

УДК 004.8

DOI <https://doi.org/10.32782/tnv-tech.2025.2.20>

МЕТОДИКА ОПТИМІЗАЦІЇ ПРОЦЕСІВ ОБРОБКИ І ПЕРЕНЕСЕННЯ ДАНИХ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

Сологуб В. Р. – аспірант кафедри
електронних засобів інформаційно-комп'ютерних технологій
Національного університету «Львівська політехніка»
ORCID ID: 0000-0003-1553-530X

Пашкевич В. З. – доктор технічних наук, доцент кафедри
електронних засобів інформаційно-комп'ютерних технологій
Національного університету «Львівська політехніка»
ORCID ID: 0000-0002-6849-652X

Бешлей М. І. – доктор технічних наук, професор кафедри
інформаційно-комунікаційних технологій
Національного університету «Львівська політехніка»
ORCID ID: 0000-0002-7122-2319

З огляду на стрімкий розвиток цифрових технологій, компанії дедалі частіше зіштовхуються з необхідністю обробки великих обсягів даних, що ускладнює ефективне виконання бізнес-аналітики на основі операційних баз даних. Виконання аналітичних операцій безпосередньо на базах даних, призначених для щоденної бізнес-діяльності, не завжди є оптимальним рішенням. Нормалізована реляційна база даних, яка використовується у таких випадках, може складатися з десятків або навіть тисяч таблиць, кожна з яких має свої унікальні назви та рівень деталізації. У результаті процес пошуку та вилучення потрібних даних для конкретного аналізу може бути складним і тривалим. Крім того, сучасні компанії часто застосовують декілька бізнес-додатків, що працюють із різними базами даних. Об'єднання таких архітектурно несумісних систем з метою проведення аналітики може призводити до зниження якості даних. Саме тому, у роботі розроблено методику оптимізації процесів обробки та перенесення даних на попередньо змодельованому сховищі даних.

Основний акцент дослідження було зроблено на розробці реляційної бази даних та сховища даних за гібридним підходом, налаштуванні ETL-процесу між реляційною базою даних і сховищем даних за допомогою пакетів SSIS, а також порівнянні ефективності перенесення даних на таблицях різної розмірності. Для оцінки ефективності запропонованої методики враховувалися такі критерії, як швидкість перенесення заданого обсягу даних з імітованої реляційної системи до аналітичної, а також мінімізація використання додаткових інструментів і людського залучення під час виконання цього процесу.

Дослідження включало розробку реляційної моделі даних, її наповнення неоптимальними даними, створення аналітичної системи та налаштування процесів перенесення й обробки даних із використанням різних засобів і параметрів. Результати дослідження показали ефективність використання типу SSIS Fast Load для перенесення та обробки різних обсягів даних, що виявилось на 11 % швидшим за використання внутрішніх засобів T-SQL і на 98 % швидшим за використання типу SSIS Plain Load.

Ключові слова: сховище даних, ETL-процес, аналітична система, ефективність перенесення даних, обробка даних.

Solohub V. R., Pashkevych V. Z., Beshley M. I. A methodology for optimizing data processing and transfer in information systems

Given the rapid development of digital technologies, companies increasingly face the need to process large volumes of data, which complicates the efficient execution of business analytics based on operational databases. Performing analytical operations directly on databases intended for daily business activities is not always an optimal solution. A normalized relational database

used in such cases may consist of dozens or even thousands of tables, each with its own unique names and levels of detail. As a result, the process of locating and retrieving the necessary data for specific analyses can be complex and time-consuming.

Moreover, modern companies often use multiple business applications that interact with different databases. The integration of such architecturally incompatible systems for analytical purposes can lead to a decline in data quality. Therefore, this study proposes a methodology for optimizing data processing and transfer on a previously modeled data warehouse.

The main focus of the research was on designing a relational database and a data warehouse using a hybrid approach, configuring the ETL process between the relational database and the data warehouse using SSIS packages, and comparing the efficiency of data transfer across tables of varying sizes. To assess the effectiveness of the proposed methodology, the following criteria were considered: the speed of transferring a specified volume of data from a simulated relational system to the analytical environment, as well as the minimization of the use of additional tools and human involvement during the process.

The study included the development of a relational data model, its population with suboptimal data, the creation of an analytical system, and the configuration of data transfer and processing processes using various tools and parameters. The results demonstrated the effectiveness of using the SSIS Fast Load type for transferring and processing data of different volumes, showing an 11 % performance improvement over the use of native T-SQL tools and a 98 % improvement compared to the SSIS Plain Load type.

Key words: data warehouse, ETL process, analytical system, data transfer efficiency, data processing.

Постановка проблеми. З розвитком цифрових технологій та зростанням потреб у глибокій бізнес-аналітиці, системи підтримки прийняття рішень набули широкого поширення. Проте у багатьох випадках доступ до даних, що зберігаються в операційних базах даних на основі систем OLTP (Online Transaction Processing), виявився неефективним. Це пояснюється тим, що такі системи перебувають під значним навантаженням і містять велику кількість взаємопов'язаних таблиць, що ускладнює швидкий та зручний доступ до необхідної інформації. Як наслідок, час отримання даних та їх якість часто є недостатніми для виконання повноцінних аналітичних завдань. У відповідь на зазначені виклики була запропонована концепція сховищ даних (Data Warehouse) для інформаційних систем, що об'єднують інтегровані набори даних з різних джерел [1]. Сховища даних орієнтовані на забезпечення ефективного середовища для аналізу та прийняття рішень, що є особливо важливим для сучасних бізнес-процесів.

