데이터 웨어하우스의 뷰를 활용한 고객구매 데이터 관리

이단영°, 안형근, 고재진
울산대학교 컴퓨터·정보통신공학부
{danyoung64°, hkahn, jjkoh}@mail.ulsan.ac.kr

Customer-Purchasing Data Maintenance using View of Data Warehouse

Dan-Young Lee°, Hyoung-Keun Ahn, Jae-Jin Koh
School of Computer Engineering & Information Technology, University of Ulsan

요 약

본 논문은 기존 쇼핑몰의 데이터베이스에서 쇼핑몰 운영자가 여러 각도에서의 올바른 자료 분석, 신속한 의사 결정, 데이터의 무결성과 일관성을 제공하고 효율적인 데이터 사용을 위해 고객의 구매데이터에 빠르게 접근할 수 있도록 데이터 웨어하우스에서 뷰를 활용한 관리 방법을 제시하고자 한다. 제시된 기법들 중에서 부가 파일을 이용하여 효과적으로 실체뷰를 자체적으로 관리하는 방안을 알아보고자 한다.

1. 서 론

기존의 쇼핑몰에서 발생되는 고객들의 구매 데이터는 실시간으로 계속 축적되고 있다. 이렇게 축적된 데이터베이스의 양은 점점 많아진다. 현재까지 누적된 방대한 양의 데이터를 통합적으로 관리하여 의사결정에 필요한 통계값이나 요약된 정보를 제공하는 데이터 웨어하우스로 구축한 뒤, 데이터 웨어하우스 내의 정보는 사용자의 질의에 대한 빠른 응답을 위해 실체뷰(Material View)로 저장된다. 실체뷰는 자주 요구되는 질의에 관계되는 투플들이 요약된 형태로 실제로 저장되는 테이블이다. 실체뷰는 기본 릴레이션으로부터 추출된 요약된 결과를 별도로 저장하고 있으므로 소스 데이터에서 삽입, 삭제, 수정이 발생하였을 경우, 질의에 정확한 결과를 제공하기 위하여 기본 릴레이션의 변경사항과 동일하게 갱신되어야 한다.

소스데이터의 갱신에 따라 뷰를 최신의 정보로 유지하는 것을 뷰 관리(View maintenance)라 한다. 데이터 웨어하우스에서 뷰 관리는 뷰 내의 데이터가 여러 데이터베이스에 분산되어 저장되어 있기 때문에 많은 시간이 소요되기도 하며, 네트워크 상황에 따라 데이터 소스에 대한 접근이 불가능할 수도 있어 뷰 관리에 어려움이 많다.[10]

뷰 관리 기법으로는 뷰의 재정의 기법(view redefinition)과 점진적 뷰 관리 기법(incremental view maintenance)이 연구되어 왔다[1,2,3,7,8,9]. 뷰 재정의 기법은 데이터 소스의 변경사항 발생시 뷰의 정의에 따라 기본 릴레이션으로부터 뷰를 재계산하는 방법이다[1]. 뷰 재정의 기법은 기본 릴레이션의 접근으로 인해 많은 계산 비용이 소요된다는 단점이 있다.

이에 비해 점진적 뷰 관리 기법은 기본 릴레이션의 변경사항만을 가지는 보조 릴레이션(auxiliary relation) 또는 부가 화일(Differential File : DF)을 이용하여 뷰를 관리하는 기법이다[1,7,8]. 보조 릴레이션과 부가 화일을 사용하는 방법은 소스데이터 갱신(삽입, 삭제, 수정)에 대해 데이터 소스의 접근없이도 뷰에 변경사항을 반영하는 뷰의 자체적 관리 기법이다 [4,6]. 이러한 뷰의 자체적인 관리는 단지 실체뷰의 내용과 기본 릴레이션의 변경사항만을 가지고 뷰를 관리한다[4]. 따라서 갱신된 데이터 소스에 대한 직접적 접근이 없기 때문에 질의에 대한 빠른 응답과 높은 이용 가능성을 제공하므로 대형의 뷰들을 관리하는데 매우 효율적이다.

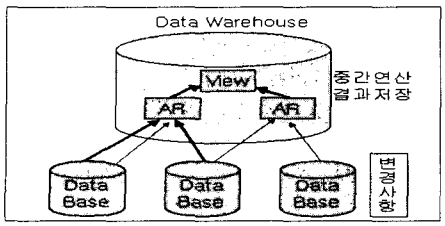
본 논문에서는 제시된 기법들 중에서 부가 파일을 이용하여 효과적으로 실체뷰를 자체적으로 관리하는 방안을 제시하고자 한다. 부가 파일을 이용한 방법에서는 기본 릴레이션에 대한 추가 변경 사항을 별도로 유지하고 이를 이용하여 뷰를 갱신하는 방법이다.

