ตัวแบบการควบรวมดาต้าไซโลแบบมีโครงสร้างด้วยทะเลสาบข้อมูล

An Integration Model on Data Lake for Solving Structured Data Silo Problems

ศศิธร สุชัยยะ[1*] สมนึก คีรีโต[2] และ สรพงษ์ เรือนมณี[3]

Sasithorn Suchaiya[1*], Somnuk Keretho[2] and Sorapong Ruanmanee[3]

[1]ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ, คณะวิทยาศาสตร์และวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตเฉลิมพระเกียรติ จังหวัดสกลนคร 47000

[2]ภาควิชาวิศวกรรมคอมพิวเตอร์, คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ (วิทยาเขตบางเขน) กรุงเทพฯ 10900

[3]สำนักบริหารการศึกษา, มหาวิทยาลัยเกษตรศาสตร์ (วิทยาเขตบางเขน) กรุงเทพฯ 10900

[1]Department of Computer and Information Science, Faculty of Science and Engineering Kasetsart University, Chalermphrakiat Sakon Nakhon Province Campus, Sakon Nakhon, 47000, Thailand.

[2]Department of Computer Engineering, Faculty of Engineering, Kasetsart University, Bangkok, 10900, Thailand

[3]Office of Education Administration Kasetsart University, Bangkok, 10900, Thailand

[*]Corresponding Author, E-mail: ssrithon.s@ku.th

Received: 7 November 2021 | Revised: 25 February 2022 | Accepted: 28 February 2022

บทคัดย่อ

ดาต้าไซโล (Data Silos) เป็นปัญหาที่เกิดขึ้นในองค์กรทั้งภาครัฐและเอกชน เนื่องจากองค์กรมีการแบ่งหน้าที่การทำงานออกเป็นหลายฝ่าย แต่ละฝ่ายมีการทำงานที่แยกกันอย่างชัดเจน แต่ละฝ่ายมีการพัฒนาโปรแกรม แอปพลิเคชัน และการจัดเก็บข้อมูลที่ซ้ำซ้อนกัน โดยมีการแชร์หรือแลกเปลี่ยนข้อมูลระหว่างฝ่ายงานที่น้อยมาก ผลกระทบจากการกระทำดังกล่าว ทำให้เกิดปัญหาข้อมูลไม่สอดคล้องหรือไม่ตรงกัน ข้อมูลเดียวกันแต่มีความแตกต่างกัน ทั้งชื่อข้อมูล โครงสร้าง และมีความหมายข้อมูลที่แตกต่างกัน ทำให้เกิดความสับสนในการนำข้อมูลนั้นไปใช้งาน งานวิจัยนี้ได้นำแนวคิดทะเลสาบข้อมูลมาประยุกต์ใช้กับการแก้ไขปัญหาดาต้าไซโล ขอบเขตของงานวิจัยจะให้ความสนใจเฉพาะดาต้าไซโลแบบมีโครงสร้าง วัตถุประสงค์ของงานวิจัยนี้คือ ออกแบบสถาปัตยกรรมทะเลสาบข้อมูล เฟรมเวิร์คการทำงานภายใน การใช้เทคโนโลยี Hive และ Spark ในการบูรณาการข้อมูลภายในทะเลสาบข้อมูลและเขียนโปรแกรมทดสอบการทำงานด้วยภาษาจาวา ผลการทำงานของโปรแกรมตามเฟรมเวิร์คที่ออกแบบลงรายละเอียดการบูรณาการดาต้าไซโลบนทะเลสาบข้อมูลสามารถลดความแตกต่างของข้อมูลและความไม่สอดคล้องของข้อมูลได้ 100% ในกรณีที่ข้อมูลมีความหมายเหมือนกัน และสามารถลดความซ้ำซ้อนของข้อมูลทดสอบได้ 78.6% ตามกรณีทดสอบ 13 กรณี

ABSTRACT

Data siloes are a major data management challenge in both public and business organizations. As a result of the organizational structure of work functions into numerous departments, each different department has distinct responsibilities and tends to create dependent software, applications and data systems to locally support its individual needs. Similar data are always stored in multiple silos or databases, and normally their data schema likes names and meanings are different from each other. As a result, users are perplexed on how to use those data coming from different silos of software applications. This research applies the data lake concept to solve the data silo problem. The scope of the research focuses on structured data silos. The objective of this research is to design a data lake architecture and its internal working framework by using Hive and Spark technologies to integrate data within a data lake and write functional testing programs in Java Spark. According to the result of testing based on a detailed developed framework, integrating data silos on data lakes can reduce data heterogeneity and data inconsistencies by 100%, and it was able to reduce the redundancy of the test data by 78.6% from the total of 13 separate data cases.

