

데이터베이스 시스템 기술

李 潤 俊

(正 會 員)

韓國科學技術院 電算科 教授

I. 서 론

현대를 정보의 홍수 시대라고 할 만큼 우리는 많은 양의 정보 속에 살고 있다. 이러한 정보를 얼마나 잘 관리하는가 하는 것이 기업의 성패에 큰 영향을 미치고 있다. 이러한 많은 양의 정보를 정확히 수집하고, 빠르게 분석하며, 효율적으로 관리하기 위해 컴퓨터의 화일 시스템(file system)이 널리 이용되고 있다. 그러나 컴퓨터를 이용하는 범위가 점차 확대되고, 컴퓨터를 이용하여 처리하고자 원하는 자료의 양이 점점 방대해짐에 따라 기존의 화일 시스템으로는 여러가지 문제를 낳게 되었다. 먼저 컴퓨터를 잘 이해하지 못하는 일반 업무자가 컴퓨터를 이용해야 하며, 비슷한 업무를 하는 부서간에는 같은 자료를 중복해서 갖게 되므로 저장 공간이 낭비되고 자료간의 불일치성이 발생할 수 있다. 또 기존의 화일 시스템은 프로그램과 데이터가 밀접히 연결되어 있기 때문에 프로그램이 변화되면 이전의 자료를 계속 사용할 수 없는 경우가 발생한다.¹⁾

이러한 문제를 해결하고, 방대한 양의 자료를 보다 효율적으로 관리하기 위해 데이터베이스 관리 시스템(DBMS)이 개발되었다. 데이터베이스 관리 시스템이란 기업에서 사용할 자료를 여러 응용 프로그램들이 공유할 수 있도록 불필요한 중복을 피해 데이터를 저장하고, 그 데이터를 필요로 하는 응용 프로그램이나 사용자들에게 보다 쉽고 효율적으로 제공하기 위한 시스템이다. 즉 그림 1과 같이 각 응용 프로그램이 원하는 데이터가 있을 때 그 자료를 각각의 화일에 저장하지 않고 그림 2와 같이 디스크내에 공통적으로 저장하고 각 응용 프로그램에 자료를 제공하기 위한 시스템이다.²⁾ 자료의 관리를 DBMS에게 맡기므로 각 사용자들은 보다 편리하게 자료를 관리하고 각자의 응용 프로그램을 개발할 수 있을 것이다.

본 고에서는 화일 시스템을 사용하지 않고 데이터베

이스 시스템을 사용했을 때 얻을 수 있는 장점을 조사하였으며, 그러한 장점을 가능케하는 DBMS의 구조에 대해 살펴 보았다. 또한 DBMS는 주로 사무용(business)에 사용되었으나 최근 들어서 이 시스템을 이용하는 새로운 응용과 DBMS가 추구하는 새로운 기술들을 소개하여, 국내에서 좀 더 많은 곳에서 데이터베이스를 이용하며 또 그에 대한 연구를 촉진시키고자 하였다.

II. 데이터베이스 시스템을 사용하는 이유와 장점

컴퓨터를 이용해서 자료를 처리하고자 하는 모든 응용 프로그램들은 그 처리를 위해 입출력 데이터를 가진다. 다루어야 할 자료의 양이 방대할수록 이들을 관

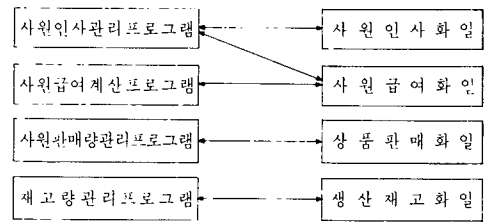


그림 1. 각 응용들과 그 데이터

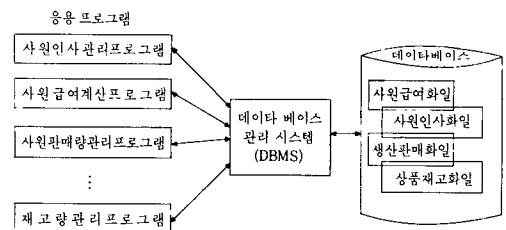


그림 2. DBMS를 이용하는 응용 프로그램

리하는 방법이 전체 프로그램의 성능에 큰 영향을 미치며, 효율적으로 데이터를 관리하는 것이 점점 어려워진다. 이러한 응용들이 다루게 될 데이터들을 데이터베이스 관리 시스템을 이용해서 처리할 경우 다음과 같은 장점들이 있다.^{1,9,10)}

