Journal of Theoretical and Applied Information Technology
30th June 2023. Vol.101. No 12
© 2023 Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PRODUCT RECOMMENDATION SYSTEM DESIGN USING GRAPH DATABASE

FAUZAN RISQULLAH[1], EDWIN TUNGGORO[2], DHIKA NUR AISYAH[3], INDRAJANI SUTEDJA[4]

[1,2,3,4]Information Systems Department. BINUS Undergraduate Program, School of Information Systems,
Bina Nusantara University, Jakarta, Indonesia 11480

E-mail: [1]fauzan.risqullah@binus.ac.id, [2]edwin.tunggoro@binus.ac.id, [3]dhika.aisyah@binus.ac.id,
[4]indrajani@binus.ac.id

ABSTRACT

The growth of data in the world is getting bigger, each person currently produces a lot of data that comes from their daily activities. To accommodate large data, appropriate data storage devices are needed. One of the data storage tools that is currently popular is the graph database. Therefore, this study aims to prove the capabilities of the graph database by creating a product recommendation system using one of graph database tools, TigerGraph. This research began by doing literature research for previously related researches that had been done. This research will use the database development cycle method in the database development process. The results obtained from this study are graph database capable of providing product recommendations that are in accordance with the trends and preferences of each user. The conclusion from this study is that graph database has reliable performance in processing data and it relationships and has a short time to process it so that users can get appropriate recommended product in a short time.

Keywords: *Graph database, Product Recommendation, TigerGraph, Cosine Similarity*

1. INTRODUCTION

Data growth in the world is getting bigger, each person is currently generating a lot of data that comes from their daily activities**.** This is due to the large number of human activities carried out through cyberspace. Before cyberspace began to develop, human-generated data were only related to transactional activities. But currently, in the era of *big data*, almost all activities carried out by humans generate data such as when using a computer, using a smartphone, or using *Internet of Thins (IoT)* related products.

One of the uses of this data growth is the existence of a product recommendation system. The product recommendation system can provide product recommendations to users based on the results of processing previously stored user data. In order for the system to provide best recommendation results, a database that can accommodate and process large data quickly is needed.

Graph databases have the ability to process large interrelated data so that they can find hidden patterns and relationships between entities more efficiently than *relational databases***.** In addition, the graph database has an *agile* and flexible structure, the representation of relationships between entities *(nodes)* is explicit. Graph database produces *real-time* and the speed depends on the number of relationships between entities [1][2].

When doing a complex query, that is a query that require multiple joined tables in SQL or by using a path with various links to different type of nodes that can be recursive in graph database. The graph database can outperform the relational database up to 146 times. Which indicate that relational database is far less suitable for a system that handles and analyze complex data and big data compared to graph database [3].

Based on the advantages provided by the graph database and some limitations with relational database when handling large of data with its relationships. A graph database system is best suited for analytical application that heavily analyze the relationships between the data.

Therefore, this study will help to see how the application of graph database for an analytical system, which is a product recommendation system. And hypothetically, this graph product recommendation system will perform better compared to relational database-based product recommendation system.

The motivation of this study is to make research on the development and implementation of graph database system**.** The method used is literature study, data collection, and implementing the database development life cycle. The purpose of this study is to find out whether the implementation of a product recommendation system using graph database, especially using TigerGraph is viable or not. Because at this time there are still not a lot of research that uses TigerGraph when implementing a *graph-based* product recommendation system.

The contribution of this research to the academic community is to provide concrete evidence of the capabilities of graph database and TigerGraph. Contribution to business is to provide a reliable product recommendation system, which has high speed and accuracy. That way, it will be able to increase the level of user satisfaction.

This paper begins with discussing about product recommendation system and graph database as shown in the Introduction part. Then followed by Literature Review which discusses the theory used for this paper and some limitation of existing tools. Followed with Research Method which discusses the methodology on how the system development using graph database will be conducted. And for this Research Method section, this study will use the database design and development method by Coronel et al (2016). This paper also provides the limitation of the paper as seen on the Limitation part. And lastly this paper also provides the conclusion of this study.

2. LITERATURE REVIEW

2.1. Product Recommendation System

Product recommendation system is a system that recommends products that meet user needs using statistical techniques and knowledge discovery technology and focuses on cross-selling and increasing sales. This refers to a technology that recommends previously purchased products or products that are in the same category based on search records, or products with similar patterns and conditions [4].

