실체 뷰와 차원 계층을 이용한 OLAP 질의 재작성 방법

(A Method Rewriting OLAP Queries using Materialized Views and Dimension Hierarchies)

박 창 섭 † 김 명 호 †† 이 윤 준 ††
(Chang-Sup Park) (Myoung Ho Kim) (Yoon-Joon Lee)

요 약 데이타 웨어하우스 시스템에 대한 OLAP 질의들은 대량의 데이타를 대상으로 복잡한 분석 및 집계 연산을 수행한다. 이러한 고비용의 OLAP 질의들을 효율적으로 실행하는 것은 시스템의 성능 향상을 위해 매우 중요하다. 이를 위해 본 논문에서는 데이타 웨어하우스 시스템에 존재하는 여러 종류의 실체 집계 뷰들을 이용하여 주어진 OLAP 질의를 재작성하는 방법을 제안한다. 본 논문에서는 차원 계층들로부터 유도되는 그룹 격자를 이용하여 OLAP 질의와 실체 뷰의 선택 단위, 선택 영역, 집계 단위 등을 정의하고, 이들로부터 OLAP 질의와 실체 뷰에 대한 정규형을 정의한다. 그리고 정규형으로 표현된 질의와 실체 뷰 사이의 관계를 이용하여 실체 뷰가 질의의 재작성에 이용 가능하기 위한 조건을 제시한다. 제안하는 질의 재작성 방법은 데이타 웨어하우스의 메타 정보들과 OLAP 질의 및 실체 뷰들의 특성을 고려하여 다양한 실체 뷰들을 효과적으로 활용한다. 특히 서로 다른 집계 단위와 선택 단위 및 선택 영역을 가진 실체 뷰들을 함께 이용할 수 있으므로, 시스템에 존재하는 실체 뷰들의 효용성을 높이고 주어진 질의를 효율적으로 처리할 수 있다.

Abstract OLAP queries involve complex analyses and aggregations on a large amount of data in data warehouses. To process these expensive queries efficiently, we propose a new method to rewrite a given OLAP query using various classes of materialized aggregate views which already exist in data warehouses. We consider selection granularities, selection regions, and aggregation granularities of OLAP queries and materialized views, which are derived from the grouping lattice of dimension hierarchies, and define normal forms of OLAP queries and materialized views using them. We present conditions for usability of a materialized view in rewriting an OLAP query, which are defined by relationship between the query and materialized view in normal forms. Considering meta-information of data warehouses and the characteristics of OLAP queries, our query rewriting method exploits materialized views effectively. Materialized views with different selection granularities, selection regions, and aggregation granularities can be used together in rewriting a given query to produce a much efficient rewritten query.

1. 서 론

최근 데이타 웨어하우스(Data Warehouse: DW)를 기반으로 한 의사 결정 시스템이 널리 사용되고 있으며, 온라인 분석 처리(On-Line Analytical Processing: OLAP)에 관한 연구가 활발히 진행되고 있다. OLAP에서 사용되는 질의들은 기존의 OLTP 응용에서 실행되는 질의들보다 훨씬 더 복잡하며 여러 가지 다른 특성들을 가진다. OLAP 질의의 대상이 되는 DW는 일반적으로 크기가 매우 크다. 의사 결정 시스템의 사용자는 세부적인 데이타들에 대해 질의하기보다는 대량의 데이타들 속에서 어떤 경향을 찾거나 비즈니스 프로세스에

† 비 회 원 : 한국과학기술원 전산학과
parkcs@dbserver.kaist.ac.kr
†† 종신회원 : 한국과학기술원 전산학과 교수
mhkim@dbserver.kaist.ac.kr
yjlee@dbserver.kaist.ac.kr
논문접수 : 2000년 6월 29일
심사완료 : 2001년 3월 15일

대한 분석을 수행한다. 이 때 이용되는 OLAP 질의들은 주로 기초 데이타들에 대한 다차원적인 그룹화(grouping) 및 집계(aggregation) 연산으로 구성된다. 이러한 대량의 데이타와 질의의 복잡성은 OLAP 질의의 비용을 크게 증가시키며, 이는 의사 결정 시스템의 효율과 생산성에 큰 영향을 미친다. 따라서 고비용의 OLAP 질의들은 가능한 한 효율적으로 실행되어야 한다.

관계형 데이타베이스를 기반으로 하는 DW 시스템에서 OLAP 질의를 효율적으로 처리하기 위해 자주 발생되는 OLAP 질의의 결과를 미리 계산하여 요약 테이블, 즉, 실체 뷰(Materialized View)로 저장해 두고 이를 이용해서 주어진 질의를 처리하는 방법에 관한 연구들이 제안되었다. 실체 뷰는 기초 데이타들에 대해 그룹화 및 집계 연산을 수행한 결과이므로 일반적으로 사실 테이블에 비해 크기가 작다. 이러한 실체 뷰를 효과적으로 이용하면 기초 데이타에 대한 OLAP 질의들을 효율적으로 실행할 수 있다. 이를 위해서는 주어진 질의를 실체 뷰들을 이용하여 재작성하는 방법이 필요하며, 이와 관련된 연구로는 [1,2,3,4,5,6,7] 등이 있다. 그러나 기존의 연구들은 OLAP가 아닌 일반적인 데이타베이스 시스템의 질의를 대상으로 하거나 DW와 OLAP 질의의 여러 가지 특성들을 충분히 활용하지 못하기 때문에 생성할 수 있는 재작성 질의의 종류가 제한적이다.

본 논문에서는 기존 연구들의 단점을 크게 개선한 OLAP 질의 재작성 방법을 제안한다.

본 논문에서 고려하는 OLAP 질의는 각 차원의 차원 계층에 속한 애트리뷰트들의 조합에 대해 선택, 그룹화 및 집계 연산을 실행하는 질의로서, OLAP 시스템에서 많이 사용되는 롤-업(roll-up), 드릴-다운(drill-down), 슬라이스&다이스(slice&dice) 질의 등을 포함한다. 본 논문에서는 DW 스키마의 의미 정보를 이용하여 대상이 되는 OLAP 질의와 실체 뷰에 대한 정규형을 정의한다. 정규형으로 표현된 질의와 실체 뷰들 사이의 관계를 이용하여 질의 재작성에 이용 가능한 실체 뷰의 조건을 제시한다. 그리고 서로 다른 종류와 형태의 실체 뷰들을 함께 이용하는 질의 재작성 방법을 제안한다.

제안하는 재작성 방법의 장점은 다음과 같다.

- 임의의 영역에 대해 정의된 집계 실체 뷰들을 이용한다. 즉, 각 차원 애트리뷰트 값의 임의의 범위에 대한 그룹화 및 집계의 결과들을 재작성에 이용할 수 있다. 기존의 실체 뷰 선택 방법들은 각 차원 애트리뷰트의 모든 범위의 값이나 또는 특정 값에 대한 집계 결과만을 고려함으로써 이용 가능한 실체 뷰가 제한적이다.
- DW 스키마의 의미(semantic) 정보와 메타 데이타들을 이용한다. 특히, DW 스키마에 포함된 차원 계층 정보들은 OLAP 질의 및 실체 뷰의 정의에 많이 사용된다. 이러한 정보들을 이용함으로써 다양한 실체 뷰들을 재작성에 이용할 수 있다.
- 실체 뷰들을 이용한 여러 가지 재작성 방법들 중 효율적으로 실행될 수 있는 것을 생성한다. 일반적으로, 주어진 OLAP 질의에 대해 실체 뷰들을 이용하여 재작성하는 방법은 다수이며, 이들의 실행 비용은 서로 다르다. 따라서, 이들 중 효율적으로 실행될 수 있는 재작성 질의를 생성하는 것이 바람직하다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구의 동기에 대해 설명하며, 3장에서는 본 연구의 대상이 되는 DW, OLAP 질의, 그리고 실체 뷰 등의 개념들을 정의한다. 4장에서는 실체 뷰를 이용한, OLAP 질의의 재작성 방법을 제안한다. 5장에서는 본 논문과 관련된 기존 연구들을 소개하고 제안한 방법과 비교한다. 마지막으로 6장에서 결론을 맺는다.

2. 동기 및 예

본 장에서는 예를 통해 본 논문의 연구 동기와 필요성에 대해 설명한다.

