OLAP을 위한 객체-관계 DBMS 기반 다차원 데이타 모델의 설계 및 구현

정회원 김 은 영*, 용 환 승**

Design and Implementation of Multidimensional Data Model for OLAP Based on Object-Relational DBMS

Eun-Young Kim*, Hwan-Seung Yong** *Regular Members*

요 약

OLAP(On-Line Analytical Processing) 기법에서 스타 또는 눈송이(snowflake) 스키마에 기반한 ROLAP(Relational OLAP)은 성능 저하라는 문제가 있고, 다차원 데이터베이스에 기반한 MOLAP(Multidimensional OLAP)은 데이터 크기 증가에 따른 공간 문제가 있다.

본 논문에서는 기존의 OLAP 시스템이 이러한 문제점을 해결하기 위해서 객체-관계 DBMS에 기반한 다차원 데이터 모델을 제안하였다. 객체-관계 DBMS가 가지는 확장성 특징을 사용하여 다차원 데이터 모델에 최적화된 다차원 개념과 함수를 정의할 수 있었다. 또한 객체-관계 DBMS의 객체간 계승 기능을 통하여 상위 테이블을 계승받는 요약 다차원 데이터 큐브의 다차원 데이터 모델을 설계하였다. 이와 같은 OLAP을 위한 데이터 타입과 함수가 정의되면, 새로운 객체-관계 DBMS 엔진과 같이 내장된 기능처럼 동작되어 성능 향상이 가능하다. 또한 객체-관계 DBMS의 하나인 Informix Universal Server와 클라이언트 개발 도구를 이용하여 제안된 다차원 데이터 모델을 구현하였다.

ABSTRACT

Among OLAP(On-Line Analytical Processing) approaches, ROLAP(Relational OLAP) based on the star, snowflake schema which offer the multidimensional analytical method has performance problem and MOLAP (Multidimensional OLAP) based on Multidimensional Database System has scalability problem.

In this paper, to solve the limitations of previous approaches, design and implementation of multidimensional data model based on Object-Relation DBMS was proposed. With the extensibility of Object-Relation DBMS, it is possible to advent multidimensional data model which more expressively define multidimensional concept and analysis functions that are optimized for the defined multidimensional data model. In addition, through the hierarchy between data objects supported by Object-Relation DBMS, the aggregated data model which is inherited from the super-table, multidimensional data model, was designed. Once these data models and functions are defined, they behave just like a built-in function, with the full performance characteristics of Object-Relation DBMS engine.

* 한국 SAS(주)(koreyk@seoul.kor.sas.com),
** 이화여자대학교 컴퓨터학과(hsyong@ewha.ac.kr)
논문번호 : 99348-0830, 접수일자 : 1999년 8월 30일
※ 이 연구는 1996년도 이화여자대학교 교내연구비 지원에 의한 연구임.

Ⅰ. 서 론

최근 들어 데이타베이스의 합리적 관리와 신속한 의사결정 지원을 위한 각종 방안이 심도 있게 논의되고 있는 가운데 특히 최종 사용자로 하여금 기업 내에 방만하게 흩어져 있는 데이타에 손쉽게 접근하고 활용하도록 하는 데이타 웨어하우스(Data Warehouse) 개념이 전 세계적으로 주목을 받고 있다. 이러한 데이타 웨어하우스의 핵심은 온라인 분석 처리(On-Line Analytical Processing: OLAP) 시스템으로, 이는 전사적으로 통합된 방대한 양의 데이타를 갖는 데이타 웨어하우스 환경에서 사용자의 전략적 의사결정 지원을 위한 분석적이고 다차원적인 정보를 제공한다[1-6, 8].

이러한 OLAP은 기존의 온라인 트랜잭션 처리(On-Line Transaction Processing: OLTP) 시스템과는 대별되는 특징을 가진다. 즉 OLTP가 처리하는 트랜잭션은 대부분 그 길이가 짧고 적은 수의 레코드를 대상으로 하는 조회나 삽입 등이 주된 연산이므로 트랜잭션 성능의 향상을 위해 개체-관계 모델(Entity-Relationship Model: ER-모델)을 기반으로 하는 데이타베이스 시스템이 연구 개발되어져 왔다. 그러나 이러한 OLTP와는 다르게 OLAP은 하나의 트랜잭션이 여러 테이블간의 조인을 통해 수천 내지 수십만 개의 레코드를 대상으로 다차원적이고 복합적인 질의를 수행하게 되므로 기존의 ER-모델에 기반한 RDBMS의 SQL로는 이를 효과적으로 처리할 수 없다[6,8,30].

따라서 OLAP의 다차원 분석을 위해서 지금까지 크게 두 가지 구조가 제안되고 있다. 물리적으로 다차원 데이타 모델을 RDBMS의 전위 (Front-End)에 외부 캐쉬 형태로 저장하고 질의를 수행하는 다차원 데이타베이스 접근법(Multidimensional OLAP: MOLAP)과 기존의 RDBMS를 기반으로 하여 관계형 테이블을 차원 모델링을 통하여 다차원적으로 재 구성하고, 다차원 분석 질의를 수행하도록 하는 관계형 온라인 분석 처리(Relational OLAP: ROLAP) 시스템이 연구되고 있다[3,6,7].

그러나 전자의 경우 다차원 배열을 기반으로 하기 때문에 불필요한 항목의 발생에 따른 저장 공간의 낭비와 원천 데이타에 대한 조회가 어렵다. 특히 차원이나 항목의 추가에 따라 저장공간이 기하 급수적으로 늘어나게 되어 결과적으로 전체 다차원 데이타베이스를 재구성해야 한다.

또한 후자의 경우 데이타의 차원 모델링이 복잡하며 차원 모델링에 기반한 다차원 질의는 조인의 횟수를 최소화 시킬 수는 있으나 애플리케이션에 대한 확장성과 유연성을 제한하게 된다. 특히 표준 SQL로는 비교 연산은 물론 행 연산이 불가능하므로 다차원 질의를 적절한 표준 SQL로 변환을 통한 질의의 수행은 RDBMS 서버의 성능을 충분히 활용할 수 없다[6,8,30].

특히 지금까지의 수치 데이타 중심의 업무용 데이터 처리 응용에 있어 점차 문자열, 문서, 비디오와 오디오 또는 다양한 형태의 이미지 데이타와 같은 복합 데이타에 대한 의사 결정 질의 기능의 필요성이 점차 증대 할 것이다. 따라서 다양한 자료형에 대한 분석 요구에 대해 OLAP시스템은 새로운 데이타 타입에 대한 정의 및 이를 피연산자로 하는 다양한 연산자에 대한 정의를 지원해야 한다[9,10].

본 논문에서는 기존의 OLAP시스템이 지닌 차원 항목에 대한 유연성의 결여 및 분석 질의의 변환에 따른 한계점 등을 보완하기 위해 객체-관계 DBMS의 확장된 성능을 기반으로 다차원 데이터 모델을 설계하고 구현하였다. 즉, 객체-관계 DBMS에서 제공하는 사용자 정의 데이타 타입을 기반으로 다차원 데이타 모델을 정의하고 계층성에 기반한 상속성을 바탕으로 다차원 데이타 모델 형태의 요약 데이타를 구축하는 생성자를 설계하며 객체-관계 DBMS 인 Informix Universal Server를 이용하여 구현하였다[9].

본 논문의 구성은 다음과 같다. 2장에서는 MOLAP과 ROLAP을 비교 분석하고 관련 연구에 대해서 소개한다. 3장에서는 다차원 데이타 모델을 위한 스키마를 정의하고 객체-관계 DBMS를 기반으로 다차원 데이타 모델을 설계한다. 4장에서는 본 논문에서 제안하는 다차원 데이타 모델과 연산자들을 객체-관계 DBMS인 Informix Universal Server를 이용하여 구현한 모델에 대해 서술한다. 마지막으로 5장에서는 앞으로의 연구 방향을 제시한다.

Ⅱ. 관련 기술 및 연구 동향

본 장에서는 데이타의 다차원 분석을 위하여 지금까지 개발된 OLAP시스템의 대표적 구조인 다차원 데이타베이스 접근법(Multidimensional On-Line Analytical Processing: MOLAP)과 관계형 온라인 분석 처리(Relational On-Line Analytical Processing : ROLAP)에 대하여 알아본다.

2.1 다차원 데이타 베이스 접근법 (MOLAP)

다차원 데이타 베이스 접근법(MOLAP)은 다차원 모델에 대한 연구를 바탕으로 출발했으며 다차원 배열을 기반으로 하여 다차원 분석을 위한 효과적인 접근법의 하나로써 연구되고 있다[13]. 다음 그림1은 다차원 데이타베이스로 구성된 다차원 데이타 모델과 이를 기반으로 하는 MOLAP의 질의를 나타낸다.

