«Системні технології» 5 (124) 2019 «System technologies»
DOI 10.34185/1562-9945-5-124-2019-08
УДК . 004.4:6

В.В. Гнатушенко, М.Г. Виноградов, О.В. Виноградова

РОЗРОБКА МОДЕЛІ ЛОГУВАННЯ В ETL СИСТЕМАХ

Анотація. *У статті розроблено модель логування запису даних по всім процесам ETL системи та її складовим: збереження вхідних даних, параметрів та результатів виконання. Основним підходом до логування є запис всієї необхідної інформації у кінцеву систему накопичення, яка має обмеження по часу життя інформації (перезапуск/вимкнення програми, обмеження простору). Модель показує свою ефективність на відміну від вже знаних систем, таких як використання тригерів чи вбудованих в систему розробки моделей.*
Ключові слова*: модель, логування, ETL система.*

Постановка проблеми. Сучасні підприємства та організації використовують технологічні системи, в складі яких містяться реляційні бази даних, головною перевагою яких є безвідмовність, стабільність та можливість швидкого отримання інформації.

Узгодженість і адаптивність є ключовими поняттями в концепції інформаційної архітектури, яка визначає набір вимог, принципів і моделей, необхідних для гнучкого спільного використання та обміну інформацією. Це приводить до створення послідовних чи паралельних моделей процесів руху даних, які можуть бути трансформовані, згруповані чи доповнені у ETL-моделях.

Аналіз останніх досліджень. Дані, що завантажуються у систему, як правило, потрібно не тільки зберігати всередині, а передавати для обробки і аналізу. Для цього використовуються сховища даних (СД або DWH - Data Warehouse), розроблені і орієнтовані спеціально для підготовки звітів і бізнес-аналізу, з метою підтримки прийняття рішень на підприємстві [1]. Виділяють три етапи в процесі роботи з даними:

1. Витяг (Extract) – відбираються і описуються дані зовнішніх джерел (починають формуватися метадані СД), які повинні зберігатися в СД (релевантні дані).

2. Перетворення (Transform) – релевантні дані перетворюються в формат представлення даних в СД, правила перетворення зберігаються в метаданих СД, формуються ключові поля таблиць фізичної структури СД, виконується очищення даних.

3. Завантаження (Load) – дані завантажуються в СД, виконується побудова агрегатів.

Ці три етапи і складають абревіатуру ETL - одного з основних процесів управління даними при отриманні їх з множини джерел і завантаження в СД, з метою отримання достовірної інформації (рисунок 1). Процес ETL реалізується шляхом або розробки програми ETL, або створення комплексу вбудованих програмних процедур, або використання ETL-інструментарію [2].

ETL-система повинна відповідати наступним характеристикам: незалежність платформи та масштабованість. Для обробки великих обсягів даних повинні бути доступні три варіанти:

Паралелізм: дозволяє паралельно запускати множину потоків, використовуючи сучасні багатоядерні апаратні архітектури;

Поділ: дозволяє інструменту ETL використовувати переваги схем розподілу даних по паралельним потокам;

Кластеризація: дозволяє процесу ETL розділити робоче навантаження на кілька комп'ютерів чи процесорів [5].

