

# ROLAP 기반의 인터넷 쇼핑몰 데이터 웨어하우스 설계에 대한 연구

이단영, 이원조, 고재진  
울산대학교 대학원 컴퓨터 정보통신공학부  
hundo@chollian.net, wjlee@mail.ulsan-c.ac.kr, jikoh@cic.ulsan.ac.kr

## Design of data warehouse for internet shopping mall based on ROLAP

Dan-Young Lee, Won-Jo Lee, Jae-Jin Koh  
Division of Computer Engineering and  
Information Technology, University of Ulsan

### 요 약

고객 DB를 통해 세분 고객별로 구매량, 주요 고객을 파악케 하며 각종 마케팅 활동이나 판촉활동이 고객의 구매/이용 패턴에 어떤 영향을 미치며, 물품을 구매하는 소비자의 다양한 구매 패턴을 분석하기 위해서 쇼핑몰 운영자가 여러 각도에서 문제 분석과 의사 결정을 빠르고 신속하게 할 수 있도록 기존 쇼핑몰의 관계형 DB를 이용하여, 다차원적 데이터 모델링을 통해서 다차원적인 분석이 가능하도록 하는 ROLAP를 이용한 인터넷 쇼핑몰의 데이터 웨어하우스 구축 방안을 제시하고자 한다.

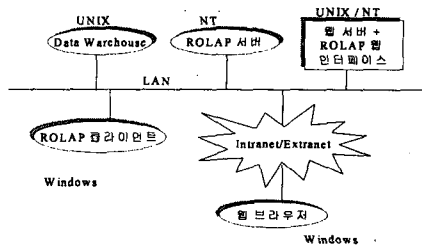
### 1. 서론

인터넷 쇼핑몰에서 물품을 구매하는 소비자의 다양한 구매 패턴을 분석하기 위해서, 소비자 개개인의 연령, 취향, 품목에 대한 구매량, 구매 시기 등등 소비형태를 파악하고, 쇼핑몰 운영자가 여러 각도에서 문제 분석과 의사 결정을 빠르고 신속하게 할 수 있도록 ROLAP(Realational Online Analytical Processing)를 이용한 데이터 웨어하우스(data warehouse) 구축 방안을 제시하고자 한다.

ROLAP은 이미 많이 사용하고 있는 관계형 DB를 이용하여, 다차원적 데이터 모델링을 통해서 다차원적인 분석이 가능하도록 하는 방식이다. ROLAP은 관계형 DB를 근간으로 하기 때문에 확장성이 좋으며 특히, 전사적 데이터 웨어하우스(enterprise data warehouse)를 구축하거나 고객이름, 주소, 전화번호와 같이 원시 데이터를 출력하는 것이 중요한 데이터베이스 마케팅(DB marketing) 같은 분야에 좋다.

등에 대한 자료를 액세스하거나 저장하도록 되어 있다. 이렇게 저장된 데이터베이스로 데이터 웨어하우스 구축하기 위해서, 4 Tier 구조 형태로 ROLAP을 설계하고자 한다.

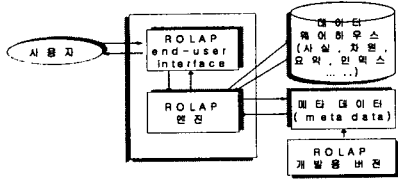
ROLAP 4 Tier 구조 형태는 웹브라우저, 웹서버, ROLAP 서버, 데이터웨어하우스로 구성되어 있다. 이는 웹 브라우저 상에서 데이터 웨어하우스에 있는 데이터를 액세스하며, 웹 서버에는 ROLAP 서버와 인터페이스가 있어 데이터 웨어하우스에 있는 데이터를 HTML 형식으로 바꾸어 주도록 구성한다.



