# 관계 데이터베이스 시스템에서 Form Model을 이용한 개체-관계 스키마의 추출

김미화, 황태희, 배석찬
군산대학교 컴퓨터과학과
e-mail:kk8475@cs.kunsan.ac.kr

# Extracting Entity-Relationship Schema Using Form Model in Relational Database Systems

Mihwa Kim, Taehee Hwang, Seokchan Bae
Dept. of Computer Sciences, Kunsan National University

요 약

데이터 모델링 접근은 주로 순공학을 사용하여 미래의 필요구조에 초점을 맞추고 있기 때문에 새로운 데이터베이스를 설계하기 위해서는 많은 시간과 노력, 개발비가 필요하다. 그러나 기존의 데이터베이스를 재사용 한다면 개발에 필요한 시간과 노력, 개발비를 줄일 수 있을 뿐 아니라 자료의 재활용면에서 효율적이다. 본 논문에서는 사용 가능한 데이터베이스 application을 재사용하기 위해 Form model을 이용하여 새로운 데이터베이스 설계의 바탕이 되는 개체-관계 스키마를 추출하는 방법에 대해 연구하였다.

## 1. 서론

많은 기관에서는 기존의 데이터베이스 시스템이 부적당하거나 존재하지 않는 문서화의 문제와 이전의 설계자들이 그 이후로 업무를 바꾸었거나 업무인원이 감소되고 또 그만두게 되는 경우로 인해 겪게 되는 어려움 때문에 그들의 성장과 변화하는 능력을 방해받고있다.

역공학은 이러한 문제를 해결하기 위한 방법으로 정규적인 유지보수로는 더 이상 관리할 수 없는 데이터베이스 시스템을 재설계, 즉 물리적 데이터베이스로부터 개념적 데이터 모델을 회복하는 것이라고 할 수 있다. 역공학은 전혀 문서화되지 않았거나 문서화가 부실한 시스템에 적용될 수 있으며 이러한 역공학의 목적은 첫째, 잃어버리거나 대신 할 수 있는 문서화를 제공하기 위해 둘째, 유지보수를 돕기 위해 셋째, 하드웨어 또는 소프트웨어 플랫폼으로부터 다른 것으로 이주하기 위함이다.

본 논문에서는 form을 이용하여 사용 가능한 관계 데이터베이스에서 form을 생성하고, 이 생성된 form을 분석한 form 모델 스키마들로부터 개체-관계 스키마를 추출하기 위한 방법을 연구하였다. 여기에서 form은 대부분 사용자에 알맞게 설계되며 반드시 기본적인 데이터베이스의 구조를 반영하지는 않는다. Form 모델 스키마는 구조화된 정보와 그들의 구조와 인스턴스로부터 추출된 데이터사이의 제약 조건등과 같은 정보를 가지고 있다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 관련연구를 기술하였고, 3장에서는 관계 데이터베이스에서 개체-관계 스키마를 추출하기 위한 과정을, 4장에서는 결론 및 향후 연구방향을 기술하였다.

## 2. 관련연구

[2]는 관계 데이터베이스 스키마의 개체-관계 모델로의 역공학에 대한 체계적인 접근 방법을 제안하였다. 그들의 목표는 관계 데이타베이스 스키마로부터 개체-관계 스키마로의 전환을 가능하게 하는 변환에 있다. 그러나 그들의 접근은 몇 가지 한계를 가진다. 그들은 상속을 무시하였고, 입력이 되는 관계 데이터베이스 스키마가 제3정규형이어야만 한다. 그리고 어떠한 동음이의어와 동의어도 존재하지 않는다는 제한을 두고 있다.

[3]은 [2]보다 상속을 고려하는 더욱 강력한 접근법을 제안하였다. 그러나 제3정규형으로 전환하는 의미적인 전처리단계를 요구한다는 것과, 동음이의어와 동의어는 제거하였기 때문에 외부키를 인식함에 있어서 모호함이 존재한다는 약점이 있다. 또한 설계자에 의해 너무 일찍 의미적 입력을 요구하는 약점을 가지고 있다. 예를 들면, 그들의 기술을 이용하기 전에, 설계자는 후보키 가능성으로부터 각각 테이블을 위한 기본키를 선택하여야만 한다. 이 기본키의 선택은 공유되는 기본 키들을 통하여 일반화의 승인을 용이하게 해야 하며, 게다가 역공학자는 다른 후보키 대신에 기본키를 통해 발생하는 다른 테이블에 대한 모든 참조를 보증해야만 한다.

