

경영정보학연구
제5권 2호
1995년 12월

분산 데이터베이스를 위한 양식을 이용한 분산 설계 방법론

이 희 석¹⁾ 김 희 진²⁾ 김 영 삼³⁾

A Form-Based Distribution Design Methodology for Distributed Databases

This paper proposes a form-based distributed database design methodology (FD³). The methodology consists of five design phases such as (i) form requirement analysis (ii) schema integration (iii) distribution analysis (iv) distribution design, and (v) local logical/physical design. In the FD³, all the important design information for each phase is obtained by using an organizations forms. Users requirements are analyzed by using forms that contain logical and quantitative information for distribution design. FD³ resolves naming conflicts by employing SQLs based on the form field data in the schema integration phase. Furthermore, FD³ enhances the quality of distributed database design by incorporating communication costs into the design model. A real-life case is illustrated to demonstrate the usefulness of the FD³.

-
- 1) 한국과학기술원 경영정보공학과
 - 2) 한국 오리클 주식회사
 - 3) 한국과학기술원 경영정보공학과

I. 서 론

분산 데이터베이스 시스템 (Distributed Database System)은 최근의 정보 산업 환경에서 가장 주목 받고 있는 정보 기술 중의 하나이다. 분산 데이터베이스 시스템은 다수의 기술적 우위와 정보 기술 발전의 한 형태이기 때문에 많은 전문가들이 수년 내로 대부분의 조직들이 중앙 집중식 데이터베이스 시스템 (Centralized Database System)에서 분산 데이터베이스 시스템으로 이전할 것이라는 예측을 하고 있다 [Stonebraker, 1989]. 그러나, 현재까지는 실제 기업 데이터베이스 구축에 적용 가능한 분산 데이터베이스 관리 시스템 (DDBMS; Distributed Database Management System)이 존재하지 않을 정도로 실질적인 구현에는 해결되지 않은 많은 문제점들이 존재하고 있다 [Ozsu and Valduriez, 1991]. 이중에서도 특히, 분산 데이터베이스 시스템의 궁극적인 목표인 성능 (Performance), 비용 (Costs) 그리고 가용성 (Availability)등을 최적화하는 분산 설계 (Distribution Design)는 필수적이다.

분산 설계는 단편화 (Fragmentation)와 할당 (Allocation)의 두 설계 단계로 대별된다. 데이터베이스 단편화는 Chang과 Cheng [1980]이 처음으로 시도하였다. 단편화 방법은 수평 (Horizontal) 단편화와 수직 (Vertical) 단편화로 대별된다 [Ceri, 1984]. Ceri et al.[1983]는 수평 단편화 문제를 연구하여 비

중복 할당 문제의 수식을 선형 정수 계획법 (Linear Integer Programming)으로 제안했으며, Navathe et al.[1984]은 분산 데이터베이스에서의 수직 단편화에 관한 연구를 수행하였다. Apers[1982]는 수평 및 수직 단편화에 관한 연구를 하여 질의 처리 (Query Processing)를 기준으로 한 비중복 및 중복 할당을 위한 이론적 근거와 방법론을 제시하였다. 할당에 관한 연구는 Chu[1969]가 최초로 시도하였다. 그의 연구는 전체 운용 비용 (Overall Operation Cost) - 통신비용과 저장비용-을 최소화 시킬 수 있는 곳에 화일을 비중복 할당 (Nonredundant Allocation)하는데 중점이 주어졌다. Casey[1972]는 데이터 접근을 질의 (Query)와 갱신 (Update)으로 분리하여 중복 할당 (Redundant Allocation)에 관한 연구를 수행했다. 할당 문제는 관련된 여러 결정 요소를 수리적 모형화한 다양한 연구가 매우 활발하게 진행되고 있다 [Levin and Morgan, 1977; Dowdy and Foster, 1982; Wah, 1983; Lien et al., 1988; Gladney, 1989; Ram and Marsten, 1991; Lee and Sheng, 1992; Lee, 1993; Lee et al., 1994; Lee and Sheng, 1995; March and Rho, 1995].

단편화와 할당의 문제는 그 복잡성 때문에 완전히 개별적인 것으로 간주되어 대부분의 연구가 진행되어 왔으나 실제로는 두 설계 문제가 상호 영향을 미치는 관계에 있기 때문에 동시에 최적화되어지는 연구가 필수적이다 [Ozsu and Valduriez, 1991]. Ceri는 단편화

와 할당 문제를 동시에 고려하여 요구 분석 (Requirement Analysis), 개념 설계 (Conceptual Design), 광역 논리 설계 (Global Logical Design), 분산 설계, 지역 논리 설계 (Local Logical Design)와 지역 물리 설계 (Local Physical Design)의 6단계로 구성된 DATAID-D라는 분산 데이터베이스 설계 방법론을 제안하였다 [Ceri, Pernici and Wiederhold, 1987]. DATAID-D에는 분산 설계에 필요한 설계 정보를 추출하는 분산 요구 분석 단계가 추가되었다. 이 단계에서는 분산 사용자의 요구 사항 (User Requirement), 광역 데이터 (Global Data) 및 운용 스키마 (Operation Schemata)를 입력으로 하여, (i) 각 지역별 응용 업무 실행 횟수를 나타내는 빈도표 (Frequency Tables), (ii) 수평 분할의 기준이 되는 분할표 (Partitioning Tables), 그리고 (iii) 분할이 응용 업무 처리의 지역성 (Locality)에 어떠한 영향을 미치는지를 계량적으로 표시해 주는 극성표 (Polarization Table)를 작성한다. 이상의 분산 요구 사항 (빈도표, 분할표 및 극성표)과 광역 데이터 스키마, 논리 접근표 (Logical Access Tables)를 기준으로 데이터 할당 지역을 설정한다. 빈도표는 사용자에게 실행 횟수를 문의함으로써 비교적 용이하게 획득되는 반면, 분할표와 극성표는 설계자의 직관에 의존하여 작성됨으로서 객관성이 미흡하다. Teorey[1989]는 3NF (Normal Form) 광역 개념 스키마에 근거하여, 해당 정보 통신망상의 원격지 사용자의 각 요구 사항을 기준으로 단편화와 비중

복 할당을 수행했다. 트랜잭션 (Transaction) 과 관계형 스키마 (Relational Schema)의 관계를 분석하여 수평 단편화를 했고, 트랜잭션이 이용하는 데이터 크기 (Data Volume)을 일별 튜플 (Tuples/Day)수로 계산하여 특정 단편을 가장 많이 참조하는 곳에 비중복 할당을 하였다. 또한, 원격 질의와 지역 질의에 드는 비용의 차이를 혜택 (Benefit)으로 하고, 특정 단편을 참조하는 모든 원격지 갱신에 드는 비용과의 차이를 계산하여 혜택이 비용보다 큰 경우 중복 할당을 하는 방법을 제안하였다.

이와 같은 기존의 개발된 분산 데이터베이스 설계 방법들은 하기의 설계상 난제가 수반되고 있다. 첫째, 요구 사항을 반영한 특정한 양식의 미비로 요구 사항이 수정될 경우 수정 사항이 즉시 설계에 반영되기 어렵다. 둘째, 일반적으로 사용자는 분산 요구 사항을 구분할 수 없으므로, 설계자가 요구 분석 단계에서부터 분산 설계에 필요한 요구 사항들을 분리 관리해야 한다. 셋째, 분산 설계에서 실질적으로 필요한 분산 설계 정보들을 정확하게 측정하기 어렵다.

한 조직에서의 사용자 요구 사항은 매우 다양하다. Batini는 사용자의 요구 사항은 자연어 (Natural Language), 양식 (Form) 및 레코드 포맷 (Record Formats)등의 세 가지 형태로 표현되어 질 수 있다고 보았다 [Batini, Ceri and Navathe, 1992]. 이러한 세 가지 형태의 요구 사항 중에서 특히 양식은 어떤 조직이 대내외적인 의사소통을 하기 위해, 혹은 조직 내의 업무 처리를 위해 사용하는 가장 일반

적인 수단이다. 여기서 언급하는 양식이란 데이터의 입력과 검색을 지원해 주는 양식화(Formatted)된 변수들의 구조화(Structured)된 집합을 말한다 [Choobineh, 1992]. 양식은 크게 종이로 된 양식과 전산화된 I/O(Input/Output) 양식으로 구분할 수 있는데, 여기서는 두 가지 모두를 포함하는 개념으로 사용하였다. 본 논문에서는 두 가지 양식을 입력 자원으로 하여 분산 데이터베이스를 설계하였다. 이런 양식들을 데이터베이스 설계의 자료로 활용할 수 있는 이유로 Choobineh [1992]는 두 가지를 언급하였는데, 첫 번째로는 현업 실무자들이 각자의 작업에 숙련됨에 따라 그들이 사용하는 양식을 통해 그들이 원하는 거의 대부분의 요구 사항들을 효과적으로 상호교환할 수 있게 되었다는 것이고, 두 번째로는 일반적으로 가장 많이 사용되고 중요하게 관리되어야 할 데이터들을 양식 안에서 찾을 수 있다는 것이다. 그러므로 양식은 개념적 데이터베이스 설계 단계에서 중요한 입력 자원이 될 수 있는 것이다.

Shu[1983]는 양식이 데이터베이스 설계에 관련된 핵심적인 정보를 내포하고 있음에 주목하였다. 하지만, 그가 사용한 양식은 사용자가 사용하는 업무 관련 양식이 아니라 데이터베이스 정보를 명확하게 형식화하기 위해 약속된 양식으로 요구 사항을 기술하는 형식의 양식이었다. Choobineh[1992]는 전산화된 I/O 양식을 기준으로 데이터베이스 설계를 할 수 있는 EDDS (Expert Database Design System)라는 전문가 시스템을 제안하였다. EDDS는

Shu의 방법론과는 달리 전산화된 양식을 사용하며 추론기능, 피드백(Feedback)과 뷰(View) 통합 규칙등을 제공하고 개념 스키마를 도출해 내지만 중앙 집중식 데이터베이스 설계를 위한 지원 시스템이다. 이와 같은 양식을 이용한 데이터베이스 설계에 관한 연구는 매우 활발이 진행되고 있지만 양식을 분산 데이터베이스 설계에 직접적으로 반영한 연구는 현재까지 제시되지 않았다.

