

	(19) 대한민국특허청(KR) (12) 공개특허공보(A)	(11) 공개번호 10-2009-0068581 (43) 공개일자 2009년06월29일
(51) Int. Cl. <i>G06F 17/00</i> (2006.01) <i>G06F 17/30</i> (2006.01) (21) 출원번호 10-2007-0136258 (22) 출원일자 2007년12월24일 심사청구일자 2007년12월24일	(71) 출원인 계명대학교 산학협력단 대구 달서구 신당동 1000번지 (72) 발명자 홍동권 대구 달서구 이곡동 서한2차아파트 104-708호 (74) 대리인 정홍식	
전체 청구항 수 : 총 3 항		

(54) 동적 XML의 노드 번호 부여방법

(57) 요약

동적 XML의 노드 번호 부여방법이 개시된다. 본 발명의 노드 번호 부여방법은 노드들의 제1필드에 부모노드 및 자식노드 간의 관계를 나타내는 단계, 및 노드들의 제2필드에 형제노드들 간의 관계를 나타내는 단계를 포함한다. 이에 따라, 노드의 삽입 및 삭제를 손쉽게 처리하여 데이터를 효율적으로 관리할 수 있게 된다.

대표도 - 도1

P필드	S필드
-----	-----

특허청구의 범위

청구항 1

XML(eXtensible Markup Language)트리에 포함된 노드들에 번호를 부여하는 동적 XML의 노드 번호 부여방법에 있어서,

상기 노드들의 제1필드에 부모노드 및 자식노드 간의 관계를 나타내는 단계; 및

상기 노드들의 제2필드에 형제노드들 간의 관계를 나타내는 단계;를 포함하는 것을 특징으로 하는 노드 번호 부여방법.

청구항 2

제1항에 있어서,

상기 XML트리에 새로운 노드가 삽입되는 경우, 삽입된 노드의 제1필드에는 상기 삽입된 노드의 형제노드들의 제1필드에 부여된 번호 중, 최대값을 1만큼 증가시킨 번호가 부여되며, 상기 삽입된 노드의 위치에 따라, 상기 삽입된 노드의 제2필드에 번호가 부여되며, 상기 형제노드들의 제2필드에 부여된 번호가 변경 혹은 유지되는 것을 특징으로 하는 노드 번호 부여방법.

청구항 3

제1항에 있어서,

상기 XML트리에 존재하는 노드가 삭제되는 경우, 삭제된 노드의 위치에 따라, 상기 삭제된 노드의 왼쪽 혹은 오른쪽에 존재하는 형제노드들의 제2필드에 부여된 번호가 변경되는 것을 특징으로 하는 노드 번호 부여방법.

명세서

발명의 상세한 설명

기술 분야

- <1> 본 발명은 동적 XML의 노드 번호 부여방법에 관한 것으로, 더욱 상세하게는 대용량 XML(eXtensible Markup Language)데이터를 관리하는 관계형 데이터베이스에서, 동적 XML의 노드 번호 부여방법에 관한 것이다.

배경 기술

- <2> 일반적으로, 관계형 데이터베이스에서 XML데이터를 관리하기 위해서는 XML노드에 번호를 부여하여 관계형 테이블에 저장하는 방식을 사용한다. 여기서, 사용되는 번호 부여방식은 XML데이터의 계층 정보를 모두 표현할 수 있어야 하며, 새로운 노드의 삽입, 기존 노드의 삭제, 및 노드의 검색 질의를 효율적으로 처리할 수 있어야 한다. 그러나, 초기의 번호 부여방식은 새로운 노드의 삽입과 삭제시 모든 번호를 다시 부여해야 하므로, XML데이터를 효과적으로 관리할 수 없다.
- <3> 최근에는, XML데이터에 효과적으로 번호를 부여하여, XML데이터를 관리하기 위해 Dewey Order방식 및 ORDPATH방식을 사용하고 있다. Dewey Order방식은 노드의 삽입 및 삭제가 없는 정적인 XML문서에 주로 사용되고 있으며, ORDPATH방식은 XML데이터의 일부분이 변경되어 새로운 노드의 삽입 및 노드의 삭제가 발생하는 동적인 XML문서에 주로 사용되고 있다.
- <4> 여기서, Dewey Order방식은 새로운 노드가 삽입될 때마다, 삽입된 노드의 위치 이후에 존재하는 모든 노드들의 번호를 다시 부여해야 하는 단점이 있다. 그리고, ORDPATH방식은 홀수만을 사용하며, 노드의 삽입 및 삭제가 발생하면 번호의 레벨이 깊어져, 같은 레벨에 있는 노드들의 번호레벨이 서로 다르게 되며, XML 질의 처리에 많은 데이터 액세스가 필요한 단점이 있다.

