

경로 인덱스를 이용한 데이터 웨어하우스의 질의 처리 기법

이정남¹, 조완섭², 이충세¹, 김홍기¹

¹충북대학교 전자계산학과, ²충북대학교 경영정보학과

Query Processing Techniques for Data Warehouses using Path Indices

Jeong-Nam Lee¹, Wan-Sup Cho², Chung-Set Rhee¹, Hong-Ki Kim¹

¹Dept. of Computer Science, ChungBuk Nat'l Univ., ²Dept. of MIS, ChungBuk Nat'l Univ

본 논문에서는 객체-관계형 데이터베이스 관리 시스템(Object-Relational DBMS : ORDBMS) 기반의 데이터 웨어하우스(Data Warehouse)에서 성능 향상을 위한 인덱싱 기법과 이를 이용한 질의 처리 기법을 제안한다. 지금까지 관계형 DBMS를 기반으로 한 데이터 웨어하우스의 성능 향상에 관한 연구는 활발히 이루어져 왔으나, ORDBMS에 기반한 데이터 웨어하우스의 구축 및 질의 처리 성능에 관한 연구는 거의 이루어지지 않고 있다. 데이터 웨어하우스는 기존의 데이터베이스와는 비교할 수 없을 만큼의 대용량 데이터를 저장하므로 ORDBMS를 이용하여 데이터 웨어하우스를 구축하는 경우에도 적절한 성능의 보장이 필수적으로 요구된다. 제안된 인덱싱 기법을 사용함으로써 데이터 웨어하우스 분석용 질의에 포함된 비용이 큰 조인과 그룹핑 연산은 비용이 저렴한 인덱스 액세스 연산으로 대체되며, 데이터의 량과 거의 무관하게 질의 처리 비용이 고정되는 효과를 얻을 수 있다.

1. 서론

현대 사회의 분산된 많은 양의 정보를 이용해서 기업의 의사 결정에 필요한 정보를 추출해내기 위해서는 현재의 데이터베이스 기술로는 불가능하다. 따라서 하나의 저장소에 공동의 데이터를 저장할 수 있는 새로운 저장 구조가 필요하다. 데이터 웨어하우스(Data Warehouse: 데이터 웨어하우스)는 기업의 의사결정을 지원하기 위한 주제 중심적이고, 통합적이며, 시간성을 가지는 비휘발성 자료의 집합이라고 정의할 수 있다[1,3,5,10,12]. 이러한 데이터 웨어하우스에 최종 사용자가 직접 접근하여 대화식으로 정보를 분석하고 의사 결정에 활용하는 과정을 온라인 분석 처리 과정(On-Line Analytical Processing : OLAP)이라 한다[7,11]. OLAP의 구현 방식은 다차원 데이터베이스(Multidimensional Database)에 기반한 MOLAP(Multidimensional OLAP)과 관계형 데이터베이스에 기반한 ROLAP(Relational OLAP)으로 나눌 수 있다[7,11]. 그러나 최근의 데이터는 행과 열로 표현되는 정형적인 데이터에 텍스트, 이미지, 비디오, 시공간 데이터 등의 다양한 멀티미디어 데이터가 증가하고 있으므로 비정형적인 데이터를 지원할 수 있는 객체-관계형 데이터베이스관리시스템(ORDBMS)과 모델링을 이용한 데이터 웨어하우스를 구축하여 유용하고 복잡한 데이터를 의사결정에 이용할 수 있어야 한다[8].

데이터 웨어하우스는 대량의 데이터에 접근과 복잡한 질의를 통해서 유용한 정보를 분석하고 예측하기 때문에 신속한 자료 분석을 위한 성능 향상을 위한 기법이 필수적이다. 기존의 ROLAP에서는 스타 스키마나 비트맵 인덱스와 같은 성능향상 기법을 이용했다[1,3]. 객체-관계형 데이터베이스에 기반한 데이터 웨어하우스는 상속 및 메소드를 통해서 데이터와 연산을 계산할 수 있으며, OID를 통한 클래스의 탐색(traversal)을 함으로써 조인의 수를 줄일 수 있는 장점이 있지만, 빠른 질의처리를 위한 성능 향상 기법의 연구가 필요한 실정이다.

