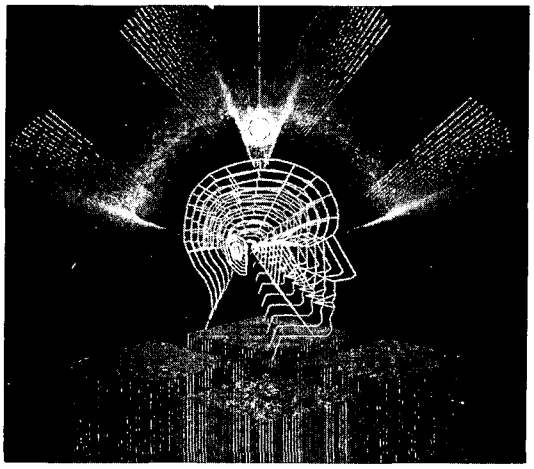
데이터베이스 설계

전문가 리포트

차 례

나 민 영
육군사관학교 전산학교수
University of Florida
전산학박사



1. 데이터베이스 설계 기초

1.1 데이터베이스와 데이터베이스 설계

우리 주위에는 수많은 데이터가 여러 형태로 존재하고 있다. 기업의 경우에는 먼저 종업원에 대한 신상명세에 대한 사항뿐만 아니라 각 부서의 인적구성, 부서별 판매형황등 많은 데이터가 있을 것이며 학교의 경우에는 학생의 등록과 교과목 및 성적에 관한 데이터가 있을 것이다. 또한 관공서와 같은 정부 기관도 취급해야 할 데이터가 이루 헤아릴 수 없이 많다.

그러나 일반적으로 이들 데이터는 구조화되어 있지 못하여 기록이 중복되고 기록 양식이 상이하며 또 검색 및 수정에 어려움이 있는 등 효율적인 업무 추진에 많은 지장을 초래하여 왔다. 더군다나 데이터 관리를 컴퓨터에 의존해왔던 일부 영역에서조차도 늘어나는 데이터 양의 관리가 점차 어려워져 과거의 화일 시스템으로는 이를 감당하기가 어렵게 되었다. 그러나 다행히도 컴퓨터 관련 기술이 급속히 발달하여 많은 양의 데이터를 처리하고 관리할 수 있는 기술들이 제공되고 있다. 이러한 기술중 가장 대표적인 것이 데이터베이스 시스템이다. 데이터베이스 시스템(Database System)이란 컴퓨터를 중심으로 데이터를 저장하고 관리해서 정보를 생성해내는 시스템을 말하는 것으로서 이는 크게 데이터베이스 관리시스템(DBMS)과 데이터베이스(DB)로 이루어진다. 데이터베이스 관리 시스템(Data Base Management Systems : DBMS)이란 화일시스템에서 야기된 데이터의 종속성과 중복성의 문제를 해결하기 위해 제안된 시스템으로 데이터베이스의 구성, 접근방법, 관리유지 등에 대한 모든 책임을 지고 있는 소프트웨어 패키지이다.

데이터베이스란 서로 연관이 있는 데이터의 집합을 말하는 것으로서 데이터베이스의 구축은 모든 분야에 있어 정보관리의 주요 관건이 되고 있다. 여기서 데이터란 기록될 수 있거나 의미있는

사실을 가리킨다. 물론 데이터베이스란 일반적으로는 모든 데이터의 집합을 가리키나 여기에서 말하는 데이터베이스는 약간 더 제한적으로 사용된다. 따라서 데이터베이스는 다음과 같은 특성을 같는다.

- 데이터베이스는 논리적으로 밀접하게 연관된 데이터의 집합이다.
- 데이터베이스는 특별한 목적을 가지고 설계되고 구축된다. 즉 사용자 그룹이 정해지며 이 사용자 그룹이 관심을 갖는 응용 프로그램이 있게 된다.
- 데이터베이스는 현실세계의 일부 측면을 표현한다. 따라서 현실세계의 변화는 데이터베이스에 반영되어야 한다.

