Int. J. Advance Soft Compu. Appl, Vol. 7, No. 1, March 2015
ISSN 2074-8523

A Selection Method of Database System in Bigdata Environment: A Case Study From Smart Education Service in Korea

Jong Sung Hwang[1], Sangwon Lee[2], Yeonwoo Lee[1], and Sungbum Park[1]*

[1]National Information & Society Agency, Big Data Strategy Center
e-mail: jshwang@nia.or.kr, leeyw@nia.or.kr, parksb@nia.or.kr
[2]Wonkwang University
Division of Information and Electronic Commerce (Institute of Convergence and Creativity)
e-mail: sangwonlee@wku.ac.kr
*corresponding author

Abstract

Relational databases have been widely used in organizations for decades. Although relational databases have significant advantages for data preservation and concurrency controlling as the standard model in databases, there are disadvantages as well, such as object relationship inconsistency and heavy dependence on strict database schema. Recently, NoSQL became the alternative for big data utilization in that it can be operated without a specific schema while works well in cluster environment in the 21st century. However, its utilization still has pros and cons. In this backdrop, we do not only enumerate the advantages and disadvantages of NoSQL on relational databases, but also show the comparison between the NoSQL databases. More importantly, this research does not necessarily recommend the specific single type database to build an information system. Instead, using the polyglot persistence concept, we suggest the database management system selection criteria. Then we present an optimal combination of the database system in order to demonstrate our suggestion from the example of the smart education service in Korea.

Keywords: *Database Selection, Big Data, Polyglot Persistence.*

1 Introduction

Currently, firms create a large amount of data in their business processes and manage them using a database. However, in the 21st century, it is impossible for traditional relational databases (RDBs) which have been widely used for decades to store and

analyze such a large amount of data due to the cost and performance issues. Accordingly, organizations that manage large amounts of unstructured data are turning to non-relational database now frequently called Nosql database [4]. In this respect, we define the key elements of NoSQL database and its critical factors required for the application of the big data area. Then, after introducing various types of big data DBs, we propose the database selection criteria by inclusively considering the merits of existing database. In fact, Nosql became an alternative for big data utilization not only because it can be operated without specific schema, but also it works well in cluster environment in the 21st century. However, even though relational databases have disadvantages such as object relationship inconsistency and heavy dependence on DB schema, there are advantages as well, such as their significant ability for data preservation and concurrency controlling as a de facto standard model in databases. Against this backdrop, in this paper, we do not necessarily recommend the use of single type database in order to build a big data based information system. Especially, using the polyglot persistence concept as a basis, we exemplify the optimal DB management system (DBMS) and present its combination from specific areas of the information system such as smart education system. The example presented in this paper enables organizations to use data storage suitable to their business circumstances.

2 Changes in Database Model

In this paper, we investigate the various database management models, and find technical changes which cannot be resolved properly with the relational database model used for decades.

2.1 Issues on Relational Database

Databases provide more outstanding flexibility than file systems when storing a large volume of data using relational databases. We can find and collect the necessary information in many application programs. As another important merit of relational database, enterprise applications have many users. It generates errors and warnings when end users operate same function in the same application concurrently and simultaneously. Relational databases help to manage these problems using concurrency control functions. Even though there's complexity in the concurrency control, it gained acceptance with the efficiency in the transaction mechanism. Finally, in an ecological system where many applications exist and collaborate mutually, relational databases successfully store many applications in the single database using the integrated DB sharing method. In this wise, relational database systems have generally been accepted by the corporations if the data didn't contain many relationships requiring joins of large tables [5].

However, relational databases have significant disadvantages: inconsistency of object relationships between the relational model and the data structure within single user interface. Another defect is the fact that relational database cannot efficiently manage

a large volume of unstructured data which should be supported in clusters, while it usually deals with structured data within the server.

2.2 Advent of Cloud Computing and Internet of Things (IoT) Systems

In the past, data were usually defined in advance, processed and then stored in the single database. Current IT technology is evolving into the diffusion of the cloud computing and internet of things system (Fig 1), which commercializes low-cost computing devices by interconnecting them. On the contrary, massive structured/unstructured data sources are generated from various devices and are stored in the clusters. They are social media data, streaming data, Open Graph (OG) data, and internet of things (IOT) devices data [1]. Therefore, it requires that the data model that support complex and various structures. Recently there has been interest in data stores that do not use a relational database and its manipulation language 'SQL' exclusively. It is so called Nosql movement. Examples are Google's BigTable and Facebook's Cassandra [5].

