Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Метод застосування бізнес-аналітики для візуалізації фрагментів сховищ даних
(тема)

Виконав:

студент II курсу, групи СПм-22-3

Радченко І.В.
(прізвище, ініціали)

Спеціальність

123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма

Системне програмування
(повна назва освітньої програми)

Керівник: Зав. кафедри. Коваленко А.А.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20___ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Радченко Івану Владиславовичу
(прізвище, ім’я, по батькові)

1. Тема роботи Метод застосування бізнес-аналітики для візуалізації фрагментів сховищ даних

затверджена наказом по університету від “ 01 ” квітня 2024 р. № 257 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 15 червня 2024 р.

3. Вхідні дані до роботи Сети даних myntra_products_catalog.csv, countries.csv, states.csv, операційна система – Windows 11. Сервер баз даних – SQL Server. Середовища розробки – Microsoft SQL Server Management Studio, Talend Open Studio, Power BI Desktopб.

4. Перелік питань, що потрібно опрацювати у роботі
1) аналітичний огляд,
2) розробка системи контролю ходу виконання завдань,
3) програмна реалізація системи та її тестування,
4) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 14

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз літературних джерел-	02.04.24-08.04.24	
2	Вибір та обгрунтування методики дослідження	09.04.24-16.04.24	
3	Вибір інструментальних засобів	17.04.24-22.04.24	
4	Розробка моделей протоколів	23.04.24-06.05.24	
5	Проведення експериментів	07.05.24-23.05.24	
6	Оформлення матеріалів кваліфікаційної роботи	24.05.24-03.06.24	
7	Подання кваліфікаційної роботи керівникові та	04.06.24-07.06.24	
	її попередній захист		
8	Подання кваліфікаційної роботи на рецензу-	08.06.24-12.06.24	
	вання		

Дата видачі завдання 01 квітня 2024 р.

Студент _____
(підпис)

Керівник роботи _____ Зав. кафедри. Коваленко А.А.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 89 с., 29 рис., 11 табл., 3 дод., 27 джерел.

АНАЛІЗ ДАНИХ, DATA WAREHOUSE, BUSINESS INTELLIGENCE, ОПТИМІЗАЦІЯ, СХОВИЩЕ ДАНИХ.

Метою дослідження є підвищення швидкості отримання та аналізу даних шляхом розробки моделі візуалізації даних.

Об’єктом дослідження є процес візуалізації даних.

Предметом дослідження є засоби та методи структуризації, збереження та безпосередньо візуалізації даних.

В кваліфікаційній роботі розглянуто основні підходи до розробки моделі візуалізації інформаційного наповнення спеціалізованої комп’ютерної системи. Наведена структурна схема моделі візуалізації даних, які зберігаються у сховищах даних та розглянуті основні етапи її роботи.

ABSTRACT

Master’s thesis: 89 pages, 29 figures, 11 tables, 3 appendices, 27 sources.

DATA ANALYSIS, DATA WAREHOUSE, BUSINESS INTELLIGENCE, OPTIMIZATION, DATA STORAGE.

The object of research is the process of data visualization.

The subject of research is data visualization models.

The research method is the development of a data visualization model and its study to increase the speed of data retrieval and analysis.

The main approaches to the development of a visualization model of the information content of a specialized computer system were developed in the thesis. The structural diagram of the data visualization model, which is stored in data warehouses and formulates the main stages of its work, is given.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

BI –Business Intelligence

БД – база даних

ВД – відомість документів

ЄСКД– єдина система конструкторської документації

ЄСПД – єдина система програмної документації

КІ – комп’ютерна інженерія

КІТ – комп’ютерні та інформаційні технології

СКБД – система керування базами даних

НДРС – науково-дослідна робота студента

СТЗВО – стандарт закладу вищої освіти

ВСТУП

У сучасному світі обсяг інформації, яка зберігається в комп'ютерних системах, постійно зростає. Однак, сама наявність великих обсягів даних не гарантує їх ефективне використання та аналіз. Інформація повинна бути доступною, зрозумілою і легко інтерпретованою для тих, хто не має спеціалізованого фахового знання.

У зв'язку з цим, розробка методів візуалізації інформаційного наповнення комп'ютерних систем є актуальною задачею. Вона спрямована на поліпшення процесу представлення даних у зрозумілій та доступній формі. Ефективна візуалізація даних дозволяє здійснювати швидкий аналіз, виявляти закономірності та взаємозв'язки між даними, а також приймати обґрунтовані рішення на підставі отриманих результатів.

Подальший розвиток методів візуалізації інформаційного наповнення спеціалізованих комп'ютерних систем є важливим завданням, оскільки це дозволить забезпечити зручний та ефективний доступ до великих обсягів даних, що зберігаються у сховищах даних.

Об’єктом дослідження є процес візуалізації даних.

Предметом дослідження є моделі візуалізації даних.

Метою роботи є підвищення швидкості отримання та аналізу даних шляхом розробки моделі візуалізації даних з використанням методу організації та зберігання даних Data Warehouse та методу візуалізації даних Business Intelligence.

1 ІСТОРІЯ ТА СУЧАСНИЙ СТАН РОЗВИТКУ БІЗНЕС-АНАЛІТИКИ

1.1 Історичні аспекти візуалізації даних

На самому початку візуалізації даних, люди використовували різні методи графічного відображення інформації для поліпшення сприйняття та аналізу. Одним із ранніх прикладів є використання діаграм та графіків для представлення даних. У цей період головним чинником була потреба передати інформацію у вигляді, доступному для розуміння та інтерпретації.

На цьому етапі в історії візуалізації даних важливим було використання графіків для подання різних видів даних. Стародавні картографи використовували графічні зображення для відображення географічних об'єктів, а також використовувались діаграми для ілюстрації числових даних. Наприклад, різні види графіків використовувалися для представлення економічних, соціальних та наукових даних.

Важливою точкою на цьому етапі була роль графічних методів у зрозумінні інформації та передачі знань. Графічні зображення дозволяли швидше та ефективніше сприймати дані, що відігравало ключову роль у виникненні культури візуалізації даних.

У 18-19 століттях вчені та статистики почали використовувати графіки для представлення статистичної інформації. Важливою постаттю цього періоду був британський економіст і статистик Вільям Плейфеєр [1], який використовував графіки для ілюстрації економічних та соціальних явищ. Одним із визначальних моментів стало винайдення інфографіки, яке сприяло розвитку графічних методів у представленні даних. Люди стали використовувати різні види графіків, кругові діаграми, гістограми та інші засоби для передачі інформації у більш зрозумілій формі.

Таким чином, цей період в історії візуалізації даних свідчить про

зростання інтересу до графічних методів та їх важливу роль у представленні складних статистичних даних [1].

Наступним етапом в історії візуалізації даних стала поява механічних обчислювальних пристроїв [1, 2].

З появою механічних обчислювальних пристроїв у кінці 19 століття та на початку 20 століття, відбулося ще одне значуще зрушення в історії візуалізації даних. Технології, такі як калькулятори, дозволили створювати більш складні графічні представлення.

На приклад, у 1920-1930-х роках механічні комп'ютери були використані для створення механічних діаграм, що дозволяли автоматично представляти дані у графічній формі. Це відкрило нові можливості для візуалізації більш широкого спектру даних та роботи з ними.

З іншого боку, важливим кроком стало поширення електронних обчислювальних машин та комп'ютерів у середині 20 століття. Це призвело до з'яви електронних таблиць, першою з яких була VisiCalc, випущена в 1979 році [2]. Ці інновації в комп'ютерних технологіях відкрили нові горизонти для візуалізації даних, зробивши її більш доступною для широкого кола користувачів.

В 1960-1970-х роках виникли перші електронні таблиці, такі як IBM's Fortran IV. Проте революцію у цьому напрямку викликав випуск VisiCalc в 1979 році. Цей програмний продукт став першим електронним аркушем, який значно полегшив введення даних та автоматизував обчислення. Люди тепер могли легко взаємодіяти з даними, користуючись графічним інтерфейсом.

У цей період також почали з'являтися перші графічні пакети для обчислення та візуалізації, такі як MATLAB. Цей інструмент дозволяв користувачам створювати складні графіки та діаграми, що відкрило нові можливості для вивчення та розуміння даних.

З цими інноваціями, електронні засоби обробки даних стали більш доступними та широко використовуваними, що стало кроком у напрямку

ефективнішої візуалізації та розуміння великих обсягів інформації.

З 1980-х років розвиток комп'ютерної техніки та програмного забезпечення зазнав великих змін, що відзначилися появою нових інструментів для візуалізації даних та їх аналізу [2].

В цей період стали поширеними графічні пакети, які дозволяли створювати різноманітні графіки та діаграми для представлення даних. Програми, такі як Microsoft Excel, дозволяли користувачам виготовляти графіки в зручний спосіб, використовуючи електронні таблиці.

Поширення особистих комп'ютерів у домогосподарствах та бізнесі також відіграло важливу роль у візуалізації даних. Люди отримали змогу особисто працювати з даними та створювати графіки без значних зусиль.

В 1990–2000 роках відбулася суттєва трансформація в області візуалізації даних, пов'язана зі зростанням використання Інтернету та розвитком технологій. Цей період визначається новими можливостями в області доступності, інтерактивності та широкого використання візуалізацій.

1.2 Сучасні підходи до візуалізації даних

З поширенням веб-технологій стали доступними нові можливості для створення візуалізацій, які можна легко відображати у веб-браузерах. Це дозволило користувачам взаємодіяти з даними без необхідності встановлення спеціального програмного забезпечення.

На цьому етапі з'явилися нові інструменти для створення інтерактивних візуалізацій. D3.js (Data-Driven Documents) став важливим інструментом для розробки веб-графіків та діаграм. Цей JavaScript-фреймворк дозволяє легко зв'язувати дані з елементами сторінки та створювати складні та динамічні візуалізації [2, 3].

Згаданий період в історії візуалізації даних відзначається значними технологічними та концептуальними досягненнями, які вплинули на розвиток різних методів зберігання та обробки даних. Поява інтерактивних

візуалізацій, оновлюваних в реальному часі, стала кроком у напрямку використання Data Virtualization, де різні джерела даних об'єднуються для створення динамічних та актуальних візуальних представлень [2 – 4].

Розвиток графічних можливостей в браузерах та використання технологій, таких як CSS, HTML5 Canvas та WebGL, вказують на сучасні підходи до веб-візуалізацій, що може включати в себе елементи Data Cube, де дані агрегуються в многовимірних просторах для забезпечення більш широкого розуміння контексту [4].

Розширення обсягів даних та необхідність їхньої ефективної обробки та аналізу вказують на важливість технологій, таких як Hadoop та Spark, що сприяли розвитку Data Lakes [4 – 5]. Ці платформи стали ключовими для обробки великих обсягів інформації та підтримки аналізу даних у режимі реального часу.

Отже, зазначений період в історії візуалізації даних відображає синергію різних технологій та методів, включаючи Data Virtualization, Data Cube, і Data Lakes, які сприяли появі інноваційних та ефективних засобів роботи з інформацією в умовах швидкозмінюваного сучасного бізнес-середовища.

В контексті зазначеного періоду в історії візуалізації даних, варто врахувати внесок систем організації та збереження даних Data Warehouse та Data Marts [4].

Data Warehouse, як ключовий компонент інфраструктури аналізу даних, виникло в середині 1980-х років. Здатність централізованого зберігання та обробки великих обсягів структурованих даних робить його важливим елементом для забезпечення доступності та інтерактивності візуалізацій, що активно використовується в сучасних технологіях візуалізації даних.

Data Marts є логічним продовженням концепції Data Warehouse та розвинулися пізніше, практичною необхідністю для локалізованого зберігання та аналізу даних для конкретних відділів чи груп користувачів. Передбачаючи створення окремих Data Marts для різних бізнес-одиниць, цей

підхід дозволяє забезпечити більшу фокусованість та ефективність аналізу для специфічних потреб окремих сегментів компанії. Такий підхід доповнює та розширює можливості загального Data Warehouse, щоб відповісти на конкретні потреби підрозділів.

Отже, взаємодія Data Warehouse та Data Marts, на тлі розширення використання інших технологій, визначила напрямок розвитку інфраструктури для аналізу та візуалізації даних.

Основні аспекти візуалізації даних в сучасному інформаційному середовищі визначаються методами організації та збереження даних. У вищеописаному контексті важливою є взаємодія Data Warehouse, Data Marts, Data Cube, Data Lakes та Data Virtualization як ключових технологій для забезпечення доступності та оптимізації аналізу даних [3 – 6].

Data Warehouse (Сховище даних) - це централізована система зберігання і управління даними, призначена для аналізу та звітності. Основною метою Data Warehouse є інтеграція даних з різних джерел і їх трансформація в структуровану форму, що спрощує аналітичний доступ і використання [4, 7].

Розглянемо основні характеристики Data Warehouse.

Інтеграція даних, Data Warehouse об'єднує дані з різних джерел, таких як операційні бази даних, зовнішні системи, файлові сховища тощо. Ця інтеграція дозволяє отримати глобальний погляд на дані.

Дані, зібрані з різних джерел, піддаються процесу трансформації, включаючи очищення, фільтрацію, об'єднання та перетворення в структуровану форму, яка підходить для аналітики.

В Data Warehouse дані зберігаються в спеціальному сховищі, яке оптимізоване для аналітики. Це забезпечує високу швидкість доступу до даних та підтримує запити аналітики.

Підтримка історії: Data Warehouse може зберігати історичні дані, дозволяючи аналізувати тенденції та зміни в часі.

Зручний доступ для аналітики: Data Warehouse надає інструменти для

виконання аналітичних запитів, створення звітів та дашбордів. Вони допомагають бізнес-аналітикам та фахівцям здобувати інсайти з даних.

Data Warehouse використовується для впровадження бізнес-аналітики, включаючи OLAP (Online Analytical Processing) для аналізу даних, виявлення тенденцій і розробки стратегій.

Зменшення навантаження на операційні системи: Оскільки Data Warehouse відділяє аналітичні запити від операційних систем, це зменшує навантаження на основні системи та дозволяє їм продовжувати нормальну роботу.

Data Warehouse грає ключову роль у вдосконаленні процесів прийняття бізнес-рішень, сприяє підвищенню продуктивності та конкурентоспроможності компанії. Він допомагає організаціям вибирати найбільш оптимальний напрямок розвитку на основі об'єктивних даних та аналітичних висновків.

Data Lakes (Озеро даних) - це сховище для зберігання великої кількості різноманітних даних, включаючи структуровані, напівструктуровані та неструктуровані дані, без потреби відразу їхньої обробки або трансформації. Це схоже на велике озеро, де дані можуть бути зібрані та зберігатися в їхньому природному вигляді. Озеро даних дозволяє легко розширювати обсяги збережених даних та використовувати їх для подальшого аналізу [4, 7].

Розглянемо основні характеристики Data Lakes.

Складність структури даних: Data Lakes приймають дані незміненими, навіть якщо вони не мають жорсткої структури, і не вимагають трансформації перед зберіганням. Це робить їх ідеальними для зберігання даних з різних джерел.

Data Lakes легко масштабуються, що дозволяє зберігати великі обсяги даних. Вони можуть бути розширені в залежності від потреби.

Можливість аналізу даних в натуральному вигляді: Озеро даних дозволяє аналізувати дані в їхньому природному стані, що корисно для

виявлення нових зв'язків;

Data Lakes можуть приймати дані з різних джерел, включаючи бази даних, сенсори Інтернету речей (IoT), логи серверів та інші.

Data Lakes підтримують обробку великих обсягів даних та можуть бути інтегровані з різними інструментами аналізу даних.

Data Lakes можуть зберігати дані в історичному контексті, дозволяючи аналізувати розвиток даних з часом.

Data Lakes стали популярними завдяки збільшенню обсягу та різноманітності даних, які компанії збирають та обробляють. Вони дозволяють зберігати дані в їхньому первинному вигляді, що сприяє гнучкості та можливості аналізувати їх у майбутньому для отримання цінних інсайтів і прийняття бізнес-рішень.

Data Virtualization (Віртуалізація даних) - це методологія об'єднання та інтеграції даних з різних джерел і представлення їх користувачам як єдиний, об'єднаний доступний ресурс, без фізичного переміщення або копіювання даних. Віртуалізація даних використовується для створення враження, ніби всі дані розташовані в одному сховищі, незалежно від їхнього фактичного розташування або формату [4, 7].

Розглянемо основні характеристики Data Virtualization.

Віртуалізація даних дозволяє об'єднувати дані з різних джерел, таких як бази даних, хмарні сховища, веб-сервіси та інші;

Один доступ до даних: Користувачі отримують зручний доступ до об'єднаних даних через одне з'єднання. Це спрощує процес аналізу і звітності.

Дані не копіюються або не переміщаються, що допомагає зменшити проблеми, пов'язані з дублюванням даних і збереженням їх в актуальному стані.

Віртуалізація даних може включати створення віртуальних моделей даних, які визначають, як дані з різних джерел взаємодіють та інтерпретуються.

Підтримка безпеки і контролю доступу. Data Virtualization може надавати рівні доступу до даних, а також механізми безпеки для забезпечення конфіденційності і цілісності даних.

Швидкість доступу до даних. Віртуалізація даних може забезпечувати швидкий доступ до даних, оскільки користувачі отримують результати запитів без затримок, пов'язаних з переміщенням або копіюванням даних.

Data Virtualization дозволяє організаціям максимально використовувати свої дані і забезпечує гнучкість та швидкість доступу до них. Цей метод особливо корисний у великих організаціях, де дані зберігаються в різних системах і джерелах. Віртуалізація даних сприяє поліпшенню аналізу даних, спрощенню інтеграції та впровадженню бізнес-аналітики.

Data Mart (Сховище даних) – це фрагмент або підмножина Data Warehouse, який містить дані, спрямовані на задоволення потреб певного відділу або групи користувачів в організації. Це вузькоспеціалізована ізольована частина, яка фокусується на певних бізнес-потребах і має структуру, оптимізовану для конкретного типу аналізу або бізнес-діяльності [4, 7].

Розглянемо основні риси Data Mart.

Спрямованість на бізнес-потреби. Data Mart розробляється та організовується з урахуванням конкретних вимог певного підрозділу чи команди в організації. Він створюється для забезпечення оптимального доступу до даних для конкретних викликів бізнесу;

Локалізовані дані. Data Mart містить тільки ті дані, які є необхідними для роботи конкретного відділу або групи користувачів. Це допомагає зменшити обсяг даних та спростити їх розуміння;

Оптимізована структура. Дані в Data Mart можуть бути структуровані та оптимізовані саме для певного типу аналізу або бізнес-процесу. Це забезпечує швидший доступ до даних та покращену ефективність аналітичних операцій.