คำสำคัญ: ดาต้าไซโล ทะเลสาบข้อมูล การบูรณาการข้อมูล อาปาเช่ ไฮฟ์
Keywords: Data Silos, Data Lake, Data Integration, Apache Hive

INTRODUCTION

Many data silos result in redundant data and inconsistent data in each silo, which has a negative influence on the company. Although they are the same data, the names, structures, and meanings of the data are frequently different, so this is called data heterogeneity. As a result, there are inconsistencies or inaccuracies in the data, creating confusion about how to use it.

Data silos are a problem that affects not only structured data, but also semi-structured and unstructured data, as well as text, photos, documents, e-mail, and other media data. The data lake concept was established in 2014 to solve the data silo problem, and LaPlante & Sharma (LaPlante and Sharma, 2016) claimed that the data lake is appropriate for it. This is because it functions similarly to a data warehouse system (Fang, 2015), but it is far less expensive, and the speed for which data is imported into the central is also faster. Due to the fact that data lakes are a new notion. Using a data lake has a variety of limits, including data import. The data that has been imported is raw data, which has not been edited or transformed. A data lake is a place where you can store structured, semi-structured, and unstructured data. Moreover, the challenge of developing applications to connect to a data lake and integrating data silos within a data lake.

This research applied the data lake concept to solve the data silo problem. The scope of the research focuses on structured data silos. The objective of this research is to design a data lake architecture (Giebler et al, 2021) and its internal working framework by using Hive and Spark technologies to integrate data within a data lake and write functional testing programs in Java Spark.

The following are the reasons why Hive and Spark were chosen for this project. Using SQL, the Apache Hive TM data warehouse software makes it easier to read, write, and manage big datasets stored in distributed storage. Data that has previously been stored can be projected with structure. Users can connect to Hive using a command line tool and a JDBC driver. Apache Spark is a data processing platform that can handle extremely large datasets quickly.

It's not new to use a data lake to solve a data silo problem. Traditional methods involved creating an integrated data silo outside of the data lake and then storing the integrated data within the lake. The integration silo process took place in the data lake for this study, and the results were stored in a master data table within the lake.

The remainder of the paper is organized as follows: The ways to solve the data silo problem and the traditional data lake notion are introduced in Section II. In part III, we suggest a data lake architecture and mechanisms for data silo integration in this architecture, and in section IV, we explain how the experimental results from the running program have influenced the design of the architecture. Section V summarizes the findings and recommendations derived from the integration of a structured data silo based on a data lake and using Hive and Java Spark.

1. The current data silo solution

The data lake concept (Dixon, 2019) was designed to store and analyze large amounts of data. The data lake had a centralized data component. Its architecture was flat (Walker, 2015), with no deep or hierarchical relationships, and it was defined as a new concept that had just been born. Data lakes, according to numerous academics, could be used to solve data silos in a variety of formats, including structured, semi-structured, and unstructured silos.

The concept of using data lakes to break down data silos is new. The Enterprise Data Lake: Greater Integration and Deeper Analytics was the title of this paper in 2014 (Stein and Morrison, 2014). According to LaPlante and Sharma (LaPlante and Sharma, 2016), the data lake was excellent for resolving data silo issues. Because its functions were similar to a data warehouse system (Fang, 2015), but it is also significantly less expensive. The speed in which data was imported into the central was also increased. The data stored in a data lake is both raw and object data. A unique identifier is assigned to each data entity in a data lake. A set of metadata is available for users to use. Define schemas for on-demand queries; the data lake is compatible with SQL and NoSQL languages, as well as Online Analytical Processing (OLAP) and Online Transaction Processing (OLTP) (Miloslavskaya and Tolstoy, 2016). Daniel E. O'Leary published a research paper in the IEEE Intelligent Systems journal in 2014 on data integration in data lakes. This research used master data management (MDM) and other procedures such as data governance, meta-data, and the ability to merge data from several silos. Furthermore, Daniel stated that data lakes were similar to data warehouses in terms of functionality, and he presented a detailed comparison chart between them. In the same year, Brian Stein and Alan Morrison (Stein and Morrison, 2014) published a paper in the electronic journal Technology Forecast: Rethinking Integration that supported the idea that data lakes were suited to breaking down data silos. In 2015, Huang Fang (Fang, 2015) referred to the concept of data lake could be used to tackle