Cho et al. (2020) propose a hypothesis that there are five (5) variables used to measure the characteristics and customer satisfaction of the product recommendation system [4]. Wu et al. (2013) prose a hypothesis that two independent variables that correspond to user satisfactory is user's initial product awareness and recommender strategy such as collaborative-filtering or content-based [22]. While those two studies propose their own variables used to measure customer satisfaction, but Kinkar et al. (2021) found that there are several challenges for product recommendation system. And one of them is about the privacy concern by the customers, since in most cases customers must provide their personal information for the recommendation system [6]. Thus, when measuring customer satisfaction, everything regarding data privacy should be taken into account.

Product recommendation systems can effectively help users filter irrelevant information, find information that is of interest to users, and improve information utilization by analyzing and digging up relevant information about users and products [5]. With many benefits, product recommendation also has several challenges, such as [6]:

a. Synonymous

Product names will be a challenge for product recommendation systems. Where there are products that have two or more different meanings, such as apple which can have two meanings, namely apples or Apple brand products. The product recommendation system will have difficulty determining the intended user.

b. Scalability

The scalability of the algorithms that access the database is a problem in product recommendation systems. The algorithm used is overloaded due to the large amount of product and customer data that must be processed. This becomes a problem in data sets and also reduces efficiency.

c. Privacy

In order for the product recommendation system to work optimally, users need to provide their personal information.

However, it causes a lot of privacy and security issues which makes consumers feel reluctant to enter their personal data into the product recommendation system. Therefore, a product recommendation system must be able to ensure that user data that is managed is safe in order to maintain user trust.

d. Data availability issues

In some cases, there are users who don't want to review the product they bought. This can cause data availability issues. The product recommendation system will have difficulty finding the same user preferences.

e. Latency

Latency issues will occur when adding many new products. The new product will be difficult to recommend to users because there are no reviews or transactions for the new product. So that this problem does not occur, categories can be added such as new product categories.

At this time, the problem regarding scalability in the product recommendation system above is no longer relevant because it can be resolved with the existence of a graph database that can process large amounts of data in real time [11].

2.2. Database Management System

Database Management System (DBMS) is a software used to organize and process data. The DBMS makes it easier for organizations to collect data, manage data efficiently, and provide access data for application programs [7]. The purpose of this DBMS is to save data storage space, data security, eliminate data duplication and inconsistency and handle large or large amounts of data.

2.2.1. Relational database management system (RDBMS)

The data model of RDBMS is in Table/Relation form, with a primary key to uniquely identify records and a foreign key to connect tables so that they can join tables. This RDBMS is also a database system whose main entity consists of tables that have relationships from one table to another. RDBMS has several advantages such as abstraction, multiuser access, automatic optimization for searching, ACID which allows using query language in transaction support [8].

2.2.2. Key value database

Key-Value database is a non-relational database type that uses a simple method of key-value pairs in the form of tables. These key-values can store a lot of data without using difficult queries during the storage process. Some "Key" is in a unique key-value, while "Value" can be JSON, BLOB, String etc. For users in cloud computing, distributed systems and storing unstructured data where multiple systems may change over time, NoSQL data-based systems are more efficient and allow faster scaling [9].

2.2.3. Graph database

Graph database is a data store in the form of nodes, edges, and relationships so as to provide index-free adjacency [10]. Based on the results of research conducted by Gartner, the trend of implementing Graph Technology is predicted to increase by 80 percent by 2025 compared to the total utilization of Graph Technology, which until 2021 is only around 10 percent. Graph databases can best express relationships among individual data elements by creating nodes for each data element and connecting nodes via edges to establish relationships between them.

In the graph database it is also possible to query data in real time [11]. Graph databases can reduce storage costs for storing large-scale graphs, although it may require more storage when the graph size is small when compared to relational database with the same amount of data stored [12]. Although it has many advantages, graph database is not always superior in some aspects. Therefore, implementing a graph database also needs to consider its purpose and context [13].

2.3. Cosine Similarity

Cosine similarity is a measure of popular text similarity [18]. Cosine similarity serves to measure or calculate the similarity between two documents based on the terms of the subject matter [19]. Cosine similarity is not affected by the length of the document, but is affected by the term value in each document. In addition, cosine similarity has an accuracy rate of 98% so that the error rate is quite low [20].