Децентралізований підхід. Data Mart може бути реалізований як

окремий об'єкт або підсистема, що дозволяє різним групам в організації зосереджуватися на своїх специфічних завданнях без необхідності взаємодії з загальним Data Warehouse.

Підтримка аналізу та прийняття рішень. Data Mart допомагає вирішувати конкретні питання і виконувати аналіз, який є важливим для певної групи користувачів, сприяючи прийняттю бізнес-рішень.

Види Data Mart: Enterprise Data Mart (призначений для цілісного підходу до організації даних для всієї організації), Departmental Data Mart (фокусується на задоволенні конкретних потреб певного відділу), Individual Data Mart (розробляється для задоволення потреб конкретного користувача чи групи користувачів).

Data Mart є ефективним інструментом для вдосконалення доступу до даних та аналітики, дозволяючи організаціям ефективно використовувати інформацію для прийняття бізнес-рішень.

Data Cube (куб даних) – це структура даних, яка дозволяє організовувати і представляти дані у тривимірному (іноді більше) просторі. Це поняття часто використовується в області аналізу даних та бізнес-інтелекту для зручного візуального представлення та аналізу багатовимірних даних [4, 7].

Розглянемо основні характеристики Data Cube.

Багатовимірність. Основна ідея полягає в тому, щоб організовувати дані в багатовимірних просторах. Найчастіше це означає, що дані можуть бути відображені вздовж трьох вимірів: X, Y, і Z (іноді також використовуються додаткові виміри).

Візуалізація. Куб даних надає можливість візуально аналізувати дані у всіх трьох вимірах одночасно. Це може включати графіки, графіки, таблиці, які дозволяють швидко виявляти залежності та патерни в даних.

Олап-аналіз. Куб даних часто використовується в системах OLAP (Online Analytical Processing), де користувач може взаємодіяти з даними, здійснюючи вибірки та агрегації в режимі реального часу.

Типи аналізу. Куб даних дозволяє виконувати різні види аналізу, такі як аналіз по часу, простору, або іншим атрибутам, залежно від характеру даних.

Розглянемо основні типи кубів даних. Розрізаний куб (Sliced Cube), відображає частину куба в одному вимірі, зберігаючи інші виміри. Куб агрегації (Rolled-up Cube), відображає куб на більш високому рівні абстракції шляхом агрегації даних вздовж одного чи більше вимірів. Динамічний куб (Drill-down/up Cube), дозволяє користувачам динамічно збільшувати (Drill-down) чи зменшувати (Drill-up) деталізацію даних.

Використання в бізнесі:

1 бізнес-аналітика, Data Cube використовується для проведення різних видів аналізу даних, що допомагає виявляти тенденції та патерни в бізнес-процесах;

2 прогнозування, Дозволяє здійснювати прогнозування шляхом аналізу даних в тривимірному просторі;

3 олап-системи, використовується в системах OLAP для реалізації різних операцій аналізу.

Data Cube є потужним інструментом для візуалізації та аналізу багатовимірних даних, який знаходить широке застосування в різних галузях, зокрема в бізнес-інтелекті та аналізі даних.

Є різні шляхи для безпосередньої візуалізації даних. Головні з них включають використання Business Intelligence (BI) платформ, розробку власних додатків для візуалізації, а також застосування різних мов програмування для створення власних та індивідуально налаштованих графічних представлень даних.

Бізнес-інтелект (BI) є ключовою областю для візуалізації даних. BI-платформи, такі як Tableau, Microsoft Power BI, та Qlik, надають зручний інтерфейс для створення дашбордів та звітів, які можуть взаємодіяти з різними джерелами даних. Вони відзначаються високою швидкістю розгортання та легкістю використання, спрощуючи процес аналізу для не-

технічних користувачів.

Попереджаючи обмеження готових BI-рішень, деякі організації обирають розробку власних додатків для візуалізації даних. Цей підхід дозволяє забезпечити максимальну гнучкість та індивідуалізацію в графічних представленнях, але вимагає значних технічних знань та ресурсів для розробки та підтримки.

Крім того, можна використовувати різні мови програмування, такі як Python, JavaScript, або R, для створення власних візуалізацій даних [4]. Цей підхід відзначається максимальною гнучкістю та повнотою контролю над графічним представленням, але вимагає від користувача глибоких знань програмування та обробки даних.

Такий варіант особливо актуальний у випадках, коли потрібно створити індивідуальні візуалізації, не обмежуючись стандартними інструментами.

2 ОБҐРУНТУВАННЯ МЕТОДІВ ТА ЗАСОБІВ ДОСЛІДЖЕННЯ

З метою проведення дослідження методів організації та зберігання даних та їхньої подальшої візуалізації, планується розробити спеціалізовану комп'ютерну систему, яка включатиме в себе реляційну базу даних. Ця база даних буде створена як основа для вивчення та аналізу різноманітних методів візуалізації даних. Обрано неіснуючий мультибрендовий магазин як об'єкт дослідження, і саме його інформація буде використана для наповнення цієї бази даних. Такий підхід дозволить систематизувати та вивчити різні аспекти організації даних, а реляційна структура бази даних сприятиме ефективному аналізу та використанню цих даних у майбутньому.

У висвітленому контексті вибір реляційної бази даних (РБД) визначається як стратегічно обґрунтоване рішення для системи зберігання та обробки даних мультибрендового магазину, який функціонує у шести різних країнах [4]. Реляційна база даних виявляється оптимальною вибором завдяки своїй здатності ефективно керувати великою кількістю структурованих даних, що характерні для торговельної діяльності, забезпечуючи нормалізацію та стандартизацію інформації. Це забезпечує єдність та консистентність даних, сприяючи високій точності та достовірності збережених інформаційних ресурсів.

У порівнянні з альтернативами, такими як NoSQL бази даних, реляційні системи видаються більш гнучкими та адаптованими до різноманітних бізнес-потреб, зокрема в умовах глобального магазину, який діє в міжнародному середовищі. Реляційні бази даних сприяють ефективному використанню ресурсів та раціональному управлінню даними завдяки використанню стандартів SQL, що спрощує роботу програмістів та адміністраторів баз даних. Враховуючи необхідність оптимізації використання даних у контексті глобальної торгівлі, реляційна база даних видається стратегічним інструментом для забезпечення ефективності та

стабільності інформаційного середовища мультибрендового магазину.

2.1 Обґрунтування вибору методу організації архітектури зберігання та обробки даних задля їх візуалізації

Задля аналізу та побудови візуалізацій буде використано централізовану систему зберігання і управління даними Data Warehouse.

Вибір Data Warehouse над іншими методами організації та зберігання даних, такими як Data Mart, Data Cube, Data Lakes та Data Virtualization, може бути обґрунтований рядом факторів, особливо при розгляді магазину з реляційною базою даних.

Загальна Інтеграція та Єдність Даних: Data Warehouse надає можливість інтегрувати дані з різних джерел та підрозділів магазину в єдиний, централізований репозитарій. Це дозволяє отримати загальну картину діяльності магазину та легко взаємодіяти зі стандартами зберігання даних.

Оптимізація Запитань та Аналітика: Data Warehouse оптимізований для проведення складних аналітичних операцій. Це дозволяє легше виконувати аналізи продажів, попиту, відстежування тенденцій та інші завдання, які можуть виникнути у сфері роздрібної торгівлі.

Підтримка Історичних Даних: Data Warehouse дозволяє зберігати історичні дані, що дозволяє аналізувати зміни та тенденції в часі. Це корисно для вивчення довготривалих стратегій, ефективності маркетингових кампаній та інших аспектів діяльності.

Ефективне Управління Даними: Data Warehouse надає засоби для ефективного управління даними, включаючи інструменти для підтримки ефективного ETL (екстракція, трансформація, завантаження) процесу та забезпечення цілісності та безпеки даних.

Підтримка Динамічного Аналізу та Запитань Користувача: Data Warehouse забезпечує можливість використовувати гнучкі запити та аналіз "на льоту", що дозволяє користувачам створювати та налаштовувати свої

власні запитання без необхідності попереднього визначення стандартних аналітичних шаблонів. Це важливо для користувачів, які потребують індивідуальний підхід до аналізу даних та можливість отримання конкретної інформації в реальному часі з урахуванням потреб бізнесу, які змінюються.

Хоча інші методи, такі як Data Mart, Data Cube, Data Lakes та Data Virtualization, можуть бути ефективними в певних сценаріях, Data Warehouse, завдяки своїм можливостям інтеграції, аналітики та управління даними, є більш адаптованим для потреб роздрібного магазину з реляційною базою даних.

2.2 Обґрунтування вибору системи управління базами даних та середовища розробки для баз даних

Існує багато різних реляційних СУБД зі своїми перевагами та недоліками.Розглянемо найпопулярніші.

MySQL – це відкрита реляційна СУБД, розповсюджується під GNU General Public License. Вона підтримує SQL-запити та використовується для різноманітних веб-застосунків. MySQL має широку спільноту користувачів і ефективно працює в середовищах з великим обсягом читань [9];

PostgreSQL – це відкрита реляційна СУБД з акцентом на розширену функціональність та стандартизацію. Вона підтримує деякі об'єктно-реляційні та геопросторові можливості, а також забезпечує високий рівень надійності та відновлення даних [10];

Microsoft SQL Server – це комерційна реляційна СУБД, розроблена корпорацією Microsoft. Вона пропонує різні можливості, такі як вбудовані процедури, бізнес-аналітика та інтеграцію з іншими продуктами Microsoft [11];

Oracle Database – це комерційна реляційна СУБД, відомою своєю масштабованістю та можливостями для великих підприємств. Вона має широкі функціональні можливості, включаючи продвинуті опції для обробки транзакцій та аналізу даних [12];

MariaDB – це відкритий форк MySQL, розроблений спільнотою і підтримується MariaDB Corporation. Вона зберегла сумісність з MySQL та пропонує деякі додаткові функції та покращення продуктивності [13].

Для розробки бази даних для мультибрендового магазину було обрано Microsoft SQL Server на основі кількох ключових факторів.

Інтеграція з іншими продуктами Microsoft – SQL Server є продуктом Microsoft, і його використання може бути вигідним в екосистемі, де вже використовуються інші продукти Microsoft, такі як Windows Server, Active Directory, та інші. Інтеграція між цими продуктами може полегшити управління та взаємодію системи.

Бізнес-аналітика та звітність – SQL Server надає розширені можливості для бізнес-аналітики та створення звітів. За допомогою інструментів, таких як SQL Server Reporting Services (SSRS) та SQL Server Analysis Services (SSAS), можна легко створювати звіти та аналізувати дані для прийняття управлінських рішень.

Масштабованість та продуктивність – SQL Server володіє добре розвинутою системою масштабованості та продуктивності. Це особливо важливо для магазинів з великою кількістю транзакцій та широкою аудиторією. SQL Server може ефективно обробляти великі обсяги даних та забезпечити стабільну роботу системи в умовах високого навантаження.

Захист та керування доступом – SQL Server надає різноманітні засоби безпеки для захисту даних, включаючи шифрування, автентифікацію та авторизацію. Керування доступом до бази даних може бути ефективно налаштоване для забезпечення конфіденційності та цілісності даних.

Підтримка кластерів та високої доступності – SQL Server має можливості для роботи в кластері та забезпечення високої доступності. Це важливо для уникнення перерв в роботі магазину та забезпечення неперервності обслуговування клієнтів.

Розширена підтримка мови T-SQL – SQL Server використовує мову запитів T-SQL, яка є потужним інструментом для взаємодії з базою даних. З

багатьма вбудованими функціями та процедурами T-SQL може полегшити розробку та оптимізацію запитів.

Загалом, вибір SQL Server для мультибрендового магазину обумовлений його розширеними можливостями бізнес-аналітики, високою продуктивністю, широкою підтримкою Microsoft та забезпеченням безпеки та доступності системи.

Для розробки та виконання T-SQL запитів, процедур та скриптів існують різні середовища розробки. Розглянемо з найпопулярніших середовищ для роботи з T-SQL.

SQL Server Management Studio (SSMS) – SSMS є офіційним інструментом для розробки та адміністрування SQL Server. Воно надає широкі можливості для написання, виконання та налагодження T-SQL запитів, процедур та скриптів. SSMS є найбільш розповсюдженим середовищем для роботи з SQL Server.

Visual Studio з розширенням для роботи з даними (Data Tools) – Visual Studio, зокрема версії, які мають розширення для роботи з даними (Data Tools), може використовуватися для розробки баз даних, включаючи SQL Server. Воно дозволяє створювати та управляти базами даних, розробляти процедури та зберігати дані, використовуючи різноманітні інструменти.

Azure Data Studio – це легковаге, багатофункціональне середовище для розробки баз даних, яке підтримує різні системи управління базами даних, включаючи SQL Server. Azure Data Studio має багато інструментів для роботи з T-SQL та надає можливості для взаємодії з базами даних в хмарному середовищі.

Query Editors в інтерфейсах односторінкових застосунків (Online Query Editors) – деякі хмарні сервіси та хостинги баз даних надають онлайн редактори запитів, які дозволяють виконувати T-SQL запити безпосередньо в веб-браузері, такі як Azure Data Studio Query Editor, SQL Server Online, або редактори, які входять до складу хмарних послуг баз даних.

Third-Party IDEs та редактори – існують також сторонні інтегровані

середовища розробки (IDEs) та текстові редактори, які підтримують T-SQL. Наприклад, Redgate SQL Prompt, dbForge Studio для SQL Server, або JetBrains DataGrip.

Було обрано SQL Server Management Studio як середовище для розробки баз даних.

Вибір SQL Server Management Studio (SSMS) для розробки та роботи з T-SQL має свої вагомі переваги:

Інтеграція з SQL Server: SSMS розроблено безпосередньо для роботи з SQL Server. Воно повністю інтегроване з цією СУБД, що забезпечує максимальну сумісність та ефективність при роботі з базою даних [11];

Розширені Засоби Розробки: SSMS має розширені засоби для розробки, такі як редактор запитів з можливістю автодоповнення, графічний конструктор запитів, візуальні засоби налагодження (debugging) та інші, які полегшують написання та оптимізацію T-SQL коду [11];

Моніторинг та адміністрування: SSMS дозволяє не лише розробляти SQL-запити, а й проводити моніторинг та адміністрування SQL Server. Ви можете використовувати інструменти для відслідковування використання ресурсів, виконання запитів та налаштування параметрів сервера [11];

Широка функціональність: SSMS надає доступ до багатьох додаткових функцій SQL Server, таких як створення збережених процедур, функцій, переглядів, робота з планами виконання та інші, що робить його потужним інструментом для повноцінної розробки та адміністрування [11];

Спільнота та підтримка. Спільнота користувачів SSMS та SQL Server велика, тому користувачі завжди можуть знайти відповіді на свої питання або отримати поради в Інтернеті. Також існує багато документації та ресурсів для вивчення [11];

Оновлення та Підтримка від Microsoft: SSMS регулярно оновлюється та підтримується Microsoft. Це гарантує сумісність із сучасними версіями SQL Server та виправлення можливих помилок або вразливостей [11].

Загалом, вибір SSMS обґрунтовується його спеціалізацією на роботі з

SQL Server, розширеними можливостями розробки та адміністрування, а також підтримкою та активною спільнотою користувачів.

2.3 Обґрунтування вибору ETL застосунку для розробки додатків для вивантаження, трансформування та завантаження даних

ETL (Extract, Transform, Load) - це процеси витягування даних з одного джерела, їх трансформації та завантаження в інше місце, часто в дата-склад (data warehouse) для аналізу та звітності [4, 14, 16]. Існує багато ETL-інструментів, які використовуються для автоматизації цих операцій. Ось кілька популярних ETL-інструментів:

1 Apache NiFi – це відкритий програмний продукт Apache, який дозволяє автоматизувати процеси ETL. Він має візуальний інтерфейс для конфігурації та моніторингу потоків даних, дозволяючи легко визначати джерела, трансформації та місце завантаження;

2 Talend – це ETL-інструмент з відкритим вихідним кодом, який надає інтегроване середовище для розробки, тестування та виконання процесів ETL. Talend підтримує різні джерела даних і має велику бібліотеку компонентів для трансформації та завантаження;

3 Microsoft SQL Server Integration Services (SSIS) – це інтегрований інструмент для ETL, який входить до складу платформи Microsoft SQL Server. Він має велику кількість вбудованих компонентів для роботи з різними джерелами даних та можливостей для програмування;

4 Informatica PowerCenter від Informatica – це комерційний ETL-інструмент, який надає широкі можливості для витягування, трансформації та завантаження даних. Він підтримує велику кількість джерел та форматів даних;

5 Apache Spark. Хоча Spark не є чисто ETL-інструментом, він широко використовується для обробки великих обсягів даних в реальному часі. Spark може використовуватися для ETL завдань за допомогою PySpark або Spark SQL;

6 Oracle Data Integrator (ODI) – це інтегрований інструмент для ETL в екосистемі Oracle. Він надає різноманітні можливості для роботи з базами даних Oracle та іншими джерелами даних;

7 SAS Data Integration Studio – це ETL-інструмент від SAS, який дозволяє розробляти та виконувати процеси витягування, трансформації та завантаження даних в середовищі SAS.

У рамках даного проекту для реалізації процесів ETL та оптимальної інтеграції з реляційною базою даних магазину, що ґрунтується на SQL Server, найбільш підходять такі ETL-інструменти як Talend, SQL Server Integration Services (SSIS) та Apache NiFi. Кожен з цих інструментів має свої особливості, які сприяють ефективній реалізації завдань витягування, трансформації та завантаження даних.

Talend відзначається високою гнучкістю та масштабованістю, що робить його ідеальним для проектів різного масштабу. Його візуальний інтерфейс дозволяє легко розробляти та підтримувати складні ETL-процеси, а також використовувати широкий спектр підключень до різноманітних джерел даних [14].

SSIS, інтегрований інструмент у складі платформи SQL Server, має високий рівень сумісності з іншими компонентами Microsoft та глибоку інтеграцію з SQL Server. Це робить його ефективним рішенням для проектів, де вже використовуються продукти Microsoft [15].

Apache NiFi славиться своєю простотою в конфігурації та моніторингу потоків даних. Його можливості роботи в режимі реального часу роблять його привабливим для проектів, де важливо оперативно обробляти та завантажувати дані [19].

