

비분배적 계층구조 스키마의 분배적 계층구조 스키마로의 변환

오미화*, 최인수**

Transforming Non-distributive Hierarchy Schemas into the Distributive Hierarchy Schema

Mi-Hwa Oh *, In-Soo Choi **

요약

현 OLAP 체제에서는 모든 계층구조 스키마가 분배적으로 되는 경우에만 제대로 구현되며, 실제계에서 생길 수 있는 여러 비분배적 계층구조 스키마의 경우는 구현 불가능하다. 이전 연구에서는 집성에 있어서 현실적으로 발생할 수 있는 모든 계층구조 스키마를 하나의 분배적 계층구조와 일곱 개의 비분배적 계층구조 스키마로 분류한 바 있다. 본 연구에서는 이러한 일곱 개의 비분배적 계층구조 스키마를 분배적 계층구조 스키마로 변환시키는 기법을 제안하고 있다. 우선적으로 복잡한 비분배적 계층구조 스키마를 더 단순한 계층구조 스키마로 분할시킨 다음, 이를 분배적 계층구조 스키마로 변환시키는 것이 이 기법의 특징이다. 따라서 이 기법은 실제계에서 일어날 수 있는 복잡한 계층구조 스키마를 현 OLAP 체제에서 구현 가능하게끔 해준다고 볼 수 있다. 본 연구에서 제안하는 방법은 구현의 관점과 스키마 설계의 관점에서 볼 때에 유용한 방법이 되리라 생각한다.

▶ Keyword : 계층구조 스키마, 분배적, 비분배적, 변환 기법, 구현, 스키마 설계

Abstract

Online analytical processing (OLAP) architecture performs only well on the facts that are summarizable along each dimension, where all hierarchy schemas are distributive, and consequently many kinds of non-distributive hierarchy schemas arising in real-world situations are not addressed by current OLAP implementations. In previous work, we presented the classification

• 제1저자 : 오미화 • 교신저자 : 최인수

• 투고일 : 2011. 09. 23, 심사일 : 2011. 10. 02, 게재확정일 : 2011. 10. 07.

* 숭실대학교 대학원 산업·정보시스템공학과 박사 과정(In a Ph.D. program of Industrial & Information Systems Engineering, Soongsil Univ.)

** 숭실대학교 산업·정보시스템공학과 교수 (Professor of Industrial & Information Systems Engineering, Soongsil Univ.)

of hierarchy schemas in aggregation, including one distributive and seven non-distributive hierarchy schemas. This paper presents a transforming approach of seven non-distributive hierarchy schemas into the distributive hierarchy schema. The first thing in the approach is cutting complex non-distributive hierarchy schemas into more simpler ones, and the next is the transformation of those simpler ones into the distributive hierarchy schema. This approach will let complex hierarchy schemas arising in real world situations be addressed by current OLAP implementations. It is thought that this approach will be useful one from an implementation and schema design point of view.

▶ Keyword : Hierarchy schema, Distributive, Non-distributive, Transforming approach, Implementation , Schema design

I. 서론

실세계에서의 복잡한 시스템을 표현하는 방법 중의 하나가 추상(abstraction)인데, 추상은 실세계에서의 어떤 전체를 나타내는데 있어서 세세한 모든 부분까지 나타내지 않고 대표적으로 중요한 부분만을 골라서 나타내고자 하는 한 가지 방법론이다. 현실적으로 모든 세세한 부분을 다루는 것이 불가능해서 추상이란 방법을 채택하고 있지만, 이러한 추상에서 다루는 중요한 대표적인 부분까지도 너무 많아서 이들의 지적 관리(intellectual management)가 어렵다는 문제가 생기게 된다. 이러한 문제를 해결하기 위해서 등장한 것이 추상 모델에서의 계층구조(hierarchy)이다[1].

계층구조를 다루는 RDB(relational database)와 OLAP(on-line analytical processing) 양 분야에서 모두 특정 기준에 따라 요약된 정보, 즉 집계 데이터(aggregate data)를 얻는데 신경을 쓰고 있다[2]. 이러한 집계 데이터를 빠르고 효율적으로 생성시키고자 먼저 RDB에서 별모양 스키마(star schema)를 제안해서 필요한 여러 집계 데이터를 SQL(structured query language) 기능을 활용하여 생성하였으나, SQL 기능이 내장(built-in)된 형태가 아니어서 각광을 받지 못하게 되었다. 반면, OLAP에서는 RDB에서 먼저 제안한 별모양 스키마를 사용할 때에 SQL 기능을 내장시키고 있다[3]. 이러한 연유로 오늘날 집계 데이터를 구하는 데에는 SQL 보다 OLAP을 선호하고 있다.

데이터 큐브(data cube)는 집계(summarization) 데이터 자체인 여러 큐브 뷰(cube view)가 모인 것인데, 여기서 큐브 뷰는 해당 차원 모두에 걸쳐서 각 차원의 수준을 조합시켜서 얻게 되는 격자점에서 집계한 데이터로 형성되게 된다. OLAP 지향 다차원 데이터 모델(multidimensional data models)에서의 각 차원에는 해당 차원의 수준들을 DAG

(directed acyclic graph)로 표현한 어떤 구조가 있기 마련인데, 이러한 구조를 계층구조 스키마라 부른다. 계층구조 스키마에 있는 각 수준에는 여러 개의 멤버가 있을 수 있으며, 이들 멤버 간에는 부모/자식 관계(child/parent relation)가 맺어져 있다[4-6].

OLAP에서 일컫는 합계가능조건(summarization conditions)이란 어떤 하나의 큐브 뷰를 다른 여러 큐브 뷰로 부터 정확하게 이끌어 낼 수 있는 조건을 말하는데, 이러한 합계가능조건을 만족시키기 위해서는 데이터 큐브가 분배적(distributive)이 되어야만 한다. 좀 더 구체적으로 말할 것 같으면, 계층구조 스키마가 분배적이어야 한다는 뜻이다 [2],[5],[7].

