Research on Data Synchronization and Interconnection Scheme between Different Application Systems

Zhuo Liu[1], Botao Liu[1,*], Lixia He[1], Xiaoxi Liu[1], Liuyang Wang[1], Jun Xie[1] and Tianliang Guo[1]

[1]Yangtze University, Jingzhou 434000, China.

*Corresponding Author. liubotao920@163.com

Abstract

With the advent of the Big Data era, people are gradually realizing that "data is the new oil". Therefore, both governments and enterprises are building or have built their own data centers. As the cooperation between different units deepens, different data centers and the data interaction between different applications under the same data center become the key to information construction. This paper introduces three data synchronization techniques and their advantages and disadvantages, and proposes a proven data synchronization scheme based on JSON, and summarizes some problems encountered in the docking scheme and the areas that need improvement. This solution has been applied to the "AI-Well" system and has proven to be a viable solution that can greatly facilitate the actual data interfacing process.

Keywords

RDBMS; MySQL; Data Docking.

1. Introduction

Under the guidance of the national information development strategy, Internet applications and information technology are developing at a rapid pace, and the degree of informationization in various industries is increasing, the "big data era" has coming, According to McKinsey, "Data, which has permeated every industry and business function today, has become an important factor of production. The mining and use of massive amounts of data by people heralds a new wave of productivity growth and consumer surpluses." "Big Data" has been taken seriously by countries all over the world. "Data" has become the main form of information circulation and dissemination in all walks of life, and is becoming an important factor of production. Data systems have become an indispensable tool in the office, research, and even administration of all units [1].

Various enterprises, universities and work units have established independent data systems for each other according to their own needs. However, with the cooperation and sharing among different units, some of the data in their data systems need to be interacted with. Due to the dramatic increase in the number of various systems, data interactions are becoming more and more complex. In order to meet the increasingly complex data interfacing needs, developers have conducted comprehensive and deep research on data synchronization in recent years. Data synchronization technology allows data to be extracted from different data centers and loaded into different data centers after transformation. While ensuring data consistency, it is also necessary to ensure data timeliness, and data reusability. In the literature [2], the synchronization technology provided by DBMS is proposed as the core of the implementation of data synchronization between different databases, the advantages of this scheme are simple implementation, relatively high security and reliability, but due to the inherent defects of triggers in DBMS synchronization technology limits its use in data synchronization and has poor support for non-pairing types of data systems; the literature [3] studies and improves on data synchronization protocols, but the article is more theoretical and lacks practical implementation ideas; The literature [4] proposes data docking based on data exchange format and data conversion technology. Due to the complexity of XML technology, it is difficult to efficiently complete the docking task in the case of large amount of data. However, this one synchronization technology is more cross-platform, scalable and more reliable than traditional solutions, and will greatly enhance the feasibility of the solution if it can improve efficiency.

In summary, it is necessary and meaningful to study the data synchronization interconnection scheme between different application systems and ensure the security, timeliness and scalability of the scheme.

1.1 Data docking technology

Data docking is the process of transferring data from a source data source to another target data source or sources after matching, cleaning and transformation, and the data performance of both parties is consistent after the transfer is completed.

1.2 XML Technology

XML technology was the most adopted data interfacing technology in the early days, XML generally refers to Extensible Markup Language, which is self-descriptive and simple to understand. Based on XML it is possible to exchange data between incompatible systems and is often used in data exchange [5].

The advantages of XML are that the format is uniform and standards-compliant; it is compatible with different types of data systems and data sharing is easier. However, when transferring large amounts of data, huge XML files are formed and the file format is complex, which will significantly consume bandwidth in transmission and affect the efficiency of the database; both server-side and client-side need to spend a lot of code to parse XML, resulting in server-side and client-side code becoming extremely complex and not easy to maintain.

1.3 DML Technology

DML generally refers to the data manipulation language, which refers to the programming statements for accessing the objects and data in the database, including adding, deleting, modifying and checking operations [6]. The main principle of data docking in DML technology is to establish database association between docking databases, update them by trigger mechanism and timing mechanism, and synchronize data through DML statements.

