

데이터 웨어하우스

최 용 락 . 이 정 일

1. 데이터 웨어하우스 개념

1.1 데이터 웨어하우스의 정의

데이터 웨어하우스는 업무 트랜잭션을 처리하는 데이터베이스 시스템에서 사용자들이 필요로 하는 정보를 추출해서 가공된 데이터 형태로 구성되는 업무 분석을 위한 데이터베이스이다. 이는 데이터베이스 관련자들이 업무 처리와 관련된 데이터들은 잘 저장하지만, 저장된 데이터들을 제대로 활용하지 못한다는 점에 착안하여, 어떻게 하면 데이터베이스에 저장되어 있는 데이터들을 보다 유익하게 효율적으로 활용할 수 있는가하는 관점에서 시작한 개념으로써, 1980년대 후반 IBM에서 도입한 개념에서 시작되었고, “W. H. Inmon”에 의해 데이터 활용 전략으로 데이터 웨어하우스의 개념을 사용함으로써 관련자들의 관심을 집중시켰다.

데이터 웨어하우스에 대한 정의는 학자들에 따라 조금씩 다르게 표현 하였지만, “기업 또는 조직들에서 일정기간 정보시스템을 운영하며 발생한 기간계 업무 데이터와 외부 데이터들을 주제(Subject)별로 통합하여 다양한 분석을 제공함으로써 사용자들이 필요로 하는 의사결정을 지원하기위한 데이터들이 저장된 저장공간”을 의미한다.

먼저 “Barry Devlin”은 “다양한 운영환경으로부터 얻어지며 최종사용자가 사용하고 이해할 수 있는 방법으로 정리된 유용하며 완전하게 일관성을 유지하는 데이터 저장소”라고 정의했으며, “Sean Kelly”는 “기업 내의 수많은 플랫폼과 아키텍처에 구현된 다중의 데이터 모델에 대한 주소를 갖고 있는 주제 지향적인 기업의 데이터베이스”라고 정의한다. 그리고 버틀러 그룹(Butler Group)에서는 “데이터 웨어하우스를 구성한다는 것은 물리적으로 여러 곳에 분산되어 있는 데이터베이스 내에 존재하는 데이터들을 이용하여 사용자들의 의사결정 지원을 위한 하나의 논리적인 뷰(View)를 창출하는 것이다.”라고 정의하여 데이터 웨어하우스를 정의했다.

특히, “Bill Inmon”의 정의에 의하면 데이터 웨어하우스는 “기업이 경영 관리를 위해 필요로 하는 의사 결정 지원에 효율적으로 사용될 수 있는 응용 프로그램들이 활용하는 정보 기반을 제공하는 하나의 통합된 데이터 저장공간으로서 다양한 운영 환경의 운용 시스템들로부터 데이터들을 추출, 변환, 통합해서 요약한 읽기 전용 데이터베이스”로 정의한다.

1.2 데이터 웨어하우스의 필요성

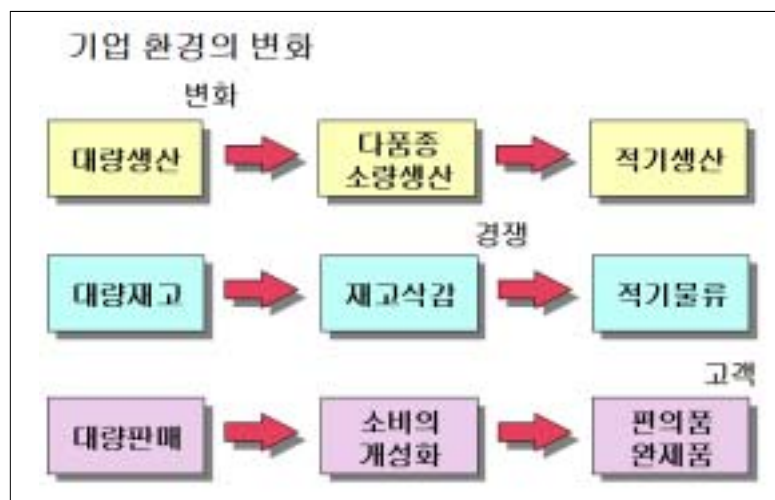
급변하는 기업의 환경 변화와 무한 경쟁 시대에서 생존 및 성장을 위해 기업을 운영하는 최근의 기업들은 전략적인 정보를 획득하여 이를 기반으로 해당 산업의 미래를 예측하고 경쟁적 우위를 차지하지 못한다면 기업이 자연도태하게 되는 현실이다. 최근의 기업 환경은 사업 영역의 확대 뿐 아니라 국제적 시장 경제 체제로의 변화 등에 적응해야 하며 무한 경쟁 체계의 사업 환경에서 고객들의 다양한 요구와 전문화된 요구에 적응할 수 있어야 한다.

이러한 현실에서 기업들이 전략적 정보를 획득하기 위해서는 기업들의 경영 관리를 위해 필요로 하는 의사결정 정보를 신속하고 정확하게 제공하는 정보시스템의 구축이 필요조건이다. 의사결정 정보를 제공하는 정보시스템이란 단순히 실적 통계를 제공하는 정보시스템이 아니라, 환경의 변화에 적응

하고 경쟁 기업들과의 차별화된 경쟁 전략을 수립할 수 있도록 함으로써 산업 내에서 지속적인 발전을 할 수 있도록 하는 정보시스템을 의미한다. 따라서 이러한 정보시스템은 다음과 같은 정보들을 파악할 수 있도록 정보를 제공해야 한다.

- ♦ 산업의 발전 형태 및 방향을 파악할 수 있는 미래 지향적 정보의 제공
- ♦ 경쟁 유발 요인을 파악하여 경쟁에 대응할 수 있는 최선의 방법을 찾을 수 있는 정보의 제공
- ♦ 유리한 경쟁 위치에 설 수 있는 요인을 파악할 수 있는 정보의 제공

[그림 1-1] “기업 환경의 변화”는 단순히 소품종 다량 생산의 시장 환경에서 제품을 대량으로 생산하여 대량 소비에 의한 이익 창출에 기여하던 정보시스템으로부터 다품종 소량 생산의 시장 환경 변화 및 고객들의 다양한 요구변화를 반영하여 변화해야하는 새로운 정보시스템의 방향을 제시한다.



[그림 1] 기업 환경의 변화

이는 기존의 정보시스템들에서 제공되던 정보들이 단순히 생산을 관리하거나 판매를 관리하는 실적 관리 위주의 정보시스템이었다면, 새로운 정보시스템은 고객의 요구사항을 분석하고, 시장 동향을 분석할 수 있는 정보를 제공할 수 있어야 한다는 것을 의미한다.

따라서 새로 구축되는 정보시스템에는 다음과 같은 경영 환경의 변화를 반영한 정보들을 제공할 수 있도록 구축되어야 한다는 것을 알 수 있다.

- ♦ 고객 및 시장 환경의 급격한 변화에 대한 분석 및 예측
- ♦ 고객 성향 및 요구를 반영한 새로운 마케팅
- ♦ 제품의 품질 및 높은 품질의 서비스 요구
- ♦ 신규 시장 개척 및 신제품 개발을 통한 새로운 비즈니스 창출

기존의 트랜잭션 데이터 처리 위주의 정보시스템들은 이러한 정보들을 최고 경영자 또는 관리자들에게

게 제공하기에는 미흡한 점이 많은 정보시스템들이다. 다시 말해 최고 경영자나 관리자들이 새로운 효과적인 의사결정을 할 수 있도록 정보를 제공하기에는 한계를 가지고 있으며, 제공되는 정보의 품질도 매우 낮은 정보시스템이다. 이러한 문제점은 먼저 기존의 정보시스템들이 기업의 전체 업무를 대상으로 구축되어 통합적으로 운영되는 시스템이 아니라, 담당 업무별로 개별적인 필요성에 의해 단위 업무 위주의 정보시스템으로 구축되어 정보가 고립(Isolate)된 상태로 유지되기 때문이다. 그리고 기존의 정보시스템들은 고립된 정보시스템들에 존재하는 데이터들의 신뢰성과 일관성에 관련된 많은 문제를 내포하고 있다. 예를 들어 영업관리 시스템에 있는 판매수량과 생산관리 시스템의 출고수량이 차이를 보일 수 있고, 수량은 맞는다고 하더라도 해당 제품을 구매한 고객의 정보가 각각의 시스템에 서로 다르게 존재할 수도 있다. 이러한 문제는 다수의 중복된 데이터들이 서로 다른 단위 시스템들에 존재하기 때문에 발생할 수 있으며, 일관성을 가지고 단위 시스템들의 데이터를 처리하지 못하기 때문에 발생할 수도 있다. 그리고 데이터를 수집하는 곳과 사용되는 곳이 분리됨으로써 발생하기도 하며, 업무를 처리하는 담당 부서들 사이의 이해관계 및 문화적인 차이에서도 발생할 수 있다. 그러므로 이러한 정보시스템들에서 제공하는 정보를 이용하여 최고 경영자 또는 관리자들이 올바른 의사결정을 할 수 없다는 것을 의미하는 것으로서, 기존에 관리하는 데이터들을 정제하여 사용해야 한다는 것을 의미한다. 그리고 기존에 존재하는 다양한 단위 업무 시스템들로부터 필요한 의사 결정 지원 정보를 취합해서 제공하기 위해서는 많은 시스템 자원 뿐 아니라 시간 및 비용을 필요로 한다.

따라서 기업 또는 조직들은 경쟁력을 확보하기 위해서 기존의 단위 시스템들에 축적되어있는 방대한 과거의 데이터들을 활용하여 효과적인 분석 기능을 가지며, 신속하고 정확한 의사 결정을 지원할 수 있는 새로운 정보시스템인 데이터 웨어하우스의 구축을 필요로 한다.

1.3 데이터 웨어하우스의 특징

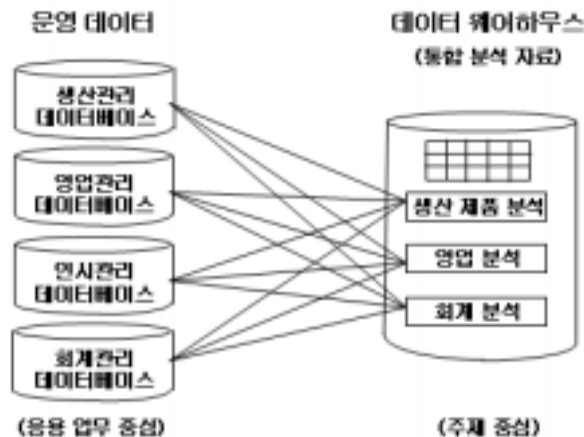
데이터 웨어하우스는 기존의 운용 데이터베이스에 비교하여 의사 결정을 지원할 수 있는 분석 정보를 제공한다는 것이 가장 큰 차이점이며 특징이라고 할 수 있다. 이와 관련하여 데이터 웨어하우스의 기본적인 사상을 정립한 “W. H. Inmon”의 저서에 의하면 데이터 웨어하우스는 다음과 같은 4가지 특징들을 지닌 데이터의 집합체를 의미한다고 그의 저서 “Building the Data Warehouse (2nd Edition)”에 서술했다.

- ◆ 주제 지향성 (Subject Oriented)

주제 지향성이란 데이터 웨어하우스 내에 존재하는 데이터들은 조직에서 일반적인 트랜잭션 처리를 위한 다양한 종류의 프로세스 중심의 데이터와는 달리 기업의 의사 결정에 필요한 특정 주제 (Subject) 즉, 기업의 주요 업무 기능과 관련된 주제 영역별로 데이터들을 구성하는 특성을 의미한다.

예를 들어, 보험회사의 경우 프로세스 중심의 시스템으로는 “자동차 보험”, “생명보험”, “개인연금보험”, “교육보험” 등이 해당되지만, 이들의 주제 영역을 보면 분석이 되는 “고객”, “약

관”, “청구”, “보험상품” 등이 될 수 있다. 금융 기관의 단위 시스템들은 “예금”, “대출”, “신용카드”, “신탁” 등과 같은 단위 업무 기능들을 처리하는 데이터베이스를 중심으로 운영되며, 데이터 웨어하우스는 금융기관과 거래를 하는 “고객”, “거래처”, 또는 금융 기관의 “상품”, “영업 활동” 등과 같은 주요 주제 영역을 중심으로 조직화 된다.



[그림 2] 데이터 웨어하우스의 특징 - 주제지향성

[그림 2] “데이터 웨어하우스의 특징 - 주제지향성”의 경우는 현재 운영되는 단위 시스템들로 “생산관리”, “영업관리”, “인사관리”, “회계관리” 시스템들로부터 주제 영역이 “생산 제품”, “영업”, “회계”인 데이터 웨어하우스를 데이터를 취합하는 내용을 도식한 것이다.

