ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

INFORMATION TECHNOLOGY

DOI: 10.20998/2079-0023.2025.02.09
УДК 004.65

І. К. БАБІЧ, аспірант, Національний технічний університет «Харківський політехнічний інститут», м. Харків, Україна; e-mail: Ihor.Babich@cs.khpi.edu.ua; ORCID: https://orcid.org/0009-0003-0433-4913
Д. Л. ОРЛОВСЬКИЙ, кандидат технічних наук (PhD), доцент, Національний технічний університет «Харківський політехнічний інститут», професор кафедри програмної інженерії та інтелектуальних технологій управління, м. Харків, Україна; e-mail: Dmytro.Orlovskyi@khpi.edu.ua; ORCID: https://orcid.org/0000-0002-8261-2988
А. М. КОПП, доктор філософії (PhD), доцент, Національний технічний університет «Харківський політехнічний інститут», завідувач кафедри програмної інженерії та інтелектуальних технологій управління, м. Харків, Україна; e-mail: Andrii.Kopp@khpi.edu.ua; ORCID: https://orcid.org/0000-0002-3189-5623

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ІНТЕГРАЦІЇ ДАНИХ ПРО КЛІЄНТІВ ТА СПОЖИВАЧІВ

На прикладі книжкового підприємства, що поєднує функції видавця, дистриб'ютора та ритейлера, показано, як багатоканальна операційна діяльність призводить до накопичення у базах даних величезних масивів інформації, яка є фрагментованою, неповною, неструктурованою та містить дублікати. Такий стан унеможливлює ефективний аналіз клієнтської поведінки, зокрема точний розрахунок ключових показників ефективності. Актуальність роботи полягає у зменшенні цього критичного розриву між обсягом накопиченої інформації та здатністю бізнесу приймати на її основі ефективні управлінські рішення. Метою даної роботи є розробка методологічного підходу до створення сховища даних за архітектурою «зірка» та реалізація адаптивного ETL-ланцюга із вбудованими правилами контролю якості. Проведено аналіз сучасних методів проєктування сховищ даних, включно з переходом від моделі «сутність-відношення» до схеми «зірка». На основі структури транзакційної бази даних та бізнес-вимог до аналізу даних спроєктовано аналітичне сховище за схемою «зірка», визначено ключові факти і виміри, необхідні для підтримки всебічної клієнтської аналітики. Для перенесення даних з оперативної системи до сховища розроблено процес вилучення, перетворення та завантаження даних, описано його логіку: вибірку даних із джерел, їх очищення та трансформацію у проміжній зоні, а також завантаження у цільові таблиці сховища. Ефективність розроблених процесів оцінено на основі даних журналу реєстрації подій. Результати аналізу підтверджують надійність та високу продуктивність запропонованого рішення. Запропонований у статті підхід забезпечує автоматизоване, надійне й ефективне оновлення сховища даних, створюючи єдине джерело достовірних даних для бізнес-аналітики.

Ключові слова: база даних, сховище даних, інтеграція даних, схема «зірка», таблиці вимірів/фактів, ETL.

Вступ. Сучасні компанії накопичують величезні масиви транзакційної, поведінкової та контактної інформації про клієнтів та їх поведінку, яка може стати справжнім джерелом цінних аналітичних звітів. Проте ці дані часто неструктуровані, задубльовані, неповні, особливо коли йдеться про показники споживчої лояльності. Точність і повнота клієнтських даних безпосередньо впливають на персоналізацію пропозицій, сегментацію споживачів, оцінку життєвої цінності клієнта (Customer Lifetime Value – CLV), визначення частоти покупок, середнього чека та ймовірності повторного звернення.

Досліджувана база даних обслуговує діяльність книжкового підприємства, яке одночасно виконує функції видавця, дистриб'ютора й роздрібного продавця. Книги реалізуються через мережу офлайн-магазинів, інтернет-платформу та прямі продажі через кол-центр. Така багатоканальність ускладнює аналіз лояльності – дані про замовлення, клієнтів, товари, бонуси та канали продажу зберігаються у різних підсистемах, що унеможливлює комплексне оцінювання взаємодії з покупцем.

Таким чином, виникає розрив між величезним обсягом накопиченої інформації про клієнтів і здатністю бізнесу ефективно використовувати ці дані для прийняття рішень. Виходом із ситуації є створення окремого аналітичного шару – сховища даних, яке займатиметься очищенням, інтеграцією та збереженням історії даних, не заважаючи при цьому роботі операційних систем.

Мета цієї роботи полягає у розробці методологічного підходу до створення сховища даних за архітектурою «зірка», яке забезпечує перетворення недосконалої операційної бази книжкової компанії на повноцінний аналітичний ресурс. Такий підхід дозволяє автоматизувати оцінювання клієнтів та покращити персоналізацію програм лояльності.

Аналіз стану питання. Операційні бази даних, що збирають транзакційну й поведінкову інформацію про клієнтів, створювалися передусім для швидкої фіксації операцій, а не для глибокого аналізу. Із часом це переросло у системну проблему: записи дублюються, ключі відсутні або нечітко визначені, формати дат та валют різняться, історичних змін немає. У

Дослідницька стаття: Цю статтю опубліковано видавництвом ***НТУ «ХПІ»*** у збірнику «Вісник Національного технічного університету "ХПІ" Серія: Системний аналіз, управління та інформаційні технології». Ця стаття поширюється за міжнародною ліцензією Creative Common Attribution (CC BY 4.0). **Конфлікт інтересів:** Автор/и заявив/или про відсутність конфлікту.

таких умовах навіть базові показники лояльності (Recency, Frequency та Monetary, CLV, імовірність відтоку) обчислюються неточно або вимагають непропорційних ресурсів. В результаті маркетинг і програми лояльності спираються на неповні чи спотворені дані, персоналізація стає нерелевантною, а витрати на утримання клієнтів зростають.

Щоб обійти обмеження OLTP-структур (Online Transaction Processing), компанії створюють окремі копії баз, але це призводить до збільшення ізольованих сховищ даних і підвищує ризики безпеки. Світові аналітичні огляди показують: більше половини часу аналітиків витрачається на очищення й підготовку даних, тоді як бізнес очікує швидких аналітичних результатів.[1] Це створює помітний розрив між обсягом клієнтської інформації та можливістю її ефективного використання.

Очевидним рішенням є виокремлення аналітичного шару – сховища даних з ретельно спроєктованим процесом перенесення, обробки та завантаження інформації, а також засобами аналітичного опрацювання.[2] Таке сховище бере на себе очищення, нормалізацію та історичну зміну даних, не втручаючись у роботу транзакційної системи. Саме існує потреба в методології побудови подібного сховища, яке здатне перетворити оперативну базу на повноцінний аналітичний ресурс.

Аналіз основних досягнень і літератури. Питанню перетворення «сирої» клієнтської інформації на аналітичний ресурс присвячено значний масив праць, який умовно складається із трьох взаємопов'язаних блоків: проєктування сховища, забезпечення якості й завантаження даних та використання отриманих аналітичних результатів у бізнес-процесах.