1. 데이터의 공유

한 기업체에 소속된 각 부서의 업무는 비슷한 데이터를 이용하는 것이 보통이다. 이 경우 각 부서에서 자신들이 필요로 하는 데이터를 따로 따로 가질 경우 데이터들이 중복해서 저장된다. 이러한 중복은 기억장소의 낭비 뿐만 아니라 같은 자료가 여러 곳에 저장되어 있으므로 자료의 일관성을 잃기 쉽다. 또 중요한 자료를 여러 곳에서 사용할 경우 그 자료의 보안 문제도 보장하기 힘들어진다. 그러나 그림 2에서 보는 바와 같이 모든 응용 프로그램들이 사용할 데이터들을 중복을 제거하여 저장하고 그 데이터를 데이터베이스 시스템이 맡아서 관리하므로써 기억장소의 효율을 높이고 데이터의 일관성과 안정성을 높일 수 있다. 나아가 자료가 중앙에서 통제되므로 데이터 사용에 대한 표준화를 이룰 수 있다.

2. 코드의 공유

모든 응용 프로그램들은 자신이 사용할 입출력에 대한 코드를 갖는데, 이러한 입출력을 위해 코드를 작성하는 노력이 전체 프로그램 작성에 드는 노력에 상당한 부분을 차지한다.¹⁶⁾ 특히 이러한 입출력은 컴퓨터 시스템에 밀접하게 종속되므로 일반 사용자들이 이해하기 힘들고 프로그램을 작성하기도 힘들다. 다루어야 할 데이터의 양이 클 경우 프로그램의 성능을 고려해서 데이터를 저장할 구조를 결정해야 하며 그의 적절한 인덱스 구조도 함께 작성하여야 한다. 하지만 디스크에 저장된 자료의 관리와 접근을 담당하는 데이터베이스 시스템을 이용하면 각 응용 프로그램들은 이 시스템이 제공하는 데이터 접근 루틴을 공유하게 되어 응용 프로그램 작성이 용이해지며 데이터의 관리도 훨씬 효율적이 된다.

3. 데이터의 독립성

데이터베이스 시스템이 제공하는 또 하나의 장점은 저장된 데이터의 양식이 바뀌거나 그것을 접근하는 방법이 바뀌어도 그 데이터를 이용하던 응용 프로그램들이 바뀌지 않아도 되는 장점이 있다. 일반적으로 각 응용의 요구사항들은 변화하기 마련이고 따라서 프로그램이나 데이터의 양식이 바뀌는 것이 보통이다. 하지

만 데이터베이스 시스템을 이용할 경우 데이터의 양식이 바뀌어도 프로그램이 변화될 필요가 없고, 프로그램의 내용이 변화되어도 데이터가 따라서 바뀔 필요가 없으므로 프로그램 수정이 용이하고 프로그램의 유지 및 보수가 용이하다.

4. 사용하기 간편한 언어 제공

데이터베이스 시스템이 제공하는 데이터를 컴퓨터를 잘 알지 못하는 사람들도 쉽게 사용할 수 있도록 이해하기 쉬운 언어를 제공하고 있다. IBM에서 개발한 System R의 SQL 언어나¹⁷⁾ Berkeley 대학에서 개발한 INGRES의 QUEL 언어¹⁸⁾ 등이 그 대표적인 언어이다. 이러한 간단한 언어에 의해 데이터가 관리되고 이용된다.

5. 그외

데이터베이스 시스템을 이용하므로써 각 응용 프로그램들을 하나의 시스템으로 구축할 수 있을 뿐만 아니라 데이터베이스 시스템이 제공하는 여러가지 유용한 기능을 이용할 수 있다. 여러 응용이 같은 자료를 동시에 접근하는 것을 조정하기 위한 병행수행 제어기능, 데이터의 저장매체에 이상이 발생했거나 자료의 접근이 비정상적으로 끝났을 경우 이를 정상적인 상태로 회복시키기 위한 기능, 자료를 최적으로 접근하기 위한 기능등의 다양한 기능을 이용할 수 있는 장점이 있다.