При порівнянні цих інструментів, Talend виділяється своєю відкритістю, широким спектром підтримуваних джерел даних та досвідом спільноти. Talend видається не лише технічно оптимальним вибором для реалізації процесів ETL в поточному проекті, але й має значущі переваги в аспектах засвоєння та вартості порівняно з SQL Server Integration Services

(SSIS). Простий інтерфейс Talend робить його швидким у освоєнні, спрощуючи розробку та підтримку ETL-процесів. В перспективі короткого терміну, Talend виявляється більш динамічним і ефективним рішенням порівняно з SSIS, що може також сприяти значній економії витрат і ресурсів проекту. Тож для реалізації ETL процесів буде використаний Talend.

2.4 Обґрунтування вибору методу та додатку візуалізації даних

У контексті даного випадку для візуалізації даних краще використовувати підхід, заснований на інструментах Business Intelligence (БІ). Business Intelligence визначається як сукупність методів, процесів та технологій, що спрямовані на збір, обробку, аналіз та візуалізацію даних з метою надання бізнес-користувачам зрозумілих та корисних інформаційних звітів [3 – 8, 16].

Переваги використання BI для візуалізації даних обумовлені кількома ключовими аспектами.

По-перше, BI-інструменти, такі як Tableau, Power BI або Qlik, надають високий рівень інтерактивності та гнучкості при створенні звітів та графіків, що дозволяє користувачам ефективно аналізувати дані.

До того ж, ці інструменти часто володіють інтуїтивним інтерфейсом, спрощуючи процес створення візуалізацій.

Другий аспект полягає у тому, що BI-платформи мають вбудовані механізми для обробки та підготовки даних, включаючи можливості автоматичного з'єднання з різними джерелами даних та автоматизовану обробку агрегацій. Це дозволяє покращити ефективність та точність аналізу.

Таким чином, використання Business Intelligence в даному випадку обґрунтовано його здатністю надавати широкий функціонал для візуалізації даних, забезпечуючи при цьому ефективність та зручність в користуванні для кінцевих користувачів.

4 найпопулярніших BI інструментів:

1 Tableau є відомим інструментом для візуалізації даних, який

дозволяє створювати інтерактивні та динамічні графіки, звіти та дашборди [20];

2 Microsoft Power BI є інструментом для аналізу даних та візуалізації, розробленим Microsoft. Інтегрується з іншими продуктами Microsoft [21];

3 QlikView/Qlik Sense – це інструменти, які спеціалізуються на асоціативному аналізі та інтерактивних візуалізаціях [22];

4 Looker – це інструмент для аналітики даних, який дозволяє створювати та спільно використовувати звіти та дашборди [23].

Power BI буде використаний для візуалізації через декілька ключових причин, наведених нижче.

Інтеграція з екосистемою Microsoft. Power BI від Microsoft тісно інтегрований з іншими продуктами Microsoft, зокрема з SQL Server. Це дозволяє зручно і ефективно обмінюватися даними між SQL Server і Power BI, спрощуючи процес витягування та візуалізації даних [21].

Автоматична взаємодія з Power Query та Power Pivot. Power BI має вбудовані інструменти, такі як Power Query для ефективного витягування та перетворення даних, а також Power Pivot для створення високопродуктивних моделей даних. Ці інструменти легко інтегруються з SQL Server [21].

Спрощена візуалізація та створення дашбордів. Power BI пропонує інтуїтивний інтерфейс для створення візуалізацій та дашбордів. Велика кількість готових візуальних компонентів дозволяє швидко та з легкістю створювати представлення даних, що зрозумілі для користувачів [21].

Підтримка реального часу та оновлення даних. Power BI забезпечує можливість роботи з даними в реальному часі та автоматичне оновлення звітів. Це особливо важливо для бізнесу, де актуальність даних є критичною [21].

Загалом, вибір Power BI для SQL Server обгрунтовується його інтеграцією, легкістю використання та широкими можливостями для створення потужних візуалізацій та дашбордів на основі даних з SQL Server.

3 МОДЕЛЮВАННЯ ВІЗУАЛІЗАЦІЇ ІНФОРМАЦІЙНОГО НАПОВНЕННЯ

3.1 Розроблення бази даних для мультибрендового магазину

Назва магазину ClothesStore. Логічні сутності бази даних: магазин, склад, товари, категорії і підкатегорії товарів, постачальники, виробники, ціна, покупці, знижки.

Бізнес логіка наведена нижче. Виробники товарів — характеризуються ID, власною унікальною назвою.

Категорії — кожна категорія товарів, доступних у ClothStore містить унікальні ID та назву.

Підкатегорії — характеризуються ID, власною унікальною назвою, відноситься за логікою до певної категорії.

Продукти (товари) — об'єкти матеріальної цінності, які виготовлені певним виробником та реалізуються у мережі ClothStore, за відповідну ціну. Характеризується власним ID, унікальною назвою, відноситься до певної підкатегорії продуктів, виготовлений певним виробником, який вказує модель.

Ціна – грошове вираження вартості товару, за яким він реалізовується покупцю.

Знижка – відсоток, на який зменшується ціна товару. Знижка залежить від певного свята протягом року.

Опис сутностей та таблиць:

- Vendors – містить інформацію про назву компанії виробника товару та його країну;
- BusinessUnits – описує ідентифікатор типу (Склад/Магазин) об’єкта торговельної мережі, і вказує чи обраний об’єкт є активним в даний час;
- BusinessUnitAddresses – несе в собі інформацію про адресу складу чи магазину (країна, штат, місто, вулиця);
- BusinessUnitTypes – типи складів/магазинів товарів;

- Categories – таблиця, яка містить інформацію про категорії товару та її предка;
- Genders – таблиця, яка містить інформацію про стать, як категорії до яких належать різні товари, переважно одягу;
- Categories_Genders – є таблицею зв’язку, яка забезпечує відношення багато до багатьох між таблицями Genders та Categories;
- Countries – несе в собі інформацію про країни, в яких працює мережа магазинів;
- States – несе в собі інформацію про назву області або регіону;
- Cities – несе в собі інформацію про назву міста, регіон та країну до якого це місто належить;
- Supplies – несе в собі інформацію про постачальника, об’єкт торгівельної мережі, в який здійснюється замовлення товару, та дату замовлення;
- SupplyDetails – описує деталі замовлення товару. Зберігає в собі дані про товар та його кількість;
- Customers – містить в собі інформацію про покупців - ім’я, прізвище, стать, дату народження, емейл та номер телефону;
- Employees – містить інформацію про працівників - ім’я, прізвище, стать, емейл, номер телефону, посаду та магазин/склад в якому він працює;
- Invoices – містить інформацію про товарно-транспортну накладну, а саме ідентифікатор накладної, дату оформлення накладної, ідентифікатор працівника, який оформив товарно-транспортну накладну;
- InvoiceDetails – містить деталі про товарно-транспортну накладну, а саме ідентифікатор товару, кількість товару;
- Orders – містить інформацію про учасників купівлі-продажу, тобто унікальний номер (ID) покупця і продавця, дату відправлення, очікувана дата отримання, тип оплати і коментар;
- OrderDetails – містить інформацію про деталі замовлення, а саме про ціну замовлення і товару, знижку, якщо така наявна;

- OrderStatuses – містить інформацію про статус замовлення, а також дата призначення замовленню різних статусів;

- PayTypes – таблиця, яка містить інформацію про способи оплати. (карта, готівка, бонуси);

- Products – таблиця, яка містить інформацію про наявні товари в мережі;

- Colors – таблиця, яка містить інформації про кольори товарів, переважно одежі;

- Roles – таблиця, яка містить інформацію про посади працівників;

- Discounts – діскаунт айді, старт дейт, енд дейт, відсоток знижки, список категорій, на які діє знижка у форматі, опис, або причина знижок, статус знижки;

- Shipments – містить інформацію про переправлення товарів між різними складами та магазинами. Наприклад спершу товар постачальники поставляють на склади в Америці, а потім їх везуть в інші країни.

Частину SQL запитів, які створюють вище перелічені таблиці, наведено на рисунку 3.1.

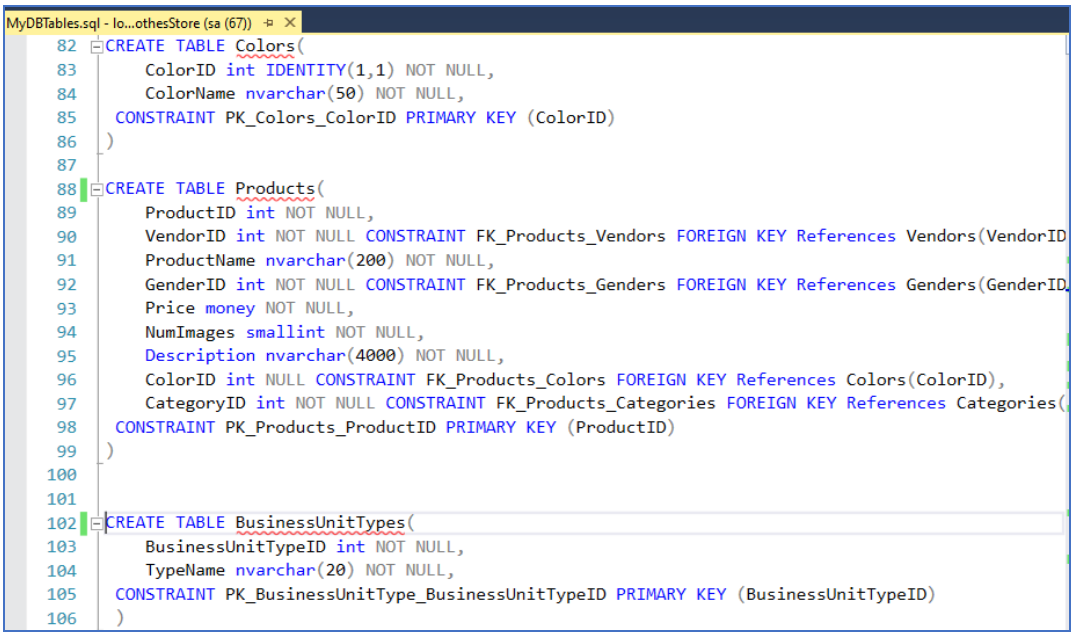

Рисунок 3.1 – SQL запит на створення таблиць

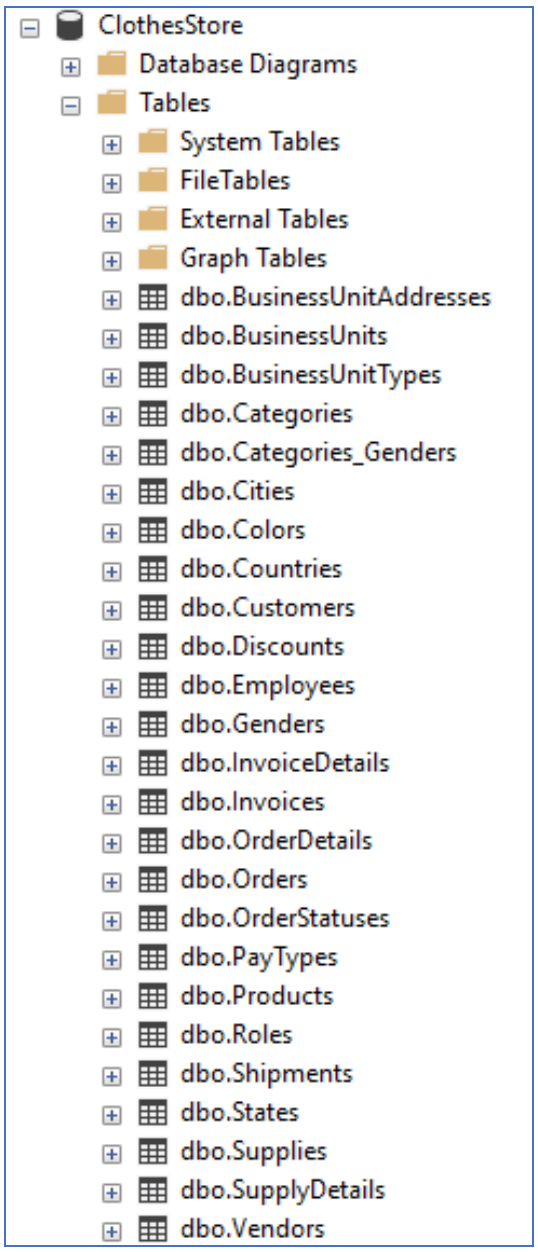

Рисунок 3.2 – Список створених таблиць для бази даних ClothesStore

На рисунку 3.2 наведено список створених таблиць.

3.2 Генерація даних для мультибрендового магазину

На наступному етапі розробки бази даних для неіснуючого магазину визначається важливість завантаження реалістичного та репрезентативного датасету товарів. Наявність вірогідних даних про товари є ключовою умовою для створення точної та деталізованої моделі візуалізації, яка відображатиме

реальний асортимент продуктів у магазині. З цією метою обрано використання відомої та надійної платформи Kaggle для завантаження датасету товарів.

Найбільший акцент робиться на датасеті товарів, оскільки він буде визначальним для аналізу та візуалізації. Забезпечення якості цього датасету є важливим елементом створення автентичної моделі магазину. Дані, завантажені з Kaggle, відображатимуть реальний асортимент товарів, їхні характеристики та атрибути, що сприятиме більш точному відтворенню візуалізації.

Додатково, інші датасети, такі як інформація про країни, регіони/області та міста, будуть завантажено з інтернету для створення повноцінного контексту та можливості аналізу продажів за різними географічними параметрами.

Усі інші дані, що не мають прямого відображення на реальному світі, будуть згенеровані за допомогою бібліотек випадкової генерації даних, зокрема, на мові програмування Python та бібліотек Pandas [24, 25] (для роботи з сетами даних) та Faker (для генерації випадкових даних) [24, 25]. Також частина даних буде згенерована із застосуванням процедур T-SQL. Це дозволить забезпечити достатню кількість даних для тестування та визначення функціональності бази даних, при цьому не порушуючи конфіденційність реальних даних та дотримуючись етичних стандартів у використанні інформації.

Список таблиць, які будуть заповнені реальними даними: products, countries, states, cities.

Частину датасету таблиць Products зображено на рисунку 3.3.

	A	B	C	D	E	F	G	H
1	ProductID	ProductName	ProductBrand	Gender	Price (INR)	NumImag	Description	PrimaryColor
2	10017413	DKNY Unisex Black & Grey Printed Medium Trolley Bag	DKNY	Unisex	11745	7	Black and grey p	Black
3	10016283	EthnoVogue Women Beige & Grey Made to Measure Custom Made	EthnoVogue	Women	5810	7	Beige & Grey m	Beige
4	10009781	SPYKAR Women Pink Alexa Super Skinny Fit High-Rise Clean Look S	SPYKAR	Women	899	7	Pink coloured w	Pink
5	10015921	Raymond Men Blue Self-Design Single-Breasted Bandhgala Suit	Raymond	Men	5599	5	Blue self-desigr	Blue
6	10017833	Parx Men Brown & Off-White Slim Fit Printed Casual Shirt	Parx	Men	759	5	Brown and off-v	White
7	10014361	SHOWOFF Men Brown Solid Slim Fit Regular Shorts	SHOWOFF	Men	791	5	Brown solid low	Brown
8	10017869	Parx Men Blue Slim Fit Checked Casual Shirt	Parx	Men	719	5	Blue checked ca	Blue
9	10009695	SPYKAR Women Burgundy Alexa Super Skinny Fit High-Rise Clean L	SPYKAR	Women	899	7	Burgundy colou	Burgundy
10	10000571	Parx Men Brown Tapered Fit Solid Regular Trousers	Parx	Men	664	5	Brown solid reg	Red
11	10017421	DKNY Unisex Black Large Trolley Bag	DKNY	Unisex	17360	5	Black solid large	Black

Рисунок 3.3 – Датасет Products

Частину датасету Countries зображено на рисунку 3.4.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
	id	name	iso3	iso2	numeric_c	phone_co	capital	currency	currency_	currency_	tld	native	region	subregion	nationality	timezones	latitude	longitude
	1	Afghanist	AFG	AF	004	93	Kabul	AFN	Afghan af	Ш‹	.af	Ш§Щ‡Ш	Asia	Southern Asia	Afghan	[{zoneNar	33	65
	2	Aland Isla	ALA	AX	248	340	Mariehamr	EUR	Euro	в‚¬	.ax	Г…land	Europe	Northern Euro	Aland Island	[{zoneNar	60.11667	19.9
	3	Albania	ALB	AL	008	355	Tirana	ALL	Albanian l	Lek	.al	ShqipГ«ri	Europe	Southern Euro	Albanian	[{zoneNar	41	20
	4	Algeria	DZA	DZ	012	213	Algiers	DZD	Algerian d	ШЇШ¬	.dz	Ш§Щ„Ш¬	Africa	Northern Afric	Algerian	[{zoneNar	28	3
	5	American	ASM	AS	016	-683	Pago Pago	USD	US Dollar	$	.as	American	Oceania	Polynesia	American Sam	[{zoneNar	-14.3333	-170
	6	Andorra	AND	AD	020	376	Andorra la	EUR	Euro	в‚¬	.ad	Andorra	Europe	Southern Euro	Andorran	[{zoneNar	42.5	1.5
	7	Angola	AGO	AO	024	244	Luanda	AOA	Angolan k	Kz	.ao	Angola	Africa	Middle Africa	Angolan	[{zoneNar	-12.5	18.5
	8	Anguilla	AIA	AI	660	-263	The Valley	XCD	East Carib	$	.ai	Anguilla	Americas	Caribbean	Anguillan	[{zoneNar	18.25	-63.1667
	9	Antarctica	ATA	AQ	010	672		AAD	Antarctica	$	.aq	Antarctica	Polar		Antarctic	[{zoneNar	-74.65	4.48
	10	Antigua A	ATG	AG	028	-267	St. John's	XCD	Eastern Ca	$	.ag	Antigua a	Americas	Caribbean	Antiguan or Ba	[{zoneNar	17.05	-61.8
	11	Argentina	ARG	AR	032	54	Buenos Ai	ARS	Argentine	$	.ar	Argentina	Americas	South America	Argentine	[{zoneNar	-34	-64
	12	Armenia	ARM	AM	051	374	Yerevan	AMD	Armenian	ЦЏ	.am	ХЪХЎХµХ	Asia	Western Asia	Armenian	[{zoneNar	40	45
	13	Aruba	ABW	AW	533	297	Oranjesta	AWG	Aruban fl	Ж’	.aw	Aruba	Americas	Caribbean	Aruban	[{zoneNar	12.5	-69.9667
	14	Australia	AUS	AU	036	61	Canberra	AUD	Australian	$	.au	Australia	Oceania	Australia and N	Australian	[{zoneNar	-27	133
	15	Austria	AUT	AT	040	43	Vienna	EUR	Euro	в‚¬	.at	Г–sterreic	Europe	Western Europ	Austrian	[{zoneNar	47.33333	13.33333

Рисунок 3.4 – Датасет Countries

Датасети States та Cities мають аналогічну структуру, що і Countries.