The data docking based on DML technology independently developed can adopt different mechanisms according to different business, such as the trigger mechanism is required for data tables with uncertain data frequency and high timeliness requirements, while the time-setting mechanism can be adopted for data tables with no time limit requirements.

But in the practical application, it is found that data loss exists in the synchronization of DML trigger mechanism, which results in data loss, source database and target database can not guarantee data consistency. In addition, timing mechanism is needed to complete data. Database synchronization adopts multiple mechanisms to ensure data consistency and cause loss of database performance [6].

1.4 JSON Technology

JSON (JavaScript Object Notation) is a lightweight data interchange format [7]. JSON format has a concise and clear hierarchy that is easy for developers to write, read, inspect and maintain, and also easy for machines to parse and generate; JSON is easy to parse, it has broad support for browser and operating system compatibility, the format is compressed, takes up little bandwidth, and is fast to transmit. JSON-based data transmission is extremely fine-grained and can be accurate to each field, and there are many programming languages that support JSON format, so JSON format is one of the most popular data exchange formats on the Web. At the same time the Web provides a large number of libraries for manipulating JSON, such as: fastjson, jackson, gson, and so on.

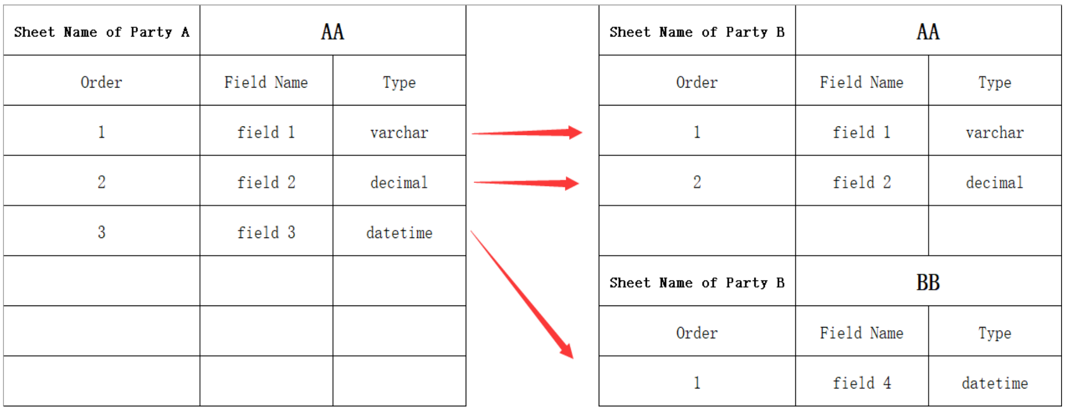
JSON has no error handling for calls and can be hacked when used with untrusted services or untrusted browsers.

2. Main body

2.1 Common situations in data synchronization docking

Due to the different ideas of database designers, it is inevitable that the same type of table or field will be designed in different forms. How to solve the differences between the fields. Taking the common relational database as an example, the following situations may occur:

1) One-to-One: in system A, "field 1", "field 2" and "field 3" exist in the AA table, while in system B, "field 1" and "field 2" are in the "AA" table (different system names will also have different names, and here, the table with similar functions is taken), and "field 3" exists in the "BB" table.



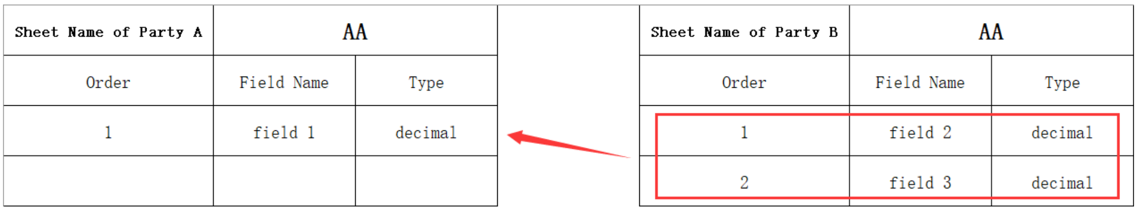
Sheet Name of Party A	AA	
Order	Field Name	Type
1	field 1	varchar
2	field 2	decimal
3	field 3	datetime

