вимагає використання специфічних методів регуляризації або перегляду всієї структури алгоритму та його блок-схеми.

Висновки. Проведене дослідження показало, що похибки обчислень є невід'ємним і часто визначальним фактором будь-якого комп'ютерного моделювання. Точність результату залежить як від обраного математичного апарату (що генерує похибку усічення) [3], так і від апаратних та програмних обмежень обчислювальної системи (похибки округлення стандарту IEEE 754) [5]. Виявлено фундаментальний обчислювальний парадокс: надмірне зменшення кроку або збільшення кількості ітерацій для мінімізації похибки методу неминуче призводить до катастрофічного накопичення похибок машинного округлення.

Існує чітка математична межа (оптимальний крок), перетин якої погіршує роботу алгоритму. Протидія цим явищам вимагає не просто сліпого переведення математичних формул у програмний код, а свідомого проєктування стійких алгоритмів: використання компенсаційних методів підсумовування, рефакторингу алгебраїчних виразів для уникнення катастрофічного скасування та встановлення динамічних умов виходу з ітераційних циклів на основі машинного епсилон [1; 2]. Отже, розробка ефективного та надійного програмного забезпечення вимагає від фахівців глибокого розуміння природи машинної арифметики та обов'язкової попередньої оцінки стійкості застосовуваних чисельних методів.

Abstract. This article investigates the main sources of errors when using numerical methods in computer algorithms. A thorough analysis of method errors (truncation) and rounding errors is conducted; the concepts of absolute and relative errors are examined, as well as the impact of the IEEE 754 standard for the representation of real numbers on the accumulation of errors. Based on computational data, the impact of these errors on the accuracy of iterative computation results is demonstrated using the example of the Maclaurin series expansion of functions and the numerical differentiation problem. The problem of catastrophic cancellation and algorithmic approaches to improving the computational stability of software are considered. The results obtained allow us to assess the limits of application of approximate computational methods, determine optimal step parameters, and optimize algorithms to minimize loss of accuracy under the conditions of finite precision in machine arithmetic.

Keywords: numerical methods, absolute error, relative error, rounding error, truncation error, computer calculations, machine epsilon, numerical stability, Kahan summation algorithm.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Комп'ютерне моделювання процесів та систем. Чисельні методи: підручник / С. П. Вислоух, О. В. Волошко, Г. С. Тимчик, М. В. Філіппова. Київ: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2021. 228 с.

2. Третиник В. В., Любашенко Н. Д. Методи обчислень. Частина 1. Чисельні методи алгебри: навч. посіб. Київ: КПІ ім. Ігоря Сікорського, 2023. 182 с.

3. Голубєва К. М., Кашпур О. Ф., Клюшин Д. А. Чисельні методи: навч. посіб. Київ: ВПЦ «Київський університет», 2022. 145 с.

4. Sauer T. Numerical Analysis. 3rd ed. (Updated). Pearson, 2021. 664 p.

5. Методи та алгоритми комп'ютерних обчислень. Теорія і практика: підручник / Р. Н. Квєтний, Я. В. Іванчук, І. В. Богач, О. Ю. Софина, М. В. Барабан. Вінниця: ВНТУ, 2023. 280 с.

УДК 004.738.5:004.896.2:004.421.2

ЗБІР ТА СТАНДАРТИЗАЦІЯ ГЕТЕРОГЕННИХ ДАНИХ У ХМАРНИХ АНАЛІТИЧНИХ СЕРЕДОВИЩАХ

Д. Ю. Кохан, Т. В. Січко

Анотація. У статті розглянуто проблему інтеграції різнорідних за форматом і походженням даних у єдине хмарне сховище. Для її розв'язання запропоновано ETL-пайплайн на основі Apache Airflow, Google BigQuery та dbt. Описано трирівневу модель трансформації (Raw – Staging – Mart) і ключові операції стандартизації: уніфікацію типів, видалення дублікатів, заповнення пропусків і нормалізацію схем. Показано, що якість підготовлених даних визначає точність аналітичних метрик.