3. RESEARCH METHOD

In general, the methods used to design and develop graph databases are almost the same as other databases, the difference is when they are in the logical model stage [15]. If normally in other databases the logical model stage is normalized, in the graph database a graph database schema (GDBS) is created [16].

Therefore, this study will use the database design and development method created by Coronel et al. In his book, database design and development methods start from database initial study, database design, implementation and loading, testing and evaluation, operation, and maintenance and evolution [17]. The following are the stages of database design and development that were carried out:

a. Database initial study

In the database initial study stage, the purpose of developing the database to be created will be determined. The vision and mission of creating the database are also determined at this stage. In addition, at this stage it also determines a broad picture such as how the data collection, database design and data formatting will be carried later.

Next, the requirement gathering process will be carried out. The purpose of doing a requirement gathering is to collect and analyze the data that will be needed by the database.

b. Database design

In the first stage of Database design, a conceptual design will be made, which will produce a class diagram [16]. Next, the second stage of database design will be carried out, namely the selection of tools of the graph database that are in accordance with the requirements that have been carried out in the previous stage. After that, it will proceed to the third stage, namely logical design. In logical design, graph database schema (GDBS) will be made [15].

In the graph database schema, there will be vertices (nodes) and edges. Thus, all the entities will be described by vertices and the relationships between these entities will be described by edges. In the last stage of database design, a physical design is made from the graph database schema that was previously created. Also, in this stage the graph database schema will be created so later the schema can be created in the chosen graph database tool. And the next process is to determine who the users of the database are, what the user roles are, and how to control the user's access to the database.

c. Implementation and loading

At the implementation and loading stage, the graph database tool will be installed. After the database is installed, a database will be created based on the database design that has been made. After the database is ready, data will be transferred from the old database to the new one (data loading). In addition, database integration is also carried out with other applications if any.

d. Testing and evaluation

After the database has been successfully implemented, it will be tested first before the database is deployed. The purpose of this testing is to find anomalies in the database that need to be fixed. System Integration Testing will be carried out to ensure that the end-to-end system integration process runs well.

Also, performance testing is carried out to test the reliability of the system in executing use case queries. After testing is complete, an evaluation will be carried out to determine whether the database created is feasible to deploy or not.

e. Operation

After confirming that at the testing stage the database can run according to the initial requirements, it will proceed to the next stage, namely operation. At the operation stage, the database will be deployed to the production environment. The old database will be replaced with the new database. The old database should not be completely shut down first to anticipate if an error occurs in the new database.

f. Maintenance and evaluation

After being deployed, the database also needs to be maintained. The goal is to make sure the database is running properly. Apart from maintenance, the database also needs to be evaluated periodically to find out what things need to be improved in the future.

The following are the steps used in this study:

a. Exporting the master data tables for Customer, Product, and Transactions table into delimited files.

b. Create a graph database schema that will be used in developing this graph analytic based product recommendation system
c. Prepare queries that will be used to provide product recommendations.
d. Testing the queries that have been made using the data that has been exported.

4. RESULT AND DISCUSSION

Table 1: Tigergraph Platform Specification

CPU	2 Core
RAM	8 GB
STORAGE	50 GB
OS	Linux
SOFTWARE	TigerGraph Cloud 3.7.0

The graph database tool used to design the product recommendation system in this research is TigerGraph Cloud version 3.7.0. The platform specifications used are 2 core CPU, 8GB Memory (RAM), and 50 GB of storage. These system specifications are default specifications provided by TigerGraph with a free license. This graph database system will be used as the analytical system for product recommendation purposes.

Table 2: MariaDB Platform Specification

CPU	4 Core
RAM	16 GB
STORAGE	500 GB
OS	Windows
SOFTWARE	MariaDB 10.4.25

The RDBMS tool that is used to store master data and transactional data is MariaDB version 10.4.25. The platform specifications used for this RDBMS are a 4 core CPU, 16 GB Memory (RAM), and 500GB of storage. This RDBMS system acts as an existing database that is used to store master data and transactional data.