현 OLAP 체제에서는 모든 계층구조 스키마가 분배적으로 되는 경우에만 제대로 집계 데이터를 구할 수 있는데[8], 실제로 실세계를 개념화할 때에는 서로 다른 여러 종류의 비분배적 계층구조 스키마가 발생하게 된다. 저자들의 이전 연구 [9]에서는 실세계를 개념화할 때 발생하는 모든 계층구조 스키마를 일반화(generalization)에서 4가지, 집계(aggregation)에서 8가지의 총 12가지 계층구조 스키마로 분류한 바 있다. 이 중, 집계에서의 8개 계층구조 스키마는 한 개의 분배적 계층구조와 일곱 개의 비분배적(non-distributive) 계층구조 스키마로 분류되어 있다. 본 연구에서는 이러한 일곱 개의 비분배적 계층구조 스키마를 분배적 계층구조 스키마로 변환시키는 방안을 제시함으로써 실세계에서 일어날 수 있는 모든 계층구조 스키마를 현 OLAP 체제에서 완벽하게 다룰 수 있게끔 하자는 데에 목적을 두고 있다.

II. 관련 연구

II장에서는 추상 모델에서의 계층구조 분류방법에 대한 기존 여러 방안과 저자들이 제시한 방안을 비교 설명하고자 한다.

1. 추상모델에서의 계층구조 분류

차원의 계층구조를 Maliowski와 Zimányi[10]는 분석 기준(criterion)의 수를 출발점으로 삼아 분류하기 시작했다. 이러한 전체 하에서 다음으로 수준공유 여부를 따졌고, 세 번째로 최대 카디널 수[11]를, 마지막으로 최소 카디널 수[11]를 고려함으로써 계층구조를 분류하였다. 분석 기준의 수가 하나이고 수준을 공유하지 않으며 최대 카디널 수가 1:N(one to many)인 경우, 최소 카디널 수가 M-M(Mandatory-Mandatory)이면 직립성 대칭 계층(simple symmetric hierarchy), M-O(Mandatory-Optional)이면 직립성 비대칭 계층(simple asymmetric hierarchy) 그리고 O-O(Optional-Optional)이면서 직립성 일반화 계층(simple generalized hierarchy)으로 분류하였다. 분석 기준의 수가 하나이고 수준을 공유하지 않으며 최대 카디널 수가 N:M(many to many)인 경우에는 비직립성 계층(non-strict hierarchy)으로 분류한바 있다. 이외에 분석 기준의 수가 하나이고 수준을 공유하는 경우 다중 배타적 계층(multiple alternative hierarchy)과 다중 포괄적 계층(multiple inclusive hierarchy)으로, 분석 기준의 수가 두 개 이상이고 수준을 공유하지 않으면 병렬 비독립 계층(parallel independent hierarchy) 그리고 수준을 공유하면 병렬 독립 계층(parallel dependent hierarchy)으로 분류하였다(표 1 참조).

차원의 계층구조에는 크게 자식 수준의 멤버들이 모여서 부모 수준의 한 멤버로 집계되는 Has-A 관계의 구조와 최상위 수준의 하나의 멤버가 자식 수준의 여러 멤버로 그리고 자식 수준의 한 멤버가 손자 수준의 여러 멤버로 분할되는 식으로 계속해서 분할되는 Is-A 관계의 구조가 있는데, Smith등 [1],[12]은 보통업(bottom up) 식으로 집계해 나가는 Has-A 관계의 계층 구조를 집성 계층이라 부르고, 톱다운(top down) 식으로 분할해 나가는 Is-A 관계의 계층 구조를 일반화 계층이라 부름으로써 이 양 구조를 완전히 구별하고 있다. RDB에서도 엔티티(entity) 간의 관계를 다룰 때에 Has-A 관계 구조와 Is-A 관계 구조로 대별하고 있음[11]을 볼 때에, OLAP의 계층 구조를 분류할 때에는 Has-A 관계의 계층인지 Is-A 관계의 계층인지를 분류 하는 데에서부터 출발해야한다고 본다. 또한 Has-A 관계에 있는 엔티티들은 서로 다른 식별자(identifier)를 갖고 있고, Is-A 관계에 있는 엔티티들은 동일한 식별자를 갖고 있기에 Has-A 관계의 계층과 Is-A 관계의 계층은 분명히 다르다고 볼 수 있다. 따라서 차원의 계층구조를 분류할 때에는 분명히 다른 이 양 계층을 맨 처음 분류하고 난 다음, 분류된 계층 각각의 범위 내

에서 각각에 대해 분류 작업을 계속하는 것이 합리적이라고 생각한다[9].

표 1. Maliowski-Zimányi의 계층구조 분류
Table 1. Maliowski-Zimányi's Classification of Hierarchies

분석 기준 수	수준 공유 여부	최대 카디널 수	최소 카디널 수	계층구조
한 개	비공유	1:N	M-M	직립성 대칭 계층
			M-O	직립성 비대칭 계층
			O-O	직립성 일반화 계층
	N:M	비직립성 계층		
공유				다중 배타적 계층
				다중 포괄적 계층
두 개 이상	비공유			병렬 비독립 계층
	공유			병렬 독립 계층

표 1에서 볼 것 같으면 직립성 일반화 계층만이 Is-A 관계의 계층이고 나머지 7개의 계층은 Has-A 관계의 계층이다. 따라서 Maliowski와 Zimányi는 직립성 일반화 계층이 8가지 계층 구조 중의 하나의 계층구조로 분류하고 있어서 차원의 계층구조를 Is-A 관계인지 아니면 Has-A 관계인지로 대별해서 분류하기 시작한 것은 아니라고 볼 수 있다. 지금까지 볼 것 같으면 Maliowski와 Zimányi 외 여러 연구자들도 처음부터 Is-A 관계와 Has-A 관계로 대별해서 차원의 계층 분류를 하지 않고 있다. 예를 들면, Mansmann과 Scholl[14]은 차원의 계층구조를 이질 계층(heterogeneous hierarchy)으로 명명된 Is-A 관계의 계층을 포함한 8개로 분류하고 있어서 Maliowski와 Zimányi와 마찬가지로 차원의 계층구조를 Is-A 관계인지 아니면 Has-A 관계인지를 대별해서 분류하기 시작한 것은 아니라고 본다.

따라서 저자들[9]은 Is-A 관계의 계층과 Has-A 관계의 계층을 분류의 출발점으로 삼은 새로운 차원의 계층구조 분류법을 제안한바 있는데, 이하 이에 대해 설명하고자 한다.

