НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«На правах рукопису»
УДК 004.65

До захисту допущено:
Завідувач кафедри
_____ Віталій РОМАНКЕВИЧ
«___»_____2021 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою

«Системне програмування і спеціалізовані комп'ютерні системи»

зі спеціальності 123 «Комп'ютерна інженерія»

на тему: «Засоби забезпечення інтерактивного аналізу даних у веборієнтованих сервісах на основі багатовимірних OLAP кубів»

Виконав:
студент ІІ курсу, групи КВ-01мп
Олійник Олександр Валерійович _____

Науковий керівник:
Доцент кафедри СПСКС, к.т.н., доцент,
Тарасенко-Клятченко Оксана Володимирівна _____
_____

Рецензент:
_____
_____
_____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2021 року

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп‘ютерних систем

Рівень вищої освіти – другий (магістерський)

Спеціальність – 123 «Комп‘ютерна інженерія»

Освітньо-професійна програма «Системне програмування і спеціалізовані комп‘ютерні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ

«___»_____2021 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Олійнику Олександру Валерійовичу

1. Тема дисертації «Засоби забезпечення інтерактивного аналізу даних у веборієнтованих сервісах на основі багатовимірних OLAP кубів», науковий керівник дисертації Тарасенко-Клятченко Оксана Володимирівна, доцент кафедри СПСКС, кандидат технічних наук, затверджені наказом по університету від «5» 11. 2021 р. №3682-С

2. Термін подання студентом дисертації 7.12.2021

3. Об’єкт дослідження:

- технологія OLAP та її застосування для багатовимірного аналізу даних.

4. Вихідні дані:

- технологія з використання OLAP кубів та інтерактивний аналіз даних.

5. Перелік завдань, які потрібно розробити:

- Проаналізувати OLAP-технологію.

- Реалізувати інтерактивний аналіз, що ґрунтується на використанні OLAP-технологій.

6. Перелік графічного (ілюстративного) матеріалу:

- Презентація.

7. Перелік публікацій:

1) Прикладна математика та комп'ютинг. «XIV науково-практична конференція магістрантів та аспірантів ПМК-2021 факультету прикладної математики».

2) Міжнародний науково-технічний журнал «Інформаційні технології та комп'ютерна інженерія».

9. Дата видачі завдання 7.10.2020

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Вивчення літератури за тематикою МД	01.10.2020	
2.	Розроблення та узгодження технічного завдання	05.10.2020	
3.	Аналіз існуючих рішень	14.12.2020	
4.	Підготовка матеріалів першого розділу МД	18.05.2021	
5.	Розроблення програмного забезпечення	16.09.2021	
6.	Відлагодження програмного продукту	14.10.2021	
7.	Підготовка матеріалів другого розділу МД	19.10.2021	
8.	Розроблення інтерфейсу програмного забезпечення	26.10.2021	
9.	Оформлення документації дипломного МД	29.11.2021	
10.	Проходження попереднього захисту	30.11.2021	

Студент Олександр ОЛІЙНИК

Науковий керівник Оксана ТАРАСЕНКО-КЛЯТЧЕНКО

РЕФЕРАТ

Актуальність теми. У сфері програмного та апаратного забезпечення ведеться постійний розвиток та вдосконалення різних методів обробки даних. На будь-якому підприємстві зараз використовуються сховища даних, бази даних та системи управління базами даних. Разом з удосконаленням інформаційно-комунікаційних технологій загалом удосконалюються і бази даних. Їх структура стає складнішою, а обсяг розв'язуваних задач більшим.

Наприклад, реляційні бази даних дозволяють подавати дані про предметну область за допомогою двовимірних таблиць, між якими встановлюється зв'язок. Вони дозволяють відобразити інформацію у найбільш доступній для розуміння користувача формі. Проте проблеми реляційної бази даних у тому, що її розробка є трудомісткою, а швидкість доступу до даних є низькою. Реляційні бази даних здатні бездоганно обробляти масиви даних, що мають невелику кількість вимірів, але вони не відповідають вимогам глибшого аналізу даних. Саме тому стає актуальною проблема застосування OLAP-технологій, що передбачають використання багатовимірних кубів, зокрема в бізнес сфері.

Об’єктом дослідження є аналіз технології OLAP та її застосування для багатовимірного аналізу даних.

Предметом дослідження є програмне забезпечення для інтерактивного аналізу даних та статистики у веборієнтованих сервісах.

Мета роботи: покращення засобів забезпечення інтерактивного аналізу даних на основі багатовимірних OLAP-кубів.

Інноваційність полягає в наступному: запропоновано швидший та якісніший інтерактивний аналіз даних у веборієнтованих сервісах з використанням OLAP-систем. Також використання OLAP-технологій забезпечує стабільно високий рівень продуктивності, обслуговуючи великі обсяги даних.

Практична цінність отриманих в роботі результатів полягає в тому, що використання OLAP-систем забезпечує стабільно високий рівень продуктивності та масштабованості, обслуговуючи великі обсяги даних, доступ до яких можуть отримати тисячі користувачів. З допомогою OLAP-технологій доступ до інформації здійснюється у реальному часі, обробка запитів не уповільнює процес аналізу, забезпечуючи його оперативність та ефективність. OLAP-система надає користувачам можливість проводити складний аналіз даних, що дозволяє краще зрозуміти принципи функціонування компанії та знайти способи покращення результатів її діяльності.

Аналітичні програми, побудовані на основі OLAP-технологій, дозволяють вирішувати цілий спектр аналітичних завдань, таких як аналіз виробництва, фінансовий аналіз, маркетингові дослідження, аналіз електронного бізнесу, аналіз трудових ресурсів. OLAP дає можливість створення аналітичних додатків, що охоплюють аналіз усього виробничого та фінансового циклу підприємства.

Апробація роботи. Основні положення і результати роботи були представлені та обговорювались на XIV науковій конференції магістрантів та аспірантів «Прикладна математика та комп'ютинг» ПМК-2021 (Київ, 17-19 листопада 2021 р.) та в статті міжнародного науково-технічного журналу «Інформаційні технології та комп'ютерна інженерія».

Структура та обсяг роботи. Магістерська дисертація складається з вступу, чотирьох розділів та висновків.

У вступі подано загальну характеристику роботи, зроблено оцінку сучасного стану проблеми, обґрунтовано актуальність напрямку досліджень, сформульовано мету і задачі досліджень, показано наукову новизну отриманих результатів і практичну цінність роботи, наведено відомості про апробацію результатів і їхнє впровадження.

У першому розділі розглянуто наведено загальні відомості про OLAP-системи та наведено основні вимоги до систем з використанням OLAP.

У другому розділі наведено аналіз технологій розробки програмного забезпечення з використанням Data Mining та OLAP-технологій.

У третьому розділі та четвертому розділах наведено структуру та опис роботи програмного забезпечення, а також проведено тестування та проаналізовано результати виконаного дослідження.

У висновках представлені результати проведеної роботи.

Робота представлена на 85 аркушах, містить посилання на список використаних літературних джерел.

Ключові слова: OLAP технології, Data Mining, OLAP куби, гіперкуб, бізнес аналітика.

ABSTRACT

Actuality of theme. In the field of software and hardware is constantly evolving and improving. Databases, databases and database management systems are now used in any enterprise. Along with the improvement of information and communication technologies in general, databases are also being improved. Their structure becomes more complex, and the volume of tasks to be solved is larger.

For example, relational databases allow you to present data about the subject area using two-dimensional tables, between which a relationship is established. They allow you to display information in the most user-friendly form. However, the problem with a relational database is that its development is time consuming and the speed of data access is low. Relational databases are able to seamlessly process data sets that have a small number of dimensions, but they do not meet the requirements of deeper data analysis. That is why the problem of using OLAP-technologies, which involve the use of multidimensional cubes, in particular in the business sphere, is becoming urgent.

The object of research is the analysis of OLAP technology and its application for multidimensional data analysis.

The subject of the study is software for interactive data analysis and statistics in web-based services.

The goal of the work: analysis of existing OLAP technologies; comparison of existing OLAP technologies; analysis of methods of creation and operations on OLAP cubes; implementation of means of providing interactive data analysis based on multidimensional OLAP cubes.

The scientific novelty is as follows: faster and better interactive data analysis in web-oriented services using OLAP-systems. Also, the use of OLAP technologies provides a consistently high level of performance and scalability, serving large amounts of data.

Practical value of the results obtained in this work is that the use of OLAP-systems provides a consistently high level of performance and scalability, serving

large amounts of data that can be accessed by thousands of users. With the help of OLAP-technologies, information is accessed in real time, query processing does not slow down the analysis process, ensuring its efficiency and effectiveness. OLAP-system gives users the opportunity to perform complex data analysis, which allows them to better understand the principles of the company and find ways to improve its performance.

Analytical programs based on OLAP-technologies allow to solve a range of analytical tasks, such as production analysis, financial analysis, marketing research, e-business analysis, human resources analysis. OLAP allows you to create analytical applications that cover the analysis of the entire production and financial cycle of the enterprise.

Approbation of work. The main provisions and results of the work were presented and discussed at the XIV scientific conference of undergraduates and graduate students "Applied Mathematics and Computing" PMK-2021 (Kyiv, November 17-19, 2021).

Structure and scope of work. The master's dissertation consists of an introduction, four chapters and conclusions.

The introduction presents a general description of the work, assesses the current state of the problem, substantiates the relevance of research, formulates the purpose and objectives of research, shows the scientific novelty of the results and the practical value of the work, provides information on testing the results and their implementation.

The first section provides general information about OLAP systems and outlines the basic requirements for OLAP systems.

The second section provides an analysis of software development technologies using Data Mining and OLAP technologies.

The third and the fourth sections provide the structure and description of the software, as well as testing and analysis of the results of the study.

The conclusions present the results of the work.

The work is presented on 85 sheets, contains references to the list of used literature sources.

Keywords: OLAP technologies, Data Mining, OLAP cubes, hypercube, business analytics.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

БД – база даних.

Гіперкуб – багатовимірний куб; є багатовимірною структурою даних, з якої користувач-аналітик може запитувати інформацію. Куби створюються з фактів та вимірів.

ІАС – Інформаційно-аналітична система. Система такого типу надає можливість отримувати інформацію, створювати її, а також виконувати її аналіз та обробку.

ІТ – інформаційні технології.

МП – мова програмування.

ООП – об'єкто-орієнтоване програмування. Це парадигма програмування, в якій множина взаємодіючих між собою об’єктів розглядається як програма.

ОС – операційна система.

ПЗ – програмне забезпечення.

ПЗ – програмне забезпечення.

Підкуб (англ. «sub cube») – це частина певного простору куба у вигляді деякої багатовимірної фігури всередині куба.

ПК – персональний комп’ютер.

СД – сховище даних. Являється предметно-орієнтованою базою даних, здебільшого розроблюється спеціально для формування звітів та бізнес-аналізу з ціллю підтримки прийняття рішень в бізнес-організаціях.

СППР – Система підтримки прийняття рішень (з англ. «DSS, Decision Support System»). Це автоматизована комп'ютерна система, що відповідає за прийняття рішень в складних умовах для об'єктивного та повного аналізу певної предметної діяльності.

СУБД – Система управління базою даних. Це сукупність мовних та програмних засобів для використання БД.

API (з англ. «application programming interface») – це сукупність функцій для правильної взаємодії між ПЗ, яке надається API та іншими програмним компонентам.

BI (від англ. «Business Intelligence») - бізнес аналітика ж або інтелектуальний аналіз даних.

Big Data– це термін, який визначає величезний масив різної структурованої та неструктурованої інформації, а також методи її обробки.

FASMI (з англ. «Fast Analysis of Shared Multidimensional Information») – це спеціальний тест-перевірка OLAP-систем.

Git – це програмне забезпечення, яке використовується для контролю версій змін файлів.

HOLAP (з англ. «Hybrid OLAP») – гібридна OLAP. Тип OLAP в якому, вихідні дані залишаються в реляційній базі, а агрегати розміщуються у багатовимірній таблиці.

IDE (з англ. «Integrated Development Environment») або ж ІСВ (Інтегроване середовище розробки) – це комплексне програмне рішення для розробки та тестування ПЗ.

IoT (з англ. «Internet of Things», Інтернет Речей) – це концепція, в якій пристрої, які утворюють глобальну мережу, обмінюються інформацією між собою і, працює без участі людини.

JIT-компіляція (з англ. «Just-in-time compilation») – це методологія компіляції безпосередньо під час її роботи.

MOLAP (з англ. «Multidimensional OLAP») – багатовимірна OLAP. Тип OLAP, в якому дані спочатку обчислюються та зберігаються в MOLAP.

OLAP (від англ. «online analytical processing» - аналітична обробка в реальному часі) - це технологія, яка дозволяє користувачеві легко і вибірково витягувати і переглядати дані за різними параметрами, з різних точок зору

ROLAP (з англ. «Relational OLAP») – реляційна OLAP. Даний тип OLAP працює безпосередньо з реляційним сховищем, факти та таблиці з вимірами зберігаються у реляційних таблицях; спеціально для збереження агрегатів створюються додаткові реляційні таблиці.

SQL (з англ. «structured query language» - «мова структурованих запитів») — це спеціальна мова програмування, основною задачею якої є створення, модифікації та управління даними в реляційній БД.

XML (з англ. «Extensible Markup Language») – це спеціальний стандарт побудови мов розмітки певних структурованих даних, створений для обміну між різними додатками, наприклад, через Інтернет.

ВСТУП

На сьогоднішній день переоцінити роль комп’ютерних технологій доволі важко, адже вони проникли у всі сфери життя суспільства і допомагають вирішувати завдання найрізноманітнішого змісту. Призначення комп'ютерних технологій полегшує в усіх аспектах такі інформаційні процеси, як процеси пошуку, зберігання, передачі та обробки інформації.

Інформаційно-комунікаційні технології застосовуються скрізь: для розвитку, в початковій, середній і вищій освіті, для самоосвіти, в державних установах і, безумовно, у сфері бізнесу. Внаслідок багатогранності програмного та апаратного забезпечення потенційні можливості комп'ютерних технологій фактично є безмежними. Очевидним твердженням також є те, що в сфері IT ведеться постійний розвиток і вдосконалення в усіх технологічних аспектах. На будь-якому підприємстві сьогодні використовуються сховища даних, бази даних та системи управління базами даних. Ці технології дозволяють зберігати інформацію і забезпечують швидкий доступ до неї. Разом із удосконаленням інформаційно-комунікаційних технологій в цілому, вдосконалюються і бази даних. Їх структура стає складнішою, а об’єми задач ще більшими.

Наприклад, реляційні бази даних дозволяють подавати дані про предметну область за допомогою двовимірних таблиць, між якими встановлюється зв'язок. Вони дозволяють відобразити інформацію у найбільш доступній для розуміння користувача формі. Проте проблеми реляційної бази даних у тому, що її розробка є трудомісткою, а швидкість доступу до даних є низькою. Крім цього, використання двомірних таблиць не передбачає роботу з багатомірними даними. Рішенням може бути застосування окремих таблиць для кожного типу даних (вимірювання), однак аналіз такого набору таблиць є громіздким і потребує великої кількості часу. Реляційні бази даних здатні бездоганно обробляти масиви даних, що мають невелику кількість вимірів, але вони не відповідають вимогам глибшого аналізу даних. Саме тому стає актуальною проблема застосування OLAP-технологій, що передбачають використання багатовимірних кубів, зокрема в бізнес сфері.

1. АНАЛІЗ ІСНУЮЧИХ СИСТЕМ ДЛЯ ІНТЕРАКТИВНОГО АНАЛІЗУ ДАНИХ

1.1 Загальні відомості про OLAP-системи

Багатовимірна система управління базами даних – це тип управління базами даних, який ґрунтується на багатовимірному поданні баз даних. Багатомірні бази даних часто створюються з використанням вхідних даних вже існуючих реляційних баз даних. Така база даних здатна відповідати на запитання вигляду «скільки товарів було продано в певному регіоні в певний місяць» і аналогічні питання, пов'язані з узагальненням та аналізом бізнес-операцій та тенденцій, тоді як реляційна база даних зазвичай отримує відповідь з використанням мови структурованих запитів. Багатовимірна система управління базами даних має здатність швидко обробляти дані в базі даних так, щоб відповідь могла бути згенерована моментально.

OLAP (від англ. «online analytical processing» - аналітична обробка в реальному часі) - це технологія, яка дозволяє користувачеві легко і вибірково витягувати і переглядати дані за різними параметрами, з різних точок зору [1]. Набір технологій OLAP включає такі функції, як побудова динамічних звітів з різних аспектів, аналіз наявних даних, спостереження, оцінювання та прогнозування основних показників бізнес-процесів [2].

З точки зору концепції, багатовимірна база даних використовує ідею інформаційних кубів для того, щоб подання даних було більш зрозуміле користувачеві. Якщо ж у реляційних базах даних інформація представлена у вигляді взаємопов’язаних таблиць, то в базах даних OLAP інформація представлена у вигляді так званих OLAP-кубів.

В OLAP-кубах зберігаються відомості про взаємозв'язки та про ієрархію даних, що значно спрощує процедуру отримання необхідної інформації та створення звітів, оскільки місцезнаходження необхідних даних та їх взаємозв'язок з іншими даними відомо заздалегідь. Куби даних містять інформацію, відображену в попередньо розробленому вигляді. Це означає, що групування, фільтрація, сортування інформації здійснюється до введення даних. Саме тому вивід та вилучення інформації, що запитується користувачем, стає простою та швидкою дією [3].

OLAP є інструментом для аналізу великих обсягів даних. При взаємодії із OLAP-системою, користувач має можливість здійснення гнучкого перегляду інформації, отримання довільних зрізів даних та виконання аналітичних операцій деталізації, згортки, наскрізного розподілу, порівняння у часі. Вся робота з OLAP-системою відбувається у термінах предметної галузі.

Сховище даних (Data Warehouse) - це предметно-орієнтований, інтегрований, незмінний, що підтримує хронологію набір даних, організований з метою підтримки прийняття рішень (за визначенням засновника сховищ даних Б. Інмона). Іншими словами - це база даних, що зберігає дані, агреговані за багатьма вимірами.

OLAP-системи є частиною більш загального поняття BI (від англ. «Business Intelligence» - бізнес аналітика ж або інтелектуальний аналіз даних), яке включає окрім традиційного OLAP-сервісу, засоби організації спільного використання документів, що виникають в процесі роботи користувачів сховища. Технології BI забезпечують електронний обмін звітними документами, розмежування прав користувачів, доступ до аналітичної інформації через інтернет.

У той час, як реляційна база даних може розглядатися як двомірна, багатовимірна база даних розглядає кожен атрибут (наприклад, продукт, географічний продаж регіону та тимчасовий період) як відокремлену величину, окреме «вимірювання». Програмне забезпечення з використанням OLAP-технологій може визначати перетин різних величин (всі продукти, продані у певному регіоні, за визначеними цінами впродовж певного періоду часу) і відображати їх. Атрибути, такі як тимчасові періоди, можуть бути розбиті на субатрибути. Атрибутом, в загальному значені, називають властивість або характеристику, а в даному випадку - змінну властивість або характеристику якого-небудь компонента програми, яка може приймати різні значення.

Багатовимірна база даних забезпечує можливість миттєвої обробки інформації, що власне і дозволяє отримувати відповідь швидко. Інтерфейс багатовимірної бази даних може змінюватися в залежності від того, яке розташування даних.

OLAP може використовуватися для інтелектуального аналізу даних або для виявлення раніше не помічених зв'язків між елементами даних. База даних OLAP не обов'язково повинна бути такою ж великою, як і зберігаються дані, оскільки не всі транзакційні дані, необхідні для аналізу тенденцій. Також дані можуть бути імпортовані з існуючих реляційних баз даних для створення багатомірної бази даних для OLAP за допомогою спеціального програмного інтерфейсу ODBC (з англ. «Open Database Connectivity» – відкритий інтерфейс, доступний до бази даних).

Концепція OLAP базується на принципі багатовимірного представлення даних. За вимірами, або ж для кращого розуміння – осями, в багатовимірній моделі виділяють чинники, які є впливовими на діяльність підприємства (наприклад, продукти, час, відділення підприємства тощо) і отримують гіперкуб, який заповнюється потім показниками діяльності підприємства (план прибутку, ціни продукції, продажі, збитки тощо). Наповнення може бути як реальними даними оперативних систем, так і прогнозованими з урахуванням історичних даних, тобто даних, накопичених за певний період.

Вимірювання гіперкуба можуть мати досить складний характер, наприклад, вони можуть бути ієрархічними або ж між ними можуть бути встановлені певні відношення. Під час аналізу користувач може змінювати точку зору на дані, (так звана операція зміни логічного погляду), тобто тим самим переглядаючи дані в різних розділах і вирішуючи конкретні завдання. Над кубами можуть виконуватись операції різного виду, включаючи прогнозування та умовне планування (аналіз типу «що, якщо»).

Оперативні дані збираються з різних джерел, очищаються, інтегруються та складаються у реляційне сховище. При цьому вони вже доступні для аналізу за допомогою різних засобів побудови звітів. Далі (повністю або частково) готуються для OLAP-аналізу. Вони можуть бути завантажені у спеціальну БД OLAP або залишені у реляційному сховищі. Найважливішим його елементом є метадані, тобто інформація про

структуру, розміщення та трансформацію даних. Завдяки їм забезпечується ефективна взаємодія різних компонентів сховища.

