객체 데이타베이스를 위한 다차원 중포 색인구조의 운용과 할당

이정아, 임윤주 이종학
대구가톨릭대학교 컴퓨터정보통신공학부

Operations And Assignments Of Multidimensional Nested Indexs For Object Databases

Jung-A Lee ,Yun-Ju Lim, Jong-Hak Lee
School of Computer and Informaion Communications Engineering, Catholic Univ. of Daegu

요 약

지난 몇 년간 체세대 데이터베이스 시스템으로서 객체 데이터베이스 시스템의 객체 질의연구가 이루어지고 있으며, 특히 고급 질의의 처리비용을 줄이기 위한 연구가 활발하다. 최근에 제안된 중포 속성 색인기법은 객체지향 질의 처리의 성능 향상에 크게 기여하고 있다. 하지만 이들 색인구조들은 기존의 관계형 데이터베이스에서 사용된 단순 속성에 대한 색인구조에 비해 저장공간과 갱신 유지비용이 크다. 또한 클래스 상속에 의한 객체 데이터 모델의 특징을 반영하지 못하며, 타켓 클래스 및 도메인 클래스 대치가 있는 경로식으로 표현된 질의는 지원하지 못한다. 따라서 본 논문에서는 객체 데이터베이스의 주요 특징을 반영할 수 있는 색인구조인 다차원 중포 색인구조와 경로 색인구조에 대한 운용법을 제시한다. 또한 효과적인 질의 처리를 하기 위한 효율적인 색인할당방법을 제시한다. 이로써 객체지향 데이터베이스 시스템에서의 고급 질의의 처리비용을 줄일 수 있다.

1. 서론

객체 데이타베이스 관리 시스템(object database management system)은 객체 지향 데이터베이스를 정의하고 조작할 수 있는 데이타베이스 시스템이다. 최근 차세대 데이타베이스 시스템으로서 객체 데이타베이스 시스템의 객체 질의 연구가 이루어지고 있으며, 특히 고급 질의의 처리비용을 줄이기 위한 연구가 활발하다[2]. 최근에 제안된 중포 속성 색인기법은 객체지향 질의 처리의 성능 향상에 크게 기여하고 있다.

하지만 이들 색인구조들은 기존의 관계형 데이타베이스에서 단순 속성에 대한 색인구조에 비해 저장공간과 갱신유지 비용이 크다. 또한 클래스 상속에 의한 객체 데이타 모델의 특징을 반영하지 못하며, 타겟 클래스 및 도메인 클래스 대치가 있는 경로식으로 표현된 질의는 지원하지 못한다. 그러므로 경로상의 타겟 클래스 대치와 도메인 클래스 대치 모두 있는 질의를 지원할 수 있는 색인구조의 연구가 필요하다. 또한, 객체 데이타베이스 색인기법의 장점을 이용하여 질의 처리를 하기 위해서는 색인들을 좀 더 효율적으로 구성할 수 있는 색인할당 방법의 연구도 필요하다.

따라서 본 논문에서는 객체 데이타베이스의 주요 특징을 반영할 수 있는 색인구조인 다차원 중포 색인구조와 경로 색인구조에 대한 운용법을 제시한다. 또한 효과적인 질의처리를 하기 위한 효율적인 색인할당 방법을 제시한다.

2. 관련연구

객체 데이타 모델은 데이타를 모델링하기 위해서 복합 객체, 객체 식별자, 클래스, 캡슐화, 서브클래스 관계성, 참조 관계성 등의 유용한 객체지향 개념들로 구성된다[5]. 실세계에서 임의의 한 존재는 하나의 객체가 될 수 있으며, 각각의 객체는 시스템 내에서 유일한 식별자(object identifier : Oid)를 갖는다.

클래스는 기존의 클래스에 속성이나 메소드를 추가하여 정의할 수 있다. 새로 정의된 클래스를 기존의 클래스에 대한 서브 클래스라 한다. 결과적으로 데이타베이스 내의 클래스들은 루트를 갖는 계층을 형성하며 이를 클래스 상속 계층 또는 클래스 계층이라 한다. 클래스 계층에서 하위(상위) 클래스는 상위(하위) 클래스의 서브(수퍼) 클래스이며, 하위 클래스는 상위 클래스에서 정의된 모든 속성과 메소드를 상속한다. 하위 클래스에 속하는 인스턴스들은 그의 상위 클래스의 인스턴스가 된다.