Щоб зрозуміти переваги цього підходу, варто розрізнити функціональну суть бази даних і сховища даних. Бази даних, як правило, є організованими колекціями даних, які підтримують поточні транзакції, тоді як сховища даних інтегрують інформацію з різних джерел, зберігають історичні дані та орієнтовані на аналітичну обробку [2].

У сучасному корпоративному середовищі компанії активно застосовують велику кількість бізнес-додатків, що взаємодіють із різними базами даних. Об'єднання таких архітектурно несумісних систем для аналітичної обробки інформації створює додаткові труднощі, зокрема впливає на якість даних, призводить до появи розбіжностей та ускладнює отримання цілісної аналітичної картини [3].

У зв'язку з цим постає проблема розробки ефективної методики оптимізації процесів обробки та перенесення даних, яка дозволить підвищити якість, швидкість і надійність аналітичних досліджень, мінімізуючи залежність від архітектурних обмежень джерел даних та обмежень ETL-процесів.

Аналіз останніх досліджень і публікацій. Реляційні бази даних базуються на системах OLTP. OLTP (онлайн-обробка транзакцій) відноситься до системи обробки даних, орієнтованої на обробку транзакцій. Ці системи зазвичай зберігають інформацію, яка використовується підприємствами щодня. Результатом їхньої роботи є швидкі, ефективні запити та поточні та точні дані. Бази даних

OLTP оптимізовано для швидкого виконання операцій CRUD (створення, читання, оновлення та видалення). Однак більш складні аналітичні запити можуть значно знизити їх продуктивність. Ці бази даних розроблені для підтримки тисяч або більше користувачів одночасно без будь-якого зниження продуктивності [4].

Сховища даних, у свою чергу, є частиною систем OLAP. OLAP (онлайн-аналітична обробка) означає систему обробки даних, зосереджену на аналізі даних і прийнятті рішень, а не на продуктивності та щоденному використанні. Багато систем OLAP пов'язані з рішеннями бізнес-аналітики (BI), які полегшують пошук необхідної інформації. Ці системи можуть містити попередньо зведені дані та велику кількість історичних записів. Сховища даних OLAP можуть підтримувати лише відносно обмежену кількість одночасних користувачів. Оскільки рішення для сховищ даних включають складніші запити, які циркулюють між кількома сховищами даних, вони за своєю суттю вимагають більше ресурсів і тому менш масштабовані, ніж бази даних [5].

Підприємства та організації працюють з конкретними питаннями та цілями. Щоб відповісти на ці запити та контролювати прогрес у досягненні цих цілей, вони збирають відповідні дані, аналізують їх і визначають необхідні дії для досягнення бажаних результатів. При проектуванні структури сховища даних найчастіше використовуються два методи: схема «Зірка» та схема «Сніжинка». Основна відмінність між схемою «Зірка» та схемою «Сніжинка» полягає в тому, що перша не використовує нормалізацію, а друга застосовує нормалізацію для усунення надмірності даних.

Побудова сховища даних за допомогою схеми є найпростішим підходом. Ядро схеми зазвичай складається з єдиної таблиці фактів разом із низкою взаємопов'язаних таблиць вимірювань. Нижче наведено приклад ER-діаграми для схеми «Зірка» (рис. 2).

Сховище даних за схемою «Сніжинка» це логічне розташування таблиць у базі даних, структурованих таким чином, що ER-діаграма нагадує форму сніжинки.

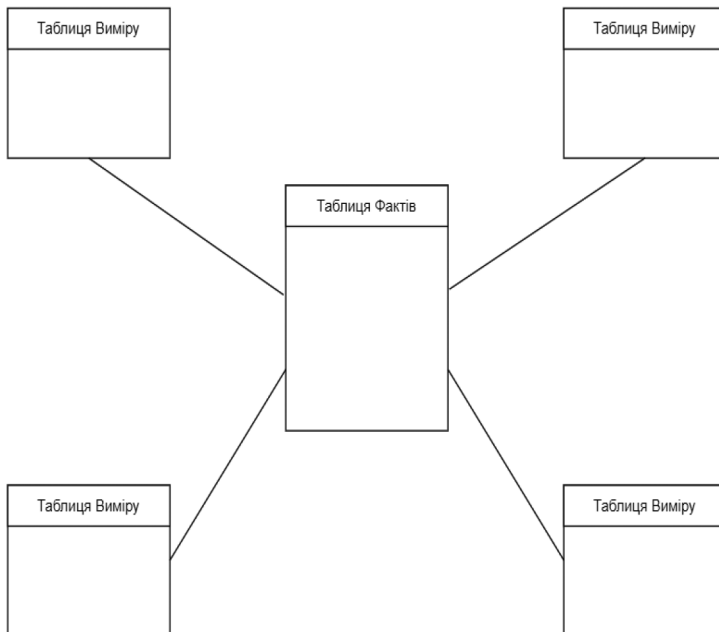


Рис. 1. ER-діаграма схеми «Зірка»

