

# 공간 데이터 웨어하우스 구축기에서 사실테이블 사전 계산 기법

○최유신, 유병섭, 박순영, 배해영  
인하대학교 컴퓨터·정보공학과  
e-mail : yschoi@billion21.com

## Pre-Computation of Fact table in a Spatial Data Warehouse Builder.

○Yu-Shin Choi, Byeong-Seob You, Soon-Young Park, Hae-Young Bae  
Dept. of Computer Science & Information Engineering, Inha Univ

### 요 약

공간 데이터 웨어하우스에서 구축기는 의사결정을 위한 기반 데이터의 구축을 담당한다. 일반적으로 공간 데이터 웨어하우스의 데이터 적재는 잦은 갱신으로 인한 서버의 부하를 줄이기 위하여 구축기에 적재할 데이터를 임시 저장하고 일정주기마다 적재하는 방법을 이용한다. 이때 구축기의 정보는 차원 테이블에 대한 갱신정보와 사실 테이블의 일부 갱신정보만을 유지하므로 여러 차원 테이블로 구성된 사실 테이블의 갱신은 공간 데이터 웨어하우스 서버에서 수행해야 한다. 사실 테이블의 갱신연산은 연관된 차원 테이블들에 의해 처리되므로 높은 처리 비용이 필요하다. 따라서 사실테이블의 처리로 인해 적재시간이 증가하며, 이는 사용자의 의사결정 응답시간을 증가시킨다.

본 논문에서는 공간 데이터 웨어하우스의 구축기에서 사실테이블의 사전 계산 기법을 제안한다. 이 기법은 차원 테이블 및 사실 테이블에 대한 메타정보와 추가적으로 기록되어야할 데이터 정보를 구축기에 유지한다. 구축기는 이 정보를 이용하여 삽입 연산시 사실 테이블에 적재할 갱신 정보를 사전에 계산하고, 이를 적재주기에 함께 적재한다. 따라서 사실 테이블의 갱신을 데이터 적재 이전에 구축기에서 계산하므로 공간 데이터 웨어하우스 서버에서 발생하는 높은 처리 비용을 감소시킬 수 있다. 공간 데이터 웨어하우스 사용자의 의사결정 응답시간을 감소시킨다.

### 1. 서론

공간 데이터 웨어하우스는 지리정보를 주제 중심적이고 통합적이며 시간성을 가지는 비휘발성 자료로 저장하여 효율적인 의사결정을 지원하는 시스템이다[1,2]. 운영계 데이터로부터 공간 데이터 웨어하우스로의 적재는 구축기를 통하여 수행되며, 구축기는 공간 데이터 웨어하우스를 처음 가동하기 위한 초기 적재와 지속적인 소스 데이터들의 변화를 반영하는 점진적 적재를 수행한다[3].

지속적인 소스데이터들의 변화를 효율적으로 반영하기 위해 구축기는 공간 데이터 웨어하우스에 적재할 데이터를 임시로 보관하고, 일정 주기마다 적재를 하는 방법을 사용하게 된다. 이때 구축기에 임시 보관되는 정보는 차원 테이블에 대한 갱신정보와 사실 테이블의 일부 갱신정보만을 유지한다[4]. 이로 인해 여러 차원 테이블의 데이터 액세스에 의해 처

리되는 사실 테이블 갱신연산을 공간 데이터 웨어하우스 서버에서 수행 하여 높은 처리 비용이 필요하다[5]. 따라서 사실테이블의 처리로 인해 적재시간이 증가고, 이는 사용자의 의사결정 응답시간을 증가시킨다.

