Discovering the Long-Term Drug Side-Effects from Big Medical Data using NoSQL Shard Schemes

Chin-Ho Lin[1], Ajit Kumar[2], Han-Fang Cheng[3], Liang-Cheng Huang[1], Seng-Cho T. Chou[1], I-Jen Chiang[2,3]
[1]Department of Information Management, National Taiwan University, Taipei 106, Taiwan
[2]Graduate Institute of Biomedical Informatics, Taipei Medical University, Taipei 110, Taiwan
[3]Institute of Biomedical Engineering, National Taiwan University, Taipei 100, Taiwan

***Abstract*—Relational data models are no longer capable in collecting and analyzing big data efficiently. This paper proposes an approach for rapid processing, analyzing, and using big medical data that can help in promoting public health. The approach and results include: 1) designing a patient-oriented medical document format and data restructuring methods for using NoSQL database, which can substantially enhance the efficiency of medical record retrieval; 2) providing additional temporal information on certain events for enabling researchers to easily perform temporal analysis; 3) using sharding-keys for data partitioning to provide more efficient data cubes; and 4) combining shard and MapReduce to facilitate high-performance targeted queries and searches, especially in filtering and analyzing the temporal sequence of events.**

Keywords—Big medical data; NoSQL; Temporal event analysis; Shard

I. INTRODUCTION

Medical records are maintained for supporting various tasks, such as patient-care assessment and decision-making, legitimate medical practice reports, healthcare institution management and planning, and clinician and nurse education. In addition, medical records are the primary sources for studies regarding the enhancement of medical care and people's health [1], [2]. The speedy growth of scientific discipline and medical technology has offered efficient and rapid verification methods for detecting, preventing, and treating diseases. This has caused 1) the rapid accumulation in the number of medical records (Velocity); 2) the increase of medical evaluation factors, such as items investigated during laboratory, biochemical, and genetic tests (Volume); and 3) the generation of diverse data forms, such as texts, numerals, diagrams, tables, images, and handwritten documents (Variety). Because of high velocity, volume, and the variety of data, processing and managing has become difficult causing more response time and costs. As an example, the National Health Insurance (NHI) service is available in Taiwan [3]. The number of people insured under NHI is approximately 23 million that generate millions of medical records daily. These data include clinical medical records, medical expenses, blood test reports, diagnosis summaries, and medication records. A considerable amount of data has been accumulated after the compilation of nineteen years medical records. The use of traditional relational databases saves storage space, avoid data redundancy, and support complex table structures that enable users to engage in various types of queries [4]. However, the medical data are rapidly increasing. As previous methods adopted for data efficiency and consistency, such as categorizing medical records in different tables or databases, are no longer efficiently capable of collecting and analyzing data [5]–[9], it creates a bottleneck in system performance and hinders the process of data integration.

This study proposes the use of NoSQL in Big Medical Data [8]–[13]. The approach includes: 1) designing a patient-oriented information format that incorporates all the medical data of a patient into a series of patient-oriented medical documents and stores them in NoSQL databases, which can substantially enhance the efficiency of medical record retrieval; 2) restructuring the relational data model to transform data into key-value or document-oriented data formats of the NoSQL; 3) providing additional temporal information on certain events for enabling researchers to easily perform temporal analysis; 4) using sharding-keys for data partitioning to provide more efficient data cubes; and 5) combining shard and MapReduce [14] to facilitate high-performance targeted queries and searches. In this study, we used MongoDB [15], [16] as a test platform and obtained the population database from Taiwanese National Health Insurance Research Database (NHIRD) [17]. A total of 1,175,186,752 medical records, collected from 1996 to 2010, was used in this study. The data were successfully imported into the test platform, and all the test requirement's features were taken into the account. Finally, we used a case study – acute pancreatitis adverse reactions can easily develop in patients with diabetes who used drugs containing Sitagliptin – was used as an example to demonstrate the outstanding performance of the system in filtering and analyzing the temporal sequence of events.

