Journal of Korea Multimedia Society Vol. 12, No. 1, January 2009(pp. 1-15)

XML 질의처리를 위한 다차원 타입상속 색인구조의 할당기법

이 종 학†

요 약

본 논문에서는 XML 데이터베이스에서 XML 질의처리를 효율적으로 지원하기 위한 다차원 타입상속 색인구조(MD-TIX)들의 할당기법을 제시한다. MD-TIX는 중첩요소와 여러 타입상속 계층으로 이루어진 중첩술어의 처리를 효율적으로 지원하기 위하여 다차원 색인구조를 이용하는 색인기법이다. 본 논문에서는 타겟 타입 또는 도메인 타입의 대치가 있는 Xpath로 표현된 여러 중첩술어들의 접속으로 구성된 복합질의의 관점에서 MD-TIX 색인들의 할당에 따른 질의처리 기법들을 분석하고, 그 결과로서 가장 효과적인 색인 할당기법을 제시한다. 먼저, XML 문서의 변경에 따른 MD-TIX 색인구조의 운용과 하나의 중첩술어를 가지는 질의처리에 대한 MD-TIX 색인의 할당에 대하여 분석한다. 그리고 경로들 사이에 공통의 부경로가 있는 겹침 경로 상에 주어지는 여러 개의 중첩술어들로 구성된 보다 일반적인 질의의 관점에서 MD-TIX 색인의 운용과 그 할당기법을 제시한다.

An Assignment Method of Multidimensional Type Inheritance Indexes for XML Query Processing

Jong-Hak Lee†

ABSTRACT

This paper presents an assignment method of the multidimensional type inheritance indexes (MD-TIXs) to support the processing of XML queries in XML databases. MD-TIX uses a multidimensional index structure for efficiently supporting nested predicates that involve both nested element and type inheritance hierarchies. In this paper, we have analyzed the strategy of the query processing by using the MD-TIXs, and presented an assignment method of the MD-TIXs in the framework of complex queries, containing conjunctions of nested predicates, each one involving an Xpath having target types or domain types substitution. We first consider MD-TIX operations caused by updating of XML databases, and the use of the MD-TIXs in the case of a query containing a single nested predicate. And then, we consider the assignments of the MD-TIXs in the framework of more general queries containing nested predicates over overlapping paths that have common subpaths.

Key words: XML Documents(XML 문서), XML Schema(XML 스키마), XML Query Processing(XML 질의처리), XML Index(XML 색인)

1. 서 론

XML[1] 데이터베이스 관리 시스템의 질의처리 성능 최적화는 중요한 연구과제이다. 최근에 제안된 XML 데이터베이스 색인기법들은 XML 질의처리의 성능 향상에 크게 기여하고 있다[2,3]. 그러나, 이들

※ 교신저자(Corresponding Author) : 이종학, 주소 : 경북 경산시 하양읍 금락1리 330(712-702), 전화 : 053)850-2746, FAX : 053)850-2746, E-mail : jhlee11@cu.ac.kr

접수일 : 2008년 8월 12일, 완료일 : 2008년 10월 16일

† 정회원, 대구가톨릭대학교 컴퓨터정보통신공학부 교수

색인구조들은 기존의 관계형 데이터베이스의 단순 속성에 대한 색인구조에 비해 저장공간 및 갱신유지 비용에 큰 부담이 있다. 또한, 색인구조의 종류에 따라 검색 성능이 다른 특성도 있다[2,3]. 그러므로 XML 데이터베이스 색인기법을 통한 질의처리의 이점이 색인구조의 저장공간 및 갱신유지를 위한 부담으로 인해 상쇄되지 않기 위해서는 색인들을 매우 신중히 할당하여야 하며, 효과적인 색인할당 방법에 관한 연구가 필수적이라 할 수 있다.

XML 데이터베이스[4]에서 XML 스키마는 타입 상속(type inheritance) 개념에 의하여 타입들 사이에 타입상속 계층을 이룬다. 즉, 하나의 타입은 여러 개의 서브타입을 가지며, 각 서브타입은 또 다른 여러 서브타입들을 가진다[5]. 이로 인하여 XML 데이터베이스에서는 하나의 질의에 대한 대상 범위를 두 가지 경우로 해석할 수 있다. 한 경우는 질의의 대상 범위를 질의에 나타나는 타입만으로 한정하는 것이고, 또 다른 경우는 질의의 대상 범위를 질의에 나타나는 타입과 그의 모든 서브타입들을 포함하는 것이다. 이러한 타입상속 계층에 대한 질의를 효율적으로 처리할 수 있는 색인구조는 특정 타입에 속하는 요소(element)들의 탐색뿐만 아니라 특정 타입을 루트로 하는 타입상속 계층의 모든 타입들에 속하는 요소들의 탐색도 효율적으로 처리할 수 있어야 한다.

XML 스키마는 또 하나의 중요한 개념인 타입 집단화(type aggregation) 개념에 의하여 한 타입이 가지는 요소의 도메인이 또 다른 타입이 될 수 있도록 함으로써(이러한 요소를 *복합요소(complex element)*라 함) 타입들 사이에 타입 집단화 계층이라는 또 다른 계층구조를 이룬다. 따라서 타입 집단화 계층을 이루는 타입들에서 정의된 어떠한 요소도 논리적으로는 루트 타입의 요소라고 볼 수 있다. 본 논문에서는 타입 집단화 계층에서 루트 타입이 아닌 타입에서 정의된 요소를 루트 타입의 *중첩요소(nested element)*[6,7]라 한다. 이로 인하여 XML 질의어에서는 중첩요소에 조건이 주어지는 *중첩술어(nested predicate)*[8]를 가진다는 특징이 있다. XML 질의어에서는 중첩요소를 표현하기 위해서 Xpath[6]와 같은 경로 표현식을 사용한다. 경로 표현식은 루트 타입으로부터 타입 집단화 계층을 따라 나타나는 요소들의 나열로 표현한다.

중첩요소를 표현하는 Xpath에는 요소들 사이의 중첩관계에 의한 요소와 요소 사이에 *암시적 조인(implicit join)*[9,10]의 의미를 가지고 있다. 이러한 암시적 조인은 XML 스키마에 의해 미리 예상이 가능하다. 따라서 질의에 자주 나타나는 중첩요소에 대한 암시적 조인을 미리 계산하여 그 결과를 색인으로 구축하여 놓음으로써, 질의처리시 이를 이용하여 성능 향상을 꾀할 수 있으며 이를 중첩요소에 대한 색인기법[2,7,11-13]이라 한다. 그러나 이러한 중첩요소에 대한 기존의 색인기법들은 모두 일차원 색인구조인 B^+-tree[6]를 이용함으로써, 타입 상속의 특징으로 인한 질의의 대상 범위가 타입상속 계층상의 임의의 타입들로 제한되거나, Xpath에 나타나는 요소의 도메인이 타입상속 계층상의 임의의 타입들로 제한이 되는 질의들을 지원하기 어려운 문제점을 가지고 있다.

이와 같은 일차원 색인구조를 이용하는 기존의 중첩요소에 대한 색인기법들이 가지는 타입상속 계층의 지원 문제를 해결하기 위하여, 다차원 색인구조[15-17]를 중첩요소에 대한 색인구조로 이용할 수 있으며 이를 *다차원 타입상속 색인구조*[18]라 한다. 다차원 타입상속 색인구조에서는 중첩요소의 키 값 도메인과 함께, Xpath에 나타나는 루트 타입의 타입상속 계층과 각 복합요소의 도메인 타입상속 계층마다 한 축의 타입 식별자 도메인을 할당하여 다차원 색인구조를 구성한다. 이와 같은 색인기법에서는 기존의 일차원 색인구조를 이용한 색인기법들에서 문제가 되는 질의의 대상 범위가 타입상속 계층상의 임의의 타입들로 제한되거나, 경로 표현식에 나타나는 요소의 도메인이 타입상속 계층상의 임의의 타입들로 제한이 되는 질의들의 처리를 한번의 색인 탐색으로 가능하게 할 수 있다.

다차원 타입상속 색인구조는 색인 엔트리를 타겟 요소로 구성하는 *다차원 타겟요소 색인구조*와 색인 엔트리를 경로 인스턴스로 구성하는 *다차원 경로 색인구조*의 두 가지 형태가 있다. 본 논문에서는 먼저, 이 두 가지 형태의 다차원 타입상속 색인구조에 대해서 XML 문서의 변경에 따른 색인구조의 운영과 유지비용 면에서 각각의 한계점을 바탕으로 하나의 중첩술어를 가지는 질의처리에 대한 색인의 할당에 대하여 분석한다. 그리고, 두 개 이상의 경로상에 각각 주어진 술어들로 구성된 복합질의의 경우, 주어진 경로 스키마상에서 다차원 타입상속 색인구조를 할당

할 수 있는 다양한 방법을 제시하고, 각 할당방법에 따른 질의처리의 성능을 비교분석 한다. 그 결과로 다차원 타입상속 색인구조의 효과적 할당기법을 제안한다.

본 논문의 구성은 다음과 같다. 먼저, 제 2절에서는 관련 연구로서 XML 데이터베이스의 색인 구축에 필요한 기본 개념으로 XML 질의어와 질의처리 전략 및 기존의 색인 기법들을 살펴본다. 그리고 제 3절에서는 XML 데이터베이스의 중첩요소에 대한 색인기법으로 다차원 색인구조를 이용하는 두 가지의 다차원 타입상속 색인구조를 소개하고, 각 색인구조의 운용기법을 제시한다. 그리고 제 4절에서는 다양한 색인 할당방법과 그에 따른 질의처리 전략을 제시하고 이들에 대한 비교분석 결과를 제시한다. 마지막으로, 제 5절에서는 결론을 내린다.