1.2 Загальні вимоги до OLAP-систем

В 1993 році Е. Ф. Кодд, винахідник концепції реляційних СУБД і, за сумісництвом, OLAP – сформулював основні критерії OLAP. Вони полягають у недоліках реляційної моделі і, насамперед, вказують на неможливість «об'єднувати, переглядати та аналізувати дані з погляду множинності вимірювань, тобто найзрозумілішим для корпоративних аналітиків способом». Загальні вимоги до OLAP-систем розширюють функціональний ряд реляційних СУБД і включають багатовимірний аналіз як одну зі своїх характеристик.

1993 року Е. Ф. Кодд опублікував працю під назвою "OLAP для користувачів-аналітиків: якою вона має бути" [4]. У ньому він виклав основні концепції аналітичної обробки та визначив 12 правил, яким мають задовольняти продукти, що надають можливість виконання оперативної аналітичної обробки.

Отже, Е. Ф. Кодд визначив 12 правил, яким має задовольняти програмний продукт класу OLAP:

1. **Концептуальне багатовимірне уявлення (Multi-Dimensional Conceptual View).** Користувач-аналітик бачить світ підприємства багатовимірним за своєю природою. Відповідно і OLAP-модель має бути багатовимірною у своїй основі. Багатовимірна концептуальна схема або уявлення користувача полегшують моделювання і аналіз, втім, як і обчислення.
2. **Прозорість (Transparency)**. Незалежно від того, чи є OLAP-продукт частиною засобів користувача чи ні, цей факт має бути прозорим для користувача. Якщо OLAP надається клієнт-серверними обчисленнями, цей факт також, по можливості, повинен бути непомітним для користувача. OLAP повинен надаватися в контексті істинно відкритої архітектури, дозволяючи користувачеві, де б він не знаходився, зв'язуватися за допомогою аналітичного інструменту із сервером. На додаток прозорість повинна досягатися і при взаємодії аналітичного інструменту з гомогенним та гетерогенним середовищами БД.

3. **Доступність (Accessebility).** Користувач-аналітик OLAP повинен мати можливість виконувати аналіз, що базується на загальній концептуальній схемі, що містить дані всього підприємства в реляційній БД, так само як і дані зі старих успадкованих БД, на загальних методах доступу та на загальній аналітичній моделі. Це означає, що OLAP повинен надавати власну логічну схему для доступу в гетерогенному середовищі БД і виконувати відповідні перетворення для надання даних користувачеві. Більше того, необхідно заздалегідь подбати про те, де і як і які типи фізичної організації даних дійсно будуть використовуватися. OLAP-система повинна виконувати доступ тільки до дійсно потрібних даних, а не застосовувати загальний принцип "кухонної вирви", який тягне за собою непотрібне введення.
4. **Постійна продуктивність розробки звітів (Consistent Reporting Performance)**. Якщо кількість вимірювань або обсяг бази даних збільшуються, користувач-аналітик не повинен відчувати будь-якої істотної деградації у продуктивності. Підтримка постійної продуктивності належного рівня критична для кінцевого користувача задля легкого використання OLAP обмеженої складності. Якщо користувач-аналітик буде відчувати суттєві відмінності у продуктивності відповідно до числа вимірів, тоді він буде прагнути компенсувати ці відмінності стратегією розробки, що, в свою чергу, може призвести до невірного подання даних іншими шляхами, але не тими, якими дійсно потрібно подавати дані. Тобто, OLAP-продукт повинен надавати стабільну роботу видачі необхідних користувачеві звітів, які можуть містити певну кількість вимірів.
5. **Клієнт-серверна архітектура (Client-Server Architecture).** Більшість даних, які сьогодні потрібно піддавати оперативній аналітичній обробці, містяться на мейнфреймах з доступом через ПК. Отже, це означає що OLAP-продукти мають бути здатні працювати в середовищі «клієнт-сервер». З цієї точки зору є необхідним, щоб серверний компонент

аналітичного інструменту був суттєво "інтелектуальним", тобто - щоб різні клієнти могли приєднуватися до сервера з мінімальними труднощами. "Інтелектуальний" сервер повинен бути здатний виконувати відображення та консолідацію між невідповідними логічними та фізичними схемами баз даних. Це забезпечить прозорість та побудову загальної концептуальної, логічної та фізичної схеми.

6. **Загальна багатовимірність; рівноправність вимірів (Generic Dimensionality).** Кожен вимір має застосовуватися безвідносно своєї структури та операційних здібностей. Додаткові операційні здібності можуть надаватися вибраним вимірам, і оскільки виміри симетричні, окремо взята функція може бути надана будь-якому виміру. Базові структури даних, формули та формати звітів не повинні зміщуватися у бік будь-якого виміру.
7. **Динамічне керування розрідженими матрицями (Dynamic Sparse Matrix Handling)** Фізична схема інструменту OLAP повинна повністю адаптуватися до специфічної аналітичної моделі для оптимального управління розрідженими матрицями. Для будь-якої взятої розрідженої матриці існує одна і тільки одна оптимальна фізична схема. Ця схема надає максимальну ефективність пам'яті й операбельність матриці, якщо, звичайно, весь набір даних не міститься в пам'яті. Базові фізичні дані OLAP-інструменту повинні конфігуруватися до будь-якої підмножини вимірювань у будь-якому порядку для практичних операцій з аналітичними моделями великого розміру. Фізичні методи доступу також повинні динамічно змінюватися та містити різні типи механізмів, таких як: безпосередні обчислення, B-дерева (B-tree) та похідні, хешування, можливість комбінувати ці механізми за потреби. Розрідженість (вимірюється у відсотковому відношенні порожніх осередків всім можливим) - це характеристика поширення даних. Неможливість регулювати розрідженість може зробити ефективність операцій недосяжною. Якщо OLAP-інструмент не може контролювати та

регулювати поширення значень аналізованих даних, модель, що претендує на практичність та базується на багатьох шляхах консолідації та вимірювання, насправді може виявитися непотрібною та безнадійною.

8. **Багатокористувацька підтримка (Multi-User Support).** Часто кілька користувачів-аналітиків мають потребу працювати спільно з однією аналітичною моделлю або створювати різні моделі з єдиних даних. Отже, OLAP-інструмент повинен надавати можливості спільного доступу (запиту та доповнення), цілісності та безпеки.
9. **Необмежені перехресні (перехресно-вимірні) операції (Unrestricted Cross-dimensional Operations).** Різні рівні згортки та шляхи консолідації, внаслідок їхньої ієрархічної природи, представляють залежні відносини в OLAP-моделі або додатку. Отже, сам продукт повинен виконувати відповідні обчислення і не вимагати від користувача-аналітика знову визначати ці обчислення та операції. Обчислення, які не виходять з цих успадкованих відносин, вимагають визначення різними формулами відповідно до певної мови, яка застосовується. Така мова може дозволяти обчислення та маніпуляцію з даними будь-яких розмірностей та не обмежувати відносини між комірками даних, не звертати увагу на кількість загальних атрибутів даних конкретних комірок.
10. **Інтуїтивна маніпуляція даними (Intuitive Data Manipulation)**. Зміна орієнтації шляхів консолідації, деталізація та інші маніпуляції, що регламентуються шляхами консолідації, повинні застосовуватися через окрему взаємодію на комірки аналітичної моделі, а також не повинні вимагати використання системи меню або іншої великої кількості дій з інтерфейсом користувача.
11. **Гнучкі можливості отримання звітів (Flexible Reporting)**. Аналіз та подання даних є простими операціями, коли рядки, стовпці та комірки даних, які візуально порівнюватимуться між собою, будуть знаходитися поблизу один до одного або за деякою логічною функцією. Засоби формування звітів повинні представляти дані, що синтезуються, або

інформацію, що випливає з моделі даних у її будь-якій можливій орієнтації. Це означає, що рядки, стовпці або сторінки повинні показувати одночасно від 0 до N вимірів, де N - число вимірів усієї аналітичної моделі. На додаток кожен вимір вмісту, який показано в одному записі, колонці або сторінці, повинен також бути здатним показати будь-яку підмножину елементів, тобто значень, що містяться у вимірі, в будь-якому порядку.

12. **Необмежена розмірність та кількість рівнів агрегації (Unlimited Dimensions and Aggregation Levels)**. Дослідження про можливу кількість необхідних вимірів, які потрібні в аналітичній моделі, показало, що одночасно може використовуватися до 19 вимірів. Звідси випливає рекомендація, щоб аналітичний інструмент був здатний надати хоча б 15 вимірів одночасно, а переважно 20. Більше того, кожен із загальних вимірів не повинен бути обмежений за кількістю рівнів агрегації, що визначаються користувачем-аналітиком і шляхів консолідації.

Дещо пізніше вони були перероблені на так званий тест FASMI, який визначає вимоги до продуктів OLAP.

FASMI – це абревіатура від назви кожного пункту тесту:

- **Fast** (з англ. «швидкий») - властивість означає, що система повинна забезпечувати відповідь на запит користувача за п'ять секунд; при цьому більшість запитів обробляються в межах однієї секунди, а найскладніші запити повинні бути оброблені в межах двадцяти секунд. Останні дослідження показали, що користувач починає сумніватися в успішності запиту, якщо він займає більше тридцяти секунд.
- **Analysis** (з англ. «аналітичний») - система повинна справлятися з будь-яким логічним та статистичним аналізом, характерним для бізнес-додатків, та забезпечує збереження результатів у вигляді, який є доступним для кінцевого користувача. Засоби аналізу можуть включати процедури аналізу розподілу витрат, часових рядів, конверсії валют, моделювання змін організаційних структур та інших.

- **Shared** (з англ. «розподілений») - система має надавати широкі можливості розмежування доступу до даних та одночасної роботи багатьох користувачів.
- **Multidimensional** (з англ. «багатовимірний») - система повинна забезпечувати концептуальне багатовимірне подання даних, включаючи повну підтримку множинних ієрархій.
- **Information** (з англ. «інформація»). Потужність різних програмних продуктів характеризується кількістю вхідних даних, що обробляються. OLAP-системи різного виду мають різну потужність: OLAP-рішення, які є передовими, можуть оперувати, принаймні, в тисячу разів більшою кількістю даних у порівнянні з найменш потужними. При виборі OLAP-інструменту слід враховувати низку факторів, включаючи необхідну оперативну пам'ять, дублювання даних, експлуатаційні показники, використання дискового простору, інтеграцію з інформаційними сховищами тощо.

1.3 Загальна інформація про гіперкуби

Головний постулат OLAP – багатовимірність у поданні даних. У термінології OLAP для опису багатовимірного дискретного простору даних використовують поняття куба, або гіперкуба.

Куб є багатовимірною структурою даних, з якої користувач-аналітик може запитувати інформацію. Куби створюються з фактів та вимірів.

Факти – це дані про об'єкти та певні події, які підлягають аналізу. Факти одного типу утворюють заходи (measures). Міра є значенням в осередку куба, іншими словами є коміркою.

Виміри – це елементи даних, над якими проводиться аналіз фактів. Сукупність таких елементів формує атрибут виміру [5].

Куб може містити фактичні дані з однієї чи кількох таблиць фактів і найчастіше містить кілька вимірів. Будь-який конкретний куб має конкретний спрямований предмет аналізу.

Як вже було зазначено, суть OLAP полягає в тому, що вихідна для аналізу інформація подається у вигляді гіперкубу (багатовимірного куба), і забезпечується можливість довільно маніпулювати нею та отримувати необхідні інформаційні розрізи – звіти. При цьому кінцевий користувач бачить куб (рисунок 1.1) як динамічну багатовимірну таблицю, яка автоматично підсумовує дані, тобто факти, у різних розрізах (вимірюваннях), і дозволяє інтерактивно керувати обчисленнями та формою звіту.

Важливою концепцією багатовимірної моделі даних є підпростір або підкуб (sub cube). Підкуб є частиною повного простору куба у вигляді деякої багатовимірної фігури всередині куба. Так як багатовимірний простір куба дискретний і обмежений, підкуб також дискретний і обмежений.

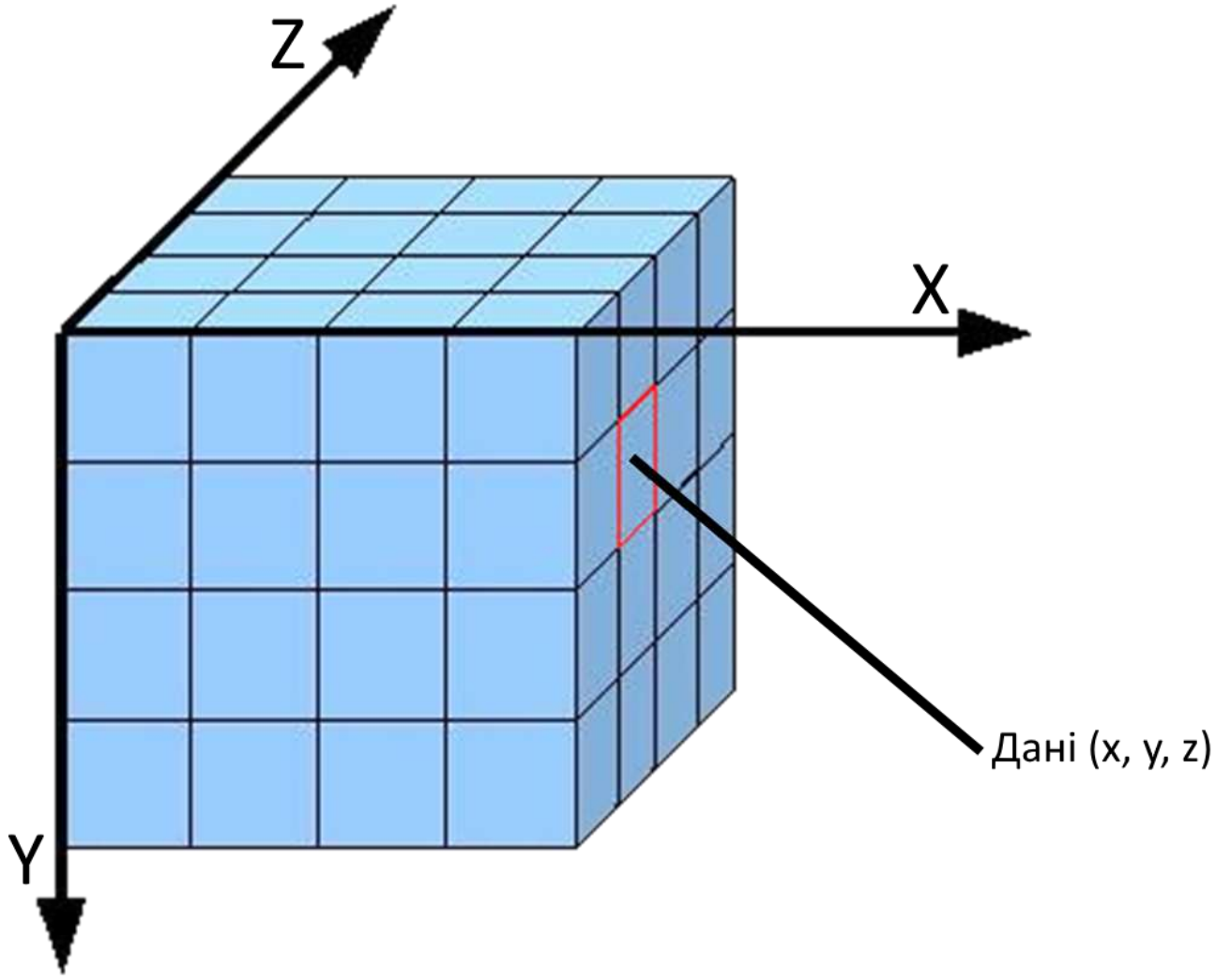

Рисунок 1.1 – Схематичне зображення гіперкубу

Над гіперкубом можуть виконуватися наступні операції:

- Зріз – це операція, під час якої формується підмножина багатовимірного масиву даних, що відповідає єдиному значенню одного або декількох елементів вимірювань, що не входять до цієї підмножини (рисунок 1.2).

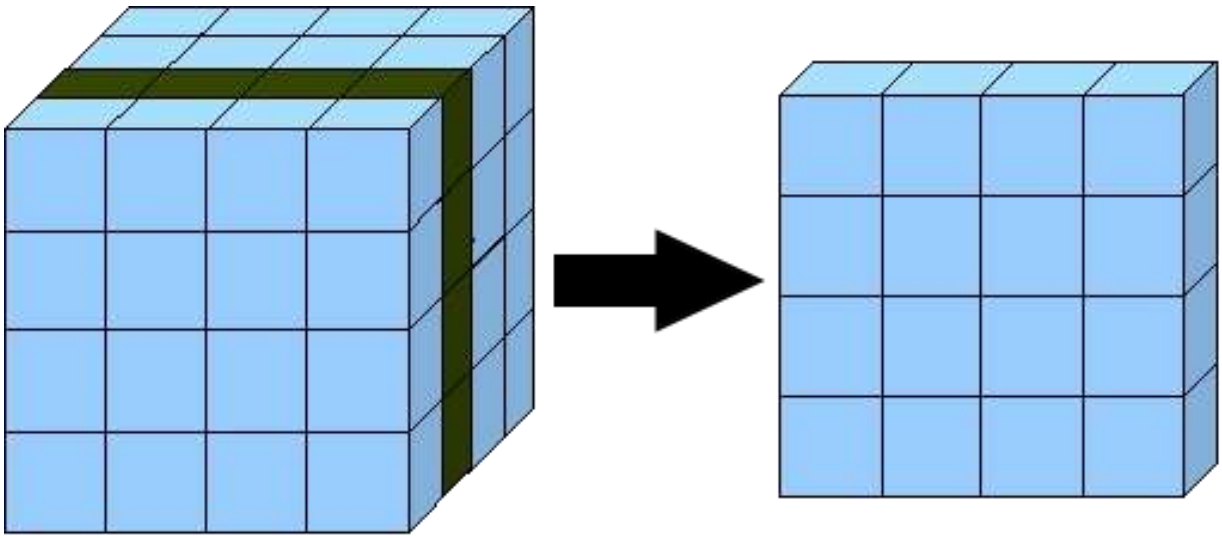

Рисунок 1.2 – Схематичне зображення операції «зріз» над гіперкубом

- Обертання – це операція під час, якої виконується зміна розташування вимірювань, представлених у звіті або на відображуваній сторінці. Транспонування (обертання) зазвичай застосовується до плоских таблиць, отриманих, наприклад, в результаті зрізу, і надає можливість зміни порядку подання вимірювань таким чином, що вимірювання, які відображалися в стовпцях, відображатимуться в рядках і навпаки. У певних випадках транспонування дозволяє зробити вигляд таблиці більш наочним. Наприклад, операція обертання може полягати у перестановці місцями рядків та стовпців таблиці. Крім того, обертанням куба даних є переміщення вимірів, як знаходяться поза таблицею, на місце вимірювань, представлених на сторінці, що відображається, і навпаки (рисунок 1.3).

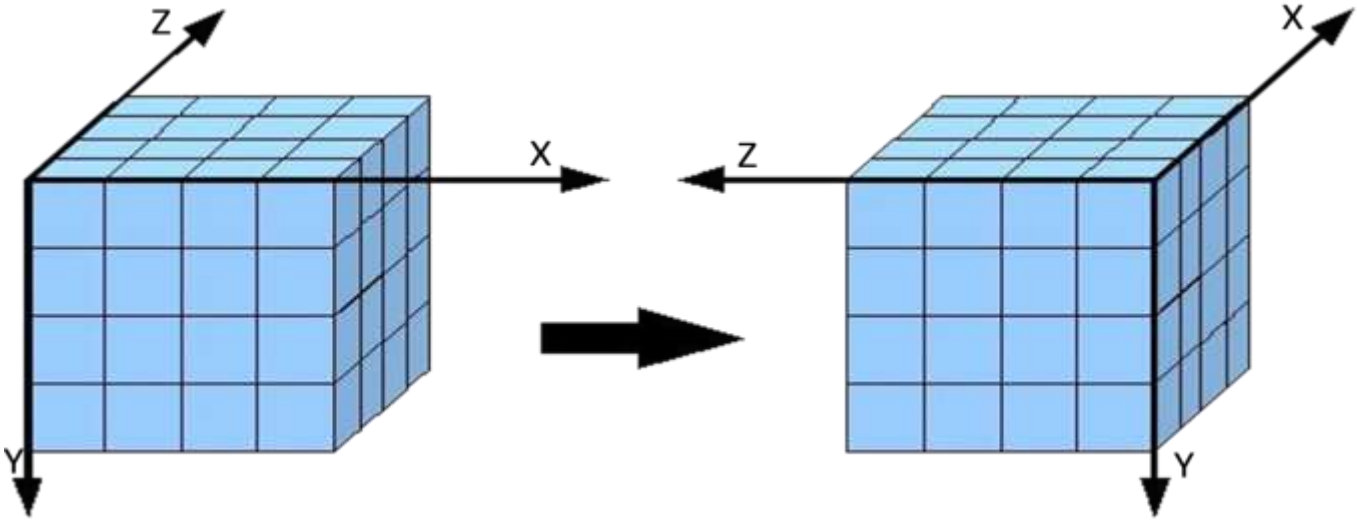

Рисунок 1.3 – Схематичне зображення операції «обертання» над гіперкубом

- Консолідація - це операція згортки, тобто перехід у напрямку від детального подання даних до агрегованого представлення даних (рисунок 1.4).