II. METHODS

The overall system architecture for our research is shown as Fig. 1. The proposed method involves a patient-oriented big medical data restructuring and transformation, which can be used for clinical, research, and management purposes. All the medical data of one patient were compiled into a single document called Patient-Oriented Medical Document (PoMeDoc), and only a single copy was created for each patient. The basic constituent elements of the PoMeDocs are paired keys and values, such as the month as key and June as value. The key is used to identify the data; therefore, within a PoMeDoc, a key must be unique. By contrast, the value is

arbitrary data. A key-value pair of one PoMeDoc can be the key-value pair of another PoMeDoc, which can form a PoMeDoc tree. The shape and structure of a PoMeDoc tree comprise the height (instant accessibility) and horizontal growth (flexible scalability) of a tree; these characteristics are suitable for big medical data applications. PoMeDocs also exhibits the advantages shown when using NoSQL in big data

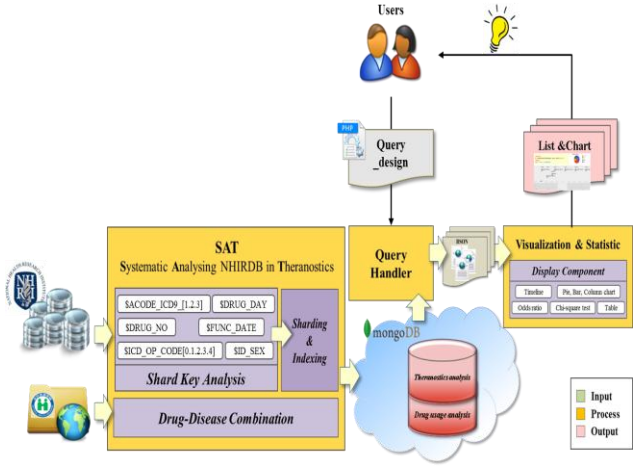

Fig. 1. System architecture

[6]–[8], [11]. For example, as all data are compiled into one PoMeDoc, clinical physicians can easily view the complete medical records of patients. Moreover, medical institutions can easily and efficiently manage and exchange patient medical records in a schema-less manner. As data search is conducted on a single patient-specific medical record called PoMeDoc, it enables researchers to effectively undertake complex searches and selection and temporal analyses. In addition, to establish different logical views (such as a shard for diabetes) on a PoMeDoc, researchers can focus on a small scope and reduce the search time. In practice, converting data format can be easily achieved by importing PoMeDoc into the NoSQL platforms. In the following section, the approach used for restructuring the data into a PoMedoc is described. In addition, data transformation, importing data to NoSQL platform, shading and using MapReduce, and conducting targeted queries and searches is explained.

A. Data Restructuring

Medical records, originally stored as the relational database, need to be restructured into PoMeDocs. This can be achieved by understanding the workflow of clinical settings. Fig. 2 illustrates the various tables and relationships generated from medical-related operations, including outpatient department, inpatient department, and contracted pharmacies. Primary Keys (PKs) and Foreign Keys (FKs) are used to Connect the

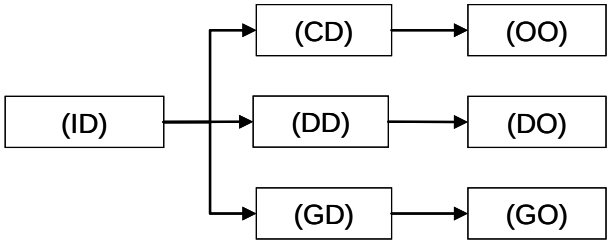

Fig. 2. Illustration of various tables and relationships, where ID – registry for beneficiaries; CD – ambulatory care expenditures by visits; DD – inpatient expenditures by admissions; GD – expenditures for prescriptions dispensed at contracted pharmacies; OO – details of ambulatory care orders; DO – details of inpatient orders; and GO – details of prescriptions dispensed at contracted pharmacies.

tables. These tables were used to reconstruct a patient's PoMeDoc. First, basic data were obtained from the ID table. Subsequently, upon tracing the relationships between the PKs and FKs, it yielded the marked data from the connecting tables, and recursive procedures were used to complete all the patients' PoMeDocs. In practice, redundant keys can be deleted during restructuring to save space.

