МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна
Факультет математики і інформатики
Кафедра теоретичної та прикладної інформатики

Кваліфікаційна робота
бакалавр

на тему «Розроблення інтегрованої програмної системи аналітики для Інтернет-маркетингу: проектування архітектури та конвеєру обробки даних»

Виконав: студент 4 курсу, групи МФ-41, спеціальність 122 – Комп'ютерні науки, освітньо-професійна програма «Теоретична та прикладна інформатика» Лазаренко Олександр Володимирович

Керівник: Меняйлов Є.С., к.т.н., доцент кафедри Теоретичної та прикладної інформатики факультету математики і інформатики Харківського національного університету імені В.Н. Каразіна

Рецензент: ……
……
……
……
……

Харків – 2024

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Факультет математики і інформатики
Кафедра теоретичної та прикладної інформатики
Рівень вищої освіти перший (бакалаврський)
Спеціальності 122 «Комп'ютерні науки»
Освітньо-професійна програма «Теоретична та прикладна інформатика»

ЗАТВЕРДЖУЮ

В.о. зав. кафедри теоретичної
і прикладної інформатики
Меняйлов Є. С.

_____ _____
підпис ініціали, прізвище

"___" _____2023 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Лазаренка Олександра Володимировича

(прізвище, ім'я, по батькові студента)

1. Тема роботи «Розроблення інтегрованої програмної системи аналітики для Інтернет-маркетингу: проектування архітектури та конвеєру обробки даних» керівник роботи _Меняйлов Євген Сергійович, кандидат технічних наук, доцент_____
затверджені наказом по університету від "___"_____20___року №___

2. Строк подання студентом роботи_____

3. Перелік питань, які потрібно розробити:

Теоретична частина. Дослідити теоретичні основи функціонування програмних аналітичних систем в галузі інтернет-маркетингу; розкрити їх задачі та структуру їх об'єктів і бізнес-процесів; класифікувати наявні на ринку програмні засоби маркетингової аналітики.

Аналітична частина. Класифікувати наявні на ринку програмні засоби маркетингової аналітики; провести аналіз провідних систем та визначити їх порівняльні переваги й недоліки; сформувати перелік стандартних функціональних вимог до сучасних систем маркетингової аналітики в

контексті їх розроблення і застосування.

Проектно-рекомендаційна частина. Провести проектування логічної моделі бази даних аналітичної програмної системи; розробити складові конвеєру збору, збереження й обробки вхідної інформації та інструменти їх оркестрації; обґрунтувати добір архітектурних та технологічних засад реалізації єдиної платформи аналітики; виробити рекомендації з підвищення продуктивності роботи системи в умовах реальної бізнес-практики.

1. План роботи

№ з/п	Назви етапів роботи
1	Ознайомлення з літературою та ПК-аналогами. Складання плану.
2	Робота над розділом 1 та висновками до нього.
3	Підбір інформації
4	Робота над розділом 2 та висновками до нього.
5	Робота над розділом 3 та висновками до нього.
6	Написання вступу та висновків до роботи.
7	Оформлення переліку використаних джерел згідно стандартів.
8	Проходження перевірки роботи на запозичення.
9	Отримання відгуку від керівника
10	Рецензування роботи
11	Складання анотацій
12	Підготовка презентації та ілюстративного матеріалу до захисту роботи

2. Дата видачі завдання_____

Студент _____ ___О. В. Лазаренко_____

підпис ініціали, прізвище

Керівник роботи _____ _____Є. С. Меняйлов_____

підпис ініціали, прізвище

ЗМІСТ

ВСТУП

Актуальність дослідження. У нинішньому світі, де найвідчутнішими тенденціями розвитку є цифровізація суспільних процесів та посилення економічної конкуренції, основний акцент збутової діяльності виробників всіх видів продукції дедалі більше зміщується убік електронного (інтернет)-маркетингу. Саме цей домен відзначається найбільшими темпами зростання, оскільки у ньому постійно знаходяться нові інноваційні механізми продажів, які забезпечують істотно ширші можливості підприємств різних галузей підвищувати фінансові результати, ніж у сфері традиційних «офлайн» продажів. Проте зворотним боком такої динаміки є посилення необхідності вироблення всебічно вивірених маркетингових стратегій, які мають враховувати безліч різноспрямованих факторів, а отже, базуватись на обліку й прогнозуванні колосальних обсягів даних. Це висуває на перший план задачу автоматизації процесів підготовки й прийняття управлінських рішень, яка передусім вирішується шляхом впровадження *систем маркетингової аналітики.* Відтак останні перетворюються на ключовий інструмент забезпечення конкурентоспроможності на будь-яких галузевих ринках [8].

Як добре відомо, програмні аналітичні комплекси економічної інформації є сьогодні однією з найбільш прибуткових форм продуктів для ІТ-компаній ([19]) що демонструється активною експансією на їх ринок з боку багатьох провідних виробників ПЗ (зокрема, корпорацій «Google», «Microsoft», «Adobe» та багатьох інших). Тому широка тематика оптимізації процесів розроблення та адаптації програмних систем маркетингової аналітики до різних умов їх застосування є вельми актуальною як для теоретичної Інформатики, так і для практики роботи сучасних ІТ-компаній. А первісні передумови такої оптимізації формуються під час проектування загальної архітектури побудови таких систем та закладених у них конвеєрів обробки даних, яке виступає головною метою даної роботи.

Метою роботи є розроблення архітектурних засад та програмної

реалізації прототипу інтегрованого ПК аналітичного забезпечення збутових і фінансових операцій ІТ-підприємства, на основі систематизації недоліків наявних на ринку систем маркетингової аналітики та виявлення загальних базових вимог до таких систем.

Поставлена мета обумовила необхідність вирішення наступних **задач:**

- описати основні інструменти збутової моделі галузі Інтернет-маркетингу;
- оцінити функціональну роль маркетингової аналітики в домені Інтернет-маркетингу та розкрити структуру її ключових бізнес-процесів;
- виокремити спектр ключових показників та метрик – об'єктів систем маркетингової аналітики;
- здійснити класифікацію провідних програмних комплексів, наявних на ринку маркетингової аналітики та визначити їх переваги і недоліки;
- провести проектування логічної моделі бази даних програмної системи, необхідної для її застосування в аналітичних процесах;
- розробити інструменти забезпечення безперебійності процесів збору, збереження й обробки вхідної інформації шляхом їх конвеєризації;
- автоматизувати виконання конвеєризованих бізнес-процесів шляхом їх інтеграції у єдину платформу оркестрації;
- здійснити вибір архітектурних та технологічних основ реалізації інтегрованої системи маркетингової аналітики;
- провести економетричне дослідження адекватності результатів роботи системи вимогам оптимізації маркетингових бюджетів замовників.

Предметом дослідження є програмні комплекси автоматизації здійснення аналітичних процесів у галузі сучасного Інтернет-маркетингу.

Об'єктом дослідження є елементи архітектури та конвеєру обробки даних прототипу інтегрованої системи маркетингової аналітики для ІТ-компанії, що спеціалізується на наданні послуг з вивчення іноземних мов через різноманітні веб-платформи за довгостроковою підпискою.

Методи дослідження. Дослідження спирається на загальнонаукові методи пізнання (зокрема порівняльний, емпірико-аналітичний, системний) та

спеціальні методологічні інструменти (теорія графів, аналіз функціональних і нефункціональних вимог, парадигми інтеграції та обробки даних, економетрична оптимізація), і ґрунтується на аналізі зарубіжних та вітчизняних науково-практичних джерел, вивчення яких передбачає систематизацію та узагальнення матеріалу, тематичний огляд, структурно-функціональний та логічно-послідовний опис.

Апробація результатів дослідження. Окремі положення та результати дослідження протягом 2023-2024 рр. оприлюднені у 6 публікаціях автора на електронних видавничих платформах Medium та Hashnode.

Структура і обсяг. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, п’яти додатків і списку використаних джерел. Основний зміст роботи викладено на 86 сторінках. У дослідженні є 14 рисунків та 3 таблиці. Список використаних джерел містить 55 найменувань.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ ДОМЕНУ ІНТЕРНЕТ-МАРЕТИНГУ

1.1. Базові бізнес-процеси та інструменти інтернет-маркетингу

Інтернет – (*електронний*) **маркетинг** – являє собою комплексну різноспрямовану діяльність із просування продукції до покупців (кінцевих і проміжних споживачів) через численні онлайн-сервіси [9].

Дана форма здійснення збутового процесу у сучасній світовій економіці відзначається найбільш динамічним зростанням: за численними прогнозами ринок послуг цифрового маркетингу зросте з 38 млрд. дол. у 2020 р. до майже 80 млрд. дол. у 2028 р., при середньорічному темпі зростання 17% [35].

Прискореному поширенню цієї бізнес-моделі в різноманітних доменах сприяє низка *переваг* інтернет-маркетингу перед класичним «офлайн» маркетингом [11; 21]:

1. Точність на цільову аудиторію, у порівнянні до традиційної реклами. Тут мається широкий спектр каналів взаємодії між продавцями й покупцями, а тому з'являється можливість підбирати саме той канал, який дозволить доставляти рекламу лише зацікавленій аудиторії, причому своєчасно – коли вона виявляє готовність здійснити покупку. Звідси також випливає значно більша, ніж у традиційному збуті, рентабельність інтернет-маркетингу.

2. Можливість відстежити та оцінити результати впливу реклами на клієнта, відібрати найефективніші інструменти та задіяти позитивний персоналізований досвід покупок (User Experience) для розширення кола потенційних споживачів.

3. Оптимізація збутових процесів за рахунок їх широкої автоматизації. Наприклад, утримання клієнтів можна налаштувати лише один раз, і надалі система працюватиме автономно, ефективно проводячи покупців за всією «вороною продажів» продукції.

4. Можливість відобразити повний шлях користувача за цією «воронкою

продажів», використовуючи великий інструментарій інтернет-аналітики.

Всі свої вказані переваги інтернет-маркетинг реалізує завдяки залученню величезної кількості **інструментів** комунікації зі споживачами. До основних із них можна віднести ([11. С. 51-56]):

Рисунок 1.1 – Основні інструменти домену Інтернет-маркетингу

Джерело: складено автором за даними [11].

1. Web-ресурс. Інтернет-сайт (інакше, «лендинг») або мобільний додаток продавця – це основа, навколо якої відбувається розбудова всієї стратегії інтернет-маркетингу. Такий ресурс є єдиним майданчиком для ознайомлення споживачів із пропонованим асортиментом товарів та послуг, комунікації з ними, підготовки замовлень та здійснення угод. Він формує вигідну альтернативу звичним магазинам та офісам, оскільки надає широкі можливості застосування чотирьох інших інструментів просування товару, недоступних в офлайн-режимі:

2. Контент-маркетинг. Здійснення просування продукту за допомогою контенту будь-якого типу: текстового, графічного, аудіо чи відео. Він створюється не стільки для ознайомлення потенційного клієнта з продуктом, скільки для цілеспрямованого підштовхування його до покупки. Причому, окрім веб-ресурсів, такий контент може бути розміщений і на інших інтернет-

майданчиках продавця – наприклад, у соціальних мережах, розсилках новин та ін. Підвидами цього широкого методу є популярні інструменти банерної реклами та SMM-маркетингу.

3. Контекстна реклама. Персоналізовані оголошення, сформовані на основі попередніх інтересів, дій та запитів користувача в Інтернеті. Націлена на клієнта, вже готового до покупки. Її підвидами є «переслідуюча реклама» (яка з'являється і «слідує» за користувачем після відвідування сайту продавця), а також PPC (pay per click – рекламна модель, в якій суб'єкт розміщує рекламу на сторонніх сайтах і, сплачуючи їх власникам за натискання користувачем на розміщений банер чи «тіло» документа, «купує» собі додаткових клієнтів).

4. SEO/ASO просування (пошукова оптимізація). Споживач за допомогою пошукача знаходить потрібні йому продукти і цілеспрямовано приходить на ресурс, який їх пропонує. Тут зусилля продавця націлені на забезпечення високих позицій свого сайту в пошукових системах за допомогою пошукового просування (внесення мета-тегів на сторінки сайту, включення ключових термінів у, побудова семантичного ядра та інше).

5. Email-маркетинг. Спосіб встановити постійний двосторонній зв'язок між рекламодавцем та користувачами, які відвідали його ресурс, діяти персоналізовано, а також оперативно інформувати про останні новини. Застосування даного інструменту допомагає ближче познайомитися з клієнтом, мотивувати його на повторну покупку за рахунок розсилки корисного контенту, інформації про майбутні акції та бонуси.

6. Маркет-плейси. Повна інформація про товар та умови його купівлі, а також контактні дані продавця розміщуються на онлайн торгових майданчиках, де користувачі можуть залишати відгуки, проводити пошук за обраними категоріями, порівнювати ціну та умови покупки у різних магазинах. Багато таких агрегаторів надають свої ресурси у формі Software-as-a-Service платформ. Найпопулярнішими у світі є *Amazon* та *Shopify*, а в Україні – *ek.ua*, *Hotline.ua*, *prom.ua* та інші.

У свою чергу інтернет-маркетинг, як і будь-який інший бізнес-процес, функціонує внаслідок активності різноманітних груп суб'єктів, яких можна розділити на елементи його *внутрішньої* та *зовнішньої* структури ринку. Принципова схема взаємодії між ними наведена на рисунку 1.2.

До *перших* відносять продавців (у т.ч. виробників і різного роду торгових посередників) та покупців (кінцевих та проміжних споживачів) основної продукції кожного конкретного ринку. Вони є традиційними та наймасовішими учасниками збутового процесу в кожній галузі економіки.

Специфікою ж інтернет-маркетингу є особливий акцент на ролі *другої* групи його суб'єктів, які формують його *зовнішню* (*інфра-*) структуру. До неї відносяться маркетингові консалтингові агентства, маркет-плейси, а також IT-компанії, які виробляють різноманітне програмне забезпечення для їх взаємодії.

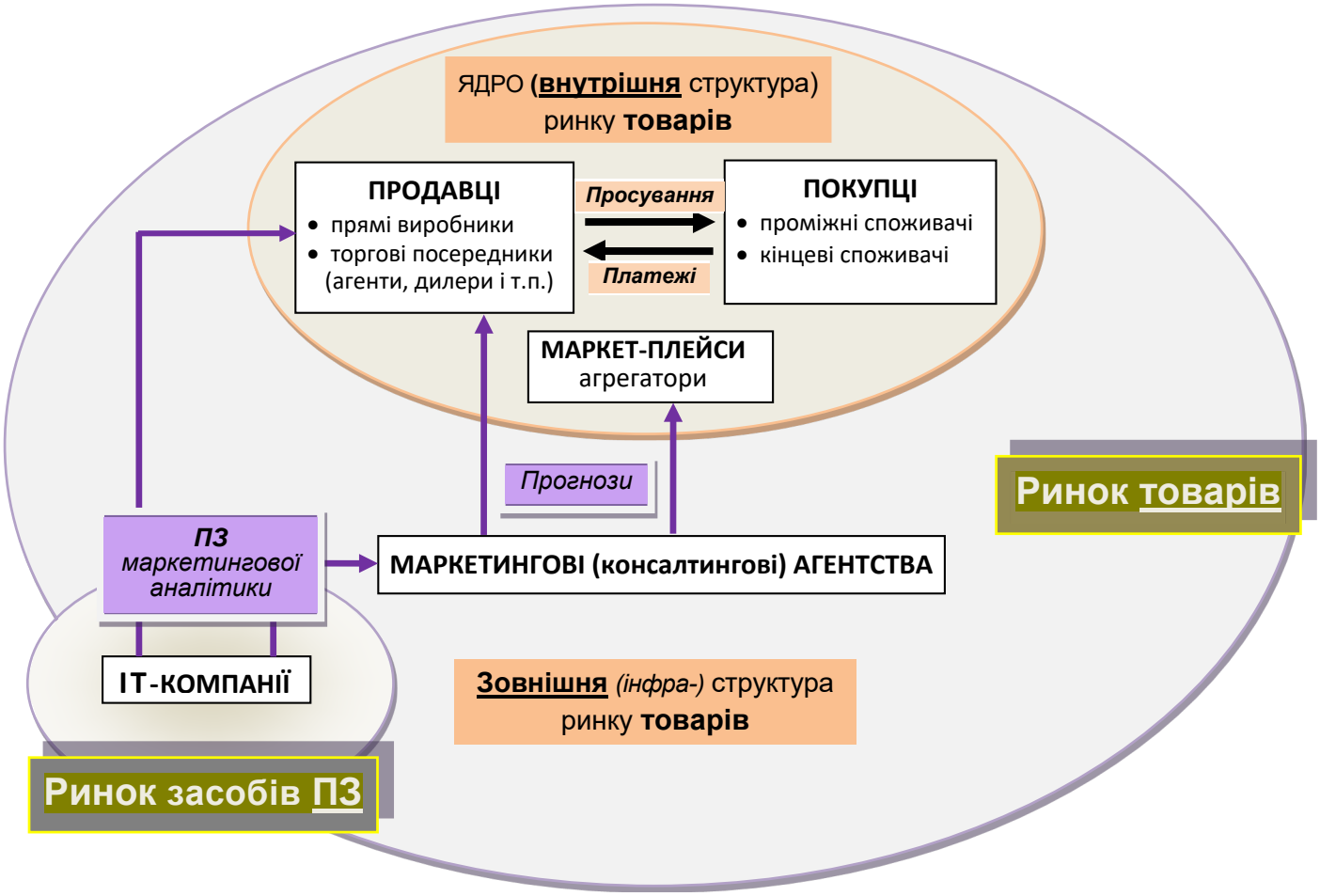


Рисунок 1.2 – Принципова схема взаємодії суб'єктів інтернет-маркетингу

Джерело: побудовано автором за даними [11. С. 23].

Кожна вказана група суб'єктів маркетингового процесу виконує у ньому особливу роль і вносить у нього специфічну форму товарів (продуктів).

Так, головним завданням продавців стає формування оптимальної **стратегії** реклами і збуту своєї продукції.

У свою чергу, *методологічну* основу цих стратегій формують маркетингові консалтингові агентства, головним результатом (формою продуктів) роботи яких є вироблення **прогнозів** (які включають виявлення найбільш вигідних каналів та способів розповсюдження продукції продавців – їх замовників).

І нарешті, *інструментальну* основу цих прогнозів становлять комплексні **системи маркетингової аналітики (надалі - СМА),** які виступають результатом діяльності (формою продуктів) IT-компаній.

Причому цільовими споживачами цих результатів стають як самі продавці основної продукції кожного ринку, так і посередники – ті ж самі консалтингові агентства та маркет-плейси.

Мета систем маркетингової аналітики – надання релевантних даних та актуальних прогнозів для прийняття рішень у сферах пошуку цільової аудиторії споживачів та проведення персоналізованих рекламних кампаній, порівняння прибутковості та вибору каналів розміщення реклами, оптимізації витрат на просування продуктів.

Ця мета конкретизується в ключових **завданнях** маркетингової аналітики**:** об'єднання органічних даних та даних із мережі, передбачення аномалій у динаміці маркетингових показників, а також спрощення роботи з кінцевими даними для осіб, що приймають рішення [20. С. 11].

Таким чином, системи маркетингової аналітики стають ключовим елементом усієї організації сучасного інтернет-маркетингу.

Дослідження цих систем слід розпочати з виокремлення їх основних об'єктів.

1.2. Основні об’єкти вивчення систем маркетингової аналітики

Однією з основних задач кожної системи МА є автоматизація виявлення кореляцій між різними компонентами маркетингової діяльності замовників. Більшість таких взаємозв’язків наперед відомі та широко стандартизовані. У маркетинговій практиці ці підлягаючі моніторингу й прогнозуванню кореляції отримали загальну назву «показники», ставши головними об’єктами дослідження систем аналітики. Ці показники прийнято поділяти на дві ключові групи: *метрики* та *KPI (Key performance Indicators)* [4; 23].

Різниця між цими групами числових показників полягає у формі виразу: KPI завжди обчислюються у відсотковому відношенні, а метрики – в абсолютних цифрах. Звідси і їх призначення: якщо метрики відбивають прості кількісні значення, то KPI демонструють, наскільки результативні маркетингові зусилля у порівнянні до плану.

При цьому обидві групи взаємопов’язані: метрики надають первинні дані, виходячи з яких обчислюються KPI маркетингу. Тому для повної оцінки результативності вироблюваної збутової стратегії потрібен аналіз обох груп показників. Саме такий аналіз виконується системами МА.

Розглянемо основні показники кожної групи для галузі Інтернет-маркетингу (див. рисунок 1.3).

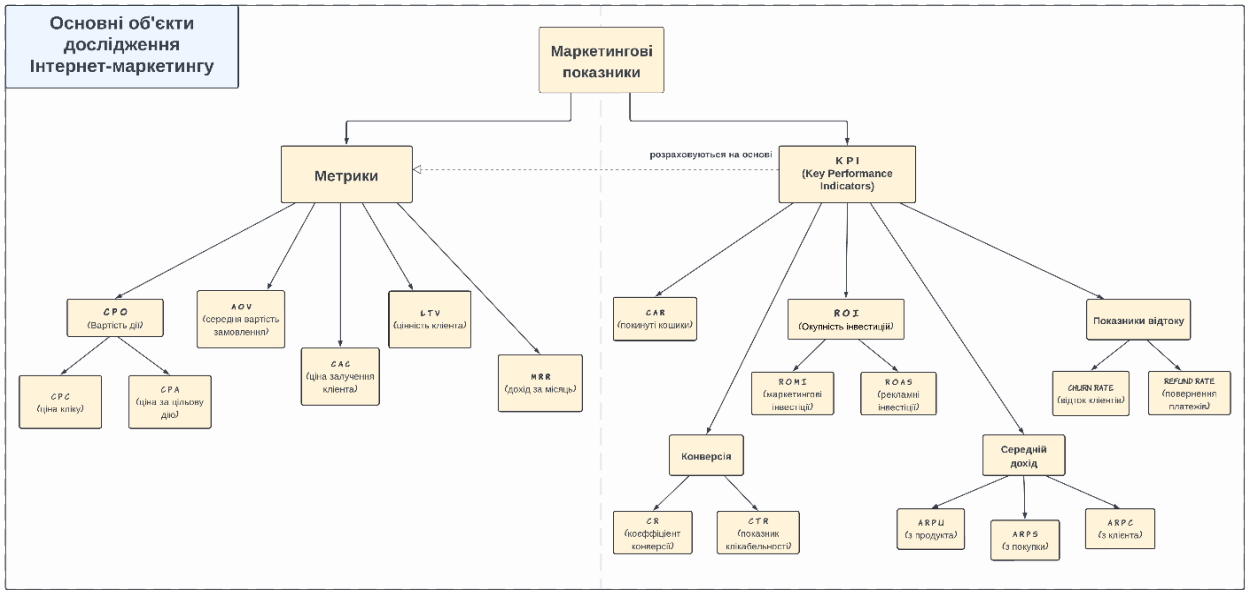


Рисунок 1.3 – Основні показники (метрики та KPI) інтернет-маркетингу

Джерело: побудовано автором за даними [4; 1].

А. Найважливіші маркетингові **метрики** —

показники, відстеження яких дозволяє оцінити результати просування продукції і оптимізувати витрати на різні компоненти маркетингової політики.

1. ***Середня вартість замовлення.*** Відслідковує середню суму, витрачену кожного разу, коли клієнт розміщує замовлення на веб-сайті або у застосунку:

$$\boldsymbol{AOV} = \frac{\text{Доходи від клієнтських платежів}}{\text{Кількість розміщених замовлень}}.$$

Даний показник є еталоном поведінки клієнтів, він стає базою для розрахунку цілого ряду інших важливих індикаторів, що дозволяють прогнозувати довгострокову цінність окремих клієнтів (показник **LTV** – див. нижче), оцінювати ефективність роботи з існуючою клієнтською базою (середній дохід з клієнта **ARPU**, ціна кліка **CPC**) та виявляти можливості нарощування доходів за рахунок оптимізації окремих компонент маркетингової стратегії – рекламної, цінової тощо (ціна за дію **CPA**, ціна залучення клієнта **CAC**, окупність інвестицій **ROMI, ROAS** та ін.).

2. ***Вартість замовлення*** *(cost-per-order)*. Це показник ціни залучення однієї покупки (оформлення одного замовлення). Ця метрика дозволяє оцінювати ефективність маркетингових кампаній та різних рекламних каналів:

$$\boldsymbol{CPO} = \frac{\text{Загальні витрати на рекламу}}{\text{Кількість створених замовлень}}.$$

В Інтернет-маркетингу цю ключову метрику часто доповнюють (або заміняють) рядом деталізуючих її показників, напр., ціною кліка та ціною дії**:**

- ***Ціна кліку*** (*cost-per-click*). Показує, скільки коштує 1 клік користувача по рекламному оголошенню. Чим нижче CPC, тим дешевше коштує залучення відвідувачів на сайт:

$$\boldsymbol{CPC} = \frac{\text{Витрати на рекламу}}{\text{Кількість кліків на рекламу}}.$$

- ***Ціна за дію*** *(cost-per-action).* Це сума, яку рекламодавець сплачує за цільову дію користувача: покупку, підписку тощо. Чим нижче метрика, тим вигіднішою є рекламна кампанія:

$$\boldsymbol{CPA} = \frac{\text{Витрати на рекламу}}{\text{Кількість дій користувача}}.$$

3. ***Ціна залучення клієнта. Відображає*** загальні витрати на залучення одного нового клієнта. Включає всі витрати, пов'язані з просуванням та продажами, необхідні для отримання нового клієнта (email-маркетинг, контент-маркетинг, SEO та ін.):

$$\boldsymbol{CAC} = \frac{\text{Витрати на залучення клієнтів}}{\text{Кількість залучених клієнтів}}.$$

Ця ж метрика може служити показником фінансової успішності здійснення *окремих компонент* маркетингової стратегії – рекламних кампаній. У такому разі її розраховують за формулою:

$$\boldsymbol{CAC} = \frac{\text{Витрати на рекламну кампанію}}{\text{Кількість залучених клієнтів}}.$$

4. ***Дохід* за *місяць*.** Щомісячний ***повторюваний*** дохід від підписок або регулярних платежів клієнтів. Ця метрика є важливою для сервісів, що надають послуги *за передплатою:*

$$\boldsymbol{MRR} = \begin{matrix}\text{Середній дохід}\\ \text{з користувачів}\\ \text{за період } (ARPU)\end{matrix} \times \begin{matrix}\text{Кількість}\\ \text{користувачів}\end{matrix}.$$

5. ***Цінність клієнта.*** Метрика відображає загальний прибуток, який приносить підприємству клієнт за період співпраці з ним. LTV дозволяє оцінити довгострокову цінність споживача з огляду на всі його покупки, підписки та інші джерела доходу для компанії. Чим вище LTV, тим ефективніше працює бізнес-модель вилучення цінності з клієнтської бази:

$$\boldsymbol{LTV} = \begin{matrix}\text{Дохід від клієнта}\\ \text{за весь період}\end{matrix} - \begin{matrix}\text{Витрати на його}\\ \text{залучення та утримання}\end{matrix}.$$

Для оцінки ефективності окремої маркетингової (чи рекламної) стратегії дану метрику можливо розраховувати і в іншому вигляді, особливо важливому для підприємств, розповсюджуючи свої продукти *за підпискою:*

$$\boldsymbol{LTV} = \begin{matrix}\text{Середня сума}\\ \text{покупок клієнта}\end{matrix} \times \begin{matrix}\text{Період}\\ \text{реалізації}\\ \text{стратегії}\end{matrix} \times \begin{matrix}\text{Частота}\\ \text{здійснення}\\ \text{покупок}\end{matrix}.$$

Б. <u>KPI</u> (*key performance indicators,* **ключові показники ефективності**) — інструменти оцінки результативності любої (індивідуальної і колективної) виконаної роботи. До них відносять [1; 4]:

1. ***Коефіцієнт конверсії (CR).*** Демонструє, яка частка аудиторії здійснює бажану дію**:** реєстрацію, купівлю, підписку, скачування програми. Чим вище CR, тим краще для продавця, оскільки більше відвідувачів стають покупцями:

$$\boldsymbol{CR} = \frac{\text{Кількість конверсій}}{\text{Кількість відвідувачів сайту}} \times 100.$$

В інтернет-маркетингу цю метрику часто замінюють таким її різновидом, як:

- ***показник клікабельності*** *(click transfer rate).* З його допомогою оцінюють привабливість та релевантність рекламного повідомлення для цільової аудиторії. Це відсоток користувачів, які клікають на оголошенні або посиланні, відносно загальної кількості показів:

$$\boldsymbol{CTR} = \frac{\text{Кількість кліків}}{\text{Кількість показів реклами}} \times 100.$$

2. ***Окупність інвестицій (ROI*** *– return on investment).* Це відношення прибутку від виконаних дій до витрат на них. Дозволяє оцінити окупність вкладень у будь-який маркетинговий або рекламний проект – для подальшого вибору вигідніших каналів рекламного трафіку чи способів збуту продукції:

$$\boldsymbol{ROI} = \frac{\text{Дохід з проекту} - \text{Витрати на проект}}{\text{Витрати на проект}} \times 100.$$

Значення показника понад 100% говорить про ефективність вкладень у ту чи іншу рекламну кампанію та результативність обраної стратегії збуту.

Це – найбільш важлива метрика бізнесу, яка насамперед відстежується системою маркетингової аналітики. До того ж її облік найпростіше піддається автоматизації через наявні на кожному підприємстві ПК бухобліку та CRM.

На практиці цю метрику часто доповнюють (чи замінюють) деталізуючими її показниками – окупність маркетингових (збутових) та рекламних витрат**:**

- ***Окупність інвестицій у маркетинг.*** Це відношення прибутку від виконаних збутових проектів до витрат за них. Показник дозволяє оцінити віддачу від роботи маркетингової і рекламної служби:

$$\boldsymbol{ROMI} = \frac{\text{Доходи підприємства} - \text{Витрати на збутові проекти}}{\text{Витрати на збутові проекти}} \times 100.$$

- ***Окупність реклами.*** Показник ефективності вкладень у рекламу як один з елементів маркетингу. Це відношення виручки від рекламної кампанії до витрат на неї:

$$\boldsymbol{ROAS} = \frac{\text{Дохід від рекламної кампанії}}{\text{Витрати на рекламну кампанію}} \times 100.$$

3. ***Показники середнього доходу.*** Вимірюють дохід, одержуваний від одиниці проданої продукції або від одного клієнта. Ці показники уточнюють уявлення про внесок у генерацію загальних доходів під-приємства з боку його різних підрозділів чи видів продукції. Залежно від зазначених цілей цей KPI розраховують у різному вигляді**:**

- ***Середній дохід з <u>продукту</u>.*** Виявляє ступінь прибутковості для підприємства прийнятого товарного асортименту (у цьому випадку розраховується для всіх видів продукції), або окремих його елементів (розраховується відокремлено для кожного виду продукції):

$$\boldsymbol{ARPU} = \frac{\text{Загальний дохід підприємства}}{\text{Кількість проданих одиниць продукції}} \times 100.$$

- ***Середній дохід з <u>клієнта</u> (користувача).*** Показує рівень лояльності аудиторії та потенціал зростання доходів. Розраховується як середня сума грошей, витрачених одним клієнтом протягом певного періоду (“*середній чек*”). Динаміка показника дозволяє оцінити ефективність роботи з наявною базою клієнтів:

$$\boldsymbol{ARPC} = \frac{\text{Загальний дохід підприємства}}{\text{Кількість клієнтів}-\text{покупців}} \times 100.$$

Для підприємств, які продають свої продукти *за підпискою,* часто вигідно поєднувати обидва вищевказані KPI в один показник**:**

- ***Середній дохід з <u>покупки</u>.*** Демонструє ступінь успішності у продажу не лише основного продукту, але й супутніх йому додаткових послуг (дохід від усієї здійсненої *угоди*). Саме у такій формі KPI середнього доходу прийнято розраховувати у більшості IT-компаній, які застосовують бізнес-моделі Product-as-a-Service чи Software-as-a-Service (особливо – за підпискою):

$$\boldsymbol{ARPS} = \frac{\text{Загальний дохід підприємства}}{\text{Кількість покупців } SaaS-\text{пакету}} \times 100.$$

4. Показники ***<u>відтоку</u>*** клієнтів та *доходу:*

- індикатор відтоку ***клієнтів.*** Демонструє частку клієнтів, які *припинили* користуватися продуктом чи послугою протягом встановленого проміжку часу (дня, тижня, місяця). Зниження відтоку є одним із ключових

завдань утримання клієнтської бази. Чим нижче цей показник, тим краще для бізнесу. Це означає, що користувачі залишаються з підприємством та задоволені її продуктами:

$$\boldsymbol{Churn\ Rate} = \frac{\text{Кількість клієнтів відказників}}{\text{Початкова кількість клієнтів}} \times 100.$$

- коефіцієнт *повернення* платежів. Повернений платіж – це сума, яка повертається продавцем покупцю внаслідок ануляції останнім платежу за зроблену покупку. Таким чином, даний індикатор показує частку клієнтів, що відмовилися від використання купленого продукту:

$$\boldsymbol{Refund\ Rate} = \frac{\text{Сума чи кількість повернених платежів}}{\text{Загальна сума|кількість платежів}} \times 100.$$

5. ***Покинуті кошики.*** Це відсоток відвідувачів сайту (застосунку), які додали продукт чи послугу до кошика, але не оформили замовлення. За його допомогою можна оцінити проблеми та бар'єри в процесі покупки та покращити конверсію:

$$\boldsymbol{CAR} = \frac{\text{Кількість оплачених замовлень}}{\text{Кількість клієнтів,що задіяли кошик}} \times 100.$$

Таким чином виглядає формальна (мінімальна) структура переліку показників, які відслідковуються та прогнозуються системами маркетингової аналітики в стандартному режимі. Проте на практиці реальне наповнення цього переліку ніколи не залишається вичерпним: воно завжди ситуаційно зумовлене, і визначається передусім запитами, що надходять від клієнтів.