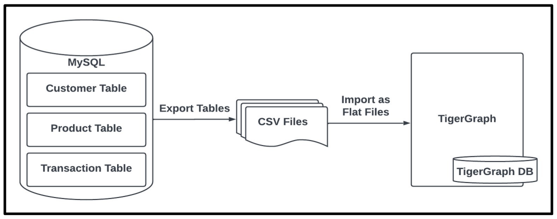


Figure 1: Data Uploading Configuration

The data uploading process begins by exporting the master data tables for Customer and Product as well as table Transactions into delimited files. Exported delimited files will be stored in one place before being imported to the TigerGraph system. Actually, the TigerGraph system supports import directly via MariaDB and PostgreSQL database systems.

However, this configuration is carried out by taking into account the impact in terms of performance given to the existing database system if a direct connection is made. Since, connecting directly to complex database applications will greatly increase system overhead to create and close database connections frequently [21]. Therefore, in designing this graph analytic based product recommendation system, a data uploading configuration that does not significantly affect the performance of the existing system is chosen. Which is by exporting the master data and the transactional data and later upload it to the graph database platform.

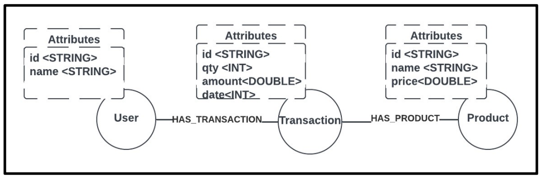


Figure 2: Graph Database Schema Configuration

Figure 2 illustrates the schema of the graph database that will be used in developing this graph analytic based product recommendation system. It can be seen that there are three (3) entities represented as vertices, namely User, Transaction, and Product along with the attributes and data types of each entity. Besides the vertices there are also two (2) edges that connect each vertex, the edges are HAS_TRANSACTION and HAS_PRODUCT. Both edges are of type undirected and have no attributes.

The graph database schema also describes how the product transaction process is carried out by the user in an outline. Let's say there is a user who makes a transaction, then the user node will be connected to the transaction node via the “HAS_TRANSACTION” edge. And in this transaction, there are also products that were purchased, so that the node of the transaction will be connected to the products via the "HAS_PRODUCT" edge.

Table 3: Master Data and Transactional Data Delimited File Details

File	Description	Rows	Columns
CUST.csv	File containing customer data	100	id name
TRANSACTION.csv	File containing transactional data of products purchased by the user	5	id user_id prod_id qty amount date
PRODUCT.csv	File containing product data	200	id name price

Table 3 above explains the data used in this research. This research uses customer, transaction, and product files. There are 100 customer data containing id and name in the CUST.csv file. In the TRANSACTION.csv file, there are 5 transaction data containing id, user_id, prod_id, qty, amount, date. There are 200 product data containing id, name, price in the PRODUCT.csv file.

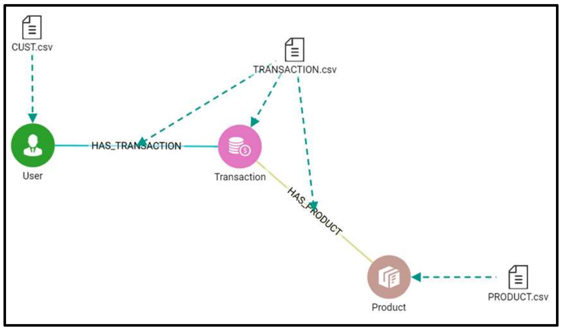


Figure 3: Data Mapping Configuration

Figure 3 explains the relationship between the three data. Users or customers entity relate to transactions entity via transaction entity because to buy products, user needs to create a transaction. Products are also related to transactions because in order for products to be sold, there needs to be transactions for products.

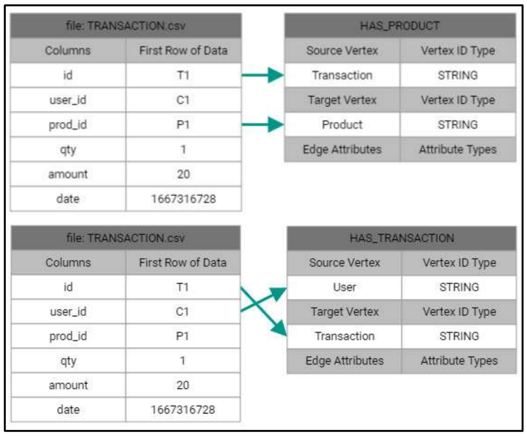
Table 4: Data Mapping from Files to Vertex

Data Mapping	
From (Files)	**To (Vertex)**
CUST.csv	**User**
id	*id (PRIMARY)*
name	*name*
TRANSACTION.csv	**Transaction**
id	*id (PRIMARY)*
qty	*qty*
amount	*amount*
date	*date*
PRODUCT.csv	**Product**
id	*id (PRIMARY)*
name	*name*
price	*price*

Table 4 above explains the relationship between the columns in the delimited file (.csv) with the attributes on the vertices in the graph database schema. And each of these vertices is connected by an edge so that the relationship between data can be defined by matching the primary id that exists on each vertex.