2. 관련 연구

본 절에서는 XML 데이터베이스의 타입상속 색인기법을 논하는데 필요한 기본 개념들을 기술한다. XML은 현재 많은 기관과 산업체에서 정보의 관리와 교환을 위하여 사용하고 있다. 또한 여러 응용분야에서의 활용을 목적으로 폭넓은 연구를 하고 있으며, XML 스키마(Schema)[5], XQuery[8], XPath (XML Path Language)[6] 등과 같은 XML 관련기술에 대한 표준이 제안되어 있다. 본 절에서는 먼저 XML 스키마에 대하여 소개한 다음, XPath와 함께XML 질의어의 특징, 그리고 XML 질의처리 전략 및 기존의 XML 색인기법들에 관하여 기술한다.

2.1 XML 스키마

XML 스키마는 XML 문서의 구조를 정의하기 위하여 제안된 XML 문서 정의어이다[5]. 그림 1은 Persons 타입에 대한 XML 스키마 그래프의 예이다. 그림에서 타입은 네모로, 요소는 동그라미로 나타내며, 타입 간의 상속관계는 점선 화살표로 나타낸다. 그리고 요소와 타입 간의 중첩관계를 실선 화살표로 나타내며, 해당 요소와 타입은 일반 실선으로 나타낸다. 그림 1에서 Persons 타입은 서브 타입인 Employees 타입과 Students 타입, 그리고 Employees 타입과 Students 타입의 서브 타입들을 포함하는 XML 타입상속 계층구조와 복합요소인

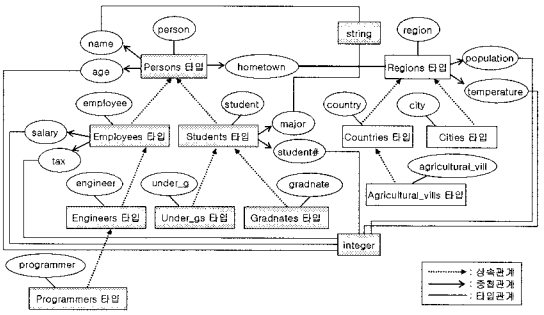


그림 1. *Persons* 타입에 대한 XML 스키마 그래프.

hometown의 도메인 타입인 Regions 타입을 포함하는 XML 타입 집단화 계층구조의 루트이다.

타입상속 계층에서 임의의 타입 T와 그의 모든 서브 타입들을 원소로 하는 집합을 T^*로 표기한다. 예를 들어 그림 1에서 $Persons^*$ 타입은 집합 {Persons 타입, Employees 타입, Students 타입, Engineers 타입, Under_gs 타입, Graduates 타입, Programmers 타입}이고, $Students^*$ 타입은 {Students 타입, Under_gs 타입, Graduates 타입}이다.

2.2 XML 질의어

XQuery는 XML 데이터베이스 질의어로 FOR, LET, WHERE, 그리고 RETURN절로 구성이 된다[8]. FOR 절은 관계형 데이터베이스 질의어인 SQL의 FROM 절과 의미상으로 유사하며, LET 절은 표현을 간략하게 하기 위해서 복잡한 식을 변수 이름에 배치할 수 있도록 한 것이다. WHERE 절은 SQL에서의 WHERE 절과 유사하며 단순 요소에 대한 조건인 단순술어와 함께 중첩요소에 대한 조건인 중첩술어를 사용할 수 있다. 그림 2는 그림 1의 XML 스키마에서 "인구가 100,000명 이상인 지역이 고향인 사람들의 이름을 검색하라"는 질의를 XQuery로 작성한 예문이다.

XPath는 중첩요소의 경로를 표현하기 위한 하나의 경로 표현식(path expression)이다[6]. 본 논문에서는 경로 표현식에서 경로를 이루는 요소들의 타입

```
FOR $h IN Persons*
WHERE $h/hometown[population] >= 100,000]
RETURN <name> $h/name </name>
```

그림 2. XQuery 예문.

을 타입상속 계층상의 일부 타입들로 한정(limit)하여 표현할 수 있도록 XPath를 확장하여 이를 *확장된* XPath라 한다. 확장된 XPath는 각 요소 다음에 한정된 타입 이름들이 ( )속에 올 수 있도록 확장한 것으로 다음과 같은 형태를 가진다. 단, E_i 뒤의 중괄호{ }는 선택적임을 나타내는 표시이다.

$$XP = T_1/E_1\{(T_2)\}/E_2\{(T_3)\}/ \ldots /E_n\{(T_{n+1})\} \quad (1)$$

경로 *XP*에서 타입 T_1을 *타겟타입*, T_{i+1}을 요소 E_i의 *도메인타입*이라 한다. 타겟타입과 도메인타입은 경로에서 타입상속 계층구조에 속하는 특정 타입으로 한정될 수 있으며, 이를 타입 대치(type substitution)라 한다. 이러한 타입 대치는 질의의 범위를 특정 타입으로 한정할 수 있도록 하여 타입상속의 개념을 XML 질의에 표현하도록 한 것이다. 다음 중첩술어들은 그림 2의 질의로부터 확장된 XPath로 표현된 타입 대치에 대한 예를 보여주고 있다.

Np1: Persons*/hometown[population >= 100,000]
Np2: Persons/hometown[population >= 100,000]
Np3: Persons*/hometown(Countries*)
[population >= 100,000]
Np4: Persons/hometown(Countries)
[population >= 100,000]

중첩술어 *Np1*은 질의 대상을 *Persons* 타입의 타입상속 계층에 속하는 모든 타입에 속하는 요소들인 조건식이며, *Np2*는 질의 대상을 Persons 타입에만 속하는 요소들로 한정하는 조건식이다. *Np3*은 *Np1*에서 복합요소 hometown의 타입을 Countries* 즉, Countries 타입, Agricultural-vills 타입으로 한정하는 조건식이며, *Np4*는 *Np2*에서 복합요소 hometown의 타입을 Countries 타입만으로 한정하는 조건식이다.

확장된 XPath식 *XP*에서 *경로 인스턴스(path instance)*는 다음 조건을 만족하는 요소들의 리스트 (E_1, E_2, ..., E_{n+1})로 정의한다. (1) 요소 E_1은 타입 T_1의 요소이다. (2) 요소 E_i ($1 < i <= n+1$)는 타입 T_i의 요소로서 요소 E_{i-1}의 구성 요소이다.

2.3 질의처리 전략 및 기존 색인기법

XML 데이터베이스의 중첩술어를 가지는 질의를 처리하기 위한 스키마 그래프의 운행 전략으로 순방향 운행(FT: Forward Traversal) 전략[19]과 역방향 운행(BT: Backward Traversal) 전략[19]이 있다. FT 전략은 스키마 그래프의 집단화 계층구조에서 상위 타입을 먼저 탐색하고 하위 타입을 나중에 탐색하는 방법으로 깊이-우선 순서로 집단화 계층구조를 운행한다. 그리고, BT 전략은 하위 타입을 먼저 탐색하고 상위 타입을 나중에 탐색하는 방법으로 집단화 계층구조를 아래에서 위로 운행한다. 역방향 운행시에는 하위 타입의 요소를 참조하는 상위 타입의 요소를 탐색하기 위해서는 많은 비용을 필요로 한다.

그리고, 방문한 타입의 요소들을 검색하는 검색 전략으로 중첩 루프(NL: Nested Loop) 전략[20]과 정렬 도메인(SD: Sort Domain) 전략[20]이 있다. NL 전략은 타입을 구성하는 각 요소를 각각 따로 탐색하는 방법이다. 즉, 한 타입에 단순술어가 주어지면, 그 타입의 각 요소별로 술어를 만족하는지 검사하고, 만족하게 되면 그 요소는 하위 타입(순방향 운행시) 또는 상위 타입(역방향 운행시)로 보내진다. 그리고, SD 전략은 타입을 구성하는 모든 요소들을 한꺼번에 탐색하는 방법이다. 즉, 한 타입에 단순술어가 주어지면, 먼저 그 타입의 모든 요소들에 대해서 한꺼번에 술어를 만족하는지 검사하고, 만족하는 모든 요소들을 한꺼번에 하위 타입(순방향 운행시) 또는 상위 타입(역방향 운행시)로 보낸다.

그래프 운행 전략과 검색 전략들을 함께 결합함으로써, 순방향 운행 중첩 루프 검색(FTNL: Forward Traversal with Nested Loop), 순방향 운행 정렬 도메인 검색(FTSD: Forward Traversal with Sort Domain), 역방향 운행 중첩 루프 검색(RTNL: Reverse Traversal with Nested Loop), 그리고 역방향 운행 정렬 도메인 검색(RTSD: Reverse Traversal with Sort Domain)의 네 가지 기본적인 질의처리 전략을 얻을 수 있다.

XML 데이터베이스에서 색인을 유지하는 요소가 단순요소이면 관계형 데이터베이스 시스템에서와 같이 색인된 요소에 대한 제한 술어를 만족하는 요소의 신속한 탐색을 위해서 사용될 수 있고, 색인을 유지하는 요소가 복합요소이면 역방향 운행시 상위 타입의 요소를 탐색하기 위한 비용을 줄일 수 있다. 따라서, XML 데이터베이스에서 색인의 키 값은 단순요소의 도메인이 되는 기본 타입의 값들뿐만 아니라 복합요소의 도메인이 되는 사용자 정의 타입의 요소

식별자인 EID들도 될 수 있다. 그리고, 중첩요소에 대한 색인기법으로 중첩요소를 표현하는 Xpath에 의한 암시적 조인을 미리 계산하여 색인구조로 유지함으로써, 질의처리시 타입 집단화 계층에 대한 운행을 생략할 수도 있다[11,13].

