조사분석

DB2 데이터베이스 관리시스팀의 활용기술 분석

홍기채* 우동진**

목 차

* 정보유통개발실 선임기술원
** 정보유통개발실 실장

〈요 약〉

어떤 특정분야에서 데이터베이스 관리에 관한 한 관계형시스팀이 앞으로의 대안이라는 것은 의심할 여지가 없으며, 이와 같은 관계형 원리에 근거한 DB2 관계형 데이터베이스 시스팀은 매우 Cost-Effective하며 다양한 방법으로 어플리케이션 개발을 지원할 뿐 아니라 아주 많은 이용자들에게 다양한 서비스를 제공하기 위해서 상대적으로 소수의 데이터 처리 전문가만을 요구한다.

본 고에서는 이러한 DB2 관계형 데이터베이스 시스팀을 이용하여 특정 어플리케이션을 개발하고자 할 경우에 있어서 필요한 간단하고 기본적인 DB2의 소개, 구성, 기억장소 산출방식 및 기술동향에 대해 아주 일부분적인 내용만을 언급하였다.

Ⅰ. DB2의 소개

DB2는 IBM DataBase 2의 약어이며, IBM 컴퓨터에서 운영되는 MVS(Multiple Virtual Systems)라는 운영체제의 서브시스템인 관계형 데이터베이스 관리시스템이다. DB2 관계형 데이터베이스 관리시스템(이하 DB2라고 함)은 이미 잘 알려진 관계형언어인 SQL(Structured Query Language) 또는 QMF(Query Management Facility) 등의 프로덕트를 통해 많은 MVS 이용자가 DB2 데이터베이스를 액세스할 수 있도록 해주고 있다.

또한 DB2는 데이터베이스 관리시스템과 관련하여 IBM사가 오래전부터 계획하고 개발해 왔던 가장 성공적인 시스템으로 1983년 6월에 발표되었으며, IBM 3090계열 컴퓨터의 MVS 하에서 운영되고 있다.

현재 IBM사는 DB2를 SAA(System Application Architecture) 프로덕트 세트의 주 멤버로 생각하고 버전 향상을 위해 보완 등을 계속하고 있다. 여기서 SAA를 잠깐 소개하면 IBM이 일관성, 호환성 및 가장 손쉬운 방법으로 모든 IBM 컴퓨팅환경을 위해서 그리고 그 환경내에서 어플리케이션을 개발할 능력, 즉 "Cross-System Consistency"를 제공할 목적으로 한데 묶어 놓은 표준 인터페이스, 약정, 프로토콜의 집합으로 구성되어 있다.

SQL 언어는 SAA내에 공통의 데이터베이스 인터페이스 언어로 지정되었고 DB2는 MVS 환경에서 그러한 인터페이스 언어의 제공자가 되었다.

IBM사는 DB2를 이용할 수 있도록 보조역할을 해주는 QMF, CSP(Cross System Product : 어플리케이션을 만들어줌)와 같은 기존 프로덕트의 확장버전 등을 많이 개발하였으며, 이외에도 다른 프로덕트들의 개발을 계속하고 있다.

이러한 프로덕트들은 IBM의 관계형 프로덕트의 집합체로, QMF, CSP, SQL/DS(Structured Query Language/Data System), AS(Application System), DXT(Data Extract), DBRAD, ECF 등은 이 집합체의 멤버들이다.

이외에 다른 컴퓨터 회사들도 이와 같은 일들을 하고 있으며 DB2를 가지고 또는 DB2를 이용할 수 있게 특별히 개발된 새로운 프로덕트들(CDB Software Inc.의 Let's See DB2' 등)이 시장에 많이 나와 있으며, 기존의 많은 프로덕트(현재 DB2에서 실행할 수 있는 MUST Software International사의 NOMAD2 등)들도 여러 형태의 DB2 인터페이스를 제공할 수 있도록 확장되고 있는 중이다.

사실상, 모든 데이터베이스 시장은 이제는 "Gone Relational"이며, 관계형기술이 대부분의 용용분야에 적용될 수 있는 방법론이라는 것을 부정할 사람은 없다.

또한 관계형기술은 어플리케이션 개발 및 유지보수의 용이성, 생산성 향상 등 참으로 중요한 개념을 가지고 있다.

