CHAPTER «ENGINEERING SCIENCES»

PROCESSING AND STORAGE OF DIFFERENT DATA WITH THE HELP OF BI-TECHNOLOGIES

ІНФОРМАЦІЙНА СИСТЕМА ОБРОБКИ І ЗБЕРІГАННЯ РІЗНОТИПОВИХ ДАНИХ ЗА ДОПОМОГОЮ BI-ТЕХНОЛОГІЙ INFORMATION SYSTEM FOR

Nataliya Boyko[1]

DOI: https://doi.org/10.30525/978-9934-26-195-4-11

Abstract. Conditions for the development of modern information space indicate the need to process large amounts of structured, poorly structured and unstructured data, which are in heterogeneous sources. The study describes the approaches, models, methods and tools for building data warehouses, formulates the purpose and objectives of the study, and the concept of building corporate repositories. This section discusses the input and output data features that the system will process. The information model of OLTP systems and data warehouses is also designed; logical essences and their business needs are described. The input data will be the result of the OLTP system, which simulates the operation of the online store. Then, by various means, different amounts of data will be transferred with specific processing to the data warehouse, from the structure of which it will be possible to obtain the original data for further analysis. This section explores the various tools for implementing the system describes their advantages and disadvantages. SQLServer was chosen as the data processing mechanism and DatabaseEngine / SSIS packages as the tools for forming the integration layer and ETL processes. The necessary software and hardware have been included. The object of the study is the process of migrating data from a database that works directly with a business program (OLTP system) to a

[1] Candidate of Economic Sciences, Associate Professor,
Associated Professor at the Department of Artificial Intelligence,
Lviv Polytechnic National University, Ukraine

data warehouse (OLAP system) for further archival storage and analysis. The work aims to create a relational database that will simulate the work of the online store, create an appropriate data warehouse and study the speed of the ETL process of data transfer to the data warehouse in different ways. The main tasks for the realization of the research goal should include construction of a relational database and its filling with suboptimal data; construction of a data warehouse; creation of ETL process with the help of SSIS packages; design of ETL process using T-SQL; comparison of the obtained results and conclusions. We also considered the software implementation, which conducted a study of the speed of data transfer using two selected tools and felt some functionality of the system so that the user interacted correctly with it. To achieve this goal, developing a data warehouse on a hybrid approach is necessary and configuring the ETL process between the relational database and the data warehouse using SSIS packages. To perform experiments, you need to compare data transfer efficiency on tables of different dimensions. As a result of the study, a three-tier system was created, consisting of an OLTP system as a transactional layer, ETL processes using DatabaseEngine and SSIS packages, as an integration layer and a data warehouse built on a hybrid principle as an analytical layer. The implemented system is analogous to the latest self-service system, as it can provide its business needs without third-party funds.

1. Вступ

У наш час компанії часто оперують великими обсягами даних, тому запроваджувати різного роду аналіз, базуючись на даних з баз, які напряму використовуються для бізнес потреб не завжди є оптимальним варіантом. Нормалізована реляційна база даних, яка у більшості випадків використовується для таких потреб, може складатись з десятків або навіть з тисяч таблиць, з різною назвою і гранулярністю, тому виявити дані які будуть необхідні для певного аналізу і отримати її є непросто. Сучасні корпорації можу використовувати не одну бізнес програму, які працюють на багатьох базах даних. І об'єднання архітектурно не пов'язаних систем для проведення певного аналізу може вплинути на якість даних.

Часто, вектор дослідження може бути напрямлений на аналіз історичних даних, а такі системи в основному збирають інформацію, яка

актуальна лише на певному проміжку часу, так як зберігати і підтримувати весь історичний об'єм даних у такого типу системах є ресурсозатратним.

Метою даного дослідження є визначення оптимальних засобів для внутрішнього переміщення різних обсягів даних, які б потенційно використовувались організацією для проведення аналізу з метою покращення діяльності.

Основними засобами переміщення даних, які будуть розглянуті у даній роботі являються:

– переміщення даних за допомогою внутрішніх засобів MSSqlServer та T-SQL;

– переміщення даних за допомогою генерації динамічного BIML скрипта, який буде створювати SSIS пакет для інтеграції даних.

Отож типовим рішенням для таких проблем є створення сховища даних. Сховище даних – це централізоване місце зберігання даних для підприємства, яке містить об'єднані, очищені та історичні дані. У сховищі акумулюється інформація, яка отримана з різних джерел, які визначають бізнес сторону діяльності підприємства. Інформація зберігається у такому вигляді, щоб легко задовольнити практичні потреби користувачів. Схеми даних у таких сховищах спрощені і більш придатні для формування аналітичних звітів, ніж нормалізовані реляційні бази даних.

Метою сховища даних є орієнтація на бізнес: вона покликана сприяти прийняттю рішень, дозволяючи кінцевим користувачам консолідувати та аналізувати інформацію з різних джерел.

2. Опис предметного середовища

При набутті широкого попиту серед систем прийняття рішень, спосіб звертання до баз, які базуються на OLTP-системи виявився не оптимальним у багатьох випадках, тому що саме такий тип системи має постійне велике навантаження і дані часто розподіленні по багатьох таблицях. Тому час отримання даних та їх якість будуть далекі від ідеалу. Отож була запропонована концепція сховища даних, що являє собою інтегровані набори даних, зібрані з різних джерел [1].

Для отримання розуміння чому сховища даних є ефективнішим рішенням для проведення аналітичних досліджень, потрібно зрозуміти

ключові відмінності сховищ даних від бази даних. Основна відмінність полягає в тому, що бази даних – це організовані колекції збережених даних. Сховища даних – це інформаційні системи, побудовані з декількох джерел даних – вони використовуються для аналізу даних [2].

Реляційні бази даних базуються на OLTP-системах. OLTP (онлайн обробка транзакцій) – це термін для системи обробки даних, що фокусується на транзакціях.

Зазвичай такі системи містять інформацію, що використовується бізнесом щодня. Результатом їх функціонування є швидкі, ефективні запити та інформація, яка є актуальною та точною. Бази даних OLTP оптимізовані для швидкої роботи CRUD-операцій (створення, читання, оновлення та видалення). Однак більш складні аналітичні запити можуть швидко знизити їх ефективність. Такі бази даних були розроблені для підтримки тисяч або більше користувачів одночасно, без будь-якого погіршення продуктивності.

В свою чергу сховища даних є частиною OLAP-систем. OLAP (онлайн-аналітична обробка) – це термін для системи обробки даних, що фокусується на аналізі даних та прийнятті рішень, а не на продуктивності та повсякденному використанні. Багато систем OLAP пов'язані з рішеннями бізнес-аналітики (BI), які полегшують отримання необхідної інформації. Такі системи можуть містити в собі вже агреговані дані та велику кількість історичних записів. Сховища даних OLAP можуть підтримувати лише відносно обмежену кількість одночасних користувачів. Оскільки рішення для сховища даних використовує більш складні запити, що циркулюють у багатьох різних сховищах даних, воно обов'язково вимагає більше ресурсів і, отже, не таке масштабоване, як база даних[3].

Отже нижче буде наведена узагальнена таблиця ключових відмінностей між сховищем даних та базою даних (таблиця 1).

