Data Query Method of Science and Technology Management Based on Relational Engine

Dong Wang
National Science and Technology Plan Management Support Center
Institute of Scientific and Technical Information of China
Beijing, China
wangd@istic.ac.cn

Zhigang Zhang
Comprehensive Service Center
Institute of Scientific and Technical Information of China
Beijing, China
zzg@istic.ac.cn

Chenyang Xu, Zhuohao Wang
National Science and Technology Plan Management Support Center
Institute of Scientific and Technical Information of China
Beijing, China
{xucy, wangzh}@istic.ac.cn

2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA) | 978-1-6654-2311-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCCBDA51879.2021.9442601

***Abstract*—As the country promotes the reform of science and technology project management, the number of science and technology projects has been increasing. At the same time, science and technology management data is growing rapidly with the characteristics of complex correlation and multi-dimensionality. How to effectively store the enormous data and mine the contained value of the data have become a crucial problem for current researches. First, we propose a science and technology management data model to illustrate the internal relation of project, topic, unit, and person, and then define the model by mathematical representation. According to science and technology management data query requirements, we present a data query method of science and technology management based on relational engine, which comprises query parser, query optimizer, and query executor. Furthermore, we design a prototype system to achieve the query requirements of science and technology management data. This research contributes to data query by providing a query method with relational engine, and improves the efficiency of data query in science and technology management.**

Keywords—data query, data model, relational engine, query optimization, science and technology project

I. INTRODUCTION

With the continuous advancement of national science and technology management reform, the support for science and technology projects has been increasing gradually. Science and technology management data is growing rapidly with the characteristics of complex correlation and multi-dimensionality. How to effectively integrate, store, mine, and analyze data throughout the entire process of science and technology management has become particularly important. Due to the diverse types of science and technology projects, complex management processes, and continuous changes in business logic, the characteristics of science and technology management data are multi-source and heterogeneous. The data contains significant value to help the development of a country and the analysis of government decision-making. Fully mining the potential value of data can help the government implement the reform of science and technology management, grasp the trend of science and technology, and respond to the technological innovation market to produce new decisions. Therefore, how to perform correlation analysis on science and technology management data and dig out greater value has become the focus of this paper.

S. P. L. Filho et al. proposed some heuristics aiming to mapping system from relational model data to graph representation [1]. V. Filatov and V. Semenets proposed an approach to detecting previously unknown functional dependences based on the analysis of a set of relational database [2]. S. Salloum and J. Ming et al. proposed a new method on the construction of distributed data model [3], [4]. The above methods are all focus on the improvement of the data relation model. Considering the actual characteristics of science and technology management data, this paper investigates data model based on the above models and the analysis of data model tools [5]. T. Z. Emara and M. Guizhi et al. carried out the research on data analysis methods from the perspectives of data analysis methods, analysis techniques, and big data analysis [6]-[11]. However, the methods above connect closely to their specific research fields, they are not effective to science and technology management data query. Therefore, this paper proposes a data query method of science and technology management based on relational engine.

II. DATA MODEL

D. Wang [12] proposed a project state model which contains proposal, review, implementation, and acceptance states of project life cycle. Considering to the previous study, we divide the whole process of science and technology management into declaring, approval, implementation, and acceptance. The science and technology projects take the project as the main body, each project is composed of several topics, each topic is participated by different units, and each unit contains hundreds of persons. Combining the characteristics of science and technology management business and project composition, we propose a science and technology management data model (STMDM) shown in Fig. 1.

As we can see from STMDM, the project is the core component of the entire model, and the relations of Project:Topic, Topic:Unit, Unit:Person are *1:N*. Projects, topics, units, and persons have unique dimensional characteristics and are linked through the subsidiary table. Based on the above relations, we define the data model of science and technology management by using mathematical symbols.