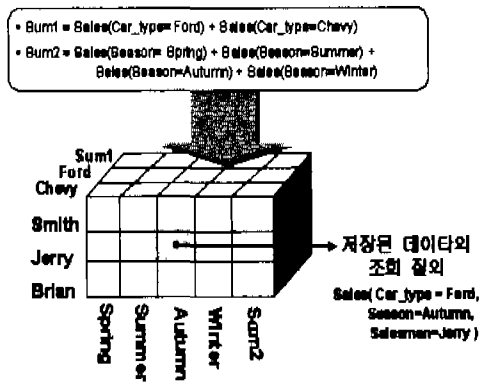


그림 1. MOLAP의 다차원 데이타 베이스의 예

MOLAP을 위한 다차원 데이타베이스는 분석의 기준이 되는 차원을 설정하고 각 차원의 항목들을 설정하여 구축되는데, 위의 예에서는 계절(Spring, Summer, Autumn, Winter)과 차종(Chevy, Ford) , 판매원(Smith, Jerry, Brian) 의 세가지 차원으로 다차원 데이타베이스를 구성하였다. 특히 다차원 데이타베이스의 생성시 차종별 판매량의 합계와 계절별 판매량의 합계를 위한 연산식 Sum1과 Sum2를 정의하여 다차원 데이타베이스의 데이타 로드 시 결과 셀을 생성하게 된다. 따라서 데이타 셀과 연산식에 의한 결과 셀이 다차원 데이타베이스 내에 함께 존재하게 되어 주어진 집계 함수 질의에 대하여 별도의 연산이 필요치 않아 질의의 처리 성능을 향상시킬 수 있다[13,14].

그러나 다차원 데이타베이스를 기반으로 하는 MOLAP의 경우 차원과 항목으로 이루어진 다차원 배열을 기반으로 하기 때문에 불필요한 항목의 발생에 따른 저장구조의 낭비는 물론 데이타 로드 시간이 많이 소요된다. 또한 다차원 데이타베이스에 정의된 데이타에 대한 조회는 빠르나 원천 데이타에 대한 질의가 용이하지 않다. 특히, 차원이나 항목의 추가나 변경에 따른 저장공간은 기하 급수적으로 늘어날 수 밖에 없으며 결과적으로 몇 가지 요소의 변화만으로도 데이타베이스 구조 자체를 재구성해야 하는 유연성의 한계가 있다[8].

2.2 관계형 온라인 분석 처리(ROLAP)

OLTP 시스템 구축을 위해 주로 사용되는 ER-모델은 데이터 웨어하우스의 환경에 부적절하며, 이를 해결하기 위해 비 정규화의 특수한 형태로 차원모델링이 제안된다[15]. 차원 모델링은 사용자의 관점에서 데이타를 사실 테이블(Fact Table)과 차원 테이블(Dimension Table)로 구조화 한다. 사실 테이블은 수치적인 실제 데이타를 저장하고 있으며, 각 사실은 모든 차원의 조합인 결합키(Combined Key)로 이루어진 주키(Primary Key)를 갖는다. 따라서 사실 테이블의 결합키를 통해서 차원 테이블과 조인을 수행할 수 있다. 차원 테이블은 주요 분석 요인을 서술하는데 각 차원은 차원 속성을 통해 계층적으로 구조화 된다. 다음 그림2는 스타 스키마로 데이타 차원 모델링의 예를 보여 준다.

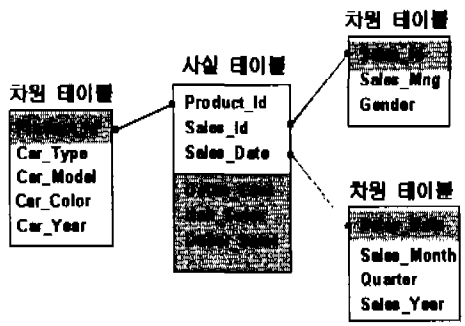


그림 2. ROLAP을 위한 데이터 차원 모델링의 예

그림2에서 Product_Id를 주키로 하는 차원 테이블과 Sales_Id를 주키로 하는 차원 테이블 및 Sales_Data를 주키로 하는 차원 테이블에 대한 각각의 주키를 포함하고 있는 사실 테이블로 차원 모델링이 구성되었다.

이러한 스타 스키마에서 질의는 통상 한 개의 사실 테이블과 여러 개의 차원 테이블을 조인하는 스타 조인을 통해서 처리되는데 가장 보편적인 방법은 질의에 포함된 차원 테이블간의 조인을 통해 중간 테이블을 작성하고 이를 사실 테이블과 조인하는 방식이다. 이때 차원 테이블의 조인에 사용되는 결합 인덱스는 질의 처리 성능에 크게 영향을 주게 되므로 조인을 위한 인덱싱 기법과 이를 기반으로 하는 질의의 최적화에 대한 연구가 활발히 진행되

고 있다[17,20].

또한 사실 테이블에는 단위 정보에 해당하는 기초 레코드와 함께 기초 레코드를 요약한 집계(Aggregation) 사실을 함께 저장하여 질의 처리 성능을 향상시킨다. 즉, 데이타 웨어하우스 환경에서 다차원 분석 질의는 시간성을 가지는 대 용량 데이타에 대한 집계 연산이 기반이 되므로 이러한 집계 연산의 처리가 성능에 크게 영향을 미치게 된다[18,19].

그러나 이러한 ROLAP의 경우 데이타의 차원 모델링이 복잡하며 조인의 횟수를 최소화 시킬 수는 있으나 애플리케이션에 대한 확장성과 유연성을 제한하게 된다. 또한 표준 SQL로는 비교 연산 및 행 연산이 불가능하므로 데이타의 다차원 분석 질의를 표준 SQL로 변환하거나 표준 SQL을 확장 시킨 별도의 OLAP 연산 엔진이 필요하다. 특히, 다차원 질의의 응답 성능을 향상 시키기 위해 데이타의 분할 및 질의의 최적화 기법 등이 요구된다[8,20].

따라서 다음 장에서는 차원간의 분류를 통하여 기존 OLAP 시스템의 한계점을 보완하는 다차원 데이타 모델을 객체-관계 DBMS를 기반으로 설계하는 방안을 기술하고자 한다.

Ⅲ. 다차원 데이타 모델 설계

3.1 차원 항목의 분류

데이타의 다차원 모델은 기본적으로 하나 이상의 차원과 각 차원의 교차점에서 발생하는 값으로 구성된다. 그러나 차원들간의 중요도가 반영될 수 없고 차원 테이블의 수가 많아져서 다차원 질의를 수행하기 위해서는 연속적인 조인이 필요하게 된다[8,15,30].

표 1. 차원 항목의 분류

차원 항목	차원 항목의 속성
시간 속성(T)	시간에 따른 계층성을 갖는 차원
차원(D)	계층성을 포함하지 않는 단순 속성 차원
변수(V)	시간 속성 및 차원간의 결합에 의해 발생하는 수치값 및 사용자가 정의한 연산식의 결과값을 포함하는 차원

본 논문에서는 이러한 차원을 특성에 따라 분류하고 이를 기반으로 기존의 다차원 모델에서 제시한 차원 테이블과 사실 테이블을 결합시킨 다차원 데이타 스키마를 정의한다. 특히, 다차원 분석의 기반이 되는 시간 항목을 별도로 분류하여 시간 속성이라 정의함으로써 시간 항목에 기반한 분석 질의 및 요약 데이타의 생성과 분석에 대한 효율적인 수행을 제공하도록 한다.

본 논문에서 제안하는 다차원 데이타 스키마를 위한 차원의 분류는 다음과 같다.

차원간의 분류를 통한 다차원 데이타 스키마는 다음과 같이 정의된다.

정의 1. 다차원 데이타 스키마

다차원 데이타 스키마 $S = \{ D, V, T \}$ 는 $D = \{ d_i \in D \mid d_i$ 는 다차원 데이타 모델을 구성하는 각 테이블의 단순 속성 $\}$으로 이때 $V = \{ v_i \in V \mid v_i$ 는 수치값 또는 연산의 결과값 $\}$, T 는 계층성을 가지는 시간 속성으로 정의된다.

그림3은 본 논문에서 제안하는 다차원 데이타 스키마를 나타낸다.

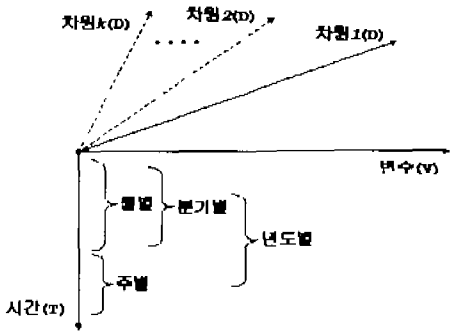


그림 3 차원 분류에 따른 다차원 데이터 모델 스키마

특히 시간 속성(T)은 시간 단위별로 상속을 지원할 수 있도록 구성한다. 즉 주별 요약 데이타와 월별 요약 데이타는 각각 일별 데이터로부터 상속 받아 구성하고 분기별 요약 데이타는 월별 데이타로부터, 년도별 요약 데이타는 월별, 분기별 요약 데이타로부터 다중 상속을 받도록 구성한다.