Sheet Name of Party B	AA	
Order	Field Name	Type
1	field 1	varchar
2	field 2	decimal

Sheet Name of Party B	BB	
Order	Field Name	Type
1	field 4	datetime

Figure 1. One-to-One situation

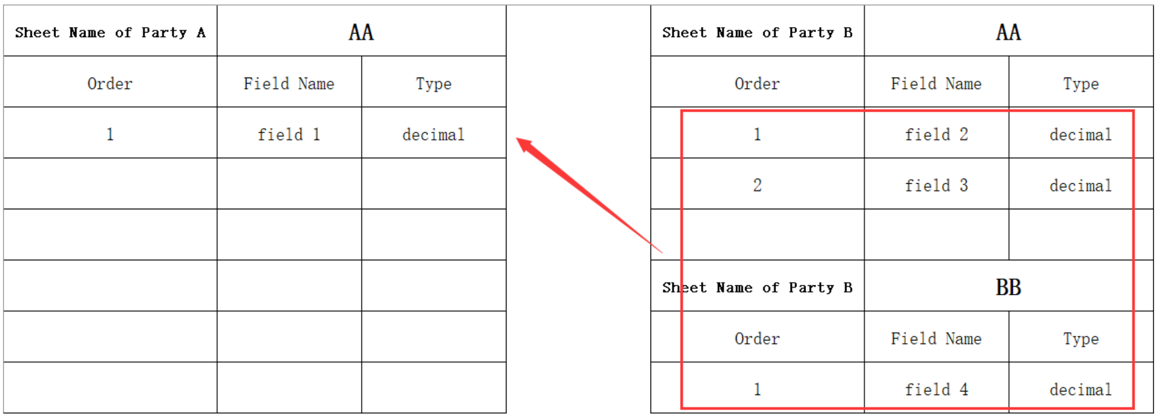
2) One-to-Many: in system A, "field 1" exists in the "AA" table, while in system B, "field two" and "field three" exist in the "BB" table, and "field two" and "field three" can be calculated by the algorithm to get "field one".



Sheet Name of Party A	AA	
Order	Field Name	Type
1	field 1	decimal

Sheet Name of Party B	AA	
Order	Field Name	Type
1	field 2	decimal
2	field 3	decimal

Figure 2. One-to-Many situation1



Sheet Name of Party A	AA	
Order	Field Name	Type
1	field 1	decimal

Sheet Name of Party B	AA	
Order	Field Name	Type
1	field 2	decimal
2	field 3	decimal

Sheet Name of Party B	BB	
Order	Field Name	Type
1	field 4	decimal

Figure 3. One-to-Many situation2

In system A, "field one" exists in the "AA" table, while in system B, "field two" and "field three" exist in the "BB" table, and "field four" exists in the "CC" table, and "field two", "field three" and "field four" can be calculated by the algorithm to arrive at "field one".

3) The "many-to-many" relationship can be derived from "one-to-one" and "one-to-many", and the above are the typical cases summarized in the development. If the fields required by system A do not exist in system B, it is necessary to use algorithm generation and manual entry to complete the synchronization.

2.2 Data synchronization docking scheme

Based on the analysis of the actual problems in the process of data synchronization, the basic idea of the data synchronization scheme is as follows.