978-0-7381-0533-8/21/$31.00 ©2021 IEEE

The overall science and technology management data model is defined as:

$$\begin{aligned} &\boldsymbol{STMDM} \\ &= \{\boldsymbol{Project}, \boldsymbol{Topic}, \boldsymbol{Unit}, \boldsymbol{Person}, \boldsymbol{Subsidiary}\} \\ &= \{\boldsymbol{Pro}, \boldsymbol{Top}, \boldsymbol{Uni}, \boldsymbol{Per}, \boldsymbol{Sub}\} \end{aligned}$$

The project model can be defined as:

$$\boldsymbol{Pro} = \{\boldsymbol{pro_1}, \boldsymbol{pro_2}, \boldsymbol{pro_3}, \dots, \boldsymbol{pro_I}\}$$

The topic model can be defined as:

$$\boldsymbol{Top} = \{\boldsymbol{top_1}, \boldsymbol{top_2}, \boldsymbol{top_3}, \dots, \boldsymbol{top_J}\}$$

The unit model can be defined as:

$$\boldsymbol{Uni} = \{\boldsymbol{uni_1}, \boldsymbol{uni_2}, \boldsymbol{uni_3}, \dots, \boldsymbol{uni_M}\}$$

The person model can be defined as:

$$\boldsymbol{Per} = \{\boldsymbol{per_1}, \boldsymbol{per_2}, \boldsymbol{per_3}, \dots, \boldsymbol{per_N}\}$$

The logical relation of a single project can be described as:

$$\boldsymbol{Pro_i} = \{\boldsymbol{Top_{1-j}}\} \cup \{\boldsymbol{Uni_{1-m}}\} \cup \{\boldsymbol{Per_{1-n}}\} \cup \{\boldsymbol{Sub}\}$$

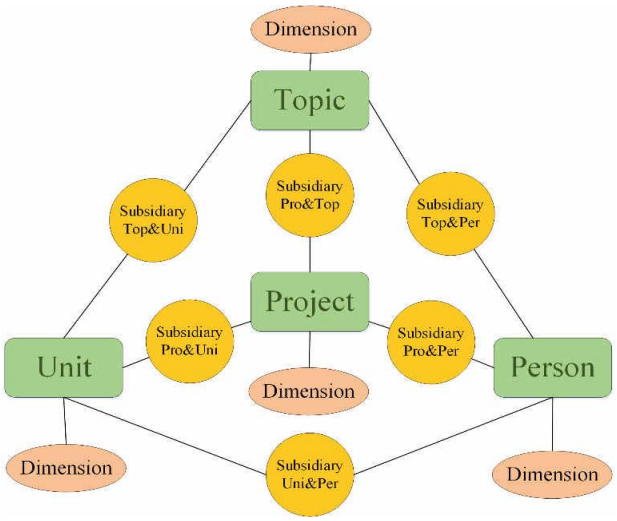

Fig. 1. STMDM.

An example of project model is shown in Fig. 2. The structure of the model is a tree, which adopts a project as a root node. The second level is constituted by topics, while the third level is constituted by units. At last, the bottom of the model is composed by leaf nodes that are persons.

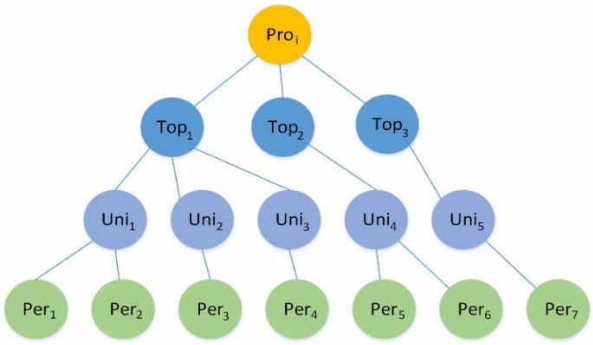

Fig. 2. An example of project model.

III. Data Query Method

In this section, we propose a data query method of science and technology management, which comprises data storage layer, query process layer, and query input layer. It is shown in Fig. 3.

A. Data Storage Layer

Data storage layer generates a data storage structure based on STMDM, and uses ETL tools to integrate science and technology management data from original business system.

B. Query Process Layer

Query process layer is a bridge between query input layer and data storage layer. In query process layer, we adopt a relational engine, which contains query parser, query optimizer, and query executor. The details of the relational engine are introduced in the following section.

C. Query Input Layer

Query input layer is an interface for users to employ the proposed method in data query. It is used for user interaction by query condition and result display. When user executes a set of query conditions, a query command is created and sent to query process layer for data query.