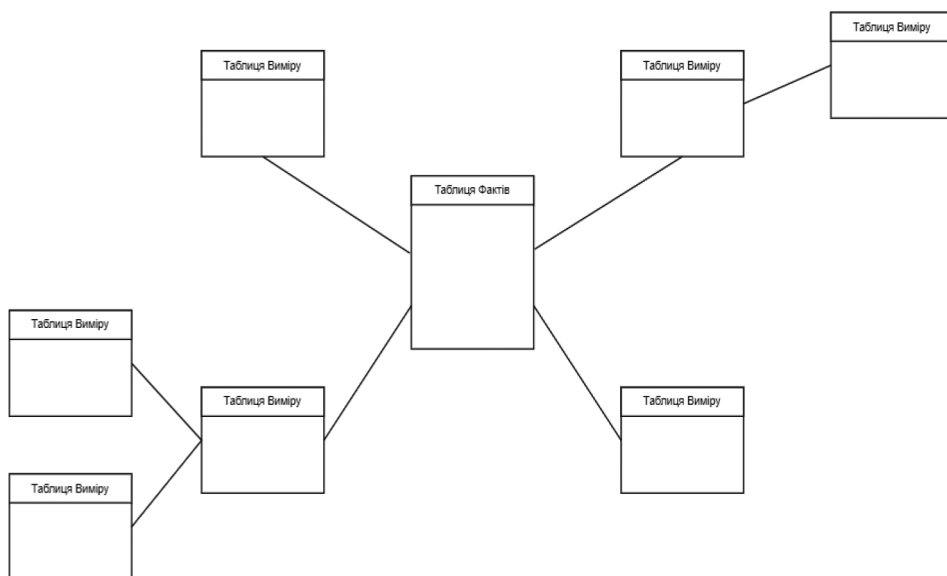


Рис. 2. ER-діаграма схеми «Сніжинка»

Ця схема є логічним розширенням схеми «Зірка», де ненормалізовані розміри нормалізуються та розбиваються на додаткові таблиці. Приклад ER-діаграми для схеми «Сніжинка» буде представлено на наступному малюнку (рис. 3).

Перенесення даних, які не оптимізовані для аналізу, з реляційної бази даних до сховища даних часто є одним із найскладніших аспектів, оскільки цей процес вимагає очищення та структурування «брудних» даних для подальшого зберігання та аналізу. Метою роботи буде створення реляційної бази даних, що імітує роботу інтернет-магазину, створення відповідного сховища даних та дослідження продуктивності процесу ETL для передачі даних у сховище даних різними методами.

Виклад основного матеріалу. Управління даними є критично важливим компонентом для організацій, оскільки створені дані є цінним ресурсом. Останніми роками ефективно управління даними стає все більш життєво важливим, особливо тому, що сучасні організації стикаються зі значним зростанням обсягу інформації,

MOCK_DATA (1) - Notepad

id	first_name	last_name	email	gender	phone
1	Clarke	Blackah	cblackah@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@uisc.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. 3. Приклад вхідних даних

що зберігається. Ретельний і структурований аналіз цих даних має важливе значення для досягнення комерційного успіху.

Основна увага дослідження зосереджена на процесі міграції даних з бази даних, яка безпосередньо взаємодіє з бізнес-додатком (система OLTP), до сховища даних (система OLAP), спрямованого на довгострокове зберігання та аналіз історичних даних [6].

Таким чином, вхідні дані для цього дослідження будуть надходити з реляційної бази даних, що моделює роботу інтернет-магазину. Оскільки дослідження проводитиметься з використанням тестових даних, вхідні набори даних будуть розділені на два типи – статичні та динамічні.

Статичними даними буде вважатись інформація, яка безпосередньо не імітує бізнес-діяльність організації. Цей тип даних містить відомості про користувачів, перевізників, магазини, категорії продуктів і послуги, які пропонує організація. Для створення таких даних використовуватимуться зовнішні ресурси з наборами даних. Нижче буде надано приклад вхідного файлу (рис. 3).

Цей тип файлу буде створено для всіх статичних об'єктів, а потім завантажено у відповідні сутності в реляційній базі даних за допомогою SQL Server Management Studio.

Динамічні дані, з іншого боку, містять інформацію та об'єкти, які моделюють бізнес-діяльність організації. Наприклад, це може включати дані, пов'язані з продажами або поставками. Такі дані будуть згенеровані на основі статичних об'єктів за допомогою збережених процедур. Зі статичних таблиць будуть вибрані

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. 4. 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	B008LZZOW4	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. 5. Представлення з даними продажі

випадкові атрибути, а кількість бізнес-дій відповідного типу буде розраховано випадковим чином. Згенеровані дані будуть безпосередньо вставлені у відповідні сутності. Приклад динамічних даних буде показано нижче (рис. 4).

Оскільки дослідження зосереджено на визначенні найефективнішого способу передачі різних обсягів даних із реляційної бази даних до сховища даних, вихідні дані будуть агреговані та структуровані у вигляді представлень. Ці перегляди базуватимуться на таблицях, які вже об'єднано та згруповано. Отримані представлення будуть корисними для проведення різноманітних аналізів, які можуть надати розуміння відповідних бізнес-контекстів організації. Нижче наведено приклад перегляду, що містить загальні неагреговані дані про продажі організації (рис. 5).

Також представлення можна використовувати для швидкого аналізу, роблячи необхідні агрегації даних і використовуючи необхідні формули всередині представлення (рис. 6).