B. Data Transformation

Occasionally, the data that clinical physicians or researchers are interested in cannot be obtained directly from the existing data. By analyzing previous NHIRD-related studies, we found that the information, such as the age and year at which patient sought medical consultation, patient's place of residence, categories of prescribed drugs, days of drug use before the next consultation, total days of drug use, duration of medical history, and occurrence time of various diseases, can be obtained by additional processing of data. However, when the data is huge, the additional processing may take a long time. To overcome the long processing time problem, we define new fields (Table I), dynamically collect the known data, and

TABLE I. NEWLY ADDED DATA ITEMS AND GENERATING METHODS

Newly added key fields	Computing methods	Property
FUNC_AGE	*FUNC_DATE — ID_BIRTHDAY*	*Numerical*
FUNC_YEAR	*FUNC_DATE（YYYY）*	*Time*
Residential_Area	*AREA_NO_I*	*Category*
Drug_Category	*Searching classification table*	*Category*
Drug_Use_Day	*Total_QTY / Drug_Use × Drug_Fre*	*Numerical*
Total_Drug_Use_Day	*SUM（Drug_Use_Day）*	*Numerical*
Disease_History	*CURRENT_DATE — FIRST_ FUNC_DATE*	*Time*
Interval_Diseases	*FIRST_ FUNC_DATE_D1 — FIRST_ FUNC_DATE_D2*	*Time*

perform additional computation and organization, which would enable a researcher to obtain the data without performing recalculations. Therefore, we transformed PoMeDocs into readily available fields.

In this data restructuring (as mentioned in A. Data Restructuring) and transformation of data, the massive medical

records were imported into the NoSQL BSON (binary-encoded serialization of JSON-like documents) data format [18].

C. *Data Sharding and Using MapReduce for Targeted Queries and Searches*

A PoMeDoc is a high dimensional entity, which can be categorized based on certain dimensions or views using sharding. Sharding is a horizontal partitioning method [19], [20], which is highly scalable and can enhance search performance. When the data are divided into multiple small shards, simple searches can be conducted, and multiple shards can be combined with MapReduce for concurrent and parallel computing to enhance searching speed.

Key-value is the basic element of PoMeDocs and is suitable for key-based sharding [19] during data partition processing. One or more key fields can comprise a sharding-key, and its value can be used as a basis for data partitioning. The value can be a single value or range of values. For example, the data for the time, when medical consultation was sought, was stored in this format – yyyy-mm-dd. The specific year (yyyy), month (mm), or year and month (yyyy-mm) can be used to shard the data. In addition, the time interval can be used to shard the data, such as an interval of a three-month period – January to March, April to June, July to September, or October to December. However, only a meaningful shard can enhance search performance. Thus, to select the appropriate sharding-key and its value, the characteristics of the data and the purpose of the search must be considered. For example, when a disease code is used as a sharding-key, the key's value is determined based on a code arrangement approach. In this study, the proposed sharding-keys and their values are shown on Table II.

Invoking MapReduce with sharding-keys for conducting targeted searches can improve retrieval and computation performance. When users are searching for data, the search criteria are mapped to the corresponding sharding-keys (Map). Subsequently, searches are conducted concurrently on the corresponding shards. Finally, all the search results are compiled (Reduce). In Fig. 3, for example, the field sex is used as a sharding-key for sharding, and data are divided into two shards – Sex = 'M' and Sex = 'F.' The corresponding shards can be used for searches and calculations to check the number of patients based on sex.

