

OLAP이란 무엇인가?

박 종수, 성신여자대학교 컴퓨터정보학부

jpark@cs.sungshin.ac.kr

(프로그래밍세계, 2001년 10월호, pp. 186-193)

1. OLAP의 위치

정보화 시대에서 우리는 무수한 용어들과 만나고 있다. 우리의 주제에 관련된 용어들을 나열해보면, DB, DBMS, DM, DSS, DW, ERP, EIS, KDD, MDB, OLTP, OLAP, OLAM, RDB, SQL 등이 있다. 이런 다양한 용어 중에서 OLAP이 위치하고 있는 부분이 어디인지를 쉽게 알 수 있다면 앞으로의 설명에 많은 도움이 될 것이다.

그림 1은 최근에 많은 연구가 이루어졌고 실제 기업 단위에서 구현되고 있는 3-계층 데이터 웨어하우스 구조를 나타낸다[6]. 이 그림은 80년대 이후부터 쌓여져 온 대용량의 데이터에서 정보나 지식을 추출해내기 위한 흐름도이다. 맨 아래쪽의 데이터는 내부의 기본적인 운영 데이터베이스에서 나오거나 외부에서 가져온 데이터에 해당되고, 이런 데이터를 통합하여 데이터 웨어하우스에 저장한다. 이 데이터 웨어하우스의 데이터를 입력으로 하여 OLAP 도구들은 원하는 보고서, 분석, 또는 정보를 추출해내고 있다.

국내 상황을 살펴보면, 80년대 중반 이후 다수의 기업들이 관계형 데이터베이스 시스템을 도입하여 운영 데이터베이스로 많이 활용되고 있다. 그리고, 90년대 말 이후 데이터 웨어하우스에 관심을 가지게 되었고, 요즈음 이런 시스템을 구축하고 있는 기업들이 늘어나고 있다. 대용량의 데이터를 처리하는 시스템을 갖추고 나서, 이런 데이터를 조작해서 정보나 지식이나 주요한 패턴을 찾아내는 도구로 OLAP이 사용되고 있다. 물론 MS의 Excel과 같이 소량의 데이터에서도 보고서를 만들 수 있고 그러한 데이터를 분석할 수는 있지만, 역시 대용량의 데이터를 다루고 저장해놓은 기업체나 관공서 등에서 OLAP 도구를 적용해서 좋은 결과를 얻을 수 있다고 판단된다.

1.1 OLAP의 목적

그림 1에서 OLAP을 기준으로 설명하면, 먼저 데이터 웨어하우스(DW, Data Warehouse)는 대용량 데이터가 들어 있는 저장고(repository)이고, OLAP(On-Line Analytical Processing)은 이 저장고에 있는 데이터에 액세스하여 데이터를 조작하고 분석하는 방식이다. 운영 데이터베이스(Operational databases)는 주로 관계형 데이터베이스 관리 시스템(RDBMS)에 의해 운영되고 관리되는 데이터베이스다. 이러한 데이터베이스 시스템에서 매출 전표 기록이나 재고 관

