□ 기술해설 □

차원 모형과 데이타 웨어하우스 설계 기법

동덕여자대학교 최동훈*

1. 소 개

1960년대 데이타베이스 시스템의 출현은 원하는 정보를 쉽게 접근할 수 있는 데이타베이스를 구축하여 정보의 가용성을 증가시켰다. 이들 데이타베이스는 조직을 운용하는 데 필요한 운용(operational) 데이타베이스로 주로 구축되었다. 운용 데이타베이스의 구축은 각 주요 조직을 중심으로 이루어졌으며 데이타베이스의 내용도 현재 상태를 위주로 유지되었다. 그러나 최근 들어 의사결정 지원 시스템(decision support system : DSS)이나 관리 정보 시스템(executive information system : EIS)의 필요성이 증가함에 따라 데이타베이스에서도 이러한 시스템을 효율적으로 지원하기 위한 새로운 요구사항이 나타나게 되었다. 즉, DSS나 EIS에서 요구하는 요약, 분석 작업을 통해 의사 결정에 필요한 정보를 사용자에게 효율적으로 제공하기 위해 과거로부터 누적된 이질적이고 방대한 양의 데이타를 통합 관리할 필요성이 증가하게 되었다. 기존에는 이러한 요구사항을 운용 데이타베이스에서 직접 연산하여 필요한 정보를 제공해왔으나 필요한 정보를 추출하기까지에는 엄청난 비용과 시간을 필요로 하거나 경우에 따라서는 불가능할 정도였다.

데이타 웨어하우스는 이들 문제에 대한 해답을 제공한다. 데이타 웨어하우스는 현재 상태의 데이타뿐만 아니라 과거의 데이타도 유지하므로, 누적된 통합 데이타를 분석하여 필요한 정보를 추출하는데 효과적이다.

* 정회원

1.1 데이타 웨어하우스의 정의

데이타 웨어하우스는 의사 결정을 지원하기 위한 데이타베이스로 다음의 특징[1]을 갖는다.

• 주제 중심적(subject-oriented) 구성

운용 시스템이 업무 처리 중심으로 구성되어 있는 것과 대조적으로, 데이타 웨어하우스는 각 조직의 주요 주제를 중심으로 데이타를 구성한다. 예를 들어 은행 시스템의 운용 데이타베이스가 대부, 저축, 은행 카드, 신용 등으로 구성된 반면, 데이타 웨어하우스는 고객, 상품 등에 관한 추세를 분석할 수 있도록 고객, 상품, 활동 등 주제 중심으로 구성된다.

업무 처리 중심과 주제 중심의 차이는 데이타의 내용의 차이로 나타난다. 운용 데이타베이스는 의사 결정에는 필요하지 않더라도 업무 처리에 필요한 데이타는 모두 관리의 대상이 되지만, 데이타 웨어하우스는 의사 결정에 필요한 데이타만을 유지한다.

운용 데이타베이스와 데이타 웨어하우스는 데이타간의 관계에서도 큰 차이가 난다. 운용 데이타베이스에서 데이타간의 관계는 업무 처리상의 상관 관계에 의해서 테이블간의 관계가 유지되지만, 데이타 웨어하우스에서는 시간을 비롯한 여러 분석 관점에 따른다.

• 통합된(integrated) 내용

데이타 웨어하우스의 데이타는 통합되어 있다. 여기서 통합이란 속성의 이름, 자료의 표현, 도량형의 단위 등이 일관성이 있음을 뜻한다. 예를 들어 길이를 cm, inch, feet등 여러 단위로 나타낼 수 있고, 날짜도 yymmdd, mmddyy 등 다양할 것이다. 이와 같이 필요에

따라 다른 표현, 단위 등이 데이타 웨어하우스에서는 분석 및 비교를 위해 일관적이다.

- 시간에 따라 변화되는(time-variant) 값의 유지

운용 데이타베이스는 접근하는 현재 시간을 기준하여 최신의 값을 유지하지만, 데이타 웨어하우스는 시간에 따라 모든 순간의 값을 유지한다. 따라서, 운용 데이타베이스는 필요할 때마다 갱신되지만 데이타 웨어하우스는 일단 일련의 스냅 샷(snapshot)으로 올바르게 기록되면 갱신되지 않는다.