예 1: 어떤 체인점의 판매 데이타에 대해 그림 1과 같은 DW 스키마를 생각하자. 이 DW는 하나의 사실(fact) 테이블과 4개의 차원(dimension) 테이블들로 구성되며, 각 차원마다 하나의 차원 계층이 존재한다. 그리고, 그림 2와 같이 세 개의 실체 뷰들이 있다고 가정하자.

이 DW에 대해 그림 3의 OLAP 질의 Q_1을 고려하자. 이 질의는 사실 테이블 대신 실체 뷰 MV_1, MV_2, MV_3를 이용하여 그림 3의 Q_1'과 같이 재작성 될 수 있다.

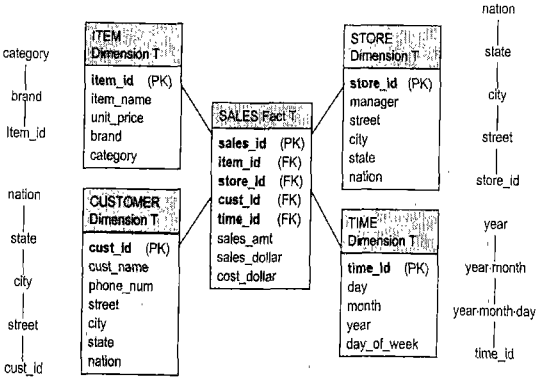


그림 1 예제 데이타 웨어하우스 스키마

MV_1: 1997년부터 현재까지 주별, 년도별 매출액의 합

```
SELECT  state, year, SUM(sales_dollar)
        AS sum_dollar1
FROM    Sales, Store, Time
WHERE   Sales.store_id = Store.store_id
        AND Sales.time_id = Time.time_id
        AND Time.year ≥ 1997
GROUP BY state, year
```

MV_2: 미국 내의 상점들의 주별, 월별 매출액의 합

```
SELECT  state, year, month, SUM(sales_dollar)
        AS sum_dollar2
FROM    Sales, Store, Time
WHERE   Sales.store_id = Store.store_id
        AND Sales.time_id = Time.time_id
        AND Store.nation = 'USA'
GROUP BY state, year, month
```

MV_3 : 캐나다 내의 상점들의 1996년까지의 도시별, 월별 매출액의 합

```
SELECT    city, year, month, SUM(sales_dollar) AS sum_dollar3
FROM      Sales, Store, Time
WHERE     Sales.store_id = Store.store_id AND Sales.time_id = Time.time_id
          AND Store.nation = 'CANADA' AND Time.year ≤ 1996
GROUP BY  city, year, month
```

그림 2 예제 실체 뷰

Q_1: 미국이나 캐나다에 위치한 상점들의 1996년 부터 1999년까지의 주별, 연도별 매출액의 합

```
SELECT  state, year, SUM(sales_dollar)
FROM    Sales, Store, Time
WHERE   Sales.store_id = Store.store_id
        AND Sales.time_id = Time.time_id
        AND (Store.nation = 'USA' OR
             Store.nation = 'CANADA' )
        AND Time.year ≥ 1996
        AND Time.year ≤ 1999
GROUP BY state, year
```

Q_1':

```
(SELECT       state, year, sum_dollar1
FROM    MV1, (SELECT DISTINCT state
              FROM   Store
              WHERE  nation = 'USA' OR
                     nation = 'CANADA') S
WHERE         MV1.state = S.state
        AND MV1.year ≤ 1999)
UNION
(SELECT state, year, SUM(sum_dollar2)
FROM    MV2
WHERE   MV2.year = 1996
GROUP BY state, year)
UNION
(SELECT       S.state, year, SUM(sum_dollar3)
FROM    MV3, (SELECT DISTINCT city, state
              FROM   Store) S
WHERE         MV3.city = S.city
        AND MV3.year = 1996
GROUP BY S.state, year)
```

그림 3 예제 질의 Q_1과 재작성된 질의 Q_1'

재작성된 질의 Q_1'에는 각 실체 뷰를 대상으로 하는 세 개의 부질의(subquery) 블록이 존재한다. 각 부질의 블록들은 Q_1의 집계 그룹(state, year)들 중 일부분을계산한다(그림 4 참조). 즉, MV_1을 이용하여 1997년부터 1999년까지 미국과 캐나다의 판매 데이타들에 대한 주별 연도별 집계를 수행한다.

MV_2로부터는 1996년 미국 내의 판매 데이타들에 대해, 그리고 MV_3로부터는 1996년 캐나다의 판매 데이타들에 대해 주별 연도별 집계를 수행한다. 각 실체 뷰로부터 계산되는 Q_1의 집계 그룹들의 영역이 서로 분리되고, 그 영역들의 합집합이 Q_1에서 구하고자 하는 집계 그룹들의 영역과 일치하므로, Q_1'과 같이 세 개의 부질의 결과들을 UNION하면 Q_1의 결과와 동일한 결과를 얻을 수 있다. □

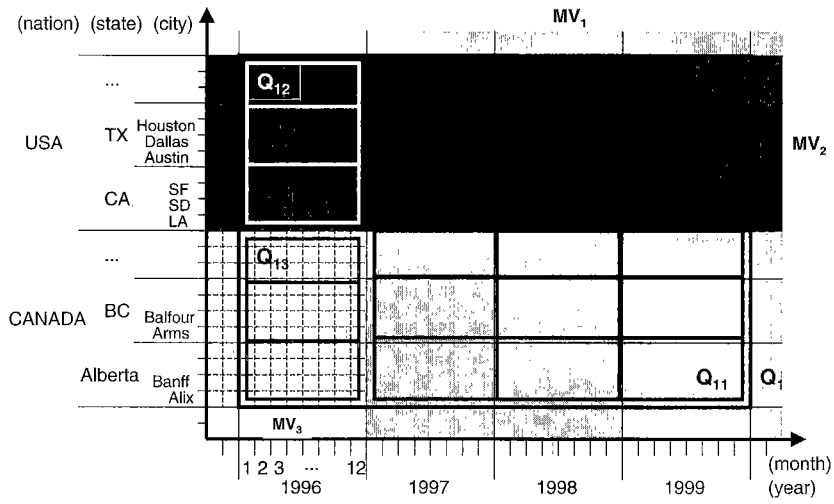


그림 4 Q_1'의 각 부질의 블록의 영역

Q_2 : 캐나다에 위치한 상점들의 주별 매출액의 총합

```
SELECT  state, SUM(sales_dollar)
FROM    Sales, Store
WHERE   Sales.store_id = Store.store_id
        AND Store.nation = 'CANADA'
GROUP BY  state
```

Q_2':

```
SELECT state, SUM(psum)
FROM (SELECT state, SUM(sum_dollar1) AS psum
      FROM   MV1,
             (SELECT DISTINCT state
              FROM  Store
              WHERE nation = 'CANADA') S
      WHERE  MV1.state = S.state
      GROUP BY state
      UNION ALL
      SELECT S.state, SUM(sum_dollar3) AS psum
      FROM   MV3,
             (SELECT DISTINCT city, state
              FROM  Store) S
      WHERE  MV3.city = S.city
      GROUP BY S.state)
GROUP BY state
```

그림 5 예제 질의 Q_2와 재작성된 질의 Q_2'

다음 예는 다른 형태의 질의 재작성 방법을 보인다.

예 2: 그림 1의 판매 DW에 대해 그림 5에 있는 질의 Q_2를 고려하자. 이 질의는 사실 테이블 대신 MV_1과 MV_3를 이용하여 그림 5의 Q_2'과 같이 재작성 될 수 있다.

Q_2는 캐나다의 상점들의 모든 연도에 대한 판매 데이타를 집계한 결과를 원한다. 그러나 MV_1은 1997년부터 현재까지의, MV_3는 1996년까지의 판매 데이타에 대한 집계 결과이다.