발명의 내용

해결 하고자하는 과제

- <5> 본 발명의 목적은 정적인 XML문서에 사용할 수 있을 뿐만 아니라, 동적인 XML문서에서 노드의 삽입 및 삭제를 효율적으로 처리하고, 데이터 액세스 횟수를 줄일 수 있는 동적 XML의 노드 번호 부여방법을 제공함에 있다.

과제 해결수단

- <6> 상기 목적을 달성하기 위한 본 발명에 따른 관계형 데이터베이스에서 XML(eXtensible Markup Language)트리에 포함된 노드들에 번호를 부여하는 동적 XML의 노드 번호 부여방법은 상기 노드들의 제1필드에 부모노드 및 자식 노드 간의 관계를 나타내는 단계, 및 상기 노드들의 제2필드에 형제노드들 간의 관계를 나타내는 단계를 포함한다.
- <7> 상기 XML트리에 새로운 노드가 삽입되는 경우, 삽입된 노드의 제1필드에는 상기 삽입된 노드의 형제노드들의 제1필드에 부여된 번호 중, 최대값을 1만큼 증가시킨 번호가 부여되며, 상기 삽입된 노드의 위치에 따라, 상기 삽입된 노드의 제2필드에 번호가 부여되며, 상기 형제노드들의 제2필드에 부여된 번호가 변경 혹은 유지되는 것을 특징으로 한다.
- <8> 상기 XML트리에 존재하는 노드가 삭제되는 경우, 삭제된 노드의 위치에 따라, 상기 삭제된 노드의 왼쪽 혹은 오른쪽에 존재하는 형제노드들의 제2필드에 부여된 번호가 변경되는 것을 특징으로 한다.

효과

- <9> 이상 설명한 바와 같이 본 발명에 따르면, 노드의 삽입 및 삭제를 효율적으로 처리하고, 데이터 액세스 횟수를 줄여, 관계형 데이터베이스에서 XML데이터를 효율적으로 관리할 수 있게 된다.