임의의 클래스에 속하는 객체는 속성의 값으로 다른 클래스에 속하는 객체들을 가질 수 있다. 여기서 속성이 정의된 클래스를 루트 클래스라 하고, 속성의 값이 되는 객체가 속한 클래스를 도메인 클래스라 한다. 즉 데이타베이스 내의 클래스들은 루트 클래스와 도메인 클래스의 관계에 의해서 클래스 집단화 계층을 형성한다.

객체지향 질의어는 관계 질의어인 SQL처럼 select, from, where절로 구성하여, 각 절에서 객체지향 개념을 지원하도록 확장하여 사용한다. 그리고 Where절에서는 중포 속성에 대한 조건인 중포 술어(nested predicate)를 사용할 수 있다. 객체지향 질의에서 중포 술어는 경로식으로 표현한다[2, 3]. 경로식은 클래스 집단화 계층구조상에서 클래스 이름과 속성의 교차적인 나열로서 다음과 같은 형태를 가

진다.

$$P=C_1.A_1\{[C_2]\}.A_2\{[C_3]\}\cdots A_n\{[C_{n+1}]\}$$

클래스 C_1을 타겟 클래스, 속성 A_i의 도메인이 되는 C_{i+1}을 A_i의 도메인 클래스라 정의한다.

경로식에서 속성의 도메인 클래스가 특정 클래스로 한정되는 것을 도메인 대치라 하고, 타겟 클래스가 특정 클래스로 한정되는 것을 타겟 대치라 한다. 타겟 클래스나 도메인 클래스의 대치를 통칭하여 클래스 대치라고 부른다. 이러한 클래스 대치는 질의의 범위를 특정한 클래스로 한정할 수 있도록 하여 클래스 상속의 개념을 객체지향 질의에 표현하도록 한 것이다[4,5].

객체지향 데이타베이스에서 기존의 색인구조로서 중포 색인[1], 경로 색인[1], ASR(Access Support Relation)[3], JIH(Join Index Hierarchy)[6] 등이 있다. 그러나 이러한 색인구조들은 클래스 상속에 의한 객체 데이타 모델의 특징을 반영하지 못하므로 타겟 클래스 대치, 도메인 클래스 대치가 있는 경로식으로 표현된 질의는 지원하지 못한다. 또한 Bertino 등이 제안한 NIX(Nested-Inherited Index)[2] 색인구조는 타겟 클래스 대치가 있는 경로식으로 표현된 질의를 지원하지만 도메인 클래스 대치가 있는 경로식을 가지는 질의를 지원하지 못한다.

3. 중포 속성에 대한 다차원 색인구조

객체 데이타베이스는 한 클래스가 가지는 속성의 도메인이 또 다른 클래스가 되게 함으로써 클래스 집단화 계층을 이루고 있다. 이러한 클래스 집단화 계층상의 모든 클래스에서 정의된 속성은 논리적으로 루트 클래스의 속성이라 하는데 이 속성을 중포 속성이라 한다[5].

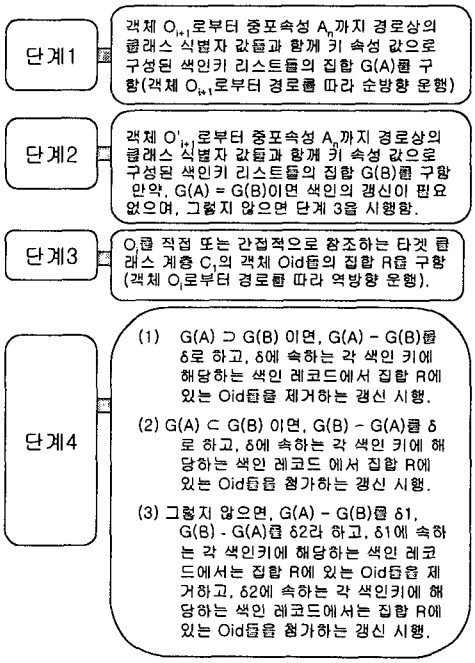
다차원 화일구조는 여러 애트리뷰트 값들을 기반으로 한 레코드를 액세스하는 다중 애트리뷰트 액세스를 효과적으로 처리하기 위한 화일구조이다[8]. 또한 이는 다차원 클러스터링을 지원하는 파일구조이다. 여러 개의 애트리뷰트에 대한 검색을 할 경우에는 하나의 다차원 화일구조를 사용하는 것이 여러 개의 B'-tree와 같은 일차원 색인구조를 사용하는 것에 비해 검색 성능이 우수하다.