[4]는 관계 데이터베이스 스키마의 개체-관계 스키마로의 역공학에 대한 수학적인 것에 기반을 둔 완전하고 이론적으로 철저한 취급법을 제시하였다. 그러나 여기에서는 실제 문제로 발생하는 최적화 단계와 완벽하지 못한 설계는 다루지 않고 있다.

본 논문에서는 기존의 관련연구에서 다루지 못한 실제로 종종 직면하게 되는 제3정규형이 아닌 관계 데이터베이스에 대해서 살펴보았다. 그리고 현재 많은 상업적인 관계 데이터베이스 시스템은 외부키를 적용하지 않는 것에 의해서 역공학 작업을 복잡하게 만드는데, 본 논문에서는 의미적인 이해와 스키마의 분석 그리고 데이터 분석과 같은 많은 source로부터 얻은 정보를 활용하였다.

## 3. 관계 데이터베이스에서 개체-관계 스키마 추출

### 3.1 관계 데이터베이스와 form

데이터베이스 역공학은 소프트웨어 역공학의 한 분야로서 소프트웨어 역공학이 프로그램이 포함하고 있는 문제와 일치하는 설계 명세서의 회복과 소프트웨어 시스템의 개선을 위해 적용되는 것과 마찬가지로 데이터베이스 역공학도 현존하는 데이터베이스의 설계 명세서를 결정하는데 적용되고 있다.

그림 1은 역공학의 개념을 대략적으로 표현한 것이다[1].

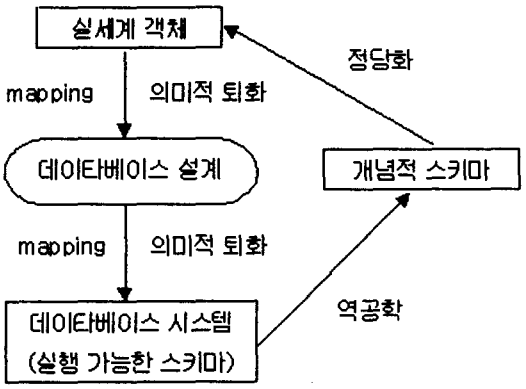


그림 1 역공학의 개념

그림 2의 관계 데이터베이스 스키마에서는 customer, order, product, order detail에 대한 정보가 포함되어있다.

| | |
|---|---|
| CUSTOMER | (CustomerID, CompanyName, ContactName, ContactTitle, AddressCity, Region, PostalCode, Country, Phone, Fax) |
| ORDER DETAILS | (OrderID, ProductID, UnitPrice, Quantity, Discount) |
| ORDERS | (OderID, CustomerID, OrderDate, RequiredDate, ShippedDate) |
| PRODUCTS | (ProductID, ProductName, UnitPrice, UnitsInStock) |

그림 2 관계 데이터베이스 스키마

그러나 이것만 보고는 다음과 같은 두 가지의 구조 외에는 정의할 수 없다.
첫째, 각각의 order는 하나의 customer에 속한다
둘째, 하나의 order는 몇 개의 products를 포함한다.
이러한 구조적인 정보는 데이터베이스 설계의 forward process과정 중에 잃어버리게 되며, 이 forward process동안 그림 1에서 보는 바와 같이, 개념 스키마에서 물리적 스키마까지 mapping은 스키마의 의미적 퇴화를 유도한다[5]. 데이터베이스는 해를 거듭할수록 조직 내에서 성장하고, 다른 사람들에 의해 수정될 수 있다. 문서화는 특히 오래된 시스템과 함께 효력이 없어지거나 존재하지 않기 때문에 누구도 어떤 데이터와 데이터들 사이의 관계를 정확히 알지 못하게 된다.

본 논문에서는 이러한 구조적인 정보와 의미들을 회복하기 위해 Form 개념과 역공학을 이용하였으며, 회복된 구조적 정보와 의미를 바탕으로 개체-관

계 스키마를 추출하는 과정을 제시하였다.
그림 3과 같은 관계 데이터베이스로 그림 4와 같은 form을 고려해 볼 수 있다.

| WINESNUM | VINTAGE | PRODNUM | PRODNAME |
|---|---|---|---|
| BD15 | Bordeaux | P103 | DUPONT |
| BG17 | Bourgogne | P088 | MARTIN |
| AL38 | Alsace | P015 | DURAND |
| BD18 | Bordeaux | P103 | DUPONT |
| BD33 | Bordeaux | P097 | BUBOIS |
| BG20 | Bourgogne | P103 | DUPONT |