Сховище даних є центральним елементом системи BI, побудованої для аналізу даних та звітності. Основна мета створення сховища даних полягає в тому, щоб представити дані, які мають вагу в управлінні бізнесом, у стандартизованій та доступній формі та зробити їх придатними для аналізу та отримання необхідних звітів. Це забезпечується процесом ETL (extract, transform, load) – збирання даних з внутрішніх і зовнішніх джерел, їх обробка, очищення та структуризація.

Таблиця 1

Відмінності між сховищем даних та базою даних

Параметр	База даних	Сховище даних
Використання	Запис даних	Аналіз даних
Методи обробки	OLTP	OLAP
Кількість користувачів	Тисячі	Обмежена кількість
Випадки використання	Невеликі транзакції	Комплексний аналіз
Оптимізація	CRUD-операції	аналіз великих масивів
Тип даних	Деталізовані дані в реальному часі	Структуровані історичні дані

Незважаючи на відмінності в підходах та реалізаціях, усім сховищам даних властиві спільні риси (рис. 1).

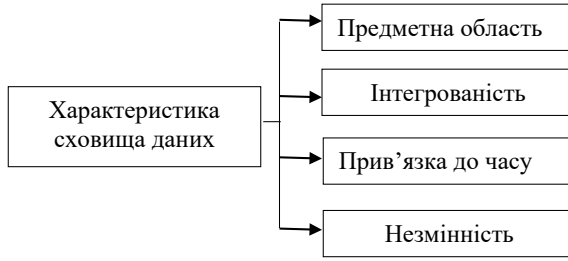


Рис. 1. Спільні характеристики сховища даних

3. Постановка задачі

Міграція неоптимальних для аналізу даних з реляційної бази даних у сховище часто є одним з найскладніших моментів так як цей процес вимагає очищення та структуризації "брудних" даних, для їх подальшого зберігання та аналізу. Метою роботи буде створення реляційної бази даних, яка буде імітувати роботу інтернет-магазину, створення відповідного сховища даних та дослідження швидкодії ETL процесу перенесення даних у сховище даних різними методами.

Дане дослідження можна розділити на декілька підзадач:

– Побудова реляційної бази даних та її наповнення неоптимальними даними;

– Побудова сховища даних;

– Створення ETL процесу за допомогою SSIS пакетів;
– Створення ETL процесу засобами T-SQL;
– Порівняння отриманих результатів та висновки.

4. Аналіз предметної області

Управління даними є дуже важливим моментом, оскільки дані, які створює організація, є дуже цінним ресурсом. Ефективне управління даними набуває все більшої важливості в останні роки, оскільки в сучасних організаціях відбувається значне збільшення обсягу інформації, що зберігається. Чіткий і структурований аналіз цих даних є важливою складовою при здобутті комерційного успіху організації.

У підприємств та організацій є питання та цілі. Щоб відповісти на ці запитання та відстежити результативність цих цілей, вони збирають необхідні дані, аналізують їх та визначають, які дії вжити для досягнення своїх цілей.

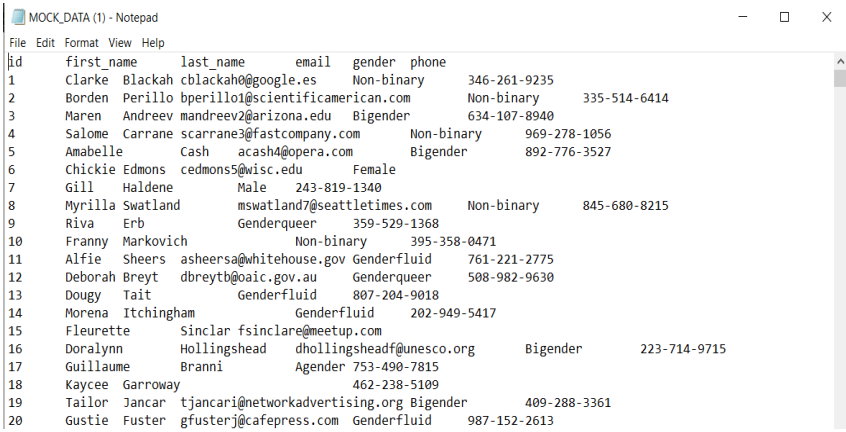
Об'єктом дослідження є процес міграції даних з бази даних яка напряму працює з бізнес-програмою (OLTP система) в сховище даних (OLAP система) для подальшого історичного зберігання та аналізу.

Отже, вхідними даними для дослідження будуть дані з реляційної бази даних, яка буде імітувати роботу онлайн магазину. Так як, дане дослідження буде проводились на тестових даних, набори вхідних даних умовно буде поділено на два типи – статичні і динамічні [4].

До статичних даних будуть відноситись такі дані, які не явно імітують бізнес діяльність організації. До переліку такого типу даних можна віднести інформацію про користувачів, перевізників, магазинів, категорії товарів та послуг, що надаються організацією. Для генерації таких даних буде використано сторонні ресурси з датасетами. Нижче буде наведено приклад вхідного файлу (рис. 2).

Даний тип файлів буде згенеровано для всіх статичних об'єктів та загружено у відповідні об'єкти в реляційну базу даних засобами SQLServerManagementStudio.

До динамічного типу відноситься такі дані/об'єкти, які моделюють бізнес активність організації. Це наприклад дані про продажі або поставки. Даний тип даних буде генеруватись на основі статичних об'єктів за допомогою збережених процедур. Випадковим чином зі статичних таблиць буде відбиратись атрибут, також випадково буде



MOCK_DATA (1) - Notepad

File Edit Format View Help

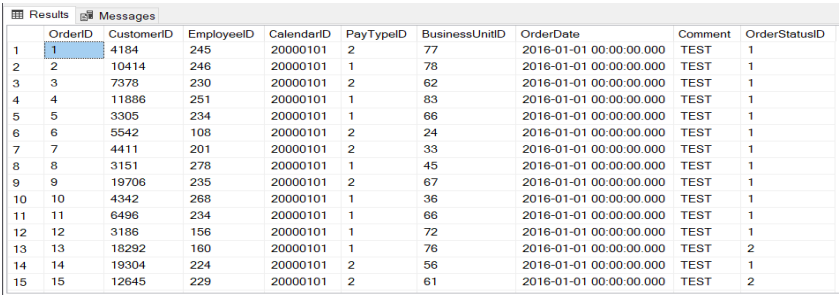
id	first_name	last_name	email	gender	phone
1	Clarke	Blackah	cblackah0@google.es	Non-binary	346-261-9235
2	Borden	Perillo	bperillo1@scientificamerican.com	Non-binary	335-514-6414
3	Maren	Andreev	mandreev2@arizona.edu	Bigender	634-107-8940
4	Salome	Carrane	scarrane3@fastcompany.com	Non-binary	969-278-1056
5	Amabelle	Cash	acash4@opera.com	Bigender	892-776-3527
6	Chickie	Edmons	cedmons5@wisc.edu	Female	
7	Gill	Haldene	Male	243-819-1340	
8	Myrilla	Swatland	mswatland7@seattletimes.com	Non-binary	845-680-8215
9	Riva	Erb	Genderqueer	359-529-1368	
10	Franny	Markovich	Non-binary	395-358-0471	
11	Alfie	Sheers	asheersa@whitehouse.gov	Genderfluid	761-221-2775
12	Deborah	Breyt	dbreytb@oaic.gov.au	Genderqueer	508-982-9630
13	Dougy	Tait	Genderfluid	807-204-9018	
14	Morena	Itchingham	Genderfluid	202-949-5417	
15	Fleurette	Sinclar	fsinclare@meetup.com		
16	Doralynn	Hollingshead	dhollingsheadf@unesco.org	Bigender	223-714-9715
17	Guillaume	Branni	Agender	753-490-7815	
18	Kaycee	Garroway		462-238-5109	
19	Tailor	Jancar	tjancari@networkadvertising.org	Bigender	409-288-3361
20	Gustie	Fuster	gfusterj@cafepress.com	Genderfluid	987-152-2613