3.2 다차원 데이타 모델의 설계

본 논문에서 제안하는 다차원 데이타 모델의 스키마를 기반으로 하여 다차원 데이타 모델을 확장된 객체-관계 DBMS의 기능을 이용하여 정의한다. 다차원 데이타 모델의 구조는 다음 그림과 같이 정의된다. 그림 4에 나타낸 객체-관계 DBMS 기반 다차원 데이타 모델 구조. 즉, 객체-관계 DBMS는 내장된 데이타 타입 뿐 아니라 사용자가 새로운 데이

타 타입과 이를 피연산자로 하는 연산식을 정의할 수 있게 하는데 본 논문에서는 다차원 데이타 모델을 위한 새로운 데이타 타입을 정의한다[23].

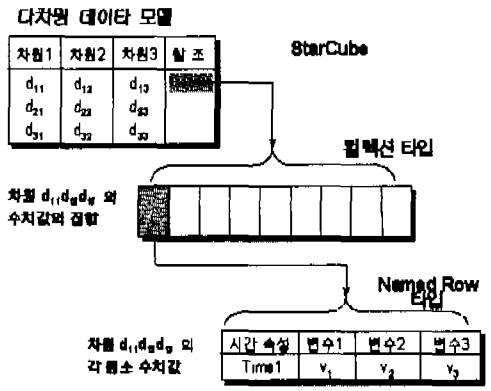


그림 4. 객체-관계 DBMS 기반 다차원 데이터 모델 구조

다차원 데이타 모델은 기존 차원 모델링에서 사실 테이블에 저장 되었던 실제 데이타 즉 차원의 결합에 의해 발생하는 수치값 및 연산식 결과값의 집합을 시간 속성을 기준으로 구조화 시켜 형성된다.

즉, 다차원 데이타 모델의 각 행은 분석의 기반이 되는 단순 속성 차원간의 결합으로 이루어지고 각 차원의 결합은 하나의 참조 칼럼을 가진다. 이러한 참조 칼럼은 차원간의 결합에 의해 발생하는 실제 데이타를 포함하는 Row 데이타 타입의 컬렉션 데이타 타입을 나타낸다. 각 Row 데이타 타입은 시간 속성을 나타내는 시간 칼럼과 실제 수치 값이나 연산식의 결과값을 나타내는 변수의 Named 칼럼들로 이루어 진다. 특히, Row 데이타 타입의 데이타들은 시간에 근거한 분석을 용이하게 하기 위해 시간 칼럼에 따라 순서화 한다.

또한, 시간 칼럼은 본 논문에서 정의한 시간 속성 데이타 타입의 변수 값과 이에 대한 OffSet 값으로 구성 되는데 이는 시간 속성에 근거한 질의 및 집계 연산을 효율적으로 처리할 수 있으며 실제 데이타의 다차원적 개념을 명확하게 표현할 수 있다.

3.2.1 다차원 데이타 모델을 위한 사용자 정의 데이타 타입

객체-관계 DBMS 기반의 다차원 데이타 모델을 위하여 다음의 사용자 정의 데이타 타입을 정의한다[24,25].

Ⅰ. StarCube: StarCube 데이타 타입은 다차원 데이타 모델에서 Row데이타 타입의 컬렉션 데이타 타입으로 정의되는 StarCube라는 부 데이타 타입의 타입 생성의 역할을 한다. StarCube의 결과값은 Degree 정보를 포함하고 있으며 Cast 연산자에 의해서 다양한 형식으로 변환될 수 있다.

Ⅱ. Degree: 다차원 데이타 모델의 시간 속성 항목을 지원한다. Degree 는 Named Row 데이타 타입으로 다음과 같이 구성된다.

표 2. Degree 의 Named Row 구성

Interval	Valid	Invalid	Start
{Day, Week, Month, Quarter, Year}	유효한 연속 기간	유효하지 않은 연속 기간	시작 시간

다차원 데이타 모델은 각각 고유한 Degree 데이타 타입의 변수를 가지며 이는 각 다차원 데이타 모델의 시간 속성 항목의 유효성을 보장한다. 또한 다차원 데이타 모델에 대한 분석 질의 및 요약 데이타에 대한 연산에 이용된다.

예를 들어 Degree 가 ("Month", 5, 1, "01/15 /1997") 로 정의되는 다차원 데이타 모델은 1997년 1월부터 5개월 간의 유효한 데이타가 그리고 6월을 제외하고 다시 5개월간 각 월별 유효 데이타가 존재함을 나타낸다. 따라서 다차원 데이타 모델의 시간 칼럼은 각기 고유한 Degree와 이에 대한

표 3. 다차원 데이타 모델을 위한 시스템 테이블

시스템 테이블	시스템 테이블 속성
DegreeTable	각 행은 Degree이름과 이를 참조하는 Star Cube 및 Degree 데이타 타입의 칼럼으로 이루어진다. 따라서, StarCube에 대한 Degree가 생성될 때마다 Degree Table에 한 행씩 추가된다.
VariableTable	다차원 데이타 모델의 생성자의 피 연산자가 되는 관계형 데이블의 칼럼 중 시간 속성 및 차원간의 결합에 의해 발생하는 수치값 및 사용자가 정의한 연산식의 결과값을 포함하는 차원으로 분류되는 칼럼들에 대한 정보를 포함한다.
Dimension Table	다차원 데이타 모델의 생성자의 피 연산자가 되는 관계형 데이블의 칼럼 중 계층성을 포함하지 않는 단순 속성 차원으로 분류되는 칼럼들에 대한 정보를 포함한다.
SCTable	StarCube에 의해 생성된 다차원 데이타 모델에 대한 정보와 이의 요약 정도를 나타내는 Degree에 대한 정보를 포함한다.

OffSet 값으로 구성되므로 Degree의 Start 칼럼의 값에 대한 OffSet 연산만으로 시간 속성 기반의 분석 질의를 수행하게 된다.

3.2.2 다차원 데이타 모델을 위한 시스템 테이블

다차원 데이타 모델을 위한 사용자 정의 데이타 타입을 지원하기 위한 시스템 테이블을 다음과 같이 정의한다[26].

3.2.3 다차원 데이타 모델을 위한 사용자 정의 연산자

다차원 데이타 모델을 위한 사용자 정의 데이타 타입과 시스템 테이블에 대해 적용되는 사용자 정의 연산자가 표4에 나타나있다. 특히 요약 데이타 생성자 Aggregate에 대해서 다음 장에서 자세히 언급하도록 하겠다.

표 4. 다차원 데이타 모델을 위한 연산자

연산자	연산자 속성
Valid Time	다차원 데이타 모델에 대해서 주어진 시간에 대한 유효한데이타가 존재하는지 검사한다. ValidTime(StarCube_Name Varchar(10), Someday Date) return Boolean; StarCube_Name: 다차원 데이타 모델
Expand	다차원 데이타 모델의 Degree를 주어진 단위 시간으로 변환한다. Expand(StarCube_Name Varchar(10), UnitTime Date) return Degree; StarCube_Name: 다차원 데이타 모델 UnitTime: 시간 단위 Expand 연산자는 다차원 데이타 모델의 요약 데이타를 생성하는 Aggregate 연산자에 이용되어 새로운 Degree를 리턴한다.
Which Degree	다차원 데이타 모델에 대한 Degree를 조회한다. WhichDegree(StarCube_Name Varchar(10)) return Degree; StarCube_Name: 다차원 데이타 모델
Aggregate	다차원 데이타 모델에 대한 요약 데이타를 생성한다. Aggregate(StarCube_Name Varchar(10), UnitTime Date) return StarCube; StarCube_Name: 다차원 데이타 모델 UnitTime: 요약 정도를 나타내는 시간 단위

3.2.4 계층 구조의 상속성에 기반한 요약 데이타 생성자: Aggregate

객체-관계 DBMS는 계층 구조에 기반하여 사용자 정의 데이타 타입과 이에 대한 사용자 정의 연산식에 대한 상위 타입으로부터 하위 타입으로의 상속을 제공한다. 본 논문에서는 이러한 객체-관계 DBMS의 상속성을 바탕으로 다차원 데이타 모델로부터 요약 데이타를 추출하여 동일한 스키마의 다차원 데이타 모델을 생성하는 사용자 정의 연산식 Aggregate를 정의한다. 다음 그림 5는 Aggregate의 질의 처리 과정을 나타낸다.

