2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA) | 978-1-6654-4276-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICPECA53709.2022.9719151

An Approach to Converting Relational Database to Graph Database: from MySQL to Neo4j

Hui Feng
College of Computer Science and Technology
Chongqing University of Posts and Telecommunications
Chongqing, China 400065
s200231125@stu.cqupt.edu.cn

Meigen Huang*
College of Computer Science and Technology
Chongqing University of Posts and Telecommunications
Chongqing, China 400065
huangmg@cqupt.edu.cn

***Abstract*—At present, there are few research methods that can convert any relational database into a graph database, and most of them are based on a specific field data set to build a relational database, and then perform simple conversion through the characteristics of the data set.Aiming at this problem, a universal conversion method is proposed. Firstly, converted the most basic component tables name, records, and fields in the relational database into labels, nodes, and corresponding attributes of the nodes under the graph database; secondly, used the intermediate connection table method to convert the foreign keys in the relational database into the relationship of a graph database between the nodes; then some constraint issues in relational databases, such as multiple primary key issues, indexes, and no default values, were optimized to form a final graph database model that met expectations; finally, Realized the effective migration of large quantities of data in the relational database to the constructed graph database model. In the experiment, the above method was used to successfully convert a relational database to a graph database, and the database construction, data import, SQL query and Cypher language query were performed for the database before and after the conversion, and through the analysis and comparison of data integrity, time cost, result validity,which shows that the integrity and operability of the database before and after conversion are consistent, and the data processing efficiency of the database is much higher than that of the relational database, which verifies that the method in this paper is feasible.**

Keywords—Relational database, Graph database, Database conversion, Data migration, Neo4j, MySQL

I. INTRODUCTION

Graph database has become more and more popular in recent years, especially its advantages in high-performance processing capabilities and strong flexibility and agility embodied in intensive data [1]. The core of the graph database is a graph model, and a graph is composed of a collection of edges and vertices. Its appearance is similar to a nautical chart. Each vertex represents a location, and each edge represents a path between two locations. In the graph data model, its structure is node-relationship-node, these nodes may correspond to different labels, and each corresponding node also has some attributes of its own, in which the edge between each two nodes corresponds to the relationship between two nodes, the relationship also has own attributes.

The graph database is the structured storage of the connections and changes between entities and relationships, so online analysis and processing can be performed efficiently. The most frequent operation in the graph database is online analysis and processing of relational data, and as the scale of relational data grows, the processing and analysis efficiency of the graph database will gradually increase. When the traditional relational database analyzes and processes this kind of data, with the expansion of the relational data set, the efficiency of the traditional relational database processing these related data is significantly reduced [2]. Compared with relational databases and NoSQL databases, graph databases have obvious advantages in relational query and analysis and mining of data [3] [4] [5]. Such related data sets mostly appear in social networks [6] [7], recommendation systems [8], network and data center management [9], and logistics management [10]. When traditional relational databases and graph databases process this type of data separately, the analysis and processing efficiency of graph databases is several orders of magnitude higher than that of relational databases. At this time, it is necessary to deal with such a problem, that is, how to convert the tables in the data set under the relational model and the connections between the tables into nodes and corresponding relationships in the graph database, and to migrate the related data.

However, from the current research on database conversion and data migration [11], only Neo4j officially provides some data import and data migration methods. For example, the data import methods include Load CSV files and dump methods based on Neo4j databases. The migration is also completed by data migration with the ETL tool provided by Neo4j. However, there are few related researches on the conversion and migration of arbitrary data sets in two kinds of databases. Most of them are based on the conversion and migration of data sets in some specific scenarios. For example, Unal [12] studied Hierarchical legal document system, and De Virgilio [13] studied the direct mapping of relational data into RDF stored in the graph database to realize database conversion and so on.

In general, there are three main problems in the conversion of traditional relational databases to graph databases: one is to sort out the relationship between the metadata schema in the relational database before the conversion, and to prepare for the conversion; The second is to solve the completeness problem of mapping from the table in the relational database and the foreign key relationship between the tables to the graph database and deal with some data constraints in the table during the conversion process; the third is how to solve the problem of data migration after the database conversion is completed. Therefore, this article focuses on proposing a custom method to solve these three problems.