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. 6. Приклад представлення з агрегованими даними

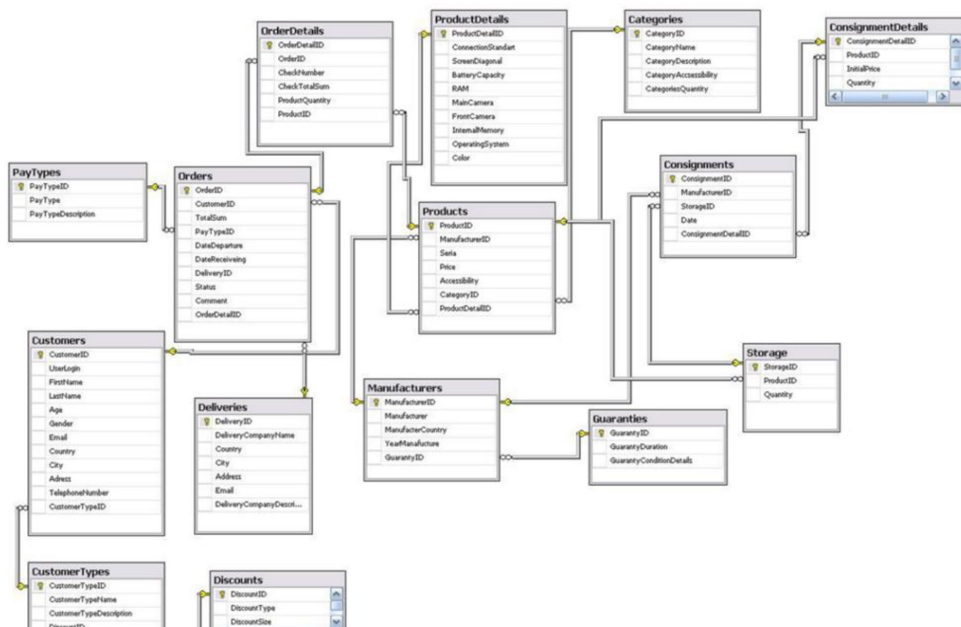


Рис. 7. Структура реляційної бази даних

Для проведення досліджень необхідно створити інформаційну модель системи. Інформаційна модель системи це структуроване уявлення про дані, які обробляються, зберігаються і використовуються в інформаційній системі. Вона визначає, які саме дані потрібні, як вони взаємопов'язані, де і як зберігаються, а також як змінюються в процесі роботи системи. Насамперед потрібно визначити які логічні сутності будуть представлені у системі, яка буде досліджуватись.

У результаті детального аналізу було сформовано перелік основних логічних сутностей, що становлять основу інформаційної моделі системи. До таких сутностей належать: Магазин, Склади та Товари, Категорії та підкатегорії товарів, Постачальники та Виробники, Ціна, Покупці та Знижки. Визначення цих сутностей є ключовим етапом у проектуванні реляційної бази даних, оскільки вони відображають основні об'єкти предметної області та взаємозв'язки між ними. Проаналізувавши логічні сутності на їх бізнес контекст, можна реалізувати структуру OLTP системи. Структура реалізованої реляційної бази даних (рис. 7).