data silo issues, comparing it to a data warehouse and claiming that data lake was considerably less expensive. Furthermore, the data lake was easier than data warehouse to import data to central. Bill Inmon (Inmon, 2016) proposed in 2016 that using an integration mapping approach, it would be able to integrate data silos within a data lake. However, this method could create difficulties. Because a data lake contains a massive amount of data, both usable and garbage data. When users needed data, it took them longer to find (swim), grasp, and match it.

In 2018, Pwint and his colleagues (Khine and Wang, 2018) advocated using a data lake to break down data silos. The difficulty of obtaining data from various sources and dealing with data that was always changing (transactional data). Moreover, working with SQL in a data lake poses challenges such as data searching (swamp) and database manipulation (CRUD: Create, Read, Update, Delete).

Jayesh Patel (Patel, 2019) stated in 2019 that the data lake concept could be used to eliminate data silos, and that this approach could be implemented within a data lake. Following that, he outlined five steps for integrating data in a data lake:1) data discovery 2) data extraction3) data export 4) data loading and 5) data processing were the next steps. He described data integration within a data lake as a difficult task.

RESEARCH AND METHODOLOGY

There are four processes involved in the research: Designing a data lake architecture is the first step. The second phase is to create a mechanism for integrating data from the data lake and a class diagram for Apache Spark programming; the third step is to create a program and execute it according to the designs; and the fourth step is to record the experiment's results. The test data consists of 13 silos, which divide the data lake integration into 13 cases.

1. Data Lake Integration Model and Workflow in Six Steps

On the data lake, we designed a data integration framework for structured data. There are six steps in this structure, all of which are done within the data lake.

Ingestion is the first step: The data engineer will import the data from the data silo into the data lake at this stage. The information committee that was formed approves the data that is imported into this data lake. The data that has been loaded is stored in the staging table.

Step 2: Make an agreement: The list of data loaded from each silo in the staging table will be checked by an information committee. If an issue arises when loading the data, they will notify the data owner. The owner will then modify the data before reloading it into the data lake.

Step 3: Data Integration: This is the process of integrating data in the data lake, and it is divided into two parts:

3.1 Preparation of Data: Data preparation entails defining the Master Table's structure, which is the primary table used to store data from multiple silos, as well as a group of meta-data tables that help to combine data, such as those used to define the meaning of the data, store the consolidation conditions, and store business key data.

3.2 Data Harmonization: This is a data integration method that uses Java Spark and the Levenshtein distance programming language to

verify structured data integration within a data lake based on the conditions specified between the data silo and the data in the master table. Following the complete completion of the data integration process. The data that has been processed will be saved in the master table.

Step 4: Changed and Conflict Management: This is used to handle data when there is a problem with the integration process, such as when data comes from two (or more) sources and the system is unable to determine which is correct. Users will be told which data items are valid by the information committee. Then go back to step 1 and repeat the process.

Step 5: Data Publication: Integrating and publishing data that has been consolidated from different silos (data in a master table) is the publication of the data for other systems to use.

Step 6: Monitor: This step involves double-checking the information to ensure that it is correct by relying on humans to carry out this task.