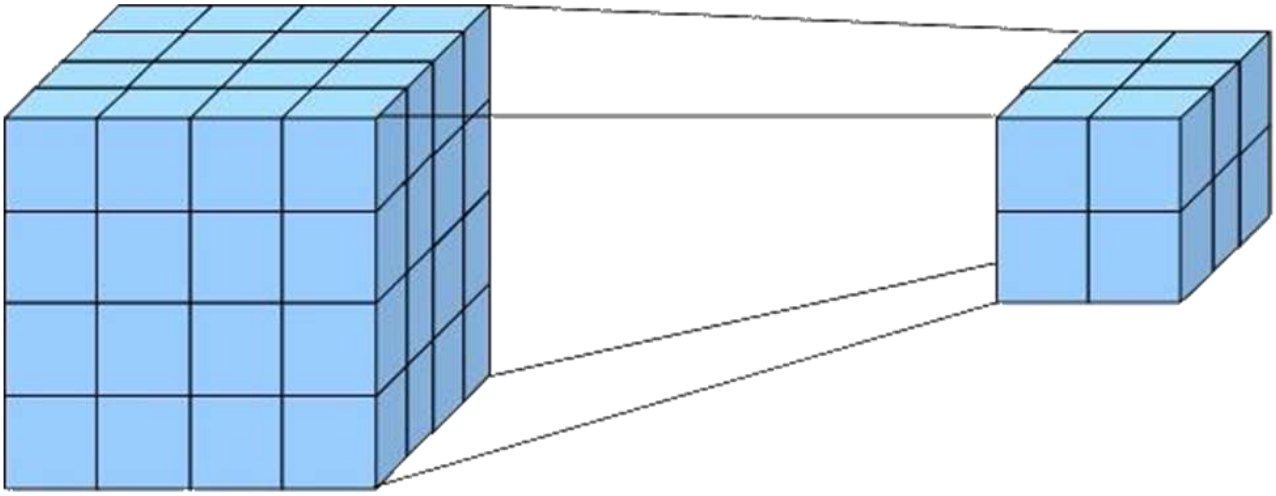

Рисунок 1.4 – Схематичне зображення операції «консолідації» над гіперкубом

- Деталізація – операція узагальнення, яка є оберненою до операції консолідації (згортки), тобто перехід від агрегованого подання даних до більш загального. Напрямок деталізації (узагальнення) може бути заданий як за ієрархією окремих вимірів, так і згідно з іншими відносинами, встановленими в рамках вимірів або поміж вимірами (рисунок 1.5).

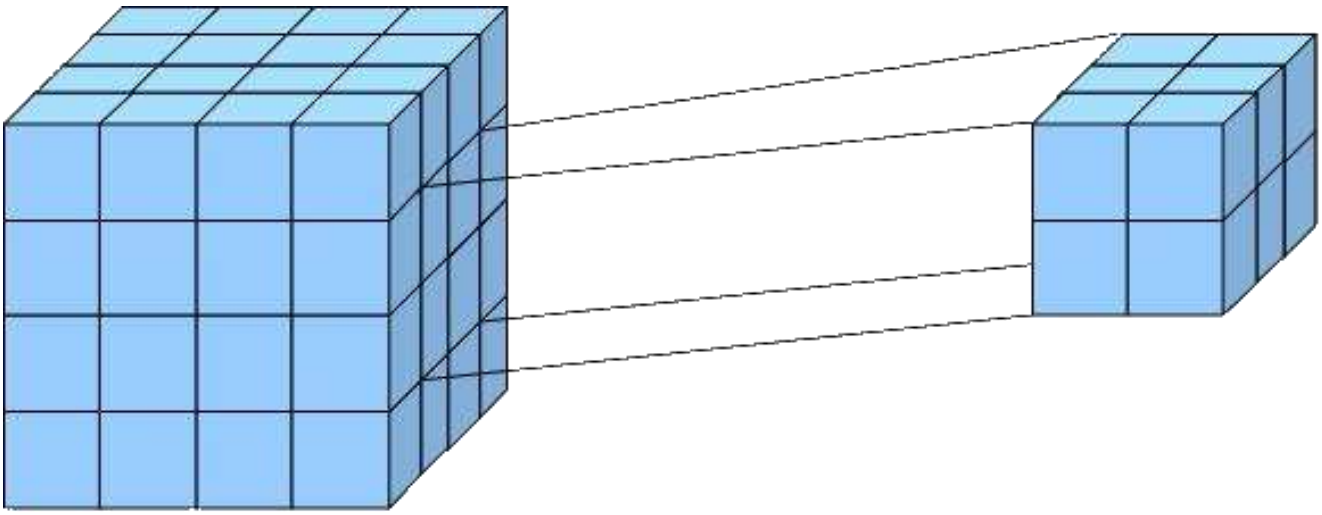

Рисунок 1.5 – Схематичне зображення операції «деталізації» над гіперкубом

1.4 Постановка задачі магістерської дисертації

На сьогоднішній день в сфері бізнесу існує велика кількість технологій, які дозволяють ефективно обробляти дані у великих масштабах, тобто, отриманих з різноманітних джерел даних (реляційні та не реляційні дані) - величезні обсяги даних, все частіше отримуються та обробляються в реальному часі. Серед додатків, які обробляють Big Data, найбільшим попитом серед підприємств є додатки, пов'язані з бізнес-аналітикою або ж Business Intelligence (BI). За допомогою додатків такого типу можна отримувати корисні ідеї з даних великого обсягу, які можна використовувати для покращення процесів прийняття рішень. Завдяки цьому програмне забезпечення в області BI часто є досить вигідним для різноманітних організацій.

В першу чергу користувачеві від створеного продукту необхідна швидка обробка інформації, що власне і дозволяє отримувати відповідь швидко. Використання OLAP-систем забезпечує стабільно високий рівень продуктивності та масштабованості, обслуговуючи великі обсяги даних, доступ до яких можуть отримати тисячі користувачів. Завдяки використанню OLAP-технологій доступ до інформації здійснюється у реальному часі, обробка запитів не уповільнює процес аналізу, забезпечуючи його оперативність та ефективність. OLAP-програма дозволяє користувачам проводити складний аналіз даних, який надає можливість краще розуміти принципи функціонування компанії та знайти способи покращення результатів у її бізнес діяльності.

Застосування реляційних баз даних у підтримці якісних бізнес процесів не відповідає потребам та запитам керівників, менеджерів та співробітників організацій. З розвитком економіки як науки кількість бізнес-показників збільшується з розширенням підприємства. Саме тому застосування багатовимірних баз даних є панацеєю у сфері бізнесу. OLAP-технології, що використовують метод зберігання та обробки даних у вигляді кубів, дозволяють автоматизувати стратегічний рівень управління

підприємством. Складання звітів, аналіз та прогнозування подій стає не таким ресурсоємким процесом при застосуванні технологій OLAP.

Головною задачею магістерської дисертації є реалізація засобів забезпечення інтерактивного аналізу даних у веборієнтованих сервісах на основі багатовимірних OLAP-кубів. Створене програмне забезпечення матиме всі переваги OLAP-технології, адже воно базується безпосередньо на використанні OLAP.

Висновки до першого розділу МД

У першому розділі магістерської дисертації наведено загальні відомості про OLAP-системи, наведено основні вимоги до систем з використанням OLAP. Також наведено загальний вигляд багатовимірного кубу, як основної структури даних, яка використана у даній роботі для інтерактивного аналізу даних у веборієнтованих сервісах.

2. АНАЛІЗ ТЕХНОЛОГІЙ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ OLAP-КУБІВ

2.1 Класифікація аналітичних систем

Агентство «Gartner Group», яке займається аналізом ринків інформаційних технологій, 80-х роках минулого сторіччя запровадило термін "Business Intelligence" (BI) - діловий інтелект або бізнес-інтелект. Цей термін запропонований для опису різних концепцій та методів, які покращують бізнес-рішення шляхом використання систем підтримки прийняття рішень.

Пізніше у 1996 році агентство уточнило визначення цього терміну. Business Intelligence – це програмні засоби, які функціонують у рамках підприємства та забезпечують функції доступу та аналізу інформації, яка знаходиться в сховищах даних, а також надають для користувачів прийняття правильних та обґрунтованих управлінських рішення.

На основі цих засобів створюються BI-системи, мета яких – підвищити якість інформації для ухвалення BI-системи; також відомі під назвою системи підтримки прийняття рішень (СППР, DSS, Decision Support System). Ці системи перетворюють дані на інформацію, на основі якої можна приймати рішення, тобто вони допомагають в прийнятті управлінських рішень.

Поняття BI поєднує в собі різні засоби та технології аналізу та обробки даних масштабу підприємства.

Загальний склад ринку BI продуктів, визначений агентством «Gartner Group», а також короткий опис параметрів наведено в таблиці 2.1.

Таблиця 2.1 – Аналітичні системи та їх класифікація

Клас програмного продукту	Тип класифікації	Вид, спеціалізація
Засоби побудови сховищ даних (data warehousing)	Засоби проектування сховищ даних	У складі СУБД
		Універсальні засоби
		Студії (готові рішення)
	Засоби вилучення, перетворення та завантаження даних	У складі СУБД
		Універсальні засоби
	Готові, предметно-орієнтовані СД	
Системи аналітичної обробки в реальному часі (OLAP)	Спосіб зберігання даних	MOLAP
		ROLAP
		HOLAP
	Розташування OLAP-продукту	OLAP-сервери
		OLAP-клієнти
	Ступінь готовності до застосування	OLAP-компоненти
		Інструментальні OLAP-системи
		OLAP-застосування
Інформаційно-аналітичні системи (Enterprise Information Systems, EIS)	Типовий список задач, які вирішуються	Аналіз фінансового стану Інвестиційний аналіз Підготовка бізнес-планів

Продовження таблиці 2.1

Клас програмного продукту	Тип класифікації	Вид, спеціалізація
Інформаційно-аналітичні системи (Enterprise Information Systems, EIS)	Типовий список задач, які вирішуються	Маркетинговий аналіз Управління проєктами Бюджетування Фінансове управління
	Масштаб задачі, що вирішується	Автоматизація праці окремого працівника
		Для колективної роботи групи працівників
		Для територіально-розподіленої корпорації
	Технологічна побудова	Монолітна
		Налагоджувальна
Засоби інтелектуального аналізу даних (data mining)		Дерево рішень Генетичні алгоритми Асоціативні правила Нейронні мережі
	Спосіб представлення	В складі OLAP-систем
		Самостійні рішення
Інструменти для виконання запитів та побудови звітів (query and reporting tools)		В складі OLAP-систем
		Самостійні рішення

Як можна побачити з вищенаведеної таблиці 2.1, дана класифікація базується на методі функціональних завдань, де програмні продукти кожного класу виконують певний набір функцій або операцій з використанням спеціальних технологій.

2.2 Data Mining як частина ринку інформаційних технологій

У минулому процес видобутку золота у гірничій промисловості складався з вибору ділянки землі та подальшого просіювання ґрунту велику кількість разів. Іноді шукач знаходив кілька цінних самородків або міг натрапити на золотоносну жилу, але в більшості випадків він взагалі нічого не знаходив, і йшов далі до іншого багатообіцяючого місця, або ж зовсім кидав добувати золото, вважаючи це заняття марною тратою часу.

Зараз же з'явилися нові наукові методи та спеціалізовані інструменти, що зробили гірничу промисловість набагато точнішою та продуктивнішою. Data Mining в галузі даних розвинулася майже в такий же спосіб. Старі методи, що застосовувалися математиками та статистиками, забирали багато часу, щоб у результаті отримати конструктивну та корисну інформацію.

Зараз на ринку представлено безліч інструментів, що включають різні методи, які роблять Data Mining прибутковою справою, дедалі все більш доступною для більшості компаній.

Історично термін Data Mining одержав свою назву з двох понять: пошуку важливої інформації у даних, власне з англ. «data», та видобутку гірничої руди (з англ. «mining»). Обидві процедури вимагають або «просіювання», тобто обробки дуже великої кількості необробленого матеріалу, або інтелектуального дослідження та пошуку потрібних властивостей.

Концептуально поняття Data Mining з’явилося ще в 1978 році, але набуло свого сучасного термінологічного тлумачення приблизно в першій половині 90-х років. Раніше до цього аналіз та обробка даних відбувалися в межах прикладної статистики, при цьому самі задачі вирішувалися лише методом опрацювання баз даних невеликого розміру.

Технології сховищ даних, бізнес-аналітики та аналітики почали з’являтися наприкінці 1980-х та на початку 1990-х років, забезпечуючи розширені можливості аналізу зростаючих обсягів даних, які створювали та збирали організації. Термін Data Mining використовувався до 1995 року, коли в Монреалі відбулася Перша міжнародна конференція з вивчення знань і Data Mining.

Спонсором заходу виступила Асоціація сприяння розвитку штучного інтелекту (AARI), яка також проводила конференцію щорічно протягом наступних трьох років. Починаючи з 1999 року, конференцію – широко відому як KDD 2021 і так далі – організовує SIGKDD, група спеціальних інтересів з виявлення знань та Data Mining в рамках Асоціації обчислювальної техніки.

Технічний журнал Data Mining and Knowledge Discovery опублікував свій перший випуск у 1997 році. Спочатку щоквартально, тепер він публікується раз на два місяці і містить рецензовані статті про теорії, методи та практики пошуку даних та відкриття знань. Інше видання, American Journal of Data Mining and Knowledge Discovery, було запущено в 2016 році.

Data Mining фактично є предметом мультидисциплінарним, бо виник він та розвивається на основі таких наук, як прикладна теорія баз даних, статистика, машинне навчання, штучний інтелект, алгоритмізація та інших (рисунок 2.1)

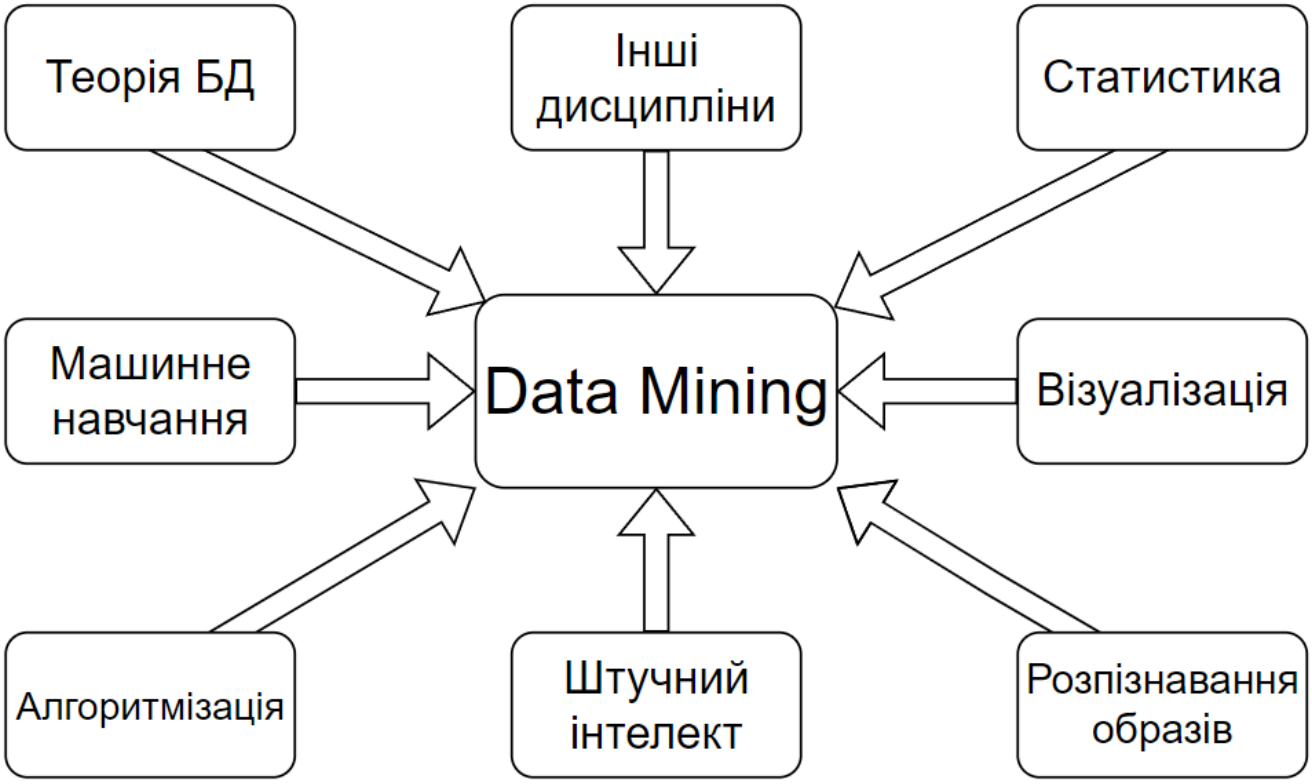

Рисунок 2.1 – Data Mining як мультидисциплінарна сутність

Нижче наведено короткий опис основних дисциплін, на перетину яких з'явилася технологія Data Mining.

2.2.1 Розвиток баз даних

Історично технологічний розвиток баз даних сягає корінням у 60-і роки XX століття. В ті часи дані зберігалися у файлах. Тобто кожен файл містив певні відомості й, відповідно, для охоплення всієї предметної області необхідним критерієм була наявність декількох файлів. Наприклад, відомості про товари зберігалися в одному файлі, а інформація щодо клієнтів – в іншому та, в свою чергу, дані про покупки товарів певними клієнтами – у третьому.

Очевидно, що організація даних такого виду вносила певні складнощі:

- подання даних у кожному файлі було різним;
- необхідно було узгоджувати дані в різних файлах для забезпечення несуперечності інформації;

- необхідно було обирати, які дані та в якому вигляді будуть фігурувати в таких файлах, як файл придбань товарів у прикладі;
- складність розробки програмного забезпечення та їх оновлення при зміні даних.

Ситуація була доволі важкою з точки зору розробки та вимагала поліпшення. Саме тому велика кількість фахівців старанно працювала над створенням чогось зручнішого у використанні. Приблизно через 10 років, на початку 1970-х років, ситуація почала покращуватися і з'явилися перші бази даних.

Британський вчений Е. Ф. Кодд у 1970 опублікував статтю, яка послужила основою для створення реляційної моделі даних. Головна перевага такої моделі зберігання даних полягає в мінімальному дублюванні даних та виключенні деяких типів помилок, властивих іншим моделям.

Згідно з цією моделлю, дані зберігаються у вигляді таблиць зі стовпцями та рядками. Не всі види таблиць прийнятні для реляційної моделі; небажані таблиці можуть бути нормалізовані для задоволення вимог реляційної моделі. У процесі нормалізації таблиця, як правило, розбивається на дві або більше прийнятних таблиць.

У 1979 році для мікрокомп'ютерів dBase-II невелика компанія Ashton-Tate випустила продукт, назвавши його реляційною СУБД. Компанії завдяки успішній тактиці вдалося поширити більш ніж 100 тисяч копій продукту серед користувачів комп'ютерів Osborne. Багато користувачів створювали програми для них, і незабаром dBase стала дуже популярною СУБД. Пізніше компанія Ashton-Tate була куплена фірмою Borland.

Продукт dBase насправді не був реляційною СУБД, а являв собою мову програмування з розширеними функціями для обробки файлів. Поки йшов розвиток dBase, інші виробники розпочали процес перенесення на мікрокомп'ютери своїх комерційних СУБД великих електронних обчислювальних машин. Прикладами таких СУБД є Oracle, Ingress та Focus. Перенесення СУБД на мікрокомп'ютери спричинило поліпшення інтерфейсу користувача, що призвело до збільшення кількості мікрокомп'ютерів, які працюють із БД.

У середині 1980-х років користувачі почали об'єднувати свої комп'ютери в локальні мережі, що призвело до виникнення клієнт-серверної моделі, а також моделі із спільним використанням файлів. Мережа дозволяла спільно використовувати дискові накопичувачі великої ємності, а також дорогі принтери. Користувачі хотіли спільного використання їх БД, що стимулювало розвиток розрахованих на багато користувачів додатків баз даних для локальних мереж.

Оскільки розрахована на багато користувачів обробка даних в локальній мережі відрізняється від розрахованої на багато користувачів обробки даних на мейнфреймі наявністю декількох обчислювачів, виникали додаткові складнощі по координації дій обчислювачів. Так виникла клієнт-серверна архітектура обробки даних. Існує і простіша, але менш надійна архітектура, заснована на спільному використанні файлів.

Зараз активно розвиваються вебдодатки баз даних, а також бази даних із використанням Internet-технологій. Вебдодатки баз даних роблять дані доступними для клієнтського перегляду користувачем, тоді як БД з використанням Internet-технологій

просто використовують клієнтські оглядачі та технології типу DHTML та XML для роботи з базою даних, не публікуючи дані через Internet.

Також існує ще дві технології БД, які є можливими, але на даний момент ще не є реалізованими – це об'єктно-орієнтовані бази даних та розподілені бази даних.

Розподілені бази даних є базою даних організації, розподілені кількома комп'ютерами локальної мережі організації. Завдяки такій архітектурі можливий гнучкіший поділ навантаження по відділах підприємства, але реалізація такої системи пов'язана з рядом проблем, деякі з яких не вирішені досі.