Об’єктом нашого дослідження виступатиме ІТ-компанія, яка входить до консорціуму «Genesis» і спеціалізується на наданні послуг з вивчення іноземних мов через різноманітні інтернет-платформи. Самою природою даної діяльності зумовлені й обрана компанією збутова модель (продукти просуваються у вигляді “Product-as-a-Service” за довгостроковою *підпискою*), і пріоритетні для неї маркетингові показники (насамперед, пов’язані з обліком *платіжних* даних, причому часто *нестандартних*).

Це змушує дану компанію висувати досить своєрідні вимоги до системи маркетингової аналітики, не всі з яких спроможні задовольняти найбільш розповсюджені «традиційні» системи маркетингової аналітики (див. розділ 2

нашого дослідження), і підштовхнуло її до створення власної СМА, архітектурні та функціональні елементи якої будуть розглянуті у проектному розділі 3 даної роботи.

Проте перед цим, нам слід отримати загальне уявлення про поточний стан світового ринку СМА та виокремити значущі особливості найбільш розповсюджених на ньому систем – потенційних конкурентів розроблюваного прототипу.

1.3. Огляд ринку програмних комплексів маркетингової аналітики

Дедалі зростаюче значення аналітичних маркетингових систем для оптимізації процесів вироблення й реалізації стратегій сучасних підприємств не пройшло повз уваги дослідників. Зокрема, для підкреслення цього факту всесвітньовідома прогностична компанія «Gartner» у 2023 р. змінила назву своєї щорічної Доповіді з «Платформи Бізнес аналітики» на «Платформи Бізнес та Маркетингової аналітики». За спостереженнями експертів «Gartner», замовники дедалі частіше переходять від «підготовки описових звітів, відображаючих поточний стан продажів, до поглибленого діагностичного аналізу й візуального представлення даних, та до прогнозної аналітики» [13].

У свою чергу, найбільш важлива роль систем маркетингової аналітики фіксується у розбудові та функціонуванні домену інтернет маркетингу. Тому саме у даній сфері бізнесу вони отримали найбільш прискорене поширення.

Так, у 2022 р. загальний обсяг світового ринку СМА оцінювався у 18,2 млрд. дол., з яких більша частина припадала на Північну Америку [17]. Згідно зі звітами авторитетного американського центру "Mordor Intelligence", тільки у США розмір ринку маркетингової аналітики в 2024 р. складе 6,31 млрд. дол. і, як очікується, до 2029 року досягне 11,54 млрд. дол. (середньорічний темп приросту – 12,9%) [15]. А за відомим прогнозом дослідницької компанії "Universal Data Solutions", глобальний ринок маркетингової аналітики в 2023-2030 роках зростатиме ще значнішими темпами – до 14% щорічно [19].

На цьому ринку налічується 10-15 основних гравців – виробників ПЗ та програмних комплексів. Найбільш вагомими є корпорації «Google», «Microsoft», «Adobe», «Oracle», «HubSpot», «Salesforce» та «Teradata». Вони надають клієнтам безліч систем – як генералізованих, широко інтегрованих всередині і назовні, проте із дуже обмеженим типовим функціоналом, так і таких, що легко адаптуються під вимоги конкретних замовників.

На відміну від розвинених країн Заходу, у Східній Європі більшою популярністю користуються саме СМА у вигляді генералізованих платформ, чому насамперед сприяє дешевизна їх впровадження, простота використання та поточної підтримки. Ці системи переважно постачаються компаніями «Google» (*Google Analytics*), «Hubspot» (*Hubspot Marketing Hub*), «Microsoft» (*Dynamics 365* та *Fabric*), а також «Adobe» (*Adobe Analytics*). Як показав наш аналіз, у цьому регіоні (в т.ч. й в Україні) повністю **відсутні власні виробники** подібних програмних комплексів.

Однак слід враховувати наявність у систем такого типу кількох суттєвих недоліків. По-перше, у них закладено моніторинг обмеженої кількості показників – часто недостатньої для вироблення ефективних прогнозів та маркетингових стратегій підприємств - замовників. По-друге, пропонований ними інструментарій аналізу навіть цих показників дуже скудний. Тому, на практиці замовники досить рідко використовують платформи МА в чистому вигляді, часто віддаючи перевагу доповненню їх *стороннім* ПЗ, яке надає істотно ширші можливості кастомізації як самих аналізованих показників, так і всіх процедур їх життєвого циклу (збору, зберігання, обробки та візуалізації). Щодо України, тут найбільшою популярністю серед останніх користуються зарубіжні «Owox» і «Woopra», а також вітчизняні «Livepage», «Ringostat» та «Netpeak».

Розгорнутий порівняльний аналіз зазначених СМА, з фокусом на розкритті їх переваг та недоліків, буде проведено у розділі 2 нашої роботи. Його метою стане виявлення набору загальних вимог до програмних комплексів обробки аналітичної інформації.

Висновки до розділу 1

У даному розділі були розглянуті загальні засади (концепції, структура та механізми) функціонування Інтернет-маркетингу як ключової форми організації збутової роботи у цифрову епоху. Були розкриті переваги даної бізнес-моделі перед «офлайн» маркетингом та можливості застосування її ключових інструментів – web-ресурсів, контент-маркетингу, контекстної реклами, SEO-просування, email-маркетингу і маркет-плейсів. Окрім цього, було встановлено, що провідним чинником їх ефективного застосування є розвиненість програмних комплексів і систем маркетингової аналітики (СМА) – інструментальних засобів автоматизації прийняття стратегічних рішень на всіх етапах «воронки продажів».

Також, було проведено аналіз основних об'єктів СМА – показників ефективності маркетингової діяльності, у ході якого було пояснено їх роль в оцінці ефективності рекламних та інших збутових кампаній та проведено їх класифікацію на 2 групи – первинні метрики та KPI (комбінації метрик).

Багатокритеріальний огляд ринку СМА дозволив оцінити сучасний стан і тенденції його розвитку та виокремити комплекс взаємовідносин між виробниками і споживачами цих прогнозних інструментів. Зокрема, була пояснена наявність на ньому великого числа ПК різної спрямованості, ступеня зовнішньої і внутрішньої інтеграції та можливостей пристосування до вимог замовників. Першу групу становлять генералізовані платформи маркетингової аналітики, виробниками яких є всесвітньовідомі корпорації Google, Microsoft, Adobe, HubSpot та ін., які набули популярність саме завдяки охопленню широкого спектру процесів Інтернет-маркетингу. Проте зворотнім боком «всеохопност» стає їх надмірна стандартизація, що суттєво знижує потенціал їх підлаштування під різні бізнес-ситуації. Тому на практиці споживачі часто віддають перевагу другій групі СМА – доповнюючим ПЗ, які, базуючись на функціоналі вказаних платформ, намагаються усунути їх означений недолік, надаючи можливості кастомізації і самих аналізованих показників, і процедур їх життєвого циклу (збору, зберігання, обробки даних та візуалізації результатів). Також було встановлено, що в Україні найпопулярнішими з них є зарубіжні ПК «Owox» і «Woopra», а також вітчизняні «Livepage», «Ringostat» та «Netpeak».

РОЗДІЛ 2. ПОРІВНЯЛЬНИЙ АНАЛІЗ КЛЮЧОВИХ ЕЛЕМЕНТІВ ПРОВІДНИХ СИСТЕМ МАРКЕТИНГОВОЇ АНАЛІТИКИ

Безперечно, однією з важливих складових успішного планування ПЗ є функціональний та порівняльний аналіз уже існуючих та перевірених часом рішень – аналогів. Тут необхідно сформувати перелік базових вимог до програмної системи, виділити сильні та слабкі сторони пропозицій конкурентів, а також врахувати їх позитивний ы негативний досвід на різних етапах процесу розробки і ринкового просування продукту. У попередньому розділі було показано, що має сенс розділяти СМА на єдині інтегровані платформи та доповнюючі їх SaaS *(Software-as-a-Service)*-комплекси. У цьому розділі роботи будуть описані програмні системи обох груп – з одного боку, популярні в регіонах Північної Америки та Східної Європи HubSpot та Google Analytics 4, а з іншого – менш іменити платформи-доповнення Owox BI та Woopra.

2.1. «Google Analytics 4»: Можливості та обмеження найвідомішого програмного комплексу інтернет-аналітики

Корпорація Гугл є розробником двох послідовних версій систем маркетингової аналітики – Google Universal Analytics (далі – GUA) та Google Analytics 4 (далі – GA4).

Google Analytics 4 (початкова назва – Google Analytics App+Web) – остання версія аналітичного інструменту від «Google», який подає дані про відвідувачів і користувачів веб-ресурсів. Продукт був представлений у жовтні 2020 р. як революційний прорив в індустрії інтернет-аналітики, виступивши новим поколінням після Universal Analytics.

Платформа GA4 надає клієнтам широкі можливості для детального аналізу даних, прогнозування поведінки користувачів та різнобічної оптимізації маркетингових стратегій. При цьому, її використання абсолютно

безкоштовне, лише з деякими нефункціональними обмеженнями – наприклад, кількість подій, що збираються за день, або кількість ГБ даних, мігрованих у сховище Google BigQuery.

1. Модель збору даних ([28])**.**

GA4 збирає дані на основі подій та параметрів (а не короткоживучих сесій, як у GUA), дозволяючи більш гнучко та докладно аналізувати взаємодію користувачів з веб-ресурсом продавця, включаючи глибину прокручування сторінки, взаємодію з пошуковим рядком, вихідні кліки та ін.

1. Події. Це діяння, які здійснюють користувачі (перегляд сторінки, клік, відправлення форми та ін.). Кожна подія в GA4 має назву та набір параметрів, які описують контекст виконання та додаткові відомості про неї.

2. Параметри. Дають додаткову інформацію про події та допомагають розділити їх на більш специфічні категорії. Так, можна додати параметри для вказівки типу контенту, категорії продукту або джерела трафіку, щоб зручніше класифікувати дані та проводити аналіз конкретних зрізів подій.

Для збору даних GA4 використовує спеціальний "тег" Google (т.зв. "gtag"), який необхідно впровадити у код веб-ресурсу (лише один раз, якщо використовувати інтеграцію з Google Tag Manager). Цей тег являє собою звичайний JS-скрипт, який відстежує дії, що виконуються користувачем, в межах ресурсу. Він відстежує взаємодію користувачів за допомогою подій та параметрів, а потім передає ці дані до GA4 для аналізу. Крім автоматичного збору, додаток також може відправляти релевантні події в GA, використовуючи програмний інтерфейс (т.зв. API) Measurement Protocol-у.

2. Ідентифікація користувачів.

У GUA ідентифікація користувачів здійснювалась за допомогою кукі-файлів та ідентифікаторів клієнтів. GA4 працює по-іншому: вона анонімно ідентифікує користувачів за допомогою унікального ключа та подій, пов'язаних із конкретним клієнтом. Таким чином ми можемо дізнатися більш точну поведінку користувача без строгих обмежень конфіденційності скоєних дій.

3. Машинне навчання ([28])**.**

У GA4 вбудовано функціонал машинного навчання для попередження клієнтів про прогнозовані зміни в даних. Система автоматично виявляє приховані патерни і передбачає поведінку користувачів на базі наявних даних. Доступні 2 способи застосування ML**:** підказки та прогнозовані показники.

Підказки або сповіщення від системи можна побачити, натиснувши кнопку «Підказки» праворуч угорі. У випадаючому меню наявна не тільки вся інформація, яку система вважала за необхідне показати, а й рядок пошуку.

Прогнозовані показники доступні через вкладку «Аналіз» у лівому меню.

Таким чином, за допомогою ML «Google Analytics» здатна**:**

- Показати аномалії у звітах.
- Передбачити можливість конверсії (і надіслати нові аудиторії у «Google Ads»)
- Заздалегідь дізнатися про підвищення попиту на товари.
- Передбачити можливість відтоку клієнтів і перешкодити цьому, та багато іншого.

4. Розширений функціонал.

GA4 пропонує детальні шляхи для розуміння взаємодії користувачів, інтеграцію з додатками для аналізу поведінки користувачів усередині них, розширене коло метрик, що відстежуються за замовчуванням, і вбудовані звіти з інтелектуальними пропозиціями.

5. Крос-платформеність.

У GA4 можна відстежувати та аналізувати поведінку користувача на різних пристроях (смартфонах, планшетах, десктопах). При цьому система автоматично (через міжплатформену ідентифікацію користувачів) об’єднує і видаляє дублі взаємодій користувачів, що з'явилися під час збору первинної інформації – тобто враховує лише реальних користувачів, які взаємодіяли з продавцем на різних платформах, а не їх входи на сайти чи додатки. У GUA аналітика мобільних додатків та веб-сайтів розглядалися як два різні потоки даних, які потребують інтеграції для повного розуміння поведінки

користувачів. А в GA4 не потрібно використовувати кілька інструментів аналітики для сайтів та програм: вся інформація потрапляє до єдиної звітності.

6. Інтеграція з іншими інструментами.

Google GA4 має покращену інтеграцію з багатьма продуктами екосистеми Google – зокрема, сервісами Google Ads (менеджер рекламних кампаній) та BigQuery (OLAP сховище даних). Завдяки цьому можна легко відстежувати зв'язки між рекламними каналами та конверсіями, а також оптимізувати інвестиції в рекламу на основі результатів аналізу.

Таким чином, проведений нами вище функціональний аналіз дозволяє виділити ключові переваги та недоліки GA4 як у порівнянні зі своєю попередньою версією (GUA), так і альтернативними системами маркетингової аналітики.

Основні переваги GA4 ([28]):

1. Вбудована багатоканальність**.** Можна відстежувати користувацькі взаємодії та конверсії по всіх каналах маркетингу, включаючи органічний пошук, соціальні мережі, прямий трафік, рекламні кампанії та ін. Крім того, система здатна реєструвати величезну кількість дій користувача, пов'язуючи кожну із внутрішньою подією. Це дозволяє формувати детальні хронологічні зрізи за певними групами відвідувачів чи відстежувати відгук аудиторії рекламні кампанії.
2. Безкоштовність**.** Більшість функціоналу GA4 надається абсолютно безкоштовно, лише кількісно обмежуючи виконання деяких операцій. Хоча СМА також має і просунуту «корпоративну» версію (GA4 360), вони здебільшого відрізняються лише нефункціональними можливостями.
3. Наскрізне відстеження користувачів**.** GA4 унікально ідентифікує кожного споживача, а не девайс, що дозволяє пов'язати дані про взаємодію з одним і тим самим користувачем на різних пристроях та платформах.
4. Інтеграція з продуктами Google**.** Може взаємодіяти з трьома продуктами – Google Ads, Google Tag Manager і Google BigQuery. При цьому:
   - інтеграція з BigQuery дозволяє будувати точну та гнучку аналітику

поведінки покупців та створювати складні запити на великих обсягах даних, що важливо насамперед для великого бізнесу.

- інтеграція з Google Ads дозволяє прямо в інтерфейсі GA4 переглянути дані щодо рекламних кампаній.
- інтеграція з Google Tag Manager (GTM) дозволяє змінювати показники, що збираються, і налаштовувати процес збору даних з ресурсів, не змінюючи при цьому їх програмний код.

5. Поведінкове прогнозування. Крім стандартних прогнозів за основними показниками маркетингу, GA також вміє передбачати подальшу поведінку користувача (або цілої аудиторії) на підставі вже скоєних ним(і) дій.
6. Доступний інтерфейс. UI системи відрізняється досить широким функціоналом для просунутих експертів, і водночас простий у освоєнні для новачків у маркетинговій аналітиці.
7. Обробка на стороні сервера**.** Усі СМА збирають, обробляють та надсилають дані на стороні клієнта. Основним недоліком такого підходу є неможливість кінцевого споживача системи отримає доступ до «сирих» аналітичних даних, оскільки подібний рівень контролю не передбачений їхньою SaaS-природою. Однак, GA4 дозволяє перенести такі навантаження на бік сервера, тим самим забезпечуючи можливість переглядати та змінювати події до того, як вони будуть доставлені до системи. Для цього в налаштуваннях GA4 необхідно впровадити Google Tag Manager у ролі менеджера збору даних, а також створити серверний контейнер, який керуватиме трьома цими процесами згідно зі списком заданих правил.

Недоліки та **обмеження** Google Analytics 4 ([16]):

1. Ліміт для зберігання історичних даних. У багатьох інших платформах маркетингової аналітики можна встановити необмежений термін зберігання історичних даних, а в GA4 максимальний строк їх зберігання становить 14 місяців (за замовчуванням – лише 2 місяці). Це змушує клієнтів робити резервні копії з метою збереження цінної інформації, вдаючись до використання інших сховищ даних – що не є безкоштовним.

2. Складнощі з міграцією даних з інших систем.
   - Не реалізований імпорт даних з альтернативних систем МА, а отже клієнти не можуть аналізувати ефективність рекламних кампаній, не пов'язаних із Google.
   - Крім того, GA4 не підтримує імпорт зібраних даних навіть зі своєї попередньої версії (Universal Analytics).
3. Затримка обробки даних. У разі необхідності аналізу *великих обсягів* даних, GA офіційно визнає неможливість виконувати запити у режимі реального часу (затримка може досягати 24-48 годин).
4. Замкненість платформи**.** Google Analytics не може похвалитися гарною підтримкою сторонніх інтеграцій, в основному покладаючись на інструменти та ресурси власної хмарної платформи. Через це система часто замикає клієнтів в екосистемі Google, обмежуючи варіанти вибору інструментів ззовні.
5. Недоліки моделі аналізу даних. У GA4 введено величезну кількість відстежуваних показників і метрик, проте більша частина з них не є стандартною для маркетологів. З іншого боку, з моделі вилучено низку показників, важливих саме для Інтернет-маркетингу (а додавання нових показників моделлю не передбачено).

   До того ж, коло аналізованих метрик задається системою за замовчуванням і не може бути *розширено* (показники не поділяються на звичайні та розширені). Тому для аналізу додаткових (необхідних клієнтам) метрик потрібно користуватися сторонніми сервісами.
6. GA4 пропонує мало вбудованих звітів. Деякі звіти, які вважаються стандартними у бізнесі, виключені із переліку. На відміну від системи Universal Analytics, яка надавала понад 100 готових шаблонів звітів, Google Analytics 4 розбиває звітність на 6 блоків, кожен із яких містить лише кілька представлень. Тому клієнтам часто доводиться створювати власні звіти, або навіть вдаватися до допомоги сторонніх інструментів.

2.2. «HubSpot Marketing Hub»: широка інтеграція на шкоду аналітичному потенціалу платформи

Дане хмарне ПЗ однойменної американської компанії є аналітичним компонентом загальної **CRM**-платформи – однієї з найпопулярніших у сучасному маркетингу систем управління взаємовідносинами з клієнтами ([27]) проте окрім цього включає багато інших продуктів і рішень**:**

- ПЗ для електронного маркетингу;
- Інструменти просування в соціальних мережах;
- ПЗ для відстеження електронної пошти;
- Рекламне програмне забезпечення;
- Багатофункціональний конструктор веб-сайтів [29].

Як CRM, сервіс пропонує низку безкоштовних інструментів та опцій (що дозволяє клієнту налаштовувати його під конкретні бізнес-завдання), а також більш ніж 1340 зовнішніх інтеграцій (включаючи Microsoft Dynamics і Salesforce), а також дозволяє розробникам створювати власні інтеграції, доступні на спеціальному внутрішньому маркет-плейсі.

Маркетинговий центр HubSpot

Marketing Hub – один із численних програмних модулів HubSpot, спеціалізований під цілі маркетингу. Його задачами стали збільшення трафіку, залучення відвідувачів та організація повних кампаній вхідного маркетингу.

У модулі доступно більше 60 функцій, серед яких є автоматизація**:**

- поточного виконання основних маркетингових операцій;
- бізнес-аналітики (хоча й досить обмежена – див. нижче);.
- лідогенерації та SEO-інструментів, просування в соціальних мережах.
- Керування рекламними кампаніями Google, Facebook, LinkedIn за допомогою зібраних даних.
- ПЗ для створення звітів. Надаються докладні звіти про продаж, продуктивність, індивідуальну ефективність виконавців робіт. Число користувачів – не обмежене. CRM підтримує до 1 млн. контрактів без

обмежень за часом та терміном дії.

- Дуже важливе значення має інший модуль HubSpot – його Операційний центр (Operations Hub). Він подає єдиний набір програмних інструментів, який поєднує програми, очищує та курирує дані клієнтів, автоматизує процеси на одній центральній CRM платформі. Тут варто виділити найпопулярніші функції модуля:
- Синхронізація та очищення баз даних клієнтів — доступна без знання коду, включає зіставлення налаштовуваних полів, фільтрацію та історичну синхронізацію.
- Програмована автоматизація.
- Підбір даних для звітності (число метрик стандартизовано та обмежено у безкоштовній версії).
- Автоматизація якості даних, включаючи автоматичне виправлення властивостей дати, імен форматів.

Таким чином, можна зробити висновок, що популярність HubSpot як єдиної платформи для онлайн аналітики зумовлена не стільки високими функціональними можливостями її маркетингового модуля, скільки переліком інструментів, що його супроводжують (напр., CRM та Sales Hub), зручно зібраних в одному місці [29].

На відміну від Google Analytics 4, будь-яка компонента платформи HubSpot надається за умовно безкоштовною моделлю. Так, у кожного модуля є кілька безкоштовних інструментів досить обмеженого функціоналу, а доступ до його розширених функцій можливий лише на платній основі – за умови підключення преміум тарифів.

Ціна Marketing Hub:

- Безкоштовний тариф включає: конструктор форм та лендингових сторінок, налаштування поштових розсилок, живий чат із клієнтами (без бізнес-телефонії), менеджер рекламних кампаній у Facebook, Google і LinkedIn, а також базові можливості формування звітів та проведення аналітики на даних.
- Тариф «Початковий» — (від $20/місяць) додає автоматизацію базових

бізнес-процесів, а також знімає обмеження на кількість листів, що відправляються в розсилці.

- Тариф «Професійний» — (від $800/місяць) додає розширені аналітичні можливості, підтримку ір-телефонії, а також автоматизацію різних видів контент-маркетингу.
- Тариф «Корпоративний» — (від $3600/місяць) є найбільш значущим у контексті нашої роботи, оскільки додає можливість збирання довільних даних із користувачів ресурсів, аналізувати шляхи покупців по воронці продажів, а також прогнозувати ключові показники маркетингової аналітики.

Переваги HubSpot [24]:

На сьогоднішній день цей програмний комплекс можна вважати однією з найкращих інтегрованих платформ автоматизації ведення операційної (поточної) маркетингової діяльності в будь-якому домені. При цьому ПЗ добре адаптоване під потреби різних типів споживачів – b2b, b2c і т.п. [29].

1. Просте використання. Інтерфейс платформи відрізняється інтуїтивно-зрозумілою навігацією для швидкого доступу до ключового функціоналу, включаючи грамотно додані гіперпосилання на релевантні ресурси та модулі. Крім того, компанія має чудову підтримку співтовариства і маркетологів (для яких створено безліч посібників), і розробників ПЗ (активно ведеться технічний блог). І нарешті, HubSpot підтримує власну академію та пов'язаний з нею освітній портал, який займається акредитацією фахівців по завершенню ними відповідних сертифікаційних програм. Завдяки цьому з використанням системи розбереться навіть недосвідчений в інтернет-маркетингу користувач.
2. Широкий функціонал. Впровадивши HubSpot, можна забути про використання інших сервісів, адже система дозволяє керувати всіма аспектами взаємодії з клієнтами, від трекінгу взаємодій з рекламними кампаніями до звітності та аналітики з продажів.
3. Оптимізація часових витрат. За рахунок всебічної автоматизації найважливіших бізнес-процесів, платформа дозволяє витрачати значно

менше часу на зведення та обробку даних, роботу з клієнтами, адміністрування поточних та планування майбутніх рекламних кампаній.

4. Великий спектр зовнішніх інтеграцій. HubSpot має вбудовану підтримку величезної кількості (більше 1500) підключень до сторонніх постачальників послуг. Це дозволяє підприємствам – клієнтам централізувати всю свою суб'єктну діяльність (включаючи маркетинг, продаж, зв'язок із покупцями та ін.) у межах однієї системи. Такий підхід підкреслює високий ступінь інтегрованості платформи, що дозволяє вільно замінювати одні компоненти на інші (напр., «шар» передобробки даних).
5. Гнучкість та масштабованість. Якщо бізнес постійно розширюється і поточних можливостей платформи стає недостатньо – можна перейти на тариф вищого рівня: у тарифній сітці представлені плани як для малих та середніх (Starter і Professional), так і для великих компаній (Enterprise). Крім того, оплата може здійснюватися як окремо за кожен підключений модуль (це дозволяє масштабувати лише деякі компоненти із загального числа використовуваних), так і оптом за весь набір інструментів.

З іншого боку, попри великий функціонал і багато переваг цієї платформи, вона не позбавлена недоліків. Головні з них випливають із природи HubSpot, що позиціонується насамперед як CRM платформа:

Недоліки HubSpot**:**

1. Слабка аналітична частина. В основному аналітика платформи налаштована на відстеження шляху клієнта за угодами і воронками продажів, що *вже відбулися,* і не дозволяє робити прогнози для *майбутніх* маркетингових дій (це можливе лише в «корпоративній» версії ПЗ). Крім того, такий функціонал розкиданий за різними модулями, що може ускладнити процес зведення результатів.
2. Обмеженість безкоштовної версії. Насправді, безкоштовним у базовому тарифі HubSpot є знов-таки лише обслуговування *поточних* маркетингових операцій, а доступ до аналітики та прогнозування *майбутніх* дій передбачено лише у *найдорожчому* «корпоративному» тарифі.

3. Складність первинного налаштування. Хоча в цілому (як ми вказували) користуватися платформою нескладно, проте її початкове налаштування неможливе без допомоги технічного спеціаліста та доменного експерта. Це особливо вірно у випадках, коли клієнт потребує індивідуалізованих рішень, які вимагають інтеграцій з зовнішніми сервісами та/або послугами.
4. «Парадокс композитності». Висока модульність платформи є «двосічним мечем»: хоча це і дозволяє користувачам бути незалежними від конкретної програмної екосистеми, проте в той же час викликає труднощі при розширенні вимог до системи аналітики. Для побудови нетипових аналітичних рішень на основі платформи часто доводиться вдаватися до сторонніх SaaS-сервісів для вирішення специфічних завдань, відповідальність за виконання яких HubSpot перекладає на плечі спільноти. Наприклад, додавання етапу передобробки вхідних даних (тобто ETL-конвеєра даних) у самому HubSpot не передбачено.
5. Необхідність навчання персоналу. Продовжуючи попередній пункт – через неминуче обширний технічний стек, установи-клієнти HubSpot змушені інвестувати додаткові ресурси (часові та фінансові) на навчання свого персоналу хитросплетінням звітності за виконуваними проектами. Саме з цієї причини компанія HubSpot вдалася до відкриття при собі власну освітню філію та сертифікації фахівців.
6. Відсутність української мови. Хоча ПЗ платформи локалізовано для багатьох європейських й азійських регіонів, підтримка української відсутня.

2.3. Загальна характеристика нішевих аналітичних програмних комплексів

Як показано вище, в обох популярних інтегрованих платформ маркетингової аналітики, окрім явних переваг, є й низка значущих для сфери інтернет-маркетингу недоліків. Їх присутність породжує різні спроби *доповнення* зазначених єдиних платформ сторонніми програмними

комплексами, покликаними частково виправити або поліпшити їх у певних аспектах. Подібне ПЗ створюється різними за розміром і національною приналежністю компаніями, з яких не всі відносяться до сфери IT: нерідко їх виробниками стають самі маркетингові агенції і навіть маркет-плейси. У результаті на ринку МА кожної розвиненої країни можна зустріти безліч різноманітного доповнюючого ПЗ, яке розширює й адаптує функціонал генералізованих платформ під специфічні потреби замовників [27].

Розглянемо основні особливості 4 таких програмних комплексів маркетингової аналітики, які набули найбільшого поширення на внутрішньому ринку України – одного зарубіжного (Owox) та трьох вітчизняних (Netpeak, Livepage.ua та Ringostat).

Сервіс бізнес-аналітики OWOX BI.

OWOX — американська компанія, що поширює три основних продукти - SaaS-інструменти бізнес-аналітики – Smart Data, Attribution та Pipeline. Разом вони складають потужний і гнучкий інструмент для збору, зберігання й аналізу даних про різні аспекті діяльності підприємств, на основі яких можна приймати якісні рішення щодо подальшого розвитку бізнесу.

Для цілей нашого дослідження найбільший інтерес становить OWOX BI Pipeline — пропрієтарна платформа для збору та об’єднання даних з різних систем – наприклад Google Analytics, CRM, веб-сайтів, додатків, рекламних служб та інших (включаючи скасовані та корпоративні замовлення, покупки кол-центру та транзакції офлайн-магазину). Крім того, дана система здатна брати на себе виконання всіх операцій наскрізної маркетингової аналітики: вона *збирає* і обробляє первинні дані, відправляє їх до Google BigQuery та дозволяє поглиблено *аналізувати*, а потім – і наочно *відображати отримані результати* у різноманітних звітах для оцінки ефективності рекламних кампаній. Також, можна створювати персоналізовані звіти із ключовими маркетинговими показниками, що будуть розраховуватися автоматично (напр. ROI, CPA, CAC та LTV).

Внаслідок широкої інтеграції аналітика в OWOX стає детальною, а завдяки інструментам довільного розширення чи обмеження кола відстежуваних метрик – ситуаційною, конкретизованою під вимоги кожного клієнта. Інтегрувавши всі свої джерела даних, клієнт може побачити реальний прибуток, отриманий за кожним атрибуційним каналом чи рекламною кампанією. Таким чином, ця СМА дозволяє оптимізувати витрати на збутові канали та підвищити ROI підприємств доменів Інтернет-маркетингу, фінансів, ІТ та інших на основі детальних, очищених йі оброблених первинних даних. До того ж, вона відзначається від описаних інтегрованих платформ маркетингової аналітики інтуїтивністю UI та простотою роботи з нею для користувачів будь-якого рівня технічної компетенції, оскільки більшість операцій не потребує участі ІТ-спеціалістів (тобто, є *codeless*).

Усе це зумовило популярність ПК «OWOX» серед понад 150 тисяч цифрових аналітиків у доменах Інтернет-маркетингу, фінансів, SaaS та інших ніш у США та країнах Західної і Східної Європи. Дедалі більшого розповсюдження він наразі набирає й у наших підприємств.

Українські ПК маркетингової аналітики.

При характеристиці сегмента доповнюючих програмних комплексів для МА українського виробництва слід виходити з того, що вони представлені дуже невеликою кількістю рішень. Це зумовлено тим фактом, що їх творцями зазвичай стають зовсім не ІТ-компанії, а насамперед *консалтингові агенції,* які бажають заповнити конкретну ринкову нішу на ринку. Звідси випливають основні особливості таких систем [27]:

1. ПЗ цілковито базується на можливостях популяризованих єдиних платформ маркетингової аналітики (передусім Google Analytics) і доповнює їх вихідний інструментарій несуттєвим спектром інструментів. При цьому такі доповнення зазвичай дуже обмежені і фокусуються лише на деяких ключових аспектах вибраного домену.

2. Українські продукти не розраховані на охоплення всього комплексу бізнес-процесів маркетингової аналітики: у кращому випадку вони

розширюють набір засобів *збору* первинних даних (пропонуючи поєднання каналів, які не враховуються основними платформами – напр., бізнес-телефонію) та *візуалізації* отриманих результатів клієнтам (але знову-таки за допомогою підключення сторонніх зарубіжних ПК – Power BI, Tableau або Google Data Studio – яке часто не безкоштовне). При цьому ніяких можливостей розширення інструментів *обробки* та *прогнозування* динаміки даних (напр., додавання показників користувача або нестандартних методів їх зіставлення) не передбачено.

3. Всі основні вітчизняні СМА постачаються на ринок виключно у вигляді SaaS (Software-as-a-*Service*) - продуктів, що для клієнтів означає необхідність постійних додаткових витрат на експлуатацію таких систем. Загальна сума передплати тут включає, окрім власне абонентської плати українському постачальнику послуг, повну оплату всіх сторонніх інтеграцій (з Google Analytics або Woopra, із хмарним сховищем Google Cloud, із системами статистичних розрахунків SPSS або SAS тощо) за їх тарифами, що у підсумку формує суми набагато більші, ніж при використанні закордонних СМА напряму (див. [26]).