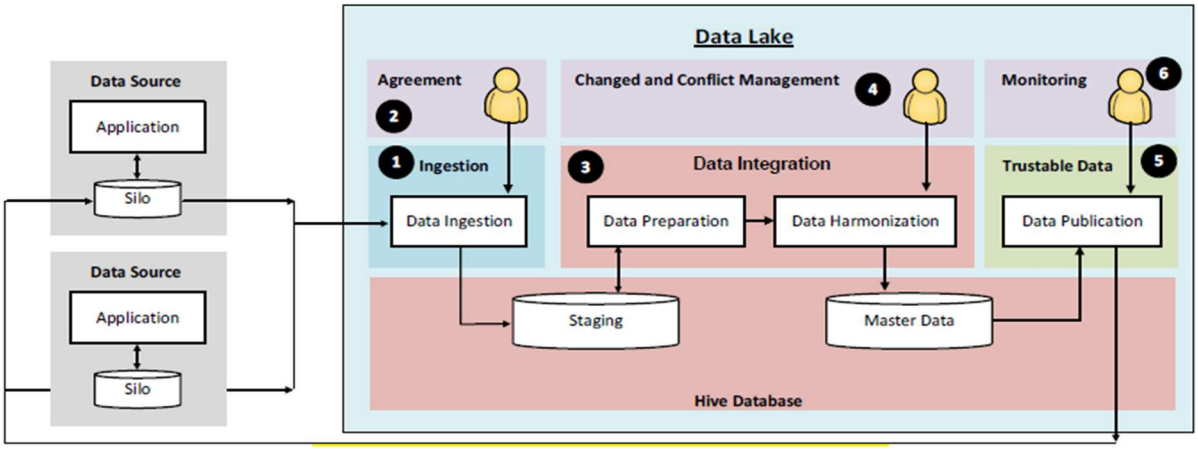

Figure 1. Integration Model on Data Lake

Many silos have been merged into a master table that was constructed and designed in a data lake to integrate all data. The data in the master table is referred to as master data, and it is a reliable source of information. Organizations or associated agencies have reference to that information. The structure of a master table is made up of "key fields," which are business keys used to identify data, and also names, meanings, and data values. We define an integrated case determination as being possible that is integration of data from 13 silos into the master table. In this research framework, there are four alternative examples, totaling 13 cases, as shown in table 1-4.

Table 1 Data silos able to integration data in master data all fields

Case	Conditions
1	Data structure order like master table, data of key fields, number of key fields, meaning and data value between master table and silo are completely match.
2	Data structure in-order like master table, but data of key fields, number of key fields, meaning and data value between master table and silo are completely match.
3	Name of key fields are differing, but data structure order like master table, data of key fields, amount of key fields, meaning and data value between master table and silo are completely match
4	Amount fields of data silos are shorter than master table, but the total number of key fields of data silos that are complete, and data structure order like master table, data of key fields, number of key fields, meaning and data value between master table and silo are completely match.
5	Amount fields of data silos are longer than master table, but the total number of key fields of data silos that are complete, and data structure order like master table, data of key fields, number of key fields, meaning and data value between master table and silo are completely match.

The table 1 shown that the result of five cases in group 1 are able to integrate the silo into the master table.

Table 2 Data silos able to integration data in master data all fields and update some fields in master table

Case	Conditions
6	Data structure order like master table, data of key fields, number of key fields, and meaning between master table and silo are completely match, but data value mismatch. Program will report problem to data committee and then update data value in master table.
7	Data structure order like master table, data of key fields, number of key fields, and meaning between master table and silo are completely match, but data value mismatch and amount fields of data silos are shorter than master table (except number of key fields is completely match). Program will report problem to data committee and then update data value in master table.
8	Data structure order like master table, data of key fields, number of key fields, and meaning between master table and silo are completely match, but data value mismatch and amount fields of data silos are long than master table (except number of key fields is completely match). Program will report problem to data committee and then update data value in master table.

Table 3 Data silos able to integration data by adding new record in master data.

Case	Conditions
9	Data of key fields are differ, but data structure order like master table, amount of key fields, data value, and meaning between master table and silo are completely match. Program will report problem to data committee and then add data of key fields in master table.
10	Data of key fields are differ and amount fields of data silos are shorter than master table, but data structure order like master table, amount of key fields, data value, and meaning between master table and silo are completely match. Program will report problem to data committee and then add new record in master table.
11	Data of key fields are differ and amount fields of data silos are longer than master table, but data structure order like master table, amount of key fields, data value, and meaning between master table and silo are completely match. Program will report problem to data committee and then add new record in master table.

