

ER 도형의 연결 위상구조를 이용한 스타 스키마 추출 알고리즘

신성현⁰ 김진호
강원대학교 컴퓨터학과
{brain94, jhkim}@cs.kangwon.ac.kr

A Star Schema Extraction Algorithm Using Connection Topologies of ER Diagrams

Sung-Hyun Shin⁰ Jin-Ho Kim
Dept. of Computer Science, Kangwon National University

요 약

관계형 OLAP 시스템에서는 다차원 분석에 사용될 데이터물 하나의 사실과 여러 개의 차원 테이블로 구성된 스타 스키마로 표현한다. 사실 테이블은 측정치의 속성을 포함하며, 차원 테이블들은 이러한 사실 테이블에서 중심으로 뻗어져 나온 형태로서, 측정치에 대한 정보를 제공한다. 방대한 크기의 OLTP 소스 데이터베이스로부터 사실과 차원 테이블로 구성된 스타 스키마 설계하기 위해서는 풍부한 경험과 많은 시간을 필요로 한다. 따라서 스타 스키마를 효과적으로 설계하기 위한 체계적인 설계 방법이 필요하다. 본 논문에서는 소스 데이터베이스들에 대한 개념적 모델인 ER 도형을 이용하여 여러 개의 사실 엔티티를 추출하는 알고리즘과 이를 통해 자동적으로 스타 스키마를 설계하는 알고리즘을 제시하였다. 스타 스키마가 하나의 사실 테이블에 여러 개의 차원 테이블이 M:1의 관계로 연결되었다는 성질을 활용하여, 이 방법에서는 ER 도형이 연결된 위상 구조를 이용하여 각 엔티티에 연결된 M:1 관계의 개수에 근거하여 사실과 차원을 추출하는 방법을 제시하였다. 본 논문에는 이러한 방법을 통해 데이터 웨어하우스의 스타 스키마를 자동적으로 쉽게 설계하여 소요되는 시간과 노력을 줄일 수 있도록 하였다.

1. 서 론

데이터 웨어하우스는 해당 응용의 운영 시스템으로부터 생성된 과거로부터 현재까지 누적된 방대한 데이터들로부터 다양한 분석 정보를 추출하고 요약하여 모아놓은 데이터의 저장고이다 [1][2]. 이렇게 함으로써 OLAP 질의에 응답하기 위해서 운영 시스템의 소스 데이터를 참조하지 않고 빠르게 처리할 수 있다. 이 데이터 웨어하우스는 사용자들의 다양한 분석을 편리하게 제공하기 위하여 분석의 기준이 되는 여러 차원들과 분석할 내용에 해당하는 측정값들로 구성된 다차원 데이터 형태로 표현된다. 그리고 OLAP은 다차원 데이터를 저장하는 방법에 따라, ROLAP, MOLAP, HOLAP로 나누어지고, 가장 널리 사용되는 ROLAP에서는 다차원 데이터를 표현하기 위해 스타 스키마 또는 스노우플레이크 스키마를 이용하여 다차원 데이터를 저장하기 위한 테이블들을 설계한다. 스타 스키마란 사실 테이블을 중심으로 여러 차원 테이블들이 뻗어져 나온 별 모양과 같은 형태를 취한다 [3][4]. 이 스타 스키마는 다차원 데이터를 관계 데이터베이스의 테이블들을 이용하여 저장하면서, 비교적 단순한 형태로 표현하기 때문에 사용자들이 쉽게 이용할 수 있다는 장점이 있어 널리 활용되고 있다.

스타 스키마는 OLTP 운영 데이터베이스 스키마로부터 차원과 사실을 추출하여 스타 스키마를 설계해야 한다. 보통의 운영 데이터베이스들은 ER 도형과 같은 개념적 모델을 활용하여 해당 데이터베이스의 개념적 스키마를 설계하며 [5][6], 이 개념적 스키마를 토대로 논리적 스키마 및 물리적 스키마를 설계하는 체계적 방법론과 알고리즘 및 도구들이 개발되어 있다. 그러나 데이터 웨어하우스의 스타 스키마는 설계자의 경험과 직관에 의해 주로 설계되고 있다. 이러한 스타 스키마를 편리하고 효과적으로 설계하기 위해서는 이를 설계하기 위한 체계적 설계 방법론과 알고리즘을 개발하는 것이 필요하다 [2][7].