그림 3 관계 데이터베이스의 예

| Producers' vintages | |
|---|---|
| PRODUCER NR : p103 | NAME : DUPONT |
| WINENUM | VINTAGE |
| BD15 | Bordeaux |
| BD18 | Bordeaux |
| BG20 | Bourgogne |

그림 4 form의 예

그림 4의 form은 생산자와 그들이 생산하는 와인에 대한 정보를 포함하고 있다. 이 form을 보면 하나의 생산자는 하나 이상의 와인을 생산할 수 있고, 주어진 하나의 와인은 오직 하나의 생산자에게 속한다는 것을 알 수 있다. 그러나, 앞서 언급했듯이 이러한 정보는 단지 직관적일 뿐이며 form의 표현은 대부분 사용자에 알맞게 설계된다. 이렇게 설계된 form은 데이터베이스 역공학의 입력으로 사용된다.

### 3.2 form의 기본 개념

관계 데이터베이스로부터 개체-관계 스키마를 추출하는 역공학 작업에 알맞은 form model은 object와 그들의 상호연관성 뿐 아니라 그것의 구성요소와 필드들을 명백하게 하기 위해 모든 데이터베이스 form을 추상화하는 것을 허용한다[6]. 그것은 form 인스턴스들을 활용하는 것에 의해 데이터베이스 form 구성요소들 사이에서 관계가 있는 제약조건들을 발견해 낸다. form의 기본개념은 다음과 같다.

1) form type
form type은 그림 4처럼 데이터베이스와 연결을 위해 적절하게 형식화 되어있는 form 필드를 말한다.

2) structural unit type
structural unit type은 같은 종류의 정보들의 그룹으로서 밀접하게 관련된 form 필드의 그룹을 말한다. 예를 들면, 그림 4 에서 와인과 생산자가 구조적 단위 형태가 된다.

3) form field
form field는 표제과 목록으로 구성되며, 표제은 form에서 미리 표시되어 있고 목록은 오퍼레이터에 의해 입력되거나 form 처리 시스템으로부터 표현되는 실제적인 데이터이다.

4) underlying source와 linked attribute
각각의 form 필드의 목록은 일반적으로 기본적인 데이터베이스 내에서 하나의 테이블의 속성과 연결되어 있다.

5) parent-child relationship type
parent-child relationship type은 두 개의 구조적인 단위 타입 사이에서 1:1 또는 1:다의 관계이다. relationship의 발생은 parent의 하나의 구조적 단위의 발생과 child 구조적 단위의 하나 또는 여러 개의 발생으로 이루어진다.

6) hierarchical tree
계층적 tree는 많은 parent-child 관계를 나타내며, 이 tree의 구조는 form 타입의 논리적 구조와 일치한다. tree의 root는 form의 표제가 되고, 다른 node들은 구조적 단위를 나타낸다.

7) cardinality constraints
cardinality constraints은 parent-child관계를 위해 구조적인 단위가 참여할 수 있는 인스턴스들의 수를 지정한다.

8) form model schema
form 모델 스키마는 구조적 단위 형태와 관계 형태 그리고 연관된 제약 조건들의 집합으로서 역공학의 입력으로 사용된다.

### 3.3 form 분석

그림 3의 관계 데이터베이스를 바탕으로 만들어진 그림 4의 form을 분석하여 form model schema를 산출해 내는 것이 이 분석 과정이다. 분석과정은 크게 정적 분석과 동적 분석으로 나뉘는데, 정적 분석은 구조적인 정보에 관계가 있는 반면, 동적 분석은 form 인스턴스에 의해 나타나게 되는 데이터 사이의 제약조건들에 관계가 있다. 정적 분석 동안, 논리적인 관점인 form 타입의 내포와 물리적인 관점인 레이아웃은 그것의 구조적 구성요소와 그들의 상호연관성을 확인한다. 동적 분서 동안에는 form 타입