Зазначені негативні особливості характерні для переважної кількості українських ПК маркетингової аналітики – і в першу чергу для продуктів таких компаній – лідерів нашого ринку, як «Netpeak», «Livepage» та «Ringostat». На жаль, вони не дозволяють розглядати вітчизняні продукти в якості серйозних конкурентів (чи аналогів) зарубіжних систем.

Сказаним зумовлюється загальна методологія *запропонованого нами* прототипу ПК маркетингової аналітики, в якому переваги генералізованих СМА будуть поєднані з можливістю адаптації їх функціоналу під вимоги конкретних замовників. Архітектурні та технологічні основи вирішення цього завдання розглядаються у третьому розділі нашої роботи.

Висновки до розділу 2

Проведене дослідження призводить до висновку про те, що всі системи маркетингової аналітики відзначаються наявністю багатьох схожих моментів, які можна умовно виділити, розглянувши їх у чотирьох релевантних доменах – ринкового збуту, інтернет-маркетингу, а також математики та інформатики. При такому поділі серед основних рис можна назвати такі:

1. З позиції кінцевого **користувача**, всі описані СМА покликані максимально автоматизувати виконання чотирьох основних бізнес-процесів – збору, зберігання, обробки (зведення, агрегування, сегментації), аналізу та прогнозування динаміки зміни даних. Також вони зобов'язані відображати фінальні звіти у зрозумілому та доступному користувачеві форматі, використовуючи популяризовані візуальні елементи (таблиці, графіки, діаграми та ін.).

2. З погляду **маркетингу**, будь-які СМА оперують схожим набором показників споживацької активності, який є стандартизованим (тому мало змінюваним) і дуже обмеженим. Він включає дві групи індикаторів – базові первинні метрики і певні KPI, що розраховуються на їх основі та переважно ґрунтуються на групі ROI (коефіцієнту окупності вкладень у проекти). Наголосимо, що такі системи в принципі неспроможні досліджувати *довільні* набори показників (метрик), перелік яких визначається проміжними замовниками з урахуванням конкретних вимог до аналітичного продукту.

3. З точки зору **математики**, досліджені СМА застосовують:

- детерміновані формули для розрахунку цільових (не прогнозних) маркетингових метрик та показників. Частково ці формули були наведені раніше в розділі 1.

- статистичні методи, такі як кореляційний, когортний та дисперсійний аналіз для виявлення *лінійних* закономірностей між масивами даних ([18]).

- алгоритми машинного навчання, такі як дерева рішень, випадкові ліси та градієнтний бустинг для поведінкового (тобто *нелінійного*) аналізу даних.

Вони виділяють найімовірніші результати шляхом вирішення завдань класифікації, регресії та кластеризації. Такі алгоритми здатні моделювати складні взаємозв`язки між ознаками, знаходячи їх приховані кореляції та визначаючи ключові фактори успіху/тиску маркетингових кампаній.

- моделі часових рядів, такі як (S)ARIMA та методи експоненційного згладжування для прогнозування *тенденцій у даних*. Вони також виявляють нелінійні тренди та сезонні коливання, що робить їх корисними для прогнозування різних часових показників (напр., очікуваних обсягів продажу або відсоток відтоку клієнтів) [18. С. 116].

- Байєсовські методи (мережі, регресії і т.п.) та фільтри Калмана для аналізу даних та складання прогнозів у *режимі реального часу* (тобто на основі динамічних спостережень з постійним потоком нових даних).

4. Нарешті, з погляду **програмної архітектури,** всі СМА характеризуються глибоко інтегрованою (тобто модульною) природою, існуючи у вигляді єдиної програмної платформи, проте сформованої з багатьох *окремих компонентів* (напр. менеджера збору даних, «шару» обчислювальної обробки, інструментів бізнес-аналітики та візуалізації). Така гнучкість дозволяє виробникам СМА охоплювати більший попит, пропонуючи численні інтеграції з різними постачальниками послуг, а клієнтам вільно замінювати їх на аналоги у разі потреби. Крім того, СМА є справжніми представниками індустрії *«великих даних»*, відмінністю яких є освоєння великих масивів даних для оптимізації бізнес-процесів (див. [14. С.54]). Для зберігання та обробки таких даних часто використовуються хмарні технології (об’єктні сховища та DWH), внаслідок дешевизни послуг та простоти розгортання останніх.

РОЗДІЛ 3. ПРИКЛАДНІ АСПЕКТИ РОЗРОБКИ АРХІТЕКТУРИ ПРОТОТИПУ СИСТЕМИ МАРКЕТИНГОВОЇ АНАЛІТИКИ

3.1. Постановка задачі

Як було зазначено у вступі до цієї роботи, головною задачею нашого дослідження є розробка архітектурних засад та програмної інженерії інтегрованого ПК аналітичного забезпечення збутових і фінансових операцій ІТ-підприємства, на основі систематизації недоліків наявних на ринку систем маркетингової аналітики та виявлення загальних базових вимог до таких систем. Означена систематизація, проведена у розділі 2 роботи, засвідчила, що кожна система маркетингової аналітики повинна виковувати визначені фундаментальні операції, автоматизація яких є основою вироблення ефективних збутових стратегій підприємств-замовників на вибраних ринках. Усі такі операції можна розділити на три ключових групи, згідно визначених у розділі 1 бізнес-процесів маркетингової аналітики:

Збір та збереження інформації:

1. Проведення первинного *збору* необроблених даних – включаючи визначення джерел інформації та встановлення механізмів її отримання. Важно переконатися, що зібраних даних буде цілком достатньо для формування прогнозів по вимогам замовників.
2. *Збереження* отриманих даних у постійному сховищі – сюди входить створення інфраструктури задля збереження і подальшої обробки великих наборів користувацької інформації, яка відповідає нефункціональним критеріям надійності, масштабованості, можливостей до оптимізацій, а також має механізми архівації та видалення застарілих даних.

Попередня обробка даних:

1. Здійснення *фільтрації* вхідних даних – тобто видалення чи виправлення помилкових, неповних чи неузгоджених даних. На цьому етапі проводиться аналіз присутності пропущених значень, “викидів” чи невідповідностей обраному формату. Мета етапу – забезпечити високу якість даних для подальшого аналізу, виключаючи спотворення результатів через зіпсовані дані.

2. Проведення *систематизації* проміжних даних – тобто їх класифікації по спільним ознакам задля скорочення загального обсягу оброблених даних.

Формування та представлення результатів:

1. Виконання *аналізу* оброблених даних – з метою виявлення прихованих закономірностей, трендів у порівняльній динаміці відстежуваних маркетингових показників та визначення шляхів удосконалення збутових стратегій замовників. На цьому етапі система має бути спроможною до застосування методів статистичного, економетричного аналізу та інших аналітичних підходів для вилучення корисної інформації. Отримані результати можуть бути використані для оптимізації чинних бізнес-процесів або виявлення не відкритих раніше збутових можливостей.
2. *Відображення* отриманих *результатів* – включаючи візуалізацію результатів аналізу та/або прогнозування з метою подання звітів у зрозумілій і наочній формі. На даному фінальному етапі своєї роботи система, за допомогою графіків, діаграм та інших візуальних елементів, повинна максимально полегшити кінцевим користувачам сприйняття отриманих результатів та їх трансформацію в актуальні бізнес-рішення.

Для автоматизації виконання зазначених процесів, а також їх об'єднання у єдиний "наскрізний" процес, побічною метою нашої роботи було визначено створення конвеєру даних, який буде відповідати за реалізацію безперебійного алгоритму доставки, інтеграції та обробки інформації від її первинного джерела до кінцевої візуалізації та бізнес-аналітики на основі її фінального зведення. Крім того, обов’язково треба врахувати обробку ключового фактору ризику – відмови одного з етапів конвеєру, і відповідним чином налаштувати відтворюваність всього алгоритму і його окремих кроків. Також, нам необхідно забезпечити можливість планового запуску конвеєра та його різних етапів за визначеним розкладом, що дозволить автономно запускати й виконувати поставлений «на потік» алгоритм в обумовлені із замовником проміжки часу.

Таким чином, ми можемо формалізувати основні задачі даного дослідження по побудові архітектури ПК маркетингової аналітики:

- розробити прототип інтегрованої системи маркетингової аналітики, адаптованої під запити підприємства-замовника;
- спроектувати та обґрунтувати архітектурні елементи програмної системи, забезпечивши логічний зв’язок між її компонентами;
- обґрунтувати найбільш релевантні для контексту роботи методології та парадигми обробки/інтеграції даних;
- визначити технологічний базис інструментів збереження та обробки первиної та кінцевої інформації;
- спроектувати логічну і реалізувати фізичну модель зберігання різних форматів даних;
- спланувати відповідні до функціональних вимог системи процедури предобробки початкових даних;
- спланувати окремі алгоритми, етапи та створити інтегрований конвеєр обробки і інтеграції даних;
- консолідувати описані програмні процеси у єдиний робочий процесс за допомогою системи-оркестратору;
- забезпечити відповідність етапів конвеєру даних нефункціональним вимогам системи - постійної доступності, відмовостійкості та ідемпотентності;
- забезпечити можливість задання розкладу автономних запусків конвеєра обробки даних;
- провести економетричне дослідження та оцінити адекватність математичної моделі розробленої системи та її кінцевих результатів;

Детальний опис реалізації цих вимог, а також умови їх досягнення у нашій системі наводиться у наступних параграфах даного розділу роботи.

3.2. Сучасні парадигми і підходи в управлінні великими даними

Необхідний перехід парадигми: заміна OLTP на OLAP.

В епоху глобалізації та неминучого цифрового розвитку, обсяг щодня створюваних даних збільшується з неймовірною стрімкістю. По даним сайту

“ExplodingTopics” [37], щодня генерується понад 328 терабайт (328 трлн байт) інформації. А згідно з аналізом, проведеним компанією “IBM” [40], лише за останні два роки обсяг даних, створених користувачами всесвітньої мережі, перевищив 90% від загальної кількості даних за всю попередню історію. Таке блискавичне зростання закономірно розширює виклики для підприємств усіх масштабів у сферах зберігання, обробки та аналізу даних. Це особливо справедливо для програмних розробок у домені маркетингової аналітики, що є головним об'єктом дослідження нашої роботи.

Розробку ефективної стратегії використання вхідних масивів інформації та вилучення з них реальної цінності для бізнесу необхідно розпочати з розуміння переваг і недоліків ключових парадигм обробки великих даних та обґрунтування критеріїв вибору кожної з них. Таких парадигм виділимо дві [31]: оперативна транзакційна (Online Transaction Processing, далі – OLTP) та оперативна аналітична (Online Analytical Processing, далі – OLAP).

Почнемо з OLTP. Це підхід до проектування систем обробки даних, орієнтований виконання незалежних, ізольованих операцій - транзакцій. Наразі OLTP є домінуючою парадигмою для СУБД загального призначення (напр., PostgreSQL або CockroachDB), що використовуються для широкого спектру сценаріїв. Системи такого типу зобов'язані:

- надавати механізм об'єднання та управління групою незалежних операцій як єдиною сутністю – транзакцією;
- відповідати набору вимог ACID (атомарність, узгодженість, ізоляція, надійність) [25], щоб гарантувати достовірність вхідних транзакцій;
- організовувати дані в таблиці із заздалегідь нормалізованими відношеннями для мінімізації надмірності та збереження цілісності даних;
- бути спроможними зберігати відносно великі набори даних, надаючи доступ до них у режимі реального часу;
- забезпечувати можливість обслуговування великого потоку паралельних транзакційних запитів до різних частин чи елементів загальної моделі даних.

Іншими словами, підхід OLTP націлений на підтримку процесу обробки великого числа операцій читання і запису, що потребують швидкого доступу до

невеликих обсягів даних. Системи OLTP зосереджені на забезпеченні надійності та цілісності інформації, а також швидкості виконання вказаних операцій. Вони підходять для обслуговування поточної діяльності підприємства, яка базується на комбінаціях нескладних запитів до СУБД.

Проте, коли ставиться задача зведення збережених даних для проведення аналізу чи аудитів, такі системи стикаються з рядом обмежень. Схеми баз даних OLTP роблять акцент на оптимізації операцій запису та зміни, а тому менш ефективні при виконанні складних аналітичних запитів, які потребують вкладених агрегацій та залучення великих обсягів інформації. Задля мінімізації надмірності, структура даних у таких системах часто нормалізована, що ускладнює вказані процедури. До того ж, нас мало цікавить узгодженість або ізольованість інформації, яка надходить до системи, оскільки часткового браку первинних даних уникнути не вдасться. І нарешті, СМА на базі систем OLTP навряд чи будуть використовуватись для створення багатьох аналітичних висновків одночасно – інакше, маркетологи підприємства-замовника не встигатимуть виробляти своєчасні бізнес-рішення на основі наданих аналітики та прогнозів.

Тепер розглянемо другу парадигму – OLAP. Це альтернативний підхід до реалізації систем обробки даних, зосереджений на втіленні комплексних аналітичних сценаріїв і розробці прогнозів на основі сукупних, проте структурованих даних. Важливо наголосити, що основним акцентом цієї методології є саме взаємодія з ключовими процесами інтернет-маркетингу: систематизацією, сегментацією та комбінованим аналізом даних з метою отримання цінної інформації для прийняття обґрунтованих бізнес-стратегій. Значною мірою це досягається завдяки впровадженню спеціалізованих багатовимірних моделей (т.зв. “кубів” [22]), які дозволяють ефективно переглядати одні й ті самі дані в різних розрізах. Тут варто згадати, що системи OLTP зазвичай працюють тільки в одновимірному просторі і фокусуються лише на окремих аспектах моделі даних.

У зведеному вигляді результати порівняння двох описаних парадигм обробки даних представлені в **таблиці 3.1.**

Таблиця 3.1 – Порівняльна характеристика парадигм OLAP та OLTP

№	Ознака	Парадигма OLTP	Парадигма OLAP
1	Цільове призначення	Обробка великого об'єму операційних даних у реальному часі	Обробка та багатовимірний аналіз дуже великого об'єму консолідовані даних підприємства
2	Основна спеціалізація	Короткі та швидкі CRUD операції	Довгі та повільні операції читання Рідкісні операції запису або оновлення
3	Об'єкти реалізації парадигми	Традиційні СУБД загального призначення, що слідують вимогам ACID	Сховища та озера даних (ROLAP, HOLAP) Спеціалізовані багатовимірні сховища (MOLAP)
4	Специфіка проєктування	Підлаштовується під функціональні вимоги кокретного застосунку	Підлаштовується під бізнес-вимоги суб'єктної галузі (напр., продажів або найму)
5	Одиниця роботи	Проста, атомарна транзакція	Комплексний, багатовимірний запит
6	Схематика даних	Виключно структурована Виключно нормалізована (мінімум 2 НФ, зазвичай до НФБК)	Структурована та неструктурована Переважно денормалізована (нормалізована до 1-2 НФ у моделі "Snowflake")
7	Загальний обсяг даних	Вимірюється мега- або гігабайтами	Вимірюється тера- або петабайтами
8	Механізм збору даних	Витяг інформації з традиційних СУБД	Зведення даних із різних джерел, зокрема традиційних СУБД та об'єктних сховищ
9	Підтримуваний паралелізм	Великий обсяг транзакцій	Невелика кількість запитів
10	Час обробки запитів	Вимірюється у мілісекундах	Вимірюється хвилинами або годинами
11	Узгодженність даних	Обов'язкова або остаточна (eventual)	Остаточна (eventual) або необов'язкова
12	Ізольованість даних	Обов'язково підримується (згідно ACID)	Зазвичай не підтримується

Джерело: складено автором за даними [2; 48].

Таким чином, перехід парадигми від звичної транзакційної обробки даних (OLTP) до аналітичної (OLAP) дозволяє набагато ефективніше вирішувати завдання маркетингової аналітики, що є основними об’єктами даної роботи. А це робить системи OLAP ідеальним інструментом для досягнення цілей нашого дослідження.

Фактори невідповідності традиційних БД цілям аналітики.

На користувацькому рівні під терміном “база даних” (далі – БД) розуміється будь-який інструмент зберігання й управління певним обсягом даних. Однак глибший погляд показує, що далеко не кожен подібний

інструмент є БД у такій традиційній інтерпретації. У більшості випадків саме бази даних використовуються для обліку активних даних конкретного веб-ресурсу (сайту чи додатку), критично важливих для його поточної коректної роботи. Залежно від потреб ресурсу, їх розмір може сильно варіюватися - від однієї електронної таблиці Excel чи Spreadsheet до створення цілого кластера екземплярів СУБД. Усі класичні БД схожі в тому, що хоч і можуть обробляти великі масиви даних, але лише за тривалі проміжки часу й апріорі не в межах одного запиту. Їх архітектура та функціональні можливості орієнтовані на оперативне обслуговування транзакційних даних, і не передбачають ефективної обробки великого обсягу інформації або комплексних запитів. До того ж, усі дані всередині БД змушені постійно перебувати у «бойовій» готовності, через що дуже важко пріоритизувати один набір даних над іншим, і, як наслідок, виділити окремі ресурси (напр., вузол кластеру) для обслуговування його індивідуальних потреб. Через ці та багато інших факторів, процес горизонтального масштабування баз даних надзвичайно складний [30] і вимагає високої технічної компетенції IT-відділу підприємства. Отже, використання традиційних (SQL чи NoSQL) баз даних як «ядра» системи маркетингової аналітики ми вважаємо не ефективним рішенням як для її компанії-виробника ПЗ, так і підприємства-замовника.

На сучасному ринку існує безліч альтернативних пропозицій, тією чи іншою мірою покликаних вирішити зазначені проблемі. Серед них варто виділити окремий тип систем, що істинно слідують парадигмі OLAP. Вони отримали назву «сховища даних» (*Data Warehouse,* скорочено *DWH*).

Теоретично, Сховище Даних (СД) – це централізований репозиторій, який поєднує та зберігає інформацію з багатьох розрізнених джерел у корпоративних межах підприємства. Концептуально, СД можна вважати надбудовою поверх одного або кількох джерел OLTP, що використовує їх вміст для реалізації різних сценаріїв суб’єктних областей (у т.ч. маркетингової аналітики). Таке представлення формує загальне уявлення про архітектуру сучасних СД [49], що включає три взаємопов’язані рівні (див. рисунок 3.1):

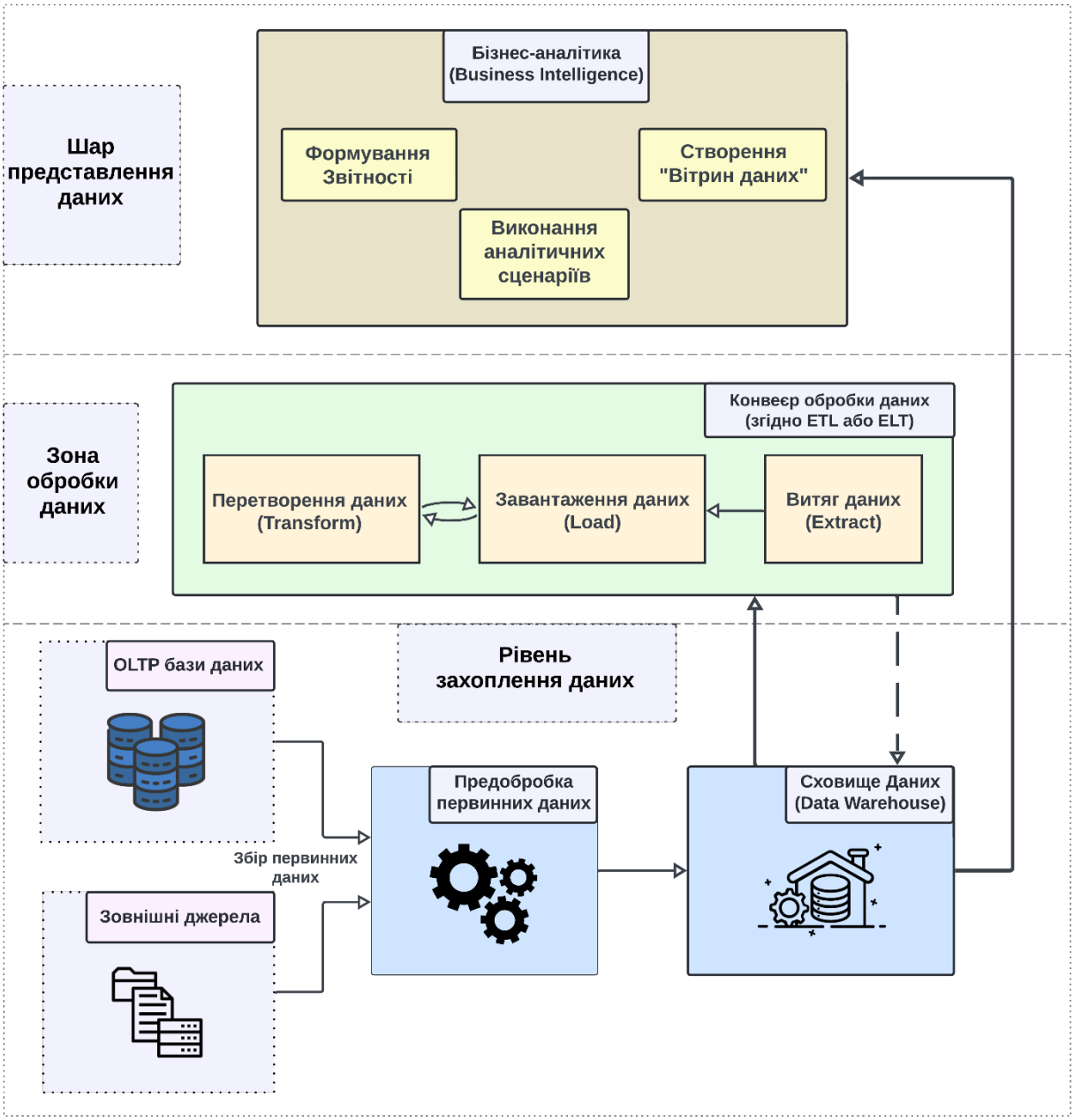


Рисунок 3.1 – Три архітектурні рівні систем типу «сховищ даних»

Джерело: побудовано автором за даними [49].

- Рівень **захоплення даних** *(прикладний)***.** Тут збираються «сирі» (необроблені) дані, зведені з різних підконтрольних підприємству джерел, переважно систем OLTP чи зовнішніх інфраструктурних інтеграцій. Зібрані дані зберігаються або у СД, або попередньо відправляються до підготовчої зони для подальших перетворень.
- Зона **обробки даних** *(посередницька)***.** Цей рівень можна вважати посередником між початковими даними та кінцевим користувачем, який

використовує їх в цілях OLAP. На цьому етапі первинна інформація проходить через процедури обробки даних (тобто конвеєр), зазнаючи необхідних для створення структурованих даних змін. Після цього удосконалені дані можуть бути використані на наступному рівні.

- **Шар представлення даних** *(клієнтський).* Це найвищий з точки зору кількості абстракцій рівень архітектури СД, у якому підготовлені дані подаються кінцевим користувачам у зручній і зрозумілій їм формі. Саме цей етап становить найбільшу цінність для інтернет-маркетингу, оскільки тут створюються зрізи (відомі як “вітрини” [51]) даних, виконуються аналітичні запити і складаються релевантні звіти. Тут часто залучаються різні інструменти бізнес-аналітики (Business Intelligence, BI), що дозволяє конвертувати програмні результати у формат графіків, діаграм та інших візуальних компонентів, спрощуючи формування бізнес-стратегій.

Як і класичним базам даних, СД властива структурованість – всі дані у ньому мають відповідати певній схематиці, тобто мати строго заданий набір атрибутів. Такий архітектурний підхід нагадує реляційний, де дані поділяються на таблиці, а всередині них – на рядки та стовпці. Однак тут існує й ключова відмінність: якщо в системах OLTP дані завжди згруповані за рядками, то в СД вони організовані згідно з характеристиками загальної структури даних – за стовпцями. Через це дані всередині сховища зберігаються не у рядковому, а у стовпчастому форматі. Такий підхід суттєво полегшує аналітичну обробку великих масивів структурованих даних, оскільки це часто потребує знаходження глобальних (відносно усього набору) кореляцій між тими чи іншими атрибутами моделі даних.

На відміну від БД, які зберігають лише обмежений спектр даних підприємства, СД завжди використовуються з метою зберігання інформації про його всебічну діяльність, як у поточному часі, так і в минулому (історичні дані) і майбутньому (прогнозні дані). Вони від початку призначені для обліку величезної кількості гетерогенних даних, і тому повинні мати високу здатність до масштабування, як у компоненті зберігання даних, так і в обчислювальній компоненті для обробки складних запитів. Ця особливість сховищ даних

задовольняє базовим вимогами будь-якої системи МА, і тому робить їх найкращим варіантом джерела даних для останньої.

Таблиця 3.2– Порівняльна характеристика баз даних (БД) і сховищ даних (СД)

№	Ознаки	База Даних	Сховище даних
1	Основне призначення	Обслуговування поточної діяльності конкретного активу (ресурсу або відділу) підприємства	Проведення аналітики на великих обсягах даних задля вироблення стратегічних бізнес-рішень
2	Кінцеві користувачі	Розробники ПЗ Інженери/аналітики даних	Маркетингові аналітики Топ-менеджмент підприємства
3	Модель споживання	Самостійне розгортання або PaaS	Переважно SaaS, рідше PaaS
4	Схематика даних	Структурована Нормалізована Одновимірна	Переважно структурована Переважно денормалізована Багатовимірна
5	Формат зберігання	Переважно рядковий	Стовпчастий
6	Основний вид робіт	OLTP (транзакційні) навантаження	OLAP (аналітичні) навантаження
7	Переважний тип запитів	Нескладні транзакційні	Комплексні аналітичні
8	Термін зберігання даних	Зберігає лише операційну інформацію	Зберігає історичні та поточні дані
9	Релевантність даних	Усі дані доступні у режимі реального часу	Дані оновлюються коли і якщо необхідно
10	Інтегрованість	Збирає дані з одного або декількох програмних ресурсів	Збирає дані з багатьох різних джерел
11	Масштабування	Переважно вертикальне	Переважно горизонтальне

Джерело: складено автором за даними [49].

У якості висновку, формалізуємо **сім ключових відмінностей** між Базою Даних (БД) та Сховищем Даних (СД):

- Оскільки традиційні БД спроектовані обслуговування операційної діяльності конкретного активу підприємства, вони заточені під виконання високошвидкісних операцій над невеликими обсягами даних у режимі реального часу. При цьому, БД дозволяють одночасно обробляти велику кількість таких операцій, наприклад додавання, часткового читання, оновлення або видалення записів (також відомих як CRUD - Create, Read, Update, Delete). Однак, якщо з'являється потреба у складніших сценаріях, особливо тих, що вимагають залучення даних з кількох незв'язаних таблиць,

їх ефективна реалізація стає значною проблемою. Сховище даних, навпаки, призначене насамперед для обслуговування аналітичних потреб, надаючи користувачам можливість виконувати складні багатофакторні запити, потребуючих доступу до величезних обсягів даних.

- БД можуть обробляти безліч (десятки тисяч) паралельних запитів від різних користувачів і до різних частин набору даних, тоді як СД заточені на виконання кількох аналітичних запитів за одиницю часу. Це зв’язано з тим, що обробка складних запитів потребує великих обчислювальних ресурсів. І хоча час відгуку залишається важливим показником, більш вагомою проблемою для СД є якість виконуваного ним аналізу.
- Завдяки зазначеній вище властивості, класичні БД є основою систем транзакційної обробки даних (парадигма OLTP), тоді як СД краще задовольняють вимоги аналітичної обробки даних (парадигма OLAP).
- Традиційні БД намагаються максимально оптимізувати процедуру зберігання інформації, що включає нормалізацію структури даних для виключення дублікатів, аномалій у даних і зайвих транзитивних залежностей. На відміну від цього, СД приділяють меншу увагу такому процесу, в основному зосереджуючись на забезпеченні ефективності аналітичних запитів. Це дозволяє їм використовувати денормалізовану схему, поєднуючи множини зв'язаних таблиць в одну централізовану, що спрощує виконання комбінованих аналітичних запитів.
- Через необхідність нормалізації відносин усередині схематики баз даних, вони не можуть використовуватися для складання комплексних запитів, які включають дані з великої кількості різних таблиць. У той же час, єдина (денормалізована) схема сховищ даних дозволяє їм здійснювати вкладені запити до великого числа характеристик без особливих зусиль.
- БД зазвичай містять лише операційну інформацію про діяльність підприємства чи конкретного ІТ-продукту, що ускладнює аналіз змін динаміки даних у часі - особливо, якщо не впроваджено процеси резервного копіювання та архівування. На відміну від БД, сховища даних зберігають усю коли-небудь отриману інформацію, що дозволяє проводити комплексну аналітику за

значно більший проміжок часу.

- Традиційні БД можуть масштабуватися вертикально (шляхом збільшення ресурсів одному сервері), але процес їх горизонтального масштабування є досить складним [30]. Сховища даних набагато легше масштабуються горизонтально завдяки своїй здатності ефективно обробляти великі обсяги даних та аналітичні запити на розподіленій інфраструктурі.

Необхідність у моделі “Data Warehouse as a Service”.

Так само, як і з будь-якою іншою технологією, при виборі відповідного функціональним вимогам програмного комплексу сховища даних, архітектору ПК слід визначитися з єдиним форматом його розгортання і використання. Так, можна або використовувати закриті (пропрієтарні), але добре підтримувані SaaS-сервіси з широким спектром запропонованого функціоналу, або покладатися на менш конкурентоспроможні рішення з відкритим кодом, що дозволяє розгорнути їх на власних серверах та доопрацювати їх під потреби конкретного проекту.

Ми вважаємо, що вибір на користь існуючих DWH-рішень (Data Warehouse as a Service, DWaaS) має ряд переваг перед останнім варіантом. Ключовим з них є їх постійна готовність до використання, оскільки підприємство-орендар отримує доступ до вже налаштованої і повністю функціонуючої інфраструктури, таким чином заощаджуючи час на її початкове розгортання. Наприклад, такі платформи (R)OLAP, як Google BigQuery або Amazon Redshift, відпочатку надають готові інструменти завантаження, зберігання та аналізу даних, які дозволяють маркетологам зосередитись на підготовці первинних даних, а інженерам – на розробці решти компонентів ПК. Такий висновок підтверджується дослідженням центру “Dimensional Research” у 2020 р. [36], яке виявило, що більше 96% опитаних підприємств використовують хмарні OLAP сховища даних.

Ще однією сильною стороною моделі DWaaS є вроджені здібності до високої масштабованості та неймовірної гнучкості використання, що зумовлені використанням хмарних ресурсів. Більшість таких рішень реалізовано на основі безсерверних обчислень, що дозволяє автоматично змінювати обсяги

обчислювальних ресурсів та ресурсів зберігання залежно від поточних потреб проекту. Так, якщо використовуваних обсяг даних починає зростати, можна легко масштабувати систему горизонтально, додаючи обчислювальні вузли або розширюючи обсяг сховища без необхідності внесення прямих змін до архітектури системи. І навпаки, якщо система не використовується за призначенням, кількість вузлів можна зменшити, залишивши лише необхідний мінімум для збереження завантажених у сховище даних. До того ж, DWaaS-рішення зазвичай надають широкий спектр інструментів для аналізу даних та подальшої візуалізації результатів. Вони можуть включати інтегровані засоби для створення звітів, діаграм та інших складових інфографіки, що робить процес аналізу більш зручним і ефективним як для проміжних користувачів, так і для кінцевих споживачів.

Проте, незважаючи на численні переваги, формат DWaaS має й певні недоліки. Головним є високі витрати на оренду СД, що однаково стосується як інноваційних стартапів з обмеженим бюджетом, так і корпорацій-гігантів з великими обсягами даних. Крім того, у разі виникнення проблем із обслуговуванням або зміною цінової політики постачальника, абсолютна залежність від його послуг може стати серйозною перешкодою для аналізу роботи суб’єктних областей підприємства-замовника. На відміну від DWaaS, сховища даних з відкритим вихідним кодом (Open Source Data Warehouse, OSDW) пропонують велику гнучкість та контроль над інфраструктурою та даними. Вони дозволяють налаштовувати систему під свої унікальні потреби та змінювати її відповідно до змін у бізнес-вимогах. Наприклад, використання таких інструментів, як Hadoop або Spark, дозволяє створювати високопродуктивні і масштабовані системи аналітики даних із контрольованими витратами. Проте підхід OSDW також має недоліки. Головний з них полягає у необхідності великих витрат часу та інших ресурсів на розробку та підтримку релевантної інфраструктури. На відміну від готових DWaaS-рішень, що надаються вже сконфігурованими, ефективне використання OSDW вимагає участі висококваліфікованих ІТ-фахівців для розгортання, налаштування та подальшого обслуговування системи. Також треба

відзначити, що OSDW не мають такого ж рівня стабільності, як пропрієтарні DWH сервіси. Незважаючи на те, що багато з них мають активну спільноту розробників та регулярно оновлюються, залишається ризик виникнення помилок або проблем сумісності між різними компонентами системи, що також потребує постійної наявності IT-фахівців у вільному доступі.

Таким чином, у контексті нашого дослідження використання підходу Data Warehouse as a Service є більш доцільним, оскільки такі рішення мають готову до використання інфраструктуру, гарантують високу надійність збереженої інформації та надають широкий спектр інструментів для аналізу даних. Це дозволяє зосередитися на розробці програмного комплексу маркетингової аналітики та аналізі даних, оминаючи складності мануального налаштування та підтримки цільового сховища даних.