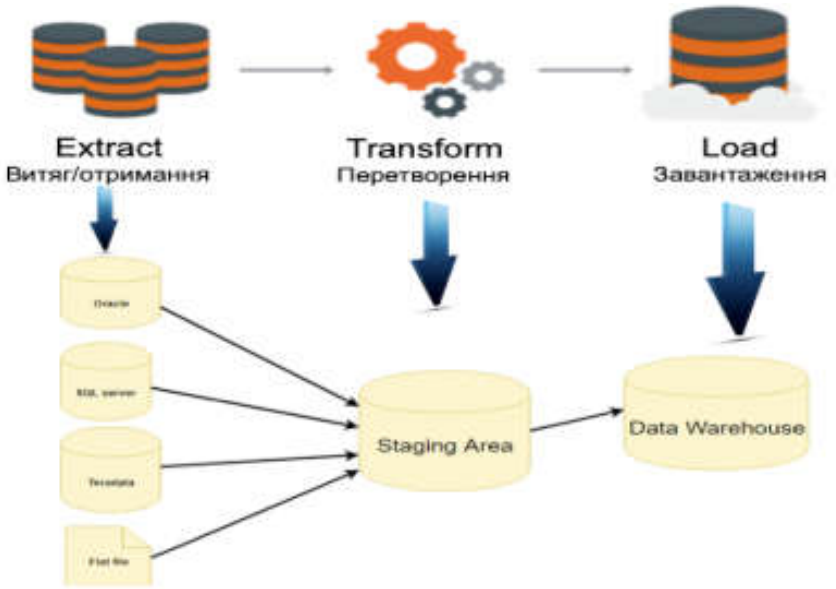

Рисунок 1 – ETL система

Будь-який процес потребує логування та фіксації всіх етапів та всіх процесів (таблиця 1). Основним підходом до логування є запис всієї необхідної інформації у кінцеву систему накопичення. Так за системою зберігання інформації можна виділити наступні системи запису:

- кеш процесу,
- файлова система,
- база даних (локальна чи хмарна).

Зберігання даних у кеші процесу реалізується у об'єктних системах. Недоліком цього підходу є час життя інформації. Тобто при перезапуску чи вимкненні програми всі дані будуть загублені та/або операційна пам'ять обмежена та на багато порядків менше ніж доступна пам'ять на жорсткому диску.

Збереження логів у файлову систему є дуже поширеним методом логування. Всі операції, дії, помилки чи транзакції записуються до файлової системи. Мінусом є неможливість запису чи більш складний процес запису, при використанні складних процедур чи функцій на стороні бази даних.

Системи логування з використанням баз даних передбачають записи операцій у базу даних, створюються окремі таблиці. Перевагою такого підходу є те, що таблиця бази даних не є тимчасовою та існує можливість отримувати дані за запитом чи створення окремої системи для швидкого отримання звітів. Також ця таблиця буде доступна після відключення програми або перевантаження та дані збережуться. Недоліком є те, що потрібно у процесах, які відбуваються не на стороні бази даних, створити підключення до бази процедур та функції.

Таблиця 1

Типи логування

Приналежніть	За місцем зберігання	За характером інформації
Внутрішні	Кеш процесу	Логування помилок
Зовнішні	Фаилова система	Логування кінцевих результатів
	База даних	Запис усіх даних

 ISSN 1562-9945

На сьогодні на ринку існує велика кількість можливих реалізацій ETL-систем, таких як SSIS, JAVA, talend, BigData Hadoop. Найбільш використані такі системи легування:на базі тригерів, вбудовані системи та платні зовнішні системи. Але всі ці системи стикаються з такою проблемою, як потреба логування різноманітних елементів.

Розробка системи запису (логування) інформації про процеси, їх етапи, характеристики, параметри та результати є актуальної задачею, що дозволить користувачам користуватися даними у будь-який час, система не буде потребувати зберігання даних у оперативній пам'яті та не буде потребувати додаткового устаткування.

Формулювання цілей статті (постановка завдання). Метою даного дослідження є розробка моделі логування процесів ETL-системи будь-якої складності та рівнем вкладеності для використання у складі інформаційної системи підприємства.

Основна частина. На сьогодні майже всі процеси та технології мають так чи інакше хоч якусь систему логування, тому і виникає питання як впровадити таку систему, яка буде працювати з будь-якими додатками. Система, що розробляється, повинна мати можливість імплементації в будь-яку технологію чи у вже працюючий функціонал, будь-яку реляційну базу даних. Тобто, цю систему можуть використовувати користувачі як з Oracle, так з MS SQL, PostgreSQL, Sybase, Informix чи іншими.

Система зберігає інформацію про процеси, етапи та результати виконання цих процесів. Головною її перевагою є те, що процеси можуть бути розподілені по різноманітним системам але результати будуть доступні та прозорі в одній базі даних. Ще однією перевагою є те, що логуватись можуть навіть внутрішні логічні складові процесів чи кроків ETL-моделі, наприклад логічна частина .Net чи Java методу, чи логічна частина збереженої процедури на стороні бази даних. Дана система може бути розширена розробниками для більшого розгалуження та для збільшення різноманітностей сутностей. Проаналізувавши поставлені задачі, вирішено реалізувати задану систему на базі Microsoft SQL Server Database та за допомогою інструментарію розробки Microsoft SQL Server Management Studio, мови C# .NET та Microsoft Visual Studio.

Головними об’єктами бази даних виступають дві основні таблиці з даними по процесу та код функціоналу для завантаження. Для тестування додатково розроблено дві таблиці та один тригер на стороні бази даних.