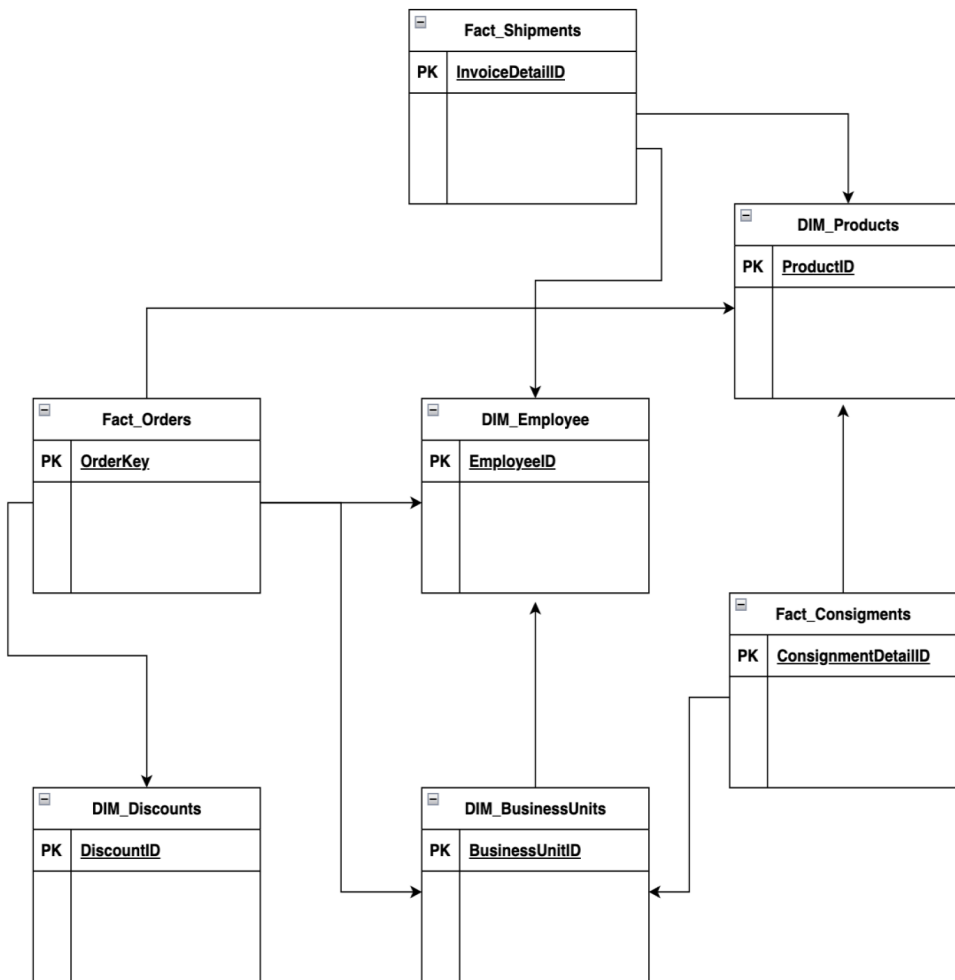


Рис. 8. Структура сховища даних

Під час включення даних у сховище даних серія кроків денормалізації є важливою для оптимізації продуктивності системи та збереження цілісності даних. Під час цього процесу сутності, які мають спільну бізнес-цілі, об'єднуються в уніфіковані таблиці – таблиці фактів або таблиці розмірів. Дані, визначені як статичні на етапі введення, організовуються та перетворюються на розміри, забезпечуючи стабільну основу для аналізу. Водночас динамічні дані, які відображають поточні зміни, спрямовуються в таблиці фактів, забезпечуючи ефективність сховища та можливість його адаптації до мінливих потреб. Враховуючи вище наведену структуру і опираючись на бізнес потреби організації було сформовано структуру сховища даних. Нижче буде наведено модель спроектованого сховища даних (рис. 8):

Для відображення структури системи обрано діаграми UML. Ці типи UML-діаграм допоможуть зобразити OLTP систему, сховище даних та візуалізувати інтеграційні зв'язки між цими двома системами. Діаграма зображена на рис. 9.

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. Діаграми UML наведено на рис. 10–15.

В діаграмі задіяні чотири актора, кожен з яких виконує свої функції:

- Покупець робить замовлення і отримує товар.
- Менеджер обробляє замовлення покупця і укладає договір.
- Постачальник надає товар згідно договору.
- Логістична служба формує поставку і відправляє її Покупцю.

Дослідження ефективності процесу ETL проводитиметься на наборах даних різного розміру. Спочатку ми виконаємо аналіз за допомогою збережених процедур