(1) Mapping of metadata

Firstly, use the tool MySQL Workbench to draw the E-R diagram under the relational database, and sort out the connections between different tables. After that, the mapping between metadata is completed through the E-R diagram.

(2) Retaining data constraints and foreign key connections

Special methods are used to process the foreign keys of various tables in the database, and some constraints on the data in the table are retained or deleted.

(3) Mass data migration

After the basic database conversion is completed, a basic graph database model can be obtained. On this model, data migration under specific rules will be carried out. Data migration mainly includes two steps, namely, obtaining the metadata in the table and importing the data into the graph database management system using the graph database API.

II. DATABASE CONVERSION RULES

In traditional relational databases, the most basic element is the data table. Firstly, the table name of each data table is named as the label of the node in the graph data model, indicating that it belongs to the same type of node, and then the table in the table Each record is mapped to a node in the graph data model, and each field in the table is mapped to an attribute on the corresponding node.

A. Mapping rules for nodes and relationships

1) Node Conversion

Firstly, a basic structure of the table can be clearly obtained through the drawn E-R diagram, and the primary key in the diagram is used to name the node, and then all candidate codes are used as the attributes of this node.In addition, for some data tables, there will be both business primary keys and logical primary keys. Delete the logical primary key first, and keep the corresponding business primary key, and then use the business primary key to name the node. At the same time, you can also add an index to the business primary key to improve query efficiency. For some data tables, there are composite primary keys, and the conversion method is the same as above. Finally, for those non-canonical data tables without a primary key, a combination of multiple fields is adopted for conversion and naming. The algorithm process of node conversion is as follows:

Algorithm 1: node conversion algorithm

Input: the table name (table I, table II...) and the field name of the corresponding table (tableI.filed1, tableII.filed1.....)

Output: a set of nodes with labels(A1, A2....) and corresponding node properties(A1.property1, A2.property....)

```
1. i←1,j←1
2.   For table[i]≠∅ do
       For table[i].filed[j]≠∅ do
         A[i][j]← table i and table i. filed j
           j←j+1
           i←i+1
3.   Return A[i][j]
```

Based on conversion algorithm 1, taking a Product table as an example, the fields in the Product table are converted to attributes on the Product label node, and the conversion result is shown in Figure 1.

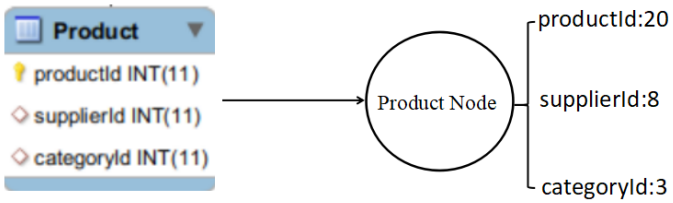

Fig. 1. Node conversion

2) Relationship Conversion

The relationship in the graph data model can be constructed in two ways.One is composed of the same fields in different tables. This method is called direct construction; The second is composed of foreign key combinations in different tables.This method is also called indirect construction method.Firstly, the foreign keys in the two tables are formed into an intermediate connection table. The connection table is used to form a directional relationship between different nodes, and each field in the connection table becomes a attribute of the relationship. The relationship conversion algorithm is as follows:

Algorithm 2: relationship conversion algorithm

Input: the table name (table I, table II...) and the field name of the corresponding table (tableI.filed1, table II. filed1.....)

Output: Relationship between nodes (Relationship:A1---[....]---->A2)

```
1.  i←1,j←1,m←1,n←1,tag←0
2.  For table[i] ≠∅ and table[j] ≠∅ do
        For table[i].filed[m]≠∅ and table[j].filed[n]≠∅ do
            If table[i].filed[m]==table[j].filed[n] then
                tag←1
                Break
          m←m+1
          n←n+1
    i←i+1
    j←j+1
3.  If tag==1 then
          Use direct construction
          Return Relationship
    Else
          Use indirect construction
          Return Relationship
```

Based on the conversion algorithm 2, taking the Organization table and the Person table as an example, it can be seen from the analysis that the personID of the Person table and the orgID of the Organization table are combined into a Dept_Member intermediate connection table using the indirect construction method, and Finally, the relationship that the Person node belongs to the Organization node is formed.the conversion result is shown in Figure 2.