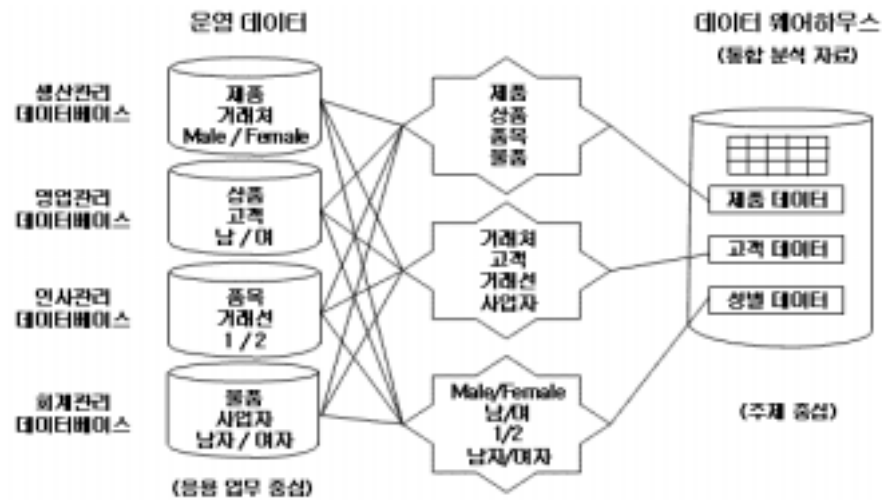
◆ 통합성 (Integrated)

통합성은 데이터베이스에 다양한 형태 또는 종류로 존재하는 원시 데이터들로부터 일관된 명명 규칙이나, 저장 규칙 등을 도출하여 데이터 웨어하우스에 하나의 유형의 물리적 데이터로 저장해야 한다는 특성을 의미한다.

기존의 어플리케이션 중심의 환경에서는 데이터베이스 내에 관리되는 남자와 여자를 구분하는 데이터 도메인(Domain)으로, “생산관리” 데이터베이스에서는 “Male/Female”, “영업관리” 데이터베이스에서는 “남/여”, “인사관리” 데이터베이스에서는 “1/2”, 그리고 “회계관리” 데이터베이스에서는 “남자/여자” 등으로 다양하게 적용할 수 있으나, 데이터 웨어하우스에서는 이들을 통합하여 하나의 성별로 관리하도록 하는 특성을 말한다. 즉, 각각의 단위 시스템들에서 사용되는 남자와 여자를 구분하는 내용에 대하여 '남'과 '여'로 통합하여 관리하도록 하는 특성이다.

또, 다른 예제로 기존의 데이터베이스에서는 업무 영역별로 센티 미터, 미터, 킬로 미터와 같은 미

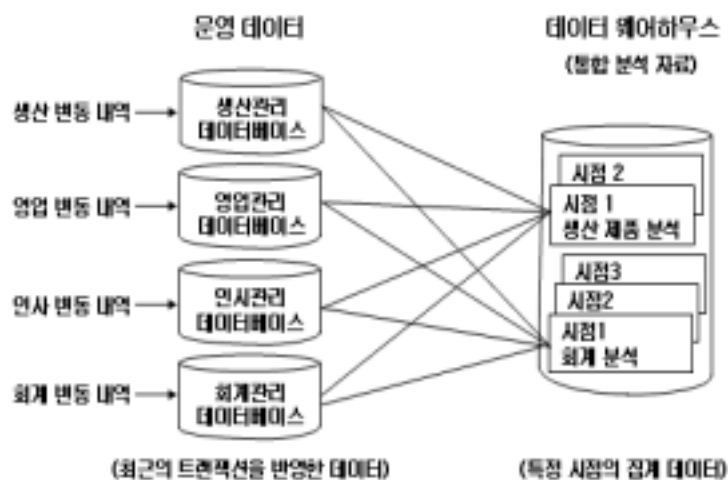
터법을 사용하는 데이터와 인치, 야드, 피트와 같은 서로 다른 여러 가지 측정 단위를 사용할 수 있지만 데이터 웨어하우스에서는 측정 단위를 통합하여 일관되게 데이터를 유지할 수 있도록 해야 한다.



[그림 3] 데이터 웨어하우스의 특징 - 통합성

♦ 시-계열성 (Time-Variant)

시-계열성이란 데이터베이스에 존재하는 데이터들은 데이터를 액세스하는 특정 시점에 정확성을 유지해야하는 반면에 데이터 웨어하우스의 데이터들은 일정 시간(기간) 동안의 시간 변화에 따른 데이터들의 변화 정보를 저장 관리해야 한다는 특징을 말한다. 즉, 데이터 웨어하우스에 저장되는 데이터들은 수시로 갱신이나, 변경하기 보다는 시간에 따라 변화된 데이터를 저장해야한다는 것을 의미한다.



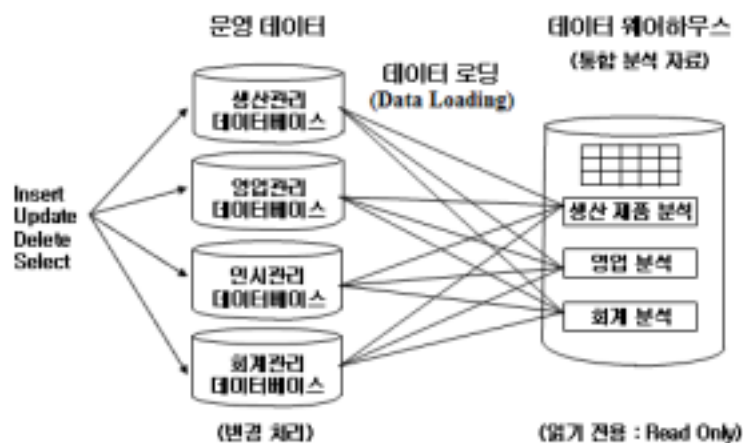
[그림 4] 데이터 웨어하우스의 특징 - 시-계열성

예를 들어, “지난 20 년간의 예산 편성금액 증가율 대비 경제성장 증가율”, “작년 같은 해당 월의 생산량 대비 판매량 실적”, “분기별 판매 계획 대비 현재의 판매 실적” 등과 같이 시간 변화에 따른 추이 등의 정보를 관리한다.

데이터 웨어하우스에 존재하는 대부분의 데이터들은 정형 또는 비정형적으로 “날짜”, “주”, “월”, “분기”, “년도”와 같은 시간 요소를 포함한다.

♦ 비-휘발성 (Non-Volatile)

비-휘발성은 데이터베이스에 존재시키는 데이터들은 트랜잭션 관리, 복구 기법, 동시성 제어 기법 등을 고려하여 수시로 입력, 수정, 삭제 등이 발생하는 “휘발성”의 성격을 가진 반면에 데이터 웨어하우스의 데이터들은 기존의 운영 데이터베이스로부터 데이터들을 데이터 웨어하우스로 데이터 로딩>Loading)되면 특별한 경우를 제외하고는 수정, 삭제 등이 발생하지 않고 읽기-전용(Read-Only) 데이터로 존재한다는 특성을 의미한다.



[그림 5] 데이터 웨어하우스의 특징 - 비-휘발성

예를 들어, 지난 1 개월간 고객들이 이용한 거래 유형별 사용실적 집계와 같이 데이터베이스의 원시 데이터가 변경되는 경우를 제외하고는 데이터 웨어하우스에 한번 생성된 데이터는 변경 또는 수정이 발생하지 않도록 관리해야 한다.

데이터 웨어하우스에 존재하는 데이터들은 일반적인 처리과정에서 갱신은 이루어지지 않으며, 주로 초기 데이터 탑재와 데이터 액세스만을 수행한다.

1.4 데이터베이스와 데이터 웨어하우스의 비교

흔히 데이터베이스는 OLTP(On-Line Transaction Processing) 데이터를 저장하는 자료저장소이고, 데이터 웨어하우스는 OLAP(On-Line Analytical Processing) 데이터를 저장하는 자료저장소라고 비교를 한다. 이는 저장되는 데이터의 성격에 따라 비교하는 방법으로써, 저장되는 데이터의 성격을 이해한

다면 쉽게 이해할 수 있다.

♦ OLTP (On-Line Transaction Processing) : 온-라인 거래 처리

OLTP 데이터는 주로 비즈니스 업무를 처리하는 과정에서 발생하는 데이터들로서 네트워크상의 여러 이용자들이 실시간으로 데이터를 갱신하거나 조회하는 등의 단위 작업을 수행하며 발생한다.

OLTP는 데이터를 발생시키는 이벤트들에 대한 내용을 데이터로 처리하는 것으로 제한된 숫자의 요약된 보고서들을 생성한다.

- 예) 은행의 창구 업무
 항공사의 예약 업무
 신용카드 조회 업무
 발주 품목의 입고처리 업무
 영업관리의 판매처리 업무

♦ OLAP (On-Line Analytical Processing) : 온-라인 분석 처리

OLAP 데이터는 최종 사용자들이 다양한 분석을 하고 의사결정을 하기위하여 요구하는 다차원 정보들을 활용할 수 있게 하는 데이터들을 OLTP 시스템과 기업 외부의 데이터들을 처리하며 발생한다.

사용자들이 요구하는 분석적 질의들을 처리하기 위하여 데이터베이스에 존재하는 수많은 데이터 레코드와 테이블 데이터들을 집계 또는 요약하여 저장한다.

분석가들에게 모든 유용한 관점에서의 데이터들에 대한 분석 뷰를 가능하게 하는 데이터를 처리한다.

- 예) 시장 동향과 고객 선호도 변화 분석 업무
 경제 성장률과 급여 상승 변화 분석 업무
 산업의 성장률과 제품의 변화 분석 업무

[표 1] OLTP 시스템과 OLAP 시스템의 비교

특 징	OLTP	OLAP
주요용도	오퍼레이셔널 트랜잭션	정보 분석
사용자	은행 창구원 등의 일반 사용자, DBA등	조직 관리자, 분석가 등의 지식 근로자
데이터베이스 설계 기법	ER기반, 응용 프로그램 중심	스타/스노우 플레이크 기반, 주제 중심
데이터	현재 값 중심	시간에 따른 변화 중심
데이터 요약도	요약 대신 개개 데이터 값 중심	데이터의 요약 중심
직업 단위	짧고 간단한 트랜잭션 중심	대다수가 복잡한 질의
액세스 유형	읽고/쓰기 모두 필요	주로 읽기

액세스 레코드 수	수십 개 정도로 많지 않음	수백만 개 정도로 상대적으로 매우 많음
사용자 수	수천 명 단위로 상대적으로 많음	수백 명 단위로 상대적으로 적음
데이터베이스 크기	100MB에서 수 GB 정도 단위	100 GB에서 수 TB 정도 단위
시스템 우선순위	높은 성능과 높은 유용성 우선	높은 유연성과 사용자 자율성 우선
시스템 성능 평가 척도	트랜잭션 처리량	질의 처리량과 응답시간

1.5 OLAP의 종류

데이터 웨어하우스에서 거론되는 OLAP은 [그림 6] “OLAP의 분류”와 같이 도식할 수 있으며, 각각의 특성은 다음과 같이 정리 할 수 있다.



[그림 6] OLAP의 분류

(Data Warehouse\최용락_데이터아키텍처(DA) p.496)

1) MOLAP (Multidimensional OLAP)

다차원 OLAP는 사용자 질의에 빠른 응답을 위하여 다차원으로 구성되는 데이터베이스에 직접 색인 (Index)을 반영한 다차원 배열을 사용한다. 처리 가능한 모든 조합의 데이터들이 다차원 배열에 이미 반영되어 원하는 셀(Cell)의 정보를 제공한다. 사용자 질의에 대해 신속하게 응답하는 장점이 있으나, 요구되어질 수 있는 다양한 경우에 대해 다차원 테이블을 구성해야하는 것이 문제점이다.

2) ROLAP (Relational OLAP)

ROLAP 엔진은 RDBMS와 사용자 사이에서 복잡한 SQL을 대신 생성하고, 다차원 연산을 수행가능하게 한다. 관계형 데이터베이스 내에 저장되어 있는 데이터를 요구하여 분석을 위한 정보를 제공한다. 점은 SQL의 단점인 비교 능력의 결여와 순차적 연산을 지원하지 못하는 점이다. 주로 스타 스키마

형태의 구조를 지원하며, Cartesian product, Parallelism, Bitmap index 기법을 사용한다.

3) HOLAP (Hybrid OLAP)

신속한 응답의 장점(MOLAP의 장점)과 미가공 데이터(Raw Data) 검색(ROLAP의 장점)의 장점 모두를 수용한다. 차원 데이터베이스나 관계형 데이터베이스 시스템에 저장되어 있는 데이터들을 동시에 분석할 수 있는 기능을 제공할 수 있는 도구를 의미한다. 다차원 데이터의 저장공간으로 MDB, RDB를 사용한다. (요약된 데이터나 계산된 데이터는 MDB에 저장하고 상세 데이터는 RDB에 저장한다.

4) DOLAP (Desktop OLAP 또는 Database OLAP)

다차원 데이터 분석을 위한 데이터의 저장 및 데이터 프로세싱이 모두 클라이언트에서 발생, 처리 가능하므로 적은 비용으로 구축이 가능하지만, 대용량 데이터의 처리에는 한계가 있다. 대용량 분석 데이터 처리에 한계를 가지며, 사용자별로 클라이언트에 저장된 분석 데이터의 일관성에도 문제가 제기될 수 있다.