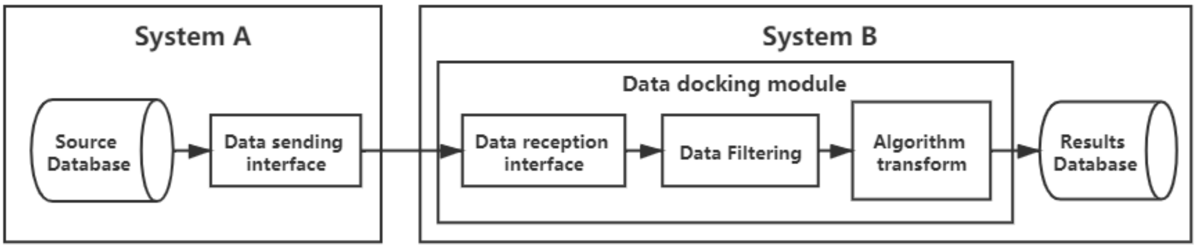


Figure 4. Data synchronization scheme

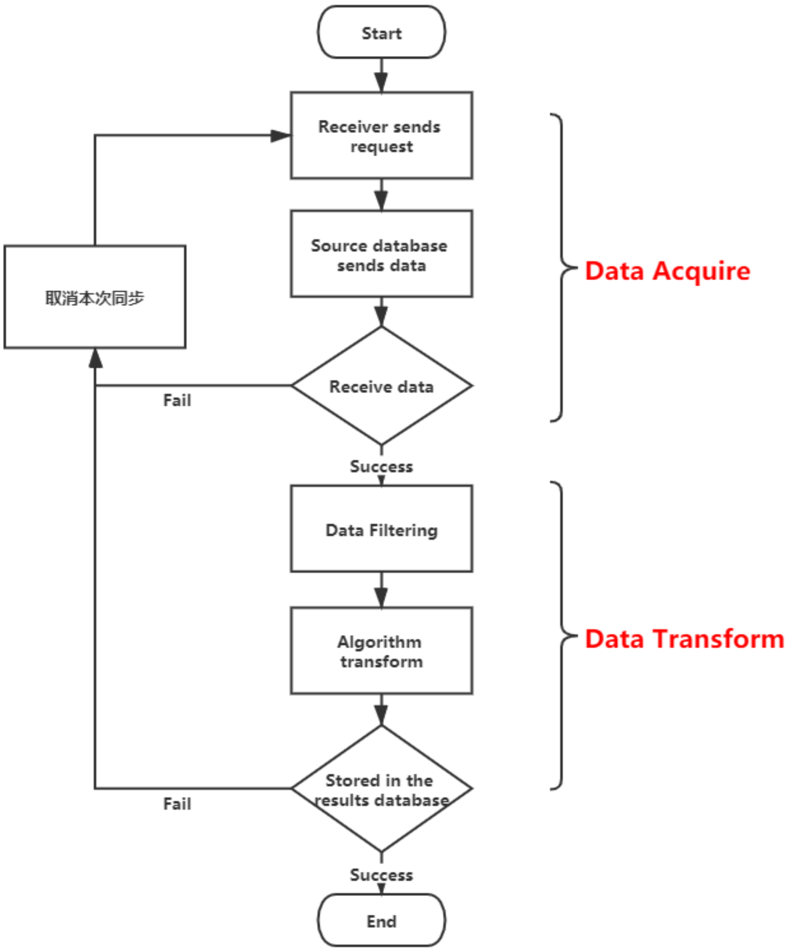


Figure 5. The specific process of data synchronization scheme

Analyze and compare the advantages and disadvantages of three data docking technology solutions, and choose the lightweight JSON technology as the root of the docking solution. A system provides the required data and interface name specification, and system B outputs the corresponding interface to transmit JSON data. After system A obtains JSON data through the interface, the data is cleaned and transformed into a collection of entity objects of system B. As there are missing fields in system B, the entity objects of system B are transformed into entity objects of system A after the algorithm transformation, and then stored in the result database of system A to ensure data consistency.

Based on the above analysis of problems and ideas, the following flow chart gives the specific process of data synchronization scheme.

The algorithm is implemented by Java, and the UML diagram of the algorithm design is given below, taking "Drilling Daily Report" as an example.