본 논문에서는 객체-관계형 데이터베이스에 기반한 데이터 웨어하우스의 성능 향상을 위한 인덱싱 기법을 제안한다. 그리고, 객체-관계형 OLAP 질의어를 처리하는 경우에 질의 처리비용을 수직적으로 모델링하고, 경로 인덱스를 이용하여 데이터 웨어하우스의 질의 처리 성능을 향상시킬 수 있음을 입증한다.

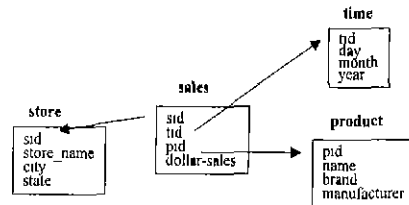
본 논문의 구성은 다음과 같다. 제2장은 객체-관계형 데이터베이스에 기반한 데이터 웨어하우스에 관련된 연구를 기술한다. 제3장에서는 경로 인덱스의 구축을 이용한 객체-관계형 데이터 웨어하우스의 성능 향상 기법을 제시하고, 제4장에서 결론을 맺는다.

2. 관련 연구

본 장에서는 관련연구로써 객체-관계형 데이터 웨어하우스의 설계와 질의를, 기술하고, 구축된 데이터 웨어하우스의 성능 향상을 위한 방안을 살펴본다.

2.1 객체-관계형 스타 스키마

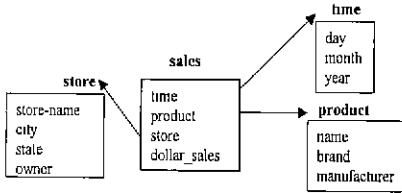
ROLAP의 스타 스키마는 그림1과 같이 사실 테이블이 각 차원 테이블 대한 외래키를 가지는 형태로 데이터 웨어하우스를 구축한다. 그러나, 객체-



—> 외래키-주키 관계

< 그림 1 > 관계형 스타 스키마

관계형 모델을 이용하는 경우에 관계형 모델에서의 외래키-주키 관계는 속성-도메인 관계를 나타내는 쿼리 속성으로 변환된다. 즉, 사실 테이블은 사실 클래스, 차원 테이블은 차원 클래스로 변환되고, 사실 클래스의 참조 속성이 차원 클래스를 참조하는 형태이다. 그림 2는 그림 1의 관계형 스타 스키마를 객체-관계형 DW 스키마로 변환한 예를 보여준다. 그림 2에서 sales는 사실 클래스이고, time, store, product는 각각 차원 클래스이다. 사실 클래스와 차원 클래스 간의 연결선은 에트리뷰트-도메인 관계를 나타낸다. 객체-관계형 스타 스키마에서는 외래키-주키가 아니라 경로식(path expression)을 사용하여 명시적 조인이 이루어지고 다차원적 질의가 가능하게 된다. 또한, 객체-관계형 모델링이므로 각 클래스는 서브 클래스(subclass)를 가질 수 있고, 각 차원 클래스의 에트리뷰트도 사용자 정의 클래스를 도메인으로 가질 수 있다 [14].



→ 애틀리뷰트-도메인 관계

< 그림 2 > 객체-관계형 DW 스키마

2.2 객체지향 질의어를 이용한 OLAP 질의

동일한 질의내용에 대해서 관계형 스키마에 기반한 OLAP 질의문과 객체-관계형 데이터 웨어하우스 스키마에 기반한 OLAP 질의문을 작성하여 서로 비교하고 분석한다.

[질의내용] 5 월에 서울에서 판매된 상품을 제조회사별로, 상품이름 별로 총액을 구하라.