즉, MV_1이나 MV_3는 Q_2의 각 집계 그룹(state)의 영역 ('CANADA', 모든 연도)의 일부분에 대한 결과만을 갖고 있으므로 Q_2의 각 집계 그룹을 계산할 수 없다. 그러나 각 실체 뷰로부터 계산되는 집계 영역이 서로 완전히 분리되어 있고 두 영역의 합집합이 Q_2의 집계 영역과 일치하므로, 두 실체 뷰를 함께 이용하면 Q_2의 결과를 구할 수 있다(그림 6 참조). Q_2'과 같이 먼저 각

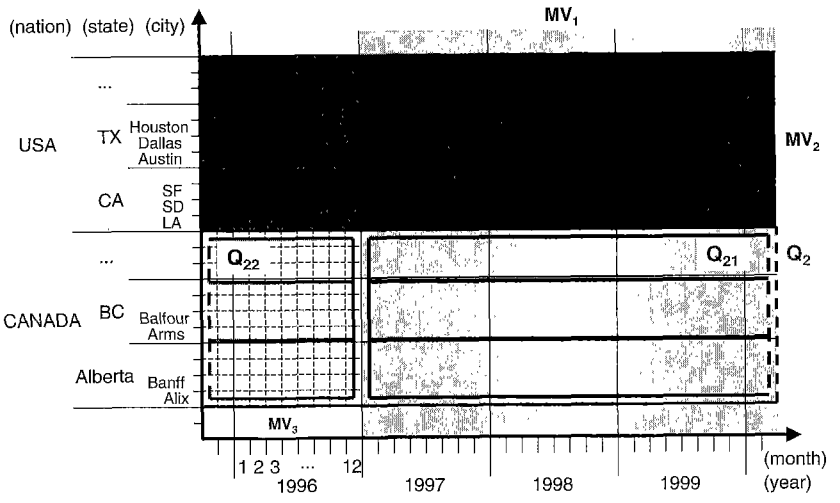


그림 6 Q_2'의 각 부질의 블록의 영역

실체 뷰로부터 부영역(subrange)에 대해 Q_2의 모든 집계 그룹들을 계산한다.

각 부질의의 결과는 같은 집계 그룹들에 대한 부분 집계 결과들이므로, 두 결과를 UNION ALL로 합한 후, 다시 그룹화하고 집계하면 Q_2와 동일한 결과를 얻을 수 있다. □

기존의 연구들에서는 앞의 예와 같은 질의 재작성을 수행하지 못한다. 따라서 시스템에 존재하는, 여러 가지 차원 애트리뷰트들의 조합에 대한 집계 실체 뷰들을 이용하여 효과적으로 주어진 질의를 처리할 수 없다. 본 논문에서는 기존의 방법들과는 달리 DW 스키마의 메타 정보와 OLAP 질의 및 실체 뷰의 특징을 고려하여 앞의 두 예와 같이 서로 다른 집계 및 선택 단위와 선택 영역을 가진 실체 뷰들을 함께 이용하는 질의 재작성 방법을 제안한다.

3. 개념 및 정의

3.1 데이타 웨어하우스 스키마와 그룹 격자

본 논문에서 고려하는 데이타 웨어하우스는 그림 1과 같이 하나의 사실 테이블과 여러 개의 차원 테이블들로 구성되는 스타(star) 스키마 구조를 갖는다. 사실 테이블에 저장된 튜플들을 기초 데이타라 부른다. 각 차원 테이블에는 하나의 차원 계층이 존재한다고 가정한다.

정의 1: 데이타 웨어하우스의 스키마를 구성하는 구성 요소들은 다음과 같이 정의된다.

1. 사실 테이블 $FT = (k, f_1, f_2, ..., f_d, m_1, m_2, ..., m_n)$: k는 주 키 애트리뷰트이고, $f_i\ (1 \le i \le d)$는 차원 테이블 DT_i에 대한 외래 키(foreign key)이다. $m_j\ (1 \le j \le n)$는 집계 연산의 대상이 되는 측정(measure) 애트리뷰트들이다.

2. 차원 테이블 $DT_i = (a_0^i, a_1^i, ..., a_{d_i}^i, b_1^i, b_2^i, ..., b_{n_i}^i)$, $1 \le i \le d$: 차원 테이블의 애트리뷰트들은 두 종류로 구분된다. $a_j^i (1 \le j \le d_i)$는 차원 계층에 포함되는 차원 애트리뷰트들이고, $b_j^i\ (1 \le j \le n_i)$는 그렇지 않은 비-차원 애트리뷰트들이다.

3. 차원 계층 $DH_i = (L_0^i, L_1^i, ..., L_{h_i}^i, NONE), 1 \le i \le d$: 차원 계층은 레벨(level) L_j^i 들의 순서화된 집합이다. L_j^i $(0 \le j \le h_i)$는 차원 테이블 DT_i에 포함된 차원 애트리뷰트들의 부분 집합이다. j를 L_j^i의 높이(height)라고 부른다. L_0^i는 DH_i의 최하위 레벨로서, DT_i의 주 키 애트리뷰트를 포함한다. NONE은 DH_i의 최상위 레벨을 나타내는 기호로서, 공집합이다. 두 차원 레벨 L_{j-1}^i과 L_j^i사이에는 함수적 종속 $L_{j-1}^i \le L_j^i (1 \le j \le h_i)$가 존재한다. □

차원 계층의 각 차원 레벨은 그 차원에 대해 기초 데이타들이 그룹화될 수 있는 기준이 된다. 모든 차원 계층들의 카티전 프로덕트(Cartesian product) DH는 각 차원 계층에 속한 차원 레벨들의 순서화된 집합들로 이루어진, 부분적으로 순서화된 집합이다. 이 집합의 원소들 사이에는 다음과 같은 부분 순서 관계(partial ordering relation)가 존재한다.

정의 2: 집합 DH에 속하는 두 원소들 사이의 부분 순서 관계는 다음과 같이 정의된다.

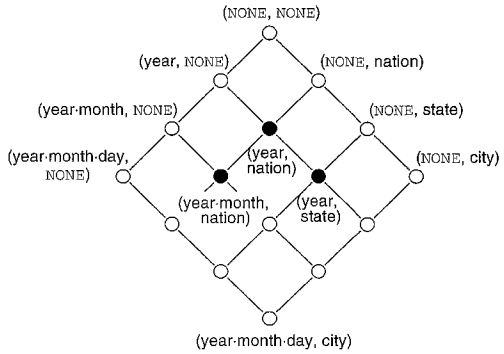


그림 7 Store 차원 계층과 Time 차원 계층에 의해 생성되는 그룹 격자의 일부

(a) $(L^1_{l_1}, L^2_{l_2}, ..., L^d_{l_d}) \le (L^1_{m_1}, L^2_{m_2}, ..., L^d_{m_d})$ 이면, 모든 i $(1 \le i \le d)$에 대해 $L^i_{l_i} \rightarrow L^i_{m_i}$ 또는 $L^i_{l_i} = L^i_{m_i}$ 이다.

(b) $(L^1_{l_1}, L^2_{l_2}, ..., L^d_{l_d}) < (L^1_{m_1}, L^2_{m_2}, ..., L^d_{m_d})$ 이면,

$(L^1_{l_1}, L^2_{l_2}, ..., L^d_{l_d}) \le (L^1_{m_1}, L^2_{m_2}, ..., L^d_{m_d})$ 이고 $L^j_{l_j} \neq L^j_{m_j}$ 인 j가 존재한다. 또, 그 역도 성립한다.

(c) $(L^1_{l_1}, L^2_{l_2}, ..., L^d_{l_d}) <> (L^1_{m_1}, L^2_{m_2}, ..., L^d_{m_d})$ 이면,

$(L^1_{l_1}, L^2_{l_2}, ..., L^d_{l_d}) \le (L^1_{m_1}, L^2_{m_2}, ..., L^d_{m_d})$ 이고

$(L^1_{l_1}, L^2_{l_2}, ..., L^d_{l_d}) \ge (L^1_{m_1}, L^2_{m_2}, ..., L^d_{m_d})$ 이다. 또, 그 역도 성립한다. □

이 집합은 하나의 격자(lattice)로 표현될 수 있으며, 이를 그룹 격자라 부른다. 그림 7은 예 1의 Store와 Time 차원의 두 차원 계층들로부터 정의되는 그룹 격자의 일부를 나타낸다.

3.2 OLAP 질의 및 실체 뷰

본 논문에서 재작성의 대상으로 고려하는 OLAP 질의는 중첩(nesting)이 없는 단일 블록 집계 질의로서, 선택 애트리뷰트 집합, 선택 술어, 그룹화 애트리뷰트 집합, 프로젝션 애트리뷰트 집합, 집계 함수 등으로 기술된다.