지금까지 제안된 XML 데이터베이스의 중첩요소에 대한 색인기법으로는 Index Fabric[7], APEX[2], DataGuide[11], 구조 요약(structural summary)[12], 1-Index[13] 등이 있다. DataGuide는 비결정적(non-deterministic) 오토마타를 결정적(deterministic) 오토마타로 변환하는 과정과 동일한 과정으로 경로를 색인하는 기법이다. 일반적으로 비결정적 오토마타를 결정적 오토마타로 바꿀 경우, 크기가 커지게 되지만, XML 문서 내에 동일한 경로들이 많이 존재할수록 색인의 크기는 줄어든다. 1-index는 루트로부터 시작되는 경로의 집합이 동일한 노드들을 모아 색인을 구축하는 기법으로서, DataGuide와 마찬가지로 XML 문서 내에 동일한 경로가 매우 많이 존재한다는 점을 이용하는 색인기법이다. 그러나 이러한 기존의 색인기법들은 일차원 색인구조인 B^+-tree를 이용한다. 따라서, XML 데이터 모델의 타입상속의 특징을 반영하지 못하는 것들로서, 타겟 타입의 대치 또는 도메인타입의 대치가 있는 질의는 지원하지 못한다. 즉, 질의의 대상 범위가 타입상속 계층상의 임의의 타입들로 제한되거나, XPath식에 나타나는 어떠한 요소의 도메인 타입이 타입상속 계층상의 임의의 타입들로 제한이 되는 질의들을 지원할 수 없다.

3. 다차원 타입상속 색인구조

XML 데이터베이스 질의어의 중첩술어에는 타입 대치가 있을 수 있으며, 이러한 중첩술어의 처리를 지원하기 위한 색인구조로 다차원 색인구조[15-17]를 이용할 수 있다. 즉, 색인할 중첩요소의 키 값 도메인과 함께 중첩요소를 표현하는 경로상의 각 타입상속 계층마다 타입 식별자들로 구성된 한 차원씩의 타입 식별자 도메인[21]을 할당함으로써, (경로길이+1)차원의 도메인 공간을 구성하여 이를 다차원 색인구조에 적용한다. 예를 들어, 그림 1과 같은 스키마 그래프에서 Persons 타입의 중첩요소 population (Regions 타입의 단순요소)에 대한 색인구조로서 중첩요소 population의 키 값 도메인, Persons 타입상속 계층의 타입 식별자 도메인, 그리고 Regions 타입상속 계층의 타입 식별자 도메인으로 삼차원 도메인 공간을 구성할 수 있다.

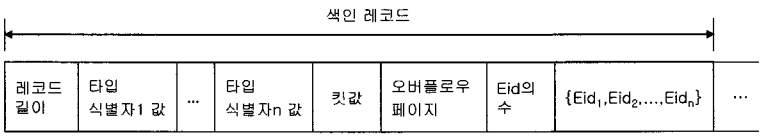
본 논문에서는 XML 데이터베이스의 중첩요소에 대한 색인구조를 다차원 색인구조의 하나인 계층 그리드 파일(MLGF: MultiLevel Grid File)[17]을 이용하여 구성하고, 이를 *다차원 타입상속 색인구조 (MD-TIX: Multidimensional Type Inheritance indeX)[18]*라 한다. MD-TIX는 디렉토리와 색인 페이지로 구성된다. 디렉토리는 다단계의 균형된 트리 구조를 가지며, 디렉토리를 구성하는 디렉토리 페이지의 구조는 MLGF에서와 마찬가지이다. 색인 페이지는 색인 레코드들로 구성되며, 각 색인 레코드에는 경로상의 각 타입 식별자 값(type-id value) 필드, 키 값(key value) 필드, 요소 또는 경로 인스턴스의 개수 필드, 및 이들에 대한 색인 엔트리들의 리스트 필드가 있다. 그리고, 레코드의 크기가 페이지의 크기보다 크게될 때 오버플로우 페이지를 할당하고 이를 포인트하기 위한 오버플로우 페이지(overflow page) 필드가 있다.

MD-TIX는 색인 레코드에 있는 색인 엔트리의 구성방법에 따라 *다차원 타겟요소 색인구조*와 *다차원 경로 색인구조*의 두 가지 색인구조로 분류할 수 있다[18]. 다차원 타겟요소 색인구조는 색인 엔트리를 색인된 중첩요소의 타겟 타입상속 계층에 속하는 요소에 대한 요소 식별자(즉, Eid)들로 구성하며, 다차원 경로 색인구조는 색인 엔트리를 색인된 중첩요소에 대한 경로 인스턴스(즉, Eid 리스트)들로 구성한다. 다차원 경로 색인구조와 같이 색인 엔트리를 경로 인스턴스들로 구성하는 것은 색인 엔트리를 타겟 타입상속 계층의 요소 식별자만으로 구성하는 경우에 발생하게 되는 데이터베이스의 변경에 따른 색인구조의 막대한 유지비용을 줄이기 위함이다. 그림 3은 이러한 다차원 타입상속 색인구조들의 색인 페이지 구조를 각각 나타낸다.

중첩요소에 대한 색인기법은 중첩술어를 만족하는 타겟요소들의 탐색에는 매우 유용하지만, 상대적으로 색인구조의 유지비용을 많이 필요로 한다[2].

3.1 다차원 타겟요소 색인구조

다차원 타겟요소 색인구조는 경로 인스턴스를 유지하는 다차원 경로 색인구조에 비해 저장 공간의



(a) 다차원 타겟요소 색인구조

(b) 다차원 경로 색인구조.

그림 3. 다차원 타입상속 색인구조의 색인 페이지 구조

오버헤드가 적은 반면에, 데이터베이스의 변경에 따른 색인구조의 유지 비용에 대한 오버헤드가 많다. 경로 *XP*에서 i번째 타입상속 계층에 있는 임의의 요소 E_i의 구성요소가 E_{i+1}에서 새로운 E_{i+1}'로 변경될 경우, 다차원 타겟요소 색인구조의 갱신을 위해서는 다음과 같은 단계의 작업이 필요하다.

① 요소 E_{i+1}로부터 중첩요소 E_n까지 경로상의 타입식별자 값들과 함께 키 요소 값으로 구성된 색인키 리스트들의 집합 KS(A)를 구한다. 이를 위해서는 요소 E_{i+1}로부터 경로를 따라 순방향 운행을 하여야 한다. 여기서, 경로상의 임의의 요소 E_i가 다중 값을 갖지 않으면 KS(A)는 하나의 원소만을 가진다.

② 요소 E_{i+1}'로부터 중첩요소 E_n까지 경로상의 타입식별자 값들과 함께 키 요소 값으로 구성된 색인키 리스트들의 집합 KS(B)를 구한다. 여기서, KS(A) = KS(B)이면 색인의 갱신이 필요 없으며, 그렇지 않으면 다음을 시행한다.

③ E_i를 직접 또는 간접적으로 참조하는 타겟 타입상속 계층 T_1의 요소 Eid들의 집합 ES를 구한다. 이를 위해서는 요소 E_i로부터 경로를 따라 역방향 운행을 하여야 한다. 여기서, i = 1이면 ES는 {E_i}가 된다.

④ 색인의 갱신을 다음과 같이 시행한다.

- KS(A) ⊃ KS(B) 이면, {KS(A) - KS(B)}를 R로 하고 R에 속하는 각 색인키 리스트에 해당하는 색인 레코드에서 집합 ES에 있는 Eid들을 제거한다.
- KS(A) ⊂ KS(B) 이면, {KS(B) - KS(A)}를 R로 하고 R에 속하는 각 색인키 리스트에 해당하는 색인 레코드에서 집합 ES에 있는 Eid들을 첨가한다.
- 그렇지 않으면, {KS(A) - KS(B)}를 R1, {KS(B) - KS(A)}를 R2라 하고, R1에 속하는 각 색인키 리스트에 해당하는 색인 레코드에서 집합 ES에 있는 Eid들을 제거하고, R2 에 속하는 각 색인키 리스트에 해당하는 색인 레코드에서 집합 ES에 있는 Eid들을 첨가한다.

이와 같이 다차원 타겟요소 색인구조에서는 데이터베이스의 변경에 따른 색인구조의 갱신을 위해서 경로상의 역방향 운행이 필요하다. 그리고 요소의 삽입과 제거에 따른 색인구조의 갱신을 위해서는 한 번의 순방향 운행만이 필요하며, 이는 변경의 경우와 같다.

3.2 다차원 경로 색인구조

다차원 경로 색인구조는 색인 레코드에 경로 인스턴스를 저장하기 때문에 다차원 타겟요소 색인구조에 비해 많은 양의 저장 공간을 필요로 하는 반면에, 데이터베이스의 변경에 따른 색인구조의 유지 비용에 대한 오버헤드가 다차원 타겟요소 색인구조에 비해 적게 된다. 경로 *XP*에서 *i*번째 타입상속 계층에 있는 임의의 요소 E_i의 구성요소가 E_{i+1}에서 새로운 E_{i+1}'로 변경될 경우, 다차원 경로 색인구조의 갱신을 위해서는 다음과 같은 단계의 작업이 필요하다.

① 요소 E_{i+1}로부터 중첩요소 E_n까지 경로상의 타입식별자 값들과 함께 키 요소 값으로 구성된 색인키 리스트들의 집합 KS(A)를 구한다. 이를 위해서는 요소 E_{i+1}로부터 경로를 따라 순방향 운행을 하여야 한다. 이는 다차원 타겟요소 색이구조에서와 마찬가지

이다.