위에서 열거한 이러한 장점들로 인해서, 다루어야 할 자료의 양이 많아질수록, 업무가 복잡하고 자주 변화할수록 데이터베이스 시스템을 사용하는 것이 유리하고 응용 프로그램을 개발하는데 드는 비용을 줄일 수 있다. 이러한 장점에도 불구하고 국내에서는 아직 데이터베이스 시스템을 사용하는 곳이 드물다. 그 이유는 먼저 데이터베이스 시스템의 가격이 상당히 비싸므로 처음 구입하는 비용이 크고, 이제까지의 프로그램들이 화일 시스템을 이용해서 개발되었기 때문에 이전 프로그램을 계속 사용해야 하고, 데이터베이스 시스템에 익숙치 않아 교육기간이 필요한 등등이 있다. 그러나 프로그램을 처음 개발할 때의 비용뿐 아니라 그것을 수정하고 유지보수하는 과정, 전체 소프트웨어 생명주기(life cycle) 동안의 가격을 고려한다면 데이터베이스 시스템을 사용하는 것이 더 효과적일 것이다.¹⁶⁾ 따라서 새롭게 개발될 소프트웨어들은 데이터베이스 시스템을 이용하여 개발하는 노력을 기울여야 할 것이다.

Ⅲ. 데이터베이스 시스템의 구조

앞 절에서 데이터베이스를 이용하여 얻을 수 있는 장

점을 살펴보았다. 이 장에서는 DBMS가 어떻게 구성 되었으며 앞에서 설명한 장점이 어떻게 구현되어지나 를 간단히 살펴보겠다. 1978년 ANSI/SPARC(american national standard institute / standard planning and requirements committee)에서 발표한 DBMS 보고서에 의하면,¹¹⁾ DBMS는 아래 그림 3 과 같은 골격 으로 구성된다고 볼 수 있다.

각 사용자나 응용 프로그래머들은 전체 데이터베이스 중에서 일부분만 사용하며, 이 각자가 데이터베이스 를 보는 관점을 외부 스키마 또는 뷰(view)라고 한다. 개념적 스키마는 각 응용 프로그램이 요구하는 데이터 들을 종합한 기업 전체의 데이터베이스 구조를 말하며 데이터베이스 접근 권한, 보안정책, 무결성 규정등을 시행하는데 필요한 조건들을 기술하고 있다. 내부 스키마는 개념 스키마의 물리적 저장 구조에 대한 정의 를 말한다. 내부 스키마는 실제로 저장될 내부 레코드 형식을 정의하며, 인덱스 사용, 저장 데이터 항목의 표현방법, 내부 레코드의 물리적인 순서를 기술한다.

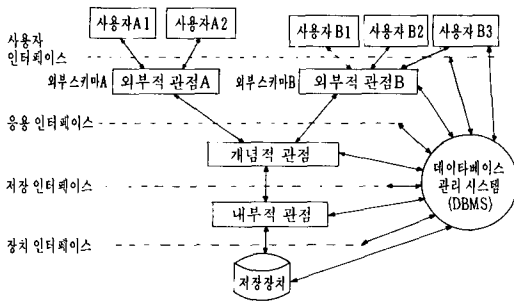


그림 3. 3 단계 데이터베이스의 논리적 구조

데이터베이스 관리 시스템은 데이터베이스의 세단계를 관리하기 위한 인터페이스를 제공한다. 즉, 사용자 인터페이스를 통하여, 사용자들이 각자의 관점으로 데이터베이스를 사용하면 이를 응용 인터페이스(외부-개념 사상)를 통하여 개념적 데이터베이스를 접근하도록 변화시킨다. 개념적 데이터베이스에 대한 접근은 다시 저장 인터페이스(개념/내부 사상)를 통하여 내부 데이터베이스의 접근으로 사상(mapping)되고 이는 장치 인터페이스를 통하여 실제 저장 장치를 접근하여 저장된 자료를 얻는다. 얻어진 자료는 역순으로 사상되어 사용자에게 자료가 전달된다. 데이터베이스 관리 시스템을 더 잘 이해하기 위하여 각 인터페이스를 살펴보기로 하자.¹²⁾

1. 사용자 인터페이스

DBMS를 사용하는 일반 사용자들은 이 인터페이스를 통하여 데이터베이스에 접근한다. 이 인터페이스는 이해하기 쉬운 데이터베이스 언어를 제공하여 사용자들이 데이터베이스 구조를 정의하고 데이터를 입력하며 검색하도록 한다. 뿐만 아니라 그래픽 시스템, 보고서 작성기, 그리고 데이터베이스 설계 도구들에 대한 인터페이스도 제공하여 사용자로 하여금 데이터의 관리를 쉽게 할 수 있도록 도와준다. 일반 사용자들이 DBMS를 사용하는데 알아야 할 지식은 이 사용자 인터페이스에 대한 지식뿐이다.