선택 애트리뷰트 집합은 선택 술어에 나타나는 애트리뷰트들의 집합으로서, 각 차원마다 어느 한 차원 레벨에 속하는 차원 애트리뷰트들로 이루어 진다[1]. 따라서 선택 애트리뷰트 집합은 그룹 격자상의 한 노드, 즉, 각 차원의 특정 차원 레벨들의 순서화된 집합에 대응될 수 있으며, 이를 질의의 선택 단위(Selection Granularity: SG)라 부른다. 선택 술어는 선택 애트리뷰트들에 대한 비교 술어들의 부울 결합(Boolean combination)이다. 비교 술어는 a op c의 형태로서, a는 차원 애트리뷰트, c는 상수, 그리고 op∈{<,≤,=,≥,>}이다. 선택 술어는 논리합 정규형(disjunctive normal form)으로 변환될 수 있다. 이 때, 각 논리곱 요소(conjunct)는 기하학적으로, 선택 단위로부터 형성되는 다차원 공간상의 하나의 하이퍼-사각형(hyper-rectangle)으로 표현될 수 있다. 즉, 선택 술어는 이러한 하이퍼-사각형들의 집합으로 표현될 수 있으며, 이 집합을 질의의 선택 영역이라 부른다. 그룹화 애트리뷰트 집합은 집계의 기준이 되는 애트리뷰트들의 집합으로서, 각 차원마다 어느 한 차원 레벨에 속하는 차원 애트리뷰트들로 이루어 진다. 선택 애트리뷰트 집합과 같이, 그룹 격자의 한 노드에 대응될 수 있으며, 이를 질의의 집계 단위(Aggregation Granularity: AG)라 부른다. 한편, 프로젝션 애트리뷰트 집합은 질의의 결과로 선택되는 애트리뷰트들의 집합으로서, 그룹화 애트리뷰트 집합과 동일하다.

이러한 요소들을 이용하여 고려하는 OLAP 질의의 정규형(normal form)을 정의한다.

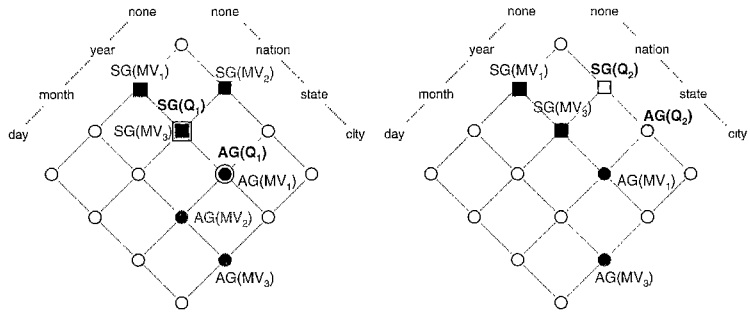
정의 3: OLAP 질의 Q의 정규형은 다음과 같이 정의된다.

$$Q(SG, R, AG, AGG, HAV)$$

- $SG = (S_1, S_2, ..., S_d)$는 질의의 선택 단위로서, S_i $(1 \le i \le d)$는 차원 테이블 DT_i의 차원 계층 DH_i의 차원 레벨이다.
- $R = \{R_i\}$는 질의의 선택 영역으로서, 하이퍼-사각형들의 집합이다. 각 하이퍼-사각형 $R_i = (r_{i1}, r_{i2}, ..., r_{id})$는 선택 술어에 포함된 차원 레벨 값의 구간(interval)들의 순서화된 집합이다. 즉, r_{ij}는 차원 테이블 DT_j내의 특정 차원 레벨 값의 구간을 나타낸다. 구간은 개구간(open interval), 폐구간(closed interval), 또는 반폐구간(half-closed interval)이 될 수 있다. 구간의 한계 값이 없는 경우에는 $-\infty$, $+\infty$를 이용하여 표기한다.
- $AG = (A_1, A_2, ..., A_d)$는 질의의 집계 단위로서, A_i $(1 \le i \le d)$는 차원 테이블 DT_i의 차원 계층 DH_i의 차원 레벨이다.
- AGG = { agg(m) | agg∈{MIN, MAX, SUM, COUNT}, m은 측정 애트리뷰트 }[2].
- HAV는 AG에 포함된 애트리뷰트나 AGG에 포함

1) 본 논문에서 제안하는 방법은 선택 애트리뷰트 집합이 차원 계층의 서로 다른 차원 레벨들에 속하는 애트리뷰트들을 포함하는 경우에 대해 확장 가능하다.

2) AVG는 SUM과 COUNT로부터 계산될 수 있으므로, 기술의 편의상 본 논문에서는 고려하지 않는다.



(a) MV_1, MV_2, MV_3, Q_1 (b) MV_1, MV_3, Q_2

그림 8 질의와 실체 뷰들의 선택 단위와 집계 단위

된 집계 함수들에 대한 비교 술어들의 논리식으로서, SQL로 작성된 질의에서 HAVING 절의 조건식을 의미한다. 이러한 조건식이 없는 경우에는 NULL로 표기한다. □

*SG(Q), R(Q), AG(Q), AGG(Q), HAV(Q)*는 질의 *Q*의 *SG, R, AG, AGG, HAV*를 의미하고, *AG(Q, i)*와 *SG(Q, i)* ($1 \le i \le d$)는 각각 *AG(Q)*와 *SG(Q)*에 포함된 DT_i의 차원 레벨을 가리킨다.

일반적으로 OLAP에서 주로 사용되는 질의들은 드릴-다운, 롤-업, 슬라이스&다이스 등이 있다[8]. 이러한 OLAP 질의들은 정의 3에서 정의한 정규형으로 표현될 수 있으며, 선택 단위와 집계 단위 사이의 관계에 따라 분류될 수 있다. 즉, 드릴-다운 질의는 *SG(Q)>AG(Q)*를 만족하고, 롤-업 질의는 *SG(Q)<AG(Q)*를 만족한다. 슬라이스&다이스 질의의 경우는 선택 술어가 *SG(Q)*상의 한 점에 해당되므로 선택 애트리뷰트 집합을 집계 애트리뷰트 집합에 포함시키면 *SG(Q)>AG(Q)*가 성립한다고 볼 수 있다.

본 논문에서 질의 재작성에 이용하는 실체 뷰는 정규형 질의의 결과를 하나의 요약 테이블로 저장한 것이다. 단, 실체 뷰의 효용성을 고려하여 다음과 같은 두 조건을 만족하는 실체 뷰들만 고려한다: (1) $SG \ge AG$. 이 경우 *MV*의 각 그룹의 집계 값들은 사실 테이블 내의, 그 그룹에 속하는 모든 기초 데이타들에 대한 집계 결과이다. (2) 집계 결과에 대한 조건, 즉, SQL 질의의 HAVING절에 해당되는 조건을 갖지 않는다. 실체 뷰의 정규형은 *MV(SG, R, AG, AGG)*로 표현되고, 질의의 정규형과 동일하게 정의된다.

예 3: 예 1의 세 실체 뷰 MV_1, MV_2, MV_3와 질의 Q_1의 정규형은 다음과 같다.

- MV_1 *(SG, R, AG, AGG)* = MV_1 ((NONE, year, NONE, NONE), {((-∞, +∞), [1997, +∞), (-∞, +∞), (-∞, +∞))}, (state, year, NONE, NONE), {SUM(sales_dollar)})
- MV_2 *(SG, R, AG, AGG)* = MV_2 ((nation, NONE, NONE, NONE), {(['USA', 'USA'], (-∞, +∞), (-∞, +∞), (-∞, +∞))}, (state, yearmonth, NONE, NONE), {SUM(sales_dollar)})
- MV_3 *(SG, R, AG, AGG)* = MV_3 ((nation, year, NONE, NONE), {(['CANADA', 'CANADA'], (-∞, 1996], (-∞, +∞), (-∞, +∞))}, (city, yearmonth, NONE, NONE), {SUM(sales_dollar)})
- Q_1 *(SG, R, AG, AGG, HAV)* = Q_1 ((nation, year, NONE, NONE), {(['USA', 'USA'], [1996, 1999], (-∞, +∞), (-∞, +∞)), (['CANADA', 'CANADA'], [1996, 1999], (-∞, +∞), (-∞, +∞))}, (state, year, NONE, NONE), {SUM(sales_dollar)}, NULL)

이들의 선택 단위와 집계 단위를 그룹 격자상에 함께 표현하면 그림 8과 같다. □

정의 4: 질의(또는 실체 뷰) *Q*와 실체 뷰 *MV*의 두 선택 영역들 사이의 영역 교집합(intersection)과 영역 차(difference)는 각각 다음과 같이 정의된다.

$$R(Q) \cap^{s} R(MV) = \{R_i \cap R_j \mid R_i \in R(Q), R_j \in R(MV)\}$$

$$R(Q) -^{1} R(MV) = \{R_k \mid R_k \in (R_i - R_j), R_i \in R(Q), R_j \in R(MV)\}$$

이 때, $R_i \cap R_j$ 는 두 하이퍼-사각형 R_i와 R_j 의 교집합의 결과인 하이퍼-사각형이고, $R_i - R_j$ 는 두 하이퍼-사각형 R_i와 R_j 의 차의 결과인 하이퍼-사각형들의 집합으로서, 이들은 각 차원의 두 구간들 사이의 겹침(overlap) 관계로부터 계산될 수 있다. □

영역 교집합이나 영역 차의 선택 단위는 *Q*의 선택 단위와 *MV*의 선택 단위의 최대 하한(Greatest Lower

Bound)과 같다.