Рис. 9. Структура системи

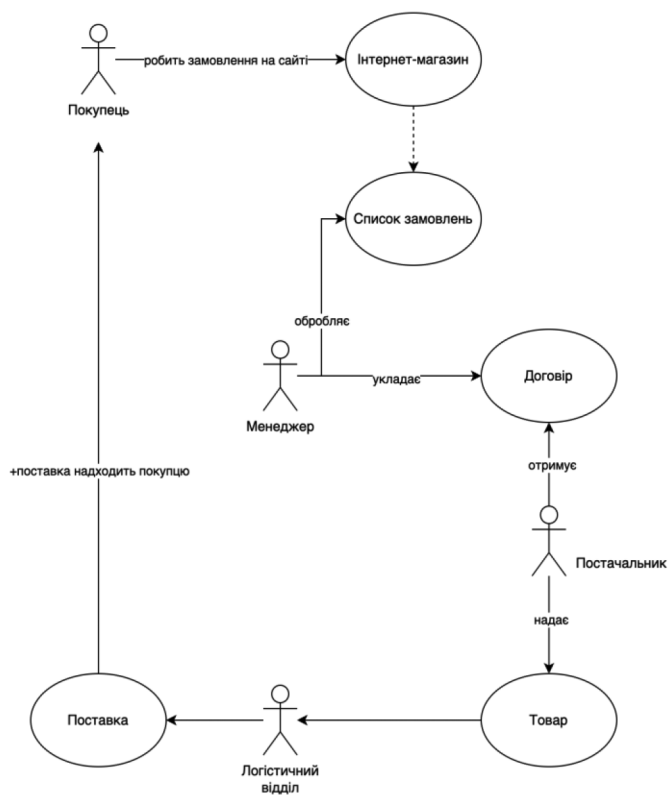


Рис. 10. Use Case діаграма

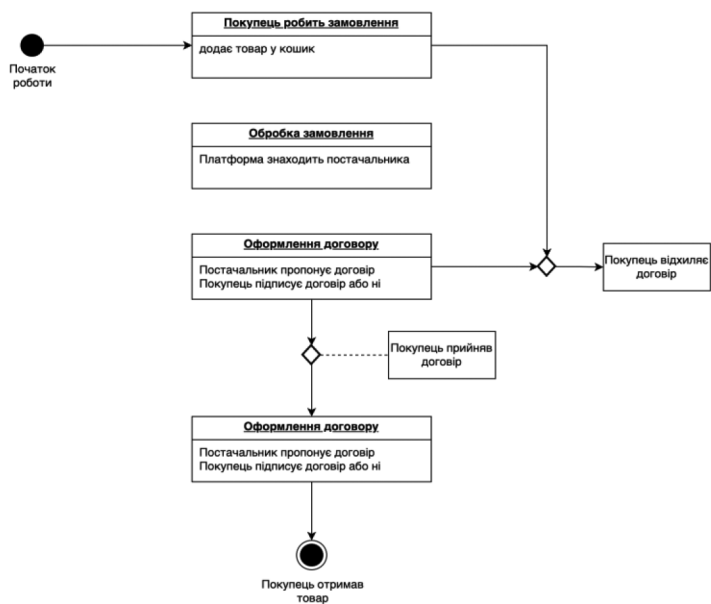


Рис. 11. Діаграма станів

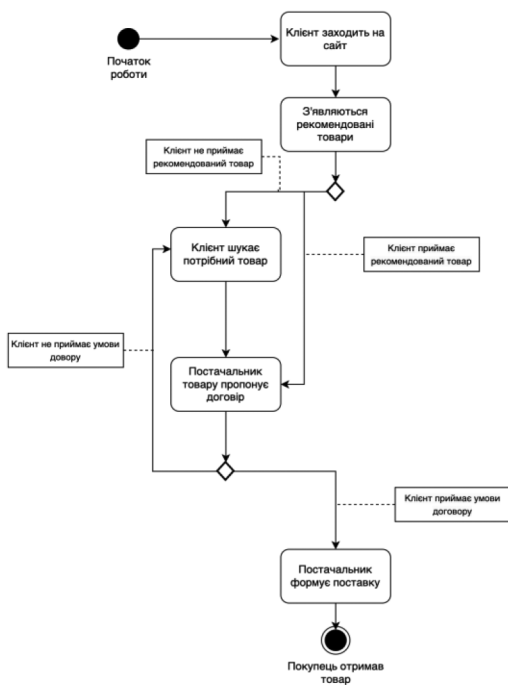


Рис. 12. Діаграма діяльності

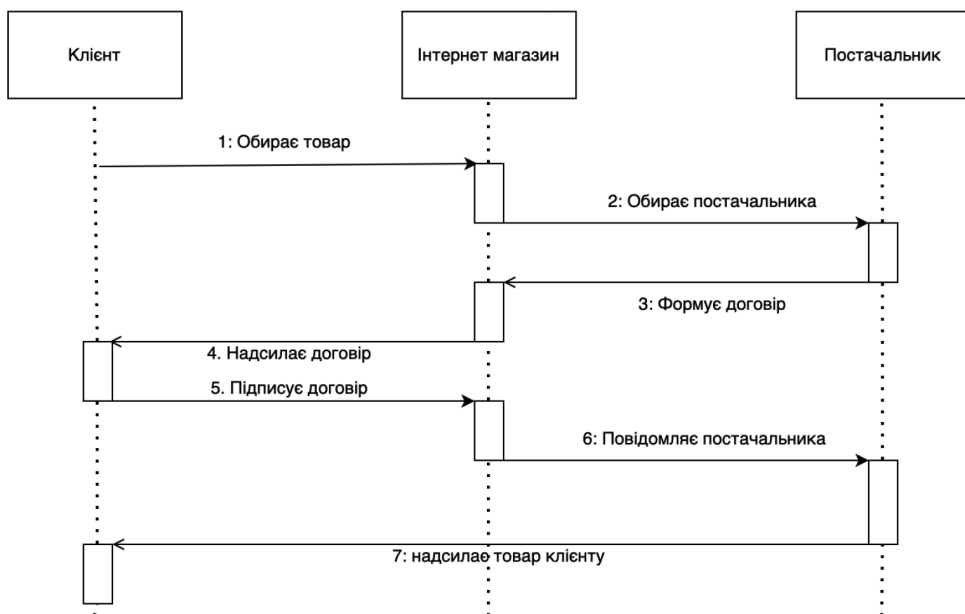


Рис. 13. Діаграма послідовності

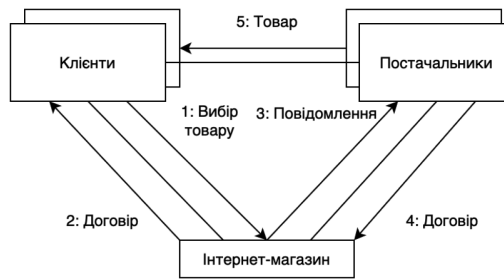


Рис. 14. Діаграма кооперації

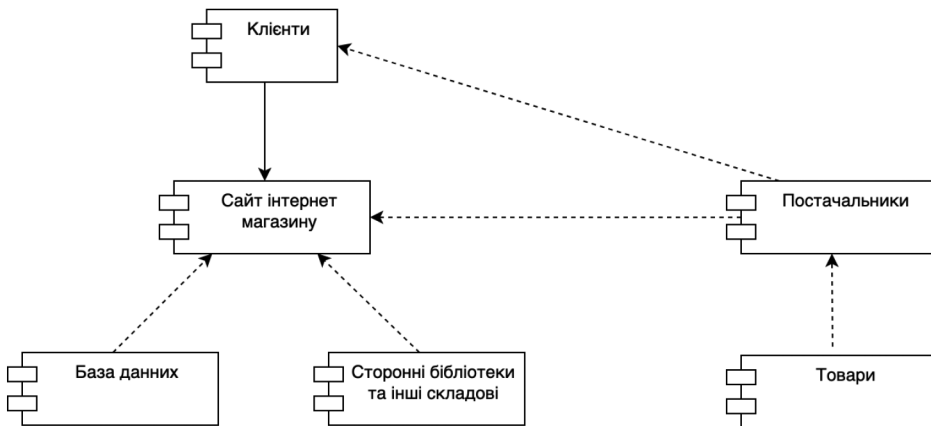


Рис. 15. Діаграма компонентів

у T-SQL. Для цієї передачі даних було створено процедуру, яка застосовуватиме спеціальні фільтри під час міграції даних. Експерименти проводитимуться на вибраних обсягах даних, поступово збільшуючи кількість записів. Зобразимо у вигляді лінійного графіка залежність зміни часу переміщення від об'єму даних, що передається.