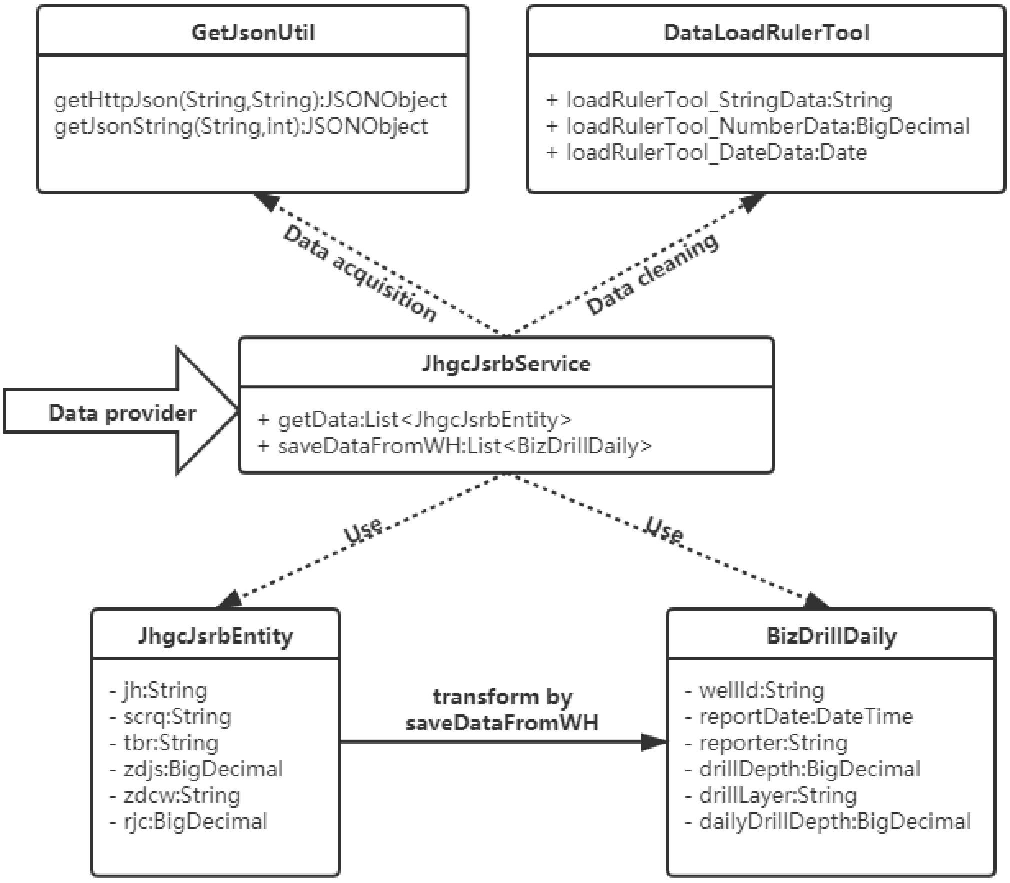


Figure 6. The UML diagram of the algorithm design

The core design of the docking scheme is the class “JhgcJsrbService”. Data acquisition is realized by the method “getData(List<JhgcJsrbEntity>)“. The data cleaning and transformation are carried out by the method “saveDataFromWH(List<BizDrillDaily>)”. Data docking and data transformation are two independent parts, which can quickly find out the problem according to different abnormal conditions.

3. Instance testing

As an example, some of the docking data of a well in an oil field is used to test whether the program is effective. Some of the data from the daily drilling report for this well is shown in table 1.