따라서 본 연구에서는 운영 데이터베이스의 ER 도형을 이용하여 스타 스키마를 자동적으로 추출하는 방법을 개발하고자 한다. 스타 스키마는 하나의 사실 테이블에 여러 개의 차원 테이블이 연결된 형태이며, 사실과 차원 간에는 M:1의 대응 관계를 갖는다. 이러한 성질을 활용하면 ER 도형의 각 엔티티에 연결된 M:1 관계의 개수에 근거하여 사

실과 차원을 분리할 수 있다. [8]의 연구에서는 이러한 ER 도형의 엔티티와 관계가 연결된 형태에 따른 위상 값을 계산하는 식을 제시하였다. 본 연구에서는 이러한 ER 도형의 연결 위상 값을 이용하여 많은 수의 엔티티와 관계로 구성된 대형의 ER 도형으로부터 여러 개의 사실과 이에 연결된 차원들을 자동적으로 추출하는 알고리즘을 개발한다. 이를 통해 데이터 웨어하우스의 스타 스키마를 자동적으로 쉽게 설계하여 스타 스키마 설계에 소요되는 시간과 노력을 줄일 수 있도록 하였다.

2. 관련연구

데이터 웨어하우스에서 다양한 분석정보를 위해 다차원 데이터를 효율적으로 표현할 수 있는 여러 개념적 모델들이 제안되었다 [9][10][11]. [9]은 데이터 웨어하우스를 설계하기 위한 체계적 단계를 제시하며, 데이터 웨어하우스의 다차원 데이터를 개념적으로 표현하기 위한 DFM(Dimensional Fact Model)을 제시하였다. [10]에서는 운영 데이터베이스의 개념적 설계에 널리 활용되는 ER 도형을 확장하여 다차원 데이터를 개념적으로 모델링하기 위한 starER이라는 모델을 제시하였다. 이와 유사하게 [11]은 노드와 아크(Arc)로 이루어진 차원 그래프를 활용하여 개념적으로 다차원 데이터를 표현하도록 하였다. 그러나 이 방법들은 데이터 요구 사항을 추출하고 이를 제안된 개념적 데이터 모델로 표현하는 전체 과정이 설계자의 직관과 경험에 의존하고 있기 때문에 많은 노력과 시간이 든다. 따라서 이러한 다차원 데이터를 좀 더 체계적이고 자동화된 방법으로 찾는 방법이 필요하다.

이러한 자동화된 설계 방법에 대한 연구들은 주로 ER 도형을 활용하여 다차원 데이터를 추출하는 방법들에 집중되어 있다 [5][6]. ER 도형에서 다차원 데이터로 변환하는 방법으로 [11]에서는 기존의 ER 도형을 토대로 설계자가 사실과 차원을 구분한 후, 이를 다차원 데이터로 쉽게 표현될 수 있도록 ER 도형을 변형하도록 하였다. 이렇게 변형된 ER 도형을 이용하여 다차원 데이터로 변환하는 방법을 제시하였다. [9][12]에서는 ER 도형에서 다차원 데이터를 스타 스키마로 변환하는 방법을 소개하고 있다. 이 방법에서는 차원의 분류 계층을 반자동화된 방법으로 찾을 수 있지만 여전히 설계자가 ER 도형의 의미를 파악하여 사실 엔티티를 지정해야 하고, 애트리뷰트들 간의 계층 구조를 변형하는 등 설계자의 개입을 여전히 요구하고 있다.

스타 스키마의 사실 테이블은 여러 개의 차원 테이블과 연결되며, 차원과 차원들은 서로 별개의 애트리뷰트들로 구성된다는 성질에 착안하

1 이 논문은 첨단정보기술연구센터(AITrc)를 통하여 한국과학재단의 지원을 받았다.