DB2같은 관계형 시스템은 컴퓨터시장을 지배하기 시작하였고 앞으로도 계속 그렇게 될 것이다. Oracle Corp., Relational Technology Inc. 등의 독립적인 DBMS 판매회사들도 컴퓨터 네트워크내에 한 지역에서 다른 지역에 DB2로 저장된 데이터를 액세스할 수 있는 그

들 자신의 소유 프로덕트인 분산 데이터베이스의 개발 가능성을 검토하고 있는 중이다.

현재 관계형 데이터베이스를 다루기 위한 공식적인 표준 인터페이스, 즉 ANSI(American National Standards Institute)/ISO(International Standard Organization) 등의 SQL 언어가 있으며, 이러한 표준은 DB2 내에서 실현된 IBM의 SQL 언어와 거의 비슷하다.

Ⅱ. DB2의 구성과 기능

DB2는 데이터베이스에 관한 일반적인 모든 기능을 제공하고 우수한 성능을 갖는 관계형 데이터베이스 관리시스템이다.

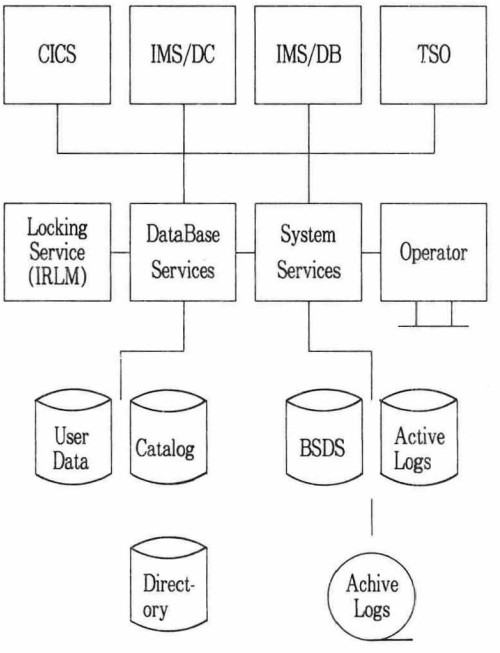
DB2의 내부구조는 (그림 1)과 같이 매우 복잡하며, 많은 프로시저들과 자료구조로 구성되어 있다. 예를 들어, 데이터베이스 언어를 제공하기 위한 기능의 프로시저 및 회복(Recovery) 제어, 권한(Authorization) 등을 수행하는 기능들의 프로시저 집합이다.

그러나 DB2는 이러한 많은 프로시저들을 포함하는 3가지 주 구성요소, 즉 시스템 운영, 오퍼레이터 통신, 로깅(Logging) 등을 지원하는 시스템 서비스, 데이터의 동시접근을 관리하는 로킹 서비스, 유저 및 시스템 데이터의 정의, 검색, 변경 등을 지원하는 데이터베이스 서비스 등으로 나뉘어진다.

1. 시스템 서비스

- 시스템 서비스(System Service)는 CICS, IMS/DC, TSO 등과 같은 MVS 서브 시스템과의 연관관계 제어
- 시스템 Start Up 및 Shutdown, 오퍼레이터 통신 등을 조작
- 시스템 오류시 시스템과 데이터베이스를 복구하기 위한 정보를 기록하기 위해 이용되는 미리 정의된 디스크 데이터세트인 시스템 로그를 관리하며, 각각의 현재 사용중인 액티브 로그 데이터세트가 꽉 차면 DB2는 새로운 로그데이터세트로 교체하며 이전의 로그 데이터세트는 디스크 또는 테이프의 아카이브 로그 데이터세트에 저장한다.

  현재 사용중인 로그 데이터세트와 아카이브 로그 데이터세트를 중복해서 저장하는 것은 로그자체에 에러가 발생할 경우 DB2가 로그데이터를 복구



(그림 1) DB2 Structure

할 수 있도록 하기 위해서이다.

모든 로그데이터세트 정보는 BSDS (Boot Strap Dataset)로 알려진 이중화된 시스템 데이터세트에 저장되어지며, BSDS를 출력해보고 유지할 수 있는 유틸리티가 제공된다.

- 시스템 전체적인 통계, 성능, Accounting 정보를 수집하고, 이러한 정보는 DB2 실행자에 의하여 모아지고 MVS 시스템관리자 또는 범용화된 추적자 데이터세트에 쓰여진다.

