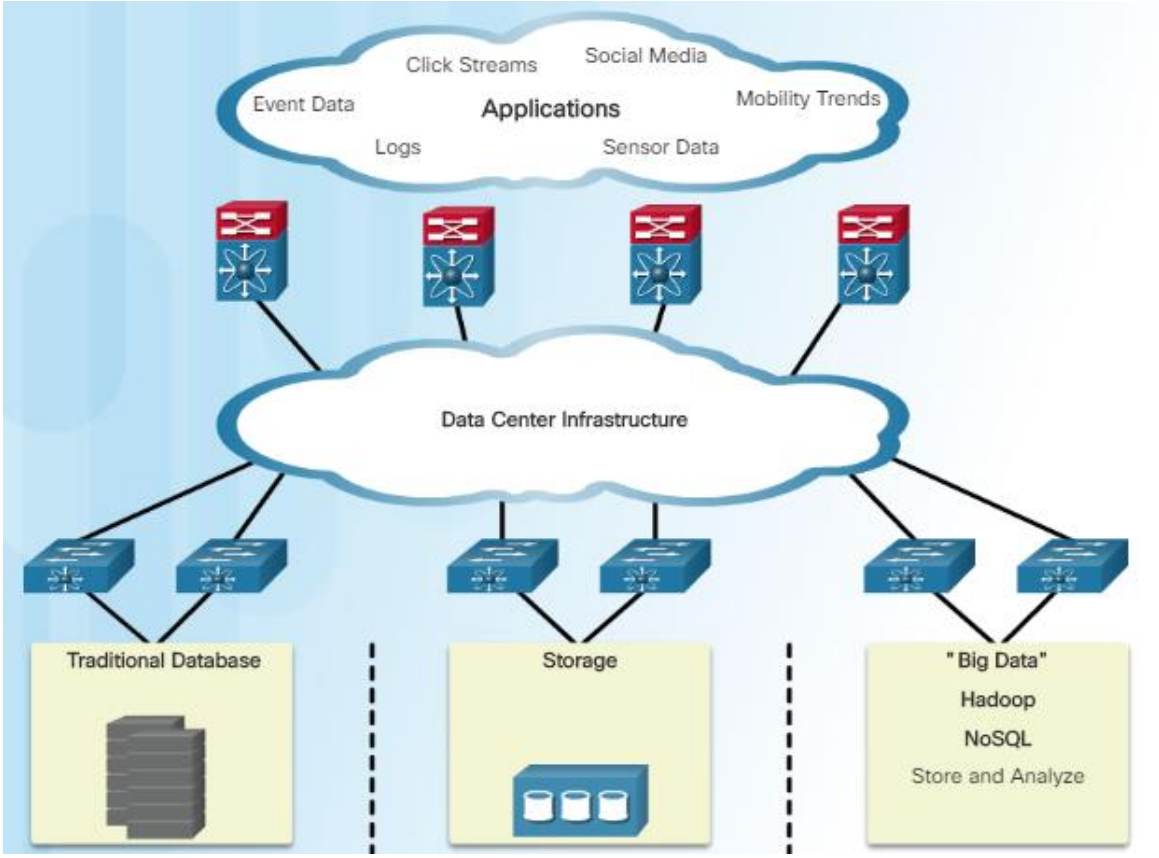


Рис. 1.8. Інфраструктура великих даних [2]

1.12. Розподілені дані та їх обробка

Наступне покоління управління даними з'явилося за допомогою системи управління реляційними базами даних (Relational database management system, RDBMS). Протягом 30 років це був стандартний підхід до управління даними. Реляційні бази даних фіксують зв’язки між різними наборами даних, створюючи більш корисну інформацію.

Більшість комерційних рішень RDBMS використовують SQL в якості мови запитів до сьогодні. Прикладом запиту SQL є: SELECT id, ім'я, ціна інвентаря, де ціна, наприклад, є <20. Приклади продуктів, які використовують структуровану мову запитів для доступу до даних, включають MySQL, SQLite, MS SQL, Oracle та IBM DB2.