В рамках оптимізації та стандартизації даних, необхідно провести процес очищення цих чотирьох датасетів. Ця процедура включатиме в себе вилучення зайвих колонок, які не несуть значущої інформації або дублюють функціональність інших стовпців. Додатково, буде здійснено перейменування необхідних колонок відповідно до встановленого формату, що дозволить забезпечити єдність та зрозумілість структури даних в усьому проекті, а також буде проведено видалення зайвих рядків, які не будуть використані в подальших етапах розробки.

Також таблиці, які пов’язані з таблицею Products та будуть заповнені існуючими даними, виходячи з датасету Products (Categories, Colors, Genders,

Vendors).

У процесі створення категорій для товарів, які відзначаються відсутністю окремої колонки категорій у датасеті, буде використано мови програмування Python та T-SQL для проведення аналізу назв товарів. Цей аналіз дозволив автоматизовано визначити та призначити кожному товару відповідну категорію на основі ключових слів, семантики та характеристик назв.

Всі інші таблиці будуть наповнені випадково згенерованими даними, створеними за допомогою скриптів, розроблених мовою програмування Python, та процедур генерації даних, виконаних на T-SQL.

Вибір мови програмування Python для генерації датасетів обґрунтовується кількома ключовими аспектами, що сприяють швидкому та ефективному написанню скриптів генерації даних.

По-перше, Python відзначається простим та лаконічним синтаксисом, що робить його доступним для широкого кола користувачів. Мова дозволяє швидко виражати ідеї та реалізовувати концепції, що важливо кули в пріоритеті продуктивність та ефективність в процесі генерації даних [24, 25].

По-друге, наявність розгалужень та великої кількості готових бібліотек для обробки даних (таких як Pandas, NumPy, та Faker) робить Python ідеальним вибором для завдань з генерації невеликих датасетів. Ці бібліотеки спрощують операції з обробки, трансформації та заповнення даними, що підвищує ефективність розробки та дозволяє швидко втілювати концепції [24, 25].

3.3 Розроблення Data Warehouse бази даних для мультибрендового магазину

Реляційні бази даних (RDBS) та Data Warehouse бази даних (DWH) представляють собою два різних типи систем зберігання та управління даними з різними характеристиками та призначеннями. Розглянемо основні різниці між ними.

Орієнтація на завдання [4, 26]. RDBS зазвичай використовуються для операційного забезпечення бізнес-процесів, обробки транзакцій та щоденного функціонування компанії. Data Warehouse спеціально призначені для зберігання та обробки великого обсягу даних для виконання аналітичних запитів та створення звітів.

Типи даних [4, 26]. RDBS зберігають структуровані дані та використовують табличну структуру для організації інформації. Data Warehouse можуть обробляти різноманітні дані, включаючи структуровані, напівструктуровані та неструктуровані дані, з метою підтримки аналітики.

Обсяг та ретенція даних [4, 26].RDBS зазвичай призначені для обробки та зберігання обмеженого обсягу даних. Data Warehouse розраховані на роботу з великим обсягом історичних даних, що дозволяє здійснювати аналіз та прогнозування на основі довготривалих трендів.

Архітектура та оптимізація запитів [4, 26]. RDBS орієнтовані на оптимізацію швидкості виконання транзакцій та операційної ефективності. Data Warehouse орієнтовані на оптимізацію аналітичних запитів та звітів, що може включати довгі агрегації або велику кількість даних.

Моделі даних [4, 26]. RDBS зазвичай використовують модель даних, що базується на нормалізації для уникнення дублювання даних та підтримки транзакцій. Data Warehouse використовують модель з денормалізацією для підвищення продуктивності аналітичних операцій та полегшення роботи з великими наборами даних.

Часові характеристики [4, 2]. RDBS орієнтовані на операційний час та потребують негайного доступу до даних для забезпечення транзакцій. Data Warehouse часто працюють із затримкою часу, оскільки дані аналізуються та агрегуються попередньо для підтримки аналітичних потреб.

Структура бази даних Data Warehouse включає два основних типи таблиць. Facts та Dimensions. Fact таблиця є агрегованим сховищем для числових величин, які представляють собою ключові факти аналізу [4].

Dimension таблиця визначається як для зберігання текстових значень та бізнес-термінів, що надають контекст та характеристики даних [4].

Для надійного дизайну таблиць фактів важливо чітко визначити бізнес-процес, який вони будуть описувати. Прикладами таких процесів можуть бути: продажі, фінансові транзакції, виробничі показники. Fact таблиця спрямована на зберігання числових значень та ключів, які посилаються на таблиці Dimension, тоді як Dimension таблиці в основному містять текстові значення та бізнес-терміни, призначені для забезпечення зрозумілості та ефективної інтерпретації даних кінцевими користувачами.

При процесі дизайну Data Warehouse також варто враховувати необхідність денормалізації даних, що дозволяє зберігати деякі повторювані дані безпосередньо в Dimension таблицях. Це може полегшить запити та забезпечить швидший доступ до даних у випадках, коли великі обсяги інформації часто використовуються у звітах та аналізі. Пріоритетом системи Data Warehouse є швидкість та легкість отримання даних.

Процес реєстрації продажу одиниць товарів є найбільш значимим етапом [4]. Ця процедура виступає ключовим елементом для ефективного аналізу та візуалізації основних бізнес-метрик. Зафіксовані дані про продажі надають можливість подальшого вивчення та спостереження за різними аспектами фінансової діяльності компанії.

Передусім, цей процес дозволяє відстежувати прибутковість на рівні окремих товарів. Шляхом аналізу обсягів продажів різних товарних позицій можна визначити ефективність та популярність кожного товару. Також може проводитися порівняльний аналіз прибутків від різних категорій товарів, що є важливим для стратегічного планування асортименту.

У системі баз даних для реєстрації продажів одиниць товарів використовуються дві таблиці: Orders та OrderDetails. Fact таблиця, яка фіксує інформацію про продані одиниці товарів, отримує назву SalesFact. Ця таблиця ідентифікує окремі одиниці продажів продукції та містить розширену інформацію стосовно замовлень. Перейменування таблиці є

необхідним етапом з метою полегшення сприйняття вмісту таблиці кінцевими користувачами, що сприяє збільшенню зрозумілості та зручності взаємодії з базою даних.

Наступним етапом у проектуванні системи Data Warehouse є вибір схеми: Star або Snowflake.

Star схема є структурою бази даних, де факт таблиця (у данному випадку, SalesFact) з'єднується безпосередньо з декількома Dimension таблицями, утворюючи центральний вузол, схожий на зірку. Кожна Dimension таблиця представляє собою різні аспекти часу, продуктів чи регіонів, і надає конкретний контекст для аналізу фактів [4].

Snowflake схема, навпаки, є розширенням Star схеми, де Dimension таблиці додатково нормалізуються, розбиваючи їх на більш деталізовані підтаблиці. Це призводить до більшої нормалізації даних, але може збільшити кількість з'єднань та складність запитів [4].

Обрано Star схему для структури Data Warehouse бази даних. Цей вибір зумовлений необхідністю оптимізації швидкодії при аналізі продажів одиниць товарів. У Star схемі факт таблиця SalesFact з'єднується безпосередньо з Dimension таблицями, створюючи ефективний центр для аналізу даних. Така структура дозволяє здійснювати швидкі та прості запити, оскільки немає значущого збільшення кількості з'єднань.

Star схема також полегшує розуміння та використання бази даних кінцевими користувачами. Її простота та зрозумілість дозволяють здійснювати швидкий доступ до необхідної інформації та аналізувати ключові аспекти продажів. У контексті обраної задачі, де основний фокус на візуалізації бізнес-метрик, Star схема є оптимальним варіантом, забезпечуючи зручність аналітичних запитів та високу продуктивність системи.

Також в усіх таблицях цієї бази даних буде використаний сурогатний ключ.

Сурогатний ключ – це унікальний ідентифікатор, який

використовується для унікальної ідентифікації кожного запису в базі даних [4]. Цей ключ не має природного значення і створюється самою системою бази даних для забезпечення унікальності та ефективного управління даними. Сурогатний ключ зазвичай використовується там, де немає однозначного чи стійкого до змін набору полів для використання як первинного ключа, і він допомагає покращити швидкодію операцій пошуку та з’єднань в базі даних.

Далі буде створено Dimension таблиці, такі як DateDim, StoresDim, SellersDim, ProductsDim, DiscountsDim, PayTypesDim, CustomersDim та OrderStatusesDim, призначені для деталізації здійснених замовлень в системі Data Warehouse. Кожна з цих таблиць володіє відповідною інформацією, що описує характеристики зазначених аспектів замовлень, такі як дата, магазин, продавець, товар, знижка, тип оплати, клієнт та статус замовлення.

Ці Dimension таблиці взаємодіють із факт таблицею SalesFact за допомогою Foreign Key, який включений до SalesFact та посилається на відповідні ключі в кожній Dimension таблиці. Такий підхід формує Star схему, що дозволяє ефективно організовувати та використовувати дані для подальшого аналізу.

Таким чином, база даних Data Warehouse для магазину одягу, буде сформована із сутностей SalesFact, DateDim, StoresDim, SellersDim, ProductsDim, DiscountsDim, PayTypesDim, CustomersDim, та OrderStatusesDim, та отримає назву ClothesStoreDW. Ця база даних включає таблиці розглянуті нижче.

SalesFact містить факти продажу одиниці товару, фактичну ціну продажу, звичайну ціну товару (без знижок), а також, ідентифікатори дати, магазину, продавця, продукту, клієнта, способу оплати, статусу замовлення та знижки.

DateDim містить атрибути дати, такі як день, місяць, рік, та інші. Використовується для подробичної сегментації продажів за часом. Оскільки ця таблиця призначена лише для зберігання дат і повинна охоплювати весь календарний період з 2000 по 2019 рік, адже дані щодо продажів були

згенеровані за період з 2016 по 2019 рік, для автоматизованого заповнення цієї таблиці буде розроблена процедура генерації даних на мові запитів T-SQL.

StoresDim містить інформацію про магазини, включаючи їхній ідентифікатор, назву, місцезнаходження, статус (працює/не працює), тип (магазин/склад) та широту і довготу місцезнаходження. Таблиця була названа Stores замість BusinessUnits, як в первинній базі даних магазину, задля надання більш зрозумілого контексту вмісту таблиці кінцевим користувачам.

SellersDim містить дані про продавців, такі як їхній ідентифікатор, ім'я, позицію, імейл та стать. Таблиця була названа Sellers замість Employees, як в первинній базі даних магазину, задля надання більш зрозумілого контексту вмісту таблиці кінцевим користувачам.

ProductsDim містить дані про продукти, такі як ідентифікатор, назва, категорія, ціна, бренд, опис, колір.

DiscountsDim містить дані про доступні знижки, включаючи ідентифікатор, опис, розмір знижки, дата початку та завершення дії знижки, тривалість знижки в днях.

PayTypesDim містить дані про типи оплати, такі як готівка та кредитна карта.

CustomersDim містить дані про клієнтів, включаючи ідентифікатор, ім'я, імейл, стать та день народження.

OrderStatusesDim містить дані про статуси замовлень, такі як "в обробці", "доставлено", "відхилено".

Ці таблиці спільно взаємодіють, утворюючи структуровану базу даних для зручного аналізу та візуалізації різноманітних аспектів діяльності магазину одягу.

Схема бази даних Data Warehouse зображена на рисунку 3.5.

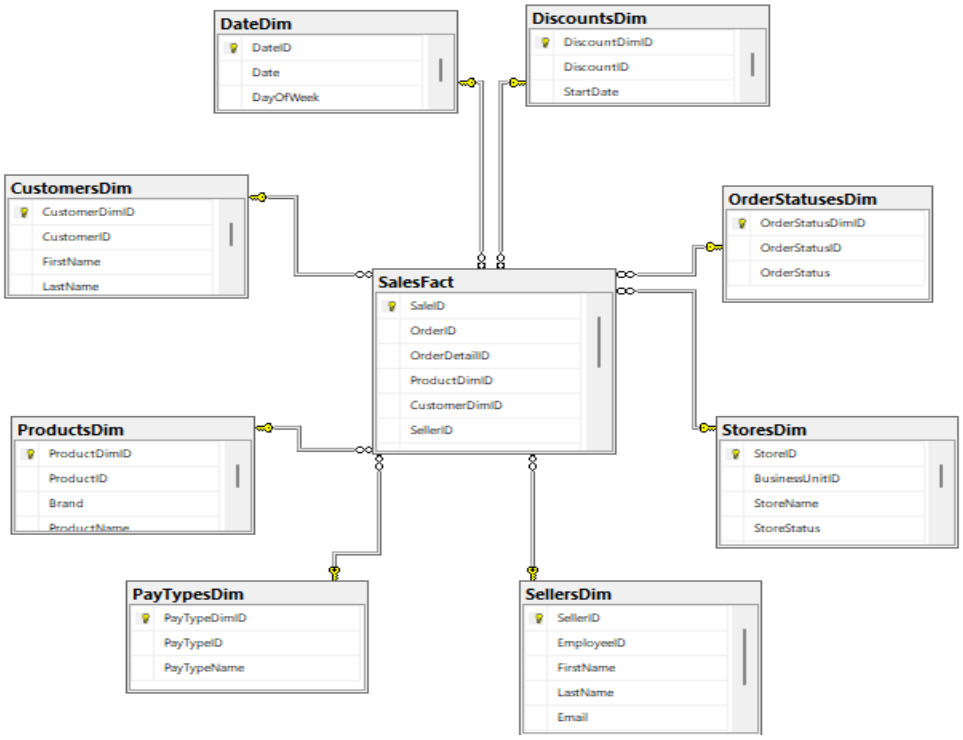

Рисунок 3.5 – Схема бази даних Data Warehouse

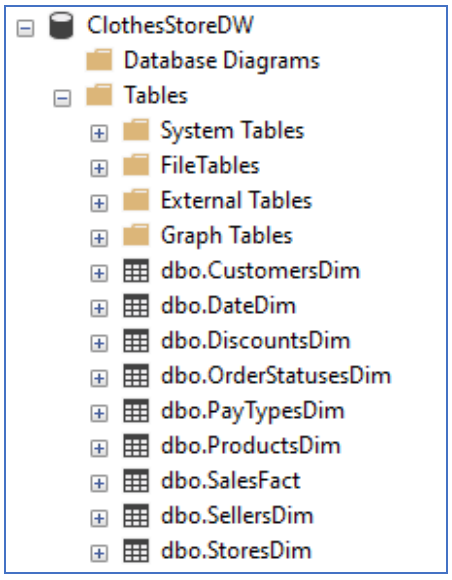

Рисунок 3.6 – Таблиці в Data Warehouse базі даних

На рисунку 3.6 зображено створені таблиці в Data Warehouse базі даних.

3.4 Розроблення програм для вивантаження, трансформування та завантаження даних в систему Data Warehouse

У початковому етапі проектування ETL процесів необхідно визначити джерела та призначення даних, що використовуються для заповнення відповідних таблиць Data Warehouse [4, 14].

Дані, призначені для таблиці ProductsDim, екстрагуються з наступних таблиць магазину: Vendors, Genders, Colors та Categories.

Щодо таблиці SellerdDim, необхідні дані знаходяться в таблицях Employees та Roles. Дані для таблиці StoresDim походять від таблиць:

- BusinessUnits;
- BusinessUnitTypes;
- BusinessUnitAddresses;
- Countries, States;
- Cities.

Також передбачається завантаження даних без подальших об'єднань для наступних таблиць:

- Customers (в таблицю CustomersDim);
- Discounts (в таблицю DiscountsDim);
- OrdersStatuses (в таблицю OrderStatusesDim);
- PayTypes (в таблицю PayTypesDim).

Ключові дані для таблиці SalesFact будуть походити з таблиць OrderDetails та Orders, і вони будуть включати посилання на всі Dimension таблиці для відповідності необхідним показникам та характеристикам у Data Warehouse.

На наступному етапі проекту передбачено розробку ETL-програм за допомогою інструменту Talend Open Studio.

Ці програми призначені для вивантаження даних з бази даних магазину, об'єднання відповідних таблиць та здійснення незначних трансформацій. Серед цих трансформацій входять операції, такі як заміна значень NULL на "Unknown", перейменування колонок, а також

впровадяться зміни, спрямовані на заміну технічних термінів бізнес-термінами, з метою полегшення сприйняття інформації кінцевими користувачами.

Графічна структура ETL програми SalesFactSourceToDWH зображена на рисунку 3.7.

На рисунку 3.7 цифрами позначено:

1 – підключення до обох баз даних;

2 – скачування даних з таблиць Orders та OrderDetails;

3 – об’єднання таблиць Orders та OrderDetails;

4 – скачування даних з Dimension таблиць;

5 – об’єднання даних SalesFact з ключами Dimension таблиць;

6 – завантаження отриманого дата сету в базу даних ClothesStoreDW;

7 – здійснення транзакцій в базах даних та закриття підключень.