- 비갱신성(non-volatile)

운용 데이타베이스에서는 레코드별 갱신이 자주 발생하지만 데이타 웨어하우스에서의 데이타 갱신은 초기 적재 이후에는 발생하지 않고 검색만이 있을 뿐이다. 따라서 데이타 웨어하우스를 설계할 때는 갱신 이상(update anomaly)을 염려할 필요가 없기 때문에 정규화와 비정규화에 관한 문제는 접근의 효율성만을 고려하여 자유롭게 선택된다. 또한 장애 발생에 대한 데이타의 복구, 트랜잭션과 데이타의 무결성 유지, 교착 상태의 탐지와 처치 등이 매우 간단하다.

1.2 데이타 웨어하우스와 운용 데이타베이스의 차이

데이타 웨어하우스와 운용 데이타베이스는 사용자의 요구 사항과 데이타의 시간의 취급 측면에서 매우 다르다.

운용 데이타베이스의 사용자는 조직의 활동을 운용하며 하루에도 수십만에서 수백만 건의 트랜잭션 처리를 요구한다. 이들 트랜잭션은 대부분의 경우 데이타베이스에서 하나의 레코드를 검색하여 데이타를 갱신하거나 새로운 데이타를 생성한다. 운용 데이타베이스에 대한 질의나 보고서는 테이블을 리스트하는 것이기 때문에 의사 결정에 필요한 정보를 추출하는데 상당한 시간이 걸린다. 게다가 운용 데이타베이스는 현재를 위주로 유지되므로 전년 대비 분석은 거의 불가능하기가 쉽다.

반면에 데이타 웨어하우스의 사용자는 조직 활동의 상황을 한 두 페이지의 전년 대비 보고서나 경향 분석을 위한 요약 정보를 주로 원한다. 이러한 요구는 수만에서 수백만 레코드의 검색을 포함하며, 발생 빈도는 운용 시스템의 트랜잭션에 비하면 매우 적으나 처리 시간은 상당히 길다.

데이타 웨어하우스에서 질의의 성능은 운용 시스템과 마찬가지로 중요하지만 특성은 상당히 다르다. 데이타 웨어하우스의 질의는 여러 테이블을 조인해야 하기 때문에 짧은 시간 안에 결과가 나오지 않아도 괜찮다. 물론 조인 질의가 수 백만 개의 레코드를 집계(aggregate)해야 하는 경우 처리 시간이 상당히 길어질 수 있으나, 집계 결과를 저장하여 질의 처리 성능을 개선한다.

데이타 웨어하우스는 시간을 운용 데이타베이스와 매우 다르게 취급한다. 운용 데이타베이스는 트랜잭션이 발생할 때마다 지속적으로 갱신되어 최신의 스냅 샷만을 보존한다. 이 때 업무에 중심적인 데이타는 시시각각으로 변한다. 데이타 웨어하우스는 전년 대비, 전월 대비 등과 같은 요약 및 분석 처리를 지원해야 하므로 과거의 데이타도 체계적으로 유지해야 한다. 이때 시간은 연속적으로 데이타에 명시적으로 포함된다.

데이타 웨어하우스는 주제 중심적인 구조를 가지며, 데이타와 함께 시간을 저장한다. 사용자는 데이타를 분석하고자 하는 관점에서 시간에 따른 경향 분석을 요구한다. 또한 내용 중에 좀더 상세하게 또는 개략적인 데이타를 얻기 위한 요구를 지속할 수 있다. 이와 같은 데이타의 특성과 사용자의 요구를 효과적으로 수용하기 위해서는 데이타 웨어하우스의 구조를 설계하는 새로운 방법을 필요로 한다.

이에 대한 대안적 데이타 모형으로 차원 모형이 있다. 이 글에서는 차원 모형을 이용한 데이타 웨어하우스의 설계 방법에 대해 설명할 것이다.