TABLE II. THE PROPOSED SHARDING-KEYS AND VALUES

Sharding-keys	Values
FUNC_AGE	[0,10], [11,20], [21,30], [31,40], [41,50], [51,60], [61,70], [71,80], [81,90], [91,100], [101, ∞]
SEX	M, F
Residential_Area	E, W, S, N
FUNC_YEAR	yyyy
Diseases	ICD9 code
Drug_Category	Metformin, DDP-4, SU, ...
Total_Drug_Use_Day	# Months
Disease_History	# Years
Interval_Diseases	# Months

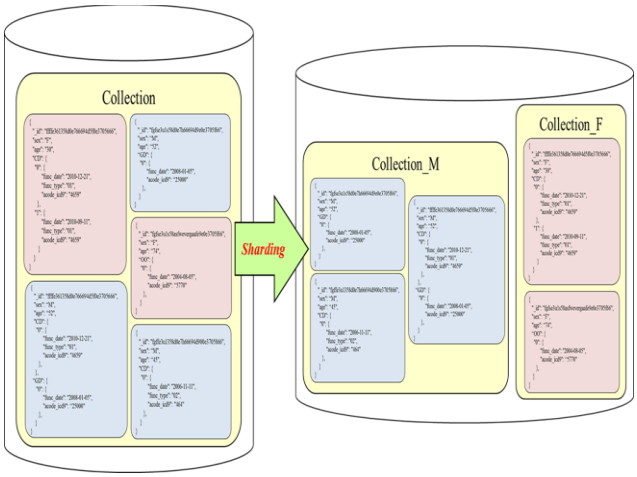

Fig. 3. The data were divided into two shards according to sex.

III. EXPERIMENTS

The materials used in this study were obtained from Taiwanese NHIRD, which contain the medical data (from 1996 to 2010) of one million people randomly sampled from the NHIRD Beneficiary Registry, 2010. These materials were combined, which yielded 1,175,186,752 medical records. The data are divided into seven files, each of which is connected with one another by PKs and FKs. These files included outpatient prescription treatment (CD), outpatient prescription order (OO), hospitalization expenses (DD), hospitalization expenses and physician order (DO), contracted pharmacy prescriptions and dispensary (GD), contracted pharmacy prescription orders (GO), and registry for beneficiaries (ID), as shown in Fig.2. As these data are available for only research purposes, users must submit applications to obtain and use the data legally. The basic patient records in the registry data do not include patient names, and only the residential area is present as the address of the patient. In addition, encrypted ID numbers are used as a primary key for linking with other detailed data.

The laboratory equipment included a Wintel based server with a 3.10GHz CPU, 16 GB RAM, 3 TB storage, and Windows 7 SP1. MongoDB 2.2.0 was used as the NoSQL database platform and Apache 2.2.x, PHP 5.3.8, and Chrome 27.x were used as development tools. Design and production programs included data restructure and transformation, BSON-format PoMeDocs, data import, search and MapReduce, user interfaces (UI), and web representations. The web query function is divided into three parts: 1) simple search – users can input a patient's ID, the International Classification of Diseases (ICD-9) or disease name, drug code, drug compositions, product name, or operation code or title to search for the proportion and distribution of people that fulfill the criteria; 2) Advanced search – combined queries can be conducted by using a diagnosis code, drug code, and an operation code to obtain population distributions, statistics regarding comorbidity, the time interval between two different search conditions, medication types, and statistical analysis; 3) Code search – the corresponding description data of disease diagnosis codes, drug codes, and operation codes can be obtained. In addition, a list of patients fulfilling the search criteria and their medical records can also be obtained.

IV. RESULTS AND DISCUSSION

All the data were successfully imported into MongoDB. Diagnostic codes, drug codes, and operation codes were used as the sharding-keys to establish the corresponding shards. Every related function passed the tests. To evaluate the system performance, patients diagnosed with diabetes were the query targets and three approaches were used to 1) directly search without sharding, 2) conduct searches based on system-defined shards, and 3) perform searches based on user-defined shards. Although the three search results were similar (88,601 people out of 1 million people were diagnosed with diabetes), significant differences were shown in the performance efficiency. The search duration was estimated to be 791.969s, 50.142s, and 20.466 s, respectively. The results indicated that the search performance can be significantly enhanced through sharding. Specifically, user-defined shards can enhance performance by approximately 40-fold. Although using system-defined shards do not necessitate defining the key's values, using the values set by users can greatly enhance the performance. Based on experience and analyses of NHIRD-related research projects, diseases, drugs, and treatment are frequently explored items by physicians and researchers. These three categories should be appropriate as sharding-keys. Nevertheless, determining appropriate key's values requires detailed analysis in order to maximize the effectiveness of shards.