Одним з найперших ґрунтовних підходів до побудови сховища даних був запропонований Вільямом Інмоном. У своїй класичній праці [3] він визначив сховище даних як «тематично-орієнтоване, інтегроване, з часовою прив'язкою та незмінюване зібрання даних, призначене для підтримки прийняття управлінських рішень». Підхід Інмона відомий як методологія «зверху вниз»: спочатку проєктується централізоване корпоративне сховище даних, яке містить інтегровану модель всіх даних підприємства, а вже потім на його основі створюються залежні тематичні вітрини даних (data marts) для окремих підрозділів. Така архітектура, також звана Corporate Information Factory, включає операційне сховище даних як проміжний рівень, централізовану базу сховища та набори вітрин, забезпечуючи глобальну узгодженість даних і стратегічну інтеграцію на рівні всього підприємства. Основне завдання, яке вирішує підхід Інмона, – це забезпечення єдиного джерела правдивих даних для організації за рахунок попереднього інтегрування та очищення даних перед їхнім аналізом.

Альтернативний підхід запропонував Ральф Кімболл – це методологія «знизу вгору», орієнтована на побудову сховища через поетапне впровадження невеликих предметно-орієнтованих сховищ (вітрин даних) та їх об'єднання за допомогою єдиних вимірів. У книзі [4] описано концепцію багатовимірного сховища: проєктування починається з окремих вітрин, а закінчується формуванням цілісного сховища підприємства. Кімболл ввів поняття шинної архітектури підприємства з узгодженими вимірами, що дозволяє паралельно розробляти незалежні вітрини для різних відділів і водночас зберігати цілісність всієї системи. Підхід Кімболла вирішує проблему швидкого впровадження аналітичних рішень у конкретних бізнес-сферах, забезпечуючи гнучкість і наочність даних завдяки використанню денормалізованих моделей (зіркова схема та сніжинка). Водночас потенційним недоліком є необхідність ретельного узгодження між вітринами через відсутність єдиної централізованої моделі на початку проєкту – саме це вирішується підходом Інмона ціною більшої тривалості впровадження.

У [5] розроблено формальний підхід до визначення гранулярності факт-таблиць та ієрархій вимірів, що долає проблему надлишковості даних і забезпечує баланс між швидкістю запитів і гнучкістю моделі. Для уточнення семантики багатовимірних структур і спрощення інтеграції з існуючими інформаційними системами запропоновано об'єктно-орієнтовані моделі. У [6] представлено концептуальну алгебру для опису операцій агрегації, що подолала обмеження традиційного ER-підходу й унормувала агрегатні функції.

В літературі зустрічаються пропозиції створення гібридних та розподілених схем. Так, Томашевський і Яцишин у роботі [7] запропонували розширену гібридну архітектуру сховища даних, що враховує різнорідні джерела даних. Йдеться про поєднання централізованого сховища з розподіленими компонентами: частина даних може залишатися у локальних базах або оперативних системах, тоді як інтегроване ядро збирає ключову інформацію. Така гібридна побудова вирішує задачу інтеграції гетерогенних джерел – автори показують, що вона покращує гнучкість системи та враховує особливості різних типів даних.

Процес Extract–Transform–Load (ETL) – невід'ємна складова будь-якого сховища, адже саме ETL відповідає за інтеграцію даних із різноманітних джерел, їх очищення, трансформацію під цільову модель і завантаження до сховища. Успішність проєкту сховища значною мірою залежить від належного планування та реалізації ETL-процесів. Наприклад, Debbarma et al. [8] систематизували ключові проблеми якості та продуктивності при завантаженні до сховища даних, показавши, що дублікати, пропуски та неоптимальні перетворення можуть збільшувати час обробки на 30–50 % та викривляти результати аналітики. Автори статті [9] визначають ETL як основний механізм консолідації гетерогенних операційних даних у єдине корпоративне сховище. Виділено три етапи: витяг даних із різнорідних джерел, їхнє приведення до спільної схеми з урахуванням якості (очищення, нормалізація) та завантаження в DW, що дозволяє забезпечити повноту, узгодженість і оперативність аналітичних запитів Перевагою цього підходу названо наявність добре відпрацьованих інструментів і методик, що дають змогу централізовано управляти процесами інтеграції й забезпечувати високу продуктивність при роботі з внутрішніми даними підприємства. У моно-

графії [10] охоплені проблеми вимірювання та поліпшення якості даних у сучасних інформаційних системах, класифіковано підходи до оцінки й покращення якості даних та описані інструменти для очищення й верифікації даних.

В літературних джерелах певна увага приділяється оптимізації та продуктивності ETL. Відомо, що вузьким місцем сховища часто стають повільні завантаження або трансформації при зростанні обсягів даних. Тому з'явились роботи, присвячені підвищенню ефективності ETL-процесів. Так, Алі та Врембель представили огляд сучасних підходів до оптимізації етапу витягу в ETL-процесах [11], де узагальнили існуючі методи прискорення і вказали на відкриті проблеми у цій галузі (наприклад, автоматизований вибір оптимальних стратегій завантаження для різних сценаріїв).

Мета та задачі дослідження. Метою статті є розробка підходу до консолідації, очищення та ефективного використання даних про діяльність книжкового підприємства в аналітичних цілях, що забезпечить підвищення точності клієнтської аналітики та автоматизовану персоналізацію взаємодії з покупцями.

Для досягнення мети поставлені наступні задачі дослідження:

- спроєктувати схему сховища даних, що охоплює основні аспекти діяльності книжкового підприємства;
- розробити адаптивний ETL-ланцюг із вбудованими правилами контролю якості для автоматичного профілювання, дедуплікації й нормалізації даних.

Побудова схеми сховища даних. Згідно [12] проєктування сховища даних повинно ініціюватися та керуватися виключно потребами бізнесу. Він зацікавлений в отриманні якісних аналітичних звітів, які формуються базуючись на даних з спеціально створеного для цього сховища даних. В рамках цієї статті розглядається побудова сховища даних для аналізу замовлень клієнтів. Потрібно реалізувати можливість отримання відповідей на питання: «які клієнти робили замовлення в періоді», «що вони замовляли», «замовлення на яку суму і в якої кількості вони зробили», «який канал продажу популярний/непопулярний», «які товари були найбільш затребувані», «в які дні/тижні/місяці/квартали проявлялась активність споживачів», «які акційні пропозиції спрацювали на замовленнях». Орієнтуючись на такі вимоги, проєктується схема сховища даних.

Операційна база даних містить таблиці, інформація в яких відповідає процесам функціонування підприємства. Розглядається фрагмент структури транзакційної бази даних книговидавничого підприємства, що займається виданням та реалізацією книжкових товарів споживачам. Асортимент складається як з книжок власного видавництва, так і інших видавців. Іноді товари можуть об'єднуватися у комплекти, що включають у себе декілька книжок однієї серії (напр. про Гарі Поттера) або книжка з аксесуаром (зазвичай закладки, календарики тощо). Продукція від постачальників та друкарень збирається на центральному складі, звідти виконується її відправка до мережі магазинів та гуртовим покупцям. Оформлення замовлень відбувається на інтернет-сайті, через операторів контакт-центру або у фірмових магазинах роздрібної мережі. Клієнти, які ідентифікуються за номером мобільного телефону, складають замовлення з наявного асортименту, обирають бажаний спосіб доставки, метод оплати тим самим формуючи своє замовлення. На центральному складі замовлення запаковуються у відправлення, які обраним перевізником доставляються споживачу.

Назви та опис найбільш значущих таблиць, які використовуються для зберігання замовлень клієнтів наведені в табл. 1. Вони мають свої супутники-довідники, які зазвичай мають атрибут ідентифікатор та відповідні ньому пояснюючи атрибути, наприклад, type_name, category_name. Сутності зазвичай мають зв'язки з іншими сутностями.