본 논문에서는 구축기의 사실테이블 사전계산 기법을 제안한다. 제안기법은 공간 데이터 웨어하우스와 동일한 차원 테이블 및 사실 테이블의 키 인덱스를 구축기 내에 유지한다. 구축기는 이 인덱스를 이용하여 소스 데이터의 삽입 연산시 차원 테이블과 사실 테이블에 적재될 갱신정보 계산을 사전에 계산하고, 주기적으로 공간 데이터 웨어하우스 서버에 적재한다. 구축기에서 사실 테이블 갱신정보 사전계산은 공간 데이터 웨어하우스 서버에서 발생하는 사실 테이블 갱신연산의 높은 처리 비용을 감소시킬 수 있다. 따라서 공간 데이터 웨어하우스 사용자의

\* 본 연구는 대학 IT 연구센터 육성·지원사업의 연구결과로 수행되었음

의사결정 응답시간을 감소시킨다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구를 살펴보고 3장에서는 본 논문이 제시하고 있는 구축기의 사실테이블 사전 계산 기법을 설명한다. 마지막으로 4장에서 결론 및 향후연구를 기술한다.

2. 관련 연구

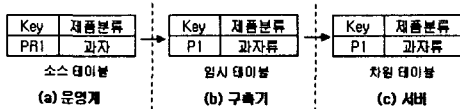
데이터 웨어하우스 구축에 있어 구축기는 데이터를 소스 시스템으로부터 추출한 후 정제, 변환작업을 거쳐 데이터 웨어하우스에 적재한다. 데이터의 적재는 초기적재 이후 소스 시스템들의 지속적인 변경사항을 일정 주기 간격으로 반영하여 적재한다.

2.1 차원 테이블 처리

운영계 소스 데이터의 갱신으로 인해 차원 테이블의 변경이 발생하게 되면 구축기는 다음과 같은 작업을 수행한다.

먼저 데이터 웨어하우스 차원 테이블의 키와 소스 데이터의 키가 다를 수 있으므로 구축기에서 소스 데이터의 키를 데이터 웨어하우스 시스템이 생성하는 키들로 변환하고, 변환된 내역을 구축기에 유지한다. 구축기에 유지된 데이터는 일정 주기로 데이터 웨어하우스에 적재된다[6]. 이때 데이터 웨어하우스 서버는 기간차원과 같이 구축기에서 처리할 수 없는 데이터의 연산을 수행한다.

예를 들어, 운영계의 소스 테이블이 [그림 1]의 (a)와 같다고 하자. 구축기는 소스 테이블의 데이터를 추출하고, 메타 데이터를 참조하여 (c)의 차원 테이블 코드와 (a)의 소스 데이터의 코드 불일치를 없애는 변환을 수행한다. 이때 Key와 제품분류 코드가 "PR1->P1"으로 "과자->과자류"로 변환된다. 변환된 데이터는 (b)의 임시테이블에 유지된다. 구축기에 임시로 유지된 데이터는 일정 주기마다 데이터 웨어하우스 서버에 적재 된다.



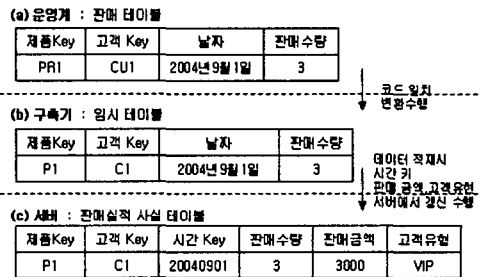
[그림 1]차원 테이블 처리

2.2 사실 테이블 처리

구축기에서 사실 테이블 구조는 참조 할 수 있어도 사실테이블에 적재되어 있는 데이터의 존재 여부는 데이터 웨어하우스에 질의를 수행해야만 알 수

있다. 구축기에서 사실 테이블과 연결된 차원 테이블의 키를 찾아 변환 작업을 수행하고, 사실 테이블을 참조하지 않고 메타 데이터만으로 구성할 수 있는 임시 데이터를 구축기에 유지한다. 사실테이블에 대해 임시로 유지된 데이터는 데이터 웨어하우스에 적재될 때 구축기가 계산 할 수 없었던 부분을 데이터 웨어하우스에서 수행하여 사실 테이블을 유지한다.[7]