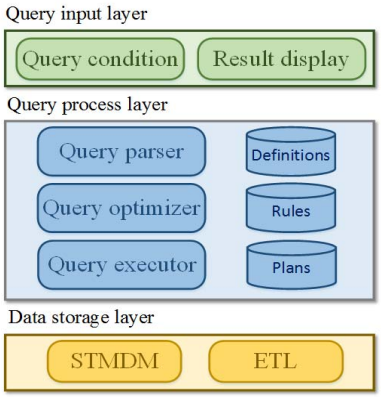

Fig. 3. The proposed data query method.

IV. Relational Engine

Relational engine is the core component of the query processing layer. In this section, we focus on the process of relational engine combining with STMDM, which enhances the efficiency of data query in science and technology management. The conceptual workflow of relational engine is shown in Fig. 4.

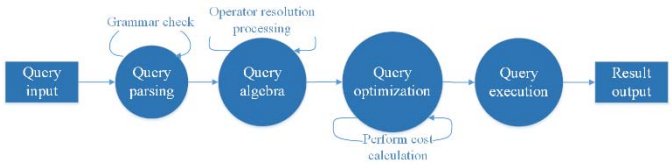

Fig. 4. The conceptual workflow of relational engine.

A. Query Parser

The main function of the query parser is to parse query command. When the query parser obtains the query command, it checks whether there is an error in the query command, and

then parse the command. After query command is processed algebraically, it is delivered to the query optimizer.

When a query command is arrived, the query parser converts the query command to a query statement, and then analyzes whether there is a syntax error in the query statement. After the grammar is recognized correctly, the query statement is processed algebraically and completely parsed into a digital schema. In general, relational algebra uses selection, projection, connection, and division to represent query, and SQL query statement is denoted by a structure (select-from-where). Adding the 'where' condition represents a selection symbol in relational algebra. The attribute column after selection is equivalent to projection. If there are several data tables after 'from', the connection attributes of each data table need to be added in the 'where' condition.

In order to illustrate the converting of query statement by relational algebra, we present the following definitions.

Definition one: Algebraic expression of query statement.

M and *N* each represent a relation, πT is projection (T is projection attribute), σ_C is selection (C is selection condition), γT is aggregation and grouping (T is attribute), ∞C is connection (C is connection condition), and × is product of the relation.

Definition two: Corresponding conversion rules.

1) $\sigma_{C1\ AND\ C2}(M) = \sigma_{C1}(\sigma_{C2}(M))$

2) $\sigma_{C1\ OR\ C2}(M) = (\sigma_{C2}(M)) \cup (\sigma_{C2}(M))$

3) $\sigma_{C1}\ (\sigma_{C2}(M)) = \sigma_{C2}\ (\sigma_{C1}(M))$

4) $\sigma_C\ (M \infty N) = \sigma_C\ (M) \infty N$

5) $\pi T(M \infty N) = \pi T(\pi O(M) \infty \pi P(N))$ *O and P are the attribute lists of M and N respectively*

6) $\pi T(M \times N) = \pi T(\pi O(M) \times \pi T(N))$

7) $(M \infty C^N) = \sigma_C\ (M \times N)$

Taking a database of STMDM as an example, projects and units have the following fields.

Project: (Pid, Pname, Pfield, Pyear);

Unit: (Uid, Pid, Uname, Ucity);

Pid is a major key of a project. *Pname* denotes a project name while *Pfield* denotes the research field of the project. *Pyear* is the year of created time of the project. *Uid* is a major key of a unit. *Uname* represents a name of the unit. *Ucity* is a location of the unit.

For example, there is a requirement of calculating the number of projects in the biological field declared by Beijing units in 2016. The query statement can be represented as:

Select count（P. Pid） from Project P,Unit U where P. Pid=U.Pid and P.Pyear=2016 and P.Pfield like 'biological' and U.Ucity like 'Beijing'.

The algebraic expression of the above statement is shown in Fig. 5.