여 [8]에서는 사실 엔티티를 자동으로 추출하는 방법을 제안하였다. 이 방법에서는 ER 도형의 연결 형태와 연결된 개수의 정량적 정보를 각 엔티티의 연결 위상 값을 계산하는 식을 제안하였다. 그러나 [8]의 방법에서는 단순히 위상 값이 제일 높은 엔티티를 사실 엔티티로 선택하고 있어서 한 개의 사실 엔티티로 구성된 ER 도형으로부터 한 개의 스타 스키마를 생성할 수 있지만, 여러 사실 엔티티로 이루어진 대형의 ER 도형으로부터 여러 개의 스타 스키마를 생성하는 경우에는 활용하기 어렵고 설계자의 개입이 필요하다. 또한 위상 값만으로 사실을 추출할 경우 연결된 관계가 많은 엔티티가 오히려 적은 위상 값을 갖는 경우도 있을 수 있으므로 다양한 ER 도형 유형에 대한 고려가 부족하다.

따라서 이 논문에서는 [8]에서 제시한 ER 도형의 연결 위상 값을 이용하여 대형의 복잡한 ER 도형으로부터 여러 개의 스타 스키마를 생성하는 알고리즘을 제시하고자 한다. 이 방법에서는 위상 값과 엔티티에 연결된 관계의 개수를 고려하여 좀 더 유용한 사실 엔티티를 추출할 수 있도록 한다. 뿐만 아니라 대형의 ER 도형은 여러 개의 사실 엔티티를 포함할 수 있으므로 주위의 엔티티들보다 상대적으로 위상 값이 큰 엔티티들을 사실로 선택하여 여러 스타 스키마를 추출할 수 있도록 한다. 또한 차원 엔티티들에 M:1로 연결된 다른 엔티티들을 추출한 후 정규화한 형태로 차원을 설계하여 스노우플레이크 형태의 스타 스키마도 추출할 수 있도록 한다.

3. 연결 위상구조 계산식

스타 스키마를 설계하는 데 구성된 엔티티들의 속성을 분석하면, 일반적으로 차원 테이블에서 외래키로 관계된 수치 데이터의 속성을 가진 엔티티들이 사실의 후보들이다. 그리고 수치 데이터를 가지지 않는 속성을 가진 엔티티들은 차원의 후보이다[2]. 이렇게 사실과 차원 테이블을 결정하는 것은 오랜 축적으로 인한 경험으로 결정된다. 하지만 일반적으로 데이터 웨어하우스 시스템에서 스타 스키마를 개발하기 위해 엔티티들의 속성으로 사실과 차원을 결정하는 대신에, 엔티티들 간의 관계를 통해 사실과 차원을 결정하는 개념 방법을 이용한다. [그림 1]은 ER 도형의 위상구조를 이용한 계산식이다. 전체 ER 도형에서 모든 엔티티들과 엔티티들 사이에 관계를 분석하여 높은 연결 위상 값을 갖는 엔티티가 사실 테이블이 되며, 그 엔티티와 관계된 다른 엔티티들이 차원 테이블이 되는 개념이다.

$$T_total = weight_w * T_width + weight_d * T_depth$$

weight_w = 1
weight_d = 0.8
T_width = 직접적 관계를 갖는 엔티티들의 수 (n)
T_depth = 모든 직접적 관계를 갖는 자식 엔티티의 위상구조 값의 총합계
T_total = 엔티티의 연결 위상 값
T_total = 직접적 M:1 관계를 갖지 않는 엔티티의 연결 위상 값은 '0'

[그림 1] 연결 위상구조식

[그림 1]에서 위상식을 보는 바와 같이, T_width는 직접적 M:1 관계를 갖는 자식 엔티티들의 개수를 나타내며, T_depth는 간접적 관계가 있는 직접적 관계인 엔티티들의 연결 위상 값을 나타낸다. 이 두개의 비중 관계는 weight_w > weight_d 가 성립된다. T_total은 계산하고자 하는 현재 엔티티의 연결 위상 값을 나타내며, 직접적 M:1 관계를 갖지 않는 엔티티에 대해서는 연결 위상 값이 0으로 된다. 첫 번째로 연결 위상구조식을 적용하기 위하여, 리프 엔티티들로부터 연결 위상구조식을 적용한다. 리프 엔티티들의 연결 위상 값들은 직접적, 간접적 M:1 관계를 갖지 않으므로, '0'의 값을 갖는다.