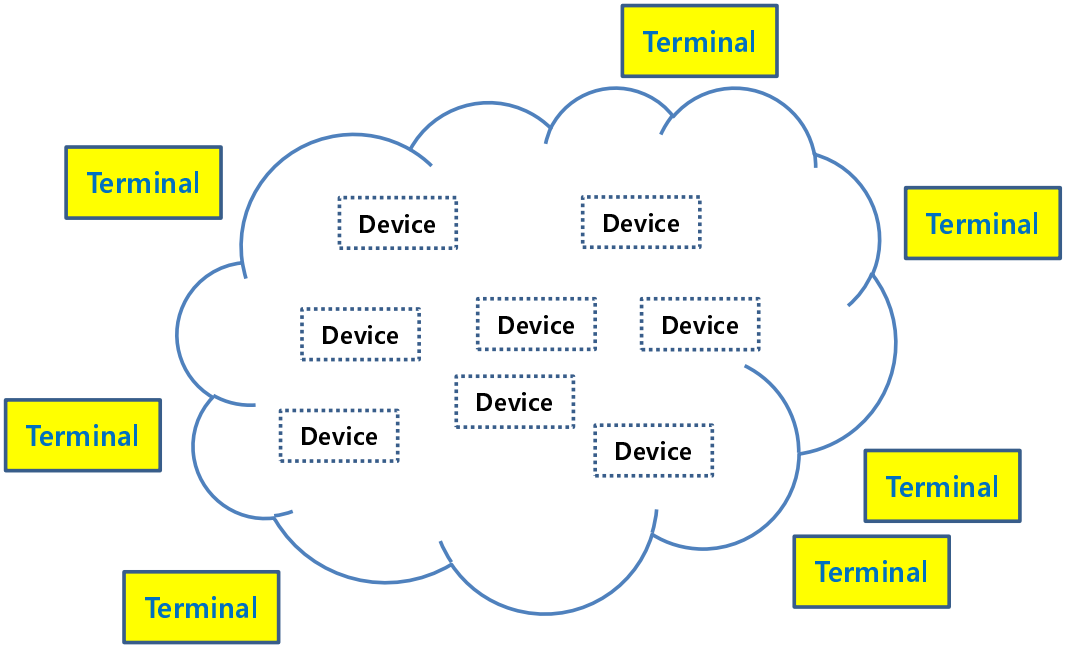

Fig. 1 Cloud Computing and IoT Environment

2.3 Nosql Era

The Nosql database provides a mechanism that employs less constrained consistency models than relational database to collect and retrieve a large amount of data. Nosql is often highly optimized key-value stores. Nosql databases are used in the various fields of traditional and emerging industries using bigdata based real-time web

applications. The purpose of Nosql is simple and easy retrieval and appending additional data processing operations. This feature supports such technical motivations as simplicity of design, horizontal scalability, and finer control over availability. Fig 2 illustates aforementioned Nosql's advantageous features on relational databases in the bigdata utilization environment.

