Journal of The Korea Society of Computer and Information
Vol. 23 No. 3, pp. 49-54, March 2018
www.ksci.re.kr
https://doi.org/10.9708/jksci.2018.23.03.049

# Development of the Design Methodology for Large-scale Data Warehouse based on MongoDB

Junho Lee*, Kyungsoo Joo**

## Abstract

A data warehouse is a system that collectively manages and integrates data of a company. And provides the basis for decision making for management strategy. Nowadays, analysis data volumes are reaching critical size challenging traditional data ware housing approaches. Current implemented solutions are mainly based on relational database that are no longer adapted to these data volume. NoSQL solutions allow us to consider new approaches for data warehousing, especially from the multidimensional data management point of view. In this paper, we extend the data warehouse design methodology based on relational database using star schema, and have developed a consistent design methodology from information requirement analysis to data warehouse construction for large scale data warehouse construction based on MongoDB, one of NoSQL.

▸Keyword: Data warehouse, NoSQL, MongoDB, Design Methodology

## I. Introduction

요즘 분석 데이터는 기존의 데이터 웨어하우스로 처리하기에는 어려운 크기로 확대되었다[1]. 기존의 데이터 웨어하우스는 관계형 데이터베이스를 기반으로 구현되었는데, 더 이상 이렇게 큰 데이터에는 적합하지 않다[2, 3, 4].

NoSQL(Not Only SQL)은 데이터 웨어하우스를 위한 새로운 가능성을 제시한다[5]. 이러한 NoSQL 데이터베이스 중 MongoDB는 고성능을 목표로 만들어진 데이터베이스이고, 빅 데이터 저장에 용이한 분산 확장을 지원하는 데이터베이스이다. 따라서 MongoDB는 기존의 관계형 데이터베이스를 기반으로 하는 데이터 웨어하우스에서 처리하기 어려운 빅 데이터를 처리하기에 유용하다.

본 논문에서는 스타 스키마를 이용한 관계형 데이터베이스 기반의 데이터 웨어하우스 설계 방법론을 확장하여 NoSQL 중 하나인 MongoDB 기반의 대규모 데이터 웨어하우스 구축을 위한, 정보 요구사항 분석부터 데이터 웨어하우스 구축까지의 일관된 설계 방법론을 개발하여 제안한다. 제안한 방법론으로 MongoDB 기반의 대규모 데이터 웨어하우스를 구축하기 위한 데이터 셋은 판매 정보 데이터를 사용하였다.

본 논문의 구성은 다음과 같다. 2장에서는 데이터 웨어하우스 및 MongoDB, 스타 스키마에 대해 기술하고, 3장에서는 본 논문에서 제안하는 스타 스키마를 이용한 관계형 데이터베이스 기반의 데이터 웨어하우스 설계 방법론을 확장하여, NoSQL 중 하나인 MongoDB 기반의 대규모 데이터 웨어하우스 구축을 위한, 정보 요구사항 분석부터 데이터 웨어하우스 구축까지의 일관된 설계 방법론을 설명한다. 그리고 제안한 방법론에 실제 데이터를 적용하여 데이터 웨어하우스를 구축하고, 구축된 MongoDB 컬렉션을 보여준다. 4장에서는 결론 및 향후 연구에 대해 기술한다.

## II. Related works

본 장의 2.1절은 데이터 웨어하우스의 필요성과 최근 연구동향

• First Author: Junho Lee, Corresponding Author: Kyungsoo Joo
*Junho Lee (wnsgh461@naver.com), Dept. of Computer Science, Soonchunhyang University
**Kyungsoo Joo (gsoojoo@naver.com), Dept. of Computer Software Engineering, Soonchunhyang University
• Received: 2017. 12. 06, Revised: 2017. 12. 30, Accepted: 2018. 03. 08.
• This work was supported by the Soonchunhyang University.

을 설명하고, 2.2절에서 NoSQL 데이터베이스 중 하나인 MongoDB에 대한 장점과 필요성과, 기존의 관계형 데이터베이스의 결점을 보완한 확장성에 대해 설명한다. 마지막으로 2.3절에서 스타 스키마와 스타스키마의 표기법을 보여주고 장을 끝낸다.

### 2.1 Data Warehouse

데이터 웨어하우스는 기업의 자원이라고 할 수 있는 데이터를 일괄적으로 통합·관리하여 경영전략을 수립할 때 필요한 의사결정을 지원하는 시스템이다.