4. 실체 뷰를 이용한 질의 재작성

질의 Q가 주어졌을 때, 어떤 질의 Q'가 (1) 임의의 데이타베이스에 대해 Q와 같은 결과를 계산하고, (2) Q'가 어떤 질의 블록의 FROM 절에 MV를 포함하면, Q'을 실체 뷰 MV를 이용한 Q의 재작성된 질의라 부른다. 또, 이러한 재작성된 질의 Q'가 존재하면, MV가 질의 Q를 재작성 하는데 이용 가능하다라고 한다[5].

본 장에서는 주어진 정규형 OLAP 질의를 여러 가지 정규형 실체 뷰들을 이용하여 재작성하는 방법에 대해 기술한다. 재작성된 질의는 SQL을 이용하여 표현된다.

4.1 후보 실체 뷰

주어진 OLAP 질의 Q를 실체 뷰들을 이용하여 재작성하기 위해서는 Q의 집계 그룹들의 전체 혹은 일부분을 계산할 수 있는 실체 뷰들을 찾아야 한다. 어떤 실체 뷰 MV가 Q의 일부 결과를 계산하는데 이용될 수 있기 위한 조건들은 다음과 같다.

(1) $R(MV) \cap^{*} R(Q) \neq \emptyset$, (2) $AG(MV) \leq AG(Q)$,
(3) $AG(MV) \leq SG(Q)$, (4) $AGG(MV) \supseteq AGG(Q)$

조건 1은 MV의 선택 술어와 Q의 선택 술어를 모두, 만족하는 기초 데이타들이 존재함을 의미한다. 조건 2를 만족하면 MV의 집계 결과들을 다시 그룹화하고 집계하여 Q의 집계 결과를 계산할 수 있다. 조건 3을 만족하면 MV는 조건 1의 영역 교집합에 속하는 기초 데이타들에 대한 $AG(MV)$상의 집계 결과를 포함한다. 그러나 그렇지 않은 경우, 일반적으로 MV는 Q의 선택 술어를 만족하지 않는 기초 데이타에 대한 집계 결과를 포함한다. 조건 4를 만족하면 $AGG(MV)$가 $AGG(Q)$의 모든 원소들, 즉, 같은 측정 애트리뷰트에 대한 같은 집계 함수들을 포함하므로 그 결과로부터 Q의 집계 함수의 결과를 계산할 수 있다.

정규형 질의 Q가 주어질 때, 실체 뷰 MV가 위의 조건들을 모두 만족하면 MV는 Q를 재작성 하는데 이용 가능하다. 즉, MV 또는 MV와 사실 테이블을 이용하는 재작성 질의가 존재한다. 위의 조건들을 만족하는 실체 뷰들을 질의 Q에 대한 후보(candidate) 실체 뷰라 부르고, 그것들의 집합을 $C(Q)$로 표기한다. 기호의 편의를 위해 사실 테이블도 실체 뷰의 일종으로 간주하여 $C(Q)$에 항상 포함된다고 가정한다. 그림 9는 주어진 질의에 대해 $AG(MV) \leq AG(Q)$와 $AG(MV) \leq SG(Q)$를 만족하는 후보 실체 뷰의 집계 단위의 영역을 나타낸다.

4.2 질의 재작성 방법

주어진 OLAP 질의는 후보 실체 뷰들 중의 일부를 이용하여 재작성 된다. 본 논문에서 제안하는 질의 재작성 방법은 다음과 같은 세 단계로 이루어 진다(그림 11 참조).

4.2.1 단계 1: 실체 뷰들의 선택

먼저, 주어진 질의 Q에 대해 후보 실체 뷰들의 집합 $C(Q)$를 구한다. 그리고 $C(Q)$로부터 재작성에 실제로 한다. 일반적으로 어떤 실체 뷰 MV_i를 이용하여 재 사용될 실체 뷰들을 선택하고 각각의 질의 영역을 결정작성된 질의는 MV_i에 속한 튜플들 중 특정 선택 술어를 만족하는 튜플들을 선택하여 집계한다. 이 선택 술어는 그룹 격자의 한 선택 단위 상에서 정의되는 선택 영역으로 표현될 수 있다. 이 선택 영역을 MV_i에 대한 질의 영역이라 부르고, $QR(MV_i)$로 표기한다. 선택된 모.

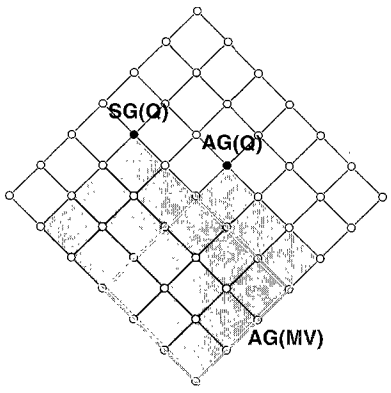


그림 9 후보 실체 뷰의 AG 영역

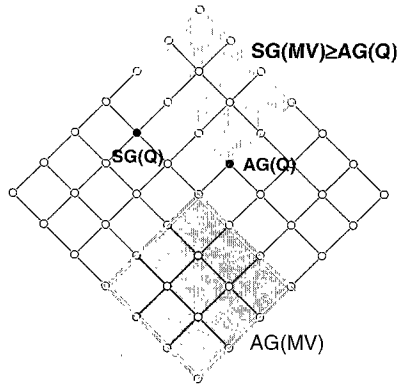


그림 10 UNION 다중 블록 결합이 가능한 실체 뷰의 AG 영역 및 SG 영역

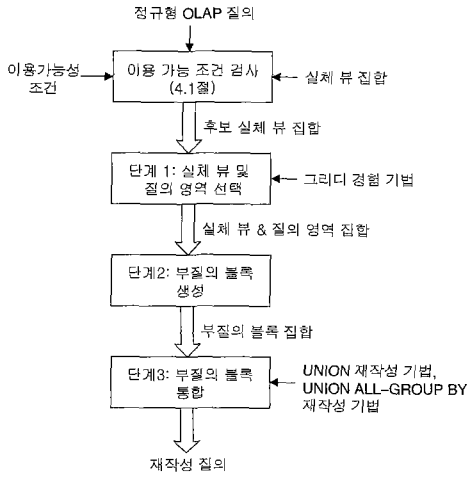


그림 11 질의 재작성 방법 개요도

든 실체 뷰들의 질의 영역들은 서로 겹치지 않는 분리된 영역이면서, 그 합집합이 질의의 선택 영역과 일치해야 한다. 즉, Q의 재작성에 선택된 실체 뷰들의 집합을 $S(Q)$라 할 때, 실체 뷰들의 질의 영역들은 다음 조건을 만족해야 한다.

$$QR(MV_i) - (R(MV_i) \cap R(Q)) = \phi,$$
$$QR(MV_i) \cap QR(MV_j) = \phi, \text{ for } \forall MV_i, MV_j \in S(Q) \text{ such that } MV_i \neq MV_j,$$
$$R(Q) - \bigcup_{MV_i \in S(Q)} QR(MV_i) = \phi$$

일반적으로 서로 다른 후보 실체 뷰들을 포함하는 여러 개의 동치 재작성 질의들이 존재한다. 이러한 재작성 질의들은 그 실행 비용이 서로 다르므로 효율적으로 실행될 수 있는 재작성 질의를 구성하는 실체 뷰들을 선택하고 그것들의 질의 영역을 결정해야 한다. 그러나 이 최적화 문제의 시간 복잡도는 NP-hard이다.[3] A^* 탐색이나 동적 프로그래밍 기법 등을 이용하여 최적해를 구할 수 있으나 최악의 경우 매우 긴 시간이 소요될 수 있다. 본 방법에서는 비교적 빠른 시간 내에 효율적인 해를 찾기 위해 그리디 경험 기법(greedy heuristic)을 사용한다. 이 방법은 후보 실체 뷰들의 집합에서 실체 뷰를 하나씩 선택하되, 매 선택 시 다음과 같은 척도를 이용하여 나머지 뷰들의 이득을 계산하여 이득이 가장 큰 것을 선택한다.

$$profit(MV_i, QR) = \frac{cost(FT, QR(MV_i))}{cost(MV_i, QR(MV_i))}$$

3) 시간 복잡도가 NP-hard인 최소 집합 커버 문제를 다항함수 시간 내에 이 문제로 변환할 수 있다.