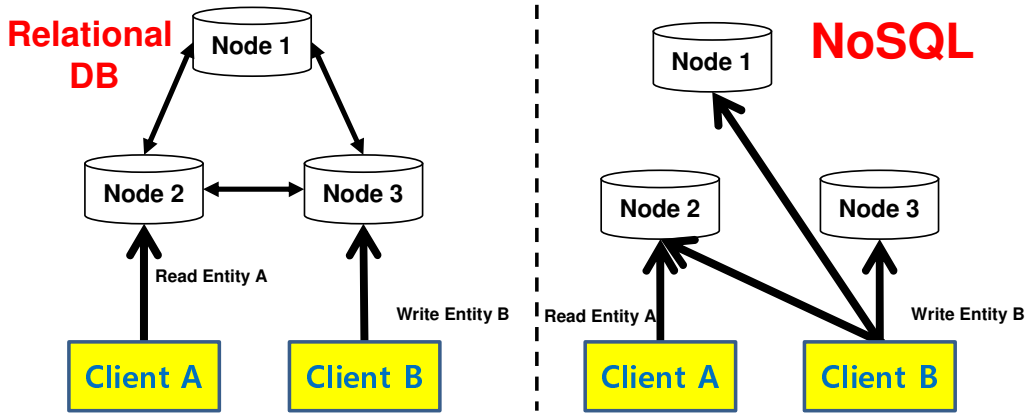

Fig. 2 Relational DB vs. NoSQL DB

Relational database horizontally distributes data load (see fig 2). This is where NoSQL solutions differ greatly. They do not distribute a logical entity across multiple tables, it's always stored in one place. They do not enforce referential integrity between these logical entities. This is what allows them to automatically distribute data across a large number of database nodes and also write them independently. If a certain user were to write entity B to a database cluster with 3 nodes, chances are he/she can evenly spread the writes across all of them. A distributed RDBMS' cases, Client 1 will either not see any changes until all three nodes acknowledged a two phase commit or will be blocked until that happened. In addition to that synchronization the RDBMS also needs to read data from other nodes in order to ensure referential integrity, all that happens during the transaction and blocks Client B. NoSQL solutions do no such thing for the most part. The fact that such a solution can scale horizontally also means that it can leverage its distributed nature for high availability. This is very important in the cloud, where every single node might fail at any moment. So it is true that NoSQL solutions lack some of the features that define an RDBMS solution due to the reason of scalability. That does however not mean that they are primitive datastores only unstructured and simple. Detail explanation will be followed by the example of Document, Column Family and Graph databases which are the three representative type of the Nosql database [6].

3 Bigdata Characteristics and Databases

3.1 Diverse Database Models of Big Data Era

3.1.1 Column-Family Stores Database

Cassandra, HBase, Hypertable, and Amazon Simple DB are notable database product in column-family stores. We define column-family data stores and their structure with reference to Cassandra. Column-families are tuples that consist of key-value pairs, where the key is mapped to a value that is a set of columns (see fig 3). The values are classified into several column-families, and each column-family becomes a data map. Column-Family Stores preserve data that meet key values instead of rows [2]. Each row has many columns related to its key (Row key A in Fig 3a). Column-Family Store guarantees consistency, transaction, availability, various queries, and extensibility of the column-family data. The query functions are performed using the Cassandra Query Language (CQL) with basic query services and indexing techniques.

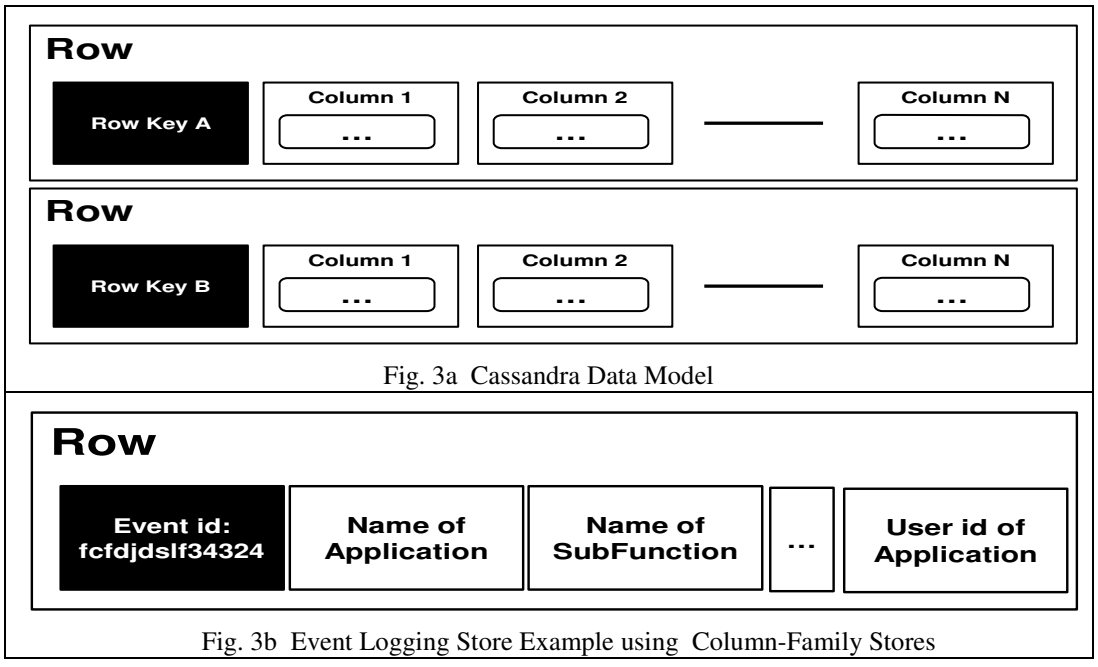

Fig. 3a Cassandra Data Model

Fig. 3b Event Logging Store Example using Column-Family Stores

Fig. 3 An example of Column-Family Stores Database

The best uses for Column-Family Stores are event logging, content management systems, blogging-platform services, and visitor countering services. In managing big data, using column-families might allow the storage of blog entries with tags, categories, links, and trackbacks in different columns (see fig 3b). However, Column-Family Stores are inefficient in terms of complex query and its frequent modifications.