Рис. 2. Приклад вхідних даних

калькулювати кількість бізнес-дій відповідного типу. Отримані дані будуть напряму вставленні в відповідні об'єкти. Нижче буде наведено приклад динамічних даних (рис. 3).

Оскільки об'єктом дослідження є оптимальний спосіб перенесення різних об'ємів даних з реляційної бази даних в сховище даних, вихідними даними будуть агреговані, структуровані дані подані у вигляді представлень, які будуть базуватись на попередньо з'єднаних та погрупованих таблицях.

Отримані представлення можна використовувати для проведення різного роду аналізу, який допоможе зрозуміти відповідні бізнес контек-



Results Messages

	OrderID	CustomerID	EmployeeID	CalendarID	PayTypeID	BusinessUnitID	OrderDate	Comment	OrderStatusID
1	1	4184	245	20000101	2	77	2016-01-01 00:00:00.000	TEST	1
2	2	10414	246	20000101	1	78	2016-01-01 00:00:00.000	TEST	1
3	3	7378	230	20000101	2	62	2016-01-01 00:00:00.000	TEST	1
4	4	11886	251	20000101	1	83	2016-01-01 00:00:00.000	TEST	1
5	5	3305	234	20000101	1	66	2016-01-01 00:00:00.000	TEST	1
6	6	5542	108	20000101	2	24	2016-01-01 00:00:00.000	TEST	1
7	7	4411	201	20000101	2	33	2016-01-01 00:00:00.000	TEST	1
8	8	3151	278	20000101	1	45	2016-01-01 00:00:00.000	TEST	1
9	9	19706	235	20000101	2	67	2016-01-01 00:00:00.000	TEST	1
10	10	4342	268	20000101	1	36	2016-01-01 00:00:00.000	TEST	1
11	11	6496	234	20000101	1	66	2016-01-01 00:00:00.000	TEST	1
12	12	3186	156	20000101	1	72	2016-01-01 00:00:00.000	TEST	1
13	13	18292	160	20000101	1	76	2016-01-01 00:00:00.000	TEST	2
14	14	19304	224	20000101	2	56	2016-01-01 00:00:00.000	TEST	1
15	15	12645	229	20000101	2	61	2016-01-01 00:00:00.000	TEST	2

Рис. 3. Приклад вхідних даних

OrderKey	OrderID	ProductID	DiscountID	FactPrice	CustomerID	EmployeeID	CalendarID	PayType	BusinessUnitID	OrderDate	OrderStatus
18	10	B003FM4XCG	1	10.71	4342	268	20000101	Card	36	2016-01-01	Done
19	10	B008KT8XOI	1	19.50	4342	268	20000101	Card	36	2016-01-01	Done
20	11	B00374PLQE	1	9.72	6496	234	20000101	Card	66	2016-01-01	Done
21	11	B005JOGO7A	1	74.11	6496	234	20000101	Card	66	2016-01-01	Done
22	11	B0046WO31A	1	30.55	6496	234	20000101	Card	66	2016-01-01	Done
23	12	B00004Z7JH	1	14.79	3186	156	20000101	Card	72	2016-01-01	Done
24	13	B003XH83N8	1	51.97	18292	160	20000101	Card	76	2016-01-01	Decline
25	13	B00CRJUG10	1	229.44	18292	160	20000101	Card	76	2016-01-01	Decline
26	13	B0009K3PFS	1	26.00	18292	160	20000101	Card	76	2016-01-01	Decline
27	14	B008LZZ0W4	1	51.94	19304	224	20000101	Cash	56	2016-01-01	Done
28	14	B003EYVOTK	1	4.54	19304	224	20000101	Cash	56	2016-01-01	Done
29	15	B004IVBCM2	1	28.28	12645	229	20000101	Cash	61	2016-01-01	Decline
30	16	B002V17LHQ	1	3.24	14016	268	20000101	Cash	36	2016-01-01	Done

Рис. 4. Представлення з даними продажі

VendorName	VendorID	Quantity
Skullcandy	SK1083	45
Kodak	KO1087	42
Synergy Digital	SY1099	42
CableWholesale	CA1035	41
Gino	GI1053	41
Asus	AS1018	39
Fenzer	FE1042	38
GE	GE1076	38
C2G	C21011	37
Panasonic	PA1017	37

Рис. 5. Приклад представлення з агрегованими даними

сти організації. Наведемо приклад представлення, яке містить інформацію про загальні, неагреговані дані продажів організації (рис. 4).

Також представлення можна використовувати для швидкого аналізу, роблячи необхідні агрегації даних і використовуючи необхідні формули всередині представлення (рис. 5).

5. Проектування системи

Для проведення досліджень необхідно створити інформаційну модель системи. Насамперед потрібно визначити які логічні сутності будуть представлені у системі, яка буде досліджуватись.

При інтеграції даних в сховище даних потрібно зробити декілька кроків денормалізації для забезпечення швидкодії роботи системи та для забезпечення цілісності даних.

Сутності, які мають однаковий бізнес напрямок будуть об’єднані в одні таблиці – таблиці фактів або вимірів. Дані, які при отриманні вхідних даних і вважалися статичними будуть згруповані і переведені у виміри, динамічні дані будуть переведені в таблиці фактів.

Розглянемо бізнес контекст кожного з об'єктів, які будуть містить в сховищі даних:

– FactOrders: таблиця фактів, яка містити дані про продажі у організації і буде містити два рівня гранулярності – рівень наявності замовлення та його деталізація;

– FactConsigments: таблиця фактів, яка міститьінформацію про поставки в середині мережі. Буде представлено два рівня гранулярності – факт поставки і деталізація;

– FactShipments: таблиця фактів, яка містить дані про поставки, які будуть надходити від зовнішніх постачальників. Буде представлено два рівня гранулярності – факт поставки і його деталізація;

– Dim_Products: таблиця виміру, яка містить інформацію про товари, які наявні в організації та виступає словником і системі для детальної інформації про товари;

– Dim_Discounts: таблиця виміру, яка містить інформацію про знижки, які існували або є діючими та надає детальну інформацію про термін дії, статус та розмір знижки;

– Dim_BusinessUnits: таблиця виміру, яка містить інформацію про точки збуту або склади, які існували або діють в межах організації. Надає інформацію про місце знаходження та статус бізнес одиниці.