Порівняльний аналіз доступних DWaaS-рішень

Після визначення підсумкового формату використання сховищ даних, необхідно провести порівняльний аналіз провідних пропозицій ринку, а саме, DWaaS-рішень від найпопулярніших хмарних платформ: Google Cloud Platform та Google BigQuery, Amazon Web Services та Amazon Redshift, Microsoft Azure та Azure Synapse Analytics, а також Snowflake.

Спочатку розглянемо їх загальні риси, серед яких варто виділити 5 основних:

- Архітектурний підхід**.** Тут можна назвати наступне:
  - Майже всі ці рішення використовують можливості власної хмарної платформи для створення необхідної інфраструктури. Виняток становить Snowflake, який може бути розгорнутий на будь-якій із трьох хмар і вільно перенесений між ними.
  - Усі зазначені рішення використовують неструктуровані об’єктні сховища [7] для зберігання завантажених у СД даних. Для BigQuery це Google Cloud Storage, Redshift – S3, Synapse Analytics – Azure Blob Storage, а Snowflake використовує можливості хмари, на якій розгорнуто його інфра-структуру. Це забезпечує необмежений ресурс компоненти «зберігання» у хмарі, а також дозволяє суттєво знизити грошові витрати, налаштувавши переведення рідко

використовуваних даних у "холодний" (архівний) режим.

- Для забезпечення максимальної ефективності виконуваних операцій, обчислювальна компонента всіх зазначених рішень покладається на використання масово-паралельної архітектури (massively parallel processing, MPP) [41]. Такий підхід дозволяє легко додавати або видаляти додаткові віртуальні вузли за поточною потребою, кожен з яких оброблятиме свою частину загального обсягу робіт. Таким чином, підприємству-орендарю не потрібно підтримувати власну обчислювальну інфраструктуру.

- Можливості розгортання**.** Залежно від функціональних вимог ресурсу або конфіденційності даних, що використовуються в ньому, орендарю може бути необхідно розгорнути його на власних фізичних (т.зв. модель “on-premise”) або підконтрольних віртуальних серверах. Більшість з розглянутих рішень підтримують обидва варіанти (тільки Google BigQuery не підтримує мануальне керування серверною інфраструктурою).

- Підтримка стандартизованої мови запитів**.** Оскільки всі СД мають структуровану модель даних, цілком логічно використовувати структуровану мову запитів SQL для виконання усіх внутрішніх операцій. Так, абсолютно всі DWaaS-рішення надають підтримку останніх стандартів цієї мови (відповідно до ANSI SQL), додаючи лише специфічні діалектні доповнення.

- Підтримка сторонніх інтеграцій**.** Представлені рішення мають безліч вбудованих підключень до зовнішніх IaaS і PaaS сервісів, що обслуговують сфери потокового перенесення даних, бізнес-аналітики та інші. Платформи відрізняються лише кількістю та якістю цих інтеграцій.

- Повна відповідність нормам безпеки**.** Окрім задоволення провідних стандартів безпеки, ці платформи забезпечують автоматичне шифрування даних, а також дозволяють встановлювати користувацькі політики доступу до створених моделей або їх атрибутів.

Таким чином можна зробити висновок, що більшість СД мають фундаментально схожу архітектуру і оперують однаковими концепціями для надання послуги DWaaS. Відмінності між ними радше полягають у нефункціональних, ніж функціональних вимогах, а також фінансових

витратах на їх експлуатацію.

Тепер перейдемо до опису переваг і недоліків кожного DWaaS-рішення:

1. Google BigQuery (GBQ) суто побудований поверх безсерверної архітектури, яка при цьому повністю розділяє компоненти зберігання та обчислення. Ця ознака дає GBQ можливість вільно масштабуватись горизонтально в межах однієї проекції, що дозволяє ефективно працювати з непрогнозованими обсягами даних. Обчислювальна архітектура GBQ заснована на MPP-механізмі "Dremel" [44]. Він дозволяє обробляти великі обсяги записів різного рівня вкладеності за малі проміжки часу. Зібрані дані зберігаються у розподілених, реплікованих сховищах (керованих двигуном "Colossus" [44]) для гарантії надійності даних. Конкурентною перевагою GBQ є фокус на виконанні запитів до неймовірно великих обсягів даних (що зрозуміло з назви: “BigQuery” – “великий запит”), через що платформа менше акцентує увагу на процесі збору даних. Ціновий план GBQ включає доволі щедрий безкоштовний рівень, доступний новим користувачам хмарної платформи Google. Поділ вузлів зберігання та обчислення дозволяє платити за обробку даних "на льоту", не вносячи фіксовану передоплату.

2. Amazon Redshift (ARS) надає 2 основних формати розгортання - безсерверне або самостійне, що часто вимагає від кінцевих користувачів значного досвіду роботи з мовою SQL і знання загальної структури сховищ даних. Крім того, користувач є відповідальним за початкове налаштування кластера та керування його вузлами, що ще підвищує вимоги до його компетенції у хмарних технологіях. Обчислювальна компонента ARS також заснована на MPP-архітектурі, де єдиний серверний кластер розбивається на деяку кількість вузлів, кожен з яких додатково розділяється на частини. Дані, що використовуються часто, зберігаються в розробленій корпорацією Amazon версією РСУБД PostgreSQL, в той час як рідко використовуються відправляються в економне об'єктне сховище S3. Основними конкурентними перевагами ARS є повна підтримка хмарної екосистеми AWS та одночасне обслуговування великої кількості аналітичних потреб. Ціновий план ARS не включає безкоштовного рівня, натомість встановлюючи фіксовану погодинну

ставку згідно з конфігурацією кластера.

3. Microsoft Azure Synapse Analytics (MASA) надає і хмарне безсерверне, і самостійне розгортання. Тому, подібно RSA, вимагає від кінцевого користувача мануальної конфігурації, зокрема виділення ресурсів і налаштування багатьох ключових процедур. Обчислювальна компонента MASA також ґрунтується на розподіленій MPP-архітектурі, де головний вузол (т.зв. “control unit”) виступає у ролі балансувальника навантаження, і рівномірно розподіляє обсяг запланованих робіт між дочірніми вузлами. Дані зберігаються в розподіленому сховищі нового покоління Data Lake Storage *Gen2*, побудованому на базі об`єктного сховища *Azure Blob Storage* (перше покоління ПЗ використовувало менш ефективну файлову систему Hadoop). Основними чинниками успіху є спрощена взаємодія з сімейством продуктів Microsoft та велика кількість вбудованих підключень до сторонніх інтеграцій. Цінова політика MASA схожа на ARS, проте на відміну від неї включає безкоштовний рівень.

4. Snowflake – надає виняткову гнучкість у розгортанні, дозволяючи обрати будь-яку популярну платформу розгортання (є хмарно-агностичним) та формат використання, з акцентом на безсерверний. На відміну від пропозицій-аналогів Amazon і Microsoft, це рішення не вимагає попереднього налаштування, проводячи його автоматично. Як і GBQ, Snowflake розділяє процедури зберігання даних та виконання фактичних запитів, що дає ті ж переваги в обробці даних. Усе це забезпечує високу зручність використання, дозволяючи працювати зі Snowflake підприємствам з обмеженими кадровими ресурсами – що й є основним конкурентним фактором платформи. Проте, у цього ПЗ є й кілька суттєвих недоліків. Головним вважають досить високу вартість послуг, особливо компоненти «зберігання». Також, оскільки Snowflake не залежить від якогось конкретного хмарного оточення, воно де-факто менш інтегроване із його екосистемою, ніж нативні послуги.

Таблиця 3.3 – Порівняльна характеристика описаних DWaaS-рішень

№	Ознаки	Google BigQuery	Amazon Redshift	Azure Synapse	Snowflake
1	Формат розгортання	Виключно безсерверний	Переважно самостійний Можливий безсерверний	Переважно самостійний Можливий безсерверний	Переважно безсерверний
2	Обчислювальна архітектура	**Масово-паралельна (MPP)**			
3	Контроль масштабування	Здійснюється автоматично	Переважно мануальний Можливе автоматичний (після налаштування)	Переважно мануальний Можливе автоматичний (після налаштування)	Здійснюється мануально або автоматично згідно політиці масштабування
4	Розділення зберігання та обчислення	Присутнє	Присутнє, але тільки для обч. вузлів типу "RA-3"	Відсутнє	Присутнє
5	Продуктивність	Ідеальна для великих наборів гетерогенних даних та аналітики у реальному часі	Ідеальна для великих наборів гомогених даних. Неефективна для малих наборів даних.	Посередня для великих та малих обсягів даних	Відмінна для великих та малих обсягів даних
6	Формат збереження даних	**Стовпчастий**			
7	Підтримка стандарту SQL	Присутня (ANSI SQL:2011)	Присутня (ANSI SQL:2008)	Присутня (ANSI SQL:2011)	Присутня (ANSI SQL:2016)
8	Діалекти мови SQL	GoogleSQL та Legacy SQL	Amazon Redshift SQL	Transact-SQL (T-SQL)	Snowflake SQL
9	Ціновий план	Фіксована плата за фактичний ГБ	Погодинна ставка або бронювання ресурсів	Погодинна ставка або бронювання ресурсів	Фіксована плата за фактичний ГБ
10	Безкоштовний рівень	Присутній	Відсутній	Присутній	Присутній
11	Модель споживання	**PaaS**			SaaS (з можливістю PaaS)

Джерело: складено автором за даними [6].

Таким чином, наш аналіз показав, що за критеріями простоти використання та інтеграції, наявності безкоштовного рівня використання, а також необхідності використання екосистеми хмарної платформи Google (причини будуть розглянуті далі), вибір DWaaS-рішення *Google BigQuery* є оптимальним для створення прототипу ПК маркетингової аналітики.

3.3. Вибір технологічного базису оркестрації робочих процесів

3.3. Підходи до управління «великими даними»: ETL vs ELT

На першому етапі розробки програмної системи маркетингової аналітики ми визначилися з її архітектурною основою – сховищем даних, а також обґрунтували вибір нетипової парадигми обробки даних (OLAP), конкретного рішення СД (Google BigQuery) та його формату використання (DWaaS). Тепер слід перейти до прототипування *конвеєра обробки даних.*

Почати необхідно з розгляду передових підходів до управління даними всередині СД, а саме до комбінованого процесу збору, обробки і відправлення даних, та виявити їх сильні і слабкі сторони. Так, на сьогоднішній день тут використовуються два інтеграційних підходи [54]: «вилучення–перетворення–завантаження» (Extract-Transform-Load) і «вилучення–завантаження–перетворення» (Extract-Load-Transform).

Extract-Transform-Load (ETL) – це класична методологія управління «великими даними», яка передбачає виконання трьох основних процесів [42] у строго заданому порядку:

1. Збір (вилучення) первинних даних із різних джерел, переважно OLTP баз даних, ERP чи CRM систем та підготовлених фахівцями-маркетологами наборів даних. По отриманні, дані або відправляються у постійне сховище, або тимчасово залишаються в оперативній пам'яті активного серверу.

2. Зміна (перетворення) і консолідація зібраних даних перед їх відправкою до системи зберігання. Цей процес забезпечує сумісність первинних даних з цільовим СД, який о вимагає від них дотримання певної структури.

3. Збереження (завантаження) перетворених даних у цільове сховище. Їх доставляють або частично, пакетами певного обсягу («пакетний» режим), або поступово, наборами нефіксованого розміру («потоковий» режим).

Цей підхід був популяризований в епоху глобалізації Інтернету (1980-1990 рр.) разом із появою OLAP [48] сховищ даних. Його фокус здебіль-шого зосереджений на посередницькому процесі перетворення даних, тоді як інші кроки виконують ролі «адресанта» (відправника початкової) й «адресата» (отримувача кінцевої) інформації. Це можна пояснити двома факторами впливу, пов’язаними з періодом розвитку ETL:

- зберігання неоптимізованих та/або неструктурованих наборів даних на той момент часу було дуже дорогим, оскільки хмарні платформи ще не отримали належного розвитку. Так, сучасні хмарні гіганти на кшталт Amazon Web Services (березень 2006), Google Cloud Platform (квітень 2008), Alibaba Cloud (вересень 2009) та Microsoft Azure (лютий 2010) були створені лише у другій половині двохтисячних років [34].

• ринок інтернет-маркетингу лише зароджувався і поки ще не став [45] домінантною сферою продажів. Підприємства на електронному ринку не зазнавали високої конкуренції, і не були змушені постійно гнатися за останніми інноваціями і трендами. І тому швидкість виконання обчислювальних навантажень для них була менш важлива, ніж потенціальна економія на зберіганні даних.

Хоча багато фахівців вважають цей підхід застарілим, він залишається популярним і на сьогоднішній день, що зумовлено цілою низкою його переваг. Так, гнучкість процесу трансформації даних дозволяє змінювати етапи обробки відповідно до поточних вимог замовників. Наявність етапу попередньої обробки гарантує, що аналіз враховуватиме лише високоякісні дані, а також зменшує витрати на зберігання первинної інформації. Крім того, на ринку існує велике розмаїття ETL-інструментів, що дозволяє користувачу вибирати потрібний варіант для конкретних завдань. А за даними центру маркетологічних досліджень «VM Reports», зібраними у лютому 2024 р., з кожним роком цей ринок зростає на 14.6%, і має досягнути обсягу 10.5 млрд. дол. упродовж 2030 р. [52].

Однак, методологія ETL не позбавлена явних недоліків. Основним ми виділимо низький ступінь абстрагування і відсутність автоматизації, що потребує ретельного планування всіх його етапів. Це робить впровадження, оновлення та зміну стратегії ETL складними та трудомісткими операціями. Згідно з опитуванням, проведеним серед інженерів великих компаній дослідницьким центром «Wakefield Research» у 2021 р. [53], у середньому вони витрачають 44% свого робочого часу лише на підтримку ETL-процесів, що призводить до втрати більше 0,5 млн. дол. на рік. Окрім цього, через неминучу затримку між вилученням даних та їх кінцевим завантаженням, зібрані дані не будуть доступні у сховищі одразу, через що неможливо дослідити їх первинні зрізи. Це підтверджується ще одним опитуванням, проведеним серед аналітиків даних компанією «Dimensional Research» у 2020 р. [36], 86% респондентів якого відмітили, що завжди змушені працювати з сильно застарілими даними (41% вказали віком таких даних 2 місяці). Таким

чином, ETL є гарним вибором у разі неординарних вимог до процедури перетворення, необхідності інтеграції гетерогенних форматів даних, а також бажання мати більший контроль за етапами виконуваних процесів.

Альтернативним та більш сучасним підходом є Extract-Load-Transform (ELT). Він базується на ETL і складається з тих самих кроків, проте представлених в іншому порядку: дані спочатку вилучаються з початкових джерел, а потім завантажуються безпосередньо в СД без проміжних операцій обробки. І лише після завершення цих двох процесів, до даних усередині сховища застосовується процедура перетворення. На відміну від ETL, цей підхід менш акцентований на обробці даних, віддаючи перевагу швидкій доставці первинної інформації до централізованого сховища.

Масове зростання показника освоєння ELT пов’язане з початком поширення (й агресивного просування) хмарних технологій [48], які пропонували необмежений і дешевий ресурс для зберігання даних і проведення обчислень будь-якого масштабу. Це дозволило перекласти відповідальність за процес перетворення на СД, використовуючи його серверні потужності не тільки для виконання складних запитів, але й для проведення обробки даних. Незабаром більшість DWaaS-рішень у хмарі стали пропонувати це як послугу, яка не потребує додаткової інфраструктури (проміжного ETL-сервера) для виконання перетворень даних, спираючись лише на власні ресурси або ресурси оточуючого віртуального середовища.

Таким чином, ELT передбачає перенесення обчислювально-складних процесів безпосередньо на рівень сховища даних (тобто підконтрольних йому серверів), у той час як ETL розгортання цих серверів є відповідальністю інженера – повною або частковою (у разі залучення допоміжної інтеграції). Хоча й на папері ця ознака і є перевагою, фактичний потенціал такого процесу перетворення повністю залежатиме від вбудованих можливостей обраного СД. Це породжує головний недолік підходу – створення трансформацій найчастіше можливе лише за допомогою мови SQL, і лише в деяких випадках за допомогою гнучкішого програмного коду (напр. на основі фреймворку обробки даних Apache Spark), що сильно ускладнює зведення різних форматів

даних (напр., JSON, Protobuf та XML). Нівелювати цей недолік вдається тільки тоді, коли у розроблюваного продукту немає особливих вимог до процесу, і можливості SQL можуть повністю покрити його вимоги. Крім того, оскільки дані не проходять етап передобробки, вони потрапляють до сховища у неструктурованому та неоптимізованому вигляді, що вимагає від СД підтримки «флюїдної» схеми даних (тобто відсутності критерію структурованості). Тому підхід виявляє свої сильні сторони при поєднанні з «озером даних» (Data Lake [51]), яке, на відміну від СД, від початку спроектовано для підтримки неструктурованих наборів даних.

Проведений аналіз дозволяє однозначно стверджувати, що підхід ELT не є прямою заміною ETL, а скоріше необхідним доповненням для ситуацій обробки величезних обсягів гомогенних даних, де простота використання і висока швидкість міграції даних є основними нефункціональними вимогами системи. З цієї причини ми не включили до матеріалів роботи порівняльну характеристику ETL та ELT.

При розробці системи маркетингової аналітики нами будуть використані обидві парадигми – ELT для лінійних перетворень даних, а ETL для нелінійних трансформацій, які вимагають більшого мануального контролю.

Вибір диспетчера задач та оркестратора робочих процесів.

Наступним етапом вироблення загальної концепції архітектури ПК маркетингової аналітики є розгляд доступних способів управління конвеєром обробки даних. Його ціллю є максимальна автоматизація розробки та підтримки даного процесу. Для цього знадобиться інструмент, який не тільки вміє запускати та змінювати конкретні програмні операції, але й може звести їх воєдино, керуючи ними як однією цілісною сутністю. Крім того, він повинен враховувати заданий порядок виконання операцій, а також коректно представляти залежності між ними. Нами було встановлено, що найбільш повно зазначеним критеріям відповідає група диспетчерів задач (ДЗ), основною метою яких є забезпечення ефективної та простої у використанні оркестрації бізнес-завдань, а також їх залучення у єдиний робочий процес. Але перед тим, як перейти до характеристики конкретних програмних рішень цієї

категорії, необхідно розібратися із частими непорозуміннями навколо ДЗ, та зрозуміти, чому багато з них не можуть бути використані при розробці нашої системи. Для цього, нам необхідно зробити історичний екскурс до часів започаткування індустрії «великих даних».

Період стрімкого зростання та масового прийняття цієї індустрії можна віднести до середини 2000 рр. [43. P. 6-9], коли загальний обсяг генерованих даних почав збільшуватися в експоненційному масштабі, а підприємства стали потребувати заходів для їх ефективного зберігання та обробки. Зокрема, в той період американський інноваційний стартап "Google" поставив перед собою вкрай амбітне завдання - створити універсальну пошукову систему [33], здатну "приборкати" міць постійно розширюваної всесвітньої мережі, та поставити усі інтернет-ресурси на внутрішній облік. На той момент, вирішити таке завдання на базі існуючих технологій було неможливо. Тому компанія провела безліч архітектурних та програмних досліджень, незабаром представивши дві переломні для індустрії наукові роботи: «The Google File System» (2003 р.) та «MapReduce: Simplified Data Processing on Large Clusters» (2004 р.), що започаткували концепцію розподіленого зберігання даних та комплексних обчислювальних навантажень. Ці ж монографії пізніше мали величезний вплив на розвиток хмарних платформ, включаючи створення сервісів EMR (Elastic MapReduce) і S3 (Simple Storage Service) від Amazon AWS, а також ABS (Azure Blob Storage) від Microsoft Azure.

На основі робіт Google, у 2005 р. було створено перший фреймворк для управління процесами зберігання та обробки великих даних у розподіленому серверному оточенні – Hadoop. Цей інструмент мав власну файлову систему HDFS (Hadoop Distributed File System), дані якої використовувалися процесами обробки даних Map (проектування), Shuffle (розподіл) і Reduce (зведення), ресурси на виконання яких виділяв кластерний менеджер YARN (Yet Another Resource Negotiator) [38]. Оскільки Hadoop надавав раніше не освоєний спосіб інтерпретації «великих даних», він швидко здобув популярність серед підприємств середнього та великого бізнесу, що могли дозволити собі розгорнути його на локальній інфраструктурі. Окрім іншого,

це допомогло їм істотно підвищити точність аналізу та прогнозу проведених рекламних кампаній, що зумовлювало їх впевнене закріплення на вибраних ринках (гарними прикладами є компанії Pinterest [47] та Spotify [55]).

Таким чином, усі подальші сценарії «великих даних» були неминучо пов'язані з екосистемою Hadoop, через що більшість існуючих ДЗ призначені виключно для управління великими наборами Hadoop-навантажень. Нами були проаналізовані: "Nifi" (створений у 2006 р. Агентством національної Безпеки США), "Oozie" (відкритий у 2007 р. компанією «Yahoo!») та "Azkaban" (представлений у 2010 р. компанією «LinkedIn»). Ці та подібні до них інструменти виявились мало пристосованими для вирішення завдань за межами Hadoop, внаслідок чого не можуть бути використані в нашій роботі.

Із плином часу, у галузі «великих даних» з'являлись і не пов'язані з Hadoop інструменти. Однак, їх не можна було інтегрувати в чинні робочі процеси, оскільки ДЗ того періоду не могли керувати чимось окрім Hadoop завдань. Тому вже на початку 2010 рр. були здійснені перші спроби узагальнення процесу оркестрації, які прагнули зробити його незалежним від використовуваних технологій. Нами досліджені 3 системи такого типу: Airflow, Prefect та Luigi. Оскільки основною метою даної роботи є побудова конвеєра обробки даних, ми зробили вибір на користь самостійного розгортання ДЗ на керованій інфраструктурі, не розглядаючи при цьому готові PaaS-рішення провідних хмарних платформ. З цієї ж причини об'єктами нашого вивчення є лише системи з відкритим кодом, вільно розповсюджувані по ліцензії "Apache 2.0".

Airflow був спочатку розроблений компанією Airbnb у 2014 р. як внутрішній проект для відстеження й автоматизації багатоступеневого процесу бронювання нерухомості. Роком пізніше проект був викладений у відкритий доступ під ліцензією вільного користування, а в 2016 р. його було вирішено передати фонду – куратору відкритого ПЗ "Apache Software Foundation", під егідою якого він продовжує свій поточний розвиток. Хоча Airflow не є першим оркестратором робочих процесів («Luigi» та «Pinball» робили спроби ще у 2012 р.), він найпершим запропонував концепцію створення повномасштабних процесів у вигляді коду (т.зв. Workflow as Code),

включаючи задання залежностей, параметризацію оточення та планування запуску задач за розкладом. Таким чином він став піонером у галузі, створивши фреймворк для спрощеного управління будь-якими процесами “великих даних” на основі мови Python. Наразі цей інструмент є абсолютним лідером ринку, надаючи багатий функціонал і набір зовнішніх інтеграцій, зручний графічний інтерфейс (включаючи календарне та графове представлення), а також хорошу здатність до масштабування за рахунок модульної архітектури. Однак, Airflow може викликати труднощі у наступних аспектах [39. P. 43]:

- Початкове опанування – оскільки він вимагає глибокого знання мови Python та основ Дискретної математики, а також обізнаності про свою внутрішню архітектуру.

Самостійне розгортання – оскільки для його роботи необхідно масштабувати сервер планувальника, вбудований веб-сервер, брокер повідомлень для обміну інформацією між задачами та БД для зберігання стану процесів. Через таку кількість компонентів Airflow переважно розгортається в середовищі контейнеризації, часто в кластері Dask або Kubernetes (це ж роблять і сервіси Amazon MWAA і Google Cloud Composer, які надають Airflow як PaaS у хмарі).

Таким чином, Airflow є гарним рішенням для автоматизації класичних процесів (напр., ETL/ELT конвеєрів) будь-якого масштабу, а також основним варіантом у разі потреби інтеграцій із великою кількістю зовнішніх ресурсів.

Найбільш відомою альтернативою Airflow слід вважати фреймворк Luigi, створений компанією «Spotify» у 2012 р. для запуску конвеєрів у рекомендаційній системі музичних композицій. Ця система являє відкритий Python-модуль, що дозволяє представляти багатоетапні процеси у вигляді єдиного процесу, та запускати їх у послідовному або паралельному режимі. При цьому дерево залежностей між етапами процесу формується автоматично. Як і у випадку з Airflow, процес повністю визначається й описується у програмному середовищі мови Python. Проте у порівнянні з останнім, Luigi має значно менші початкові можливості, що не дало йому такого ж стрибка у розвитку. Він має дуже обмежений спектр вбудованих систем оповіщення, що

включає лише електронну пошту та інтеграцію з сервісами хмари Amazon. Також, у Luigi немає вбудованої підтримки планування робочих процесів, що передбачає застосування додаткового інструмента (напр. Cron або Celery) для їх запусків по розкладу. До того ж, графічний інтерфейс інструменту надто простий, і не завжди здатен детально відобразити статус виконання задач та їх залежностей. Таким чином, Luigi добре підходить для управління відносно тривіальними робочими процесами, які не потребують динамічних змін та постійного моніторингу з боку IT-фахівців.

Prefect – набагато сучасніший інструмент, що з'явився лише на початку 2018 р. Оскільки він був створений пізніше, ніж інші оркестратори, проєкт зумів увібрати в себе більшість їх позитивних рис, врахувати негативні (переважно досвід Airflow) й адаптувати результат під поточні тренди індустрії. Так, Prefect теж підтримує задання робочих процесів у середовищі Python, але при цьому в більш доступній і простій для розуміння функціональній парадигмі, а також має підтримку інших популярних для обробки «великих даних» мов (Julia і R). На відміну від інших рішень, він додатково надає REST та GraphQL API для програмної взаємодії з сервером оркестратора, що дозволяє «на льоту» запитувати інформацію про виконувані процеси та завдання, а також відстежувати їх поточний статус. До того ж, інструмент має інтерактивну панель, яка надає можливість централізовано керувати робочими процесами і відстежувати працездатність створених конвеєрів у реальному часі. Єдиним недоліком цього рішення можна вважати відносну новизну, через що воно ще не встигло зібрати навколо себе таку ж велику спільноту розробників, як Airflow або Luigi.

Однак, незважаючи на сильні сторони Luigi і Prefect, в контексті нашої роботи ми вирішили обрати саме Apache Airflow, оскільки його можливості цілком задовольняють нефункціональним вимогам розроблюваного конвеєра обробки даних. Також варто зазначити: оскільки Airflow використовує мову програмування Python для опису робочих процесів, увесь програмний код буде представлений виключно на ньому.

Механізми функціонування та анатомія оркестратора «Apache Airflow».

Перш ніж перейти до детального дослідження основних алгоритмів та етапів конвеєра обробки даних розроблюваної СМА, необхідно заглибитись в обрану систему управління останнім – оркестратором “Apache Airflow”. Спочатку потрібно розібратися, яким чином Airflow представляє описані у ньому робочі процеси. Це дозволить коректно описувати та структурувати їх етапи, забезпечуючи надійність і передбачуваність їх виконання.

Так, оскільки будь-який процес по суті є послідовністю завдань, його реалізацію можна представити у вигляді безперервної *конвеєрної стрічки*, кожен наступний «сегмент» якої знаходиться у прямій залежності від попереднього, і не може розпочатися, доки «вищестояща» операція не завершиться успішно. При цьому, етапи цієї стрічки мають розташовуватися згідно з певним порядком, який задається при її початковому запуску. До того ж, кожне завдання повинно бути виконане лише один раз – а у разі невдачі, перезапустити виконання всього процесу.

Описані вимоги до представлення процесів відсилають нас до основ «Теорії Графів» ([10]), а саме розділу про класифікацію графів: .

- «неперервна» конвеєрна стрічка –між будь-якою парою довільно взятих вершин повинен існувати хоча б один шлях — *зв’язний граф* [10. С. 243].
- виконання задач згідно з порядком – необхідна односпрямована направленість ребер — *орієнтований граф* [10. С. 248]. Тоді, напрямок ребра буде відображати напрямок залежності: тобто ребро від операції О1 до операції О2 вказує, що задача О1 повинна бути виконана до запуску О2.
- операція виконується лише один раз – обов’язкова відсутність петель і паралельних (кратних) ребер — *ациклічний граф* [10. С. 246-247].

При цьому, комбінація таких властивостей суворо обмежує граничну кількість ребер у графі: Нехай заданий зв’язний ациклічний граф $G = (V, E)$, де V є множиною усіх вершин, а E – являє собою кортеж із пар (u_i, v_i), де $i, j \in N \ \wedge u_i, v_i \in V$. На основі вказаних критеріїв введемо такі обмеження:

- оскільки у графі не може бути петель ні на одній з вершин, у кожному елементі множини E вершини на обох позиціях не можуть співпадати – інакше вийде псевдограф $(u \neq v \mid \forall (u, v) \in E)$;

- одночасно може існувати лише одне ребро з кожної пари кратних ребер, інакше у графі з'явиться цикл довжиною 2 (ребра) - та ми отримаємо мультиграф ($\forall\,(u,v) \in E \mid (v,u) \notin E$).

Розглянемо процес побудови графа G, де кардинальність множини $|V| = n$. При додаванні початкової вершини v_1, максимальна кількість вихідних від неї ребер дорівнює $n-1$ (див. обмеження 1). Оберемо наступну вершину v_2, яка може бути одноразово з'єднана з усіма вершинами, окрім початкової v_1 (див. обмеження 2) - це означає, що вона може мати $n-2$ інцидентних ребер. Закінчимо побудову графа вершиною v_n, яка може бути зв'язаною з будь-якими вершинами, окрім $v_1, v_2, \dots, v_n$ - а отже, додавання нового ребра здійснити неможливо (оскільки $n-1-(n-1)=0$). Такий алгоритм нагадує формування простого неорієнтованого графа, який при додаванні усіх не інцидентних одній вершині ребер стає повним графом. Знаючи це, ми можемо звернутися до "Леми про рукостискання" [10. С. 251-252; 12. С. 11] і одного з її наслідків [12. С. 12], та виразити кінцеву кількість ребер як число невпорядкованих пар на n об'єктах:

$$\binom{n}{2} = \frac{n!}{2!(n-2)!} = \frac{n(n-1)(n-2)!}{2(n-2)!} = \frac{n(n-1)}{2}.$$

Такий обґрунтований висновок дозволить нам уникати циклів у графах при плануванні порядку виконуваних у межах робочих процесів операцій.

Таким чином, структуру виконання робочого процесу можна зручно представити у вигляді орієнтованого ациклічного графа (ОАГ), вершини якого відображатимуть кроки алгоритму, а ребра – заданий порядок і статус виконання залежних кроків. Істотною перевагою даного представлення є те, що воно дозволяє сформувати доволі простий алгоритм, що можна використовувати для послідовного виконання багатоетапного конвеєра даних. Так, цей алгоритм складається з наступних кроків [39. P. 30-31]:

1. Для будь-якого відкритої (тобто «незавершеної») задачі у графі виконуються наступні дії:
    a. для кожного ребра, що вказує на задачу, перевіряється, чи завершена «вищестояча» задача на суміжній вершині;

b. якщо такі задачі вже були виконані або просто відсутні, поточна задача додається у активну чергу на виконання.

2. Послідовно виконуються завдання з черги, та позначаються як виконані при успішному завершенні;

a. якщо виконання задачі було перервано, вона ініціює повторний запуск конвеєру до тих пір, поки не завершиться з позитивним результатом;

3. Здійснюється повернення до кроку 1 і повторення тієї ж послідовності дій, доки всі задані завдання не будуть виконані.