데이터 웨어하우스는 관리자의 의사결정을 지원하기 위한 주제중심의(Subject-oriented), 통합된(Integrated), 비휘발성의(Nonvolatile), 시간변이적인(Time variant) 데이터 집합이다[6].

최근의 데이터 웨어하우스는 빅 데이터의 처리를 위해 기존의 관계형 데이터베이스 기반에서 NoSQL 데이터베이스 기반으로 변경하는 여러 연구가 진행 중이다.[7, 8]

### 2.2 MongoDB

MongoDB는 문서 지향형 데이터베이스이다. 관계형 모델을 사용하지 않고 문서 모델을 채택하는 이유는 분산 확장을 쉽게 하기 위한 것이지만, 다른 이점도 가지고 있다. 내장문서와 배열을 허용함으로써 문서 지향 모델은 복잡한 계층 관계를 하나의 레코드로 표현할 수 있다. 또한 MongoDB에서는 문서의 키와 값을 미리 정의하지 않는다. 따라서 고정된 형태의 스키마가 존재하지 않는다. 고정된 스키마가 존재하지 않기 때문에 필요에 따라 필드를 추가하고 삭제하는 것이 쉬워졌고, 개발과정 또한 빠르게 이루어질 수 있다[9].

MongoDB는 분산 확장을 염두에 두고 설계된 데이터베이스로, 여러 장비에 데이터를 분산 확장하여 저장하는 기능을 지원한다. 문서를 자동적으로 재분배하고 사용자 요청을 올바른 장비에 라우팅 함으로써 클러스터 내 데이터양과 부하를 조절할 수 있다. 이러한 분산 확장은 기존의 관계형 데이터베이스의 성능 확장 보다 경제적이고 확장에 용이하다.

또한, MongoDB는 범용 데이터베이스 목적으로 만들어져 데이터의 생성, 읽기, 변경, 삭제 외에도 특별한 기능을 제공한다. 그러나 MongoDB에서는 관계형 데이터베이스에서 일반적으로 제공하는 기능 중 몇몇 기능이 없는데 그 중 대표적인 기능은 조인과 다중 트랜잭션이다. 이런 기능들은 분산 시스템에서 효율적으로 제공하기 어렵기 때문에 제외되었고, 높은 확장성을 제공하는 아키텍처를 위한 결정이다[9].

### 2.3 Star Schema

스타 스키마는 데이터를 가능한 비 정규화 함으로써 만들어진다. 유사한 객체들의 속성을 하나의 단일 테이블로 결합하기 때문에 조인 연산이 줄어 검색 성능이 개선된다. 또한, 스타 스키마는 다차원 데이터를 표현하기 위한 기법이다. 사실(Fact) 테이블과 차원(Dimension) 테이블로 구성되고, 사실 테이블을 중심으로 차원 테이블이 뻗어나가는 형태가 아래 Fig. 1과 같은 '별'형태와 유사하여 스타 스키마로 불린다[10].

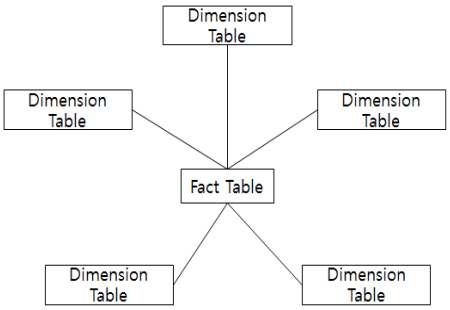


Fig. 1. Star Schema

## III. Development Data Modeling Methodology for MongoDB and is Application

데이터 모델은 데이터베이스에 필요한 데이터 구조의 개념적 표현이다. 데이터 구조에는 데이터 객체, 데이터 객체 간의 연결 및 객체에 대한 작업을 제어하는 규칙이 포함된다. 데이터 모델의 목표는 데이터베이스에 필요한 모든 데이터 객체가 완전하고 정확하게 표현되도록 하는 것이다. 데이터 모델은 쉽게 이해할 수 있는 표기법과 자연어를 사용하기 때문에 사용자가 정확하게 검토하고 확인할 수 있다[11].

본 논문에서는 MongoDB 기반의 데이터 웨어하우스를 설계함에 있어 스타 스키마를 사용하는데, 개념적 데이터 모델로 시작하여 논리적 모델, 물리적 모델을 거쳐 MongoDB 데이터 웨어하우스 컬렉션을 정의하는 것으로 끝난다.