2. 응용 인터페이스

사용자 인터페이스를 통하여 들어온 데이터베이스 요구는 각 사용자나 응용 프로그래머의 외부 관점에 대한 요구이다. 이를 개념적 데이터베이스의 참조, 삭제, 그리고 생성으로 변환시키는 것이 응용 인터페이스의 역할이다. 이때 고려해야 할 사항은 데이터 사용에 대한 권한 검사, 데이터의 무결성 검사등이 있다. 이 인터페이스에서는 외부-개념사상을 위하여 각종 정보를 담고 있는 데이터 사전(data dictionary)를 관리하는 것이 필요하다. 만약 개념적 데이터베이스의 구조가 변화될 경우 이전의 데이터베이스를 이용하던 응용 프로그램에 대한 외부 관점을 새롭게 변화된 개념적 데이터베이스로 사상시킬 경우 외부 관점과 그 외부 관점을 이용하는 응용 프로그램이 변화될 필요가 없는 장점이 있다. 이를 논리적 데이터 독립성(logical data independence)라 한다.

3. 저장 인터페이스

개념적 데이터베이스에 대한 접근은 실제 저장된 데이터를 접근하기 위해 내부적 데이터베이스의 접근으로 사상된다. 이 단계는 실제 저장장치를 접근하기 위한 방법과 순서, 인덱스의 사용 등에 대한 계획을 생성하는 단계로서 가장 구현하기 까다로운 인터페이스이다. 이 단계에서는 사용자가 요구한 데이터를 가장 효율적으로 접근하도록 사상하기 위한 최적화 기능과 여러 사용자가 같은 데이터를 접근하므로써 생기는 각종 문제를 해결하기 위한 병행수행 제어의 기능이 요구된다. 이 단계의 사상에, 의해, 실제 데이터가 저장된 레코드의 양식(format)이 변화하거나, 화일의 구조나 접근 방법이 변화하여도 개념적 데이터베이스 구조가 변화할 필요가 없고 따라서 응용 프로그램이 바뀔 필요가 없는 장점이 있다. 이를 물리적 데이터 독립성(physical data independence)라 한다.

4. 장치 인터페이스

앞에서 생성된 장치 접근 계획을 실제 저장장치의 기능과 운영체제(operating system)를 이용하여 자료를 접근하고 운영 및 관리하는 인터페이스이다. DBMS의 기능을 수행하기 위해 프로세서 자원의 관리, 데이터베이스 응용 프로그램과 그것의 제어를 위한 루틴에 대한 기억장치 할당의 제어, 데이터베이스 접근에서 생기는 입출력과 캐쉬 또는 버퍼관리등의 기능을 수행한다.¹¹¹ 그리고 보조 기억 장치의 관리, 화일 관리 서비스, 그외 여러 운영체제 서비스를 운영체제로 제공받아 실제 데이터를 저장하고 관리하는 인터페이스이다. 따라서 사용하고자 원하는 컴퓨터와 그 운영체제에 종속된 부분이다.

IV. 데이터베이스 시스템의 추세

이제까지 데이터베이스를 사용하여 응용 소프트웨어를 개발할 경우 얻을 수 있는 장점과 그 데이터베이스 관리 시스템의 구성에 대해서 살펴보았다. 앞에서 보았듯이 데이터베이스는 많은 양의 정보를 다루는 응용 프로그램의 개발을 용이하게 하는 장점이 있어 해외뿐만 아니라 국내에서도 이미 여러곳에서 DBMS를 사용하고 있다. 그러나 이제까지의 DBMS는 간단한 형태의 자료를 다루는 과학 계산 응용이나 사무 응용에 주로 이용되어 왔고, 복잡한 형태의 자료를 다루는 응용에는 부적합한 면을 가지고 있었다.¹¹⁵ 하지만 최근 들어서는 이러한 여러 응용들도 DBMS를 사용하여 처리할 수 있도록 기능을 확장하는 여러 연구들이 활발히 진행되고 있고 이러한 기능을 이용하여 응용 시스템을 개발하는 노력들이 활발히 진행되고 있다. 예를 들면 복잡한 형태의 CAD의 설계 결과들을 저장하고 관리하는 응용과 길이가 긴 문서들을 관리하는 응용이 있다. 또 데이터베이스를 사용하는 환경의 변화로 인하여 DBMS도 그에 맞춰 변화하는 추세에 있다.