즉, QR에 대한 각 실체 뷰 MV의 이득은 QR(MV)에 대한 사실 테이블과 MV의 실행 비용의 비로서, [9]에서 제안된 선형 비용 모델등을 이용하여 추정될 수 있다. 이미 선택된 실체 뷰들의 집합을 S_P라 할 때, 새로 선택될 실체 뷰 MVi의 질의 영역은 다음과 같이 결정된다.

$$QR(MV_i) = R(MV_i) \cap (R(Q) - \bigcup_{MV_j \in S_p} QR(MV_j))$$

이 방법의 시간 복잡도는 $O(n^2)$이다.

4.2.2 단계 2: 부질의 블록들의 생성

단계 1에서 선택된 각 실체 뷰 MV 마다 그것의 질의 영역 $QR(MV)$을 이용하여 부질의 블록 $SQB(MV)$을 생성한다. 먼저, 사실 테이블에 대한 다음과 같은 정규형 질의를 고려한다.

Q_{MV} (SG_{MV}, $QR(MV)$, $AG(Q)$, $AGG(Q)$, $HAV(Q)$)

선택 단위 SG_{MV}는 $QR(MV)$의 단위이다. 부질의 블록 $SQB(MV)$는 Q_{MV}와 동치이면서 사실 테이블 대신 MV를 이용하여 정의되어야 한다. MV에 포함된 차원 애트리뷰트는 $AG(MV)$에 속한 애트리뷰트들이므로, $SG(Q_{MV})$와 $AG(Q_{MV})$의 모든 애트리뷰트들이 $AG(MV)$에 포함되지 않으면 $R(Q_{MV})$에 대한 선택이나 $AG(Q_{MV})$에 대한 그룹화 및 집계를 수행하기 위해 MV와 차원 테이블 사이의 조인이 필요하다. 이 때, 만일 조인 애트리뷰트가 차원 테이블의 주 키가 아니면, 조인된 튜플의 중복을 피하기 위해 먼저 차원 테이블에 대한 중복-제거 선택 질의 블록을 생성하고 이것과 조인을 수행해야 한다. $R(Q_{MV})$에 대응되는 선택 술어 중 내포된 질의 블록에 대한 차원 선택 술어는 그 블록 내에 포함될 수 있으며, 나머지 차원 선택 술어들은 바깥쪽 블록에 포함된다.

정의 5: 질의 Q, 실체 뷰 MV, 그리고 질의 영역 $QR(MV)$가 주어질 때, MV에 대한 부질의 블록 $SQB(MV)$는 다음과 같이 정의된다. 단, GROUP BY 절은 $AG(Q)>AG(MV)$인 경우에만 필요하며, HAVING 절은 $HAV(Q)\neq$ NULL, $SG(MV)\geq AG(Q)$인 경우에만 필요하다.

SELECT	P, AGG
FROM	R
WHERE	JC, SC
GROUP BY	G
HAVING	HC

1. P = $\bigcup_{1\leq i\leq d}$ AG(Q,i) , AGG =

$\{agg_i(m_{MV})$ AS $label_i \mid agg_i(m) \in AGG(Q)$,
$m_{MV}=agg_j(m) \in AGG(MV)$, $agg_i=agg_j\}$,

$$\begin{cases} \text{if } AG(Q) > AG(MV), \\ \{m_{MV} \text{ AS } label_i \mid agg_i(m) \in AGG(Q), \\ m_{MV}=agg_j(m) \in AGG(MV), agg_i = agg_j\}, \\ \text{if } AG(Q) > AG(MV) \end{cases}$$

2. $R=\{MV, DT_i NB(DT_j) \mid$
$(AG(MV,i) \not\supseteq SG(Q_{MV},i) \vee AG(MV,i) \not\supseteq AG(MV,i) \supseteq L^i_o,$
$(AG(MV,i) \not\supseteq SG(Q_{MV},j) \vee AG(MV,j) \not\supseteq AG(MV,j) \supseteq L^i_o\}$

내포된 질의 블록 $NB(DT_i)$는 차원 테이블 DT_i에 대한 단일-블록 질의이다.

$NB(DT_i)$: (SELECT DISTINCT $AG(MV,i), AG(Q, i)$
FROM DT_i
WHERE $SP(QR(MV), i)$) NB_i

만일 $AG(MV, i)=AG(Q, i)$이면 둘 중 하나만 포함한다. $SP(QR(MV), i)$는 $QR(MV)$에 대응되는 선택 술어 중 i번째 차원에 대한 선택 술어를 의미한다. 단, i번째 차원에 대해 $QR(MV)$의 구간과 $R(MV)$의 구간의 최소(또는 최대) 한계 값들과 그것들의 구간 포함 여부가 서로 일치하면, 그 한계 값에 대한 선택 술어는 제외된다.

3. JC $= \wedge_{DT_i \in R} (MV.AG(MV,i) = DT_i.AG(MV,i)) \wedge \wedge_{NB_j \in R} (MV.AG(MV,j) = NB_j.AG(MV,j))$,

$SC= \wedge_{AG(MV,\ i) \supseteq SG(Q_{MV},\ j)} SP(QR(MV,i)) \wedge \wedge_{DT_j \in R} SP(QR(MV),j)$

4. $G = \begin{cases} \underline{AG(Q)}, \text{ if } \underline{AG(Q)} > \underline{AG(MV)} \\ \phi, \text{ if } \underline{AG(Q)} = \underline{AG(MV)} \end{cases}$

5. $H= \begin{cases} \underline{HAN(Q)}, \text{ if } \underline{HAN(Q)} \neq \text{NULL and } \underline{SG(MV)} \geq \underline{AG(Q)} \\ \phi, \text{ otherwise} \end{cases}$

4.2.3 단계 3: 부질의 블록들의 통합

만일 단계 1에서 하나의 실체 뷰만 선택된 경우에는 단계 2에서 생성한 부질의 블록이 재작성의 결과가 된다. 그렇지 않은 경우에는 부질의 블록들을 통합하여 하나의 재작성된 질의를 생성한다. 단계 1에서 선택된 실체 뷰들은 질의 Q의 집계 단위와 각 실체 뷰의 선택 단위의 관계에 따라 다음과 같이 구분된다.

1. $S_1 = \{ MV_i \mid SG(MV_i) \geq AG(Q) \}$: 이 집합에 속하는 실체 뷰는 Q의 일부 혹은 모든 그룹들의 집계를 계산할 수 있다. 그림 10은 이러한 실체 뷰의 선택 단위와 집계 단위의 가능한 영역을 나타낸다.

2. $S_2 = \{ MV_j \mid SG(MV_j) < AG(Q)$ or $SG(MV_j) <> AG(Q) \}$: 이 집합에 속하는 각 실체 뷰는 Q의 한 그룹의 집계를 계산하지 못할 수도 있으나, 전체 실체 뷰들은 Q의 각 그룹들의 집계를 계산할 수 있다.

$S_1 = \{MV_{11}, MV_{12}, ..., MV_{1m}\}$, $S_2 = \{MV_{21}, MV_{22}, ..., MV_{2n}\}$이라 할 때, 재작성된 질의는 다음과 같은 형태를 갖는다.