[질의문1] 그림1에 대한 관계형 질의

```
SELECT maker, name, sum(amount)
FROM sales, product, store, time
WHERE sales.pid=product.pid
AND sales.sid=store.sid
AND sales.tid =time.tid
AND time.month=5 andstore.city='서울'
GROUP BY product.maker, product.name,
```

[질의문2] 그림2에 대한 객체-관계형 질의

```
SELECT s.product.maker, s.product.name, sum(amount)
FROM sales s
WHERE s.time.month=5
AND s.store.city='서울'
GROUP BY s.product.maker, s.product.name;
```

질의문1과 질의문2는 동일한 의미를 갖지만 질의문2는 OODB에서의 경로식을 이용하기 때문에 명시적 조인조건이 불필요하다 따라서 질의 1에 비하여 간단하고 직관적이다 질의 2의 FROM 절에는 사실 클래스인 명시하며, WHERE 절에서 애틀리뷰트-도메인 관계의 경로식(path expression)을 사용하여 각 차원 클래스와 조인을 수행한다 GROUP BY 절과 SELECT 절에도 길이가 2 이상인 경로식을 사용하여 그룹핑과 프로젝션을 명시할 수 있다

2.3 객체-관계형 데이터 웨어하우스의 성능향상 방안

데이터 웨어하우스에 저장되는 정보는 기존의 OLTP(On-Line Transaction Processing)용 데이터베이스에 저장되는 정보의 량과는 비교할 수 없을 정도로 대용량이므로 성능의 향상을 위한 특별한 대안이 있어야 한다. 이러한 대안으로 요약 정보 및 실제화된 뷰의 생성과 OLAP 질의어의 대표적인 특성인 스타 조인과 집계 함수를 신속히 처리할 수 있도록 객체-관계형 데이터 웨어하우스에 적절한 인덱스를 고안해야 한다. 따라서 데이터 웨어하우스의 질의 처리시 새로운 인덱스와 요약 정보 및 실제화된 뷰를 최적으로 이용하는 질의 처리 및 최적화 기법이 연구되어야 한다.

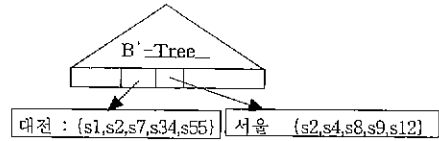
3. 경로 인덱스를 이용한 데이터 웨어하우스의 성능 향상 방안

이 장에서는 객체-관계형 데이터베이스에 기반한 데이터 웨어하우스의 성능 향상을 위한 인덱싱 기법을 제안하고, 이를 이용하여 OLAP 질의를 처리하는 경우에 질의 처리 비용을 수학적 모델링한다. 제안된 인덱싱 기법은 경로 인덱스를 이용하는 것이며, 이를 이용하여 데이터 웨어하우스의 질의 처리 성능을 향상시킬 수 있음을 증명한다.

3.1 경로 인덱스의 구조[13]

경로 인덱스는 객체-관계 스키마 그래프의 경로에 대하여 구축되는 인덱스로서 객체-관계 질의 중에서 경로에 대한 조건을 처리하는데 적합한 인덱스이다. 그림 3은 그림 2의 스키마에서 경로 s.store.city에 구축된 경로 인덱스의 일부를 보여준다. 그림 3에서 보는 바와 같이 경로 s.store.city에 구축된 경로 인덱스는 각 인덱스 엘리먼트의 키 값으로 경로의 끝 노드(여기서는 속성 city 의 도메인인 String 클래스를 의미함)의 값을 가지며, 포인트 필드에는 경로

의 시작 노드(여기서는 sales 클래스를 의미함)의 객체에 대한 OID 리스트를 가진다. 그림 3에서는 대전과 서울의 상점에서 판매된 sales 객체들의 OID는 각각 s1,s2,s7,s34,s55 와 s2,s4,s8,s9,s12 인을 나타낸다



< 그림 3 > 경로 s.store.city에 구축된 경로 인덱스

경로 인덱스를 이용한 질의 처리 과정을 설명하기 위하여 먼저 경로 인덱스의 질의에 포함된 술어에 대하여 다음 용어들을 정의한다.

[정의 1] 경로 인덱스와 술어의 일치 (match)

질의의 where 절에 술어 P path θ value가 존재하고, 경로 path에 대하여 경로 인덱스 PI(path)가 술어 P의 경로 인덱스 PI(path)는 일치한다(match)고 정의한다.