Таблиця 1 – Центральні сутності фрагменту схеми бази даних

Назва сутності	Призначення
member_order	замовлення
order_contents	склад замовлення
shipment	відправлення
bonus	довідник бонусів
bonus_order	прив'язка бонусу до замовлення
member	особисті дані клієнтів
member_additional	контактні номери телефонів
member_address	адреси клієнтів
channel	довідник каналів продажу
complect	довідник комплектів товарів
product	довідник товарів
advertisement	довідник кодів реклами

На рис. 1 наведено фрагмент структури транзакційної бази даних BOOKS_DB, що зберігає дані про замовлення. Кольором згруповані сутності зі спільним смисловим навантаженням.

Для кожного клієнта в таблиці member зберігаються його особисті дані: прізвище, ім'я, стать, дата народження, дата першого замовлення, унікальний код клієнта. Крім того клієнти мають статус із таблиці member_status (активний, неактивний, заблокований, службовий) та тип з таблиці member_type (член клубу, не член клубу, співробітник, роздрібний). В окремій сутності member_additional зберігаються контактні телефони з датою їх актуалізації. Адреса проживання або бажана адреса доставки організуються у сутності member_address. Її атрибутами є повний рядок адреси, структурована адреса, тип з адреси (проживання або доставки), статус (активність адреси). Кожне замовлення містить від одного до декількох товарів різної кількості. Товари зібрані у таблицю product, де їх індивідуальні відмінності відповідають певним атрибутам: ідентифікатор продукту, каналу продажу, комплекту, тип, статус, код продукту, розміри.

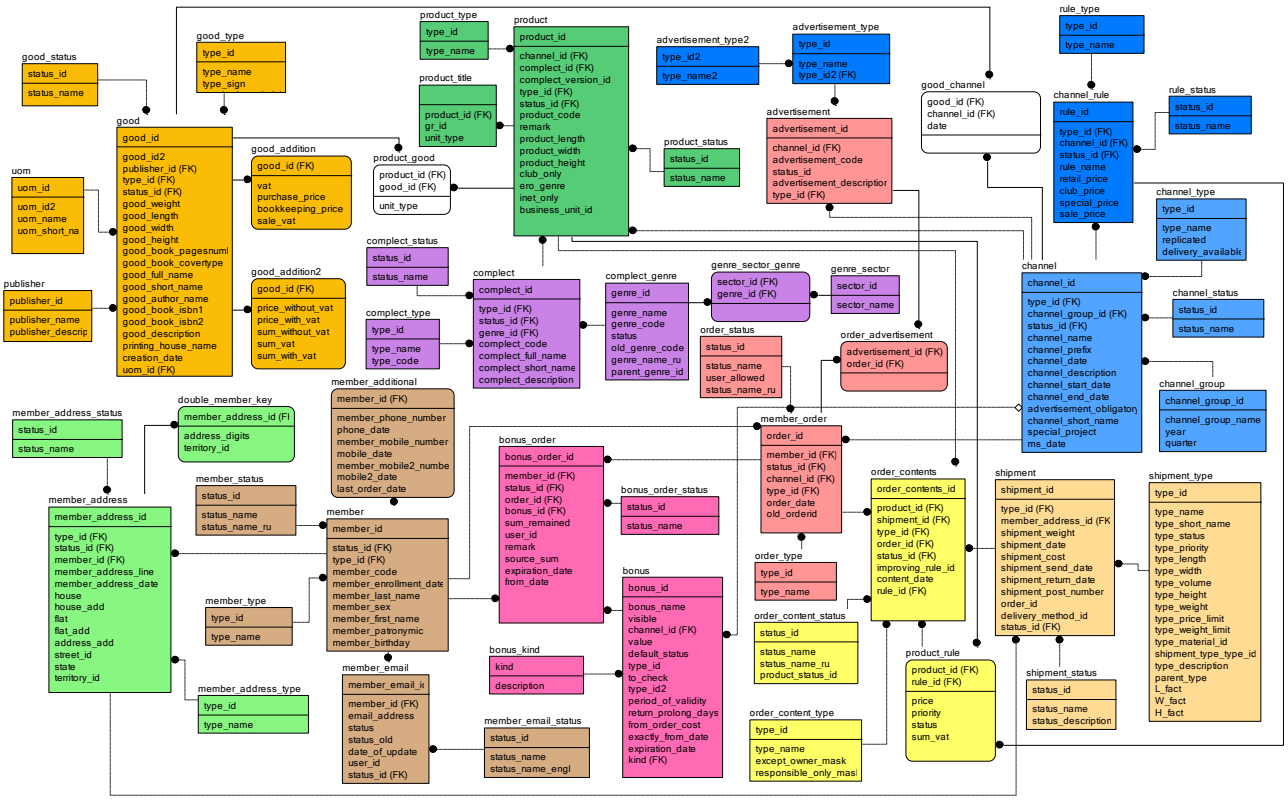


Рис. 1. Фрагмент структури бази даних BOOKS_DB (замовлення клієнтів)

Товари групуються в комплекти, які можуть складатися з одного або більшої кількості товарів. Для цього призначена таблиця complect, яка має атрибут для збереження коду комплекту, повну та коротку назву комплекту, зв'язок з довідником жанрів complect_genre, тип та статус відповідно у довідниках complect_type та complect_status. Усі товарно-матеріальні цінності, у тому числі товари для продажу, зорганізовані у глобальному довіднику good. В ньому зібрано найбільш повний перелік властивостей товарів, які продаються, та супутніх матеріалів. Ціни товарів в замовленні формуються залежно від стажу клієнта (строк перебування в якості клієнта), поточних акцій та накопичених бонусів та регулюються ціновими правилами. Цінові правила channel_rule залежать від каналу продажу і визначають роздрібну ціну, клубну ціну, спеціальну ціну та ціну розпродажу. Заголовкова таблиця замовлень member_order окрім ідентифікатору клієнта містить ідентифікатор каналу продажу, дату створення замовлення, тип, статус. Зміст замовлення міститься в таблиці order_contents з такими атрибутами: ідентифікатор замовлення, ідентифікатор продукту, час додавання продукту до замовлення, цінове правило, тип та статус. Ціна товару price у замовленні визначається через product_rule залежно від обраного товару product_id та цінового правила rule_id, яке «спрацювало» за визначеними умовами. Якщо спрацювало декілька умов, то згідно зі значенням priority обирається ціна с найменшим його значенням.

Після оформлення замовлення формуються одна або декілька відправлень (посилок). Кожне відправлення shipment має адресу доставки, обраний спосіб доставки, масу, дату формування, дату відправки, дату повернення, статус та тип. Для кожного товару в замовленні повинне визначатися відправлення shipment_id. Відправлення пакується у коробки залежно від розміру товарів. Обмежений набір коробок визначає тип відправлення shipment_type. Не виключена відправка замовлення декількома місцями в коробках відповідного розміру.

Операційна база даних, орієнтована на виконання транзакцій у реальному часі, має низку обмежень щодо застосування в задачах аналітики клієнтів та реалізації програм лояльності. Основними недоліками можна назвати відсутність історичних змін у сутностях, дублювання даних, неуніфіковану структуру довідників.