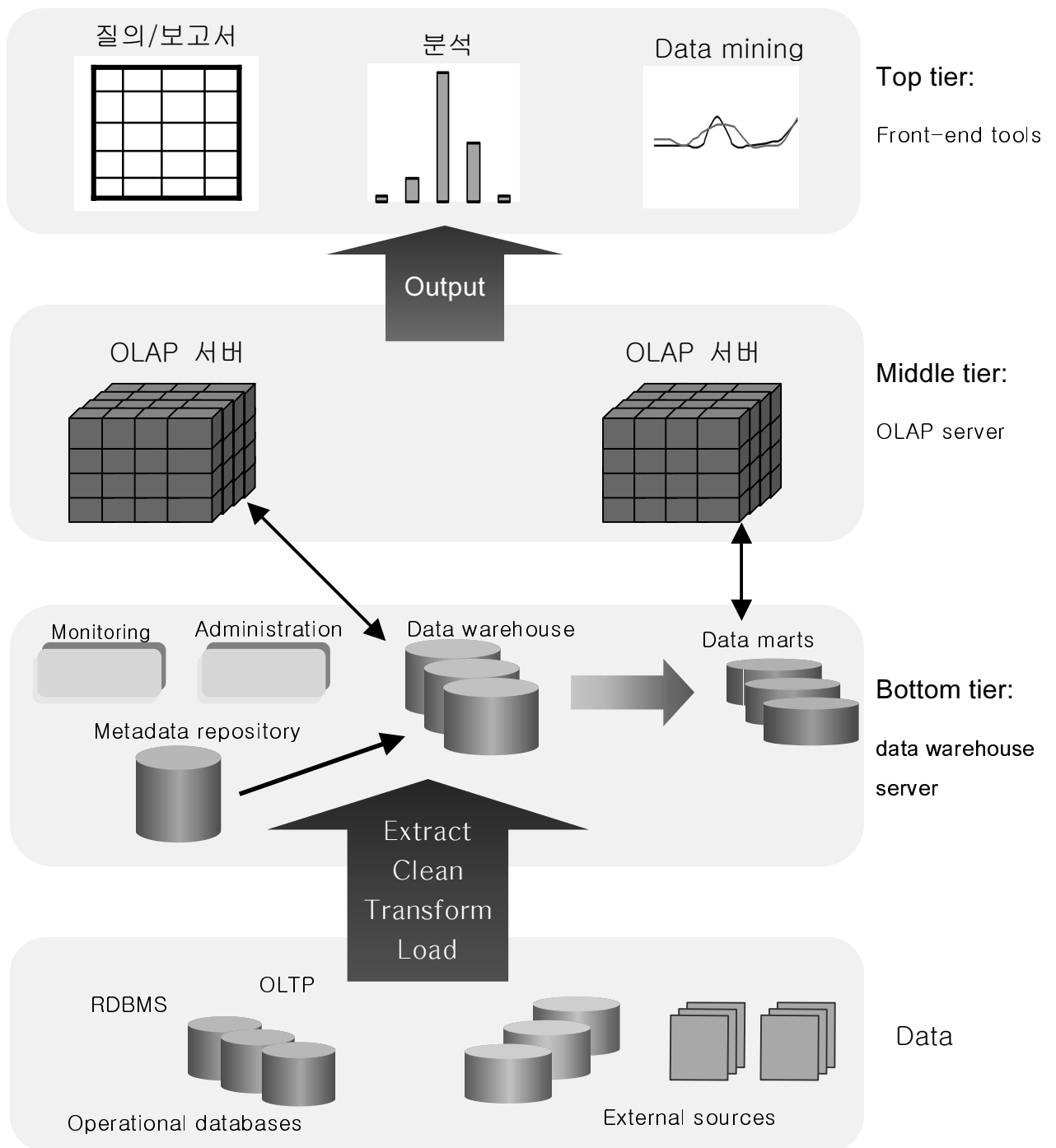


그림 1. 3계층 데이터 웨어하우징 구조

리 등의 기본 데이터를 수집하고 관리하는 운영 시스템을 OLTP(On-Line Transaction Processing)라고 한다.

앞에서 설명한 바와 같이 **OLAP**은 분석이나 관리 목적으로 다차원 데이터를 수집하고 관리하고 처리하고 보여주기 위한 응용과 기술들의 집합 도구다. 앞에서 이야기한 것과 같이 **OLAP**이 적용될 수 있는 환경은 대 용량의 데이터가 저장되어 있는 기업 단위의 데이터 웨어하우스나 또는 특정 부서에서 관리하는 데이터 마트(**Data Mart**)가 존재해야 되는 것이다. **OLAP**의 목적은 최종사용자가 기업의 전반적인 상황을 이해할 수 있게 하고 의사결정을 지원하는 데 있다. **OLTP**는 매일 매일의 기업운영을 가능하게 하는 반면, **OLAP**은 기업이 나아가야 할 방향을 설정할 수 있게 한다.

1.2 OLAP과 관련된 용어 설명

외부 자료(**External sources**)

내부의 데이터베이스 시스템이 관리하는 데이터가 아니라, 외부에서 구해오는 여러가지 데이터를 총칭해서 외부 소스라고 한다. 주로 파일 단위로 구성된다. 신문이나 정부 통계 자료나 환율표 등을 나타낸다.

관계형 데이터베이스(**Relational databases**)

관계형 데이터베이스는 **Codd** 박사가 제안한 관계형 모델에 따라 이루어진 데이터베이스다. 데이터베이스는 테이블들의 집합이고, 각 테이블은 유일한 이름이 배정된다. 각 테이블은 애트리뷰트들(attributes, columns or fields)의 집합이고 보통 튜플들(tuples, records or rows)의 큰 집합으로 저장된다. 관계형 테이블에서 각 튜플은 유일한 키로 구분되고 그리고 애트리뷰트들의 값의 집합으로 묘사되는 오브젝트를 나타낸다. 관계형 데이터베이스 시스템(**RDBMS**)의 예로는 오라클, IBM사의 DB2, 그리고 MS사의 SQL 2000 등이 있다.

데이터 웨어하우스(**Data warehouse**)

1996년 W.H. Inmon에 의하면, “데이터 웨어하우스란 관리자의 의사 결정을 지원하기 위한 주제 중심의(subject oriented), 통합된(integrated), 비휘발성의(nonvolatile), 시간변이적인(time variant) 데이터 집합이다” 라고 하였다[1]. 데이터 웨어하우스는 보통 전사적 웨어하우스로 모델링되어 단계적으로 구축되고 있다. 데이터 웨어하우스를 위해 가장 많이 사용하는 모델은 다차원 모델이고, 그런 모델로는 스타 스키마(star schema), 스노우플레이커 스키마

(snowflake schema), 또는 사실 컨스텔레이션 스키마(fact constellation schema) 등이 있다.

- 1) 주제 중심의: 데이터 웨어하우스는 주요 주제 중심으로 구성한다. 주요 주제는 고객, 공급자, 생산, 판매 등이다. 일일 운영에 집중하거나 조직의 트랜잭션 처리 보다는 오히려 데이터 웨어하우스는 의사 결정자를 위해 데이터의 모델링과 분석에 초점을 맞춘다. 그러므로, 데이터 웨어하우스는 의사결정 지원 과정에 유용하지 않은 데이터를 제외함으로 특징적으로 특정 주제 문제에 대한 간단하고 간결한 관점을 제공한다.
- 2) 통합된: 데이터 웨어하우스는 보통 관계형 데이터베이스, 단순한 파일, 그리고 온라인 트랜잭션 레코드와 같은 다양하고 이종의 소스들을 통합함으로 만들어진다. 명칭 부여에 대한 관례, 암호화 구조, 에트리뷰트 측정, 등등에서 일관성을 보장하기 위해 데이터 클리닝과 데이터 통합 기술이 적용된다.
- 3) 시간변이적인: 데이터는 역사적 관점에서 정보를 제공하기 위해 저장된다(예를 들면, 과거 5 - 10년). 데이터 웨어하우스에서 모든 키 구조는 묵시적이거나 명시적으로 시간 요소를 포함한다.
- 4) 비휘발성의: 데이터 웨어하우스는 항상 운영 환경에서 만들어진 응용 데이터에서 변환된 데이터를 물리적으로 분리되게 저장한 것을 일컫는다. 이 분리로 인하여, 데이터 웨어하우스는 트랜잭션 처리, 복구, 그리고 동시 제어 메커니즘을 요구하지 않는다. 이것은 데이터를 액세스함에 있어서 보통 단지 2개의 연산만 필요로 한다: 초기 데이터 로드 연산과 데이터 액세스 연산.