객체 데이타베이스 질의어인 중포 술어에는 클래스 대치가 있을 수 있다. 이러한 중포 술어의 처리를 지원하기 위한 색인구조로 다차원 색인구조를 이용할 수 있다. 즉, 색인할 중포 속성의 키 속성 도메인과 중포 속성을 표현하는 경로상의 각 클래스 계층마다 클래스 식별자들로 구성된 한차원씩의 클래스 식별자 도메인을 할당하여 다차원의 도메인 공간을 형성한다.

우리는 이와 같은 색인구조를 다차원 중포 속성 색인(multidimensional nested attribute index : MD-NAI)구조라 한다[7]. MD-NAI는 색인 레코드에 있는 색인 엔트리의 구성방법에 따라, 다차원 중포 색인(multidimensional nested index : MNI)구조와 다차원 경로 색인(multidimensional path index : MPI)구조의 두 가지 색인구조로 분류할 수 있다. MNI는 색인 엔트리를 색인된 중포 속성의 타겟 클래스 계층에 속하는 객체에 대한 객체 식별자들로 구성되고, MPI는 색인 엔트리를 색인된 중포 속성에 대한 경로 인스턴스(Oid들의 리스트)로 구성된다. MPI와 같이 색인 엔트리를 경로 인스턴스들로 구성하는 것은 색인 엔트리를 타겟 클래스 계층의 객체 식별자만으로 구성하는 경우 발생하게 되는 데이타베이스의 변경에 따른 색인구조의 막대한 유지비용을 줄이기 위함이다[1, 2].

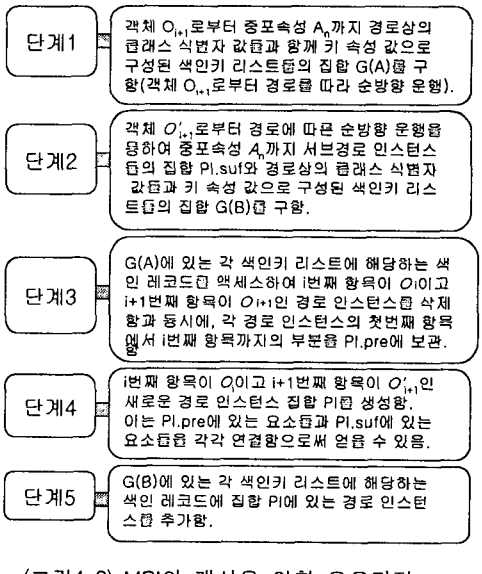
4. 다차원 중포 속성 색인구조의 운용

(그림4-1)은 경로 P상의 i번째 클래스 계층에 있는 임의의 객체 O_i에서 속성 A_i의 값으로 O_{i+1}를 새로운 O'_{i+1}로 변경될 경우의 MNI의 갱신을 위한 운용과정을 나타낸다.



(그림 4-1) MNI의 갱신을 위한 운용과정

(그림4-2)은 경로 P상에서 i번째 클래스 계층에 있는 임의의 객체 O_i에서 속성 A_i의 값으로 O_{i+1}을 새로운 O'_{i+1}로 변경될 경우의 MPI의 갱신을 위한 운용과정을 나타낸다.



(그림4-2) MPI의 갱신을 위한 운용과정

위에서 살펴본 바와 같이 각 색인구조의 운용에 따른 MNI색인구조와 MPI색인구조의를 상호 비교하여 요약하면

(표 4-1)과 같다.

(표 4-1) 운용에 따른 MNI와 MPI의 비교

구조 비교 \ 색인	MNI	MPI
저장공간 오버헤드	적음	많음
유지비용 오버헤드	많음	적음
갱신시	역방향 운행 필요	역방향 운행 불필요
역참조자	필요	불필요

5. 다차원 중포 속성 색인구조의 할당

5.1 단일 술어에 대한 할당

검색 질의에 대한 두 색인구조의 성능에 대해서는 MNI가 MPI에 비해 좋은 성능을 가진다. 이는 MPI에서는 색인 엔트리를 색인된 중포속성의 경로 인스턴스로 구성하기 때문에 색인 엑트리를 타켓 클래스 계층에 속하는 객체 식별자만으로 구성하는 MNI에 비하여 많은 저장 공간의 오버헤드 때문이다. 그러나, 두 색인구조의 운용에 따른 유지비용에 대한 오버헤드는 색인된 중포속성의 경로 길이에 따라 많은 차이를 보이게 된다.