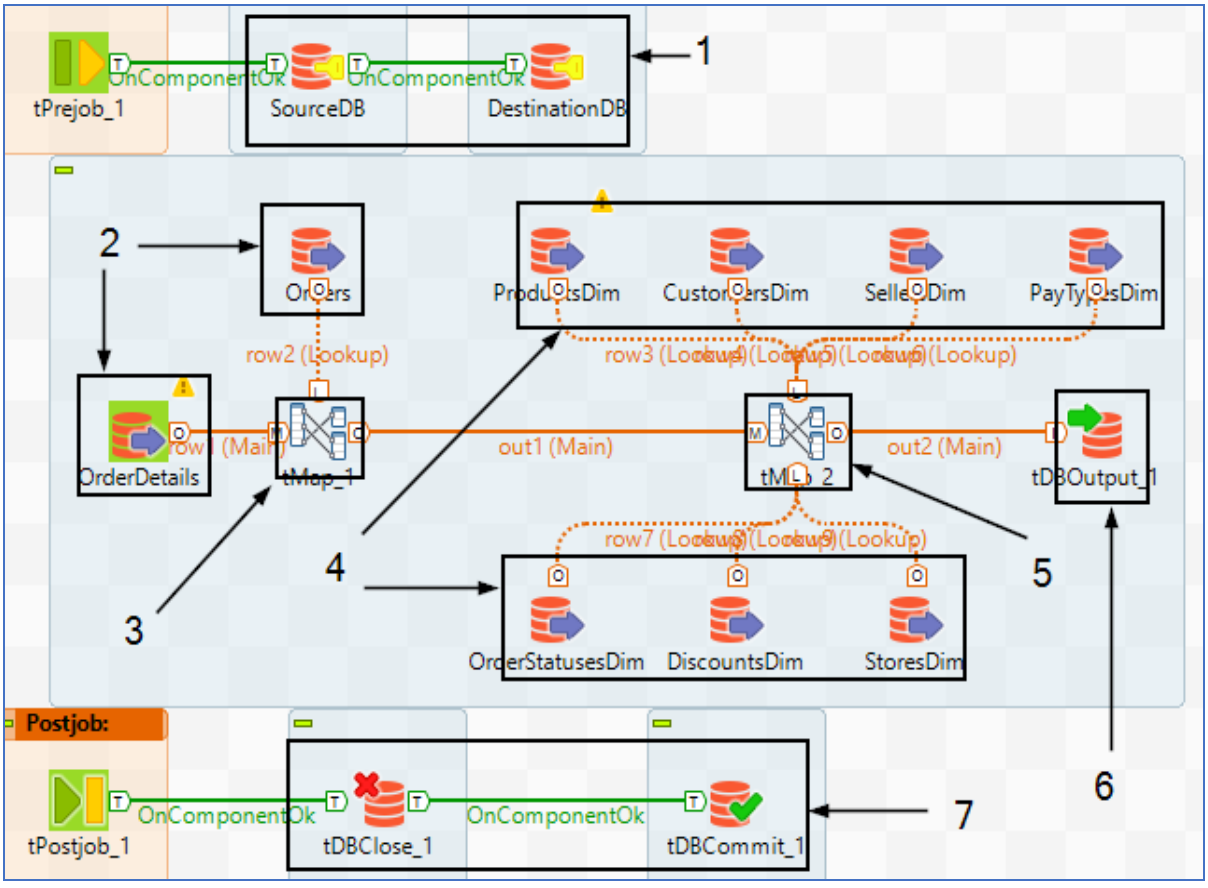

Рисунок 3.7 – ETL програма SalesFactSourceToDWH

Список усіх розроблених ETL програм зображено на рисунку 3.8.

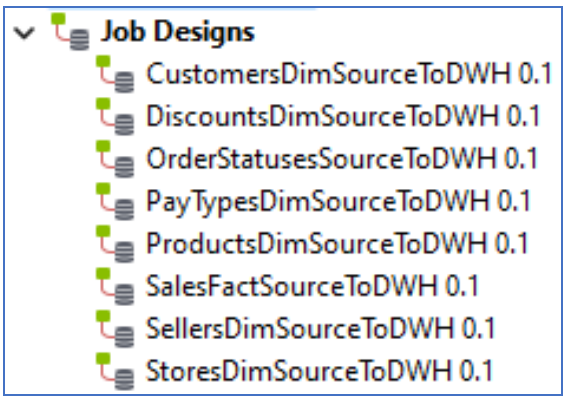

Рисунок 3.8 – Список ETL програм

Після розробки та реалізації ETL-процесів з використанням інструменту Talend Open Studio, трансформовані дані ефективно вивантажено в Data Warehouse базу даних. Цей процес включав в себе взаємодію з різними таблицями магазину, об'єднання необхідної інформації та застосування різноманітних трансформацій, що покращили якість та структуру даних. У результаті завершення ETL-процесів, дані стали готовими для подальшого використання у візуалізаціях та аналітичних звітах, надаючи користувачам можливість ефективного вивчення та розуміння накопиченої інформації.

3.5 Розроблення графічних звітів

Побудова ефективних BI-дашбордів (графічних звітів) базується на кількох ключових концептах, що визначають їхню функціональність та корисність [2 – 8]:

- ключові показники ефективності (KPIs), ідентифікація та відображення основних показників, які відображають стан бізнесу та дозволяють здійснювати стратегічне прийняття рішень;

- візуалізація даних, використання графіків, діаграм, карт та інших візуальних елементів для зрозумілого та ефективного представлення інформації;

- інтерактивність, забезпечення можливості взаємодії користувача з

дашбордом, дозволяючи вибирати параметри, фільтрувати дані та отримувати докладніше розгорнуту інформацію;

- групування за тематикою, організація даних на дашборді за логічними групами або темами для кращого сприйняття та розуміння;

- збалансованість, уникнення перенасиченості інформацією та фокус на важливих показниках без зайвого завантаження.

Буде створено дашборд, який буде мати сторінки, кожна з яких націлена на відображення конкретних аспектів бізнес-аналітики:

- сторінка "Revenue Summary" (Зведення прибутку) – це головна сторінка призначена для надання огляду ключових показників прибутку впродовж часу. На ній будуть відображені загальні показники прибутку, чистий прибутку, кількість замовлень, замовлення за категоріями товарів, а також визначено десять найбільш прибуткових товарів;

- сторінка "Customer Details" (Деталі клієнтів) – ця сторінка буде включати інформацію про загальну кількість клієнтів, середній прибуток від одного клієнта, а також перелік топ-клієнтів. Це допоможе виокремити ключових учасників та зрозуміти характеристики споживачів у контексті прибутковості;

- сторінка "Product Details" (Деталі продуктів). На цій сторінці будуть відображені показники продажів та чистого прибутку, а також забезпечено можливість фільтрації за конкретним продуктом. Це дозволить детально проаналізувати вплив окремих продуктів на фінансові показники;

- сторінка "Map" (Мапа) – ця сторінка буде відображати географічну мапу з позначенням кількості замовлень у кожній країні. Це надасть візуальний контекст розподілу активності замовлень по різних країнах.

Грубий макет усіх сторінок дашборду зображено на рисунках 3.9 –3.12.

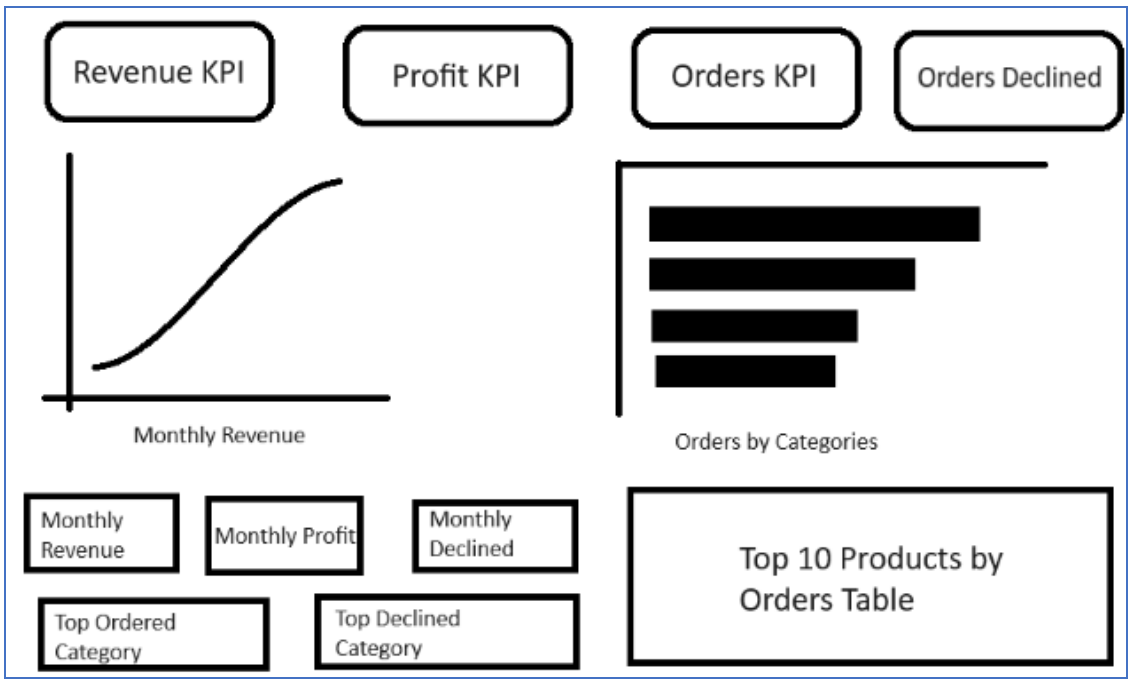

Рисунок 3.9 – Макет сторінки Revenue Summary

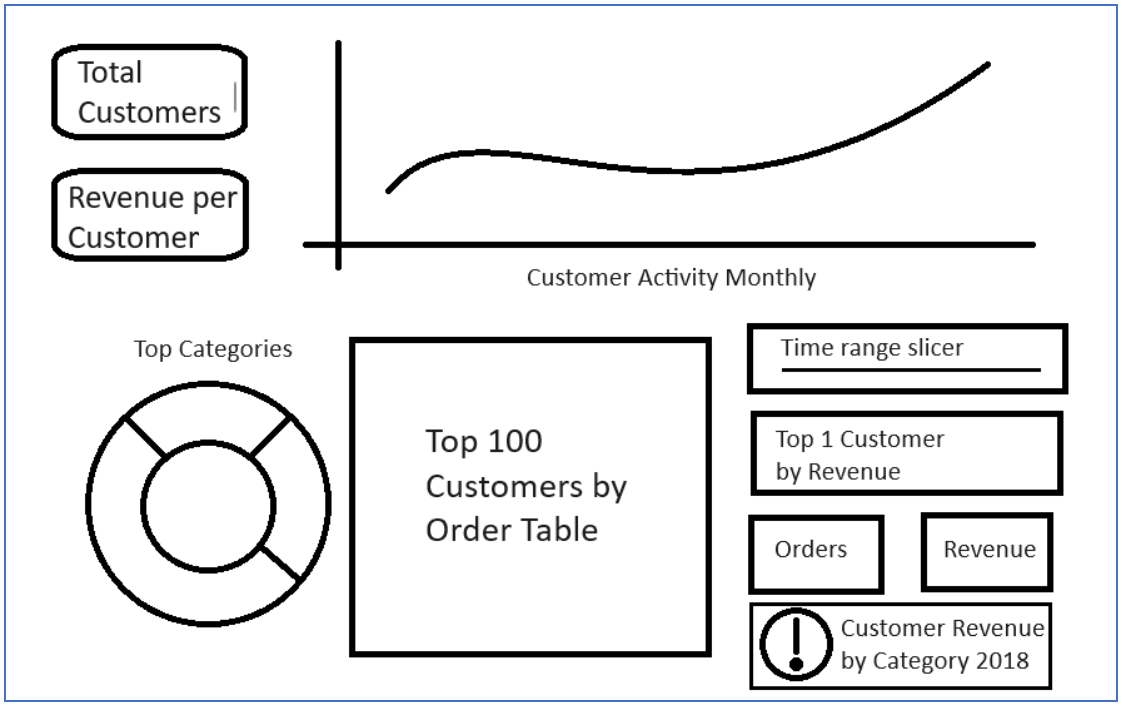

Рисунок 3.10 – Макет сторінки Customer Details

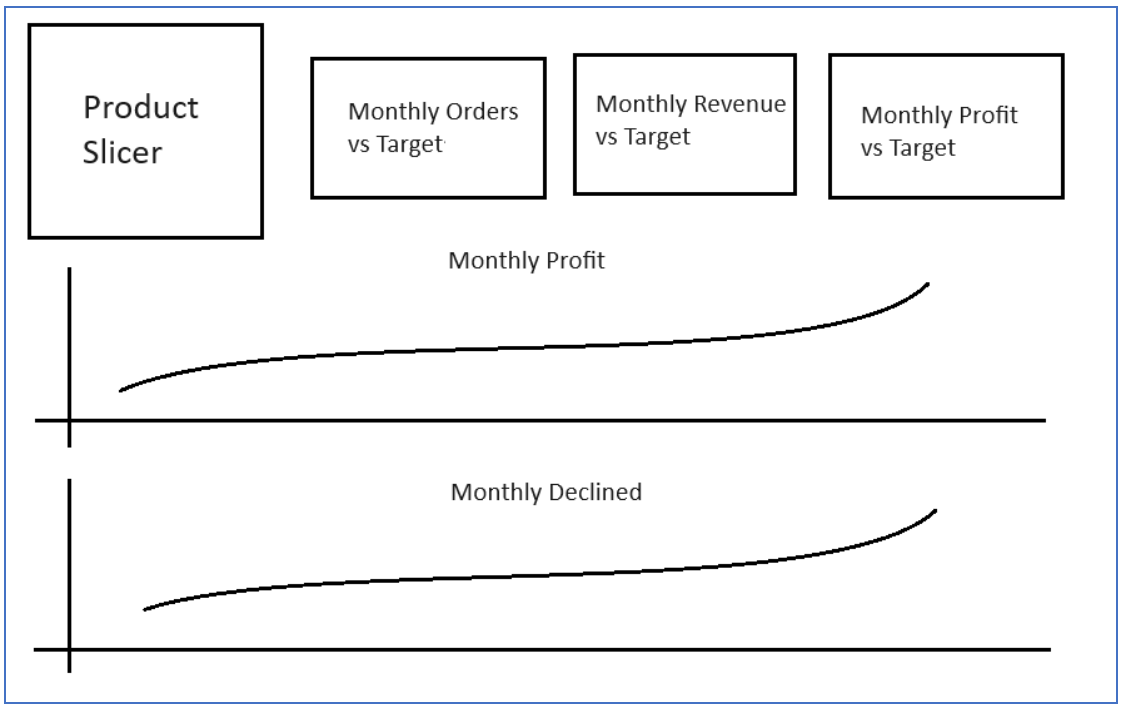

Рисунок 3.11 – Макет сторінки Product Details

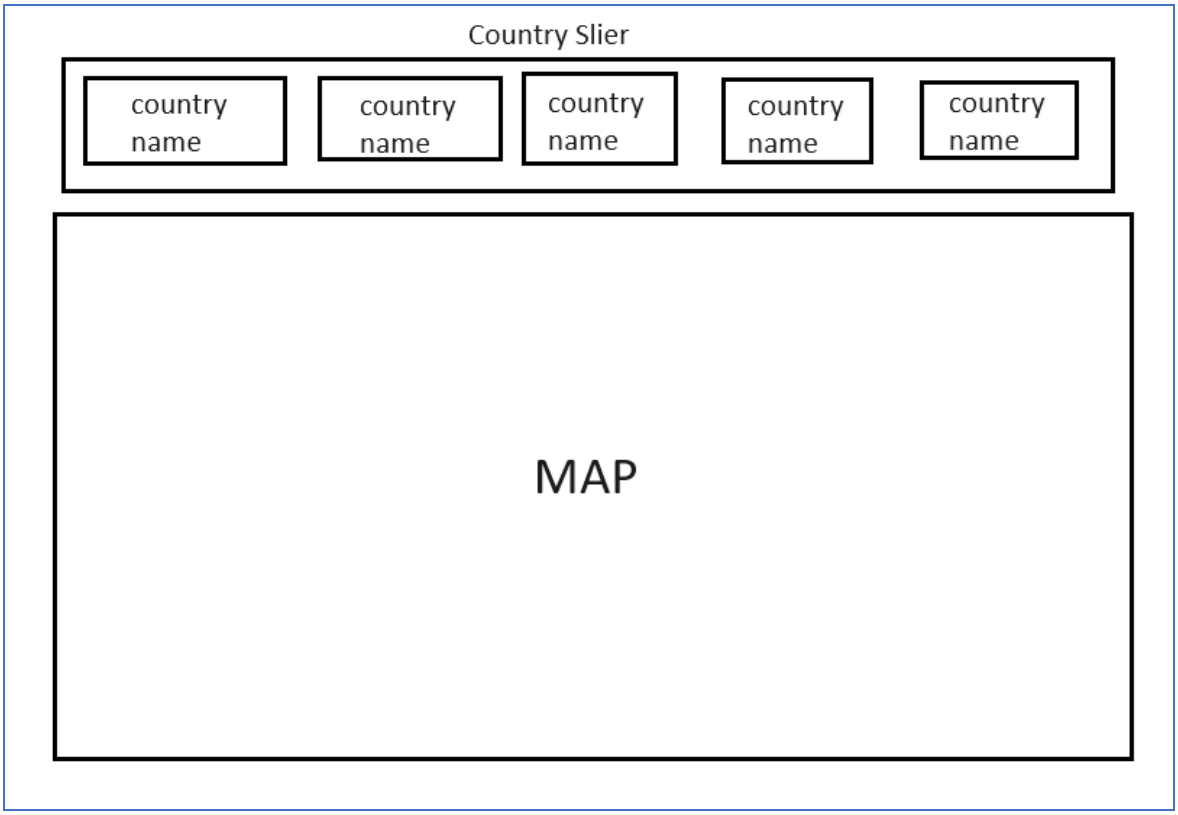

Рисунок 3.12 – Макет сторінки Map

На наступному етапі розробки передбачається створення BI-дашборду в середовищі Power BI та його подальше підключення до бази даних

ClothesStoreDW. Так як трансформацій таблиць в дашборді не проводиться, модель даних повністю відповідає схемі бази даних ClothesStoreDW.

З метою представлення відповідних метрик та показників у складі дашборду, необхідно визначити та реалізувати відповідні вимірювання (measures).

Одними з ключових параметрів, які заслуговують на особливу увагу, будуть:

- Total Orders (загальна кількість замовлень);
- Total Revenue (загальний обсяг прибутку);
- Total Profit (загальний обсяг прибутку);
- Total Declined (загальна кількість відхилених замовлень) та інші важливі показники.

Повний список створених вимірювань зображений на рисунку 3.13.