② 요소 E_{i+1}'로부터 경로에 따른 순방향 운행을 통하여 중첩요소 E_n까지 서브경로 인스턴스들의 집합 SP.new와 경로상의 타입식별자 값들과 키 요소 값으로 구성된 색인키 리스트들의 집합 KS(B)를 구한다.

③ KS(A)에 있는 각 색인키 리스트에 해당하는 색인 레코드를 액세스하여 i번째 항목이 E_i이고 $i+1$번째 항목이 E_{i+1}인 경로 인스턴스를 삭제함과 동시에, 각 경로 인스턴스의 첫 번째 항목에서 i번째 항목까지의 부분만을 T에 보관한다.

④ i번째 항목이 E_i이고 $i+1$번째 항목이 E_{i+1}'인 새로운 경로 인스턴스 집합 PI를 생성한다. 이는 T에 있는 요소들과 SP.new에 있는 요소들을 각각 연결함으로써 얻을 수 있다.

⑤ KS(B)에 있는 각 색인키 리스트에 해당하는 색인 레코드에 집합 PI에 있는 경로 인스턴스를 첨가한다.

이와 같이 다차원 경로 색인구조에서는 경로 인스턴스가 색인 레코드에 저장되어 있기 때문에 데이터베이스의 변경에 따른 색인구조의 갱신을 위한 역방향 운행이 필요 없다. 따라서 다차원 경로 색인구조는 데이터베이스의 요소 내에 역 참조자가 존재하지 않아도 사용할 수 있다.

3.3 두 색인구조의 운용에 따른 유지비용

다차원 경로 색인구조에서는 저장 공간의 오버헤드 때문에 검색 질의처리에 대해서는 다차원 타겟요소 색인구조보다 좋은 성능을 보이지 못한다. 그러나 두 색인구조의 운용에 따른 유지비용에 대한 오버헤드는 색인된 중첩요소의 경로 길이에 따라 많은 차이를 보이게 된다. 따라서 본 절에서는 두 색인구조의 운용에 따른 유지비용을 비교 분석한다.

색인구조를 구축할 경로의 길이가 2인 경우, 역 참조자가 데이터베이스의 요소 내에서 제공될 경우에는 다차원 타겟요소 색인구조에서도 역방향 운행이 필요 없기 때문에 다차원 경로 색인구조의 유지비용은 같게 된다. 그리고 데이터베이스의 변경이 첫 번째 타입상속 계층 T_1에서 발생할 경우, 다차원 타겟요소 색인구조와 다차원 경로 색인구조의 갱신을 위해 필요한 순방향 운행에 대한 오버헤드는 색인구조와 무관하게 데이터베이스 자체의 변경에서도 필요하다. 즉, 경로상의 첫 번째 타입상속 계층에 있는 요소 E_1에서 구성요소 E_2의 값이 두 번째 타입상속 계층의 다른 요소로 변경될 경우, 다차원 타겟요소 색인구조와 다차원 경로 색인구조에서는 색인키 값을 얻기 위하여 E_2의 변경전의 값과 변경후의 값에 대한 요소 E'와 E''를 액세스하는 두 번의 순방향 운행이 필요하다. 그러나, 이 두 요소의 액세스는 E_1에 대한 역 참조자의 값을 변경하기 위하여 색인구조와 무관하게 데이터베이스 자체적으로 반드시 액세스해야 한다.

색인구조를 구축할 경로의 길이가 2보다 크게 되면, 다차원 타겟요소 색인구조에서는 역 방향 운행이 필요하게 되므로 다차원 경로 색인구조에 비해 유지비용이 증가하게 된다. 다차원 타겟요소 색인구조의 유지비용을 지배하는 것은 역방향 운행에 의한 것으로, 역방향 운행에 필요한 요소의 액세스 개수는 요소 참조 공유도에 의해 결정된다. 다차원 경로 색인구조에서는 역방향 운행이 필요 없기 때문에 다차원 타겟요소 색인구조에서보다 유지비용이 매우 적게 된다. 한편, 데이터베이스의 변경이 첫 번째 타입상속 계층 T_1과 두 번째 타입상속 계층 T_2에서 발생할 경우에는, 다차원 타겟요소 색인구조에서도 역방향 운행이 필요 없게 되므로 색인구조의 유지비용이 다차원 경로 색인구조에서와 같게 된다.

따라서, 이와 같은 유지 비용의 분석으로부터 다음과 같은 결과를 얻을 수 있다. 먼저, 색인을 구축할 경로의 길이가 2인 경우에는 다차원 타겟요소 색인구조가 적합하다. 이것은 경로의 길이가 2인 경우의 유지비용은 두 가지 색인구조 모두 비슷한 반면에 다차원 타겟요소 색인구조에서 검색 성능이 더 좋기 때문이다. 그리고 색인을 구축할 경로의 길이가 3이상인 경우에는 일반적으로 다차원 경로 색인구조가 적합하다. 이것은 다차원 경로 색인구조의 검색 성능은 다차원 타겟요소 색인구조에 필적하는 반면에 데이터베이스 변경에 의한 유지비용이 적게 되기 때문이다.

4. 다차원 타입상속 색인구조의 할당기법

본 절에서는 제 3절의 다차원 타입상속 색인구조의 운용에 따른 유지비용의 분석 결과로 데이터베이스의 각 요소가 역 참조자를 가지고 있지 않거나 경

로의 길이가 3이상인 경우에 유지비용 면에서 유리한 다차원 경로 색인구조를 할당하여 질의를 처리하는 방법을 분석한다. 먼저, 두 개의 중첩술어가 주어지는 질의의 경우를 고려하고, 이를 여러 개의 중첩술어로 구성된 일반적인 복합질의 경우로 확장한다.

경로 $XP = T_1E_1E_2 \ldots E_n$이 주어지면, XP상에서 대응되는 타입상속 계층들은 $\{T_1, T_2, \ldots, T_n\}$이다. 다음 표 1은 본 절의 색인할당에 따른 질의처리 분석을 위한 비용 모델에서 필요한 매개변수들이다.

타입상속 계층 T_i에서 요소 E_i의 구성요소 값으로 동일한 값을 가지는 요소의 평균 개수를 나타내는 매개변수 $S_i(1 \leq i \leq n)$는 구성요소의 공유정도를 나타내며, 이것은 질의처리 비용에 가장 크게 영향을 미친다. 앞으로 연속된 구성요소 공유도의 곱으로 다음과 같은 표기법을 사용한다.

$$S(i, j) = S_i \times \cdots \times S_j \quad (2)$$

그리고, 비용 모델을 간단히 하기 위해서 다음과 같은 몇 가지 가정을 한다.

-타입 T_i의 각 요소는 타입 T_{i-1}의 요소들에 의해 반드시 중첩된다. 이것은 매개변수의 관점에서는 $U_i = N_{i-1}$를 의미한다.

-모든 키 값들은 동일한 길이를 가진다. 이것은 모든 비단말 노드 색인 레코드들이 길이가 동일함을 의미한다.

표 1. 비용 모델에서 사용한 매개변수.

매개변수	의미
h	색인구조 디렉토리 트리의 높이
P	색인 페이지의 크기(4K 바이트)
U_i	타입상속 계층 T_i에서 요소 E_i의 구성요소가 가지는 유일한 값의 개수 $(1 \leq i \leq n)$
N_i	타입상속 계층 T_i의 요소의 개수$(1 \leq i \leq n)$
S_i	타입상속 계층 T_i에서 요소 E_i의 구성요소 값으로 동일한 값을 가지는 요소의 평균 개수$(S_i = \lceil N_i/U_i \rceil)$
XLL	다차원 경로 색인구조의 평균 색인 레코드 길이
IDL	요소 식별자의 길이(8 바이트)
$PN(T_i)$	타입상속 계층 T_i 요소들의 디스크 페이지 개수
IAC	다차원 경로 색인구조의 액세스 비용$(IAC = h + \lceil XLL/P \rceil)$

-구성요소의 값들은 요소를 정의하는 타입의 요소들 중에서 균일하게 분포된다.

-모든 색인구조는 클러스터링 색인구조가 아니다. 이것은 요소들이 단말노드 색인 레코드들에 저장되어 있는 EID들의 순서에 따라서 저장되지 않음을 의미한다.

본 논문에서의 비용 함수는 요소 공유도를 나타내는 매개변수 S_i와 타입의 요소의 개수 N_i에 중점을 둔다. 이들은 다차원 경로 색인구조의 액세스 비용에 가장 큰 영향을 미친다.

4.1 질의처리의 비용식

색인이 제공되지 않을 때에는 술어가 요소들 사이의 참조에 의해서 명시적으로 평가되어야 한다. 본 절에서는 제 2.3절에서 소개한 순방향 운행법(FT)과 중첩 루프(NL) 검색법에 의한 질의처리 전략인 FTNL 전략에 대한 질의처리 비용식[1]을 나타낸다. 이는 앞으로 여러 색인할당에 대한 비교 평가를 위해 필요한 비용식이다. 비용식에서 다음과 같은 표기법을 사용한다.

• $C_s(T, N, E, S)$: 타입 T의 요소 수 N의 집합에서 중첩요소 E를 위한 값으로 집합 S에 속하는 EID들을 가지는 요소들을 결정하는 비용을 나타낸다. 예를 들어, 그림 1의 스키마에서 집합 $S = \{region[i], region[m]\}$이 주어질 때, $C_s(Persons, N_{Persons}, hometown, S)$는 $homtown$으로 $region[i]$ 또는 $region[m]$을 가지는 타입 $Persons$의 요소를 결정하는 비용을 나타낸다.