2. 새로운 차원의 계층구조 분류

Has-A 관계의 집성계층이나 아니면 Is-A 관계의 일반화계층이나에 따라 차원의 계층구조 분류를 시작하고, 다음으로 집성에서는 분석하는 기준의 수, 수준 공유 여부, 최대 카디널 수 그리고 최소 카디널 수에 따라 순차적으로 분류 하였고, 일반화에서는 최대 카디널 수에 따라서 먼저 분류하고 다음으로 최소 카디널 수에 따라 분류하였다. 이하, 집성계층과 일반화계층에 있어서의 계층구조의 분류에 관해 상세히 설명하고자 한다.

2.1 집성계층의 분류

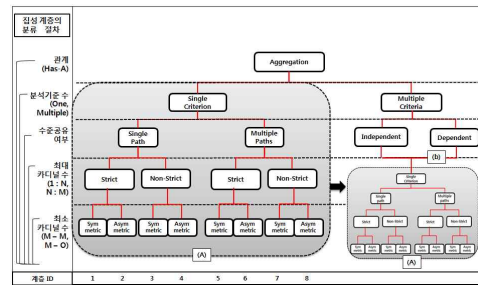
집성이라는 것은 연관성이 없는 여러 가치를 모아서 하나의 체계를 이룬다는 뜻이다. 여기서 연관성이 없다는 것은 엔티티의 식별자가 서로 다르다는 것을 의미한다. 따라서 집성은 서로 다른 정체성을 가진 여러 하위 엔티티가 모여서 상위 엔티티를 이루는 보통업 과정이다. 집성계층에서의 계층구조 분류는 분석하는 기준의 수, 수준 공유 여부, 최대 카디널 수 그리고 최소 카디널 수에 따라 순차적으로 이루어진다. 이때 분석 기준의 수가 하나인 경우를 단일기준(single criterion) 계층, 둘 이상의 경우를 다중기준(multiple criteria) 계층이라 한다. 먼저, 단일 기준 계층 분류를 살펴보면 수준 공유 여부에 따라서 공유되는 계층이 없는 단일경로(single path)와 서로 다른 경로에 의해 공유되는 수준이 하나라도 있는 다중경로(multiple paths) 계층으로 분류한다. 그리고 단일경로 계층과 다중경로 계층 각각을 대상으로, 모든 인접한 두 수준 멤버 간의 최대 카디널 수가 1:N가 되는 직립성(strict) 계층과 인접한 두 수준 간의 최대 카디널 수 중에서 최소한 하나가 N:M가 되는 비직립성(non-strict) 계층으로 분류한다. 마지막으로 직립성 계층과 비직립성 계층 각각을 대상으로 모든 인접한 두 수준의 멤버 간에 최소한으로 참여해야만 하는 멤버의 숫자를 말하는 최소 카디널 수에 따라 M-M의 관계에 놓여 있는 대칭(symetric)과 M-O의 관계에 놓여 있는 비대칭(asymmetric) 계층으로 분류하고 있다.

전술한 단일기준 계층이 여러 개 모여서 이룩되는 것이 다중기준 계층이다. 이러한 다중기준 계층의 경우는 이 계층에 속하는 여러 개의 단일기준 계층이 수준을 공유하지 않는 독립(independent) 계층과 공유하는 비독립(dependent) 계층으로 분류하고 있다. 따라서 독립계층의 경우는 각 단일기준 계층별로 전술한 단일기준 계층 분류법에 따라서 분류하고(그림 1의 (a)-(A)부분 참조), 비독립 계층의 경우는 후술할 일반화 계층으로 먼저 분류한 다음(그림 1의 (b)부분 참조) 다시 단일기준 계층 분류법(그림 1의 (a)-(A)부분 참조)에 따라서 분류하고 있다. 본 연구에서 사례로 삼은 계층이 비독립 계층임을 밝혀둔다.

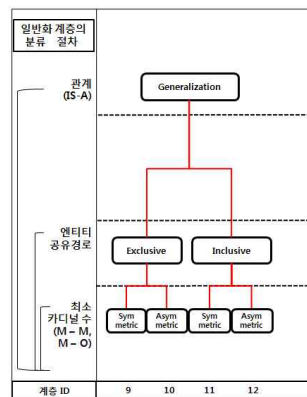
2.2 일반화계층의 분류

일반화는 최상위 수준에 있는 하나의 최상위 엔티티를 이 수준에 해당되는 한 개의 분석 기준에 따라 자식 수준에서의 여러 개의 하위 엔티티로 분할하고, 이들 각각의 하위 엔티티에 해당되는 각각의 한 개씩의 분석 기준에 따라 다시 손자 수준의 여러 하위 엔티티로 분할해 나가는 톱다운 과정이다. 일반화계층에서는 수준별로 다른 분석 기준을 가지고 있다.

그러므로 집성과 같이 분석 기준의 수와 수준공유 여부에 따라 계층을 분류하는 것은 아무런 의미가 없다. 이렇듯 독립된 각각의 경로만 존재하는 일반화에서는 엔티티 공유 경로의 존재여부에 따라 공유하는 경로가 없는 경우를 배타적(exclusive) 계층 그리고 공유하는 경로가 있는 경우를 포괄적(inclusive) 계층으로 먼저 분류한다. 그 다음으로 각각에 대해 최소 카디널 수에 따라 대칭 계층과 비대칭 대칭으로 분류한다(그림 1의 (b)부분 참조).



(a)



(b)

그림 1. 차원의 계층구조
Fig. 1. Dimension Hierarchies

이상의 새로운 계층구조 분류법을 종합해 보면 그림 1과 같다. 그림 1의 (a)는 집성계층에 관한 것이고, (b)는 일반화 계층에 관한 것이다.

그림 1에서의 각 계층을 계층구조 스키마로 나타내면 그림 2와 같다. 그림 2에서는 총 네 개의 수준을 가정하였고, DAG에 최대 카디널리티 표시(crow's foot notation)[11]를 하였으며, 각 수준의 멤버를 노드(node)로 표시하고 있다.