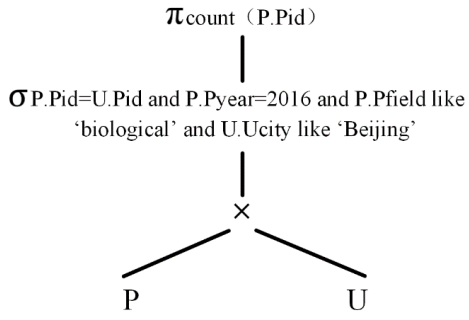

Fig. 5. The algebraic expression of a statement.

B. *Query Optimizer*

Query optimization is an intermediate process in the whole relational engine. When the query optimizer receives a query statement, it checks the diverse operations of the query statement (such as selection, conditions, joins, and so on), lists all execution policies for the query statement. It estimates the cost of every operation in different execution policies. According to the results, an effective execution plan is generated [13]-[16]. The process of query optimizer is shown in Fig. 6.

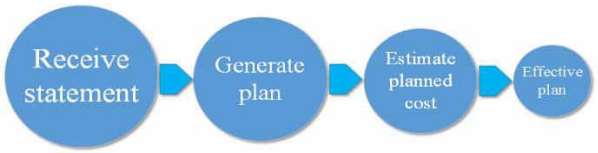

Fig. 6. The process of query optimizer.

When the query statement is very simple, the query optimizer generates each possible execution plan and estimates the cost of each plan immediately. Oppositely, when the query statement is very complex, it takes more time to estimate the cost of every possible query plan, and the query optimizer may list some inefficient execution policies. Therefore, some basic optimization rules need to be defined to ensure that the policies listed by the query optimizer are as efficient as possible for the query statement.

We define four basic rules, which are used in the optimization process of query statements.

- The selection and projection operations are performed as early as possible, which would reduce the number of tuples and the size of the relation.
- Combining selection operations with projection operations, the cost of intermediate process results can be reduced effectively.
- The query optimizer can perform multiple selection and projection operations on the same relation simultaneously, which prevents repetitive execution of same relation.
- Appropriately, the query optimizer can perform a combination of projection and join operations.

We use a query example to describe how the query optimizer executes the query statement. The query statement of searching unit names of each project in Beijing is as follows.

Select U.Uname from Project P, Unit U where P.Pid=U.Pid and U.Ucity like 'Beijing'.

The query optimization process is as follows, shown in Fig. 7.

- Convert the query statement into a relational algebraic expression: $\pi L(\sigma_C(P \times U))$. L="*U.Uname*", C="*P.Pid=U.Pid and U.Ucity like 'Beijing'*".
- Split "*P.Pid=U.Pid and U.Ucity like 'Beijing'*" into "*P.Pid=U.Pid*" and "*U.Ucity like 'Beijing'*" by using rules.
- Move "*U.Ucity like 'Beijing'*" down, divide the projection operation $\pi U.Uname$ into two and move them under the product respectively. Then, combine the selection operation and the projection operation into a join operation.

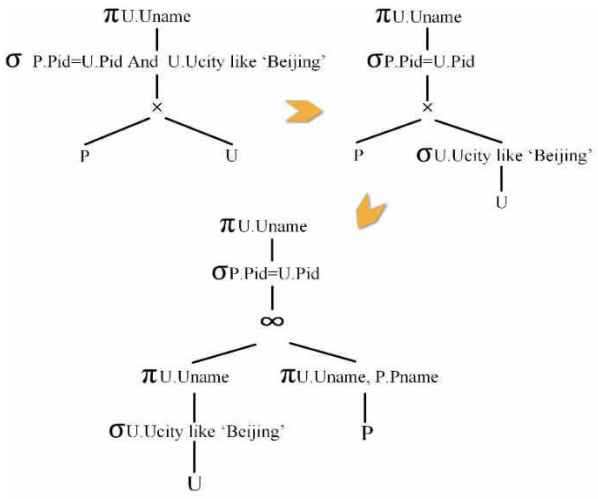

Fig. 7. An example of query optimization process.