База даних не передбачає гнучкої агрегації інформації або збереження змін стану сутностей. Зокрема, зміна статусу клієнта, каналу замовлення чи типу товару не фіксується у розрізі часу, через що унеможливлюється побудова ретроспективної аналітики або аналізу життєвого циклу споживача. Деякі сутності дублюють атрибути або містять надлишкові зв'язки, що ускладнює інтеграцію в єдину аналітичну модель.

Виникає необхідність винесення аналітичного функціоналу за межі бази даних через створення окремого сховища даних із підтримкою історичності, уніфікованих вимірів та подієвої моделі.

Пропонується в якості моделі даних для сховища використати схему «зірка». Її простота й ефективність зробили її стандартом для проєктування сховищ даних, орієнтованих на аналітичні запити. Денормалізована структура «зірки» скорочує кількість необхідних операцій з'єднання між таблицями сховища. Модель даних у вигляді зірки більш інтуїтивно зрозуміла для розробників і бізнес-користувачів. Кожна таблиця фактів представляє бізнес-процес або подію, а таблиці розмірностей – контекстні довідники. Структура зі зрозумілим поділом на факти і виміри спрощує спілкування між IT-фахівцями і аналітиками, оскільки відображає природний спосіб мислення про бізнес-показники. Більшість сучасних BI-платформ (Power BI, Tableau, MicroStrategy тощо) та інструментів звітності розраховані на роботу зі сховищами у вигляді фактів і вимірів. Модель «зірка» спрощує інтеграцію з такими інструментами, оскільки забезпечує єдиний інтерфейс до даних: користувачі можуть будувати дашборди, перетягуючи поля вимірів (для фільтрів чи групування) та показники фактів (для агрегатів) без глибокого знання мови структурованих запитів. Простота також полегшує впровадження самообслуговування в аналітиці – коли бізнес-користувачі самі формують запити та звіти. Схема «зірка» ідеально підходить для OLAP-систем (On-Line Analytical Processing) та створення багатовимірних OLAP-кубів. У такій моделі дані вже організовано по вимірах, тому нарізка і обертання даних за різними розрізами виконується природно.

Структура сховища даних реалізує концепцію поділу інформаційного простору на факти та виміри. Факти у сховищі даних відображають події бізнес-процесів, які мають кількісне вимірювання. У моделі «зірка» факти утворюють центральну таблицю, що фіксує показники діяльності підприємства. Кожен запис у факт таблиці характеризує одну подію – наприклад, оформлення замовлення на придбання товару чи нарахування бонусу.

У рамках статті розглядається побудова фрагменту сховища, що слугує джерелом даних про замовлення клієнтів. У фрагменті сховища головною фактовою таблицею є fct_Order, яка відображає процеси замовлення товарів клієнтами. Її призначення – зберігати усі кількісні показники, за якими здійснюється подальший аналіз: сума замовлення (order_total_amount), кількість товарів у замовленні (order_total_items), бонус, нарахований за програмою лояльності (order_bonus_value), частка бонусу від загальної суми (order_bonus_pct), ціна одиниці товару та кількість (unit_price, quantity). Фактова таблиця fct_Order зберігає лише числові показники та зовнішні ключі на вимірні таблиці. Такий підхід мінімізує надлишковість даних і дає змогу виконувати швидке агрегування для побудови звітів за будь-якими зрізами – клієнтами, продуктами, каналами продажів або періодами часу.

Виміри описують контекст фактів – тобто, хто, що, де і коли здійснив певну дію. Вони містять текстові, категоріальні та довідкові дані, необхідні для інтерпретації числових значень фактів. Кожен вимір має унікальний ключ (surrogate key), який використовується для зв'язку з фактовою таблицею.

Вимір dim_Member описує атрибути клієнта, що є суб'єктом замовлення. До його складу входять поля ідентифікатора, коду клієнта, ПІБ, дати народження, статі, статусу, типу клієнта, контактних даних; використовується для побудови профілю клієнта, сегментації за типом і статусом. Вимір dim_Product містить описові характеристики товарів: тип продукту, назву, автора, жанр, тип обкладинки, кількість сторінок, валюту та тип виробництва. Цей вимір дозволяє аналізувати структуру продажів за видами продукції, авторами, категоріями, аналізувати структуру асортименту та популярність продуктів. Таблиця dim_Channel описує спосіб здійснення замовлення: ідентифікатор каналу, назву, тип, групу, рік, квартал і статус. Цей вимір необхідний для оцінки ефективності каналів продажу, аналіз конверсій та мультиканальної поведінки клієнтів. Вимір dim_Shipment узагальнює дані про процеси відправлення замовлень. Поля включають тип відправлення, дати формування, надсилання й повернення, метод доставки, вагу, територію. Цей вимір використовується для підтримки аналітики витрат на доставку, своєчасності виконання замовлень і географічної доступності сервісу. Календарний вимір dim_Date є обов'язковим у будь-якому сховищі даних. Він містить ключ дати (date_key), саму дату, номер дня, місяця, кварталу та року. Завдяки цьому виміру забезпечується можливість аналізу динаміки продажів у часі, побудови часових зрізів і сезонних моделей поведінки клієнтів.

Для схеми «зірка» необхідна наявність поєднань факта з вимірами за типом «багато-до-одного»: декілька записів у fct_Order можуть посилатися на один запис у певному вимірі. Це дозволяє зберігати детальність фактів і водночас уникати дублювання описової інформації у структурі сховища.

На базі вищеозначеного підходу створюється схема сховища даних, фрагмент якої представлений на рис. 2.

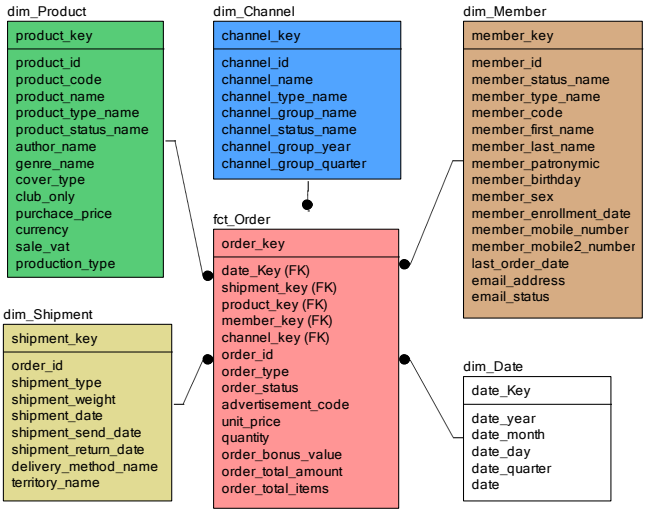


Рис. 2. Фрагмент структури сховища даних BOOKS_DWH

Таким чином, побудова схеми сховища даних у форматі «зірки» дозволила сформувати логічну модель, яка відокремлює аналітичні потреби від обмежень транзакційної структури. Визначено ключові вимірювання, факти та зв'язки, необхідні для підтримки подальшого аналізу клієнтських даних.

ETL-процес. Наступним етапом є реалізація ETL-процесу, який забезпечить перенесення даних з оперативної бази до сховища з урахуванням особливостей їх структури, якості та семантики. Для переведення сирих OLTP-даних у придатну для аналізу форму використовується трискладова процедура: витяг (Extract), перетворення (Transform) і завантаження (Load) (рис. 3).

На етапі витягу збираються необроблені дані з різних джерел даних. Ці джерела можуть бути різноманітними, починаючи від структурованих джерел, таких як бази даних (SQL, NoSQL), до напівструктурованих даних, таких як JSON, XML, або неструктурованих даних, таких як електронні листи або плоскі файли.