Table 1. Basic data of drilling daily report of a well in an oil field

jh	scrq	tbr	zdjs	zdcw	rjc
ShunXi 5X	(Null)	Yang Li	0	Fourth Department	0
ShunXi 5X	2020/11/8 0:00	Yang Li	0		0
ShunXi 5X	2020/11/9 0:00	Yang Li	0	Fourth Department	0
ShunXi 5X	2020/11/10 0:00	LiMing Dong	0	(Null)	0
ShunXi 5X	2020/11/11 0:00	LiMing Dong	0	(Null)	0
ShunXi 5X	2020/11/12 0:00	Yang Li	0		0
ShunXi 5X	2020/11/13 0:00	Yang Li	0	Fourth Department	0
ShunXi 5X	2020/11/14 0:00	Yang Li	0	Fourth Department	0
ShunXi 5X	2020/11/15 0:00	(Null)	66	Fourth Department	66
ShunXi 5X	2020/11/16 0:00	Yang Li	66	Fourth Department	0
ShunXi 5X	2020/11/17 0:00	Yang Li	66	Fourth Department	0
ShunXi 5X	2020/11/18 0:00	Yang Li	66	Fourth Department	-66
ShunXi 5X	2020/11/19 0:00	Yang Li	-66		0
ShunXi 5X	2020/11/20 0:00		66	Fourth Department	0
ShunXi 5X	2020/11/21 0:00	Yang Li	66	Fourth Department	0
ShunXi 5X	2020/11/22 0:00	Yang Li	66	Fourth Department	0
ShunXi 5X	2020/11/23 0:00	Yang Li	0		0
ShunXi 5X	2020/11/24 0:00	Yang Li	66	Fourth Department	0
ShunXi 5X	2020/11/25 0:00	Yang Li	66	Fourth Department	0
ShunXi 5X	2020/11/26 0:00	Yang Li	66	Fourth Department	0
ShunXi 5X	2020/11/27 0:00	Yang Li	66	Fourth Department	0
ShunXi 5X	2020/11/28 0:00	(Null)	1259	Kangcun Group	934
ShunXi 5X	2020/11/29 0:00	Yang Li	1503	Kangcun Group	503
ShunXi 5X	2020/11/30 0:00	Yang Li	1503	Kangcun Group	0
ShunXi 5X	2020/12/1 0:00	Yang Li	1529	Kangcun Group	20
ShunXi 5X	2020/12/2 0:00	Yang Li	1979		101
ShunXi 5X	2020/12/3 0:00	Yang Li	2700	Bashkichik group	791
ShunXi 5X	2020/12/4 0:00	Yang Li	3072	Halahatang Group	492
ShunXi 5X	2020/12/5 0:00	Yang Li	3487	Kotur Group	435
ShunXi 5X	2020/12/6 0:00	Yang Li	3706	Kotur Group	243
ShunXi 5X	2020/12/7 0:00	Yang Li	3905	Permian	296
ShunXi 5X	2020/12/8 0:00	Yang Li	4112	Permian	168

Table 2. Basic cleaned data of drilling daily report of a well in an oil field

jh	scrq	tbr	zdjs	zdcw	rjc
ShunXi 5X	1940/1/1 0:00	Yang Li	0	Fourth Department	0
ShunXi 5X	2020/11/8 0:00	Yang Li	0	[---]	0
ShunXi 5X	2020/11/9 0:00	Yang Li	0	Fourth Department	0
ShunXi 5X	2020/11/10 0:00	LiMing Dong	0	[---]	0
ShunXi 5X	2020/11/11 0:00	LiMing Dong	0	[---]	0
ShunXi 5X	2020/11/12 0:00	Yang Li	0	[---]	0
ShunXi 5X	2020/11/13 0:00	Yang Li	0	Fourth Department	0
ShunXi 5X	2020/11/14 0:00	Yang Li	0	Fourth Department	0
ShunXi 5X	2020/11/15 0:00	[---]	66	Fourth Department	66
ShunXi 5X	2020/11/16 0:00	Yang Li	66	Fourth Department	0
ShunXi 5X	2020/11/17 0:00	Yang Li	66	Fourth Department	0
ShunXi 5X	2020/11/18 0:00	Yang Li	66	Fourth Department	0
ShunXi 5X	2020/11/19 0:00	Yang Li	0	[---]	0
ShunXi 5X	2020/11/20 0:00	[---]	66	Fourth Department	0
ShunXi 5X	2020/11/21 0:00	Yang Li	66	Fourth Department	0
ShunXi 5X	2020/11/22 0:00	Yang Li	66	Fourth Department	0
ShunXi 5X	2020/11/23 0:00	Yang Li	0	[---]	0
ShunXi 5X	2020/11/24 0:00	Yang Li	66	Fourth Department	0
ShunXi 5X	2020/11/25 0:00	Yang Li	66	Fourth Department	0
ShunXi 5X	2020/11/26 0:00	Yang Li	66	Fourth Department	0
ShunXi 5X	2020/11/27 0:00	Yang Li	66	Fourth Department	0
ShunXi 5X	2020/11/28 0:00	[---]	1259	Kangcun Group	934
ShunXi 5X	2020/11/29 0:00	Yang Li	1503	Kangcun Group	503
ShunXi 5X	2020/11/30 0:00	Yang Li	1503	Kangcun Group	0
ShunXi 5X	2020/12/1 0:00	Yang Li	1529	Kangcun Group	20
ShunXi 5X	2020/12/2 0:00	Yang Li	1979	[---]	101
ShunXi 5X	2020/12/3 0:00	Yang Li	2700	Bashkichik group	791
ShunXi 5X	2020/12/4 0:00	Yang Li	3072	Halahatang Group	492
ShunXi 5X	2020/12/5 0:00	Yang Li	3487	Kotur Group	435
ShunXi 5X	2020/12/6 0:00	Yang Li	3706	Kotur Group	243
ShunXi 5X	2020/12/7 0:00	Yang Li	3905	Permian	296
ShunXi 5X	2020/12/8 0:00	Yang Li	4112	Permian	168