В роботі розроблено систему ETL, яка завантажує дані з двох джерел та вивантажує дані у базу даних, запускає три процедури та вивантажує результат у два різних вихідні файли. Структура системи містить (рисунок 2):

1. В кожній ETL-системі є події (events). Це деякий функціонал, який спрацьовує при початку процесу, завершенні, помилці і т.д. Саме на ці події налаштовується система.

2. При спрацюванні події, в системі використовується виклик блоку коду, та підключення до бази даних.

3. В базі даних створені дві таблиці, одна по загальному запуску ETL-системи та друга – по кожному процесу чи блоку. Цей виклик може бути імпортований в будь-який блок коду, процедуру, чи будь-який функціонал

4. У лог процесі (блоці) можливо включити будь-яку інформацію, будь-які параметри, та статуси.

5. В базу даних записується інформація про процес чи блок коду.

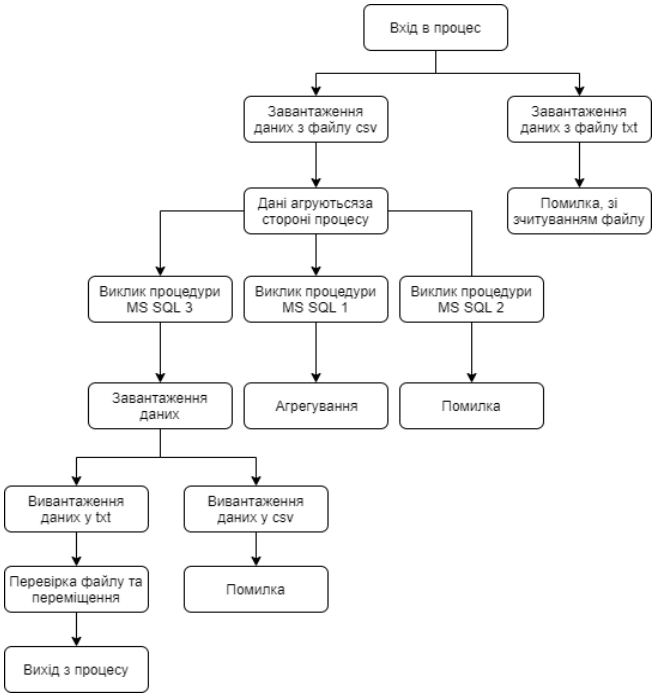

Рисунок 2 – Схема ETL процесу для легування

 ISSN 1562-9945

Призначенням системи є не логічність чи цілісність даних, а можливість розробленої моделі для логування будь-яких процесів, збереження вхідних даних, параметрів та результатів виконання кроків системи.

При запуску ETL отримуємо унікальний ідентифікатор поточного процесу, який зберігається до кінця виконання та передається на кожний блок подій кожного кроку.

Таблиця 2

Логування процесу запису

1	При запуску ETL системи отримаємо унікальнии ідентифікатор запуску і записуємо у таблицю dbo.Opr_Run	Записуємо у runId дані – унікальнии ідентифікатор oprRunId
2	Для кожного блоку перед виконанням записуємо дані у dbo.Opr_Log з oprRunId із першого кроку, та інші дані.	Отримаємо дані – унікальнии ідентифікатор отримуємо oprId
3	Якщо виникає помилка, то система в таблиці dbo.Opr_Log оновлює дані по oprId з попереднього пункту та записує помилку.	Зберігається статус помилки. Для того, щоб по завершені не було оновлення до позитивного статусу.
4	Після завершення процесу – оновлюємо таблицю dbo.Opr_Log зі статусом – «коректно» по полю oprId – унікальному ідентифікатору	Якщо є помилка в попередньому кроці – цей крок не виконується.
5	Після завершення всієї системи ETL йде оновлення dbo.Opr_Run по раніше генерованому oprRunId	Якщо процес завершується – то це буде в будь-якому випадку успішний статус. Якщо вся система закінчується помилкою – лише тоді буде статус помилки.

В таблиці запусків є унікальний ідентифікатор, за яким можна однозначно визначити запуск, та в таблиці логів є унікальний ідентифікатор для кожного підпроцесу, навіть для найменшої складової частини буде свій унікальний код та дані по цьому процесу (кроку).

Розробник чи адміністратор системи може включити будь-який блок коду, операцію у логування, запис до таблиці. Навіть, якщо є бізнес-потреба в логуванні найменших частин процесу, вони можуть бути записані у таблицю логів.