이 [그림 1]의 식을 일반화하여 다음과 같이 연결 위상구조식을 정의하였다.

$$T_total (parent) = n + 0.8 * \sum_{i=1}^n T_total (child(i))$$

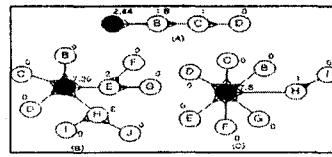
[그림 2] 연결 위상구조식의 일반화

4. 스타 스키마의 사실 테이블 추출 알고리즘

데이터 웨어하우스 시스템에서 빠른 질의를 처리하는 스타 스키마가 이상적이다. 스타 스키마 구조에서 사실과 차원 테이블 사이에 관계는 M:1 관계로 이루어져 있지만 실제 세계 데이터 웨어하우스의 스타 스키마는 사실과 차원 테이블 사이에 M:N 관계를 자주 포함하고 있다[2]. 이러한 관계의 구조를 가진 ER 도형에서 스타 스키마를 추출할 때 발생하는 문제가 생길 수 있고, 스타 스키마의 구조를 잃어버리기 쉽다. 그리고 이러한 관계는 복잡성이 오히려 증가되고 질의 처리 성능을 저하시킨다. 이때, 그러한 관계의 구조를 가진 ER 도형으로부터 스타 스키마를 추출할 때에 가장 중요한 점은 사실 테이블과 차원 테이블을 어떻게 결정하는 지가 중요한 문제이다.

4.1 사실 테이블과 M:1 관계인 엔티티 추출

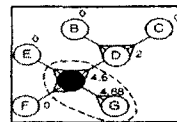
연결 위상구조식을 이용하여 일반적 M:1 관계인 ER 도형에서 사실 테이블을 결정하려고 한다. [그림 3]에서 나타나고 있는 ER 도형들은



[그림 3] 일반적 M:1 관계의 ER 도형

모두 일반적 M:1 관계를 유지하고 있다. 연결 위상구조식을 적용하면 가장 큰 연결 위상 값을 갖게 된 엔티티가 사실로 결정될 수 있다. 하지만 연결 위상구조식의 결과값으로 스타 스키마의 사실 테이블을 추출할 때 여러 문제들이 발생할 수 있다.

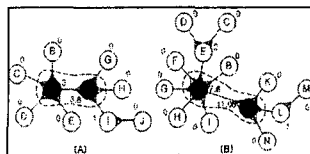
4.1.1 사실 테이블과 M:1 관계인 엔티티 추출



[그림 4] ER 도형에서 사실 추출

[그림 3.9]에서 엔티티 A는 인접한 엔티티들과 많은 M:1 관계를 가지고 있고 연결 위상구조식을 통해 계산하게 되면 엔티티 A는 사실 테이블이 될 수가 없고, M:1 관계로 인접한 엔티티 G가 사실 테이블이 되는 문제가 발생한다. 이러한 문제점을 해결하기 위한 방법은 사실 테이블을 구별하기 위해 직접적 엔티티들의 개수를 분석하고, 직접적 엔티티를 갖지 않은 리프 엔티티로부터 연결 위상구조식을 계산한다. 마지막으로, 높은 연결 위상 값을 갖는 엔티티가 사실 테이블로 결정되는 방법이다.

4.1.2 여러 개의 사실 엔티티 추출



[그림 5] 여러 개의 사실 추출(1)

실세계에서 분석자들은 하나의 사실 테이블이 아닌 다른 사실 테이블에서도 분석 정보를 얻기 원한다. 그래서 요구분석에 따라 하나의 사실만을 추출하는 것이 아닌 두 개의 엔티티들을 사실로 추출한다.

연결 위상구조식에만 계산하는 경우, 하나의 엔티티만이 사실 테이블이 된다. 하지만 이 경우에는 ER 도형에서 하나의 엔티티가 사실 테이블이 되는 것이 아니라 두 개 이상의 엔티티들이 많은 자식 엔티티들을 포함하고 있으므로, [그림 5]에서와 같이 모두 사실 테이블로 결정될 수 있는 경우이다.

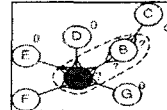
큰 연결 위상 값을 갖는 엔티티를 사실 테이블로 결정하고, 결정된 사실 엔티티와 인접한 엔티티가 포함하는 자식 엔티티들의 개수를 비교한다. 그래서 사실 테이블이 갖는 자식 엔티티들의 개수보다 크거나 같을 경우 역시 인접한 엔티티도 사실 테이블로 결정한다.