Ключові слова: гетерогенні дані, ETL-пайплайн, BigQuery, dbt, стандартизація даних.

Вступ. Більшість сучасних підприємств зберігають операційні дані одразу в кількох несумісних системах: реляційних базах даних, хмарних сервісах, файлових звітах і зовнішніх API.

Кожна з них формує власну структуру і формат, тому навіть прості запитання на кшталт «Скільки продажів було минулого місяця?» потребують ручного зведення даних із кількох джерел. Неузгодженість схем і семантики знижує якість аналітичних моделей і збільшує час на підготовку звітів [1; 2]. На практиці це виражається в тому, що аналітики витрачають значну частину робочого часу не на аналіз, а на узгодження форматів і усунення суперечностей між джерелами. Розв'язати цю проблему дає змогу ETL-процес (Extract, Transform, Load) – набір методів вилучення, перетворення та завантаження інформації в єдине сховище [3].

Проблема гетерогенності даних досліджується у роботах, присвячених архітектурам озер даних [1], управлінню пайплайнами [2] та методам очищення інформації [6]. У роботі [3] розглянуто підходи до побудови озер даних як єдиного репозиторію з розмежуванням зон зберігання. Автори [4] класифікують проблеми сумісності схем під час інтеграції різнорідних масивів. У [5] показано, що консолідація у хмарному сховищі типу Lakehouse скорочує час аналітичних запитів і спрощує управління метаданими. Попри наявність цих рішень, питання побудови повного циклу обробки – від збору різнорідних джерел до готових метрик – у контексті сучасних хмарних інструментів залишається відкритим.

Мета статті – описати архітектуру та методи збору й стандартизації гетерогенних даних на основі стеку Airflow – BigQuery – dbt.

Основна частина. Під гетерогенними даними розуміють масиви, що відрізняються за структурою, форматом збереження та способом доступу [4]. За ступенем структурованості їх поділяють на три категорії. Структуровані дані – реляційні таблиці SQL-баз із фіксованою схемою (PostgreSQL, MySQL). Напівструктуровані дані – ієрархічні формати JSON та XML, що надходять через REST API або формуються у вигляді файлів вивантаження. Неструктуровані дані – довільні файлові формати: CSV, Excel-звіти, текстові документи.

Для кожної категорії характерні специфічні проблеми сумісності. По-перше, відмінності у форматах дат і часових зон: одне джерело може зберігати дату у форматі DD.MM.YYYY, інше – у вигляді Unix timestamp. По-друге, неоднорідне кодування рядків: поля з однаковою семантикою (наприклад, назва країни) можуть містити різні варіанти написання або абревіатури. По-третє, наявність дублюючих записів унаслідок паралельного введення даних у кілька систем. По-четверте, пропущені значення, що виникають через незаповнені поля або збої під час вивантаження. Нарешті, несумісність схем між джерелами – однойменні поля можуть мати різні типи даних або різну семантику в різних системах. Через ці відмінності дані з різних джерел не можна об'єднати без попередньої обробки.

Запропонована архітектура ETL-пайплайну (рис. 1) реалізує повний цикл обробки даних і складається з чотирьох функціональних рівнів. На рівні оркестрації Apache Airflow виконує функцію централізованого планувальника: формує DAG-графи (Directed Acyclic Graph) для кожного джерела, контролює часові розклади запуску та забезпечує обробку збоїв і повторних спроб виконання завдань. Ключовою перевагою Airflow є декларативний опис залежностей між задачами, що дає змогу візуалізувати та відлагоджувати складні пайплайни. Кожен DAG відображає повний ланцюжок операцій для конкретного джерела: від моменту вилучення даних до підтвердження успішного завантаження. Airflow не зберігає дані – він управляє виключно потоком їх вилучення та передачі між компонентами.