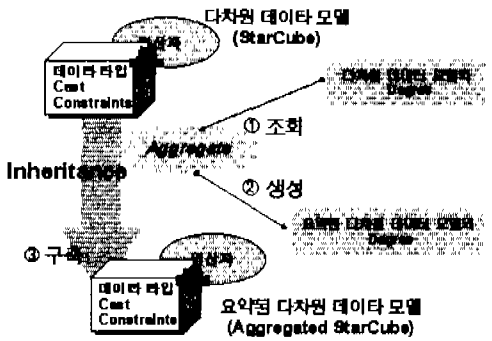


그림 5. 상속성에 기반한 요약 데이터 생성자 Aggregate의 질의 처리 과정

요약 데이타 생성자 Aggregate는 먼저 다차원 데이타 모델의 Degree에 대한 Expand 연산을 통해 새로운 Aggregated Degree를 생성하고 시스템 테이블인 DegreeTable에 시간 속성 정보를 포함하는 새로운 Degree를 추가한다.

다차원 데이타 모델의 데이타 타입과 Cast 연산자 및 연산자를 상속 받는 요약 다차원 데이타 모델을 정의하고 다차원 데이타 모델의 Row 데이타 타입의 각 변수 항목에 대한 집계 연산을 통해 요약 데이타를 구한다. 또한 객체-관계 DBMS에서 제공하는 함수 오버 로딩을 통해 생성된 요약 다차원 데이타 모델에만 적용되는 연산자를 다차원 데이터 모델에 독립적으로 새롭게 정의할 수 있다[24].

3.3 다차원 데이타 분석 질의 정의

다차원 데이타 분석 질의는 많은 수의 레코드를 조회하거나 요약하여 데이타 간의 관계 분석이나 비교 또는 통계 기능을 포함한 질의라고 할 수 있다. 본 논문에서는 앞에서 제시한 다차원 데이타 모델 생성자 StarCube를 통해서 구축된 다차원 데이타 모델을 기반으로 다차원 분석 질의는 다음과 같다.

3.4 객체-관계 DBMS 기반 다차원 분석 시스템의 특징

본 논문에서 제안하는 객체-관계 DBMS 기반 다

표 5. 다차원 데이터 분석 질의

분석 질의	분석 질의 속성
Slicing (SC, D, P)	SC 는 구축된 다차원 데이타 모델 D = { di D | di 는 SC의 차원 테이블의 속성 } 단, n(D) = n(D of SC) - 1 P는 시간 속성의 요약 정도로 정의되며, 다차원 데이타 모델의 부분 집합에 대한 단순 조회 및 비교 연산을 수행한다.
Dicing (SC, C, P)	SC 는 구축된 다차원 데이타 모델 C = { (di , c) , di D | c 는 속성 d의 제약 조건 } P는 시간 속성의 요약 정도로 정의되며, 다차원 데이타 모델의 제약 조건을 만족 시키는 부분 집합에 대한 단순 조회 및 비교 연산을 수행한다.
Rollup (SC, D, P)	SC 는 구축된 다차원 데이타 모델 D = { di D | di 는 SC의 차원 테이블의 속성 } P는 시간 속성의 요약 정도로 정의되며, 다차원 데이타 모델의 모델의 부분 집합의 상위시간차원에 대한 단순 조회 및 비교 연산을 수행한다.
DrillDown(SC, D, P)	SC 는 구축된 다차원 데이타 모델 D = { di D | di 는 SC의 차원 테이블의 속성 } P는 시간 속성의 요약 정도로 정의되며, 다차원 데이타 모델의 모델의 부분 집합의 하위시간차원에 대한 단순 조회 및 비교 연산을 수행한다.

차원 데이타 모델과 이를 기반으로 하는 분석 연산자 및 다차원 요약 데이타 모델 생성자 Aggregate는 객체-관계 DBMS의 내부 시스템 카탈로그에 정의되어 있으므로 DBMS 서버의 성능을 그대로 활용할 수 있다. 즉, 다차원 분석 시스템이 객체-관계 DBMS 엔진의 한 부분으로 확장됨으로써 다차원 데이타 모델에 대한 분석 질의가 DBMS 외부에서 프로그래밍 언어를 통해서 이루어 지거나 표준 SQL언어로 변환되어 처리되는 기존의 OLAP 시스템에 비해 효율적이라 할 수 있다.

먼저 객체-관계 DBMS의 사용자 정의 데이타 타입을 기반으로 다차원 데이타 모델을 정의하고 사용자 정의 연산자를 기반으로 다차원 데이타 모델 생성자 StarCube를 정의하여 복수의 관계형 테이블로부터 다차원 데이타 모델을 생성한다. 본 논문에서는 이러한 다차원 데이타 모델 스키마를 위해 차원을 특성에 따라 분류하고 이를 기반으로 분석의 기반이 되는 시간 항목을 별도로 분류하여 시간 속성이라 정의함으로써 시간 항목에 기반한 분석 질의 및 요약 데이타의 생성과 분석에 대한 효율적인 수행을 제공할 수 있다[22]. 이러한 다차원 데이타 모델에 대해 사용자 정의 연산자를 기반으로 Query, Slicing / Dicing, Rollup / Drilldown등의 다차원 분석 질의를 정의한다.

다음 그림 6는 본 논문에서 제안하는 객체-관계 DBMS 기반 다차원 데이타 모델 및 분석 질의를 기반으로 하는 OLAP 시스템의 구조를 나타낸다.

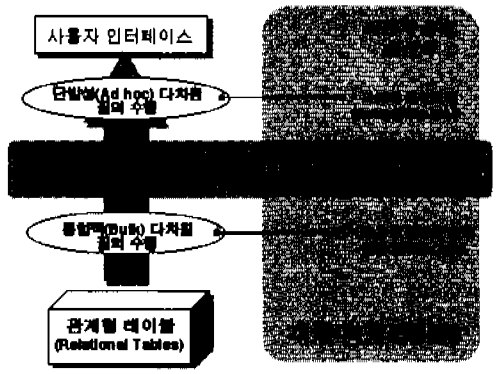


그림 6. 객체-관계 DBMS 기반 OLAP 시스템 구조

객체-관계 DBMS에서는 사용자 정의 연산자를 특정 사용자 정의 데이타 타입에 종속 시켜 최적화된 질의 수행을 위한 환경을 제공한다. 따라서, 본 논문에서는 제안한 다차원 메이타 모델 생성자 StarCube와 다차원 분석 질의를 다차원 데이타 모델에 종속 시켜 정의한다.

객체-관계 DBMS에서는 객체의 계층성에 기반한 상속을 지원하는데 본 논문에서 제시하는 다차원 데이타 모델의 요약 데이타를 DBMS의 시스템 카탈로그에 정의되어 있는 다차원 데이타 모델을 상속하여 생성함으로써 다차원 메이타 모델에 종속되어 있는 데이타 타입 및 다차원 분석 질의를 함께 상속 받도록 한다.

Ⅳ. 다차원 데이타 모델의 구현

4.1 시스템 구현 환경

본 논문에서 제안한 객체-관계 DBMS기반 다차원 데이타 모델의 효용성을 검증하기 위해서 객체-관계 DBMS를 기반으로 다차원 분석을 위한 프로토타입 시스템을 구현하였다. 프로토타입 시스템의 구현 환경은 다음과 같다.

객체-관계 DBMS 기반 다차원 분석을 위한 프로토타입 시스템의 구현을 위해 서버측에는 Sun 사의 Solaris 2.5 기반의 객체-관계 DBMS으로써 Informix Universal Server 를 사용하였다. Informix Universal Server 는 Informix-OnLine Dynamic

표 6. 객체-관계 DBMS기반 다차원 분석 시스템 구현 환경

서버측 운영체제	Sun 사의Solaris 2.5
객체-관계 DBMS	Informix Universal Server V9
SQL 미들웨어	Informix SetNet32 V9.0
클라이언트측 운영체제	Microsoft Windows 95
클라이언트 도구	Informix SQL Editor V2.0 Informix Schema Knowledge V2.32

Server 의 DSA를 기반으로 2D, 3D 이미지, 사운드, 비디오 및 전자 문서와 웹 문서 등 다양한 형태의 데이타와 이를 지원할 수 있는 연산자를 정의할 수 있도록 기존 RDBMS의 성능에 대한 확장성을 제공한다. 또한 각 데이타 객체에 대한 계층과 이에 기반한 상속성 등의 객체 지향적 개념을 지원하는 대표적인 객체-관계 DBMS 이다[9,23].

클라이언트 측에는 대화식 SQL 편집기인 Informix SQL Editor V2.0 과 데이타베이스 스키마를 그래픽 사용자 환경으로 표현하여 주는 Informix Schema Knowledge V2.32를 사용하였다. 또한 네트워 미들 웨어로는 Informix SetNet32 V9.0을 사용하였다.