양식을 이용한 분산 데이터베이스 설계시는 하기의 장점을 가지게 된다. 첫째, 양식을 사용함으로써 응용 업무를 도출하지 않고도 사용자의 뷰를 쉽게 설계에 반영할 수 있다. 둘째, 양식으로 부터 직접 정량화된 분산 요구 정보를 획득할 수 있으므로 (예, 개체/관계별 튜플 수), 추가의 분산 설계를 위한 요구 분석 없이 설계를 수행할 수 있다. 이상의 장점에 주목하여 본 논문에서는 사용자 요구 분석에서 부터 실제 데이터베이스 구현에 이르기까지의 분산 데이터베이스 설계의 제반 단계에서 필요한 사용자 요구 사항 및 분산 설계 정보를 양식을 이용하여 추출하는 분산 데이터베이스 설계 방법론인 FD³ (Form-Based Distributed Database Design Methodology)를 제시하고자 한다.

II. 분산 데이터베이스 설계 방법론: FD³

FD³의 데이터베이스 설계는 외부 단계, 개념 단계, 분산 단계, 그리고 내부 단계로 크게 수직적인 4단계로 구성된다. 이상의 제반 단계

를 내포한 FD³의 설계 아키텍처가 <그림 2-1>에 묘사되었다. 이상의 4단계는 구체적으로 각각 양식 요구 분석 (Form Requirement Analysis), 스키마 통합 (Schema Integration), 분산 설계, 그리고 지역 논리/물리 설계 단계로 구분한다. 추가로 분산 설계의 입력 자료를 위한 분산 분석 단계가 요구 되며 이단계는 분산 설계 단계 이전 외부 단계 또는 개념 단계 어느 단계와 함께 수행되어도 된다.

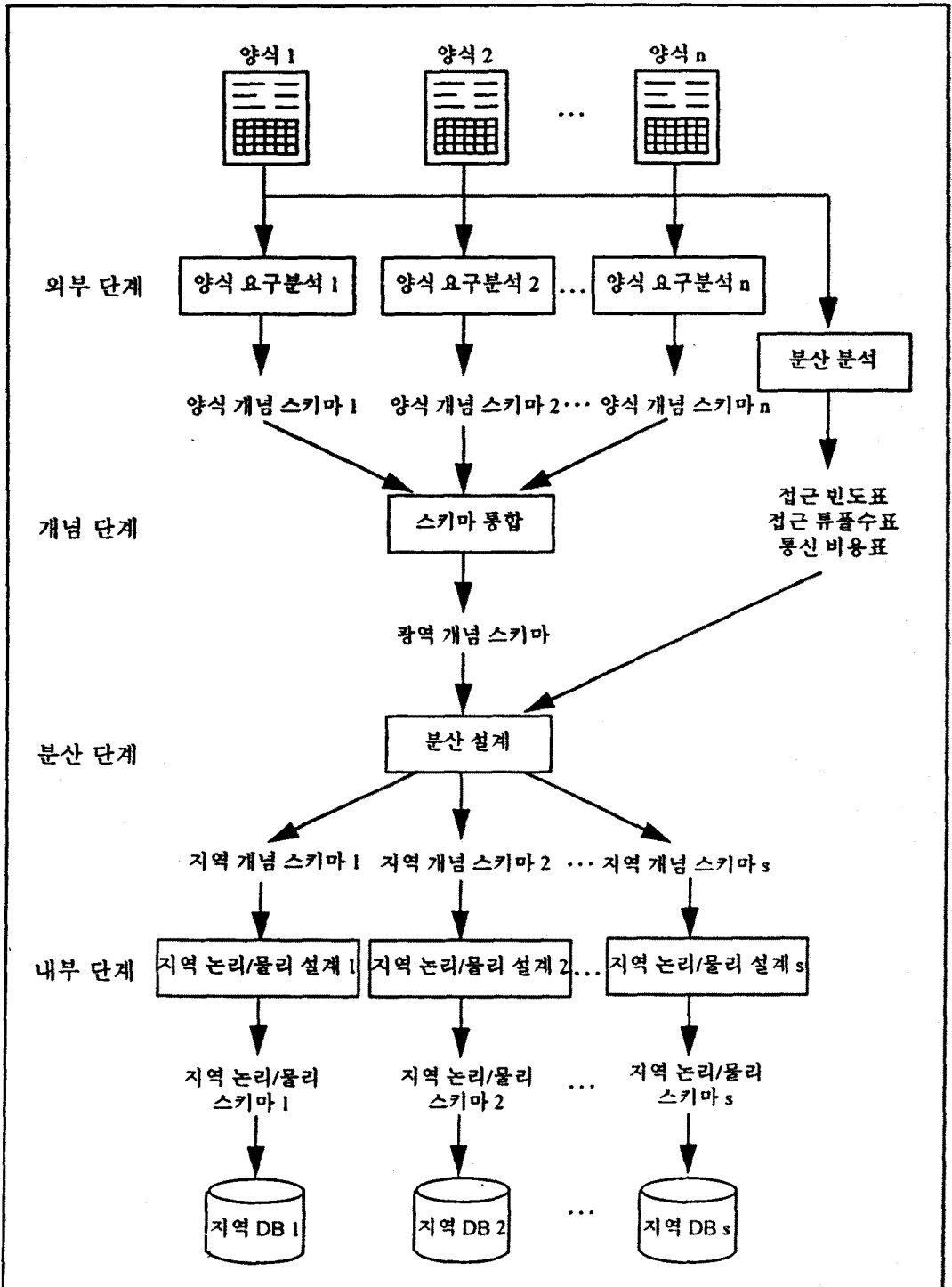
FD³를 이용하여 설계자는 우선 양식을 지역별 사용자 요구 사항으로 간주하여 각 양식별 양식 개념 스키마 (Form Conceptual Schema)를 생성한다. 이상의 양식 개념 스키마를 통합하여 하나의 광역 (Global) 개념 스키마 도출하기 위해서는 스키마 통합이 수행되어야 하는데, FD³를 위하여 간단한 SQL 문을 이용하여 스키마 불일치 중의 이름 상충 (Naming Conflict)을 해소하는 방법이 개발되었다. 한편, 분산 분석에 필요한 분산 요구 사항은 제시된 양식으로 부터 직접 획득된다. 분산 분석에서는 (i) 양식의 지역별 사용 빈도를 가리키는 접근 빈도표 (Access Frequency Table), (ii) 특정 지역에서 한번의 양식 사용 시 광역 개념 스키마에 존재하고 있는 개체에 접근 하는 튜플 갯수를 나타내는 접근 튜플수표 (Access Tuples Table), (iii) 지역간의 통신 비용표 (Communication Cost Table) 등의 분산 정보가 추출된다. 광역 개념 스키마와 분산 분석을 통해 추출된 분산 요구 사항을 이용하여 분산 설계를 수행함으로써 개체를 지역별로 비 중복 (Nonredundant), 그리고 나아가서 중복

할당 (Redundant Allocation)한 후, 그 결과를 사용하여 지역별 개념 스키마를 재구성하여 지역별 논리/물리 설계를 수행한다. FD³의 각 단계별 자세한 기술은 하기와 같다.

II.1. 양식 요구 분석

이 단계에서는 기업 조직에서 사용하는 여러 양식을 사용자 요구 사항으로 간주하여, 양식별 개념 스키마를 작성한다. 현재까지 다양한 형태의 개념 데이터 모델이 제안되었으나 대부분이 실제 현업에 적용하기에 복잡하였다. 그러나, Chen[1976]이 처음으로 제시한 ER (Entity-Relationship) 모델은 개념 설계 단계에서 가장 보편적으로 사용되는 데이터 모델이다. Chen의 ER은 그후 여러 형태로 보완되었으며, 그 중에서도 Teorey는 주 클래스 (Superclass), 부 클래스 (Subclass), 그리고 상속 (Inheritance)의 개념을 추가한 EER (Extended Entity-Relationship) 모델을 제시하여 현재 실제 문제에 광범위하게 적용되고 있다 [Teorey et al., 1986; Teorey, 1991].

FD³에서의 데이터 스키마 (양식 개념 스키마, 광역 개념 스키마, 지역 개념 스키마) 구현은 Teorey가 제안한 EERD (Extended Entity-Relationship Diagram)를 사용한다. FD³는 사용자 요구 사항이 양식의 구체적인 형태로 구현된 후 적용되는 것이다. 이때, 양식을 분석하여 지역별 EERD를 생성하는 과정은 Choobineh가 제안한 EDDS을 사용하거나 데이터베이스 설계 전문가가 직접 양식을 분석하



<그림 2-1> FD³ 설계 아키텍처 (Architecture)

여 양식 개념 스키마를 생성한다.

II.2. 스키마 통합

분산 데이터베이스에서는 이상의 여러 양식 개념 스키마를 논리적으로 하나의 광역 개념 스키마로 통합해야 하는데 이 단계가 스키마 통합이다. FD³의 스키마 통합은 선 통합 (Preintegration), 스키마 비교 (Comparison of the Schema), 스키마 순응 (Conforming the Schema), 그리고 합병 및 재구성 (Merging and Restructuring)의 단계를 거치게 된다 [Batini and Lenzerini,1986].

FD³의 특징은 스키마 순응 중의 이름 상충에 해당되는 이음 동의어 (Synonym)와 동음이의어 (Homonym)를 해결하기 위해 상용 RDBMS (Relational Database Management System)가 이용된 점이다. 양식에 존재하는 모든 필드 (Field)와 양식 이름을 저장한 데이터베이스로 생성하고 SQL 구문을 사용하여 이음 동의어와 동음이의어를 해소하였다.

II.2.1 이름 상충 해소

이음 동의어는 동일한 대상을 각기 다른 이름으로 표현할 때 발생한다. 예를 들면, 한 양식에서는 손님으로 다른 양식에서는 고객으로 표현하는 경우이다. 동음이의어는 상이한 대상을 동일한 필드 이름으로 표현할 때 발생한다. 예를 들면, 한 양식에서는 설비라는 이름이 건물에 부착된 설비를 지칭하고, 다른 양식에서는 컴퓨터의 주변장치의 의미를 표현할 때 발

생한다.