이 글의 나머지 부분은 다음과 같다. 2절에서 차원 모형을 설명하고, 3절에서 차원 모형(dimensional model)을 이용한 데이타 웨어하우스의 설계 기법과 설계의 예를 보여준다. 4절에서 데이타 웨어하우스에 대한 질의 처리 효율을 개선하는 집계 결과의 저장 방법을 제시하고, 5절에서 결론을 맺는다.

2. 차원 모형

2.1 차원 모형의 필요성

운용 시스템은 트랜잭션 처리 성능이 가장 중요하기 때문에 데이타베이스를 설계할 때 개체-관계 모형(entity-relationship model)을 사용하고 데이타의 의존성(data dependency)을 이용하였다. 이들은 데이타의 중복을 제거하는 데 효과적이고, 데이타의 중복이 없으면 갱신 트랜잭션은 한 군데의 데이타만을 접근하면 되기 때문에 트랜잭션 처리 속도에 절대적인 기여를 해왔다.

개체-관계 모형화는 데이타를 여러 개의 이산적인 개체로 나누고 각 개체를 하나의 테이블로 정의한다. 예를 들어 판매 주문 데이타베이스에서 각 주문을 개체로 설정하고 관계 있는 개체를 추가해 가면서 그림 1과 같이 복잡한 스키마로 모형화 된다. 물론 이 그림은 판매 주문의 모든 면을 포함한다. 인접한 테이블간에는 서로 연결되므로 떨어져 있는 두 테이블간에는 연결 경로가 매우 복잡하고 다양하다. 이와 같은 스키마는 데이타 웨어하우스의 질의 처리에 관한한 다음과 같은 문제점을 포함하고 있다.

스키마가 개체 및 개체간의 관련성, 데이타의 의존성 등 데이타의 특성을 중심으로 설계되면 테이블간의 중요도에 차이가 없고, 많은 수의 테이블이 포함되므로 복잡하다. 그림 1은 판매 업체의 운용 데이타베이스의 스키마이다. 여기서 노드는 테이블을 나타내며, 링크는 조인 경로를 나타낸다. 이러한 스키마를 사용자 관점에서 보면, 원하는 중요한 데이타가 여러 개체에 흩어져 있기 때문에 이들을 모으려면 여러 번의 조인을 포함하는 질의를 작성해야 한다. 복잡한 스키마는 이해하기에 복잡하며, 사용자의 관점에 따른 질의 작성을 무척 어렵게 한다.

스키마가 매우 복잡하여 질의에 인접하지 않은 두 테이블이 포함된다면 그들 간의 연결 경로는 매우 많다. 예를 들어 그림 1에서 계약과 상품간의 연결 경로를 생각해 보면 여러 경로를 통해서 같은 답을 얻을 수 있음을 알 수 있다. 각 경로는 비용이 서로 다르고 매우 다양

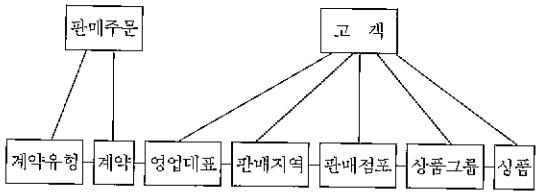


그림 1 판매 업체의 운용 데이타베이스 스키마

하기 때문에 최적 경로를 찾기가 무척 어렵다. 최선의 경로를 선택하더라도 경로에 따라 조인 연산을 해야 하므로 응답을 얻으려면 여러 번의 조인 연산을 해야 한다. 연속적인 조인 연산은 응답 시간을 크게 지연시키므로 개체-관계 모형에 따른 데이타베이스 탐색은 여러 번의 조인을 수행해야 하는 질의의 처리 성능을 크게 떨어뜨린다.

데이타 웨어하우스는 질의를 중심으로 사용되기 때문에 위에 나열한 문제점을 가진 개체-관계 모형은 데이타 웨어하우스의 설계에 부적합하며, 이에 대한 대안으로 차원 모형[2]이 있다. 차원 모형은 사용자 관점에서 스키마를 작성하며 업무상 중요한 수치 데이타와 이들을 설명하는 설명 데이타를 다른 테이블로 분리하여 설계한다. 이러한 구조는 사용자가 중요하게 다루는 정보를 효과적으로 보존하고 볼 수 있도록 한다. 또한 테이블간의 경로도 간단해서 사용자가 질의를 쉽게 작성할 수 있다.