Після ознайомлення з структурою інструмента, перейдемо до розгляду його основних концепцій та архітектурних компонентів (див. рисунок 3.2):

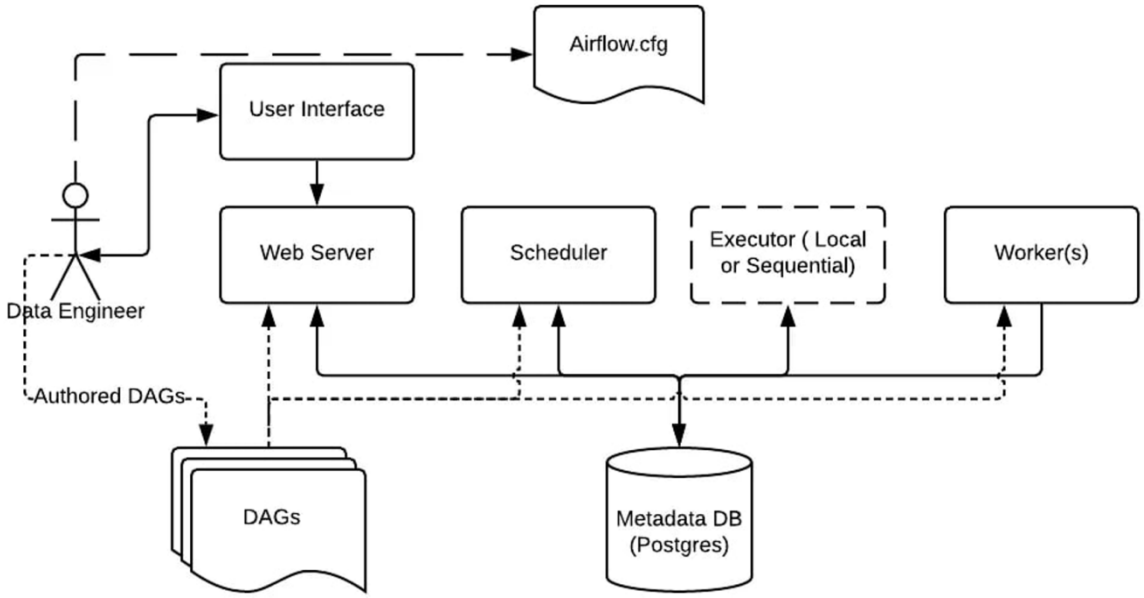


Рисунок 3.2 – Архітектурна схема оркестратора «Apache Airflow»

Джерело: Офіційна документація проекту “Apache Airflow” [32]

- ОАГ (орієнтований ациклічний граф) — основний будівельний блок, який представляє собою загальний план виконання конвеєра.
- Оператор — абстракція, яка вказує на тип виконуваної роботи кожного етапу (тобто "вершини") ОАГ.
- Задача — програмний код на мові програмування Python, що описує бізнес-логіку відповідного оператора.
- Планувальник — ключова складова Airflow, яка відповідає за запуск описаних інженером даних ОАГ. Цей компонент також забезпечує

відмовостійкість завдань та систему екстреного оповіщення у разі неможливості їх виконання.

- Виконавець — сутність, яка розподіляє заплановані до виконання завдання між робочими процесами.
- Робітник (воркер) — Python-процес, який є відповідальним за безпосереднє виконання бізнес-логіки (тобто конкретного завдання) ОАГ.
- БД метаданих — внутрішня база даних, яку використовують планувальник та виконавець системи для зберігання допоміжної інформації, пов'язаної з виконанням ОАГ (наприклад, статус конвеєра або задачі).
- Веб сервер/Інтерфейс — користувацький застосунок, що побудований на HTTP-фреймворку “Flask” та запускається через HTTP-сервер Python - “gunicorn”. Його роллю є графічне відображення ходу виконання ОАГ.

І нарешті, нам необхідно зрозуміти, яким чином ці компоненти взаємодіють один з одним у межах екосистеми Airflow. Отже, загальний алгоритм виконання ОАГ складається з наступних етапів:

- за розкладом або за ручним запитом, «планувальник» запускає конвеєр обробки даних і очікує статусу його завершення;
- для кожного запланованого завдання «планувальник» перевіряє, чи були виконані залежності завдання (тобто "ребра" ОАГ). Якщо так, то воно додається до черги виконання;
- список всіх активних завдань всередині черги виконання відправляється «виконавцю»;
- залежно від налаштувань паралелізації, «виконавець» створює один або кілька процесів «робітників2, розподіляючи заплановані завдання між ними;
- кожен «робітник» створює власне середовище виконання Python, якому передається відповідне завдання, програмний код та його параметризована конфігурація;
- після завершення завдання «робітник» повідомляє про статус виконання «виконавцю», який позначає його як виконане або невдале у базі метаданих. У разі останнього, «виконавець» повідомляє про помилку «планувальнику», який запускає конвеєр заново, тим самим забезпечуючи відповідність критерію відмовостійкості.

Отримавши фундаментальне уявлення про внутрішню будову Airflow та усвідомивши відповідні ролі його основних компонентів, ми можемо перейти до реалізації етапів конвеєра обробки даних для розроблюваної СМА.

3.4. Архітектура аналітичної системи та реалізація етапів конвеєра інтеграції даних

Архітектура та технологічний базис СМА.

В основу вирішення головної мети нашого проекту – розбудови архітектурних елементів ПК маркетингової аналітики – ми будемо щакладати низку висновків, отриманих у двох попередніх розділах нашого дослідження. Зокрема, ми аргументували необхідність прийняття наступних високорівневих аспектів програмної архітектури СМА: використання спеціалізованих сховищ даних замість стандартних баз даних, аналітичної парадигми обробки даних (OLAP) замість транзакційної (OLTP) у розділі 3.2, а також використання підходів ELT та ETL як декларативної основи конвеєра обробки даних у розділі 3.3. У тому ж розділі, ми обґрунтували обраний для розробки ПК маркетингової аналітики технологічний стек: сховище даних Google BigQuery, оркестратор завдань Apache Airflow та імперативну мову програмування Python. Також слід зауважити, що окрім перелічених технологій, інфраструктура розроблюваної системи також включатиме об'єктне сховище Google Cloud Storage, хмарні безсерверні функції Google Cloud Functions та BI-інструмента Tableau. Проте причини залучення цих програмних продуктів будуть розглянуті у поточному розділі пізніше.

Таким чином, загальний погляд на архітектуру системи розроблюваної програмної системи маркетингової аналітики показаний на рисунку 3.3:

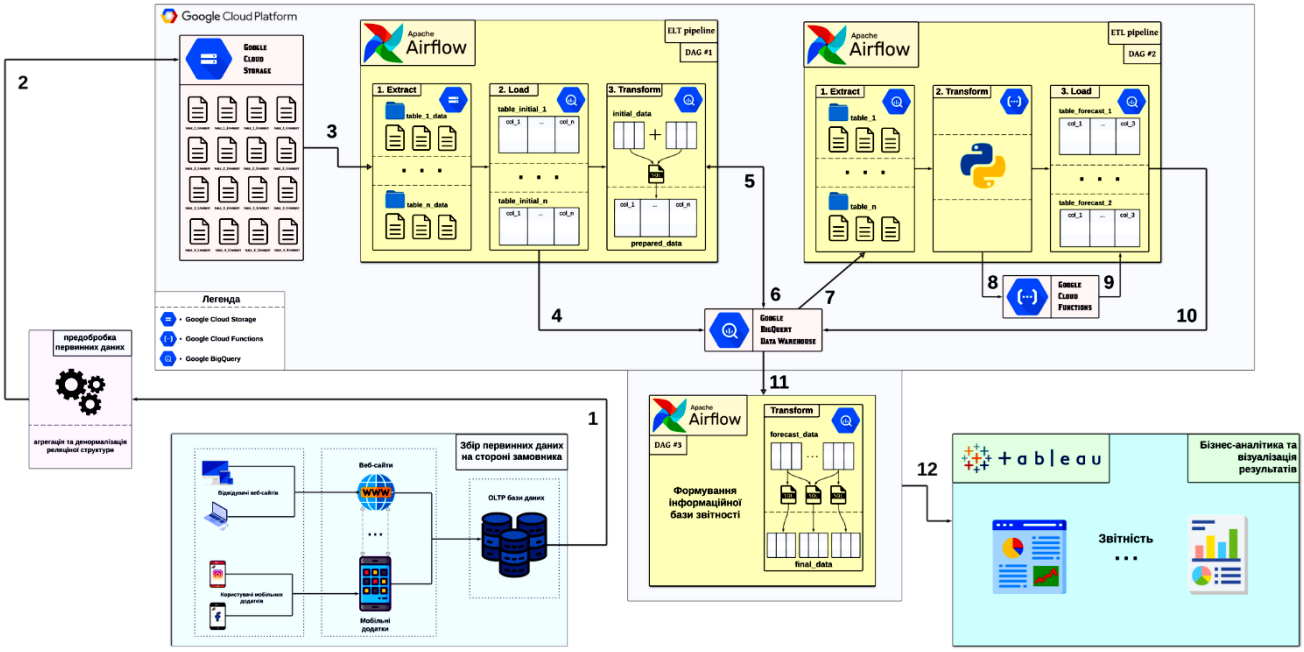


Рисунок 3.3. – Архітектурна діаграма компонентів розроблюваної СМА

Джерело: складено автором.

Формалізуємо кожну окрему компоненту діаграми, згідно зі шляхом даних всередині системи – від їх первинного збору до формування релевантної звітності:

Перший блок (на рис. блакитний) відображає процес збору первинних даних компанією-замовником: користувачі взаємодіють з веб-ресурсами (сайтами, мобільними додатками) – здійснюють підписки на її продукцію – дані про які надсилаються до внутрішньо-корпоративних OTLP баз даних.

Другий блок (на рис. фіолетовий) описує процес передобробки вихідних даних: агрегація даних з OLTP джерел – перетворення та денормалізація реляційної структури – формування набору аналітичних даних – його подальше завантаження в об'єктне сховище Google Cloud Storage.

Третій блок (на рис. «DAG #1») представляє собою основний конвеєр даних, який готує вихідні дані до подальшого аналізу – витягування інформації з Google Cloud Storage – завантаження отриманого у СД Google BigQuery – створення ключової таблиці-агрегату по поточному прибутку з кожного регіону та клієнта (базису всіх подальших дій).

Четвертий блок (на рис. «DAG #2») описує аналітичний конвеєр даних, який формує прогнозні показники та вираховує очікувані прибутки з

рекламних кампаній: витягування оброблених даних з СД Google BigQuery – виклик хмарної функції на мові Python, яка виконує ряд аналітичних алгоритмів – завантаження отриманих результатів у BigQuery.

П’ятий блок (на рис. «DAG #3») відображає фінальний етап зведення даних: створення фінальної бази звітності шляхом агрегації сформованих раніше таблиць у СД Google BigQuery.

Шостий блок (на рис. бірюзовий) вказує на візуалізацію результатів роботи системи в інструменті бізнес-аналітики Tableau: збір фінальних даних зі сховища BigQuery – формування в інструменті відповідних звітів.

Попередня обробка даних: денормалізація реляційної структури.

Деякі СМА (у т.ч. розглянуті у розд. 2) здійснюють процес збору даних самостійно, що дозволяє їм відпочатку формувати задовільний набір даних певної структури, який згодом використовується для проведення аналізу. Проте, такий підхід в аналітичній системі ми не вважаємо продуктивним, оскільки він не дозволяє адаптувати зібрані дані до запитів конкретних клієнтів (тим самим суттєво обмежуючи можливості тонкого налаштування). До того ж, багато компаній – клієнтів таких систем навряд чи можуть дозволити останнім напряму підключатися до своїх джерел даних, через високий рівень конфіденційності даних та наявності пропрієтарної інформації. Тому, у розроблюваному прототипі СМА буде запропоновано альтернативний варіант розвитку подій – вхідні дані будуть підготовлені заздалегідь, а системі доведеться лише зчитати їх з віддаленого джерела.

Враховуючи описану специфіку нашої системи, маркетинговим та технологічним відділами підприємства-замовника було вироблено стратегію динамічного збору інформації з внутрішньо-корпоративних джерел даних. Це дозволило сформувати вичерпний набір необроблених даних, які можна використовувати в основі СМА для виконання аналітичних завдань. Так, компанією була надана така схематика даних (неключові атрибути опущено):

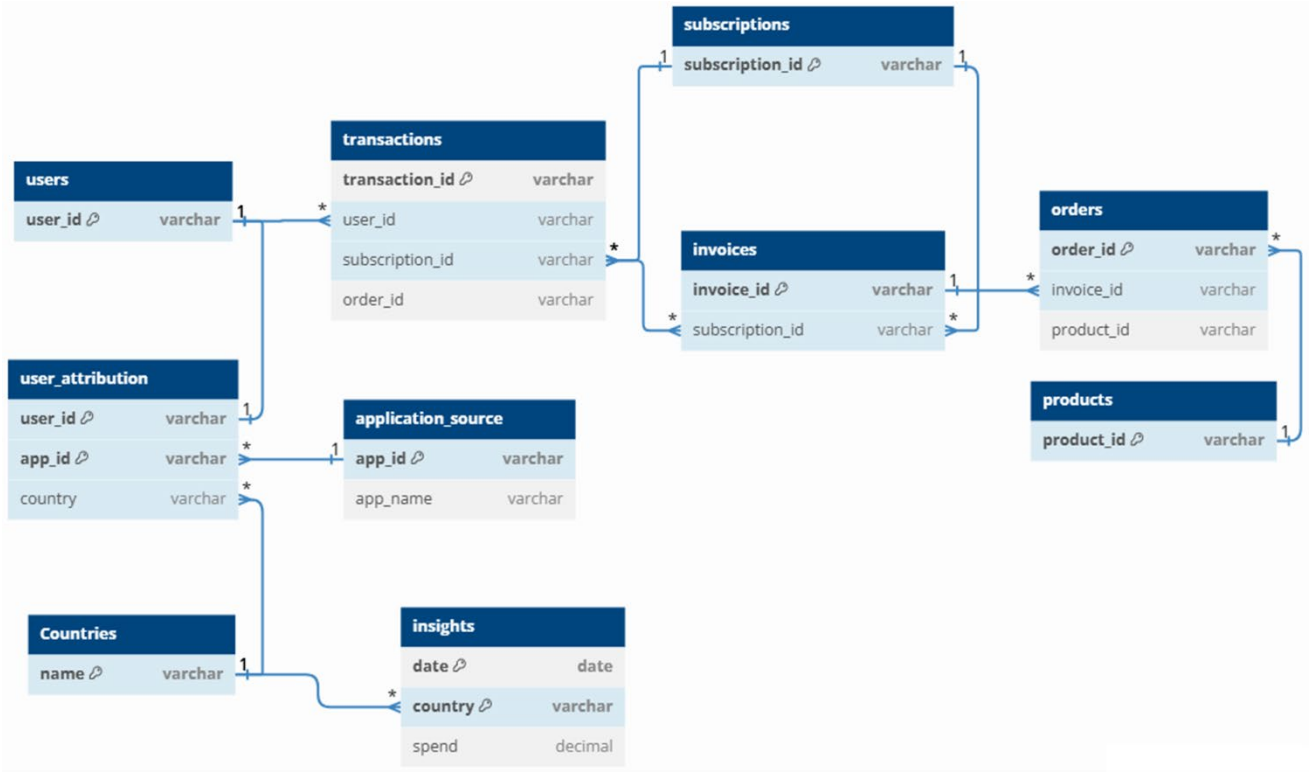


Рисунок 3.4 – Початкова схематика зберігання даних замовника

Джерело: складено автором.

На наведеній діаграмі сутностей можна помітити, що таблиці знаходяться або у 2 (у випадку прямої залежності неключових стовпців від первинного ключа), або 3 (у разі відсутності транзитивних залежностей між неключовими стовпцями) нормальній формі [5]. Крім того, всі вони пов’язані одна з одною, або безпосередньо, або опосередковано через своїх «сусідів», тим самим утворюючи ієрархію підтаблиць. Подібні спостереження наштовхують на закономірний висновок - зображена схематика відповідає багатовимірній моделі представлення даних "Snowflake" (див. дет.: [50]) що використовується в OLAP системах для гарантії відсутності аномалій даних. Однак, хоча такий підхід і був релевантним на етапі первинного збору та узгодження даних, він не є оптимальним для виконання багатофакторних аналітичних навантажень. Варто зазначити, що нормалізована структура насамперед є критерієм реляційних баз даних, який забезпечує відсутність надмірності та цілісність збереженої інформації при великій кількості паралельних операцій запису. Таким чином, це є однією з ключових вимог транзакційної (OLTP) парадигми. Раніше ми обґрунтували неефективність використання такого підходу під час

проектування СМА, змінивши його на аналітичний (OLAP). У його контексті основними вимогами є можливості багатовимірного аналізу даних та створення агрегатів, а також швидкість і ефективність виконання високоселективних запитів. В той же час, зазначені переваги нормалізації не особливо важливі в аналітичній обробці, а їх негативні наслідки ("викиди" та аномалії) цілком допустимі. Необхідність розроблюваної системи відповідати зазначеному переліку вимог приводить нас до концепції «денормалізації» структури даних, що передбачає централізацію даних у межах меншого числа таблиць (тобто об’єднання «сутностей»), а також суттєве зменшення кількості проективних операцій типу “JOIN”. Отже, нашим подальшим кроком має бути зведення кількох незалежних таблиць до одного денормалізованого набору даних.

Ключовим елементом моделі Snowflake є так звана «факт-таблиця» – центральна багатомірна модель, навколо якої будується схема БД. Сама по собі, вона містить мінімальну кількість метричних даних, і пов’язана з декількома таблицями «вимірів» через зовнішні ключі. Ці зв’язки дозволяють деталізувати дані факт-таблиці (напр., пов’язувати продаж із конкретним продуктом чи геопозицією). На схемі вище, такою центральною моделлю можна виділити “transactions”, тоді як інші таблиці швидше представляють її складові, виділені в окремі сутності. Знаючи це, ми можемо базувати нову таблицю-агрегат на її основі, перенісши туди атрибути інших моделей (зокрема, “users”, “aplication_source”, “orders”, “invoices” і “products”). Здійснивши це, ми отримаємо значно спрощену схематику даних (рисунок 3.5).

Після внесення описаних змін, ми переклали відповідальність за всі пов’язані з платежами «виміри» на модель “purchases”, водночас залишивши решту таблиць недоторканими. Таке розділення має сенс, оскільки аналітичні запити будуть охоплювати весь спектр фінансових даних, але далеко не завжди у зв’язці з іншою інформацією. На цьому етапі можна вважати процес первинної підготовки даних завершеним. Поглиблений опис структури даних у кожній із таблиць наведено у Додатку А.

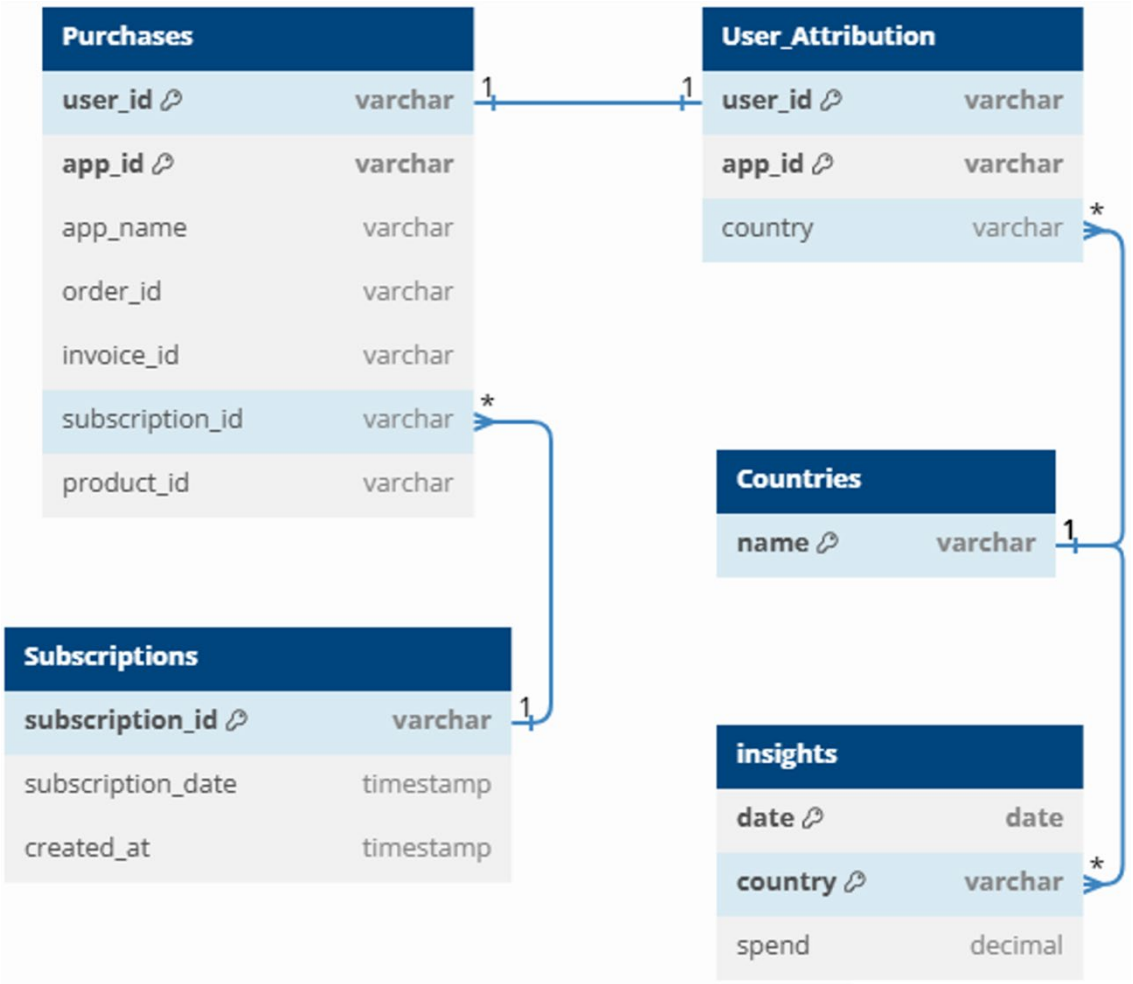


Рисунок 3.5 – Денормалізований варіант схематики даних замовника

Джерело: складено автором.

Остаточно визначившись із структурою даних, ми можемо розпочати обговорення основної тематики даної роботи, перейшовши до обговорення структури конвеєра обробки даних.

Підготовчий конвеєр даних: етап 1 – Extract (витяг даних).

Почнемо обговорення етапів підготовчого конвеєра даних з першого процесу методології *Extract*-Load-Transform (ELT) – а саме, збору первинної інформації. Раніше ми вказали, що розроблювана нами система не збирає вихідні дані самостійно – замість цього вона зчитує їх із віддаленого джерела. У контексті нашої роботи, таким джерелом виступить об’єктне сховище хмарної екосистеми Google – Google Cloud Storage (GCS), у якому вже будуть знаходитися заздалегідь агреговані дані з денормалізованою структурою (див. рисунок 3.5). Задля зменшення витрат на зберігання цих даних і оптимізації їх

подальшого відправлення в СД, вони були сконвертовані у спеціальний бінарний формат “Parquet”. Parquet – це колонково-орієнтований файловий формат, основною перевагою якого є ефективне представлення великих наборів структурованої інформації у стиснутому вигляді. Файли цього типу добре інтегруються з системами «великих даних» (в т.ч. й аналітичними комплексами) як вихідне джерело інформації. Збережені дані будуть розділені по папках відповідно до розглянутих вище таблиць, а також додатково розбиті на партиції за часовим проміжком – днями, що дозволяє ефективно мігрувати їх у стовпчасте сховище даних по частинах у пакетному (тобто “batch”) режимі.

Отже, алгоритм виконання етапу “витягування” (Extract) конвеєра обробки даних складається з наступного (див. рисунок 3.6):

- система запитує облікові дані, необхідні для підтвердження подальших дій, у сервісу відкритої авторизації платформи Google – *Google OAuth Service,*;
- здійснюється віддалений запит до сховища Google Cloud Storage, який перевіряє, чи існує заданий бакет у системі, повертаючи посилання на нього у разі наявності;
- далі здійснюється ще один запит, який перевіряє, чи існує вказаний об'єкт у бакеті, повертаючи активне посилання на нього у разі успіху;
- нарешті, система робить останній запит на витягування строкового вмісту необхідного об’єкта зі сховища, очікуючи позитивної відповіді. Перед зчитуванням даних, сховище BigQuery робить запит до сервісу авторизації з метою підтвердження задіяних в облікових даних дозволів;
- якщо дію було авторизовано і виконано успішно, отримане вміст об’єкта перетворюється у двійковий формат (т.зв. “blob” – Binary Large OBject) для подальшої серіалізації;
- останнім кроком є повернення серіалізованого GCS-об'єкта виконавцю Airflow, який передає його наступній задачі у черзі на виконання.

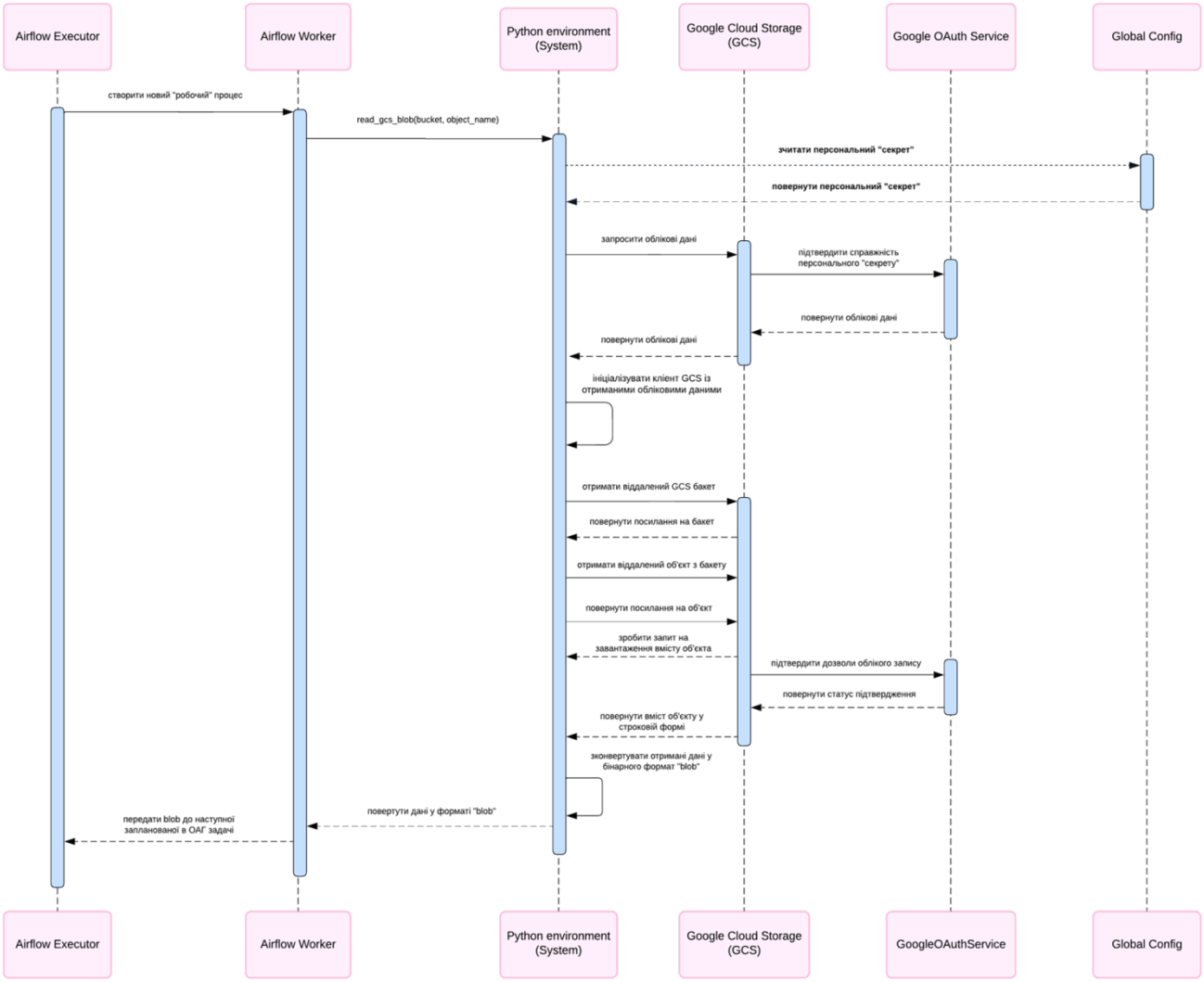


Рисунок. 3.6 – Діаграма послідовності виконання етапу «витягу» (Extract) даних
Джерело: побудовано автором.

Описаний вище і представлений на рис 3.6 процес буде здійснюватися для кожної з початкових таблиць – отже, буде повторений сумарно n разів (де «n» - кількість завантажених на попередньому кроці у GCS таблиць).

Підготовчий конвеєр даних: етап 2 – Load (завантаження даних).

На попередньому етапі ми завершили процес вилучення даних з об'єктного сховища Google Cloud Storage (GCS). Відповідно до методології Extract-*Load*-Transform (ELT), наступним етапом конвеєра має бути завантаження отриманих даних у кінцеве цільове сховище. Раніше ми вже обґрунтували ефективність використання спеціалізованих на OLAP сховищ даних для цілей проведення аналітики, а також необхідність використання інструменту Google BigQuery у якості такого. Отже, поточний етап передбачає

процес міграції зібраних з GCS даних до СД BigQuery.

Алгоритм виконання етапу "завантаження" (Load) конвеєра обробки даних полягає в наступному (див. рисунок 3.7):

- GCS-об'єкт, зчитаний на попередньому етапі, передається системі як один з вхідних параметрів у форматі “blob”;
- далі система конвертує отриманий бінарний об'єкт у формат “Parquet”, який є сумісним з BigQuery;
- оточення запитує облікові дані у сервісу відкритої авторизації Google OAuth Service, необхідні для підтвердження подальших дій в API хмари Google;
- у випадку, якщо задачі був переданий параметр “первинний запит”, він надсилається та виконується на сервері сховища BigQuery (це необхідно для початкового створення таблиці та видалення конфліктних даних). Наявність такого кроку у алгоритмі забезпечує ідемпотентність виконання задачі завантаження;
- встановлюється спосіб додавання нових даних у BigQuery - запит буде або очищати усі наявні у таблиці дані перед здійсненням операції, або просто приєднувати їх до кінця таблиці;
- система робить фінальний запит на додавання даних у форматі “Parquet” до віддаленої таблиці на сервері BigQuery. Перед здійсненням операції, сховище BigQuery запитує до сервісу авторизації з наміром підтвердити задіяні в облікових даних дозволи;
- якщо операція була авторизована і пройшла успішно, її статус повертається виконавцю, який переходить до виконання наступного завдання конвеєра.

Описаний і представлений на рисунку 3.7 процес завантаження даних у BigQuery буде виконуватися для кожного вилученого на першому етапі набору даних – а отже, буде повторений таку ж саму кількість разів.

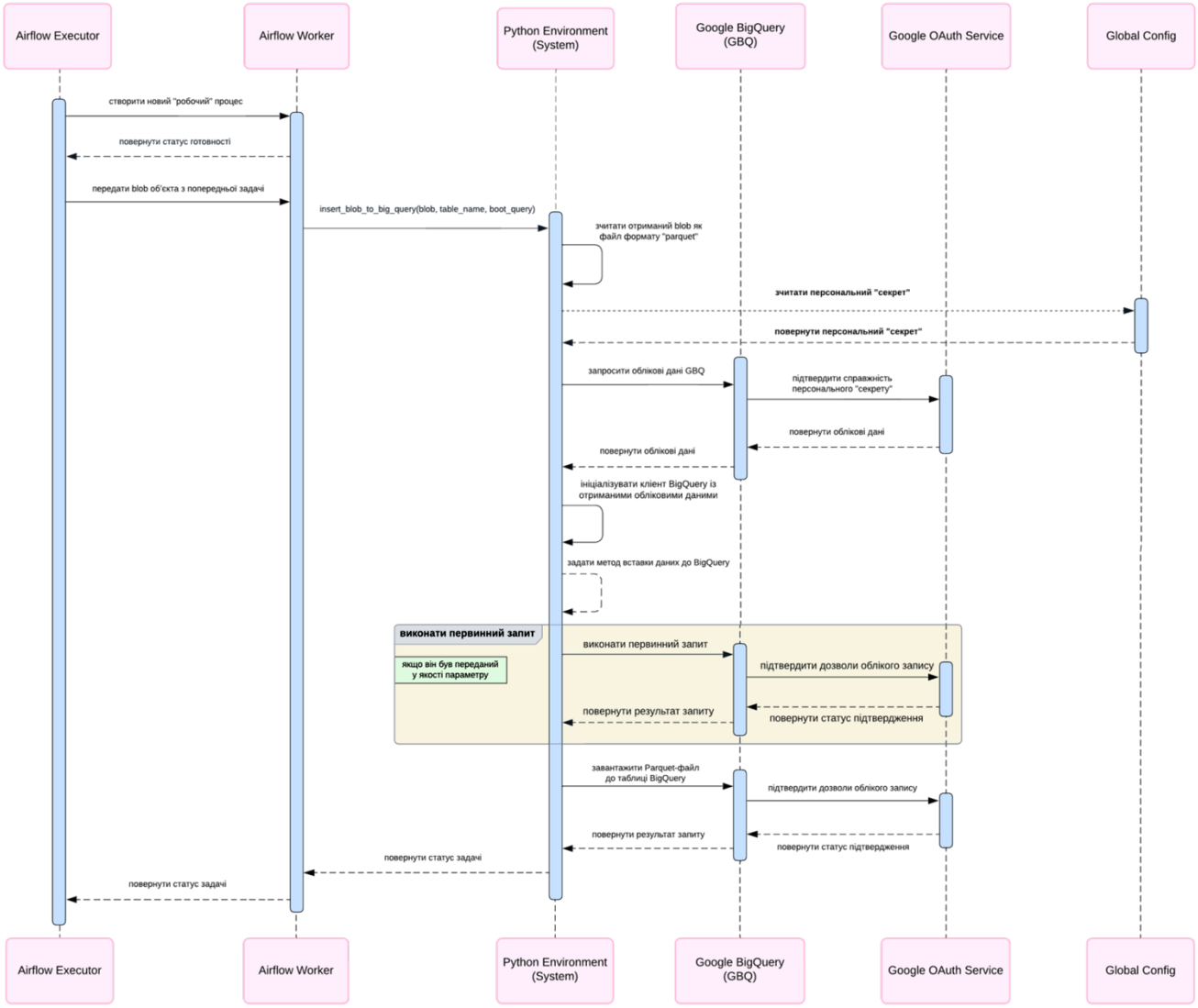


Рисунок 3.7 – Діаграма послідовності виконання етапу «завантаження» (Load)
Джерело: побудовано автором.