FD³에서는 수 많은 양식 필드들간의 중복성 (Redundancy), 이음 동의어, 동음이의어 문제를 최소화 하기 위해 양식에 존재하는 모든 필드들을 RDB (Relational Database)에 저장하고, SQL 구문으로 동음어를 찾아낸다. 필드를 저장하기 위한 테이블 (본 고에서는 IOFORM이라 칭한다)을 생성하는 SQL 구문은 <그림2-2>와 같이 구성된다. IOFORM 테이블은 양식 이름, 양식 필드 이름, 필드의 이음 동의어, 그리고 필드의 동음이의어 속성 (Attribute)을 가진다. <그림2-2>는 IOFORM 테이블의 데이터 선언문 (Data Definition Language)이다. 각 양식 필드를 RDB의 테이블에 입력하기 위해서는 <그림2-3>과 같은 입력 SQL이 필요하다. 입력한 필드 데이터를 이용하여 동음이의어를 검색하고 분류하는 SQL은 <그림2-4>와 같이 구성된다.

```
create table IOFORM
(FormName char(30) not null,
FieldName char(30) not null,
Entity char(30) null,
Synonym char(30) null,
Homonym char(30) null,
Description char(50) null.)
```

<그림 2-2> 양식 필드 저장 테이블 생성 SQL

```
insert into IOFORM
value(FormName, FieldName, Entity,
Synonym, Homonym, Description)
```

<그림 2-3> 양식 필드 데이터 입력 SQL

```
select      FieldName,  FormName
from        IOFORM
order by    FieldName
```

<그림 2-4> 동음이의어 검색 SQL

여러 양식들의 필드들은 동일 단어 필드별로 그룹 지어 출력된다. 필드들은 특정 순서로 정렬되어 있으므로 데이터베이스 설계자와 사용자는 토론을 거쳐 필드를 분류하고, 동음이의어인 경우 필드 이름을 수정하여 동음이의어 속성에 새로운 이름을 부여한다. 그 후 귀속되어야 하는 개체를 결정한 후, <그림 2-5> 와 같은 SQL 을 이용하여 양식 정보를 갱신한다.

```
update IOFORM
set      Entity = EntityName, Homonym
         = NewFieldName
where    FormName = FormName and
         FieldName = FieldName
```

<그림 2-5> 양식 필드 정보 수정 SQL

이음 동의어의 경우는 이음 동의어 사전을 만들어 해당 필드의 이음 동의어를 모두 검색하여 대표명으로 표현한 후, 귀속되는 개체를 결정해야 한다. <그림 2-6>와 <그림 2-7>은 특정 필드 명과 동일한 의미를 가진 이음 동의어 사전 테이블 (SYNONYM)을 생성하고, 데이터를 입력하는 SQL 구문이다. 하나의 필드는 여러 개의 이음 동의어를 가지므로, 하나의 대표명 (ReprName)에 대해 여러 개의 이음 동의어 필드가 존재할 수 있다. SYNONYMS 테이블의 정보를 사용한 <그림

2-8>과 같은 SQL을 사용하여 이음 동의어를 찾고 하나의 이름으로 표현한 후 귀속될 개체를 결정한다.

```
create table SYNONYMS
(ReprName char(30) not null,
Synonym char(30) null)
```

<그림 2-6> 이음동의어 사전 테이블 생성 SQL

```
insert into SYNONYMS
values(ReprName, Synonym)
```

<그림 2-7> 이음 동의어 사전 데이터 입력 SQL

```
update IOFORM
set      Synonym = ReprName
where    FieldName =
         (select Synonym
          from SYNONYMS)
```

<그림 2-8> 이음 동의어 해소 SQL

II.2.2. 구조 상충 해소

속성, 개체 및 관계들의 상호 연관하의 개념적 불일치를 구조 상충으로 간주한다 [Batini et al., 1992]. 구조 상충 해소란 비교하는 스키마들내에서 동일한 이름이 발견되었을 때 이들이 하나로 합병될 수 있는지를 살펴보는 것이다. 이러한 구조 상충은 동일한 대상을 서로 다른 설계 모델을 택하여 표현한 경우에 발생하게 된다. 예를 들어, 출판이라는 개체와 관련하여 출판사와의 관계를 표현하는 경우, A에서

는 출판사를 출판의 속성으로 표현하고, B에서는 출판사를 하나의 개체로 두고 출판과의 관계를 설정하여 표현하는 경우이다.

지금까지의 연구는 중앙 집중식 데이터베이스 설계와 관련하여 상충된 뷰를 뷰 통합(View Integration) 방법을 거쳐 조정을 하였으나, 최근에는 분산 데이터베이스 설계에 있어서, 기존의 데이터베이스에 표현된 스키마의 상충을 통합하는 문제에 연구가 활발히 진행되고 있다. 이러한 문제는 분산 데이터베이스 관리 시스템이 지원된 경우에 질의 수정 프로시저(Query Modification Procedure)를 광역 스키마에 저장함으로써 해결될 수 있다 [Ceri et al., 1987; Deen et al., 1987; Reddy et al., 1994]. 그러나, 본 연구에서 이름 상충을 SQL 문을 사용하여 해소한 방법과 마찬가지로, 질의 수정 프로시저를 사용하지 않고도 개체와 속성간의 구조 상충을 SQL 구문으로 검색할 수 있다. 즉, 이름 상충 해소에서는 양식 필드만을 사용하여 상충 해소를 시도했으나, 이름 상충 해소에서 입력된 필드들의 개체 이름과 양식 필드 이름들을 SQL 문으로 검색한다면, 구조 상충이 용이하게 발견될 수 있다.

II.3. 분산 분석

분산 설계에 필요한 분산 요구 사항은 분산 분석 단계에서 취합된다. 분산 분석 단계에서는 양식을 분석하여 접근·빈도표, 접근 튜플수표, 지역간의 통신 비용표의 분산 설계 정보를 추출한다.

양식별 지역별 접근 빈도표는 일정 기간 동안의 지역별 해당 양식의 사용 횟수, 실제로는 트랜잭션 횟수를 가리키는데, 이 정보는 해당 지역 사용자로 부터 획득될 수 있다. 양식별 지역별 개체별 접근 튜플수표는 지역별로 해당 양식을 사용할 때 한번의 트랜잭션 발생 시에 가져오게 되는 개체별 사용 튜플수를 가리킨다. 이 정보는 양식의 개체별 연관도(Cardinality)로부터 용이하게 얻어지는데 이것은 한번의 트랜잭션으로 화면에 출력할 수 있는 개체의 튜플 최대 갯수와 동일하다. 지역간 통신 비용표는 한 개의 튜플을 질의 하거나 갱신하는데 소요되는 비용을 가리키는데, 본 연구에서는 튜플은 통신 패킷의 단위로 하여 튜플당 지역간의 전송 비용을 도출한다. FD³에서는 광대역 통신망(WAN; Wide Area Network)상에 존재하는 지역에 개체를 할당하는 것으로 간주한다. 근거리 통신망(LAN; Local Area Network) 환경과는 달리 WAN 환경하에서의 분산 설계는 일반적으로 통신 비용을 최소화하는 문제로 귀결되고 있으며 [Sheng and Lee, 1992], FD³에서도 통신 비용을 최소화하는 측면에서 개체가 할당되었다.

II.4. 분산 설계

통합된 광역 개념 스키마를 각 지역별로 최적의 분산을 하기 위해서는 분산 설계가 수행되어야 한다. 할당의 단위는 광역 개념 스키마 내의 개체이며, 분산 분석에서 도출된 분산 정보에 근거하여 개체를 비중복 및 중복 할당 한

후, 지역별 개념 스키마를 생성한다.

FD³에서의 분산 설계는 다음과 같은 3단계로 나누어 수행된다.

(i) 수평 단편화 (Horizontal Fragmentation) 및 비중복 할당 (Nonredundant Allocation);

(ii) 중복 할당 (Redundant Allocation);

(iii) 지역 개념 스키마 재구성 (Reconstruction of Local Conceptual schemata).

II.4.1. 수평 단편화 및 비중복 할당

FD³에서는 할당의 대상이 되는 개체를 사용하기 위한 원격 통신 비용이 최소인 지역에 그 개체를 비중복 할당 한다. 특정 지역에 개체가 할당되어 있을 때, 타 지역에서 양식을 사용한 데이터 접근을 수행하는 경우, 정보 통신망 (Data Communication Network)을 사용하여야 하는데 이 경우 통신 비용이 소요되게 된다. 따라서, 어떤 한 지역에 해당 개체가 할당 되었을 경우 타 지역에서 소요해야 하는 원격 질의, 통신 비용을 계산하여 그 비용이 최소인 지역에 개체를 할당한다.

한편, 한 개체 내의 튜플들이 지역별로 요구되어 이들 튜플들이 해당 지역에 의존적이며 상호 배타적인 경우, 개체 전체가 아닌 개체의 일부 튜플을 할당의 대상으로 하기 위해 개체를 수평 단편화해야 한다. 예를들어, 판매이라는 개체의 튜플들이 지역별로 분산되어 요청되고, 그 튜플들은 해당 지역에서만 사용된다고 할때, 이 개체는 개체 전체가 아닌 튜플별로 할당되어야 한다. 즉, 개체가 분할되어 일부 튜플

로 구성된 개체 일부분이 할당의 단위가 된다. 이 경우 편의상 개체 일부분을 하나의 개체로 간주하기로 한다.

분산 설계에 사용되는 개체 할당 모형의 각 매개변수 (Parameter)는 다음 표기법을 이용하여 표시한다.