Therefore, if measuring distance or calculating on the shortest route are as important as the data itself, using column-family store is not a good solution.

3.1.2 Document Database

Organizations create and manage significant amounts of document in the administration of their business. It is impossible for general relational databases to manage and analyze such a large amount of document data directly. Among the commercialized Document DBs such as CouchDB, Terrastore, and OrientDB, MongoDB is the representative schema-free document DB written in C++ and developed in an open-source.

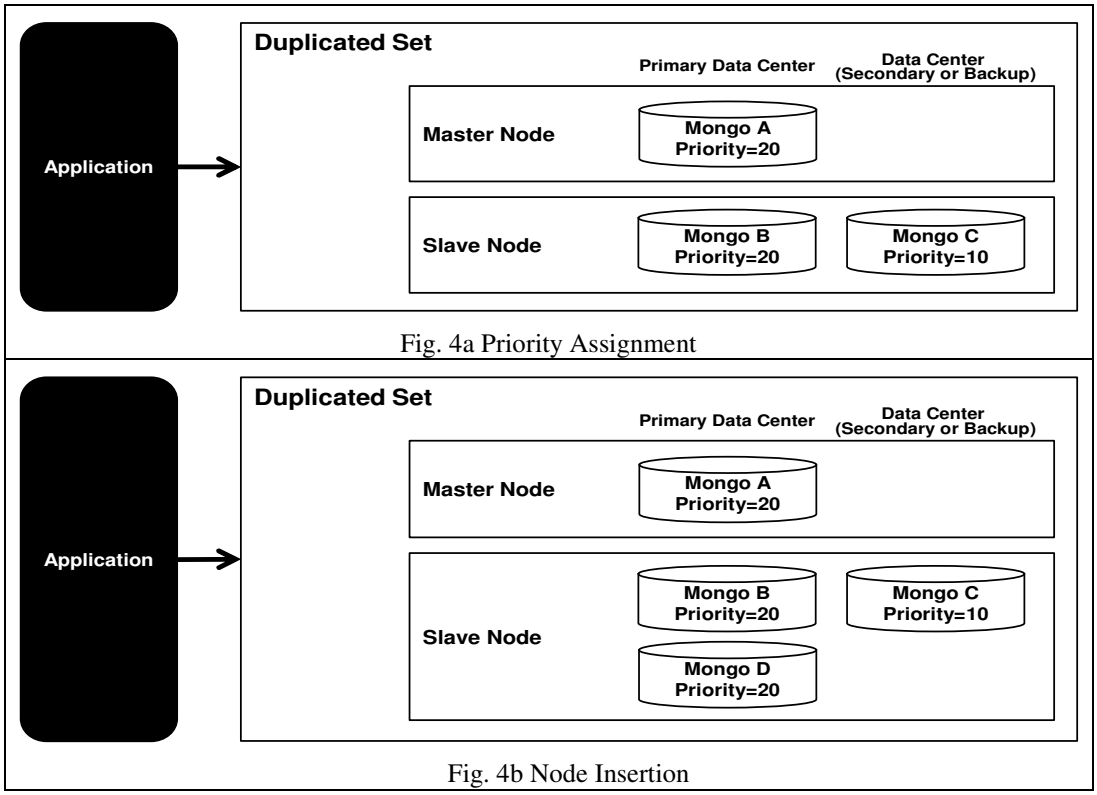

Fig. 4 An Example of Document Database

MongoDB closed the gap between the fast and highly scalable column-family-stores and the feature-rich traditional relational database management systems. Document databases are considered to be the next logical step of simple column-family-stores with slightly more complex and meaningful data structures. Document database encapsulates a key/value-pairs in documents. On the other hand, there is no strict schema to which documents have to conform, and this eliminates the need for schema migration efforts. Documents in the form of XML, JSON, BSON, and so on are

major concepts of Document databases. When inserting a new attribute, there is no need to define the attribute and change the existing document. MongoDB guarantees consistency, atonic transaction, availability, various queries, and extensibility of document data. Fig 4a shows an example of assigning priority to each node. Fig 4b shows an example of inserting a new node MongoDB into the existing duplicated set. The best uses for Document DBs are content management systems, blogging platforms, Web analysis, real-time analysis, and applications for electronic commerce. However, Document DBs are not a solution for web site with complex transactions or queries having dynamically changing calculation structure. The document data model - Document DBs - facilitate website construction because the data model supports unstructured data natively and does not require costly and time-consuming migrations when the application requirements change while Column-Family stores are not adequate for prototyping or technical reviewing premised on the website refactoring.