[정의 2] OID 인터섹션(intersection)과 유니온(union)

질의에 포함된 술어 P₁과 P₂에 대하여 경로 인덱스 PI₁과 PI₂가 존재할 때, 인덱스 PI₁를 사용하여 술어 P₁를 만족하는 OID 집합(OIDset-i)을 구하고, 인덱스 PI₂를 사용하여 술어 P₂를 만족하는 OID 집합(OIDset-j)을 구할 수 있다. 이 경우 조건 P₁ AND P₂를 모두 만족하는 OID 집합은 OIDset-i ∩ OIDset-j로 구할 수 있으며 (OID 인터섹션이라고 부름), 조건 P₁ OR P₂를 모두 만족하는 OID 집합은 OIDset-i ∪ OIDset-j로 구할 수 있다 (OID 유니온이라고 부름),

다음은 경로 인덱스를 이용한 OLAP 질의 처리 알고리즘을 소개한다

(i) where 절에서 경로 인덱스와 일치(match)하는 where 절의 조건들을 처리한다.

(ii) 일치하는 술어의 개수가 2개 이상인 경우 술어들을 연결한 연산자 AND 이면 OID intersection을, OR이면 OID union을 시행하여 인덱스로 처리 가능한 술어들을 만족하는 OID 집합을 구한다. ·OID-idx-qualified

(iii) OID-idx-qualified 집합에 속하는 OID들을 데이터베이스로부터 하나씩 가져와서 인덱스로 처리할 수 없는 질의 조건을 처리한다: Object-qualified
(iv) Group by 절이 존재하는 경우에는 group by 절에 포함된 경로들과 인덱스의 일치 여부를 검사한 다음에 일치하는 경로들은 인덱스를 이용하여 처리하고, 나머지 경로들은 데이터베이스의 객체를 직접 접근하여 처리한다.

3.2 성능 분석

경로 인덱스를 사용하는 경우가 그렇지 않은 경우에 비하여 얼마 만큼의 성능이 향상되는지를 수학적 모델링 방법으로 분석한다. 성능 분석을 위하여 일반적인 형태의 다음 질의를 가정하자

[질의문3]

```
SELECT spath1, spath2, , spathn-1 ,
FROM C
WHERE wpath1 θ v1 AND
wpath2 θ v2 AND
... AND
wpathn θ vn
GROUP BY gpath1, gpath2, , , gpathn-2
```

성능 분석은 경로 인덱스를 사용하지 않은 경우와 경로 인덱스를 사용하는 경우로 나누어 분석하며, 각각에 대하여 WHERE 절의 처리 성능과 GROUP BY 절의 처리 성능을 분석한다

(가) 경로 인덱스를 사용하지 않은 경우

(1) WHERE 절의 처리 비용

$$P(C) + n(C) \times sel(C) \times cost(wpath_1) + n(C) \times sel(C) \times sel(wpath_1) \times cost(wpath_2) + n(C) \times sel(C) \times sel(wpath_1) \times sel(wpath_2) \times cost(wpath_3) + \dots$$

$$= P(C) + \sum_{i=1}^n n(C) \times sel(C) \times \prod_{j=1}^{i-1} sel(wpath_j) \times cost(wpath_i) \quad (\text{수식 1})$$

수식 1에서 사용된 기호는 다음과 같은 의미를 가진다

- cost(wpath_k) wpath_k에 포함된 속성의 값의 평균들의 곱
- sel(wpath_k) 조건 wpath_k θ v_k를 만족하는 wpath_k의 시작 노드 키 객체의 비율 (nested selectivity[Cho97])
- sel(C) 클래스 C에 포함된 지역 술어들의 선택률 곱