Підготовчий конвеєр даних: етап 3 - Transform (перетворення даних).

Завершальний етап підготовки даних полягає у зведенні даних про здійснені в межах підприємства платежі та атрибуції покупців підписної продукції. Ця операція формує нову модель «actual_revenue», яка являє собою детальний опис кожної здійсненої покупки відносно клієнта, який її ініціював, та рекламного каналу, що залучив клієнта. Такі дані можна застосовувати для проведення наскрізної аналітики у розрізі конкретних клієнтів чи споживчих груп. Для створення «actual_revenue», на стороні сервера Google BigQuery виконується спеціальний SQL-запит, який об'єднує дві розглянуті раніше таблиці – «purchases» і «users_attribution» (див. рисунок 3.8):

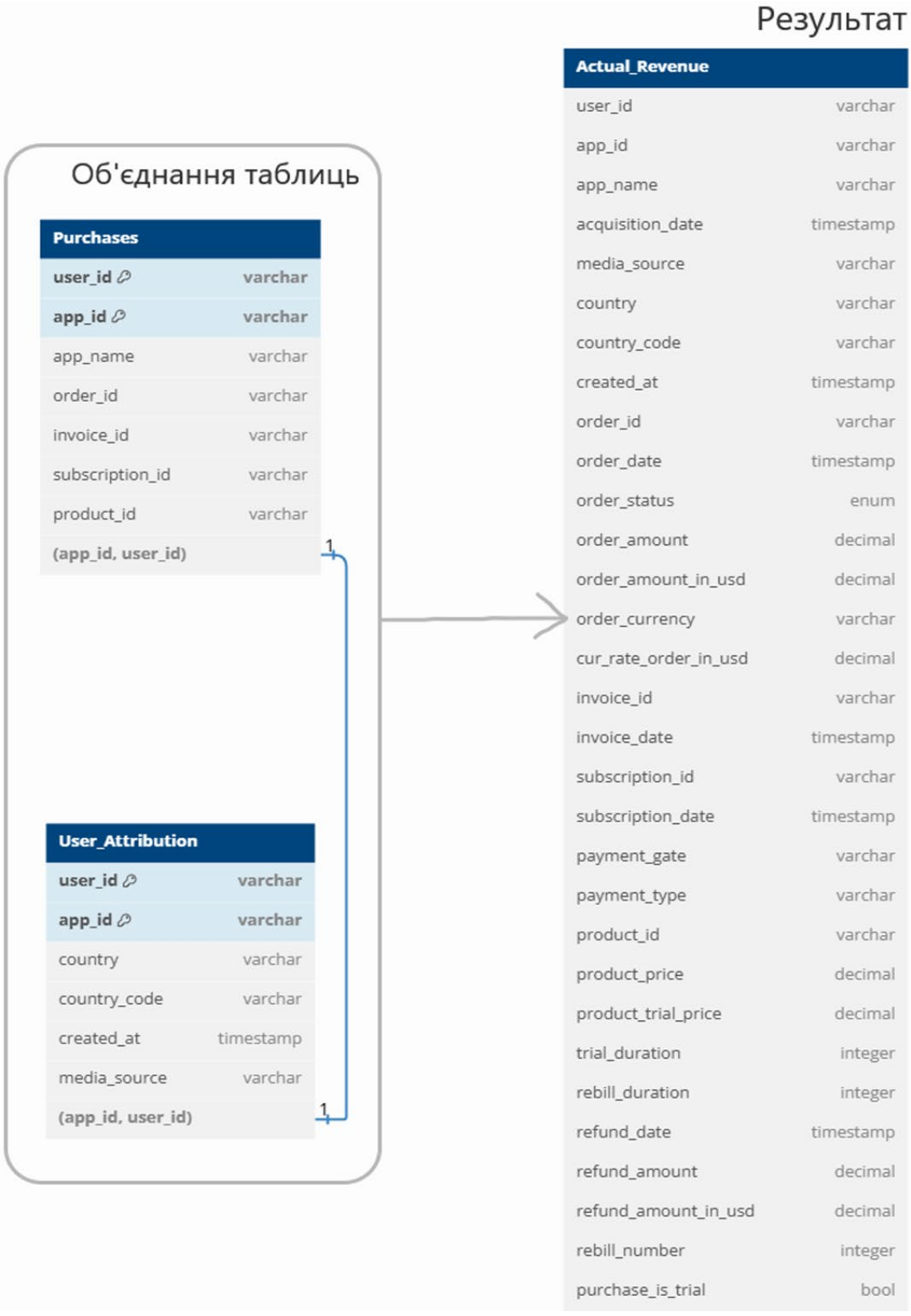


Рисунок 3.8 – Схема створення агрегаційної моделі «actual_revenue»

Джерело: складено автором.

Розглянемо його ключові моменти:

```
CREATE OR REPLACE TABLE final.actual_revenue
    PARTITION BY                                                                                    -- (1)
        DATE_TRUNC(acquisition_date, DAY)
    CLUSTER BY                                                                                      -- (2)
        app_id, user_id
AS (
    WITH purchases_full AS                                                                          -- (3)
        SELECT
          , purch.subscription_date                                         AS acquisition_date
          , users.country_code
          -- ...
          -- інші атрибути проєкції (див. рис. 3.2.x вище)
          -- ...
          , COALESCE(users.media_source, 'organic')                         AS media_source         -- (4)
          , COALESCE(users.country, 'Not Defined')                          AS country              -- (5)
          , COALESCE(users.country_code, 'Not Defined')                     AS country_code

          , DENSE_RANK() OVER (PARTITION BY subscription_id ORDER BY invoice_date) AS precalc_rebill_number   -- (6)
        FROM `initial.purchases` AS purchases                                                       -- (7)
           LEFT JOIN
               USING (app_id, user_id) -- об'єднання даних по стовбцах "app_id" и "user_id" з обох таблиць    -- (8)
    )
    SELECT
      purchases_full.user_id
      -- ...
      -- інші атрибути проєкції (див. рис. 3.2.x вище)
      -- ...
      , IF(trial_duration IS NOT NULL, precalc_rebill_number - 1, precalc_rebill_number) AS rebill_number     -- (9)
    FROM purchases_full
);
```

1. Оскільки завантажені у BigQuery дані заздалегідь розбиті на партиції по дням, має сенс розділити цю таблицю по цьому ж часовому проміжку. Це оптимізує виконання усіх запитів, пов’язаних з датами залучення клієнтів.

2. Фізично згруповуємо дані всередині таблиці по композитному первинному ключу, що дозволить ефективно виконувати JOIN-операції.

3. Створюємо іменований набір даних, що буде використаний у подальшій проекції.

4. Якщо атрибуційний канал не відстежено, вважаємо, що клієнт був залучений органічно.

5. Якщо країна походження клієнта не знайдена або не існує, ініціалізуємо стовпці «country» та «country_code» значенням за замовчуванням (уникаємо обробки недетермінованих NULL-значень).

6. Підраховуємо, скільки разів клієнт здійснював повторну підписку на один і той самий продукт (на момент поточного платежу) за допомогою віконної функції. Це інструмент агрегації, який спочатку розбиває загальний набір даних на окремі "вікна" (сукупності рядків), а потім виконує вказану

функцію для кожного рядка у їх межах. У нашому випадку, рядки будуть розбиті на групи за значенням «subscription_id» (тобто для кожного унікального subscription_id буде створено своє «вікно»), і впорядковані за датою виставлення рахунку «invoice_date». А функція «dense_rank» дозволяє визначити порядковий номер поточної сплати в межах кожної підписки.

7. Включаємо до вибору усі записи із таблиці «purchases» та відповідні записи із «users_attribution».

8. Об`єднуємо дані з обох таблиць по стовпцях «app_id» и «user_id».

9. Якщо одна з підписок є пробним періодом, її не можна вважати прибутковою, і тому вона виключається з загальної послідовності.

Аналітичний конвеєр: загальний опис и функціональні особливості.

На поточному етапі ми успішно виконали основний конвеєр обробки даних, і таким чином завершили процес підготовки первинних даних до подальшого аналізу та вироблення прогнозів на їх основі. Тут слід зазначити: оскільки основним фокусом цього дослідження є проектування архітектури CMA, внутрішня логіка алгоритму аналізу і прогнозування не буде розглянута в контексті нашої роботи. Тому сам алгоритм буде надано у готовому вигляді зацікавленою стороною – проте релевантне завдання його інтеграції у загальну послідовність дій, а також специфіка роботи будуть описані у цьому пункті.

Розглянемо функціональні особливості алгоритму:

- для проведення аналізу, на вхід алгоритму надаються зведені дані з користувацьких продажів, які зберігаються у СД BigQuery.
- після виконання аналітичних навантажень, отримані дані використовуються для створення нової таблиці, яка описує коефіцієнти відтоку та лояльності клієнтських аудиторій.
- на основі цієї таблиці формується кінцева прогнозна модель «expected_revenue», яка поєднує дані щодо наявних та очікуваних продажів.

Оскільки алгоритм вимагає використання BigQuery у якості вхідного і вихідного джерела даних, а його часова складність ступінно зростає разом з розміром таких даних, нами було прийняте рішення розгорнути його в середовищі безсерверних функцій *Google Cloud Functions.* Такий вибір дозволяє заощадити на доставці великих обсягів інформації з BigQuery до

функції, оскільки обидва ці хмарні ресурси будуть розгорнуті в одному регіоні, що мінімізує мережеву затримку двостороннього обміну даними. Завдяки тому, що перенесення даних займатиме менше часу, алгоритм загалом виконуватиметься швидше, ніж якби він був розгорнутий на локальній інфраструктурі. Оскільки алгоритм не вимагає зберігання проміжного стану, він може легко масштабуватись горизонтально, що дозволяє розподілити будь-який обсяг обчислювальних навантажень між репліками функції.

Таким чином, підготовлені вихідні дані спочатку будуть вилучені зі сховища BigQuery, потім використані з метою аналітики алгоритмом (в т.ч. породжуючи нову таблицю), результати роботи якого повинні бути знову завантажені в СД BigQuery у вигляді нової моделі. Така послідовність дій – вилучення-перетворення-завантаження – нагадує етапи підходу ETL (extract-transform-load). Отже, нам необхідно створити додатковий конвеєр даних, який слідуватиме цій класичній методології.

Алгоритм виконання конвеєру включає такі етапи (див. рисунок 3.9):

- воркер Airflow ініціює запуск Python «оператора», який здійснює параметризований виклик функції *cloud_functions_api_call.* Ця функція є відповідальною за запит до API хмарних функцій Google.
- перед викликом самої функції, система має отримати необхідні для авторизації цієї операції облікові дані, що можна отримати на основі персонального «секрету». Тому, система має спочатку зчитати цей секрет із глобальної конфігурації середовища, а потім зробити запит до сервісу відкритої авторизації Google OAuth, та створити токен доступу на його основі.
- далі отриманий JWT-токен потрібно додати у заголовок “авторизація”, що підтвердить справжність HTTP-запиту. Після цього, система здійснює POST-запит до розгорнутої хмарної функції.
- оскільки першим етапом виконання функції є витяг даних зі сховища BigQuery, необхідно запросити ще один набір облікових даних у сервісу Google OAuth, необхідних для підтвердження подальших дій в API Google.
- після отримання облікової інформації, функція виконує спеціальний SQL-запит на зчитування необхідного набору вхідних даних.
- BigQuery повертає результат запиту у форматі JSON – проте, заради

сумісності з алгоритмом, його необхідно привести до вигляду датафрейму (двовимірної моделі рядків і стовпців). На основі отриманих даних, виконується аналітичний алгоритм та виробляються релевантні прогнози.

• результат роботи алгоритму завантажується у СД BigQuery у вигляді нових таблиць – прогнозних моделей коефіцієнтів відтоку та лояльності клієнтів та таблиці очікуваних прибутків “expected_revenue”.

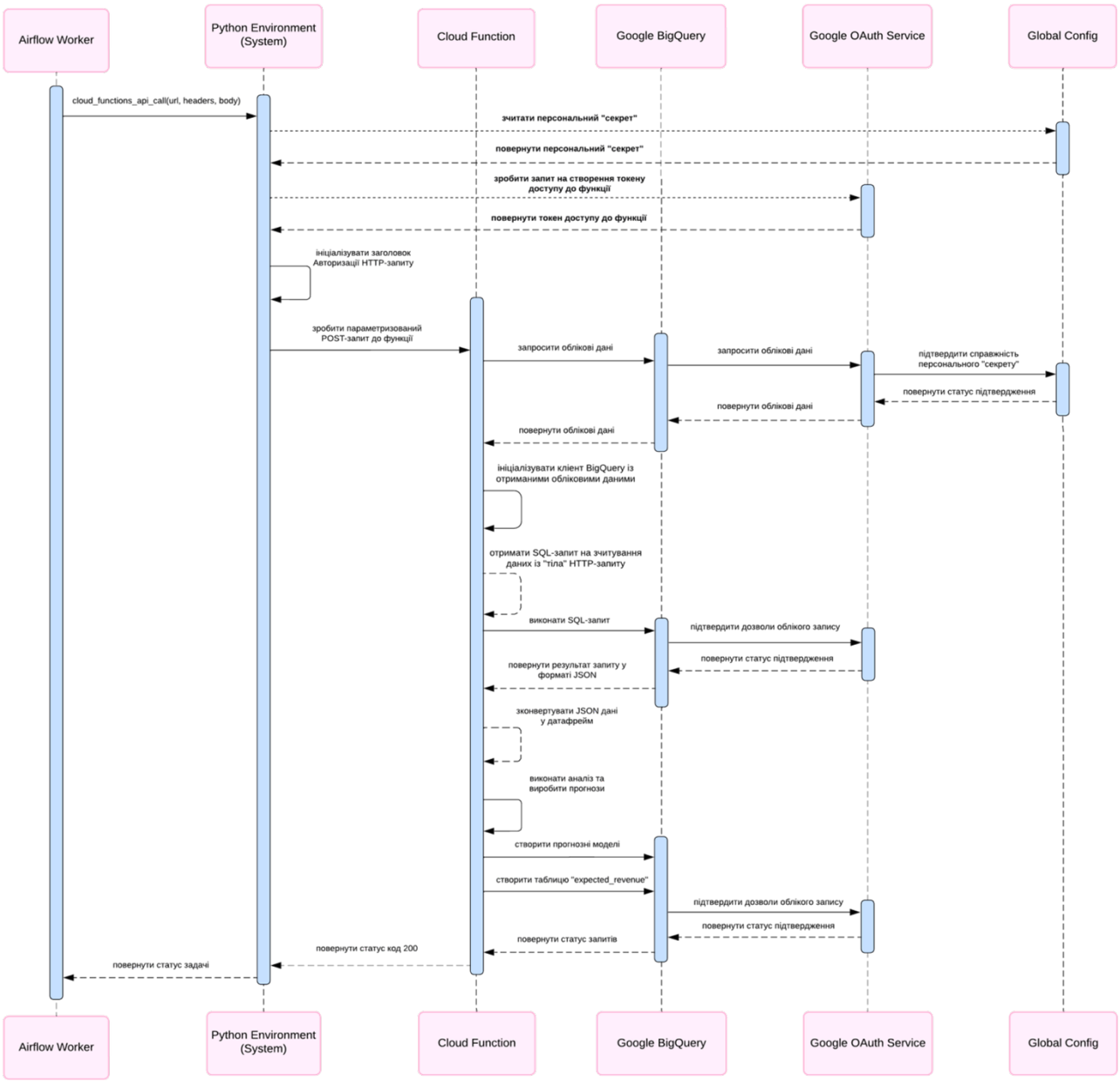


Рисунок 3.9 – Діаграма послідовності виконання аналітичного конвеєру даних

Джерело: побудовано автором.

Успішне завершення цього етапу активує запуск останнього активного процесу у архітектурі розроблюваної СМА, який є відповідальним за зведення

первинних оброблених та прогнозних даних.

Фінальний етап: зведення підготовлених та прогнозних даних.

Як ми вказали раніше, останній етап обробки даних передбачає зведення моделей, отриманих у результаті виконання двох попередніх активних етапів (конвеєрів), а також їх приведення у сумісний з інструментом візуалізації звітності «Tableau» формат. Вибір цього BI-інструменту викликаний запитом компанії-замовника, яка вже має розгорнутий і налаштований сервер із цим ПЗ, а також необхідну компетенцію для освоєння результатів його роботи. Так, нам необхідно об’єднати сформовані раніше таблиці у дві нові фінальні – «user_purchases» (яка відображає максимально докладну інформацію про користувальницькі покупки), і «final_report» (яка відображає доходи і витрати на проведені рекламні кампанії у різних країнах за відповідні дати).

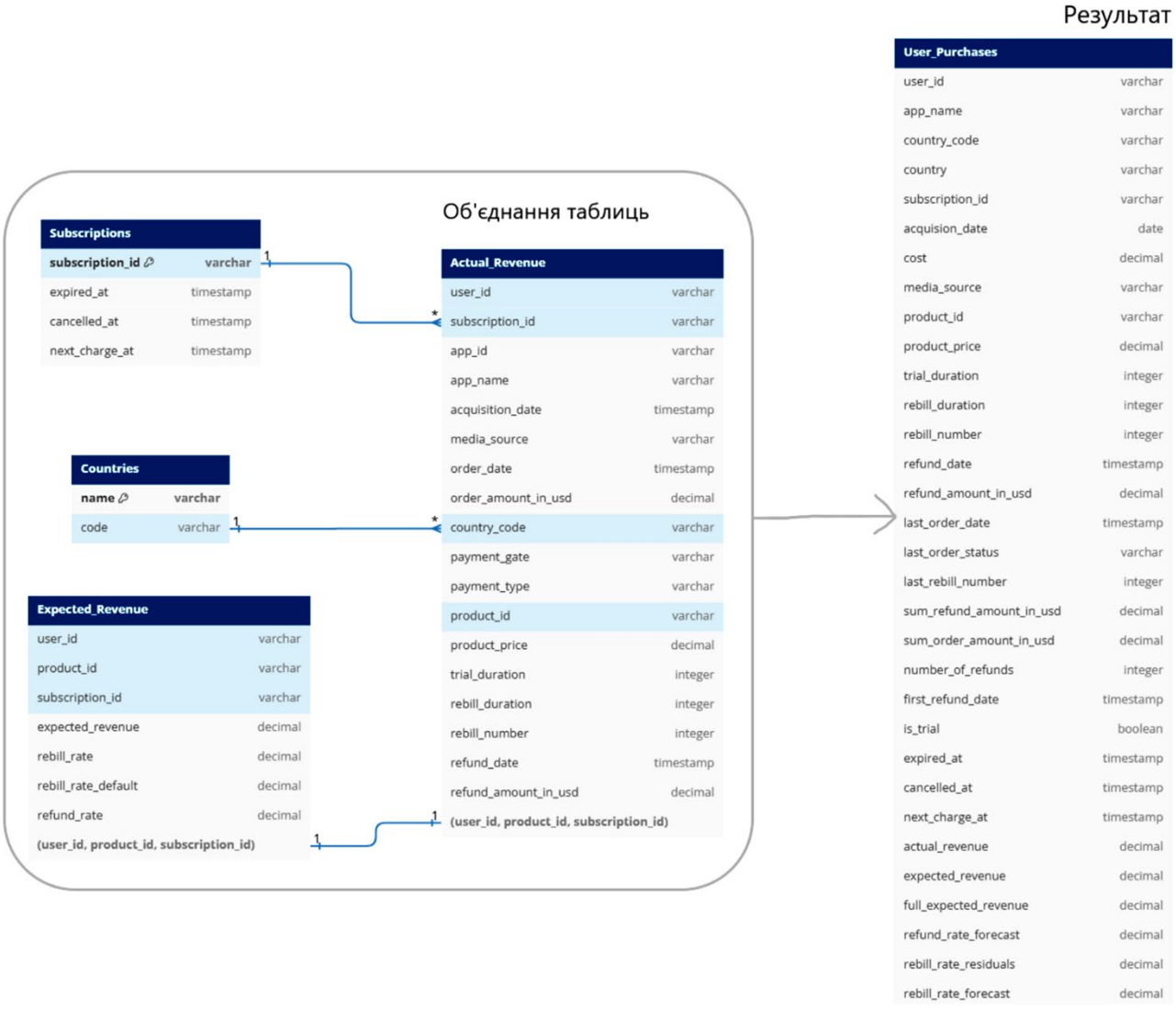


Рисунок 3.10 – Схема створення фінальної моделі звітності «user_purchases»

Джерело: складено автором.

Почнемо з «user_purchases», яка є поєднанням таблиць «subscriptions», «countries», «actual_revenue» та «expected_revenue». Логічна схема її формування на основі цих сутностей наведена на ER-діаграмі на рисунку 3.10.

Лістинг SQL-запиту, що створює таблицю «user_purchases», буде наведений у додатках до нашої роботи (див. Додаток Б).

Розглянемо другу таблицю «final_report» – у якій ми доповнюємо «user_purchases» даними по витратах на рекламні кампанії з «insights». Логічна схема її формування наведена на ER-діаграмі на рисунку 3.11:

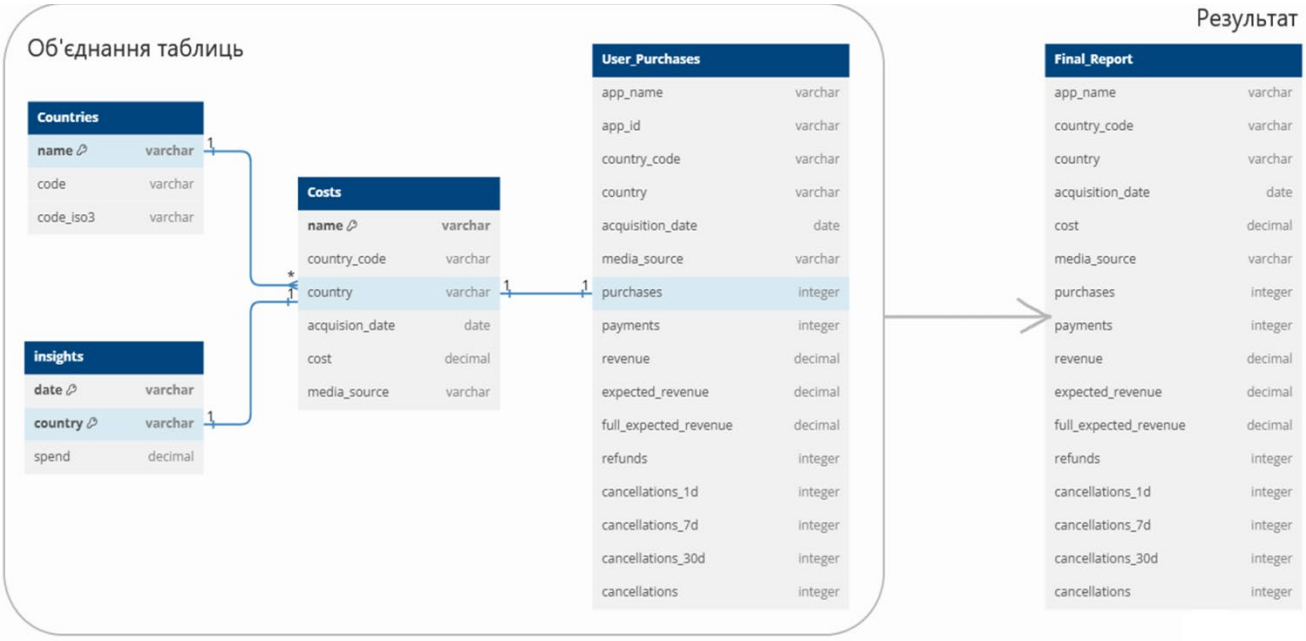


Рисунок 3.11 – Схема створення фінальної моделі звітності «final_report»

Джерело: складено автором.

Лістинг SQL-запиту, що створює таблицю «final_report», буде наведено у додатках до нашої роботи (див. Додаток В).

3.5. Тестування працездатності розробленої системи

У якості одного зі способів тестування працездатності системи та оцінки адекватності результатів, ми можемо провести економетричне дослідження на основі сформованих прогнозних даних. Так, маючи вичерпну інформацію про

очікувані показники прибутку та залучення клієнтів, ми можемо виробити стратегію ефективного розподілу маркетингового бюджету підприємства-замовника по задіяних рекламних каналах на найближчі місяці.

Для цього, ми формалізуємо та розв'яжемо розширену мінімаксну задачу [3, С. 20-22], в якій потрібно знайти найбільш вигідні розподіли інвестицій залежно від вхідних параметрів та переліку відомих обмежень. Ми представимо її у вигляді «задачі задоволення обмежень» (*Constraint Satisfaction Problem,* ЗЗО), а для її розв'язання ми будемо використовувати набори методів й алгоритмів парадигми «програмування з обмеженнями» (Constraint Programming), що є потужним інструментом для вирішення будь-яких комбінаторних задач. Формально, ЗЗО (позначається як $P = < X, D, C >$) визначається набором з трьох множин: вирішальні змінні $X = \{x_1, x_2, \dots, x_n\}$, «домени» змінних $D = \{d_1, d_2, \dots, d_n\}$ та обмеження $C\{c_1, c_2, \dots, c_k\}$ [46. P. 202]. Будь-яка змінна x_i має непусту область визначення (домен) D_i, що включає всі її можливі значення. При цьому, кожне обмеження c_i залежить від певної підмножини вирішальних змінних, і задає допустимі комбінації значень $v_1, v_2, \dots, v_n$ для неї. Поточний стан задачі на кожному етапі вирішення визначається шляхом присвоєння значень деяким або всім заданим змінним, породжуючи комбінацію виду $\{x_i = v_i, \dots, x_j = v_j\}$. Комбінація, яка не порушує зазначених обмежень, називається «сумісною», а та, в якій бере участь кожна змінна з множини X, називається «*повною*». Розв`язком задачі є вибір найбільш відповідної до її запитів *повної комбінації*, що задовольняє усім накладеним обмеженням [46. P. 203]. Знаючи це, ми звернемося до мови визначення ЗЗО і формалізуємо поставлену задачу, виділивши її ключові і важливі елементи:

Мета задачі:

Знайти оптимальний розподіл бюджету по маркетинговим каналам і місяцях з метою максимізації доходу замовника та кількості залучених клієнтів. Таким чином, ми можемо покращити стратегію виділення рекламного бюджету задля досягнення найкращих результатів для підприємства.

Набір вхідних даних:

- C - множина маркетингових каналів.
- M - множина розглянутих місяців відповідних років.
- $G_{r,m}$ - загальний прибуток підприємства від каналу r за місяць m.
- $S_{r,m}$ - загальні витрати на просування каналу r за місяць m.
- $C_{r,m}$ - кількість залучених покупців від каналу r за місяць m.

Параметри:

- B - загальний рекламний бюджет підприємства-замовника.
- $B_{min,r}$ та $B_{max,r}$ - мінімальний/максимальний бюджет каналу r за весь період
- b_{min} - мінімальний бюджет, що має бути виділений кожному каналу, кожного місяця.
- t_{min} та t_{max} - коефіцієнти обмеження місячного бюджету на всі канали.
- α та β - вагові коефіцієнти, які визначають важливість прибутку та кількості клієнтів відносно одне одного.

Вирішальні змінні:

- $x_{r,m}$ – сума грошей, яку компанія виділяє на просування каналу r за місяць m. Ці змінні є дискретними, а отже мають скінченні області визначення D_r.

Обмеження:

1. Обмеження на загальний рекламний бюджет: Сума грошей, виділених на рекламу по всіх каналах і місяцях, не повинна перевищувати загальний рекламний бюджет «B» компанії: $\sum_{r\in R}\sum_{m\in M} x_{r,m} \leq B$.

2. Мінімаксні обмеження бюджету по кожному каналу: Сума грошей, виділених на рекламу каналу r за всі місяці, повинна бути в межах від $B_{min,r}$ до $B_{max,r}$, де $B_{min,r}$ та $B_{max,r}$ - мінімальні и максимальні бюджети для каналу r:

$$B_{min,r} \leq \sum_{m\in M} x_{r,m} \leq B_{max,r} \; \forall \; m \in M.$$

3. Обмеження на мінімальний бюджет для кожного каналу: Щоб гарантувати щомісячну маркетингову присутність компанії, ми маємо встановити мінімальний бюджет b_{min}, який повинен бути виділений на кожен канал кожного місяця: $x_{r,m} \geq b_{min} \; \forall \; r \in R, \; m \in M$.

4. Мінімаксні обмеження бюджету на кожен місяць: Сума грошей, виділених на проведення реклами кожного місяця, повинна бути у межах від $t_{min} * B$ до $t_{max} * B$, тобто не бути менше або більше певного відсотка від загального бюджету:

$$t_{min} * B \leq \sum_{r \in R} x_{r,m} \leq t_{max} * B \ \ \forall\, m \in M, 0 \leq t_{min}, t_{max} \leq 1.$$

Цільова функція:

Ми прагнемо максимізувати дві ключові категорії: загальний прибуток компанії та кількість залучених покупців за певний період часу. Для цього, ми враховуємо співвідношення між інвестиціями в рекламу та доходами, використовуючи показник *ROI*, а також співвідношення між витратами на рекламу та кількістю клієнтів, взявши за основу обернений коефіцієнт метрики *CAC* (оскільки нам потрібно максимізувати значення функції). Вказані показники та метрики були описані у розділі 1.2 даної роботи.

$$roi_{r,m} = \frac{G_{r,m} - S_{r,m}}{S_{r,m}} * x_{r,m} \ \ \forall\, r \in R, m \in M.$$

$$cac_{r,m} = \frac{S_{r,m}}{C_{r,m}} * x_{r,m} \ \ \forall\, r \in R, m \in M.$$

Замість звичайної максимізації прибутку, ми створимо комбіновану функцію, яка враховує як і доходи підприємства, так і кількість залучених клієнтів. Додатково, обом категоріям можна вказати коефіцієнти "ваги" (α та β відповідно), що вплинуть на розподіл пріоритетів при виборі найкращих рішень. Таким чином, нам необхідно максимізувати наступну функцію:

$$f_{max} = \alpha \sum_{r \in R} \sum_{m \in M} roi_{r,m} + \beta \sum_{r \in R} \sum_{m \in M} \frac{1}{cac_{r,m}}, \ \ \forall\, \alpha, \beta \in Q \mid \alpha + \beta = 1.$$

$$<=> \sum_{r \in R} \sum_{m \in M} \left(\alpha * roi_{r,m} + \beta * \frac{1}{cac_{r,m}} \right).$$

$$<=> \sum_{r \in R} \sum_{m \in M} x_{r,m} \left(\frac{G_{r,m} - S_{r,m}}{S_{r,m}} \alpha + \frac{C_{r,m}}{S_{r,m}} \beta \right).$$

$$<=> \sum_{r \in R} \sum_{m \in M} x_{r,m} \left(\frac{G_{r,m}\alpha - S_{r,m}\alpha + C_{r,m}\beta}{S_{r,m}} \right).$$

Інфографіку дослідження при різних наборах вхідних параметрів, а також лістинг програмного коду, що автоматизує вирішення задачі, можна знайти у секції з додатками даної кваліфікаційної роботи (див. **Додаток Г та Д)**.

ВИСНОВКИ

Основною метою нашого дослідження є розробка архітектурних засад інтегрованого ПК аналітики для автоматизації поточного та оптимізації майбутнього здійснення збутових операцій ІТ-компанії – замовника. У ході досягнення цієї мети було вирішено широке коло завдань.

Було визначено концептуальні засади функціонування як самої галузі Інтернет-маркетингу, і ролі, що грають у розвитку системи маркетингової аналітики. Шляхом дослідження структури взаємодії суб'єктів даної збутової моделі та застосовуваних ними інструментів продажів було з'ясовано, що реалізувати свої переваги перед традиційним маркетингом вона може лише завдяки різкому розширенню можливостей для автоматизації проведення всіх видів поточних та перспективних бізнес-операцій – а єдиним інструментом такої автоматизації є задіяння програмних комплексів збору, відстеження, аналізу та прогнозування гетерогенної економічної інформації. Також були виявлені головні об'єкти відстеження кожної аналітичної системи – набір суворо упорядкованих даних про результати збутової, рекламної та фінансової діяльності підприємств, які прийнято об'єднувати під поняттям маркетингових показників, та розділяти на дві групи – первинні метрики та KPI (ключові індикатори ефективності), які є їхньою комбінацією.