Об'єктно-орієнтовані бази даних позиціонуються як засіб зберігання структур даних, які використовуються в об'єктно-орієнтованому програмуванні. Так як об'єкти є на порядок складнішими за структури, то й реалізація БД буде досить складною. На додачу, розвиток об'єктно-орієнтованих БД стримується існуванням величезної кількості реляційних баз даних, у яких вже зберігаються масиви інформації величезних розмірів.

Взагалі існують різні визначення поняття бази даних. Зазвичай вони або неповні, або надто громіздкі. Нижче наведено просте визначення, яке розширюється з появою нових понять.

Базою даних (БД) називають сукупність взаємозалежних даних на машинних носіях, призначених для використання в інтерактивному (діалоговому) режимі доступу та програмних додатках. Зазвичай БД створюється для зберігання та доступу до даних з деякої предметної області, тобто є інформаційною моделлю класу об'єктів.

Система управління базою даних (СУБД) - це мовні та програмні засоби для організації, поповнення, модифікації та використання БД. Розрізняють універсальні та спеціалізовані СУБД. Універсальні СУБД є системами широкого профілю і не мають чітко розмеженого застосування, а спеціалізовані СУБД створюються для БД конкретного призначення: банківських, бухгалтерських тощо. Спеціалізовані СУБД найбільшою мірою враховують специфіку предметної області, що відображається в інтерфейсі та процедурах обробки інформації.

Також виділяють клас промислових чи комерційних СУБД як систем, розроблених професійними компаніями у сфері створення програмного забезпечення, апробованих практично і тиражованих на певних комерційних умовах. Промислові СУБД відносно дешеві, досить надійні та доволі часто мають гарну документацію. Зазвичай їхній появі передують досвідчені розробки, пробні версії, попередні публікації тощо.

2.2.2 Поняття статистики

Статистика - це наука про методи збору, обробку та аналіз даних для подальшого виявлення закономірностей, що властиві досліджуваному явищу. Статистика - це сукупність методів планування експерименту, збору даних, їх подання та узагальнення, а також аналізу та отримання висновків на основі цих даних. Статистика оперує даними, отриманими у результаті спостережень або експериментів.

2.2.3 Поняття машинного навчання та штучного інтелекту

На сьогоднішній день немає єдиного визначення машинного навчання. Його можна охарактеризувати як процес отримання нових програм знань. У 1996 році Мітчелл дав наступне визначення: "Машинне навчання - це наука, яка вивчає комп'ютерні алгоритми, які автоматично покращуються під час роботи". Нейронні мережі – це один із найбільш популярних прикладів алгоритму машинного навчання.

В свою чергу, штучний інтелект – це науковий напрям, який передбачає постановку і вирішення задач апаратного чи програмного моделювання видів людської діяльності, які вважаються інтелектуальними.

Термін інтелект (з англ. «intelligence», з лат. «intellectus») означає свідомість, розум, розумові здібності людини, а штучний інтелект (AI, Artificial Intelligence), відповідно, тлумачиться як властивість автоматичних комп’ютерних систем брати окремі функції інтелекту людини. Таким чином, штучним інтелектом можна назвати властивість інтелектуальних систем виконувати творчі функції, які вважаються прерогативою людини.

Усі напрямки, які сформували Data Mining, мають свої унікальні особливості. Порівняємо деякі з них з Data Mining.

2.2.4 Коротке порівняння статистики, машинного навчання та Data Mining

- Статистика
  - ❖ Більше зосереджується на перевірці гіпотез.
  - ❖ Базується на теорії більше, ніж Data Mining.
- Машинне навчання
  - ❖ Є більш евристичним, тобто в більшій мірі використовує процеси відкриття, створення чогось нового.

- ❖ Є сконцентрованим на поліпшенні роботи агентів навчання.

- Data Mining
  - ❖ Являється прямим поєднанням теорії та евристики.
  - ❖ Концентрується на єдиному процесі аналізу даних, який включає очищення даних, навчання, інтеграцію та відображення, тобто візуалізацію результатів.

2.3 Data mining. Ключові особливості, поняття та концепції.

Data mining не є чимось таким, що з'явилось у цифрову епоху. Концепція існує вже понад століття, але привернула увагу громадськості у 1930-х роках. Один із перших прикладів Data mining був продемонстрований у 1936 році, коли Алан Т'юрінг представив ідею універсальної машини, яка могла б виконувати обчислення, подібні до обчислень сучасних комп'ютерів.

Відтоді людство пройшло довгий шлях. Зараз компанії використовують Data mining та машинне навчання, щоб покращити все: від процесів продажів до інтерпретації фінансових даних для інвестицій. У результаті цього спеціалісти Data science стали життєво важливими для організацій по всьому світу, оскільки компанії прагнуть досягти більших цілей за допомогою науки про дані, ніж будь-коли раніше.

Data mining – це процес аналізу великих обсягів даних для виявлення бізнес-аналітики, яка допомагає компаніям вирішувати проблеми, зменшувати ризики та використовувати нові можливості. Цей напрямок науки про дані отримав свою назву через схожість між пошуком цінної інформації у великій базі даних і видобутком корисних копалин (mining). Обидва процеси вимагають просіювання величезної кількості матеріалу, щоб знайти приховану цінність.

Data mining може відповісти на питання бізнесу, які традиційно займали занадто багато часу для вирішення вручну. Використовуючи різноманітні статистичні методи для аналізу даних різними способами, користувачі можуть визначити закономірності, тенденції та зв'язки, які вони інакше могли б упустити. Вони можуть застосувати ці висновки, щоб передбачити події, які, ймовірно, стануться в майбутньому, та вжити відповідних заходів.

Data mining використовується в багатьох сферах бізнесу та досліджень, включаючи продажі та маркетинг, розробку продуктів, охорону здоров'я та освіту. При правильному використанні Data mining може забезпечити значну перевагу перед конкурентами, дозволяючи дізнатися більше про клієнтів, розробити ефективні маркетингові стратегії, збільшити дохід і зменшити витрати.

2.3.1 Ключові концепції Data Mining

Щоб досягти найкращих результатів від аналізу даних, потрібен цілий ряд інструментів і методів. Найбільш часто використовувані функції включають наступні наведені поняття.

Очищення та підготовка даних — етап, на якому дані перетворюються у форму, придатну для подальшого аналізу та обробки, наприклад виявлення та видалення помилок.

Штучний інтелект (ШІ) —відповідає за аналітичну діяльність, схожу на з людський інтелект, таку як планування, навчання, міркування та вирішення проблем.

Навчання правил асоціації — ці інструменти, також відомі як аналіз ринкового кошика, шукають зв'язки між змінними в наборі даних, наприклад визначають, які продукти зазвичай купуються разом.

Кластеризація — процес поділу набору даних на набір значущих підкласів, які називаються кластерами, щоб допомогти користувачам зрозуміти природне групування або структуру даних.

Класифікація — ця методика призначає елементи в наборі даних цільовим категоріям або класам з метою точного прогнозування цільового класу для кожного випадку в даних.

Аналітика даних — процес обчислення цифрової інформації в корисну бізнес-аналітику.

Сховища даних — велика колекція бізнес-даних, яка використовується, щоб допомогти організації приймати рішення. Це основоположний компонент більшості масштабних проектів з аналізу даних.

Машинне навчання — техніка комп'ютерного програмування, яка використовує статистичні ймовірності, щоб дати комп'ютерам можливість «вчитися» без явного програмування.

Регресія — метод, який використовується для прогнозування діапазону числових значень, таких як продажі, температури або ціни на акції, на основі певного набору даних.

2.3.2 Переваги Data Mining

Дані надходять у бізнес у багатьох форматах з великою швидкістю та у великому обсязі. Успіх бізнесу залежить від того, наскільки швидко можна знайти інформацію у великій кількості даних і впровадити їх у бізнес-рішення та процеси, забезпечуючи краще функціонування підприємства. Однак з великою кількістю даних такий аналіз може здатися нездоланним завданням.

Data mining дає можливість підприємствам оптимізувати майбутнє, розуміючи минуле і сьогодення, а також роблячи точні прогнози щодо того, що, ймовірно, станеться далі.

Наприклад, аналіз даних може дати інформацію, які потенційні клієнти, імовірно, стануть прибутковими клієнтами на основі профілів минулих клієнтів, а хто, швидше за

все, відповість на конкретну пропозицію. Маючи ці знання, можна підвищити рентабельність інвестицій (ROI), зробивши пропозицію лише тим потенційним клієнтам, які, ймовірно, відреагують і стануть цінними клієнтами.

Можна використовувати Data mining для вирішення практично будь-якої бізнес-проблеми, яка включає дані, зокрема:

- Збільшення доходу.
- Розуміння побажань клієнтів.
- Залучення нових клієнтів.
- Покращення перехресних продажів і підвищення продажів.
- Утримання клієнтів і підвищення лояльності.
- Підвищення рентабельності інвестицій від маркетингових кампаній.
- Виявлення шахрайства.
- Виявлення кредитних ризиків.
- Моніторинг операційних показників.

Загалом, переваги для бізнесу від Data mining полягають у підвищеній здатності виявляти приховані закономірності, тенденції, кореляції та аномалії в наборах даних. Цю інформацію можна використовувати для покращення прийняття бізнес-рішень і стратегічного планування за допомогою комбінації традиційного аналізу даних і прогнозної аналітики.

Конкретні переваги Data mining включають наступне:

- **Більш ефективний маркетинг і продажі**. Data mining допомагає маркетологам краще розуміти поведінку та вподобання клієнтів, що дозволяє їм створювати цільові маркетингові та рекламні кампанії. Аналогічно, відділи продажів можуть використовувати результати аналізу даних, щоб підвищити коефіцієнт конверсії потенційних клієнтів і продавати додаткові продукти та послуги наявним клієнтам.
- **Краще обслуговування клієнтів**. Завдяки аналізу даних компанії можуть швидше виявляти потенційні проблеми з обслуговуванням клієнтів і надавати агентам контакт-центру актуальну інформацію для дзвінків та онлайн-чатів із клієнтами.
- **Покращене управління ланцюгом поставок**. Організації можуть помічати ринкові тенденції та точніше прогнозувати попит на продукцію, що дозволяє їм краще керувати запасами товарів.

Менеджери ланцюга постачання також можуть використовувати інформацію з аналізу даних для оптимізації складських, розподільних та інших логістичних операцій.

- **Збільшений час безвідмовної роботи**. Видобуток робочих даних із датчиків на виробничих машинах та іншому промисловому обладнанні підтримує програми прогнозного технічного обслуговування для виявлення потенційних проблем до того, як вони виникнуть, допомагаючи уникнути незапланованих простоїв.
- **Посилене управління ризиками**. Ризик-менеджери та керівники бізнесу можуть краще оцінити фінансові, юридичні, кібербезпекові та інші ризики для компанії та розробити плани управління ними.
- **Менші витрати**. Data mining допомагає заощадити кошти за рахунок операційної ефективності в бізнес-процесах і зменшити надлишковість у витратах.

Зрештою, Data mining ініціативи можуть призвести до підвищення доходів і прибутків, а також до конкурентних переваг, які відрізняють компанії від їхніх конкурентів.

Завдяки застосуванню методів аналізу даних, рішення можуть ґрунтуватися на справжній бізнес-аналітиці, а не на інстинктах чи інтуїтивних реакціях. Таким чином, забезпечуються постійні результати, які тримають бізнес попереду конкурентів.

Оскільки широкомасштабні технології обробки даних, такі як машинне навчання та штучний інтелект, стають доступнішими, компанії тепер можуть отримувати терабайти даних за хвилини чи години, а не за дні чи тижні. Це допомагає їм впроваджувати інновації та розвиватися швидше.

2.3.3 Як працює Data Mining

Побудова типового проєкту з використанням Data mining починається з постановки правильного бізнес-питання, збору правильних даних для відповіді на нього та підготовки даних для аналізу. Успіх на пізніх етапах залежить від того, що відбувається на попередніх етапах. Погана якість даних призведе до поганих результатів, тому потрібно забезпечити перш за все хорошу якість даних.

Практикуючі спеціалісти з аналізу даних зазвичай досягають своєчасних надійних результатів, дотримуючись структурованого, повторюваного процесу, який включає шість кроків:

1) **Розуміння бізнесу** — розвиток глибокого розуміння параметрів проекту, включаючи поточну бізнес-ситуацію, головну бізнес-ціль проекту та критерії успіху.
2) **Розуміння даних** — визначення даних, які знадобляться для вирішення проблеми, збір цих даних з усіх доступних джерел.
3) **Підготовка даних** — підготовка даних у відповідному форматі, щоб відповісти на бізнес-питання. Виправлення будь-яких проблем із якістю даних, таких як відсутність або повторення даних.
4) **Моделювання** — використання алгоритмів для визначення закономірностей.
5) **Оцінка** — визначення того, чи допоможуть результати, отримані даною моделлю, досягти бізнес-цілі і наскільки. Часто ця фаза є ітеративною, щоб знайти найкращий алгоритм для досягнення найкращого результату.
6) **Розгортання** — надання результатів проекту для осіб, які приймають рішення.

Протягом цього процесу тісна співпраця між експертами в певній предметній області та Data Mining спеціалістами є важливою для розуміння значення Data Mining для бізнес-питання, яке досліджується.

В цілому процес Data Mining можна розділити на чотири основні етапи, які зображено на рисунку 2.2.

Рисунок 2.2 – Чотири основні етапи Data Mining

Розглянемо їх докладніше.

- **Збір даних**. Виявлено та зібрано відповідні дані для аналітичної програми. Дані можуть розташовуватися в різних вихідних системах, сховищах даних або озері, що стає все більш поширеним сховищем для великих даних, які містять поєднання структурованих і неструктурованих даних. Можна також використовувати зовнішні джерела даних. Звідки б не надходили дані, data science спеціаліст часто переміщує їх в озеро даних для решти кроків процесу.
- **Підготовка даних**. Цей етап включає в себе набір кроків для підготовки даних до майнінгу, тобто процесу обробки. Він починається з дослідження, профілювання та попередньої обробки даних, а потім очищення даних для усунення помилок та інших проблем із якістю даних. Трансформація даних також виконується, щоб зробити набори даних узгодженими, якщо тільки спеціаліст з даних не хоче проаналізувати нефільтровані вихідні дані для певної програми.
- **Видобуток даних**. Після того, як дані підготовлені, спеціаліст з даних вибирає відповідну техніку аналізу даних, а потім реалізує один або кілька алгоритмів для видобутку даних. У програмах машинного навчання алгоритми, як правило, повинні бути навчені на тестових наборах даних, перш ніж вони будуть запущені з повним набором даних.
- **Аналіз та інтерпретація даних**. Результати аналізу даних використовуються для створення аналітичних моделей, які можуть допомогти сприяти прийняттю рішень та іншим бізнес-діям. Дослідник

даних або інший член команди з науки про дані також повинен повідомити про результати керівникам бізнесу та користувачам.

2.3.4 Приклади використання Data Mining

Нижче наведено приклади того, як організації деяких галузей використовують Data Mining як частину аналітичних програм:

- **Роздрібна торгівля**. Інтернет-магазини збирають дані про клієнтів та записи кліків в Інтернеті, щоб допомогти їм націлити маркетингові кампанії, оголошення та рекламні пропозиції на окремих покупців. Data Mining і прогнозне моделювання також забезпечують рекомендаційні механізми, які пропонують відвідувачам веб-сайту можливі покупки, а також діяльність з управління запасами та ланцюгом поставок.
- **Фінансові послуги**. Банки та компанії, що займаються кредитними картками, використовують інструменти аналізу даних для створення моделей фінансового ризику, виявлення шахрайських транзакцій і перевірки заявок на позику та кредит. Data Mining також відіграє ключову роль у маркетингу та у визначенні потенційних можливостей збільшення продажів із наявними клієнтами.
- **Страхування**. Страховики покладаються на аналіз даних, щоб допомогти в ціноутворенні страхових полісів і вирішувати, чи схвалювати програми полісів, включаючи моделювання ризиків та управління ними для потенційних клієнтів.
- **Виробництво**. Додатки для аналізу даних для виробників включають зусилля, спрямовані на покращення часу безперебійної роботи та ефективності роботи на виробничих підприємствах, продуктивність ланцюга поставок та безпеку продукції.

- **Розваги**. Служби потокової передачі здійснюють аналіз даних, щоб аналізувати, що користувачі дивляться або слухають, і давати персоналізовані рекомендації на основі звичок людей до перегляду та прослуховування.
- **Охорона здоров'я**. Data Mining допомагає лікарям діагностувати захворювання, лікувати пацієнтів та аналізувати результати рентгенівських знімків та інших медичних зображень. Медичні дослідження також сильно залежать від аналізу даних, машинного навчання та інших форм аналітики.

Також можна розглянути наступний приклад. Компанія «Air France KLM» обслуговує клієнтів з питань подорожей — тобто, авіакомпанія використовує методи аналізу даних для створення всебічного огляду клієнта, інтегруючи дані пошуку поїздок, бронювання та операцій польотів із взаємодіями з Інтернетом, соціальними мережами, кол-центром та залами аеропорту. Вони використовують це глибоке розуміння клієнтів, щоб створити персоналізований досвід подорожей.

2.3.5 Майбутнє Data Mining

Майбутнє є світлим для Data mining та інших наук про дані, оскільки обсяг даних буде тільки збільшуватися.

Подібно до того, як методи видобутку корисних копалин розвивалися та вдосконалювалися завдяки вдосконаленню технологій, так само існують і технології для отримання цінної інформації з даних. Колись лише такі організації, як NASA, могли використовувати свої суперкомп’ютери для аналізу даних — вартість зберігання та обчислення даних була надто великою. Зараз компанії роблять різноманітні корисні речі за допомогою машинного навчання, штучного інтелекту та глибокого навчання за допомогою хмарних сховищ даних.

Наприклад, Інтернет речей та ґаджети, які завжди знаходяться з користувачами (телефони, смарт-годинники, фітнес-браслети тощо), фактично перетворили людей і пристрої на машини для генерації даних, які можуть давати необмежену інформацію про людей і організації — якщо компанії можуть збирати, зберігати та аналізувати дані досить швидко.

До 2025 року кількість пристроїв IoT (з англ. «Internet of Things», Інтернет Речей), ймовірно, буде понад 27 мільярдів з'єднань. Дані, згенеровані в результаті їх діяльності, будуть доступні в хмарі, що створить нагальну потребу в гнучких, масштабованих інструментах аналітики, які можуть обробляти масу інформації з різнорідних наборів даних.

Хмарні аналітичні рішення роблять доступ до великих даних і обчислювальних ресурсів для організацій більш практичними та економічно ефективними. Хмарні обчислення допомагають компаніям швидко збирати дані з продажу, маркетингу, Інтернету, систем виробництва, складів та інших джерел; зберегти та підготувати їх; проаналізувати їх; і діяти так, щоб покращити результати діяльності підприємства.

Інструменти аналізу даних з відкритим вихідним кодом також надають користувачам нові рівні потужності та гнучкості, задовольняючи аналітичні вимоги так, як багато традиційних рішень не можуть. Вони пропонують широкі спільноти аналітиків і розробників, де користувачі можуть ділитися проектами та співпрацювати над ними. Крім того, передові технології, такі як машинне навчання та штучний інтелект, тепер доступні практично будь-якій організації, яка має потрібних людей, дані та інструменти.

2.4 Загальний перелік OLAP-систем та їх класифікація

У сфері бізнесу є три основні цілі застосування OLAP-технологій: аналіз даних, планування бюджету, фінансова консолідація. Багатовимірна модель даних, можливість проводити аналіз великого обсягу даних та швидке оброблення запитів роблять OLAP-технології безальтернативним механізмом для аналізу продажів, маркетингових кампаній та інших завдань з великою кількістю вихідних даних. OLAP-куб містить дані та інформацію про виміри (агрегати) [6].

Зараз представлено величезну кількість різноманітних OLAP-систем. Розроблено декілька класифікацій продуктів цього типу: наприклад, класифікація за способом зберігання даних, за місцезнаходженням OLAP-машини, за ступенем готовності до застосування. Нижче наведено першу із наведених класифікацій.

На даний момент є три способи зберігання даних в OLAP-системах або ж три архітектури OLAP-серверів [77]:

- багатовимірна OLAP (MOLAP - Multidimensional OLAP);
- реляційна OLAP (ROLAP - Relational OLAP);
- гібридна OLAP (HOLAP - Hybrid OLAP).

Таким чином, згідно з цією класифікацією OLAP-продукти можуть бути представлені трьома класами систем.

- **Багатовимірна OLAP, MOLAP (з англ. «Multidimensional OLAP»).** Фактично цей тип є класичною формою OLAP, тому її часто називають просто OLAP. Дані попередньо обчислюються та зберігаються в MOLAP. Використовуючи MOLAP, користувач може використовувати дані багатовимірного подання з різними аспектами. У даному випадку, вихідні та багатовимірні дані зберігаються в багатовимірній БД або в багатовимірному локальному кубі. Такий спосіб зберігання забезпечує високу швидкість виконання OLAP-операцій. Але багатовимірна база в цьому випадку скоріш за все буде надлишковою. Куб, який є побудованим на її основі, буде дуже залежним від числа вимірів. При збільшенні кількості вимірів об'єм кубу буде експоненційно зростати. В деяких випадках це може призвести до різкого "вибухового зростання" обсягу даних, що в результаті може «паралізувати» запити користувачів.