3.1.3 Graph Database

Bigdata analysis requires expressing various relationships between entities in graphical form. Graph databases use graph structures with nodes, edges, and properties to represent and store data. Nodes are extremely similar in nature to objects, with which object-oriented programmers are familiar. Based on graphic theory, every element contains a direct pointer to its adjacent element. In addition, no index lookups are necessary (fig 5). Compared to relational databases, graph databases are faster for associative data sets and map more directly to the structure of object-oriented applications. Because Graph databases do not typically require expensive join operations, they can scale to large data sets. In the meantime, relational databases are typically faster at performing the same operation on large numbers of data elements, but Graph databases are more suitable for managing ad hoc and changing data with te evolving schema because they depend less on rigid schemas. Similar to Column-Family Stores and Document databases, Graph databases guarantee consistency, transaction, availability, various queries, and extensibility of column-family data. The examples shown in Fig 6 are relationships with attributes and node partitions, respectively.

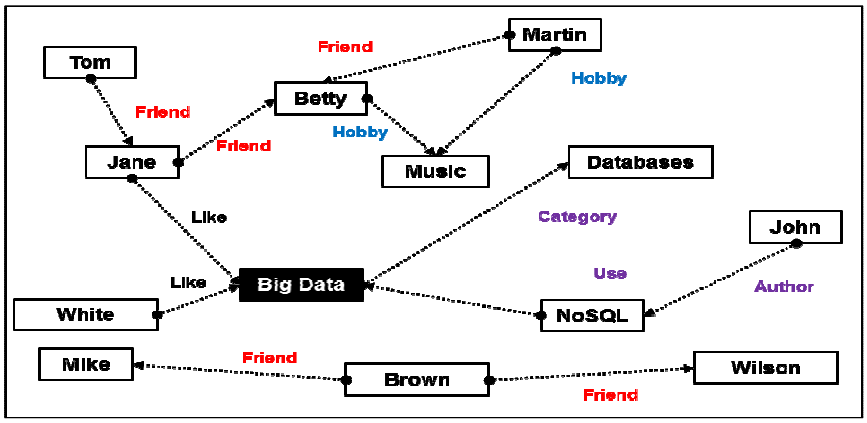

Fig. 5 Graph Database Structure Example

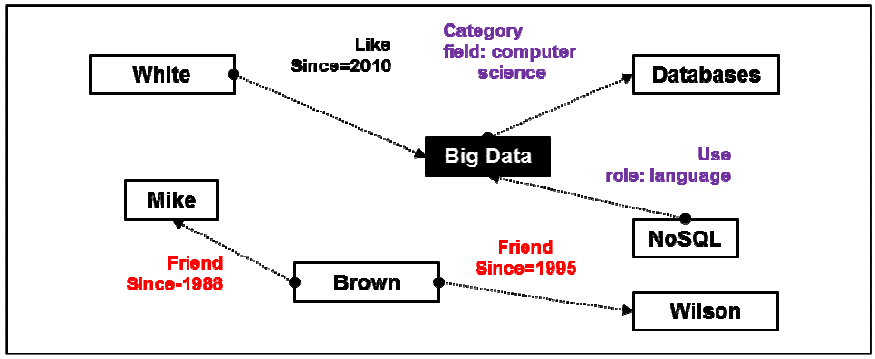

Fig. 6 Attribute Relationship Example

The best uses for Graph databases are linked data, routing, dispatch, location-based services, and recommendation services. However, Graph databases do not support an optimal solution for entity updates, especially when modifying entities or attributes. Consequently, Graph databases are a powerful tool for graph-like queries, for example, computing the shortest path between two nodes in a graph. Other graph-like queries can be performed with Graph databases in a natural way, such as graph diameter computations or community detection.

3.2 Polyglot Persistency and Bigdata DB selection criteria

This research introduced three representative type of Nosql database and their features. Table 1 is a list of three Nosql data models and the commercial products. The list is not inclusive of the entire database, but is the list of the Nosql databases commonly encountered. The list also includes the appropriate use cases, according to aforementioned database selection criteria such as scalability, data size, and data structure complexity.