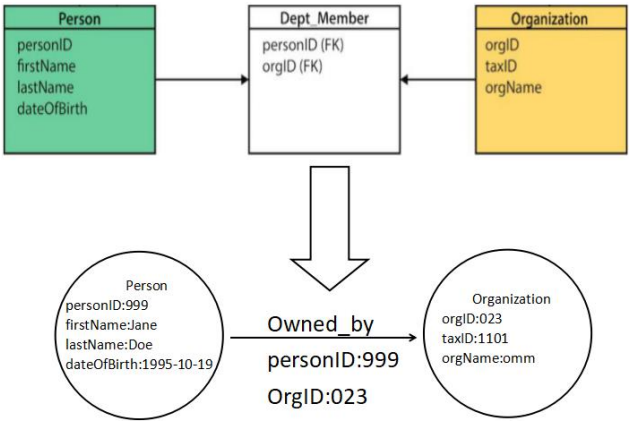

Fig. 2. Relationship conversion

B. Constraint handling

Relational data tables will add some restrictive conditions to certain fields, such as initial default values, uniqueness of field values, and auto-increment fields located on the logical primary key. The methods to keep these constraints are shown in Table I.

TABLE I. CONSTRAINT HANDLING

Constraint type	Approach
With initial default value	Delete the default value
Field uniqueness	Use constraint statements (Equation 1)
Auto-increment field	Delete auto increment field

The constraint statement (Equation 1) in the table is:

CREATE CONSTRAINT ON (variable name: label name) ASSERT variable name.Attribute name IS UNIQUE

In addition, for some irregular data tables that contain duplicate records, you need to create nodes separately. Finally, the data type and null value processing are handled by the CQL (Cypher Query Language) function officially provided by graph database.

C. Data migration rules

On the basis of data conversion, the basic framework of graph data model has been built. The next task is to transfer specific data. The data migration method proposed in this section is different from the conventional data import method, which is the CSV file import method officially proposed by the graph database. However, the CSV import method cannot filter some illegal data, which affects the validity and accuracy of the imported data. The advantage of this method is to transfer the data accurately and without errors.

Based on the data conversion rules, data migration requires two steps: one is the extraction of metadata and table data; the other is to import the extracted data into the graph database. Data extraction refers to the use of Python technology to crawl the fields and records needed for effective metadata from the relational database. The import of data is based on the established graph data model, and using the extracted valid data as the input terminal,according to established the graph data model,is input into the API provided by the graph database.Import and migration methods are shown in Figure 3.

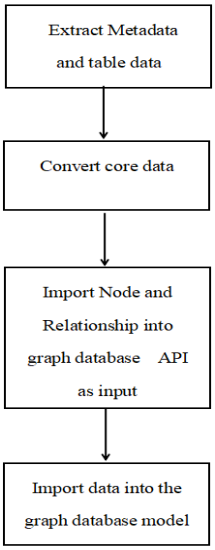

Fig. 3. Data migration process

III. DATABASE CONVERSION FROM MYSQL TO NEO4J

This section will show the data conversion and data migration methods proposed in the previous section from two specific different databases. First of all, the traditional database uses the MySQL database, and the graph database uses the Neo4j database. In addition, these two databases can be switched to other databases of the same type at will, such as other relational databases PostgreSQL, Oracle, etc, graph databases TigerGraph [14], HBase [15], etc. Therefore, this chapter mainly introduces the specific implementation and application based on the method proposed above.

A. Data conversion

Based on the mapping rules of nodes and relationships proposed above, this conversion will be based on conversion to Neo4j. In the graph database Neo4j, a node can be assigned multiple labels, and a node can have several attributes. The data type of each attribute is consistent with the data type of the corresponding field in the relational database,which is handled by Neo4j's built-in CQL function.Take the three tables in Figure 4 as an example.