### 3.1 CDM(Conceptual Data Modeling)

개념적 데이터 모델링은 핵심 개념과 관계를 토대로 시스템의 범위를 파악하기 위한 모델링 단계로, CDM 과정은 아래 Fig. 2와 같다[12].

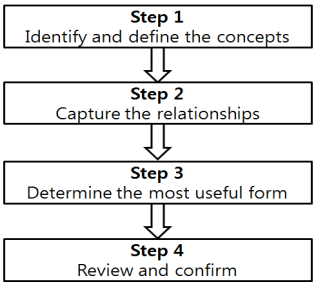


Fig. 2. Flowchart of CDM

첫 번째 단계는 개념을 식별하는 단계로서, Dimensional면에서 대답해야하는 구체적인 질문들을 결정한다. 이 질문들은 실제

로 구축할 시스템이 분석해야하는 것에 대한 질문들로 구성된다.

구체적인 질문들은 다음과 같다.
① 제품별 기간별 매출현황
② 제품별 판매지역별 매출현황
③ 프로모션별 제품별 매출현황

두 번째 단계에서는 각 개념들 간의 관계를 파악하는 단계로서, 위의 구체적인 질문을 토대로 Grain 행렬을 만들게 된다. Grain 행렬은 질문의 측정값이 열이 되고 질문의 번호가 행이 되는 스프레드시트이다[12].

구체적인 질문들은 매출 현황이라는 것에 중점을 둔다. 따라서 Table 1과 같은 Grain 행렬을 만들 수 있다.

Table 1. Grain Matrix

| | Sales Status |
|---|---|
| Product | 1, 2, 3 |
| Term | 1 |
| Area | 2 |
| Promotion | 3 |

세 번째 단계는 파악한 개념들의 관계를 토대로 해당 데이터 모델을 표현하기 위한 표기법을 결정해야 한다. 정보공학 방법론에 의해 표기할 수 있고, Axis기법 이라고 하는 비즈니스 친화적 모델링 형식을 사용 할 수도 있다[12].

본 논문에서는 데이터 웨어하우스를 구축하기 위해 기본 표기법으로 스타 스키마를 채택한다. 개념 스타 스키마는 Fig. 3과 같다.

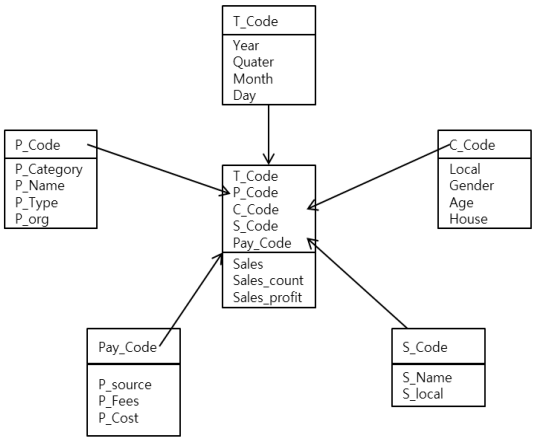


Fig. 3. Conceptual Star Schema

네 번째 확인 및 검토 단계에서는 도출된 모델이 올바른지 확인하는 단계이다. 올바른 모델이라고 판단 될 때까지 위의 모델링 방법을 반복 적용하여 수정할 수 있고, 올바르다고 판단된다면 논리적 모델링 단계로 넘어간다.

### 3.2 LDM(Logical Data Modeling)

논리적 데이터 모델링은 상세 내용을 파악하는 단계로 CDM 과정에서 나온 데이터 모델을 토대로 다음 Fig. 4와 같은 LDM 과정을 거쳐 만들게 된다.

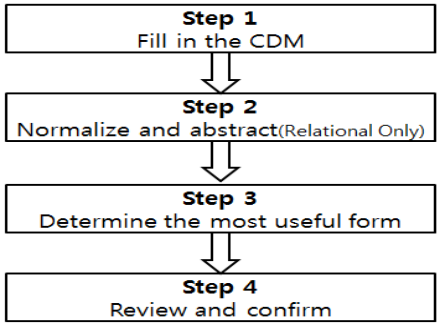


Fig. 4. Flowchart of LDM

첫 번째 단계는 개념의 속성을 식별하고 정의하는 것이다. CDM에서 정의한 개념은 LDM의 엔티티가 되며, 이 단계를 마치게 되면 속성 템플릿과 속성 특성 템플릿을 작성할 수 있다. 각 엔티티의 속성 또는 데이터요소를 파악하여 속성 템플릿을 작성한다.