의 외연, 즉 form 인스턴스들이 그들이 구성요소들 사이의 cardinality 제약조건, 함수 종속관계, 존재 종속관계와 같은 제약조건을 발견하게 된다. 동적 분석은 한번에 하나의 form 타입으로 처리된다.
이러한 분석을 통해 얻어진 결과는 데이터베이스 개념적 스키마(즉, 개체-관계 스키마와 같은)를 추출하는 과정의 출발점이 된다.
그림 5는 그림 4의 form model schema를 나타낸다.

```
Structural-Unit-Type =   {Wines, Producers}
LinkedAttributes =       {Winenum, Vintages, Prodnum, Prodname}
Relationship-type =      {Produce}
RelationNames =          {Wines, Producers, Produce}
RolesNames =             {WineOf, VintagesOf, ProducerNrOf, NameOf,
                          WinWith, Prodwith, surr, ProdProducing,
                          WinProducedBy}
Structural-Unit-Attributes = {Wines    (Winwith: Wines, WineOf: Winenum,
                                        VintagesOf: Vintages)
                              Producers (ProdWith: Producers,
                                        ProducerNrOf: Prodnum, NameOf: ProdName)
                              }
Relationship =           {Produce   (surr: Prodece, ProdProducing: Producers,
                                        Winproducedby: Wines)
                          }
Underlying-Sources =     {Wines}
-structural units cardinality constraints
CARDINAL-SU(Produce, Producers, 1, N)
CARDINAL-SU(Produce, Wines, 1, 1)
-Other constraints
EQEXIST-FIELD(Producers, Prodnum, Prodname)
EQEXIST-FIELD(Wines, Winenum, Winename)
```

그림 5 그림 4의 form model schema

이 form model schema는 구조적 단위 형태인 와인과 생산자라고 부르는 두 개의 동질적인 정보의 그룹을 포함하고 있으며, 그들의 각각은 구조적 단위 속성에 의해 표현된다. 또한 생산자와 와인 사이에는 생산이라는 관계가 존재한다. 또, 이 form에 포함된 기본적인 데이터베이스가 존재하는데 이것은 기본적인 source를 포함하며, form 필드 이름은 같은 이름을 가질 수 있는 연결된 속성으로부터 구별하기 위해 Of를 붙인다.
CARDINAL-SU(R, $r_i$ : $T_i$, min, max)는 parent-child 관계 R에서 $T_i$의 구조적 단위 $r_i$의 최소, 최대 cardinality를 나타내며, EQEXIST-FIELD(R, $r_1$ : $T_1$, ... , $r_n$ : $T_n$)은 구조적 단위 R에서 필드 $r_1$ : $T_1$, ... , $r_n$ : $T_n$필드에 동시에 존재하는 값이 있다는 것을 나타낸다.

### 3.4 개체-관계 스키마의 추출

관계 데이터베이스로부터 개체-관계 스키마를 추출하는 과정의 주요한 입력으로는 데이터베이스 form 모델이 사용된다. 추출 과정은 다음의 여섯 단계를 거친다.

1) 개체 유도

개체 타입은 구조적 단위의 타입으로부터 유도된다. 각각의 구조적 단위 타입은 같은 기본적인 테이블에 속한 연결된 속성의 필드를 모은다. form 모델 스키마에 있는 각각의 구조적 단위는 항상 오직 하나의 기본적인 데이터베이스 테이블과 관계가 있고 이 테이블은 개체 타입으로 옮겨지며, 테이블의 연결된 속성들은 개체 타입의 속성이 된다.
그림 4의 form에서 개체로 유도될 수 있는 구조적 단위 타입은 Wines, Producers이다.

2) 관계 유도

두 개의 구조적 단위 타입 사이의 각각의 관계 타입은 구조적 단위 타입으로부터 유도된 개체 타입들 사이의 관계로 옮겨진다.
여기에서는 두 개체, Wines와 Producers사이에 Produce라는 관계가 만들어진다.

3) 속성 첨부

이 단계에서는 이전에 확인된 개체 타입들에게 기본적인 source의 속성들을 첨부하는 것을 목표로 하는데 다음과 같은 두 가지의 상태가 발생 할 수 있다.
첫째, 기본적인 테이블은 오직 하나의 개체 타입과 관계된다. 상응하는 개체 타입은 모든 그것의 속성과 기본키를 상속받는다.
둘째, 몇몇 개체들은 하나의 기본적인 source로부터 유도된다. 그들의 속성은 그들 개체들 사이에서 나누어지게 된다.
개체 속성의 cardinality는 form 모델 스키마에서 상응되는 form 필드의 cardinality가 된다.

4) cardinality 결정

form 모델 스키마에서 구조적인 단위 사이의 cardinality는 개체-관계 스키마의 결과로서 생기는 개체들 사이의 cardinality이다.

5) 개념적 정규화

이 단계는 역공학 또는 의미-저장 스키마 변환을 이용하여 정상화, 최소화와 명확화와 같은 질을 주는 것을 목표로 한다[7].