– Dim_EmployeeAndRoles: таблиця виміру, яка містить інформацію про працівників та їхні ролі, надає дані про історію переміщень працівників між позиціями та історичні дані працівників.

Враховучи вище наведену структуру і опираючись на бізнес потреби організації формується структура сховища даних.

6. Засоби розробки

Метою даного дослідження є визначення оптимальних засобів для внутрішнього переміщення різних об'ємів даних, які б потенційно використовувались організацією для проведення аналізу з метою покращення діяльності [6].

Основними способами переміщення даних, які будуть розглянуті у даній роботі було вибрано наступні засоби:

– переміщення даних за допомогою внутрішніх засобів MSSqlServer та T-SQL;

– переміщення даних за допомогою генерації динамічного BIML скріпта, який буде створювати SSIS пакет для інтеграції даних;

SQL Server – це механізм обробки даних, запроваджений корпорацією Майкрософт. Він забезпечує середовище, що використовується для створення та управління базами даних. Це забезпечує безпечне та ефективне зберігання. Він надає інші компоненти та послуги, які підтримують платформу бізнес-аналітики для формування звітів та аналізу даних [7].

SQL Server характеризується такими особливостями як:

– Продуктивність: SQL Server працює дуже швидко.

– Надійність і безпека: SQL Server надає шифрування даних.

– Простота: З даною СКБД відносно легко працювати і вести адміністрування.

SQL Server містить ряд компонентів. Кожен компонент надає певні послуги та підтримку клієнтам, підключеним до сервера.

На наступній схемі показані компоненти SQL Server (рис. 6):

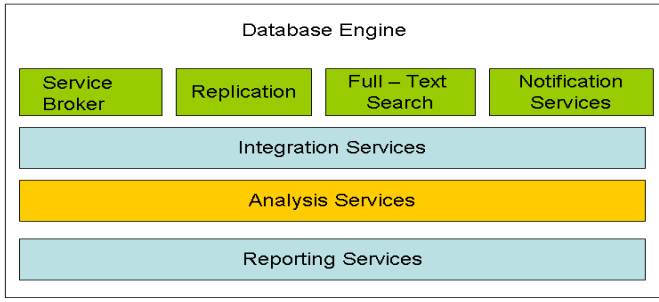


Рис. 6. Компоненти SQLServer

Виходячи з різноманітності технології і засобів, які надаються при використанні SQLServer, можна зробити висновок що даний механізм обробки даних є найоптимальніший і найбільш конкурентно здатний у порівнянні зі своїми конкурентами.

Розглянемо переваги основних засобів, які надаються даним механізмом обробки даних:

– Database Engine:

Компонент забезпечує підтримку зберігання запитів, обробки та захисту даних на сервері. Це дозволяє користувачеві створювати та

керувати об'єктами бази даних. Такі фонові послуги, як реплікація, повнотекстовий пошук, сервіси сповіщень і т.д. надаються даним компонентом

– Сервіси інтеграцій(SSIS):

Даний засіб дозволяє збирати та інтегрувати різноманітні дані у єдиному форматі в сховище даних.

– Сервіси аналізу (SSAS):

Сховища даних призначені для полегшення складання звітності та аналізу. Додатки широко використовують дані засоби в аналітичних цілях. Програми, що використовуються для цієї мети, відомі як програми BI.

– Сервіси звітування (SSRS):

Засоби надають підтримку для створення повних звітів про дані в механізмі баз даних у сховищі даних. Ці послуги надають набір інструментів, які допомагають створювати звіти у різних форматах та керувати ними.

Отже, використання SQLServer у дослідженнях надають нам різноманітність у виборі засобів для роботи з даними. Нижче перелічимо ще декілька переваг даного механізму:

– Масштабованість: це дозволяє розподіляти дані у великих таблицях у різні групи файлів. Сервер може одночасно отримувати доступ до груп файлів;

– Сервісно-орієнтована архітектура: це забезпечує розподілену асинхронну структуру додатків для великомасштабних додатків;

– Високий рівень безпеки: реалізовано високий рівень безпеки шляхом додавання політики для входу та паролів;

– Регулятор ресурсів: використовується для управління робочим навантаженням сервера шляхом розподілу та управління ресурсами.

T-SQL або Transact-SQL – це розширення мови структурованих запитів (SQL) від Microsoft, яке має додаткові транзакційні структури або аспекти з SQL і використовується для роботи з будь-якою з реляційних баз даних на основі сервера SQL [8].

Переваги T-SQL:

– Модульність: великий перехід технологій у бік мікросервісної та модулізованої архітектури. Це впливає на швидкодію роботи системи та допоможе зменшити залежність частин системи один від одного.

– Безпека: дані зберігаються на сервері з власною мірою безпеки як комерційна таємниця. Захист побудований на основі допоміжних знань про реєстрацію та транзакцій у навколишньому середовищі, що сприяє надійності.

– Ефективність: мінімізує трафік на сервері. завдання, що виконуються з даними, обробляються з мінімальними накладними витратами при передачі в програмі. Таким чином, складні нетривіальні завдання можна легко вирішити за допомогою T-SQL.

BIML – це мова, базована на XML, що дозволяє повністю моделювати рішення BI та використовується для автоматизації створення ETL процесів [9].

BIML найкраще працює, якщо вам потрібно створити кілька пакетів SSIS, які всі мають однаковий шаблон. Іншими словами, якщо всі пакети мають приблизно однакову мету та структуру, але, змінюються лише таблиця джерела та призначення, тоді використанням BIML допоможе автоматизувати більшу частину ручної роботи.

7. Опис програмної реалізації

Для створення системи було обрано SQLServer, як один з найоптимальніших механізмів обробки даних. Розробка розпочалась з того, що необхідно створити OLTP систему та сховище даних, зобразити потік даних, та поєднати системи інтеграційним шаром [10].

Для відображення потоку даних було обрано DataFlow діаграму. Даний тип UML-діаграми допоможе зобразити OLTPсистему та сховище даних та візуалізувати інтеграційні зв'язки між цими двома системами. Data Flow діаграма зображена на рис. 7.

Першим компонентом системи є OLTP система. Дана система моделює діяльність інтернет магазину, тобто всі логічні операції будуть відбуватися на стороні реляційної бази даних. Цілісність даних забезпечується ключами та тригерами.