술어가 있는 경로의 길이가 2이고 역 참조자가 데이타베이스의 객체 내에서 제공될 경우에는 MNI와 MPI으 유지비용은 같게 된다. 그 이유로서, 먼저 경로상의 두 번째 클래스 계층에 있는 객체 O_2에서 색인된 속성 A2가 변경되면 객체 O_2내에 역 참조자가 존재하기 때문에 객체 O_2를 참조하는 첫 번째 계층의 객체를 결정하기 위한 역방향 운행이 필요 없기 때문이다. 그리고, 술어가 있는 경로의 길이가 3이상인 경우에 MNI에서는 역 방향 운행이 필요하게 되므로 MPI에 비해 유지비용이 증가하게 된다. MNI의 유지비용을 지배하는 것은 역방향 운행에 의한 것으로, 역방향 운행에 필요한 객체의 액세스 개수는 객체 참조 공유도[1]에 의해 결정된다. MPI에서는 역방향 운행이 필요 없기 때문에 MNI에서보다 유지비용이 매우 적게 된다.

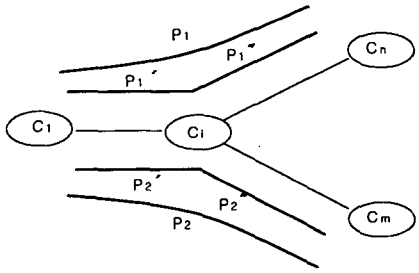
따라서, 위와 같은 분석의 결과로서 다음과 같은 결론을 얻을 수 있다. 첫째로 색인을 구축할 경로의 길이가 2인 경우에는 MNI가 적합하다. 이것은 유지비용은 두 가지 색인구조 모두 비슷한 반면에 MNI에서 검색 성능이 좋기 때문이다. 둘째로 색인을 구축할 경로의 길이가 3이상인 경우에는 일반적으로 MPI가 적합하다. 이것은 MPI의 검색 성능이 MNI에 필적하는 반면에 데이타베이스 변경에 의한 유지비용이 적게 되기 때문이다. 그리고 MPI는 각 객체 내에 역 참조자가 존재하지 않을 경우에도 사용이 가능하다.

5.2 두 개의 술어에 대한 할당

단일 술어로 된 (그림 5-1)은 집단화 계층에 따른 경로 구성을 본 논문에서 앞으로 사용할 경로들의 이름과 함께 나타낸 것이다. (그림 5-1)의 분리된 경로에서 분리가 시작되는 클래스 계층은 C_i이다. 그리고, 색인구조를 이용하지 않는 순방향 운행법(FT:Forward Traversal)에 의한 질의처리 비용식은 다음 식(1)과 같으며, 앞으로 각 색인할당 방법에서 질의처리 비용을 계산하기 위하여 이 식을 이용한다.

$$Cost\_FT = PC(C_i) + 2 \times N_i \times (n-i) \quad (1)$$

여기서, $PC(C_i)$는 클래스 C_i의 인스턴스들을 가지는 디스크 페이지의 개수($1 \le i \le n$)이다.



(그림 5-1) 분리된 경로구성 스키마.

(1) 색인할당 방법(1): P_1과 P_2에 MPI색인을 할당하는 경우

이 색인할당 방법의 경우는 두 경로 P_1과 P_2 모두 분리하지 않고 색인을 할당하는 경우이다. 이 경우의 질의처리 전략은 다음과 같다.

① $pred_n$을 평가하기 위해 경로 P_1상의 색인을 액세스한다.
② $pred_m$을 평가하기 위해 경로 P_2상의 색인을 액세스한다.
③ ①과 ②의 평가결과를 교집합한다.

색인할당 방법(1)의 비용식은 다음과 같이 나타낼 수 있다.

$$Cost(1) = MPA\_Cost(P_1) + MPA\_Cost(P_2) \quad (2)$$

여기서, MPA는 MPI 색인구조의 액세스 비용이다.

(2) 색인할당 방법(2): P'_1, P''_1과 P''_2에 MPI색인을 할당하는 경우

이 색인할당 방법의 경우는 두 경로 모두 분리하여 색인을 할당하는 경우이다. 여기서 두 경로의 첫 번째 부경로의 색인들은 동일하다. 이 경우의 질의처리 전략은 다음과 같다.