  DB2 PM(Performance Monitor) 프로덕트는 이와 같은 데이터세트로부터 배치 보고서 및 인터랙티브한 그래픽 등을 만들어준다.

2. 로킹 서비스

로킹서비스(Locking Service)는 IRLM(IMS Resource Lock Manager)라고 불리는 MVS 서브시스템에 의해 제공된다. 이러한 IRLM은 범용의 Lock 관리자로서 시스템내에 IMS가 있든지 없든지 관계없이 데이터의 동시액세스를 제어하기 위해서 DB2에 의해 이용된다.

3. 데이터베이스 서비스

데이터베이스 서비스 요소의 주목적은 정의, 검색 그리고 데이터베이스 데이터의 변경 즉 SQL 언어의 기능을 수행하는 것을 지원하는 것이다.

이러한 지원은 다음과 같은 5개의 부요소에 의하여 제공되며, 응용 프로그램의 준비, 실행을 허용해준다.

- 프리컴파일러

프리컴파일러(PreCompiler)는 호스트 프로그래밍 언어(PL/I, COBOL 등)의 프리프로세서이다. 이것의 기능은 호스트언어 소스 모듈을 분석하고 모든 SQL 문을 찾아서 호스트언어 Call 문으로 대치하며, 이 SQL문에 대해 바인드 부요소의 입력이 되는 DBRM(Database Request Module)을 구성한다(런타임시에 이러한 Call 문들은 간접적으로 런타임 수퍼바이저에게 제어가 넘어간다).

- 바인드

바인드(Bind) 부요소의 기능은 응용 플랜(Plan)을 만들기 위해서 하나 이상의 DBRM들을 컴파일한다. 응용 플랜은 이미 만들어진 DBRM으로부터 원래 SQL문의 컴파일형태를 이루고 있는 내부 제어구조의 집합을 포함한다. 특별히 바인드는 데이터관리 구성요소라고 한다.

- 런타임 수퍼바이저

런타임 수퍼바이저(Runtime Supervisor)는 응용프로그램이 실행될 때 주기억장소에 상주하며 실행을 관리한다. 응용프로그램이 수행되어져야 하는 어떤 데이터베이스 오퍼레이션을 요구할 때, 데이터관리자에게 적절한 오퍼레이션을 요구하기 위해 응용플랜내의 제어정보를 이용하는 런타임 수퍼바이저에게 먼저 제어가 넘어간다.

- 데이터관리자

데이터관리자(Data Manager)는 매우 정교한 액세스 방법으로 탐색, 검색, 변경, 인덱스

관리 등의 모든 정상적인 액세스 기능을 수행한다. 좀더 넓은 의미로 말하자면 데이터관리자는 물리적 데이터베이스를 관리하는 요소이다.

또한 로킹(Locking), 로깅(Logging), 물리적 I/O 오퍼레이션 등과 같은 복잡한 기능을 수행하기 위해 필요에 따라 다른 시스템 구성요소를 호출한다.

- 버퍼관리자

버퍼관리자(Buffer Manager)는 외부 매개체와 가상 기억장소사이에 물리적인 데이터전송(I/O 오퍼레이션)에 대해 처리하는 구성요소이다.

이것은 실질적으로 수행되는 물리적인 I/O양을 최소화하고, 버퍼풀(Buffer Pool)로부터 최상의 성능을 얻기 위해서 Read-Ahead 버퍼링 및 Look-Aside 버퍼링같은 복잡한 기술을 이용한다.

- 시스템테이블

데이터베이스 서비스 구성요소는 시스템데이터 테이블을 관리한다. 이러한 테이블들은 어떤 제어와 유저의 데이터테이블과 그들의 칼럼들, 데이터베이스 백업, 오퍼레이션, 데이터베이스 인덱스 등의 디스크립터 정보를 포함한다.

시스템테이블은 카탈로그와 디렉토리의 2개 그룹으로 나누어진다.

유저의 관점에서 볼 때 카탈로그는 데이터베이스 관리자가 이용할 수 있는 정보를 만들어내는 SQL 데이터 검색문에 의해 액세스 가능하며, 디렉토리는 SQL 문으로 액세스가 되지 않으며 단순히 DB2 자신의 내부적인 이용을 위해서 존재한다.

- 유틸리티

데이터베이스 서비스 구성요소는 또한 데이터베이스 로딩 및 데이터베이스 재구성과 같은 그러한 기능을 수행하기 위한 유틸리티들의 집합을 포함한다.