Рівень завантаження представлено Google BigQuery – колонковою OLAP-системою, що призначена для аналітичних запитів над великими обсягами даних. На цьому рівні сирі дані потрапляють у Raw-шар – незмінений знімок джерела на момент вилучення. BigQuery забезпечує горизонтальне масштабування, партиціювання таблиць за датою, кластеризацію та вбудовану підтримку SQL-сумісних запитів [5]. Відокремлення обчислення від зберігання дає змогу масштабувати запити незалежно від обсягу даних. Це означає, що один і той самий запит однаково виконується як на таблиці в тисячу рядків, так і на таблиці в кілька мільярдів – без змін у коді і без ручного налаштування кластера.

Рівень трансформації реалізується засобами dbt (data build tool) – фреймворка для декларативного визначення SQL-моделей безпосередньо у сховищі даних. dbt перетворює сирі дані у стандартизовані структури, виконує тести якості та документує залежності між моделями у

вигляді лінійного графу [7]. Вихідний рівень – Power BI – забезпечує візуалізацію готових метрик у вигляді інтерактивних дашбордів для кінцевих користувачів.

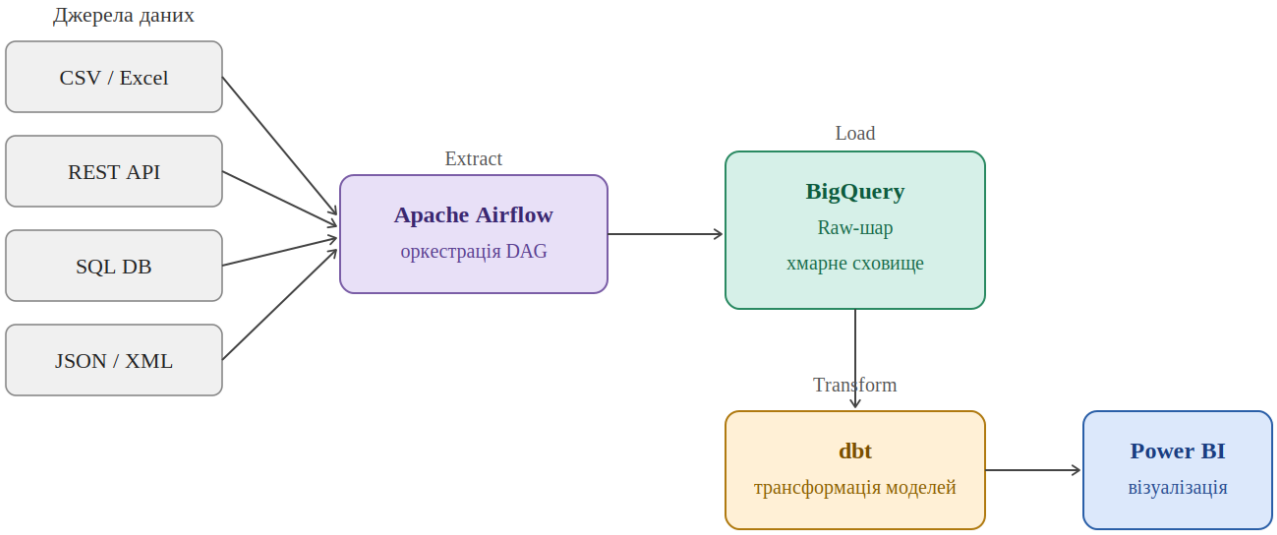

Рис. 1. Архітектура ETL-пайплайну для збору та обробки гетерогенних даних

Процес стандартизації організовано за трирівневою архітектурою шарів dbt (рис. 2). Raw-шар зберігає дані у вигляді, ідентичному джерельному. Жодних змін до даних на цьому рівні не вноситься – це гарантує відтворюваність і можливість повторної обробки за будь-яких змін логіки трансформацій. Незмінність Raw-шару є критично важливою для аудиту та відновлення даних у разі виявлення помилок на наступних етапах.