[표 2] OLAP 비교표

항목	MOLAP	ROLAP	HOLAP	DOLAP
특징	다차원 데이터 저장과 프로세싱에 동일한 엔진인 다차원 DBMS가 사용되기 때문에 다른 아키텍처에 비해 네트워크 상의 데이터 이동이 최소화될 수 있다.	RDBMS에서 수행 하며 ROLAP 엔진이 사용자의 복잡한 SQL을 대신 생성하고 다차원 연산을 수행한다. 메타 데이터를 관리 및 저장 한다	데이터 저장공간으로 다차원 DB나 관계 DB나 모두 사용하는 구조이다. DB: 요약 데이터, 관계식생성데이터 관계 DB: 상세 데이터 저장	다차원 데이터의 저장 및 프로세싱이 모두 클라이언트에서 이루어지며, 사용자는 클라이언트에 저장된 데이터를 대상으로 제한된 다차원분석을 한다.
장점	사용자 질의에 대해 빠른 응답 성능을 제공한다.	원시데이터의 취급이 가능하며 대용량 데이터를 처리할 수 있다.	MOLAP, ROLAP의 장점을 모두 제공한다.	적은 비용으로 구축할 수 있다
단점	차원을 구성하는 항목들에 변화가 생길 경우 DB나 전체적으로 완전히 재구축될 수 있다.	SQL 비교능력의 결여되고, 순차적 연산 지원이 곤란하다	HOLAP: 활용하기 위한 구축과 관리가 어렵다	대용량 데이터 처리에 한계가 있으며 각 사용자는 별도의 데이터를 가지므로, 일관성 유지 곤란하다
제품	Cognos's PowerPlay Hyperion's Essbase Oracle's Express Pilot's Decision Support Suite Applix's TM/1	Group1's Sagent OLAP Informix's MetaCube Information Advantage's DecisionSuite MicroStrategy's DSS Agent	Seagate Software's Holos Oracle's Express Microsoft SQL Server for OLAP	BrioTechnology's BrioQuery Business Objects's Business Objects

1.6 데이터 웨어하우스와 데이터 마트의 비교

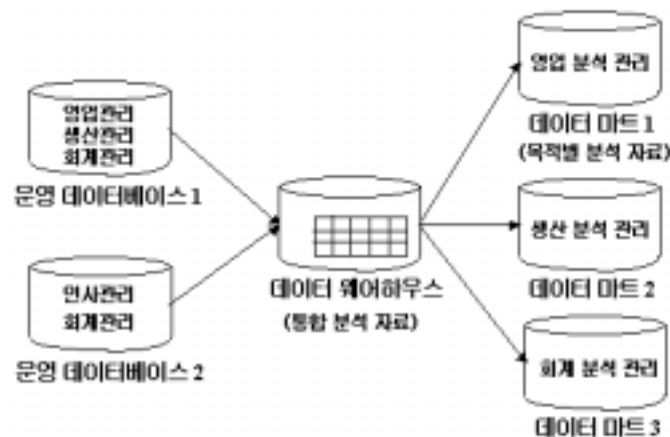
데이터 웨어하우스가 전사적인 데이터를 한곳에 모으는 것이라면, 데이터 마트(Data Mart)는 부서 단위, 또는 업무 단위로 구축한 소규모 데이터 웨어하우스에 해당한다. 데이터마트는 데이터 웨어하우스에 있는 데이터들 중에서 해당 부서 또는 특정 목적에 적합한 데이터들만을 따로 내려 받아 구축하는 것으로 실제 트랜잭션은 데이터 마트에서 일어나는 경우가 많다.

데이터 웨어하우스

- ◆ 방대한 양의 이력 데이터를 가지고 있는 자료 저장소
- ◆ 데이터 웨어하우스의 변경은 시간 순서를 가지고 입력
- ◆ 데이터 웨어하우스의 쿼리는 실시간 분석 처리(OLAP)의 특성을 가짐

데이터 마트(Data Mart)

- ◆ 데이터 웨어하우스와 유사하지만 제한된 업무 도메인을 가짐
- ◆ 데이터가 특정 유형, 특정 업무 기능, 또는 특정 업무 단위로 제한됨

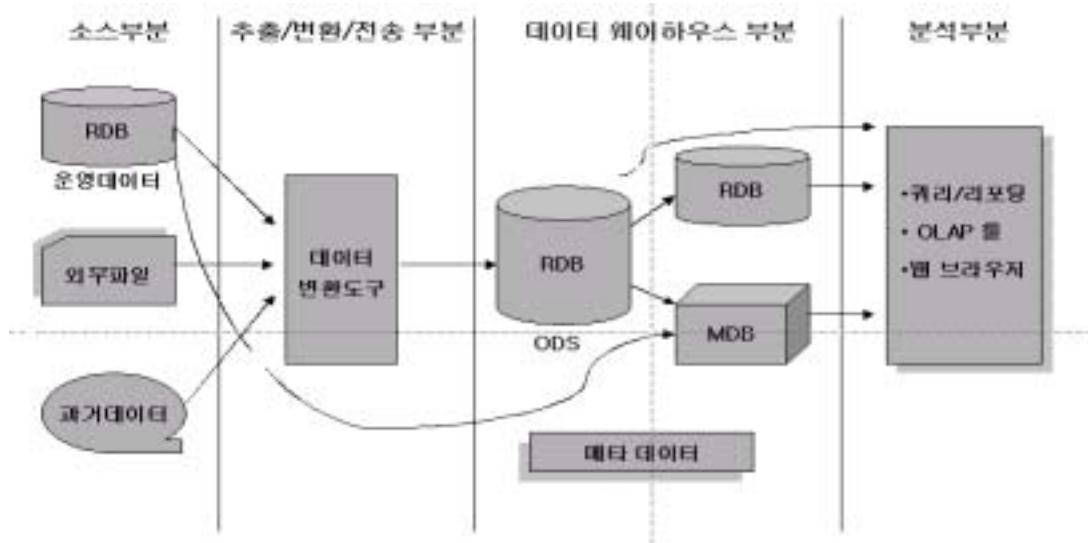


[그림 7] 데이터 웨어하우스 / 데이터 마트

1.7 데이터 웨어하우스의 구조

데이터베이스와 데이터웨어하우스의 연관 관계를 고려하여 데이터 웨어하우스의 위치를 간단하게 도식해보면 [그림 8] “데이터 웨어하우스의 구조”와 같이 표현할 수 있다.

데이터 웨어하우스는 실제 운영중인 데이터베이스의 기존 데이터들을 소스 데이터로 이용하여 다양한 데이터 도출 도구들을 이용하여 원하는 데이터 웨어하우스의 데이터들을 데이터 웨어하우스 데이터베이스에 저장한다. 이렇게 저장되거나, 관리되는 데이터들은 데이터 웨어하우스 접근 도구들을 이용하여 사용자에게 분석 정보를 제공하여 활용되게 된다.



[그림 8] 데이터 웨어하우스의 구조

“소스 부분”은 실제로 운영중인 소스 데이터를 포함하는 “운영 데이터베이스”부분을 의미하며, “운영 데이터”에는 트랜잭션 처리를 위한 데이터들은 물론, 모든 고유 데이터(Legacy Data)들을 포함한다. “추출/변환/전송 부분”은 소스 데이터로부터 데이터 웨어하우스 관리 소프트웨어인 데이터 변환 도구들을 활용하여 데이터 웨어하우스에 데이터를 적재(Load)하는 과정을 의미한다. “데이터 웨어하우스 부분”은 실제로 구축된 데이터 웨어하우스를 의미하며, 데이터 웨어하우스 관리를 위한 메타 데이터와, 구축되는 대표적인 데이터베이스인 관계형 데이터베이스와 다차원 데이터베이스가 존재한다는 것을 의미한다. “분석 부분”은 실제로 데이터 웨어하우스를 활용하는 단계를 의미하며, “마케팅”, “EIS”, “DSS”, “SIS” 등의 업무에 데이터웨어하우스 접근 도구들을 활용하여 접근하는 것을 표현한다.

앞에 제시한 데이터 웨어하우스의 구조를 기준으로 각각의 구성 부분을 간략히 설명하면 다음과 같이 정리할 수 있다.

1) 소스(Source) 부분

소스 부분은 기업 및 기관들의 업무 처리를 수행하며 발생하는 운영계 데이터뿐 아니라, 기업 및 기관들에 영향을 크게 미치는 외부의 데이터들도 포함한다.

소스 데이터는 플랫폼 파일(Flat File) 형태로 존재할 수도 있으며, R-DB(Relational Database), H-DB(Hierarchical Database), N-DB(Network Database) 등의 다양한 형태로 존재한다.

2) 추출(Extraction) / 변환(Transformation) / 전송(Transmission) 부분

소스에서 데이터를 추출하여 스테이징 서버(Staging Server) 혹은 타겟 서버(Target Server) 쪽으로 SAM(Sequential Access Method) 파일 형태로 보낼 수도 있고, 스테이징 데이터베이스에 데이터베이스의 형태로 보낼 수도 있다.

추출된 데이터는 타깃 서버 데이터 웨어하우스의 표준에 맞게 여러 가지 형태로 변환되며, 적절하게 변환된 데이터는 타깃 서버 데이터 웨어하우스에 전송되고, 적재(Loading)된다.

3) 데이터 웨어하우스 부분

데이터 웨어하우스의 데이터는 주로 R-DBMS(Relational Database Management System)나 M-DBMS(Multi Dimensional Database Management System)에 저장된다.

전사적 데이터 웨어하우스인 경우에는 거의 R-DBMS에 저장되며, 데이터 마트(Data Mart)인 경우에는 R-DBMS에 저장되기도 하며 M-DBMS에 저장되기도 한다.

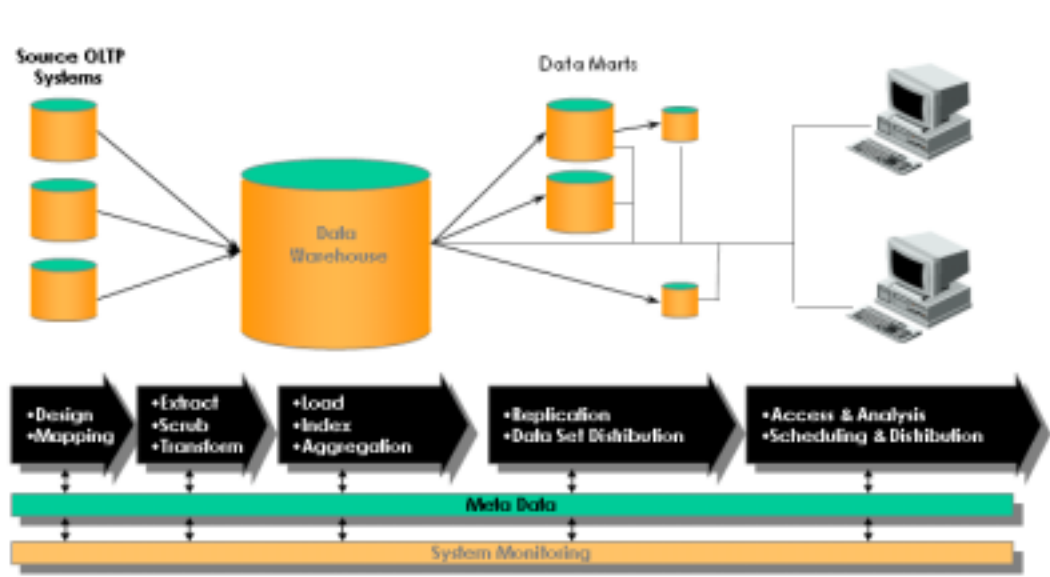
데이터가 R-DBMS에 저장되는 경우에는 빠른 속도를 보장하기 위해서 데이터마트와 별도로 요약 데이터를 따로 두기도 한다.

데이터 웨어하우스에 데이터가 들어가기 전에 중간 단계의 데이터 저장고인 ODS(Operational Data Store)를 별도로 두기도 하는데, 운영계 시스템에서 데이터 웨어하우스에서 필요하다고 생각되는 부분의 모든 데이터를 보관한다.

4) 분석 부분

데이터 웨어하우스에 저장된 데이터를 이용하여 사용자가 원하는 보고서를 만들기 위해서 리포팅 도구를 사용하며, 사용자가 원하는 보고서는 물론이고 드릴-업(Drill-Up), 드릴-다운(Drill-Down) 등을 통해 다차원 분석을 하기 위해 OLAP 도구를 사용한다.

사용자에게 의미 있으며, 유용한 데이터의 관련 패턴을 분석하기 위해서 데이터 마이닝(Data Mining) 도구를 사용한다.



[그림 9] 데이터 웨어하우스 구조와 단계

(Data Warehouse\최용락_데이터아키텍처(DA) p.492)

[그림 9] “데이터 웨어하우스 구조와 단계”는 [그림 8] “데이터 웨어하우스의 구조”에다 각 단계

에서 수행해야 하는 작업을 표현한 그림이다.