S : 네트워크상의 지역들의 집합

F : 네트워크상의 양식들의 집합

E : 네트워크상의 개체들의 집합

j, n : 특정 지역, 즉, $j, n \in S$

i : 특정 양식, 즉, $i \in F$

k : 특정 개체, 즉, $k \in E$

f_{ij}^a : 지역 j에서의 양식 i의 일별 질의 (Query) 횟수

f_{ij}^u : 지역 j에서의 양식 i의 일별 갱신 (Update) 횟수

t_{ij}^a : 개체 k에 대한 지역 j에서의 양식 i의 질의별 접근 튜플 갯수

t_{ijk}^a : 개체 k에 대한 지역 j에서의 양식 i의 갱신별 접근 튜플 갯수

c_{in}^a : 지역 j와 지역 n간의 원격 질의 통신 비용 (원/튜플)

c_{in}^u : 지역 j와 지역 n간의 원격 갱신 통신 비용 (원/튜플)

f_{ij}^a, f_{ij}^u 는 사용자로부터 획득되는 것으로 지역 사용자들의 양식에 대한 일별 질의 혹은 갱신 횟수, 즉, 양식 조회/갱신 트랜잭션이 발생하는 횟수를 가리킨다. t_{ijk}^a, t_{ijk}^u 는 특정 지역에서 양식 조회/갱신 트랜잭션이 발생할 때 특정

개체에 접근하는 튜플 갯수이다. 예를 들어, 서울에서 발주 현황 조회 라는 양식을 이용하여 발주 현황을 조사하고자 할 때, 해당 양식에서는 제품, 고객등의 개체를 병합하여 필요한 정보를 구성하게 된다고 하자. 이때, 양식에서는 고객 한 사람에 대한, 최대 5개의 제품의 발주 현황만을 출력할 수 있다면, 개체 고객에 대한 지역 서울에서의 양식 발주 현황 조회의 질의 별 접근 튜플 갯수는 1이고, 개체 제품에 대해서는 5가 된다. c_{in}^a 와 c_{in}^u 는 지역간의 질의 및 갱신에 소요되는 통신 비용을 말한다. 통신 비용은 거리에 비례하는 통신 설비 이용료를 가리키는데 갱신의 경우, 데이터 갱신이 추가로 발생하여 이에 따른 전송 지연 (Transmission Delay) [Stalling, 1994]이 소요됨으로써, 질의에 비해 통신 비용은 커지게 된다.

개체 k를 지역 j에 할당할 경우 개체 k에 접근하기 위해 필요한 질의 및 갱신 통신 비용 C_k 은 다음과 같이 나타낼 수 있다.

$$C_k = \sum_{i \in \{a, e, r, v, k\}} \sum_{n \neq j} f_{in}^a t_{ink}^a c_{in}^a + \sum_{i \in \{u, d, t, e, s, k\}} \sum_{n \neq j} f_{in}^u t_{ink}^u c_{in}^u$$

따라서, 개체 k는 다음과 같은 지역에 비중복 할당 된다.

II.4.2. 중복 할당

중복 할당은 비중복 할당된 지역 외에 해당

개체가 할당될 수 있는 추가적인 지역을 선정하는 것이다. 즉, 특정 지역에 개체를 중복 할당 함으로써 얻어지는 혜택이 중복에 드는 비용을 초과할 때 중복 할당을 결정한다.

혜택은 원격 질의를 하지 않고 지역 질의를 함으로써 얻게 되는 원격 질의 통신 비용 절감을 의미하며, 비용은 그 지역에 개체를 새로이 하나 중복함으로써 발생하는 원격 갱신 통신 비용을 의미한다.

m : 개체 k가 비중복 할당된 지역, $\min \in S$

c_{jm}^a : 지역 j와 지역m간의 원격 질의 통신 비용(원)

c_{pm} : 지역 j와 지역n간의 원격 갱신 통신 비용(원)

m 은 특정 개체 k가 비중복 할당된 지역이나 고려 대상이 되는 지역 j에 개체 k를 두게 될 경우에는, m 으로 원격 질의를 해야 하는 통신 비용이 절감되는 혜택을 가지게 되나, 그와는 반대로, 개체 k에 대해 지역 j 이외의 지역의 갱신이 수행되어야 하므로, 원격 갱신 비용이 추가로 소요된다.

따라서, 개체 k를 지역 j에 추가로 할당함에 따르는 혜택과 비용의 차이 B_k 는 다음과 같이 표현될 수 있다.

$$B_k = \sum_{i \in \{a, e, r, v, k\}} f_{ij}^a t_{ijk}^a c_{jm}^a$$

$$- \sum_{i \in \text{updates } k} \sum_{n \neq j} f_{ij}^u t_{ijk}^u c_{jm}^u$$

결과적으로, 혜택이 비용을 초과할 때, 즉, $B_k > 0$ 일 때 지역 j 에 개체 k 를 중복 할당한다.

II.4.3. 지역 개념 스키마 재구성

지역 개념 스키마 재구성 단계에서는 광역 개념 스키마, 중복 할당 및 비중복 할당에 의한 개체의 할당될 지역을 고려하여 각 지역별 개념 스키마를 재구성한다.

비중복 및 중복 할당에서는 개체만이 할당의 대상이었으며, 개체와 개체간의 관계는 고려 대상이 아니었다. 그러나, 개체간에 일대다 (1:M)와 일대일 (1:1)의 관계가 존재하는 경우 논리 개념 스키마로 변환 시에 일대 다의 관계에서는 일 (One)측의 기본 키 (Primary Key)가 다 (Many)측의 외래 키 (Foreign Key)가 되며, 일대 일에서는 둘 중의 어느 한 개체가 상대방의 기본 키를 자신의 외래 키로 가지게 된다. 그러므로, 해당 지역에 할당된 개체와 해당 지역에 할당되지 않았지만 현 지역의 개체와 외래 키로 연관되는 개체를 지역별 개념 스키마에 개념적으로 도식화할 필요가 있다. FD³에서는 타 지역에 할당된 개체의 기본 키를 현 개체의 외래 키로 연관시켜야 하는 경우 그 개체와 그 개체와의 관계는 점선으로 표시하며, 속성은 외부 키만을 나타내어 추후 지역 논리/물리 설계 단계에 이용되도록 하였다.

II.5. 지역 논리/물리 설계

이상의 비중복 및 중복 할당의 결과는 지역/데이터 상관도 (Location/Data Matrix)로 용이하게 표시될 수 있다 [Flaatten et al., 1992]. 지역 논리 설계는 각 지역별 개념 스키마를 DBMS에 의해 실행될 수 있는 데이터베이스 구조로 기술하는 것이다. 논리 스키마를 기술하기 위해서는 논리 모델 (Logical Model)을 사용해야 하는데, 관계 (Relational), 네트워크 (Network), 그리고 계층 (Hierarchical) 모델 등이 있다. 논리 설계는 DBMS가 사용하는 논리 모델에 따라 상이하게 전개된다.

지역 물리 설계는 지역 논리 설계의 산출물인 논리 스키마를 특정 DBMS에 적합한 형태로 표현한다. 또한, 효과적으로 데이터를 사용하기 위해 인덱스 (Index), 포인터 (Pointer), 레코드 클러스터링 (Record Clustering), 데이터베이스 저장 장치의 분할 (Partitioning of Database Storage), 레코드 연결 (Record Linkage), 그리고 버퍼링 (Buffering) 등의 데이터 저장 구조 (Storage Structure)와 접근 방법 (Access Method)을 결정한다. 물리 설계에서는 성능 향상을 위해 조정 (Tuning)을 하게 되는데, 이러한 것이 논리 스키마의 구조에 영향을 주므로 물리 설계와 논리 설계는 상호 보완 관계에 있다.

Ⅲ. 사례 분석

Ⅲ.1. 사례 개요 및 범위

본 장에서는 유통업계에서는 국내에서 최대 규모 중의 하나인 (주)한화유통의 유통 및 재고 분산 데이터베이스 개발에 FD³ 방법론이 적용된 사례를 소개하고자 한다. (주)한화유통은 서울에 본사를 두고 있으며, 경인, 부산, 서울에 3개의 물류센타를 두어 각 지역의 배송 업무를 수행하고 있다. 그외 전국 각 지역에 50여개의 점포를 가지고 있다. 이전에는 본사에 중앙 집중식 데이터베이스를 운영하여 유통 및 재고의 본부 업무만이 정보화된 형태였다. 향후 POS (Point of Sales) 및 EDI (Electronic Data Interface) 도입을 위한 기업 전략적 측면에서 본사, 물류센타 및 점포를 연결한 분산 데이터베이스 시스템 개발에 주력하고 있다. 실제의 유통 및 재고 업무 관련 양식은 200여 개에 달했으며 유통 및 재고 분산 DB 설계를 완료한 시점에서는 50여 개의 개체, 수백 개의 속성, 수십 개의 관계들이 나타났지만 본 논문에서는 이해를 돕기 위해 발주업무와 관련된 4개의 양식만을 사용하여 예시함으로써 FD³ 방법론의 유용성을 보이고자 한다.

Ⅲ.2. FD³ 적용 절차

Ⅲ.2.1. 양식 요구 분석

발주라는 업무는 상품을 취급하는 거래선에

판매에 필요한 상품을 특정 수량만큼 주문하는 행위이다. 발주를 하기 위해서는 발주 당사자인 점포와 발주의 대상이 되는 상품, 상품을 거래하는 거래선, 그리고 발주 업무를 수행하는 발주등의 각각에 대한 정보, 즉, 해당 양식들을 가지고 있어야 한다.

이들 네가지의 양식이 <그림3-1>에서 <그림3-4>까지 예시되었다. 점포 정보 설정 화면은 상품을 주문하는 주체가 되는 점포에 관한 정보를 입력하는 양식이다. 이 양식은 타 개체와의 관계는 존재하지 않으므로 양식 개념 스키마에는 점포라는 개체와 그 개체의 속성들만이 표현된다. 거래선 마스터는 점포의 상품 발주 대상이 되는 거래선에 관한 정보를 입력하는 양식이다. 이 양식도 거래선에 관한 정보만을 포함한다. 상품 마스터는 발주 업무의 중심이 되는 거래 대상인 상품에 관한 정보를 입력하는 양식이다. 양식 하단의 발주처코드, 원가, UNIT입수 및 보충수량에는 특정 상품에 대해 발주가 가능한 거래선이 최대 3개까지 표시된다. 따라서, 특정 상품은 1개 이상의 발주 가능 발주처를 가지며, 특정 발주처는 1개 이상의 발주 가능 상품을 가질 수 있는 다대다 (M:M) 관계를 갖고 개체화 된다. 점포 발주 관리의 발주 업무를 수행하는 양식이다. 발주는 점포, 상품, 거래처의 정보가 연관되어 하나의 발주 인스턴스 (Instance)를 유발하게 된다. 발주 인스턴스 한 개는 특정 점포, 특정 상품, 특정 거래처 및 특정 일자에만 발생 가능하다. 따라서, 발주 개체는 점포, 상품, 거래처와 각각 다대일 (M:1) 관계를 가지며, 발주 일자,

수량 및 매가 속성을 가져야 한다.