Ⅲ. DB2를 이용한 활용기술

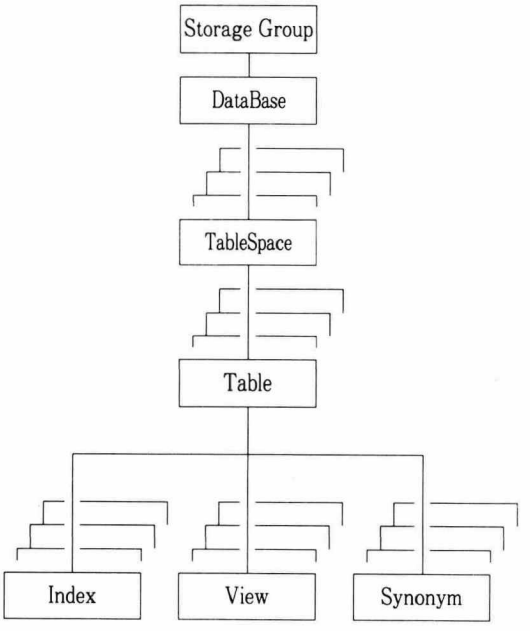
DB2 관계형 데이터베이스를 이용하여 실제적으로 응용분야별로 데이터베이스를 구축하고 운영하는 데 있어서 필요한 기억장소 구조 및 요구되는 용량(Space)과 몇가지 응용에 관한 내용을 알아보기로 한다.

1. 기억장소 구조

DB2는 (그림 2)와 같이 Storage Group, DataBase, TableSpace, Table, Index, View, Synonym이라고 불리는 계층적인 관계를 가진 7개의 오브젝트(Objects)를 정의하며, 이러한 오브젝트 관계는 SQL Create문이 실행될 때 DB2 카탈로그 테이블에 기록된다.

가. Storage Group

Storage Group은 모두 같은 형태의 디바이스인 직접액세스 가능한 디스크의 집합체로, 각 Storage Group에 대해 용량과 분할은 VSAM(Virtual Storage Access Method) ESDS (Entry-Sequenced Datasets)를 이용하여 저장된다. DB2는 디스크관리, 데이터세트 카탈로깅, 페이지(Pages)와 주기억장소 사이의 물리



(그림 2) DB2 오브젝트들의 계층적인 관계도

적 데이터 전송 등을 위해 VSAM을 이용한다. 그러나 페이지내에 용량관리는 VSAM이 아닌 DB2가 직접한다.

나. DataBases

DB2 내에 데이터베이스는 테이블(Tables)과 인덱스(Indexes)를 포함하는 TableSpace, IndexSpace 등의 논리적으로 관련된 오브젝트들의 집합체이다. 또한 데이터베이스는 콘솔 오퍼레이터가 Start/Stop 명령어로 데이터베이스의 사용 또는 사용불가를 처리할 수 있는 의미에서 데이터베이스의 사용시작/끝의 한 단위로, 물리적으로 존재하는 오브젝트가 아닌 개념상의 오브젝트라고 볼 수 있으며 정의할 수 있는 데이터베이스의 수는 제한이 없다.

다. TableSpace

TableSpace는 하나 이상의 테이블을 갖기 위해서 이용되는 보조 기억장치(디스크)에 존재하는 논리적인 어드레스 스페이스다(여기서 논리적이라는 의미는 물리적으로 반드시 인접되지 않아도 되는 기억장소의 집합이란 뜻임). Table의 크기가 커짐에 따라 TableSpace의 기억장소는 적절한 크기의 용량이 필요하며, 최고 64G바이트까지 확장가능하며, 데이터베이스내에서 TableSpace의 수는 제한이 없다.

또한 TableSpace는 같은 크기의 4K 또는 32K 바이트의 페이지(Page)들로 구성되며, Simple TableSpace, 분할 TableSpace, 세그먼트 TableSpace로 나뉘어지며 각각의 TableSpace는 다음과 같은 특성을 가지고 있다.

- Simple TableSpace

분할이나 세그먼트 TableSpace가 아닌 TableSpace를 Simple TableSpace라고 하며, 서로간에 밀접한 관계를 가지는 테이블들의 경우에 바람직한 TableSpace로 하나 이상의 테이블을 가질 수가 있다.