데이터 웨어하우스의 데이터 형태

- ◆ **메타 데이터 (Meta Data)**

데이터 웨어하우스에 어떤 데이터를 어떻게 저장할 것인가를 세부적으로 기술하는 '설명자'의 역할을 하며, 운영 시스템의 데이터 구조를 데이터 웨어하우스로 매핑(Mapping)하는데 필요한 데이터를 요약 (사용되는 알고리즘 포함)

- ◆ **현재의 상세 데이터 (Current Detail Data)**

대개 디스크에 저장

가장 최근의 데이터 변경 사항을 반영

가장 하위 레벨에 위치하므로 대부분 데이터 량이 많음

- ◆ **과거의 상세 데이터 (Past Detail Data)**

가끔씩 필요에 의해 액세스되는 데이터로서 테이프 등과 같은 보조 저장장치에 수록

현재의 상세 데이터도 정의되는 시간 개념에 따라 과거의 상세 데이터로 바뀌게 됨

- ◆ **약간 요약된 데이터 (Low Summary Data)**

주로 디스크에 저장

현재의 상세 데이터를 약간 요약한 것

- ◆ **고도로 요약된 데이터 (High Summary Data)**

주로 디스크에 저장

간결하게 요약되어 쉽게 액세스가 가능함

약간 요약된 데이터로부터 재차 요약된 데이터

1.8 데이터 웨어하우스 구축 방법

데이터 웨어하우스를 구축 방식에는 전사적으로 관리되는 통합적 데이터 웨어하우스를 먼저 구축하고, 필요에 따라서 데이터 마트를 구축하는 방식이 있다. 그리고 현재 운영되고 있는 단위 업무별로 데이터 마트를 먼저 구축하고, 필요에 따라 구축된 데이터 마트를 활용하여 통합적 데이터 웨어하우스를 구축하는 방식이 있다. 데이터 마트를 확장하는 방식은 실제로 데이터 마트가 다차원 데이터 베이스를 근간으로 하기 때문에 데이터 웨어하우스의 근간이 되는 관계형 데이터베이스와의 호환성이 없으므로 불가능하다. 따라서 데이터 웨어하우스를 구축하기 위해 데이터 마트를 구축한다면, 반드시 관계형 데이터베이스를 사용하여야 한다. 그리고 관계형 데이터베이스 설계는 전사적 데이터 웨어하우스로의 확장성을 고려하여 데이터베이스 설계를 해야 한다. 만일 이것이 보장되지 않은 상태에서

데이터 마트를 구축한다면, 운영 시스템으로부터 데이터 추출의 일관성 결여 및 데이터 무결성(integrity) 등을 보장할 수 없으므로, 데이터 웨어하우스를 구축하고자 하는 시점에는 많은 어려움이 발생한다. 또한, 데이터 마트로 확장하거나, 재구축할 때에는 관계형 데이터베이스의 설계를 바뀌어야만 하는 문제에 직면하게 된다.

이렇게 단계적으로 데이터 웨어하우스를 구축해 가는 장점으로는,

- 적은 비용으로 안전하게 소규모 프로젝트로 추진 가능
- 전사적 데이터 웨어하우스를 한꺼번에 구축하기는 어렵지만, 데이터 웨어하우스를 단계별로 구축하면 전사적 데이터 웨어하우스 구축으로 인한 위험 부담의 최소화
- 필요할 경우 신속하게 데이터 마트 구축 가능
- 데이터 마트 구축 후에도 데이터 마트를 확장하거나, 재구축을 필요로 할 때 언제든지 쉽게 전환 가능 (데이터는 모두 데이터 웨어하우스에 존재 하므로)

(Data Warehouse\dw개념_ORACLE.ppt p.26-27)

데이터 웨어하우스 구축과정

데이터 웨어하우스 구축과정은 다음과 같이 4 단계로 고려해볼 수 있다.

1단계 : 사용자 요구조건 정의(Business Requirement)

2단계 : 통합 환경의 구축

3단계 : Pilot Project 실시

4단계 : 데이터 모델링 및 구축

1단계 : 요구 조건 정의

사용자 요구조건의 정의 및 기업 특유의 데이터 웨어하우스 성격 규정 첫 단계는 사용자 요구조건을 파악하는 과정이다. 현재 산업별로 15개까지의 다양한 데이터 모델이 나와 있을 만큼 산업별 또는 기능별로 사용자 요구조건이 상이하다고 볼 수 있는데, 기업 고유의 또는 산업 특유의 사용자 요구조건을 파악하는 일이 최우선 과제라고 볼 수 있다.

사용자 요구조건이 파악되면 다음 단계는 기업 고유의 데이터 웨어하우스 성격을 사전에 규정하는 과제가 남아 있다. 예를 들어 주제 영역(Subject Area)은 어느 정도까지 설정할 것인지 여부, 데이터의 통합은 어떻게 할 것인지 여부, 또한 어느 정도의 기간에 해당하는 데이터를 가져올 것인지 여부 등을 결정하는 일이다.

2단계 : 통합 환경의 구축

데이터 웨어하우스 성격이 규정되면, 본격적으로 통합 환경을 구축하여야 한다. 하드웨어뿐만 아니라 관계형 데이터베이스 관리 시스템, 데이터 웨어하우스 관리도구, OLAP 툴 등이 이에 해당한다. 통합 환경의 구성요소 및 선택기준은 아래의 표와 같다.

[표 3] 통합 환경의 구성요소 및 선택기준

구 분	선 택 기 준
DBMS	병렬처리기술 제공 여부, 데이터 파티셔닝 기술정도, 장애대비책, 온라인 백업 및 복구기능 등
DW 관리도구	Meta Data 생성 여부, 지원언어, 사용 환경
OLAP 도구	Query 성능, 가격 대비 성능

3단계 : Pilot Project 실시

데이터 웨어하우스는 본질적으로 데이터베이스 규모가 크고 환경이 복잡하다. 따라서 본격적인 데이터 웨어하우스 구축에 앞서 Prototype으로 Pilot Project를 실시해 보는 것이 실패의 위험을 미연에 방지하고 경험을 축적할 수 있다. 대부분의 데이터 웨어하우스 공급사는 Pilot Project 지원을 위하여 자사의 고객센터에 데이터 웨어하우스를 구현하고자하는 고객들을 위하여 관련 시설 및 컨설팅 지원을 할뿐만 아니라, 고객 확보를 위해 수주가 확정된 경우 무료로 Pilot Project를 실시하기도 한다.

4단계 : 데이터 모델링 및 구축

Pilot Project를 전후하여 중요한 과정이 데이터 모델링 과정이다. 실제 프로젝트의 60% 이상의 시간이 소요될 정도로 데이터 모델링은 중요하며, 가장 중요한 고려사항으로 스키마 디자인(정규 또는 비정규), 미래의 성장 가능성에 미리 대비(데이터베이스 규모, 트랜잭션 건수), 응답 시간에 대한 조건이 완화되어 있는 만큼 어느 정도의 성능을 바라는지 여부, 사용자수 등을 고려해야 한다. 데이터 웨어하우스 구축의 마지막 단계는 실제 데이터 웨어하우스의 구축이다. 이는 다시 Information Service(정보 제공) 기능, 그리고 데이터 웨어하우스 Operation Service(운영 제공) 기능의 둘로 나눌 수 있다. 3000GB에 달하는 대용량의 데이터 웨어하우스를 구축한 MCI의 경우를 예로 들면 아래와 같다. 즉, Operation System상에 존재하는 데이터를 통합 변형한 뒤(Harvesting 과정), 데이터 웨어하우스를 구축한 다음 이를 필요에 따라, 데이터 마트로 보낸다.(Shipping 과정) 이 과정에서 Operational service는 필수적으로 수반되는데, 데이터 갱신 간격을 설정하는 과정, 데이터 웨어하우스에 사용자별로 접근 권한이나 접근 시간을 제한시키는 기능, 보안 유지 기능, 예외 관리 기능 등이 Operational service의 핵심 과제이다.

(Data Warehouse\데이터웨어하우스\논문_임진홍 p.33-35)

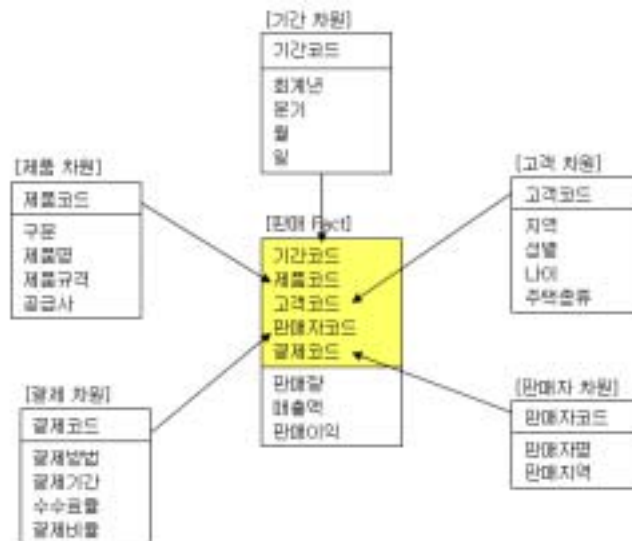
2. 데이터 웨어하우스 구축

2.1 다차원 모델링(Multidimensional Modeling) 방법

최근에 많이 활용되는 ROLAP은 MOLAP의 단점인 원시 데이터의 취급과 대용량 데이터를 처리할 수 없는 확장성의 한계를 극복하기 위해서 개발 되었다. ROLAP은 관계형 데이터베이스를 그대로 사용하며, 다차원적인 데이터 모델링을 통해서 다차원적 분석이 가능하도록 하는 방식으로써 사실(Fact) 테이블과 차원(Dimension) 테이블로 구성이 된다. 다차원 모델링은 사용자의 다양한 요구사항들을 충족시키는 복잡한 질의를 신속하게 처리하는 데 목적이 있으며, 스키마 구성 방식으로는 스타 스키마(Star Schema) 방식과 스노우 플레이크(Snowflake Schema) 스키마 방식이 널리 활용되고 있다.

1) 스타 스키마(Star Schema)

스타 스키마는 각 차원 테이블과 사실 테이블 사이에 일대다 관계를 가진 단순한 관계형 모델이지만, 정규화 된 모델은 아니다. 차원 테이블들은 질의들을 처리하기 위해 최적화를 제공할 목적을 갖고 정규화하지 않았으며, 이것이 스타 스키마와 OLTP 시스템을 위한 관계형 스키마 사이의 기본적인 차이점이다. 스타 스키마는 단일 업무에 사용하면 보다 좋은 성능을 유지할 수 있다.



[그림 10] 판매 정보 Star Schema

[그림 10] “판매 정보 Star Schema”는 “영업관리” 업무에서 발생하는 판매 정보를 스타 스키마로 다차원 모델링 한 내용이다. 판매 정보인 판매 사실(Fact) 테이블과 “기간”, “제품”, “결제”, “고객”, “판매자”의 차원 테이블로 표현하고 있다.

스타 스키마와 관련된 장/단점은 다음과 같이 정리할 수 있다.

장점)

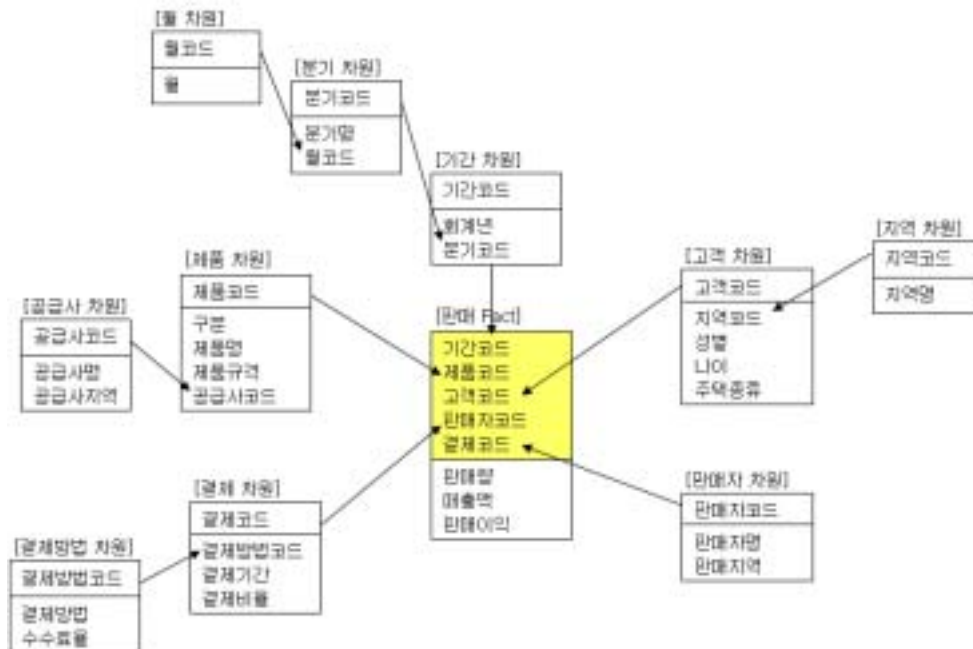
- 사용자들에 의해 직관적으로 이해하기 쉬움
- 조인(Join) 수가 적고 데이터베이스를 통하여 항해를 최적화
- SQL이 간단하면서 질의 처리에 아주 적합
- 일정한 성능을 유지하는 설계 가능

단점)

- 공간이 많이 필요하며 차원 테이블(Dimension Table)의 확장성이 없음
- Cross Dimensional Modeling이 안됨
- 차원(Dimension) 간의 n: n 경우는 안 됨(항상 1:n의 경우만 지원)
- 차원 구조가 복잡해져서 차원 요소가 많아지면 많은 중복을 감수해야 함

2) 스노우 플레이크 스키마(Snowflake Schema)

차원 요소마다 1개의 차원 테이블을 가지고 있고, 이 차원 테이블에는 차원 요소에 대한 정보와 상위 차원 테이블로 조인하기 위한 키를 가지고 있다. 각 차원 테이블은 정규화가 잘 되어 있어 중복이 최소화 되며, 따라서 테이블의 크기도 스노우 플레이크 스키마에 비교하여 크지 않다. 스노우 플레이크 스키마에서는 스타 조인(Star Join)이라고 하는 여러 개의 작은 테이블과 한 개의 큰 테이블을 조인하는 형태의 조인을 많이 수행해야 한다.



[그림 11] 판매 정보 Snowflake Schema

[그림 11] “판매 정보 Snowflake Schema”는 “영업관리” 업무의 판매 정보를 스노우 플레이크 스키마로 다차원 모델링 한 내용이다, 판매 정보인 사실 테이블과 “기간”, “제품”, “결제”, “고

객”, “판매자”의 차원 테이블로 표현하고 있다. 각 차원 테이블도 세부 차원 테이블을 가지고 있는 구조이다.