Багатокритеріальний огляд ринку CMA дозволив виділити оцінити сучасний стан і тенденції його майбутнього розвитку та виокремити повний комплекс взаємовідносин між виробниками і споживачами цих прогнозних інструментів. Зокрема, була пояснена наявність на ньому великого числа програмних продуктів різної спрямованості, ступеня зовнішньої і внутрішньої інтеграції та можливостей пристосування до функціональних вимог замовників, та здійснена їх класифікація. Першу групу становлять генералізовані платформи, виробниками яких є всесвітньовідомі корпорації ІТ-сектору Google, Microsoft, Adobe, HubSpot та ін., які набули популярність саме завдяки охопленню широкого спектру процесів Інтернет-маркетингу. Проте зворотнім боком такої «всеохопності» стає їх надмірна стандартизація,

що суттєво знижує потенціал їх налаштування під різні бізнес-ситуації. Тому на практиці споживачі часто віддають перевагу другій групі СМА – доповнюючим ПЗ, які, базуючись на основному функціоналі вказаних платформ, намагаються усунути їх означений недолік, надаючи ширші можливості у налаштуванні як самих аналізованих показників, так і всіх процедур їх життєвого циклу (збору, зберігання, обробки даних та візуалізації результатів).

Було проведено розгорнутий порівняльний аналіз зазначених СМА. У цьому контексті були вивчені продукти як зарубіжних (Google Analytics, HubSpot, Woopra BI), так і вітчизняних (Ringostat, Livepage, Netpeak) виробників.

Результатом стало виявлення їх конкурентних переваг і недоліків, обґрунтування вибору генералізованих платформ як *аналогів* розроблюваної у даній роботі системи, а також вироблення мінімального списку функціональних і нефункціональних вимог до подібних інтегрованих систем.

Дослідження існуючих парадигм обробки та зберігання даних – OLTP та OLAP – дозволило обґрунтувати релевантність використання останньої для розроблюваного ПК. Крім того, виявлено основні недоліки використання класичних баз даних для втілення комплексних аналітичних сценаріїв, та необхідність залучення концепції «Data Warehouse» для досягнення зазначених цілей. Визначеність в останньому дозволила провести порівняльний аналіз існуючих систем на ринку сховищ даних та розгорнуто обґрунтувати вибір конкретного програмного продукту – Google BigQuery.

З іншого боку, було обґрунтовано необхідність об'єднання всіх операцій систем маркетингової аналітики в єдиний безперервний бізнес-процес та виявлено можливості їхньої конвеєризації. Проведено порівняння ключових методологій інтеграції «великих даних» – ETL та ELT, та вироблено стратегію їх комбінованого залучення до проектування етапів конвеєру даних. Проведено аналіз вимог компанії-замовника до процесу обробки інформації, які були згодом зведені до формату єдиного робочого процесу. Детально вивчено клас систем для управління такими багатоетапними бізнес-процесами

- диспетчерами завдань, та проведено характеристику доступних рішень цієї ринкової ніші з детальним обґрунтуванням вибору конкретної технології – Apache Airflow.

На основі проведених досліджень, було дано вичерпне обґрунтування технологічного стеку програмного проекту, в тому числі інтеграції з сервісами хмарної платформи «Google Cloud Platform» - зокрема, BigQuery, Cloud Storage і Cloud Functions. До того ж, був розглянутий загальний шлях та життєвий цикл даних у системі, а також детально описані існуючі етапи та реалізація алгоритмів єдиного конвеєра обробки даних.

Таким чином, кінцевим результатом дослідження стало створення уніфікованого програмного комплексу, адаптованого до функціональних вимог підприємства-замовника. Розроблена система здатна автономно оркеструвати та узгоджувати конвеєризовані робочі процеси, що включають всі етапи життєвого циклу обробки даних – від їх вилучення з первинних джерел до формування бази аналітичної інформації та складання графічних звітів бізнес-аналітики на їх основі.

З метою оцінки продуктивності роботи розробленого ПК, було математично формалізовано та програмно вирішено *економетричну мінімаксну задачу*, результати виконання якої дозволяють передбачити найоптимальніші стратегії виділення рекламного бюджету підприємства-замовника у майбутньому, що дозволяє системно оцінити відповідність математичної моделі системи та сформованих нею результатів адекватності. Для вирішення задачі використовувалася мова Python, а також теоретичний апарат «задоволення обмежень» та парадигма «програмування в обмеженнях».

Таким чином, всі поставлені перед цим кваліфікаційним дослідженням завдання було виконано. Отримані в його ході теоретичні результати мають елементи новизни, поглиблюючи наукові уявлення про домен Інтернет-маркетингу, а також структуру та архітектурні підходи до розробки сучасних систем маркетингової аналітики, а розроблений аналітичний програмний комплекс може досить ефективно застосовуватися різними компаніями доменів Інтернет- та традиційного маркетингу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 15 ключових KPI інтернет-магазину для маркетингу та відділу продажів [Електронний ресурс]. Режим доступу: https://pricer24.com/uk/blog/15-klyuchovih-kpi-v-e-commerce-dlya-marketingu-ta-viddilu-prodazhiv/ (Останнє відвідування: 12.07.2023).

2. Апостол А. В. Системи оперативного аналізу даних. *Наукові записки НаУКМА.* Т. 99 : Комп'ютерні науки. С. 61-66.

3. Ачкасов А. Є. Конспект лекцій з курсу «Економіко-математичне моделювання» (для студентів 3 курсу заочної форми навчання бакалаврів за галуззю знань 0305 «Економіка і підприємництво» / А. Є. Ачкасов, О. О. Воронков; Харк. нац. акад. міського госп-ва. Х.: ХНАМГ, 2011. 204 с.

4. Бергер М. 10 KPI-метрик эффективности маркетинга [Електронний ресурс]. Режим доступу: https://netpeaksoftware.com/ru/blog/10-kpis-every-digital-marketer-should-track (Останнє звернення: 02.04.2024).

5. Буй Д. Б., Пузікова А. В. Теорія нормалізації в реляційних базах даних: сучасний стан [Електронний ресурс]. Режим доступу: https://ekmair.ukma.edu.ua/server/api/core/bitstreams/71ef12e4-e84f-4b66-8770-c3ecdf311e5e/content (Дата звернення: 05.2024).

6. Даценко М. Snowflake vs BigQuery: Порівняння двох популярних рішень для сховищ даних. URL: https://cloudfresh.com/ua/cloud-blog/snowflake-vs-bigquery-porivnyannya-dvoh-rishen/ (Дата звернення: 05.2024).

7. Дронь, К. Об’єктне сховище: легкість керування масштабними масивами даних. URL: https://netwave.ua/obyektne-shovyshche-danyh/ (Дата звернення: 11.05.2024).

8. Іванов Ю. Б., Ус М. І. Складові інформаційного забезпечення маркетингової діяльності промислового підприємства. Бізнес Інформ. 2016. № 1. С. 299-305.

9. Интернет-маркетинг и его инструменты. URL:https://sendpulse.ua/ru/support/glossary/internet-marketing (Дата звернення: 20.03.2024).

10. Комп’ютерна дискретна математика: Підручник/ М. Ф. Бондаренко, Н. В. Білоус, А. Г. Руткас. Харків: “компанія СМІТ”, 2004. 480 с.

11. Котлер Ф. Маркетинг 4.0. Від традиційного до цифрового / Ф. Котлер. Г. Катарджая, Ї. Сетьяван; під редакцією В. Олександрова. Київ : КМ-БУКС, 2019. 224 с.

12. Курс лекцій з дисципліни «Дискретна математика», розділ «Теорія графів» для факультету «Комп'ютерно-інформаційних систем і програмної інженерії» / Упоряд.: Н.Р. Крива, Н.І. Блащак. Тернопіль: ТНТУ, 2023. 40 с.

13. Кухар А. Мировой рынок BI: аналитика только начинается. URL: https://ko.com.ua/mirovoj_rynok_bi_analitika_tolko_nachinaetsya_74198 (Дата звернення: 12.05.2024).

14. Легенчук С. Ф., Завалій Т.О. Big Data в маркетинговій аналітиці: можливості та проблеми використання. *Проблеми теорії та методології бухгалтерського обліку, контролю і аналізу,* Вип. 1(54), 2023. С. 52–58.

15. Маркетинговая аналитика. Анализ размера и доли рынка. Тенденции роста и прогнозы (2024–2029 гг.). URL: https://www.mordorintelligence.com/ru/industry-reports/marketing-analytics-market (Дата звернення: 06.04.2024).

16. Обзор Google Analytics 4. URL: https://serpstat.com/ru/blog/obzor-google-analytics-4/ (Дата звернення: 23.02.2024).

17. Отчет о перспективах развития рынка услуг цифрового маркетинга. URL: https://exactitudeconsultancy.com/ru/reports/2437/digital-marketing-services-market/ (Дата звернення: 05.04.2024).

18. Приймак В. Математичні методи економічного аналізу / В. Приймак. Київ : Центр навчальної літератури, 2019. 296 с.

19. Рынок маркетинговой аналитики: текущий анализ и прогноз (2022-2030 гг.). URL: https://univdatos.com/ru/report/marketing-analytics-market/ (Дата звернення: 10.05.2024).

20. Світвуд А. Маркетингова аналітика. Як підкріпити інтуїцію даними / пер. з англ. О. Асташова. Київ : Наш формат, 2019. 152 с.

21. Стратегии, инструменты и тренды интернет маркетинга. URL: https://tilda.education/courses/marketing/internet-marketing-beginning#rec4390008 (Дата звернення: 21.03.2024).

22. Сховище даних, OLAP - куб. *Веб портал Тернопільського національного технічного університету ім. Івана Пулюя.* URL:

https://wiki.tntu.edu.ua/Сховище_даних,_OLAP_-_куб (Дата звернення: 16.04.2024).

23. Ткаченко А. Показатели эффективности интернет-маркетинга: ключевые метрики и KPI [Електронний ресурс]. Режим доступу: https://wezom.com.ua/blog/pokazateli-effektivnosti-internet-marketinga-klyuchevye-metriki-i-kpi (Дата звернення: 13.03.2024).

24. Топ 5: сервіси для маркетингової аналітики або не Google Analytics єдиним. URL: https://www.theinstapreneurs.com.ua/blog-posts/top-servisiv-dlya-marketingovoyi-analitiki (Дата звернення: 28.02.2024).

25. Транзакція у базах даних. URL: https://www.wikiwand.com/uk/Транзакція_(бази_даних)#Властивості_транзакцій (Дата звернення: 14.04.2024).

26. Удод Є. Порівняння сервісів мобільної аналітики: як вибрати оптимальний інструмент для ваших цілей. URL: https://netpeak.net/uk/blog/porivnyannya-servisiv-mobil-noi-analitiki-yak-vibrati-optimal-niy-instrument-dlya-vashikh-tsiley/ (Дата звернення: 13.05.2024).

27. Федорова Х. 7 дієвих маркетингових інструментів для бізнесу. URL: https://hub.kyivstar.ua/articles/7-diyevyh-marketyngovyh-instrumentiv-dlya-biznesu (Дата звернення: 18.05.2024).

28. Google Analytics 4: плюсы и минусы перехода на новую систему аналитики. URL: https://streamtele.com/ru/google-analytics-4-plyusy-i-minusy-perehoda-na-novuyu-sistemu-analitiki/ (Дата звернення: 23.02.2024).

29. HubSpot — CRM-система, CMS, платформа для управления маркетингом, саппортом и операциями. URL: https://partnerkin.com/services/hubspot (Дата звернення: 21.02.2024).

30. Allen, M. Relational Databases Are Not Designed For Scale. URL: https://www.progress.com/blogs/relational-databases-scale (Last acessed: 23.04.2024)

31. Astera Analytics Team. OLTP проти OLAP: дві сторони однієї медалі даних? URL: https://www.astera.com/ru/knowledge-center/oltp-and-olap/ (Дата звернення: 14.04.2024).

32. Apache Airflow Concepts. URL: https://airflow.apache.org/docs/apache-airflow/2.0.1/concepts.html (Last accessed: 04.05.2024).

33. Brin, S., & Page, L. *The anatomy of a large-scale hypertextual Web search*

engine [Електронний ресурс]. Режим доступу: https://snap.stanford.edu/class/cs224w-readings/Brin98Anatomy.pdf (Дата звернення: 06.05.2024).

34. Davies, V. The history of cloud computing. URL: https://cybermagazine.com/cloud-security/history-cloud-computing (Last accessed: 30.04.2024).

35. Digital Marketing Services Market / Annual Report. URL: https://exactitudeconsultancy.com/reports/2437/digital-marketing-services-market/ (Last accessed: 22.05.2024).

36. Dimensional Research. A Global Survey of Data and Analytics Professionals [Електронний ресурс]. Режим доступу: https://get.fivetran.com/rs/353-UTB-444/images/Dimensional Research Data Analyst Survey Report 6.5.20.pdf (Дата звернення: 26.04.2024).

37. Duarte, F. Amount of Data Created Daily in 2024. URL: https://explodingtopics.com/blog/data-generated-per-day (Last accessed: 11.04.2024).

38. Hadoop Architecture and Components Explained. URL: https://www.simplilearn.com/tutorials/hadoop-tutorial/hadoop-architecture (Last accessed: 17.05.2024).

39. Harenslak, B., De Ruiter, J. Data Pipelines with Apache Airflow. Shelter Island (NY): Manning, 2021. 502 с.

40. IBM Marketing Cloud. 10 Key Marketing Trends for 2017 [Електронний ресурс]. Режим доступу: https://paulwriter.com/wp-content/uploads/2017/10/10-Key-Marketing-Trends-for-2017.pdf (Дата звернення: 20.04.2024).

41. Imran Abdul Rauf. What are MPP Systems - Benefits, Types and Examples. URL: https://www.royalcyber.com/blogs/what-is-massively-parallel-processing-mpp/ (Last accessed: 24.04.2024).

42. Khan, B., Jan, S., Khan, W., & Chughtai, M. *An Overview of ETL Techniques, Tools, Processes and Evaluations in Data Warehousing*. URL: https://www.techscience.com/jbd/v6n1/55252/html (Last accessed: 29.04.2024).

43. Mayer-Schönberger V., & Cukier, K. *Big Data: A Revolution that Will Transform how We Live, Work, and Think*. New York: Houghton Mifflin Harcourt, 2013. 242 с. 6-9.

44. Melnik, S., Gubarev, A, Jing Jing, L., Romer, G., Shivakumar, S., Tolton,

M. & Vassilakis, T. *Dremel: Interactive Analysis of Web-Scale Datasets*. URL: https://research.google/pubs/dremel-interactive-analysis-of-web-scale-datasets-2/ (Last accessed: 25.04.2024).

45. Monnappa, A. The History and Evolution of Digital Marketing. URL: https://www.simplilearn.com/history-and-evolution-of-digital-marketing-article (Last accessed: 27.04.2024).

46. Russel, S.J. & Norvig, P. Artificial Intelligence: A Modern Approach. Lnd: Prentice Hall, 2010.

47. Shahangian, M. Powering big data at Pinterest. URL: https://medium.com/pinterest-engineering/powering-big-data-at-pinterest-3c4836e2b112 (Last accessed: 30.04.2024).

48. Simitsis, A., Skiadopoulos, S. & Vassiliadis, P. (2023). The History, Present, and Future of ETL Technology [Електронний ресурс]. Режим доступу: https://ceur-ws.org/Vol-3369/invited1.pdf (Дата звернення: 27.04.2024).

49. Snowflake, Redshift, BigQuery, and Others: Cloud Data Warehouse Tools Compared. URL: https://www.altexsoft.com/blog/snowflake-redshift-bigquery-data-warehouse-tools (Last accessed: 24.04.2024).

50. Snowflake schemas. URL: https://www.ibm.com/docs/en/ida/9.1.2?topic=schemas-snowflake (Last accessed: 11.05.2024).

51. Taylor, D. What is Data Mart in Data Warehouse? Types & Example. URL: https://www.guru99.com/data-mart-tutorial.html (Last accessed: 20.04.2024).

52. Verified Market Reports. Global ETL Tools Market Insights. URL: https://www.verifiedmarketreports.com/product/etl-tools-market/ (Last accessed: 02.05.2024).

53. Wakefield Research. The State of Data Management Report [Електронний ресурс]. Режим доступу: https://get.fivetran.com/rs/353-UTB-444/images/2021-CDL-Wakefield-Research.pdf (Дата звернення: 30.04.2024).

54. What is data integration? URL: https://www.ibm.com/topics/data-integration (Last accessed: 15.05.2024).

55. Wojdyla, R., & Baer, J. The Evolution of Hadoop at Spotify - Through Failures and Pain. URL: https://www.slideshare.net/slideshow/the-evolution-of-hadoop-at-spotify-through-failures-and-pain/45837771 (Last accessed: 29.04.2024).

ДОДАТКИ

«РОЗРОБЛЕННЯ ІНТЕГРОВАНОЇ ПРОГРАМНОЇ СИСТЕМИ АНАЛІТИКИ ДЛЯ ІНТЕРНЕТ-МАРКЕТИНГУ: ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА КОНВЕЄРУ ОБРОБКИ ДАНИХ»

ДОДАТОК А – Опис атрибутів денормалізованої структури даних

- "**purchases**" – набір, що включає детальну інформацію про здійснені в межах підприємства платежі.
  - user_id – унікальний ідентифікаційний номер ("**ID**") залученого покупця.
  - payment_gate – платіжний шлюз, що обслуговує транзакцію.
  - payment_type – обраний користувачем електронний спосіб оплати.
  - **Онлайн-ресурс/Веб-сайт:**
    - app_id – унікальний ID онлайн-ресурсу, який залучив користувача до продукції підприємства. Для більш ефективного впливу на локалізовані аудиторії, компанією були викуплені доменні імена з різною подачею інформації у декількох країнах світу.
    - app_name – назва онлайн-ресурсу.
  - Замовлення **на оплату:**
    - order_id – унікальний ID створеного клієнтом замовлення на покупку або запиту на повернення коштів. При отриманні замовлення, клієнту або підприємству виставляється рахунок для оплати (т.зв. invoice). Оскільки не можна гарантувати успішність платежу, система намагатиметься здійснити операцію кілька разів у випадку невдачі. При цьому, для відображення кожної такої спроби буде створено нове "замовлення".
    - order_date – дата додавання замовлення/запиту.
    - order_status – стан замовлення (успішна оплата, задоволений запит тощо).
    - order_currency – обрана валюта для здійснення платежу.
    - order_amount – обсяг транзакції в обраній валюті.
    - order_amount_in_usd – грошовий еквівалент транзакції в USD.
    - order_cur_rate_to_usd – фіксований курс обраної валюти до USD на момент затвердження транзакції.
  - **Виставлений рахунок:**
    - invoice_id – унікальний ID виставленого клієнту/підприємству рахунку. Після проходження платежу здійснюється купівля - підписка клієнта на продукт підприємства (subscription).
    - invoice_date – дата успішного проведення платежу за вибраним продуктом, або повернення коштів.
  - **Покупки (підписки):**
    - subscription_id – унікальний ID покупки користувача. Це значення прив'язане до користувача, і залишається постійним до скасування підписки однієї зі сторін угоди.
    - subscription_date – початкова дата купівлі продукту, інакше кажучи, перший термін підписки.
  - **Об'єкт купівлі – продукт:**
    - product_id – унікальний ID продукту, що надається підприємством, що є об'єктом інтересу користувача.
    - product_price – ціна продукту в USD.
  - **Пробний період використання:**
    - trial_duration – обрана користувачем тривалого пробного періоду.
    - product_trial_price – вартість пробного періоду в USD.
    - purch_is_trial – вказує, чи є об'єктом транзакції *пробна* підписка.
  - **Повторний платіж:**
    - rebill_duration – кількість днів до закінчення поточного періоду оплати.
    - rebill_duration_group – кількість оплачених користувачем днів користування продуктом.
  - **Повернення коштів:**
    - refund_amount - обсяг запитаного повернення коштів у вибраній користувачем валюті.
    - refund_amount_in_usd – обсяг запитаного повернення коштів у USD.
    - refund_date – дата повернення коштів покупцю у повному обсязі.

- "**subscriptions**" – набір, що описує часові метрики оформлених підписок.
  - subscription_id – унікальний ID покупки користувача.
  - subscription_date – початкова дата останнього оплаченого періоду.
  - expired_at – дата закінчення поточного періоду оплати.
  - created_at – дата першої здійсненої підписки по продукту.
  - next_charge_at – дата виставлення наступного рахунку клієнту.
  - cancelled_at – дата скасування передплати користувачем.

- "**users_attribution**" – набір, що містить дані про ефективність каналів залучення покупців електронного маркетингу:
  - user_id – унікальний ID залученого покупця.
  - created_at – дата залучення потенційного покупця.
  - app_id - унікальний ID ресурсу, що залучив покупця.
  - country – повна назва країни проживання користувача.
  - country_code – подання країни у скороченому (кодовому) форматі.
  - media_source - назва джерела, звідки здійснено перехід на рекламний ресурс.

- "**insights**" – набір даних, що описують фінансові витрати підприємства на проведення рекламних кампаній у різних країнах:
  - date – дата проведення рекламної кампанії.
  - country – країна проведення кампанії.
  - spend – обсяг інвестицій у рекламну кампанію.

- "**countries**" – допоміжний набір даних, описуючий країни, у яких підприємство проводило маркетингові кампанії.
  - - name – повна назва країни;
  - - code – скорочене представлення країни у форматі "ISO ALPHA-2";
  - - code_iso_3 – скорочене уявлення країни у форматі "ISO 3166".

ДОДАТОК Б

Лістинг SQL-запиту створення моделі «user_purchases»

```
CREATE OR REPLACE TABLE final.user_purchases
    PARTITION BY
        DATE_TRUNC(acquisition_date, DAY)
AS (
    WITH grouped_actual_revenue AS (
        SELECT
          user_id
          , subscription_id
          , product_id
           -- ...
          -- інші атрибути проєкції (див. рис. 3.2.x вище)
          -- ...

          , MAX(order_date)                                          AS last_order_date
          , MAX(COALESCE(rebill_number, 1))                          AS last_rebill_number
          , SUM(order_amount_in_usd)                                 AS sum_order_amount_in_usd
          , COUNT(refund_date)                                       AS number_of_refunds
          , SUM(IF(refund_amount_in_usd IS NULL, 0, refund_amount_in_usd))  AS sum_refund_amount_in_usd
          , MIN(refund_date)                                         AS first_refund_date

         FROM final.actual_revenue

         GROUP BY
          user_id
          , subscription_id
          , product_id
          -- ...
          -- інші не-агреговані атрибути, що оминуті у проєкції вище
          -- ...
    ),

    expected_revenue_data AS (
        SELECT
          er.user_id                                                 AS user_id
          , er.product_id                                            AS product_id
          , er.subscription_id                                       AS subscription_id
          , COALESCE(er.expected_revenue, 0)                         AS expected_revenue
          , er.rebill_rate_default                                   AS rebill_rate_forecast
          , er.rebill_rate                                           AS rebill_rate_residuals
          , er.refund_rate                                           AS refund_rate_forecast
       FROM final.expected_revenue AS er
    ),
```

```
  -- підтягуємо дати оформлення/відміни/наступної сплати по підписці
  subscriptions AS (
      SELECT
        subscription_id
        , DATE(expired_at)       AS expired_at
        , DATE(next_charge_at)   AS next_charge_at
        , cancelled_at
     FROM initial.subscriptions
  )

  SELECT
      grouped_actual_revenue.*
      , IF(last_rebill_number = 0, True, False)                    AS is_trial
      , CASE
            WHEN number_of_refunds > 0
                THEN 'refunded'
            ELSE 'approved'
         END                                                       AS last_order_status

      , subs.* EXCEPT (subscription_id)

      , sum_order_amount_in_usd - sum_refund_amount_in_usd         AS revenue
      , expected_revenue
      , sum_order_amount_in_usd - sum_refund_amount_in_usd
          + expected_revenue                                       AS ful_expected_revenue

      , rebill_rate_forecast                                       AS rebill_rate_forecast
      , rebill_rate_residuals                                      AS rebill_rate_residuals
      , refund_rate_forecast                                       AS refund_rate_forecast

  FROM grouped_actual_revenue AS base

  LEFT JOIN subscriptions subs
      USING (subscription_id)

  LEFT JOIN expected_revenue_data
      USING (user_id, product_id, subscription_id)

  LEFT JOIN exodus.countries AS countries
      ON base.country_code = countries.code
;
```

ДОДАТОК В

Лістинг SQL-запиту створення моделі «final_report»

```
CREATE OR REPLACE TABLE final.final_report
    PARTITION BY
        acquisition_date
AS (
    WITH costs AS (
       SELECT
           `date` AS acquisition_date
           , 'facebook' AS media_source
           , country AS country_code
           , countries.name AS country
           , spend AS cost
        FROM `initial.insights` AS ins
        LEFT JOIN `initial.countries` AS countries
            ON ins.country = countries.code
    ),

    ceo_report AS (
        SELECT
            DATE(acquisition_date)                          AS acquisition_date
            , app_name, app_id
            , country_code, country
            , media_source
            , COUNT(DISTINCT user_id)                       AS purchases -- усі покупки
            , SUM(IF(NOT is_trial, 1, 0))                   AS payments -- усі покупки без пробних періодів
            , SUM(revenue)                                  AS revenue
            , SUM(expected_revenue)                         AS expected_revenue
            , SUM(ful_expected_revenue)                     AS full_expected_revenue
            , SUM(number_of_refunds)                        AS refunds
            , COUNT(DISTINCT
                IF(
                  DATE_DIFF(cancelled_at, acquisition_date, DAY) < 1,
                  user_id, NULL
                )
              )                                             AS cancellations_1d
            , COUNT(DISTINCT
                IF(DATE_DIFF(cancelled_at, acquisition_date, DAY) < 7, user_id, NULL)
              )                                             AS cancellations_7d
            , COUNT(DISTINCT
                IF(DATE_DIFF(cancelled_at, acquisition_date, DAY) < 30, user_id, NULL)
              )                                             AS cancellations_30d
            , COUNT(DISTINCT
                IF(cancelled_at IS NOT NULL, user_id, NULL)
              )                                             AS cancellations

        FROM final.user_purchases AS purchases
        GROUP BY
            DATE(acquisition_date)
            , app_name
            , app_id
            , country_code
            , country
            , media_source
```

```
-- підтягуємо дати оформлення/відміни/наступної сплати по підписці
subscriptions AS (
    SELECT
      subscription_id
      , DATE(expired_at)       AS expired_at
      , DATE(next_charge_at)   AS next_charge_at
      , cancelled_at
   FROM initial.subscriptions
)

SELECT
    grouped_actual_revenue.*
    , IF(last_rebill_number = 0, True, False)                      AS is_trial
    , CASE
          WHEN number_of_refunds > 0
              THEN 'refunded'
          ELSE 'approved'
       END                                                         AS last_order_status

    , subs.* EXCEPT (subscription_id)

    , sum_order_amount_in_usd - sum_refund_amount_in_usd           AS revenue
    , expected_revenue
    , sum_order_amount_in_usd - sum_refund_amount_in_usd
        + expected_revenue                                         AS ful_expected_revenue

    , rebill_rate_forecast                                         AS rebill_rate_forecast
    , rebill_rate_residuals                                        AS rebill_rate_residuals
    , refund_rate_forecast                                         AS refund_rate_forecast

FROM grouped_actual_revenue AS base

LEFT JOIN subscriptions subs
    USING (subscription_id)

LEFT JOIN expected_revenue_data
    USING (user_id, product_id, subscription_id)

LEFT JOIN exodus.countries AS countries
    ON base.country_code = countries.code
```

ДОДАТОК Г

Результати вирішення мінімаксної задачі по оптимальному розподілу маркетингового бюджету замовника

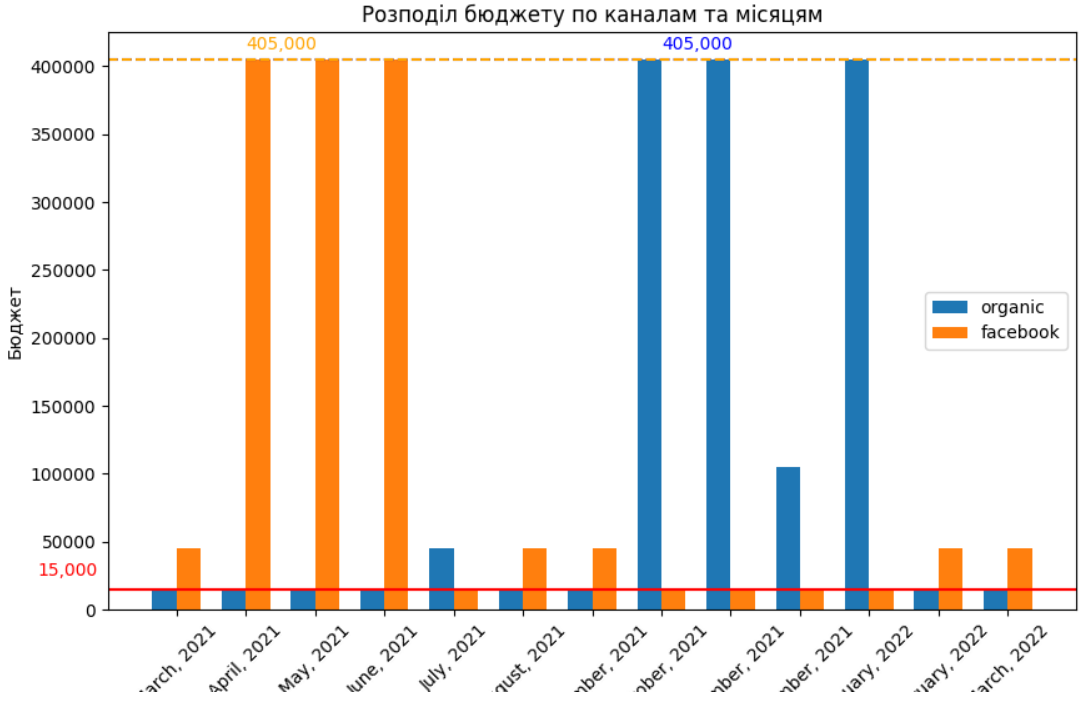
Набір вхідних показників:

B = 3_000_000; b_min = 15_000

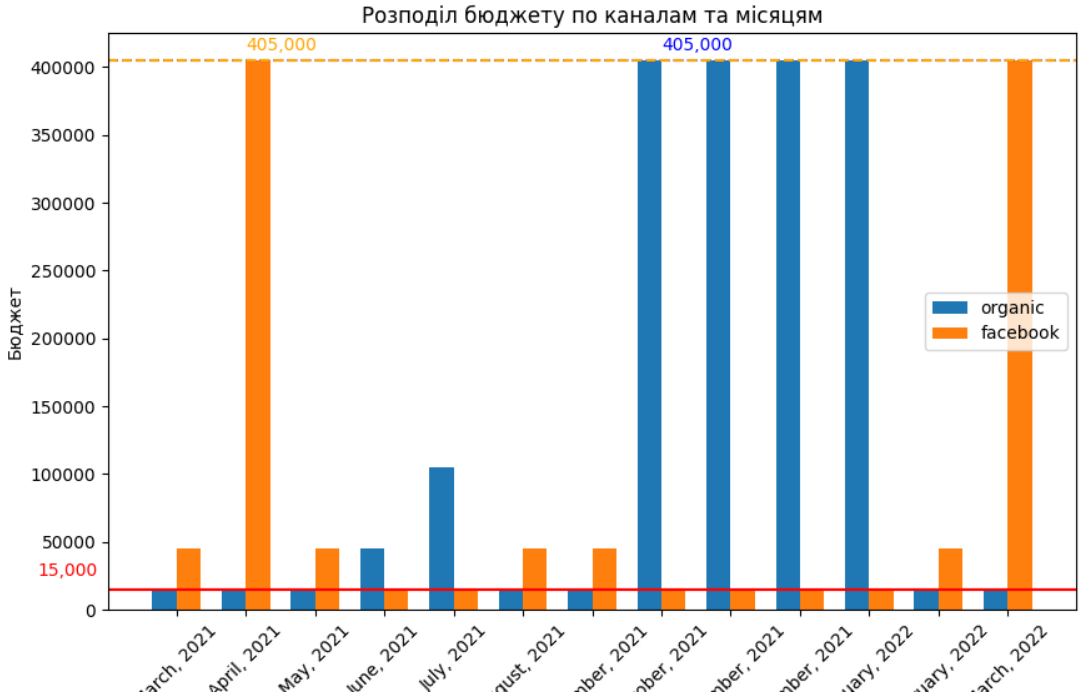
$B\_min, organic$ = 300_000; $B\_min, facebook$ = 350_000

$B\_max, organic$ = Infinity; $B\_max, facebook$ = Infinity

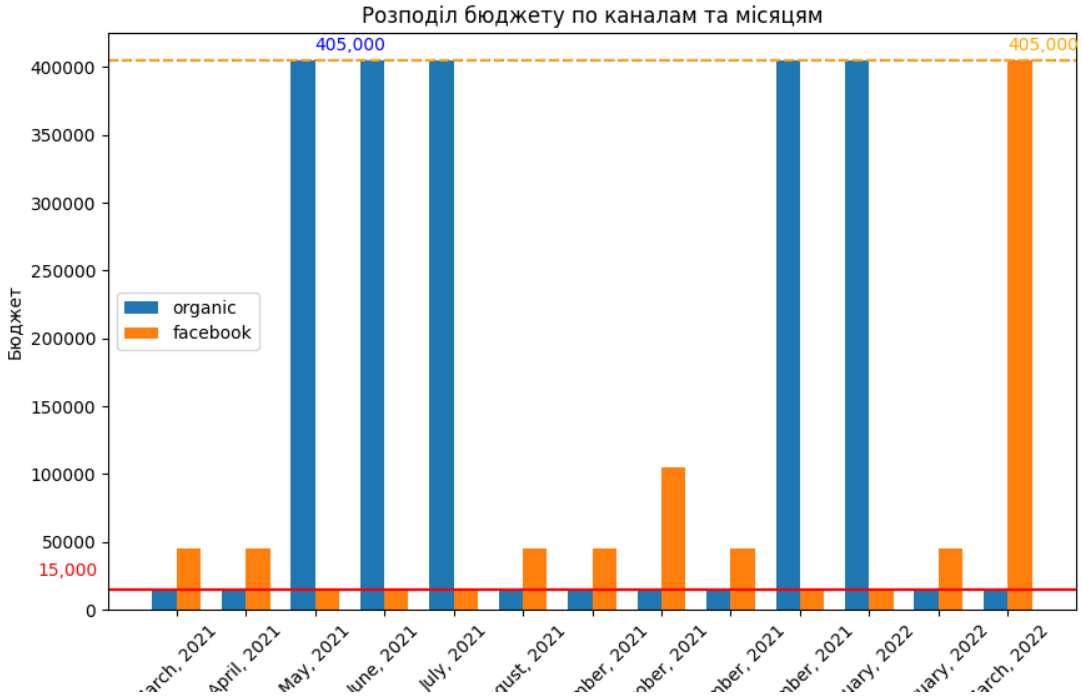
t_min = 14%; t_max = 2%



При $\alpha = 1$ та $\beta = 0$.