• $C_p(T, N, E, pred)$: 타입 T의 요소 수 N의 집합에서 술어 $pred$를 만족하는 중첩요소 E를 가지는 요소들을 결정하는 비용을 나타낸다. 예를 들어, 술어 "$population = 100,000$"를 고려할 때, $C_p(Persons, N_{Persons}, population, population = 100,000)$는 $population$이 $100,000$인 $region$이 $hometown$인 타입 $Persons$의 요소를 결정하는 비용을 나타낸다.

FTNL 전략은 요소들을 하나하나 방문하는 전략

1) 본 논문에서는 질의처리 전략으로 순방향 운행법(FT)과 중첩 루프(NL) 검색법에 FTNL 전략만을 사용한다. 역방향 운행(RT)은 대부분의 경우에서 매우 많은 비용을 필요로 하고, 중첩 루프(NL) 전략이 정렬 도메인(SD) 전략보다 각 색인할당의 비교를 쉽게 하기 때문이다.

이다. 요소 탐색에는 타겟타입 전체로부터 시작하는 탐색(a)와 중간 질의처리의 결과에 따른 탐색(b)로 두 가지의 형태가 있다. 타입 T_i의 요소 수가 N_i일 때 탐색(a)의 비용식은 다음과 같다.

$$C\_FTNL = PN(T_i) + 2 \times N_i \times (n-i) \quad (3)$$

식 (3)은 다음과 같이 유도된다. T_i의 요소들을 검색하기 위해서, 모든 페이지들이 액세스되어야 하고, 이 페이지들의 수는 $PN(T_i)$이다. 그리고, 각 요소들에 대해서 요소 E_n의 값에 도달해야 한다. 각 요소 E_j(i ≤ j ≤ n)에 대해서, 각 요소의 위치가 저장되어 있는 시스템 테이블의 액세스와 요소 자체의 액세스가 필요하다.

탐색(b)는 중간 질의처리 결과의 요소 수를 N이라 할 때, 타겟 타입 요소들의 EID는 알려져 있으므로 비용식은 다음과 같다.

$$C\_FTNL = 2 \times N \times (n-i+1) \quad (4)$$

식 (4)는 탐색 (a)의 비용식과 동일하게 유도된다. 단지, 첫 번째 타입의 액세스가 필요 없기 때문에 $PN(T_i)$가 생략된 것이다. 이 식들을 종합하면 다음과 같다:

$$C_s(T_i, N, E_n, S) = C_p(T_i, N, E_n, pred) = PN(T_i) + 2 \times N_i \times (n-i) \text{ if } N = N_i,$$
$$2 \times N \times (n-i+1) \quad \text{if } N < N_i \quad (5)$$

여기서 C_s와 C_p는 동일하다. 왜냐하면 이 운행법에서는 탐색의 종류에 관계없이 모든 인스턴스들이 액세스되어야 하기 때문이다.

4.2 두 경로 사이의 색인할당

본 절에서는 동일한 타입에서 시작하여 경로의 일부가 겹치는 두 개의 경로상에서, 각 경로의 마지막 요소에 조건이 주어지는 술어를 가지는 질의에 대해서 각 색인할당에 대한 질의처리 전략을 분석한다.

두 개의 경로 $XP_1 = T_1.E_1.E_2. \ldots .E_n$과 $XP_2 = T_1.E_1'.E_2'. \ldots .E_m'$에서, $1 \le i \le j$ 에 대해서는 $E_i = E_i'$이고, $j < i$ 에 대해서는 $E_i \neq E_i'$ 인 j(< min(n, m))가 존재한다면, 이 두 경로는 겹쳐진 경우이다. 여기서, 먼저 분리된 경로구성에 대해서 소개한다. 분리된 경로구성은 하나의 경로를 두 개의 부경로로 분리하여서, 각 부경로에 색인구조를 할당한 경우이다. 그림 4는 집단화 계층에 따른 경로구성을 본 논문에서 앞으로 사용할 경로들의 이름과 함께 나타낸 것이다. 그림 4의 분리된 경로에서 분리가 시작되는 타입상속 계층은 T_i이다.

그림 4와 같은 경로 상에 주어진 두 개의 술어로 구성된 질의 형태에 따른 다차원 경로 색인구조의 할당방법은 표 2와 같이 분류할 수 있다. 표 2에서는 먼저 하나의 경로에만 색인을 할당하는 경우와 두 경로 모두에 색인을 할당하는 경우로 분류하고, 각각에 대해서 경로를 분리하지 않거나 분리하여 각 경로

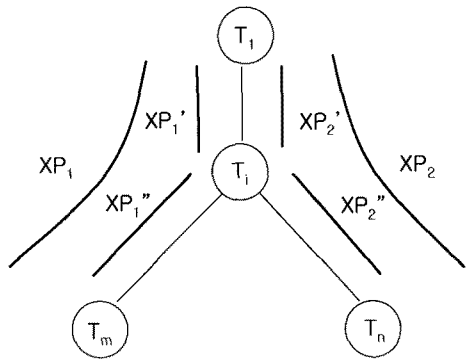


그림 4. 두 개로 분리된 경로 스키마

표 2. 두 경로상에 가능한 색인할당의 분류

색인할당 분류		색인할당 방법
하나의 경로 XP_1에만 색인할당	색인할당①	XP_1 전체에 하나의 색인할당
	색인할당②	XP_1을 겹침부분과 비겹침 부분으로 분리하여 각각 색인할당
두 개의 경로 XP_1과 XP_2에 모두 색인할당	색인할당③	XP_1 전체, XP_2 전체에 각각 색인할당
	색인할당④	XP_1, XP_2의 겹침부분, XP_1의 비겹침부분, XP_2의 비겹침부분에 각각 색인할당
	색인할당⑤	XP_1에는 전체 색인할당, XP_2에는 겹침부분과 비겹침부분으로 분리하여 각각 색인할당
	색인할당⑥	XP_1에는 전체 색인할당, XP_2에는 비겹침부분에만 색인할당

에 색인을 할당하는 방법으로 나누어 각 색인의 할당 방법에 대하여 번호를 매기고, 앞으로 이 번호로서 각 할당방법을 대신해서 사용한다. 다음은 색인할당의 효율성을 분석하기 위하여 각 색인할당 방법에 대한 질의처리 전략을 제시하고 이들의 질의처리 비용을 평가한다.

(1) XP_1 전체에 하나의 색인이 있는 색인할당 ①의 경우

표 2의 색인할당 ①에서처럼 겹침 경로에 대해서 경로를 분리하지 않고 경로 XP_1에만 하나의 다차원 경로 색인구조를 할당할 경우에는 다음과 같은 네 가지 질의처리 전략이 있을 수 있다. 이 전략들 사이의 차이점은 단지 T_{i+1}에서 T_n까지 방문하는 방법에 있다. 이러한 전략들에서는 다차원 경로 색인구조의 색인 엔트리인 경로 인스턴스에서 요소 식별자들의 튜플을 생성하는 프로젝션 연산이 사용된다. 이 네 가지 전략의 세부사항은 다음과 같다.

(가) 술어 $pred_m$은 XP_1상에 정의된 다차원 경로 색인구조에 한번 액세스하는 것으로 해결된다. 먼저, 탐색된 경로 인스턴스들에 대하여 프로젝션 연산을 수행하여 (EID_1, EID_i)쌍들을 생성한다. 그리고, 각 쌍들의 두 번째 원소들로서 정렬하여 술어 $pred_n$을 해결하기 위하여 T_n까지 FTNL 전략을 사용한다.

(나) (가)의 전략에서 T_i에서 T_n까지의 요소들의 탐색을 위하여 FTSD 전략을 사용한다.

(다) (가)의 전략에서 T_i에서 T_n까지의 요소들의 탐색을 위하여 RTNL 전략을 사용한다.

(라) (가)의 전략에서 T_i에서 T_n까지의 요소들의 탐색을 위하여 RTSD 전략을 사용한다.

본 논문에서는 단지 순방향 운행(FT)을 사용한 전략을 위한 비용식만 제시한다. 역방향 운행(RT)은 대부분의 경우에서 매우 많은 비용을 필요로 하기 때문이다. 또한 각 색인할당의 비교를 쉽게 하기 위하여 중첩 루프(NL) 전략만을 고려한다. 따라서, 전략 (가)의 비용식을 비용식 (5)를 이용하여 나타내면 다음과 같다.

$$\begin{aligned} 비용(가) &= IAC(XP_1) + C_p(T_i, S(i, m), E_n', E_{n+1}' \\ &\quad = value_n) \\ &= IAC(XP_1) + 2 \times S(i, m) \times (n-i+1) \end{aligned} \quad (6)$$

식 (6)은 다음과 같이 설명할 수 있다. IAC(XP_1)은 경로 XP_1에 할당된 다차원 경로 색인구조의 액세스에 대한 비용이다. 이것은 경로 XP_1의 인스턴스들을 반환한다. 이 인스턴스들에 대한 프로젝션 연산은 $S(i, m)$개의 쌍들을 반환한다. 이 쌍들의 두 번째 원소들에 대해서 그 요소의 값을 얻기 위하여 두 번의 페이지 액세스가 필요하고, 이것은 경로 XP_2''의 길이인 (n-i+1)만큼 반복되어야 한다.