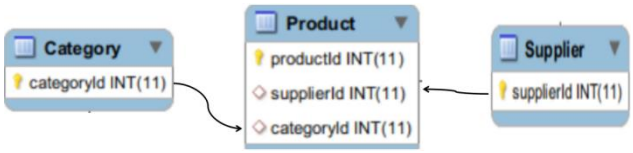

Fig. 4. E-R diagram before conversion

1) Node Conversion

As shown in Figure 4,firstly,using MySQL Workbench draws the structure of the table in the relational database and the relationship between different tables.Then,convert the Product table.At first , name the table name as the label name after which all records in the table are converted to nodes. All fields in the record are converted into attributes of the corresponding node, and the data type of each corresponding attribute is consistent with the field type. The specific creation statement is as follows:

CREATE (n:Product)
set n = row,
n.productId = toInteger(row.productId),
n.supplierId = toInteger(row.supplierId),

n.categoryId = toInteger(row.categoryId)

Follow the above statement to create the overall structure of the node, other data tables also follow this method to create nodes, and then perform the constraint processing in Table I. When all the data tables have been processed, the creation of all nodes is completed. The created node is shown in Figure 5.

Fig. 5. *Node graph after conversion*

2) Relationship Conversion

Since the associations between the canonical relational data tables basically belong to the type shown in Figure 2 and Figure 4 above, this section will take the data tables in Figure 2 and Figure 4 as examples.

Firstly, take Figure 4 as an example. From the ER diagram, it can be seen that the primary key categoryId in the Category table is a field in the Product table, and the primary key supplierId in the Supplier table is also a field in the Product table, so it can be constructed The relationship between the three nodes in the graph data model, and the relationship is also called a directed edge in the graph data model, so they will form a directional relationship.After constructing the node relationship of different labels with the direct construction method, the data will be migrated to the graph data model in the next step. The relationship creation statement is as follows:

a) Product-----PART_OF------→Category

MATCH (p:Product),(c:Category)
WHERE p.categoryID = c.categoryID
CREATE (p)-[:PART_OF]->(c);

b) Supplier----SUPPLIES----→Product

MATCH (p:Product),(s:Supplier)
WHERE p.supplierID = s.supplierID
CREATE (s)-[:SUPPLIES]->(p)

This direct construction method is suitable for the parts that directly overlap the fields in different tables. It is not limited to the overlap of the primary key. Any field overlap can be converted by this method.

The indirect construction rule applies to tables that have no overlap in table fields, but are related in business logic. The main solution is to solve the problem through creating the intermediate connection table. Taking the two tables in Figure 2 as an example, The white Dept_Member table is the intermediate connection table constructed by the indirect construction method, and the Person and Organization tables are the business data tables. From the structure and business logic of the two tables, it can be judged that the person belongs to a certain organization or certain organizations. An intermediate table containing the primary keys of the two business data tables is created with this relationship, and the fields on the intermediate table are relationships attributes. The creation statement is the same as above, so that the conversion of the relationship is completed through the indirect construction method.The relationship creation statement is as follows:

c) Person----OWNED_BY----→Organization

MATCH(p:Person),(m:Dept_Member),(n:Organization)
WHERE p.personID = m.personID And
n.orgID = m.orgID
CREATE (p)-[:OWNED_BY]->(n)

B. Data migration

The steps of data migration mainly obtain valid data from log analysis and metadata analysis, and then carry out the process of data import. The main process of data migration is as follows:

1) JDBC connects to the background database;

2) Use Python crawler technology to extract core data, mainly extracting table structure and fields in table data;

3) The required core data obtained by calling Java library functions;

4) After converting the obtained core data into nodes and relationships through data conversion rules, the nodes and relationships are used as the two parameters of the graph database API input terminal, and finally processed by the graph database batch processor.

In the process of data import, the process of data type conversion is also involved. The realization of this process is mainly completed by the built-in CQL function of Neo4j. The migration process is shown in Figure 6.