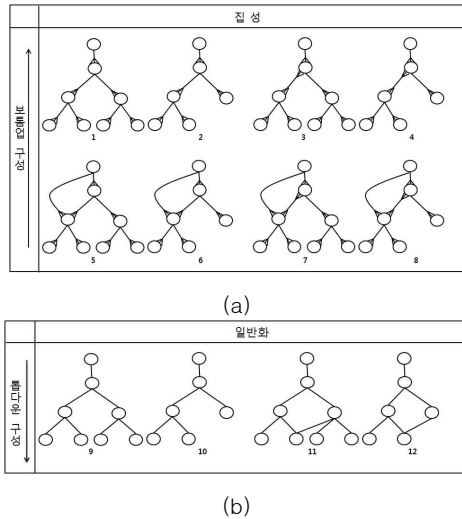


그림 2 계층구조 스키마
Fig. 2. Hierarchy Schemas

그림 2의 각 계층구조 스키마 밑에 계층 ID를 첨부하였는데, 계층 ID 1은 그림 1에서의 단일기준-단일경로-직립성-대칭 집성계층의 스키마이고, 계층 ID 2는 단일기준-단일경로-직립성-비대칭 집성계층의 스키마이며, 계층 ID 3은 단일기준-단일경로-비직립성-대칭 집성계층의 스키마이며, 계층 ID 4는 단일기준-단일경로-비직립성-비대칭 집성계층의 스키마이다. 그리고 계층 ID 5는 단일기준-다중경로-직립성-대칭 집성계층의 스키마이고, 계층 ID 6은 단일기준-다중경로-직립성-비대칭 집성계층의 스키마이며, 계층 ID 7은 단일기준-다중경로-비직립성-대칭 집성계층의 스키마이며, 계층 ID 8은 단일기준-다중경로-비직립성-비대칭 집성계층의 스키마이다. 또한 계층 ID 9는 배타적-대칭 일반화계층의 스키마이고, 계층 ID 10은 배타적-비대칭 일반화계층의 스키마이며, 계층 ID 11은 포괄적-대칭 일반화계층의 스키마이며, 계층 ID 12는 포괄적-비대칭 일반화계층의 스키마이다.

집성계층에서 볼 것 같으면, 계층 ID 1만이 분배적 계층이고 나머지 7개의 계층은 비분배적 계층이다. 실세계에서 일어날 수 있는 모든 계층을 현 OLAP 체제에서 완벽하게 다룰 수 있게 하자면 계층구조 스키마가 분배적이어야 함을 전술한 바 있다. 따라서 나머지 7개 비분배적 계층구조 스키마를 계층 ID 1과 같은 분배적 계층구조 스키마로 전환시킬 필요가 생기는 것이다. 이하 분배적 계층구조 스키마로의 전환 방법에 대하여 자세히 기술하고자 한다.

III. 분배적 계층구조 스키마

먼저, 표준 SQL에서 사용하는 분배적 집계함수에 대하여 설명하고자 한다. 그림 3에서의 같은 두 개의 차원으로 구성된 데이터 세트 $\{X_{ij} | i = 1, \dots, I; j = 1, \dots, J\}$ 을 대상으로 자세하게 살펴보면 다음과 같다.

함수 G(가

$$F(\{X_{i,j}\}) = G(\{F(\{X_{i,j} | i = 1, \dots, I\}) | j = 1, \dots, J\})$$

를 만족시킨다면, 집계함수 F(는) 분배적이 된다. COUNT, MIN, MAX, SUM 집계함수는 모두 분배적이 되는데 COUNT를 제외한 나머지 집계함수에 있어서는 $F=G$ 가 된다. SUM을 예로 설명하면,

X_{11}	X_{12}	X_{13}	-	X_{1J}
X_{21}	X_{22}	X_{23}	-	X_{2J}
X_{31}	X_{32}	X_{33}	-	X_{3J}
\vdots	\vdots	\vdots	\vdots	\vdots
X_{I1}	X_{I2}	X_{I3}	-	X_{IJ}
$j=1$	$j=2$	$j=3$	-	$j=J$

그림 3 $\{X_{ij} | i = 1, \dots, I; j = 1, \dots, J\}$ 데이터 세트

Fig. 3. A two dimensional set of values $\{X_{ij} | i = 1, \dots, I; j = 1, \dots, J\}$

$F(\{X_{i,j} | i = 1, \dots, I\})$ 는 그림 3에서 $j=1$ 인 경우 $\sum_{i=1}^I X_{i1}$ 가 되며, $j=2$ 인 경우 $\sum_{i=1}^I X_{i2}$, ..., $j=J$ 인 경우 $\sum_{i=1}^I X_{iJ}$ 가 되고,

$$G(\{F(\{X_{i,j} | i = 1, \dots, I\}) | j = 1, \dots, J\}) = \sum_{i=1}^I X_{i1} + \sum_{i=1}^I X_{i2} + \dots + \sum_{i=1}^I X_{iJ} = F(\{X_{i,j}\})$$

가 되어서 $F=G$ 가 되는 것이다. COUNT의 경우는 $F=COUNT$ 가 되고, $G=SUM$ 이 된다[6],[7].

그림 4(a)와 같은 계층구조 스키마는 비분배적이다. 왜냐하면 잎 수준(leaf level) 멤버의 합이 중간 수준 멤버의 합과 동일하게 되지 않고, 루트 수준(root level) 멤버의 합과도 동일하게 되지 않아서 위에서 말한 분배적 집계함수의 개념을 적용할 수 없는 스키마가 되기 때문이다.

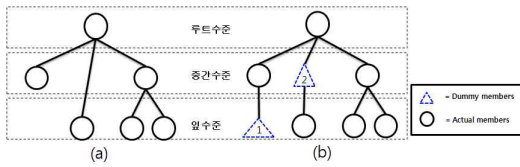


그림 4. 비분배적 계층구조 스키마
Fig. 4. A Non-distributive Hierarchy Schema

계층구조 스키마가 비분배적인 그림 4(a)를 분배적인 스키마로 변환시킨 것이 그림 4(b)이다. 그림 4(a)에 더미1 멤버를 삽입함으로써 후손존립성(onto) 문제를 해결하였고, 더미2 멤버를 삽입함으로써 망라성(covering) 문제를 해결한 것이 그림 4(b)이다[13]. 그림 4(a)에는 나타내지 않았지만 부모가 둘 이상이 되어서 분배적이 되지 못하는 경우도 있다. 이러한 경우 하나의 부모만 갖도록 직립성 스키마로 변환시켜야 한다. 후손이 존립하게 하고, 망라되게 하며 그리고 직립적으로 되게 하는 이러한 3가지 조건이 OLAP에서 일컫는 합계가능조건[5],[14-17]인데 이러한 합계가능조건을 충족시키도록 전환시킨 것이 분배적 계층구조 스키마인 그림 4(b)인 것이다.