[그림 6]에서 위와 같은 방법으로 사실을 추출할 경우 사실 테이블이 갖는 모든 자식 엔티티의 개수가 작으므로 인접한 엔티티는 사실 테이블이 될 수 없다. 하지만 ER 도형에서 사실 테이블보다 자식 엔티티의 총 개수가 작더라도 많은 간접적 관계를 이용하여 여러 사실 테이블들을 결정할 수 있다. 이러한 방법은 간접적 관계인 엔티티들로부터 계산된 직접적 관계인 엔티티들의 연결 위상값을 서로 비교하여 위상값이 클 경우 사실 테이블로 결정한다.

4.2 사실 테이블과 M:N 관계인 엔티티 추출

실세계에서 설계자들은 자주 사실 테이블과 차원 테이블 사이의 M:N 관계를 포함하여 설계한다. 이런 경우, 복잡성이 증가되어 스타 스키마의 구조를 잃기 때문에 스타 스키마에서 사용할 수 없다. 이러한 M:N 관계를 갖는 예제들을 다루면서 발생할 수 있는 문제점과 이를 해결하기 위한 방법을 논한다.

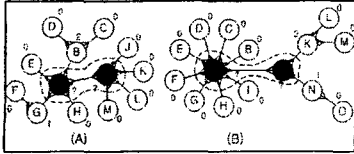
4.2.1 M:N 관계에서 사실 엔티티 추출



[그림 7] M:N 관계의 사실 추출

연결 위상구조식은 M:1 관계에서만 적용되기 때문에 [그림 7]과 같이 M:N 관계에서 연결 위상구조식을 적용하면 정확한 결과값을 나타낼 수가 없다. 비록 M:1 관계가 아닌 M:N 관계라도 정보 분석을 얻기 위해서는 사실 테이블로 결정해야 한다. 그래서 이런 M:N 관계인 두 엔티티들을 사실 테이블로 결정하기 위한 방법은 두 엔티티 사이에 인접한 자식 엔티티들의 개수를 비교하여 사실 테이블을 결정한다.

4.2.2 M:N 관계를 갖는 엔터티가 서로 사실 테이블일 경우



[그림 8] M:N 관계에서 사실 추출

직접적 관계인 자식 엔터티들의 총 개수를 비교하여, 사실 테이블이 갖는 자식 엔터티들의 개수보다 크거나 같을 경우 사실 테이블을 결정한다. 하지만 [그림 8]의 (B)와 같이 직접적 관계를 갖고 있는 자식 엔터티들의 총 개수가 작기 때문에 모든 간접적 관계를 갖는 엔터티들로부터 얻어진 연결 위상 값을 계산하여 비교한다. 이때 계산된 연결 위상 값에 따라 두 엔터티들이 사실 테이블을 결정할 수 있다.

```

Algorithm Extract_Fact
Input : an entity E
Output : a set of fact entities Sf
Begin
1. 연결 위상 값이 제일 높은 엔터티 E를 선택한다.
2. 추출된 사실 엔터티 집합인 Sf에 E를 저장한다.
3. E와 연결된 자식 엔터티들을 찾아 집합에 Sc에 저장한다.
4. For each C in Sc {
    if (Child_count(E) <= Child_count(C)) {
        C를 Sf에 포함시킨다.
        // 자식 엔터티들의 전체 개수(Child_count())를 서로 비교
    } else { // 전체 개수가 작을 경우
        if (Topology_Sum(Sf - (C)) < Topology_Sum(C와 child 집합)) {
            // 사실 엔터티는 비교하고자 하는 자식 엔터티를 제외한
            // 모든 다른 엔터티들이 갖는 연결 위상 전체값(Topology_Sum())을 비교
            // 사실 엔터티로 결정되는 자식 엔터티를 집합 Sf에 저장
        }
    }
}
End;
    
```

[그림 9] Extract_Fact 알고리즘

사실 테이블을 결정할 때, 하나의 사실 테이블만을 결정하는 것이 아닌 여러 개의 사실 테이블을 결정하기 위하여 [그림 9]와 같은 Extract_Fact 알고리즘을 제안한다. 이 알고리즘은 연결 위상 값으로 사실 테이블을 결정하고 나서, 여러 개의 다른 사실 테이블들을 결정하기 위하여 결정된 사실 엔터티들의 자식 엔터티들의 개수를 이용하여 자식 사실 테이블로 결정하거나, 작을 경우에는 간접적 관계를 통해 얻어진 모든 자식 엔터티들의 연결 위상 값을 이용하여 사실 엔터티의 여부를 결정한다. 그래서 Extract_Fact 알고리즘을 통해 단순히 하나의 사실 엔터티가 아닌 여러 개의 사실 엔터티들을 결정할 수 있다.