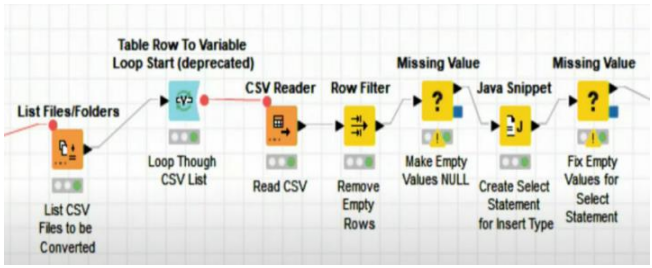

Fig. 6. Data migration

IV. Experiment

The experimental hardware configuration and experimental software environment are shown in Table II.

TABLE II. Experimental software and hardware environment

Experimental software and hardware environment	Specific name
processor	Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz
RAM	DDR4 2400 8GB
hard disk	Mechanical hard disk500GB
operating system	CentOS Linux release 7.8.2003 (Core) and Windows10
software platform	MySQL5.7.31Community Edition、Neo4j3.5.28Community Edition and Pycharm2020.1.1Professional Edition

The Northwind dataset used in the experiment is a sample dataset provided by SQL Server. Based on this dataset, we compare the query efficiency of the two databases in different business scenarios. In the experiment, Neo4j uses Cypher query language, and MySQL uses SQL statements for query. This data set Northwind describes the

relevant data about the orders of the freight sales system, which contains 13 tables, as shown in Table III.

TABLE III. NORTHWIND DATA SHEET

Name	Core field	Data entry	Function meaning
Category	categoryId(PK)	8	Type of goods and description of goods
Product	productId(PK) supplierId categoryId	77	Commodity information
Supplier	SupplierId(PK)	29	Supplier Information
Employee	employeeId(PK)	9	employee information
EmployeeTerritory	employeeId(FK) territoryId(FK)	49	Employee location connection table
Region	regionId(PK)	4	Regional distribution information
OrderDetail	orderDetailId(PK) orderId productId	2155	Commodity order connection table
Territory	territoryId(PK) regionId	53	Location and area connection table
SalesOrder	orderId(PK) custId(FK) shipperId	830	Freight order sales information
Customer	custId(PK)	91	Customer information
Shipper	shipperId(PK)	3	Shipping information
CustomerDemographics	customerTypeId(PK)	0	Customer type details
CustCustDemographics	custId(FK) customerTypeId(FK)	0	Customer Type Connection Table

The experiment will first perform relational data modeling, import the data into MySQL, and then use the custom conversion method proposed above to perform data conversion to form a graph data model, then use Python for data extraction, and Finally, it is imported into Neo4j by Neo4j API to complete graph data modeling. The next step is to compare the two databases, and the main ones are data integrity comparison and query efficiency comparison experiments to verify the feasibility of the method.

Based on the node and relationship conversion method proposed in the third section, Firstly, draw the E-R diagram of 13 tables by Table III to analyze the structure of the entire data set, and then convert them into nodes of Category, Customer, Supplier, Product, and Order five types of labels according to the node conversion method.The E-R diagram of freight order system is shown in Figure 7.