C. Query Executor

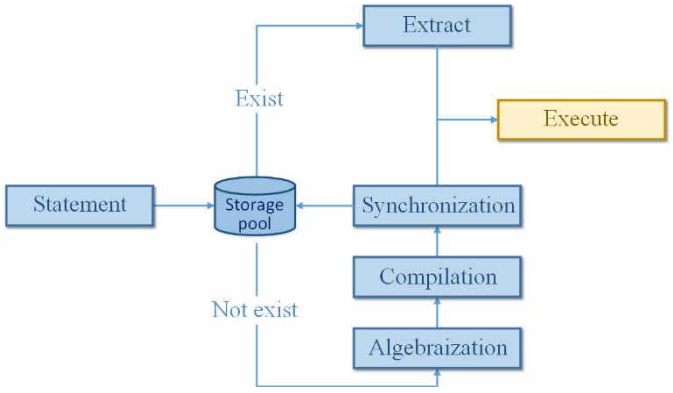

Fig. 8. The process of query executor.

There is a storage pool to store execution plans in the query executor. An execution plan is a data structure that can be used repeatedly. When a query executor executes a statement, it scans the storage pool to search if there is a same structured plan. If so, the same plan in the storage pool is extracted to improve the efficiency of the statement execution. If not, the query executor generates a new execution plan for the statement, which is executed after algebraization, compilation, and synchronization to the storage pool. The process is shown in Fig. 8.

On the basis of the process of query executor, we propose a statement execution algorithm.

Statement execution algorithm:
Input: *A Sql Query Statement (SQS)*
For each SQS **do**
Search from storage pool;
If exist then
do Extract(SQS);
else
do Algebraization(SQS);
do Compilation(SQS);
do Synchronization(SQS);
End if
Result = **Execute**(SQS);
End for
Output: *Result*

V. The Prototype System

After investigating the specific development workflow of data query system [17]-[19], we start the implementation of our proposed method. Based on STMDM and relational engine, we present a prototype system to prove the processes and algorithm of relational engine. As shown in Fig. 9.

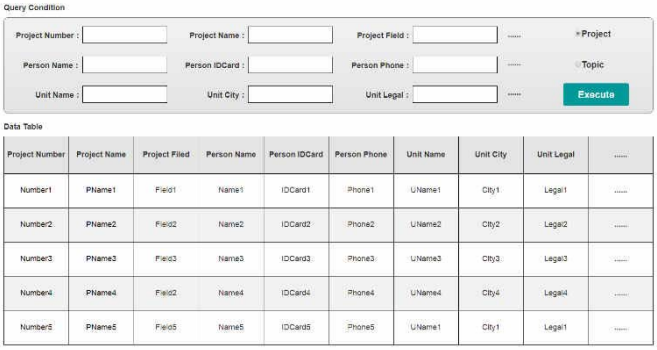

Fig. 9. A Prototype system.

The prototype system comprises query condition, execute button, and data table. The query condition satisfies the need of multiple dimensions of project, topic, unit, and person based on STMDM. The execute button is used by user to perform the filled query conditions. The data table displays the query results which are generated by relational engine. The system is mainly used to accept query condition and display query results that are more intuitive and satisfy decision analysis for different dimensions and query requirements.

VI. Conclusion

This paper proposes STMDM for the needs of science and technology management data storage and query application. For scenarios of science and technology management data query, we propose a data query method based on relational engine. Based on STMDM and relational engine, this paper constructs a science and technology management data query prototype

system to support complex query requirements. In the face of ever-increasing data, how to support the greater demand for data storage and analysis is the key issue we will focus on in the future.

ACKNOWLEDGMENT

The paper is sponsored by fund (Innovation Research Fund Project of the Institute of Scientific and Technical Information of China, MS2019-06).

REFERENCES

[1] S. P. L. Filho, M. C. Cavalcanti and C. M. Justel, "Graph modeling from relational databases," 2017 XLIII Latin American Computer Conference (CLEI), Cordoba, Sept. 2017, pp. 1-10, doi: 10.1109/CLEI.2017.8226384.

[2] V. Filatov and V. Semenets, "Methods for Synthesis of Relational Data Model in Information Systems Reengineering Problems," 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, Oct. 2018, pp. 247-251, doi: 10.1109/INFOCOMMST.2018.8632144.

[3] S. Salloum, J. Z. Huang and Y. He, "Random Sample Partition: A Distributed Data Model for Big Data Analysis," in IEEE Transactions on Industrial Informatics, vol. 15, no. 11, pp. 5846-5854, Nov. 2019, doi: 10.1109/TII.2019.2912723.