Для генерації даних у системі було створено наступний перелік збережених процедур [11]:

– dbo.STP_GenerateDataForConsignmentDetails
– dbo.STP_GenerateDataForConsignments
– dbo.STP_GenerateDataForDiscounts
– dbo.STP_GenerateDataForShipments

– dbo.STP_GenerateDataForShipmentDetails
– dbo.STP_GenerateDataForOrders
– dbo.STP_GenerateDataForOrderDetails

Вище подані процедури мають схожу логіку генерації даних. Вибираючи випадкове значення з таблиць, які складають логічну сутність таблиці, формується рядок і зберігається у таблиці. Таким чином буде отримано моделювання взаємодії користувачів з системою. Нижче

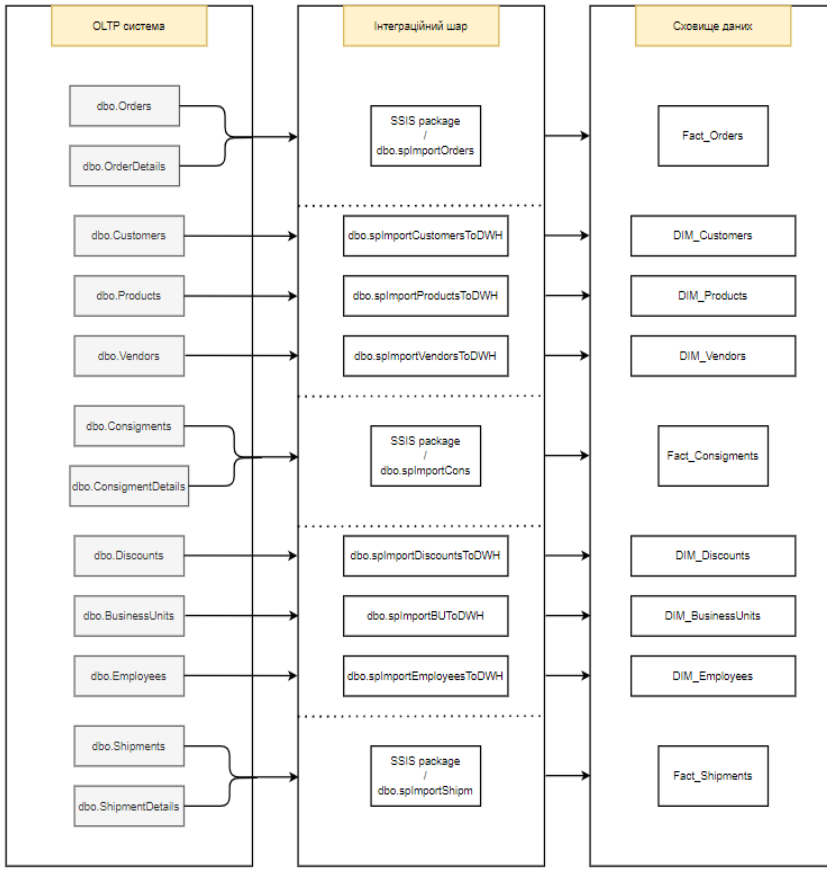


Рис. 7. DataFlow діаграма

буде подано рисунок (рис. 8) на якому буде зображено кількість записів у кожній з таблиць, які будуть використовуватись у подальшому досліджені.

TableName	RowCount
[dbo].[OrderDetails]	1642470
[dbo].[Orders]	821436
[dbo].[Customers]	20538
[dbo].[ConsignmentDetails]	16913
[dbo].[Consignments]	3087
[dbo].[Shipments]	2904
[dbo].[Employees]	323
[dbo].[Discounts]	201
[dbo].[BusinessUnits]	84

Рис. 8. Кількість записів у таблицях

При генерації даних, також було встановлено відсоток невалідних даних. Тобто це так звані пошкодженні рядки, які мають відсутні певні логічні значення. Такі рядки будуть відсіюватись у інтеграційному шарі, так як не несуть ніякої аналітичної цінності.

Отже, на даному етапі реалізації отримано готову OLTP систему, наповнену реалістичними даними. Далі розглянемо побудову сховища даних та його структуру.

Сховище даних є одним з компонентів системи, де будуть зберігатись очищені і структуровані дані. Побудова сховища даних необхідне для подальшого зберігання історичних даних. Також наявність структурованого сховища надає можливість відокремлювати певні вітрини даних та проводити аналіз.

Нижче буде подано діаграму зав'язків, за принципом якої буде реалізована система (рис. 9).

Розглянемо дану структуру в межах реалізованої системи. Сорсовими системами в системі виступає вище описана OLTP система [12].

Інтеграційний шар буде розглянуто далі, так як буде проведено дослідження різних інтеграційних засобів на різних об'ємах даних. Отже, необхідно розглянути як в системі реалізовано аналітичний шар. В реалізованій системі можна виділити три основних бізнес напрямки: Продажі; Поставки; Переміщення.

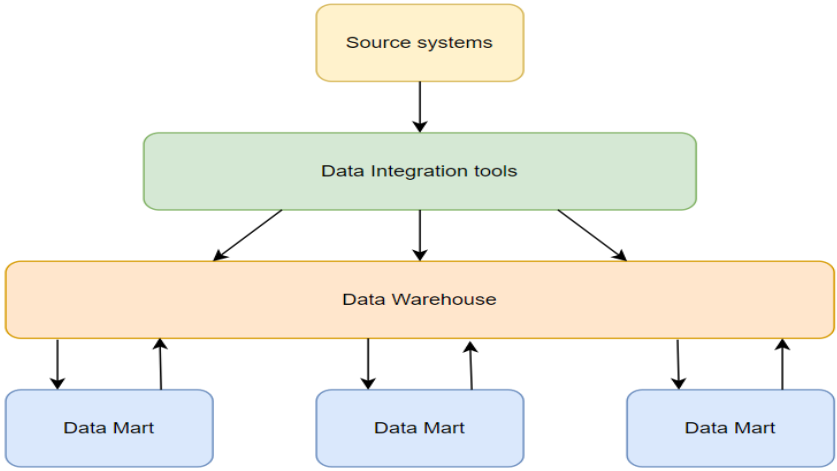


Рис. 9. Діаграма зав'язків системи

На рисунку нижче (рис. 10) буде накладено структуру аналітичного шару системи на рис. 13, що був описаний раніше.

Для побудови повноцінної вітрини даних до визначених основних таблиць було додано таблиці вимірів, які були описані вище (рис. 11).

Також у даному сховищі даних реалізовані певні представлення, які допоможуть швидко отримати певну аналітику.

Найважливішим компонентом у системі буде інтеграційний шар, так як він буде виконувати не тільки роль переміщення даних, але й очищення і структуризації, тобто ETL процес [13].

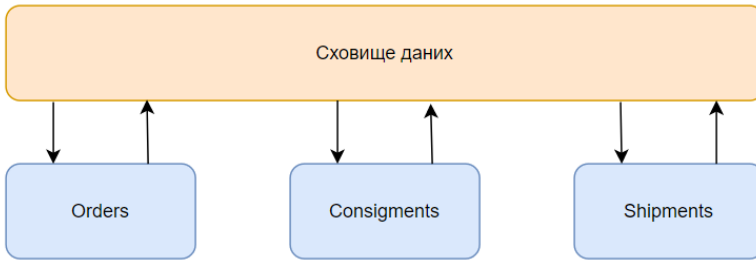


Рис. 10. Реалізація аналітичного шару

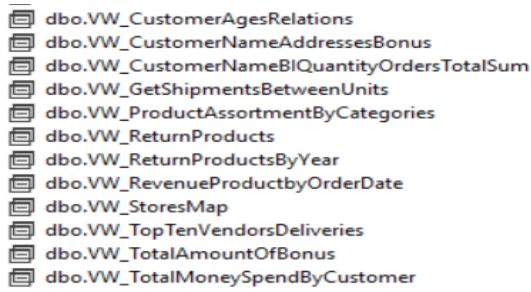


Рис. 11. Аналітичні представлення у сховищі даних

Виконуючи даний процес проведемо порівняння між двома способами переміщення, для визначення найоптимальнішого. Серед розглянутих, будуть наступні способи: SSIS пакети; збережені процедури на T-SQL.