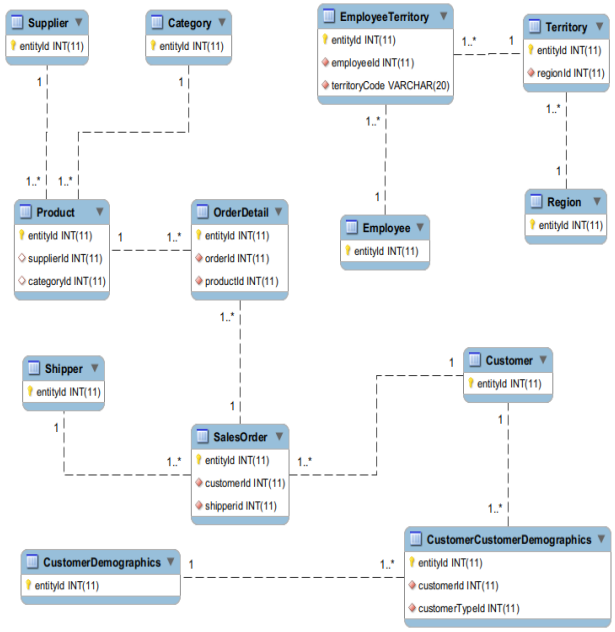

Fig. 7. E-R diagram of freight order system

The relationship between the nodes is constructed as follows: the relationship between Category and Product, Supplier and Product, and Order and Customer are constructed by the direct construction method in the above-mentioned relationship conversion. The categoryId field in the Category table and the supplierId field in the Supplier table exist in the Product table, and the custId field in the Customer table exists in the SalesOrder table. Therefore, three relationships can be constructed: Product is a part of Category, Supplier supplies Product, and Customer owns Order.In addition, because the Product table and the SalesOrder table are not directly related to the fields, an intermediate connection table of OrderDetail is created to connect the Product and SalesOrder tables. The OrderDetail table contains the productId in the Product table and the orderId field in the SalesOrder table, based on indirect construction Method can construct the relationship that Order contains Product. The structure model result of the graph database after conversion is shown in Figure 8.

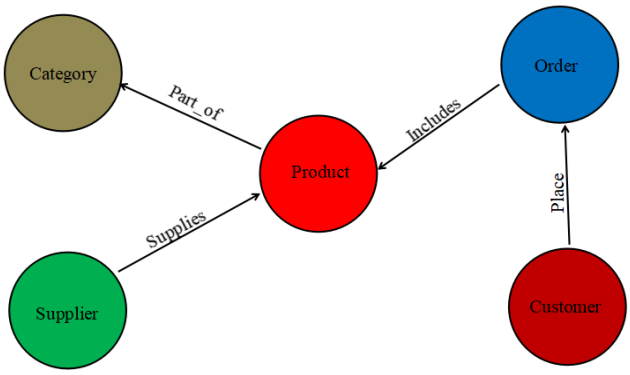

Fig. 8. Node and relationship graph

After creating the graph data model, perform data migration based on this model, which is the process of creating a large number of data nodes and relationships. The result of the creation is compared with Table III to verify the data integrity of the graph database.The results are shown in Table IV and Table V.

TABLE IV. NODE DATA

Node label type	Number of nodes	Node attributes
Product	77	productId supplierId categoryId
Category	8	categoryId
Supplier	29	SupplierId
Customer	91	custId
Order	830	orderId custId shipperId

TABLE V. RELATIONSHIP DATA

Relationship type	Number of relationship
Product--[Part_of]-->Category	77
Order--[Includes]-->Product	2155
Supplier--[Supplies]-->Product	77
Customer--[Place]-->Order	830

In order to verify the validity of the data conversion, data query comparison between the two databases was performed. The query results are shown in Table VI and Table VII.

TABLE VI. NEO4J OF QUERY TIME

Meaning of query statement	Cypher query statement	Result	Query time(ms)
What types of goods does each company produce	MATCH (s:Supplier)-->(:Product)-->(c:Category) RETURN s.companyName as Company, collect(distinct c.categoryName) as Categories	BWGYE--Confections,.....ZWZDM--Grains/Cereals(29 records)	8
The number of companies that have produced 'Produce' products	MATCH (c:Category {categoryName:"Produce"})<--(:Product)<--(s:Supplier) RETURN DISTINCT s.companyName as ProduceSuppliers	JNNES,QOVFD,QWUSF,STUAZ,SVI YA	2

TABLE VII. MYSQL OF QUERY TIME

Meaning of query statement	SQL query statement	Result	Query time(ms)
What types of goods does each company produce	select companyName,group_concat(distinct categoryName) from Product join Category on Product.categoryId=Category.categoryId join Supplier on Product.supplierId= Supplier.supplierId group by companyName	BWGYE--Confections,.....ZWZDM--Grains/Cereals(29 records)	50
The number of companies that have produced 'Produce' products	select companyName from Product join Category on Product.categoryId=Category.categoryId join Supplier on Product.supplierId= Supplier.supplierId where categoryName='Produce' group by companyName	JNNES,QOVFD,QWUSF,STUAZ,SVIYA	20

From the query results and query time in Table VI and Table VII, it can be concluded that the connection query speed of graph database far exceeds that of relational database and the results are also accurate, which proves that the custom conversion method proposed above is effective and highly usable.it can be predicted that in the future, as the volume of the data set gradually increases and the relationship becomes more and more complicated, the advantages of the graph database will become more prominent. This is also the purpose of this method.