1. CAD/CAM 응용

DBMS는 자연수, 실수, 문자등의 간단한 형식의 자료만을 사용하는 응용들을 위하여 개발되어 왔다. 그러나 VLSI 설계등과 같은 응용들은 그러한 간단한 형식의 자료외에 몇 K bytes 또는 몇 M bytes 크기의 데이터를 다루며 데이터 자체도 계층적 구조를 갖는다. 또 데이터베이스의 스키마도 동적으로 변화하며, 설계를 위한 고차원적 그래픽 인터페이스를 필요로 한다. 이러한 새로운 요구사항을 갖는 CAD의 설계 결과를 저장하고 관리하기 위한 DBMS의 개발이 국내에서 활발히 시도되고 있다. 대표적인 예로 IBM에서는 기존

의 범용 DBMS인 System R을 이 응용에 맞도록 확장하였으며,¹¹² 미국의 MCC(microelectronics and computer technology corporation)에서는 이 응용에 적합한 새로운 시스템의 구현을 시도하고 있다.¹⁴⁵

2. 문서 관리 응용

이 응용분야는 길이가 길고 텍스트, 그림, 그리고 사진이 섞여 있는 문서를 저장하는 새로운 방법을 요구한다. 또 정확히 일치되는 내용만 검색하던 기존의 응용과는 달리 키워드(keyword)에 의한 검색, 부분적으로 일치되는 내용의 검색, 비슷한 내용의 검색을 요구한다. 이 응용에 적합한 DBMS를 개발함으로써 각종 문서의 처리와 서류처리를 효율적으로 할 수 있을 것이며 DBMS의 응용 범위가 더욱 넓어질 것이다. 이러한 응용에 적합한 시스템을 구현하기 위해, 캐나다에 있는 Toronto대학등에서 많은 연구가 진행되어 왔으며¹¹⁸ 우리나라에서도 산업연구원에서 해외 문헌 관리, 대법원의 판례관리, 그외 각 대학의 도서관에서 장서관리와 목록관리를 위해 이러한 응용의 시스템을 이용하는 시도가 진행되고 있다.¹¹⁶

3. 분산 데이터베이스 시스템

데이터베이스에 대한 새로운 응용들이 나타난 것 외에도 컴퓨터 기술이 발달됨에 따라 DBMS에 새로운 변화가 요구되고 있다. 컴퓨터의 이용이 증대되고 개인용 컴퓨터의 개발로 인해 모든 정보를 한 컴퓨터에 수록된 정보뿐만 아니라 컴퓨터끼리의 통신을 이용하여 다른 컴퓨터에 저장된 데이터들도 접근할 수 있는 것이 필요하다. 지역적으로 떨어져 있는 컴퓨터들의 데이터베이스를 한 컴퓨터에서 마치 자기 자신의 데이터베이스내에 있는 것처럼 사용할 수 있도록 하는 DBMS를 분산 데이터베이스 시스템(distributed DBMS)라 한다.¹⁶ 이 분산 DBMS에서는 컴퓨터끼리의 통신과 이에 소용되는 비용을 고려하여 자료를 접근하는 방법을 요구한다. 이러한 시스템에 의해 한 기업체나 한 국가는 여러 지역의 정보를 편리하고 일관성 있게 관리할 수 있다. 분산 데이터베이스는 IBM에서 개발한 R*나 Berkeley대학에서 개발한 분산 INGRES 등이 있다.

V. 결 론

컴퓨터를 이용하는 모든 응용 프로그램들은 입출력 자료를 가지며 이 입출력을 검색, 조작, 관리하는 루틴이 전체 프로그램의 상당한 부분을 차지한다. 이제까지는 대부분의 프로그램들이 화일 시스템을 이용하

여 이러한 데이터들을 관리하여 왔다. 이러한 방식의 경우, 한 기업체내의 정보가 중복해서 저장될 수 있으므로 기억장치의 낭비와 자료의 불일치성을 낳을 수 있다. 또 자료의 양이 많아질 경우 이를 효율적으로 관리하기 위해 인덱스등의 구조를 모든 응용 프로그램이 공통적으로 가져야 하므로 노력이 중복된다. 그러나 데이터베이스 관리 시스템을 이용하여 응용 시스템을 개발할 경우 이러한 문제들을 해결할 수 있다.