그러나 하나 이상의 테이블을 가질 경우 한 테이블의 액세스만을 할 경우에도 모든 TableSpace가 액세스되어지므로 성능이 줄어들 뿐 아니라 페이지 로킹(Locking)을 할 경우에도 다른 테이블까지 로킹이 되므로 한 테이블만의 로킹을 할 수가 없는 단점도 있다.

- 세그먼트 TableSpace
  - 하나 이상의 테이블을 저장할 수 있으며, 테이블 액세스시 해당

테이블에 관련되는 페이지만이 액세스되며, 비어 있는 페이지는 액세스가 되지 않는다.

- 테이블 로킹시 다른 테이블에 해당하는 세그먼트는 로킹이 되지 않는다.
- 테이블을 삭제하면 이에 관련되는 세그먼트는 즉시 이용가능하게 되므로 TableSpace의 재구성이 필요없게 된다.
- 테이블내의 대량의 레코드 삭제시 매우 빠르며 작은 양의 로그 정보를 기록한다.
- 스페이스 맵(Space Map)이 페이지내에 어느 정도의 여분이 있는지의 정보를 갖고 있기 때문에 새로운 레코드를 추가시 불필요한 Read 액세스를 피할 수 있다.

이상과 같이 세그먼트 TableSpace는 전체적으로 많은 장점과 보다 빠른 성능을 기대할 수가 있으므로 분할 TableSpace와 함께 권고되는 TableSpace이다. 그러나 4K 바이트로 구성되는 세그먼트의 수를 적절하게 적용하지 않으면 기억장소 낭비의 결과를 초래할 수도 있다.

- 분할 TableSpace
  - TableSpace를 64개까지 여러 개의 Partition으로 분할할 수 있으며, 한 테이블만을 저장한다.
  - 매우 큰 테이블을 여러 개의 Storage Group 또는 데이터세트에 저장.
  - 반드시 클러스터 인덱스를 가져야 하며, 매우 큰 테이블일 경우에 권고되는 TableSpace임.

라. Table

- 레코드의 집합으로 구성되며, 릴레이션이라고도 함.
- 한 레코드는 반드시 한 페이지내에 포함되어야 하나 한 테이블은 여러 개의 페이지를 가질 수가 있다.
- 가변장 레코드를 가질 경우 고정장 필드의 뒤에 가변장 필드를 놓는 것이 성능을 높일 수 있다는 점에서 권고되어진다.
- 4K 바이트 페이지내에 저장할 수 있는 레코드의 길이는 오버헤드를 제외한 4056 바이트를 초과해서는 안된다.

마. Indexes, IndexSpaces

- 인덱스와 IndexSpaces는 같은 의미로 볼 수 있으며, 둘 사이의 관계는 1 : 1이다. IndexSpaces에 대한 데이터 정의 명령은 없으나 인덱스 생성시 파라미터로서 지정되며 인덱스가 생성되면 자동적으로 Indexspaces도 만들어진다.
- IndexSpaces내에 페이지는 TableSpace내의 페이지와 마찬가지로 4K 바이트의 크기를 갖는다.
- DB2의 인덱스는 트리가 항상 균형된 특성을 갖고 많은 레벨, 트리구

조 인덱스인 B-트리 구조로 되어 있다.

바. View

뷰(View)는 가상의 테이블로 물리적으로 기억장소를 갖지 않으며 다음과 같은 경우에 생성한다.

- 특정한 이용자에게 원 테이블에서 약간 변형된 테이블의 내용을 제공하기 원할 때.
- 한 테이블내에 존재하는 모든 칼럼(애트리뷰트)의 내용을 어떤 이용자가 액세스하지 못하도록 할 경우 등에 뷰(View)를 생성한다.
- 응용프로그램에 영향을 주지 않고 테이블의 변경을 허용할 때 등이다.

사. Synonym

- 테이블 또는 뷰의 대치명

2. 기억장소 용량

여기서는 불필요한 기억장소의 용량을 최소화시키고 최적의 성능을 얻기 위해 DB2에 의해 요구되는 기억장소 용량 산출방법에 대해 살펴보고자 한다.

가. DB2 설치시 필요한 디스크 용량

우선 응용 영역 크기별로 DB2를 설치하기 위한 디스크 용량을 살펴보면 〈표 1〉과 같다.