V. CONCLUSION

In order to overcome the dependence of traditional conversion methods on data sets, this paper proposes a custom database conversion method, and adds a data migration function in the middle. This method realizes the conversion of various relational databases and ensures the consistency of the data before and after the conversion and the consistency of the functions.However, this method still has certain shortcomings. For example, after completing some data set conversions, it is less efficient to implement some simple queries.

ACKNOWLEDGMENT

Project supported by the National Natural Science Foundation of China(Grant 61672004).

REFERENCES

[1] IAN ROBINSON, JIM WEBBER, EMIL EIFREM. Graph Databases[M]. O'Reilly Media, Inc:Cambridge,2015:4-5.

[2] Sack H,Schuetz Christoph G.,Bozzato L,Neumayr B,Schrefl M,Serafini L. Knowledge Graph OLAP[J]. Semantic Web, 2021, 12(4):649-683.

[3] Bouamrane K,Matallah H,Belalem G. Comparative Study Between the MySQL Relational Database and theMongoDB NoSQL Database[J]. International Journal of Software Science and Computational Intelligence(IJSSCI),2021,13(3):38-63.

[4] OZGUR C, COTO J, BOOTH D. A comparative study of network modeling using a relational database (eg Oracle, mySQL, SQL server) vsNeo4j[C]//Conference Proceedings By Track. 2017:156-165.

[5] MAGORZATA PW, RYKOWSKI D. Comparison of Relational, Document and Graph Databases in the Context of the Web Application Development[M]// Information Systems Architecture and Technology: Proceedings of 36th International Conference on Information Systems Architecture and Technology. Springer International Publishing, 2016:3-13.

[6] Cooray Sh. A qualitative enhancement to quantitative social network analysis[J]. The Information Society,2021,37(4):227-246.

[7] Anoop V.S.,Deepak P.,Asharaf S.. A distributional semantics-based information retrieval framework for online social networks[J]. Intelligent Decision Technologies,2021,15(2):189-199.

[8] Shawni Dutta,Samir Kumar Bandyopadhyay. Recommender System for Term Deposit Likelihood Prediction using Cross-validated Neural Network[J]. South AsianJournal of Social Studies and Economics,2021:21-28.

[9] K Sai Prasanthi,A K Subash,B Manoj,P Sunil Kiran. Cost Optimization Algorithm for Data Center Manage-ment in Cloud[J]. International Journal of InnovativeTechnology and Exploring

Engineering (IJITEE),2019,9(2):4778-4781.

[10] TANG Jianping, YE Xinliang, SUN Ruihong. Analysis of Management Research Based onCitespace[J]. Logistics Technology, 2021:40-43.

[11] De VIRGILIO R, MACCIONI A, TORLONER. R2G:a Tool for Migrating Relations to Graphs[C]// International Conference on Extending Database Technology,2014:640-643.

[12] Yelda Unal,Halit Oguztuzun. Migration of data from relational database to graph database[P]. Information Systems and Technologies, 2018:1-5.

[13] Roberto De Virgilio,Antonio Maccioni,Riccardo Torlone. Converting relational to graph databases[P]. Graph Data Management Experiences and Systems,2013:1-6.

[14] GUO Tao.TigerGraph allows everyone to analyze graphs[J]. Network Security and Informatization,2020(09):13-16.

[15] Ghalem Belalem,Houcine Matallah,Karim Bouamrane. Evaluation of NoSQL Databases: MongoDB, Cassandra, HBase, Redis, Couchbase, OrientDB[J]. International Journal of Software Science and Computational Intelligence (IJSSCI),2020,12(4):71-91.