Основною метою вилучення є збір даних без зміни їх формату, що дозволяє їх подальшу обробку на наступному етапі. Нові та модифіковані рядки бази даних переміщуються у в окремі (або окрему) сутність (таблицю) у Staging area. На практиці це реалізується через роботу тригерів бази даних, які реагують на вставку або зміну рядків. З одного боку такий підхід мінімізує час доставки даних у сховище для оперативних даних. З іншого боку, такий підхід не придатний для початкового переміщення даних в сховище, коли його робота тільки розгортається.

На фазі перетворення дані, отримані на попередньому етапі, часто є сирими та суперечливими. Під час трансформації дані очищаються, агрегуються та форматуються відповідно до бізнес-правил. Це важливий крок, оскільки він гарантує, що дані відповідають стандартам якості, необхідним для точного аналізу. До поширених перетворень відносяться: видалення неактуальних або неправильних даних, сортування даних у необхідному порядку, узагальнення даних для надання значущої інформації (наприклад, усереднення даних про продажі). Етап трансформації також може включати більш складні операції, такі як конвертація валют, нормалізація тексту або застосування правил для конкретного домену для забезпечення відповідності даних потребам організації.

Етап завантаження передбачає перенесення перетворених даних у сховище даних, озеро даних або іншу цільову систему для зберігання. Залежно від випадку використання, існує два способи завантаження: усі дані завантажуються в цільову систему (часто використовується під час початкового заповнення сховища); завантажуються лише нові або оновлені дані (більш ефективний для поточних оновлень даних).

В рамках статті процеси ETL реалізовані за допомогою стандартних SQL-запитів та збережених процедур, що забезпечує контроль над логікою обробки даних.

Нижче надана діаграма діяльності ETL-процесу формування даних для виміру dim_Member (рис. 4). Така схема перетворень характерна саме для вимірів,

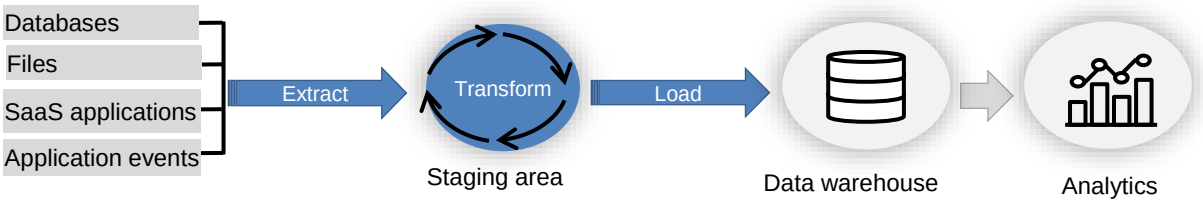


Рис. 3. Етапи та місце ETL-процесу

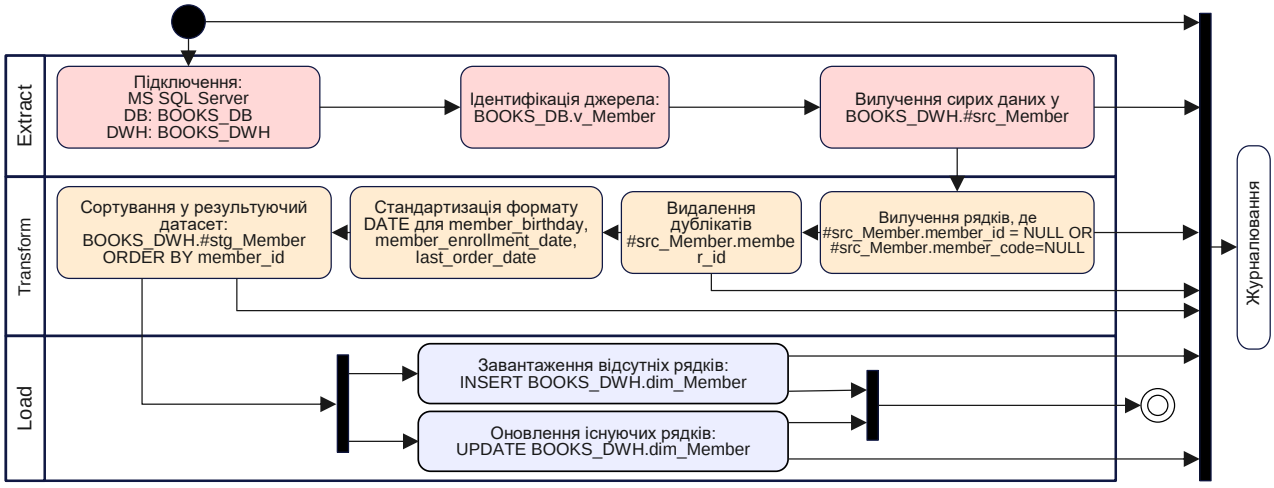


Рис. 4. Діаграма діяльності ETL-процесу формування виміру dim_Member

але для наочності наведена діаграма перетворень даних клієнтів.

Діаграма відображає логіку виконання ETL-процесу у трьох основних фазах: Extract, Transform, Load з обов'язковим елементом є ведення журналу операцій для контролю якості. Процес починається з ініціалізації – створення запису про запуск, що фіксує початок виконання, назву процесу, цільову таблицю та початковий статус RUNNING. Після цього відбувається фаза Extract, де дані витягуються з подання BOOKS_DB.dbo.v_Member, яке агрегує інформацію про клієнтів з оперативних таблиць (member, member_additional, member_email, member_status, member_type). На цьому етапі перевіряється заповненість ключових полів (member_id, member_code), видаляються записи з пропущеними значеннями, а їх кількість реєструється як метрика rows_dropped_nulls.

Далі відбувається фаза Transform, де дані очищуються та уніфікуються. Застосовується механізм усунення дублікатів за member_id, а також виконується перетворення типів даних: поля member_birthday, member_enrollment_date, last_order_date приводяться до типу DATE. Додатково виконується нормалізація текстових значень (обрізання пробілів, переведення у стандартний регістр). Результат перетворення зберігається у проміжній таблиці #stg_Member, яка використовується для подальшого завантаження у вимір.

У фазі Load відбувається синхронізація виміру з даними з джерела. Через оператор MERGE здійснюється порівняння проміжної таблиці з цільовою BOOKS_DWH.dbo.dim_Member: якщо запис уже існує – він оновлюється (UPDATE), якщо ні – додається новий (INSERT). Після завершення завантаження виконується повторна реєстрація подій з фіксацією фактичних результатів обробки – кількість рядків, швидкість обробки, наявність помилок чи попереджень.

Формування даних для таблиці фактів fct_Order реалізує повний цикл перетворення транзакційних даних про замовлення у аналітичну форму, придатну для інтеграції зі схемою «зірка» сховища даних BOOKS_DWH. На діаграмі діяльності (рис. 5) відображено послідовність операцій, що виконуються у трьох основних фазах.

На початку процесу створюється запис, який фіксує початок роботи, назву процесу (ETL_fct_Order), цільову таблицю (fct_Order) та час старту. Далі виконується етап Extract, під час якого здійснюється вибірка даних із представлень v_Order та v_Bonus. Ці джерела містять інформацію про замовлення, товари, клієнтів, канали продажу та бонусні нарахування. На цьому етапі видаляються рядки з відсутніми ключовими полями (order_id, member_id, product_id), що унеможливлює порушення цілісності зв'язків у моделі.