- 〈표 1〉의 3330, 3350, 3380은 DASD 디바이스 모델이며, Small Site는 100개의 플랜(Plan), 50개의 데이터베이스, 500개 정도의 테이블, Medium Site는 200개의 플랜, 200개의 데이터베이스, 2000개 정도의 테이블, Large Site는 400개의 플랜, 400개의 데이터베이스, 4000개 정도의 테이블을 가진다는 가정에 근거한 용량이다.

  그러나 DB2의 설치 또는 이전시에 생성되는 Image Copies, Archive Logs, 임시 데이터세트 등은 포함되지 않았다.
- 또한 디바이스 모델에 따른 Size는 〈표 2〉와 같다.

〈표 1〉 DB2 설치시 필요한 디스크 용량

(단위 : 실린더수)

Dev. Type / Site Size	3330	3350	3380
Small	1086	528	438
Medium	1775	854	700
Large	11637	5539	4452

〈표 2〉 디바이스 모델별 디스크 용량

(* : KBytes)

Device Type	3380	3375	3350	3340	3330
Track Size*	47476	35616	19069	8368	13030
Pages per Track	10	8	4	2	3
Factor Value	1.16	1.09	1.16	1.02	1.06

나. 각종 DB2 데이터세트 설치시 필요한 디스크 용량

DB2 설치시 실제적으로 디스크 용량을 차

지하는 각각의 데이터세트에 관한 정보를 가정된 응용 영역 크기별로 〈표 3〉에 나타내었다.

〈표 3〉 DB2 데이터세트별 디스크 용량

(단위 : MBytes)

Site Size	DB2 Libraries	DB2 Catalog	Directory	Active Logs	Bootstrap Data Sets	Temporary Database
Small	134	20	8	124	1	25
Medium	134	73	18	248	1	25
Large	134	145	36	2465	1	390

- DB2 Libraries는 어떤 Site에 관계없이 고정된 용량을 가지게 되며 Active Logs와 카탈로그는 Site의 크기에 따라 크게 증가한다.

다. TableSpace 생성시 필요한 스페이스 산출방식

- Row(레코드)의 길이
  테이블 내의 각 칼럼 길이의 합계 +Null을 허용하는 칼럼의 수+가변장을 갖는 컬럼의 수*2+6 (Header)
- 페이지당 Row의 수(한 페이지당 127개의 Row를 초과할 수 없음)
  Floor(페이지 크기-22-(PctFree *페이지 크기)/(Row 길이+2 (Id Map)))
- 총 페이지 수
  - FreePage가 없는 경우
    Ceiling(총 Row 수/페이지당 Row 수)+2(Space Maps+Header)
  - FreePage가 하나 이상인 경우
    Floor((Ceiling(총 Row 수/페이지당 Row 수)+2(Space Maps+Header)) * ((FreePage + 1) / FreePage))
- TableSpace 용량(단위 : KBytes)
  총 페이지 수*4

예) 다음과 같은 조건으로 하나의 테이블을 포함하는 TableSpace를 예로 요구되는 기억장소 용량을 산출해 보겠다.

조건) 레코드의 수=100,000건
평균 레코드 크기=80바이트
페이지 크기=4K
PctFree=5(각 페이지당 5퍼센트를 남김)
FreePage=20(20페이지가 이용될 때마다 1페이지를 남김)

산출) 이용가능한 페이지 크기=4074*0.95=3870바이트
페이지당 레코드 수=Floor(3870/80)=48
이용된 페이지 수=2+Ceiling (100000/48)=2086

라. IndexSpace 생성시 필요한 스페이스 산출 방식

- 변수의 정의
  K : 인덱스키의 길이, 즉 키(Key)를 구성하는 모든 칼럼의 길이+키를 구성하는 칼럼중 Null을 허용하는 칼럼의 수
  S : 이용가능한 스페이스

((100−P)/100, 여기서 P는 PctFree의 값)

N : 인덱스내에 중복되는 키의 평균 수 (Unique 인덱스인 경우는 1임)