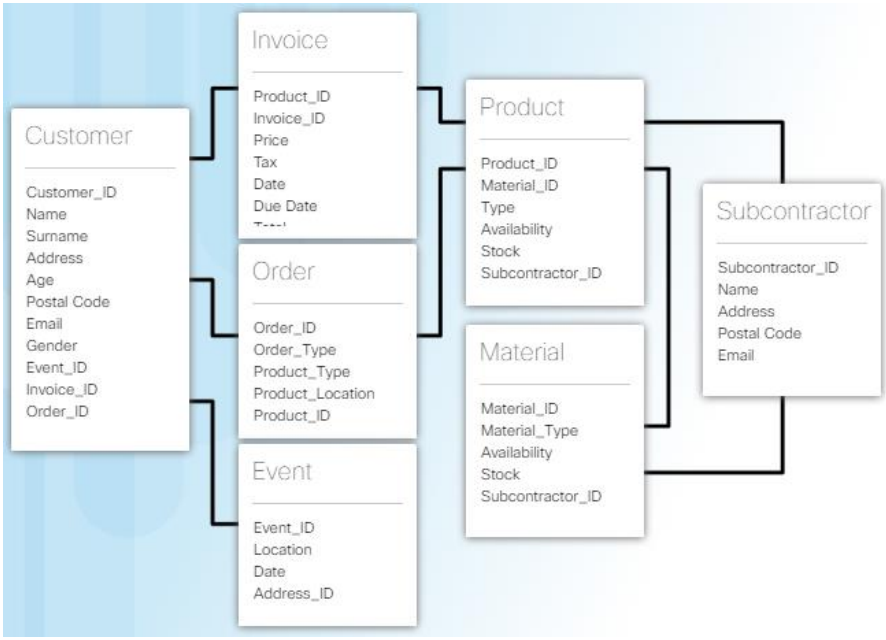


Рис. 1.9. Приклад реляційної бази даних [2]

Ще одна характеристика реляційних баз даних – це відмінність між базою даних та системою управління, що використовується для запиту бази даних. Зазвичай із RDMS та базовою базою даних багато користувачів можуть одночасно запитувати реляційну базу даних. Користувач зазвичай не знає всіх відносин, що існують усередині бази даних, адже, скоріше за все, користувач резюмує представлення бази даних, яка відповідає його потребам.

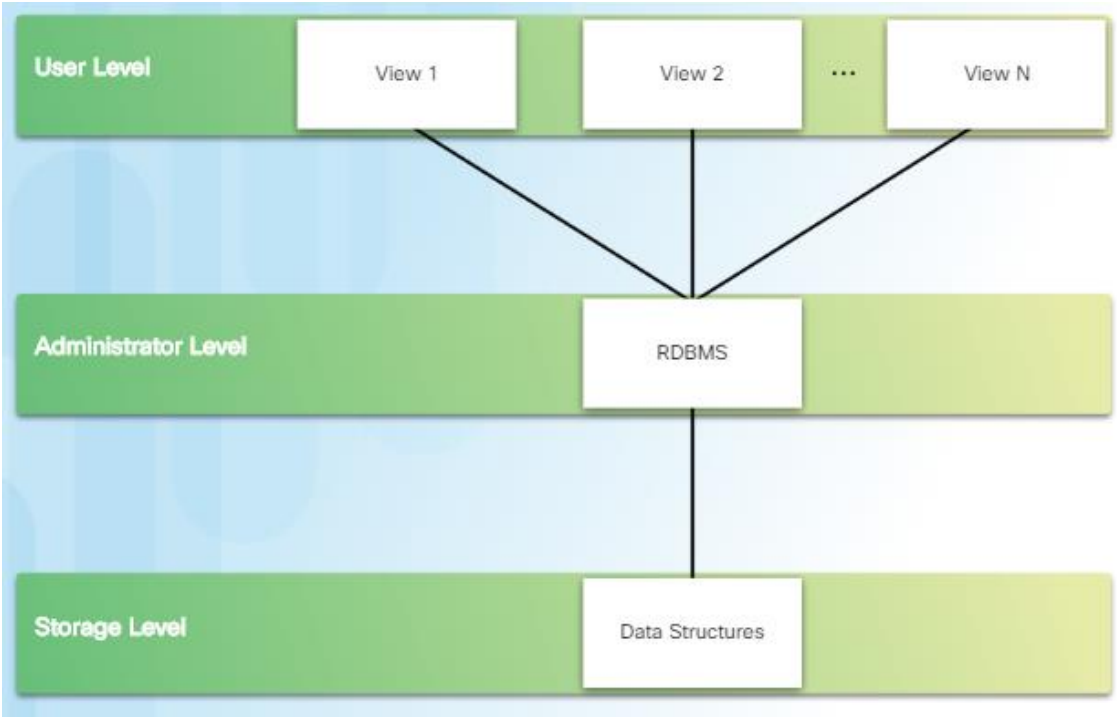


Рис. 1.10. Абстрагування даних у реляційній базі даних [2]

Найнижчий рівень абстрагування описує, як фізично зберігаються дані. Наступний рівень описує, які дані зберігаються та зв’язки між даними. Це рівень, на якому працює адміністратор бази даних. Користувацький рівень – це найвищий рівень, який описує, до якої частини бази даних певний користувач або група користувачів можуть отримати доступ. У будь-який момент часу може бути визначено багато різних одночасних підключень до бази даних. Нереляційні бази даних SQL (NoSQL) дуже добре масштабуються з розподіленими базами даних, оскільки NoSQL може обробляти великі дані та веб-додатки в режимі реального часу краще, ніж RDBMS, запити до баз даних NoSQL зосереджені збору документів, таких як інформація, зібрана з веб-сайтів. NoSQL також дозволяє кластерам машин обробляти дані та забезпечувати кращий контроль над їх наявністю. Бази даних NoSQL широко використовуються для вирішення бізнес-проблем.