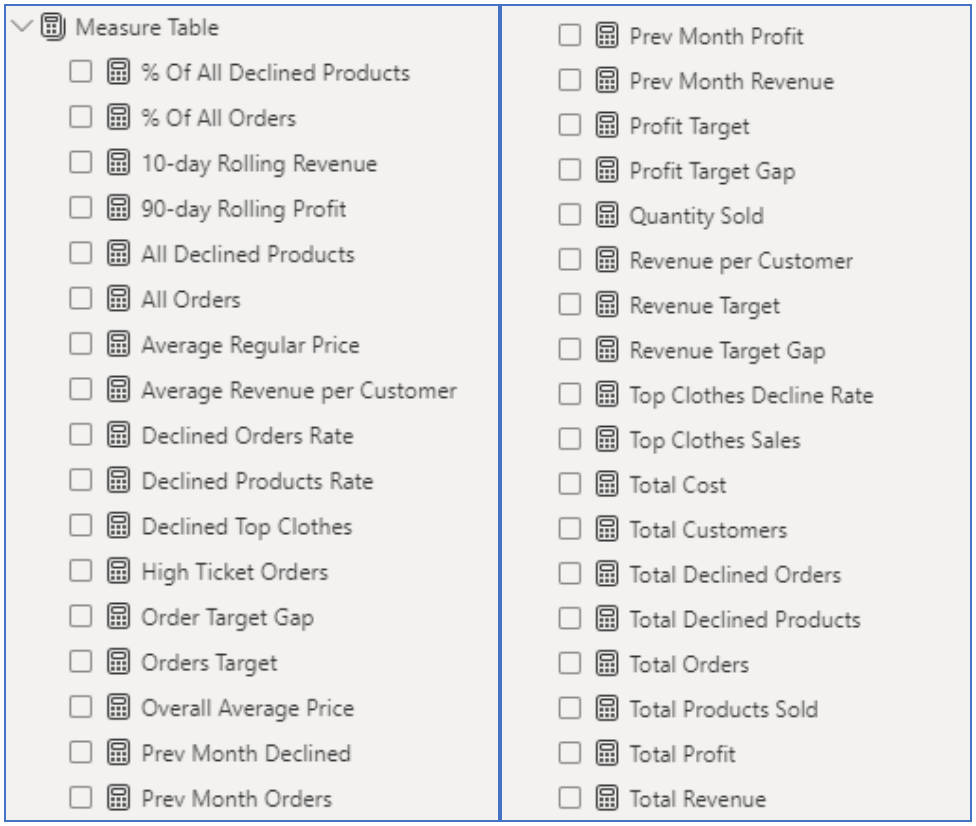

Рисунок 3.13 – Список вимірювань

Далі у розробці передбачається приступити безпосередньо до формування сторінок дашборду відповідно до раніше розроблених макетів.

На рисунках 3.14 – 3.17 зображено 4 створених сторінки дашборду.

Рисунок 3.14 – Сторінка Revenue Summary

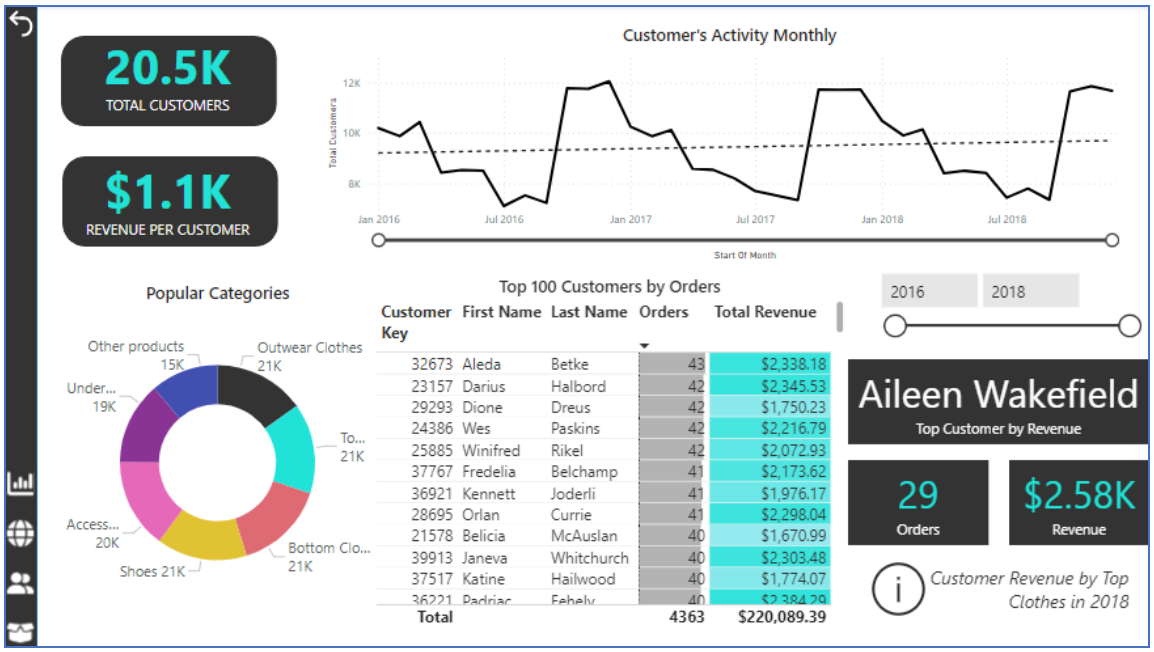

Рисунок 3.15 – Сторінка Customer Details

Основними кольорами для теми дашборду було обрано:

- білий (#FFFFFF) для фону сторінок;
- темно сірий (#333333) для фону певних візуалізацій;
- бірюзово-блакитний (#20E2D7) для значень з темно сірим фоном.

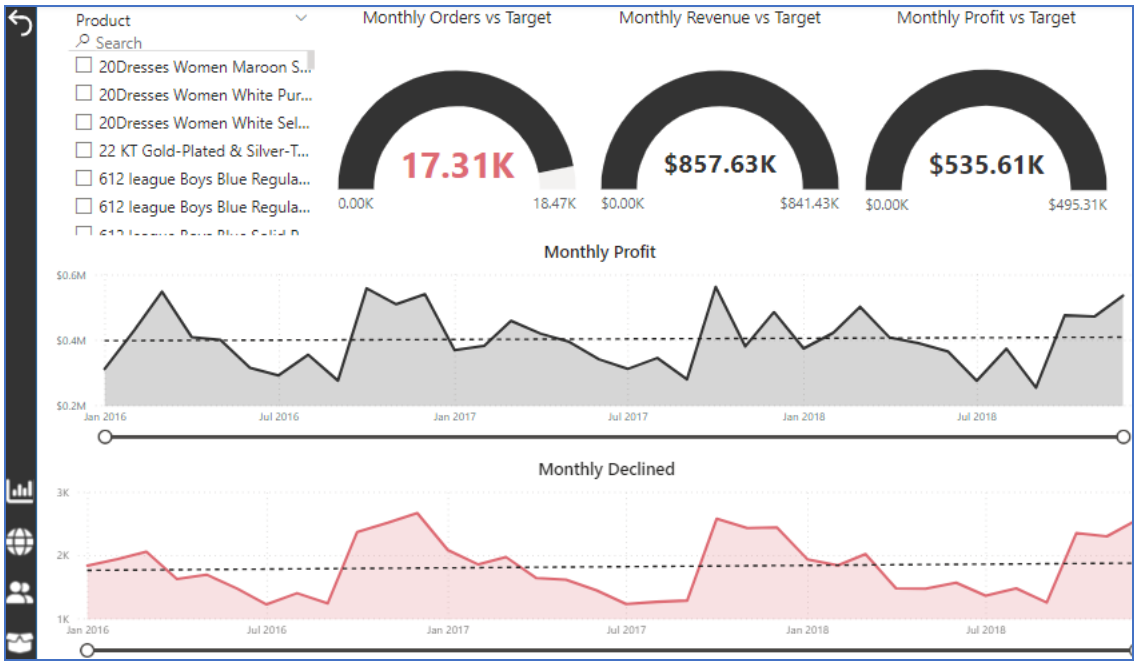

Рисунок 3.16 – Сторінка Product Details

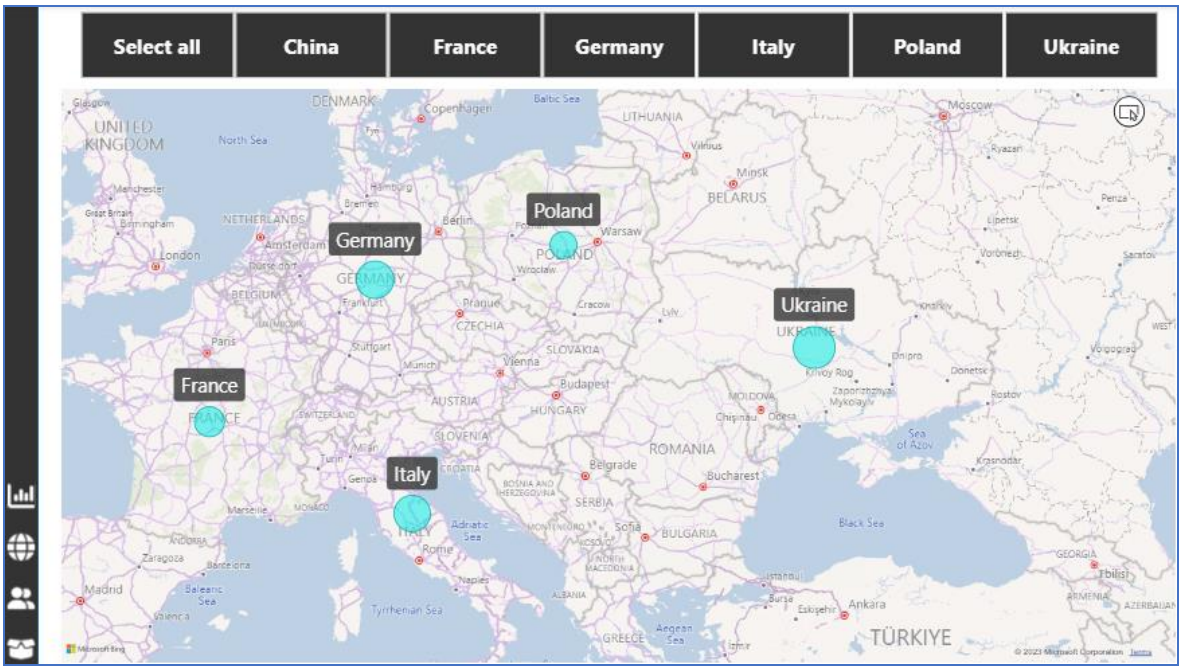

Рисунок 3.17 – Сторінка Map

4 ДОСЛІДЖЕННЯ РОЗРОБЛЕНОЇ МОДЕЛІ ВІЗУАЛІЗАЦІЇ ІНФОРМАЦІЙНОГО НАПОВНЕННЯ

Для перевірки розробленого підходу до візуалізації даних була створена модель мультибрендового магазину (далі ClothesStore) та відповідна база даних на платформі SQL Server. Вказана база даних представляє собою реляційну структуру, спеціально призначену для оптимального функціонування в умовах підприємницької діяльності. Крім того, було розроблено базу даних Data Warehouse (далі ClothesStoreDW), її призначенням є збереження інформації у форматі, сприятливому для подальших візуалізацій та аналізу даних. Одним із ключових аспектів Data Warehouse є швидкість отримання даних, яка має перевищувати темпи доступу до інформації важливої дня аналітики даних у порівнянні з реляційною базою даних. Важливо відзначити, що оптимізація жодної з баз даних не проводилась, тому вони перебувають у рівних умовах.

Для порівняння швидкості отримання даних було розроблено по чотири SQL-запити для обох баз даних. Кожен із цих запитів має ідентичну логіку, структурою та ідентичні агрегаційні функцій. Єдина відмінність полягає у різниці кількості таблиць, з яких необхідно отримати дані, через особливості структури відповідних баз даних. Також для відображення часу виконання запитів буде використано команду “SET STATISTICS TIME ON” [11].

На рисунку 4.1 зображено запит №1, який повертає наростаючий прибуток помісячно кожного року продажів для бази даних ClothesStore.

Результат виконання запиту №1 зображено на рисунку 4.2.

На основі рисунку 4.2 можна спостерігати, що загальний час виконання запиту склав 334 мс (elapsed time), однак час виконання запиту процесором склав 1294 мс (CPU time), що перевищує загальний час виконання запиту. Це явище виникло внаслідок розбиття даного запиту на частини, які виконувалися паралельно. В результаті сумарний час виконання паралельних

обчислень становить 1294 мс.

```
WITH MonthlyRevenue AS (
    SELECT YEAR(o.OrderDate) AS SaleYear, MONTH(o.OrderDate) AS SaleMonth,
        SUM(od.FactPrice) AS Revenue
    FROM Orders as o
        inner join OrderDetails as od on o.OrderID = od.OrderID
    GROUP BY YEAR(o.OrderDate), MONTH(o.OrderDate)
)
SELECT mr1.SaleYear, mr1.SaleMonth, mr1.Revenue AS CurrentMonthRevenue,
    ISNULL(mr2.Revenue, 0) AS PreviousMonthRevenue,
    mr1.Revenue - ISNULL(mr2.Revenue, 0) AS RevenueDifference
FROM MonthlyRevenue mr1
LEFT JOIN MonthlyRevenue mr2 ON mr1.SaleYear = mr2.SaleYear
        AND mr1.SaleMonth = mr2.SaleMonth + 1
WHERE mr1.Revenue >= 0
order by SaleYear, SaleMonth
```

Рисунок 4.1 – Запит №1 до ClothesStore, помісячного прибутку

Results | Messages

	SaleYear	SaleMonth	CurrentMonthRevenue	PreviousMonthRevenue	RevenueDifference
1	2016	1	582257.6268	0.00	582257.6268
2	2016	2	683143.9242	582257.6268	100886.2974
3	2016	3	822031.188	683143.9242	138887.2638
4	2016	4	613460.88	822031.188	-208570.308
5	2016	5	610267.506	613460.88	-3193.374
6	2016	6	520420.512	610267.506	-89846.994
7	2016	7	454381.5672	520420.512	-66038.9448
8	2016	8	532022.004	454381.5672	77640.4368
9	2016	9	442951.413	532022.004	-89070.591
10	2016	10	886950.2304	442951.413	443998.8174
11	2016	11	837392.1096	886950.2304	-49558.1208

Results | Messages

```
SQL Server parse and compile time:
   CPU time = 15 ms, elapsed time = 27 ms.

(36 rows affected)

 SQL Server Execution Times:
   CPU time = 1294 ms,  elapsed time = 334 ms.

Completion time: 2023-12-10T22:51:44.8825385+02:00
```

Рисунок 4.2 – Результат виконання запиту №1 до бази даних ClothesStore

Запит №1 було переписано для бази даних ClothesStoreDW. Логіка та агрегаційні функції змінені не були. Змінений запит №1 зображено на рисунку 4.3.

```sql
WITH MonthlyRevenue AS (
    SELECT d.CalendarYear, MONTH(d.Date) as CalendarMonth,
        SUM(s.FactPrice) AS Revenue
    FROM SalesFact as s
        inner join DateDim as d on d.DateID = s.OrderDateID
    GROUP BY d.CalendarYear, MONTH(d.Date)
)
SELECT mr1.CalendarYear, mr1.CalendarMonth,mr1.Revenue AS CurrentMonthRevenue,
    ISNULL(mr2.Revenue, 0) AS PreviousMonthRevenue,
    mr1.Revenue - ISNULL(mr2.Revenue, 0) AS RevenueDifference
FROM MonthlyRevenue mr1
LEFT JOIN MonthlyRevenue mr2 ON mr1.CalendarYear = mr2.CalendarYear
        AND mr1.CalendarMonth = mr2.CalendarMonth + 1
WHERE mr1.Revenue >= 0
order by mr1.CalendarYear, mr1.CalendarMonth
```

Рисунок 4.3 – Запит №1, помісячного прибутку для бази даних ClothesStoreDW

Результати виконання запиту №1 до ClothesStoreDW зображено на рисунку 4.4.

Results | Messages

	CalendarYear	CalendarMonth	CurrentMonthRevenue	PreviousMonthRevenue	RevenueDifference
1	2016	1	582257.6268	0.00	582257.6268
2	2016	2	683143.9242	582257.6268	100886.2974
3	2016	3	822031.188	683143.9242	138887.2638
4	2016	4	613460.88	822031.188	-208570.308
5	2016	5	610267.506	613460.88	-3193.374
6	2016	6	520420.512	610267.506	-89846.994
7	2016	7	454381.5672	520420.512	-66038.9448
8	2016	8	532022.004	454381.5672	77640.4368

```
(36 rows affected)

 SQL Server Execution Times:
   CPU time = 637 ms,  elapsed time = 160 ms.
```

Рисунок 4.4 – Результат виконання запиту №1 до ClothesStoreDW

На основі рисунку 4.4 можна спостерігати, що загальний час виконання запиту склав 160 мс (elapsed time), а час виконання запиту процесором склав 637 мс (CPU time).

Подальше виконання включало в себе запити №2-4, які були відправлені до обох баз даних, і час їх виконання був зафіксований та порівняний в таблиці 4.1.

Таблиця 4.1 – Порівняння часу виконання SQL запитів в двох базах даних

Запит	CPU Time (t1), мс (ClothesStore)	CPU time (t2), мс (ClothesStoreDW)	$\frac{t1}{t2}$	Elapsed time (t3), мс (ClothesStore)	Elapsed time (t4), мс (ClothesStoreDW)	$\frac{t3}{t4}$
№1	1294	637	2.03	334	160	2.08
№2	3002	2202	1.36	1285	1791	0.72
№3	720	686	1.05	366	310	1.18
№4	1000	734	1.36	258	129	2

На основі порівняльного аналізу часу виконання чотирьох запитів в базах даних ClothesStore та ClothesStoreDW можна зробити висновок, що отримання даних з ClothesStoreDW в цілому є ефективним процесом. Особливо виділяються запити №1 та №4, які виконались значно швидше. Це можливо завдяки структурі зберігання даних Data Warehouse, яка оптимізована для швидкого витягування та обробки інформації.

Наступним етапом дослідження є аналіз розробленого графічного звіту в середовищі Power BI. У цьому випробуванні розглядатимуться такі параметри, як швидкість завантаження звіту, функціональні можливості, які надає звіт, та гнучкість у внесенні змін для адаптації до вимог користувача. Аналіз цих аспектів дозволить визначити ефективність та придатність графічного звіту для вирішення конкретних бізнес-задач.

Час запуску візуалізацій графічного звіту складає 42 секунди, що зафіксовано на рисунку 4.5 за допомогою використання таймеру Windows.