M : 페이지당 서브페이지의 수

- 페이지당 Entries 수
  - 페이지당 하나의 서브페이지로 구성된 인덱스인 경우
    페이지당 Entries 수=
    Floor(S∗N∗4050/(K+(4∗N)))
  - 페이지당 하나이상의 서브페이지로 구성된 인덱스인 경우
    페이지당 Entries 수=
    Floor(S∗N∗(4067−M∗(K+21))/(K+(4∗N)))
- 리프 페이지의 수
  Ceiling(테이블의 총 Rows/페이지당 Entries 수)
- 2nd 레벨에서의 논리프 페이지 수
  Ceiling(리프 페이지 수/페이지당 Entries 수)
- 3rd 레벨에서의 논리프 페이지 수
  Ceiling(2nd 레벨에서의 논리프 페이지 수/페이지당 Entries 수)
- 모든 레벨에서의 총 인덱스 페이지 수
  리프 페이지 수+2nd 레벨에서의 논리프 페이지 수+3rd레벨에서의 논리프 페이지 수
- IndexSpace의 용량(단위 : KBytes)
  (모든 레벨에서의 총 인덱스 페이지 수)∗4

예) TableSpace와 마찬가지로 다음과 같은 인덱스 구성시 소요되는 기억장소 용량을 산출해 보겠다.

조건) Unique인덱스를 가짐
테이블은 100000개의 레코드(Row)를 가지고 있음
키는 Null이 아닌 10바이트의 문자형태로 하나의 칼럼만 포함
인덱스는 각 리프페이지에 대해서 하나의 서브페이지만을 가짐

산출) 키의 길이 (K)=10바이트
이용가능한 스페이스(S) : (100−10)/100=0.9
페이지당 서브페이지의 수(M)=1
인덱스내에 중복되는 키의 평균 수 (N)=1
페이지당 Entries수=Floor(4050∗0.9/(10+4))=260
리프 페이지의 수=Ceiling(100000/260)=385
2nd 레벨에서의 논리프 페이지 수=Ceiling(385/260)=2
3rd 레벨에서의 논리프 페이지 수=Ceiling(2/260)=1
모든 레벨에서의 총 인덱스 페이지 수=385+2+1=388
IndexSpace용량=4∗(388+2)=1560K

Ⅳ. DB2의 기술동향

DB2같은 관계형시스템은 외부구조와 내부

구조, 논리적인 레벨과 물리적인 레벨사이의 관계가 비선형 시스템보다는 훨씬 정교하다.

논리적인 레벨에서 이용성(Usability)을 강조한다. 즉, 시스템은 이용자의 생산성을 높이기 위해 간단한 데이터 구조, 그리고 그 구조를 오퍼레이터가 조작하기 쉽도록 해준다.

또한 물리적인 레벨에서 무관함(Freedom)을 강조한다. 즉, 2개의 레벨로 분리시키는 것은 설치시 이용자에게 전혀 영향을 주지 않기 위해서이다.

앞으로 DB2는 특별히 이 양쪽 레벨에서 몇가지의 중요한 개발을 해야한다고 생각한다. 물리적인 레벨에서 액세스 방법론(해싱, 포인터 체인 등) 등이 인덱스 기법의 대안으로 제공되어야 하며, 최적자(Optimizer)는 그러한 새로운 구조를 이용하기 위해서 강화되어야 한다(물론 그러한 새로운 방법론은 논리적인 레벨에서는 보이지 않는다).

논리적인 레벨에서 SQL언어가 실행되어져야 하는 외부 조인(Outer Join) 같은 기능을 직접 지원하기 위해서 확장되어야 하며, 특별히 무결성(Integrity), 참조 무결성(Referential Integrity) 등을 지원하기 위해서 향상되어야 한다.

이와같은 문제 등을 해결하기 위해 IBM은 DB2 시스템을 SAA 프로덕트의 주 멤버로 생각하고 버전 업 등을 계속하고 있으며, 현재 참조무결성, 그리고 제한적이나마 다른 Site에서 서로 상대방의 데이터베이스를 액세스할 수 있는 분산처리 기능 등은 DB2 버전 2.2에서 가능하도록 하였다.

이외에도 DB2 관련 툴등과 관련하여 최근에 여러 곳에서 몇몇의 상업적이고 경쟁적인 프로덕트화 경향이 있으며, 이미 나와있는 툴들도 있다. 이에 따라 DB2는 다음과 같이 시장에 나올 프로덕트들을 위해 표준화를 정립해야 한다.