Table 4 Data silos cannot integration data in master data

Case	Conditions
12	Data of key fields and amount fields of key are differ, although some data structure order like master table, some data value, and meaning between master table and silo are match. Program will report problem which cannot integrate data silos into master table.
13	Data of key fields, amount fields of key, and the meaning of data are differ, although some data structure order like master table, and some data value between master table and silo are match. Program will report problem which cannot integrate data silos into master table.

The table 2 shown that the result of three cases in group 2 are able to integrate the silo into the master table. If silo fields are shorter or longer (except number of key fields is completely match), program will report problem to data committee and then update data value in master table. The table 3 shown that the result of three cases in group 3 are able to add some record of the silo into the master table. If data of key fields are differ, but data structure order like master table, amount of key fields, data value, and meaning between master table and silo are completely match.

2. The workflow program of data integration in the data lake

The system integrates data by loading metadata at the first stage to explore each data source and identify which silo is master. Afterward, the structure of the master data silo will be validated and identified with the key and meaning before the next stage.

In the next stage, the system integrates data silos into master data by starting analysis through the data structure of each silo, resulting in a list of columns, key columns and non-key columns. The master data columns will be compared to silo data columns one by one to justify the similarity of string as shown in this figure 2.

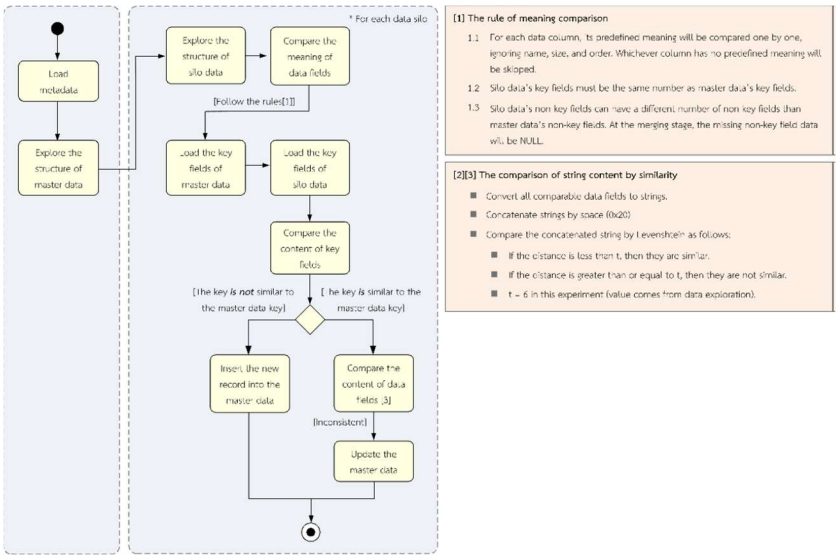

Figure 2. The workflow program of data integration in the data lake using Java Spark and similarity comparison.

3. Coding program to prove the framework according to procedures

The program code to process data on the Data Lake was developed by Apache Spark and Java. Spark is a big data processing technology that works with HDFS or other data sources like Cloud Storage, NoSQL, or RDBMS. In this research, the Spark application will be run over a Hadoop cluster with YARN interactively. The data lake in this research is used as a showcase for data comparison and integration.

The Driver Program and Spark Context work together. The Driver Program is an application code to create a SparkContext object for analyzing and planning the processing of data into a directed acyclic graph, and to proceed and monitor the processing step-by-step according to the planned graph.

The Cluster Manager acts as a dispatcher to delegate tasks to worker nodes. There are 3 worker nodes in this research as shown in Figure 7. The Executor, located within the worker node, is in charge of running tasks and reporting progress to SparkContext, including resource consumption, which is also reported to the Cluster Manager.

In the case of multitasking and parallel processing, the Cluster Manager makes a decision on which worker node deserves to run a task based on its available resources. As a result, the Cluster Manager can maximize resource utilization throughout Hadoop properly. The output for each task will be collected by SparkContext and combined into a final result, which will be returned back to the client application.The parallel data processing in Spark will be stored in RDD format (RDD-Resilient Distributed DataSet), which may be

called a DataSet in Java, and also be called a DataFrame in Python. The RDD is stored

As we mentioned earlier, Spark conducts the parallel data processing in RDD format as a read-only dataset, unable to manipulate the dataset, insert, delete, or update, which is unsupported. To overcome data manipulation over RDD, the researcher uses other techniques to integrate data with Spark, such as the union () function, to combine additional data from silos into the master, as shown in case 4.