The algorithm will clean up the illegal data according to the cleaning rules, and then transform it. The rule premise agreed here is: "show the obvious problem data to the user, and leave it to the user to identify." The specific implementation rules are as follows: for irregular field string values, convert them into "[---]" and store them in the achievement database; For irregular date value, it is converted to "1940-01-01 00:00:00"; for irregular value type, it is converted to "0" first, and special problems are encountered before labeling. The final results are shown in table 2:

The above results show that the scheme in data acquisition, data conversion results are in line with expectations.

4. Conclusion

Data synchronization between different application systems is a common practical problem in development. In the process of data docking, the accuracy and real-time of data need reliable implementation scheme and code logic to ensure. Before the system is officially launched, it must be ensured that it has undergone rigorous functional testing. According to the actual situation of different systems, docking rules suitable for the system can be formulated.

The data synchronization scheme proposed in this paper can effectively realize the data docking between two application systems. Based on the principle of aspect oriented programming, the code is loosely coupled and has strong scalability. The docking system module developed based on this scheme has been applied to the "smart well site" system. In the follow-up work, we consider using microservice architecture to form a data synchronization service module to cope with the data synchronization docking in the case of multiple data sources in the future.

Acknowledgments

This research is supported by five projects: The demonstration construction projects of “Ideological and Political Course and Ideological and Political Course” in Yangtze University in 2021 (No.41); The project of young people in the Education Hall of Hubei (No.Q20161311); The Yangtze Youth Fund (No.2015cqn53); The Yangtze University Students' Innovation and Entrepreneurship Training Program Project (No. Yz2020157,Yz2020150).

References

[1] Liu Shu, Xu Shuo, Huang Qiquan, sun Lu, Ding ran, Wang Lihua. Research and implementation of a trigger based data synchronization system [J]. Computer applications and software, 2012, 29(12): 189-191 + 200.

[2] Chen zuoxia, Zhang Kai, Li Zhenkun. Design and implementation of data synchronization in remote disaster recovery system and data warehouse [J]. Computer application research, 2007(05):229-230+233.

[3] Liu Zhen, Zhou Yunyi. Research and improvement of data synchronization protocol [j]. Computer application and software, 2009,26 (02): 235-237 + 253.

[4] Zhang Yueqin, yuanxinkun. Research and implementation of a heterogeneous database synchronization technology [j]. Microcomputer information, 2008 (33): 182-184.

[5] Zhang Zhenyou, Sun Yan. Research and implementation of distributed heterogeneous database synchronization technology [J]. Computer knowledge and technology, 2015,11 (28): 16-19.

[6] Kou Yuanyuan, Wang Xiaoming, Qiao Wenwen. Research and application of database synchronization technology [J]. Computer knowledge and technology, 2019,15 (15): 3-5 + 34.

[7] Chen Shufang, Li Juan, Liu Limei, Xue Qing. Research on JSON-based special equipment heterogeneous data sharing and exchange method[J]. Information Technology and Informatization,2020(12):20-21+24.

[8] Zhao Wei, Deng Xiawei, Fu Ruiling, Jiang Jiarui. A data transmission method of network information security system based on JSON format [J]. Journal of Communication University of China (NATURAL SCIENCE EDITION), 2020,27 (03): 22-28.