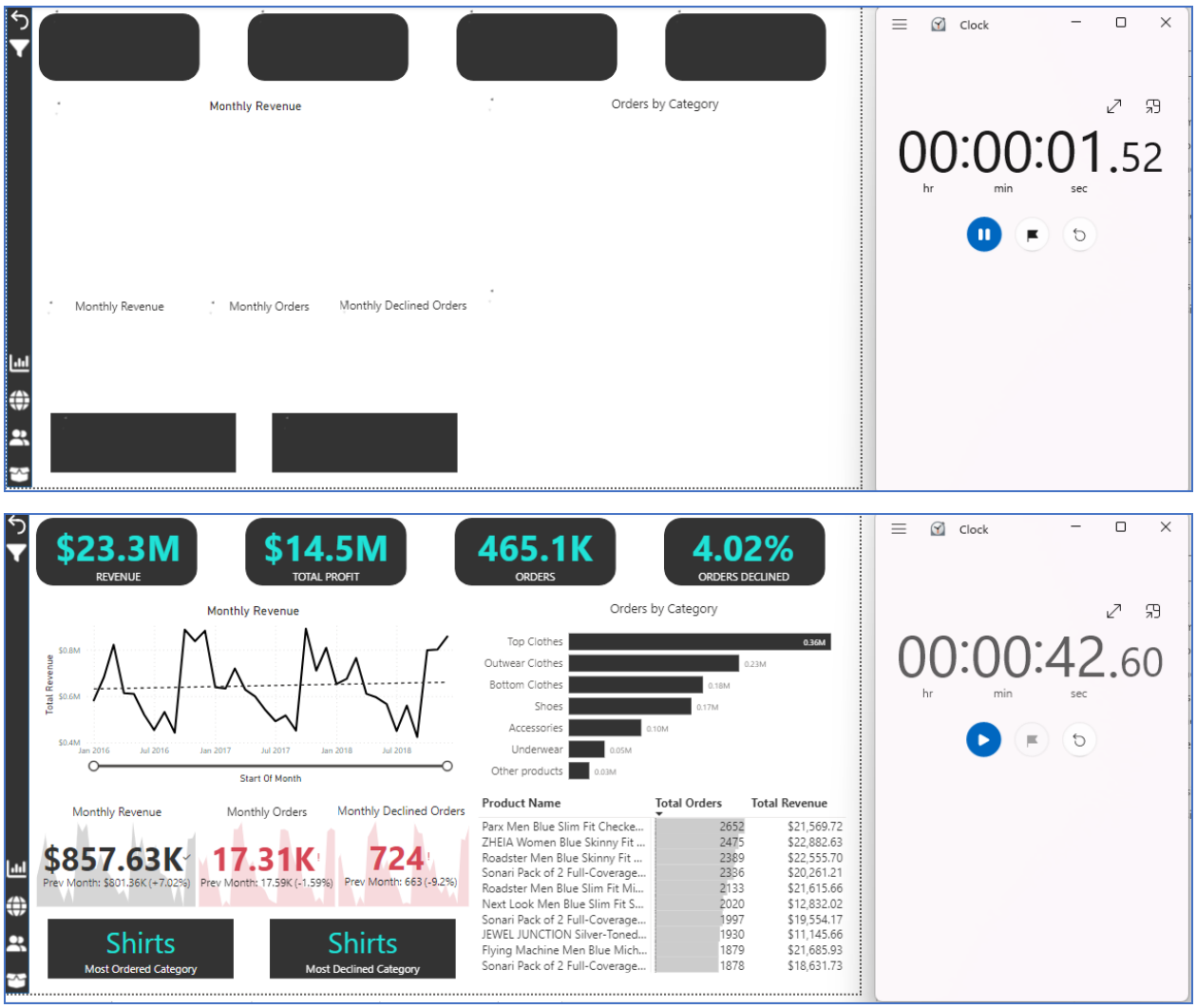

Рисунок 4.5 – Час запуску візуалізацій графічного звіту Power BI

Наступною точкою розгляду є перша сторінка "Revenue Summary". Ця сторінка включає загальну інформацію про прибуток, кількість замовлень, топ товарів та категорій. Звіт є інтерактивним, надаючи можливість фільтрування всієї сторінки за допомогою одного натискання на конкретну категорію в візуалізації топ категорій. Це покращує зручність користування та дозволяє отримувати точкову інформацію за допомогою взаємодії з візуальними елементами звіту. Результат фільтрації звіту зображено на рисунку 4.6.

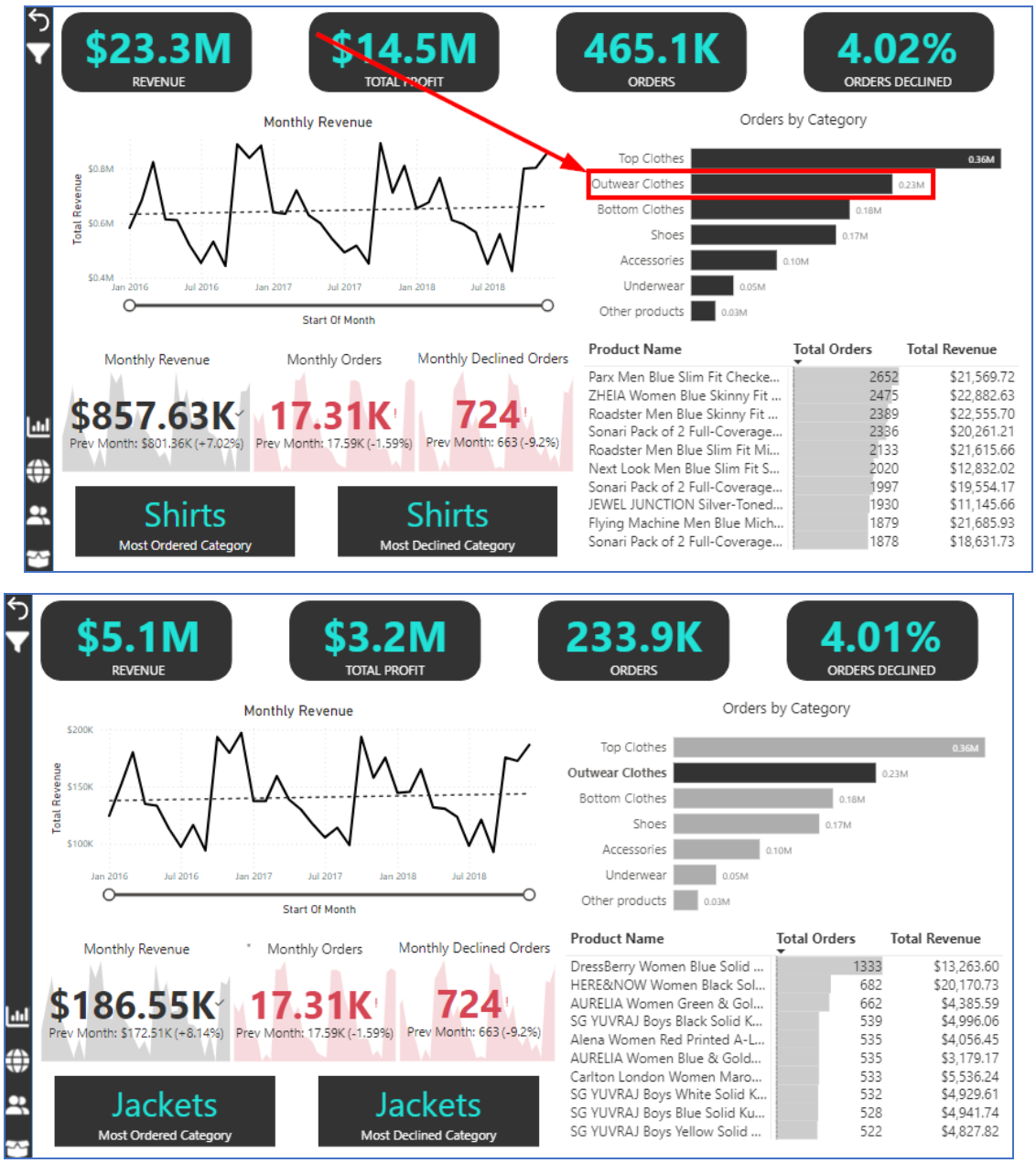

Рисунок 4.6 – Результат фільтрації звіту за обраною категорією

Також можливо використовувати фільтрацію звіту, вибираючи конкретний місяць або місяці у візуалізації "Monthly Revenue" (місячний прибуток). Цей функціонал дозволяє точно визначати період аналізу та отримувати відомості, зорієнтовані на обраний місяць, забезпечуючи додаткову гнучкість у виведенні інформації на звіті. Результат фільтрації звіту за місячними прибутками зображено на рисунку 4.7.

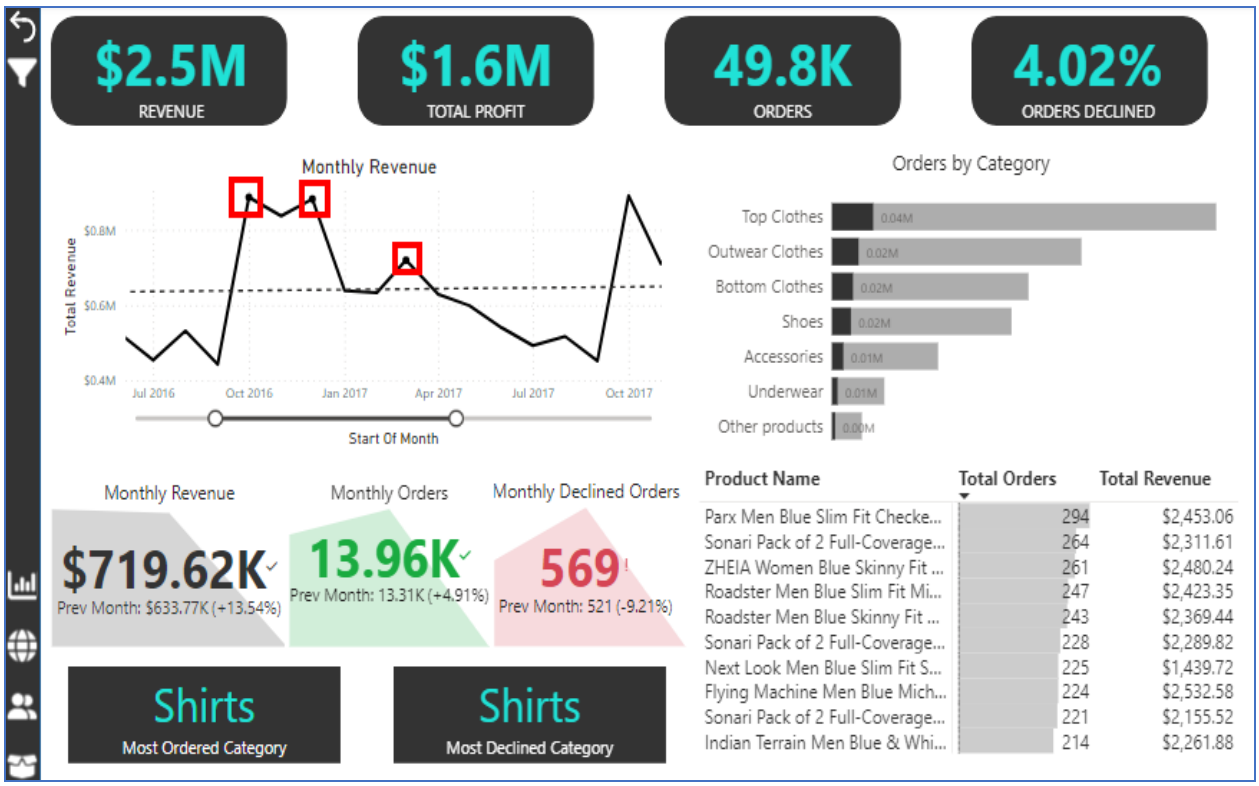

Рисунок 4.7 – Результат фільтрації звіту за місячними прибутками

Power BI також включає функціонал "Bookmarks" (закладки), який забезпечує можливість зберігання конкретних налаштувань звіту та повернення його до певного стану за допомогою використання закладок. Додатково, закладки можуть бути пов'язані з кнопками у звіті, і при їх натисканні вони миттєво викликають відповідну закладку, змінюючи конфігурацію звіту згідно визначеним параметрам. Це забезпечує зручний та швидкий доступ до різних налаштувань звіту залежно від потреб користувача.

Результат використання кнопки "Reset" зв’язаною з закладкою, яка вимикає усі фільтри застосовані на звіті, зображено на рисунку 4.8.

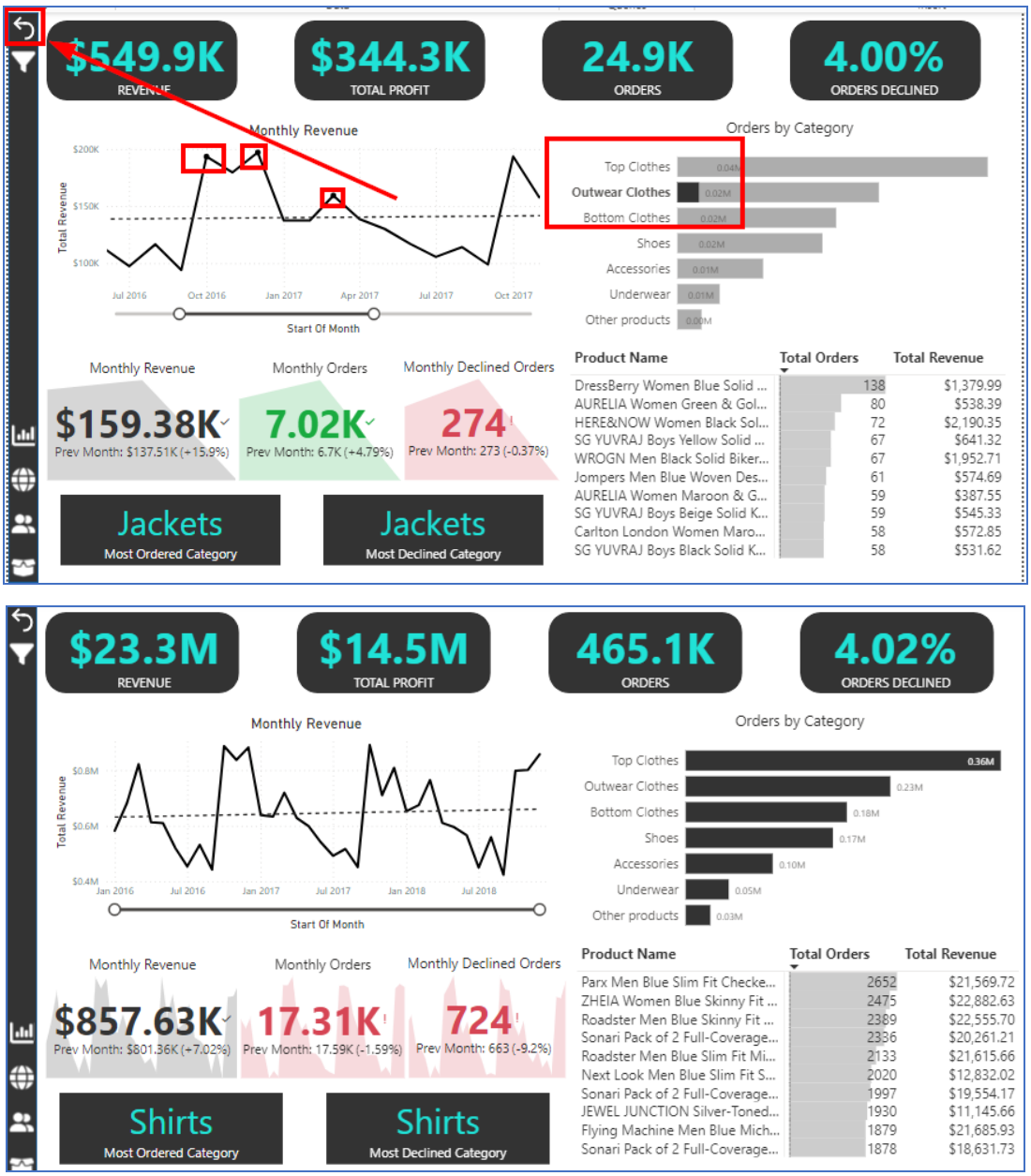

Рисунок 4.8 – Результат очищення фільтрів за допомогою кнопки та закладки

Крім того, були розроблені індивідуальні кнопки для навігації між різними сторінками звіту. Результат переходу на сторінку "Map" із візуалізацією карти завдяки відповідній кнопці відображено на рисунку 4.9.

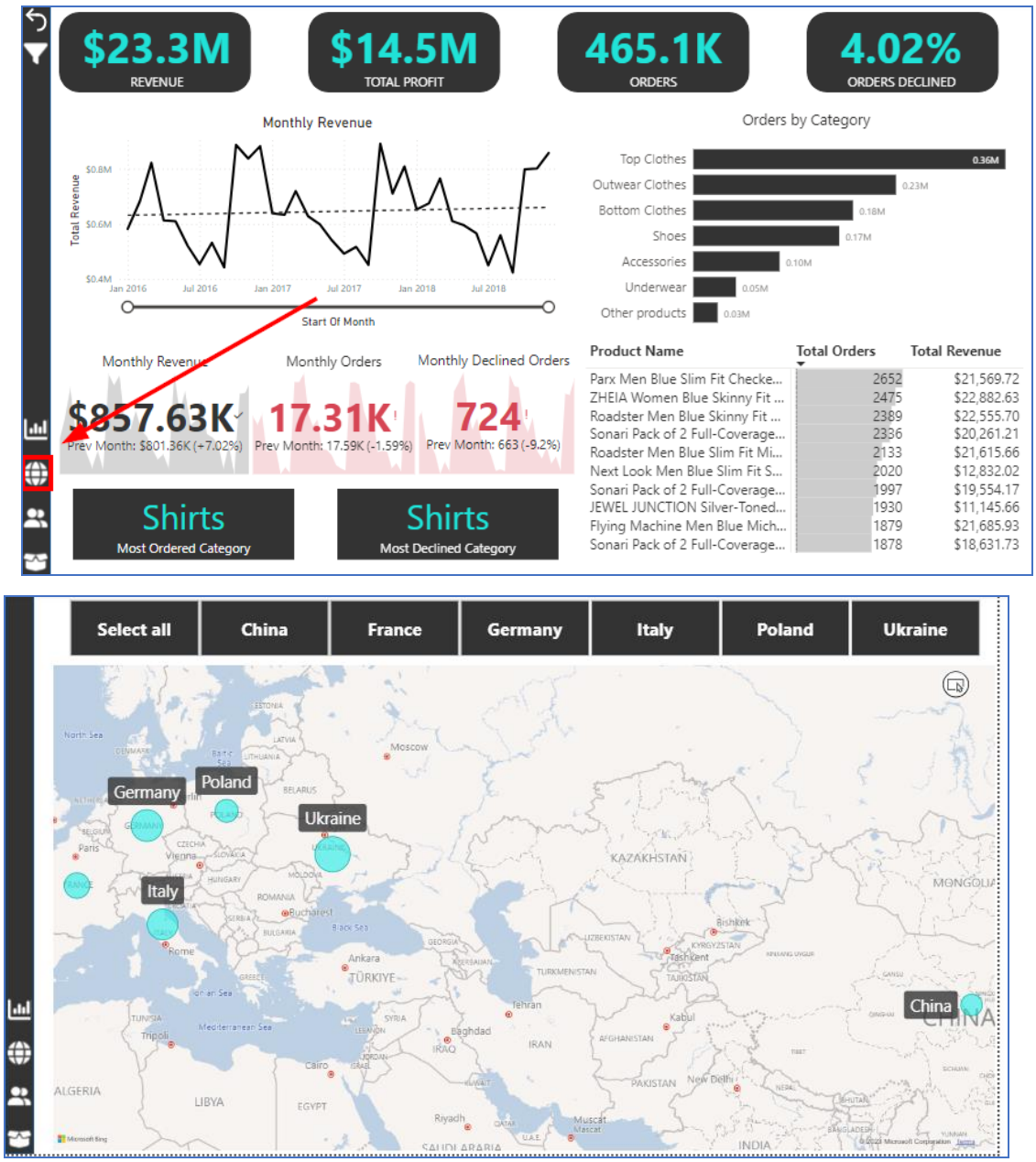

Рисунок 4.9 – Результат переходу на сторінку “Map” завдяки індивідуальній кнопці

Ще однією корисною функцією у Power BI є можливість використання підказок ("Tool Tip"). Коли курсор миші наводиться на конкретну візуалізацію, з'являється підказка, яка надає більш детальну інформацію. Появу підказок зображено на рисунку 4.10.