Розроблено модель логування запису даних по всім процесам ETL системи та деяким їх складовим. Модель показує свою ефективність на відміну від вже знаних систем, таких як використання тригерів чи використання вбудованих в систему розробки моделей.

При використанні системи логування на базі тригерів можливо отримати неповні результати при залежності однієї таблиці від іншої за допомогою тригеру – в залежну таблицю будуть попадати лише успішні записи. Помилки чи ролбеки не будуть логуватись.

При використанні вбудованих систем немає можливості записати внутрішні процеси, чи блоки, які підлягають логуванню. Так, наприклад, крок з виконанням зовнішньої збереженої процедури на стороні бази даних може бути записаний лише зовнішні дані (вхідні-вихідні параметри). А якщо потрібно залогувати внутрішні блоки (окремі DML команди), то це неможливо.

В системі SSIS є можливість вбудувати систему логування та запису процесів. Більшість систем ETL включають в себе додаткові процеси. Так в розробленій системі ETL є такі блоки, як використання коду .NET C#, виклик складних навантажених збережених процедур на стороні бази даних, використання BAT файлів та інші вбудовані блоки. Так і в будь-якій іншій системі можуть бути використані зовнішні блоки. В системах Jenkins, Informatica чи Talend можуть бути використані збережені процедури бази даних, чи BigData, чи використані блоки коду Java, Pyton, JavaScript чи .NET в залежності від системи. І ці блоки можуть мати навантажений заплутаний код зі складною логікою, яку потрібно логувати, вхідні параметри чи проміжні результати, від яких залежить результат виконання блоку чи кроку системи, які також потрібно записувати. Тому використання представленої моделі задовольнить вимогу запису (логування) даних у будь-якій системі та з використанням будь-якої реляційної бази даних. Навіть можливо записувати дані в базу NoSql, але вико-

ристання таких баз даних для процесів логування не є найкращим вибором.

Висновки та перспективи подальших досліджень. Проаналізовано та досліджено існуючи моделі логування систем ETL та створена система логування за допомогою реляційної бази даних та подій у системі ETL на базі SSIS пакету, яка дозволяє виводити детальну інформацію по всім крокам ETL-системи з параметрами, результатами, та блокам. Система легко може підлаштовуватись під будь-яку потребу кінцевого користувача та має переваги у порівнянні з існуючими моделями на базі тригерів та вбудованими.

ЛИТЕРАТУРА / ЛІТЕРАТУРА

1. Jos van Dongen, Matt Casters, Roland Bouman. Pentaho Kettle Solutions: Building Open Source ETL Solutions with Pentaho Data Integration. – John Wiley & Sons, 2010. – 674p./ ISBN: 9780470942420

2. Точилкина Т.Е. Хранилища данных и средства бизнес-аналитики: учебное пособие / Т.Е. Точилкина, А.А. Громова – М.: Финансовый университет, 2017. – 161 с.

3. Алексей Полев ETL – технология, сопутствующая любой BI-инициативе // Jet Info – 29 марта 2012. - №2.

4. Ralph Kimball, Joe Caserta. The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. — Wiley Publishing, Inc., 2004. – 491p.

REFERENCES

1. Jos van Dongen, Matt Casters, Roland Bouman. Pentaho Kettle Solutions: Building Open Source ETL Solutions with Pentaho Data Integration. – John Wiley & Sons, 2010. – 674p./ ISBN: 9780470942420

2. Tochilkina T.E. Data Warehousing and Business Intelligence Tools: Textbook / T.E. Tochilkina, A.A. Gromova - Moscow: Financial University, 2017 .-- 161 p.

3. Alexey Polev ETL - a technology that accompanies any BI-initiative // Jet Info - March 29, 2012. - No.

4. Ralph Kimball, Joe Caserta. The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. — Wiley Publishing, Inc., 2004. – 491p.

Received 28.10.2019.
Accepted 04.11.2019.

Розробка моделі логування в ETL системах

Три етапи складають ETL (Extract, Transform, Load) - одного з основних процесів управління даними при отриманні їх з множини джерел і завантаження в сховище даних (СД), з метою отримання достовірної інформації. Процес ETL реалізується шляхом або розробки програми ETL, або створення комплексу вбудованих програмних процедур, або використання ETL-інструментарію.

Будь-який процес потребує логування та фіксації всіх етапів та всіх процесів. Основним підходом до логування є запис всієї необхідної інформації у кінцеву систему накопичення.