① $pred_n$을 평가하기 위해 경로 P''_1상의 색인을 액세스한다.
② $pred_m$을 평가하기 위해 경로 P''_2상의 색인을 액세스한다.
③ ①과 ②의 평가결과를 교집합한다.
④ 경로 P'_1상의 색인을 액세스 하여 최종결과를 얻는다.

색인할당 방법(2)경우의 질의처리전략들의 비용식은 다음과 같다.

$$Cost(2) = MPA\_Cost(P''_1) + MPA\_Cost(P''_2) + no \times MPA\_Cost(P'_1) \quad (3)$$

여기서, $no = \lceil k(i, n)/D'_m \rceil$로써 두 개의 술어 $pred_n$과 $pred_m$을 동시에 만족하는 객체의 수를 나타낸다.

(3) 색인할당 방법(3): P_1, P'_2와 P''_2에 MPI색인을 할당하는 경우

이 색인할당 방법의 경우는 경로 P_1, P'_2와 P''_2에 색인을 할당하는 것이다. 이 경우의 질의처리전략은 다음과 같다.

① $pred_n$을 평가하기 위하여 경로 P_1상의 색인을 액세스한다.
② $pred_m$을 평가하기 위하여 경로 P'_2상의 색인과 경로 P''_2상의 색인을 차례로 액세스한다.
③ ①과 ②의 평가 결과를 교집합한다.

색인할당 방법(3)경우의 질의처리전략의 비용식은 다음과 같다.

$$Cost(3) = MPA\_Cost(P''_2) + no \times MPA\_Cost(P'_2) + MPA\_Cost(P_1) \quad (4)$$

여기서, $no = k'(i, m)$로서 $pred_m$을 만족하는 클래스 C_i의 객체 수이다.

(4) 색인할당 방법(4): P_1과 P''_2에 MPI색인을 할당하는 경우

1) 클래스 계층 Ci에서 속성 Ai의 값으로 동일한 값을 가지는 객체의 평균 개수.

이 색인할당 방법의 경우는 주어진 두 개의 겹침 경로 P_1과 P_2에서 경로 P_1은 분리하지 않고 색인을 할당하고, P_2는 분리하였지만 P''_2에만 색인을 할당할 경우이다. 이 경우의 질의처리전략은 다음과 같다.

① $pred_n$을 평가하기 위하여 경로 P_1상의 MPI색인을 액세스하여 (O_1, O_i)쌍들의 집합을 프로젝션하여 구한다.

② $pred_m$을 평가하기 위하여 경로 P''_2상의 색인을 액세스하여 O_i 객체들의 집합을 구한다.

③ ①의 결과에서 두번째 항이 ②의 결과에 있는 첫번째 항을 구한다.

색인할당 방법(4)경우의 질의처리 전략의 비용식은 다음과 같다.

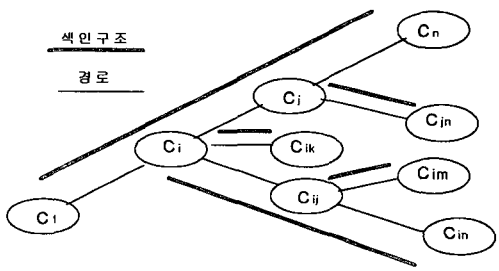
$$Cost(4) = MPA\_Cost(P_1) + MPA\_Cost(P''_2) \quad (5)$$

(5) 각 색인구조의 할당에 따른 질의처리 비용식의 비교

색인할당 방법(1), (2), (3), (4)에 대한 질의처리 비용식을 비교한 결과로 두 경로 모두에 색인을 할당하는 경우에는, 두 경로 모두 부경로로 분리하지 않는 색인할당 방법(1)이 두 경로 모두 또는 한 경로를 분리하는 색인할당 방법(2), (3)보다 질의처리 비용이 적다. 그리고 한 경로를 분리하여 공통의 부경로(P'2)에는 색인을 할당하지 않는 색인할당 방법(4)가 두 경로 모두 부경로로 분리하지 않는 색인할당 방법(1)보다 질의처리 비용이 적다. 따라서 가장 최적의 색인할당방법은 색인할당방법(4)이다.