- $N(C)$ 클래스 C의 객체의 개수
- $P(C)$ 클래스 C의 객체들이 차지 하는 페이지 수

절의분 1에서 절의 대상이 되는 클래스 C는 모두 읽어야 하므로 $P(C)$ 개의 페이지 출력이 일어난다 다음으로 클래스 C의 객체중에서 C의 지역 조건을 만족하는 객체($N(C) \times \text{sel}(C)$ 개입) 각각에 대하여 첫 번째 술어 $wpath_1 \theta v_1$ 를 처리하는 비용은 $\text{cost}(wpath_1)$ 이 되므로 이 비용은 $N(C) \times \text{sel}(C) \times \text{cost}(wpath_1)$ 으로 표시된다 다음으로 첫 번째 술어를 만족하는 C의 객체 v_2 , $N(C) \times \text{sel}(C) \times \text{sel}(wpath_1)$ 개입) 각각에 대하여 두 번째 술어 $wpath_2 \theta v_2$ 를 검사하는 비용도 유사하게 $N(C) \times \text{sel}(C) \times \text{sel}(wpath_1) \times \text{cost}(wpath_2)$ 로 표시된다 유사한 방법으로 n개의 술어에 대한 처리 비용을 모두 합산하면 수식 1을 얻는다.

(2) GROUP BY 절의 처리 비용

인덱스가 없는 경우에 group by의 처리 비용은 다음 수식 2와 같이 모델링된다.

$$\begin{aligned} & n(C') \times \text{cost}(gpath_1) + P(C') \log_2 P(C') + \\ & n(G_1) \times P(G_1) \log_2 P(G_1) + \text{cost}(gpath_2) + \\ & n(G_2) \times P(G_2) \log_2 P(G_2) + \text{cost}(gpath_3) + \\ & \dots \\ & = n(C') \times \text{cost}(gpath_1) + P(C') \log_2 P(C') + \\ & \sum_{i=1}^{n-1} n(G_i) \times \{P(G_i) \log_2 P(G_i) + \text{cost}(gpath_{i+1})\} \text{ (수식 2)} \end{aligned}$$

수식 2에서 C' 은 클래스 C의 객체들 중에 where 절의 조건을 모두 만족하는 객체들의 집합이다. 수식의 앞부분인 $N(C') \times \text{cost}(gpath_1)$ 은 정렬의 기준이 되는 $gpath_1$ 의 마지막 속성값을 확인하는데 필요한 페이지 출력 회수이다. 다음에 나오는 $P(C') \log_2 P(C')$ 은 C' 의 객체들 $gpath_1$ 의 마지막 속성값을 기준으로 그룹핑(정렬)하는 비용이다. 마지막 항은 group by 절에 두 개 이상의 경로가 나오는 경우에 필요한 비용식이다. 여기서 $n-2$ 는 group by 절에 포함된 경로의 개수이고, $N(G_i)$ 는 $gpath_i$ 까지 그룹핑한 결과 생성되는 그룹의 개수이며, $P(G_i)$ 는 $gpath_i$ 까지 그룹핑한 결과 생성되는 그룹들의 평균 페이지 개수이다 이 경우 $gpath_i$ 까지 그룹핑한 결과 생성되는 각 그룹을 $gpath_{i+1}$ 의 값을 기준으로 정렬해야 하므로 정렬 자체의 비용 $N(G_i) \times P(G_i) \log_2 P(G_i)$ 가 되고, 정렬의 기준이 되는 속성($gpath_{i+1}$)값을 액세스하는 비용은 $\text{cost}(gpath_{i+1})$ 이 되므로 이들의 합으로 표현된다 또한, group by 절에 3 개 이상의 경로가 나와도 두 개의 경로가 나타난 경우와 동일한 방법으로 비용을 예측하여 더하면 되므로 전체 비용은 수식 2와 같이 표현된다.

(나) 경로 인덱스를 사용하는 경우

(1) 검색 비용

k ($\leq n$) 개의 where 절의 술어가 인덱스의 일치한다고 가정하자 이 경우에 where 절의 처리 비용은 다음과 같다

$$\sum_{i=1}^k H_i + \sum_{i=1}^{n-1} n(C^i) \times \text{sel}(C) \times \prod_{k=1}^{i-1} \text{sel}(wpath_k) \times \text{cost}(wpath_i) \text{ (수식 3)}$$

수식 3에서 C^i 는 C의 객체중에서 인덱스로 처리 가능한 술어들을 만족하는 객체집합이다. 수식3 에서 $k = n$ 이면 (즉, where 절의 모든 술어들을 인덱스로 처리할 수 있으면) n 개의 인덱스만을 액세스하여 절의 where 절을 모두 만족하는 객체의 OID를 얻을 수 있다 따라서 이 경우 where 절의 처리 비용은 이들 인덱스의 불이 합(H, 불이 합)이 된다. 단일, 하나의 술어라도 인덱스로 처리할 수 없는 것이 존재하면 수식 1의 원리와 유사하게 인덱스로 처리할 수 없는 술어들의 처리 비용을 더한다.