이들 네 양식들은 <그림 3-5>, <그림 3-6>, <그림 3-7>, <그림 3-8>과 같은 EER 모델의 양식 개념 스키마로 각각 표현될 수 있다. 양식 요구 분석 단계에서 한 개의 양

식은 한 개의 양식 개념 스키마로 도식화된다. 즉, 양식 하나는 사용자 뷰 하나로 간주되었다. 양식 그 자체가 사용자 요구 사항이므로 인터 뷰와 같은 기존 방법에 비해 사용자 요구 사항 기술의 오류를 줄이기에 용이하다고 하겠다.

점포정보설정

점포코드	1820	잠실점		
사업자번호	12345-67890	총인원	50 명	점포면적 300 m2
주 소	서울시 송파구 잠실			
전화번호	123-4567			
점장명	홍길동			
개점일		폐점일		
지역현황				

<그림 3-1> 점포정보 설정 화면

거래선 마스타

거래선코드	거래선명	대표자명	주 소	전화번호
457000	제일제당서울지사	전 도안	서울시 서대문구	222-2222
464200	한국존슨(주)	노 태오	서울시 동대문구	444-3333

<그림 3-2> 거래선 마스타 화면

상품 마스타

상품코드 1002676

상품명 제일비트 1.5KG

단위 EA 단가 100 매가 110

원가 90 재고량 50 발주종료일

입수 1 용량 1.5 KG

발주처코드	원가	UNIT입수	보충수
457000	90	5	

<그림 3-3> 상품 마스타 화면

점포 발주 관리

점포코드 1820 잠실점

발주일자 95/06/01

상품코드	수량	상품명	거래처	매가	입수
1122206	10	해태고향만두100입	해태제과서울지사	3,000	1.0
1119725	5	해태불고기피자380	해태제과서울지사	4,300	1.0

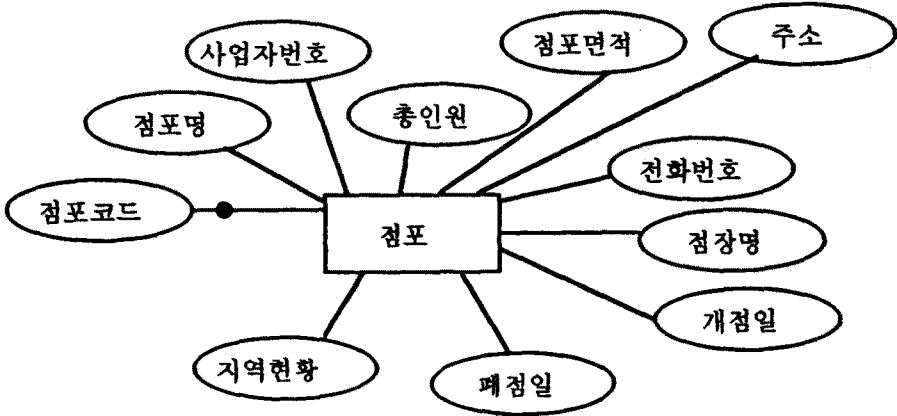
<그림 3-4> 점포 발주 관리 화면

Ⅲ.2.2. 스키마 통합

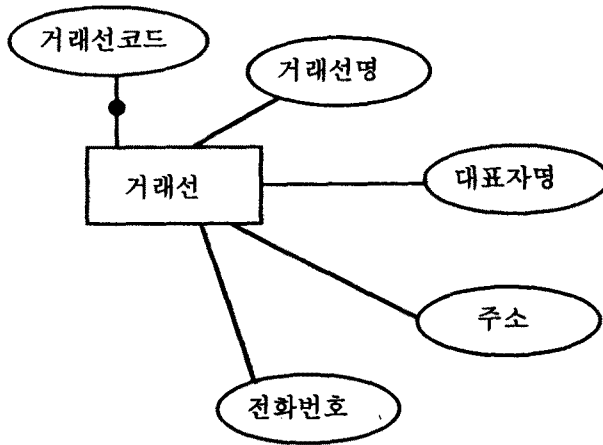
본 스키마 통합 단계에서는 4개의 양식 개념 스키마들을 동시에 비교하여 이름 상충 및 구조 상충 여부를 판단한 후 스키마 순응을 하였다. 우선 이름 상충을 해소하기 위해서, 양식에 존재하는 모든 필드, 양식 이름을 RDB에 저장

한 후, SQL 문장을 이용하여, 이음 동의어와 동음이의어를 제거하였다.

동음이의어의 예로서는 상품마스타내의 원가와 상품마스타의 하단의 원가이다. 앞의 원가는 상품에 대해 고유하게 가지는 원가 정보이며, 하단의 원가는 제공하는 거래선별 상품



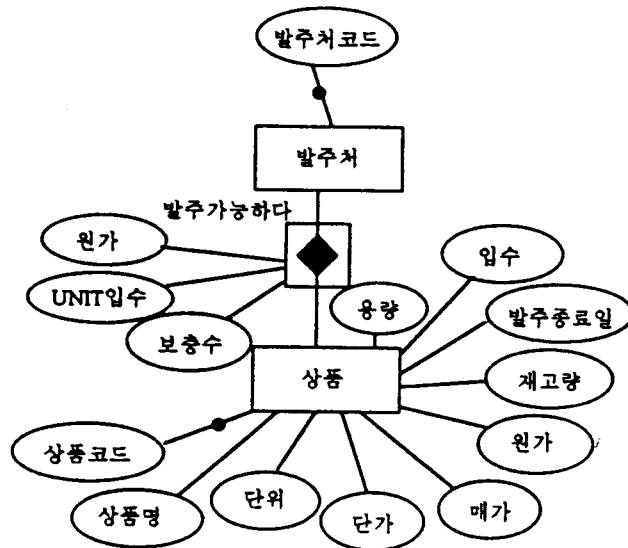
<그림 3-5> 점포 정보 설정 양식 개념 마스타



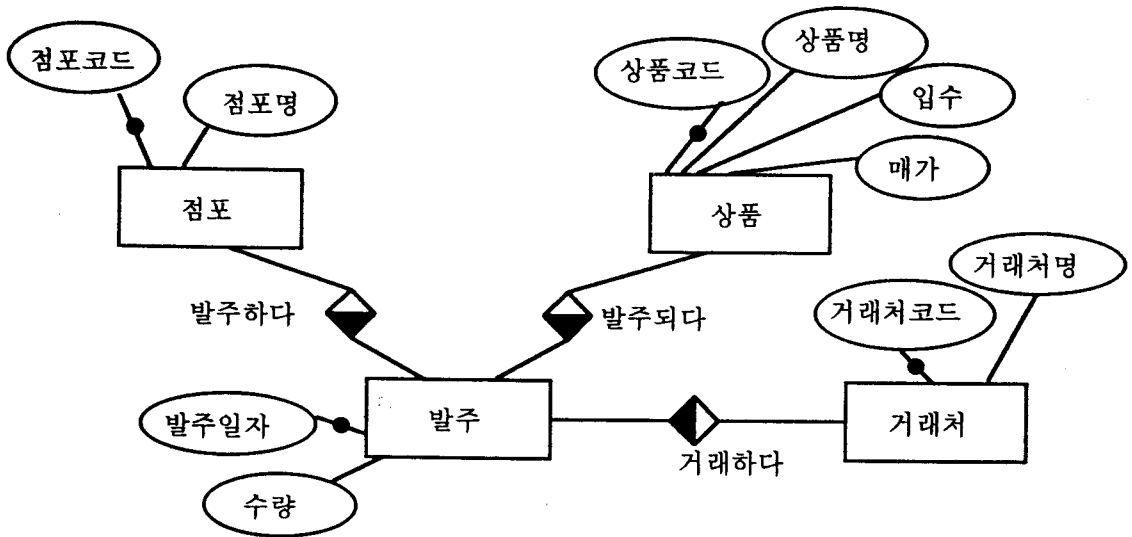
<그림 3-6> 거래선 마스타 양식 개념 스키마

제공 원가이다. 따라서, 이들은 <그림 3-9>에 예시된 것과 같은 SQL 구문에 의하여 재명명되었다. IOFORM의 키는 FormName와 FieldName으로 구성된 종합키 (Combination Key)이므로 동일 양식에서 동음이의어가 발생할 경우는 데이터가 입력되지 않는다. 따라

서, <그림 3-9(b)>와 같이 원가를 원가1, 원가2로 분류하여 입력해야 한다. <그림 3-9(c)>의 문장들이 수행되면, 동일 필드끼리 오름차순으로 정렬되어 출력된다. 이 데이터를 데이터베이스 설계자와 사용자간의 의견 수렴을 통해, 동음이의어로 판단된 경우 새로운 이



<그림 3-7> 상품 마스터 양식 개념 스키마



<그림 3-8> 점포 발주 관리 양식 개념 스키마

를 부여하고, 귀속 개체를 결정한 후, <그림 3-9(d)>와 같은 문장을 이용하여 양식 정보를 갱신했다.

```
create table IOFORM
```

(FormName	char(30)	not null,
FieldName	char(30)	not null,
Entity	char(30)	null,
Synonym	char(30)	null,
Homonym	char(30)	null,

Description char(30) null)

(a) 양식 필드 저장 테이블 생성 SQL

```
insert into IOFORM
values(상품마스타, 원가 1, , , )
```

```
insert into IOFORM
values(상품마스타, 원가 2, , , )
```

(b) 양식 필드 데이터 입력 SQL

```
select FieldName, FormName
from IOFORM
order by FieldName
```

(c) 동음이의어 검색 SQL

```
update IOFORM
set Entity=거래선, Homonym =
                                     상품제공원가
where FormName = 상품마스타 and
                                     FieldName= 원가2
```

(d) 양식 필드 정보 수정 SQL

<그림 3-9> 동음이의어 해소 SQL의 적용예

한편, 이음 동의어에는 거래선, 발주처 및 거래처가 해당되며, 이들에 대한 대표명을 거래선코드로 하여 이름 상충을 해소할 수 있다. SQL 구문을 예시하면 <그림 3-10>과 같다.

```
insert into IOFORM
values(거래선마스타, 거래선코드, , , )
insert into IOFORM
values(상품마스타, 발주처코드, , , )
```

(a) 양식 필드 데이터 입력 SQL

create table SYNONYMS

```
(ReprName char(30) not null,
Synonym char(30) null)
```