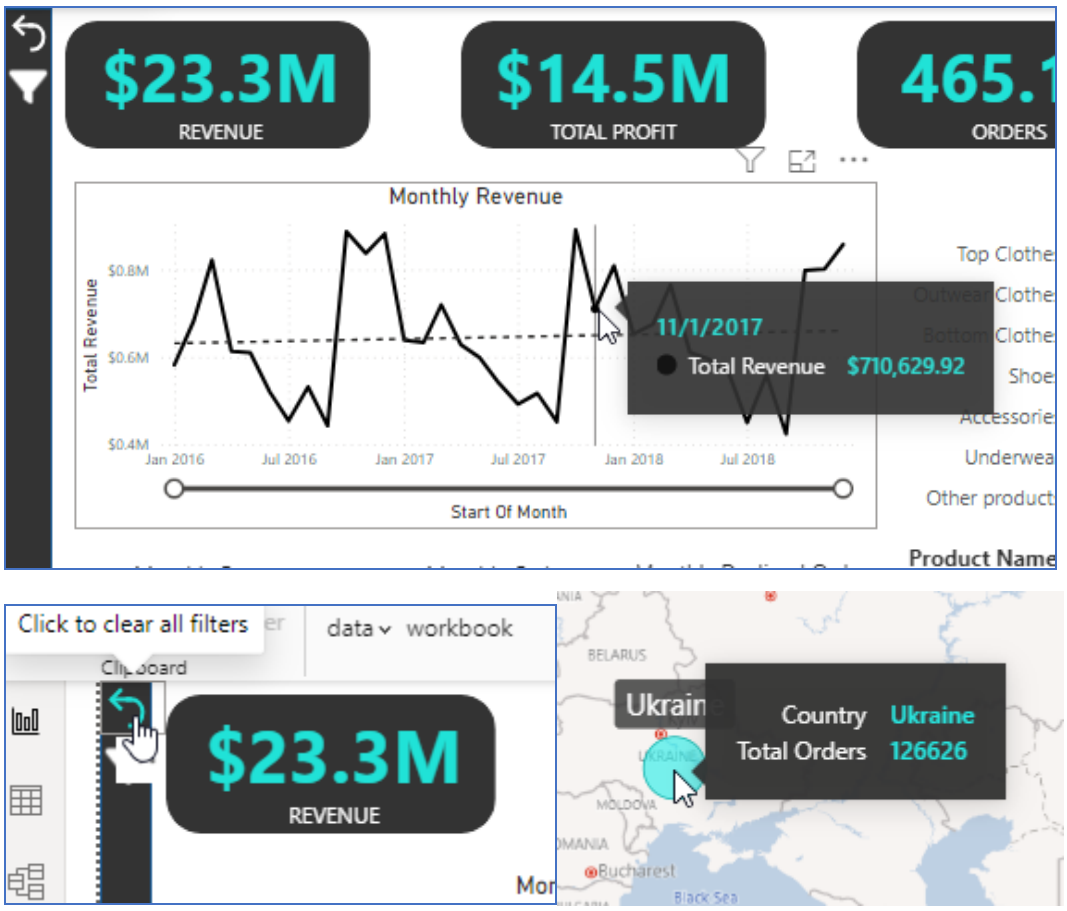

Рисунок 4.10 – Поява підказок

Наступною функцією, що варта уваги, є "Slicer" в Power BI , яка використовується для створення інтерактивних фільтрів, що дозволяють користувачам швидко вибирати та фільтрувати дані на сторінці звіту. За допомогою фільтрів "Slicer" можна створювати різні типи фільтрів, такі як випадаючі списки, календарі, чекбокси тощо. Ця функція дозволяє спростити процес аналізу даних, надаючи зручний інструмент для вибору конкретних значень чи категорій. Приклад використання фільтру “Slicer” зображено на рисунку 4.11.

Наступним буде проведено аналіз часових та фізичних витрат при створенні нової сторінки у звіті Power BI. Вивчимо, скільки часу займає виконання даного завдання та яка кількість кліків миші та натискань кнопок на клавіатурі є необхідною для його виконання, не враховуючи друку тексту для найменувань візуалізацій, сторінок тощо.

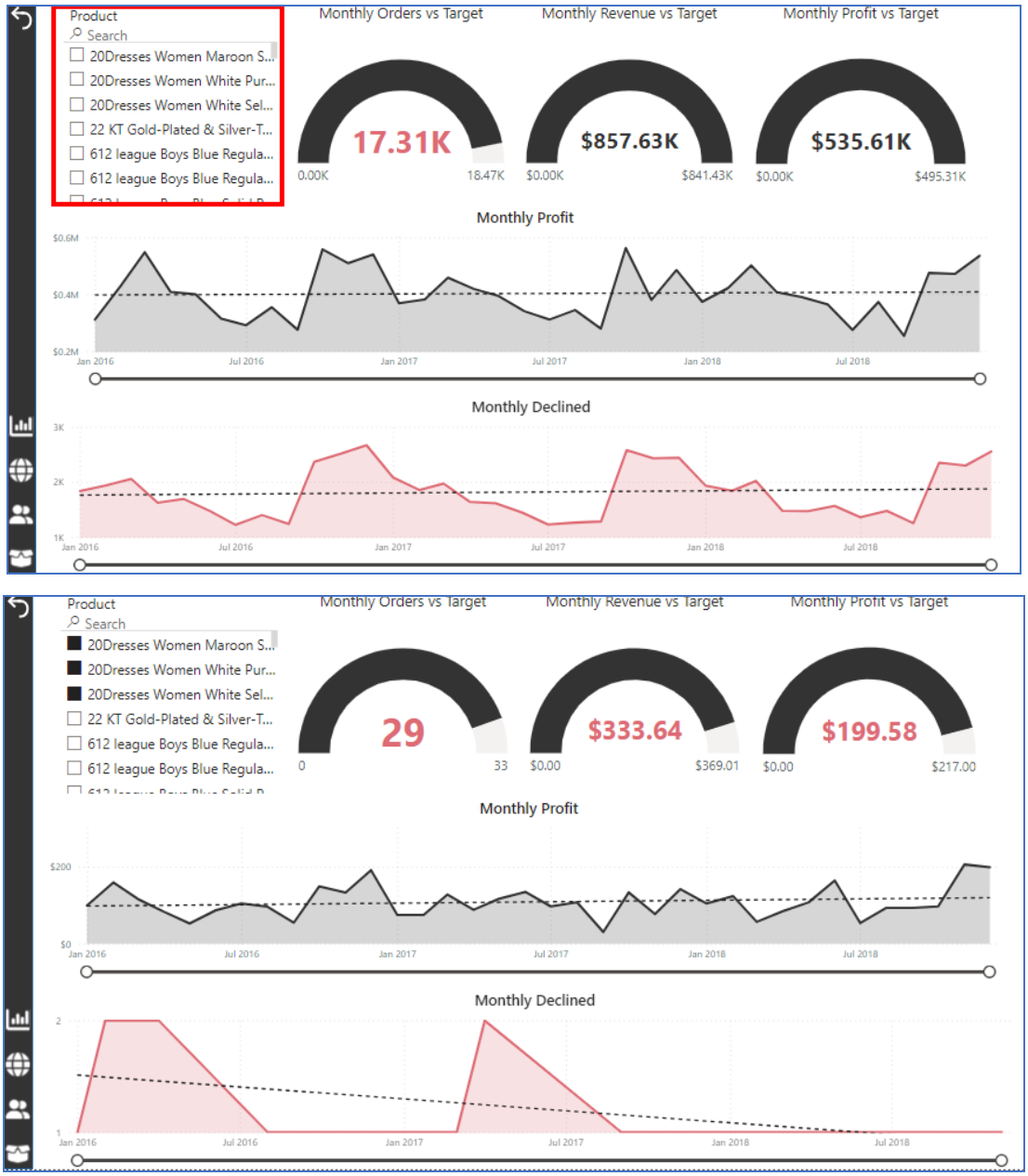

Рисунок 4.11 – Використання фільтру “Slicer” на сторінці “Product Details”

На новій сторінці всі візуалізації були розроблені абсолютно з початку, і не включали жодних елементів, скопійованих з інших сторінок. Важливо відзначити, що використання інших візуалізацій як шаблонів також є ефективним для економії часу. Створена сторінка включає в себе таблицю, що відображає топ-5 міст за прибутком, два слайсера та дві мапи, які ілюструють прибуток та кількість замовлень у містах, де розташовані магазини мережі.

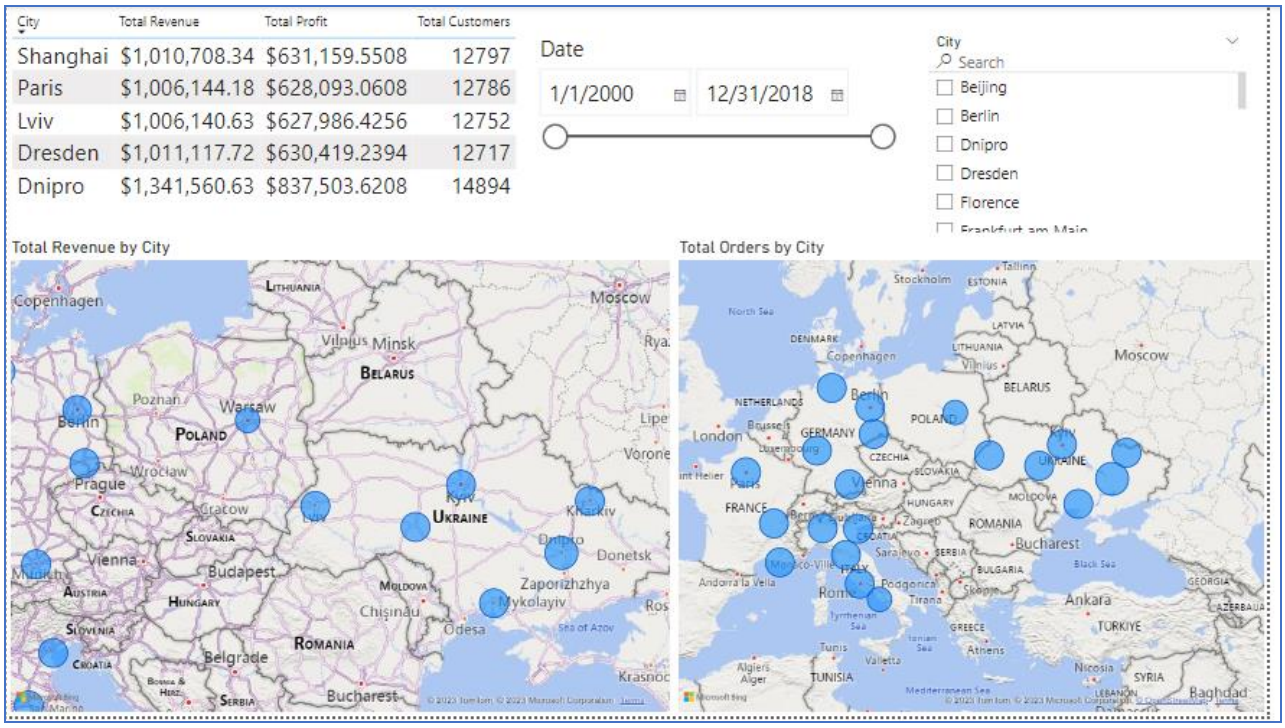

Рисунок 4.12 – Нова сторінка графічного звіту

Витрати часу на створення сторінки склали 25 хвилин, було здійснено 114 натискань кнопок миші та клавіатури. Ці показники є дуже задовільними, оскільки протягом такого короткого періоду часу кінцевий користувач має можливість отримати доступ до значущої інформації, відповідаючи на актуальні питання без зайвого затримання.

Створену нову сторінку зображено на рисунку 4.12.

ВИСНОВКИ

В сучасному світі, де досягнення високих технологій порівняно з минулим століттям надають великі можливості, візуалізація даних стає складним та важливим завданням. Це вимагає комплексного підходу та вибору оптимальних рішень, враховуючи різні вимоги галузей бізнесу. Організація даних через системи, такі як Data mart, Data Cube, Data Lakes, Data Virtualization та Data Warehouse, відіграє ключову роль у впорядкуванні цього завдання.

Необхідність візуалізації даних передбачає три основні підходи. Перший - використання платформ для бізнес-інтелекту (BI), які надають інструменти для створення візуалізацій без глибоких технічних знань. Другий - розробка власних додатків для візуалізації, що дозволяє індивідуалізацію графічних представлень. Третій - використання мов програмування, таких як Python, JavaScript або R, для самостійного створення та налаштування візуалізацій.

Для розробки системи організації та зберігання даних для мультибрендового магазину була використана реляційна база даних, зокрема, SQL Server, яка виступає основою для вивчення методів Data Warehouse та Business Intelligence. Застосунок Talend Open Studio був використаний для створення процесів ETL. Power BI був використаний для візуалізації та аналізу даних.

Виявлено, що база даних Data Warehouse оптимізована для аналітичних запитів та ефективніше обробляє складні запити порівняно з реляційною базою даних. Power BI вражає своєю виразністю та інтерактивністю, роблячи його потужним інструментом для аналізу та візуалізації даних. В цілому, розроблені модель та інструменти надають ефективну систему аналітичних процесів для взаємодії з даними та прийняття обґрунтованих рішень.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1 Радченко І. В., Шеховцов О. В., Коваленко А. А., Ситник О. В. Формування кластерів на одноплатних комп'ютерах у мережах IoT. Системи управління, навігації та зв’язку. Полтава : Національний університет «Полтавська політехніка імені Юрія Кондратюка», 2024. Вип. 2(76). С. 132–136.

2 Rosenberg D. Cartographies of time / D. Rosenberg, A. Grafton – New York: Princeton Architectural Press, 2010 – 272 p.

3 Rendgen S. Information Graphics / S. Rendgen, J. Wiedemann – Cologne. TASCHEN, 2012 – 480 p.

4 Few S. Information Dashboard Design Display data for at-a-glance monitoring Second Edition /S. Few – Burlingame. Analytics Press, 2013 – 260 p.

5 Kimball R. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Third Edition / R. Kimball, M. Ross – Indianapolis: John Wiley & Sons, Inc, 2013 – 564 p.

6 McDaniel E. The Accidental Analyst / E. McDaniel, S. McDaniel – Seattle. Freakalytics, 2012 – 300 p.

7 Cairo l. Functional Art, The: An introduction to information graphics and visualization / l. Cairo – Berkeley. New Riders, 2012 – 384 p.

8 Nussbaumer Knaflic C. Storytelling with Data: A Data Visualization Guide for Business Professionals / C. Nussbaumer Knaflic – Indianapolis. Wiley Publishing, 2015 – 288 p.

9 Yau N. Visualize This: The FlowingData Guide to Design, Visualization, and Statistics / N. Yau– Indianapolis. Wiley Publishing, 2011 – 384 p.

10 MySQL. Офіційна документація [Електронний ресурс] — Режим доступу: https://dev.mysql.com/doc/

11 PostgreSQL. Офіційна документація [Електронний ресурс] —

Режим доступу: https://www.postgresql.org/docs/

12 SQL Server. Офіційна документація [Електронний ресурс] — Режим доступу: https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16

13 Oracle Database. Офіційна документація [Електронний ресурс] — Режим доступу: https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/#Oracle%C2%AE-Database

14 MariaDB Server. Офіційна документація [Електронний ресурс] — Режим доступу: https://mariadb.com/kb/en/documentation/

15 Reis J. Fundamentals of Data Engineering / J. Reis, M. Housley – Sebastopol. O'Reilly Media, 2022 – 447 p.

16 Steele J. Beautiful Visualization / J. Steele, Iliinsky N. – Sebastopol. O`Reilly Media, 2010 – 415 p.

17 Wexler S. The Big Book of Dashboards / S. Wexler, J. Shaffer, A. Cotgreave – Hoboken. John Wiley & Sons, 2017 – 448 p.

18 Talend. Офіційна документація [Електронний ресурс] — Режим доступу: https://help.talend.com/r/en-US/8.0/release-notes/esb-migration-from-7.x.x-to-8.0.1

19 SQL Server Integration Services. Офіційна документація [Електронний ресурс] — Режим доступу: https://learn.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-ver16

20 Apache NiFi. Офіційна документація [Електронний ресурс] — Режим доступу: https://nifi.apache.org/docs.html

21 Tableau. Офіційна документація [Електронний ресурс] — Режим доступу: https://help.tableau.com/current/pro/desktop/en-us/default.htm

22 Power BI. Офіційна документація [Електронний ресурс] — Режим доступу: https://learn.microsoft.com/en-us/power-bi/

23 QlikView. Офіційна документація [Електронний ресурс] — Режим доступу: https://help.qlik.com/en-US/qlikview/May2023/Subsystems/QMC/Content/QV_QMC/QMC_Documents.htm

24 Looker. Офіційна документація [Електронний ресурс] — Режим доступу: https://cloud.google.com/looker/docs

25 Ramalho L. Fluent Python: Clear, Concise, and Effective Programming 2nd Edition / L. Ramalho – Sebastopol. O`Reilly Media, 2022 – 1012 p.

26 Python. Офіційна документація [Електронний ресурс] — Режим доступу: https://www.python.org/doc/versions/

27 Petrov A. Database Internals / A. Petrov – Sebastopol. O'Reilly Media, 2019 – 370 p.