데이터 마트(Data mart)

데이터 마트는 특정 그룹의 사용자들에게 가치있는 전사적 데이터의 부분 집합을 포함한다. 그 범위는 특별히 선택된 주제로 제한된다. 예를 들면, 마케팅 데이터 마트는 고객, 품목, 그리고 판매에 관한 주제로 제한될 것이다. 데이터 마트에 포함된 데이터는 요약되는 경향이 있다. 데이터 마트는 UNIX나 Windows/NT 기반의 낮은 비용으로 한 부서 단위의 서버에 보통 구축되고 있다. 데이터 마트의 구축 주기는 월 또는 년 단위 보다는 주 단위로 하는 경우가 많다. 그러나, 그것의 설계와 계획이 전사적이지 않다면 오랜 사용으로 복잡한 통합을 내포할 수도 있다.

OLTP와 OLAP 시스템간의 비교

특징(Feature)	OLTP	OLAP
특성	운영 처리	정보 처리
방침	트랜잭션	분석
사용자	직원, DBA, 데이터베이스 전문가	지식 노동자 (즉, 관리자, 이사, 분석가)
기능	일일 운영	장기간 정보 요구사항, 의사결정지원
DB 설계	ER 기반, 응용 위주	스타/스노우플레이커, 주제중심
데이터	현재; 최신 데이터 보장	역사적; 시간에 걸쳐 유지된 정확성
요약	기초적인, 아주 상세함	요약된, 통합 정리됨
뷰	상세, 평면적인 관계형	요약된, 다차원적인
일의 단위	짧고, 간단한 트랜잭션	복잡한 질의
액세스	읽기/쓰기	대개 읽기
조점	데이터 입력	정보 출력
운영	주키에 인덱스/해쉬	많은 스캔
액세스된 레코드의 수	수십개	수백개
사용자의 수	수천명	수백명
DB 크기	100MB에서 GB	100GB에서 TB
우선순위	고성능, 높은 가용성	높은 융통성, 최종 사용자 자율성
측정기준	트랜잭션 처리량	질의 처리량, 응답 시간

데이터 마이닝(Data mining)

데이터 마이닝은 대량의 데이터에서 유용한 패턴을 발견해내는 작업이다. 여기서 데이터는 데이터베이스, 데이터 웨어하우스, 또는 다른 정보 저장소에 저장되어 있을 수 있다. 이것은 데이터베이스 시스템, 데이터 웨어하우징, 통계학, 기계 학습, 데이터 시각화, 정보 검색, 그리고 고성능 컴퓨팅 등의 영역을 아우르는 초기 단계의 학제간 제휴 분야다. 이 이외에 기여하는 영역으로는 신경망, 패턴 인식, 공간 데이터 분석, 이미지 데이터베이스, 신호 처리, 그리고 많은 응용 분야를 포함한다. 여기서 응용 분야는 사업, 경제학, 그리고 생물정보학

등이다.

KDD

많은 사람들이 데이터 마이닝을 또 하나의 널리 사용되는 용어인 데이터베이스에서 지식 탐사((Knowledge Discovery in Databases , KDD)에 대한 동의어로 취급하고 있다. 반대로, 그와 다른 사람들은 데이터 마이닝을 데이터베이스에서 지식 탐사의 과정에서 단순히 중요한 단계로 보고 있다. 이 글에서도 후자에 동의하고 있다. KDD를 하나의 프로세스로 간주하면 다음과 같이 7 단계로 구성되고[6], 단계 5가 데이터 마이닝임을 알 수 있다.

- 1) 데이터 클리닝(Data cleaning): 필요 없거나(noise) 일관성이 없는 데이터를 삭제한다.
- 2) 데이터 통합(Data integration): 다중 데이터 소스들(sources)이 결합된다.
- 3) 데이터 선택(Data selection): 데이터베이스에서 분석 작업에 관련된 데이터만 가져온다.
- 4) 데이터 변환(Data transformation): 예를 들면, 데이터에 요약이나 총계 연산을 수행함으로써 마이닝에 적절한 형태로 데이터를 변환하거나 통합 정리한다.
- 5) 데이터 마이닝(Data mining): 데이터 패턴을 추출해내기 위하여 지능적인 방법이 적용되는 필수적인 과정.
- 6) 패턴 평가(Pattern evaluation): 어떤 유용성 측정을 통하여 지식을 나타내는 확실히 유용한 패턴을 구분해내는 것.
- 7) 지식 표현(Knowledge representation): 사용자에게 탐사된 지식을 보여주기 위하여 시각화와 지식 표현 기술이 사용된다.

2. OLAP이란 무엇인가?

“OLAP”이란 용어는 맨 처음 E.F. Codd가 쓴 글에서 1993년에 갑자기 등장하였다. Codd 박사는 OLAP을 위한 12가지 규칙을 제안하였다. 그리고, 1995년에 Pendse와 Creeth는 기술적 측면과 업무의 요구사항을 적절히 조합하여 OLAP을 ‘공유되는 다차원 정보에 대한 신속한 분석(FASMI)’으로 정의하였다[7]. 펜지와 크리스의 정의를 보완하여 OLAP을 ‘최종 사용자가 다차원 정보에 직접 접근하여 대화식으로 정보를 분석하고 의사결정에 활용하는 과정’으로

정의하면[2], 우리는 먼저 **OLAP**의 가장 기본적인 특성인 다차원 (정보)와 (대화식인) 발견적 분석에 대해서 알아본다. 그리고, **Pendse**와 **Creeth**의 **FASMI**에 대해서 알아보고, 다음으로 **Codd** 박사가 제시한 **OLAP**을 위한 규칙(rules)과 특징(features)에 대해서 알아본다.