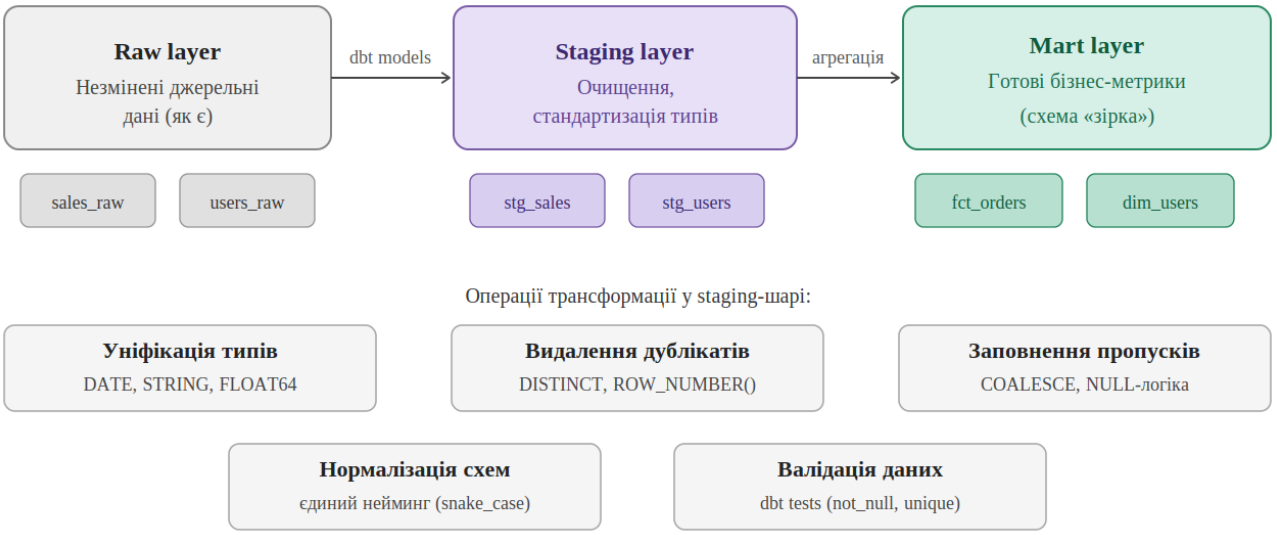

Рис. 2. Шари трансформації даних у dbt та операції стандартизації

Staging-шар є ключовим рівнем стандартизації. Тут виконуються такі операції: уніфікація типів даних (приведення полів дат до єдиного формату DATE, числових значень – до FLOAT64, текстових – до STRING); видалення дублікатів за допомогою конструкцій DISTINCT або віконних функцій ROW_NUMBER() з розбивкою за ключовими полями; заповнення пропущених значень засобами COALESCE з логікою підстановки значень за замовчуванням залежно від бізнес-правил; нормалізація іменування полів – приведення до єдиного регістру і конвенції snake_case для забезпечення консистентності між джерелами. Додатково на цьому рівні виконуються вбудовані тести dbt: перевірки на not_null (відсутність порожніх значень у ключових полях), unique (унікальність первинних ключів) і relationships (цілісність зовнішніх ключів між таблицями) [7].

Mart-шар консолідує стандартизовані дані у бізнес-орієнтовані моделі. Тут формуються таблиці фактів (fct_orders, fct_revenue) та виміри (dim_users, dim_products), що разом реалізують схему «зірка» – оптимальну для аналітичних запитів у колонкових СУБД. Схема «зірка» мінімізує кількість операцій JOIN під час виконання агрегаційних запитів, що безпосередньо впливає на швидкість побудови звітів у Power BI. Саме Mart-шар є безпосереднім джерелом для підключення інструментів візуалізації. Контроль якості даних на кожному шарі реалізується через метадані: dbt зберігає версії моделей, журнали виконання та результати тестів, що дає змогу виявляти регресії якості між запусками пайплайну [3].