CDM에서 얻은 데이터 모델을 기반으로 Table 2와 같은 속성 템플릿을 작성한 후, Table 3과 같은 속성 특성 템플릿을 작성한다.

Table 2. Properties Template

| | Product | Term | Custom er | Paymen t | Seller |
|---|---|---|---|---|---|
| Name | p_name , org_na me | | u_name | | s_name |
| Text | p_type, p_categ ory | | l o c a l , gender, house | pay_so urce | s_local |
| Code | p_code | t_code | c_code | pay_co de | s_code |
| Date | | m o n t h , d a y , year | | | |
| Quater | | quater | | | |
| Number | | | age | f e e s , cost_pe rcent | |

두 번째 단계는 관계형 데이터베이스를 사용할 경우 진행하는 정규화 단계이기 때문에 본 논문에서는 진행하지 않는다.

세 번째 단계는 유용한 표기법을 결정하는 단계이다. CDM 단계에서 사용하였던 스타 스키마를 기본 표기법으로 사용한다. 위의 Table 3, 4를 작성 한 후 스타 스키마로 변환시키게 되면 다음 Fig. 5와 같은 논리 스타 스키마를 얻을 수 있다.

Table 3. Properties Characteristics Template

| Property | Definition | Sample Value | Format | Length |
|---|---|---|---|---|
| p_name | production name. | PC | char | 100 |
| org_name | supply company name. | SCH Univ. | char | 100 |
| u_name | customer name. | JunHo Lee | char | 30 |
| s_name | seller name. | GilDong Hong | char | 30 |
| p_type | product specification. | AP_0001 | char | 50 |
| p_category | product category. | 0001A | char | 50 |
| local | customer's purchase area. | Asan | char | 100 |
| gender | customer's gender. | M or F | char | 1 |
| house | type of the customer's house. | APT | char | 100 |
| pay_source | payment method. | Card or Cash | char | 20 |
| s_local | sales area of the product. | Asan | char | 100 |
| p_code | product code. | Product0001 | char | 50 |
| c_code | customer code. | C0001 | char | 50 |
| t_code | term code. | A2017M4 | char | 50 |
| pay_code | billing code. | P_0001 | char | 50 |
| s_code | seller code | A0001 | char | 50 |
| month | sales period (month). | 10 | | |
| day | sales period (day). | 5 | date | |
| year | fiscal year. | 2017 | | |
| quater | sales period (quarter). | 1 | integer | 2 |
| age | customer's age. | 45 | integer | 10 |
| fees | commission rate. | 3.5 | float | 20 |
| cost_percent | payment rate. | 90 | float | 20 |

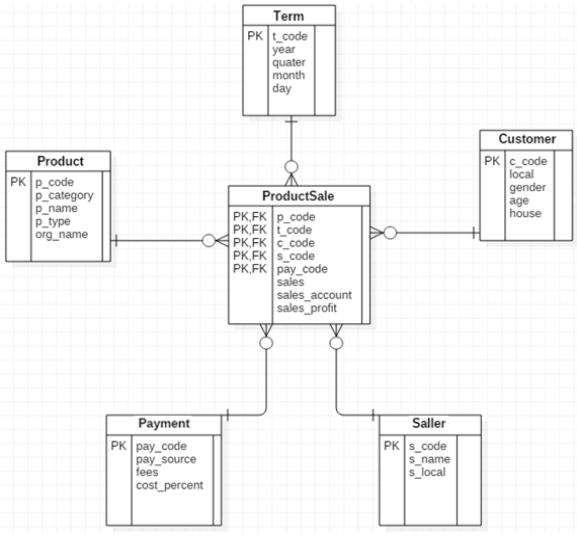


Fig. 5. Logical Star Schema

네 번째 단계는 CDM단계와 마찬가지로 데이터 모델을 확인하는 단계이다.

### 3.3 PDM(Physical Data Modeling)

물리적 데이터 모델링(Physical Data Modeling)은 실제 MongoDB 기반의 데이터 웨어하우스 스키마를 확정하는 단계로, LDM 과정에서 나온 데이터 모델을 토대로 만들어지게 된다. PDM 과정은 아래 Fig. 6과 같다.