The performance of the system in the screening and analyzing temporal sequence of events was a primary concern of this research. In this study, scenarios were used to conduct tests. The clinicians and researchers observed that the number of diabetic patients hospitalized for acute pancreatitis is increasing; therefore, three search criteria were set – hospital records, population with diabetes, and hospitalizations because of acute pancreatitis. As events occurred in sequence, a fourth criterion was added – the time on which the patient was first diagnosed with diabetes must be earlier than the time that the patient was hospitalized for acute pancreatitis. After conducting the search, the number of patients who met these criteria in the last six months was determined to have increased by approximately 90% compared to that from over the past three years, as shown in Fig. 4. Subsequently, the inspected system provided a variety of figures and tables, such as population distribution (Fig. 5) and the drug category statistics (Fig. 6). These tables and figures indicated that Januvia was a new drug that was covered under the National Health Insurance starting in 2009 (the experimental data was from 2010). Subsequently, a fifth criterion (Januvia) was added; the search results indicated that an increasing number of people has been using this drug recently (Fig. 7). The final search results are shown in Fig. 8, revealing a 1.626 odds ratio, which indicated that diabetic patients who are taking Januvia are 1.6 times more likely to develop acute pancreatitis than those who do not take the drug. This search test was conducted based on the drug safety information announced by the U.S. Food and Drug Administration (FDA) on September 25, 2009 [21]. The verification results corresponded to the warnings issued by the FDA, and the entire testing process exhibited the excellent performance of the system.

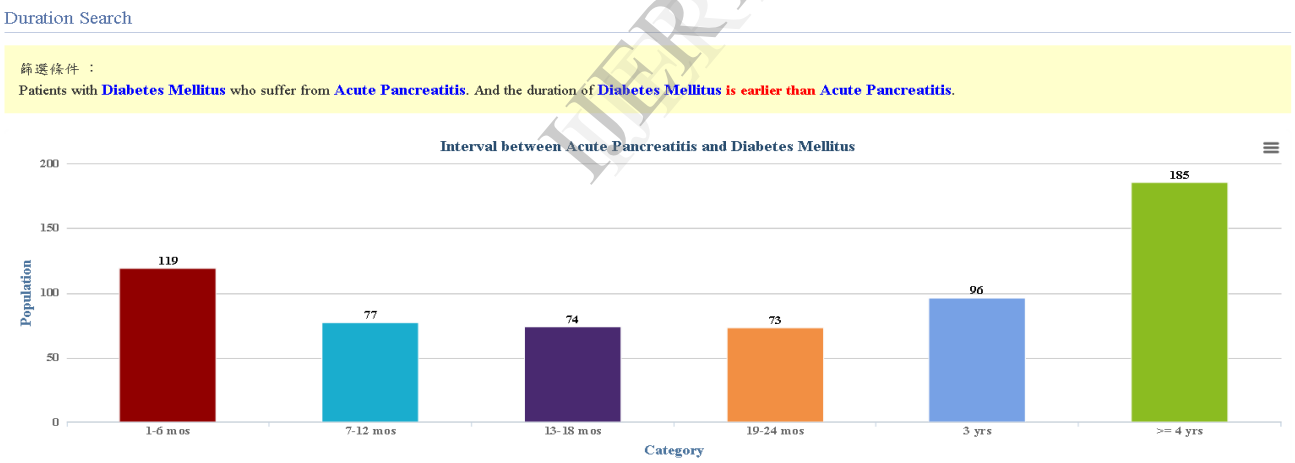

Fig. 4. The data were divided into two shards according to sex.