<그림 1> ROLAP 4 Tier 구조

### 2. 시스템 구성요소 및 설계

인터넷 쇼핑몰의 데이터베이스(database) 시스템은 일반적으로 웹브라우저를 통해서 데이터베이스의 고객, 물품, 구매



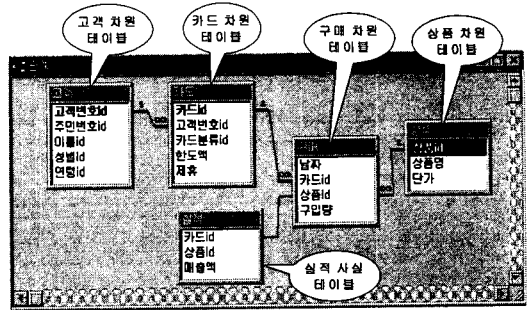
<그림 2> ROLAP 프로세싱의 내부 구조

데이터 웨어하우스를 구축하기 위해서 먼저 사용자의 요구 사항을 분석한다. 그리고 이 요구사항에 따라서 다차원 모델링을 한다. 데이터 웨어하우스 개발자(data modeller)들이 다차원 모델링에서 정의된 내용에 따라서 사실 테이블, 인덱스(index), 키(key), 제약조건(constraint), 뷰(view)등을 만든다. 입력이 끝나면 ROLAP 개발용 버전은 위의 입력 정보를 받아서 메타 데이터(meta data)를 생성한다. 메타 데이터는 데이터 웨어하우스가 있는 서버 또는 ROLAP 서버에 둘 수도 있다. 이 메타 데이터 역시 관계형 DB 테이블이다. 사용자가 원하는 데이터를 보기를 원한다면, ROLAP 사용자 버전에는 사용자 화면 인터페이스가 ROLAP 엔진에게 이러한 사실을 알리면, ROLAP 엔진은 메타 데이터에 SQL문으로 질의(query)를 던진다. SQL문으로 원하는 메타 데이터를 추출한다. 이것은 메타 데이터가 잘 설계되어 있어야 쉽고 신속하게 데이터를 잘 가져올 수 있다. 이러한 정보를 분석하고 나서 여러 가지 복잡한 법칙에 따라서 SQL문을 만든다. 이 SQL문은 실제 데이터 웨어하우스에 보내서 프로세싱이 되어야 하므로 SQL문문이 정확해야 함은 물론이고 퍼포먼스(performance)를 좋게 하기 위해서 여러 가지 기법을 사용한다. 위의 SQL문장을 데이터 웨어하우스가 있는 관계형 DB로 보낸다. 관계형 DB는 이를 처리한다. 이것은 RDBMS가 기계적으로 프로세싱해주는 것이다. 관계형 DB에서 프로세싱이 끝나면 그 결과를 ROLAP 엔진으로 보내 준다. 이 때 ROLAP 엔진에서는 이 결과를 캐싱(caching)해준다. 결과 데이터를 가지고 사용자가 원하는 형식, 폰트, 칼라, 배열을 맞추어 화면에 디스플레이 해준다.

2.1. 다차원 모델링(Multidimensional Modeling)

다차원 모델링은 차원과 사실컬럼을 복합하여 관계형 DB의 테이블로 만든다. 예를들면, 한 쇼핑몰이 고객과 카드에 대한 정보를 가지고 있고 카드사의 상품별로 매출액을 분석한다고 가정한다면, 매출 실적에 대한 테이블과 SQL문은 다음과 같다.

- \* 차원(dimension) : 고객(고객id, 주민id, 성별, 이름, 연령,...), 카드(카드id, 고객번호id, 카드분류, 한도액, 제휴,...), 상품(상품id, 상품명, 단가.), 구매(날짜, 카드id, 상품id, 구입량)
- \* 사실 컬럼(fact column) : 매출액, 매출수량



<그림 3> 다차원 모델링 예

```

SELECT 카드.카드id, 상품.상품id,
Sum([구매][구입량]*[상품][단가])
AS 매출액 INTO 실적
FROM (고객 INNER JOIN 카드
ON (고객.고객번호id = 카드.고객번호id)
AND (고객.고객번호id = 카드.고객번호id))
INNER JOIN (상품 INNER JOIN 구매
ON 상품.상품id = 구매.상품id)
ON 카드.카드id = 구매.카드id
GROUP BY 카드.카드id, 상품.상품id;
    
```

2.2. 데이터 추출, 가공, 로드(ETT)

ETT는 데이터의 추출(Extraction), 가공(Transformation), 전송(Transportation)의 약자로 데이터를 소스 시스템(source system)에서 추출하여 데이터 웨어하우스에 로드시켜 정제 작업까지 이르는 전 과정을 말한다.