($SQB(MV_{11})$ *UNION* $SQB(MV_{12})$ UNION... UNION $SQB(MV_{1m})$)

UNION
(SELECT $AG(Q)$, $AGG'(Q)$
FROM ($SQB(MV_{21})$ UNION ALL $SQB(MV_{22})$ UNION ALL...UNION ALL $SQB(MV_{2n})$)
GROUP BY $AG(Q)$
HAVING $HAV(Q)$)

즉, S_1에 속하는 실체 뷰들의 부질의 블록들은 UNION 연산자를 사용하여 하나의 UNION 다중 블록 질의로 통합된다. S_2에 속하는 실체 뷰들의 부질의 블록들은 UNION ALL로 연결되고 $AG(Q)$를 이용한 GROUP BY 절을 포함하는 UNION ALL-GROUP BY 질의로 통합된다. S_2의 경우, $HAV(Q)$가 NULL이 아니면 $HAV(Q)$를 포함한 HAVING 절이 필요하다. 또, $AGG'(Q)$에 속하는 집계 함수 $agg'(m)$은, 질의의 집계 함수 $agg(m)$이 $agg \in$ {MIN, MAX, SUM}인 경우에는 $agg' = agg$이고, agg = COUNT 인 경우에는 agg' = SUM 이 된다. 마지막으로, S_1과 S_2의 두 결과를 UNION 연산자를 통해 통합함으로써 하나의 재작성된 질의를 생성한다.

예 4: 예 1의 질의 Q_1에 대한 재작성 과정은 다음과 같다.

단계 1: 4.1절의 조건에 의해 $C(Q_1)$={MV_1, MV_2, MV_3, Sales}이다. 단계 1의 기준을 적용하면 MV_1, MV_2, MV_3 순서로 선택되고, 각각의 질의 영역은 다음과 같이 결정된다 (그림 4 참조).

$QR(MV_1)$ = { (['USA', 'USA'], [1997, 1999], (-∞, +∞), (-∞, +∞)), (['CANADA', 'CANADA'],[1997, 1999],(-∞, +∞), (-∞, +∞))}

$QR(MV_2)$={ (['USA', 'USA'], [1996, 1996], (-∞, +∞), (-∞, +∞)),

$QR(MV_3)$=(['CANADA','CANADA'],[1997,1999],

(-∞, +∞), (-∞, +∞))}

단계 2: MV_1에 대한 부질의 블록의 작성 과정은 다음과 같다. $QR(MV_1)$에 대한 정규형 질의는 Q_{MV} ((nation, year, NONE, NONE), {(['USA','USA'], [1997,1999], (-∞, +∞), (-∞, +∞)), (['CANADA', 'CANADA'], [1997,1999], (-∞,+∞), (-∞, +∞))}, (state, year, NONE, NONE), {SUM(sales_dollar)}) 이다.

부질의 블록 $SQB(MV_1)$의 각 요소들은 다음과 같이 정의된다.

P={state, year}, AGG={sum_dollar$_1$}. $AG(MV_1)$ $=AG(Q_1)$이므로 SUM(sales_dollar)가 MV_1의 sum_dollar$_1$으로 치환된다. R={MV_1, NB(Store)}이다. $AG(MV_1,1)$={state}$\not\supseteq${nation}=$SG(Q_{MV},1)$이고, AG(MV$_1$, 1)={state}$\not\supseteq${store_id}=L^i_0이므로, Store 차원 테이블에 대한 내포된 질의 블록 NB(Store)와의 조인이 필요하다. $AG(MV_1, 1)=AG(Q_1, 1)$={state}이고, $SP(QR(MV_1), 1)$= "nation = 'USA' OR nation = 'CANADA'" 이므로, NB(Store)는 다음과 같이 정의된다.

(SELECT DISTINCT state
FROM Store
WHERE nation='USA' OR nation='CANADA') NB_1

$AG(MV_1, 2)=SG(Q_{MV}, 2)=AG(Q_1, 2)$={year}이므로 Time 차원 테이블과는 조인할 필요가 없다. JC = "MV_1.state = NB_1.state", SC = "year≤1999" 이다. $SP(QR(MV_1), 2)$ = "year≥1997 AND year≤1999"이나, "year≥1997"이 $SP(R(MV_1), 2)$에 포함되어 있으므로 제외된다. 한편, $AG(MV_1)=AG(Q_1)$, $HAV(Q_1)$ =NULL 이므로 GROUP BY 절과 HAVING 절이 필요 없다. 따라서, MV_1에 대한 부질의 블록은 다음과 같다.

$SQB(MV_1)$: SELECT state, year, sum_dollar$_1$
FROM MV$_1$, (SELECT DISTINCT state
FROM Store
WHERE nation = 'USA' OR
nation = 'CANADA') NB$_1$
WHERE MV$_1$.state = NB$_1$.state AND year≤1999

MV_2, MV_3에 대한 부질의 블록들도 같은 방법으로 정의될 수 있다 (그림 3의 Q_1' 참조).

단계3: MV_1, MV_2, MV_3는 각각 $SG(MV_1)> AG(Q_1)$, $SG(MV_2)>AG(Q_1)$, $SG(MV_3)>AG(Q_1)$을 만족한다. 따라서, MV_1, MV_2, MV_3의 각 부질의 블록들을 UNION으로 연결하여 하나의 UNION 다중 블록 재작성 질의를 생성할 수 있다. 그 결과는 Q_1'과 같다. □

한편, 예 2의 Q_2'에서 재작성을 위해 선택된 실체 뷰는 MV_1, MV_3이다. 이들은 각각 $SG(MV_1)<>AG(Q_2)$, $SG(MV_3)<>AG(Q_2)$을 만족한다. 따라서, Q_2'과 같이, MV_1, MV_3의 각 부질의 블록들을 하나의 UNION ALL-GROUP BY 질의로 통합하여 재작성된 질의를 생성할 수 있다.

5. 관련 연구

기존의 데이타베이스 질의 처리 분야에서 실체 뷰를 이용하여 주어진 질의를 재작성하는 연구들이 진행되어 왔다[1,2,3,4,5,6,7]. [1,2]에서는 집합 의미론(setsemantics)을 사용하여 논리곱 질의들에 대한 질의 재작성을 위해 뷰의 이용 가능성을 연구하고, 뷰의 정의와 질의의 정의 사이의 포함 사상(containment mapping)에 기반한 재작성 방법을 제안하였다.

[3]에서는 DW에서 집계 질의를 실행하기 위해 실체 집계 뷰들을 이용하는 문제를 고려하고, 질의 변환 규칙과 이를 이용한 재작성 방법을 제안하였다. 이 방법은 질의를 연산자 트리로 표현하고, 그것에 대해 변환 규칙들을 이용하여 구문적 변환을 수행한다. 그러나, 어떤 실체 뷰를 이용하기 위해서는 질의의 정의의 일부분이 실체 뷰의 정의와 일치되도록 변환되어야 한다. 따라서 이용할 수 있는 실체 뷰와 재작성의 결과가 매우 제한적이다.

[4]는 논리곱 Select-Project-Join(SPJ) 질의를 실체 뷰들을 이용하여 재작성하는 방법을 제안하고, 재작성된 질의들을 질의 최적화 과정에 포함시키는 방법에 대해 연구하였다. 실체 뷰를 고려한 질의 최적화를 위해 기존의 질의 최적화기에서 사용되는 조인 나열 알고리즘을 확장하였다. 그러나 이 논문에서는 OLAP에서 많이 사용되는 집계 질의나 실체 집계 뷰에 대해서는 고려하지 않았으며, 재작성된 질의로서 단일-블록 질의만 고려하였다.

[5]에서는 논리곱 뷰나 집계 뷰를 이용해서 집계 질의를 재작성하는 방법을 제안하였다. [5]에서 제안한 재작성 알고리즘에서 이용될 수 있는 실체 뷰는 기본적으로 다음과 같은 두 조건을 만족해야 한다.

(1) 실체 뷰의 정의에 사용된 모든 테이블들이 질의에 나타나야 한다.

(2) 질의의 SELECT 절이나 GROUP BY 절에 나타난 애트리뷰트가 실체 뷰의 정의에 사용된 테이블들 중 하나에 속한 애트리뷰트이면, 그것이 실체 뷰의 SELECT 절에도 나타나야 한다.

그러나 본 논문의 예 1, 예 2의 경우 실체 뷰들이 이러한 조건을 만족하지 않으므로 [5]에서 제안한 방법으로는 이 두 예와 같은 재작성 질의를 생성할 수 없다. 즉, [5]에서는 DW 스키마의 메타 정보들을 고려하지

않았고, 또 본 논문에서 제안하는 UNION ALL-GROUP BY 형태의 재작성에 대해서도 고려하지 않았다. 그 결과 OLAP 질의의 재작성에 이용될 수 있는 실체 뷰와 재작성 방법이 본 논문에 비해 매우 제한적이다.