스노우 플레이크 스키마와 관련된 장/단점은 다음과 같이 정리할 수 있다.

장점)

- 한 차원(Dimension) 내에 여러 개의 계층(Hierarchy) 지원
- Cross Dimensional Modeling, n: n Dimension Relation 지원
- 차원 테이블(Dimension Table)이 확장 가능함
- 대용량 데이터 처리 지원하며 조직변화 등 변화를 잘 수용
- 저장공간은 다소 절약
- 정규화 된 구조들은 더 쉽게 갱신되고 유지된다.

단점)

- 스키마는 덜 직관적이고 최종사용자들은 복잡도에 의해 흥미를 잃음
- 내용을 통해 브라우저 하는 능력을 어렵게 함
- 추가적인 조인의 수가 많으며 SQL이 복잡하고 성능이 저하됨
- 차원 테이블이 많기 때문에 스타 조인을 많이 해야 함

2.2 시스템 구축 사례

시스템 구축 사례에서는 일반적인 기업의 “영업관리” 시스템을 대상으로 하여 데이터 웨어하우스를 구축한 사례를 설명한다. 데이터 웨어하우스 구축 대상은 “영업관리” 시스템이지만 데이터 웨어하우스를 구축하기 위해서 필요한 관련 업무 시스템들인 “생산관리”, “회계관리”, “인사관리” 시스템들에 대해서도 간략히 설명했다.

• 영업관리 및 관련 업무 시스템 소개

“영업관리”는 기업의 업무 중 핵심적인 업무이며, 정보화에 대한 가장 빠르게 효과를 얻을 수 있다. 정확하고 빠른 업무처리로 “영업관리” 업무의 효율성을 제고하는 것은 물론, 고객 만족을 극대화 할 수 있는 기업이 경쟁력을 갖추는 시스템을 보유하게 되는 것이다.

“영업관리”는 영업의 대상이 되는 “고객관리”, 고객이 필요로 하는 견적 정보를 관리하는 “견적관리”, 고객이 필요로 하는 “제품/상품”을 주문하는 정보를 관리하는 “주문관리”, 판매 및 수금을 관리하는 “판매관리”, 수금되지 않은 미수 금액을 관리하는 “미수금관리” 등으로 구성된다. 그리고 “영업관리”와 관련되어 매일 발생하는 데이터들은 “집계”, “현황”, “재고” 등의 내용을 각 기간별로 필요로 한다.

1) 영업관리 시스템의 기능 및 통계 정보

영업관리 시스템의 주요 업무 구분과 기능 및 통계정보

[표 4] 영업관리 시스템 기능 구성

기능	설명	통계정보
고객관리	고객의 정보에 대한 등록/수정/삭제관리를 하며, 고객 정보에 대한 조회 및 고객리스트, 고객카드 등의 출력 및 통계정보를 제공한다.	고객 유형별 현황, 고객별 매출현황
견적관리	제품의 견적에 대한 등록/수정/삭제관리를 하며 견적 정보에 대한 조회 및 견적서, 견적현황 등의 출력 및 통계정보를 제공한다.	거래처별 견적 현황
주문관리	주문 정보에 대한 등록/수정/삭제관리를 하며, 주문 정보에 대한 조회 및 주문전표, 주문대장 등의 출력 및 통계정보를 제공한다.	고객별 주문조회, 기간별 주문현황
판매관리	판매 정보에 대한 등록/수정/삭제관리를 하며, 판매 정보에 대한 조회 및 판매일지, 판매 현황 등의 출력 및 통계정보를 제공한다.	거래처별 판매조회, 기간별 판매현황
미수금관리	미수금 정보에 대한 등록/수정/삭제관리를 하며, 미수금 정보에 대한 조회 및 미수금현황 등의 출력 및 통계정보를 제공한다.	거래처별 미수금현황

2) 영업관리 관련 업무 시스템의 기능 및 통계 정보

관련 업무로는 “생산관리”, “회계관리”, “인사관리” 시스템이 있으며, 주요 기능 및 통계정보는 다음과 같다.

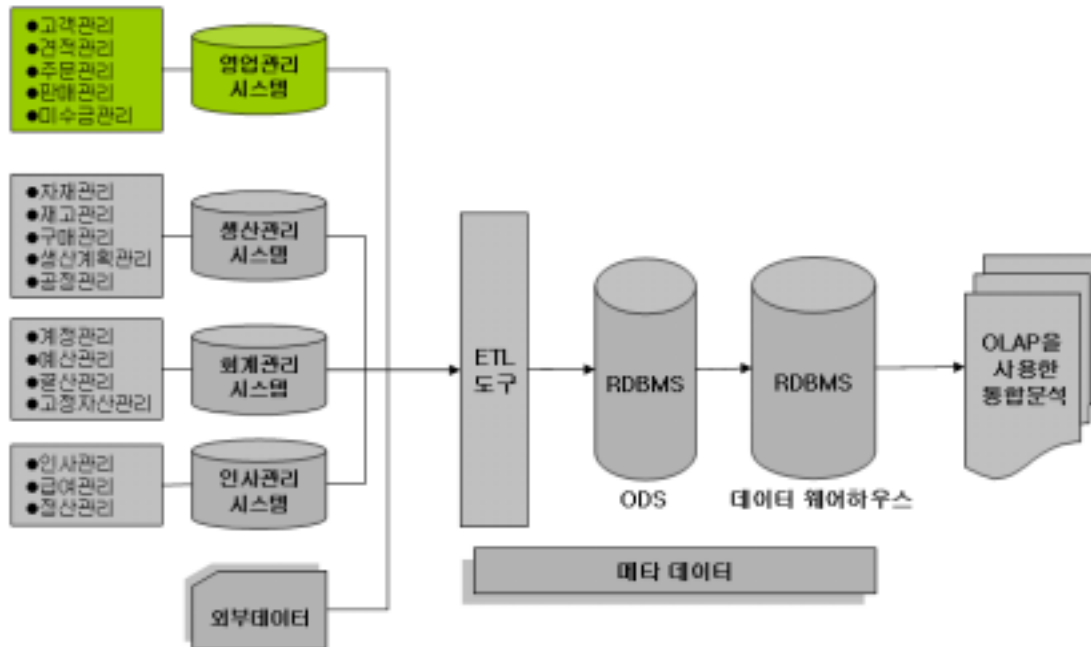
[표 5] 영업관리 관련 업무 시스템 기능 구성

기능영역	설명	통계정보
생산관리	제품을 생산할 때 필요한 각종 원재료, 부재료, 반제품 등에 대한 구매, 재고를 관리한다.	기간별 재고현황, 기간별 생산현황, 항목별 자재현황
회계관리	기업의 자금현황, 매출대비 실제 이익에 대한 현황, 미수채권에 대한 상황 등 기업의 실시간 회계 현황을 관리한다.	예산대비 결산현황, 기간별 수입지출현황, 기간별 예산증감현황, 기간별 결산현황
인사관리	발령사항, 상벌, 교육, 자격증 등 각종 사항을 관리하고, 근태관리와 퇴직자에 대한 관리 및 급여 지급과 세금관리 및 연말정산에 대한 관리한다.	직급별 인사현황, 월별 급여지급현황, 세목별 정산현황

• 영업관리 시스템 데이터 웨어하우스 구축

“영업관리” 시스템의 통계 정보는 시간에 따른 각 기능별 통계로서, 정보에 대한 시간의 흐름이나 각 대상에 대한 일반적인 현황만 조회가 가능하며 의사결정에 한계가 있다. “영업관리” 시스템의 OLTP 환경의 데이터와 연관된 타 업무 시스템의 데이터 및 외부 데이터를 통합하고, 성과 관리 기준

및 의사결정 지원을 위한 주제 지향적으로 전사적인 데이터 웨어하우스를 구축하여 설계에 따른 성능 비교와 분석을 한다.



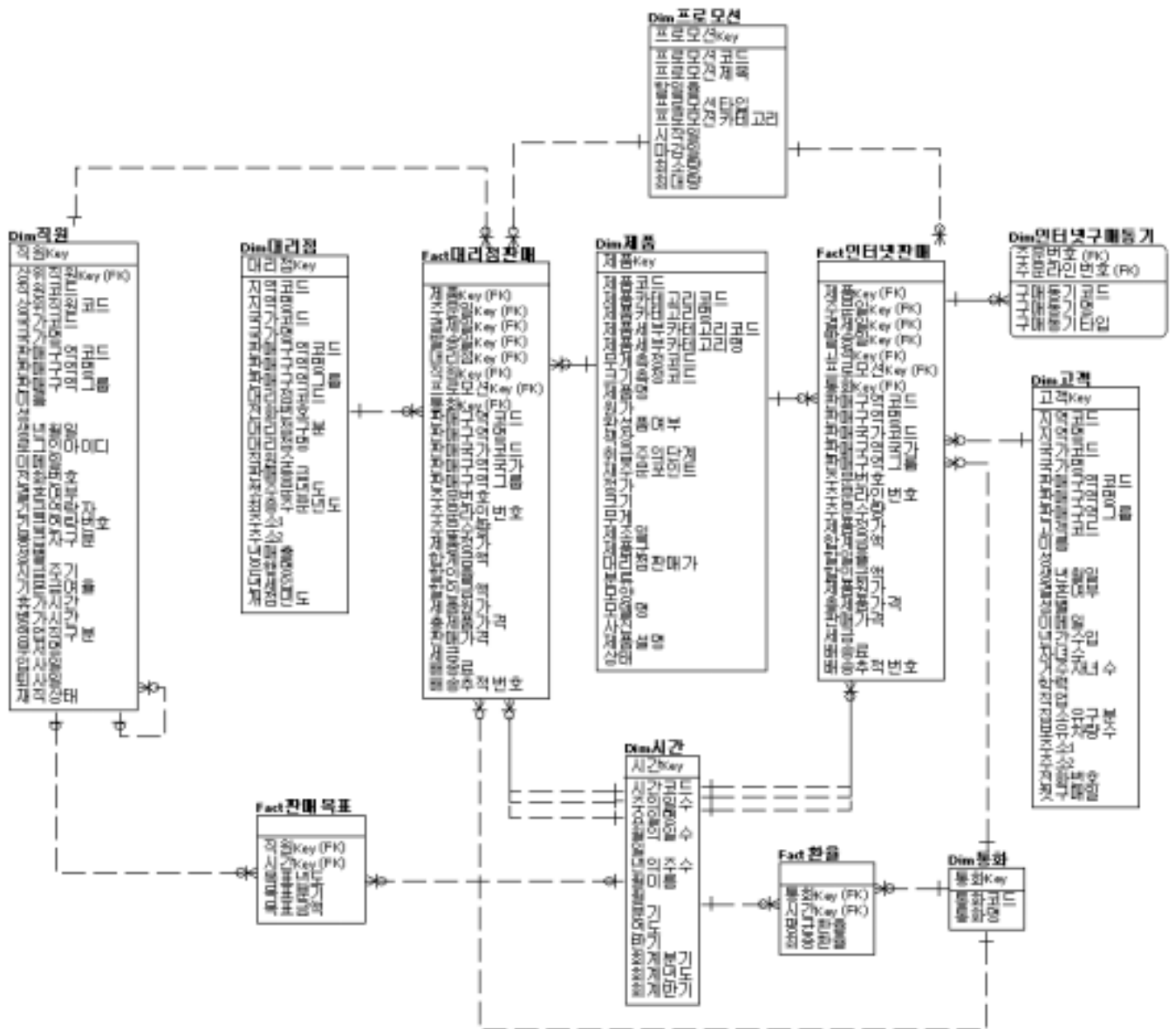
[그림 12] 영업관리 시스템 DW 구성도

[그림 12] “영업관리 시스템 DW 구성도”는 데이터 웨어하우스 구축의 주요 기능영역인 “영업관리”의 기능들과 관련 업무들의 기능 영역에 해당하는 기능들을 표기하였으며, 각 기능 영역의 시스템의 데이터들은 ETL(Extraction, Transformation, Loading)도구를 통하여 데이터가 추출, 전송되어 ODS에 저장된다. ODS에 저장된 데이터를 통합하고 정제하여 데이터 웨어하우스에 저장한다. 그리고 OLAP도구를 사용하여 통합적인 분석을 사용한다. ODS를 사용하는 구조의 가장 큰 장점으로는 미래에 전사적 데이터 웨어하우스(EDW: Enterprise Data Warehouse)로 발전하기 매우 쉽다.