데이터베이스는 어느 규모이든 관계없다. 예를 들어, 주소록 같은 단순한 레코드들로만 구성된 것도 있는 반면 도서관 장서처럼 엄청난 양을 다루는 것도 있다. 단일 사용자나 몇명의 사용자에 의해 사용되는 조그만 데이터베이스에 있어서는 데이터베이스 설계는 그리 복잡하지 않을 수도 있다. 그러나 큰 조직체내에서 중간규모 또는 그 이상의 데이터베이스가 정보시스템으로서 구성될 때의 데이터베이스 설계는 그리 쉽지가 않다. 그 이유는 많은 사용자들이 데이터베이스를 사용하게 되므로 그들의 요구사항을 만족시켜 주어야 하기 때문이다.

데이터베이스 설계가 잘못되면 많은 수고와 노력을 들여 데이터베이스를 구축했다 하더라도 사용되지 않고 그냥 사장될 가능성이 많다. 이는 매우 비효율적인 일이다. 따라서 데이터베이스를 제대로 설계하는것은 데이터베이스 시스템 구축 및 운용을 위한 가장 우선적이고 기초가 되는 작업이라 할 수 있다. 좋은 데이터베이스를 설계하기 위해서는 앞으로 다룰 데이터베이스 설계 절차를 따라 설계하여야 한다.

1.2 데이터 모델링

데이터 모델이란 현실세계의 데이터 구조를 기술하는 개념적이 도구이다. 데이터베이스 구조란 데이터 타입, 관계성, 데이터에 관한 제약조건 등에 관한 것들을 말한다. 설계자들은 현실세계의 표현인 스키마를 만드는데 모델을 사용한다. 이와같이 데이터 모델을 사용하여 현실세계를 표현하는 작업을 데이터모델링이라 한다. 스키마의 우수성은 데이터베이스 설계자의 기술뿐만 아니라 선택된 모델에도 의존한다. 모든 데이터 모델에서는 이해하기 쉽고 간결하게 표현하기 위해 추상화 메카니즘을 사용한다. 추상화(Abstraction)란 어떤 객체에서 관계가 적은 특성은 생략하고 주요 특성을 선택할때 사용되는 지적인 과정을 말한다. 다시 말하면 추상화는 어떤 객체를 가장 기본적인 특성으로 압축시킬때 사용된다. 그림 1은 추상화된 자전거를 보여준다. 데이터베이스 설계에 쓰이는 추상화에는 계층화(Classification), 집단화(Aggregation), 일반화(Generalization)의 3가지가 있다.

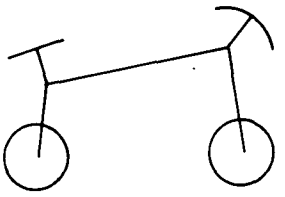


그림1. 추상화된 자전거

● 계층화

계층화는 같은 성질을 갖는 현실 세계의 객체들을 클래스라는 하나의 개념으로 정의하는데 사용된다. 예를 들어 자전거라는 개념은 모든 자전거를 멤버로 갖는 클래스이고, 달(month)이라는 개념은 일월부터 십이월까지를 멤버로 같는 개념이다. 계층화의 표현은 트리구조로서 이루어질 수 있는데 루트는 클래스를, 잎은그 클래스의 원소를 나타낸다(그림2 참조). 트리의 각 간선은 리프토드는 루트에 의해 표현되는 클래스의 멤버가 됨을 나타낸다(IS_MEMBER_OF). 현실 세계에서의 한 객체는 여러가지 방법으로 계층화될 수 있다.