2.1 다차원

모든 OLAP 도구의 밑바탕은 항상 다차원 데이터 모델(Multidimensional data model)이라고 지칭되는 개념 데이터 모델이다. 업무 처리를 다차원 모델로 변환하는 과정은 “차원 모델링(dimensional modeling)” 또는 좀더 일상적인 용어로 “다차원 모델링(multi-dimensional modeling: MDM)”이라고 불리운다. MDM은 업무 모델을 업무의 일상적인 측면으로 설명되는 일련의 수단으로 개념화하기 위한 기법이다. 이것은 분석을 용이하게 하기 위해 데이터를 이전, 요약 및 배치하는 데 특히 유용하다. MDM은 사실, 차원, 계층 및 회소성과 같은 구조물을 사용한다. 다차원 모델들은 값, 계수, 가중치, 그리고 발생횟수와 같은 숫자 데이터를 중심으로 설계된다. 전형적인 OLTP의 문제가 “주문 이행 과정을 모형화”하는 것이라면, MDM의 문제는 “시간이 지남에 따라 고객에 의한, 조직에 의한 나의 수익성은 무엇인가?”이다. 분석을 위해 활용되는 정보의 형태는 다차원적이라는 사실이다. 다차원 정보는 사용자들에 의해 이해되는 기업의 실제 차원(기간, 제품, 부서, 지역 등)을 반영한다. 구체적인 예는 그림 2에 있고, 이를 참고하면 다차원 큐브를 쉽게 이해할 수 있다.

2.2 발견적 분석

OLAP이 항상 여러 통로를 통과하는 한 줄의 분석 끈을 따르는 대화식 데이터 조회를 포함한다는 점은 훨씬 더 명백하다. 한 가지 예는 요약 데이터를 표시하고, 그 뒤를 따라 연속적으로 더 낮은 수준의 세부 사항으로 탐구해 내려가는 것이다. 흔히 OLAP을 “발견적”이라고 말하는데, 이것은 분석 세션이 하나의 질문으로 시작하고, 각각의 연속된 질문이 이전의 한 질문에서 파생된다는 것을 의미한다. 대화식의 발견적 분석은 느린 데이터베이스의 응답을 기다리다가 사고의 흐름이 흩어지지 않도록 특별한 성능을 필요로 한다. 그리고, 최종 사용자는 중간 매개자(전산부서)나 매개체(리포트) 없이 온라인상에서 직접 데이터에 접근한다.

2.3 Pendse와 Creeth의 FASMI

펜지와 크리스는 1995 년에 OLAP 의 정의를 기억하기 쉽게 5 단어로 요약하였다. 즉, **Fast Analysis of Shared Multidimensional Information** — 또는 짧게 표현하면 **FASMI**. 이런 정의는 OLAP 을 어떻게 구현하느냐에 관계없이 OLAP 응용의 특성을 정의하려는 것이다. 아래는 각 단어에 대해서 설명한다.

- 1) **FAST**: 빠르다는 의미는 시스템이 사용자에게 약 5 초 이내에 대부분의 응답을 하도록 목표하는 것을 말한다. 간단한 분석은 1 초를 넘어서지 않고 아주 복잡한 것은 20 초를 넘어서는 경우는 작아야 된다. 시스템이 응답이 늦어질 것이라고 알려주더라도, 사용자는 늦어지면 정신이 산만해질 수 있고 사고의 연속성이 깨질 수도 있기 때문에 분석의 질이 떨어질 수 있다. 벤더들은 이 목표를 달성하기 위해 다양한 기술을 적용하고 있다. 이 중에는 특별한 형태의 데이터 저장 기술, 광범위한 사전-계산, 그리고 특별한 하드웨어 요구사항 등을 포함한다.
- 2) **ANALYSIS**: 분석은 시스템이 응용과 사용자에게 관련되어 있는 어떤 사업 논리와 통계 분석에도 대처할 수 있고, 그리고 목표하는 사용자에게 그것이 충분히 쉽게 유지되어야 한다는 것을 의미한다. 이런 분석에는 시계열 분석, 비용 할당, 환율 변동, 목표 도달, 일반적인 다차원 구조 변화, 비절차적 모델링, 예외 경고, 데이터 마이닝 그리고 다른 응용에 따른 특성 등을 포함한다. 이런 성능은 제품의 목표 시장에 따라 제품간에 아주 다르게 된다.
- 3) **SHARED**: 공유는 시스템이 기밀을 위해서 셀(cell) 수준까지 보안 요구사항을 구현해야 되고, 만약 다중 쓰기 액세스가 필요하면 적절한 수준에서 동시 업데이트 로킹이 구현되어야 한다는 것을 의미한다. 모든 응용이 사용자에게 데이터를 다시 쓰는 것을 요구하지는 않지만, 이렇게 하도록 하는 요구가 증가하고 있으므로 시스템은 제때 안전하게 다중 업데이트를 처리할 수 있어야 한다. 이것이 많은 OLAP 제품에서 취약한 주요 영역이다. 이런 제품들은 모든 OLAP 응용들이 간단한 보안 제어로 읽기-전용이라고 가정하는 경향이 있다.
- 4) **MULTIDIMENSIONAL**: 다차원은 핵심적인 요구 사항이다. 만약 OLAP 을 한-단어 정의로 요구한다면, 이 단어이다. 시스템은 데이터의 다차원 개념 시각을 지원해야 한다. 이는 계층과 다중 계층을 완전히 지원하는 것을 포함하고, 이런 계층 지원은 확실히 사업과 조직을 분석하는 대개의 논리적인 방법인 것이다.
- 5) **INFORMATION**: 정보는 데이터의 전부이고 필요하다면 유도되어진 정보를 나타낸다. 그것이 어디에 있던지 그리고 응용에 얼마나 많이 관련되었는지 간에 상관을 앎는다. 여러 제품의 성능을 얼마나 큰 입력 데이터를 처리할 수 있는 가에 따라 측정할 수 있지만, 얼마나 많은 기가바이트로 저장하는 정도로는 측정하지 않는다.