4.2 다차원 데이타 모델을 위한 사용자 정의 데이타 타입 및 시스템 테이블

4.2.1 사용자 정의 데이타 타입: Degree

다차원 데이타 모델의 시간 속성 항목을 지원하는 Degree 는 Named Row 타입으로 다음과 같은 Degree_t 타입을 통해 구현된다.

```
CREATE ROW TYPE Degree_t (
        INTERVAL    VARCHAR(8),
        VALID       INTEGER,
        INVALID     INTEGER,
        START       DATE );
```

그림 7. 사용자 정의 데이타 타입: Degree_t 정의

위의 사용자 정의 데이타 타입인 Degree_t 는 다차원 데이타 모델 내의 유효 데이타의 주기, 연속 기간, 유효하지 않은 연속 기간 및 시작 시간을 포함한다. 따라서 다차원 데이타 모델은 각각 위에서 정의된 Degree_t 타입의 Degree를 가지며 이는 다차원 데이타 모델의 시간 속성 항목의 유효성을 보장하고 다차원 데이타 모델에 대한 분석 질의 및 요약 데이타에 대한 연산에 이용된다.

4.2.2 시스템 테이블

다차원 데이타 모델을 위한 사용자 정의 데이타 타입 및 이에 대응하는 연산식에 대한 정보를 관리하기 위하여 시스템 테이블들을 다음과 같이 구현한다.

```
CREATE ROW TYPE Dgr_t (
    DGR_NAME    VARCHAR(10),
    SC_NAME     VARCHAR(20),
    DGR_TYPE    Degree_t );
CREATE TABLE DegreeTable OF
        TYPE Dgr_t;
```

그림 8. 시스템 테이블: DegreeTable 정의

사용자 정의 데이타 타입인 Dgr_t는 각 다차원 데이타 모델에 대한 Degree 정보를 관리하기 위한 시스템 테이블의 사용자 정의 데이타 타입으로 Named Row 타입이다. Dgr_t는 Degree 이름 및 이를 참조하는 StarCube의 이름과 Degree_t의 형태인 Degree 정보로 이루어 진다. 따라서 다차원 데이타 모델에 대한 각종 질의문 및 요약데이타 생성시 시스템 테이블인 DegreeTable로부터 해당 다차원 데이타 모델의 Degree 정보를 입력 받아 질의를 수행한다.

또한 각 다차원 데이타 모델의 생성시 차원 항목 및 변수 항목에 대한 정보를 포함하는 시스템 테이블들을 다음과 같이 구현한다.

```
CREATE ROW TYPE Dimension_t (
        DIM_NAME        VARCHAR(10),
        TBL_NAME        VARCHAR(20));
CREATE TABLE DimensionTable OF
        TYPE Dimension_t;
CREATE ROW TYPE Variable_t (
        VAR_NAME        VARCHAR(10),
        TBL_NAME        VARCHAR(20),
        VAR_TYPE        DEGREE_t);
CREATE TABLE VariableTable OF TYPE
        Variable_t;
```

그림 9. 시스템 테이블: DimensionTable및 VariableTable 정의

차원 항목에 대한 정보를 저장하기 위해서 그림

4.3에서 정의한 Dimension_t 형태의 시스템 테이블인 DimensionTable을 구현한다. 또한 변수 항목에 대한 정보를 저장하기 위해서 그림 4.3 에서 정의한 Variable_t 형태의 시스템 테이블 VariableTable을 구현한다. 특히 각 변수 항목은 Degree_t을 가지고 있다.

4.3 사용자 정의 연산자: 다차원 데이타 모델 생성자

4.3.1 다차원 데이타 모델 생성자: StarCube

앞서 구현한 사용자 정의 데이타 타입 및 시스템 테이블을 기반으로 복수개의 관계형 테이블로부터 다차원 데이타 모델을 구축하는 StarCube 연산자는 그림10과 같다.

먼저 다차원 데이타 모델에 저장된 데이타의 특성에 따라 참조할게 될 시간 속성 정보 Degree를 정의하고시스템 테이블인 DegreeTable에 추가한다. 다차원 데이타 모델 생성자 StarCube는 먼저 차원 항목간의 Cartesian Product에 대해서 각 Named Row 타입의 컬렉션 타입인 Ref 칼럼을 생성시키고 초기치를 설정한다. 따라서 원래 관계형 테이블에 저장되어 있던 유효 데이타의 수만큼 Ref 칼럼의 원소가 할당되게 한다. 다음으로 각 차원 항목간의 Cartesian Product 의 실제 데이타 값 및 연산식의 결과값을 원래의 관계형 테이블로부터 구해서 Ref 칼럼의 각 Name Row 타입의 변수 항목에 대입한다. 이때 Named Row 타입의 컬렉션 타입인 Ref 칼럼은 컬렉션 타입의 임시 변수와 Row 타입의 임시 변수를 이용해서 변환하도록 한다[23].

이때 Ref 칼럼은 해당 다차원 데이타 모델이 참조하는 Degree이름과 Degree에 저장된 정보를 이용해서 Degree의 시작 시간을 나타내는 Start 칼럼으로부터의 Offset을 함께 저장하여 질의에 이용하도록 한다. 또한 다차원 데이타 모델이 생성되면 이에 대응하는 정보를 시스템 테이블인 SCTable에 저장시킨다.

4.3.2 요약 다차원 데이타 모델 생성자: Agg_ StarCube

앞서 구현한 다차원 데이타 모델을 기반으로 요약 다차원 데이타 모델을 생성하는 Agg_StarCube 연산자는 다음과 같다.

먼저 다차원 데이타 모델의 사용자 정의 데이타 타입을 상속 받는 요약 다차원 데이타 모델의 데이타 타입을 정의하고 이를 기반으로 다차원 데이타

```
① 다차원 데이타 모델 생성자 StarCube 선언
CREATE FUNCTION StarCube (DGR_NAME
        VARCHAR(10),
    V_D1  VARCHAR(10), V_D2  VARCHAR(10),
    V_V1  VARCHAR(10), V_V2  VARCHAR(10),
    V_V3  VARCHAR(10) )
    RETURNING  VARCHAR(10);
② 참조 칼럼에 수치값 입력시 사용할 임시 변수 선언
    DEFINE      TEMP     COLLECTION;
    DEFINE      R        ROW;
    DEFINE      S, P, L  VARCHAR(10);
    LET  R      =  ROW (100 , 200 , 300);
③ 다차원 데이타 모델의 정의
   CREATE TABLE StarCube  OF
                TYPE  SC_t;
④ 차원 항목간의 Cartesian Product 으로 커서 생성
   FOREACH    CURSOR1    FOR
      SELECT  V_D1, V_D2  INTO  S, P
        FROM    PRODUCT,  SALES_PERSON

⑤ 차원 항목간 교차점의 Named Row 형의 컬렉션 초기화
  FOREACH     CURSOR2    FOR
  SELECT V_V1, V_V2, V_V3   INTO  S, P, L
  FROM    SALES_DATA, PRODUCT
  WHERE SALES_DATA.P_id = PRODUCT.P_id
              // Null Value 대입 //

⑥ 차원 항목간 교차점의 Named Row 형의 수치값 대입
  SELECT  REF       INTO    TEMP
  FROM StarCube;
  FOREACH   CURSOR3       FOR
  SELECT  *  INTO   R     FROM
          TABLE(TEMP)
     UPDATE TABLE(TEMP)   SET
        // 해당 Value 대입 //
     WHERE        CURRENT  OF  CURSOR3;
       ...
⑦ 임시 컬렉션 변수를 Ref 칼럼에 대입
  UPDATE StarCube SET  REF  =  TEMP
  RETURN DGR_NAME;
  END  FUNCTION;
```

그림 10. 다차원 데이타 모델 생성자: StarCube

모델 객체를 상속 받는 요약 다차원 데이타 모델을 구현한다. 다음으로 다차원 데이타 모델의 차원 항목에 대해서 각 Named Row 타입의 컬렉션 타입인 Ref 칼럼을 생성 시키고 초기치를 설정한 후 다차원 데이타 모델의 데이타 값 및 연산식의 결과값을 요약 연산자를 통해 계산하여 Ref 칼럼의 각 Name Row 타입의 변수 항목에 대입한다. 이 때 Named Row 타입의 컬렉션 타입인 Ref 칼럼은 컬렉션 타입의 임시 변수와 Row 타입의 임시 변수를 이용해서 변환하도록 한다. 각 Ref 칼럼은 해당 다차원 데이타 모델이 참조하는 Degree 이름과 Degree에

저장된 정보를 이용해서 요약 연산을 수행한다. 또한 요약 다차원 데이타 모델이 생성되면 이에 대응하는 정보를 시스템 테이블인 SCTable에 저장시킨다. 단 본 논문의 Named Row 타입에 대한 지원 부분은 Informix Universal Server V9.2이상부터 가능하다[21].

```
① 요약 다차원 데이타 모델 생성자
          Agg_StarCube 선언
 CREATE  FUNCTION      Agg_StarCube
      (SC_NAME  VARCHAR(10),
       Agg_M  VARCHAR(10))
 RETURNING            VARCHAR(10);

② 참조 칼럼에 수치값 입력시 사용할 임시 변수 선언
 DEFINE   TEMP      COLLECTION;
 DEFINE   R         ROW;
 DEFINE   S, P      VARCHAR(10);
 LET      R         =  ROW (100, 200, 300);

③ 요약 다차원 데이타 모델의 정의
 CREATE TABLE Agg_StarCube
  OF  TYPE  Agg_StarCube_t     UNDERStarCube;

④ 차원 항목간의 Cartesian Product 으로 커서 생성
 FOREACH      CURSOR1        FOR
   SELECT     D1, D2   INTO   S,  P
   FROM      StarCube;

⑤ 차원 항목을 요약 다차원 데이타 모델에 저장
 INSERT  INTO   Agg_StarCube( D1, D2 )
 VALUES ( S , P );
 END   FOREACH

⑥ 차원 항목간 교차점의 Named Row 형의 요약 수치값 대입
 SELECT  REF  INTO  TEMP  FROM      StarCube;
         FOREACH     CURSOR2          FOR
         SELECT  *  INTO  R  FROM
                    TABLE(TEMP)
         UPDATE   TABLE(TEMP)          SET
         WHERE CURRENT   OF   CURSOR2;
         END   FOREACH
         UPDATE  StarCube
           SET     REF   =   TEMP
 RETURN              Agg_M;
 END      FUNCTION;
```

그림 11. 요약 다차원 데이터 모델 생성자:Agg_StarCube

4.4 다차원 데이타 모델의 구현 예제

4.4.1 예제 데이타

다음 그림 12는 본 논문에서 이용한 예제 데이타의 관계형 테이블들을 나타낸다.