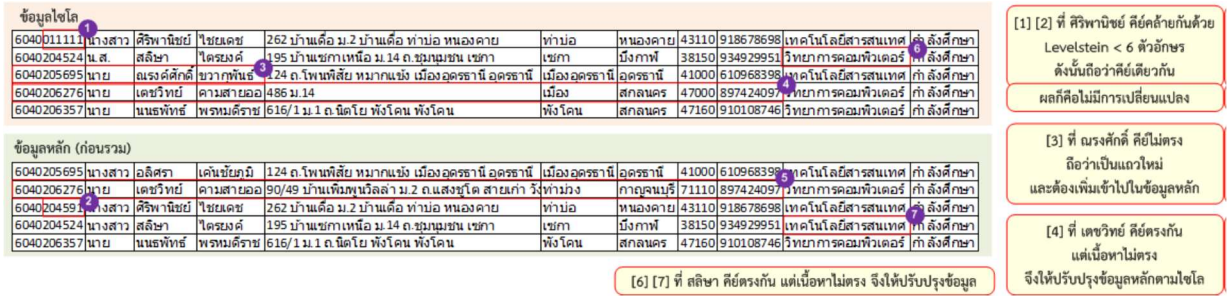

Figure 3. The approach to integrate the case 4 data silo into the master data.

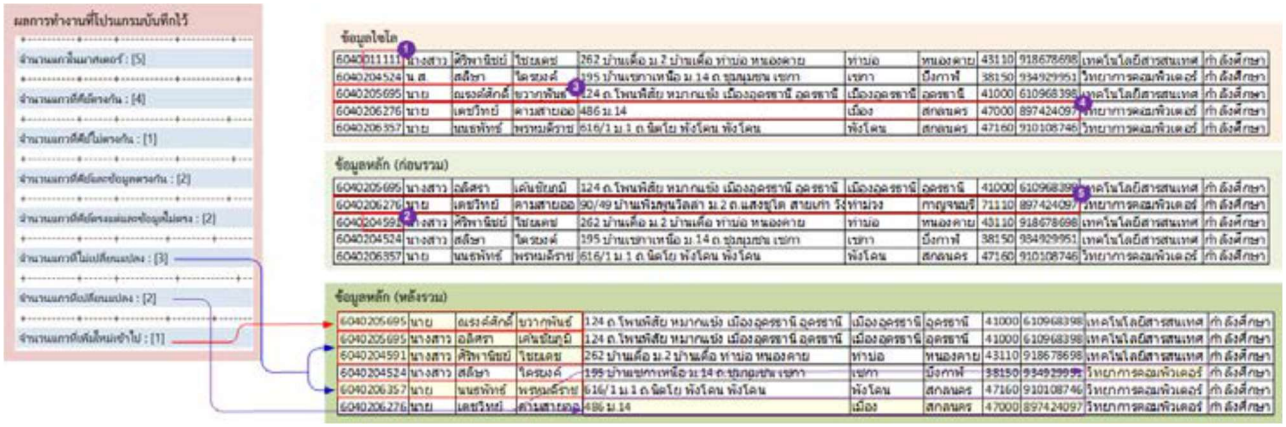

Figure 4. The result of data integration between silo and master.

THE RESULTS OF EXPERMENT

Data integration is the 3rd stage of the proposed framework. A data lake consists of the following data tables: A master data table is used to store combined data. 2) Sematic table, which is used to define the meaning of fields in the master table, especially key fields. 3) A report table is used to log the results of each data integration activity that is used in the system's operation.