이러한 OLAP 의 목표를 달성하기 위한 기술로는 클라이언트/서버 구조, 시계열 분석, 객체-지향, 최적화된 전용 저장 장치, 멀티스레딩과 벤더들이 자랑하는 다양한 특허로 인정된 아이디어 등이 포함된다.

2.4 Codd 규칙과 특징들

1993년에 E.F. Codd는 그 당시 아머 소프트웨어 사에 'Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate'이라는 제목의 백서를 제출하였다. 코드 박사는 1960년대와 1980년대 후반에 걸쳐 유명한 데이터베이스 연구자로 잘 알려져 있었고, 관계형 데이터 모델의 제안자로 인정받고 있다. 그러나, 그의 OLAP 규칙은 수학적인 기본보다는 상품 판매상의 입장에서 제안되었기에 논란이 많은 것으로 여겨지고 있다. 그래서, 이 백서는 학술 논문으로 보다는 오히려 판매상-출판 소책자로 간주되어질 수도 있다. 그러나, Codd 박사가 제시한 규칙들은 많은 사람들이 크게 유용하지 않다는 데 동의하지만, 대부분의 그의 분석은 OLAP 주제에 관한 결정적인 언급으로 남아있다. 원래 OLAP 백서는 12개의 규칙들을 포함하고 있고, 이는 잘 알려져 있다. 1995년에 이 규칙들에다 잘 알려지지 않은 또 다른 6개의 규칙들이 추가되었고, 코드 박사는 이 규칙들을 특징(feature)이라 부르는 4개의 그룹으로 재조정하였다. 표 1은 첫 번째 그룹인 기본 특징 B에 속한 규칙들을 나열하였다.

표 1. OLAP을 위한 Basic Features B

F1: 다차원 개념 시각(Original Rule 1)
F2: 직관적인 데이터 조작 (Original Rule 10)
F3: 접근성: OLAP as a Mediator (Original Rule 3)
F4: Batch Extraction vs Interpretive (New)
F5: OLAP 분석 모델 (New).
F6: 클라이언트/서버 구조 (Original Rule 5).
F7: 투명성 (Original Rule 2)
F8: 다중-사용자 지원 (Original Rule 8)

나머지 3개 그룹과 해당 특징은 다음과 같다. 두 번째 그룹 Special features S는 F9: 비정규화된 데이터의 처리 (New), F10: OLAP 결과 저장: Keeping Them Separate from Source Data

(New), F11: 분실된 값의 추출 (New), F12: 분실된 값의 처리 (New)로 구성된다. 세 번째 그룹 Reporting Features R은 F13: 융통성있는 보고 (Original Rule 11), F14: 일관성있는 보고 수행 (Original Rule 4), F15: 물리적 수준의 자동조정(원래는 동적 회소 행렬 취급) (Supersedes Original Rule 7)로 이루어진다. 마지막으로 네 번째 그룹인 Dimension Control D는 F16: 총체적 차원성 (Original Rule 6), F17: 무제한 차원과 총계 수준 (Original Rule 12), F18: 무제한 교차 차원 운영 (Original Rule 9)으로 구성된다.

18개의 특징들 중에서 OLAP을 위한 Codd의 기본 특징들 5가지에 대해서는 상세히 알아보고, 나머지 특징들은 참고 문헌에서 찾을 수 있다[7].

- 1) F1: Multidimensional Conceptual View (Original Rule 1). 누구도 이 특징에 대해서 논란을 제기하지 않을 것이다. OLAP의 중심적인 핵심으로 여긴다. 이 요구 사항을 위해 Codd 박사는 ‘slice and dice’ 연산을 포함시켰다.
- 2) F2: Intuitive Data Manipulation (Original Rule 10). 코드 박사는 메뉴나 다중 동작(actions)에 의지하지 않고 뷰에서 셀들에 직접적인 동작을 통해 이루어지는 데이터 조작을 선호했다.
- 3) F3: Accessibility: OLAP as a Mediator (Original Rule 3). 코드 박사는 기본적으로 OLAP 엔진을 상이한 데이터 소스들과 OLAP 전위 사이에 위치한 미들웨어로 설명하고 있다.
- 4) F4: Batch Extraction vs Interpretive (New). 이 규칙은 제품들이 OLAP 데이터를 위한 그 자신의 중간 적재 데이터베이스 기능 뿐만 아니라 외부 데이터를 직접 액세스할 수 있도록 하는 것을 요구하고 있다. 코드 박사는 다차원 데이터 적재와 더불어 대규모의 다차원 데이터베이스의 부분적인 사전-계산을 찬성하였다. 더구나 이런 데이터에 대해서는 기본이 되는 상세에 이르기 까지 투명한 접근을 요한다. 오늘날, 이것은 hybrid OLAP의 정의로 간주될 수 있고, 유행하는 구조로 되고 있다.
- 5) F5: OLAP Analysis Models (New). OLAP 제품은 네 종류의 분석 모델을 지원할 것을 코드 박사는 요구하고 있다. 백서에 설명된 네 모델은 Categorical, Exegetical, Contemplative, 그리고 Formulaic 모델이다. 보통 모든 OLAP 도구들은 첫 두 모델들을 지원하고, 대개는 세 번째 모델을 어느 정도까지 지원한다. 그러나, 네 번째 모델을 사용할 정도로 지원하는 OLAP 도구는 드물다. 아마도 코드 박사는 이 규칙에

대해서 데이터 마이닝을 기대했을 것으로 예상된다.