2. 관련 연구

데이터 웨어하우스에서 기본 릴레이션의 변경에 따라 실체뷰를 최신의 정보로 유지하는 뷰 관리 비용을 줄이기 위하여 기본 릴레이션의 변경사항만을 뷰에 점진적으로 반영하는 뷰의 자체적인 관리에 대한 연구가 관심 깊게 연구되고 있다. 뷰의 자체적인 관리 방법은 보조 릴레이션(auxiliary relation)과 부가 화일(Differential File : DF)을 사용하는 방법으로 크게 구분된다.

2.1 보조 릴레이션

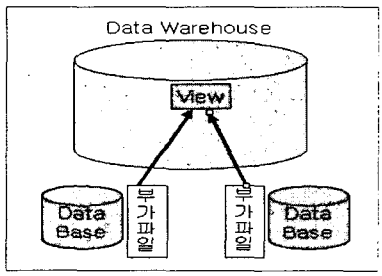
보조 릴레이션을 이용하는 방법은 (그림 1)과 같이 뷰 정의에 사용된 연산들의 중간 결과를 별도의 릴레이션(즉, 보조 릴레이션)에 유지해 두며, 기본 릴레이션이 갱신될 때마다 이 갱신 투플과 연산의 중간 결과들을 결합하여 최종 뷰에 반영할 정보를 결정하고 이를 반영하는 방법이다[3,7].



(그림 1) 보조릴레이션을 이용한 뷰 갱신

2.2 부가 화일

부가 파일을 이용하는 방법은 (그림 2)에서 보여지는 것처럼 각각의 기본 테이블에 대한 변경사항들만 별도의 부가 화일에 유지한 후 일정 시간마다 이 변경 사항들을 뷰에 반영하는 방법이다[10]. 부가 릴레이션에는 기본 테이블의 변경사항만 저장하므로 유지해야 할 정보의 양이 더 작다. 또한 기본 테이블이나 중간 연산, 결과를 접근하지 않고, 부가 릴레이션만으로 최종 뷰를 수정하므로 처리시간이 적게 소요된다는 장점이 있다.



(그림 2) 부가 파일을 이용한 뷰 갱신

부가 파일을 이용한 뷰 관리 기법에 대한 기존연구로, [11]은 부가 파일에 갱신 투플들을 분석하여 여러번 갱신된 투플들은 최종 갱신 내용만 남기고 나머지는 제거하는 스크린 테스트 방법을 제안하였다. 이렇게 함으로써 부가 릴레이션의 양을 줄이고 따라서 뷰 변경 시간을 줄일 수 있도록 하였다.

2.3 조인 뷰(Join View)

[5]에서는 기존의 부가 파일 외에 다른 기본 릴레이션에 새로운 투플이 삽입될 때 참조 무결성 제약 조건을 이용하여 이 투플과 조인될 투플을 찾아 별도의 조인 부가 파일에 유지하여, 기본 릴레이션들을 전혀 접근하지 않고 조인 뷰만을 관리하는 방법을 제안하였다. 이 방법은 기본 릴레이션보다는 크기가 매우 작은 부가 파일과 조인 부가 파일만 사용하므로 뷰 갱신 시간을 현저히 줄일 수 있다는 장점이 있다.

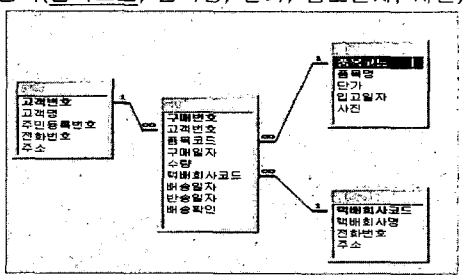
3. 데이터 웨어하우스를 위한 뷰 구조

본 논문에서 제안하고자 하는 쇼핑몰의 고객 구매 데이터에 빠른 접근을 위한 데이터 웨어하우스의 뷰 관리를 위해 예제로 사용할 기본 릴레이션과 부가 파일에 대해 설명한다. 기본 릴레이션은 3.1절에서와 같이 3개의 릴레이션으로 구성되어 있고, 데이터 웨어하우스는 3개의 릴레이션에서 추출된 데이터로 가정한다.