As can be seen in the table in one of the delimited files, namely "TRANSACTION.csv", only four (4) columns are associated with attributes at the vertex, even though there are six (6) columns in the delimited file. The other two columns in the file contain foreign key data which in this graph database becomes the reference data for creating relationships between vertices through edges. Therefore, the two columns will be connected not to the vertex but to the edge.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



file: TRANSACTION.csv	
Columns	First Row of Data
id	T1
user_id	C1
prod_id	P1
qty	1
amount	20
date	1667316728

HAS_PRODUCT	
Source Vertex	Vertex ID Type
Transaction	STRING
Target Vertex	Vertex ID Type
Product	STRING
Edge Attributes	Attribute Types

file: TRANSACTION.csv	
Columns	First Row of Data
id	T1
user_id	C1
prod_id	P1
qty	1
amount	20
date	1667316728

HAS_TRANSACTION	
Source Vertex	Vertex ID Type
User	STRING
Target Vertex	Vertex ID Type
Transaction	STRING
Edge Attributes	Attribute Types

Figure 4: Data Mapping for Edges

Figure 4 above explains how the data mapping process is carried out for the "HAS_PRODUCT" edge which connects the Transaction vertex with the Product vertex and "HAS_TRANSACTION" edge which connects the Transaction vertex with the User vertex. It can be seen on the "HAS_PRODUCT" edge, from the delimited file "TRANSACTION.csv" the product_id column is connected to the target vertex on the edge. As well as the id of the transaction associated with the source vertex. So that a relationship will be created between the transaction vertex and the product vertex whose id is connected to the "HAS_PRODUCT" edge.

The same thing is also done for the "HAS_TRANSACTION" edge where the user_id column from the delimited file "TRANSACTION.csv" is linked to the source vertex and the id of the transaction is linked to the target vertex. Therefore, a relationship will be created where transactions will be connected with users who make transactions based on the id of that user.

Query for product recommendation starts with the user inputting the product code. Then the query will check all transactions from users who have purchased the product entered. That way the query will get a collection of users who have purchased the inputted product and from here the query will look for other products that have been purchased by these users for example Product X, Product Y, and Product Z. And those products will be referred as the related products.

After that, the similarity of the inputted product and the related products is calculated using cosine similarity. Calculation of the similarity value is based on the total transaction spent of the inputted product and the total spent of related products. If the nominal spending between the two products is close, the similarity value will be higher.