Спершу, необхідно обрати об'єкт бази даних на якому будуть проводитись дослідження. Проаналізувавши об'єкти які були представлені в системі було обрано бізнес напрям замовлень, так як у даних таблицях є не тільки числові, але й текстові поля, що можуть суттєво впливати на час переміщення даних (рис. 12, 13).

Також дані об'єкти дають можливість зробити витяг різної кількості даних так як при повному переміщенні даних об'єктів буде опрацьовано близько 2.5 мільйона записів.

Отже, дослідження швидкодії ETL процесу буде проводитись на наборах даних різної розмірності.

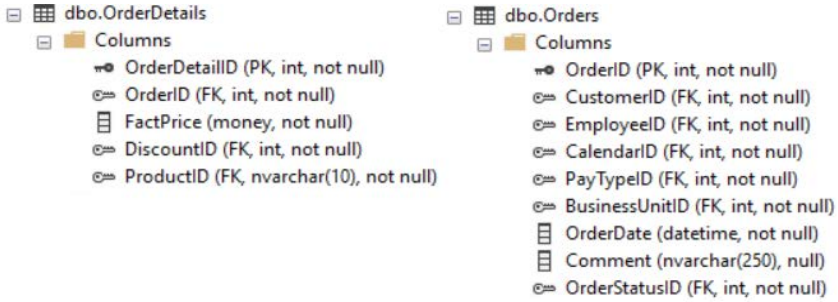


Рис. 12. Структура замовлень в OLTP системі

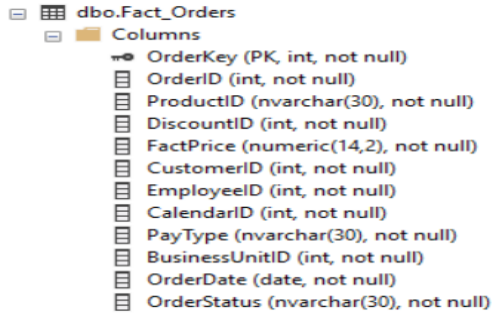


Рис. 13. Структура замовлень в сховищі даних

Спершу проведемо дослідження використовуючи збережені процедури на T-SQL. Для здійснення даного переміщення було створено процедуру, в якій, при переміщені даних, будуть відбуватись певні фільтрації [14].

Проведемо експерименти на обраних об'ємах даних, поступово збільшуючи кількість записів. Результати відобразимо у вигляді таблиці (табл. 2).

Таблиця 2

Час на переміщення даних за допомогою T-SQL

Кількість записів	Час виконання(мс)
1000	176
10000	203
20000	220
50000	356
100000	496
200000	816
500000	2336
1000000	4010
1500000	6203
2000000	7850

Аналізуючи результати, які були отримані під час експериментів та занесені у таблицю 3, видно, що зі збільшенням об'єму інформації що опрацьовується збільшується час роботи.

Рис. 14. Графік залежності часу до об'єму інформації обробленої за допомогою T-SQL

Зобразимо у вигляді лінійного графіка залежність зміни часу переміщення від об'єму даних, що передається (рис. 14).

Тепер проведемо такі ж експерименти, тільки з використанням SSIS пакетів. При використанні SSDT для роботи з даними пропонує на вибір параметр UseFastLoadIfAvailable. Розглянемо два варіанти із застосуванням даного параметру та без нього.

– Fast Load – <UseFastLoadIfAvailable=»false»>

– Plain Load – <UseFastLoadIfAvailable=»true»>

Результати будуть відображені у таблиці нижче (табл. 3).

З отриманих результатів видно, що використання Plain Load починаючи з 10000 записів різко збільшує час виконання і у порівнянні з Fast Load працює довше в середньому у 50 разів довше. Це випливає з того, що при використанні PlainLoad опрацьовується не весь об'єм даних, а дані подаються частково – по 9000 рядків. Тільки після опрацювання однієї порції, наступна перейде у роботу. З опцією FastLoad дані опрацьовуються динамічно, що значно впливає на швидкість роботи.

На рисунках нижче зобразимо відношення часу виконання до об'єму даних, що опрацьовуються для кожного з досліджуваних способів використання SSIS пакетів (рис. 15, 16).

Таблиця 3

Час для переміщення даних двома варіантами використання SSIS

Кількість записів	Час виконання(мс)	
	Fast Load	Plain Load
1000	156	343
10000	160	2266
20000	203	4390
50000	328	13641
100000	516	21515
200000	860	43422
500000	1859	149641
1000000	3547	337109
1500000	5391	533922
2000000	6985	671078

З вище поданих графіків видно, що використання FastLoad не тільки дає значний приріст в часі виконання, але і зростання даного часу у порівнянні з PlainLoad є більш стабільним. З рисунку 17 видно що після різкого збільшення даних час виконання збільшується не

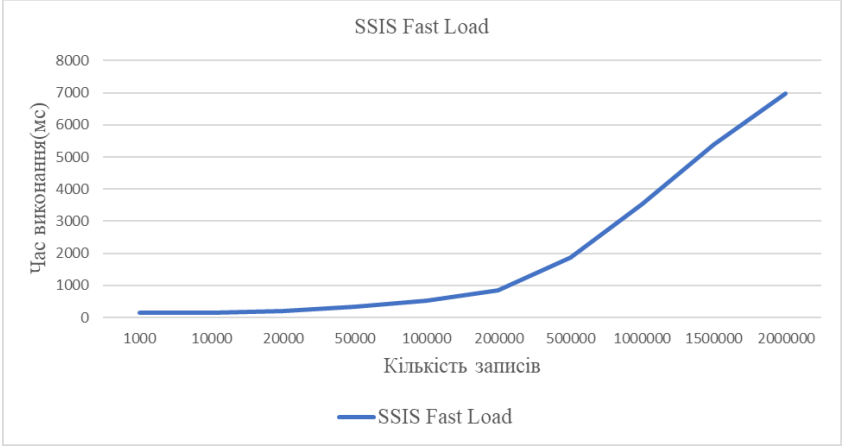


Рис. 15. Графік порівняння швидкості часу завантаження за допомогою SSIS пакетів використовуючи Fast Load

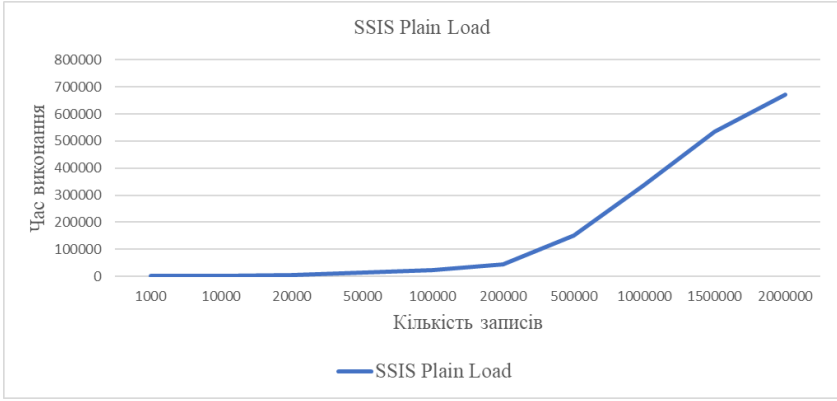


Рис. 16. Графік порівняння швидкості часу завантаження за допомогою SSIS пакетів використовуючи PlainLoad

пропорційно, в той час як використання FastLoad дає лінійний приріст в часі відносно об'єму даних.