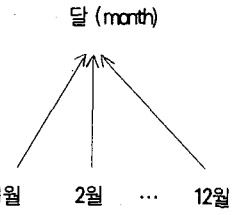


그림2 계층화의 예

● 집단화

집단화는 클래스들의 셑으로 구성되는 새로운 클래스를 정의하는데 사용되는 개념이다. 각 원소 클래스는 새로운 클랫의 구성요소임을 의미한다. 예를 들어, 자전거는 페달, 바퀴, 및 핸들과 같은 클래스들로 구성되어 있다. 또한 사람이란 크랠스는 이름, 성별, 직위 등의 클래스로 구성된다. 집단호는 트리구조로서 표현될 수 있다. 루트는 집단화되어 이루어진 클래스를 나타내고 간선은 리크클래스가 루트에 의해 표현되는 클래스의 부품임을 나타낸다(IS_PART_OF). 계층화와의 차이를 위해 간선은 이중선으로 표시한다(그림3 참조).

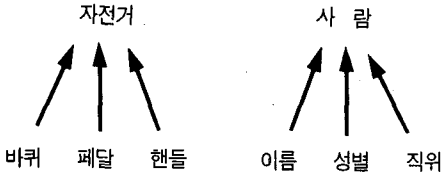


그림3. 집단화의 예

● 일반화

일반화란 여러 클래스 간의 공통적인 특성을 파악하는 과정을 말한다. 즉, 일반화는 둘 또는 그 이상의 클래스의 요소간에 서브셑 관계를 정의하는 개념이다. 예를 들어 운송기구는 자전거의 일반화이다. 왜냐하면 모든 자전거는 운송수단이기 때문이다. 마찬가지로 클래스 사람은 그림 4에서 보는 바와 같이 클래스 남성과 클래스 여성의 일반화이다. 일반화 역시 트리구조로서 표현될 수 있다. 이 트리에서 모든 노드는 클래스를 나타내는데 루트 노드는 generic 클래스를 나타내고 간선은 리프 클래스가 루트 클래스의 서브셑임을 의미한다(IS_A). 일반화가 다른 추상화 개념과 다른점은 방향을 가진 간선이 하나 밖에 없다는 점이다. 일반화에 있어서 generic 클래스에 적응된 모든 특성은 서브셑 클래스로 계승(inherit)된다. 예를 들어 그림 3에서 사람 클래스는 이름, 성별, 직위가 합해져 이룩된 클래스이고, 그림 4에서 남성과 여성은 사람의 서브셑이므로 남성 역시 이름, 성별, 직위의 집단화로 생각할 수 있다. 이 계승특성때문에 일반화는 매우 중요하게 취급된다.

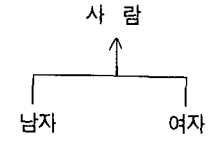


그림4. 일반화의 예

데이터 모델은 그 기능상 크게 개념 데이터 모델과 구현 데이터 모델의 두가지로 나뉜다. 개념 데이터 모델이란 사용자가 데이터를 감지하는 방법에 가까운 개념을 제공하고 구현 데이터 모델이란 데이터가 컴퓨터내에서 조직되는 방법에 크게 무관하지 않으면서도 사용자가 이해할 수 있는 개념을 제공한다. 개념 데이터 모델의 대표적인 것으로서는 ER 모델을 들 수 있고, 구현 데이터 모델의 대표적인 것으로서는 관계형 모델을 들 수 있다. 데이터베이스 설계에 있어서 먼저 현실세계를 개략적으로 기술하는데 개념적 모델을 이용하고 그 다음이 개념적 스키마를 논리 스키마로 전환한다. 스키마(schema)란 현실 세계의 특정한 한 부분의 표현으로서 특정 데이터 모델을 이용해서 만들어진다. 좀 더 자세히 설명하면 스키마는 한 조직에서 관심있는 부분의 데이터 구조를 기술하는 언어 또는 그래프 표현의 집합으로써 시간에 따라 불변인 특성을 갖는다. 스키마에 의해 정의된 데이터 구조에 맞는 데이터의 집합을 인스턴스(instance)라 하며 이는 시간에 따라 변하는 다이내믹한 특성을 갖고 있다. 각 스키마는 여러 인스턴스를 가질 수 있으며 특정 시간에 있어 데이터베이스의 상태는 그 당시의 인스턴스들에 의해 결정된다.