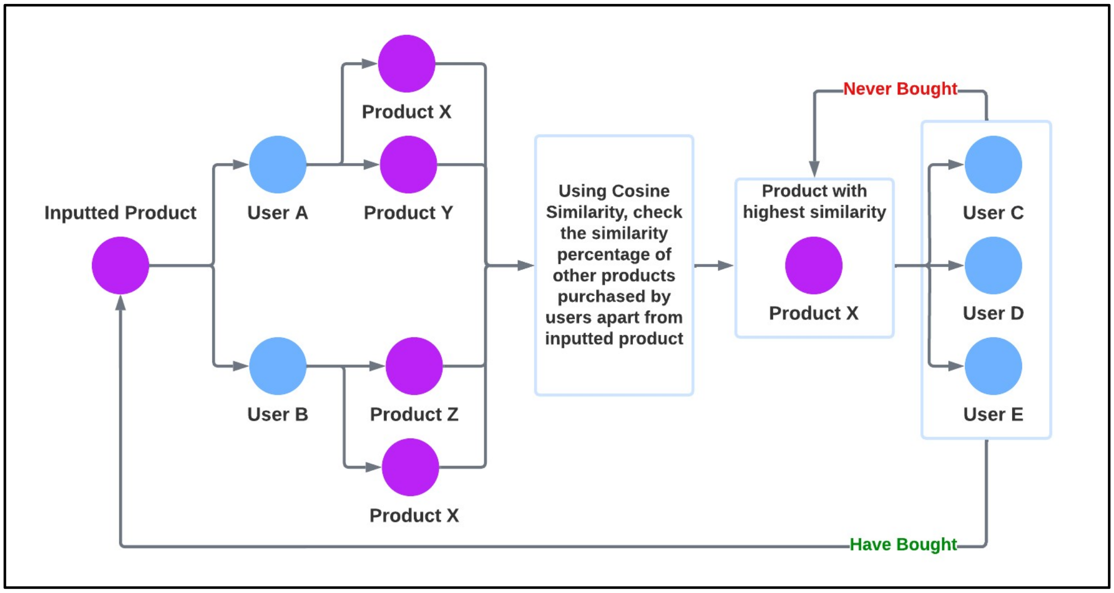

Inputted Product
User A
User B
Product X
Product Y
Product Z
Product X
Using Cosine Similarity, check the similarity percentage of other products purchased by users apart from inputted product
Never Bought
Product with highest similarity
Product X
User C
User D
User E
Have Bought

Figure 5: High Level Logic for The Product Recommendation Query

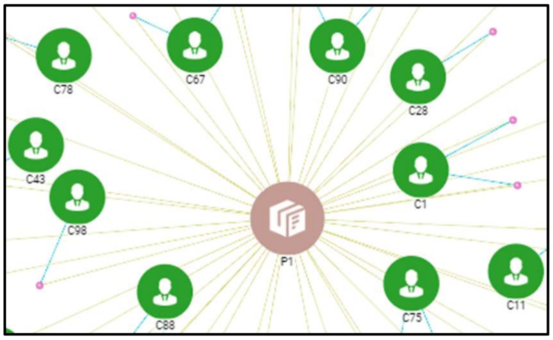
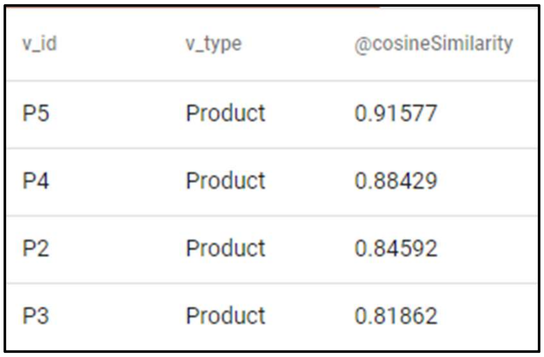


Figure 6: Query Result for Users That Have Bought the Inputted Product

Product recommendation query will ask for input in the form of a product code, then it will look for customers who have already purchased the product. As seen in the example in figure 5, the query receives inputted product code which is the 'P1' and the query will search for all customers who have purchased the product.

The query will also flag the inputted product vertex to indicate that the inputted product has been traversed by the query. Therefore, the product recommendation system will not recommend the inputted product to be recommended to the user. And also, the query will attach a variable that contains the total spent amount for that product for each user at that user's vertex.



v_id	v_type	@cosineSimilarity
P5	Product	0.91577
P4	Product	0.88429
P2	Product	0.84592
P3	Product	0.81862

Figure 7: Ranking of Similarity of Related Products Purchased by the User

The query will look for related products purchased by the customers that found earlier. The query eliminates customers who only buy the inputted product, so the result to this block of query is a unique product code.

The output will also provide a level of similarity and this number will also be used as a reference for ranking which products are most relevant to recommend.