- **Реляційна OLAP, ROLAP (з англ. «Relational OLAP»).** Даний тип OLAP працює безпосередньо з реляційним сховищем, факти та таблиці з вимірами зберігаються у реляційних таблицях, і для зберігання агрегатів створюються додаткові реляційні таблиці. Тобто в таких продуктах вихідні дані зберігаються в реляційних БД або плоских локальних таблицях на файл-сервері. Агрегатні дані можуть зберігатися у службових таблицях тієї ж БД. Перетворення даних з реляційної БД на багатовимірні куби відбувається за запитом OLAP-засобів. При цьому швидкість побудови кубу сильно залежать від типу джерела даних, і тому, в деяких випадках, час відгуку системи стає неприйнятно великим.
- **Гібридна OLAP, HOLAP (з англ. «Hybrid OLAP»).** Що стосується використання гібридної архітектури, тобто у HOLAP-продуктах, вихідні дані залишаються в реляційній базі, а агрегати розміщуються у багатовимірній таблиці. Побудова OLAP кубу виконується за запитом OLAP-засобу на основі реляційних та багатовимірних даних. Такий підхід дозволяє уникнути різкого збільшення даних. В такому випадку можна досягти оптимального часу виконання клієнтських запитів [7].

Наступною класифікацією є класифікація за місцем розміщення OLAP-продукту. За цією ознакою OLAP-продукти поділяються на:

- OLAP-сервери;
- OLAP-клієнти.

У OLAP-засобах серверного типу зберігання та обчислення агрегатних даних виконуються окремим процесом – сервером. Клієнтська програма отримує лише результати запитів до багатовимірних кубів, що зберігаються на сервері. Деякі OLAP-сервери підтримують зберігання даних лише у реляційних базах, інші - лише у багатовимірних. Багато сучасних OLAP-серверів підтримують усі три способи зберігання даних: MOLAP, ROLAP та HOLAP. Зараз одним з найпоширенішим серверним рішенням є OLAP-сервер корпорації Microsoft.

В свою чергу, OLAP-клієнт влаштований дещо іншим чином. Побудова багатовимірного куба та OLAP-обчислення виконуються у пам'яті клієнтського комп'ютера.

За допомогою використання OLAP-серверу може бути організоване фізичне зберігання обробленої багатовимірної інформації [8], що дозволяє швидко видавати

відповіді на запити користувача. Крім того, передбачається перетворення даних з реляційних та інших баз багатовимірних структур в режимі реального часу. OLAP продукти інтегруються в існуючу корпоративну інфраструктуру шляхом інтегрування з реляційними системами. Адміністратори баз даних або завантажують реляційні дані в багатовимірний кеш, або налаштовують кеш для доступу до даних SQL.

У таблиці 2.1 наведено порівняльні характеристики різних моделей керування даними [8]:

Таблиця 2.2 – Порівняльні характеристики різних моделей управління даними

Характеристики	Реляційні СУБД OLTP	Реляційні СУБД СППР / Сховища даних	Багатовимірні СУБД OLAP
Типова операція	Оновлення	Звіт	Аналіз
Рівень аналітичних вимог	Низький	Середній	Високий
Об'єм даних на транзакцію	Невеликий	Від малого до великого	Великий
Рівень даних	Детальні	Детальні та сумарні	Здебільшого сумарні
Терміни зберігання даних	Тільки поточні	Історичні та поточні	Історичні, поточні та прогнозовані
Структурні елементи	Записи	Записи	Масиви

2.5 Інтеграція OLAP та Data Mining

Обидві технології можна розглядати як складові процесу підтримки прийняття рішень. Однак ці технології рухаються в дещо різних напрямках – OLAP концентрується на забезпеченні доступу до багатовимірних даних, а методи Data Mining частіше за все працюють з одновимірними плоскими таблицями та реляційними даними.

Інтеграція технологій OLAP та Data Mining "збагачує" функціональність як однієї, так й іншої технології. Ці два види аналізу можуть бути тісно поєднані, для того, щоб інтегрована технологія могла забезпечувати одночасно багатовимірний доступ та пошук закономірностей. Зараз багато компаній створили гарні сховища даних, ідеально розклавши по «поличках» величезну кількість інформації, яка не використовується, яка сама по собі не забезпечує ні швидкої, ні достатньо грамотної реакції на ринкові події.

Деякі науковці вводять сукупний термін "OLAP Data Mining" (багатовимірний Data Mining) для позначення наступного об'єднання.

Засіб багатовимірного інтелектуального аналізу даних має знаходити закономірності як у деталізованих, так і в агрегованих з різним ступенем узагальнення даних. Аналітика багатовимірних даних повинна виконуватися над спеціального виду гіперкубом, комірки якого містять не довільні чисельні значення (кількість подій, обсяг продажів, сума зібраних податків), а числа, що визначають ймовірність відповідного поєднання значень атрибутів. Проекції такого гіперкуба (що виключають із розгляду окремі виміри) також мають досліджуватися щодо пошуку закономірностей.

Деякі науковці пропонують дещо спрощену назву - "OLAP Mining" і називають декілька варіантів інтеграції двох технологій [9].

- "**Cubing then mining".** Можливість виконання інтелектуального аналізу має забезпечуватися над будь-яким результатом запиту до багатовимірного концептуального подання, тобто над будь-яким фрагментом будь-якої проекції гіперкубу показників.
- **"Mining then cubing"**. Подібно до даних, вилучених зі сховища, результати інтелектуального аналізу повинні подаватися в формі гіперкубу для подальшого багатовимірного аналізу.
- **"Cubing while mining"**. Цей гнучкий спосіб інтеграції надає можливість автоматично активувати однотипні механізми інтелектуальної обробки над результатами кожного кроку багатовимірного аналізу (переходу між рівнями узагальнення, отримання нового фрагмента гіперкубу і т.д.).

На сьогоднішній день лише деякі розробники реалізують Data Mining для даних багатовимірного типу. Окрім того, деякі методи Data Mining, наприклад, метод найближчих сусідів або байєсівська класифікація, через їхню нездатність працювати з агрегованими даними, не застосовуються до багатовимірних даних.

2.6 Сховища даних

Варто зазначити, що інформаційні системи сучасних підприємств часто організовані в такий спосіб, щоб мінімізувати час введення та коригування даних, тобто інформаційні системи організовано не оптимально з погляду проектування бази даних. Методологія такого вигляду ускладнює доступ до історичних, тобто архівних, даних. Зміни структур у базах даних інформаційних систем дуже трудомісткі, інколи ж просто неможливі.

Для ефективного ведення сучасного бізнесу потрібна актуальна інформація, яка надається у зручному для аналізу вигляді та в реальному масштабі часу. Наявність такої інформації дозволяє оцінювати поточний стан справ та робити прогнози на майбутнє і, як

наслідок, приймати більш виважені та обґрунтовані рішення. До того ж, варто зазначити, що основою прийняття рішень мають слугувати справжні дані.

Якщо дані зберігаються в класичних БД інформаційних систем підприємства, то при їх аналізі може виникати низка складнощів, зокрема:

- значно збільшений час, потрібний для обробки запитів;
- можуть з’являтися проблеми з підтримкою форматів даних різного типу, а також їх кодуванням;
- неможливість аналізу тривалих рядів ретроспективних даних тощо.

Проблеми такого типу вирішуються використанням сховищ даних. Головними задачами таких сховищ є інтеграція, актуалізація та узгодження оперативних даних із джерел різного типу задля формування цілісного та стійкого єдиного погляду на об'єкт управління. За допомогою використання СД стає можливим складання звітності різного типу, а також виконання аналітичної обробки в реальному часі (OLAP) та Data Mining.

Американський вчений Білл Інмон визначає сховища даних як "набори даних, що підтримують хронологію; є організованими з метою підтримки управління; а також являються інтегрованими, незмінними та предметно орієнтованими" й повинні бути одним джерелом істинної інформації, яке забезпечує аналітиків та менеджерів інформацією, яка є достовірною та необхідною для оперативного аналізу інформації та прийняття рішень [10].

Нижче наведено особливості вищезгаданих властивостей.

- **Інтегрованість**. Дана властивість означає, що дані задовольняють не лише одній функції бізнесу, а вимогам всього підприємства. Цим СД гарантує, що однакові звіти, які були згенеровані для користувачів, наприклад аналітиків, будуть містити однакові результати.
- **Незмінність.** Ця властивість означає, що, дані, які потрапили в сховище один раз, ніколи не зміняться при зберіганні. Причому дані до СД можуть лише додаватися.
- **Прив'язка до часу.** Дана властивість означає, що СД є сукупністю даних, відновлення яких є можливим в будь-який момент часу. Часовий атрибут задано явно у структурах СД.
- **Предметна орієнтація.** Ця властивість СД означає, що дані, які були об'єднані в певні категорії, та зберігаються відповідно до областей, безпосередньо які вони описують, а не відповідно до використань, які ними оперують.

Розглянемо основні переваги використання сховищ даних.

- СД зберігають інформацію за весь часовий інтервал, аж до декількох десятків років, в єдиному інформаційному просторі. Завдяки цій властивості СД ідеально підходять для аналізу сезонних залежностей, виявлення трендів та аналітичних показників іншого типу.
- Зазвичай, інформаційні системи різних підприємств зберігають і подають однакові дані різними способами. Тобто, однакові показники можуть зберігатися в різних одиницях вимірювання або ж - однакові продукти чи клієнти можуть мати різні іменування. У системах СД невідповідності в даних усуваються на етапі додавання даних в єдину БД. Варто зазначити, що при цьому формуються єдині довідники, показники яких зводяться до одиниць вимірювання, які є однаковими.
- На кроці занесення інформації в СД інформація заздалегідь підлягає обробці, але доволі часто внаслідок помилок введення операторами СД містять деяку кількість даних, які є неправильними. Дані за спеціальною технологією підлягають до перевірки на відповідність заданим критеріям та, при необхідності, корегуються. Отже, дана властивість забезпечує своєчасне сповіщення адміністратора про помилки у вхідній інформації, а також забезпечує створення аналітичних звітів.
- Використання СД забезпечує універсалізацію доступу до інформації. Тобто сховище даних надає можливість отримувати звіти будь-якого типу про діяльність підприємства на основі певного конкретного джерела інформації. Завдяки інтегруванню інформації, яка вводиться з вже занесеними даними, їх доволі легко та просто порівнювати. Варто зазначити, що користувач під час цього процесу не втрачає доступу до системи.
- Збільшення швидкості отримання аналітичних звітів. Формування та отримання звітів з використанням засобів, що надаються OLAP-системами, є неоптимальним способом. Системи такого типу витрачають досить велику кількість часу на агрегацію інформації,

тобто, наприклад, розрахунок мінімальних, максимальних, сумарних, середніх значень, тощо. Варто зазначити, що в базі OLAP-системи знаходиться тільки найнеобхідніша та найсвіжіша інформація, в той час як інформація за попередні періоди поміщається в архів. У випадку, якщо дані потрібно діставати з архіву, тривалість формування звіту, відповідно, збільшується в декілька разів. Не варто також забувати, що сервер операційної системи може не забезпечити необхідну продуктивність при одночасному введенні інформації та відповідно побудові складних звітів. Це може негативно позначатися на роботі підприємства, так як оператори, або ж адміністратори, не зможуть оформляти необхідні документи, фіксувати певні дії над продукцією в той час, коли відбувається формування чергового звіту. Використання СД дозволяє позбутися цих проблем. Адже, по-перше, робота сервера СД не впливає на роботу й відповідно не перешкоджає роботі. По-друге, в сховищах окрім інформації в детальному представленні містяться також і наперед прораховані агреговані значення. І, по-третє, в СД архівні дані завжди є доступними для використання та додавання в звіти. Ці особливості можуть значно зменшити час формування звітів та, відповідно, уникнути проблем в роботі.

- Створення запитів довільного вигляду. Дані в СД недостатньо лише структурувати та централізовувати. Користувачам потрібні також засоби відображення цієї інформації, тобто інструмент, за допомогою використання якого можна легко отримувати інформацію, що потрібна для ухвалення своєчасних рішень. Однією з головних та критичних вимог кожного аналітика є легкість створення звітів, а також їх наочність. Побудова звітів може бути позбавлена деякої гнучкості, адже для виконання операції створення нового звіту, необхідно залучати фахівців інформаційного відділу, які, в свою чергу, об'єднають дані, які знаходяться в різних системах. У разі ж використання СД даного недоліка можна позбутись використанням

OLAP-технології. Як вже зазначалося раніше, дана технологія може надати доступ до даних в досить короткий проміжок часу, який є дуже критичним для аналізу даних. OLAP-технологія ґрунтується на концепції представлення інформації у багатьох вимірах. Насправді, кожне числове значення, яке міститься в СД, може мати навіть декілька десятків різних атрибутів. Наприклад, кількість проданих певних товарів певного типу в певному регіоні, які відбулися за певний проміжок часу тощо. Отже, варто зазначити, що робота в СД відбувається з гіперкубами, тобто багатовимірними структурами даних, числові значення в яких розташовані на перетинах декількох різних вимірів. Даний підхід оперування над даними застосовується в OLAP-системах. Системи такого типу надають змогу користуватися OLAP-маніпуляціями, тобто зручними засобами навігації по багатовимірних кубах. За допомогою таких маніпуляцій аналітик має змогу отримувати інформацію в різному представленні, наприклад, отримати зріз за певний період чи переглянути дані за іншими критеріями.

Таким чином, з вищезазначених переваг використання СД, можна значно полегшити, а також поліпшити швидкість й покращити процес Data Mining. Впровадження технологій, про які йшлося вище, надає як розробникам, так і звичайним користувачам переваги перед використанням розрізнених класичних баз даних різноманітних інформаційних систем при розробці систем підтримки прийняття рішень [11].

Висновки до другого розділу МД

В даному розділі магістерської дисертації наведено загальні відомості про класифікацію аналітичних систем, розглянуто технологію Data Mining як частину ринку інформаційних технологій, відповідно, перелічено основні концепції, поняття та ключові особливості технології Data Mining. Також наведено загальний перелік OLAP-систем, їх класифікація та особливості й інтеграцію OLAP та Data Mining.

3. СТРУКТУРА ТА ОПИС РОБОТИ МОДУЛІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Принципи роботи програмного забезпечення для інтерактивного аналізу даних

Для того, щоб розроблене програмне забезпечення коректно розпочало свою роботу необхідно, щоб з відкритого API компанії Clover надійшли дані про певні товари, відносно яких необхідно виконати аналітичні обчислення.

Після цього дані надсилаються в модуль програмного забезпечення, який відповідає за опрацювання цих «сирих» даних до виду, який є прийнятним для подальшого продовження аналітичної обробки.

Даний модуль прийому даних, який реалізовано мовою програмування Python, направляє дані в модуль категоризації на мові Julia, який в свою чергу виконує основний пласт важких та трудомістких обчислень. Цей скрипт розділяє все на категорії – наприклад за типами товарів, по магазинам, по брендах тощо, використовуючи основні ідеї Data mining. Адже саме завдяки використанню різноманітних методів для аналізу даних різними способами, користувачі можуть визначити закономірності, тенденції та зв’язки, які при інших обставинах можна було б упустити.

Наступний модуль програмного забезпечення відповідає за зберігання оброблених даних у сховищі даних. Завдяки властивості СД зберігати інформацію за весь часовий інтервал, аж до декількох десятків років, в єдиному інформаційному просторі, СД ідеально підходять для аналізу сезонних залежностей, виявлення трендів та аналітичних показників різних типів.

Після цього відбувається прорахунок прогнозів продажів по потрібних продуктах.

Таким чином, програмне забезпечення має можливість показу кожному власнику магазину чи, наприклад, адміністратору сформованих прогнозів по продажах певних продуктів за певний інтервал часу. Завдяки цьому можна скласти орієнтований план покупок на майбутній аналогічний період часу.

3.2 Вибір інструментарію для розробки програмного забезпечення

Зараз вибір інструментарію розробки ПЗ є доволі важким питанням. Щороку кількість МП (мов програмування), фреймворків та різноманітних бібліотек невпинно зростає. Фактично різні МП призначені для різних задач. Саме тому МП Python є слушним вибором для створення програмного забезпечення для задач, які напряму пов’язані з аналітикою даних.

Python первинно та концептуально як мова програмування був винайдений наприкінці 1980-х у голландському національному дослідницькому інституті математики та інформатики (інакше CWI, Центр математики та інформатики) Гвідо ван Россумом.

Мова програмування Python фактично є похідною від багатьох інших мов, включаючи C, C ++, ABC, Modula-3, Algol-68, SmallTalk, навіть від сімейства мультизадачних ОС Unix та інших мов написання сценаріїв.

МП Python підлягає під захист авторським правом. Як і в мові програмування Perl, весь вихідний код мови Python зараз доступний під Стандартною громадською ліцензією GNU (тобто, загальнодоступна ліцензія).

На даний момент, в основному підтримкою мови програмування Python займається група розробників в інституті. Варто також зазначити, що Гвідо ван Россум, як і раніше, відіграє доволі важливу роль розвитку мови Python.

Розглянемо основні особливості МП Python:

- Дана мова легко вивчається, має кілька ключових слів, просту структуру і чітко визначений синтаксис. В цілому це надає можливість швидкого опанування мовою.
- Написаний код має високу степінь читабельності – він доволі чітко визначається та сприймається зором.
- Вихідний код самої мови програмування Python є досить простим в обслуговуванні.
- Python має велику бібліотеку стандартних підпрограм — більша частина бібліотеки мови Python є доволі портативною та мультиплатформенною, тобто є сумісною з UNIX-системами, Windows OS та Mac OS.
- Мова програмування Python має підтримку інтерактивного режиму. Він надає можливість виконання інтерактивного тестування та відлагодження фрагментів коду.
- Python має можливість роботи на різних апаратних платформах і має однаковий інтерфейс на всіх платформах.
- Мова програмування Python може легко розширювати свої можливість за рахунок додавання до інтерпретатора низькорівневих модулів, які відповідно написані на низькорівневих МП. Ця властивість надає програмістам надає можливість налагоджувати свої проекти для підвищення ефективності.
- Python має можливості створення графічного інтерфейсу.

- Python є інтерпретованою мовою програмування. Тобто фактично до запуску програми він є звичайним текстовим файлом. Відповідно програмувати можна майже на всіх платформах, хоч на портативних гаджетах.
- Python підтримує методи функціонального та структурного програмування, а також ООП.
- Python може використовуватися як мова написання сценаріїв або може бути скомпільований у байт-код для створення великих програм.
- Він надає динамічні типи даних дуже високого рівня та підтримує динамічний контроль типів.
- Підтримує автоматичне керування звільненням динамічної пам'яті.
- Може бути легко інтегрований у мови C, C++, COM, ActiveX, CORBA та Java.

Станом на сьогоднішній день останньою стабільною версією є Python 3.9.9, її реліз відбувся в листопаді 2021 року. В кінці цього ж місяця на тестування ентузіастами була випущена версія Python 3.10, але вона не є рекомендованою для використання, адже ще не була протестована належним чином [12].

На даний момент для Python найбільшою популярністю користується середовище розробки є PyCharm. Воно створено компанією JetBrains (по ряду опитувань станом на 2019 нею користувалося близько 60% розробників на Java)

Згідно з опитувань, які були проведені в 2020 році PyCharm від компанії JetBrains є найпопулярнішим інструментом для розробки Python з 35% спільної частки для PyCharm Professional та Community [8888]. Через цілий ряд корисних функці, таких як автозаповнення тексту введених команд, повне відображення синтаксису мови програмування, можливість збереження різних версій тексту програми на різні сервіси, такі як Git, Github, Gitlab дане середовище розробки і було обрано як основне (рис. 3.4).

Рисунок 3.1 – Середовище розробки Pycharm

Також варто зазначити, що дане IDE (з англ. «Integrated Development Environment») або ж ICP (Інтегроване середовище розробки) має чудові можливості відлагодження коду програми. Наприклад, наявна можливість відслідковування та знаходження помилкових ділянок коду, які можуть слугувати проблемній компіляції програми й, відповідно, також присутня можливість запуску ручного режиму відлагодження коду програми «step-by-step», тобто покроково шляхом розставлення спеціальних точок переривання (з англ. «breakpoint»), до яких, за умови відсутності помилок, буде виконано код програми. Ця особливість забезпечує реалізацію покрокового тестування програми.

Як уже зазначалося раніше, для покращення продуктивності, стабільної роботи та підвищення швидкодії скрипт, який відповідає за категоризацію брендизацію написано з використанням мови програмування Julia.

Julia — є доволі «молодою» мовою програмування, яка призначена переважно для громіздких наукових обчислень. Концептуально її творці хотіли, щоб вона зайняла нішу, яку раніше займали Matlab, його клони та R. Творці намагалися вирішити так звану проблему двох мов: поєднати зручність R та Python та продуктивність C [13].

У 2009 році Стефан Карпінськи, вчений з Нью-Йоркського Університету, поставив за мету створити мову, яка була б позбавлена обмежень MATLAB і R. З реалізацією зголосилися допомогти колеги з інших університетів: Алан Едельман, Джефф Безансон і Вірал Шах. Протягом 3 років вони працювали над концепцією нової мови, що включає:

- висока швидкодія при роботі з великими даними;
- підтримка паралельного типу програмування;
- підтримка хмарних технологій.