1.3 데이터베이스 설계과정

데이터베이스 설계는 통산 요구사항 수집 및 분석, 개념적 설계, 논리적 설계, 물리적 설계의 4단계로 나뉘나 여기서는 좀 더 자세히 설명하기 위하여 다음 그림 5와 같은 6단계로 나뉘어 설명한다.

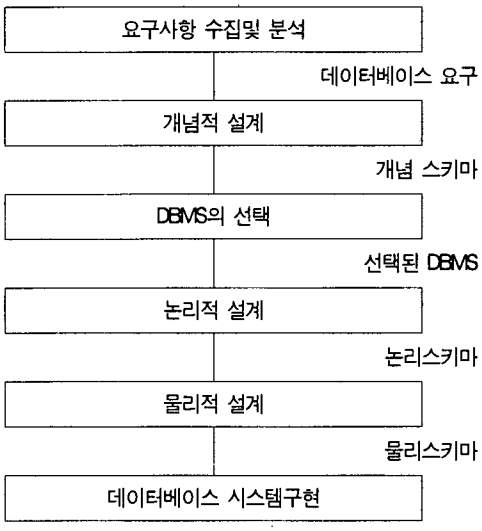


그림5. 데이터베이스 설계과정

● 요구사항 수집 및 분석

데이터베이스를 효과적으로 설계하기 이전에 사용자의 요구사항과 데이터베이스의 사용용도를 가능한한 자세히 해야한다. 이러한 단계를 요구사항 수집 및 분석단계라 하며 다음과 같은 작업들이 이에 해당된다.

○ 사용자 그룹과 응용분야의 인식

데이터베이스를 사용하는 주요 응용 분야와 사용자 그룹이 인식된다. 각 사용자 그룹에 있는 핵심 요원들이 주 참여자로 선택된다.

○ 현존하는 문서의 점검

응용에 관계된 문서 또는 폼, 보고서, 조직표등을 검토하고 분석하여 이들이 사용자 요구사항 수집 및 분석에 어떤 영향을 주는가를 검토한다.

○ 운용 처리 환경의 분석

현재의 환경과 앞으로의 계획을 연구한다. 여기에는 트랜잭션의 타입과 그 빈도, 트랜잭션의 입출력, 정보의 흐름등이 포함된다.

○ 설문조사 및 인터뷰

데이터베이스 사용 예정자들에 대하여 필요하다면 설문조사를 실시하고 핵심요원들에 대해서는 인터뷰를 통해 좀 더 정확하고 중요한 정보를 획득한다.

이상과 같은 방법들로 얻어진 결과는 대개 비구조적이고 비정형적인 형태이다. 비정형적인 결과로부터 정형적인 결과를 얻으려면 정형적인 요구사항 명세 기법을 사용해야 한다. 이러한 기법에는 HIPO(Hierarchical Input Process Output), SADT(Structured Analysis and Design), DFD(Data Flow Diagram) 등이 쓰인다.

요구사항 수집 및 분석 단계에는 많은 시간이 소요되나 앞으로 시스템의 성공을 위해서는 필수적이다. 최근에는 컴퓨터를 이용하여 요구의 일치성과 완전성을 검사/보완하는 기법이 제안되고 있다.

● 개념적 설계

데이터베이스 설계의 두번째 단계는 개념적 스키마를 설계하는 개념적 설계 단계로서 그 중요도에 비해 아직 많은 사람들이 제대로 이해하지 못하거나 또한 적용하지 못하여 데이터베이스 설계가 잘못되는 경우가 종종 있다. 이 단계에서 결과로서 산출되는 개념적 스키마는 DBMS에 무관한 상위 데이터 모델로서 데이터베이스를 구현하는데 직접 사용되지는 않으나 다음과 같은 이유 때문에 매우 중요하게 취급된다.