На фазі Transform відбувається логічна підготовка даних до завантаження. Поле order_date перетворюється з типу datetime у date для узгодження з виміром dim_Date. Далі здійснюється агрегація транзакцій до рівня замовлення: визначаються показники суми замовлення (order_total_amount), кількості товарів (order_total_items) та відсотка бонусу від загальної суми (order_bonus_pct). Після цього дані з'єднуються з вимірними таблицями dim_Member, dim_Product, dim_Channel, dim_Shipment і dim_Date, формуючи проміжну структуру #stg_Fact. Окремо перевіряється унікальність складеного ключа (order_id, product_key); при наявності дублікатів зберігається перший запис, а інформація про очищені дублікати заноситься до лог-таблиці.

Фаза Load передбачає фізичне завантаження даних у цільову таблицю BOOKS_DWH.dbo.fct_Order. Операція MERGE забезпечує синхронізацію даних: нові записи додаються (INSERT), а існуючі – оновлюються (UPDATE). У разі виникнення помилок (наприклад, через порушення унікальності або типів даних) у лог-таблиці фіксується статус FAILED і текст помилки. У випадку успішного завершення процесу оновлю-

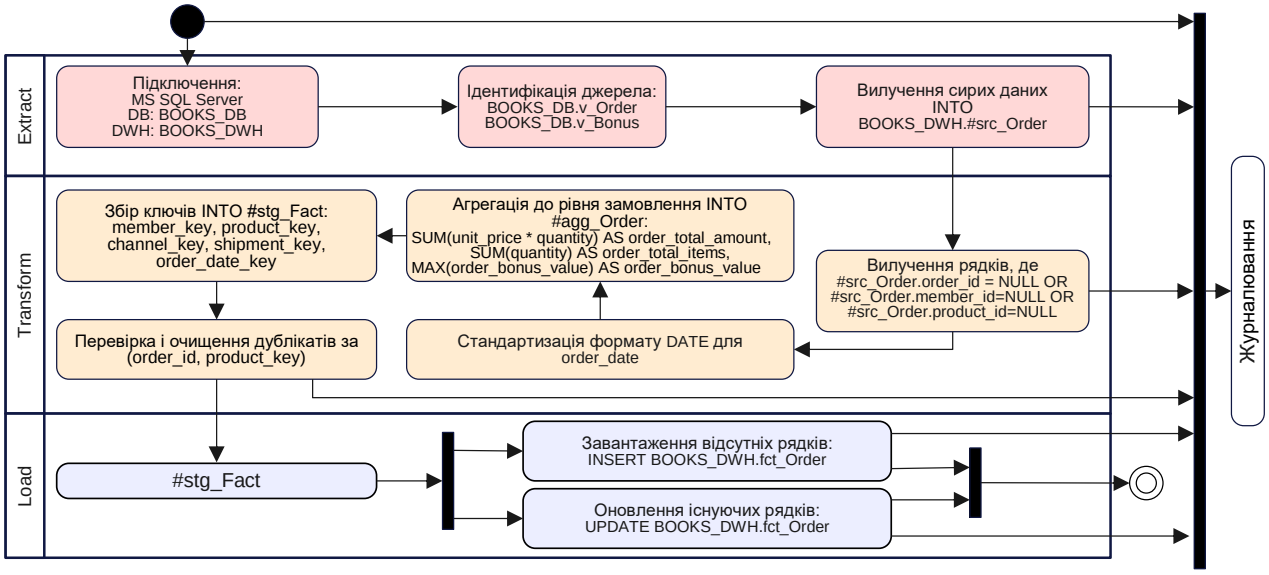


Рис. 5. Діаграма діяльності ETL-процесу формування fct_Order

ється запис у etl_metrics_log зі статусом SUCCESS та ключовими метриками – rows_extracted, rows_transformed, rows_loaded, rows_dropped, duplicates_removed, latency_sec та throughput_rps.

Аналіз ефективності ETL-процесів. З метою оцінки продуктивності та якості функціонування системи завантаження даних проведено аналіз основних показників таблиці etl_metrics_log, яка накопичує технічні метрики всіх ETL-процесів сховища BOOKS_DWH. Порівняння охоплює як фактові, так і вимірні процеси ETL_fct_Order, ETL_dim_Shipment, ETL_dim_Product, ETL_dim_Member та ETL_dim_Channel.(рис. 6)

Аналіз показників розпочато з етапу первинного завантаження, який відповідає за наповнення сховища початковими даними. Швидкодія процесів (Throughput) значно відрізняється залежно від складності обробки даних. Найвищий показник має ETL_dim_Shipment (65,8 тис. рядків/сек), що свідчить про ефективне завантаження однорідних записів. Дещо нижчий, але стабільно високий рівень демонструє ETL_dim_Product (64,9 тис. рядків/сек). У той час як процеси ETL_dim_Member (19,9 тис. рядків/сек) і особливо ETL_fct_Order (17,5 тис. рядків/сек) мають меншу пропускну здатність через складнішу логіку трансформації та багатотабличні об'єднання. Це очікувано, оскільки таблиця фактів обробляє найбільші обсяги даних і формує агреговані показники.

За тривалістю виконання (Latency) процесів спостерігається така закономірність:

- ETL_fct_Order – найдовший процес (180 сек), що зумовлено великим обсягом джерельних даних і необхідністю з'єднання з усіма вимірними таблицями;
- решта вимірних процесів виконуються швидко – від 10 сек для ETL_dim_Product до 47 сек для ETL_dim_Member;
- ETL_dim_Channel відпрацьовує практично миттєво через мінімальний обсяг даних.

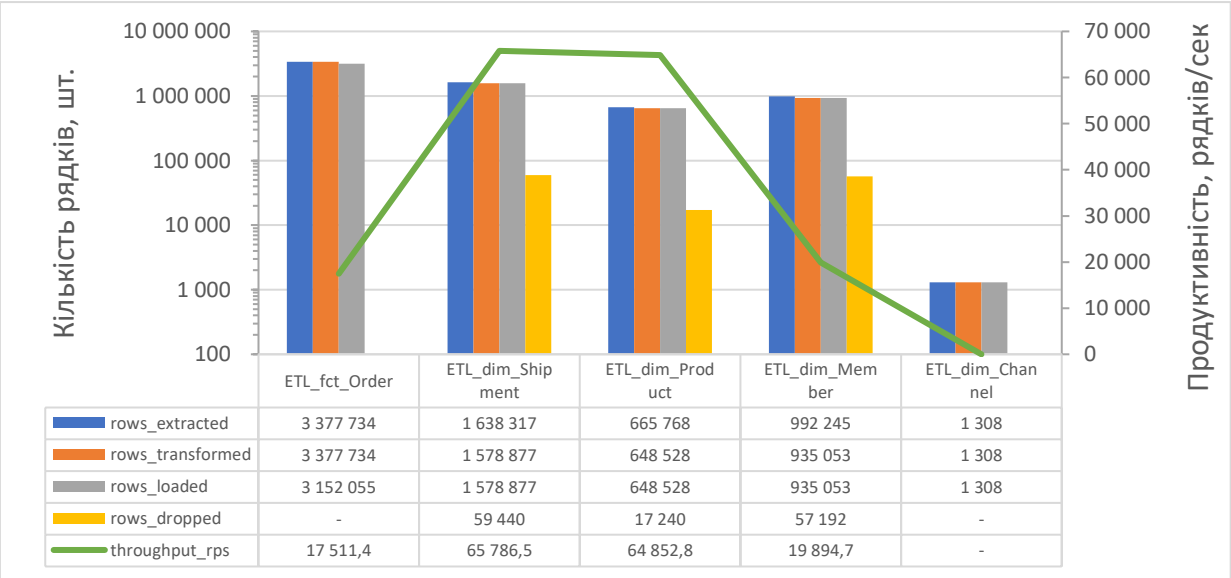
Це підтверджує правильне планування ETL-ланцюга: спочатку оновлюються невеликі виміри, потім – факт.