3. OLAP의 연산들

다차원 모델에서 데이터는 여러 차원으로 구성되고, 각 차원은 개념 계층에 의해 정의된 추상화의 여러 수준을 포함한다. 이런 구성은 사용자에게 다른 관점에서 데이터를 보는 다양성을 제공한다. 대화식의 질문하기와 바로 가까워서 데이터의 분석을 허용하면서, 이러한 다른 관점을 실재화하기 위해서 다수의 OLAP 데이터 큐브 연산이 있다. 따라서, 대화형 데이터 분석을 위해 OLAP은 사용자-편의성을 가진 환경을 제공한다. 다음은 다차원 데이터 모델에서의 OLAP 연산에 대한 설명이다[6].

다차원 데이터에서 대표적인 OLAP 연산에 대해서 살펴보자. 아래 설명된 각 연산은 그림 2에 그려져 있다. 그림 2의 가운데에 있는 것은 가전제품 판매를 보여주는 데이터 큐브이다. 이 큐브는 location, time, 그리고 item 차원을 포함한다. 여기서 location은 도시 이름에 따라 총계산이 되어 있고, time은 분기별로 총계산이 되어 있고, 그리고 item은 품목 종류에 따라 총계산이 되어 있다. 쉽게 설명하기 위해 이 큐브를 중앙 큐브라 부르자. 표시된 측정 값은 백 만원 단위의 판매액이다. 읽기 쉽게 하기 위해서 큐브의 셀 일부분에만 값이 주어져 있다. 도시 이름으로 사용되어진 데이터는 서울, 부산, 동경 그리고 경도다.

Roll-up 연산

어떤 벤더들은 **drill-up** 연산이라고도 부른다. 하나의 차원 상에서 개념 계층상 위로 오르거나 또는 차원 축소에 의해 데이터 큐브에서 총계산을 수행하는 연산이다. 그림 2는 위치(location)에 대한 개념 계층상 위로 오르는 방식으로 중앙 큐브상에서 **roll-up** 연산을 수행한 결과이다. 이 계층은 **street < city < province_or_state < country**의 전체 순서로 정의된 것이다. 그림에 나타난 **roll-up** 연산은 **city** 수준에서 **country** 수준으로 위치 계층을 오름으로 데이터를 총계산한 것이다. 다른 말로 설명하면, **city**로 데이터를 그룹하는 것에 비해 **country**로 그룹한 큐브로 결과가 나온다. **Roll-up** 연산이 차원 축소로 연산할 때, 주어진 큐브에서 하나 또는 여러 차원이 제거된다. 예를 들면, location과 time 두 차원을 갖는 판매 데이터 큐브를

고려해보자, **time** 차원을 삭제하는 **roll-up** 연산이 수행되면, 위치와 시간에 의하기 보다는 위치에 의한 전체 판매의 총계산이 결과로 얻어질 것이다.

Drill-down 연산

Drill-down 연산은 **roll-up** 연산의 반대이다. 그것은 덜 상세한 데이터에서 더 상세한 데이터로 진행시킨다. 하나의 차원에서 개념 계층을 내리거나 또는 추가적인 차원들을 도입함으로써 **drill-down**을 현실화시킬 수 있다. 그림 2는 중앙 큐브 상에서 **day < month < quarter < year**로 정의된 **time**의 개념 계층을 내림으로 수행되어진 **drill-down** 연산의 결과를 보여준다. **Drill-down** 연산은 **quarter** 수준에서 **month** 수준으로 **time** 개념 계층을 하향시킴으로 이루어진다. 결과 데이터 큐브는 분기별로 요약된 것보다는 월별로 된 전체 판매를 열거하고 있다. **Drill-down**은 주어진 데이터에 상세함을 더하기 때문에, 큐브에 새로운 차원을 추가하는 경우도 발생할 수 있다. 예를 들면, 그림 2의 중앙 큐브상에서 **drill-down** 연산은 **customer_type**과 같은 추가적인 차원을 도입함으로써 이루어질 수 있다.

Slice and dice 연산

Slice 연산은 주어진 큐브의 한 차원 상에서 선택 연산(selection)을 수행하는 것이고, 그 결과 서버큐브가 만들어진다. 그림 2에서 판단 기준인 **time = "Q1"**을 사용하여 **time** 차원에서 중앙 큐브로부터 선택된 것이 판매 데이터인데, 이 과정이 **slice** 연산을 보여준다. **Dice** 연산은 두 개 이상의 차원에서 선택 연산이 수행되어진 서브큐브로 정의된다. 그림 2는 세 차원이 연관된 다음 판단 기준에 의해 중앙 큐브 상에서의 **dice** 연산을 보여준다: (**location = "서울" or "부산"**) and (**time = "Q1" or "Q2"**) and (**item = "TV" or "PC"**).