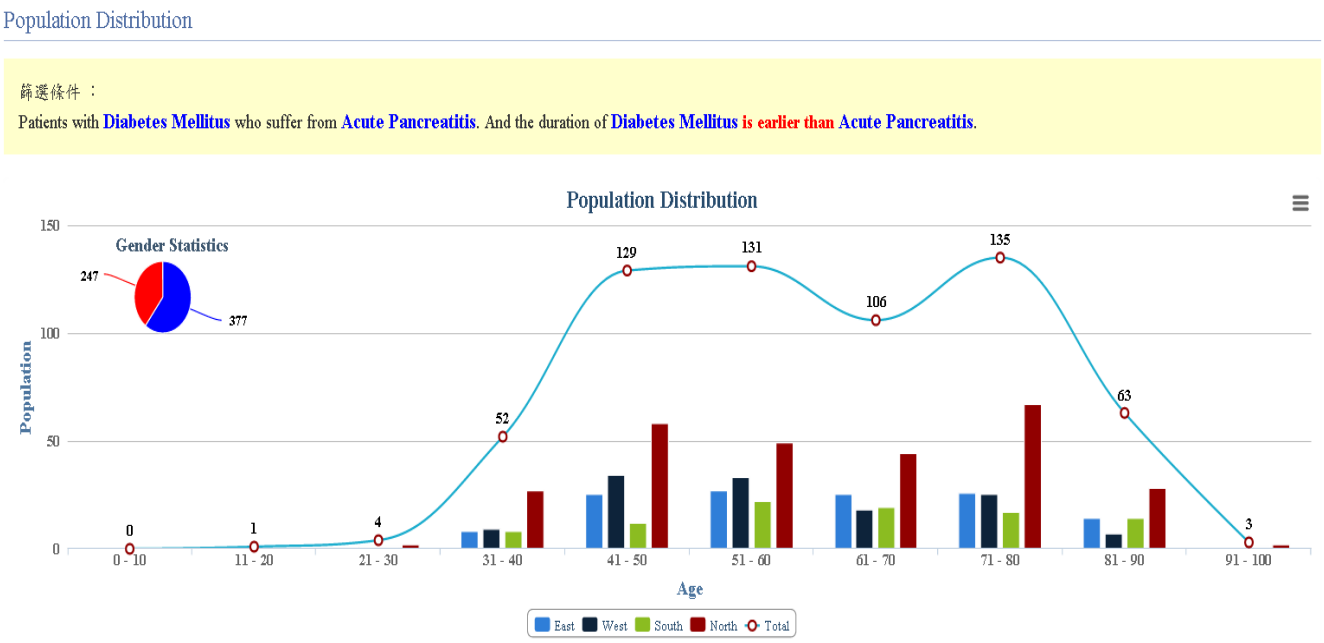

Fig. 5. The population distribution of diabetic patients hospitalized for acute pancreatitis.