Table 1: Types of Data Model

Data Model	Database Examples	Scalability	Data Size	Data Complexity
Column-Family Stores	Amazon SimpleDB, Cassandra, HDB, Hyper Table	High	High	Low
Document Database	CouchDB, MongoDB, OrientDB, RavenDB, Terrastore	Middle	Middle	Middle
Graph Database	Block DB, HyperGraphDB, Infinity Graph, Neo4J, OrientDB	Low	Low	High

To recap previous chapter, the common characteristics of Nosql are: (1) It has no schema, and Nosql systems allow the use of SQL-like query languages. (2) Nosql works well in clusters. (3) Nosql is open source and is developed for the Web environment of the 21st century._However, we do not think that the traditional relational databases will disappear, but they will continue to be used in many private and public ICT projects. The familiarity, reliability, versatility, and technical support are all compelling factors of relational databases that the majority of projects cannot be overlooked. What is different from the past is the fact that we consider relational databases an option with regard to data storage. Such perspective is referred to as the polyglot persistency [7]. Instead of selecting RDBs simply because everyone else is, one should select data storage considering the nature and characteristic of the data. Through the following cast study, we introduce a database selection criteria in bigdata environment using polyglot persistency.

4. Case Study: Database development for Smart Education using Bigdata

4.1 Smart Education Reform and Educational Bigdata in Korea

Through the diffusion of computerization in the mid 1980s, digitized educational information started to disseminate with Computer Aid Instruction (CAI) education in Korea. Later, distribution of the Internet in the 1990s led to the expansion of Web-based education. Since then, educational data format has been defined complying the Korea Data Education Metadata (KEM) belong to the Korean Industrial Standards (KS). Educational data have been classified into resource types, such as practice, simulation, questionnaire, diagram, figure, graph, index, slide, tables, statements, test, experiment, problem statements, self-diagnosis, lectures, and more, then saved and managed by relational database [3]. However, existing complex data definitions and unified storing method are too complex and inefficient considering schema-free data from multiple resources and multiple types of databases available in the bigdata environment.

With these considerations, it is required to re-classify and selecting database for educational data. Educational data can largely be classified into □unstructured or □semi-structured “teaching-learning” data that is generated from teacher-student interactions, and □non-teaching-learning data, such as structured educational administration data that can be saved into fixed fields. □Structured data refer to the information related to the school’s current state of affairs, teacher and student information, personnel-payroll information, school administration information, health and meal information, and academic achievements. □Semi-structured data refers to student/teacher evaluation data/schemas, E-learning content, e-books, teacher’s guides, and e-textbooks, etc. □Unstructured data refer to e-mails, blogs, social networking services (SNS), edutainment, digital textbooks, and image/video/audio data, etc.

Since 1996, semi-structured teaching-learning data have been shared and distributed through the website by the Korean government. In addition, Education Broadcasting Station (EBS) started televising educational video clips out of unstructured data. And it established and operates the Education Data Resource Bank (EDRB, http://www.edrb.co.kr) to store them. However, such services are systems only designed for search and collection, which is different from more student-specific, personalized instruction service through the appropriate mixed application of the current relational database system and Nosql technology.

4.2 Bigdata Service Model for Smart Education

The Korean government is currently conducting a project to provide optional personalized learning, to amplify each student's learning potential, and to provide equal education opportunities for everyone through the Bigdata based Smart Education services. The Smart Education Strategic Promotion Plan was established aiming for 2015, and it is being used to motivate reforms to the education system. In order that the smart education service is incorporated, education data should be used to implement optimal strategies for education policy, education systems, education methods, etc. The Smart Education Strategic Promotion Plan is largely classified and promoted in four different areas: (1) Expansion of self-directed time: classes and study hours that were limited to specific times become available anytime, as required. (2) Expansion of adaptive educational capacity: utilize digital textbooks, enable online lectures/evaluation, and conduct personalized education. (3) Expansion of resource-enriched educational content: increase student creativity and problem-solving skills by providing resource-enriched learning materials through the development of digital textbooks and the active use of educational content for public purpose. (4) Expansion of technology-embedded space: promote outside-the-classroom learning at home or on the global stage by creating school infrastructures based on the cloud, and conduct online classes.