그림 4(b)를 대표적으로 나타내면 그림 2(a)의 계층 ID 1이 된다. 구체적으로 계층 ID 1은 단일기준-단일경로-직립성-대칭 집성계층이다.

본 연구에서는 실세계를 개념화할 때에 생길 수 있는 7개의 비분배적 계층구조 스키마를 분배적 계층구조 스키마로 전환시킴으로써 현 OLAP 체계가 완벽하고 원활하게 돌아가게 하자는데에 목적을 두고 있다. 이하 사례를 통하여 7개의 비분배적 계층구조 스키마를 계층 ID 1로 전환시키는 방안을 설명하고자 한다.

IV. 사례 차원

과 구조(departmental structure)와 팀 구조(team structure)의 양 구조로 운영되는 하나의 회사 조직 구조를 대상 차원으로 삼은 것이 본 사례이다. 이하 이 사례 차원에 대하여 설명하고자 한다. 먼저, 과 구조로 운영되는 측면에서 볼 것 같으면,

[상황 1] 직원(employee)은 반드시 과(department)에 소속되어야하며 과에는 반드시 한명 이상의 직원이 있어야한다. 과는 반드시 부서(section)에 소속되어야하며, 부문에는 반드시 한 개 이상의 과가 있어야한다. 그리고 부문은 반드시 회사(company)에 소속되어야하며, 회사에는 반드시 한 개 이상의 부문이 있어야한다.

다음으로 팀 구조로 운영되는 측면에서 볼 것 같으면, 직원은 팀에 소속되기도 하고 그렇지 않기도 한다.

[상황 2] 직원이 팀에 소속되지 않는 경우는 바로 회사에 소속되며, 회사에는 반드시 한명 이상의 직원이 근무하여야한다. 직원이 팀에 소속되는 경우는,

[상황 3] 직원이 하나 이상의 팀(team)에 소속될 수 있으며, 팀에는 한 명이상의 직원이 근무하고 있다. 단, 초기 단계에는 팀만 구축하고 직원을 배정하지 않는 경우도 있을 수가 있다. 그리고 팀은 반드시 회사에 소속되어야하며, 회사에는 반드시 한 개 이상의 팀이 있어야한다.

이상의 사례 차원의 계층구조 스키마로 나타내면 그림 5와 같다.

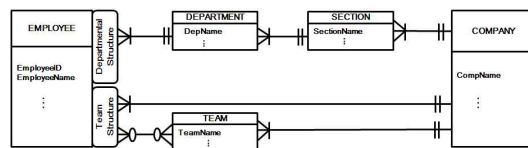


그림 5. 사례 차원의 계층구조 스키마
Fig. 5. A Hierarchy Schema for the Case Dimension

이하 차원의 계층구조 분류법과 사례 차원의 처리법을 설명하고자 한다.

V. 계층구조의 분할과 변환

1. 차원의 계층구조 분류법

저자들이 제안한 차원의 계층구조 분류법(그림 1참조)에 대해 먼저 설명함으로써 사례 차원의 처리법을 논하고자 한다.

분석기준의 수가 2개 이상인 대상 차원에 있어서 이 차원이 다중기준-독립(Multiple Criteria - Independent) 계층이면 최소한 분석기준의 수만큼 각자가 그림 1[A]의 ID 1~4 중 하나의 계층구조를 갖는 계층으로 분할한다. 이 중 ID 2~4인 것은 추가로 ID 1로 변환시킨다. 그리고 이 차원이 다중기준-비독립(Multiple Criteria - Dependent) 계층이면 그림 1(b)의 일반화 계층 개념에 따라 포괄적(Inclusive) 일반화 계층인지 배타적(Exclusive) 일반화 계층인지로 분할한다. 여기서 포괄적 계층인 경우는 그림 1[A]의 ID 1 계층으로 변환시켜야 하고, 배타적 계층인 경우는 그림 1[A]의 ID 1~8 중 하나의 계층구조를 갖는 계층으로 분할한다. 이 중 ID 2~4인 것은 ID 1로 변환시키고, ID 5~8 이면 그림 1(b)의 배타적 일반화 계층 개념에 따라 다

시 ID 2'4인 것으로 분할시킨 다음, 추가로 이들을 ID 1로 변환시킨다.

이상의 다중기준 집성계층구조 분류법에 관한 설명을 간단히 그림으로 나타내면 아래와 같다.

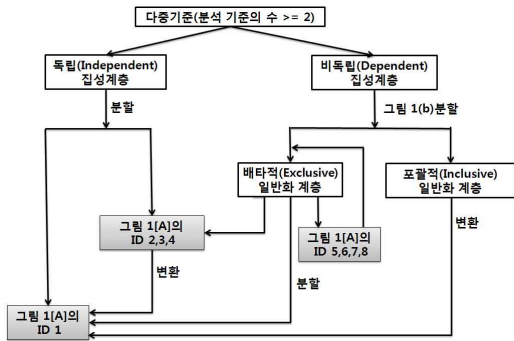


그림 6. 간단한 계층구조로 차원 계층구조의 분할
Fig. 6. A Dividing Process of the Dimension Hierarchy into Simple Hierarchies

2. 사례 차원의 분할

본 절에서는 앞의 차원 계층구조 분류법을 사례 차원에 적용시키고자 한다.

본 연구 사례를 나타내는 그림 5는 과 구조(Department Structure) 분석기준과 팀 구조(Team Structure) 분석기준의 분석기준을 2개 갖고 있는 차원으로, 계층 ID 8의 다중기준 집성계층으로 되어 있다. 그림 5에서 팀 구조 분석 기준을 갖고 있는 계층(상황 2와 3)은 COMPANY 수준을 공유하기 때문에 사례 차원은 다중기준-비독립 집성계층이다.

그림 1(b)의 개념에 따라 이 차원을 분할하는 단계를 이하 설명하고자 한다.

[단계 1] 계층 ID 8인 사례 차원은 배타적 일반화 계층 개념(계층 ID 10)에 따라 [상황1](그림 7참조)과 [상황2와 3](그림 8참조)의 계층구조로 분할한다.