(2) GROUP BY 절의 처리 비용

group by 절에 나타낸 n2 개의 경로중에서 k ($\leq n2$) 개의 경로에 인덱스기 구축되었다고 가정하자. 이 경우 k 개의 인덱스의 리프 노드를 읽으면 k 회 승렬된 그룹을 만들 수 있으며, 나머지 $n2-k$ 개의 경로식은 수식 2의 유사하게 인덱스를 액세스하지 않고 정렬의 기준이 되는 경로를 읽어서 그룹핑하도록 한다 따라서 비용은 다음과 같이 두 부분으로 표현된다.

$$\sum_{i=1}^k LP(I_i) + \sum_{j=k+1}^{n2} n(G_j) \times \{P(G_j) \log_2 P(G_j) + \text{cost}(gpath_j)\} \text{ (수식 4)}$$

수식 4에서 $LP(I_i)$ 는 인덱스 I_i 의 리프노드의 페이지 개수이고, 나머지 항들은 수식 1 ~ 수식 3에서 설명되었다 두 번째 항에서 $n(G_j)$ 는 j 개의 인덱스를 사용하여 그룹핑한 결과에서 생성되는 그룹의 개수를 나타내며, $P(G_j)$ 는 이 그룹의 평균 페이지 개수를 나타낸다

(다) 성능의 비교

where 절을 처리될 때 경로 인덱스를 사용하지 않은 경우 (수식 1)과 인덱스를 사용하는 경우(수식 3)을 비교하면 술어의 처리에 사용될 수 있는 인덱스

의 개수가 많을수록(수식 3에서 k가 n에 가까울수록), 클래스 C의 객체수 $n(C)$ 가 많을수록, 클래스 C의 지역 술어들의 선택률 곱인 $\text{sel}(C)$ 가 클수록 인덱스를 사용하는 수식 3의 비용은 인덱스를 사용하지 않은 수식 1의 비용보다 급격히 적어짐을 알 수 있다. 실제로 데이터 웨어하우스에서 사실 테이블의 크기가 전체 데이터 웨어하우스 크기의 대부분을 차지할 정도로 $n(C)$ 는 크다. 또한, 사실 테이블에 대한 조건은 거의 대부분의 OLAP 질의에서 주어지지 않으므로 $\text{sel}(C)$ 도 1에 가까운 정도로 크다. 따라서 수식 $n(C)$ 와 $\text{sel}(C)$ 가 수직되는 수식 1의 비용은 사용되는 인덱스의 개수(수식 3에서 k)가 클수록 수식 3의 비용보다 커짐을 알 수 있다 극단적인 경우로써, 수식 3의 k가 n과 동일하면 (즉, 모든 where 절의 술어들을 인덱스로 처리한다면) 수식 3의 경우 $n \times$ (인덱스 높이 ≈ 3 혹은 4) 개의 페이지를 읽어서 where 절을 처리할 수 있으나 수식 1의 경우 C의 페이지 수 $P(C)$ 와 C의 객체수 $n(C)$ 의 값을 n 회 누적하므로 비용이 커진다 특히, 간단한 경우 클래스 C의 객체수나 페이지 수에 무관하게 비용이 표시되므로 사실 테이블의 크기와 무관한 반면, 후자의 경우 사실 테이블의 크기에 의존하는 형태가 된다