З точки зору управління даними, аналітика була простою, коли дані створювали лише люди. Обсяг даних був керований. Реляційні бази даних обслуговують потреби аналітиків даних. Однак, з поширеністю систем автоматизації бізнесу та вибуховим зростанням веб-додатків та даних, що генеруються, аналітиці стає важче керувати лише рішенням RDBMS. Приблизно 90% даних, які існують сьогодні, були зібрані лише за останні два роки. Цей збільшений обсяг за короткий проміжок часу є властивістю експоненціального зростання. Цей великий обсяг даних важко обробити та проаналізувати. Замість того, щоб великі бази даних оброблялися великими та потужними комп'ютерами мейнфрейму та зберігалися в гігантських дискових масивах (вертикальне масштабування), розподілена обробка даних приймає великий об'єм даних і розбиває їх на менші частини. Ці менші обсяги даних поширюються у багатьох місцях, які обробляються багатьма комп'ютерами з меншими процесорами. Кожен комп'ютер у розподіленій архітектурі аналізує свою частину зображення Big Data (горизонтальне масштабування).

Більшість розподілених файлових систем розроблені таким чином, щоб вони не були помітні для клієнтських програм. Розподілена файлова система знаходить

файли та переміщує дані, але користувачі не можуть знати, що файли розподіляються між багатьма різними серверами чи вузлами. Користувачі отримують доступ до цих файлів так, ніби вони є локальними для власних комп'ютерів. Усі користувачі бачать однаковий вигляд файлової системи та отримують доступ до даних одночасно з іншими користувачами.

Hadoop був створений для вирішення цих великих обсягів даних. Проект Hadoop розпочався з двох граней: розподілена файлова система Hadoop (Hadoop Distributed File System, HDFS) – це розподілена файлова система і MapReduce, який є розподіленим способом обробки даних. Hadoop перетворився на всеосяжну екосистему програмного забезпечення для управління великими даними. Є багато інших програм з розподіленою файловою системою (DFS). Ось лише декілька з них: Ceph , GlusterFS та файлова система Google.

База даних NoSQL зберігає та отримує доступ до даних інакше, ніж реляційні бази даних. NoSQL іноді називають "не тільки SQL", "не SQL" або "нереляційними". Системи NoSQL можуть підтримувати SQL-подібні мови запитів. Бази даних NoSQL використовують структури даних, такі як ключ-значення, широкий стовпчик, графік або документ. Багато баз даних NoSQL надають "можливу послідовність". З можливою послідовністю зміни бази даних з часом з’являються у всіх вузлах. Це означає, що запити для даних можуть не надавати останню доступну інформацію. Причиною створення NoSQL було спрощення дизайну баз даних. Легше масштабувати кластери вузлів за допомогою NoSQL, ніж це виконувати у стандартних реляційних базах даних. Найпопулярнішими базами даних NoSQL у 2015 році були MongoDB, Apache Cassandra та Redis.

Структурована мова запитів (SQL) призначена для управління, пошуку та обробки даних, включаючи Big Data. SQLite – це бібліотека, яка використовує автономний, транзакційний двигун бази даних SQL. Код для SQLite знаходиться у відкритому доступі, що означає, що він може вільно використовуватись у комерційних та приватних цілях. SQLite – це найбільш широко розгорнута база даних у світі. SQLite також є вбудованою системою баз даних SQL. На відміну від

більшості інших баз даних SQL, у SQLite немає окремого серверного процесу. SQLite читає і записує безпосередньо у звичайні файли диска.

SQLite – популярний вибір для двигуна баз даних у мобільних телефонах, MP3-програвачах, приставках та інших електронних гаджетах. SQLite має невеликий слід коду, дозволяє ефективно використовувати пам’ять, дисковий простір та пропускну здатність диска, відрізняється високою надійністю і не потребує обслуговування адміністратора бази даних. SQLite часто використовується замість RDBMS для тестування. SQLite не потребує налаштувань, що значно спрощує тестування.

Розглянемо кілька корисних функцій SQLite.

- Ніяких налаштувань чи адміністрування не потрібно. Він має простий у користуванні API.
- Повна база даних зберігається в одному файлі міжплатформних дисків. Може використовуватися як формат файлу програми.
- Має невеликий слід коду.
- Це крос-платформна SQL. Підтримує Android, iOS, Linux, Mac, Windows та кілька інших операційних систем.
- Джерела для SQLite знаходяться у відкритому доступі.
- Має окремий інтерфейс командного рядка (CLI).
- Усі зміни в межах однієї транзакції відбуваються повністю або зовсім не відбуваються. Це справедливо навіть у випадку збою програми чи операційної системи або відмови живлення.