데이터 웨어하우스에서 필요한 최종 테이블은 사실 테이블이다. 이것에서부터 여러 가지 요약 테이블을 생성한다. 또한 종류의 테이블은 차원 테이블로 그 크기가 크지 않으므로 쉽게 데이터를 로딩할 수 있다. 따라서, 데이터 웨어하우스에서는 사실 테이블과 요약 테이블만 있으면 실제 사용자가 데이터를 볼 수 있다.

소스 시스템은 대개의 경우 그 회사의 중요한 운영 시스템이다. 매일 또는 매달, 소스 시스템에서 데이터를 SAM 파일로 만들어서 데이터 웨어하우스로 보내는 오프라인(offline) 방식이나 소스 시스템 DB와 데이터 웨어하우스 DB를 직접 연결하여 DB내에 있는 데이터를 직접 로딩하는 온라인(online) 방식을 택한다. 또는 운영 시스템내에 사용자 로그 파일을 만들어 데이터 웨어하우스 서버가 이 로그 파일을 주기적으로 체크하여 데이터 웨어하우스로 가져오는 semi-online 방식도 있다. 이 모든 경우, 소스 시스템에서 넘어온 데이터를 데이터 웨어하우스가 가공해야 한다.

2.2.1. 초기 데이터와 주기적 데이터

데이터 웨어하우스는 과거 수 년간의 데이터를 보관해야 한다. 그리고, 시스템을 가동하면 주기적으로 소스 시스템에서 데이터가 넘어 오게 된다. 따라서 데이터 웨어하우스에 들어가게 되는 데이터는 초기 데이터와 주기적 데이터로 나눌 수 있다.

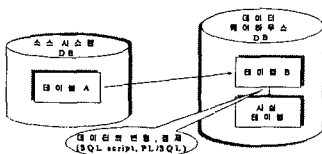
초기데이터는 과거 수년간의 데이터이다. 이러한 데이터는 대개 소스 시스템에서 갖고 있지 않고 MT 형태로 보관되어 있게 된다. 실제로는 사용자의 요구 조건에 따라서 이를 생략하고 최근 수 개월 데이터만 올리는 경우가 있으나 대개의 경우는 수 년간의 데이터를 초기에 데이터 웨어하우스로 로드해야 한다. 이를 초기 로딩(initial loading)이라고 한다. 초기 로딩시에는 많은 데이터를 한꺼번에 로드해야 하므로 그리고 데이터가 소스 시스템에 있지 않으므로 오프라인 방식을 택한다.

주기적 데이터는 시스템이 가동하기 시작하면 일 단위 또는 월 단위로 소스 시스템에서 데이터 웨어하우스로 넘어오는 데이터를 말한다. 이 데이터의 양이 많을 경우는 오프라인 방식으로 처리하는 것이 훨씬 유리하다. 데이터의 양이 많지 않다면 온라인 방식으로 처리한다.

2.2.2. 데이터의 변환(Transformation)

소스 시스템에서 추출한 데이터를 데이터 웨어하우스에서는 그대로 사용할 수 없다. 그 이유는 데이터 웨어하우스의 사실 테이블은 특정한 포맷으로 되어 있기 때문이다. 그리고 사용자의 요구가 간단하지 않기 때문에 여러 개의 소스 시스템 테이블을 조합해야 한다. 예를 들면, 매출이라는 주제 영역에서 총매출액과 신용 판매 금액을 보기를 원하는데 총매출액은 매출관리 시스템, 신용 판매 금액은 신용 판매 시스템에 따로 들어 있다면 당연히 두 시스템의 테이블을 합해야 한다. 또한, 소스 데이터의 코드가 일치 않기 때문이다. 예를 들면, 제품 코드가 매출 관리 시스템과 신용 판매 시스템과 상이하다면 이를 하나로 통일해야 한다.