Метою даного дослідження є розробка моделі логування процесів ETL-системи будь-якої складності та рівнем вкладеності для використання у складі інформаційної системи підприємства.

Проаналізувавши поставлені задачі, вирішено реалізувати задану систему на базі Microsoft SQL Server Database та за допомогою інструментарію розробки Microsoft SQL Server Management Studio, мови C# .NET та Microsoft Visual Studio. Головними об'єктами бази даних виступають дві основні таблиці з даними по процесу та код функціоналу для завантаження. Для тестування додатково розроблено дві таблиці та один тригер на стороні бази даних.

Система зберігає інформацію про процеси, етапи та результати виконання цих процесів. Головною її перевагою є те, що процеси можуть бути розподілені по різноманітним системам але результати будуть доступні та прозорі в одній базі даних. Ще однією перевагою є те, що логуватись можуть навіть внутрішні логічні складові процесів чи кроків ETL-моделі, наприклад логічна частина .Net чи Java методу, чи логічна частина збереженої процедури на стороні бази даних. Дана система може бути розширена розробниками для більшого розгалуження та для збільшення різноманітностей сутностей.

Проаналізовано та досліджено існуючи моделі логування систем ETL та створена система логування за допомогою реляційної бази даних та подій у системі ETL на базі SSIS пакету, яка дозволяє виводити детальну інформацію по всім крокам ETL-системи з параметрами, результатами, та блокам. Система легко може підлаштовуватись під будь-яку потребу кінцевого користувача та має переваги у порівнянні з існуючими моделями на базі тригерів та вбудованими.

Development of logging model in ETL systems

The three steps constitute ETL (Extract, Transform, Load) - one of the main data management processes after receiving data from multiple sources and uploaded to a data warehouse (DWH) in order to get reliable information. The ETL process implements in a different way: by developing an ETL program, by creating a set of embedded program procedures, or by using ETL tools.

Any process requires logging and fixation of all stages and all processes. The basic method to logging is to record all the necessary information in the final cumulative system.

The purpose of this investigation is to develop a model of logging ETL-system processes of any complexity and level of nesting for use in the enterprise information system.

After the task has been analyzed, it was decided to implement the system on the basis of Microsoft SQL Server Database and using development tools of Microsoft SQL Server Management Studio, C # .NET language and Microsoft Visual Studio.

 ISSN 1562-9945

After analyzing the tasks, it was decided to implement the given system on the basis of Microsoft SQL Server Database and using the development tools of Microsoft SQL Server Management Studio, C # .NET language and Microsoft Visual Studio. The main objects of the database are the two main tables with information about processes and the functional code of downloading. Two tables and one trigger on the database side have been further developed for testing.

The system keeps information about the processes, steps and results of these processes. Its main advantage is that processes can be distributed across systems, but the results will be accessible and transparent in one database. Another advantage is that even the internal logical components of the ETL model processes or steps, such as the logical parts of the .Net or Java methods, or the logical parts of the stored procedures on the database side, can be logged in. This system can be extended by developers for greater expansion and to increase the diversity of entities. The existing models of logging of ETL systems has been analyzed and investigated, new logging system was created with the help of relational database and events in ETL system on the basis of SSIS package, which allows to display detailed information about all steps of ETL-system with parameters, results, and blocks. The system can easily adapt to any end-user's needs and has advantages over existing trigger-based and built-in models.

Гнатушенко Виктория Владимировна - д.т.н., доцент, профессор кафедры информа-ционных технологий и систем Национальной металлургической академии Украины.

Виноградов Николай Геннадьевич - магистр кафедры информационных технологий и систем Национальной металлургической академии Украины.

Виноградова Оксана Владимировна - магистр кафедры информационных технологий и систем Национальной металлургической академии Украины.

Гнатушенко Вікторія Володимирівна - д.т.н., доцент, професор кафедри інформаційних технологій і систем Національної металургійної академії України.

Виноградов Миколай Геннадійович – магістр кафедри інформаційних технологій і систем Національної металургійної академії України.

Виноградова Оксана Володимирівна – магістр кафедри інформаційних технологій і систем Національної металургійної академії України.

Hnatushenko Viktoriia - doctor of technical sciences, associate professor, professor of the department of information technology and systems of the National Metallurgical Academy of Ukraine.

Vinogradov Nikolay - master of the department of information technologies and systems of the National Metallurgical Academy of Ukraine.

Vinogradova Oksana - master of the department of information technologies and systems of the National Metallurgical Academy of Ukraine.