본 논문에 사용되는 예제 데이타는 자동차 판매 실적을 관리하기 위한 판매원 테이블(Sales_person)

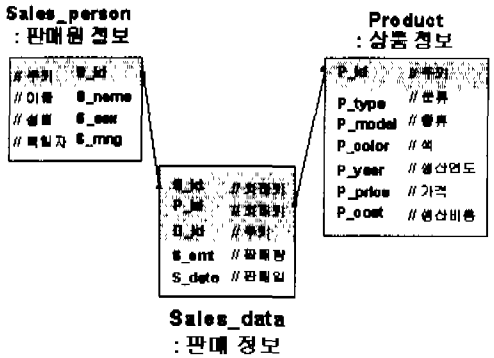


그림 12 예제 데이타의 관계형 테이블 구조

과 상품 테이블(Product), 판매 정보 테이블(Sales_data)로 구성된다. 판매원 테이블은 주 키인 S_id를 갖고 자동차 테이블은 주 키로 P_id를 가지며, 판매 정보 테이블은 참조 키를 기반으로 조인(Join)을 통해 각종 질의를 수행한다. 그림 13은 이러한 관계형 테이블에 기반한 질의문의 수행 예를 나타낸다. 이 예제는 "모든 판매원의 자동차별 판매량과 총 판매액 및 총 이익을 출력하라" 는 질의에 대한 수행 결과를 나타낸다. 이러한 질의를 수행하기 위해서는 자동차에 대한 정보를 포함하고 있는 Product 테이블과 판매 정보를 포함하는 Sales_data 테이블 간의 조인을 기반으로 해당 칼럼을 추출하고 또한, 각 칼럼을 피연산자로 하는 연산을 통한 결과값을 얻어낸다.

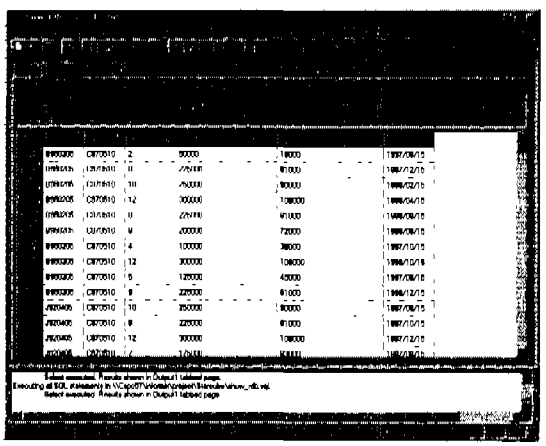


그림 13 관계형 테이블 기반 질의 수행 예

그러나 질의 수행 결과가 테이블 간의 조인에 기반한 관계형 테이블의 형태가 됨으로써 분석을 위해서는 이를 재 배치할 필요가 있다.

4.4.2 다차원 데이타 모델 구축 예제

다음은 본 논문에서 제안한 객체-관계 DBMS 기반 다차원 데이타 모델을 그림 13의 예제에 대해 구축하기 위한 실행을 보여 준다.

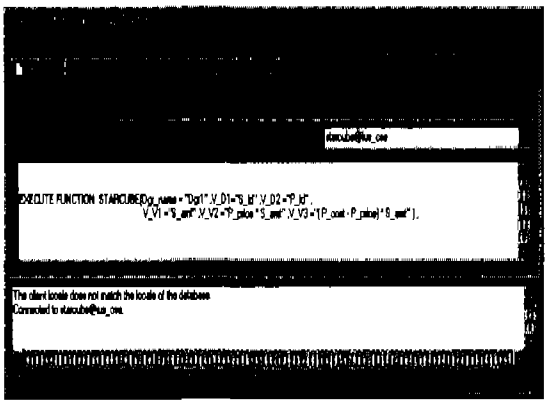


그림 14. 다차원 데이터 모델 생성자 수행 예

그림 14는 Informix Universal Server 인 Service Name을 ius_cse로 하는 서버의 Starcube 라는 데이타베이스에 접속하여 다차원 데이타 모델 생성자를 수행하는 것을 나타낸다[28]. 다차원 데이타 모델 생성자 Starcube는 매개 변수로써 시간 속성을 나타내는 Degree 타입의 'Dgr1'과 단순 속성인 차원으로 'S_id' 와 'P_id'를 그리고 변수 항목으로 'S_amt'와 총 판매액을 나타내는 'S_amt * P_price', 총 이익을 나타내는 '(P_cost ? P_price) * S_amt'를 정의하여 다차원 데이타 모델을 구축한다.
이러한 다차원 데이타 생성자를 통해 구축된 다차원 데이타 모델은 다음과 같다.

그림 15는 그림 14의 다차원 데이타 모델 생성자의 수행 결과 구축된 다차원 데이타 모델을 보여준다. 그림 15의 (A) 는 다차원 데이타 모델의 차원 항목 d1, d2와 이들의 교차점에서 발생하는 수치 값을 포함하는 칼럼인 Ref으로 구성된다. 칼럼 Ref는 실제 수치 값 및 연산식의 결과값을 시간 속성을 기준으로 구조화 시킨 Named Row 형의 컬렉션 형이다. 그림 4.9의 (B)는 d1 = 'J920405' 이고 d2 = 'C970510' 의 칼럼 Ref를 나타낸다. 그림 15의 (B) 는 툴라이언트 도구인 SQL Editor V2.0 의 Cell Viewer로써 다차원 데이타 모델의 칼럼 Ref 과 같이 컬렉션 타입의 데이타 셀을 보여 준다. 위의 예에서는 판매원 'J920405'의 자동차 'C970510' 에 대한 판매량, 총 판매액 및 총 이익과 시간 속성

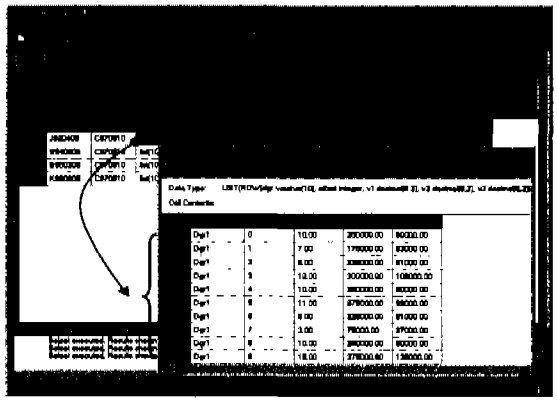


그림 15. 다차원 데이타 모델 구축 예

을 나타내는 Degre 'Dgr1'에 대한 Offset 을 함께 Named Row 타입으로 정의하고 각각을 Offset을 기준으로 정렬시킨 컬렉션 타입인 리스트로써 칼럼 Ref 가 구조화 되었음을 알 수 있다. 또한 다차원 데이타 모델 생성시 각 다차원 데이타 모델이 참조하는 Degree를 정의하고 시스템 테이블 Degree Table에 등록을 시킴으로써 각종 분석 질의 및 요약 데이타 생성시 이를 기반으로 수행하도록 한다.
다음은 그림 14의 다차원 데이타 모델 생성자를 수행시킨 후 시간 속성의 정보를 포함하는 시스템 테이블DegreeTable을 나타낸다.