Показник відкинутих рядків (rows_dropped) відображає якість сирих даних на вході. Найбільші втрати спостерігаються у процесах ETL_dim_Shipment (59,4 тис.) та ETL_dim_Member (57,2 тис.), що становить 3,6 % та 5,8 % відповідно. Причиною може бути наявність дублікатів первинних ключів або відсутність обов'язкових значень. Для ETL_dim_Product частка відкинутих записів становить 2,6 %, що відповідає типовим межам очищення довідникових даних. Фактовий процес ETL_fct_Order виконується без втрат, оскільки очищення і валідація виконуються на попередніх рівнях.

Порівняння показників вилучених рядків (rows_extracted) і завантажених рядків (rows_loaded) демонструє, що більшість ETL-процесів зберігають понад 94–97 % вхідних даних після очищення, що є хорошим результатом для промислової обробки. Фактовий процес ETL_fct_Order зберіг 93 % даних (3,15 млн із 3,37 млн), що підтверджує ефективність механізмів агрегації без втрати фактів.

Рис. 7 демонструє роботу інкрементального завантаження даних у таблицю фактів протягом року після початкового наповнення. Щомісячне завантаження має стабільний час виконання у межах від 48 до 66 секунд при змінюванні обсягу даних від 180 до 370 тисяч рядків. Це свідчить про ефективність реалізованої ETL-архітектури, яка масштабується без помітної деградації продуктивності.

Отримані показники свідчать про стабільність і ефективність роботи ETL-системи. Найкращі результати демонструють процеси, орієнтовані на довідники з простою структурою, тоді як складні аналітичні модулі (особливо fct_Order) потребують більшого часу на обробку, що є закономірним.



	ETL_fct_Order	ETL_dim_Shipment	ETL_dim_Product	ETL_dim_Member	ETL_dim_Channel
rows_extracted	3 377 734	1 638 317	665 768	992 245	1 308
rows_transformed	3 377 734	1 578 877	648 528	935 053	1 308
rows_loaded	3 152 055	1 578 877	648 528	935 053	1 308
rows_dropped	-	59 440	17 240	57 192	-
throughput_rps	17 511,4	65 786,5	64 852,8	19 894,7	-

Рис. 6. Первинне наповнення: обробка рядків і продуктивність

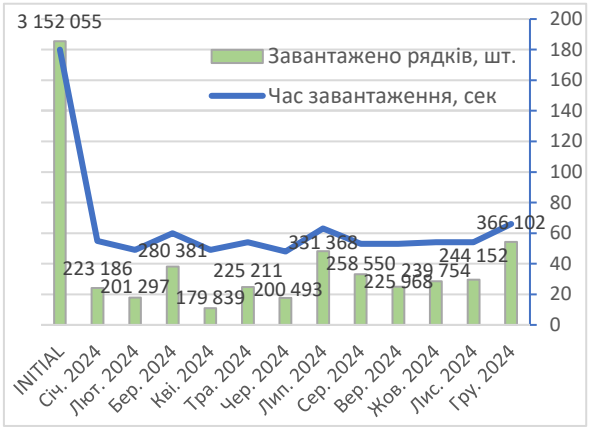


Рис. 7. Динаміка інкрементального завантаження (2024 рік)

Подальше вдосконалення ETL-ланцюга може бути спрямоване на:

- зниження кількості відкинутих записів через контроль первинних ключів у джерелі;
- паралелізацію трансформацій великих обсягів фактових даних;
- оптимізацію журналювання та розрахунку метрик.

Таким чином, результати аналізу доводять, що реалізована архітектура ETL забезпечує високу продуктивність, узгодженість даних і відтворюваність процесів, що відповідає вимогам промислового сховища даних для задач управління клієнтською лояльністю.

Висновки. Результатом роботи є побудова сховища даних з зірковою схемою в СУБД MS SQL Server, яке інтегрує різнорідні дані клієнтських замовлень у єдину аналітичну базу. Створено фактову таблицю замовлень та пов'язані з нею таблиці вимірів, що дозволило структурувати дані за основними бізнес-вимірами (продукти, клієнти, канали продажів, відвантаження) для зручного аналізу. Розроблені ETL-процеси з інкрементальним завантаженням підвищили ефективність оновлення даних, мінімізуючи обсяг обробки за рахунок вилучення лише змінених записів. Механізми виявлення дублікатів та перевірки якості даних гарантують високий рівень достовірності інформації, а ведення журналу метрик ETL забезпечує прозорість процесів та швидке виявлення можливих збоїв. Таким чином, розроблена система забезпечує узгодженість і актуальність даних, оптимізує продуктивність запитів та підвищує довіру користувачів до аналітичної звітності.

Запропоноване рішення має широкі перспективи застосування в інформаційно-аналітичних системах. Зіркова модель сховища даних полегшує інтеграцію з OLAP-інструментами і BI-системами – більшість сучасних засобів бізнес-аналітики (Power BI, Tableau тощо) оптимізовані під роботу з фактами і вимірами зіркової схеми, що спрощує побудову звітів для кінцевих користувачів. Архітектура може бути масштабована шляхом додавання нових фактів або вимірів для розширення аналітики на інші бізнес-процеси. У подальшому можливе впровадження повільно змінюваних вимірів та перехід до майже реального часу оновлення даних для ще швидшого отримання актуальної інформації. Крім того, механізм технічного журналювання ETL може бути інтегрований із системами моніторингу, що дозволить автоматично відстежувати метрики завантаження та оперативно сповіщати про відхилення. Загалом, реалізоване рішення створює гнучку й надійну основу для підтримки бізнес-аналітики на базі сховища даних, забезпечуючи якісну та своєчасну інформаційну підтримку управлінських рішень.