5. ER-to-Star 추출 알고리즘

앞서 설명했던 여러 관계에 대해 문제점을 논하고 이를 해결하기 위해 ER 도형에서 스타 스키마를 추출하기 위한 알고리즘을 제안한다.

```

Algorithm ER_to_Star
Input : ER diagram
Output : Star Schema
Begin
1. ER 도형에서 모든 엔터티를 후 집합 Sf에 저장
2. Sf내의 각 엔터티(E)에 M:1과 M:N 관계로 연결된 자식 엔터티들 Ch를 검색한다.
2.1. if(M:1 관계이면) 인접행렬 Adjoint[E][Ch] = 1;
2.2. if(M:N 관계이면) 인접행렬 Adjoint[E][Ch] = 1, Adjoint[Ch][E] = 1
2.3. Adjoint[E][Count] = E의 자식 엔터티 개수 (후 단계 1로 세팅된 개수)
3. 자식 엔터티 개수가 0, 1, 2, ... 순으로 각 엔터티의 연결 위상 값 계산
4. 계산된 모든 엔터티들의 연결 위상 값으로 엔터티들을 정렬
5. while (엔터티가 아직 남아 있으면) {
    5.1. 추출된 사실 엔터티 집합(Sf) = Extract_Fact(E)
        // 사실 엔터티를 결정하기 위해
        // 사실 추출 알고리즘인 Extract_Fact 알고리즘을 이용하여 Sf에 저장
    5.2. for(each E' in Sf) {
        5.2.1. 인접 행렬을 이용하여 E'에 연결된 자식 엔터티들을 추출
        5.2.2. 선택된 자원에 M:1 관계를 갖는 서브 차원 엔터티를 인접 행렬 이용하여 추출
        5.2.3. 추출된 사실과 차원을 결합하여 하나의 스타 스키마 생성하여 출력함
        5.2.4. 추출된 사실과 차원 엔터티들 대상 엔터티 집합 Sf에서 제거한다
    }
}
end;
    
```

[그림 10] ER-to-Star 추출 알고리즘

[그림 10]는 ER 도형에서 스타 스키마를 추출하기 위한 알고리즘을 제안한 것이다. ER 도형에서 구성된 모든 엔터티들을 검색하고, 엔터티들 사이에 M:1 관계와 M:N 관계를 검색한다. 이때 여러 관계에 대한 검색은 인접행렬을 이용한다. 검색된 엔터티들과 M:1 관계, 또는 M:N 관계로 인접한 다른 엔터티들은 인접행렬에서 M:1 관계와 M:N 관계를 기록하고, 인접행렬을 검색하여 M:1 관계와 M:N 관계를 하나도 갖지 않는 리프 엔터티들부터 연결 위상구조식을 계산한다. 그리고 연결 위상구조식의 계산을 통해 얻어진 결과값을 가지고 여러 개의 사

실 테이블이 결정될 수 있는 지에 대하여 Extract_Fact 알고리즘을 이용하고, 결정된 사실 테이블과 차원 테이블을 추출하여 스타 스키마를 설계한다.

지금까지 사실 엔터티에서 M:1 관계와 M:N 관계를 가지고 있으며, 차원 테이블이 될 수 있는 자식 엔터티들만 검색하는 방법에 대해 소개를 하였다. 사실이 갖는 수직 변수차원은 차원의 속성에 따라 결정되기 때문에, 차원 테이블을 검색해야 된다. 하지만 실제계에서 차원 테이블의 속성은 외래키들만 있을 수 있으며 사실에서 필요한 정보들을 갖지 못한다. 그래서 차원 테이블만을 검색하는 것이 아니라, 차원에 대해서도 정보를 갖고 있는 서브 차원 테이블들도 함께 검색해야 할 필요가 있다. ER-to-Star 추출 알고리즘에서와 같이 인접행렬을 이용하면 차원 테이블뿐만 아니라, 서브 차원 테이블을 갖는 스노우플레이크도 쉽게 검색 될 수 있다. 이로써 스타 스키마를 설계시 사실 테이블과 차원 테이블 사이에 단편적 정보를 가지고 있는 차원 테이블만을 검색하는 것이 아닌 스노우플레이크의 서브 차원 테이블들도 함께 검색하므로 정확하고 다양한 정보들을 획득할 수가 있다.