Аналізуючи результати, які були отримані під час експериментів, видно, що зі збільшенням об'єму інформації що опрацьовується збільшується час роботи. 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>>

З наведених графіків видно, що використання FastLoad не лише забезпечує суттєве скорочення часу виконання, а й демонструє більш стабільне зростання цього показника порівняно з PlainLoad. Як показано на рис. 17, після різкого збільшення обсягу даних час виконання зростає непослідовно, тоді як при використанні FastLoad спостерігається майже лінійна залежність часу від обсягу даних.

Отже, за результатами проведеного дослідження встановлено, що створення SSIS-пакетів із використанням FastLoad є найбільш оптимальним підходом для обробки та перенесення даних із застосуванням цього інструменту. На цьому етапі

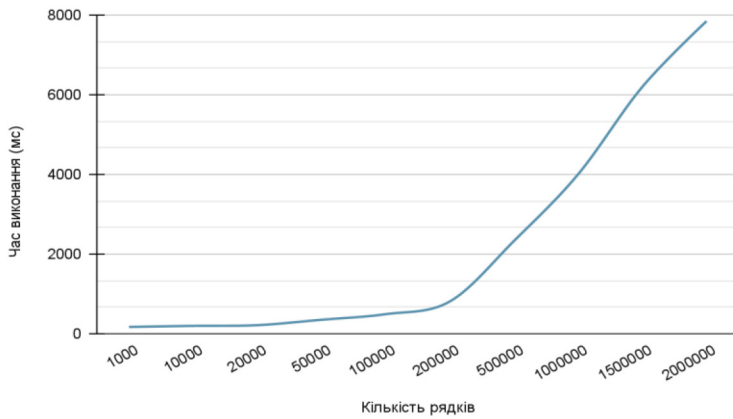


Рис. 16. Графік залежності часу до об'єму інформації обробленої за допомогою T-SQL

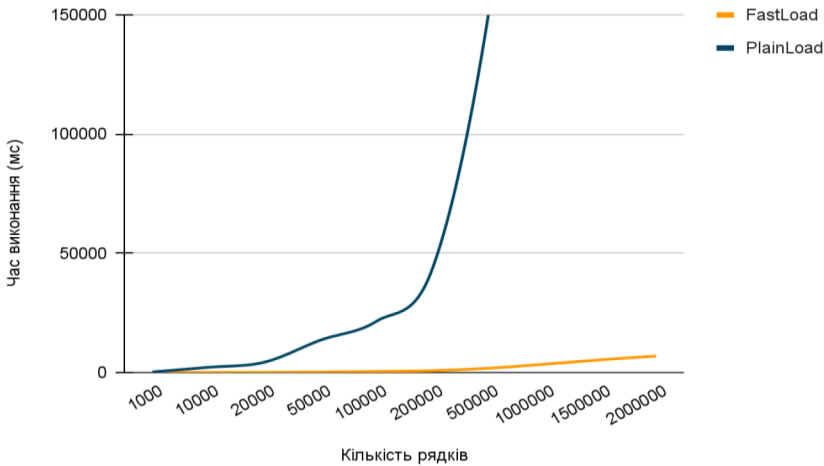


Рис. 17. Графік порівняння швидкості часу завантаження за допомогою SSIS пакетів використовуючи Fast та PlainLoad

доцільним є проведення порівняльного аналізу між T-SQL та FastLoad, з метою визначення найбільш ефективного способу з-поміж розглянутих.

Отже, з наведеного графіка видно, що перевага все ж таки на боці FastLoad, оскільки при обробці відносно великих обсягів даних спостерігається приріст продуктивності. Водночас, при обсягах до 200 000 записів цей приріст є не настільки суттєвим. Незважаючи на те, що навіть на менших обсягах даних FastLoad демонструє кращі результати, доцільно рекомендувати використання T-SQL, адже розробка за допомогою цього інструменту є простішою та не потребує додаткових засобів, як це має місце при створенні SSIS-пакетів. Проте для прийняття оптимального рішення необхідно враховувати всі чинники, зокрема специфіку завдання та бізнес-потреби.