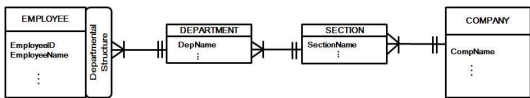


그림 7. 단일경로-직립성-대칭 집성계층(계층 ID 1)
Fig. 7. A Single-Path, Strict, Symmetric Aggregation Hierarchy

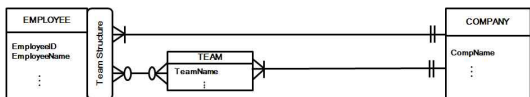


그림 8. 다중경로-비직립성-비대칭 집성계층(계층 ID 8)
Fig. 8. A Multiple-Paths, Non-Strict, Asymmetric Aggregation Hierarchy

그림 7은 ID 1의 계층구조를 갖고 있는 분배적 계층이다. 현 OLAP 체제에서 구현되는 계층이기 때문에 더 이상 고려하지 않아도 된다.

[단계 2] 그림 8은 ID 8의 계층구조를 갖고 있는 비분배적 계층이기 때문에 그림 1(b)의 개념에 따라 계층 ID 1의 [상황 2](그림 9참조)와 계층 ID 4의 [상황 3](그림 10참조)으로 분할한다. 그림 8의 팀 구조 분석기준은 팀에 소속하지 않는 분석기준(Team, Nonparticipated)(그림 9참조)과 팀에 소속하는 분석기준(Team, Participated)(그림 10참조)으로 변했음을 확인하길 바란다. 또한 그림 10에서 TEAM : EMPLOYEE의 최소 카디널 수가 M-O로 바뀌었음을 확인하길 바란다.

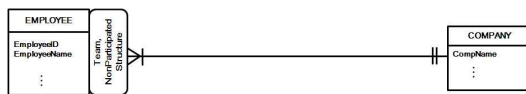


그림 9. 단일경로-직립성-대칭 집성계층(계층 ID 1)
Fig. 9. A Single-Path, Strict, Symmetric Aggregation Hierarchy

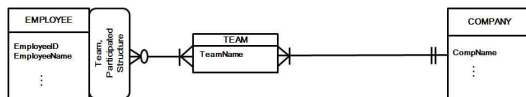


그림 10. 단일경로-비직립성-비대칭 집성계층(계층 ID 4)
Fig. 10. A Single-Path, Non-Strict, Asymmetric Aggregation Hierarchy

[단계 3] 그림 9는 ID 1의 계층구조를 갖고 있기 때문에 더 이상 고려하지 않아도 되지만, 계층 ID 4인 그림 10은 그림 1(b)의 개념에 따라 ID 12인 포괄적 비대칭 일반화 계층으로 분할함으로써 ID 1의 계층구조로 변환시켜야만 하는데, 이 과정은 VI에서 설명하기로 한다.

이상의 사례 차원 분할 과정을 점선으로 표시한 그림으로 나타낸 것이 그림 11이다.

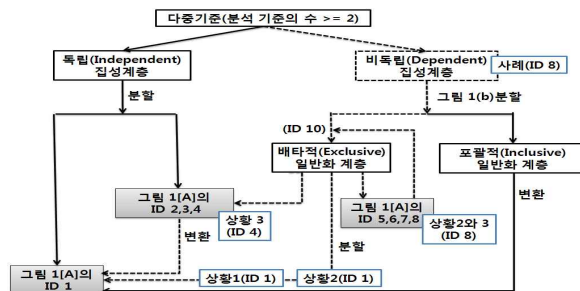


그림 11. 사례 차원의 분할
Fig. 11. Dividing the Case Dimension into Simpler Hierarchies

그림 11은 ID 8의 계층구조를 갖고 있는 비분배적 계층구조를 ID 4로 분할한 후, 이 계층 ID 4를 분배적 계층구조 스키마인 ID 1로 변환시켜야만 하는 사례를 설명한 것이다. 이외 나머지 6개 비분배적 계층구조 스키마의 분할과 변환에 대해 설명하고자 한다.

3. 비분배적 계층구조 스키마의 분할과 변환

3.1 계층 ID 7의 경우

사례에서는 초기단계에 팀만 구축하고 직원을 배정하지 않는 경우도 있는데 비해 이를 배제한 것이 계층 ID 7(그림 12 참조)의 경우이다. 계층 ID 7은 팀에 반드시 한명 이상의 직원이 근무하여야 한다. 이 계층은 비분배적 계층이기 때문에 그림 1(b)의 배타적 일반화 계층 개념(계층 ID 10)에 따라 계층 ID 1(그림 9참조)과 계층 ID 3(그림 13참조)으로 분할한다. 그림 13에서 TEAM : EMPLOYEE의 최소 카디널 수가 M-M로 바뀌었음을 확인하길 바란다. 그림 9는 계층 ID 1의 계층구조를 갖고 있기 때문에 더 이상 고려하지 않아도 되지만, 계층 ID 3은 그림 1(b)의 개념에 따라 ID 12인 포괄적 비대칭 일반화 계층으로 분할함으로써 ID 1의 계층구조로 변환시켜야만 한다.

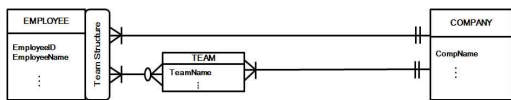


그림 12 다중경로-비직립성-대칭 집성계층(계층 ID 7)
Fig. 12 A Multiple-Paths, Non-Strict, Symmetric Aggregation Hierarchy

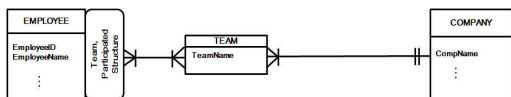


그림 13 단일경로-비직립성-대칭 집성계층(계층 ID 3)
Fig. 13 A Single-Path, Non-Strict, Symmetric Aggregation Hierarchy

3.2 계층 ID 6의 경우

계층 ID 7(그림 12참조)에서 TEAM : EMPLOYEE의 최대 카디널 수가 1:N로 바뀌고 최소 카디널 수가 M-O로 바뀌는 것이 계층 ID 6(그림 14참조)이다. 계층 ID 6은 직원이 하나의 팀에만 소속될 수 있으며, 초기 단계에는 팀만 구축하고 직원을 배정하지 않는 경우도 있을 수가 있다. 이 계층은 비분배적 계층이기 때문에 그림 1(b)의 배타적 일반화 계층 개념(계층 ID 10)에 따라 계층 ID 1(그림 9참조)과 계층 ID 2(그림 15참조)로 분할한다. 그림 9는 계층 ID 1의 계층구조를 갖고 있기 때문에 더 이상 고려하지 않아도 되지만, 계층 ID 2는 그림 1(b)의 개념에 따라 ID 12인 포괄적 비대칭 일반화 계층으로 분할함으로써 ID 1의 계층구조로 변환시켜야만 한다.