2.2 차원 모형의 정의

차원 모형은 사용자 중심으로 중요한 데이타를 구별하여 하나의 테이블에 모아 놓기 때문에 개체-관계 모형과 달리 스키마에서 테이블간의 중요도가 서로 다르다. 사용자의 관점에서 중요한 데이타를 포함하는 테이블을 스키마의 중심에 놓고, 이들을 분석하기 위한 관점을 나타내는 테이블을 주변에 배치한다. 중심 테이블을 사실(fact) 테이블이라 하고 주변의 테이블을 차원(dimension) 테이블이라 부른다. 여기서 차원이란 사용자가 데이타를 분석하고자 하는 주요 분석 요인(factor)을 의미하며, 차원 테이블은 이러한 목적을 충족시키기 위한 정보를 유지한다.

사실 테이블은 사용자가 중요하다고 생각하는 모든 데이타를 저장하므로 차원 테이블과

비교할 수 없을 정도로 크며, 차원 테이블은 사실 테이블의 내용을 보조하는 데이타를 포함하여 크기가 매우 작다. 차원 데이타베이스의 스키마는 하나의 매우 큰 중심 테이블과 주변의 작은 테이블로 구성된 스키마의 모양이 별 모양인 것에서 유래하여 스타 조인 스키마(star join schema)[3]라고도 한다. 스타 스키마는 중요한 데이타가 하나의 테이블에 모여 있기 때문에 직관적으로 이해할 수 있고 질의 작성도 쉽다.

그림 2는 판매 업체의 판매 실적을 분석하기 위해 차원 모형을 사용하여 사업 활동을 추상(abstraction)한 것으로, 판매 실적 테이블이 사실 테이블이고, 시간 차원, 상품 차원, 매장 차원들이 차원 테이블이다. 그림 1과 비교하면, 그림 2가 단순하고 판매 실적 테이블을 중심으로 판매 현황을 직관적으로 알 수 있게 한다.

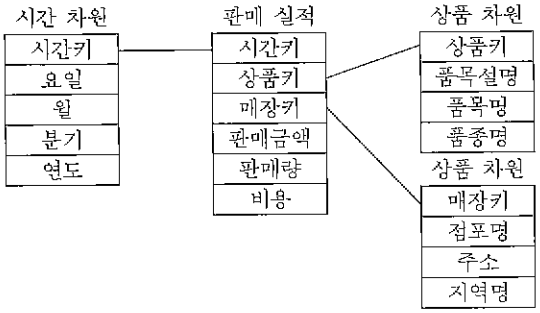


그림 2 판매현황에 대한 스타 스키마

2.2.1 사실 테이블

사실 테이블은 차원 데이타베이스에서 가장 큰 테이블로 조직의 활동에 의해 발생한 중요한 수치를 사실(fact)로 저장한다. 각 사실은 모든 차원의 교차점에서 취해진 값이며, 이러한 이유로 사실 테이블의 주키(primary key)는 모든 차원을 결합한 결합 키(combined key)이다. 이러한 결합 키는 차원 테이블과 조인할 수 있는 유일한 수단이다. 예를 들어, 그림 2에서 사실은 판매 실적 테이블의 판매 금액, 판매량, 비용이며, 이들은 매일 각 매장에서 상품을 판매한 결과를 시간, 매장 상품의 키와 함께 저장한다. 만일 판매 실적 테이블에서 x일에 y매장에서 z상품을 판매한 사실을 찾으려면, x, y, z를 결합한 형태의 결합키가 필요하다.

사실 테이블의 사실은 연속적인(continuous) 이고 가산적인(additive) 값으로 덧셈이 가능하다. 따라서, 대단히 많은 수의 레코드로부터 사실을 사용자 요구에 맞도록 계산하여 수십 개의 레코드로 요약하여 답을 제공할 수 있다.