발명의 실시를 위한 구체적인 내용

- <10> 이하에서는 첨부된 도면들을 참조하여 본 발명의 일 실시예를 보다 상세하게 설명한다. 다만, 본 발명을 설명함에 있어서, 관련된 공지 기능 혹은 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우, 그에 대한 상세한 설명은 축약하거나 생략한다.
- <11> 도 1은 본 발명의 일 실시예에 따른 노드 번호 부여방법이다.
- <12> 도 1을 참조하면, 본 발명의 노드 번호는 P필드 및 S필드로 구분된다. P필드는 부모노드와 자식노드 간의 관계를 나타내는 필드이며, S필드는 형제노드들 간의 관계를 나타내는 필드이다. 여기서, P필드에는 XML트리의 최상위계층에 속한 노드부터 깊이 우선 탐색하여 번호가 부여되는 Dewey Order방식이 적용된다. 그리고, S필드에는 정수를 사용하여 하나의 계층에 포함된 형제노드들 간의 관계를 나타낸다.
- <13> 도 2는 본 발명의 일 실시예에 따른 노드 번호 부여방법에 따라, 번호가 부여된 XML트리를 나타낸 도면이다.
- <14> 도 2를 참조하면, 모든 노드들은 부모노드와 자식노드 간의 관계를 나타내는 P필드 및 형제노드들 간의 관계를 나타내는 S필드로 구성된다. 제1계층에 포함된 노드 A의 경우, 최상위계층의 노드이므로 P필드에는 '1'이 부여되며, 제1계층에 다른 형제노드가 존재하지 않으므로 S필드에는 '1'이 부여된다. 그리고, 제2계층에 포함된 노드 B는 노드 A의 첫 번째 자식노드이므로, P필드에는 '1.1'이 부여되고, S필드에는 '1'이 부여된다.
- <15> 같은 방법으로, 노드 C는 노드 A의 두 번째 자식노드이므로 P필드에는 '1.2'가 부여되고, S필드에는 '2'가 부여되며, 노드 D는 노드 B의 세 번째 자식노드이므로 P필드에는 '1.3'이 부여되고, S필드에는 '3'이 부여된다. 제3계층 및 제4계층에 속한 노드 E,F,G,H, 및 I의 경우에도 위와 같은 동일한 방법으로 노드 번호가 부여된다.
- <16> 여기서, 각각의 노드에는 S필드의 범위를 나타내기 위해 시작번호(sno) 및 끝 번호(eno)를 나타내는 정보가 포함되며, 새로 삽입되는 노드의 P필드에 고유한 값을 부여하기 위해 자식노드의 P필드에 부여된 값 중 최대값(pmax)이 부가적인 정보로 유지된다. 도 2의 제2계층을 예로 들면, sno=1, eno=3, 그리고, pmax=3이며, 이러한 정보들을 이용하여 노드 A의 자식노드의 개수 $n = eno - sno + 1$ 을 통해 3으로 쉽게 구해진다.
- <17> 도 3a 내지 도 3c는 본 발명의 일 실시예에 따른 노드 번호 부여방법에 따라, 새로운 노드가 삽입되는 경우를 설명하기 위한 도면이다.
- <18> 새로운 노드가 삽입되는 경우는 중간 자식노드로 삽입되는 경우, 첫 번째 자식노드로 삽입되는 경우, 그리고, 마지막 자식노드로 삽입되는 경우, 이렇게 3가지 경우로 나눌 수 있다.
- <19> 도 3a에는 XML트리에서, 노드(1.2)의 두 번째 자식노드로 새로운 노드가 삽입된 경우를 나타내었으며, 이러한

노드 삽입과정을 도 3b와 같이 나타낼 수 있다. 도 3a 및 도 3b를 참조하면, 새로 삽입된 노드의 P필드는 (1.2.3)을 부여받음으로써, 부모노드(1.2)와의 관계를 나타낸다. 그리고, 새로 삽입된 노드가 부모노드(1.2)의 두 번째 자식노드가 되므로, 첫 번째 노드(1.2.1)의 S필드에 주어진 값보다 1만큼 큰 '2'를 부여받게 되며, 형제노드들 간의 순서를 유지하기 위해, 다른 형제노드(1.2.2)의 S필드에는 기존의 값보다 1만큼 증가된 '3'을 부여받게 된다.