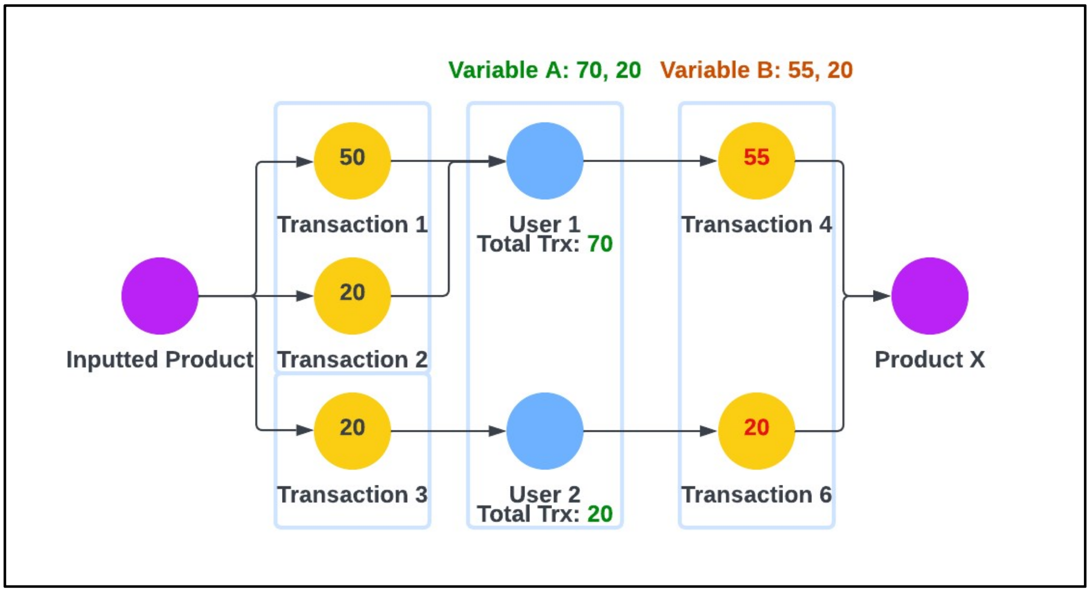


Figure 8: Determining the Cosine Similarity Calculation Variables

In this paper, the product with the highest similarity rating will be selected, and will make the product a recommended product. In figure 6 the recommended product is 'P5', because this product has a similarity level of 91.57%.

The recommendation value for this product recommendation system is calculated by using the cosine similarity method. In this calculation, two variables are used, variables A and B. Variable A is the total transaction spent for each user to purchase the inputted product. Meanwhile, variable B is the amount spent of transactions from users who have purchased the inputted product for each product other than the inputted product.

$$\square\square\square(\square) = \frac{\sum_{\square=\square}^{\square} \square_{\square}\square_{\square}}{\sqrt{\sum_{\square=\square}^{\square} \square_{\square}^{\square}}\sqrt{\sum_{\square=\square}^{\square} \square_{\square}^{\square}}}$$

(1)

In the example in Figure 7, to get the cosine value of Product X is to calculate the sum of the multiplication of variable A with variable B. Which is then the query will calculate the root of the sum of variable A raised to the power of two and the root of the sum of variable B raised to the power of two and the result of the two roots are then multiplied (1).

So that the cosine value is obtained by dividing the sum of the multiplication of the two variables and the result of multiplying the roots of the sum of the two variables raised to the power of two. The following is a product X cosine similarity calculation example by referring to the Figure 7.

Table 5: Simulation of Calculating the Cosine Similarity Value for Product X

	A	B	A x B	A²	B²
	70	55	3850	4900	3025
	20	20	400	400	400
Total			4250	5300	3425

Based on the simulation in Table 4, to calculate the cosine similarity value of Product X is to divide the number 4250 with the result of $\sqrt{5300}$ multiplied by $\sqrt{3425}$. So that the number 0.9975 will be obtained, which means that Product X has a similarity value of 99.75% with the input product. This similarity value is very high so that the system can recommend Product X to be purchased by users who have only purchased the inputted product or have purchased the inputted product but have never purchased Product X.

5. LIMITATION

The limitations in this paper are related to the data uploading configuration. The configuration of the data uploading recommendation system in this paper prioritizes ease of implementation. It is recommended for the data uploading process to be automated by utilizing various existing tools.

The process of exporting master data tables and transactional data can be automated using a scheduler and data will be ingested by TigerGraph using streaming tools such as Apache Kafka. As much as possible do not create a direct connection between TigerGraph and the existing database because it can affect the overall performance of the existing system. It probably not going to be noticed with such small data, but if the data in the existing database system is large, then it will certainly affect the performance to the existing system during the data loading process if such direct connection is made.

6. CONCLUSION

As the technology of product recommendation system evolves, the implementation of such technology also increases. Many businesses sector harness the benefit given by the product recommendation system. We can see this application of product recommendation in our daily life, for example in social media, e-commerce, media and entertainment platform and many more.

With many methodologies and tools used to develop a product recommendation system, this study proposes a solution on how to develop a product recommendation system. By utilizing the advantages provided by the graph database system, which one of them is the superiority of analyzing data with complex relationship structure compared to the relational databases. A graph analytic-based product recommendation system is definitely one of many ways to utilize the advantages given by the graph database. Since a product recommendation system relies heavily on the relationship between its data.