여기서, 다차원 경로 색인구조의 사용과 다차원 타겟요소 색인구조의 사용을 비교하면, 다차원 경로 색인구조를 사용하는 경우가 대부분의 경우에 유리하다. 이는 다차원 타겟요소 색인구조의 경우에는 술어 $pred_n$을 평가하기 위하여 T_i가 아닌 T_1으로부터 T_n까지 운행을 해야하기 때문이다. 그림 5는 색인할당 ①의 경로구성에서 다차원 경로 색인구조와 다차원 타겟요소 색인구조의 성능을 나타내며, 요소 공유도 S가 너무 크지 않을 때는 경로 XP_1에 다차원 경로 색인구조를 할당하는 것이 효율적임을 나타낸다.

(2) XP_1'과 XP_1''에 색인이 있는 색인할당 ②의 경우

그림 4의 겹침 경로에 대해서, 표 2의 색인할당 ②는 XP_1 경로를 분리하여 부경로 XP_1'과 XP_1''상에 다차원 경로 색인구조를 할당하는 경우이다. 이러한 경로구성에서 있을 수 있는 질의처리 전략들은 다음과 같다.

(가) 경로 XP_1''에서 색인을 액세스하고, $pred_n$을 평가하기 위해서 T_i에서부터 T_n까지 FTNL 전략을 사용한다. 그리고 검색된 각 요소에 대해서 경로 XP_1'에서 색인을 액세스한다.

(나) 경로 XP_1''에서 색인을 사용하지 않은 전략으로, 두 술어를 평가하기 위하여 T_i에서 T_m과 T_n까지 각각 FTNL 전략을 사용한다. 그리고 검색된 각 요소에 대해서 경로 XP_1'에서 색인을 액세스한다.

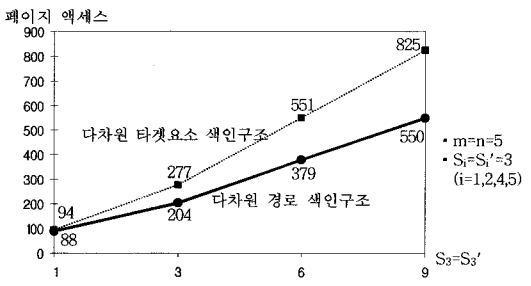


그림 5. 색인할당 ①에서 다차원 경로 색인구조와 다차원 타겟요소 색인구조의 성능 비교.

(다) 경로 XP_1'에서 색인을 사용하지 않은 전략으로, 경로 XP_1''에서 색인을 액세스하고, $pred_n$을 평가하기 위해서 T_i에서부터 T_n까지 FTNL 전략을 사용한다. 그리고 T_1에서부터 T_i까지 FTNL 전략을 사용한다.

이들 각 전략의 비용식은 다음과 같다.

$$\begin{aligned} 비용(가) &= IAC(XP_1'') + C_p(T_i, S(i, m), E_n', E_n' \\ &= value_n) + no \times IAC(XP_1') \\ &= IAC(XP_1'') + 2 \times S(i, m) \times (n-i+1) \\ &\quad + \lceil S(i, m)/U_n' \rceil \times IAC(XP_1') \end{aligned} \quad (7)$$

여기서, $no = \lceil S(i, m)/U_n' \rceil$로서 두 술어를 만족하는 타입 T_i 요소들의 추정치를 나타낸다.

$$\begin{aligned} 비용(나) &= C_p(T_i, N_i, E_m, E_m = value_m) + C_p(T_i, \\ &\quad S(i, m), E_n', E_n' = value_n) + no \times \\ &\quad IAC(XP_1') \\ &= PN(T_i) + 2 \times N_i \times (m-i) + 2 \times S(i, \\ &\quad m) \times (n-i+1) + \lceil S(i, m)/U_n' \rceil \times \\ &\quad IAC(XP_1') \end{aligned} \quad (8)$$

$$\begin{aligned} 비용(다) &= IAC(XP_1'') + C_p(T_i, S(i, m), E_n, E_n \\ &= value_n) + C_s(T_1, N_i, E_i, no) \\ &= IAC(XP_1'') + 2 \times S(i, m) \times (m-i+1) \\ &\quad + PN(T_1) + 2 \times N_1 \times (i-1) \end{aligned} \quad (9)$$

다음은 색인할당 ②에 대한 각 질의처리 전략들의 비교 분석이다. $IAC(XP_1'') > PN(T_i) + 2 \times N_i \times (m-i)$, 즉 $h_{XP_1''} + \lceil \frac{k(i,m) \times (m-i)}{500} \rceil > PN(T_i) + 2 \times N_i \times (m-i)$ 이면, (비용(가) > 비용(나))이다. 그리고, $\lceil \frac{k(i,m)}{U_n'} \rceil \times (h_{XP_1'} + \lceil \frac{k(i,m) \times (m-i)}{500} \rceil) > PN(T_i) + 2 \times N_1 \times (i-1)$ 이면, (비용(A)> 비용(다))이다. 따라서 이와 같은 비교에서 다음과 같은 요약 1을 도출할 수 있다.

요약 1: 겹침이 있는 두 경로에서 한 경로에만 두 개의 부경로로 분리하여 색인을 할당할 경우, 각 부경로의 요소 공유도가 높을 때는 그 부경로에 색인을 사용하지 않는 질의처리 전략이 더 효율적이다.

다음은 겹침이 있는 두 경로에서 한 경로에만 색인을 할당할 때, 그 경로를 분리하는 경우(색인할당 ②)와 분리하지 않는 경우(색인할당 ①)에 대한 비교이다. 먼저, 그 결과를 그림으로 나타내면 그림 6과 같다. 이는 비용식 (7)을 색인할당 ①의 비용식 (6)과 비교함으로써 알 수 있다. 그림 6에서는 요소 E_n'에 대한 서로 다른 값들의 수인 U_n'이 매우 작을 때는

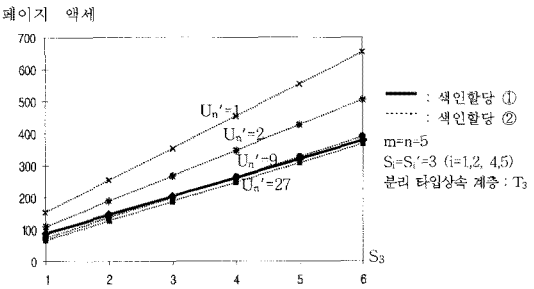


그림 6. 색인할당 ①과 ②의 성능 비교.

분리하지 않는 것이 좋음을 나타낸다.

(3) XP_1과 XP_2에 각각 색인이 있는 색인할당 ③의 경우

그림 4의 겹침 경로에서 두 경로 XP_1과 XP_2 모두 분리하지 않고 색인을 할당하는 경우이다. 이 경우의 질의처리 전략은 다음과 같다.

(가) 두 개의 색인을 모두 이용하는 전략으로 두 색인을 각각 액세스하여 각 결과의 공통된 부분을 도출한다.

(나) 두 개의 색인 중에 하나만 사용하는 전략이다.

전략 (가)의 비용식은 다음과 같이 나타낼 수 있다.

$$비용(가) = IAC(XP_1) + IAC(XP_2) \quad (10)$$

두 개의 경로 모두에 색인이 있는 경우와 하나의 경로에만 색인이 있는 경우에 대한 질의처리 전략의 비교는 다음과 같다. 이는 비용식 (10)과 색인할당 ①의 비용식 (6)을 비교함으로써 알 수 있다. 두 개의 색인을 사용하는 것이 하나의 색인을 사용하는 것보다 유용하게 되는 경우는 다음과 같은 조건이 성립할 때이다. 비용식 (10)의 값이 비용식 (6)의 값보다 작을 때이므로, $IAC(XP_2) < 2 \times S(i, m) \times (n-i+1)$일 때, 즉 $h_{XP_2} + \lceil \frac{k'(1, n) \times n}{500} \rceil < 2 \times S(i, m) \times (n-i+1)$인 경우이다. 이 경우에서는 경로 인스턴스를 색인 값으로 가지는 다차원 경로 색인구조의 이점을 이용하지 않기 때문에, 두 경로 상에서 많은 갱신이 없으면 다차원 경로 색인구조의 사용이 다차원 타겟요소 색인구조의 사용보다 시간과 공간을 낭비하게 된다.

(4) XP_1', XP_1''과 XP_2''에 색인이 있는 색인할당 ④의 경우

색인할당 ④는 겹침이 있는 두 경로에서 모두 분리된 경로구성을 선택하고, 두 경로 모두에 색인을

할당하는 경우이다. 여기에서 두 경로의 첫 번째 부경로의 색인들은 동일하다. 이 경우의 질의처리 전략은 다음과 같다.

(가) 모든 색인을 이용하는 전략으로, 술어 $pred_m$과 $pred_n$을 만족하는 타입 T_i의 요소들을 구하기 위하여 먼저 XP_1''과 XP_2''상의 색인을 각각 액세스하여 검색하고, 그들의 공통된 결과를 도출한다. 그리고 도출된 각 요소에 대하여 경로 XP_1'상의 색인을 액세스하여 최종 결과를 얻는다.

(나) XP_1'상의 색인을 이용하지 않는 전략으로, XP_1'상의 색인을 이용하지 않고 대신에 T_1에서 T_i까지 FTNL 전략을 사용한다.

(다) 경로 XP_1''이나 XP_2''상의 색인을 모두 또는 하나를 이용하지 않는 전략으로, 이 전략은 색인할당 ②에서 제시한 전략들과 같아진다.