- <20> 여기서, 노드(1.2.2)의 자식노드들은 부모노드인 노드(1.2.2)의 P필드에 부여된 번호가 변경되지 않았으므로, 자식노드들의 번호 역시 변경되지 않게 된다. 그리고, 자식노드의 개수 n 이 큰 경우, 새로운 노드의 삽입 위치는 $n/2$ 를 중심으로 왼쪽 혹은 오른쪽인지를 검사하여 형제노드들의 S필드에 부여되는 번호를 변경하게 된다. 따라서, 새로운 노드 삽입 시, 평균 $n/4$ 개의 형제노드들의 번호 변경이 필요하다. 이때, 액세스되는 노드는 부모노드인 노드(1.2), 새로 삽입되는 노드(1.2.3), 그리고, $n/4$ 개의 형제노드들이다.
- <21> 도 3c에는 XML트리에서, 노드(1.2)의 첫 번째 자식으로 새로운 노드가 삽입된 경우를 나타내었다.
- <22> 새로 삽입된 노드의 P필드는 (1.2.4)을 부여받음으로써, 부모노드(1.2)와의 관계를 나타낸다. 그리고, 새로 삽입된 노드가 부모노드(1.2)의 첫 번째 자식노드가 되므로, S필드에 '0'을 부여받게 되어, 형제노드들 간의 순서는 유지되게 된다. 여기서도 역시, 형제노드(1.2.2)의 자식노드들은 부모노드인 노드(1.2.2)의 P필드에 부여된 번호가 변경되지 않았으므로, 자식노드들의 번호 역시 변경되지 않게 된다.
- <23> 도 3c에서, $eno=3$, $sno=0$ 이므로, 노드(1.2)의 자식노드의 개수 $n = eno - sno + 1$ 을 통해 4로 구해진다. 이때, 액세스되는 노드는 부모노드인 노드(1.2) 및 새로 삽입된 노드(1.2.4)이며, 다른 노드들은 전혀 액세스되지 않는다.
- <24> 도시하지는 않았으나, 노드(1.2)의 마지막 자식으로 새로운 노드가 삽입되는 경우, 노드(1.2)의 $pmax=4$ 를 이용하여, 새로운 노드의 P필드에는 (1.2.5)가 부여되고, S필드에는 eno 값이 1만큼 증가한 '4'가 부여된다. 여기서, 액세스되는 노드는 부모노드인 노드(1.2) 및 새로 삽입된 노드(1.2.5)이며, 다른 노드들은 전혀 액세스되지 않는다.
- <25> 이상에서 설명한 과정을 통해 XML트리에 새로운 노드가 삽입되는 경우, 각각의 노드에 효과적으로 번호를 부여하는 것이 가능하다.
- <26> 도 4a 및 도 4b는 본 발명의 일 실시예에 따른 노드 번호 부여방법에 따라, 노드가 삭제되는 경우를 설명하기 위한 도면이다.
- <27> 노드가 삭제되는 경우는 중간 자식노드가 삭제되는 경우, 첫 번째 자식노드가 삭제되는 경우, 그리고, 마지막 자식노드가 삭제되는 경우, 이렇게 3가지 경우로 나눌 수 있다.
- <28> 도 4a에는 도 3c의 XML트리에서, 노드(1.2)의 두 번째 자식노드가 삭제된 경우를 나타내었으며, 이러한 노드 삭제과정을 도 4b와 같이 나타낼 수 있다. 도 4a 및 도 4b를 참조하면, 자식노드(1.2.1)가 삭제되는 경우, 형제노드(1.2.4)의 S필드에는 1만큼 증가한 '1'이 주어지고, $pmax=3$, $sno=1$ 로 변경된다. 만약, 삭제된 노드(1.2.1)에 자식노드가 존재하는 경우, 이도 함께 삭제되는 것이 바람직하다.
- <29> 그리고, 자식노드의 개수가 n 인 경우, 삭제되는 노드의 위치가 $n/2$ 를 중심으로 왼쪽 혹은 오른쪽인지를 검사하여 형제노드들의 S필드에 부여되는 번호를 변경하게 된다. 즉, 삭제되는 노드의 위치가 $n/2$ 를 중심으로 왼쪽인 경우, 삭제되는 노드의 왼쪽에 위치한 형제노드들의 S필드에 부여된 번호가 변경된다. 따라서, 노드 삭제 시, 액세스되는 노드는 부모노드인 노드(1.2), 삭제되는 노드(1.2.1), 그리고, $n/4$ 개의 형제노드들이다.
- <30> 도시하지는 않았으나, 노드(1.2)의 첫 번째 자식노드가 삭제되는 경우, $pmax$ 값은 1만큼 감소하며, sno 값은 1만큼 증가한다. 이때, 액세스되는 노드는 부모노드 및 삭제되는 노드뿐이다. 그리고, 노드(1.2)의 마지막 자식노드가 삭제되는 경우, $pmax$ 값이 1만큼 감소하며, eno 값은 1만큼 감소한다. 이때, 액세스되는 노드 역시, 부모노드 및 삭제되는 노드뿐이다.
- <31> 이상에서 설명한 과정을 통해, XML트리에서 노드가 삭제되는 경우, 효과적으로 번호를 부여하는 것이 가능하다.
- <32> 이상에서는 본 발명의 바람직한 실시 예에 대하여 도시하고 설명하였지만, 본 발명은 상술한 특징의 실시 예에 한정되지 아니하며, 청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술분야에서 통상의 지식을 가진 자라면 누구든지 다양한 변형 실시 가능한 것은 물론이고, 그와 같은 변경은 청구범위 기재의 범위 내에 있게 된다.