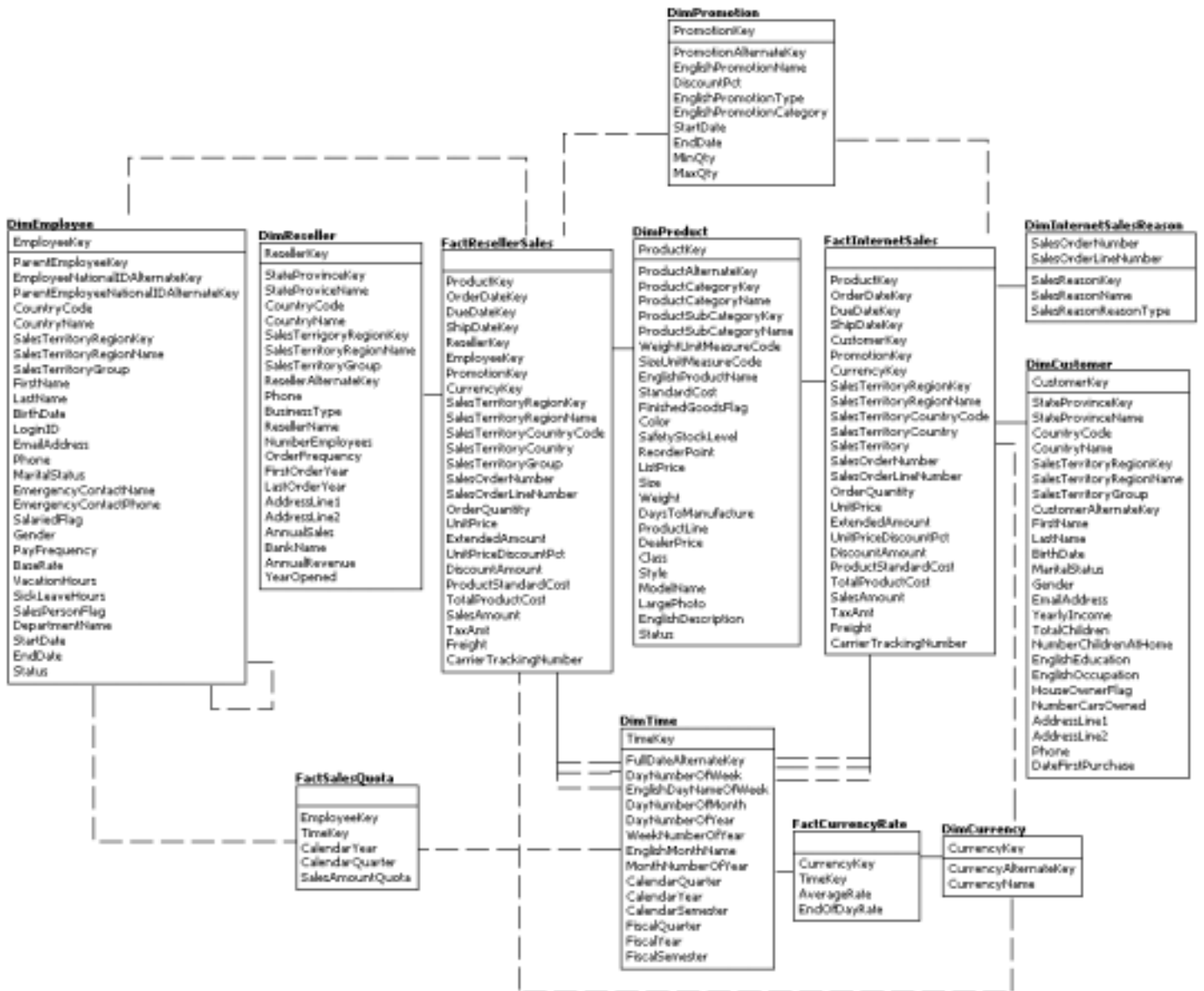
• 영업관리 시스템의 다차원 스키마 설계

“영업관리” 시스템의 다차원 모델링을 스타 스키마와 스노우 플레이크 스키마 두 가지 형태로 설계했다.

다차원 모델링을 수행할 때 고려해야 할 사항으로, 점진적으로 변화하는 차원의 데이터 중에서 그 시점의 값이 중요한 데이터는 사실(Fact) 차원에 같이 존재하게 한다. 제품 차원의 원가 및 정가는 시간이 지나면서 금액을 인상하거나 인하한다. 이렇게 변화하는 금액을 차원테이블에서 관리하면 금액이 인상되었을 때 이전 시점의 금액을 알 수가 없다. 그래서 사실 테이블에 원가와 정가 금액을 같이 저장하면 변화하는 데이터지만 그 시점의 금액을 관리할 수 있다.

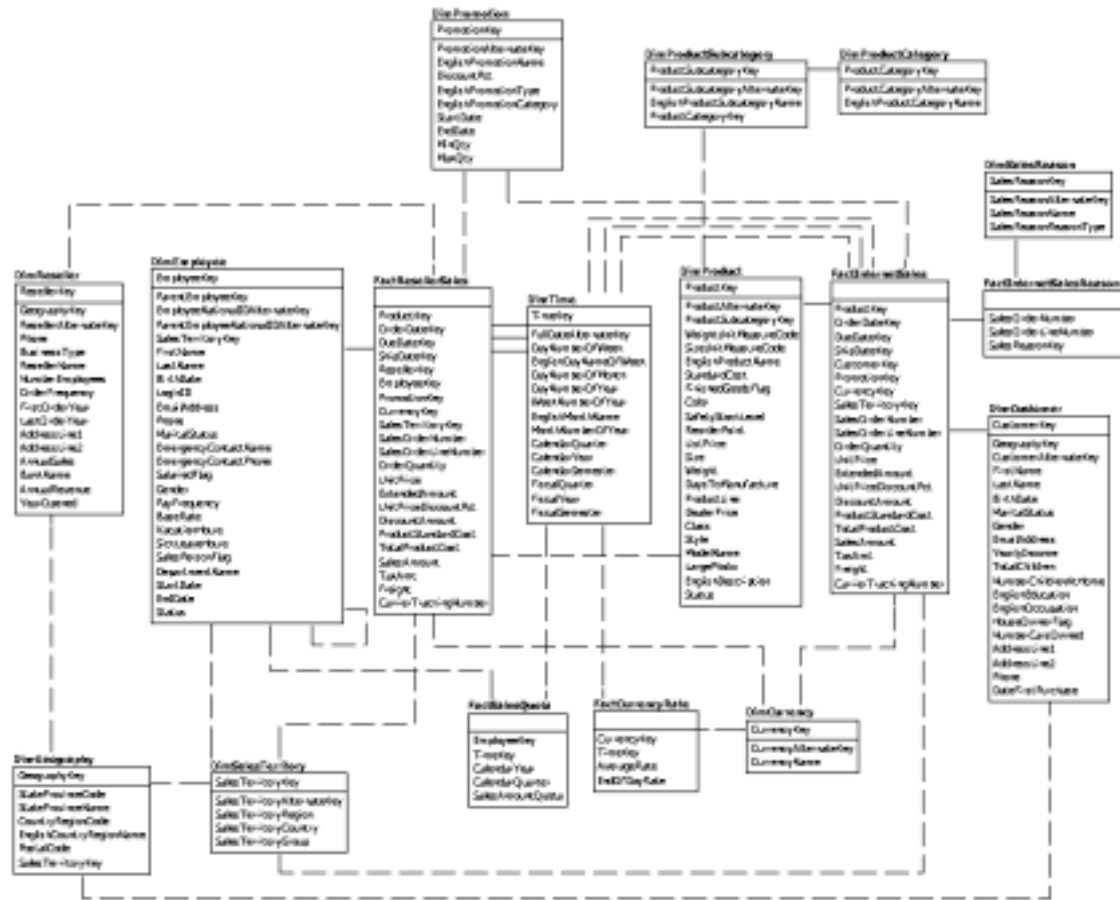


[그림 13] 영업관리 시스템 논리 스타 스키마



[그림 14] 영업관리 시스템 물리 스타 스키마

[그림 13]과 [그림 14]는 “영업관리” 시스템의 스타 스키마이다. 스타 스키마는 사용자 질의에 최적화된 데이터구조를 설계하기 위해 사용되는 논리적인 데이터 모델링 기법이다. 스타 스키마는 비정규화의 특수한 형태로 스타 스키마의 강점은 사실 테이블 내에 있는 데이터에 대한 검색기준으로 차원 테이블의 열을 사용한다는 것이다. 그리고 최종사용자의 관점에서 모델을 되도록 간단히 표현하는데 초점을 맞추었다. 즉 사용자들이 더 쉽게 즉각적으로 이해할 수 있도록 설계되어 진다면 스타 스키마에서 중복성은 크게 문제가 되지 않는다. 스타 스키마는 최적의 질의 응답성능을 위한 설계기법으로 적은 수의 테이블로 구성되면 각 테이블은 다수의 칼럼들을 가진다. 또 각 테이블간의 조인 경로는 명확하다. 이와 같이 간단한 데이터 구조는 데이터베이스 구조에 익숙하지 못한 사용자들도 쉽게 이해될 수 있으며, 사용자 질의에 빠른 응답성능을 내도록 최적화 되어질 수 있다.



[그림 16] 영업관리 시스템 물리 스노우 플레이크 스키마

[그림 15]와 [그림 16]은 “영업관리” 시스템의 스노우 플레이크 스키마이다. 스노우 플레이크 스키마는 스타 스키마의 사실 테이블 구조와 동일하게 유지하면서 차원 테이블이 정규화(일반적으로 제 3 정규형)된 구조로 모델링을 수행했다. 스노우 플레이크 스키마는 차원테이블이 정규화 됨으로써 차원 테이블의 크기가 상당히 줄어들게 되며 각 어트리뷰트가 한 장소에서 유지됨에 따라 데이터 무결성이 높은 수준에서 유지될 수 있다. 이처럼 스노우 플레이크 스키마는 데이터를 저장하는데 필요한 저장공간을 최소화하고 어플리케이션의 유연성을 증가시킨다.

- 스타 스키마와 스노우 플레이크 스키마 비교

[표 6] 스타 스키마와 스노우 플레이크 스키마의 비교표

	스타 스키마	스노우 플레이크 스키마
특징	스타 스키마는 차원테이블을 비-정규화 하여 사용자 질의에 응답하기 위해 필요한 조인의 횟수를 줄임으로써 사용자 질의에 빠르게 응답하도록 만들었다. 스타 스키마는 각 차원 테이블과 사실 테이블 사이에 일대다 관계를 가진 단순한 관계형 모델이지만, 정규화 된 모델은 아니다.	차원 테이블이 정규화 됨으로써 차원 테이블의 크기가 상당히 줄어들게 되며 각 어트리뷰트가 한 장소에서 유지됨에 따라 데이터 무결성이 높은 수준에서 유지된다. 따라서 테이블의 크기도 작다.
장점	최적의 질의 응답성능을 위한 설계기법으로, 적은 수의 테이블로 구성되며 각 테이블은 다수의 칼럼을 가진다. 또 각 테이블 간의 조인 경로는 명확하며, 사용자들에게 쉽게 이해될 수 있으며, 사용자 질의에 빠른 응답성능을 내도록 최적화되어질 수 있다.	한 차원 내에 여러 개의 계층 지원 차원 테이블의 확장이 가능하다. 대용량 데이터 처리를 지원하며, 조직의 변화 등 변화를 잘 수용한다. 정규화 된 구조들은 더 쉽게 갱신되고 유지된다.
단점	특정 차원과 관련된 어트리뷰트 정보들이 하나의 테이블에 저장되므로 차원구조가 복잡해져 차원요소가 많아지면 많은 중복을 감수해야 한다. 또한 저장공간이 많이 필요하며, 차원테이블의 확장성이 없다.	스타스키마에 비해 더 많은 차원테이블을 사용하기 때문에 그 구조가 복잡하며 따라서 유지보수 노력이 가중된다. 또 사용자 질의가 다양한 어트리뷰트가 필요할 때 차원테이블 사이에 많이 조인이 필요하게 되며, 따라서 응답 성능은 심각하게 저하될 수 있다

2.3 데이터 웨어하우스 구축 및 성능 비교

2.3.1 성능 비교 방안

데이터 웨어하우스의 구축 방법에 따른 성능 비교를 위해 일반적인 기업들에서 수행하는 중요한 업무의 하나인 “영업관리” 업무를 중심으로 데이터 웨어하우스를 구축하면서 성능 테스트의 결과를 기록하고, 비교를 통해 보다 효율적인 데이터 웨어하우스 구축 방법을 도출한다.

성능 비교 테스트를 위해 “영업관리” 업무의 스키마 설계는 앞 단원에서 설명 된 스타 스키마와 스노우 플레이크 스키마 두 가지 형태로 설계한다. 그리고 각 스키마별로 ROLAP과 MOLAP 툴을 성능 평가를 하여, 스키마 구조와 OLAP 툴에 대한 연관 관계 및 최적화된 방안을 파악하고자 한다.

성능 비교를 위한 OLAP 툴은 코그노스(Cognos) 사의 파워플레이(Power Play), 마이크로소프트사의 SSAS(SQL Server Analysis Services)를 MOLAP 툴로, 그룹원(Group1)이 인수한 세이전트(Sagent) 사의 Sagent OLAP을 ROLAP 툴로 선정하였고, 데이터 웨어하우스 구축을 위한 데이터베이스로는 오라클(Oracle) RDBMS의 버전 10g를 대상으로 작업을 수행한 결과이다.

2.3.2 데이터 로드를 위한 ETL

데이터웨어하우스 구축을 위한 데이터의 소스들을 여러 가지 형태 (R-DBMS, OO-DBMS, 스프레드시트, ISAM(Indexed Sequential Access Method), 플랫 파일 등)의 원본 데이터로 존재하며, 이 OLTP 시스템의 데이터들의 취합을 담당하는 도구가 ETL 툴이다. ETL 툴을 사용하면 데이터의 저장 형태 및 존재하는 위치에 관계없이 일관성을 유지하면서 데이터를 취합할 수 있으며, 취합되는 데이터의 모니터링을 할 수 있는 장점이 있다.

그리고 스키마에 따른 OLAP의 성능 비교가 목적이므로, ETL 도구는 하나만 선정하여 스키마별로 데이터 로드와 관련한 성능을 비교를 했다.