2.2.2 차원 테이블

차원 테이블은 기업 활동의 결과에 대해 사용자의 주요 분석 요인(factor)을 서술한다. 예를 들어, 사용자가 날짜별로 매장에 따라 특정 상품의 판매 실적을 분석하고자 하면, 판매 실적에 대한 차원은 시간, 매장, 상품 등이 될 것이다. 차원의 각 레코드는 특정 상품을 나타내며 많은 속성으로 구성된다. 속성의 자료형은 문자형이며 이산형으로, 속성은 사용자의 응답의 표제나 질의 제한 조건으로 사용된다.

차원 속성은 차원의 특징을 계층적으로 서술하기도 한다. 예를 들어 그림 2의 상품 차원에서 상품은 품목, 품종 등의 계층 구조로 분류된다. 사용자는 이러한 계층 구조를 따라 품목별 판매 금액을 집계하여 실적을 분석한다. 차원 테이블 속성의 중요한 역할은 질의에서 제한 조건으로 사용되거나 사용자의 응답에서 표제로 사용되는 것이다.

데이타 웨어하우스의 질의는 일별 품목에 대한 판매 실적을 분석하면서, 분기별, 월별, 연도별 판매 현황 또는 판매 비교 등을 요구하기도 하고, 품목이 아니라 품종에 대해서도 판매 현황을 요구한다. 이와 같이 품목에서 품종으로, 날짜에서 분기 또는 년도로 점점 추상적인 수준의 판매 현황을 요구하는 것을 drill-up이라 하며, 반대로 상세한 수준의 정보를 요구하는 것을 drill-down이라 한다. Drill-down과 drill-up은 차원 테이블에 정의된 차원의 계층 구조를 따라 진행된다.

3. 차원 데이타 웨어하우스의 설계

3.1 설계 기법

차원 모형을 이용한 차원 데이타 웨어하우스의 설계는 이론적인 근거가 있는 것이 아니고 매우 단순하며 사용자의 요구 사항을 만족하도

록 다음의 과정[2]을 따라 진행된다.

① 처리 요구 사항을 정의하고 처리의 중심 대상을 사실 테이블로 정의함.
② 사실 테이블에 저장할 중요 데이타를 결정함.
③ 사실 테이블의 데이타를 분석할 주요 요인을 차원으로 결정함.
④ 사실 테이블에 대한 집계 사실의 수준을 결정함.
⑤ 차원의 설명을 포함하여 차원의 속성을 정의함
⑥ 집계 사실의 저장 방법을 결정함.

처리 사항을 분석하면, 처리에 중요한 사실 테이블들이 각 처리 사항으로부터 설정될 것이다. 사실 테이블이 상세하게 설계되기 전에 사실 테이블의 레코드가 무슨 내용이어야 할지, 즉 단위 정보를 정의하여야 한다. 전형적인 단위 정보는 하나 하나의 거래 내용, 일별 스냅샷 또는 월별 스냅 샷 등이다.

사실 테이블의 단위 정보가 결정되면 차원을 식별하고 차원의 단위 정보도 정의할 수 있다. 차원의 선택은 사실(fact)의 분석이 차원에 의존하므로 아주 중요하다. 차원이 결정되면 사실 테이블을 위한 수치 사실을 찾고, 마지막으로 차원 레코드의 속성을 채워 넣는다.

데이타 웨어하우스를 설계할 때 수치 데이타를 사실 테이블의 사실로 해야 할지, 차원 테이블의 속성으로 해야 할지 불명확할 때가 있다. 이 때 값의 특징을 기준으로 결정한다. 만일 수치가 시시각각으로 연속적으로 변하면 사실로 정하고, 이산적이고 시간에 따라 변하지 않으면 속성으로 정의한다. 예를 들어 상품의 비용은 상수 값으로 보이지만 종종 변하므로 사실로 결정한다. 가끔 이러한 분류가 불확실한 경우도 있어서 설계자의 관점에서 결정하기도 한다.

3.2 데이타 웨어하우스 설계의 예[2]

이 절에서는 여러 지역에 여러 소매점을 포함하는 소매 연쇄점에 대한 차원 데이타 웨어하우스를 설계 절차에 따라 설명한다. 문제를 단순화하기 위해 집계 사실의 설계에 대한 부분은 생략하기로 한다.