При $\alpha = 0.5$ та $\beta = 0.5$.



При $\alpha = 0$ та $\beta = 1$.

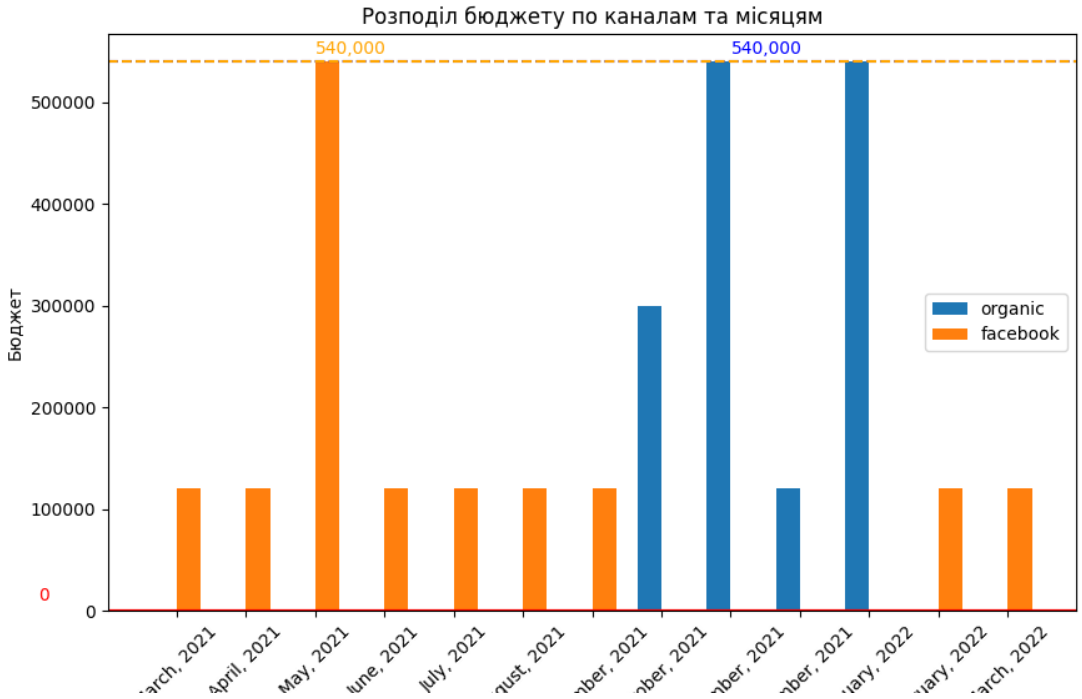
Набір вхідних показників:

B = 3_000_000; b_min = 0

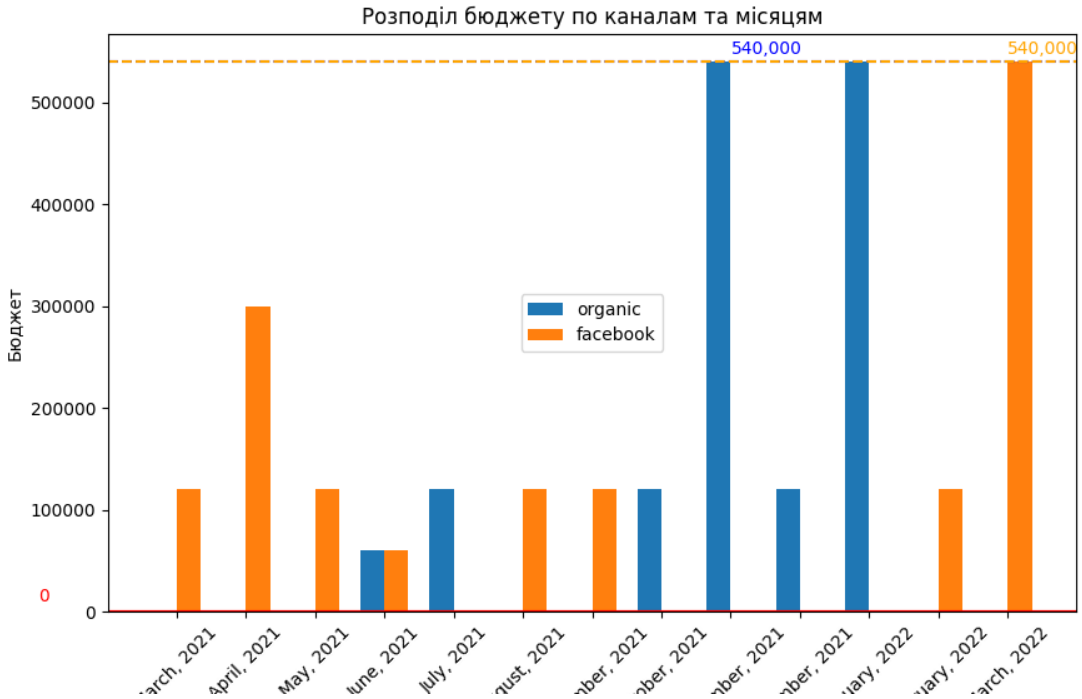
$B\_min, organic$ = 50_000; $B\_min, facebook$ = 100_000

$B\_max, organic$ = 1_500_000; $B\_max, facebook$ = 1_500_000

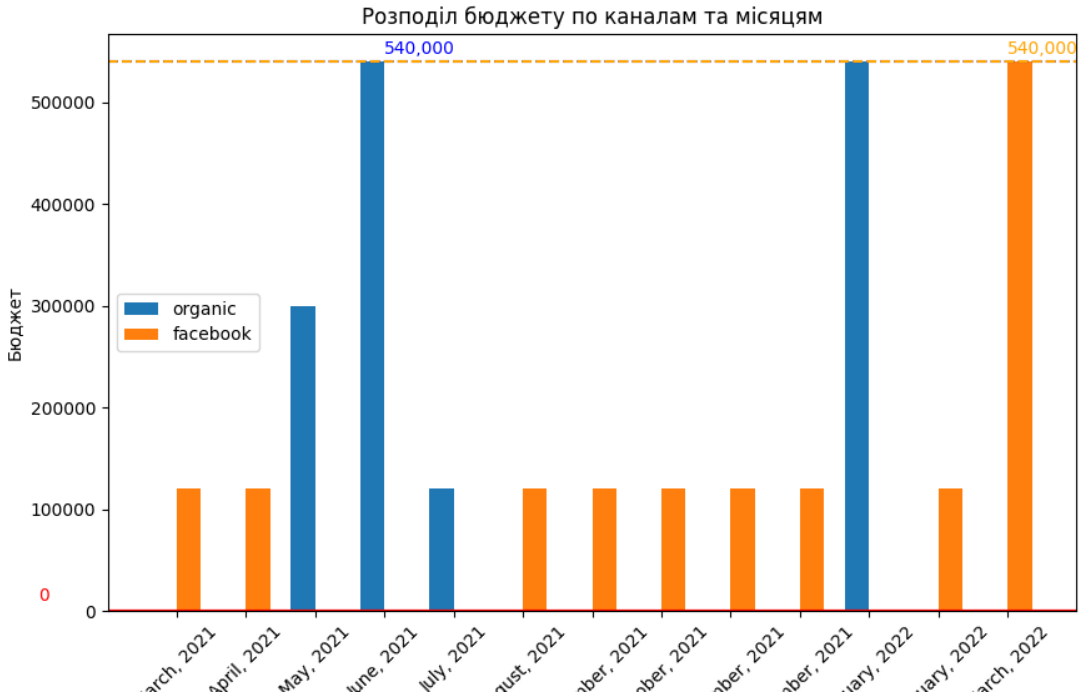
t_min = 18%; t_max = 4%



При $\alpha = 1$ та $\beta = 0$.



При $\alpha = 0.5$ та $\beta = 0.5$.



При $\alpha = 0$ та $\beta = 1$.

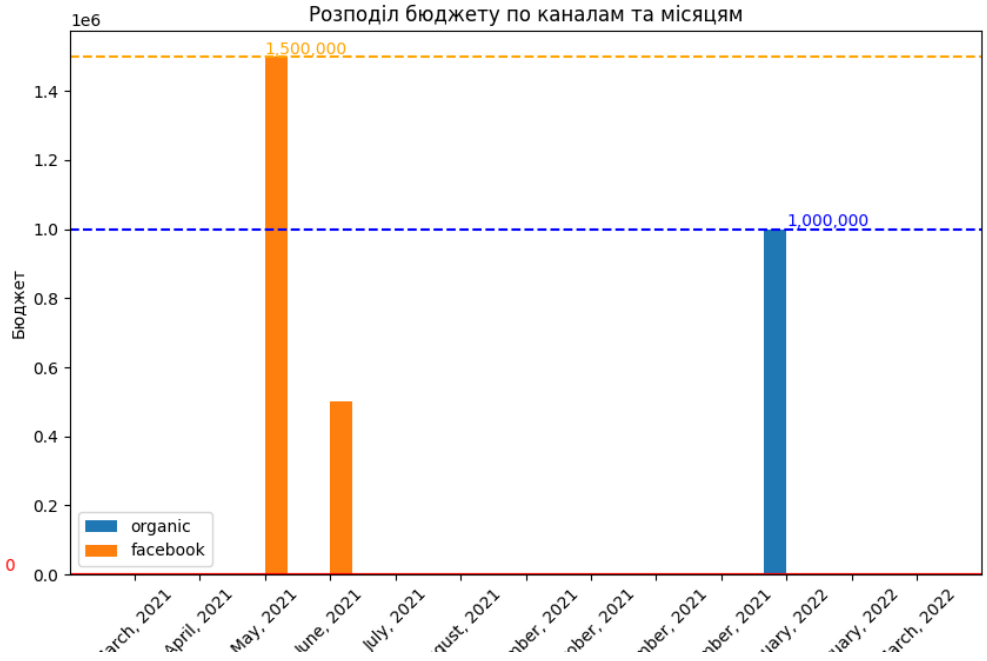
Набір вхідних показників:

B = 3_000_000; b_min = 0

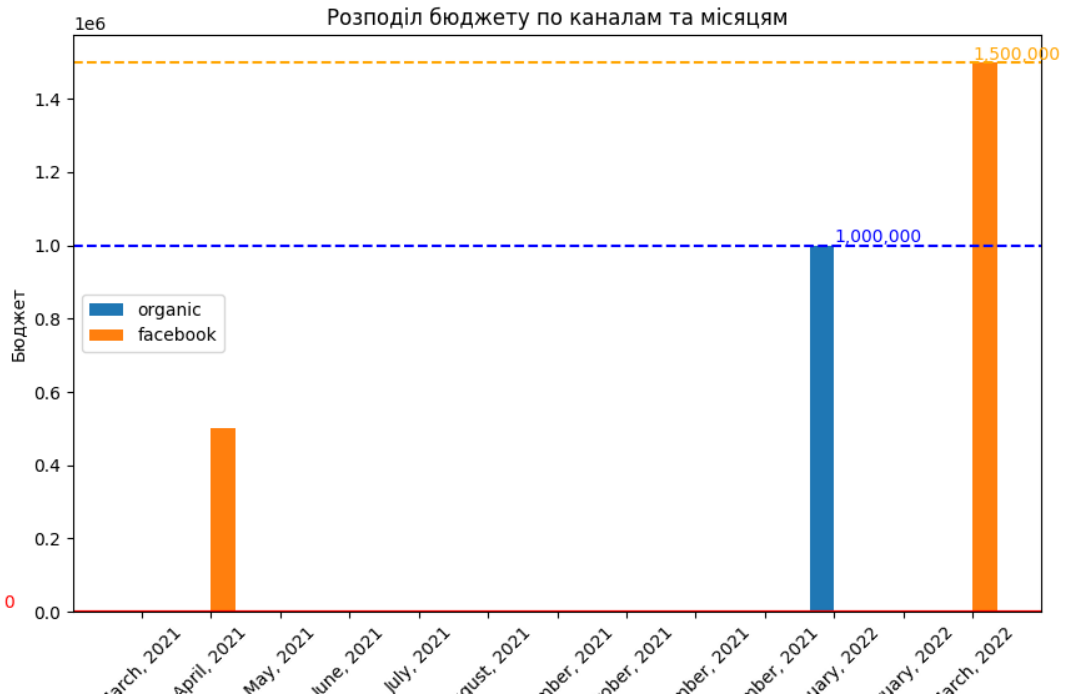
$B\_min, organic$ = 50_000; $B\_min, facebook$ = 100_000

$B\_max, organic$ = 1_500_000; $B\_max, facebook$ = 1_500_000

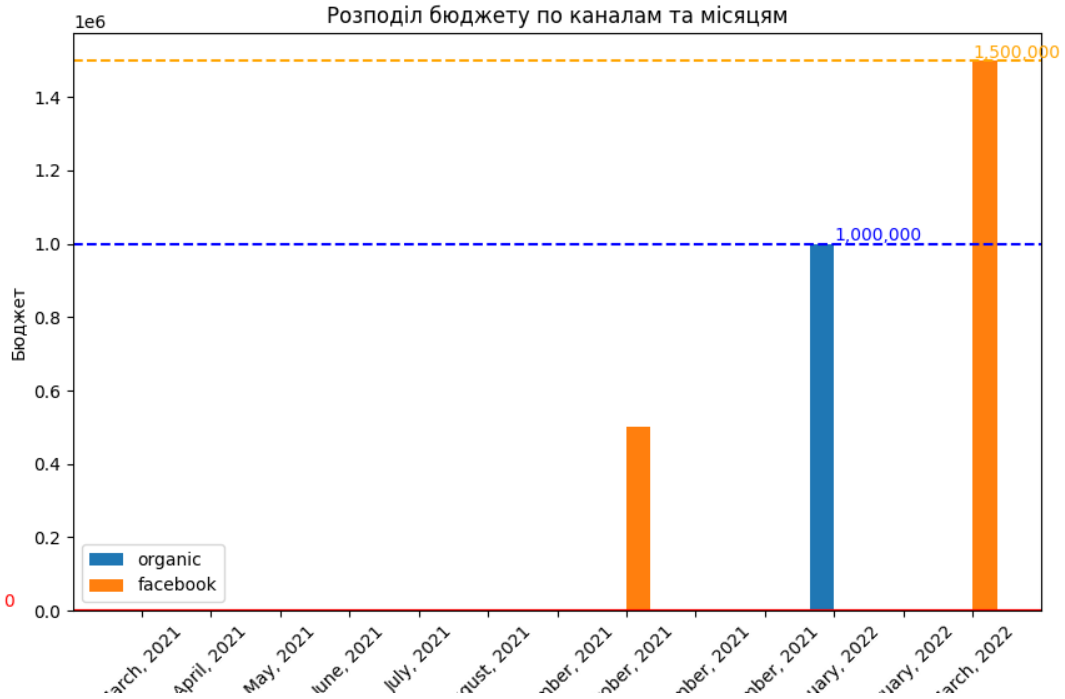
t_min = 18%; t_max = 4%



При $\alpha = 1$ та $\beta = 0$.



При $\alpha = 0.5$ та $\beta = 0.5$.



При $\alpha = 0$ та $\beta = 1$.

ДОДАТОК Д

Лістинг програмного коду вирішення мінімаксної задачі по оптимальному розподілу маркетингового бюджету замовника

Python

```python
from ortools.linear_solver.pywraplp import Solver
from itertools import product as cartesian
from .utils import *

def main(file_dataset: str) -> None:
    cli_args = parse_cli_arguments()
    channels, months, total_spent, total_gained, customers = read_csv_dataset(file_dataset)

    task_parameters, (t_max, t_min) = init_task_params_from_args(cli_args)
    total_budget, ch_min_budget, ch_max_budget, min_mon_budget, (alpha, beta) = task_parameters

    # "Z-Score" нормалізація задля приведення усіх даних до одного масштабу
    standardized_gained = {c: standardize(total_gained[c]) for c in channels}
    standardized_spent = {c: standardize(total_spent[c]) for c in channels}
    standardized_customers = {c: standardize(customers[c]) for c in channels}

    solver: Solver = Solver.CreateSolver('GLOP')

    # Вирішальні змінні: представляють бюджети для кожного каналу та місяця
    ch_budget_vars = {}
    for c, m in cartesian(channels, months):
        ch_budget_vars[c, m] = solver.NumVar(0, solver.infinity(), f'x[{c},{m}]')

    # Обмеження 1: Загальний бюджет - сума всіх budget_vars[c, m] не перевищує/рівна бюджету `B`.
    solver.Add(sum(ch_budget_vars[c, m] for c in channels for m in months) == total_budget)

    # Обмеження 2: Мінімальний та максимальній бюджети для кожного каналу за весь період.
    for c in channels:
        solver.Add(sum(ch_budget_vars[c, m] for m in months) >= ch_min_budget[c])
        solver.Add(sum(ch_budget_vars[c, m] for m in months) <= ch_max_budget[c])

    # Обмеження 3: Мінімальний бюджет `b_min` на кожний канал кожного місяця.
    for c, m in cartesian(channels, months):
        solver.Add(ch_budget_vars[c, m] >= min_mon_budget)

    # Обмеження 4: Місячні бюджети не можуть бути меньше/перевищувати частину загального бюджету `B`.
    for m in months:
        solver.Add(sum(ch_budget_vars[c, m] for c in channels) <= t_max * total_budget)
        solver.Add(sum(ch_budget_vars[c, m] for c in channels) >= t_min * total_budget)
```

```
    # Цільова функція
    objective = solver.Objective()

    for c, m in cartesian(channels, months):
        idx = months.index(m)
        roi_gained = calculate_roi(standardized_gained[c][idx], standardized_spent[c][idx])
        if total_spent[c][idx] > 0:
            cac_customers = standardized_customers[c][idx] / standardized_spent[c][idx]
        else:
            cac_customers = 0

        # разуємо значенняч цілбової функції для канала `r` у місяць `m`
        combined_metric = alpha * roi_gained + beta * cac_customers
        objective.SetCoefficient(ch_budget_vars[c, m], combined_metric)

    # мета задачі - максимізація цільової функції
    objective.SetMaximization()
    task_status = solver.Solve()

    # якщо рішення не є оптимальним або можливим, то результатів немає
    if task_status not in (Solver.OPTIMAL, Solver.FEASIBLE):
        return

    print(f'Значення максимізованої функції: {objective.Value()}')

    budgets = {c: [ch_budget_vars[c, m].solution_value() for m in months] for c in channels}
    visualize_result_plot(channels, months, budgets, min_mon_budget)


if __name__ == '__main__':
    main('./data.csv')
```

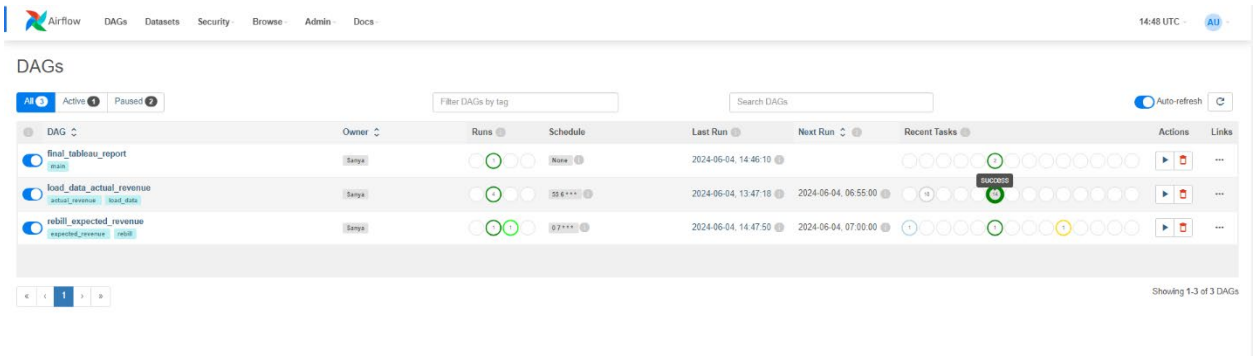
Python

```python
# utils.py
import math
import argparse

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np


def read_csv_dataset(file_name: str):
    df = pd.read_csv(file_name)

    channels: pd.DataFrame = df['attribution_channel'].unique()
    months: pd.DataFrame = df['month_date'].unique().tolist()

    total_spent = {channel: [0] * len(months) for channel in channels}
    total_gained = {channel: [0] * len(months) for channel in channels}
    customers = {channel: [0] * len(months) for channel in channels}

    for channel in channels:
        for i, month in enumerate(months):
            entry = df[(df['attribution_channel'] == channel) & (df['month_date'] == month)]
            if not entry.empty:
                total_spent[channel][i] = entry['total_spent'].values[0]
                total_gained[channel][i] = entry['total_gained'].values[0]
                customers[channel][i] = entry['customers_attracted'].values[0]

    return channels, months, total_spent, total_gained, customers


def parse_cli_arguments():
    parser = argparse.ArgumentParser(description='Оптимизация маркетингового бюджета.')

    parser.add_argument('-t', '--total_budget', type=float, default=3_000_000)
    parser.add_argument('-mino', '--min_budget_organic', type=float, default=300_000)
    parser.add_argument('-maxo', '--max_budget_organic', type=float, default=math.inf)
    parser.add_argument('-minfb', '--min_budget_facebook', type=float, default=350_000)
    parser.add_argument('-maxfb', '--max_budget_facebook', type=float, default=math.inf)
    parser.add_argument('-m', '--min_monthly_budget', type=float, default=15_000)
    parser.add_argument('-a', '--alpha', type=float, required=False, default=None)
    parser.add_argument('-b', '--beta', type=float, required=False, default=None)

    return parser.parse_args()
```

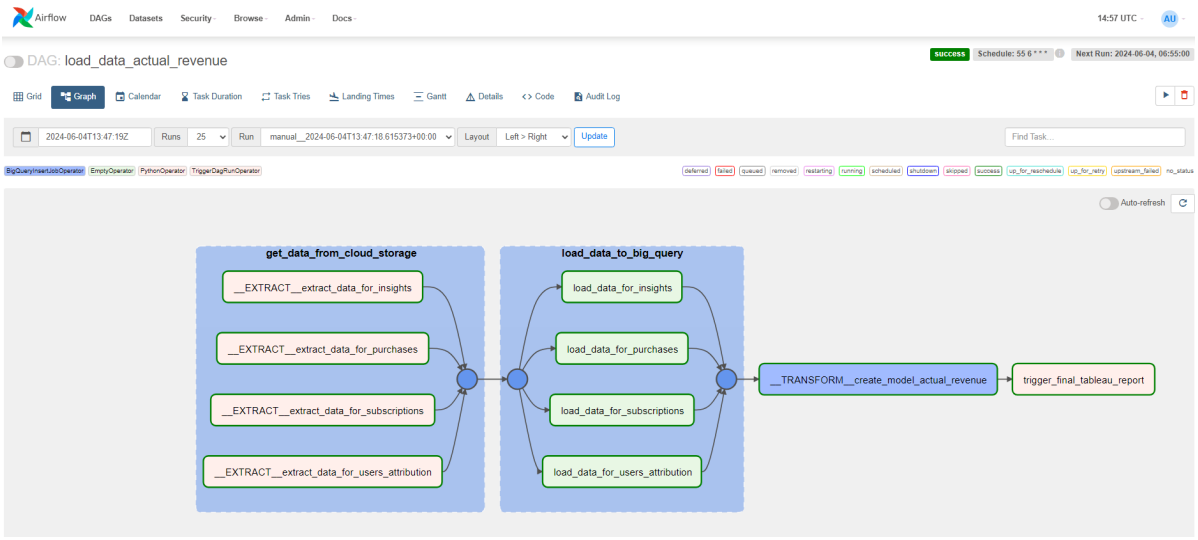
```python
def init_task_params_from_args(args: argparse.Namespace):
    total_budget = args.total_budget
    min_month_budget = args.min_monthly_budget
    ch_min_budget = {'organic': args.min_budget_organic, 'facebook': args.min_budget_facebook}
    ch_max_budget = {'organic': args.max_budget_organic, 'facebook': args.max_budget_facebook}
    alpha, beta = args.alpha, args.beta

    if alpha is not None and beta is None:
        beta = 1 - alpha
    elif alpha is None and beta is not None:
        alpha = 1 - beta
    elif None in (alpha, beta) or alpha + beta != 1:
        raise ValueError(f"Задані значення {alpha=} та {beta=} не є вірними.")

    return total_budget, ch_min_budget, ch_max_budget, min_month_budget, (alpha, beta)

def visualize_result_plot(channels: list, months: list, best_budgets: dict, month_budget: float):
    fig, ax = plt.subplots(figsize=(10, 6))

    ind = np.arange(len(months))
    width = 0.35

    for i, c in enumerate(channels):
        budget = best_budgets[c]
        ax.bar(ind + (i - 0.5) * width, budget, width, label=c)

    ax.set_xlabel('Місяці'); ax.set_ylabel('Бюджет')
    ax.set_title('Розподіл бюджету по каналам та місяцям')
    ax.set_xticks(ind); ax.set_xticklabels(months, rotation=45)
    ax.legend()

    # Додавання пунктирних ліній для піків інвестицій
    for c in channels:
        max_budget_month = months[np.argmax(best_budgets[c])]
        max_budget_value = max(best_budgets[c])
        color = 'orange' if c == 'facebook' else 'blue'

        ax.axhline(y=max_budget_value, color=color, linestyle='--')
        ax.text(months.index(max_budget_month), max_budget_value + 7500, f'{int(max_budget_value):,}', color=color)


    ax.axhline(y=month_budget, color='red', linestyle='solid', linewidth=1.6)
    ax.text(-2, month_budget + 10_000, f'{int(month_budget):,}', color='r')

    plt.show()

def calculate_roi(net_gain: float, net_spend: float) -> float:
    return 0 if net_spend == 0 else (net_gain - net_spend) / net_spend

def standardize(data) -> list:
    mean = np.mean(data)
    std = np.std(data)
    return [(x - mean) / std for x in data]
```

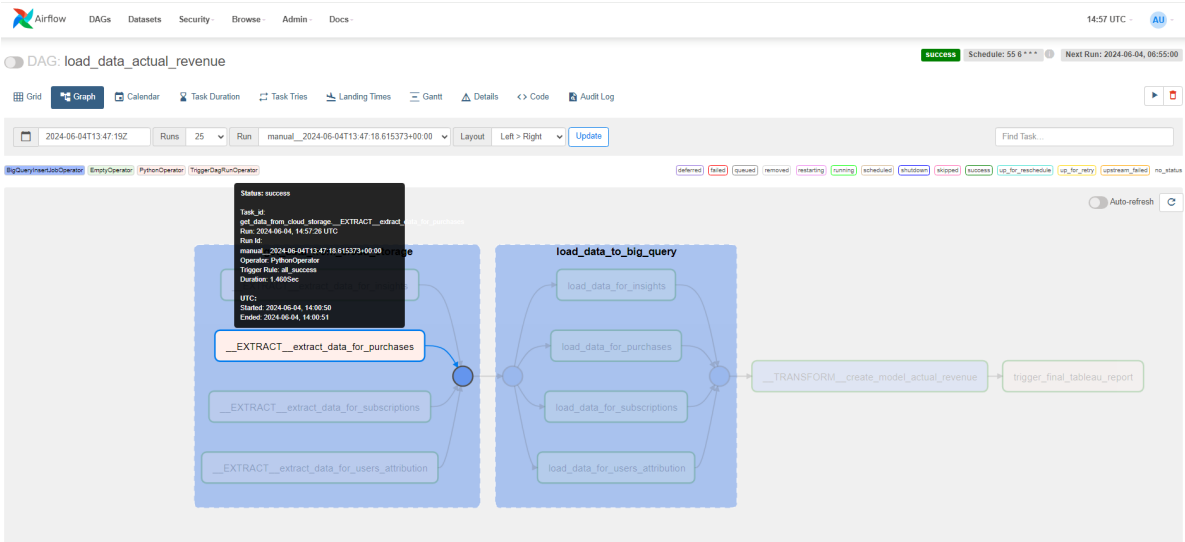
ДОДАТОК Е

Відображення послідовності етапів загального конвеєра обробки та інтеграції даних у веб-інтерфейсі оркестратора Apache Airflow

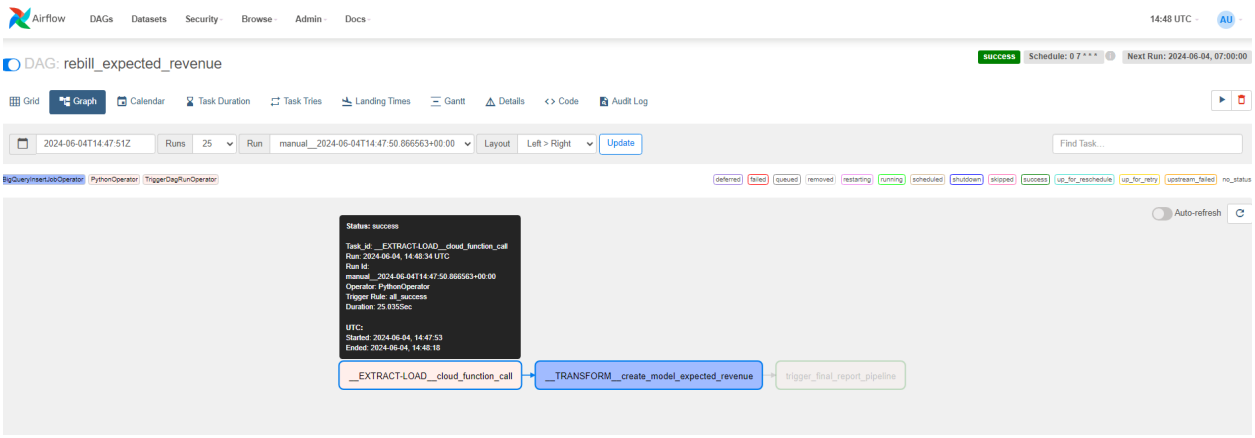


Огляд етапів загального конвеєра обробки та інтеграції інформації розробленої системи маркетингової аналітики

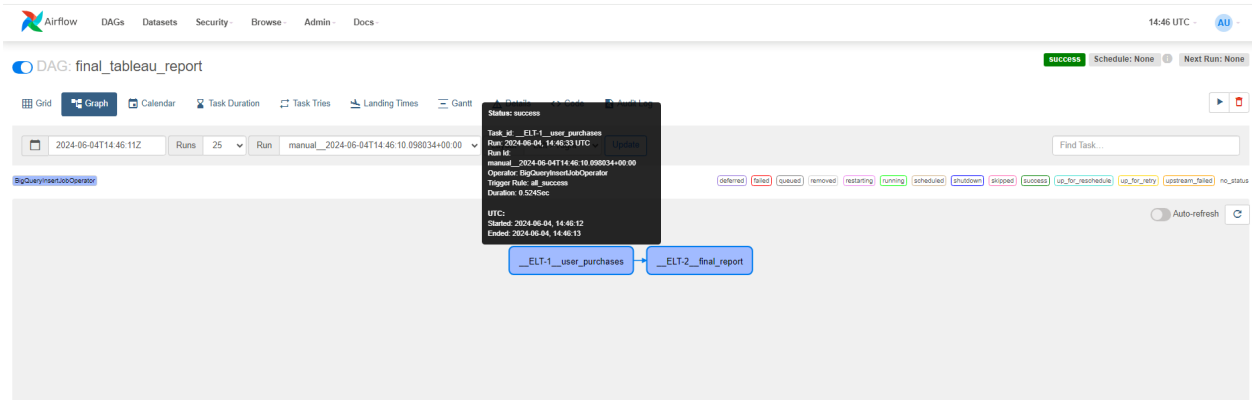




Етап підготовчого конвеєру даних, (що функціонує за методологією ELT), створення обробленої та денормалізованої моделі-агрегата



Етап аналітичного конвеєру даних (що функціонує за методологією ELT), та створення прогнозної моделей на основі алгоритмічного аналізу



Етап фінального етапу зведення даних та створення інформаційної бази для подальшої бізнес-аналітики