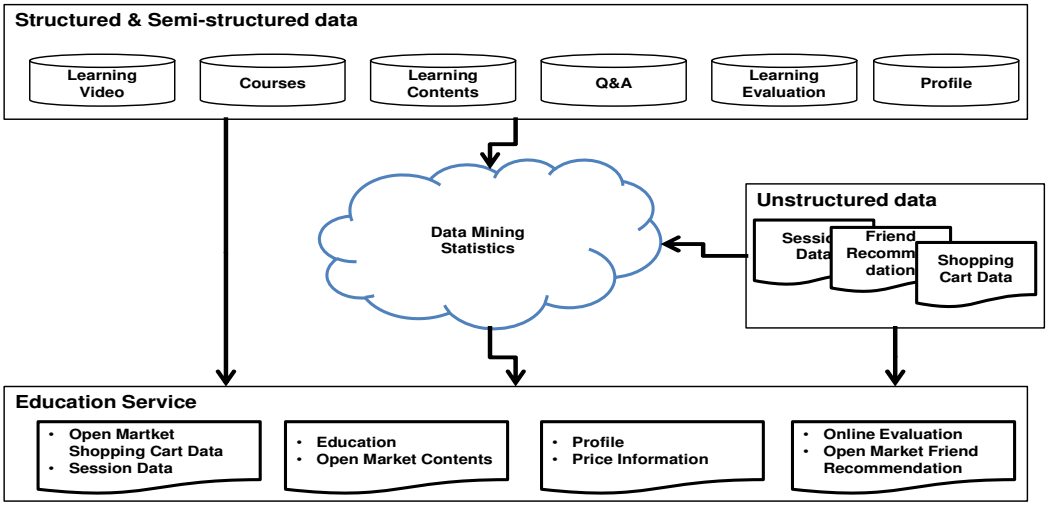

Fig. 7 Learning Support through Personalized Education Portfolio

The business model for smart education system operates based on bigdata, and performs the role of promoting the e-learning content industry through composing personal training portfolios. Various educational services, such as Session Data, Education, Open Market content, Profile, Price Information, Online Evaluation, and Open Market Friend Recommendation services are provided. Students and parents are provided with an accurate guide through the use of the educational content. Also, industry development is promoted through the use of the course data, learning content data and learning evaluation data. This model utilizes the bigdata-based personal education log, provides personalized education consulting to every student, improves equal educational opportunities and the quality of public education, and opens learning data to promote the development of educational programs necessary for students. In addition, the model provides tailored education courses, personal preferences and an appropriate learning portfolio for educational interests of individual student.

4.3 Polyglot Persistency Theory-based DB Construct

Many organizations tend to use the sngle database management system to store business transactions, session management data, and other storage needs such as reporting, BI, data warehousing, or logging information (Fig 8) using with the computer programing perspective. Complex applications combine different types of problems; therefore, choosing the correct language for each job might be more productive than attempting to fit all aspects into a single language. Similarly with the database management perspective, when working on an eSmart education service, using an appropriate data store for the open market content that is highly available and support scallability is important, but the same data store cannot assist in finding products bought by customers' friends because this is an entirely different question.

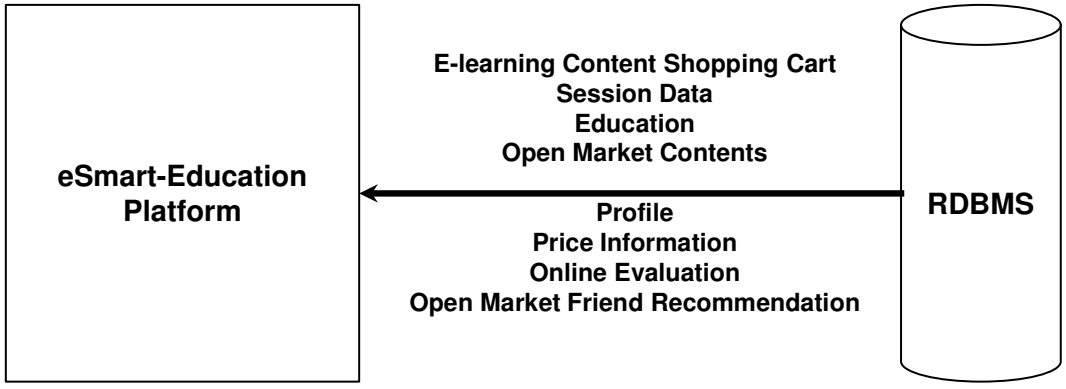

Fig. 8 Use of RDBMS for Entire Application and Storage

The relational database management system solution is advantageous when reinforcing existing relationships, but is disadvantageous when discovering new

relationships or finding different tables within the same object. For instance, using column-family stores is logical because open market e-learning content shopping carts are usually accessed through user IDs, and once confirmed and paid by customers, can be saved in the relational database system. Similarly, the session data keyed by the session ID should be stored and managed by column-family stores. If products need to be recommended to customers when they place products into their shopping carts, for example, "your friends also bought these products" or "your friends bought these accessories for this product," then introducing a graph data store becomes relevant (Fig 8). The application does not need to use a single data store for all its needs because different databases are built for different purposes, and not all problems can be elegantly solved by a single database. Even using specialized relational databases for different purposes, such as data warehousing appliances or analytics appliances, within the same application can be viewed as polyglot persistence.