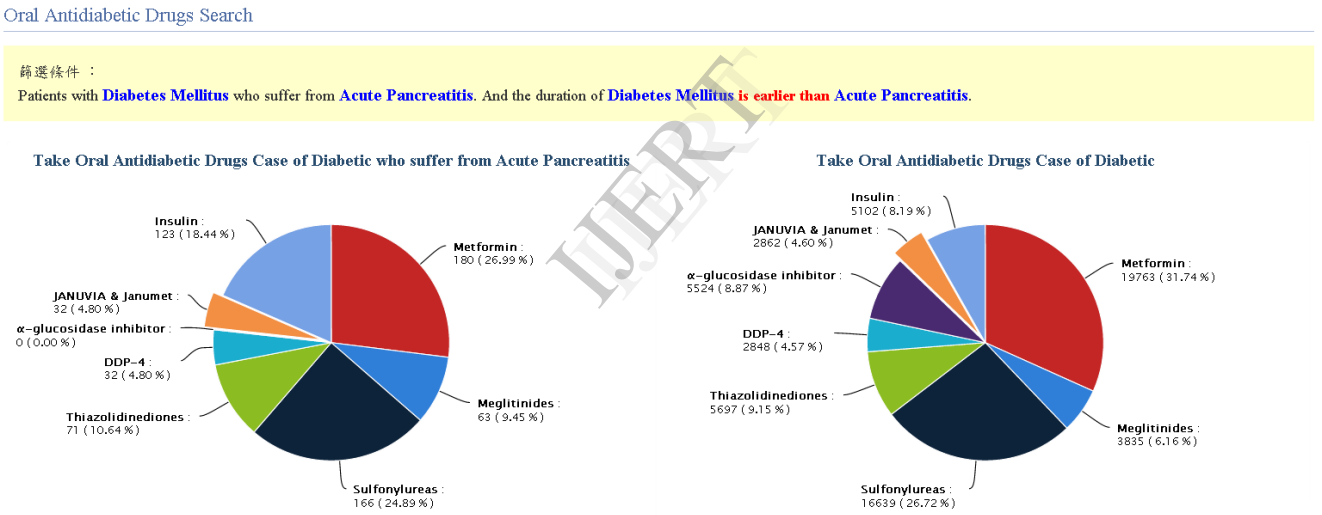

Fig. 6. Drug statistics.

Period of Medication Search

篩選條件 ：
Patients with **Diabetes Mellitus** who suffer from **Acute Pancreatitis**. And the duration of **Diabetes Mellitus** **is earlier than** **Acute Pancreatitis**.

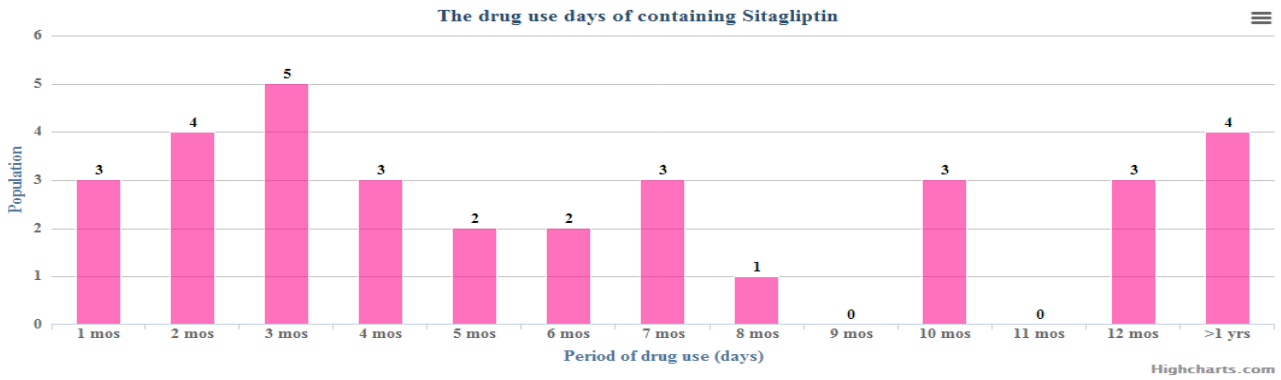

Fig. 7. Days of drug use.

Statistical Analysis

篩選條件 ：
Patients with **Diabetes Mellitus** who suffer from **Acute Pancreatitis**. And the duration of **Diabetes Mellitus** **is earlier than** **Acute Pancreatitis**.

統計檢定：

$\chi^2 = 7.24$

$\alpha = 0.05$ ， $df = 2$ ，$\chi^2_\alpha = 3.841$

H_0 = Diabetic 服用含有 Sitagliptin 成分藥物對於產生 Acute Pancreatitis 沒有影響

H_1 = Diabetic 服用含有 Sitagliptin 成分藥物對於產生 Acute Pancreatitis 有影響

$\chi^2 > \chi^2_\alpha$ ， 拒絕 H_0 ，接受 H_1 ，Diabetic 服用含有 Sitagliptin 成分藥物對於產生 Acute Pancreaftitis 可能會有影響

Odds Ratio ≒ 1.63

即 Diabetic 服用含有 Sitagliptin 成分藥物罹患 Acute Pancreatitis 的勝算是未服用的 1.63 倍

Diabetes Mellitus		Acute Pancreatitis	
		+	-
含 Sitagliptin 成分藥物	+	32	2830
	-	592	85147

Fig. 8. Statistical analysis results.