예를 들어 운영계의 판매 테이블이 [그림 2]의 (a)와 같다고 할때, 구축기는 메타데이터를 참조하여 판매실적 사실 테이블의 코드와 소스 데이터의 코드 불일치를 없애는 코드변환을 수행한다. 이때 제품키와 고객키가 "PR1->P1", "CU1->C1"으로 변환되고, 변환된 데이터는 [그림 2]의 (b)처럼 구축기의 임시 테이블에 적재 된다. 구축기에 의한 데이터 적재 시 서버는 구축기에서 연산하지 못한 판매금액과 고객유형 및 시간키에 대한 연산을 수행하여야 한다.



[그림 2] 사실 테이블 처리

따라서, 기간 차원에 대해서 서버의 연산을 필요로 하고, 여기서 나온 시간 Key와 사실 테이블에 필요한 판매금액과 고객유형을 사실 테이블 레코드에 기록하기위해 높은 처리비용이 발생되며, 사용자의 의사결정 응답시간을 증가 시킨다.

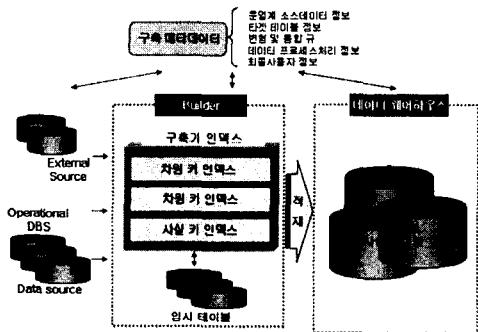
3. 구축기에서 사실 테이블 사전 계산 기법

본 장에서는 소스 데이터의 갱신이 있을 경우, 구축기에서 차원테이블 및 사실테이블의 변경된 정보를 이용하여 사전 계산하고 일정 주기마다 공간 데이터 웨어하우스에 적재하여, 공간 데이터 웨어하우스 서버의 차원 테이블과 사실 테이블의 갱신연산 비용을 줄이고 사용자의 의사결정 응답시간을 감소시키는 방법에 대해 기술한다.

3.1 공간 데이터 웨어하우스 구축기 구성

공간 데이터 웨어하우스 구축기의 기본 구성은

[그림 3]과 같다. 구축기는 여러 운영 데이터 베이스와 연결되어 갱신된 소스 데이터를 추출한다. 또한 구축기는 데이터 웨어하우스의 서버와 연결되어 일정한 주기마다 임시로 유지한 데이터를 적재한다. 구축기의 내부에는 데이터를 임시로 보관하는 저장소가 존재하며, 사실 테이블의 사전 계산을 위한 차원키 인덱스와 사실키 인덱스가 존재한다. 차원키 인덱스는 서버의 차원 테이블에 대한 키 값을 찾아낼 때 사용하며, 사실 키 인덱스는 사실 테이블에 대한 키 값을 찾아낼 때 사용한다.



[그림 3] 공간 데이터 웨어하우스 구축기 구성

### 3.2 임시 데이터 저장 구조

임시 데이터의 저장 구조는 플래그와 데이터로 구성된다. 플래그는 적재 과정에서 삽입 또는 갱신이 필요한 데이터를 구분하기 위하여 사용하고, 데이터는 서버에 적재될 레코드를 위하여 사용한다. 플래그의 종류는 3가지가 있으며, 기존 데이터의 정보가 추가되는 데이터 추가연산의 'A' 플래그, 새로운 데이터를 삽입하는 데이터 삽입연산의 'I' 플래그, 그리고 기존의 잘못된 데이터를 수정하는 데이터 수정연산의 'U' 플래그로 구성된다.