(b) 이음 동의어 사전 테이블 생성 SQL

```
insert into SYNONYMS
values(거래선코드, 발주처코드)
```

(c) 이음 동의어 사전 데이터 입력 SQL

```
update IOFORM
set Synonym = 거래선코드
where FieldName =
(select Synonym
from SYNONYMS)
```

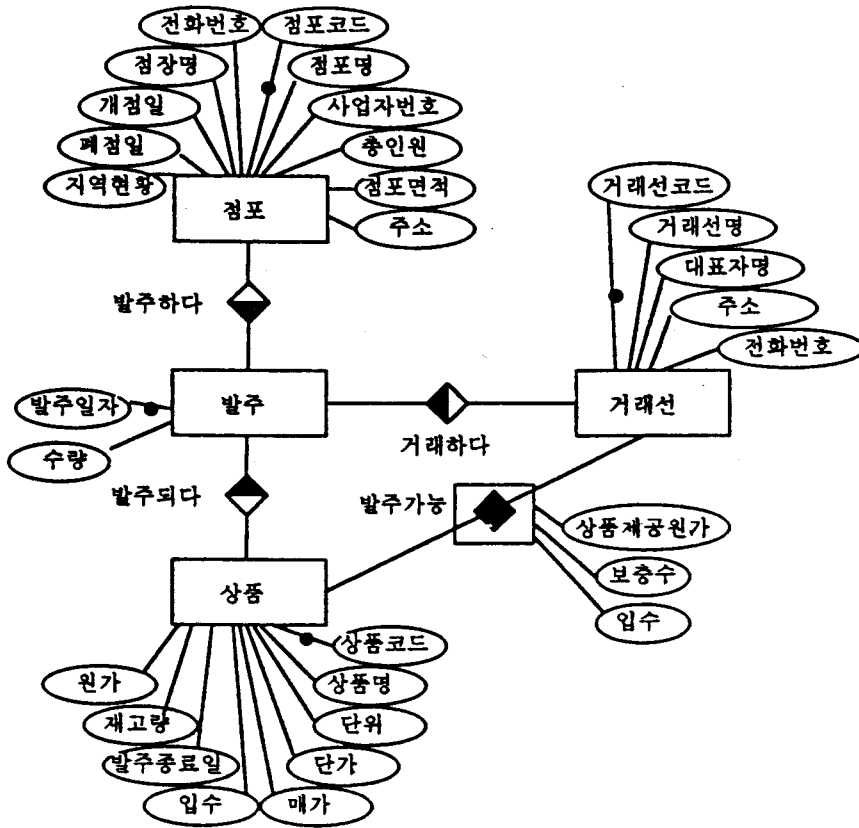
(d) 이음 동의어 해소 SQL

<그림 3-10> 이음 동의어 해소 SQL의 적용예

이상의 스키마 통합 단계를 거쳐 도출된 광역 개념 스키마가 <그림 3-11>에 도식화 되었다. 참고로 양식별 사용 개체를 광역 뷰의 관점에서 살펴보면 <그림 3-12>과 같다. 예를 들어, 상품 마스타 양식이 상품과 거래선 개체를 사용하고 있는 것을 양식과 광역 개념 스키마를 통해 알 수 있다.

III.2.3. 분산 분석

분산 설계에 사용되는 분산 정보는 다음과 같은 방법으로 분산 분석 단계에서 획득될 수 있다. 개체는 스키마 통합에서 도출된 광역 개념 스키마에서 용이하게 파악된다. 이들 개체



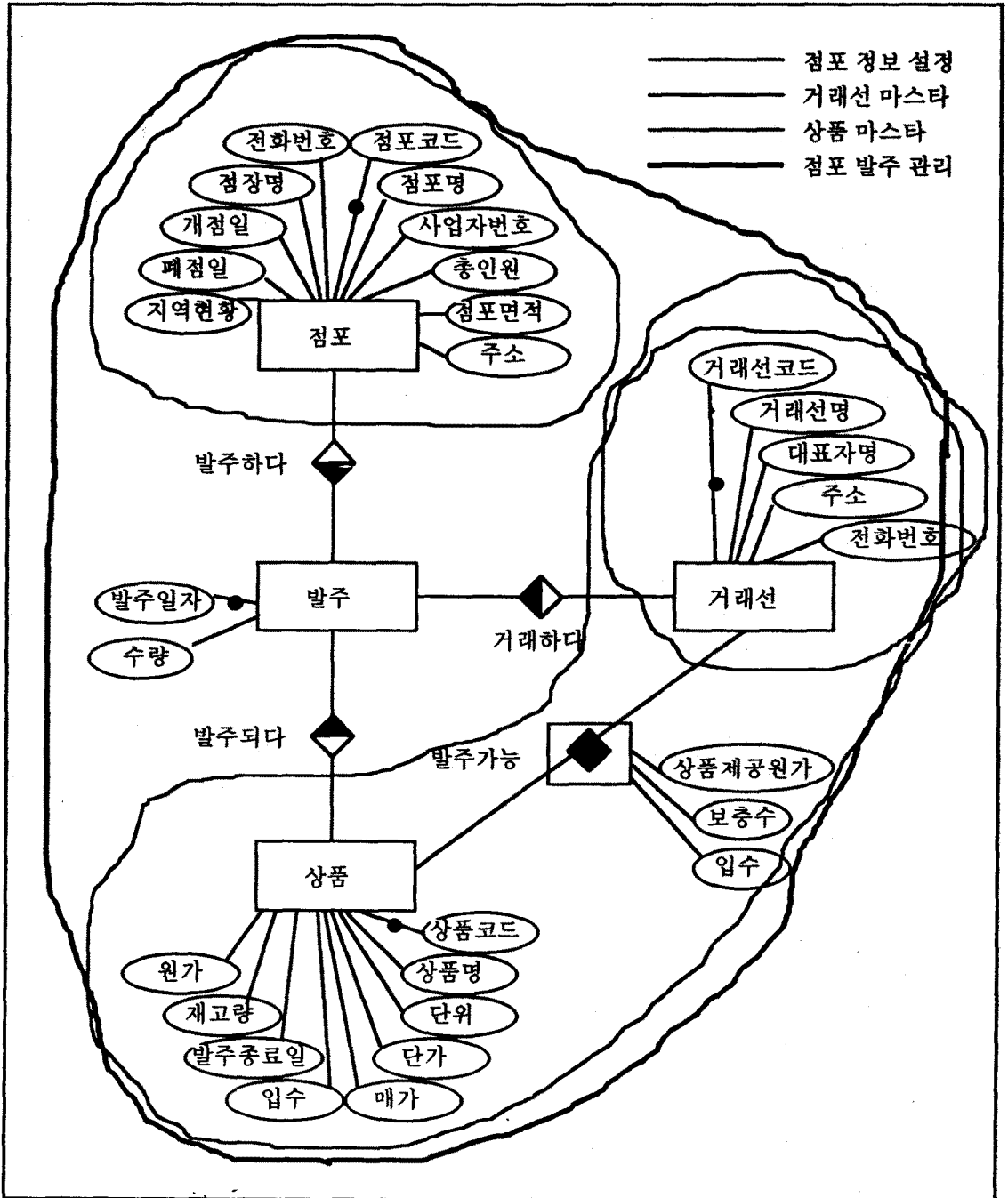
<그림 3-11> 광역 개념 스키마

가 주 할당 대상이다 (<그림3-11> 및 <그림3-12> 참조). 양식별 지역별 접근 빈도표는 해당 지역 담당자로부터 획득했다. 이 접근 빈도표가 <표3-1>과 같이 정리되었다. 접근 튜플수표는 양식에 나타난 개체별 연관도로 부터 도출하였다. 예를 들어, 지역 서울에서 거래선 마스터 양식을 한번 사용할 때 개체 거래선에 대한 질의와 갱신 모두 최대 10개의 정보를 가져오거나 보낼 수 있으므로, 접근 튜플수는 10이 된다. 이것은 한번에 트랜잭션으로 화면에 출력할 수 있는 개체 튜플의 최대 갯수와 동

일하다. 이상의 접근 튜플수표가 <표3-2>에 예시되었다. 마지막으로, 지역간 통신 비용은 한 개의 튜플을 질의 또는 갱신하는데 소요되는 비용으로 현재 (주)한화유통에서 구축되어 사용하는 정보 통신망을 분석하여 측정되었다. 이 통신 비용이 <표3-3>에 예시되었다.

III.2.4. 분산 설계

이상의 광역 개념 스키마에 포함된 4개의 개체 중 어느 개체도 지역별로 일부 튜플만이 요청되는 경우가 아니므로 개체의 수평 단편화가



<그림 3-12> 광역 뷰로써의 양식별 사용 개체

<표 3-1> 접근빈도표(Access Frequency Table)

(단위: 횟수/일)

지역 \ 양식	서 울		대 전		부 산	
	질의	갱신	질의	갱신	질의	갱신
점포 마스타	50	30	10	10	5	15
거래처 마스타	80	50	70	10	50	20
상품 마스타	200	10	100	10	150	30
점포 발주 관리	30	20	40	10	50	10

<표 3-2> 접근 튜플수표 (Access Tuples Table)

(단위: 갯수/양식)

지역 \ 양식	개체	점 포		상 품		거 래 선		발 주		발주가능	
		질의	갱신	질의	갱신	질의	갱신	질의	갱신	질의	갱신
서 울	점포 마스타	1	1	0	0	0	0	0	0	0	0
	거래처 마스타	0	0	0	0	10	10	0	0	0	0
	상품 마스타	0	0	1	1	3	0	0	0	3	3
	점포 발주 관리	1	0	10	0	10	0	10	10	0	0
대 전	점포 마스타	1	1	0	0	0	0	0	0	0	0
	거래처 마스타	0	0	0	0	9	9	0	0	0	0
	상품 마스타	0	0	1	1	2	0	0	0	2	2
	점포 발주 관리	1	0	9	0	9	0	9	9	0	0
부 산	점포 마스타	1	1	0	0	0	0	0	0	0	0
	거래처 마스타	0	0	0	0	9	9	0	0	0	0
	상품 마스타	0	0	1	1	2	0	0	0	2	2
	점포 발주 관리	1	0	9	0	9	0	9	9	0	0

<표 3-3> 통신 비용표 (Communication Cost Table)

(단위: 원/튜플)

지역 \ 지역	서 울		대 전		부 산	
	질의	갱신	질의	갱신	질의	갱신
서 울	0.0	0.0	1.0	4.0	3.0	5.5
대 전	1.0	2.5	0.0	0.0	2.0	4.0
부 산	3.0	5.5	2.0	2.5	0.0	0.0

추가로 요구되지는 않았다. 그러므로, FD³ 에서의 할당의 단위는 각 개체이며, 이들에 대한 비중복 할당이 선행된다. 지역별 해당 개체에 대한 원격 질의 및 갱신 비용은 <표3-4>와 같다.