계층 ID 2(그림 15참조)에는 후손이 존재하지 않기 때문에 더 미 멤버를 삽입함으로써 계층 ID 1로 변환되게 되는 것이다.

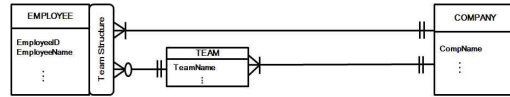


그림 14 다중경로-직립성-비대칭 집성계층(계층 ID 6)
Fig. 14 A Multiple-Paths, Strict, Asymmetric Aggregation Hierarchy

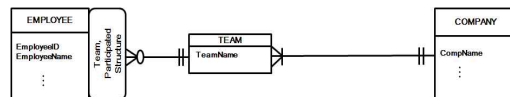


그림 15 단일경로-직립성-비대칭 집성계층(계층 ID 2)
Fig. 15 A Single-Path, Strict, Asymmetric Aggregation Hierarchy

3.3 계층 ID 5의 경우

계층 ID 6(그림 14참조)에서 TEAM : EMPLOYEE의 최소 카디널 수가 M-M으로 바뀐 것이 계층 ID 5(그림 16 참조)이다. 계층 ID 5는 팀에 반드시 한명 이상의 직원이 근무하여야 한다. 이 계층은 비분배적 계층이기 때문에 그림 1(b)의 배타적 일반화 계층 개념(계층 ID 10)에 따라 계층 ID 1(그림 9참조)과 계층 ID 1(그림 17참조)로 분할한다. 그림 9와 그림 17은 계층 ID 1의 계층구조를 갖고 있기 때문에 더 이상 고려하지 않아도 된다.

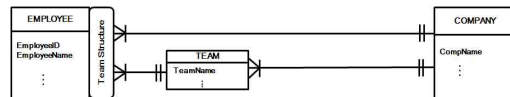


그림 16 다중경로-직립성-대칭 집성계층(계층 ID 5)
Fig. 16 A Multiple-Paths, Strict, Symmetric Aggregation Hierarchy

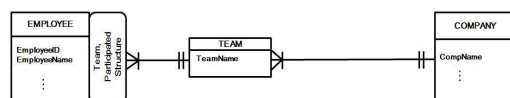


그림 17 단일경로-직립성-대칭 집성계층(계층 ID 1)
Fig. 17 A Single-Path, Strict, Symmetric Aggregation Hierarchy

이상에서 보듯이 계층 ID 6과 5의 경우는 현 OLAP 체제에서 구현될 수 있는 계층 ID 1까지 분할되었기 때문에 더 이상 고려할 필요가 없으나, 사례의 경우는 계층 ID 4까지, 계층 ID 7의 경우는 계층 ID 3까지 밖에 분할시키지 못했기 때문에 이 두 경우는 계층 ID 1까지의 계층구조로 변환시켜야만 한다.

한다. 이러한 변환과정을 VI에서 다루고자 한다.

VI. 현 OLAP 체제에서의 구현

1. 분배적 계층구조 스키마로의 변환과정

먼저 계층 ID 4까지지만 분할될 사례의 경우를 살펴보고자 한다. 계층 ID 4의 계층구조 스키마 형태인 그림 10에서 TEAM : EMPLOYEE의 최대 카디널 수가 N:M인데 이를 1:N으로 변환시키는 것이 분배적 계층구조 스키마로의 변환이다. 이하 이 변환과정을 설명하고자 한다.

TEAM : EMPLOYEE의 최대 카디널 수가 N:M이라는 것은 한명의 직원이 둘 이상의 팀에 소속된다는 뜻인데, EmployeeID 6 직원은 T1, T2, T3의 3개 팀에 소속되어 일하고 있고, EmployeeID 7 직원은 T2, T3의 2개 팀에 소속되어 일하고 있는 TEAM : EMPLOYEE의 최대 카디널 수가 N:M의 경우를 예를 들어 설명하고자 한다(그림 18참조).

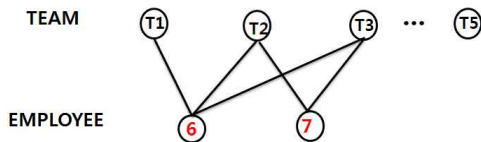


그림 18. 다 대 다 관계
Fig. 18. Many-to-Many Relationship

이 경우는 일반화 계층 개념으로 볼 때에는 포괄적 비대칭 일반화 계층(ID 12)이 된다. 계층 ID 12를 계층 ID 1로 변환시키는 작업을 본 연구에서는 각 직원을 소속된 TEAM에서 기여하는 일량(Duty)에 따라 분할시킴으로써 수행하고 있다. 직원 6은 T1에 40%, T2에 40%, T3에 20% 기여하는 식으로 근무한다고 하고, 직원 7은 T2에 40%, T3에 60% 기여하는 식으로 근무한다고 하자. 본 연구에서는 직원 6은 T1에 0.4, T2에 0.4, T3에 0.2의 일량(그림 19에서 괄호 안에 숫자로 표기)으로 근무하는 세 사람의 직원 6으로 분할하고, 직원 7은 T2에 0.4, T3에 0.6의 일량으로 근무하는 두 사람의 직원 7로 분할함으로써 TEAM : EMPLOYEE의 최대 카디널 수를 1:N으로 변환시키고 있다(그림 19, 표 4 참조).

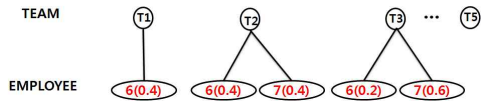


그림 19. 일 대 다 관계
Fig. 19. One-to-Many Relationship

이상의 변환 과정에 따라 그림 10을 현 OLAP 체제에서 구현 시킨 일례를 살펴보면 다음과 같다.

차원 테이블로는 EMPLOYEE 테이블(표 2 참조)과 TEAM(표 3 참조) 테이블이 있다. 표 3에서 특별행사팀(T5)은 팀원이 없는 이름만의 팀임에 유의하기 바란다.