V. CONCLUSION

Even though NoSQL has powerful expansion capabilities and flexibilities that made cloud databases abandon relational database architecture, it neither has a theoretical basis as relational database, nor universal data modeling technology. The patent-oriented and schema-less methods for data restructure, which this paper proposes, can overcome the differences in schema designs many hospitals have. It also facilitates changing existing schema fields to add new ones. PoMeDoc has key-value and document-oriented integrated data types. It is very suitable for hospital data and can be converted into various NoSQL data formats. In addition to simplifying the complication of platform integration, platform limitations can be avoided, as well. Through data transformation, the additional processed items make hospital records more valuable and help in diagnosis decision-making and research. Regarding the analysis of temporal event and complex data searching and filtering performed in the interacting progress, the powerful effect of the combination of shard and MapReduce is shown in the sample experiments. Through this method, big medical data can be quickly processed, preserved, retrieved, analyzed and used to aid in constructing a medical information environment with medical treatment, research and management as main purposes.

REFERENCES

[1] M.A. Musen and J.H. Bemmel, Handbook of Medical Informatics, Houten: Bohn Stafleu Van Loghum, 1999.

[2] M. Porta and J. M. Last, A Dictionary of Epidemiology, New York: Oxford University Press, 2008.

[3] National Health Insurance Administration, Available: http://www.nhi.gov.tw/english/index.aspx

[4] E. F. Codd, "A relational model of data for large shared data banks," Commun. ACM, vol. 13(6), pp. 377–387, 1970.

[5] N. Jatana, S. Puri, M. Ahuja, I. Kathuria, D. Gosain, "A survey and comparison of relational and non-relational database," Int. J. Eng. Res. Tech., vol. 1(6), Aug. 2012

[6] A. B. M. Moniruzzaman and S. A. Hossain, "NoSQL Database: New era of database for big data analytics – clasification, characteristics and comparison," Int. j. database theor. app., vol. 6, no. 4, pp. 1–14, 2013.

[7] R. Cattell, "Scalable SQL and NoSQL data stores," ACM SIGMOD Record, vol. 39(4), pp. 12–27, Dec. 2010.

[8] M. Stonebraker, "SQL databases v. NoSQL databases," Commun. ACM, vol. 53(4), pp. 10–11, April 2010.

[9] S. Madden, "From databases to big data," IEEE Internet Computing, vol. 16(3), pp. 4–6, 2012.

[10] B. G. Tudorica, C. Bucur, "A comparison between several NoSQL databases with comments and notes," In: RoEduNet '11, pp. 1–5, 2011

[11] J. Han, E. Haihong, L. Guan, J. Du, "A survey on NoSQL databases," Int. Conf. on Pervas. Comput. and Appl. (ICPCA), IEEE Press, Oct. 2011, pp.363–366, doi:10.1109/ICPCA.2011.6106531

[12] I. Lungu, B. G. Tudorica, "The development of a benchmark tool for NoSQL databases," Database Syst. J., vol. 4, no. 2, pp. 13–20, 2013.

[13] J. Pokorny, "NoSQL databases: a step to database scalability in web environment," Int. J. Web Inform. Syst., vol. 9, no. 1, pp. 69–82, 2013.

[14] J. Dean, S. Ghemawat, "MapReduce: simplified data processing on large clusters," Commun. ACM, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[15] 10gen. MongoDB, Available: http://www.mongodb.org

[16] M. Ward, NoSQL database in the cloud: MongoDB on AWS, Amazon Web Services, 2013.

[17] National Health Insurance Research Database, Available: http://nhird.nhri.org.tw/en/index.htm

[18] BSON, Available: http://bsonspec.org

[19] A. Pavlo, C. Curino, S. Zdonik, "Skew-aware automatic database partitioning in shared-nothing, parallel OLTP systems," ACM SIGMOD, pp. 61–72, May 2012.

[20] Y. Liu, Y. Wang, Y. Jin, "Research on The Improvement of MongoDB Auto-Sharding in Cloud Environment," IEEE ICCSE, pp. 851–854, July 2012.

[21] FDA Januvia Tablet, Available: http://www.fda.gov/Safety/MedWatch/SafetyInformation/Safety-RelatedDrugLabelingChanges/ucm121926.htm