도면의 간단한 설명

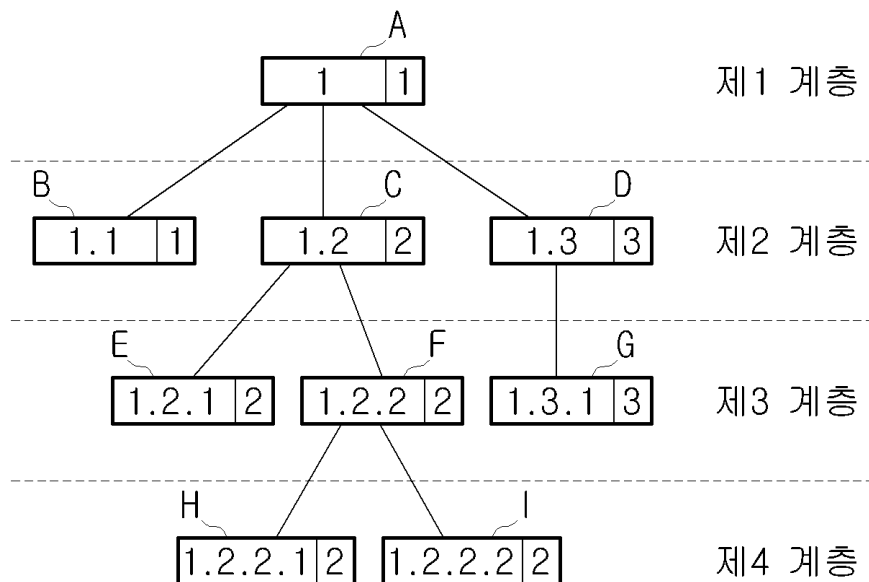
- <33> 도 1은 본 발명의 일 실시예에 따른 노드 번호 부여방법,
- <34> 도 2는 본 발명의 일 실시예에 따른 노드 번호 부여방법에 따라, 번호가 부여된 XML트리를 나타낸 도면,
- <35> 도 3a 내지 도 3c는 본 발명의 일 실시예에 따른 노드 번호 부여방법에 따라, 새로운 노드가 삽입되는 경우를 설명하기 위한 도면, 그리고
- <36> 도 4a 및 도 4b는 본 발명의 일 실시예에 따른 노드 번호 부여방법에 따라, 노드가 삭제되는 경우를 설명하기 위한 도면이다.