ETT의 온라인 방식은 소스 시스템 DB에서 데이터를 데이터 웨어하우스로 직접 내리는 방식이다. 이 테이블들은 소스 시스템의 테이블과 똑 같은 값과 포맷으로 되어 있다. 따라서 사실 테이블을 만들어 주기 위해서는 데이터의 변형, 정제 작업등이 오프라인 방식과 동일하게 수행되어야 한다. 이를 위해서는 주로 SQL script나 PL/SQL을 이용한다.



<그림 4> 온라인 추출 방식

데이터 웨어하우스 구축시 데이터의 정확성을 보장하기가 어렵다. 이것은 데이터 웨어하우스의 생명과 직결된 아주 절박한 문제이다. 왜냐하면 현업은 데이터가 틀린 시스템을 사용하지 않을 것이며, 사용되지 않은 데이터 웨어하우스는 아무리 잘 구축했다고 해도 실패한 것이나 마찬가지이다.

추출 데이터의 정확성을 보장하기 위해서는 소스 시스템에서 추출한 데이터의 건수와 특정 필드 값을 합산한 것이 데이터 웨어하우스로 데이터 로드가 끝난 후 데이터의 건수와 그 필드의 값을 합산한 것과 비교한다. 또한 데이터의 추출과 로드시 에러가 생길 경우 에러가 생긴 데이터를 모아서 에러의 원인을 밝히고 교정을 해야 한다.

ROLAP 서버는 데이터를 가지고 있지 않으며, 실제 데이터는 모두 데이터 웨어하우스에 있다.

3. 결론 및 향후과제

인터넷 쇼핑몰을 통해 물품을 구매하는 소비자의 다양한 구매 패턴을 데이터베이스로 저장하여 이 관계형 DB를 근간으로 ROLAP을 이용한 데이터 웨어하우스를 구축하거나 고객 이름, 주소, 전화번호와 같이 원시 데이터를 출력하는 것이 중요한 데이터베이스 마케팅(DB marketing) 같은 분야에 이를 활용함으로써, 최근 데이터 웨어하우스 구축이 보편화 되면서 데이터 웨어하우스와 고객 DB를 통해 세분 고객별로 구매량을 파악하거나 주요 고객이 누구이며 수익 공헌도가 높은 회원은 어떤 분류인지 파악케 하여 각종 마케팅 활동, 판촉활동이 고객의 구매/이용 패턴에 어떤 영향을 줬는지, 상품과 고객간의 인과관계 분석을 가능케 하고 신규 고객의 개척보다는 기존 고객의 이탈 방지에 더욱 비중을 두는 고객의 유지 마케팅(Retention Marketing)에 더욱 비중을 두는 데이터베이스 마케팅에 활용한다면 효과적일 것이다.

앞으로의 기업의 요구사항이 주로 고객관리에 중점을 두는 데이터베이스 마케팅쪽으로 가고 있기 때문에 이쪽의 최적의 솔루션인 데이터 마이닝은 필연적이다. 그리고 웹을 이용한 데이터의 액세스가 보편화되면서 인공지능의 기법을 사용한 데이터 마이닝도 앞으로 구축해 볼만할 것이다.

4. 참고문헌

- [1] 데이터 웨어하우스, 오라클, 장동인
- [2] The Data Warehouse Toolkit, Ralph Kimball, Wiley, 1996
- [3] Building the Data Warehouse, W. H. Inmon, 1996
- [4] 데이터 웨어하우스의 구축과 활용(Using the Data Warehouse), W. H. Inmon, Richard Hackathorn, 김성수 역, 1996
- [5] Data Warehouse from Architecture to Implementation, Barry Devlin, Addison Wesley, 1997