각 전략들의 비용식은 다음과 같다.

$$비용(가) = IAC(XP_1'') + IAC(XP_2'') + no \times IAC(XP_1') \quad (11)$$

여기서, $no = \lceil S(i, m)/U_n' \rceil$ ($= \lceil S'(i, n)/U_m \rceil$)로서 두개의 술어 $pred_m$과 $pred_n$을 동시에 만족하는 요소의 수를 나타낸다.

$$\begin{aligned} 비용(나) &= IAC(XP_1'') + IAC(XP_2'') + C_s(T_1, N_1, E_i, no) \\ &= IAC(XP_1'') + IAC(XP_2'') + PN(T_1) + 2 \times N_1 \times (i-1) \end{aligned} \quad (12)$$

이 전략들 사이의 비교는 다음과 같다. 먼저, 전략 (가)와 전략 (나)에 대해서 비용식을 비교하면, $S(1, m) \times (i-1) > 500 \times U_n' \times (PN(T_1) + 2 \times N_1 \times (i-1))$ 이면, $비용(가) > 비용(나)$이다. 이것은 경로 XP_1상의 요소 공유도가 높을 때는 경로 XP_1'상의 색인을 사용하지 않고 FTNL 전략을 사용하는 것이 좋음을 나타낸다.

일반적으로 색인할당 ④가 색인할당 ②에 비해 효율적이지만, 항상 그렇지는 않다. 색인할당 ④와 색인할당 ②의 비용식을 비교하면, $h_{XP_2''} + \lceil \frac{k'(i, n) \times (n-i)}{500} \rceil > 2 \times S(i, m) \times (n-i)$, 즉 $h_{XP_2''} + S'(i, n) > 1000 \times S(i, m)$ 이면, 비용식 (11) > 비용식 (7)이다. 따라서, 다음과 같은 요약2를 도출할 수 있다.

요약 2: 그림 4와 같은 겹침 경로에서, 부경로 XP_2''의 요소 공유도가 부경로 XP_1''의 요소 공유도보다 훨씬 클 경우에는 부경로 XP_2''에는 색인을 할당하지 않는 것이 좋다. 그리고 색인할당 ④와 같이 경로할당을 완전히 분리할 경우에는 다차원 경로 색인구조보다 다차원 타겟요소 색인구조를 할당하는 것이 더 효율적이다. 이는 다차원 경로 색인구조에서와 같은 경로 인스턴스의 사용이 필요 없기 때문이다.

(5) XP_1, XP_2'과 XP_2''에 색인이 있는 색인할당 ⑤의 경우

그림 4의 겹침 경로에서 가능한 또 다른 색인할당은 경로 XP_1, XP_2'와 XP_2''에 색인을 할당하는 것이다. 이 경우의 질의처리 전략은 다음과 같다.

(가) 모든 색인구조를 이용하는 전략으로, 먼저 술어 $pred_n$을 평가하기 위하여 경로 XP_2''상의 색인과 경로 XP_2'상의 색인을 차례로 액세스한다. 그리고 술어 $pred_m$을 평가하기 위하여 경로 XP_1상의 색인을 액세스하여 $pred_n$의 평가 결과와 교집합을 취한다.

(나) 이 색인할당에 대해서 더욱 적합한 질의처리 전략으로, 먼저 $pred_m$을 평가하기 위하여 경로 XP_1상의 다차원 경로 색인구조를 액세스하여 (E_1, E_i)쌍들의 집합을 프로젝션하여 구한다. 그리고 $pred_n$을 평가하기 위하여 경로 XP_2''상의 색인을 액세스하고, 그 결과를 $pred_m$의 평가 결과인 (E_1, E_i)쌍들의 집합에서 두 번째 요소와 교집합을 취하여 최종 결과를 구한다.

각 전략의 비용식은 다음과 같다.

$$비용(가) = IAC(XP_2'') + no \times IAC(XP_2') + IAC(XP_1) \quad (13)$$

여기서, $no = S'(i, n)$이다.

$$비용(나) = IAC(XP_1) + IAC(XP_2'') \quad (14)$$

비용식 (13)과 (14)를 비교함으로써 다음과 같은 요약 3를 도출할 수 있다.

요약 3: 주어진 두 개의 겹침 경로 XP_1과 XP_2에서 경로 XP_1은 분리하지 않고 색인을 할당하고, XP_2는 분리하여 XP_2'와 XP_2''에 각각 색인을 할당하는 경우에는 모든 색인을 이용하는 질의처리 전략보다 XP_2'상의 색인은 이용하지 않는 질의처리 전략이 항상 효율적이다.

두 경로 모두에 색인을 사용하는 경우에, 두 경로 모두 분리하지 않는 색인할당이 한 경로를 분리하는 색인할당보다 항상 질의처리 비용이 많다는 중요한

결과를 얻을 수 있다. 이는 색인할당 ③의 비용식 (10)과 비용식 (14)를 비교함으로써 얻을 수 있다. 그리고 이것은 XP_2''가 XP_2의 부경로이고, 부경로에 정의된 색인의 액세스 비용이 항상 XP_2 전체 경로에 정의된 색인의 액세스 비용보다 항상 적다는 사실로서 쉽게 증명할 수 있다. 그러나 이와 같은 결과는 다차원 타겟요소 색인구조를 사용하는 경우와는 반대의 결과이다. 이는 다차원 경로 색인구조의 경우는 프로젝션 연산을 사용함으로써, 다차원 타겟요소 색인구조에서 필요하게 되는 부경로 XP_2'상의 색인을 액세스하기 위한 비용이 절약되기 때문이다.

(6) XP_1과 XP_2''에 색인이 있는 색인할당 ⑥의 경우

색인할당 ⑤의 분석으로부터, 단지 다차원 경로 색인구조만을 사용할 경우에는 지금까지 제안하지 않은 다른 색인할당을 자연스럽게 생각할 수 있다. 이 색인할당은 경로 XP_1상에 색인을 할당하고, XP_2는 XP_2'와 XP_2''로 분리하여 XP_2''에만 색인을 할당하는 것이다. 이와 같은 색인할당에서 질의처리 전략에 대한 비용식은 색인할당 ⑤에서의 비용식 (14)와 같다. 즉, 부경로 XP_2'에는 색인을 사용하지 않는 것이 사용하는 것보다 항상 유리하므로, 가장 좋은 색인할당 전략은XP_2'상에는 색인을 할당하지 않는 것이다.

4.3 세 개 이상의 경로들 사이의 색인할당

본 절에서는 제 4.2절의 결과를 확장하여 세 개 이상의 중첩술어가 결합된 복합 질의를 처리하는 경우에 대해서 색인할당과 그에 따른 질의처리 전략을 제시한다.

먼저, 겹침이 없는 여러 경로들에 각각 정의된 술어들을 가지는 경우이다. 이런 경우는 다차원 경로 색인구조의 할당이 다차원 타겟요소 색인구조의 할당에 비해서 특별히 유용하지는 않다. 특정 경로에 하나의 색인만을 할당하는 경우에는 요소 공유도 $(S(1, n))$가 가장 낮은 경로에 색인을 할당한다. 이런 경우의 질의처리 전략은 먼저 색인된 경로에 주어진 술어들에 대해서 색인을 액세스하여 평가한 후, 그 결과의 요소들에 대해서만 FTNL 전략으로 나머지 술어들을 평가하는 것이다.

그리고, 겹침이 있는 여러 경로들에 각각 정의된 술어들을 가지는 경우이다. 만약 색인구조를 그림 7의 (a)와 같이 분리되지 않은 경로 하나에만 할당한다면, 색인할당 ①의 경우를 일반화하여 요소 공유도 $(S(1, n_h))$가 가장 낮은 경로에 다차원 경로 색인구조를 할당한다. 이런 경우의 질의처리 전략은, 먼저 다차원 경로 색인구조를 액세스하여 색인이 할당된 경로의 술어를 만족하는 경로 인스턴스들을 얻는다. 그리고 이들에 대하여 프로젝션 연산을 수행하여 (E_1, E_i)쌍들을 생성하고, 각 쌍들의 두 번째 원소들로서 나머지 경로들의 술어들을 해결하기 위하여 FTNL 전략을 사용한다. 그리고, 만약 색인구조를 경로 하나에만 할당하되 그 색인구조를 그림 7의 (b)와 같이 분리할 수 있다면, 색인할당 ②의 결과로부터 다차원 경로 색인구조보다 다차원 타겟요소 색인구조를 할당하는 것이 좋다. 이런 경우에는 다차원 경로 색인구조에서와 같은 경로 인스턴스의 사용이 필요 없기 때문이다.

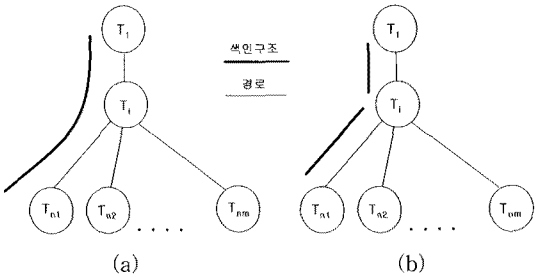


그림 7. 여러 겹침 경로에 대한 하나의 경로에 색인할당 방법.

두 개의 경로 사이에 겹침이 있는 경우에 대한 색인할당 ⑤로부터 얻은 결과를 일반화함으로써 다음과 같은 요약 4를 도출할 수 있다.

요약 4: 겹침이 있는 여러 경로들에서 모든 경로에 색인을 할당하고자 할 경우에는 먼저, 하나의 경로는 분리하지 않고 다차원 경로 색인구조를 할당하고, 나머지 경로들은 모두 분리하여 겹쳐지지 않는 부경로에 대해서만 각각 색인을 할당한다.