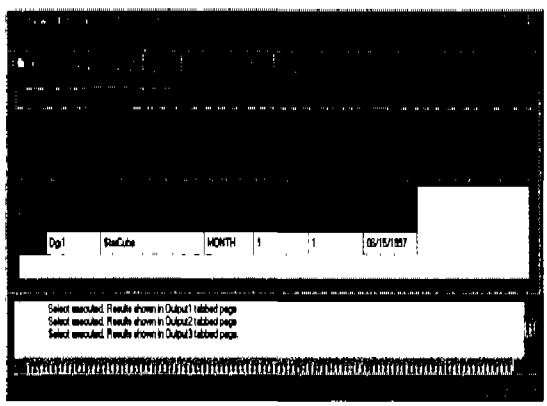


그림 16. 다차원 데이타 모델 생성 후 시스템 테이블 Degree Table의 예

그림 16에서 Degree인 Dgr1은 Dgr_type가 { "Month", 1 , 1 , "06/15/1997" } 로 정의되는데 이는 1997 년 6월부터 1개월의 유효한 데이타가 그리고 7월을 제외하고 다시 1개월의 유효 데이타가 존재함을 나타낸다. 또한 다차원 데이타 모델 생성

시 각 다차원 데이타 모델을 시스템 테이블 SCTable에 등록을 시킴으로써 각종 분석 질의 및 요약 데이타 생성시 이를 기반으로 수행하도록 한다.

그림 17은 그림 14의 생성자를 수행시킨 후 다차원 데이타 모델에 대한 정보를 포함하는 시스템 테이블 SCTable을 나타낸다.

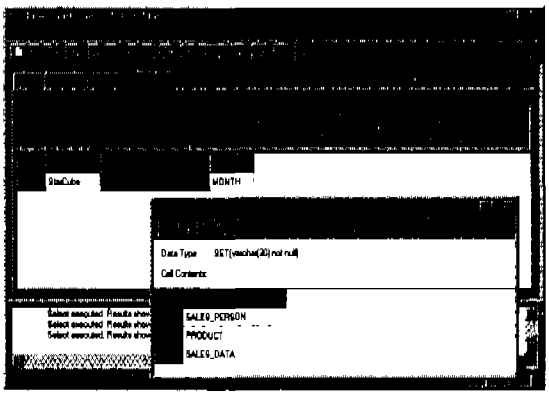


그림 17. 다차원 데이타 모델 생성 후 시스템 테이블 SCTable의 예

그림 17의 SCTable은 다차원 데이타 모델 Star Cube가 관계형 테이블 Sales_person, Product, Sales_data로 구성되었으며 요약 정도는 월별임을 나타낸다.

4.4.3 요약 다차원 데이타 모델 구축 예제

그림 18는 본 논문에서 제안한 객체-관계 DBMS의 상속성에 기반한 요약 데이타를 구축하기 위한 실행을 보여준다.

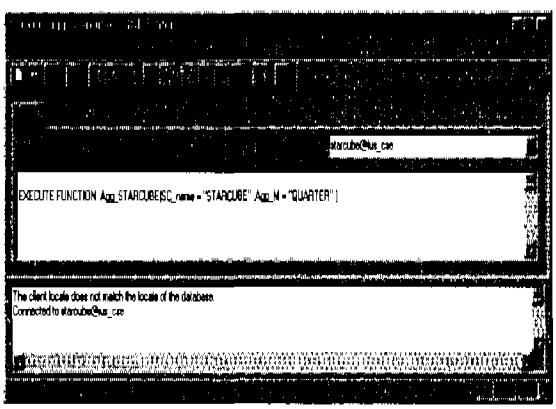


그림 18. 요약 다차원 데이타 모델 생성자 수행 예

요약 다차원 데이타 모델 생성자 Agg_Starcube는 매개 변수로 기 구축된 다차원 데이타 모델 'StarCube'와 요약 정도를 명시하는 'Quarter'를 입력 받아 다차원 데이타 모델의 데이타 타입과 연산자 등을 상속 받는 분기별 요약 데이타를 다차원적으로 구성하도록 한다.

이러한 요약 다차원 데이타 생성자를 통해 구축된 요약 다차원 데이타 모델은 다음과 같다. 그림 19의 (A)는 다차원 데이타 모델의 차원 항목들과 이들의 교차점에서 발생하는 수치 값을 포함하는 칼럼 Ref으로 구성된다. 그림 19의 (B)는 d1 = 'J920405' 이고 d2 = 'C970510'의 칼럼 Ref를 나타낸다.

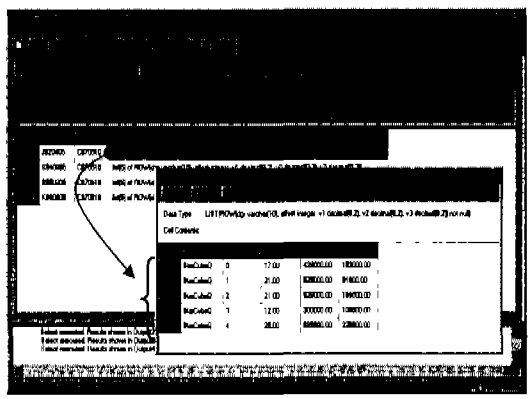


그림 19. 요약 다차원 데이터 모델 구축 예

그림 19은 그림 4.9의 다차원 데이타 모델에 대한 분기별 요약 데이타의 다차원 데이타 모델을 보여준다. 판매원 'J920405' 의 자동차 'C970510'에 대한 판매량, 총 판매액 및 총 이익의 분기별 요약 데이타와 요약된 시간 속성을 나타내는 Degree 'StarCubeQ'에 대한 Offset을 함께 Named Row 타입으로 정의하고 각각을 Offset을 기준으로 정렬시킨 컬렉션 타입 리스트로 칼럼 Ref 가 구조화 되었음을 알 수 있다.

또한 요약 다차원 데이타 모델 생성시 각 요약 다차원 데이타 모델이 참조하는 시간 속성을 나타내는 Degree를 정의하고 시스템 테이블 Degree Table에 등록을 시킴으로써 각종 분석 질의 및 요약 데이타 생성시 이를 기반으로 수행하도록 한다.

그림 20은 그림 18의 요약 다차원 데이타 모델 생성자를 수행시킨 후 시간 속성 정보를 포함하는 시스템 테이블 DegreeTable을 나타낸다.

위의 예에서 Degree 인StarCubeQ는 Dgr_type가 { "Quarter", 1, 0, "06/15/1997" } 로 정의되는데

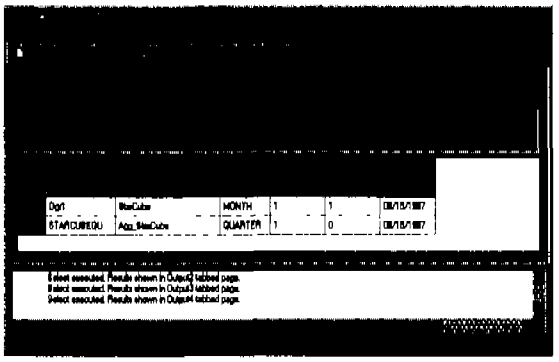


그림 20. 요약 다차원 데이터 모델 생성자 수행 후 시스템 테이블 DegreeTable의 예

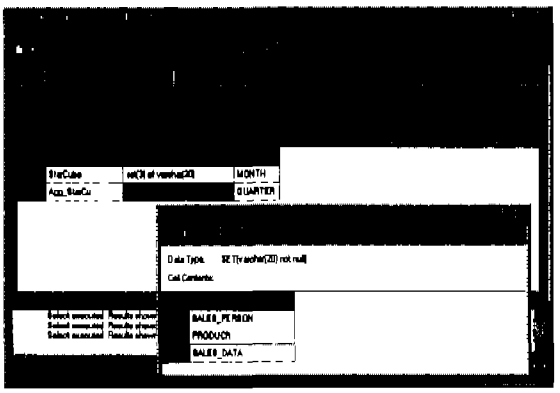


그림 21. 요약 다차원 데이터 모델 생성자 수행후 시스템 테이블 SCTable의 예

이는 1997 년 6월부터 분기별 유효 데이타가 존재함을 나타낸다.

또한 요약 다차원 데이타 모델 생성시 각 요약 다차원 데이타 모델을 시스템 테이블 SCTable에 등록을 시킴으로써 각종 분석 질의 및 요약 데이타 생성시 이를 기반으로 수행하도록 한다.

그림 21은 그림 18의 요약 데이타 생성자 수행 후 시스템 테이블 SCTable을 나타낸다. SCTable은 요약 다차원 데이타 모델 Agg_StarCube가 관계형 테이블Sales_person, Product , Sales_data로 구성되었으며 요약 정도는 분기별 임을 나타낸다.

4.4.4 계층성에 기반한 다차원 데이타 모델의 상속 예제

본 논문에서 제안하는 객체-관계 DBMS 기반 요약 다차원 데이타 모델은 객체-관계 DBMS에서 제공하는 계층성에 기반한 객체 단위의 상속을 통해 구축된다. 즉, 기 구축된 다차원 데이타 모델의 데이타 타입을 상속 받는 요약 다차원 데이타 모델을 정의하고 다차원 데이타 모델의 Row Data 타입의 각 변수 항목에 대한 집계 연산을 통해 요약 데이타를 구한다.