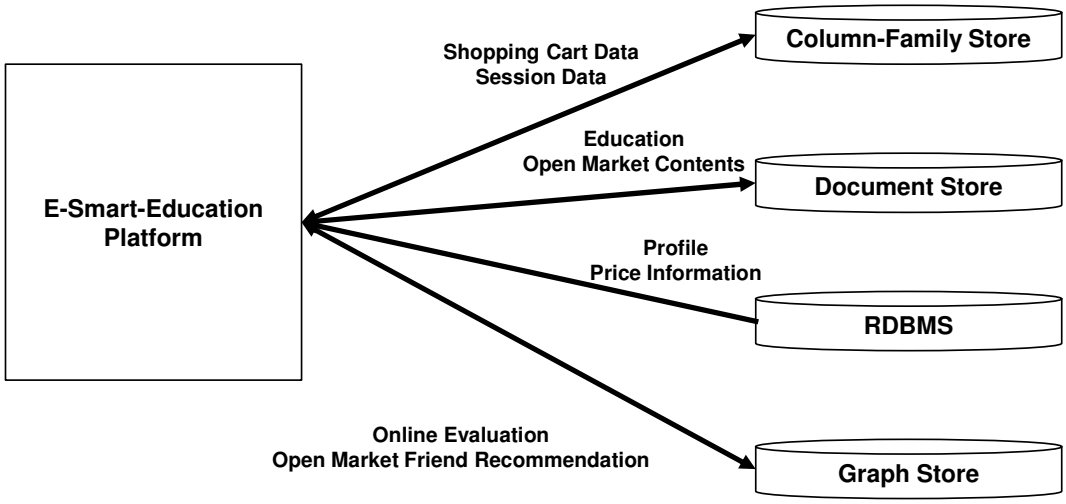

Fig. 9 Polyglot Persistence based database development for Smart Education Service

4 Conclusion

Recently, Nosql became the alternative database management system for bigdata. However, Nosql and relational database has their own pros and cons respectively. In this backdrop, we suggested the relational database selection criteria according to the specific area of information system and presented its combination example for eSmart education service cases in Kora. This case study was conducted prior to the full scale implementation of smart education service. Thus, this study provides a great opportunity for estimating the type and size of the bigdata related to educational activities, and to propose a desirable database system framework for smart education services.

The introduction of NoSQL data storage technologies will force organization database administrators to think about how to use the new storage. Organizations are used to having uniform relational database management system environments that an organization first uses. Using polyglot persistence, database administrators will have to become more poly-skilled in order to learn how some of these Nosql technologies work, how to monitor these systems, create backups for them, and retrieve and input data into these systems. Once an organization decides to use Nosql, issues such as licensing, support, tools, upgrades, drivers, auditing, and security might arise. Many Nosql technologies are open source and have an active community of supporters; in addition, there are companies that provide commercial support. In traditional relational database management systems, data can be accessed using query tools. With NoSQL databases, there are query tools as well. However, the idea is for the Nosql database management system only managing data while doesn't take care of security. With this approach, the responsibility for the security lies with the application. Now is the time that the flexibility to compose data from various perspectives suitable to the cloud computing environment that is represented by distributed parallel processing is required. In other words, in the current bigdata era, data models must demonstrate flexibly in various ways in order to establish data models in accordance with the complex changing environment of computing systems.

References

[1] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, "Big Data: The Next Frontier for Innovation, Competition, and Productivity," McKinsey & Company, 2011.

[2] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Sixth Symposium on Operating System Design and Implementation, pp. 137-150, 2004.

[3] Korea Industrial Standards Commission, Information Technology – Elementary and Middle School Education Information Metadata KSX7001, 2004.

[4] Leavitt, N., "Will NoSQL databases live up to their promise?", Computer, Vol.43, No.2, 12-14., 2010

[5] Vicknair, C., M. Macias, et al.,"A comparison of a graph database and a relational database: a data provenance perspective", Proceedings of the 48th annual Southeast regional conference, ACM., 2010

[6] Michael Kopp, "About:Performance - Application performance, scalability and architecture NoSQL or RDBMS? – Are we asking the right questions?", http://apmblog.compuware.com, 2011

[7] Sadalage, P. J. and Fowler, M., 2012, "NoSQL distilled - A brief guide to the emerging world of polyglot persistence", Pearson Education, Inc.