비중복 할당 결과 개체 점포는 지역 서울, 개체 상품은 지역 대전, 개체 거래선은 지역 서울, 개체 발주는 지역 대전, 개체 발주가능은 지역 서울에 할당된다.

추가 중복 할당 가능성 분석을 위한 혜택 대비 비용 차액분을 정리하면 <표3-5>와 같다. 혜택 대비 비용 차액분이 양수일때 개체가 중복 할당되므로, 개체 상품은 지역 서울 부산, 개체 거래선은 지역 부산, 개체 발주가능은

지역 부산에 중복 할당된다. 이상의 비중복 할당을 종합하여 지역/데이터 관계도를 작성하면 <표3-6>과 같다. 지역/데이터 관계도의 F는 전체 개체 (Full Copy)를 의미한다. 수평 단편화의 경우 개체의 일부가 할당되므로 개체 일부 (Partial)를 의미하는 P가 사용될 수도 있다.

이상의 중복 및 비중복 할당을 근거로 한 서울, 대전 및 부산 3지역에 대한 지역별 개념 스키마를 완성한 결과가 <그림3-13> <그림3-14> 및 <그림3-15>이다. 특히, 개체가 물리적으로 할당되지 않아도 개체간의 관계가 예시되어야 하는 경우가 외부키의 존재 여부이다. 예를들어, 대전의 경우 점포와 거래선

<표 3-4> 원격 질의 및 갱신 통신 비용표

(단위: 원/튜플)

지역 \ 지역	점 포	상 품	거 래 선	발 주	발주가능
서 울	* 322.5	2450.0	* 6005.0	2430.0	* 1480.0
대 전	325.0	* 1845.0	6070.0	* 2060.0	1515.0
부 산	545.0	2515.0	10590.0	3080.0	2445.0

<표 3-5> 중복 할당을 할 경우의 혜택 대비 비용 차액분(Bjk)

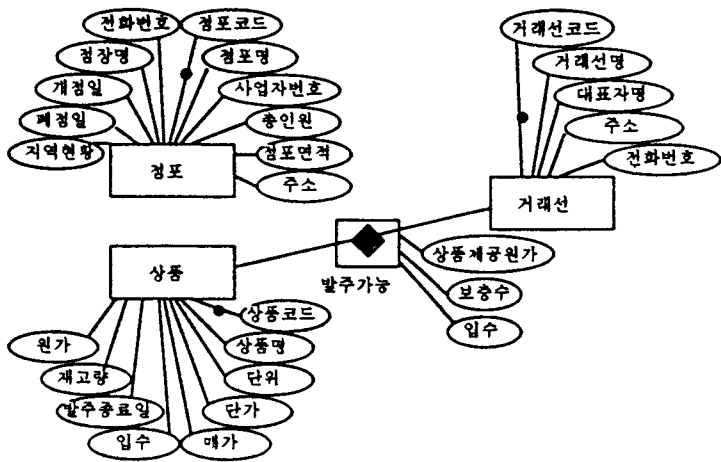
(단위: 원/튜플)

지역 \ 지역	점 포	상 품	거 래 선	발 주	발주가능
서 울	*	310.0	*	-420.0	*
대 전	-85.0	*	-780.0	*	-115.0
부 산	-40.0	1105.0	490.0	-560.0	655.0

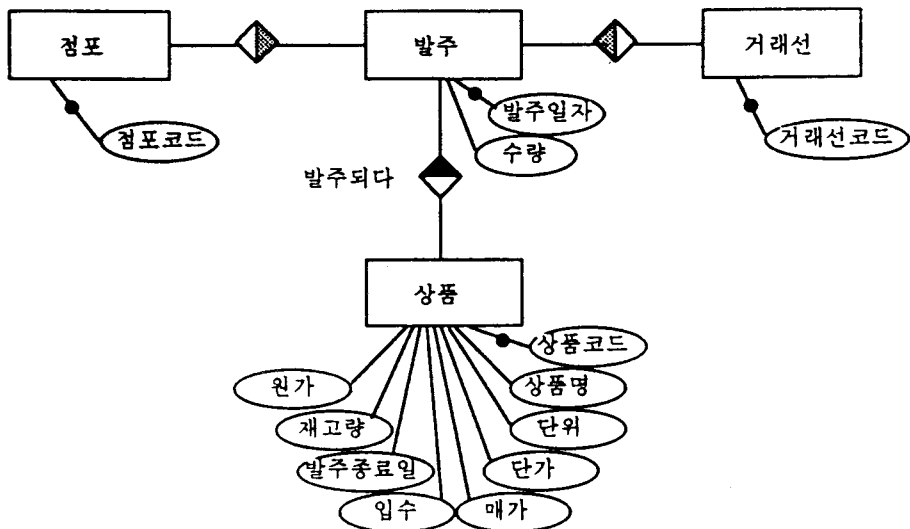
(*는 개체가 비중복 할당된 지역)

<표 3-6> 지역/데이터 관계도

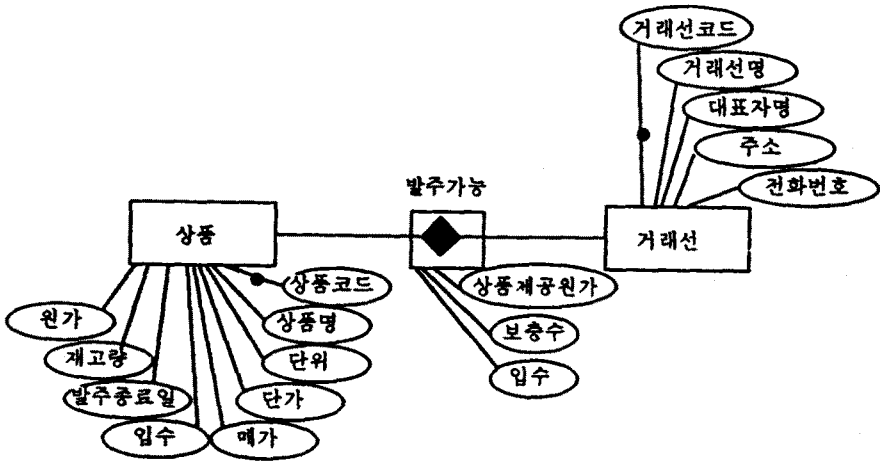
지역 \ 지역	점 포	상 품	거 래 선	발 주	발주가능
서 울	F	F	F		F
대 전		F		F	
부 산		F	F		F



<그림 3-13> 서울의 지역 개념 스키마



<그림 3-14> 대전의 지역 개념 스키마



<그림 3-15> 부산의 지역 개념 스키마

개체가 대전에 물리적으로 할당되지는 않지만, 개체 점포의 점포 코드와 개체 거래선의 거래선코드를 개체 발주가 외부 키로 가지므로 개념 스키마에 이들이 점선으로 표시되었다.

지역 논리 설계는 개념 스키마를 논리 모델로 치환하는 과정이다. 본 사례에서는 관계형 모델이 적용되었으며 그 결과의 지역별 논리 스키마가 <그림3-16> <그림3-17> <그림3-18>과 같이 구현되었다. 각 테이블의 기본 키는 검은 선으로 표시한다.

점포(점포코드, 점포명, 사업자번호, 총인원, 점포면적, 주소, 전화번호, 점장명, 개점일, 폐점일, 지역현황)

상품(상품코드, 상품명, 단위, 단가, 매가, 원가, 재고량, 발주종료일, 입수)

거래선(거래선코드, 거래선명, 대표자명, 주소, 전화번호)

발주가능(상품코드, 거래선코드, 상품제공

원가, 보충수, 입수)

<그림 3-16> 서울의 논리 스키마

상품(상품코드, 상품명, 단위, 단가, 매가, 원가, 재고량, 발주종료일, 입수)

발주(상품코드, 거래선, 점포코드, 발주일자, 수량)

<그림 3-17> 대전의 논리 스키마

상품(상품코드, 상품명, 단위, 단가, 매가, 원가, 재고량, 발주종료일, 입수)

거래선(거래선코드, 거래선명, 대표자명, 주소, 전화번호)

발주가능(상품코드, 거래선코드, 상품제공원가, 보충수, 입수)

<그림 3-18> 부산의 논리 스키마

지역 물리 설계 단계에서는 이상의 논리 스키마가 특정 DBMS에 적합한 형태로 변환된

다. 본 사례에서는 ORACLE [Oracle, 1994] RDBMS이 사용되어 ORACLE에서 사용되는 물리 설계 산출물이 구현되었다. 특히, 이상의 전 설계 과정에 System Architect라는 CASE (Computer Aided Software Engineering) 도구가 이용되어 설계 창고 (Depository)에 저장되어 있으므로 언제라도 분산 설계 과정이 수정/변경 가능하게 되었다. 이상의 한화유통 분산 데이터베이스 설계 과정은 유통회사의 초유로 알려지고 있다.

IV. 결 론

본 연구에서는 양식을 근간으로 한 분산 데이터베이스 설계 방법론인 FD³ 를 제시하였다. FD³의 주요 특징은 하기와 같다.

첫째, FD³에서는 분산 데이터베이스 설계 핵심 단계 (양식 요구 사항, 분산 분석, 분산 설계)에 걸쳐 중요한 사용자 요구 및 분산 정보를 양식 그 자체에서 용이하게 획득할 수 있다. 그러므로써 데이터베이스 설계자가 기존의 인터뷰를 통한 방법과 비교할 때 사용자의 요구 사항 및 분산 설계에 필요한 정보를 누락이나 오류 없이 최대한 정확하게 반영할 수 있다.