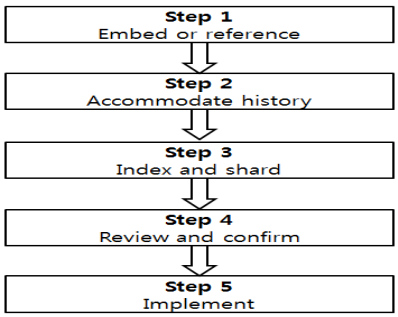


Fig. 6. Flowchart of PDM

첫 번째 단계로 LDM 단계에서 나온 논리적 데이터 모델을 물리 스타 스키마로 변경한다. 변경된 물리 스타 스키마는 Fig. 7과 같다.

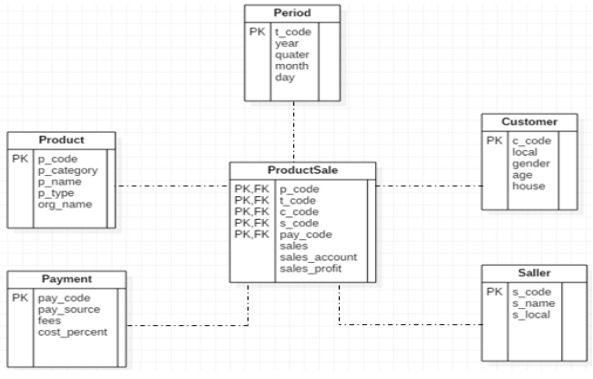


Fig. 7. Physical Star Schema

두 번째 단계는 Accommodate History로 필드 값이 시간에 따라 어떻게 변하는지에 대한 옵션을 설정한다. 필드 값의 변화를 다루기 위한 옵션은 SCD(Slowly Changing Dimension)로 불리며, 숫자(0, 1, 2, 3)로 구성된다[12].

① SCD 0 : 원 상태를 저장하고 변경사항은 저장하지 않는다.

② SCD 1 : 가장 최근의 상태를 저장한다.

③ SCD 2 : 데이터의 변경사항이 있을 때마다, 모든 변경사항을 컬렉션에 저장한다.

④ SCD 3 : 일부 기록에 대해 요구사항(최신보기, 이전보기 등...)이 있다.

각 컬렉션에 필요한 옵션을 결정해 컬렉션 기록 템플릿을 만든다[12].

해당 물리 스타 스키마에 적합한 컬렉션 기록 템플릿을 정의하면 Table 4와 같이 정의 할 수 있다.

Table 4. Collection History Template

| | Type 0 | Type 1 | Type 2 | Type 3 |
|---|---|---|---|---|
| Period | ◎ | | | |
| Product | | ◎ | | |
| Customer | | ◎ | | |
| Payment | | | ◎ | |
| Saller | | ◎ | | |
| ProductSale | | | ◎ | |

세 번째 단계는 인덱싱(Indexing)과 샤딩(Sharding)이다. 인덱싱은 논리적 데이터 모델의 기본키와 대체키를 고유 인덱스(Unique Index)로 변환하는 것이다. 샤딩은 컬렉션이 두 개 이상의 부분으로 분리되는 경우를 말하며 수평, 수직 두 가지의 분할 방식이 있다[12].

네 번째 단계는 확인 및 검토 단계로 해당 모델이 올바르게 작성된 모델인지 검토한다.

다섯 번째 단계는 설계된 데이터 모델을 토대로 MongoDB 데이터 웨어하우스 컬렉션을 정의한다.

### 3.4 MongoDB Data Warehouse Collection

3.3절의 Fig. 7의 물리 스타스키마를 통해 정의된 MongoDB 기반의 데이터 웨어하우스 컬렉션은 다음 Table 5, 6, 7, 8, 9와 같다.

Table 5. Product Collection

```
Product:
{
    p_code : "Product0001",
    p_category : "00001A",
    p_name : "PC",
    p_type : "APC_0001",
    org_name : "Soonchunhyung Univercity"
}
```

Table 6. Period Collection

```
Period:
{
    t_code : "A2017M4",
    date : [
    {
        year : "2017",
        month : "10",
        day : "30"
    ]},
    quater : "1"
}
```

Table 7. Customer Collection

```
Customer:
{
    c_code : "C0001",
    local : "Asan"
    gender : "M",
    age : "35",
    home : "APT"
}
```

Table 8. Saller Collection

```
Saller:
{
    s_code : "S0001",
    s_name : "Gil-Dong Hong",
    s_local : "Asan"
}
```