Управління метаданими в dbt реалізується через YAML-описи моделей, де фіксуються призначення полів, очікувані типи та бізнес-правила. Це робить пайплайн самодокументованим і спрощує роботу з ним у разі змін у команді. Якщо поле змінює тип або логіку обчислення, достатньо оновити YAML-опис – dbt автоматично перевірить відповідність і сигналізує про невідповідності ще до запуску. Поєднання Airflow, BigQuery і dbt формує відтворюваний і версіонований цикл обробки даних, що відповідає сучасним практикам DataOps [2].

Висновки. У статті описано архітектуру ETL-пайплайну для збору та стандартизації гетерогенних даних у хмарному середовищі на основі стеку Airflow – BigQuery – dbt. Визначено три категорії гетерогенних джерел і описано типові проблеми їх сумісності – розбіжності форматів, дублікати, пропущені значення та несумісність схем. Airflow забезпечує оркестрацію збору, BigQuery – зберігання у незмінному Raw-шарі, dbt – трирівневу трансформацію з контролем якості, а Power BI – фінальну візуалізацію метрик. Показано, що стандартизація на рівні Staging-шару безпосередньо визначає достовірність аналітичних результатів: дані, що пройшли повний цикл очищення, дають змогу будувати метрики без ручних виправлень з боку аналітика. Подальші дослідження доцільно спрямувати на автоматизацію моніторингу якості даних у реальному часі, розробку метрик оцінювання повноти й актуальності інформації у сховищі та інтеграцію механізмів виявлення аномалій безпосередньо в пайплайн.

Abstract. The article addresses the problem of integrating heterogeneous data from multiple sources into a unified cloud data warehouse for business analytics purposes. An ETL pipeline architecture based on Apache Airflow, Google BigQuery, and dbt is proposed, providing automated data collection, cleansing, and standardization. A three-layer transformation model (Raw – Staging – Mart) is described along with key standardization operations: type unification, deduplication, null handling, and schema normalization. It is demonstrated that data quality at the preparation stage directly determines the accuracy of analytical metrics. Practical application of the proposed architecture enables full reproducibility of the data processing cycle and consistent quality control at each transformation stage.

Keywords: heterogeneous data, ETL pipeline, BigQuery, dbt, data standardization.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Data Lakes: A Survey of Functions and Systems / R. Hai, C. Koutras, C. Quix, M. Jarke. *IEEE Transactions on Knowledge and Data Engineering.* 2023. Vol. 35, № 12. P. 12571–2590. DOI: 10.1109/TKDE.2023.3270101.

2. Munappy A. R., Bosch J., Olsson H. H. Data Pipeline Management in Practice: Challenges and Opportunities. *Lecture Notes in Computer Science.* 2020. Vol. 12562. P. 168–184. DOI: 10.1007/978-3-030-64148-1_11.

3. Leveraging the Data Lake: Current State and Challenges / C. Giebler, C. Gröger, E. Hoos, H. Schwarz, B. Mitschang. *Lecture Notes in Computer Science.* 2019. Vol. 11708. P. 179–188. DOI: 10.1007/978-3-030-27520-4_13.

4. Yohannis T. K., Bandung Y., Purnama J. Heterogeneous data integration: Challenges and opportunities. *PLOS ONE.* 2024. Vol. 19, № 9. e0308101. DOI: 10.1371/journal.pone.0308101.

5. .Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics / M. Armbrust, A. Ghodsi, R. Xin, M. Zaharia. *Proceedings of CIDR 2021.* 2021. URL: https://people.eecs.berkeley.edu/~matei/papers/2021/cidr_lakehouse.pdf

6. Ilyas I., Abo Kh. M., Rekatsinas T. Data Cleaning: Overview and Emerging Challenges. *ACM SIGMOD.* 2022. URL: https://dl.acm.org/doi/10.1145/3514221.3522563

7. dbt documentation: How dbt works. *dbt Labs.* 2024. URL: https://docs.getdbt.com/docs/introduction