Також, за задумом творців, мова в майбутньому становитиме конкуренцію іншим мовам програмування у галузі візуалізації даних, моделюванні та прогнозуванні.

У 2012 році група вчених виклала у мережу першу відкриту версію Julia. До речі, історії назви мови фактично не існує, це просто гарне жіноче ім'я, яке дуже сподобалося розробникам.

Мова є динамічною, проте використовує JIT-компіляцію (з англ. «Just-in-time compilation»), тобто так звана компіляція «на льоту». Завдяки цьому досягається висока швидкість роботи додатків, написаних чистою мовою, без використання низькорівневих бібліотек і векторних операцій. Підтримується перевантаження функцій та операторів, які фактично також є функціями, при цьому опціонально можна вказувати тип для аргументів функції, чого зазвичай немає в мовах, що динамічно типізуються. Це дозволяє створювати спеціалізовані варіанти функцій та операторів для прискорення обчислень. Найбільш підходящий варіант функції вибирається автоматично у процесі виконання. Також завдяки перевантаженню операторів можна створювати нові типи даних, які поводяться подібно до вбудованих типів. Одним із пріоритетних напрямів у розвитку мови є підтримка розподілених обчислень. Також наявна велика кількість стандартних конструкцій для розпаралелювання коду.

Перша версія мови, випущена в 2012 році, не привернула належної уваги, і довгий час цією мовою цікавилися лише окремі ентузіасти з роботи з великими даними.

Однак все змінилося з приходом нової версії Julia в 2018 році, яка стала показувати чудові результати, наприклад:

- мова програмування Julia стала працювати з даними швидше, ніж Python, MATLAB, R, JavaScript;
- продуктивність Julia наблизилася до таких мов, як Go, C, Lua, Fortan;
- реалізовано часткову візуалізацію даних за допомогою підтримки бібліотек PyPlot, Winston, Gadfly.

Завдяки цим нововведенням мова програмування Julia стала використовуватися набагато частіше. Вона піднялася у багатьох рейтингах із позицій за межами першої сотні у топ-50. Також варто зазначити, що цю МП стали вивчати в деяких американських університетах як основну мову при роботі з великими даними.

Як і будь-яка інша мова програмування, Julia має власні особливості, які відрізняють цю мову від інших. Серед головних та виразних особливостей можна розглянути такі:

- висока швидкість роботи з великими даними;
- динамічна типізація;
- технологічність, яка дозволяє Julia працювати з дуже великими даними та дуже складними обчисленнями;
- опціональний опис типів даних;
- адаптивна система керування різними пакетами;
- простий синтаксис, який практично не використовує «машинних слів», а лише англомовні поняття;

- вбудована JIT-компіляція;
- підтримка бібліотек Python та C без зайвої проблематичності;
- відмінна навчальна документація;
- підтримка у синтаксисі символів Unicode;

З недоліків виділяється лише низька кількість власних бібліотек. Так, можна використати бібліотеки інших мов. Але якщо використовувати бібліотеки Python, то втрачається головна властивість мови Julia - продуктивність, відповідно, пропадає сам сенс використання цієї МП.

Також використовується система для автоматизації роботи Docker.

Docker – це платформа для розробки, доставки та запуску контейнерних програм. Docker дозволяє створювати контейнери, автоматизувати їх запуск та розгортання, керує життєвим циклом. Він дозволяє запускати безліч контейнерів на одній машині.

Контейнери – це спосіб упакувати програму та всі її залежності в єдиний образ. Цей образ запускається в ізольованому середовищі, що не впливає на основну операційну систему. Контейнери дозволяють відокремити додаток від інфраструктури: розробникам не потрібно замислюватися, в якому оточенні працюватиме їхня програма, чи будуть там потрібні налаштування та залежності. Вони просто створюють додаток, упаковують усі залежності та налаштування в єдиний образ. Потім цей образ можна запускати на інших системах, не турбуючись, що програма не запуститься.

Контейнеризація схожа на віртуалізацію, але це не одне й те саме. Віртуалізація працює як окремий комп'ютер зі своїм віртуальним обладнанням та операційною системою. При цьому всередині однієї ОС можна запустити іншу ОС. У разі контейнеризації віртуальне середовище запускається прямо з ядра основної операційної системи та не віртуалізує обладнання. Це означає, що контейнер може працювати тільки в тій самій ОС, що й основна. При цьому, оскільки контейнери не віртуалізують обладнання, вони споживають набагато менше ресурсів.

На серверній частині вебдодатку, який відповідає за відображення статистики та прогнозів, використовується Node або Node.js. Це програмна платформа, яка ґрунтується на рушії V8, який транслює JavaScript в машинний код. Тобто Node.js перетворює JavaScript з вузькоспеціалізованої мови в мову загального призначення.

В основі Node.js лежить подієво-орієнтоване та асинхронне або, як ще його називають, реактивне програмування з неблокуючим введенням та виводом.

Використання Node.js дозволяє найшвидше обробляти запити на сервер та має стабільну підтримку з боку розробників. Варто зазначити, що це є популярним рішенням, і гарантує велику підтримку з боку ентузіастів, тобто самої спільноти розробників, тим самим забезпечується можливість оперативного вирішення проблем, які можуть виникнути під час розробки.

На front-end частині вебдодатку, тобто для реалізації інтерфейсу, використовується бібліотека React.

React - це найбільш популярний JavaScript- фреймворк на сьогоднішній день, він має величезну базу прихильників, яка складається, як з професійних майстрів, так і новачків. Ця популярність обумовлюється простотою, гнучкістю та поширеністю фреймворку.

React легко може конкурувати з Vue.js, а також з Ember.js і навіть з JQuery. Його основний функціонал полягає у спрощенні роботи з елементами інтерфейсу користувача, проте React на цьому не закінчується, адже на ньому вже зараз можна створювати мобільні додатки для Android і iOS, також за допомогою додаткових програм на ньому можна створювати десктопні додатки.

3.3 Опис структури розроблених модулів

Розроблюване програмне забезпечення для інтерактивного аналізу даних з використанням технологій OLAP-кубів передбачає використання чотирьох структурних модулів.

Узагальнену схему взаємодії розроблених модулів програмного забезпечення наведено на рисунку 3.2.

Рисунок 3.2 – Схема взаємодії розроблених модулів

Перший структурний модуль програмного забезпечення реалізовано мовою програмування Python, він відповідає за прийом даних від відкритого API Clover. Як зазначалося раніше, дані з Clover API надходять у неформатованому вигляді, саме тому даний модуль виконує додаткове форматування даних для їх подальшої обробки.

Також важливо додати, що даний структурний модуль програмного забезпечення може приймати великі об’єми даних, забезпечуючи високий рівень стабільності роботи, та сприяє майбутній передачі даних у «сприятливому» вигляді до наступного модуля.

Другий структурний модуль програмного забезпечення є найбільш важливим серед усіх інших, адже він виконує найбільшу частину роботи щодо категоризації, брендизації, відтворення прогнозів та обрахунку майбутніх трендів продажу того чи іншого товару.

Даний структурний модуль реалізовано мовою програмування Julia (адже, як зазначалося раніше, дана МП надає дуже високий рівень продуктивності) та може бути легко імплементований до попереднього модуля програми при цьому не зменшуючи його стабільність роботи.

Завдяки передовим технологічним рішенням, які вбудовані в мову Julia, даний модуль розроблюваного програмного забезпечення дозволяє підвищити швидкодію обробки даних на 5-7 відсотків (згідно з дослідженнями незалежних експертів).

Категоризація (рисунок 3.3) та брендизація (рисунок 3.2) продукції, яка надійшла з Clover API, відбувається завдяки розбиттю цих продуктів на певні кластери категорій, субкатегорій та суббрендів. Мова Julia відмінно справляється з цією задачею та готує дані для подальшої обробки за допомогою OLAP-технологій. Важливо відмітити, що всі дані, які передаються до сховища даних, є вже розбитими на певні категорії, субкатегорії та суббренди, що дозволяє зручно використовувати дане розбиття в інтерактивному аналізі та прогнозах продажу товарів по конкретному магазину.

```
function change_brand(row, clean_brands, df_br)
    if row in clean_brands
        return df_br[df_br[!, :clean_brands] .== row, 2][1]
    end
end

function categorization_using_brands(clusters::Array, df_br, br_categ, unique_brands)

    for cluster in clusters
        if cluster.brand in unique_brands
            cluster.category = br_categ[br_categ[!, :brand] .== cluster.brand, :category][1]
            cluster.sub_category = br_categ[br_categ[!, :brand] .== cluster.brand, :sub_category][1]
            cluster.changable = false
        end
    end

    return clusters
end
```

Рисунок 3.2 – Частина лістингу, в якому реалізовується брендизація

```
function Category_all(clusters::Array, dfp, dict_dfp, all_tokens)

    for cluster in clusters
        if cluster.changable
            center = cluster.tokens
            mas_res = Vector{Vector{Float64}}()

            for word in center
                if word in all_tokens
                    push!(mas_res, dict_dfp[word])
                end
            end

            if length(mas_res) != 0
                mas_res = hcat(mas_res...)
                sum_res = sum(mas_res, dims = 2)

                res_min = minimum(sum_res)
                index_min = findall(mas->mas==res_min, sum_res)[1][1]

                sort!(sum_res, dims=1)

                if sum_res[2] - sum_res[1] > 1
                    cluster.category = dfp[index_min, :category_name]
                    cluster.sub_category = dfp[index_min, :sub_category_name]

                else
                    cluster.category = "Other"
                    cluster.sub_category = "Other"
                end
            else
                cluster.category = "Other"
                cluster.sub_category = "Other"
            end
        end
    end

    for cluster in clusters
        if cluster.sub_category in all_categories
            cluster.sub_category = "Non food"
        end
    end
    return clusters
end
```

Рисунок 3.3 – Частина лістингу, в якому реалізовується категоризація

Третій структурний модуль програмного забезпечення дозволяє взаємодіяти із закритим сховищем даних, яке фактично представлено у вигляді взаємопов’язаних таблиць РБД (реляційної бази даних).

Серед усіх раніше згаданих OLAP-систем найбільшою популярністю користується ROLAP, тобто реляційні OLAP. В них вхідні дані перетворюються на багатовимірну модель через проміжний шар метаданих. Під час застосування ROLAP вихідні результати вибірок зберігаються в тій же реляційній БД, де і їхні вхідні дані. Агреговані дані розташовуються в спеціально створених службових таблицях РБД. Для переходу від табличного представлення даних до гіперкубу в ROLAP-технології необхідно застосувати одну з моделей представлення даних: «зірка» (з англ. «star schema») або «сніжинка» (з англ. «snowflake schema»). Для використання моделі «зірка» потрібно перевести таблиці реляційної бази у вигляд, в якому кожен вимір кубу зберігається в іншій окремій таблиці, а для реалізації зв’язку з таблицею атрибутів застосовується відношення «один-до-багатьох». В свою чергу, при використанні моделі «сніжинка» є характерною ситуація, коли інформація про один вимір міститься в декількох різних

пов’язаних таблицях між собою, але при цьому ж з таблицею фактів поєднується таблиця з ієрархії вимірів, яка є дочірньою.

Отже, узагальнений алгоритм застосування технології ROLAP для здійснення багатовимірного аналізу РБД полягає у виконанні наступних кроків:

1) Отримання на вхід сховища даних, яке представлено у вигляді взаємопов’язаних таблиць реляційної бази даних.
2) Визначення показників та рівня бажаної деталізації багатовимірного аналізу вхідних даних.
3) Формування багатовимірної OLAP-моделі із вхідної реляційної БД через її структурне представлення у вигляді моделі «зірка» (рисунок 3.4) або «сніжинка» (рисунок 3.5). Визначення структури, кількості та вмісту таблиць фактів і вимірів.
4) Застосування потрібних аналітичних операцій для розробленого гіперкубу.
5) Обробка отриманих вихідних даних.

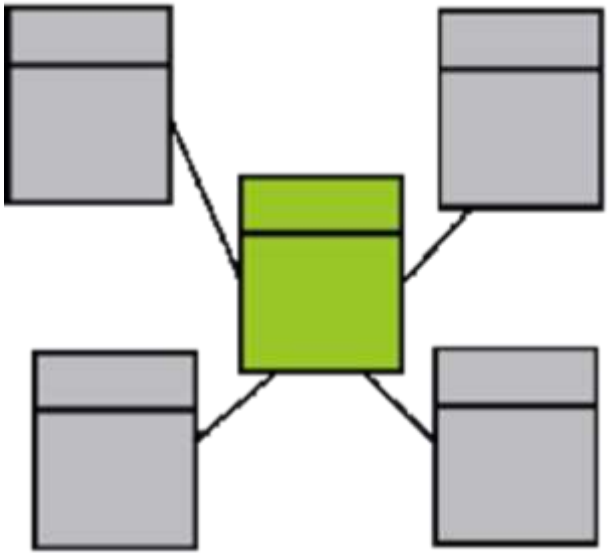

Рисунок 3.4 – Схематичний вигляд моделі «зірка» (star schema)

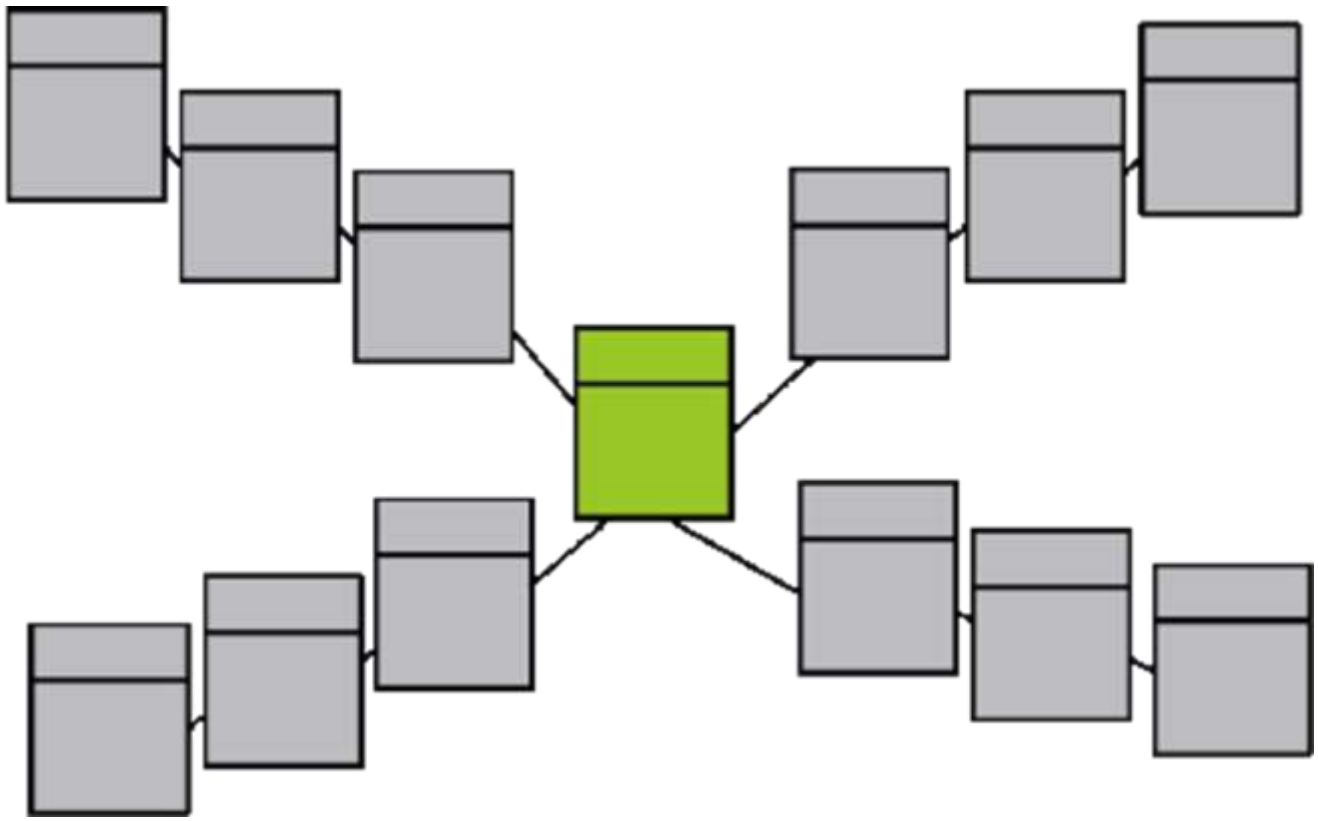

Рисунок 3.5 – Схематичний вигляд моделі «сніжинка» (snowflake schema)

Третій структурний модуль фактично є універсальним, оскільки може використовуватися із будь-якою базою даних, наприклад MySQL, ClickHouse, SQLite, тощо.

Даний функціонал є можливим завдяки наявній можливості підключення драйверу бази даних будь-якого типу.

Приклад ініціалізації сховища даних з використанням MySQL драйверу (рисунок 3.6).

```
class MysqlCubeEngine implements CubeEngine {
  _mysql: $PropertyType<Settings, 'mysqlDriver'>;
  _calculatedMeasures: $NonMaybeType<$PropertyType<Settings, 'calculatedMeasures'>>;
  _attributes: $NonMaybeType<$PropertyType<Settings, 'attributes'>>;
  _collate: $NonMaybeType<$PropertyType<Settings, 'collation'>>;

  constructor({mysqlDriver, calculatedMeasures = {}, attributes = {}, collation}: Settings = {}) {
    this._mysql = mysqlDriver;
    this._calculatedMeasures = calculatedMeasures;
    this._attributes = attributes;
    if (collation) {
      this._collate = ' COLLATE ' + collation;
    } else {
      this._collate = '';
    }
  }

  async resolveAttributes(query: AttributesQuery): Record {
    const subQuery = squel.select();
    const mainQuery = squel.select().from(subQuery, CUBE_QUERY_ALIAS);

    for (const [attribute, dimensions] of Object.entries(query)) {
      for (const [dimensionName, dimensionValue] of Object.entries(dimensions)) {
        const escapedValue = this._mysql.escape(String(dimensionValue));
        subQuery.field(escapedValue + this._collate, dimensionName);
      }
      this._attributes[attribute](mainQuery, CUBE_QUERY_ALIAS);
    }
    const records = await this._mysql.query(mainQuery);

    return records.reduce((result, record) => {
      return Object.assign(result, record);
    }, {});
  }
```

Рисунок 3.6 – Ініціалізація сховища даних з використанням MySQL драйверу

В основі третього модуля лежить бібліотека Cube.JS, яка створена на Node.JS, але з покращеними модифікаціями, які дозволяють оброблювати складні типи даних, працювати із базами даних різних типів та застосовувати складні групування даних для статистики при цьому забезпечуючи належний рівень швидкодії та стабільності роботи.

Cube.js - це фреймворк з відкритим вихідним кодом для створення аналітичних веб-застосунків. Він в основному використовується для створення внутрішніх інструментів бізнес-аналітики або додавання аналітики, орієнтованої на клієнта, в існуючу програму.

Розглянемо основні особливості Cube.js, які виділяють розробники цього ж фреймворку.

- Він надає своєрідні «будівельні блоки» для додавання аналітичних функцій у будь-який додаток, який створюється.
- Cube.js призначений для роботи з великомасштабними наборами даних та реалізує різні методи оптимізації.
- Реалізація хмарних рішень позбавляє необхідності будувати аналітику.
- Підтримує інфраструктуру та механізм запитів.

- Надає API-клієнти та SDK, а також велику документацію, для того щоб була можливість зосередитися на створенні зручного інтерфейсу користувача.

Важливо зазначити, що даний третій модуль програмного забезпечення повертає усі дані зі сховища у вигляді потоку даних, що реалізовано у вигляді відправки порцій даних зі сховища до серверної частини вебдодатку. Це дозволяє обробляти великі об’єми даних, не сповільнюючи роботу з гіперкубом. У потоці даних є певні події, які відповідають за взаємодію зі сховищем:

- data – повертаються порціями дані зі сховища;
- end – читання даних зі сховища закінчено;
- total – подія, яка відповідає за сумарне підрахування показників даних;
- error – яка відповідає за виникнення помилки під час читання даних зі сховища;
- metadata – подія, при якій додаткові метадані (тобто так звані «дані про дані») повертаються зі сховища .

```
getDataStream({
                dimensions,
                measures,
                attributes = [],
                dimensionsFilters = {},
                attributesFilters = {},
                sort = [],
                limit = 0,
                offset = 0,
                reportType = 'data_with_totals'
              }: CubeQuery): Readable {
  const cubeResult = new PassThrough({objectMode: true});
  const dimensionsInFilters = getKeysFromFilters(dimensionsFilters);
  const needToJoinAttributes = this._getAvailableAttributes(attributes, new Set([
    ...dimensions, ...dimensionsInFilters
  ]));
  const dimensionsInEqualFilters = getKeysFromFilters(extractEqualFilters(dimensionsFilters));
  const needToResolveAttributes = this._getAvailableAttributes(attributes, new Set(dimensionsInEqualFilters));
  const executionPlan = this._createExecutionPlan(measures, dimensionsInFilters, dimensions, dimensionsFilters);

  // Check available aggregations
  if (_.isEmpty(executionPlan.aggregations)) {
    appendEmptyCubeStream(cubeResult);
    return cubeResult;
  }

  process.nextTick(() => { // Wait for subscription
    cubeResult.emit('metadata', {
      measures: [...executionPlan.show],
      scores: executionPlan.scores,
      attributes: [...(new Set([...needToJoinAttributes, ...needToResolveAttributes]))]
    });
  });

  if (reportType === 'metadata') {
    return cubeResult;
  }
```

Рисунок 3.7 – Ініціалізація потоку даних зі сховища

Четвертий структурний модуль програмного забезпечення фактично являється вебдодатком для перегляду статистики, прогнозів продажу та трендів продажу продуктів в магазині. Він складається з серверної частини, яка виконує запити до Node.JS кубу для отримання статистичних даних та фронтової частини додатку, яка реалізована з використанням бібліотеки ReactJS та відповідає за відображення даних користувачу.