Based on the results of the product recommendation system that has been built, it can be concluded that the graph database is effective in analyzing data that have complex relationships. By using the cosine similarity to analyze the similarity value between products combined with the ability possessed by TigerGraph to store a vertex-attached variable. Which makes computing aggregation values becomes much easier and not to mention the elimination of the join process on the query level that is computationally expensive since the relationship between data is predefined. With these capabilities and benefits, such complex data analysis that is not limited to product recommendation systems can also be made using graph database.

REFERENCES:

[1] Lysenko, A., Roznovăţ, IA, Saqi, M., Mazein, A., Rawlings, CJ, & Auffray, C. (2016). Representing and querying disease networks using graph databases. *BioData mining*, *9*(1), 1-19.

[2] Dancuk, M. (2022, July 11). *What is a graph database? {definition, use cases & benefits}*. Knowledge Base by phoenixNAP. Retrieved October 12, 2022, from https://phoenixnap.com/kb/graph-database

[3] Do, Thi-Thu-Trang, et al. "Query-based Performance Comparison of Graph Database and Relational Database." *Proceedings of the 11th International Symposium on Information and Communication Technology*. 2022.

[4] CHO, BD, POTLURI, RM, & YOUN, MK (2020). A Study on the effect of product recommendation system on customer satisfaction: focused on the online shopping mall. *The Journal of Industrial Distribution & Business*, *11*(2), 17-23.

[5] Dabhade, MMG, & Chopde, NR (2020). A Result Review Analysis of Product Recommendation System in Domain Sensitive Manner.

[6] Kinkar, K., Zore, A., & Kulkarni, PV Product Recommendation System: A Systematic.

[7] Meiryani, USA (2019). Database Management System. International Journal of Scientific & Technology Research, 8.

[8] Raj, P., & Deka, GC (2018). *A Deep Dive into NoSQL Databases: The Use Cases and Applications*. Academic Press.

[9] Hasan, WU Exploring NoSQL (Key-Value) DB vs Relational Database Management System.

[10] Paul, S., Mitra, A., & Koner, C. (2019, March). A Review on Graph Database and its representation. In *2019 International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC)* (pp. 1-5). IEEE.

[11] Kundu, G., Mukherjee, N., & Mondal, S. (2020, May). Building a graph database for storing heterogeneous healthcare data. In *International Conference on Information and Communication Technology for Intelligent Systems* (pp. 193-201). Springers, Singapore.

[12] Chen, Y. (2016). *Comparison of Graph Databases and Relational Databases When Handling Large-Scale Social Data* (Doctoral dissertation, University of Saskatchewan).

[13] Timón-Reina, S., Rincón, M., & Martínez-Tomás, R. (2021). An overview of graph databases and their applications in the biomedical domain. *Databases*, *2021*.

[14] Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of business research*, *104*, 333-339.

[15] Roy-Hubara, N., & Sturm, A. (2020). Design methods for the new database era: a systematic literature review. Software and Systems Modeling, 19(2), 297-312.

[16] Roy-Hubara, N., Rokach, L., Shapira, B., & Shoval, P. (2017). Modeling graph database schema. IT Professional, 19(6), 34-43.

[17] Coronel, C., & Morris, S. (2016). *Database systems: design, implementation, & management*. Cengage Learning.

[18] Sohangir, S., & Wang, D. (2017). Improved sqrt-cosine similarity measurement. Journal of Big Data, 4(1), 1-13.

[19] Gunawan, D., Sembiring, C. A., & Budiman, M. A. (2018, March). The implementation of cosine similarity to calculate text relevance between two documents. In *Journal of physics: conference series* (Vol. 978, No. 1, p. 012120). IOP Publishing.

[20] Ristanti, P. Y., Wibawa, A. P., & Pujianto, U. (2019, October). Cosine similarity for title and abstract of economic journal classification. In *2019 5th International*

Conference on Science in Information Technology (ICSITech) (pp. 123-127). IEEE.

[21] F. Liu, A Method of Design and Optimization of Database Connection Pool, presented at the 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics, 2012.

[22] Wu, Ling-Ling, Yuh-Jzer Joung, and Jonglin Lee. "Recommendation systems and consumer satisfaction online: moderating effects of consumer product awareness." *2013 46th Hawaii International Conference on System Sciences*. IEEE, 2013.