Зобразимо на одному рисунку відношення для даних двох операторів, для чіткішого розуміння швидкості приросту часу з використанням PlainLoad (рис. 18). Як і було описано вище, PlainLoad не є

Рис. 17. Графік порівняння швидкості часу завантаження за допомогою SSIS пакетів використовуючи Fastта PlainLoad

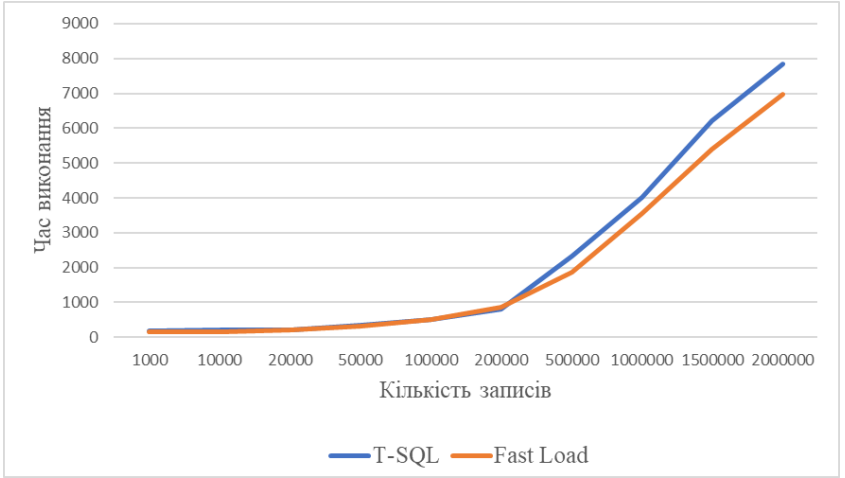


Рис. 18. Графік порівняння швидкості часу завантаження за використанням T-SQLта FastLoad

оптимальним способом для опрацювання і перенесення даних, так як з рисунку вище (рис. 19) видно, стрімке збільшення часу на відносно не великій кількості даних у порівнянні з FastLoad.

Отже, після проведення досліджень було виявлено що створення SSIS пакетів з використанням FastLoad є найбільш оптимальним способом для опрацювання і перенесення даних з використанням даного засобу.

На даному етапі необхідно провести порівняння між T-SQL та FastLoad щоб зрозуміти, який спосіб з усіх розглянутих є найбільш оптимальним.

Зобразимо у вигляді таблиці 4 раніше отриманні значення під час експериментів для способі що порівнюються

З вище поданої таблиці можна зробити висновок, що на будь якому об'ємі даних використання SSIS пакетів з FastLoad є більш швидкодійним аніж використання засобів DatabaseEngine та T-SQL.

Явний приріст по продуктивності починається після 200000 записів де FastLoad є в середньому в 1.25-1.3 рази ефективнішим та швидкодійним у порівнянні з T-SQL.

Таблиця 4

Порівняння часу виконання T-SQL та FastLoad

Кількість записів	T-SQL	FastLoad
	Час виконання(мс)	Час виконання(мс)
1000	176	156
10000	203	160
20000	220	203
50000	356	328
100000	496	516
200000	816	860
500000	2336	1859
1000000	4010	3547
1500000	6203	5391
2000000	7850	6985

На меншій кількості записів різниця не є настільки суттєвою і не перевищує приріст у 1.1 рази.

Зобразимо дане порівняння на графіку для візуального розуміння швидкості приросту часу до об'єму даних (рис. 18).

Отже, з даного графіку видно, що перевагу все ж таки має FastLoad, так як на відносно великих об'ємах даних є приріст по продуктивності, а на об'ємах даних до 200000 приріст не є настільки суттєвим.

Незважаючи на те, що на відносно не великих об'ємах даних FastLoad показав кращі результати, рекомендацією би було використовувати T-SQL, так як розробка цими засобами є більш простішою та не вимагає додаткових засобів, так як це потрібно при розробці SSIS пакетів. Але необхідно враховувати всі фактори і бізнес-потреби для прийняття оптимального рішення [15].

Створена система передбачає своє керування використовуючи засоби SQLServerManagementStudio. Керування даними передбачено як для OLTP системи так і для сховища даних. Було реалізовано ряд об'єктів бази даних для отримання швидкого аналізу даних.

Розглянемо об'єкти, якими є змога керувати користувачу. Для додавання, оновлення або видалення інформації в системі реалізовані процедури які будуть відповідати за дані операції. Також реалізовано декілька об'єктів, які відповідають за бізнес процеси, такі як звільнення працівника і т.д. Нижче буде наведено їх перелік (рис. 19).

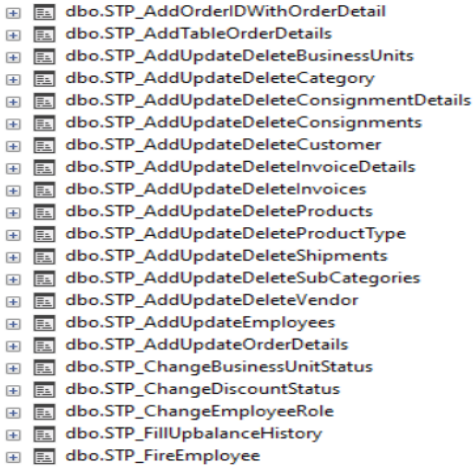


Рис. 19. Процедури для керування даними в системі

У кожній з даних процедур передбачено передавання інформації яку потрібно змінити і на основі отриманих вказівок буде виконана певна дія.

Розглянемо роботу однієї з них на прикладі додавання та видалення нового типу продуктів у систему. Перевіримо які існуючі типи вже є, для цього переглянемо дані у таблиці dbo.Products. Для обмеження кількості записів у вибірку візьмемо лише ті, що починаються на ‘W’ (рис. 20).

	ProductTypeID	TypeName
1	WAL13174	Wall Plates & Connectors
2	WAL14042	Wall Chargers
3	WAT10056	Water Cooling Systems
4	WEA14900	Weather Radios
5	WIR16584	Wireless Jack Systems
6	WIR18884	Wireless Access Points

Рис. 20. Вибірка типів продуктів

На даний момент у системі представлено 6 типів продуктів які задовольняють умови нашої вибірки. Припустимо з’явилась бізнес-потреба добавити ще один тип. Для цього скористаємось процедурою dbo.STP_AddUpdateDeleteProductType.

При виклику процедури слід вказати необхідну дію та дані, які потрібно добавити та після успішного виконання даної задачі у системі з'явиться необхідний запис (рис. 21).

	ProductTypeID	TypeName
1	WAL13174	Wall Plates & Connectors
2	WAL14042	Wall Chargers
3	WAT10056	Water Cooling Systems
4	WEA14900	Weather Radios
5	WIR16584	Wireless Jack Systems
6	WIR18884	Wireless Access Points
7	WIRTEST	Wireless Charges

Рис. 21. Модифікована вибірка типів продуктів

За допомогою виклику даної процедури, тільки з передачею іншого параметра дії можна взаємодіяти з даними в системі без додаткових засобів і не надаючи користувачам доступ до початкових таблиць.