한편, group by 절의 처리 비용인 수식 2와 수식 4는 상황에 따라서 비용의 크기 순서가 달라진다 즉, where 절의 조건을 만족하는 객체 집합 C' 의 원소수 $n(C')$ 가 작은 경우에는 수식 2의 첫 번째 항의 크기보다 수식 4의 인덱스 리프노드의 크기합보다 작게될 수 있기 때문이다 극단적으로 C' 에 속하는 객체가 하나뿐이라면 수식 1의 경우 $n(C')=1$, $n(G)=1$, $P(C')=1$, $P(G)=1$ 이 되므로 $\text{cost}(gpath_1)$, $i=1, 2$, $n2$ 를 합한 것이 된다. 그러나, 수식 4의 경우 일치하는 k개의 인덱스의 리프 노드 페이지를 모두 합한 값이 되므로 수식 2보다 클 가능성이 높게 된다.

4. 결론 및 향후 연구

본 연구에서는 객체-핀계 DBMS기반의 데이터 웨어하우스의 질의 처리 성능을 향상시키기 위한 인덱싱 기법을 제시하고, 질의 처리시 제안된 인덱싱 기법을 사용함으로써 성능 향상의 정도를 수학적인 모델링 기법을 사용하여 분석하였다 제안된 인덱싱 기법은 고비용으로 알려진 where 절에 포함된 조건과 group by 절에 포함된 경로들을 저비용인 인덱스 액세스를 키리할 수 있으므로 절의 처리의 성능이 개선된다 특히, 데이터 웨어하우스는 사실 클래스나 차원 클래스에 대한 삭제나 변경 연산을 거의 발생하지 하지 않고 사실 클래스에 대한 삽입 연산인 주로 발생하므로 객체 지향 데이터베이스에기와는 달리 경로 인덱스의 유지 비용도 높지 않다

향후 연구로서 수식 1~수식 4에 대한 시뮬레이션을 통하여 제안된 인덱싱 기법으로 인한 질의 처리 성능의 향상 정도를 정확히 분석하는 작업과 경로 인덱스 자체를 확장하여 계속적으로 구성함으로써 데이터 웨어하우스 질의의 구체화(drill-down) 혹은 추상화(drill-up) 연산을 지원하도록 하는 연구를 진행하고자 한다.

[참고 문헌]

- [1] Lyman Do et al., "Issues in Developing Very Large Data Warehouses," In Proc. Int'l Conf. on VLDB, 1998
- [2] Wilburt Labio et al., Jennifer Widom, "The WHIPS Prototype for Data Warehouse Creation and Maintenance," In Proc. Int'l Conf. ACM SIGMOD, 1997
- [3] Wu, M-C and Buchmann, A.P., "Research Issues in Data Warehousing," Submitted for publication.
- [4] Wu, M C and Buchmann, Alejandro P., "Encoded Bitmap Indexing for Data Warehouses," In Proc Int'l Conf ICDE, 1993
- [5] Imnon, W H, Building the Data Warehouse John Wiley, 1992.
- [6] Surajit Chaudhuri and Umeshwar Dayal, "Data Warehousing and OLAP for Decision Support," In Proc. Intl. Conf. DOOD, 1997
- [7] George Colliat, "OLAP, Relational, and Multidimensional Database System," In ACM SIGMOD Record, 25(3), 1996.
- [8] Yihong Zhao et al., "Array-Based Evaluation of Multi-Dimensional Queries in Object-Relational Databases System," In Proc Int'l Conf ICDE, 1998
- [9] Prasad Deshpande et al., "Cubing Algorithms, Storage Estimation, and Storage and Processing Alternatives for OLAP," IEEE Data Engineering Bulletin 20(1): 3-11, 1997
- [10] Kimball, R The Data Warehouse Toolkit John Wiley, 1996
- [11] OLAP Council White Paper, <http://www.olapcouncil.org>
- [12] Stanford Technology Group, "Designing the Data Warehouse On Relational Database," White Paper
- [13] Bertino, E and Kim, W "An indexing technique for query on nested object," IEEE Trans on Knowledge and Data Engineering, Vol, No2(1989), 196-214
- [14] 이정민, 조완섭, 이충재, "ORDBMS를 이용한 Internet기반 데이터 웨어하우스 구축," 한국통신학회 추계학술발표대회, 1999