3.4 Опис інтерфейсу на прикладі вебдодатку для статистики

В основі будь-якого додатку, створеного для користувача, завжди є інтерфейс взаємодії. Розглянемо інтерфейс розробленого додатку для інтерактивного аналізу даних.

На рисунку 3.8 зображено вікно ініціалізації.

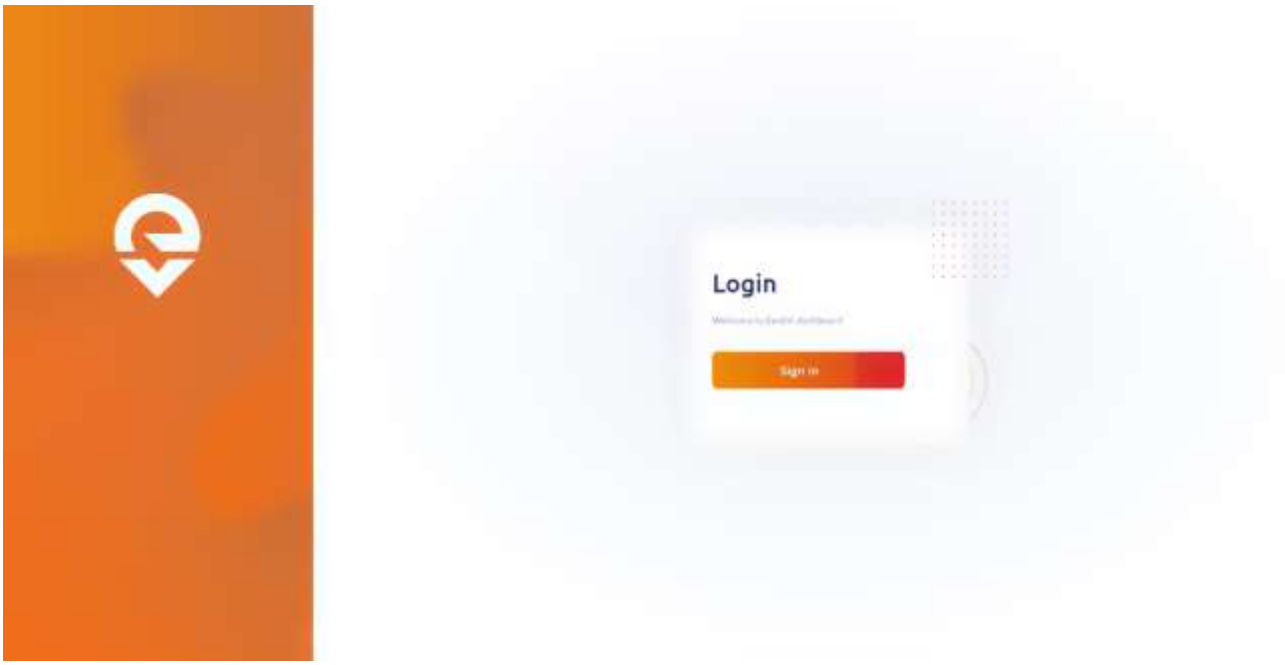

Рисунок 3.8 – Вікно ініціалізації

На рисунку 3.9 зображено типове для багатьох додатків вікно входу в систему за допомогою логіну та пароля.

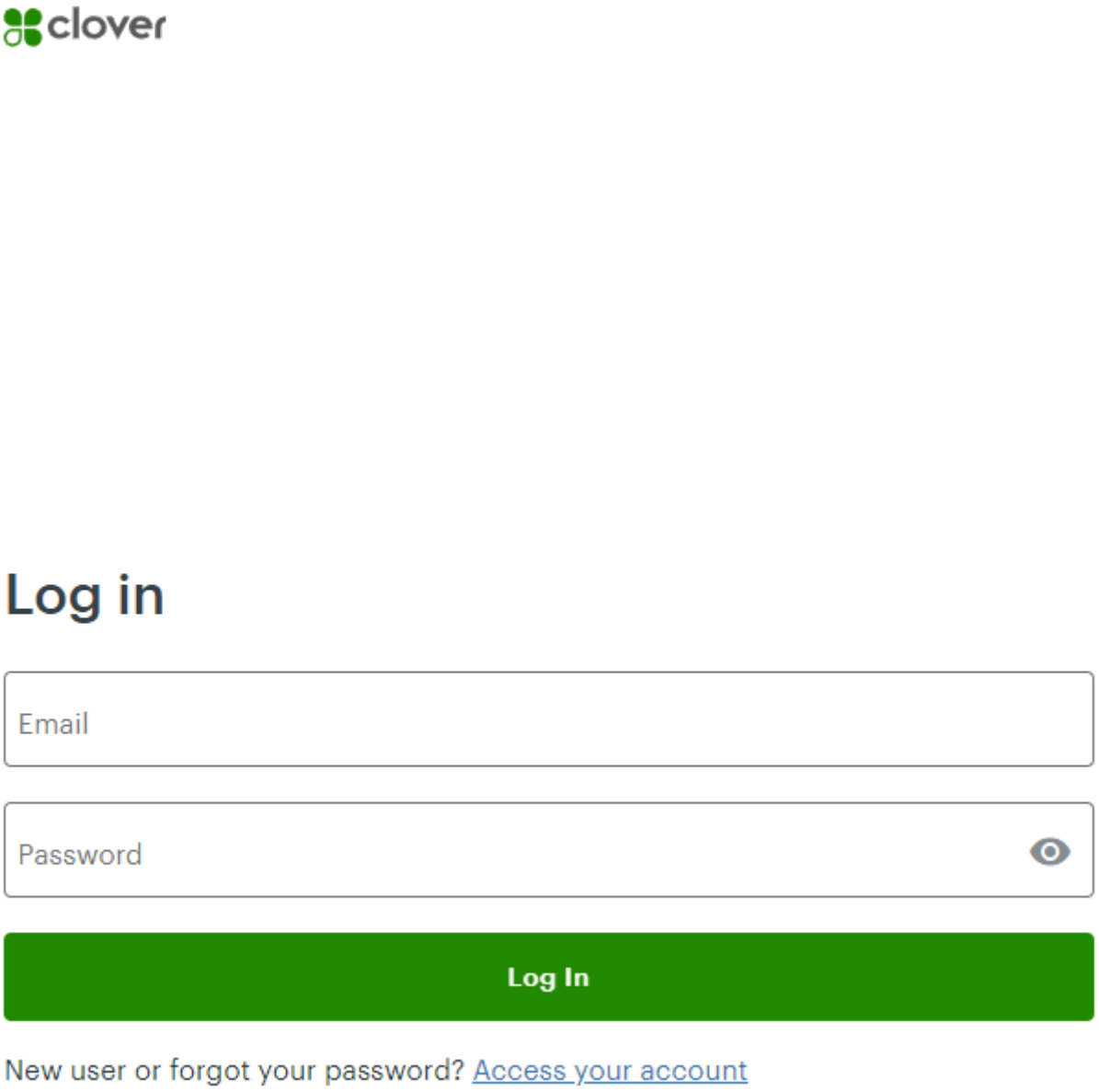

Рисунок 3.9 – Вікно входу

На рисунку 3.10 можна побачити вікно вибору магазину, в якому можна виконувати інтерактивний аналіз.

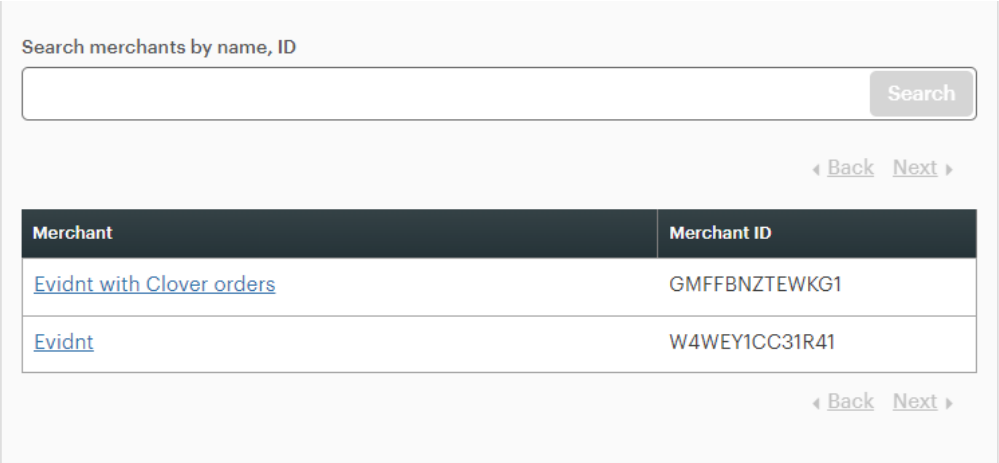

Рисунок 3.10 – Вікно вибору магазину

На рисунках 3.11 та 3.12 можна побачити інтерфейс пустого магазину, тобто магазину в якого немає даних по продажах за певний проміжок часу.

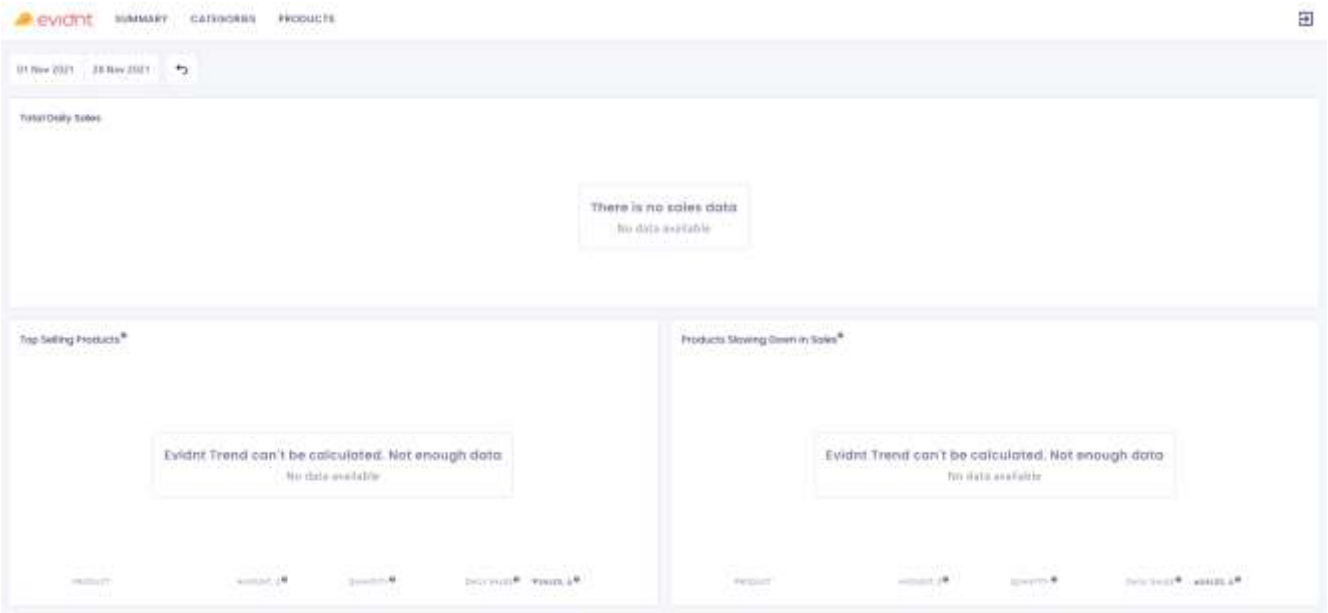

Рисунок 3.11 – Інтерфейс пустого магазину

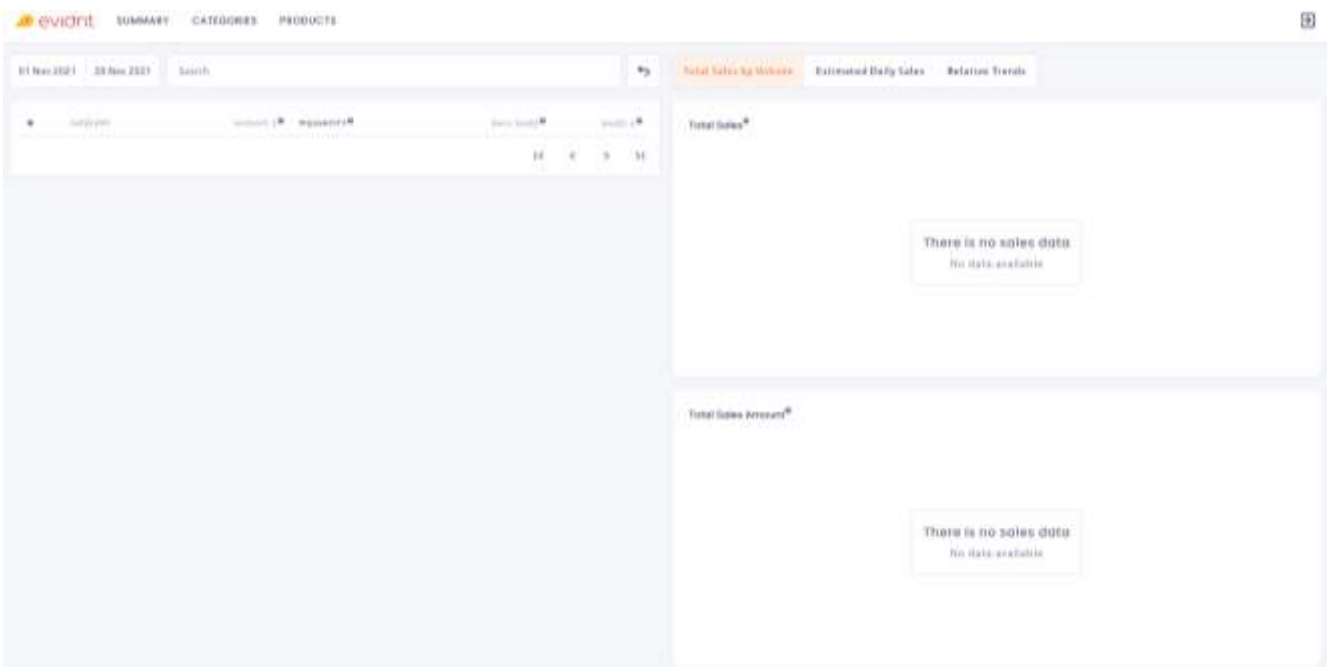

Рисунок 3.12 – Інтерфейс пустого магазину (продовження)

На рисунках 3.13 та 3.14 зображено товари, які є найпопулярнішими та користуються найменшим попитом за певний проміжок часу. Також видно на рисунках який є приблизний приріст та зменшення продажу продукції.

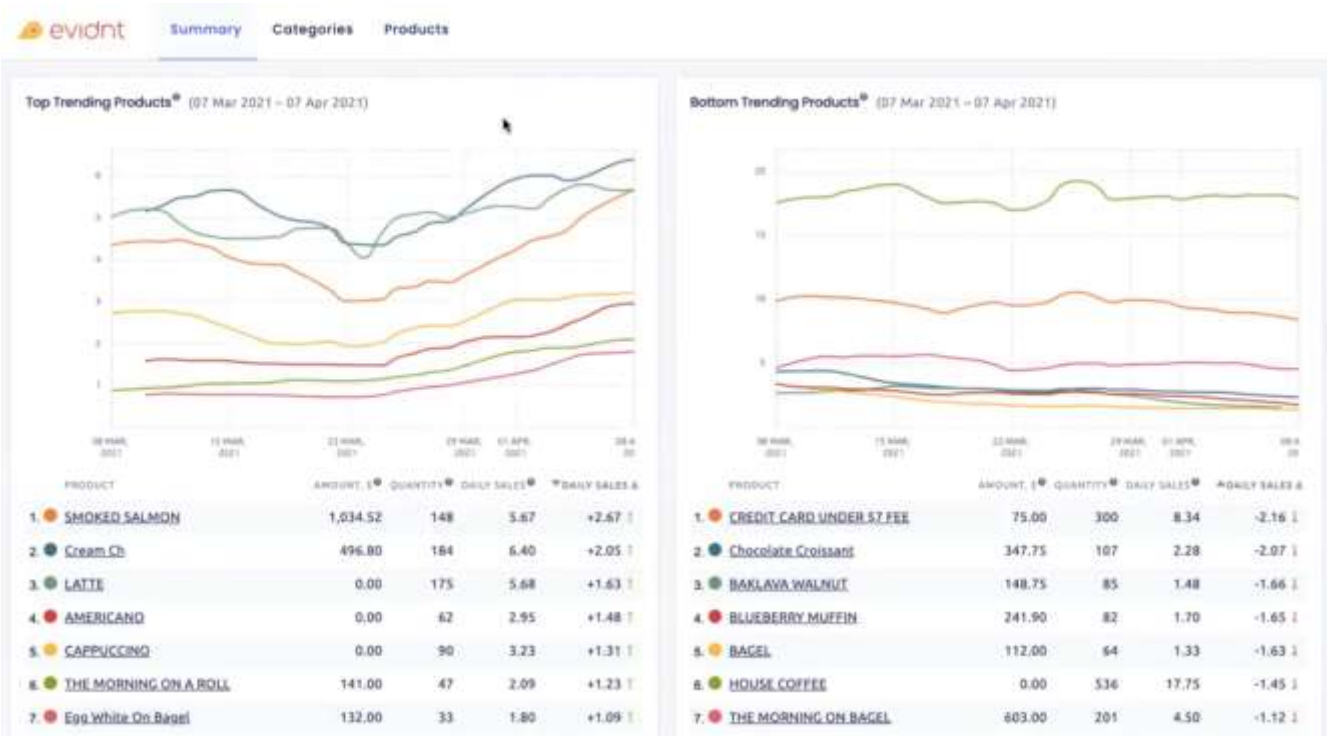

Рисунок 3.13 – Графіки продажу продуктів з прогнозами

Рисунок 3.14 – Графіки продажу продуктів з прогнозами (у відсотках)

На рисунку 3.15 можна побачити графік проданих одиниць всіх товарів за певний проміжок часу.

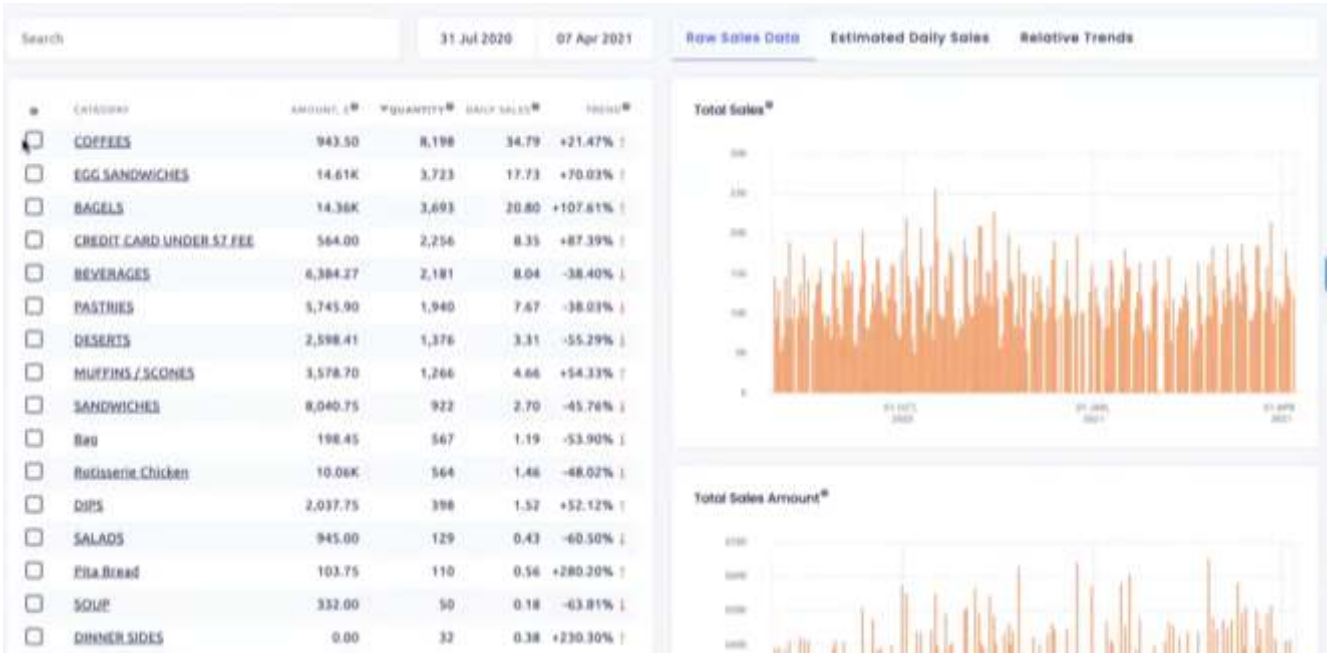

Рисунок 3.15 – Графік проданих одиниць всіх товарів

На рисунку 3.16 представлено діапазон відхилень по прогнозам та фактичні продажі одного конкретного товару. Можна побачити, що значення по прогнозах можуть варіюватися, але в цілому прогнозування відповідає справжнім продажам. Точками на графіку позначені аномальні значення тренду, які можуть виникнути протягом виникнення продаж. Такі точки прогнозуються, виходячи з попередніх різких змін за аналогічний часовий період.