Pivot(rotate) 연산

보고서를 생성할 때 또는 보고서를 생성한 후에 동적으로 보고서의 축을 바꾸어 보는 연산이다. 데이터의 다른 표현을 제공하기 위해 뷰에서 데이터 축을 회전시키는 시각화 연산이다. 그림 2는 2-D slice 연산에서 **item**과 **location** 축이 회전된 것을 보여주는 **pivot** 연산을 나타낸다. 다른 예로는 3-D 큐브에서 축을 회전시키는 것을 포함하고 또는 3-D 큐브를 일련의 2-D 면들로 변환하는 것도 포함한다.

다른 OLAP 연산들

어떤 OLAP 시스템은 추가적인 **drilling** 연산을 제공한다. 예를 들면, **drill-across** 연산은 하나의 사실 테이블보다 많은 테이블들이 연관된(즉, **across**) 질의를 실행한다. **Drill-through** 연산은 데이터 큐브의 최저 수준을 지나서 후위 관계형 테이블까지 철저히 분석하게 하는 관계형 SQL 특성들을 사용하게 한다[3]. 또 다른 OLAP 연산은 리스트에서 상위 N 또는 하위 N 항목들을 나열하는 것을 포함하기도 하고, 이동 평균, 성장률, 이율, 내부 회수를, 감가 상각, 환율, 그리고 통계 함수를 포함한다.

4. OLAP의 종류

OLAP 서버는 사업 사용자에게 데이터 웨어하우스나 데이터 마트로 부터 다차원 데이터를 제공한다. 이 데이터가 어디에 어떻게 저장되어 있는가는 관심 사항이 아니다. 그렇지만, OLAP 서버의 물리적 구조나 구현은 데이터 저장에 대해서 고려해야 한다. 그림 1에서 OLAP과 데이터 웨어하우스의 관계를 고려하여, OLAP 처리를 위한 웨어하우스 서버의 구현은 다음을 포함한다.

4.1 다차원 OLAP(MOLAP)

이를 지원하는 서버는 배열(array)-기반으로 한 다차원 저장 엔진을 통해 다차원 뷰를 지원한다. 이 서버들은 다차원 뷰들을 직접 데이터 큐브 배열 구조에 배치한다. 예를 들면, 아버사의 EssBase는 MOLAP 서버의 한 종류다. 데이터 큐브를 사용하는 장점은 미리 계산되어 요약된 데이터에 빠르게 인덱싱할 수 있다는 것이다. 다차원 데이터 저장에서 데이터 집합이 드문드문하게 분포되어 있으면 저장 이용성은 낮아짐에 유의해야 한다. 그런 경우에는 sparse matrix 압축 기법이 사용되어야 한다. 많은 MOLAP 서버들은 드문드문한 데이터 집합과 밀집한 데이터 집합을 처리하기 위해서 2-수준 저장 개념을 채택한다: 밀집한 서브큐브로 확인되면 배열 구조에 저장하고, 대신에 드문드문한 서브큐브로는 효율적인 저장 이용을 위해 압축 기술을 적용한다. 모든 OLAP은 정의에 의하면 다차원 OLAP을 지원하게 되어 있다.

다차원 데이터베이스(multidimensional database: MDD)는 놀라운 조회 성능을 제공할 수 있으며, 그것은 대개 데이터가 접근될 방식을 예측하는 기능이다. MDD에 있는 정보는 관계형 데이터베이스보다 더 거친, 세분화되지 않은 형태로 저장되기 때문에 색인은 더 작고 일반적으로 메모리에 상주한다. 일단 메모리 내의 색인이 스캔되면 데이터베이스로부터 두세 페이지가 이끌려 나온다. 어떤 도구들은 이러한 페이지들을 공유 메모리에 캐시하도록 설계되어 있어서, 더 향상된 성능을 제공한다. 응용 프로그램 설계자가 이용 패턴에 관하여 올바른 가정을 했을 경우, 이 체제는 상당히 효과를 나타낸다. MDD의 흥미로운 현상은 정보가 배열에 저장된다는 것으로, 이것은 배열에 있는 값들이 색인에 영향을 주지 않고 갱신될 수 있다는 것을 의미한다.

4.2 관계형 OLAP(ROLAP)

ROLAP을 지원하는 서버는 관계형 후위 서버와 클라이언트 전위 도구 사이에 위치하는 중간 서버를 지칭한다. 웨어하우스 데이터를 저장하고 관리하기 위해서 이 서버들은 관계형 또는 확장된-관계형 DBMS를 사용하고, 그리고 손실된 조각들을 지원하는 OLAP 미들웨어를 사용한다. ROLAP 서버는 각 DBMS 후위를 위한 최적화, 집계 순회 논리(aggregation navigation logic)의 구현, 그리고 추가적인 도구와 서비스를 포함한다. ROLAP 기술은 MOLAP 기술보다 더 큰 확장성(scalability)을 가진 것으로 여겨진다. 예를 들면, 마이크로스트래티지사의 DSS 서버와 인포믹서사의 Metacube는 ROLAP 접근 방식을 채택하였다.

일부의 전문가들은 다음과 같은 기준을 만족할 경우에만 그 제품이 관계형 OLAP이라고 간주한다[4].

- 1) 다중 경로 선택 또는 상관관계가 있는 하위 조회를 작성할 수 있고, 사소하지 않는 순위 결정, 비교 및 %-대-등급 계산을 작성할 수 있을 만큼 강력한 SQL 생성기를 가지고 있을 경우.
- 2) SQL 확장자를 포함하여 대상 데이터베이스를 위해 최적화된 SQL을 생성하는 경우.
- 3) 메타 데이터를 통해 모델을 설명하고 조회를 구축하기 위해 실시간으로 메타 데이터를 사용하는 메커니즘을 포함할 경우.
- 4) 가능한 한 사용량을 감시할 능력을 갖추고, 적어도 성능을 위한 요약 표 구축에 조언을 할 수 있는 메커니즘을 포함할 경우.