Table 9. Payment Collection

```
Payment:
{
    pay_code : "P_0001",
    pay_source : "Card"
    fees : "3.5",
    cost_percent : "90"
}
```

## IV. Conclusions

본 논문에서는 스타 스키마를 이용한 관계형 데이터베이스 기반의 데이터 웨어하우스 모델링 방법론을 확장하여 MongoDB 기반의 대규모 데이터 웨어하우스 구축을 위한, 정보 요구사항 분석부터 데이터 웨어하우스 구축까지 일관된 설계 방법론을 개발하여 제안하였다. 본 방법론에 따르면 개념, 논리, 물리 각 단계별 질문을 통해 시스템의 목적 및 필요한 개념과 관계를 질문을 통하여 식별한다. 또한, 구체적인 질문을 통해 Grain 행렬을 찾아내고 찾은 Grain 행렬을 통해 모델을 만들어 최종 단계까지 스키마를 도출할 수 있다. 또한, 각 단계별 검토 과정을 통해 수정이 간략하게 이루어 질 수 있다.

본 논문에서는 MongoDB를 기반으로 한 대규모 데이터 웨어하우스의 일관된 설계 방법론을 제안함으로써, 기존의 관계형 데이터베이스를 기반으로 한 데이터 웨어하우스의 한계인

대용량 데이터 저장 및 처리의 어려움을 극복하는, 빅 데이터를 위한 데이터 웨어하우스 구축 방법론을 제안하였다.

향후 NoSQL 중 하나인 MongoDB 기반의 OLTP(On-Line Transaction Processing) 구축을 위한 방법론과 OLAP(On-Line Analytical Processing) 구축을 위한 방법론을 통합하여, 일관된 통합 설계 방법론에 대해 개발하고자 한다.

## REFERENCES


[1] Jacobs, A., “The pathologies of big data,” Communications of the ACM, 52(8), pp. 36-44, 2009.

[2] Stonebraker, M., “New Opportunities for New SQL,” Communications of the ACM, Vol.55, issue11 pp. 10-11, 2012

[3] Cuzzocrea, Alfredo, Ladjel Bellatreche, and Il-Yeol Song., "Data warehousing and OLAP over big data: current challenges and future research directions," Proceedings of the sixteenth international workshop on Data warehousing and OLAP. ACM, 2013.

[4] Dehdouh, Khaled, Omar Boussaid, and Fadila Bentayeb., "Columnar nosql star schema benchmark," International Conference on Model and Data Engineering. Springer, Cham, 2014.

[5] Chevalier M, El Malki M, Kopliku A, Teste O, Tournier R., "Implementing Multidimensional Data Warehouses into NoSQL," 17th International Conference on Enterprise Information Systems, Vol.1, pp. 172-183, 2015.

[6] W.H.Inmon., "Building the Data Warehouse." John Wiley & Sons, 2002.

[7] Pereira, Daniel, Paulo Oliveira, and Fátima Rodrigues. "Data warehouses in MongoDB vs SQL Server: A comparative analysis of the querie performance." Information Systems and Technologies (CISTI), 2015 10th Iberian Conference on. IEEE, 2015.

[8] Bicevska, Zane, and Ivo Oditis. "Towards NoSQL-based Data Warehouse Solutions." Procedia Computer Science 104, pp 104-111, 2017.

[9] Michael Dirolf, "MongoDB The Definitive Guide." O'ReillyMedia, 2013.

[10] Christopher Adamson, “Star Schema The Complete Reference,” McGraw-Hill Osborne, 2010.

[11] Mamenko, J. "Introduction to data modeling and msaccess," Lecture Notes on Information Resources, 2004.

[12] Hoberman, S., “DataModeling for MongoDB,” Technics Publications, 2014.


## Authors


Junho Lee received the B.S. degrees in Computer Software Engineering from Soonchunhyang University, Korea, in 2015 respectivelySoftware Engineering at Soonchunhyang University, Asan, Korea, in 2015. He is currently a M. S. course in the Department of Computer Science, Soonchunhyang University. He is interested in database and big data database.

Kyungsoo Joo received the Ph.D. degrees in Computer Science from Korea University, Korea, in 1993 respectivelyKorea University, Seoul, Korea, in 1993. He is currently a Professor in the Department of Computer Software Engineering, Soonchunhyang University. He is interested in database and big data database.