3.1 기본 릴레이션과 뷰

품목, 고객, 구매, 택배회사에 관한 데이터베이스는 아래와 같은 릴레이션으로 구성되어 있다고 가정한다.

▶ 고객(고객번호, 고객명, 주민등록번호, 전화번호, 주소)
▶ 품목(품목코드, 품목명, 단가, 입고일자, 사진)



(그림 3) 쇼핑몰 데이터베이스 릴레이션

다음과 같이 정의 할 수가 있다.

▶택배회사(택배회사코드, 택배회사명, 전화번호, 주소)
▶구매(구매번호, 고객번호, 품목코드, 구매일자, 수량, 택배회사코드, 배송일자, 반송일자, 배송확인)

(그림 4)는 이 릴레이션들에 적용되는 질의에 대한 실체 뷰의 예이며, 아래 (그림 5)는 위의 릴레이션에 대한 예를 보여 주고 있다. 고객의 고객번호와 구매의 고객번호, 품목의 품목코드와 구매의 품목코드, 택배회사의 택배회사코드와 구매의 택배회사코드 사이에는 참조 무결성 규칙이 존재한다고 가정한다. 각 릴레이션에서 밑줄이 쳐진 애트리뷰트는 그 릴레이션의 기본 키를 나타낸다.

```
CREATE VIEW V(고객.고객번호, 고객.고객명, 고객.전화번호, 구매_고객번호, 구매.품목코드, 구매.수량, 구매.구매일자) AS
SELECT 고객_고객번호, 고객.고객명, 고객.전화번호, 구매_고객번호, 구매.품목코드, 구매.수량, 구매.구매일자
FROM 고객, 구매
WHERE 고객.고객번호 = 구매.고객번호;

CREATE VIEW V1(품목.품목코드, 품목.품목명, 품목.단가, 구매.품목코드, 구매.구매일자, 구매.수량) AS
SELECT품목.품목코드, 품목.품목명, 품목.단가, 구매.품목코드, 구매.구매일자, 구매.수량
FROM 품목, 구매
WHERE 품목.품목코드 =구매.품목코드;

.....
```

(그림 4) 기본 릴레이션에 대한 뷰 정의

고객번호	고객명	주민등록번호	전화번호	주소
1	이순신	801010-1908765	576-6868	울산시
2	이정희	900202-2123232	686-6868	부산시
3	김순희	601213-1675644	123-4567	울산시
4	정정희	600312-2131232	456-4534	부산시
5	박순정	780909-2123433	342-3746	서울시
6	장정장	700707-1798686	7879-7979	울산시

(1) 고객

구매번호	고객번호	품목코드	구매일자	수량	택배회사코드	배송일자	받은확인
1	이정희	비디오	2003-11-25	5 한진		2003-11-26	2003-11-27	☑
2	박순정	오디오	2003-11-25	6 대한		2003-11-26		☐
3	박순정	컴퓨터	2003-11-25	7 대한		2003-11-26		☐
4	김순희	디카	2003-11-25	4 동서택배		2003-11-26		☐
5	김순희	냉장고	2003-11-25	1 거니		2003-11-26		☐
6	박순정	컴퓨터	2003-11-25	3 우리		2003-11-26		☐
7	이순신	디카	2003-11-25	6 대한		2003-11-26		☐
8	이순신	비디오	2003-11-25	2 대한		2003-11-26		☐
9	김정희	디카	2003-11-25	1 우리		2003-11-26		☐
10	정정희	비디오	2003-11-25	2 우리		2003-11-26		☐
11	장정희	디카	2003-11-25	1 우리		2003-11-26		☐
12	정정희	비디오	2003-11-25	2 우리		2003-11-26		☐
13	이순신	비디오	2003-11-25	2 대한		2003-11-26		☐
14	이정희	냉장고	2003-11-25	2 우리		2003-11-26		☐
15	박순정	냉장고	2003-11-25	1 한진		2003-11-26		☐

(2) 구매

택배회사코드	택배회사명	전화번호	주소
1	한진	909-0909	울산시
2	대한	112-2345	울산시
3	거나	1234-4567	부산시
4	우리	345-6789	부산시
5	동서계이	456-7988	울산시

(3) 택배회사

품목코드	품목명	단가	업그레
1	컴퓨터	₩350	2003-0
2	TV	₩560	2003-0
3	냉장고	₩300	2003-0
4	디카	₩40	2003-0
5	비디오	₩50	2003-0
6	오디오	₩67	2003-0