3.2.1 처리 사항과 사실 테이블의 식별

본사의 관리자는 모든 소매점에서 하루에 발생한 판매 활동을 분석하려고 한다. 즉 날짜별로 어느 상품이 어느 매장에서 얼마에 얼마나 팔렸는지를 자세히 알고 싶어한다. 그리고 이들로부터 매장별 월별 상품 판매량 등을 요약 및 분석하려고 한다. 이러한 처리 사항을 수행하려면 관리자가 중요하게 여기는 데이타는 판매 실적에 대한 것이며, 따라서 사실 테이블은 판매 테이블이 된다.

3.2.2 사실 테이블의 단위 정보

판매 테이블에는 기본적으로 매장별 날짜별 상품의 판매량, 판매 금액, 비용 등을 포함한다. 이들로부터 지역별, 월별, 품종 판매 실적이 집계될 것이다.

3.2.3 차원 테이블의 결정

직관적으로 판매 사실은 시간(일, 월, 분기, 회기 년도), 상품(품목, 품종), 매장(점포, 지역)에 따라 분석되어야 하므로, 시간, 상품, 매장의 세 개의 차원을 쉽게 찾을 수 있다. 한편, 할인 판매, 할인권 발행, 진열대의 재배치 등의 판촉 활동도 판매 실적에 영향을 미치며, 이들 판촉 활동의 영향 분석을 위해 판촉도 새로운 차원으로 설정되어야 한다.

만일 단골 고객의 구매력이 실적에 큰 영향을 미치는 사업 활동에 대해 차원 데이타 웨어하우스를 설계한다면, 고객의 구매 성향 분석과 이들의 영향력을 영업에 활용하기 위해 고객을 하나의 차원으로 설정해야 할 것이다.

3.2.4 차원의 속성 결정

시간 차원은 시간키와 함께 날짜, 월, 분기, 회기 년도, 요일, 공휴일, 계절 등이 포함된다. 요일이나 공휴일, 계절 등은 평일과 주말, 계절별 판매 비교가 가능하므로 중요하다. 회기 년도는 전년 대비 분기별 실적 평가를 가능하게 한다. 상품 차원은 상품키와 함께 품목, 품종, 포장 유형, 크기, 무게, 색깔 등이 포함된다. 매장 차원은 매장키, 점포명, 주소, 도시, 판매 지역, 점포 관리자, 전화 번호, 점포 넓이, 시설 등을 포함한다. 판촉 차원은 판촉키, 판촉명,

할인 유형, 할인권 유형, 진열 상태, 판촉 비용, 판촉 기간(시작, 끝) 등을 포함한다.

이들 차원에는 일-월-분기-년도, 품목-품종, 점포-지역의 계층 구조가 있다. 이들 계층 구조는 좀더 개략적인 정보(drill-up)와 좀더 상세한 정보(drill-down)를 효과적으로 추출하는 데에 기여한다.

위의 절차에 따른 스타 스키마는 그림 3과 같다.