○ 개념적 설계의 목표는 데이터베이스의 구조, 의미, 관계성, 그리고 제약조건의 완전한 이해이다. 이 목표는 특정한 DBMS에 무관해야 달성할 수 있다. 왜냐하면 각 DBMS는 통상 개념적 설계에 반영되지 않는 자기자신만의 특성과 제약조건을 가지고 있기 때문이다.

o 개념적 스키마는 데이터베이스 내용을 변함없이 안정되게 기술하므로 매우 귀중하다. DBMS의 선택과 그 밖의 결정요소들은 개념적 스키마를 변경시킴이 없이 변경가능하다.

o 개념적 스키마를 잘 이해하는 것은 데이터베이스 사용자와 응용 프로그램 개발자 모두에게 필수적이다. 상위 데이터 모델의 사용은 더 일반적이고 더 표현적이기 때문이다.

o 다이아그램을 이용한 개념적 스키마의 기술은 데이터베이스 사용자, 설계자, 또는 분석자간에 아주 좋은 의사소통 수단이 된다. 왜냐하면 상위 레벨 데이터 모델은 DBMS에 귀속되는 데이터 모델보다 더 이해하기 쉽고 정확한 의사소통을 제공하기 때문이다.

이러한 개념적 스키마 설계는 상위 레벨 데이터 모델(시맨틱 데이터 모델)을 사용하여 이루어지는데 이들은 다음과 같은 특징을 갖는 것이어야 한다.

o 표현성(Expressiveness)
데이터 타입, 관계성, 제약조건 간에 발생하는 차이를 잘 나타낼 수 있어야 한다.

o 단순성(Simplicity)
일반 사용자가 이해하고 사용하기 쉽도록 단순해야 한다.

o 다이아그램 표현(Diagrammatic representation)
개념적 스키마를 이해하기 쉽도록 다이아그램으로 표현할 수 있어야 한다.

o 정형성(Formality)
데이터 모델에서 사용되는 개념은 정확하고 명확하게 정의되어 데이터를 모호하지 않게 기술할 수 있어야 한다.

많은 데이터 모델들이 개념적 데이터베이스 설계에 제안되었다. 그중 대표적인 모델로서는 Functional Data Model, ER Model 등이 있다. 여기서는 일반적으로 데이터베이스 설계에 많이 쓰이고 있는 시맨틱 모델로서 ER 모델을다룬다.

● DBMS의 선택

DBMS의 선택은 기술적 요소, 구조적 요소, 경제적 요소등 여러 요소에 의해 영향을 받는다. 먼저 DBMS를 획득 운용하는 비용을 생각해보고 이어서 DBMS를 사용하는 이익을 살펴본다. 다음은 DBMS를 선택할때 고려해야할 요소들이다.

- 소프트웨어 획득비용
- 유지비용
- 하드웨어 획득비용
- 데이터베이스 구축 및 전환비용
- 인건비
- 교육비용
- 운용비용
- 회사의 서비스 수준

데이터베이스를 획득하는 이윤은 엄청나서 사용의 편리성, 데이터의 더 광범위한 이용, 더 빠른 액세스 등을 제공받을 수 있다. 또한 개발비용의 감소, 데이터 중복의 감소, 더 나은 보안등의 부수적인 효과도 얻을 수 있다. 그러나 이미 화일 시스템의 형태로 기개발되어 사용중인 시스템이 있을때에는 이를 데이터베이스 시스템으로 전환하는 것을 신중히 고려하여야 한다. 최근들어 기존의 화일 시스템으로부터 데이터베이스 시스템으로 전환하기를 원하는 요구가 계속 증가하고 있다. 특히 비용의 관점에서 볼때 화일 시스템으로부터 DBMS로의 전환은 더욱 가속화될 것이다. 그 이유는 다음과 같은 요소에 근거해서 판단할 수 있다.

o 데이터 복잡성
데이터의 관계성이 더욱 복잡해지기 때문에 DBMS의 필요성이 더욱 대두된다.

o 응용 프로그램 간의 공유
DBMS를 사용하면 응용 프로그램간의 데이터를 공유하게 되므로 화일의 잉여를 줄일 수 있다.