Вище був розглянутий один з найпростіших варіантів взаємодії з OLTP системою. Натомість взаємодія зі сховищем даних буде відріз-

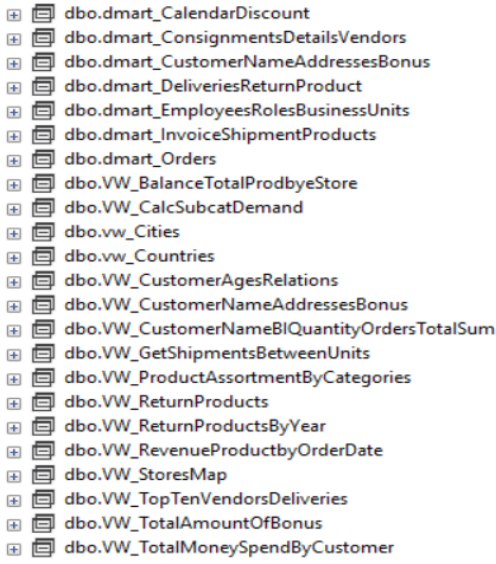


Рис. 22. Аналітичні представлення у сховищі даних

нятись від вище описаної. Так як у сховищі даних зберігаються вже трансформовані, архівні дані.

Користувач не повинен мати доступ на їх модифікацію, але повинен мати можливість отримувати дані для аналізу. Для цього було створено певний перелік представлень, які будуть слугувати вихідними даними системи.

За допомогою даних представлень користувач буде мати можливість або самостійно отримати агреговані або чисті дані для аналізу, або передати інформацію з даних представлень в сторонні програми для створення BI-рішень, такі як PowerBI, для проведення більш глибшого аналізу (рис. 22).

8. Висновки

Отже, в результаті дослідження було створено трьох шарову систему яка складається з OLTP системи, як транзакційного шару, ETL процесів за допомогою засобів DatabaseEngine та SSIS пакетів, як інтеграційного шару та сховища даних побудованого за гібридним принципом, як аналітичного шару. Реалізована система є аналогом новітньої self-service системи, так як може забезпечувати свої бізнес потреби без сторонніх засобів.

Було розглянуто різницю між реляційною базою даних та сховищем даних, наявні підходи до побудови сховищ даних, їх переваги та недоліки. Було проаналізовано предметну область та описано концепцію побудови сховища даних та OLTP системи. Також було виконано усі завдання, які були описані в постановці, тобто реалізовано інформаційну self-service систему.

При проектуванні системи було створено інформаційну модель реляційної бази даних та сховища даних, описано логічні сутності які представлені в системі та сформовано бізнес потреби кожного з об'єктів. Описано структуру вхідних та вихідних даних та запропоновано варіанти їх подальшого використання.

Для коректної роботи програми було сформовано та описано вимоги до програмного та технічного забезпечення, сформована записка з керуванням для користувача з детальним описом основного функціоналу системи.

Розроблену систему можна використовувати як аналог для взаємодії з даними на підприємстві. Використані інтеграційні засоби можна

застосувати для перенесення будь якого типу даних і це допоможе краще аналізувати структуровані дані для визначення основних бізнес потреб системи.

Список літератури:

1. Wang Y., Wu X. (2007) Heterogeneous spatial data mining based on grid, Lecture notes in computer science, vol. 4683, pp. 503–510.
2. Veres O., Shakhovska N. (2015) Elements of the formal model big date. In: The 11th Intern. conf. Perspective Technologies and Methods in MEMS Design (MEMSTEH), pp. 81–83.
3. Agrawal R., Gehrke J., Gunopulos D., Raghavan P. (2005) Automatic subspace clustering of high dimensional data. In: Data mining knowledge discovery, vol. 11(1), pp. 5–33.
4. Guimei L., Jinyan L., Sim K., Limsoon W. (2007) Distance based subspace clustering with flexible dimension partitioning. In: Proc. of the IEEE 23rd Intern. conf. on digital object identifier, vol. 15, pp. 1250–1254.
5. Procopiuc C.M., Jones M., Agarwal P.K., Murali T.M. (2002) A Monte Carlo algorithm for fast projective clustering. In: ACM SIGMOD Intern. conf. on management of data, pp. 418–427.
6. Boyko N. (2016) A look trough methods of intellectual data analysis and their applying in informational systems. In: Scientific and Technical Conference "Computer Sciences and Information Technologies (CSIT), 2016 XIth International, pp. 183–185.
7. Boyko N. (2017) Advanced technologies of big data research in distributed information systems. Radio Electronics, Computer Science, Control, vol. 4, pp. 66–77.
8. Boyko N. (2018) Machine learning on data lake. Monograph, p. 189.
9. Boyko N., Shakhovska N., Pukach P. (2018) The Information Model of Cloud Data Warehouses. In: International Conference on Computer Science and Information Technologies, CSIT 2018, pp. 182–191.
10. Shakhovska N., Vovk O., Hasko R., Kryvenchuk Y. (2018) The Method of Big Data Processing for Distance Educational System. In: Conference on Computer Science and Information Technologies, pp. 461–473.

References:

1. Wang Y., Wu X. (2007) Heterogeneous spatial data mining based on grid, Lecture notes in computer science, vol. 4683, pp. 503–510.
2. Veres O., Shakhovska N. (2015) Elements of the formal model big date. In: The 11th Intern. conf. Perspective Technologies and Methods in MEMS Design (MEMSTEH), pp. 81–83.
3. Agrawal R., Gehrke J., Gunopulos D., Raghavan P. (2005) Automatic subspace clustering of high dimensional data. In: Data mining knowledge discovery, vol. 11(1), pp. 5–33.

4. Guimei L., Jinyan L., Sim K., Limsoon W. (2007) Distance based subspace clustering with flexible dimension partitioning. In: Proc. of the IEEE 23rd Intern. conf. on digital object identifier, vol. 15, pp. 1250–1254.

5. Procopiuc C.M., Jones M., Agarwal P.K., Murali T.M. (2002) A Monte Carlo algorithm for fast projective clustering. In: ACM SIGMOD Intern. conf. on management of data, pp. 418–427.

6. Boyko N. (2016) A look trough methods of intellectual data analysis and their applying in informational systems. In: Scientific and Technical Conference "Computer Sciences and Information Technologies (CSIT), 2016 XIth International, pp. 183–185.

7. Boyko N. (2017) Advanced technologies of big data research in distributed information systems. Radio Electronics, Computer Science, Control, vol. 4, pp. 66–77.

8. Boyko N. (2018) Machine learning on data lake. Monograph, p. 189.

9. Boyko N., Shakhovska N., Pukach P. (2018) The Information Model of Cloud Data Warehouses. In: International Conference on Computer Science and Information Technologies, CSIT 2018, pp. 182–191.

10. Shakhovska N., Vovk O., Hasko R., Kryvenchuk Y. (2018) The Method of Big Data Processing for Distance Educational System. In: Conference on Computer Science and Information Technologies, pp. 461–473.