Список використаної літератури

1. Press G. *Data Quality: Best Practices for Accurate Insights*. URL: https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says (дата звернення: 15.06.2025)
2. Tsvenger D. *Leveraging data analytics to enhance customer loyalty and retention*. URL: https://paylode.com/articles/data-analytics-customer-loyalty-retention (дата звернення: 15.06.2025)
3. Inmon W. H. *Building the Data Warehouse*. URL: http://www.r-5.org/files/books/computers/databases/warehouses/W_H_Inmon-Building_the_Data_Warehouse-EN.pdf (дата звернення: 15.06.2025)
4. Kimball R., Ross M. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. URL: http://www.r-5.org/files/books/computers/databases/warehouses/Ralph_Kimball_Margy_Ross-The_Data_Warehouse_Toolkit-EN.pdf (дата звернення: 15.06.2025)
5. Golfarelli M., Rizzi S. *Data Warehouse Design: Modern Principles and Methodologies*. URL: https://cs09lects.wordpress.com/wp-content/uploads/2012/12/book-data-warehouse-design-golfarelli-_-rizzi.pdf (дата звернення: 15.06.2025)
6. Franconi E., Sattler U. *A Data Warehouse Conceptual Data Model for Multidimensional Aggregation*. URL: https://ceur-ws.org/Vol-19/paper13.pdf (дата звернення: 15.06.2025)
7. Томашевський В. М. *Особливості проектування гібридних сховищ даних з врахуванням джерел даних*. URL: https://science.lpnu.ua/sites/default/files/journal-paper/2017/jun/3398/2369.pdf (дата звернення: 15.06.2025)
8. Debbarma N., Nath G., Das H. *Analysis of Data Quality and Performance Issues in Data Warehousing and Business Intelligence*. URL: https://ijcaonline.org/archives/volume79/number15/13818-1862/ (дата звернення: 15.06.2025)
9. Шаховська Н. Б., Виклюк Я. І. *Сховища та простори даних – інформаційний фундамент систем прийняття рішень*. URL: http://nbuv.gov.ua/UJRN/etks_2012_8_17 (дата звернення: 15.06.2025)
10. Batini C., Scannapieco M. *Data Quality: Concepts, Methodologies and Techniques*. URL: https://link.springer.com/book/10.1007/3-540-33173-5 (дата звернення: 15.06.2025)
11. Ali S. M. F., Wrembel R. *From conceptual design to performance optimization of ETL workflows: current state of research and open problems*. URL: https://doi.org/10.1007/s00778-017-0477-2 (дата звернення: 15.06.2025)
12. *The DAMA Guide to the Data Management Body of Knowledge (DAMA-DMBOK2)*. URL: https://www.google.com.ua/books/edition/DAMA_DMBOK/YjacswEACAAJ (дата звернення: 15.06.2025)

References (transliterated)

1. Press G. *Data Quality: Best Practices for Accurate Insights*. Available at: https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says (accessed: 15.06.2025)
2. Tsvenger D. *Leveraging data analytics to enhance customer loyalty and retention*. Available at: https://paylode.com/articles/data-analytics-customer-loyalty-retention (accessed: 15.06.2025)
3. Inmon W. H. *Building the Data Warehouse*. Available at: http://www.r-5.org/files/books/computers/databases/warehouses/W_H_Inmon-Building_the_Data_Warehouse-EN.pdf (accessed: 15.06.2025)

4. Kimball R., Ross M. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. Available at: http://www.r-5.org/files/books/computers/databases/warehouses/Ralph_Kimball_Margy_Ross-The_Data_Warehouse_Toolkit-EN.pdf (accessed: 15.06.2025)
5. Golfarelli M., Rizzi S. *Data Warehouse Design: Modern Principles and Methodologies*. Available at: https://cs09lects.wordpress.com/wp-content/uploads/2012/12/book-data-warehouse-design-golfarelli-_-rizzi.pdf (accessed: 15.06.2025)
6. Franconi E., Sattler U. *A Data Warehouse Conceptual Data Model for Multidimensional Aggregation*. Available at: https://ceur-ws.org/Vol-19/paper13.pdf (accessed: 15.06.2025)
7. Tomashevskyi V. M. *Osoblyvosti proektuvannia hibrydnykh skhovyshch danykh z vrakhuvanniam dzherel danykh* [Features of designing hybrid data storage systems taking into account data sources]. Available at: https://science.lpnu.ua/sites/default/files/journal-paper/2017/jun/3398/2369.pdf (accessed: 15.06.2025)
8. Debbarma N., Nath G., Das H. *Analysis of Data Quality and Performance Issues in Data Warehousing and Business Intelligence*. Available at: https://ijcaonline.org/archives/volume79/number15/13818-1862/ (accessed: 15.06.2025)
9. Shakhovska N. B., Vykliuk Y. I. *Skhovyshcha ta prostory danykh – informatsiinyi fundament system pryiniattia rishen* [Data warehouses and data spaces – the information foundation of decision-making systems]. Available at: http://nbuv.gov.ua/UJRN/etks_2012_8_17 (accessed: 15.06.2025)
10. Batini C., Scannapieco M. *Data Quality: Concepts, Methodologies and Techniques*. Available at: https://link.springer.com/book/10.1007/3-540-33173-5 (accessed: 15.06.2025)
11. Ali S. M. F., Wrembel R. *From conceptual design to performance optimization of ETL workflows: current state of research and open problems*. Available at: https://doi.org/10.1007/s00778-017-0477-2 (accessed: 15.06.2025)
12. *The DAMA Guide to the Data Management Body of Knowledge (DAMA-DMBOK2)*. Available at: https://www.google.com.ua/books/edition/DAMA_DMBOK/YjacswEACAAJ (accessed: 15.06.2025)

Надійшла (received) 15.11.2025

UDC 004.65

I. K. BABICH, Postgraduate Student, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine; e-mail: Ihor.Babich@cs.khpi.edu.ua; ORCID: https://orcid.org/0009-0003-0433-4913
D. L. ORLOVSKYI, Candidate of Technical Sciences (PhD), Docent, National Technical University "Kharkiv Polytechnic Institute", Professor at the Department of Software Engineering and Management Intelligent Technologies, Kharkiv, Ukraine; e-mail: Dmytro.Orlovskyi@khpi.edu.ua; ORCID: https://orcid.org/0000-0002-8261-2988
A. M. KOPP, Doctor of Philosophy (PhD), Docent, National Technical University" Kharkiv Polytechnic Institute", Head of Software Engineering and Management Intelligent Technologies Department, Kharkiv, Ukraine, e-mail: Andrii.Kopp@khpi.edu.ua; ORCID: https://orcid.org/0000-0002-3189-5623

INFORMATION TECHNOLOGIES FOR THE INTEGRATION OF CUSTOMER AND CONSUMER DATA

Using the example of a book enterprise that combines the functions of a publisher, distributor, and retailer, it is shown how multi-channel operational activities lead to the accumulation of vast arrays of information in databases that are fragmented, incomplete, unstructured, and contain duplicates. This situation makes it impossible to effectively analyze customer behavior, including the accurate calculation of key performance indicators. The relevance of the work lies in reducing this critical gap between the volume of accumulated information and the business's ability to make effective management decisions based on it. The purpose of this work is to develop a methodological approach to creating a data warehouse based on the star schema architecture and to implement an adaptive ETL chain with built-in quality control rules. An analysis of modern data warehouse design methods was conducted, including the transition from the entity-relationship model to the star schema. Based on the structure of the transactional database and business requirements for data analysis, an analytical warehouse using the star schema was designed, and key facts and dimensions necessary to support comprehensive customer analytics were identified. To transfer data from the transactional system to the warehouse, an extract, transform, and load (ETL) process was developed, and its logic was described: data extraction from sources, its cleaning and transformation in a staging area, and loading into the target warehouse tables. The effectiveness of the developed processes was evaluated based on event log data. The analysis results confirm the reliability and high performance of the proposed solution. The approach proposed in the article provides automated, reliable, and efficient updating of the data warehouse, creating a single source of truth for business analytics.

Keywords: database, data warehouse, data integration, star schema, dimension/fact tables, ETL.

Повні імена авторів / Author's full names

Автор 1 / Author 1: Бабіч Ігор Костянтинович / Babich Ihor Kostiantynovych
Автор 2 / Author 2: Орловський Дмитро Леонідович / Orlovskyi Dmytro Leonidovych
Автор 3 / Author 3: Копп Андрій Михайлович / Kopp Andrii Mykhailovych