In data integration, three essential ideas are used: By comparing the matching data, you can perform 1) meaning mapping, 2) key mapping, and 3) data mapping. As a result of data silos arising from a variety of sources, they have different data structures, names, and meanings. We provide the conditions for integrating the data in Table 1-4. The application is then developed to work in accordance with the workflow that has been established. For program and workflow testing, we use the following three data sets: Data test set 1 was utilized to validate 13 cases, 13 silos, and 10 data records per silo using simulated student data. Data test set 2 is made up of 4,371 records of real data that have been copied in terms of structure

and format. The table structure was copied from 105,292 records of real data in Data Test Set 3 using student data (simulated).

The formula for data differential reduction is as below.

$\nabla d = ((d1-d2)/d1)*100$ (1)

d1 = summation of differences in data structure and data consistency before integration processes inside the data lake.

d2 = summation of differences in data structure and data consistency after integration processes inside the data lake.

∇d = the percentage of data differences that can be reduced.

Table 5 Table summarizing the experimental findings with three data set testing.

Group of data test	Cases	∇d	Note
1	1-11	100% of integration	They meet the test conditions.
	12-13	0 of integration	Data of key fields, amount fields of key, and the meaning of data from data silos are differ in master table
2		100% of integration	They meet the test conditions
3		100% of integration	They meet the test conditions

The formula for data redundancies reduction is as below.

$\nabla s = ((n-m)/n)*100$ (2)

n = number of silos before the integration processes

m = number of silos after the integration processes

∇s = the percentage of data redundancies reduction

Before experiment, we have 14 silos (13 testing silos and 1 master table) and after experiment, we remain 3 silos. Therefore, the percentage of data redundancies reduction is 78.6%

CONCLUSION AND RECOMMENDATION

The detailed performance of the Data Lake Framework means that integrating data silos over a data lake can reduce data heterogeneity and data inconsistencies by 100% in cases where the data is synonymous. Integration Model on Data Lake was able to reduce the redundancy of the first group of test data by 78.6%. During our investigation, we discovered that Spark does not support transactional tables. If transactional = true is set, Spark will be unable to read the data. The solution to the problem in this research is to edit the dataset using the union () and subtract () procedures, as if it were a transaction on that table.

In this research, we used the Levenshtein algorithm to do the approximate matching. To match words by edit distance, must be less than the predefined threshold. To compare whether data from silo is similar to master or not, by allowing the error threshold to be less than the predefined threshold. For example, if the threshold is 6 and the left-side string is "นางสาว" and the right-side string is "น.ส.", the error is 6 because we judge that both string data are the same. The Levenshtein algorithm

causes some errors in comparison. For instance, we define a threshold that must not exceed 6 points.

REFERENCES

Dixon, J. (2019). Retrieved from https://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-and-data-lakes/

Fang, H. (2015). Managing Data Lakes in Big Data Era: What's a data lake and why has it became popular in data management ecosystem. In: The 5th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, Shenyang, China. 820-824.

Giebler, C., Gröger, C., Hoos, E., Eichler, R., Schwarz, H. and Mitschang, B. (2021). The Data Lake Architecture Framework: A Foundation for Building a Comprehensive Data Lake Architecture. In: Conference for Database Systems for Business, Technology and Web (BTW). 351-370.

Inmon, B. (2016). Designing the Data Lake and Avoiding the Garbage Dump. USA: Technics Publications.

Khine, P.P. and Wang, Z.S. (2018). Data lake: a new ideology in big data era. ITM Web of Conferences. 1-11.

LaPlante, A. and Sharma, B. (2016). Architecting Data Lakes Data Management Architectures for Advanced Business Use Cases. USA: O'Reilly.

Miloslavskaya, N. and Tolstoy, A. (2016). Application of Big Data, Fast Data and Data Lake: Concepts to Information Security Issues. In: 4th International Conference on Future Internet of Things and Cloud Workshops, Vienna, Austria. 148-153.

Patel, J. (2019). Overcoming data Silos through big data integration. International Journal of Computer Science and Technology 3(1): 1-6.

Stein, B. and Morrison, A. (2014). The enterprise data lake: Better integration and deeper analytics. Technology Forecast: Rethinking integration Retrieved. 1: 1-9.

Walker, H.A. (2015). Personal Data Lake with Data Gravity Pull. In: Big Data and Cloud Computing (BDCloud), 2015 IEEE Fifth International Conference. 160-167.

❑❑❑❑❑