[4] J. Ming, L. Zhang, J. Sun and Y. Zhang, "Analysis models of technical and economic data of mining enterprises based on big data analysis," 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Chengdu, Apr. 2018, pp. 224-227, doi: 10.1109/ICCCBDA.2018.8386516.

[5] K. Host, D. Jaksic and P. Poščić, "Overview and comparison of the selected relational data modelling tools," 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, May. 2018, pp. 1592-1597, doi: 10.23919/MIPRO.2018.8400286.

[6] T. Z. Emara and J. Z. Huang, "Distributed Data Strategies to Support Large-Scale Data Analysis Across Geo-Distributed Data Centers," in IEEE Access, vol. 8, pp. 178526-178538, Sept. 2020, doi: 10.1109/ACCESS.2020.3027675.

[7] J. Chen, Q. Jiang, Y. Wang and J. Tang, "Study of data analysis model based on big data technology," 2016 IEEE International Conference on Big Data Analysis (ICBDA), Hangzhou, Mar. 2016, pp. 1-6, doi: 10.1109/ICBDA.2016.7509810.

[8] R. Hu, "Key Technology for Big Visual Data Analysis in Security Space and Its Applications," 2016 International Conference on Advanced Cloud and Big Data (CBD), Chengdu, Aug. 2016, pp. 333-333, doi: 10.1109/CBD.2016.065.

[9] J. Um, S. Han, H. Kim and K. Park, "Massive OceanColor Data Processing and Analysis System: TuPiX-OC," 2017 IEEE 13th International Conference on e-Science (e-Science), Auckland, Oct. 2017, pp. 450-451, doi: 10.1109/eScience.2017.66.

[10] S. R. Sukumar, M. A. Matheson, R. Kannan and S. Lim, "Mini-apps for high performance data analysis," 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, Dec. 2016, pp. 1483-1492, doi: 10.1109/BigData.2016.7840756.

[11] M. Guizhi, "Application of big data analysis in oil production engineering," 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), Beijing, Mar. 2017, pp. 447-451, doi: 10.1109/ICBDA.2017.8078859.

[12] D. Wang, M. Zhou, S. Ali, P. Zhou, Y. Liu, and X. Wang, "A Novel Complex Event Processing Engine for Intelligent Data Analysis in Integrated Information Systems," International Journal of Distributed Sensor Networks, Mar. 2016, pp.1-14. doi: 10.1155/2016/6741401

[13] W. Zheng, "Database Query Optimization Based on Parallel Ant Colony Algorithm," 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), Chongqing, Jun. 2018, pp. 653-656, doi: 10.1109/ICIVC.2018.8492789.

[14] A. P. Chendke, S. S. Sherekar and V. M. Thakare, "Advanced query processing and its optimization for mobile computing environment," 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, Jan. 2018, pp. 313-318, doi: 10.1109/ICISC.2018.8399085.

[15] Y. Xu, "Dynamic Optimization Analysis of Keyword Query Results in Relational Databases Based on Ant Colony Optimization Algorithm," 2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC), Dalian, China, Dec. 2017, pp. 721-724, doi: 10.1109/ICCTEC.2017.00160.

[16] S. Mehta, P. Kaur, P. Lodhi and O. Mishra, "Empirical Evidence of Heuristic and Cost Based Query Optimizations in Relational Databases," 2018 Eleventh International Conference on Contemporary Computing (IC3), Noida, Aug. 2018, pp. 1-3, doi: 10.1109/IC3.2018.8530467.

[17] J. Xie and J. Luo, "Construction for the city taxi trajectory data analysis system by Hadoop platform," 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), Beijing, Mar. 2017, pp. 527-531, doi: 10.1109/ICBDA.2017.8078689.

[18] T. Zhang, "Construction and Application of Big Data Analysis Platform for Ideological and Political Education in Colleges," 2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Changsha, China, Jan. 2019, pp. 249-252, doi: 10.1109/ICITBS.2019.00066.

[19] Y. Cheng, W. Shang, L. Zhu and D. Zhang, "Design and implementation of ATM alarm data analysis system," 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, Jun. 2016, pp. 1-3, doi: 10.1109/ICIS.2016.7550948.