6. 결 론

기존 연구에서는 운영 시스템에서 기존 데이터베이스에 설계된 ER 도형으로 부터 스타 스키마 설계시 설계자가 직접 추가적 설계를 해야 하므로 설계 과정이 매우 복잡하고, 설계자의 오랜 경험과 직관적으로 설계가 이루어지기 때문에 많은 노력과 시간이 소요된다. 따라서 본 논문에서는 ER 도형의 연결 위상구조 값을 이용하여 복잡하고 다양한 상황에서의 ER 도형으로부터 여러 개의 스타 스키마를 추출하는 알고리즘을 제안하였다. ER 도형에서 연결 위상 값과 엔터티 간의 연결된 관계를 통해 스타 스키마의 사실 테이블을 추출하는 방법이다. 사실 테이블을 추출하는 Extract_Fact 알고리즘을 이용하여 여러 개의 사실과 이에 연결된 차원들을 결정하여 자동적으로 스타 스키마를 추출하는 ER-to-Star 추출 알고리즘을 제안하였다. 그리고 차원 엔터티를 결정하고 이 엔터티들과 M:1로 연결된 다른 엔터티들을 추출하여 정규화한 형태로 설계하여 스노우플레이크 스키마도 추출하였다. 이로써, 데이터 웨어하우스의 스타 스키마를 자동적으로 쉽게 설계하여 설계에 소요되는 시간과 노력을 줄일 수 있었다.

향후 연구 과제로, ER 도형에서 스타 스키마를 추출하는 과정에서 엔터티의 속성들 중 사실에 대한 수치 데이터들도 검색할 수 있는 연구 과제와 본 추출 알고리즘을 통해 상업적 다차원 데이터 분석 툴에 적용하여 다양한 데이터를 저장에서부터 분석에 이르기까지의 모든 과정을 자동화 시스템을 구축하는 연구가 이루어져야 한다.

7. 참고문헌

- [1] Immon, W. H., *Building the Data Warehouse*, 2nd Ed., New York: John Wiley and Sons Inc., 1996.
- [2] Kimball, R., Reeves, L., Ross, M., and Thornthwaite, W., *The Data Warehouse Lifecycle Toolkit*, New York: John Wiley & Sons Inc., 1998.
- [3] Kimball, R., *The Data Warehouse Toolkit*, John Wiley and Sons, 1996.
- [4] Poe, V., "Building a Data Warehouse for Decision Support," Upper Saddle River, NJ : Prentice Hall, 1996.
- [5] Chen, P. S., "The Entity-Relationship Model: Toward unified view of Data," ACM TODS, 1(1):9-36, 1976.
- [6] Batini, C., Ceri, S., and Navathe, S., *Conceptual Database Design*, Benjamin/Cummings, 1992.
- [7] Kimball, R., "A Dimensional Manifesto," DBMS, August 1997.
- [8] Song, I.-Y., "A Quantitative Method to Convert Enterprise Business ER Diagram to Data Warehousing Star Schema," Technical Report, Drexel University, 2002.
- [9] Golfarelli, M. and Rizzi, S., "A Methodological Framework for Data Warehouse Design," Proc. of Int'l Workshop on Data Warehousing and OLAP, pp. 3-9, 1998.
- [10] Tryfona, N., Busborg, F., and Christiansen, J.G.B., "starER: A Conceptual Model for Data Warehouse Design," Proc. of the DOLAP 1999.
- [11] Cabibbo, L., Torlone, R., "A logical approach to multidimensional database," Proc. 6th EDBT, LNCS 1377, pp. 183-197, 1998.
- [12] Golfarelli, M., Maio, D., and Rizzi, S., "Conceptual design of data warehouses from E/R schem," Proc. HICSS-31, VII, pp. 334-343, 1998.