6) 스키마 통합

몇몇 form이 대개 데이터베이스 조작을 위해 사용되며 각각 form은 기본적인 데이터베이스 관점에서 표현되므로 이들 form의 분석은 몇 개의 개체-관계 서브스키마를 산출해 낼 수 있다. 이들 서브스키마는 모든 기본적인 데이터베이스를 표현하는 전체적인 개념적 스키마로 합병되며, 스키마 통합의 결과

는 하나의 스키마로 축적되고 전체적인 개념으로 서서히 발전시킨다. 그림 6은 그림 5의 form 모델 스키마로부터 추출된 개체-관계 스키마를 나타낸다. 이렇게 6단계를 거쳐서 추출된 최종적인 개체-관계 스키마의 양쪽 개체들은 제3정규형이 아닌 그림 3의 관계 데이터베이스로부터 유도된 것이다. 이 최종적인 개체-관계 스키마의 양쪽 개체는 비정규화의 단계 없이 그림 4에 대한 form 모델 스키마(그림 5)에 의해 직접 만들어진 것이다.

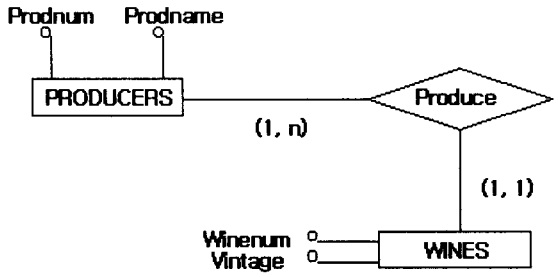


그림 6 그림 5의 form 모델 스키마로부터 추출된 ER 스키마

## 4. 결론

본 논문에서는 데이터베이스 구성요소들이 무엇인지, 그리고 그들의 상호연관성이 무엇인지 확인하는 과정으로 데이터베이스 역공학을 이용하였다. 데이터베이스 응용에 따라, 특히 오래된 시스템에서 종종 시대에 뒤떨어지거나 존재하지 않는 문서화의 문제를 가진다.

본 논문에서 제시한 form의 구조는 비록 표시되는 form이 반드시 직접적으로 데이터베이스 구조를 반영하지 않는다 하더라도 기본적인 데이터베이스 구조와 match된다. 이러한 form을 이용하여 개체-관계 스키마를 추출했을 때의 이점은 첫째, form의 구조적인 측면은 데이터베이스 구성요소와 그들의 상호 연관성을 회복하도록 도와주는 정보를 산출해 낸다. 둘째, 모든 정규 form에서의 데이터베이스를 고려하므로 의미가 form에 포함되어있기 때문에 모든 데이터베이스에서 비정규화의 문제를 찾아내고 해결하는 것을 가능하게 한다는 이점이 있다.

모든 데이터베이스 역공학은 어떻게 데이터베이스 구성요소와 그들의 상호 연관성을 회복하는가의 문제에 직면해야만 한다. 그러나 역공학은 데이터베이스 모델링에 대한 지식과 설계 그리고 실행도구들에 대한 기술적 지식 뿐 아니라 어떤 프로그래머에 의해 프로그램 되었는지, 어떤 설계자에 의해 설계되었는지에 대한 지식까지도 알아야 하기 때문에 완전히 자동화된 처리를 할 수는 없다. 본 논문에서 제시한 form의 분석에서도 사람에 의해 조작되는 부분이 여전히 필요하다. 이것은 form을 읽기 쉽고, 이해하기 쉽게 만들지만 완전한 자동화가 불가능하므로 이에 대한 연구가 계속 되어야 할 것이다.

## 참고문헌


[1] Roger H. L. Chiang, Terence M. Barron. "Quality Issues in Database Reverse Engineering : An Overview", Engineering management conference. 1995.

[2] Kathi Hogshead Davis, Adarsh K. Arora. "Converting a relational database model into an Entity-Relationship model", Proceedings of the sixth international Conference on Entity-Relationship Approach. 1987.

[3] S.B.Navathe, A.M.Awong, "Abstracting relational and hierarchical data with a semantic data model", Proc. of the Sixth International Conf. on E-R Approach. 1987.

[4] Victor M. Markowiz, Arie Shoshani. On the correctness of representing exteded Entity-Relationship wtructures in the relational model", SIGMOD. 1989.

[5] JL Hainaut, M chandelon, C Tonneau, M joris. "Contribution to theory of database reverse engineering", Proc. of the IEEE Working Conf. 1993.

[6] N.Mfourga. "A model for abstracting database forms", Technical Report YEROOS TR-97/05, IAG-QANT. 1997.

[7] JL Hainaut. "Entity-generating schema transformation for entity-relationships models", In Proc. of the 10th Int. Conf. ER. 1991.