한편, 최근에는 DW 상에서 좀 더 다양한 실체 뷰들을 이용하고 복잡한 질의를 처리하기 위한 방법들이 제안되었다[6, 7]. 기존의 연구들은 실체 뷰에서 질의로 일대일 사상이나 포함 사상이 존재하는 경우에만 그 실체 뷰를 이용 가능하나, [6]에서는 주어진 질의에 나타나지 않는 테이블들을 포함하는 실체 뷰를 이용하는 방법을 제안하였다. 또 실체 뷰와 질의의 그룹화 애트리뷰트들 사이의 함수적 의존 관계를 고려하여 실체 뷰의 이용 가능 여부를 결정한다. 그러나 본 논문과는 달리 질의와 실체 뷰의 선택 술어는 고려하지 않았으며 재작성 방법에서 생성할 수 있는 질의는 하나의 단일-블록 집계 질의만 가능하다. [7]은 복잡한 수식, 수퍼-그룹 집계 함수, 중첩된 부질의 등을 포함하는 복잡한 질의에 대한 재작성에 관한 연구이다. 질의와 실체 뷰를 질의 그래프 모델(Query Graph Model)을 사용하여 각각 그래프 형태로 표현하고, 질의 부그래프와 실체 뷰 부그래프 사이의 동치를 위한 부합(matching) 관계와 보상(compensation)을 정의하였다. 그리고 몇 가지 간단한 부그래프 부합 패턴들에 대해 부합 조건과 보상 규칙들을 정의하였다. 이 논문의 재작성 방법은 질의 그래프와 실체 뷰 그래프를 상향으로 함께 탐색하면서 부합이 가능한 부그래프 패턴들을 찾고 이에 대해 보상을 수행해서 실체 뷰를 포함한 재작성된 부질의들을 생성하는 과정으로 이루어 진다. 그러나 차원 테이블에 존재하는 차원 계층을 고려하지 않았고, 실체 뷰들의 이용 방법이 제한적이다. 또, 재작성에 사용될 실체 뷰들의 선택 문제와 질의 부합의 적용 순서에 대해서는 언급하지 않았다.

기존의 연구들과 본 논문과의 차이점을 요약하면 다음과 같다.

(1) 기존의 방법들은 차원 계층과 같은 DW의 메타 정보들과 OLAP의 특성을 효과적으로 이용하지 못하므로 실체 뷰의 활용도나 재작성된 결과가 매우 제한적이다.

(2) 본 논문에서 제안한 UNION ALL-GROUP BY 재작성을 할 수 없다.

(3) 주어진 질의와 동치인 다수의 재작성 질의들 중 효율적으로 실행될 수 있는 질의를 생성하는 문제에 대한 논의가 부족하다.

6. 결론 및 향후 연구

본 논문에서는 DW 시스템에 존재하는 여러 종류의 실체 집계 뷰들을 이용하여 주어진 OLAP 질의를 재작성하는 방법을 제안하였다. 제안한 방법은 기존의 방법들과는 달리 DW의 메타 정보들과 질의와 실체 뷰의 특성들을 고려하여 시스템에 존재하는 다양한 실체 뷰들을 질의 재작성에 이용한다. 이를 통해 시스템에 존재하는 실체 뷰들의 효용성을 높이고 보다 효율적으로 주어진 질의를 처리할 수 있다.

본 논문에서는 DW 스키마의 차원 계층들로부터 정의되는 그룹 격자를 이용하여 OLAP 질의와 실체 뷰의 정규형을 정의하고, 질의와 실체 뷰 사이의 관계를 이용하여 질의의 재작성에 이용 가능한 실체 뷰의 조건을 제시하였다. 본 논문에서 제안한 질의 재작성 방법은 세 단계로 이루어 진다. 첫째, 재작성에 이용될 실체 뷰들을 선택하고 각각의 질의 영역들을 결정한다. 둘째, 선택된 각 실체 뷰에 대해 그것의 질의 영역을 이용하여 하나의 부질의 블록을 생성한다. 셋째, 생성된 부질의 블록들을 하나의 질의로 통합한다. 통합하는 방식은 실체 뷰의 선택 단위와 질의의 집계 단위 사이의 관계에 의해 UNION 방식과 UNION ALL-GROUP BY 방식이 있다. 제안하는 방법은 서로 다른 집계 단위, 선택 단위, 선택 영역을 가진 실체 뷰들을 함께 이용하여 질의를 재작성할 수 있다. 또, 유용한 경험적 기준들을 이용하여 실체 뷰들과 질의 영역을 선택함으로써, 여러 가지 가능한 재작성 질의들 가운데 효율적으로 실행될 수 있는 재작성 질의를 생성한다.

본 논문에 대한 향후 연구로는, 질의 재작성에 이용할 최적 실체 뷰 집합을 선택하는 문제에 대한 연구가 좀 더 필요하다. 여러 가지 다양한 최적화 기법들과 경험적 기법들을 사용한 알고리즘들을 제안하고 실험을 통해 그것들의 성능을 비교 분석할 계획이다. 또, 각 실체 뷰에 대한 부질의 블록의 실행 비용을 보다 정확하고 빠르게 추정할 수 있는 방법에 대한 연구와, 재작성된 질의에 대한 최적화 방안, 질의 재작성 방법을 질의 최적화 과정과 통합하는 문제 등에 대해서도 연구가 필요하다.

참 고 문 헌

[1] H. Z. Yang and P. A. Larson, Query Transformation for PSJ-Queries, Proc. of 13th Int'l Conf. on Very Large Data Bases, Brighton, pp.245254, August 1987.

[2] A.Y. Levy, A.O. Mendelzon, Y. Sagiv, and D. Srivastava, Answering Queries Using Views, Proc. of ACM Symposium on Principles of Database Systems, SanJose, CA, pp.95104, May 1995.

[3] A. Gupta, V. Harinarayan, and D. Quass,

Aggregate-Query Processing in Data Warehousing Environments, Proc. of 21st Int'l Conf. on Very Large Data Bases, Zurich, Switzerland, pp.358369, Sept. 1995.

[4] S. Chaudhuri, R. Krishnamurthy, S. Potamianos, and K. Shim, Optimizing Queries with Materialized Views, Proc. of the 11th IEEE Int'l. Conf. on Data Engineering, Taipei, pp190200, March 1995.

[5] D. Srivastava, Sh. Dar, H.V. Jagadish, and A.Y. Levy, Answering Queries with Aggregation Using Views, Proc. of 22th Int'l Conf. on Very Large Data Bases, India, pp.318329, Sept. 1996.

[6] J. Chang and S. Lee, Query Reformulation Using Materialized Views in Data Warehouse Environment, Proc. of the First ACM Int'l Workshop on Data Warehousing and OLAP, Nov. 1998.

[7] M. Zaharioudakis, R. Cochrane, G. Lapis, H. Pirahesh, and M. Urata, Answering Complex SQL Queries Using Automatic Summary Tables, Proc. of 2000 ACM SIGMOD Int'l Conf. on Management of Data, Dallas, Texas, pp.105116 May 2000.

[8] S. Chaudhuri and U. Dayal, An Overview of Data Warehousing and OLAP Technology, SIGMOD Record, 26(1), pp.6574, March 1997.

[9] V. Harinarayan and A. Rajaraman and J. Ullman, Implementing Data Cube Efficiently, Proc. of ACM SIGMOD Int'l Conf. on Management of Data, Montreal, Canada, pp.205216, June 1996.

박 창 섭
1995년 한국과학기술원 전산학과 학사. 1997년 한국과학기술원 전산학과 석사. 1997년 3월 ~ 현재 한국과학기술원 전산학과 박사과정 재학중. 관심분야는 데이타웨어하우스, OLAP, 질의 최적화, 병렬 처리 등.

김 명 호
정보과학회 논문지 : 데이타베이스
제 28 권 제 1 호 참조

이 윤 준
1977년 서울대학교 계산통계학과 졸업. 1979년 한국과학기술원 전산학과에서 석사학위 취득. 1983년 France, INPGEN-SIMAG에서 박사학위 취득. 1983년 ~ 1984년 France, IMAG 연구원. 1984년 ~ 현재 한국과학기술원 전산학과 부교수. 1989년 MCC(미) 초빙연구원. 1990년 CRIN(불) 객원교수. 관심분야는 데이타베이스 시스템, 정보검색, 실시간 데이타베이스 등임.