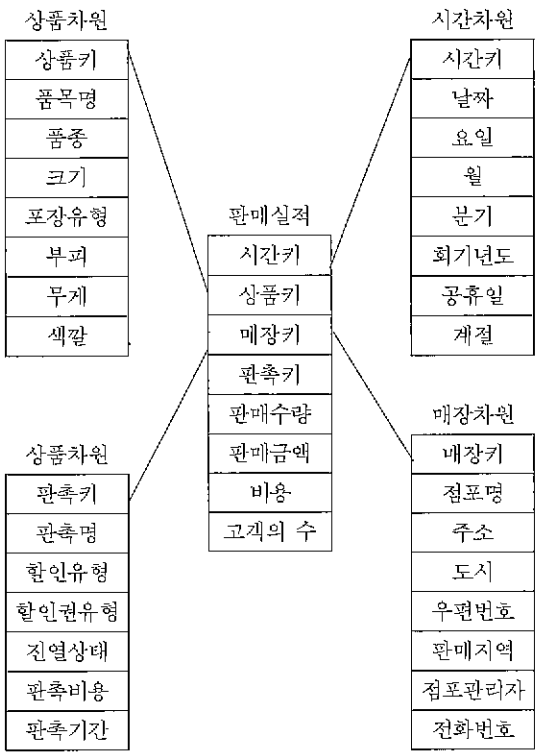


그림 3 판매실적을 위한 스타 스키마

4. 집계 사실의 저장 방법

차원 데이타 웨어하우스에서 사실 테이블은 상세한 수준의 단위 정보에 해당하는 엄청난 수의 기초 수준의 레코드, 즉 기초 레코드를 포함하고 있다. 사실 테이블이 기초 레코드만을 가지고 있다면 질의를 처리할 때 수많은 기초 레코드를 읽어서 요약을 해야 하기 때문에 처리에 오랜 시간이 필요하다. 그러나, 기초 레코드를 요약한 집계 사실을 저장하여 질의를 처리할 때 사용하면 질의 처리 시간을 훨씬 줄일 수 있다.

이러한 집계 사실로 인한 질의 처리의 성능 향상은 수 백배 내지 수 천배 이상의 개선을 기대할 수 있다. 따라서, 데이타 웨어하우스의 설계에서 집계 사실의 저장은 매우 중요한 설계 원칙으로 다루어야 한다.

새로운 집계 사실과 집계 차원을 저장하고 관리하는 방법에는 두 가지가 있다. 하나는 별도의 사실 테이블을 생성하는 것이고, 다른 하나는 수준 항목을 기존의 차원 테이블에 추가하고 기존의 사실 테이블에 집계 사실을 저장하는 것이다.

4.1 집계 사실을 위한 새로운 테이블의 생성[2]

집계 사실 테이블을 별도의 사실 테이블로 저장하면 많은 집계 사실 테이블이 새로 생성되어 스키마가 복잡해지지만, 이 방법은 다음과 같은 장점이 있다.

- 집계 사실이 따로 저장되므로 스키마 상에서 집계 사실 레코드를 기초 사실 테이블과 구분할 필요가 없다.
- 집계 항해자(aggregate navigator)를 이용하여 최종 사용자와 응용 개발자는 집계 사실 테이블이 별도로 저장되어 있다는 사실을 알 필요가 없다. 집계 테이블은 집계 항해자와 데이타 관리자만 알고 있으면 된다. 집계 항해자는 질의가 들어오면 적절한 집계 테이블을 선택하는 소프트웨어이다.
- 서로 다른 집계 수준에 의한 집계 사실 테이블을 서로 다른 시간에 저장하기 때문에, 데이타베이스를 점진적으로 관리할 수 있다.
- 집계 항해자는 메타 테이블의 집계 방법에 대한 상세한 내용을 탐색하지 않고 사용 가능한 데이타베이스 시스템의 시스템 테이블을 탐색하여 집계 테이블을 선택하기 때문에 메타 테이블을 단순하게 만든다.

4.2 집계 사실을 위한 수준 항목의 추가 [2]

집계 사실을 저장하는 또 하나의 방법은 차원 테이블에 수준 항목을 사용하여 집계 사실을 새로운 차원 키와 함께 사실 테이블에 저장하는 것이다(그림 4 참조). 예를 들어, 과자류에 대한 집계 사실을 판매 실적에 저장하고, 상품 차원에 수준 항목을 추가하여 수준의 값으로 과자류라는 품종을 명시한다. 이렇게 하면, 과자류에 대한 집계 사실은 상품 차원 테이블을 통해 사실 테이블의 집계 사실을 찾을 수 있다. 이 방법은 앞의 방법과 마찬가지로 같은 수의 집계 레코드를 생성하고, 차원 테이블과 사실 테이블에 집계 키를 생성한다.