임시 테이블

Flag	
I	

데이터 추가연산 Flag = 'A',  
 데이터 삽입연산 Flag = 'I',  
 데이터 수정연산 Flag = 'U'

[그림 4] 임시 데이터 저장 구조

### 3.3 차원 테이블 처리

구축기에서 차원 테이블 처리는 다음의 절차를 따른다. 먼저 공간 데이터 웨어하우스의 차원 테이블의 키와 소스 데이터의 키가 다를 수 있으므로 구축기에서 소스 데이터의 키를 공간 데이터 웨어하우스 시스템이 생성하는 키들로 변환한다. 변환된 키가 구축기에 유지되는 차원 인덱스에 없으면 키 값을

생성하여 구축기의 차원 인덱스에 추가하며 플래그를 'I'로 설정하고 임시 테이블에 변환 데이터를 저장한다. 구축기의 차원 인덱스에 키 값이 있을 경우 플래그를 'U'로 설정하고 기존의 방식을 따른다. 임시 테이블의 데이터 적재시 'I' 플래그가 붙은 레코드는 차원 테이블에 없는 레코드이므로 서버에서 레코드 구성을 위한 연산 없이 바로 적재가 가능하다.



[그림 5] 차원 테이블 처리

예를 들어 소스 테이블이 [그림 5]의 (a)와 같다면 코드 불일치를 없애기 위해 메타 데이터를 참조 "PR1->P1", "과자->과자류"로 변환을 수행한다. 차원키 인덱스에 키가 없을 경우 플래그를 'I'로 설정한 후 임시 /테이블에 데이터를 유지한다. 구축기에 유지된 데이터는 일정주기로 공간 데이터 웨어하우스에 적재되며, 'I' 플래그로 설정된 레코드는 차원 테이블에 없는 레코드이므로 공간 데이터 웨어하우스의 차원 테이블에 바로 적재가 가능하다.

### 3.4 사실 테이블 처리

구축기에서 사실 테이블 처리는 기존의 존재하는 데이터의 정보가 추가되는 데이터 추가연산과 사실 테이블의 새로운 데이터를 삽입하는 데이터 삽입연산, 기존의 잘못 삽입된 데이터를 수정하는 데이터 수정연산이 존재한다.

#### 3.4.1 데이터 추가연산

삽입 연산시 사실 테이블 키 인덱스에 해당 데이터가 존재하면 구축기는 임시 테이블에 갱신 레코드를 유지한다. 이 경우 플래그를 'A'로 두어 구축기에서 적재시 공간 데이터 웨어하우스에서 기존 데이터의 정보를 추가하는 연산을 수행 하도록 한다.

Flag	제품 Key	...
A	P1	...

[그림 6] 데이터 추가연산 구조

#### 3.4.2 데이터 삽입연산

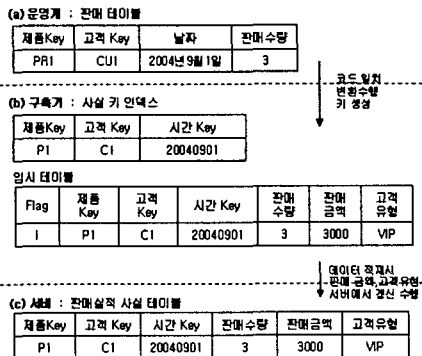
삽입 연산시 사실테이블 키 인덱스에 해당 데이터가 존재하지 않을 경우 먼저 기간차원 존재유무를

확인한다. 다음으로 구축기는 사실 테이블과 연결된 차원 테이블의 키를 찾아 키 변환 작업을 수행한다. 변환된 키가 구축기에 유지되는 사실 인덱스에 없으면 키 값을 생성 구축기의 사실 인덱스에 추가하며, 사실 테이블 레코드의 부가 정보도 메타데이터를 참조하여 생성한다. 생성된 데이터는 임시 테이블에 저장되어 주기적으로 공간 데이터 웨어하우스에 적재되며, 공간 데이터 웨어하우스 서버는 사실테이블에 대한 갱신 연산을 수행할 필요가 없다. 이는 [알고리즘 1]과 같이 나타낼 수 있다.