- 5) 클라이언트, 서버, 그리고 데이터베이스의 맥락을 관리하기 위한 중간층 사이에서 응용 프로그램을 분할할 수 있는 능력을 가진 경우.

4.3 Hybrid OLAP(HOLAP)

ROLAP의 아주 큰 확장성과 MOLAP의 더 빠른 계산성능의 장점을 취해서, 혼합 OLAP 접근 방식은 ROLAP과 MOLAP 기술을 결합한다. 예를 들며, HOLAP 서버는 관계형 데이터베이스에 저장될 수 있는 상세 데이터의 대 용량을 허용하고, 반면에 총계산된 값들은 분리된 MOLAP의 저장소에 보관된다. 그러므로, HOLAP은 다차원 데이터베이스와 관계형 데이터베이스에 저장된 데이터를 동시에 다차원 분석에 제공할 수 있다. 마이크로소프트사의 SQL 7.0 OLAP Services는 혼합 OLAP 서버를 지원한다.

4.4 데이터베이스 OLAP(DOLAP)

관계형 데이터베이스 판매업체들이 데이터베이스/OLAP을 본떠서 사업 분석가들이 DOLAP이라고 이름붙인 ROLAP 기능을 그들의 엔진 속에 제공하게 한 도구들을 DOLAP이라 한다. 이는 관계형 데이터베이스에서 증가되는 OLAP 처리의 요구에 부응하기 위해서, 몇 개의 관계형 및 데이터 웨어하우징 회사들이(예, 인포믹스사의 Redbrick) 특성화된 SQL 서버를 만들어냈다. 이 특별한 서버는 읽기-전용 환경에서 스타와 스노우플레이크 스키마를 능가하는 SQL 질의를 지원하는 진보된 질의 언어와 질의 처리를 제공한다.

DOLAP을 Desktop OLAP으로 이야기하는 경우도 있다[5, 7]. DOLAP은 다차원 데이터의 저장 및 프로세싱이 모두 클라이언트에서 이루어진다. 분석에 필요한 데이터는 데이터베이스에서 추출되어 클라이언트에 특수한 파일 형태로 저장된다. 사용자는 클라이언트에 저장된 데이터를 대상으로 제한된 기능의 다차원 분석을 행한다.

4.5 OLAP을 선택하는 기준

선택 기준은 기능성, 성능, 확장성, 그리고 미래라는 다섯 가지 주요 범주로 나눈다[4]. 기능성은 결정적이고 눈에 보이는 항목이지만 일시적이라는 것이다. 이는 주로 다차원 데이터베이스를 선택할 지 관계형 OLAP 솔루션을 선택할 지의 문제이기에, 제품들의 특징과 기능들은 급속히 변화하고 있기 때문이다. 그리고, 프로젝트의 초기에 언급된 요건들이 끝까지

지 지속되는 경우가 드물다. 성능은 계약을 좌지우지하며, 따라서 이것은 강제적이라기보다 계약적 항목이다. 맞춤성, 확장성 및 미래는 가장 중요한 문제이다.

맞춤성에 관련된 요인들은 다음과 같다: 개발, 팻 클라이언트/썬 클라이언트(fat client/thin client), 네트워크 영향, 인터넷, 연결성 등이다. 확장성 면에서는 ROLAP이 유리하지만, MDD들도 빠르게 큰 데이터베이스를 처리하는 영역을 넓혀가고 있다. 미래에 관한 사항은 판매업체를 평가하는 항목과 일치하고 있다. 그 판매업체가 계속해서 신기술을 따라잡을 수 있을 것인가 또는 판매 제품의 전반적인 품질은 어떠한가 등을 평가해야 된다. 마지막으로 다음 표는 세 가지 OLAP 방식을 간단히 비교한 것이다[3].

표 2. 세 가지 OLAP 운영 방식

	ROLAP	MOLAP	HOLAP
기존구조	관계형 데이터베이스	다차원 데이터베이스	다차원 데이터베이스와 관계형 데이터베이스
대용량 데이터	O	X	O
원시 데이터 액세스	O	X	O
분석 기능	X	O	O
핵심기술	다차원 모델링	다차원 데이터베이스	다차원 데이터베이스 + 다차원모델링
적용	전사적 데이터 웨어 하우스	데이터 마트, EIS	데이터 마트, EIS

5. 맺는 말

FIFO와 비슷하게 GIGO라는 용어가 있다. 이는 “Garbage In Garbage Out”을 나타내고, 어떤 자료 처리나 정보 처리 시스템의 입력으로 좋은 데이터가 들어가지 않으면 좋은 결과나 정보를 얻을 수 없다는 의미다. OLAP 도구를 잘 선택하는 것도 중요하지만, 이를 사용할 대상인 입력 자료에 대해서도 아주 신중히 접근해야 된다는 것을 뜻한다.

6. 참고 문헌 및 **Web Site**

- [1] 유영일의 공역, 데이터웨어하우스 구축 방법론, 홍릉과학출판사, 1997.
- [2] 조재희 박성진, 데이터 웨어하우징과 OLAP, 대청, 1998.
- [3] 장동인, 실무자를 위한 데이터 웨어하우스, 대청, 1999.
- [4] 함문성역, 데이터웨어하우스, 도서출판 니드, 1999.
- [5] 조재희 박성진, OLAP 테크놀로지, 시그마컨설팅그룹, 1999.
- [6] J. Han and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2001.
- [7] N. Pendse and R. Creeth, <http://www.olapreport.com>