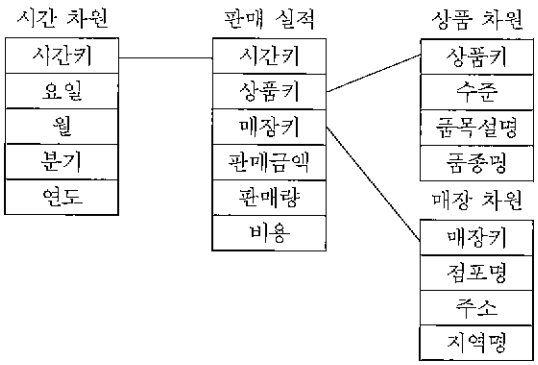


그림 4 수준 항목을 이용한 경우

이 방법의 가장 큰 문제는 만일 질의에 항목에 대한 정의가 잘 되어 있지 않으면 집계 사실뿐만 아니라 기초 사실도 검색될 수 있다는 것이다. 이러한 경우를 방지하는 것은 매우 까다롭기 때문에[2] 이 방법보다는 집계 사실을 별도의 집계 사실로 저장하는 것이 좋다.

5. 결론 및 요약

90년대 들어 데이타 웨어하우스는 정보 시스템의 중심적 역할을 하고 있다. 이 글의 내용을 요약하면 다음과 같다.

데이타 웨어하우스는 의사 결정 지원을 목적으로 기업 활동 분석에 필요한 데이타를 과거로부터 축적 및 유지하므로 데이타의 생성 시간을 명시적으로 기록한다. 이들 데이타에 대한 분석은 여러 차원을 근거로 이루어지며, 차원 분석과 함께 데이타를 효과적으로 관리하려면 차원 모형의 스키마를 사용하는 것이 바람직하다. 데이타 웨어하우스에 대한 사용자의 질의는 사용자의 관점에서 데이타를 검색 및 분석한다. 차원 모형은 사용자 중심으로 중요한 데이타와 그들에 대한 보조 데이타 등으로 구별하여, 사용자가 중요한 데이타를 직관적으로 구별 가능하다. 테이블을 업무상의 변화하는 중요한 수치를 갖고 있는 테이블과 이에 대한 고정적인 설명을 포함하는 테이블로 구분하면, 스키마를 이해하기가 쉬워서 사용자가 질의를 작성하는데 큰 어려움이 없다. 차원 모형을 이용한 데이타 웨어하우스의 설계 기법은 매우 단순하다. 사용자의 처리 사항으로부터 사실 테이블을 결정하고, 사실 테이블의 데이타를 분석하는 주요 요인을 차원으로 한다. 차원 데이타 웨어하우스에 대한 사용자의 질의는 수많은 레코드의 집계를 포함한다. 따라서, 기초 수준의 레코드를 미리 집계하여 집계 사실을 저장하면 질의 처리 성능을 개선할 수 있다.

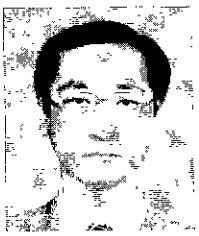
이 글에서는 집계 사실의 수준을 결정하고, 집계 사실의 크기를 예측하는 문제에 대한 설명은 지면 관계상 제외되었다. 이들 문제는 데이타 웨어하우스의 설계에서 가장 중요하고 실제적인 것으로 참고문헌 [2]를 참조하기 바란다.

참고문헌

[1] W.H. Inmon and R.D. Hackathorn, *Using the Data Warehouse,* John Wiley & Sons, 1994.

[2] R. Kimball, *Data Warehouse ToolKit,* 1996.

[3] Red Brick System, "Star Schemas and STARjoin Rechnology", *Red Brick Systems White Paper,* 1996.



최 동 훈

1981 서울대학교 계산통계학과 (학사)
1983 한국과학기술원 전산학과 (석사)
1989 Northwestern University (박사)
현재 동덕여자대학교 전자계산학과 부교수
관심분야 : 데이타베이스

● ISAAC '97 학술대회 논문모집 ●

Eighth Annual International Symposium on Algorithms and Computation

- 응모분야 : 알고리즘과 계산이론에 관련된 모든 분야
- 일 자 : 1997. 12. 17~19
- 장 소 : 싱가폴
- 제출마감 : 1997. 5. 30
- 제 출 처 : Hon Wai Leong(Emall : leonghw@lscs.nus.sg)
  Department of Information Systems and Computer Science
  National University of Singapore, Singapore 119260
- 문 의 처 : Hon Wai Leong