데이터베이스 시스템은 편리한 사용자 인터페이스의 제공과 논리적/물리적 데이터 독립성등의 다양한 기능을 제공하므로 이를 이용할 경우 소프트웨어의 개발이 용이하고 소프트웨어가 간단해지므로 유지 및 보수에 드는 비용을 줄일 수 있다. 그럼에도 불구하고 많은 사람들이 DBMS의 사용을 꺼리고 있다. 그 이유는 DBMS의 가격이 상당히 비싸고, 이제까지의 대부분의 시스템이 화일 시스템을 이용하여 개발되었고, DBMS의 지식을 새롭게 배워야 하는 부담이 있기 때문이다.

최근들어 소프트웨어의 요구가 폭증하고 있고, 소프트웨어의 개발외에 유지보수의 비용이 전체 소프트웨어 개발 비용의 50% 이상을 차지한다는 통계적 결과¹⁰를 볼 때 소프트웨어를 체계적으로 개발하는 것이 더 중요하다는 인식이 높아지고 있다. 따라서 전체 소프트웨어 생명주기에 걸친 비용을 고려할 때, DBMS를 사용하는 것이 더 효과적이며 폭증하는 소프트웨어의 요구를 만족시키는 더 좋은 방법이 될 것이다.

한편 최근 들어서 새롭게 요구되는 응용들을 해결하기 위해 DBMS의 기능이 증가되고 있다. 이러한 기능들을 사용하면 CAD/CAM, 사무자동화 등등의 응용 시스템 개발도 더욱 용이하게 될 것이다. 새로운 기능을 DBMS에 추가시키며 응용 시스템에 적용하는 방법과 더불어 앞으로 개발되어질 소프트웨어 시스템에 DBMS를 이용하는 노력이 더욱 많아져야 할 것이다.

參 考 文 獻

- [1] ANSI/X 3/SPARC Study Group, Database Management Systems, *Framework Report on Database Management Systems*, AFIPS Press, Montvale, NJ, 1978.
- [2] ANSI/X3/SPARC Database Systems Study Group, Database Architecture Framework Task Group, "Reference Model for DBMS standardization," ACM SIGMOD Record, pp. 19-58, March 1986.
- [3] Astrahan, M.M. et al., "System R: A Relational approach to Database Management," ACM Trans. on Database Systems, pp. 97-137, June 1976.
- [4] Banerjee, J. and Kim, W., "DL/CAD: A Data Language for CAD Objects," MCC Technical Report, April 1985.
- [5] Banerjee, J. and Kim, W., "Storage Structure for Evaluation Recursion in CAD Databases," MCC Technical Report DB-128-85, Nov. 1985.
- [6] Ceri, S. and Pelagatti, G., *Distributed Databases: Principles and Systems*, McGraw-Hill Book Company, 1984.
- [7] Chamberlin, D.D. et al., "SEQUEL 2: A Unified Approach to Data Definition, Manipulation, and Control," IBM J. Res. Dev., pp. 560-575, Nov. 1976.
- [8] Cristodoulakis, S. et al., "Development of a Multimedia Information System for an Office Environment," Proc. of VLDB, pp. 261-271, 1984.
- [9] Date, C.J., *An Introduction to Database Systems*, Third Edition, Addison-Wesley Publishing Company, 1982.
- [10] Fairley, R., *Software Engineering Concepts*, McGraw-Hill Book Company, 1985.
- [11] Gray, J.N., *Notes on Database Operating Systems*, RJ2188 (30001), 1978, IBM Research Laboratory, San Jose, CA.
- [12] Haskin, R.L. and Lorie, R.A., "On Extending the Functions of a Relational Database System," Proc. of SIGMOD, pp. 207-212, 1982.
- [13] Stonebreaker, M. et al., "The Design and Implementation of INGRES," ACM Trans. on Database System, pp. 189-222, Sept. 1976.
- [14] Taylor, R.W., "New Directions in Database Management," Proc. of The 1984 NYU Symp. on New Directions for Database System, pp. 5-11, 1984.
- [15] 한기준, "CAD 응용을 위한 설계 데이터베이스의 관리," 정보과학회지, pp. 3-10. 1985년 12월.
- [16] 박정란, 김경혜, 김은영, "DB산업, 정보화사회 지평을 연다." 경영과 컴퓨터, pp. 141-160. 1986년 3월. *