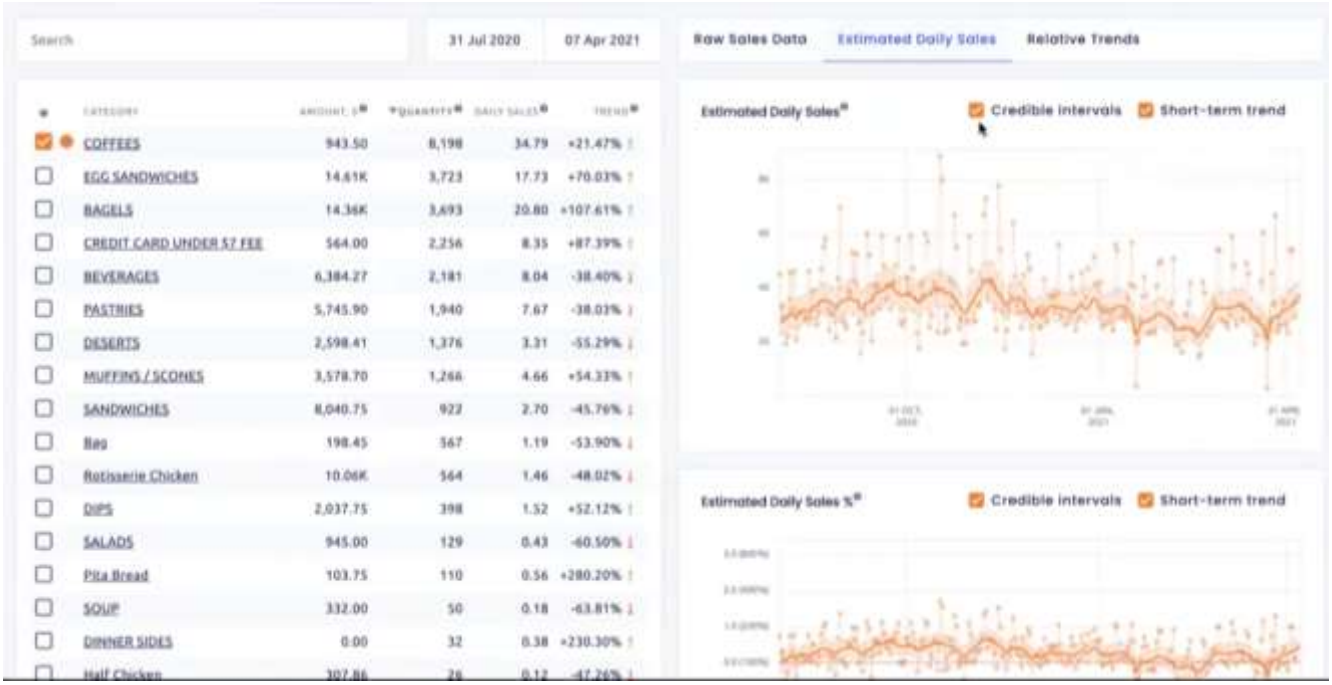

Рисунок 3.16 – Графік прогнозованих та фактичних продажів

Висновки до третього розділу МД

У третьому розділі магістерської дисертації розглянуто загальні принципи роботи створеного програмного забезпечення, обґрунтовано вибір інструментарію та методів розробки програмного забезпечення. Також проаналізовано загальну структуру, зв’язок розроблених модулів та описано інтерфейс розробленого вебдодатку.

4. ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

4.1 Тестування програмного забезпечення

Тестування програмного модуля, написаного мовою програмування Julia проводилося з тестовим набором даних у вигляді продуктів конкретних магазинів, які зберігаються в .csv файлах.

Ці дані є наближеними зразками реальних магазинів, які беруться напряму з API Clover.

На рисунку 4.1 зображено співвідношення трендів продажу двох продуктів протягом двох місяців.

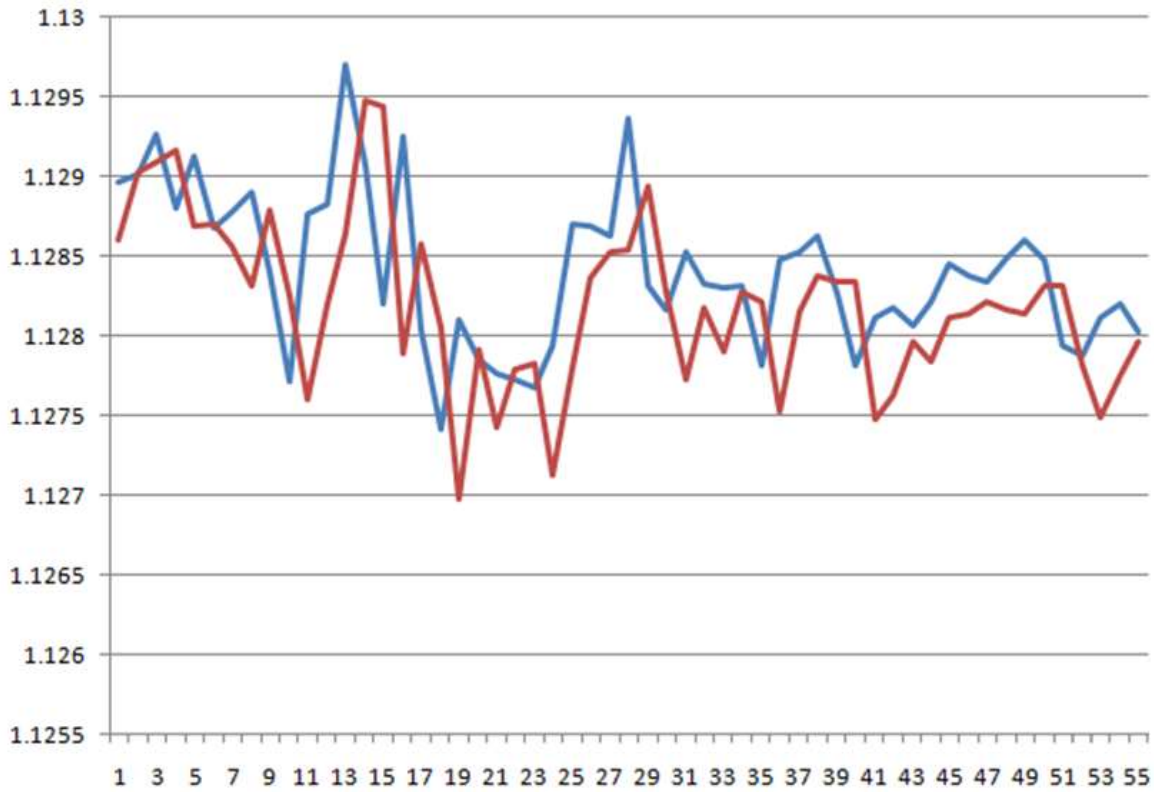

Рисунок 4.1 – Тренди продажу двох продуктів протягом двох місяців.

Для тестування програмного забезпечення, яке лежить в основі розробленого вебдодатку, використовується бібліотека Jest.JS, яка реалізована мовою JavaScript і являється найпопулярнішою бібліотекою для тестування розроблених продуктів.

На рисунку 4.2 зображено результат запуску unit тестів, тобто тестування різних розроблених компонентів.

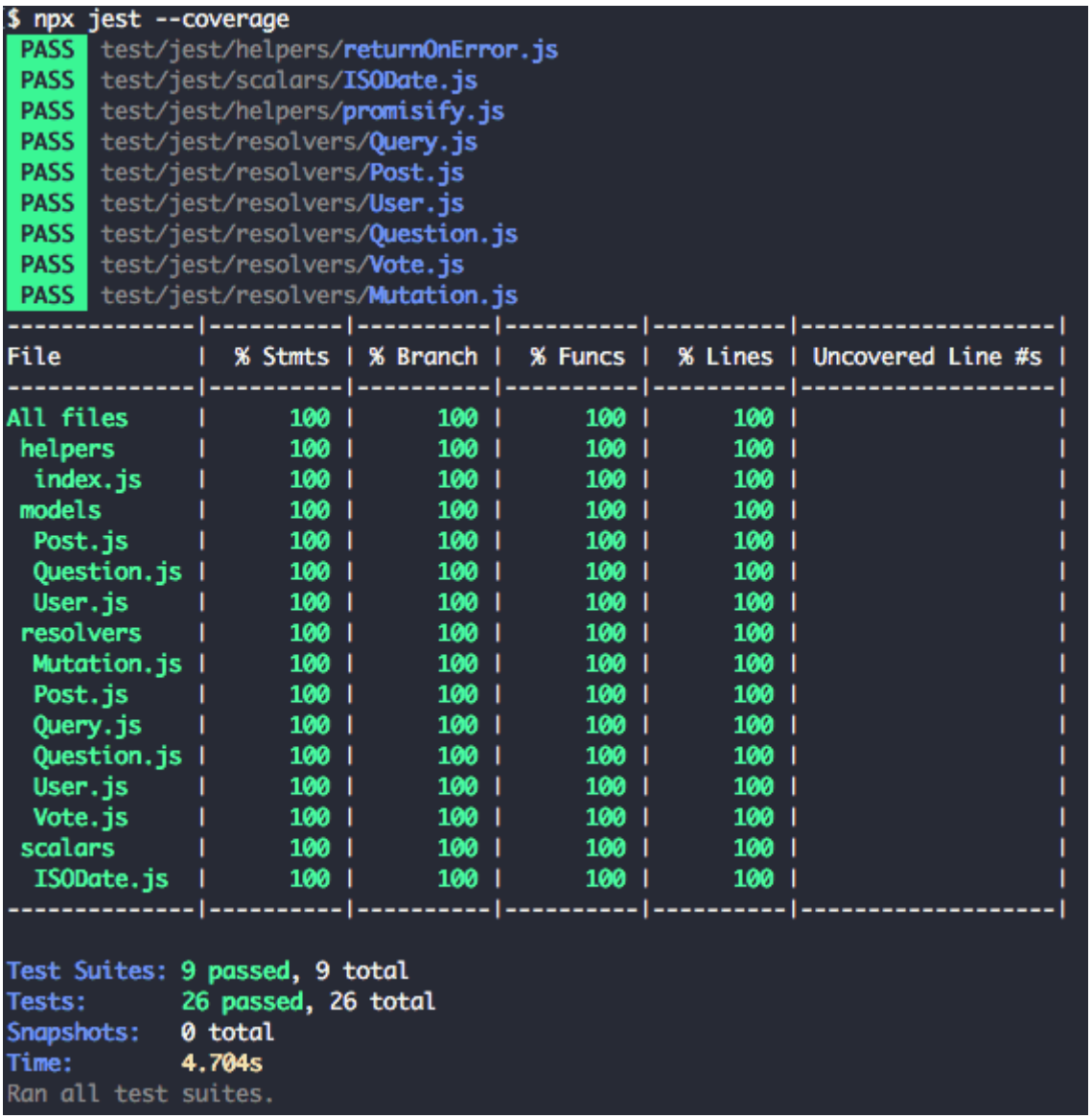

Рисунок 4.2 – Unit тести

На рисунку 4.3 зображено результат запуску інтеграційних тестів третього структурного модулю, який необхідний для взаємодії зі сховищем даних. Інтеграційні тести передбачають симуляцію реальних запитів до сховища з фільтрами групування, брендизації тощо.

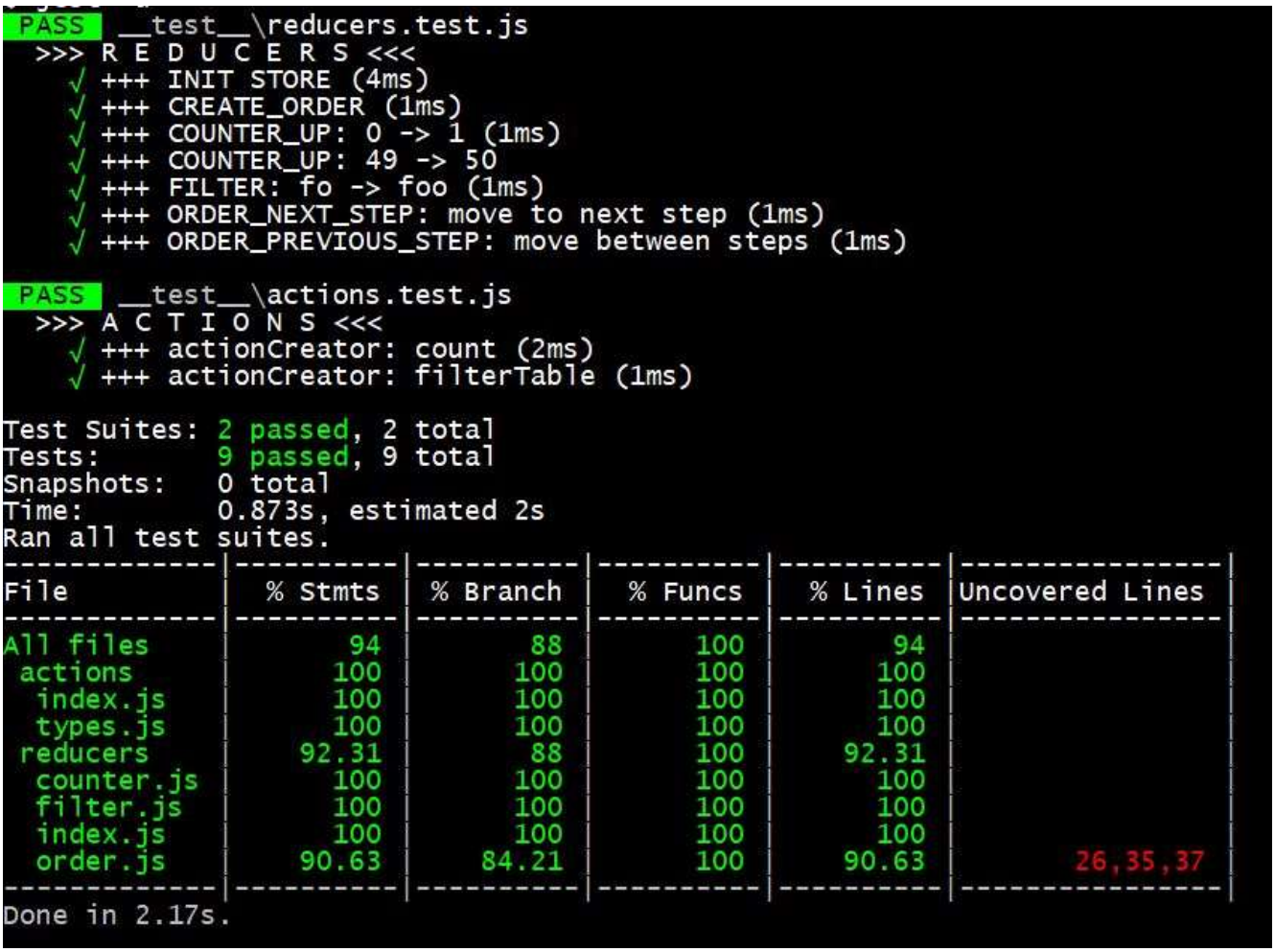

Рисунок 4.3 – Інтеграційні тести

Висновки до четвертого розділу МД

В четвертому розділі магістерської дисертації проведено тестування окремих модулів розробленого забезпечення, а саме розглянуто основні модулі ПЗ, перевірено їх працездатність, стабільність та використання всіх функцій.

Під час написання магістерської дисертації розроблено програмне забезпечення, яке повністю базується на використанні основних постулатів OLAP-технології і може виконувати категоризацію продуктів, прогноз продаж та трендів.

ВИСНОВКИ

У даній магістерській дисертації детально розглянуто особливості використання OLAP-систем та концепції Data Mining для виконання аналізу даних, а також створено програмне забезпечення для інтерактивного аналізу даних з передовими рішенням, завдяки яким відбуваються прискорені аналітичні операції.

У першому розділі магістерської дисертації наведено загальні відомості про OLAP-системи, наведено основні вимоги до систем з використанням OLAP. Також наведено загальний вигляд багатовимірного кубу, як основної структури даних, яка використана у даній роботі для інтерактивного аналізу даних у веборієнтованих сервісах.

У другому розділі магістерської дисертації наведено загальні відомості класифікацію аналітичних систем, розглянуто технологію Data Mining як частину ринку інформаційних технологій, відповідно, перелічено основні концепції, поняття та ключові особливості технології Data Mining. Також наведено загальний перелік OLAP-систем, їх класифікація та особливості й, відповідно, інтеграцію OLAP та Data Mining.

У третьому розділі магістерської дисертації розглянуто загальні принципи роботи створеного програмного забезпечення, обґрунтовано вибір інструментарію та методів розробки програмного забезпечення. Також проаналізовано загальну структуру, зв’язок розроблених модулів та описано інтерфейс розробленого вебдодатку.

У останньому четвертому розділі магістерської дисертації проведено тестування окремих модулів розробленого забезпечення. А саме розглянуто основні модулі ПЗ, перевірено їх працездатність, стабільність та використання всіх функцій.

У результаті розроблено програмне забезпечення, яке повністю базується на використанні основних постулатів OLAP-технології і може виконувати категоризацію продуктів, прогноз продаж та трендів. Для зручного та зрозумілого користування створено інтуїтивно зрозумілий для користувачів різного рівня інтерфейс.

У розроблений продукт будь-якого направлення завжди можна вносити певні зміни та покращення. Даний продукт можна покращити, наприклад, розширенням кількості продукції, тобто використанням іншого API чи створенням власного для отримання даних по продукції.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Chaudhuri, S., & Dayal, U. (1997). An overview of data warehousing and OLAP technology. ACM Sigmod record, 26(1), 65-74.
2. Kimball, R.; Ross, M. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, 3rd ed.; John Wiley & Sons: 2013. - [Електронний ресурс] – Режим доступу: http://www.gbv.de/dms/tib-ub-hannover/74588184x.pdf
3. Codd's 12 Rules for an RDBMS - [Електронний ресурс] – Режим доступу: https://link.springer.com/content/pdf/bbm%3A978-1-4302-0161-8%2F1.pdf
4. Konikov, A., Kulikova, E., & Stifeeva, O. (2018). Research of the possibilities of application of the Data Warehouse in the construction area. - [Електронний ресурс] – Режим доступу: https://www.researchgate.net/publication/329649410_Research_of_the_possibilities_of_application_of_the_Data_Warehouse_in_the_construction_area
5. Методы и модели анализа данных: OLAP и Data Mining. А.А. Барсегян, М.С. Куприянов, В.В. Степаненко, И.И. Холод, 2004 г. - [Електронний ресурс] – Режим доступу: http://213.230.96.51:8090/files/ebooks/dasturlash/Metody%20i%20modeli%20analiza%20dannykh%20olap%20i%20data%20mining%203642712.pdf
6. Mansmann, S., Neumuth, T., & Scholl, M. H. (2007, September). OLAP technology for business process intelligence: Challenges and solutions. In International Conference on Data Warehousing and Knowledge Discovery - [Електронний ресурс] – Режим доступу: https://www.researchgate.net/publication/220802446_OLAP_Technology_for_Business_Process_Intelligence_Challenges_and_Solutions
7. Бондаренко, А.В. (2012). Метод преобразования реляционных данных в многомерные кубы и построение OLAP-отчетов. Альманах современной науки и образования. - [Електронний ресурс] – Режим доступу: https://www.gramota.net/articles/issn_1993-5552_2012_6_02.pdf

8. Туманов В. Е. Проектирование хранилищ данных для систем бизнес-аналитики: учебное пособие - [Електронний ресурс] – Режим доступу: https://docplayer.com/36465757-Proektirovanie-hranilishch-dannyh-dlya-sistem-biznes-analitiki.html
9. Jiawei Han. OLAP Mining: An Integration of OLAP with Data Mining. - [Електронний ресурс] – Режим доступу: https://link.springer.com/content/pdf/10.1007/978-0-387-35300-5_1.pdf
10. Дюк В.А., Самойленко А.П. Data Mining: учебный курс. - [Електронний ресурс] – Режим доступу: https://www.azstat.org/Kitweb/zipfiles/11337.pdf
11. Барсегян А.А. Куприянов М.С. Степаненко В.В. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP. – [Електронний ресурс] – Режим доступу: http://static.ozone.ru/multimedia/book_file/1005924702.pdf
12. The Python Institute. What is Python? – [Електронний ресурс] – Режим доступу: https://pythoninstitute.org/what-is-python/
13. JuliaLang.org. Why We Created Julia – [Електронний ресурс] – Режим доступу: https://julialang.org/blog/2012/02/why-we-created-julia/