- DB2 보안관리, 성능관리, 조율(Tuning), DB2 데이터베이스 설계 등을 지원하기 위한 프로덕트
- 벤더(Vendor)들 소유의 데이터베이스 시스템, 응용프로그램 저작자 등과의 인터페이스 관련 프로덕트

현재 관계형 원리에 근거하여 대학이나 연구기관 등에서 장기간의 개발을 목표로 다음과 같은 좋은 프로덕트들을 얻기 위해서 연구활동이 활발하다.

- 분산 데이터베이스 시스템
- 공유 데이터베이스 머신
- 시멘틱 모델링
- 다양한 데이터의 통합(예를 들면, 텍스트, 이미지 등)
- 전문가 데이터베이스 시스템
- 자연어를 포함하는 새로운 형태의 인터페이스
- 공학 및 과학 데이터베이스 시스템
- 기 타

이들중의 데이터베이스 머신, 자연어처리시스템 등 같은 몇몇은 현재도 이용 할 수 있는 시스템이 시장에 나와 있으나 위와같은 모든 토픽에 대해 앞으로도 연구개발이 계속되어야 할 것이다.

DB2 관계형 데이터베이스 시스템은 이와같이 관계형 원리에 근거하여 여러가지 목표를

가지고 연구개발해 나가는데 있어서 많은 영향을 미칠 것이며, 데이터베이스관리에 관한 한 관계형시스템이 앞으로의 대안이라는 것을 의심할 사람은 없을 것이다.

V. 결 론

관계형시스템의 가장 큰 이점은 한마디로 "단순성"이라고 할 수 있으며, 공식화된 이론적인 기초 즉 이용자들이 쉽게 이해하고 이용할 수 있도록 잘 정의된 관계형모델에 근거하고 있다.

관계형 데이터베이스에서 모든 데이터는 테이블의 레코드(Row)내에 칼럼(애트리뷰트)값으로 표현된다. 또한 여러가지 기능들, 데이터정의, 보안과 권한 제어 및 DBMS의 다른 특성들을 다루기 위해 소수의 오퍼레이터만이 필요하게 된다. 관계형 모델을 전적으로 지원하는 DBMS는 없다.

DB2는 아마도 복잡한 데이터베이스시스템의 관리자가 선택할 수 있는 가장 잘 정의된 관계형 DBMS가 아닐까 한다.

DB2는 가능한 한 설치하고 운영하기 쉽도록 개발되었다. 많은 유틸리티들이 호출되고 다른 작업과 함께 동시에 데이터정의를 할 수 있다.

또한 DB2는 아주 많은 이용자들에게 DB2 서비스를 제공하기 위해서 상대적으로 소수의 데이터처리 전문가만을 요구한다.

DB2는 매우 Cost-Effective하며, 다양한 방법으로 어플리케이션 개발을 지원한다. QMF 또는 CSP 같은 DB2 Front-End 프로덕트들은 어떤 어플리케이션들을 개발하는 데 반드시 필요한 것은 아니나 개발을 더욱 쉽게 하고 유지보수를 덜 요구하게 한다.

지금까지 살펴본 바와 같이 DB2는 다른 DBMS에 비해 단점도 없지는 않으나 여러가지 장점을 더 많이 갖고 있는 DBMS로 추측된다.

끝으로 본 고에서는 DB2 관계형 데이터베이스를 이용하여 특정 어플리케이션을 개발하고자 할 경우에 있어서 필요한 아주 간단하고 기본적인 DB2의 소개, 구성, 기억장소산출 및 기술동향에 대해 일부분적인 내용만을 서술하였음을 첨언한다.

참 고 문 헌

1. C.J. Date, Colin J. White, "A Guide to DB2 Second Edition" Addison Wesley, April, 1988.
2. "DB2 : IBM'S Database Strategy", Computer Technology Research Corp., August, 1991.
3. Gabrielle Wiorkowski, David Kull, "DB2 Design & Development Guide", Addikson-Wesley Publishing Company, Inc., 1988.
4. Rick F van der Lans, "Introduction to SQL", Addison-Wesley Publishing Company, Inc., 1988.
5. "Database Programming & Design", Computer Associates, Vol 2, October, 1989.~ July, 1992.
6. "Administration Guide, Vol Ⅰ, Vol Ⅱ, Vol

Ⅲ" SC26－4374－1, IBM, 1989.

7. "Application Programming and SQL Guide", SC26－4377－1, IBM, 1989.

8. "SQL Reference", SC26－4380－1, IBM, 1989.

9. "Command and Utility Reference", IBM, 1989.