(4)품목

(그림 5) 기본 릴레이션의 예

3.2 부가 파일의 관리

기본 릴레이션의 갱신에 대해서 실체뷰는 최신의 정보로 유지되어야 한다. 이를 위해 각 기본 릴레이션마다 하나의 부가 파일을 가지며 변경사항을 부가 파일에 저장한다. 즉, 부가 파일은 기본 릴레이션의 변경사항을 기록한 일련의 부가정보이다.데이터웨어하우스의 뷰 갱신이 주기적으로 갱신되어진다면 부가 파일내에는 동일한 VTID(뷰의 기본 릴레이션의 기본 키)를 갖는 투플들의 연속적인 갱신이 존재할 수 있다. 만일 부가 파일내의 정보 중 동일한 VTID를 갖는 투플이 연속적으로 변경되어진다면 최초와 최종적인 변경사항만을 제외하고 중간 과정의 변경사항은 뷰 갱신에 있어 불필요한 과정으로 존재하게 된다. 따라서 이러한 중간 단계의 변경사항에 과난 투플을 중복된 투플이라 간주하여 삭제할 수 있다. 이런 과정을 중복삭제라 지칭하고 최초와 최종값만 남기고 모두 제거한다. 이런 중복 삭제의 과정을 거침으로써 뷰 갱신에 필요한 최소한의 정보를 얻을 수 있다.

4. 결 론

본 논문은 기존의 쇼핑몰에서 발생되는 고객들의 구매 데이터가 실시간으로 계속 축적되어지고, 이렇게 축적된 데이터베이스의 양은 점점 많아진다. 현재까지 누적된 방대한 양의 데이터를 통합적으로 관리하여 의사결정에 필요한 통계값이나 요약된 정보를 제공하는 데이터 웨어하우스로 구축한 뒤, 데이터 웨어하우스 내의 정보는 사용자의 질의에 대한 빠른 응답을 위해 실체뷰(Material View)로 저장된다. 데이터에 빠르게 접근하여 할 수 있도록 데이터 웨어하우스에서 사용되는 실체 뷰를 관리하는 방법으로 점진적 뷰 관리 기법에서 기본 릴레이션의 변경사항만을 별도로 저장하는 부가 파일을 이용하는 방법 제시하였다. 향후과제로는 참조 무결성과 여러 연산을 조합한 복합 뷰의 관리의 연구도 기대된다.

5. 참고 문헌

[1] L.Colby, A.Kawaguchi, D. Lieuwen, I. S. Mumick AND L. Ross, "Supporting Multiple View Maintenance Polices," In *proc. of ACM SIGMOD Conf.*, pp. 405-416, 1997Mumick

[2] A Gupta, H. Jagadish and I. S. Mumick, :Data Integration Using Self-maintainable Views," In *Proc. EDBT,* pp. 140-144, 1996

[3] T. Griffin and L. Libkin, "Incremental Maintenance of Views with Duplicates," In *proc. of ACM SIGMOD Conf.*, pp. 328-339, 1995

[4] A. Gupta and I. S. Mumick, "Maintenance of Materialized Views : Problems, Techniques, and Applications," In *proc. of IEEE DEBU,* Vol. 18, No. 2, , pp. 3-18, 1995

[5] W. Lee, "On the Independence of Data Warehouse from Databases in Maintaining Join Views," In *proc. Lecture Notes in Computer Science,*, , pp. 86-95, 1999

[6] M. Mohania and Y. Kambayashi, "Making Aggregate Views Self-Maintanable," DKE, vol. 32, No. 1, pp. 87-109, 2000

[7] M. Mohania, S. Konomi, and Y. Kambayashi, "Incremental Maintenance of Materialized Views," In *proc. DEXA,* pp.551-560, 1997

[8] X.Qian and G. Wiederold, "Incremental Recomputation of Active Relational Expressions," *IEEE TKDE*, Vol. 3, No. 3, pp. 337-341, 1991

[9] K. A. Ross, D. Stivastava and S. Sudarshan, "Materialized View Maintenace and Integrity Constraint Checking : Trading Space for Time," In *proc. of ACM SIGMOD conf.*, pp. 447-458, 1996

[10] A. Segev and J. Park, "Updating Distributed Materialized Views," *IEE TKDE*, Vol. 1, No. 2, pp. 173-184, 1989.

[11] J. A. Blakeley, "Updating Materialized Database Views," Research Report CS-87-32, Univ. of Waterloo, 1987.