도면

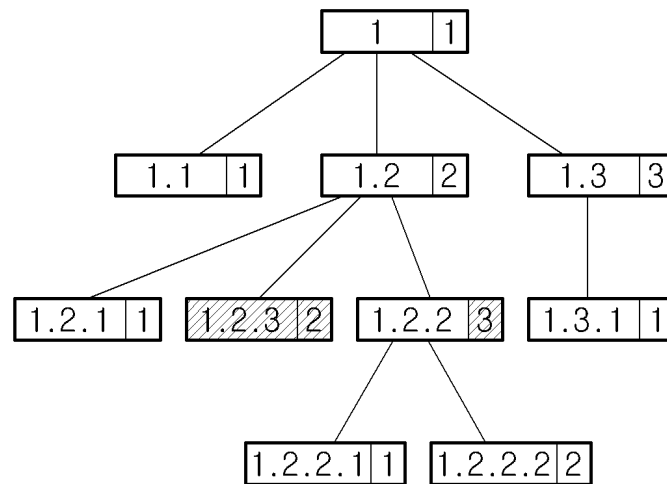
도면1



도면2



도면3a

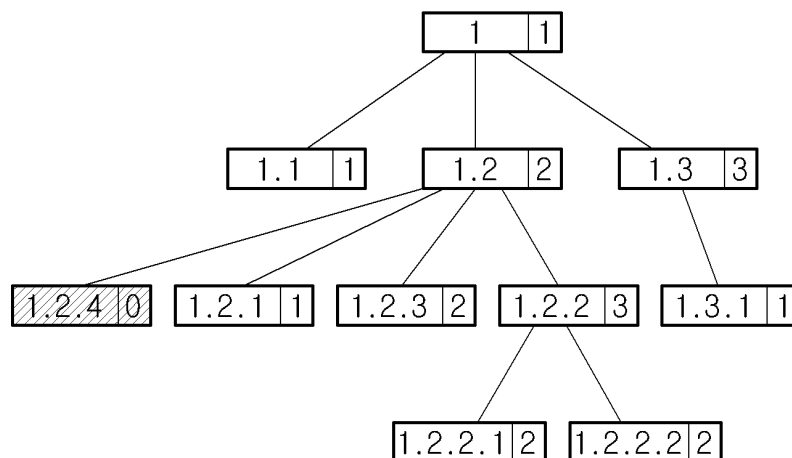


도면3b

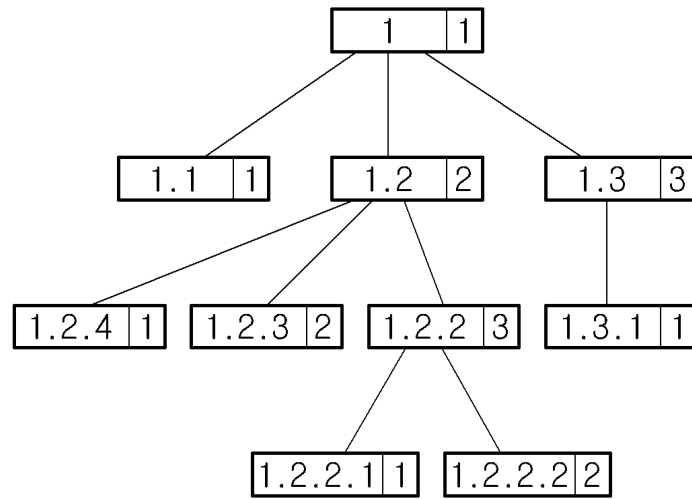
```

procedure add()
{
  pare = 부모 노드 (1.2)를 찾는다.
  max = pare.pmax
  새로운 노드의 p 필드 값을 (1.2.max + 1)로 부여 (1.2.3)
  노드 (1.2.3)은 2번째 자식이 되므로 s 필드 값이 2가 된다. (1.2.3, 2)
  노드 (1.2.3)의 모든 오른쪽 형제들의 s 필드 값은 1씩 증가.
  부모 노드의 pmax를 3으로, eno의 값을 1 증가
}
  
```

도면3c



도면4a



도면4b

```

procedure delete(node)
{
  pare = parent(node); // 노드 (1.2)
  노드 (1.2.1)의 모든 왼쪽 형제들의 s 필드 값을 1씩 증가.
  부모 노드의 pmax를 1 감소, sno의 값을 1 증가
}
  
```