- Step 1) 기간치원 존재유무를 확인한다.
- Step 2) 사실 테이블과 연결된 차원 테이블의 키를 찾아 키 변환 작업을 수행후 사실테이블 키 인덱스를 생성
- Step 3) Flag 'I' 설정
- Step 4) 사실테이블 키 인덱스 삽입
- Step 5) 사실 테이블 부가정보를 생성
- Step 6) 사실 테이블 저장 레코드 구조 형성
- Step 7) 임시테이블에 레코드 저장

[알고리즘 1] 데이터 삽입연산 알고리즘

예를 들어 판매 테이블이 [그림 7]의 (a)와 같을 때 소스데이터의 삽입 연산시 구축기는 사실 테이블과 연결된 차원 테이블의 키를 찾아 “PR1->P1”, “CU1->C1”, “2004년9월1일->20040901”로 키 변환 작업을 수행한다. 변환된 키가 구축기의 사실키 인덱스에 없으면 키 값을 생성 구축기의 사실키 인덱스에 추가하며, Flag를 'I'로 설정한다. 사실 레코드의 판매수량, 판매금액, 고객유형도 메타데이터를 참조하여 생성한다. 이렇게 생성된 데이터는 임시 테이블에 저장되며 주기적으로 공간 데이터 웨어하우스에 적재된다. Flag가 'I'로 설정된 레코드는 새로 생성되는 레코드이므로 데이터 웨어하우스는 적재시 다른 연산의 수행 없이 바로 적재가 가능하다.



[그림 7] 사실 테이블 처리

따라서 서버의 적재비용이 감소하고, 이는 사용자 응답시간을 감소시킨다.

### 3.4.3 데이터 수정연산

수정연산은 기존에 잘못 삽입된 데이터의 수정이므로 구축기는 Flag에 'U'로 표시하여 데이터 적재시 기존의 갱신 연산을 수행 한다.

Flag	제품 Key	...
U	P1	...

[그림 8] 데이터 수정연산 구조

## 4. 결론 및 향후 연구

본 논문에서는 공간 데이터 웨어하우스의 사실테이블의 갱신연산 비용을 줄이기 위해 구축기의 사실테이블 사전계산 기법을 제안했다. 제안기법은 공간 데이터 웨어하우스와 동일한 차원 테이블과 사실 테이블의 키 인덱스를 구축기 내에 유지하였다. 구축기에 유지된 키 인덱스를 이용하여 공간 데이터 웨어하우스의 접근 없이 차원 테이블과 사실 테이블에 적재될 정보를 사전에 계산 할 수 있다.

따라서 공간 데이터 웨어하우스 서버에서 발생하는 갱신연산의 높은 처리 비용을 감소시키고, 이는 사용자의 의사결정 응답시간을 감소시킨다.

향후 연구로는 본 논문에 대한 성능평가가 필요하고, 서버의 인덱스에 대한 처리를 연구해야 한다.

### 참고문헌

- [1] W.H. Inmon, "What is a Data Warehouse?", Prism Solutions Tech Topics, vol.1, no.1,1995
- [2] Lafond, P., "Designing and Building the Distributed Geospatial Data Warehouse Architecture", Proceedings of The Twelfth Annual Symposium on Geographic Information Systems. Toronto, 1998
- [3] S. Chaudhuri and U. Dayal "An Overview of Data Warehousing and OLAP Technology" ACM SIGMOD Record 26(1), 1997
- [4] Paulraj Ponniah, "data warehousing fundamentals" WILEY, 2001
- [5] R. Kimball, Data Warehouse ToolKit, John Wiley, 1996
- [6] Peter Griffiths, "Slowly Changing Dimensions: A Data Warehouse Ongoing Challenge", DM Review, 2001
- [7] "The Oracle Server as an ETL Toolkit", An Oracle White Paper, 2000
- [8] W.H. Inmon, Building the Data Warehouse 2nd Edition, John Wiley&Sons,Inc.,1996