○ 데이터의 다이내믹한 확장성

데이터가 계속 변한다면 이는 화일 시스템보다는 DBMS를 이용하여 해결해야 한다.

○ 불규칙적인 데이터 요구빈도

화일 시스템은 근본적으로 불규칙적이고 즉각적인 검색에는 맞지 않는다.

○ 데이터 양 및 제어대책 : 대량의 데이터와 이를 통제하는 방편으로서 DBMS가 훨씬 더 효율적이다.

● 논리적 설계(데이터 모델 매핑)

데이터베이스 설계의 다음 단계는 데이터 모델 매핑이라고도 하는 논리적 설계로서 선정된 사용 DBMS의 스키마를 생성하는 일이다. 이는 이전 단계에서 만들어진 상위 레벨 데이터 모델로부터 선택된 DBMS의 데이터 모델로의 매핑에 의해 이루어진다. 이 매핑은 다음 두단계로 진행된다.

시스템과 무관한 매핑 : DBMS 구현에 적용될 특별한 특성이나 경우를 고려하지 않고 매핑한다.

특정 DBMS에 스키마 조정 : 특정 DBMS 구현에 따른 특징 및 제약조건등을 고려한다.

논리적 설계단계를 거치면 그 결과로 논리스키마, 즉 선택된 DBMS의 DDL 언어에 의한 문장들이 얻어진다. 그러나 실제로 많은 경우에 DDL 문장들은 특정한 물리적 설계 요소들을 내포하고 있어야 하므로 완전한 DDL 문장은 물리적 설계에 가서야 이루어진다.

● 물리적 설계

물리적 설계는 주어진 응용 프로램에 대한 성능 향상을 위해 데이터베이스 화일에 대한 특정한 저장 구조와 액세스 패스를 결정하는 작업이다. 각 DBMS는 화일 조직과 액세스 패스를 위해 여러가지 옵션을 제공한다. 예를 들면 인덱싱, 클러스터링, 해싱 등이다. 일단 DBMS가 선택되어지면 물리적 DB 설계과정은 그 DBMS에 의해 제공되는 옵션들 가운데서 데이터베이스 화일을 위한 가장 적합한 구조를 선택하는 것으로 압축된다. 다음은 효과적인 물리적 설계를 위한 고려요소들이다.

응답시간 : 데이터베이스 트랜잭션을 요청해서 응답을 얻기까지의 시간

메모리 효율성 : 데이터베이스 화일과 액세스 패스 구조에 대한 메모리의 총공간

트랜잭션 산출물 : 일정 단위시간동안 처리되는 트랜잭션의 평균 수

물리적 설계 단계의 경과는 데이터베이스 화일을 위한 저장구조와 액세스 패스에 대한 최초의 결정이다. 보통 데이터베이스 시스템이 구현된 이후 얼마동안 사용하면서 관찰하여 물리적 데이터베이스 설계를 조정하는 것이 필요하다.

● 데이터베이스 시스템 구현

논리적 설계와 물리적 설계가 끝난후에는 데이터베이스 시스템을 구현할 수 있다. 선택된 DBMS의 데이터 정의언어(DDL) 문장과 저장매체 정의언어(SDL) 문장들은 컴파일되며 데이터베이스 스키마와 데이터베이스 화일(현재는 비어있음)을 생성해내는데 사용된다.

이제 데이터베이스에는 데이터가 들어갈 수 있게 되었다. 만일 데이터들이 이미 다른 형태로 저장되어 있다면 변환 루틴을 이용해서 새로운 데이터베이스 포맷에 맞게 조정하면 된다. 이때 응용 프로그래머에 의해 데이터베이스 트랜잭션도 구현되어야 한다. 내장된 DML 명령이 있는 응용 프로그램도 테스트되어야 한다. 이와 같이 트랜잭션이 준비되고 데이터베이스내에 데이터가 적재되면 설계 및 구현 단계는 완료되고 이제부터는 데이터베이스 시스템 운용단계가 시작되는 것이다.