◆ 스키마별 데이터 로딩

데이터 웨어하우스에 적재되는 데이터는 데이터 웨어하우스의 스키마 구조에 따라 로딩에 사용되는 비용이 많은 차이를 나타낸다. 이는 OLAP 툴을 선정하면서 적용되는 데이터 웨어하우스의 스키마 구조 설계 방법이 달라지기 때문이다. 따라서 본 예제에서는 원본 데이터를 ETL 툴을 사용하여 데이터가 스키마별로 데이터 웨어하우스에 적재되는 소요 시간을 비교했다.

약 90MB 크기에 해당되는 원본 데이터를 스타 스키마 형태의 데이터 웨어하우스와 스노우 플레이크 스키마 형태의 데이터 웨어하우스에 로딩한 결과이다. 이 결과는 동일한 크기의 데이터를 5회 전송하여 로딩되는 결과 시간과 평균 시간을 나타낸다.

[표 7] 스키마별 데이터 로딩 시간(CPU Time) 변화표

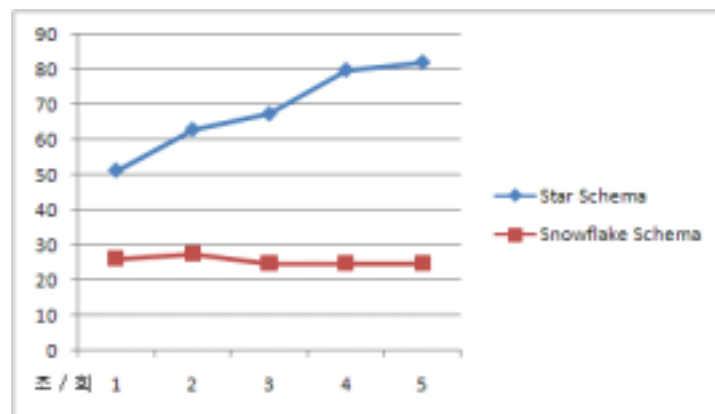
단위 : 초(second)

	1회	2회	3회	4회	5회	평균
Star Schema (큐브 생성 전)	34.234	38.891	35.375	42.860	36.078	37.488
Star Schema (큐브 생성 후)	17.0	24.0	32.0	37.0	46.0	31.200
Snowflake Schema	26.345	27.547	24.859	24.906	24.953	25.722

[표 7]은 스키마별 데이터 로딩을 수행하는데 소요된 CPU 활용 시간을 기록한 표이다. 각 스키마별 데이터 웨어하우스에 로딩하여 소요된 시간을 기록하였으며, 동일 데이터를 5회 전송하여 소요된 시간 및 산술평균값을 표기했다.

스타 스키마의 경우 MOLAP 툴을 사용하여 분석을 하는 경우가 대부분이므로 큐브까지 생성되는 것을 원칙으로 해야 한다. “큐브 생성 전 ”은 데이터 웨어하우스에 로딩되는 시간까지 표현한 것이고, “큐브 생성 후 ”는 데이터 웨어하우스의 데이터로 큐브 생성을 한 시간을 표현하였으며, 각각 기록했다.

[표 7] 스키마별 데이터 로딩 시간(CPU Time) 변화표의 기록 내용을 비교해 보면, 스타 스키마의 소요 시간이 더 많이 증가하는 것을 볼 수 있다. 이는 스타 스키마는 비정규화 된 구조로 중복된 데이터가 동일 테이블에 존재하므로 용량이 더 커지기 때문에 데이터 로딩 소요시간을 더 많이 필요로 한다. 또한 스타 스키마는 큐브에 적재되는 것도 포함해서 시간을 계산해야 하므로 데이터 로드에서 걸리는 시간은 스타 스키마가 스노우 플레이크 스키마에 비해 훨씬 더 많이 소요된다. 반면 스노우 플레이크 스키마는 정규화를 통해 중복된 데이터가 최소화되도록 설계하므로 데이터의 로딩 성능이 스타 스키마보다 상대적으로 좋게 나타난다. 스타 스키마의 경우도 적절한 정규화 설계를 통해 보다 성능이 좋아지게 된다.



[그림 17] 스키마별 데이터 로딩 시간(CPU Time) 그래프

[그림 17]은 스키마별 데이터 로딩 시간(CPU Time) 변화를 그래프로 표현했다. 그래프의 내용을 살펴 보면 스타 스키마의 데이터 로딩 시간이 스노우 플레이크 스키마보다 상대적으로 많이 소요되며, 데이터가 누적되면서 소요 시간이 점차 증가하는 것을 볼 수 있다.

2.3.3 스키마에 따른 용량 비교

데이터 웨어하우스에 누적되는 데이터 량은 시간의 흐름에 따라 지속적인 증가하는 것이 일반적이다. 왜냐하면, 전송된 OLTP 데이터는 데이터 웨어하우스의 데이터베이스에 누적되고, 요약 데이터는 시간의 흐름에 정비례하게 증가되기 때문이다. 따라서 데이터 웨어하우스 관리자는 특정 기간별로 데이터의 증가량을 예측하고, 저장장치의 적절한 용량을 확보해야 하며, 자신들의 데이터 웨어하우스에 맞는 스키마를 설계해야 한다.

다음에 제시되는 예제도 동일한 원본 데이터가 ETL 도구를 이용하여 데이터 웨어하우스에 데이터가 저장될 때, 각 스키마별 저장되는 용량을 비교했다.

◆ 스키마별 데이터 전송 용량

데이터 웨어하우스에 적재되는 데이터는 소요되는 시간과 함께 누적되는 데이터의 용량을 고려하여 설계해야 한다. 데이터 웨어하우스의 주요 역할 중의 하나는 대량의 데이터를 통합하는 역할이다. 따라서 데이터 웨어하우스에 저장되는 데이터 용량에 대해서 스키마별 증가율에 대해서 비교했다.

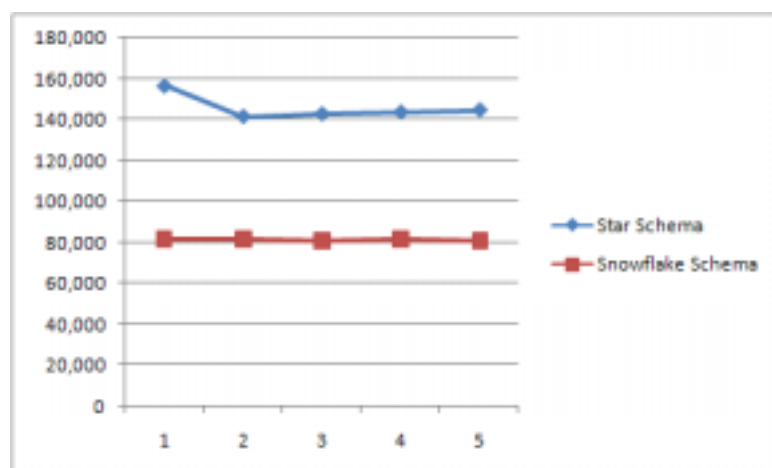
약 90MB에 해당되는 원본 데이터를 스타 스키마 형태의 데이터 웨어하우스와 스노우 플레이크 스키마 형태의 데이터 웨어하우스에 로딩한 결과이다. 이 결과도 동일한 크기의 데이터를 5회 전송하여 이에 따른 물리적인 용량의 변화를 기록했다.

[표 8] 스키마별 데이터 용량(Disk Memory) 증가표

단위 : KB

	1회	2회	3회	4회	5회	평균
Star Schema (DW database)	118,784	116,736	116,736	116,736	117,760	117,350
Star Schema (큐브)	37,763	24,687	25,975	26,858	26,964	28,449
Snowflake Schema	81,920	81,920	80,896	81,920	80,896	81,510

[표 8] 스키마별 데이터 용량(Disk Memory)의 증가를 기록한 표이다. 각 스키마별 데이터 웨어하우스에 로딩하여 증가한 물리적인 데이터 량을 기록하였으며, 동일 데이터를 5회 전송하여 증가된 용량 및 산술평균값을 표기했다.



[그림 18] 스키마별 데이터 용량(Disk Memory) 변화표

[표 8] 스키마별 데이터 용량(Disk Memory) 증가표의 기록내용을 비교해 보면 스타 스키마의 데이터 용량이 더 커지는 것을 볼 수 있다. 이는 스타 스키마는 비정규화 된 구조로 중복된 데이터가 동일

테이블에 존재하므로 용량이 더 커지게 된다. 또한 스타 스키마는 큐브에 적재되는 데이터들도 포함해서 용량을 계산해야 하므로 데이터 로드 후 증가되는 물리적인 데이터 용량은 스타 스키마가 스노우 플레이크 스키마보다 더 크다.

[그림 18]은 스키마별 데이터 용량(Disk Memory) 변화를 그래프로 표현한 내용이다. 스타 스키마는 데이터 웨어하우스에 저장되는 데이터의 물리적인 용량과 큐브의 물리적인 용량을 합한 크기를 표기했다. 그래프의 내용을 보면 스타 스키마의 데이터 증가율이 스노우 플레이크 스키마보다 상대적으로 많이 증가하는 것을 확인 할 수 있다.

2.3.4 OLAP 도구별 성능

OLAP 도구들은 데이터 웨어하우스에 저장된 주제 중심의 데이터를 사용자가 별도의 프로그래밍 없이 데이터를 분석할 수 있게 해주는 도구이다. 사용자는 분석하고자 하는 사실 데이터와 차원 데이터를 선택하여 데이터를 분석할 수 있다.

MOLAP은 다차원 데이터베이스를 큐브 형태의 데이터 뷰로 작성하여, 사용자가 OLAP 도구를 이용해 다양하게 분석할 수 있도록 구성되어 있고, 다른 데이터베이스에서 추출한 요약된 데이터를 저장하게 된다. 이 큐브에서는 보통 다음에 갱신 작업을 수행하기 전까지의 정적인 데이터가 보관되어 검색의 속도를 향상시킨다.

ROLAP은 관계형 데이터베이스를 기반으로 사용자의 요건 분석을 취합하여 사용자 관점의 뷰를 메타 데이터로 구성하여, 이를 통해 사용자 질의 보고서를 작성하고 다양한 분석을 할 수 있도록 구성되어 있다. ROLAP 서버는 미리 계산된 큐브를 사용하지 않는 대신에 데이터 웨어하우스 전용 데이터베이스나 관계형 데이터베이스를 데이터 소스로 이용하고, 중간 단계 서버(OLAP 전용 서버)에 데이터를 보관한다.

여기에서는 데이터의 로딩 후, 분석 데이터를 검색하여 성능을 기록하고, 같은 조건의 검색을 재 기록하여 최초와 이후의 검색 속도의 차이 및 스타 스키마와 스노우 플레이크 스키마의 검색 성능, OLAP 도구별 성능을 비교했다.

◆ 스키마별 검색 쿼리

스타 스키마와 스노우 플레이크 스키마의 쿼리 성능 비교를 위해 저장된 데이터들을 대상으로 제품별 기간별 매출현황, 제품별 지역별 매출현황, 그리고 프로모션별 제품별 매출현황에 대한 쿼리를 수행했다. 영업관리에서 검색과 통계를 위해 가장 많이 사용되는 제품, 기간 그리고 판매지역을 차원으로 측정지표를 매출로 선정하여 다차원 분석을 하도록 했다.

- 1) 제품별 기간별 매출현황
- 2) 제품별 판매지역별 매출현황
- 3) 프로모션별 제품별 매출현황

위 3가지 중 제품별 기간별 매출현황의 쿼리의 내용은 아래와 같다.

```
SELECT
    "DWstarSchema"."dbo"."DimTime"."CalendarYear",
    "DWstarSchema"."dbo"."DimTime"."MonthNumberOfYear",
    "DWstarSchema"."dbo"."DimProduct"."ProductCategoryName",
    "DWstarSchema"."dbo"."DimProduct"."ProductSubCategoryName",
    "DWstarSchema"."dbo"."DimProduct"."EnglishProductName" AS ProductName,
    SUM("DWstarSchema"."dbo"."FactInternetSales"."SalesAmount") AS SalesAmount
FROM
    "DWstarSchema"."dbo"."FactInternetSales"
INNER JOIN "DWstarSchema"."dbo"."DimProduct" ON
    "DWstarSchema"."dbo"."DimProduct"."ProductKey" =
        "DWstarSchema"."dbo"."FactInternetSales"."ProductKey"
INNER JOIN "DWstarSchema"."dbo"."DimTime" ON
    "DWstarSchema"."dbo"."DimTime"."TimeKey" =
        "DWstarSchema"."dbo"."FactInternetSales"."OrderDateKey"
GROUP BY
    "DWstarSchema"."dbo"."DimTime"."CalendarYear",
    "DWstarSchema"."dbo"."DimTime"."MonthNumberOfYear",
    "DWstarSchema"."dbo"."DimProduct"."ProductCategoryName",
    "DWstarSchema"."dbo"."DimProduct"."ProductSubCategoryName",
    "DWstarSchema"."dbo"."DimProduct"."EnglishProductName"
```