Висновки. У результаті проведеного дослідження розроблена та реалізована три-рівнева інформаційна модель системи, яка включає транзакційний рівень на основі системи OLTP, інтеграційний рівень з використанням ETL-процесів (Database Engine та пакетів SSIS), а також аналітичний рівень у вигляді сховища

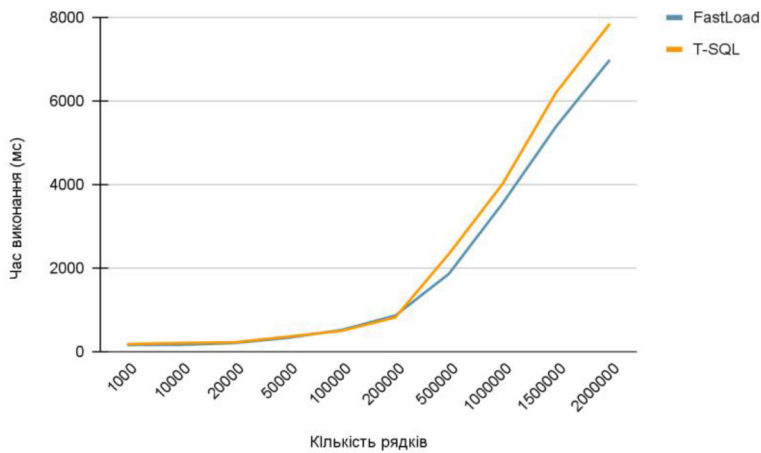


Рис. 18. Графік порівняння швидкості часу завантаження за використанням T-SQL та FastLoad

даних, побудованого за гібридним підходом. Запропонована система відповідає концепції самообслуговування, оскільки забезпечує повноцінну роботу з даними без залучення зовнішніх аналітичних платформ, що значно підвищує гнучкість і автономність бізнес-процесів.

У процесі проектування проаналізовано предметну область, описано логічні сутності, побудовано інформаційну модель реляційної бази даних і сховища даних відповідно до актуальних бізнес-потреб. Визначено структуру вхідних і вихідних даних, а також запропоновано можливі сценарії їх подальшого використання в межах підприємства.

Запропоновано методику оптимізації процесів обробки і перенесення даних в інформаційних системах, яка дає змогу ефективно реалізовувати передачу структурованих даних різного обсягу з мінімальним використанням зовнішніх ресурсів. Застосовані інструменти інтеграції, зокрема SSIS Fast Load, продемонстрували високу продуктивність і стабільність у порівнянні з іншими підходами, що підтверджує доцільність їх використання у подібних системах. Використовувані інструменти інтеграції можна застосовувати для передачі будь-яких типів даних, що допоможе краще аналізувати структуровані дані для визначення основних бізнес-потреб системи.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. Saraswat P., Raj S. Educational data mining and data warehouse design using business intelligence. *International Journal of Innovative Research in Computer Science & Technology*. 2022. P. 97–101. URL: <https://doi.org/10.55524/ijircst.2022.10.1.17>
2. Data Optimization for Industrial IoT-Based Recommendation Systems / M. Beshley et al. *Electronics*. 2022. Vol. 12, no. 1. P. 33. URL: <https://doi.org/10.3390/electronics12010033>
3. Novoselova O., Ruzheynikov A., Gavrillov A. Analytical Processing of Applied Tasks Conceptual Models at Design of Information-Active Systems. *EPJ Web of Conferences*. 2019. Vol. 224. P. 06008. URL: <https://doi.org/10.1051/epjconf/201922406008>
4. Jaroли P., Masson P. Data Warehousing and OLAP Technology (Data warehousing). *International Journal of Engineering Trends and Technology*. 2017. Vol. 51, no. 1. P. 45–50. URL: <https://doi.org/10.14445/22315381/ijett-v51p208>

5. A Data Warehouse Approach for Business Intelligence / G. Garani et al. 2019 *IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, Napoli, Italy, 12–14 June 2019. 2019. URL: <https://doi.org/10.1109/wetice.2019.00022>

6. Rachapudi S. Holistic Data Management: Integrating OLTP and OLAP in Financial Systems. *International Journal of Science and Research (IJSR)*. 2024. Vol. 13, no. 10. P. 1115–1118. URL: <https://doi.org/10.21275/sr241015110206>

REFERENCES:

1. Saraswat, P., & Raj, S. (2022). Educational data mining and data warehouse design using business intelligence. *International Journal of Innovative Research in Computer Science & Technology*, 97–101. <https://doi.org/10.55524/ijircst.2022.10.1.17>

2. Beshley, M., Hordiichuk-Bublivska, O., Beshley, H., & Ivanochko, I. (2022). Data Optimization for Industrial IoT-Based Recommendation Systems. *Electronics*, 12(1), 33. <https://doi.org/10.3390/electronics12010033>

3. Novoselova, O., Ruzheynikov, A., & Gavrilov, A. (2019). Analytical Processing of Applied Tasks Conceptual Models at Design of Information-Active Systems. *EPJ Web of Conferences*, 224, 06008. <https://doi.org/10.1051/epjconf/201922406008>

4. Jaroli, P., & Masson, P. (2017). Data Warehousing and OLAP Technology (Data warehousing). *International Journal of Engineering Trends and Technology*, 51(1), 45–50. <https://doi.org/10.14445/22315381/ijett-v51p208>

5. Garani, G., Chernov, A., Savvas, I., & Butakova, M. (2019). A Data Warehouse Approach for Business Intelligence. Y 2019 *IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. IEEE. <https://doi.org/10.1109/wetice.2019.00022>

6. Rachapudi, S. (2024). Holistic Data Management: Integrating OLTP and OLAP in Financial Systems. *International Journal of Science and Research (IJSR)*, 13(10), 1115–1118. <https://doi.org/10.21275/sr241015110206>