5.3 두 개 이상의 술어에 대한 할당

두 개의 경로 사이에 겹침이 있는 경우에 대한 색인구성의 결과를 일반화함으로써 다음과 같은 결과를 얻을 수 있다. 즉, 겹침이 있는 여러 경로들에서 모든 경로에 색인을 구성하고자 할 경우에는 먼저, 하나의 경로는 분리하지 않고 MPI 색인을 구성하고, 나머지 경로들은 모두 분리하여 겹쳐지지 않는 부경로에 대해서만 각각 색인을 구성하는 것이 좋다. 이런 경우의 질의처리 전략은 분리하지 않은 MPI 색인에 대해 프로젝션 연산을 이용하는 전략을 사용한다. 결과적으로, 겹쳐진 부경로에 별도로 구성된 색인은 이용하지 않는 것이 유리하므로 색인할당 방법(4)의 색인 구성을 일반화하여 (그림 5-2)와 같이 구성한다. 이런 경우에 가장 적합한 질의 수행 전략은 색인구성 (4)의 질의처리 전략을 일반화하는 것이다. 즉, 가장 긴 MPI 색인을 액세스하여 그 경로에 주어진 중포술어를 만족하는 경로 인스턴스들을 탐색하고, 이들에 대해 프로젝션 연산을 수행하여 (O_1, O_i, ..., O_j) 형태의 투플들을 생성한다. 그리고, 이들 중 다른 술어를 만족하지 않는 것들을 제거하기 위하여 각 부경로에 주어진 색인들을 액세스하면 된다.



(그림 5-2) 색인할당 방법(4)의 일반화.

6. 결론

객체 데이타베이스 색인구조의 유지비용을 줄이기 위하여 저장공간 및 갱신유지 비용을 최소화할 수 있는 효과적인 색인할당 방법을 제시하였다. 그리고 객체 데이타베이스의 중포 술어에 타겟 클래스 계층과 도메인 클래스 계층 모두에 클래스 대치가 있는 질의처리를 지원할 수 있는 차원 중포 색인구조와 다차원 경로 색인구조의 운용법을 제시하였다.

제시된 방법의 타당성을 확인하기 위하여 하나의 경로를 대상으로 각 색인구조의 운용에 따른 유지비용을 비교하였다. 비교 결과로 경로의 길이가 2인 경우는 다차원 중포색인을 할당하고, 경로의 길이가 3인 경우는 다차원 경로색인을 할당하는 것이 가장 효율적이었다. 특히 경로의 길이가 4이상일 경우에는 경로의 길이를 1, 2, 또는 3인 부경로들로 구성할 수 있으므로 각 부경로별로 상기에서 제시한 색인구조를 할당한다.

한편 상하 클래스간에 두 개의 겹침 경로를 가지는 경우에 색인할당 방법을 분류해서, 각각에 대한 질의처리전략과 비용식을 제시하였다. 제시된 비용식을 이용하여 각 계층 간의 공통 부경로에는 색인을 할당하지 않는 것이 가장 효과적인 색인할당 방법임을 알 수 있었다. 이는 질의 처리 시 색인 엔트리에 프로젝션 연산을 이용함으로써 비용을 최소화 할 수 있기 때문이다. 또한 상기 결과를 기반으로 하여 세 개 이상의 경로를 대상으로 하는 일반적인 색인할당 방법을 제시하였다. 즉, 여러 경로가 겹치는 경우에는 객체 공유도가 가장 낮은 경로에 대해서 다차원 경로 색인을 할당한다. 그리고 나머지 경로에는 두 개의 부경로로 분리하여 비 겹침 경로에만 색인을 할당하는 것이 가장 효율적으로 질의처리를 지원한다는 것을 알 수 있었다.

[참고문헌]

[1] Bertino, E. and Kim, W., "Indexing Techniques for Queries on Nested Objects," *IEEE Trans. on Knowledge and Data Eng.*, Vol. 1, No. 2, pp. 196-214, June 1989.

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

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

[4] Kifer, M., Kim, W., and Sagiv, Y., "Querying Object-Oriented Databases," In *Proc. Intl. Conf. on Management of Data, ACM SIGMOD, San Diego, Calif.*, pp. 393-402, May 1992.

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

[6] Xie, Z. and Han, J., "Join Index Hierarchies for Supporting Efficient Navigations in Object-Oriented Databases," *In Proc. Intl. Conf. on Very Large Data Bases, Santiago, Chile*, pp. 522-533, Sept. 1994.

[7] 이종학, "다차원 파일구조를 이용한 객체지향 데이타베이스의 중포속성 색인기법," *한국정보처리학회논문지*, 제 7권, 제 8호, pp. 2298-2309, 2000년 8월.

[8] 황규영 외 5명 역, 데이타*베이스시스템*, 영한출판사, 제 3판, 2002년 2월.