요약 4의 경우 질의처리 전략은 분리하지 않은 다차원 경로 색인구조에 대해 프로젝션 연산을 이용하는 전략을 사용한다. 결과적으로, 색인할당 ⑤에서처럼 겹쳐진 부경로에 별도로 할당된 색인은 이용하지 않는 것이 유리하므로 색인할당 ⑥의 색인 할당을 일반화하여 그림 8과 같이 할당한다. 이런 경우에 가장 적합한 질의 수행 전략은 색인할당 ⑤의 질의처리 전략 (나)를 일반화하는 것이다. 즉, 가장 긴 다차원 경로 색인구조를 액세스하여 그 경로에 주어진 중첩 술어를 만족하는 경로 인스턴스들을 탐색하고, 이들

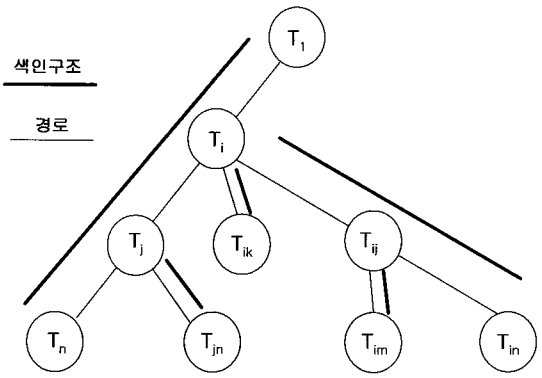


그림 8. 색인할당 ⑥의 일반화.

에 대해 프로젝션 연산을 수행하여 (E_1, ..., E_i, ..., E_j) 형태의 튜플들을 생성한다. 그리고, 이들 중 다른 술어를 만족하지 않는 것들을 제거하기 위하여 각 부경로에 주어진 색인들을 액세스하면 된다.

5. 결 론

본 논문에서는 XML 데이터베이스에서 두 개 이상의 중첩술어를 가지는 복합질의의 처리를 효율적으로 지원하기 위한 다차원 타입상속 색인구조의 할당기법을 제시하였다. 먼저, 하나의 중첩술어가 주어진 경우에는 다차원 타입상속 색인구조의 운용에 대한 분석으로 중첩술어에 나타나는 경로의 길이가 3 이상일 때는 색인 엔트리를 타겟 타입의 EID들만으로 구성하는 다차원 타겟요소 색인구조보다 색인 엔트리를 경로 인스턴스들로 구성하는 다차원 경로 색인구조가 적합함을 보였다. 이는 데이터베이스의 변경에 따른 다차원 타겟요소 색인구조의 유지비용의 오버헤드가 너무 크기 때문이다.

그리고, 두 개의 중첩술어가 주어지는 경우에는 먼저, 겹침이 있는 두 경로에서 한 경로에만 색인을 할당할 경우에는 경로 전체에 대해 하나의 다차원 경로 색인구조를 할당하는 것이 그 경로를 겹침 부경로와 비겹침 부경로로 분리하여 각각 색인구조를 할당하는 경우보다 더 효율적이다. 이는 질의처리시 비겹침 부경로의 색인 액세스의 결과에 따른 겹침 부경로의 색인 액세스의 횟수가 전체 경로에 할당된 색인구조의 크기에 따른 오버헤드를 상회하기 때문이다. 그리고 두 경로 모두 색인을 할당할 경우에는 경로상의 요소 공유도가 낮은 한 경로에 대해서는 경로 전체에 대해 다차원 경로 색인구조를 할당하고, 나머지 경로에 대해서는 비겹침 부경로에 대해서만 색인을 할당하는 것이 좋다. 이는 질의처리시 전체 경로 색인구조의 색인 엔트리에 대한 비겹침 부경로 색인의 결과를 프로젝션 연산으로 질의처리가 가능하기 때문이다.

마지막으로, 세 개 이상의 중첩술어가 주어지는 경우에는 여러 경로들에서 요소 공유도가 가장 낮은 하나의 경로에 대해서는 전체 경로에 하나의 경로 색인구조를 할당하고, 나머지 경로들은 모두 비겹침 부경로에 대해서만 색인을 할당하는 것이 가장 효율적이다. 여기서도 전체 경로에 주어진 다차원 경로 색인구조의 색인 엔트리에 대한 비겹침 부경로 색인의 결과를 프로젝션 연산으로 나머지 경로의 술어들을 처리하는 것이 질의처리 비용을 줄일 수 있기 때문이다. 향후 연구로서 본 논문에서는 색인할당 방법을 비교하기 위하여 질의처리 비용만을 고려하였으나, 색인을 유지하게 되는 경로의 길이에 따른 다차원 경로 색인구조의 유지비용의 고려가 필요하며, 본 논문에서 제시한 비용모델 분석을 실제로 구축한 다양한 데이터베이스 환경에서 실제 실험을 통한 검증작업이 필요하다.

참 고 문 헌

[1] T. Bray et al., *Extensible Markup Language, (XML) 1.0. W3C Recommendation*, http://www.w3.org /TR /REC-xml-19980210, Feb. 2004.

[2] C. W. Chung, J. K. Min, and K. Shim. "APEX: An Adaptive Path Index for XML Data," In *Proc. Intl. Conf. on Management of Data, ACM SIGMOD*, Madison, Wisconsin, pp. 121-132, June, 2002.

[3] E. Bertino and B. C. Ooi, "The Indispensability of Dispensable Indexes," *IEEE Trans. on Knowledge and Data Eng.*, Vol.11, No.1, pp. 17-27, Jan. 1999.

[4] W. Meier, "eXist: An Open Source native XML Database," *Web, Web-Services, and Database Systems, NODe 2002 Web- and Database-Related Workshops,* Revised Papers (Lecture Notes in Computer Science

Vol.2593), pp. 169-183, 2003.

[5] C. D. Fallside and P. Walmsley, *XML Schema Part 0. W3C Recommendation*, http: //www.w3.org/TR/xmlschema-0, Oct. 2004.

[6] A. Berglund et al., "XML Path Language (XPath) 2.0. W3C Working Draft 30 Apr. 2002," *http://www.w3.org/TR/xpath20,* Working Draft, 2002.

[7] B. F. Cooper et al., "A Fast Index for Semistructured Data," In *Proc. Intl. Conf. on Very Large Data Bases*, Rome, Italy, pp. 341-350, Sept. 2001.

[8] S. Boag et al., *XQuery 1.0: An XML Query Language*, http://www.w3.org/TR/xquery, Nov. 2005.

[9] A. Kemper and G. Moerkotte, "Access Support Relations: An Indexing Method for Object Bases," *Information Systems*, Vol.17, No.2, pp. 117-145, 1992.

[10] W. Kim, "A Model of Queries for Object-Oriented Databases," In *Proc. Intl. Conf. on Very Large Data Bases*, pp. 423-432, Amsterdam, Aug. 1989.

[11] R. Goldman and J. Widom, "DataGuides: Enable Query Formulation and Optimization in Semistructured DataBases," In *Proc. Int'l Conf. on Very Large Data Bases*, Athens, Greece, pp. 436-445, Aug. 1997.

[12] S. Nestorov et al., "Representative Objects: Concise Prepresentation of Semistructured, Hierarchical Data," In *Proc. IEEE Int'l Conf. on Data Engineering*, Birmingham, U.K., pp.79-90, Feb. 1997.

[13] T. Milo and D. Suciu, "Index Structures for Path Expression," In *Proc. Int'l Conf. on Database Theory*, Jerusalem, Israel, pp. 277-295, Jan. 1999.

[14] D. Comer, "The Ubiquitous B-tree," *ACM Computing Surveys*, New York, USA, Vol.11, No.2, pp. 121-137, June 1979.

[15] J. L. Bentley, "Multidimensional Binary Search Trees in Database Applications," *IEEE Trans. on Software Eng.*, Vol.5, No.4, pp. 333-340, July 1979.

[16] J. T. Robinson, "The K-D-B-Tree: A Search Structure for Large Multidimensional Dynamic Indexes," In *Proc. Int'l Conf. on Management of Data, ACM SIGMOD*, Ann Arbor, Michigan, pp. 10-18, Apr. 1981.

[17] K. Y. Whang and R. Krishnamurthy, "The Multilevel Grid File - A Dynamic Hierarchical Multidimensional File Structure," In *Proc. Intl. Conf. on Database Systems for Advanced Applications (DASFAA)*, pp. 449-459, Tokyo, Apr. 1991.

[18] J. H. Lee, "MD-TIX: Multidimensional Type Inheritance Indexing for Efficient Execution of XML Queries," *Journal of Korea Multimedia Society*, Vol.10, No.9, pp. 1093-1105, Sept. 2007.

[19] K. C. Kim et al., "Acyclic Query Processing in Object-Oriented Databases," In *Proc. Intl. Conf. on Entity-Relationship Approach*, Rome, Italy, pp. 329-346, Nov. 1989.

[20] W. Kim, *Introduction to Object-Oriented Databases*, The MIT Press, 1990.

[21] J. H. Lee, "2D-THI: Two-Dimensional Type Hierarchy Index for XML Databases," *Journal of Korea Multimedia Society*, Vol.9, No.3, pp. 265-278, Mar. 2006.

이 종 학

1982년 경북대학교 전자공학과 (전자계산 전공) 졸업 (학사)
1984년 한국과학기술원 전산학과 졸업(공학석사)
1997년 한국과학기술원 전산학과 졸업(공학박사)
1991년 정보처리기술사
1984년~1987년 금성통신(주) 부설연구소 주임연구원
1987년~1998년 한국통신 연구개발본부 선임연구원
1998년~현재 대구가톨릭대학교 컴퓨터정보통신공학부 교수
관심분야 : 색인구조, 다차원 파일구조, 데이터베이스 설계, XML 데이터베이스, 데이터 웨어하우스, 생물정보학 등