[그림 19] ROLAP 스타 스키마 쿼리

[그림 19]는 제품별 기간별 매출현황에 대한 스타 스키마의 쿼리를 나타낸 내용이다. 제품의 구성은 제품 카테고리, 서비스 카테고리, 제품명으로 구성되어 있으며, 제품 차원 테이블의 어트리뷰트로 존재한다. 기간은 년도, 월의 형태로 구성으로 되었다. 쿼리의 조건은 전체 기간과 전체 제품에 대해서 조회한다.

```
SELECT
    "DWsnowSchema"."dbo"."DimTime"."CalendarYear",
    "DWsnowSchema"."dbo"."DimTime"."MonthNumberOfYear",
    "DWsnowSchema"."dbo"."DimProductCategory"."EnglishProductCategoryName" AS
    "ProductCategoryName",
    "DWsnowSchema"."dbo"."DimProductSubcategory"."EnglishProductSubcategoryName" AS
    "ProductSubCategoryName",
    "DWsnowSchema"."dbo"."DimProduct"."EnglishProductName" AS ProductName,
    SUM("DWsnowSchema"."dbo"."FactInternetSales"."SalesAmount") AS SalesAmount
FROM
    "DWsnowSchema"."dbo"."FactInternetSales"
INNER JOIN "DWsnowSchema"."dbo"."DimProduct" ON
    "DWsnowSchema"."dbo"."DimProduct"."ProductKey" =
        "DWsnowSchema"."dbo"."FactInternetSales"."ProductKey"
INNER JOIN "DWsnowSchema"."dbo"."DimTime" ON
    "DWsnowSchema"."dbo"."DimTime"."TimeKey" =
        "DWsnowSchema"."dbo"."FactInternetSales"."OrderDateKey"
INNER JOIN "DWsnowSchema"."dbo"."DimProductSubcategory" ON
```

```

"DWsnowSchema"."dbo"."DimProductSubcategory"."ProductSubcategoryKey" =
"DWsnowSchema"."dbo"."DimProduct"."ProductSubcategoryKey"
INNER JOIN "DWsnowSchema"."dbo"."DimProductCategory" ON
"DWsnowSchema"."dbo"."DimProductCategory"."ProductCategoryKey" =
"DWsnowSchema"."dbo"."DimProductSubcategory"."ProductCategoryKey"
GROUP BY
"DWsnowSchema"."dbo"."DimTime"."CalendarYear",
"DWsnowSchema"."dbo"."DimTime"."MonthNumberOfYear",
"DWsnowSchema"."dbo"."DimProductCategory"."EnglishProductCategoryName",
"DWsnowSchema"."dbo"."DimProductSubcategory"."EnglishProductSubcategoryName",
"DWsnowSchema"."dbo"."DimProduct"."EnglishProductName"

```

[그림 20] ROLAP 스노우플레이크 스키마 쿼리

[그림 20]은 제품별 기간별 매출현황에 대한 스노우플레이크 스키마의 쿼리를 나타낸 것이다. 제품의 구성은 제품 카테고리, 서비스 카테고리, 제품명으로 구성되어 있으며, 각 정규화 된 차원 테이블로 구성되었으며, 기간은 년도, 월의 형태로 구성으로 되었다. 쿼리의 조건은 전체 기간과 전체 제품에 대해서 조회한다.

```

SELECT
  NON EMPTY { [Measures].[Sales Amount] } ON COLUMNS,
  NON EMPTY { ([Dim Product].[계층].[English Product Name].ALLMEMBERS *
    [Order Date].[Calendar Year - Month Number Of Year].[Month Number Of
    Year].ALLMEMBERS ) }
  DIMENSION PROPERTIES MEMBER_CAPTION, MEMBER_UNIQUE_NAME ON ROWS
FROM [DW TimeProduct]

```

[그림 21] MOLAP 스노우플레이크 스키마 MDX쿼리

[그림 21]은 제품별 기간별 매출현황에 대한 조회를 위해 MOLAP 도구를 사용한 MDX 쿼리를 나타낸 것이다. 제품의 구성은 제품 카테고리, 서비스 카테고리, 제품명으로 구성되어 있으며, 기간은 년도, 월의 형태로 구성으로 되었다. 차원을 계층 구조로 구성되어있고, 쿼리의 조건은 전체 기간과 전체 제품에 대해서 조회한다.

OLAP도구를 사용하여 검색한 쿼리 결과 화면은 아래와 같다.

	2013	2014	2015	2016	2017	2018
Accessories						
Bike Racks	374,400.00	374,400.00	374,400.00	374,400.00	374,400.00	374,400.00
Bike Stands	408,312.00	408,312.00	408,312.00	408,312.00	408,312.00	408,312.00
Baskets and Cages	558,547.44	558,547.44	558,547.44	558,547.44	558,547.44	558,547.44
Clearance	69,355.60	69,355.60	69,355.60	69,355.60	69,355.60	69,355.60
Fenders	447,079.20	447,079.20	447,079.20	447,079.20	447,079.20	447,079.20
Helmets	2,170,779.60	2,170,779.60	2,170,779.60	2,170,779.60	2,170,779.60	2,170,779.60
Hydration Packs	391,308.64	391,308.64	391,308.64	391,308.64	391,308.64	391,308.64
Tires and Tubes	2,390,394.36	2,390,394.36	2,390,394.36	2,390,394.36	2,390,394.36	2,390,394.36
Bikes						
Mountain Bikes	105,421,493.09	105,421,493.09	105,421,493.09	105,421,493.09	105,421,493.09	105,421,493.09
Road Bikes	163,689,356.96	163,689,356.96	163,689,356.96	163,689,356.96	163,689,356.96	163,689,356.96
Touring Bikes	37,795,108.56	37,795,108.56	37,795,108.56	37,795,108.56	37,795,108.56	37,795,108.56
Clothing						
Caps	181,163.36	181,163.36	181,163.36	181,163.36	181,163.36	181,163.36
Gloves	392,672.16	392,672.16	392,672.16	392,672.16	392,672.16	392,672.16
Jackets	1,854,103.48	1,854,103.48	1,854,103.48	1,854,103.48	1,854,103.48	1,854,103.48
Shirts	696,260.52	696,260.52	696,260.52	696,260.52	696,260.52	696,260.52
Socks	49,301.16	49,301.16	49,301.16	49,301.16	49,301.16	49,301.16
Wrist	345,186.00	345,186.00	345,186.00	345,186.00	345,186.00	345,186.00

[그림 22] 제품별 기간별 매출현황(ROLAP)

[그림 22] 제품별 기간별 매출현황(ROLAP)은 ROLAP 도구를 사용하여 검색한 결과를 캡처한 화면이다.

	2013	2014	2015	2016	2017
Accessories					
Bike Racks	374,400.00	374,400.00	374,400.00	374,400.00	374,400.00
Bike Stands	408,312.00	408,312.00	408,312.00	408,312.00	408,312.00
Baskets and Cages	558,547.44	558,547.44	558,547.44	558,547.44	558,547.44
Clearance	69,355.60	69,355.60	69,355.60	69,355.60	69,355.60
Fenders	447,079.20	447,079.20	447,079.20	447,079.20	447,079.20
Helmets	2,170,779.60	2,170,779.60	2,170,779.60	2,170,779.60	2,170,779.60
Hydration Packs	391,308.64	391,308.64	391,308.64	391,308.64	391,308.64
Tires and Tubes	2,390,394.36	2,390,394.36	2,390,394.36	2,390,394.36	2,390,394.36
Bikes					
Mountain Bikes	105,421,493.09	105,421,493.09	105,421,493.09	105,421,493.09	105,421,493.09
Road Bikes	163,689,356.96	163,689,356.96	163,689,356.96	163,689,356.96	163,689,356.96
Touring Bikes	37,795,108.56	37,795,108.56	37,795,108.56	37,795,108.56	37,795,108.56
Clothing					
Caps	181,163.36	181,163.36	181,163.36	181,163.36	181,163.36
Gloves	392,672.16	392,672.16	392,672.16	392,672.16	392,672.16
Jackets	1,854,103.48	1,854,103.48	1,854,103.48	1,854,103.48	1,854,103.48
Shirts	696,260.52	696,260.52	696,260.52	696,260.52	696,260.52
Socks	49,301.16	49,301.16	49,301.16	49,301.16	49,301.16
Wrist	345,186.00	345,186.00	345,186.00	345,186.00	345,186.00

[그림 23] 제품별 기간별 매출현황(MOLAP)

[그림 23] 제품별 기간별 매출현황(MOLAP)은 MOLAP 도구를 사용하여 검색한 결과를 캡처한 화면이다.

◆ 검색 성능의 비교와 특징

스키마별 OLAP툴을 적용하여 쿼리의 성능 결과를 기록하였으며, 쿼리의 결과는 개성이 되지 않은 최초 쿼리의 결과를 기록했다.

[표 9] ROLAP 툴을 사용한 성능 결과

단위 : 초(second)

스키마 \ 년도	2003	2004	2005	2006	2007	평균
Star Schema	15.50	15.45	14.44	15.39	16.42	15.44
Snowflake Schema	14.23	14.56	14.72	13.52	13.63	14.13

[표 9] ROLAP 툴을 사용한 성능 결과를 살펴보면 각 년도별 동일한 데이터로 구성되어 있는 데이터 웨어하우스의 데이터를 ROLAP 도구를 사용하여 조회한 쿼리 결과의 성능을 기록했다. 각 년도별 수행 시간을 기록하였으며, 5개 년도의 산술평균도 표기했다.

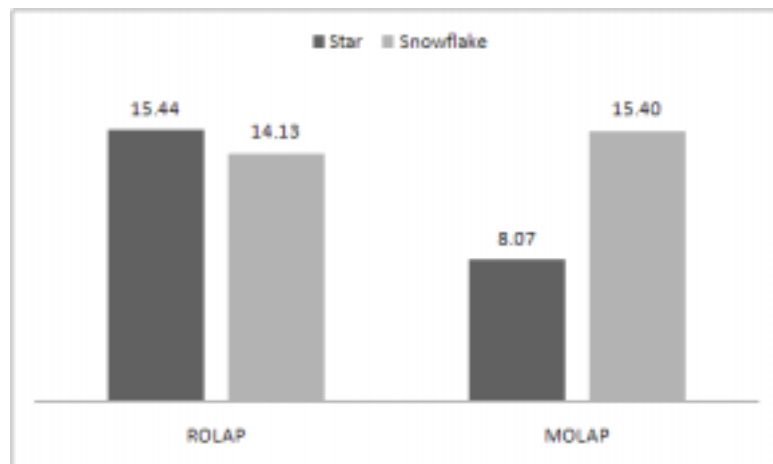
[표 10] MOLAP 툴을 사용한 성능 결과

단위 : 초(second)

구분 \ 년도	2003	2004	2005	2006	2007	평균
Star Schema	6.99	8.05	8.95	7.90	8.47	8.07
Snowflake Schema	18.95	12.42	13.20	12.95	19.49	15.40

[표 10] MOLAP 툴을 사용한 성능 결과를 살펴보면 각 년도별 동일한 데이터로 구성되어 있는 데이터 웨어하우스의 데이터를 MOLAP 도구를 사용하여 조회한 쿼리 결과의 성능을 기록했다. 각 년도별 수행 시간을 기록하였으며, 5개년도의 산술평균도 표기했다.

단위 : 초(second)



[그림 24] OLAP 도구별 쿼리 성능 비교

[그림 24] OLAP 도구별 쿼리 성능 비교는 성능의 평균을 그래프로 표현했다. ROLAP 도구는 정규화가 잘되어 있는 스노우 플레이크 스키마 설계에서 성능이 좋으며, MOLAP 도구는 스타 스키마 설계에서

성능이 월등하게 차이가 나는 것을 확인 할 수 있다.

MOLAP의 형태는 큐브가 생성되면서 집계데이터를 미리 만들기 때문에 검색 결과가 ROLAP 보다는 빠르게 나왔으나, 쿼리의 재사용에 따른 캐싱 및 집계 데이터의 활용을 통해 ROLAP 도구도 최적화된 설계를 적용할 수 있다. 따라서 본 예제에서 제시하는 내용은 각 OLAP 도구의 특성에 맞게 설계하는 것이 중요하다는 것이다.

데이터 웨어하우스의 스키마 설계 시 고려해야 할 사항으로, 기간계 시스템 데이터의 트랜잭션 변화에 따라 스키마 설계 방법을 선택해야 한다. 기간계 시스템 데이터의 변동이 많은 경우에는 데이터 웨어하우스의 데이터도 로딩되는 데이터의 범위가 사실 테이블과 함께 차원 테이블의 로드되는 데이터의 양이 많이 진다. 이런 경우는 데이터 웨어하우스의 데이터 저장공간을 작게 사용하고, 데이터를 로드 하는 시간을 최소화하는 스키마 구조의 설계 방법인 스노우 플레이크 스키마로 설계해야 한다. 반대로 기간계 시스템 데이터의 트랜잭션이 변화가 심하지 않고 정적인 요소가 많은 경우에는 데이터 웨어하우스의 사실 테이블의 집계 정보와 차원 테이블의 정보가 유지되는 기간이 길어지므로 스타 스키마로 설계하는 것이 효율적인 설계 방법이다.

기간계 시스템 데이터의 트랜잭션이 증가량이 많아지면 데이터웨어하우스의 물리적인 공간도 지속적으로 증가하게 된다. 물리적인 공간의 증가 변화율 또한 스키마 설계에 중요한 요소이다. 데이터의 증가량이 크다면 이를 효율적으로 수용하고, 관리할 수 있는 스키마를 선택해야 하는데, 정규화가 잘 되어 있어 중복 데이터가 최소화되어있는 즉, 저장공간을 작게 사용하는 스노우 플레이크 스키마로 설계하는 것이 바람직하다. 반대로 데이터웨어하우스의 공간의 증가변화율이 적은 경우는 검색의 성능을 최대로 활용할 수 있는 스타스키마로 설계하는 것이 효율적인 설계 방법이다

대부분 기업 및 조직들이 OLAP 도구에 의존하는 경우가 있다. 도구에 의존하기 보다는 자신들의 데이터 성격을 파악해서 결정하는 것이 더 좋다.