둘째, 스키마 통합 단계에서 이름 상충 문제 해결을 위해 RDB가 구현되었다. 양식에 존재하는 모든 필드, 양식 이름을 데이터베이스에 저장해 두고 SQL 구문을 사용하여 이름 동의어와 동음이의어를 해결하였다. 실제로 한화유통의 경우 속성수가 기존의 3000여개에서 본 방법 적용 결과 500여개로 감소되는 효과가 있

었다. 기존의 스키마 통합 방법들은 경험적인 성격을 지니고 있어 설계자의 숙련도에 의존하는 경향이 있었으나, FD³에서는 SQL 구문을 활용하여 보다 과학적인 스키마 통합을 수행할 수 있는 계기가 마련되었다.

셋째, FD³에서는 기존의 분산 설계 방법에 비해 몇가지의 개선점을 보이고 있다. FD³에서는 특정 지역에서 사용하는 양식에 포함된 개체들에 대한 접근 정보를 기술하는 것이므로, 해당 지역에서 한번의 양식을 사용하는 것은 결국 양식을 중심으로 양식에 포함된 개체들간의 조인(Join)을 수행한 것과 동일한 상황이 된다. 따라서, FD³는 기존의 설계 방법론에서 적절하게 반영하지 못한 조인을 일부 반영하게 되었다. 나아가서, FD³는 통신 비용이 고려된 설계 모형이 이용되어 기존의 단순한 질의/갱신 횟수에 근거한 모형에 비해 분산 설계의 질을 향상시킬 수 있었다.

마지막으로, 기존의 대부분의 방법론이 수리적인 면에 치중을 하여 특정 분산 설계 상황에서 최적 설계 방안 도출에는 우수하나 복잡한 현실 문제에 적용되기에는 어려운 면이 많았다. FD³는 분산 설계의 보다 논리적인 면에 중점을 두어 설계 과정의 수리적인 면과 조화를 이루도록 시도하여 현실 문제 실제 적용 가능성을 제고하였으며 실제 유통 회사 분산 데이터베이스 설계 결과 양호한 반응이 얻어졌다.

본 연구로 분산 데이터베이스 설계 방법론으로서의 기본적인 아키텍처가 FD³로 구현되었다. FD³에 근거한 향후 연구 방향으로는 우선

데이터베이스 설계자가 용이하게 사용할 수 있는 설계 시스템의 개발에 있다. 이와 같은 설계 시스템의 근간으로는 요구 정보 및 설계 정보를 저장하기 위한 데이터베이스와 설계자 인터

페이스 향상을 위한 개발도구이다. 현재 FD³에서 미진한 부분인 (i) 다단계 조인 (Multi-phase Join) [Seo, 1994]과 (ii) 수직 단편화에 관한 보완 연구도 필요하다.

참 고 문 헌

- Apers, P.M.G., Query processing and data allocation in distributed database systems, Ph.D. Dissertation, Vrije Universiteit, Amsterdam, The Netherlands, Sept. 1982.
- Batini, C., and Lenzerini, M., A comparative analysis of methodologies for database schema integration, *ACM Computing Surveys*, Vol.18, No.4, pp.323-364, Dec. 1986.
- Batini, C., Ceri, S., and Navathe, S.B., *Conceptual Database Design: An Entity-Relationship Approach*, Benjamin/Cummings, Redwood City, CA, 1992.
- Casey, R.G., Allocation of copies of a file in an information networks, Proceeding 1972 Spring Joint Computer Conf., AFIPS, 1972.
- Ceri, S., Navathe, S.B., and Wiederhold, G., Distribution design of logical database schemas, *IEEE Trans. Software Eng.*, Vol.SE-9, No.4, pp. 487-504, July 1983.
- Ceri, S., and Pelagatti, G., *Distributed Databases-Principles & Systems*, McGraw-Hill, 1984.
- Ceri, S., Pernici, B., and Wiederhold, G., Distributed database design methodologies, Proceeding of the IEEE, Vol.75, No.5, pp.533-545, May 1987.
- Chang, S.K., and Cheng, W.H., A methodology for structured database decomposition, *IEEE Trans. Software Eng.*, Vol.SE-6, No.2, pp.205-218, March 1980.
- Chen, P.P., The entity-relationship model: toward a unified view of data, *ACM Transactions on Database Systems*, Vol.1, No.1, pp.9-36, Jan. 1976.
- Chu, W.W., Optimal file allocation in a multiple computer system, *IEEE Trans. Comput.*, Vol. C-18, No.10, pp.885-888, Oct. 1969.
- Choubineh, J., Mannino, M.V., and Tseng, V. P., A form-based approach for database analysis and design, *Communications of the ACM*, Vol.35, No.2, pp.108-120, Feb. 1992.

Deen, S.M., Amin, R.R., and Taylor, M.C., Data integration in distributed databases, *IEEE Transactions on Software Engineering*, Vol. SE-13, No.7, pp.860-864, July 1987.

Dowdy, L.W., and Foster, D.V., Comparative models of the file assignment problem, *ACM Computing Surveys*, Vol.14, No.2, pp.287-314, June 1982.

Flaatten, P.O., McCubbery, D.J., Oriordan, P.D., and Burgess, K., *Foundations of Business Systems*, The Dryden Press, 1992.

Gladney, H.M., Data replicas in distributed information services, *ACM Transactions on Database Systems*, Vol.14, No.1, pp.75-97, March 1989.

Lee, H., Modelling and optimization of data assignment in a distributed information systems, *Systems Science*, Vol.24, pp.173-181, 1993.

Lee, H., and Sheng, O.R.L., A multiple criteria model for the allocation of data files in a distributed information system, *Computers Ops Res.*, Vol.21, pp.21-33, 1992.

Lee, H., and Sheng, O.R.L., Analysis of optimal distribution designs for distributed databases on wide area telecommunication networks, *Korea Database Journal*, Vol.2, No.1, pp.3-25, 1992.

Lee, H., Sheng, O.R.L., and Stolen, J., Allocating data files over a wide area network: goal setting and compromise design, *Information and Management*, Vol.21, pp.85-93, 1994.

Levin, K.D., and Morgan, H.L., Optimal program and data location in computer networks, *Commun. ACM*, pp.315-322, 1977.

Lien, Y.N., Chang, Y.L., and Wah, B.W., File allocation problems on homogeneous two-level local broadcast network, Proc. Fourth Intl Conf. On Data Eng., pp. 92-99, 1988.

March, S.T., and Rho, S., Allocating data and operations to nodes in distributed database design, *IEEE Trans. Knowledge and Data Eng.*, Vol.7, No. 2, pp.305-317, 1995.

Navathe, S.B., Ceri, S., Wiederhold, G., and Dou, J., Vertical partitioning algorithms for database design, *ACM Trans. Database Syst.*, Vol.9, No.4, 1984.

Oracle Co., *Introduction to Oracle Volume2-SQL/SQL*Plus*, Oracle, 1994.

Ozsu, M. T., and Valduriez, P., *Principles of Distributed Database Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1991.

Ram, S., and Marsten R.E., A model for database allocation incorporating a concurrency control

mechanism, *IEEE Trans. On Knowledge and Data Eng.*, Vol.3, No.3, pp. 389–395, Sep. 1991.

Reddy, M.P., Prasad, B.E., Reddy, P.G., and Gupta Amar, A methodology for integration of heterogeneous databases, *IEEE Trans. on Knowledge and Data Eng.*, Vol.6, No.6, pp.920–933, Dec. 1994.

Seo, P.K., File Allocation and Join Site Selection Problem in Distributed Database Systems, Ph.D. Dissertation, Louisiana State University, 1994.

Sheng, O.R.L., and Lee, H., Data allocation design in computer networks: LAN versus MAN versus WAN, *Annals of Operations Research*, Vol. 36, pp.125–150, 1992.

Shu, N. C., Wong, H. K.T., and Lum, V.Y., Forms approach to requirements specification for database design, *SIGMOD REC.(USA)*, pp.161–172, May 1983.

Stallings, W., *Data and Computer Communications*,

4th Edition, Prentice Hall International Editions, 1994.

Stonebraker, M., Future trends in database systems, *IEEE Trans. Knowledge and Data Engineering*, Vol.1, pp.33–44, 1989.

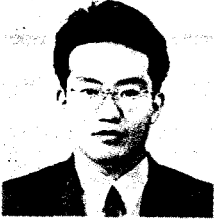
Teorey, T.J., Yang, D, and Fry, J.P., Logical design methodology for relational databases, *Computing Surveys* 18, No. 2, 1986.

Teorey, T.J., *Database Modeling: An Entity Relationship Approach*, Benjamin-Cummings, 1991.

Teorey, T.J., Distributed database design: A practical approach and example, *SIGMOD RECORD*, Vol. 18, No.4, pp.23–39, Dec. 1989.

Wah, B.W., and Ramamoorthy, C.V., The isomorphism of simple file allocation, *IEEE transactions on computers*, Vol.C-32, No.3, pp.221–231, 1983.

◇ 저자소개 ◇



공동저자 이희석은 서울대 산업공학학사, KAIST 산업공학석사, 그리고 Univ. of Arizona at Tucson에서 경영정보시스템을 전공하여 경영학박사학위를 취득하였으며 현재 KAIST 서울분원 경영정보공학과에 재직하고 있다. Information and Management, Information Systems, Information Processing and Management, Telematics and Informatics, Journal of Systems and Software, Annals of Operations Research, Computers and Operations Research, Computers and Industrial Engineering, European Journal of Operation Research, Systems Science, Engineering Economists, Omega, Korea Database Journal, Journal of MIS Research, International Encyclopedia of Business and Management 등 국내외 학술지 및 학회에 40여편의 논문을 발표하였다. 주요 연구관심분야는 데이터베이스 인프라 설계, 클라이언트/서버, 분산 데이터베이스, 정보통신망 계획 및 구축 등 이다.



공동저자 김희진은 이화여대 전자계산학학사, KAIST 경영정보공학석사를 취득하고 현재는 (주)한국 오라클의 데이터베이스 컨설팅팀에 근무하고 있다. 주요 관심분야는 분산 데이터베이스 설계, 데이터 모델링 그리고 전략적 데이터베이스 구축이다.



공동저자 김영삼은 홍익대 산업공학학사, KAIST 경영정보공학석사를 취득하고 현재는 동 대학원 박사과정을 이수중이다. 주요 관심분야는 데이터베이스 설계, 데이터베이스 구축 그리고 Business Process Reengineering이다.