Використання технологій SQL та баз даних є ефективним для отримання підмножини даних із наявного набору даних, що зберігається в базі даних. Вираз SQL, який виконує цю дію, називається SQL-запитом. У бізнесі багато важливих проблем не вдається вирішити за допомогою простого запиту SQL і потрібен більш складний аналітичний процес. Ось тут використовується більш потужна мова програмування для аналізу даних, така як R або Python. R і Python мають великі спільноти розробників. Їх користувачі відомі тим, що розробляють модулі

аналізу даних та безкоштовно надають їх спільноті. Через це будь-який користувач може завантажувати та використовувати попередньо запрограмовані модулі та інструменти. Можливість створювати інструменти аналізу даних з нуля дозволяють отримати спеціально налаштовані програми. Процес створення інструменту аналізу даних з нуля можна розділити на дві основні частини: модель та код.

Моделювання полягає у визначенні того, що робити з даними для досягнення бажаних результатів та висновків. Припустимо, потрібно створити персональний фітнес-трекер та не існує попередньо запрограмованого модуля, який би виконував саме те, що ми хочемо зробити. Модуль трекера містить акселерометр, який є датчиком, здатним вимірювати прискорення пристрою. Акселерометр можна використовувати для визначення швидкості та напрямку руху. Швидкість і напрямок руху пристрою завжди відповідають швидкості та напрямку його користувача, коли він прикріплений до користувача.

Але що робити, якщо прилад кріпиться до ваги гантелей чи тенісної ракетки? Пристрій все одно буде отримувати однакові дані, швидкість і напрямок руху, але через різні програми інтерпретацію цих даних слід адаптувати до нового використання. У цьому контексті моделювання можна розглядати як спосіб інтерпретації та обробки даних. Наприклад, якщо фітнес-трекер прикріплений до користувача, дві поспіль точки без руху (швидкість дорівнює нулю), ймовірно, представляють початок і кінець спринту. Якщо вони прикріплені до ваги гантелей, ймовірно, дані представляють момент, коли гантель був піднятий з підлоги та найвищою точкою, яку користувач зміг підняти перед тим, як повернути його на підлогу.

Код є другою частиною створення інструментів аналізу даних з нуля. Код – це програма, яка обробляє дані і повинна бути записана відповідно до створеної моделі. Хоча модель і код є двома окремими об'єктами, вони пов'язані, оскільки код побудований на основі моделі.

Висновок до лекції 1

Даними можуть бути слова в книзі, вміст електронної таблиці, фотографії, файли або потоки вимірювань, надіслані пристроєм. Чотири Vs великих даних – це об'єм, швидкість, різноманітність та достовірність. Структуровані дані – це дані, введені у фіксовані поля файлу бази даних або записі. Неструктуровані дані не мають фіксованої схеми, яка визначає тип даних. Дані в стані спокою – це статичні дані, що зберігаються у фізичному місці. Дані в русі аналізують та витягують значення з даних до їх збереження. Hadoop був створений для роботи з великими обсягами даних. База даних NoSQL зберігає та отримує доступ до даних інакше, ніж реляційна база даних.

Питання для закріплення

1. Який вплив Інтернету Речей та зростання даних?
2. Для чого використовується платформа Kaggle?
3. Яке визначення великих даних?
4. Наведіть приклади великих даних у реальному світі.
5. Які дані є відкритими?
6. Які дані є структурованими та неструктурованими?
7. Що таке хмарні та туманні обчислення?
8. Опишіть дані в спокої та дані в русі.
9. Якою є інфраструктура великих даних?
10. Яка роль Hadoop в обробці розподілених даних?

Список рекомендованої літератури

1. Byte Size Infographic: Visualising data // Електронний ресурс. Режим доступу: https://www.redcentricplc.com/resources/infographics/byte-size/
2. IoT Fundamentals: Big Data & Analytics // Електронний ресурс. Режим доступу: https://www.netacad.com/courses/iot/big-data-analytics
3. Kaggle// Електронний ресурс. Режим доступу: https://www.kaggle.com/
4. DrivenData // Електронний ресурс. Режим доступу: https://www.drivendata.org/
5. Big Data: the 3 VS explained // Електронний ресурс. Режим доступу: https://bigdataldn.com/intelligence/big-data-the-3-vs-explained/
6. Computing // Електронний ресурс. Режим доступу: https://home.cern/science/computing
7. Open Knowledge Foundation // Електронний ресурс. Режим доступу: https://okfn.org
8. Gapminder // Електронний ресурс. Режим доступу: https://www.gapminder.org
9. Портал відкритих даних України // Електронний ресурс. Режим доступу: https://data.gov.ua