그림 22는 Informix Schema Knowledge V2.32를 통해서 다차원 데이타 모델과 이를 기반으로 구축된 요약 다차원 데이타 모델의 상속 관계를 보여준다.

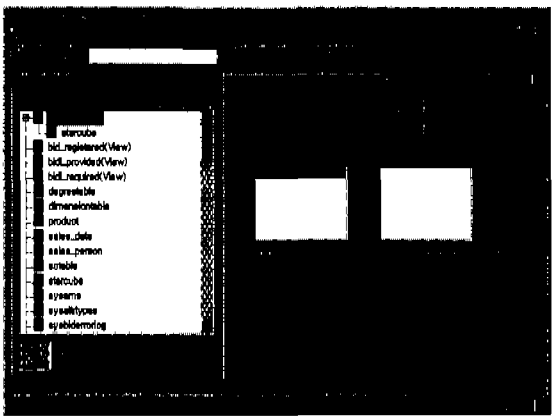


그림 22. 다차원 데이타 모델 StarCube와 요약 다차원 데이타 모델 Agg_StarCube의 계층 구조 예

그림 22는 Informix Universal Server 인 Service Name을 ius_cse로 하는 서버의 Starcube 라는 데이타베이스에 접속하여 데이타베이스 내에 정의된 모든 객체들의 계층 및 상속 관계를 보여 준다. 위의 예에서 요약 다차원 데이타 모델 Agg_StarCube 는 다차원 데이타 모델 StarCube를 상위 테이블로 하여 데이타 타입 및 연산자 등을 상속 받아 정의 되었음을 알 수 있다.

V. 결론 및 향후 연구 과제

본 논문에서는 데이타의 다차원적인 개념을 좀더 명확하게 반영할 수 있고, 분석 질의어도 간단하게 구현될 수 있으며 기존의 OLAP 시스템이 지닌 분석 질의의 변환에 따른 한계점과 차원 항목에 대한 유연성의 결여 등을 보완하기 위해 객체-관계 DBMS 기반의 다차원 데이타 모델의 설계를 제안하였다.

이는 객체-관계 DBMS가 제공하는 사용자 정의 데이타 타입과 이를 기반으로 연산자를 정의할 수 있도록 하는 확장성 및 계층 구조에 의거한 객체의 상속성을 다차원 분석을 위한 데이타의 모델 설계

에 적용하여 다차원 분석 시스템을 객체-관계 DBMS 엔진의 한 부분으로 확장 시킴으로써 보다 효율적인 다차원 분석 시스템의 설계 방안을 제시하고자 하였다.

먼저 분석의 기반이 되는 차원을 각 차원의 특성 및 중요도에 따라 시간 속성 항목, 계층성을 갖지 않는 차원 항목 및 수치 값 또는 연산식에 의한 결과값을 갖는 변수 항목으로 분류하였다. 이러한 차원간의 분류를 기반으로 기존의 차원 모델링에서 제시한 차원 테이블과 사실 테이블을 결합시켜 다차원 데이타 스키마를 정의하였으며 이에 대한 메이타의 다차원 모델을 설계하였다.

즉, 객체-관계 DBMS에서 제공하는 사용자 정의 데이타 타입을 기반으로 다차원 데이타 모델을 위한 데이타 타입과 시스템 테이블을 정의하고 이에 적용되는 연산식을 객체-관계 DBMS의 사용자 정의 연산식에 근거하여 정의하였다. 또한 데이타 객체의 계층성에 기반하여 기 구축된 다차원 데이타 모델로부터 상속을 받아 요약 데이타를 설계함으로써 다차원 데이타 모델의 데이타 타입 및 연산자를 요약 데이타에 대해서도 적용될 수 있도록 하였다.

이에 따라 본 논문에서 제안한 객체-관계 DBMS 기반 다차원 데이타 모델과 질의의 효용성을 검증하기 위해 객체-관계 DBMS 인 Informix Universal Server를 통해 다차원 분석 질의를 위한 프로토타입 시스템을 구현하고 클라이언트 도구를 사용하여 다차원 데이타 모델 및 요약 데이타의 구축에 대한 예시를 보여주었다.

본 논문에서 제안한 객체-관계 DBMS 기반 다차원 데이타 모델이 갖는 장점은 차원간의 분류를 통한 다차원 데이타 스키마에 의해서 데이타의 다차원적인 개념을 그대로 반영하는 데이타 모델을 구축할 수 있었고, 정의된 다차원 데이타 모델에 종속된 분석 질의어를 쉽게 구현할 수 있었다. 특히 다차원 데이타 모델 객체를 상속 받는 요약 데이타를 구축하여 다차원 분석에 적합하도록 할 수 있었다.

향후 연구 과제로는 본 연구에서 제안된 다차원 데이타 모델에 대한 다양한 다차원 분석 질의어를 설계, 구현하고 보다 향상된 사용자 인터페이스를 구현하고자 한다. 따라서 본 연구의 내용과 연계하여 다차원 분석 시스템을 객체-관계 DBMS의 확장된 형태로 최적화 하여 통합시킬 수 있을 것이다.

참 고 문 헌

[1] W. H. Inmon, What is a Data Warehouse?, Prism Solutions Technology Topics, Vol.1, No.1, 1995.

[2] M. H. Brackett, The data warehouse challenge, John Wiley & Sons, 1996.

[3] W. H. Inmon, Building the Data Warehouse, John Wiley & Sons, 1996.

[4] W. H. Inmon, J. D. Welch, K. L. Glassey, Managing the Data Warehouse, John Wiley & Sons, 1996.

[5] R. Kimball, The Data Warehouse Toolkit, John Wiley & Sons, 1996.

[6] 조 재희, 박 성진, 데이타 웨어하우징과 OLAP, 대청, 1996.

[7] R. Tanler, The Intranet Data Warehouse, John Wiley & Sons, 1997.

[8] E. Thomsen, OLAP Solution, John Wiley & Sons, 1997.

[9] M. Stonebraker, Object-Relational DBMSs: The Next Great Wave, Morgan Kaufmann Publishers, Inc., 1996.

[10] C. Zaniolo, S. Ceri, C. Faloutsos, R. T. Snodgrass, V. S. Subrahmanian, R. Zicari, Advanced Database Systems, Morgan Kaufmann Publishers, 1997.

[11] V. Harinarayan, A. Rajaraman, J. D. Ullman, "Implementing Data Cubes Efficiently," In Proc. on ACM SIGMOD Conf., pages 205-216, 1996.

[12] J. Gray, A. Bosworth, "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals," Microsoft Technical Report MSR-TR-95-22, 1995.

[13] R. Agrawal, A. Gupta, and S. Sarawagi, "Modeling Multidimensional Database Technology," Kenan System Corporation White Paper, Available Via: http://www.kenan.com/aumate/mddb.html, 1995.

[14] Oracle, "Oracle OLAP Products: Adding values to the Data Warehouse," Oracle White Paper, 1995.

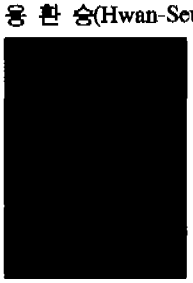
[15] Red Brick System, "Star Schemas and Starjoin Technology," Red Brick Systems White Paper, 1996.
[16] S. Kelly, "OLAP & ROLAP ? Two Ways of Giving the User Multidimensional Data Access," Data Warehouse Report, Issue5, Spring 1996.
[17] P. O'Neil, G. Graefe, "Multi-Table Joins Through Bitmapped Join Indexes," In Proc. on ACM SIGMOD, pages 8-11, September 1995.
[18] W. P. Yan, P. A. Larson, "Eager Aggregation and Lazy Aggregation," In Proc. on 21st VLDB, pages 345-357, 1995.
[19] V. Harinarayan, A. Gupta, "Generalized Projection: A Powerful Query-Optimization Technique," Stanford Technical Report No. STANCS-TN-94-1, 1994.
[20] A. Gupta, V. Harinarayan, D. Quass, "Aggregate-Query Processing in Data Warehousing Environments," In Proc. on 21st VLDB, pages 358-369, 1995.
[21] Informix, Informix Universal Server Guide to SQL: Reference, Informix Press, 1997.

김 은 영(Eun-Young Kim) 정회원

1996년 8월 : 이화여자대학교 컴퓨터학과 졸업
1998년 8월 : 이화여자대학교 대학원 컴퓨터학과 석사
1998년 8월~현재 : 한국 SAS(주) 연구원

<주관심 분야> OLAP, Data Mining

용 환 승(Hwan-Seung Yong) 정회원



1983년 2월 : 서울대학교 컴퓨터공학과 졸업
1985년 2월 : 서울대학교 대학원 컴퓨터공학과 석사
1985년 1월~1989년 1월 : 한국전자통신연구소 연구원
1994년 2월 : 서울대학교 대학원 컴퓨터공학과 박사
1995년~현재 : 이화여자대학교 컴퓨터학과 교수

<주관심 분야> 멀티미디어 데이터베이스, OLAP, Data Mining