표 2 팀 참여 EMPLOYEE 차원 테이블
Table 2. Team Participated EMPLOYEE Dimension Table

EmployeeID	EmployeeName
1	강창모
2	권일훈
⋮	⋮
6	설원석
7	송광민
⋮	⋮
20	김종진

표 3. TEAM 차원 테이블
Table 3. TEAM Dimension Table

TeamID	TeamName
T1	기획전략팀
T2	운영관리팀
T3	경영지원팀
T4	재무회계팀
T5	특별행사팀

TEAM : EMPLOYEE의 최대 카디널 수가 N:M인 것을 1:N로 변환시키는 1절의 과정을 표 4와 같이 측정값 Duty을 갖는 일량 사실 테이블을 채택함으로써 현실화 시켰다.

표 4. 일량 사실 테이블
Table 4. Duty Fact Table

FactID	EmployeeID	TeamID	Duty
1	1	T1	1
2	2	T4	1
3	3	T2	1
4	4	T2	1
5	5	T2	1
6	6	T1	0.4
7	6	T2	0.4
8	6	T3	0.2
9	7	T2	0.4
10	7	T3	0.6
⋮	⋮	⋮	⋮
25	20	T3	1

이상의 팀 참여 EMPLOYEE 차원 테이블, TEAM 차원 테이블 그리고 일량 사실 테이블을 사용하여 OLAP 큐브를 구성해 보면 그림 20과 같고 이를 실행 시켜서 팀별로 전개 시킨 결과가 그림 21이다. 그림 20에서 EMPLOYEE 차원

위 변환과정에서 보듯이, 선 분할-후 변환 원칙을 고수하고 있는 것이 기존 여러 연구[10],[18]와는 다른 본 연구의 특징이라고 할 수 있겠다. 복잡한 계층을 우선 분할하여 간단한 계층으로 전환시킨 다음 이를 분배적 계층으로 변환시킨다는 원칙을 지키고 있다는 뜻이다.

본 연구에서는 이상의 변환과정을 통해 실세계에서 일어날 수 있는 모든 계층구조 스키마를 현 OLAP 체제에서 잘 다룰 수 있게끔 하고 있다고 결론 내릴 수 있다. 단, 본 연구에서는 실세계에서 일어날 수 있는 모든 계층구조 스키마를 그림 2에서와 같이 12개로 정리하고 있는데, 이에 대한 심층 연구가 더 필요하다고 생각한다.

참고문헌

- [1] John Miles Smith, and Diane C.P.Smith, "Database Abstractions : Aggregation and Generalization", ACM Transactions on Database Systems, Vol. 2, No. 2, pp. 105 - 133, June. 1977.
- [2] Robert S. Craig, Joseph A. Vivona, and David Berkovitch, "Microsoft Data Warehousing : Building Distributed Decision Support Systems", pp. 59-66, John Wiley & Sons, 1999.
- [3] Erik Thomsen, "OLAP Solutions", pp. 38-40, John Wiley & Sons, 2002.
- [4] SeHyeon Jang, HanJu Yu, and InSoo Choi, "Design of a Hierarchical Dimension of the Bill of Materials Type", KSCI, Vol. 11, No. 4, pp. 244-250, September. 2006.
- [5] Carlos A. Hurtado, and Alberto O. Mendelzon. Jensen, "Reasoning about Summarizability in Heterogeneous Multidimensional Schemas", LNCS 1973, pp. 375 - 389, 2001.
- [6] J. Gray, A. Bosworth, A. Layman, and H. H. Pirahesh, "Data cube : A relational operator generalizing group-by, cross-tab and sub-totals", In Proceedings of the 12th IEEE-ICDE Conference, New Orleans, Los Angeles, USA, 1996.
- [7] SeungHyun Lee, DuckSung Lee and InSoo Choi, "Applying an Aggregate Function AVG to OLAP Cubes", KSCI, Vol. 14, No. 1, pp. 217-228, January. 2009.
- [8] Svetlana Vinnik, and Florian Mansmann, "From Analysis to Interactive Exploration: Building Visual Hierarchies from OLAP Cubes", LNCS 3896, pp. 496-514, 2006.
- [9] Mihwa oh, ManMo Hwang, JungWoo Choi and InSoo Choi, "An Approach to Navigating Data Cubes with a Hierarchical Visualization Technique", KSCI, Vol. 16, No. 2, pp. 290-305, February. 2011.
- [10] E. Malinowski, and E. Zimányi, "OLAP Hierarchies: A Conceptual Perspective", LNCS 3084, pp.477-491, 2004
- [11] David M. Kroenke, "Database Processing", PEARS ON, pp. 125-188, 2010.
- [12] John Miles Smith, and Diane C.P.Smith, "Database Abstractions : Aggregation", ACM, pp. 405 - 413, June. 1977.
- [13] Erik Thomsen, "OLAP Solutions", pp. 132, John Wiley & Sons, 2002.
- [14] Svetlana Mansmann, and Marc H. Scholl, "Extending Visual OLAP for Handling Irregular Dimensional Hierarchies", LNCS 4081, pp. 95 - 105, 2006.
- [15] H. J. Lenz, and A. Shoshani, "Summarizability in OLAP and statistical data bases", In Proceedings of 9th International Conference on Scientific and Statistical Database Management, pp. 132 - 43, 1997.
- [16] Torben Bach Pedersen, Christian S. Jensen, and Curtis E. Dyreson, "A foundation for capturing and querying complex multidimensional data", Information Systems Vol. 26, pp. 383 - 423, 2001.
- [17] Torben Bach Pedersen, and Christian S. Jensen, "Multidimensional Data Modeling for Complex Data", A Time Center Technical Report, pp. 1 - 27, 1998.
- [18] Carlos A. Hurtado and Claudio Gutiérrez, "Equivalence of OLAP Dimension Schemas", LNCS, Vol. 2942, pp. 176-195, 2004

저 자 소 개



오 미 화

2002 : 성결대학교 정보통신학과
공학사.

2004 : 숭실대학교 산업공학과
공학석사.

현 재 : 숭실대학교 대학원 산업·
정보시스템공학과 박사과정

관심분야 : MIS, DW, OLAP

Email : ohmihwa@empas.com



최 인 수

1985 : 서울대학교 산업공학과
공학박사

현재 : 숭실대학교 산업·정보시스템
공학과 교수

관심분야 : MIS, DW, OLAP

Email : ischoi@ssu.ac.kr