

DATABASE 정보기술 발전방향

1. DBMS 기술발전 추세 (1960-2000년 대까지의 흐름)
1960년대 후반 DBMS의 본격적인 제품화 시도

- 2) -조작성과 유연성이 매우 뛰어나
-데이터의 표현에 행과 열로 구성된 2차원의 표를 이용
-H/W Resource를 비교적 많이 차지

1.1 계층형/네트워크 DBMS

1970년대부터 1980년대 중반까지 Mainframe DBMS시장을 주도

- 1) 데이터의 집합체는 트리모양 또는 네트워크 형태로 '미리 정해진 구조'를 가진 형태로 표현
- 2) -특정한 검색요구에 대해서 매우 빠른 처리 성능을 발휘
-H/W Resource를 비교적 적게 소요
-비정형의 검색요구에 대한 응용 프로그램 구현이 어려움
-데이터 항목이나 각 데이터간의 연관관계가 변화했을 경우 유지보수가 매우 어려움

1.2 관계형 DBMS

1980년 후반부터 점진적으로 관계형으로 이행

- 1) 데이터 표현에 특정 형태를 가지지 않고 사용자가 각 레코드간의 물리적인 배치를 의식하지 않고 자연스럽게 검색 가능

1.3 객체지향 DBMS

- 1) 데이터를 일정한 형태로 저장할 필요가 없어 자유롭게 데이터 구조를 변경, 기능 확장이 용이
- 2) -종래의 관계형 DBMS에서 처리하기 어려웠던 도형 데이터처리가 용이
-관계형보다 처리속도가 우수
-대량의 도형정보를 다루는 CAD/CAM 분야나 멀티미디어 DB를 구축하기 위한 도구로 주목

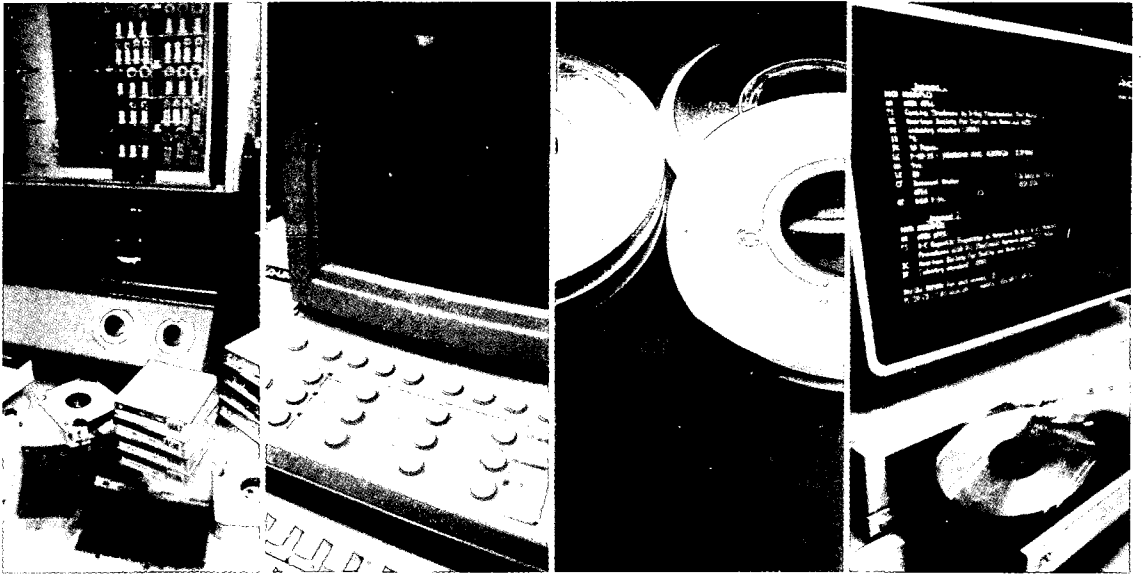
추 세

관계형 DBMS에 OODBMS의 특징을 받아들이는 형태로 진화시켜 가는 시도가 본격화됨

2. 90년대 DBMS 기술 추세

2.1 90년대 시스템구축의 새로운 조류

- 1) Downsizing의 확산
-고성능 W/S의 출현
Open System의 Unix를 탑재한 고성



능 W/S의 출현으로 고가의 Mainframe을 대체할 가능성 대두

-Network의 발전

강력한 Network기능을 가진 Unix의 확산으로 분산된 기종간에 연결이 용이해지고 Network기술은 더욱 발전

2) Open화

-이기종간의 연결 필요성 증가 및 Unix의 확산으로 Open화 급진전

-기존 Mainframe 업체도 Open System 지향 천명

2.2 Downsizing과 Open화 추진의 2가지 중요과제

1) 분산환경에서의 시스템구축 및 S/W 개발을 어떻게 전개할 것인가?

-Unix의 잇점을 살린 Business Application 구축에 명확한 방향성을 정

립하지 못한 사용자가 대다수.

이러한 상황에서 분산환경에서 작동하는 DBMS와 그것을 중심으로 하는 Application 개발환경은 매우 중요

이유 : DBMS가 Network Protocol, Unix System Call, GUI구축을 위한 프로그래밍 Interface 등의 고도기술을 사용자로부터 은폐해 줌

2) 기존 Mainframe과 PC를 Open System에 어떻게 연결시킬 것인가?

-모든 Transaction 처리나 데이터 관리기능을 Unix나 PC로 이행하는 것은 현실적이지 못함

Mainframe의 강력한 관리기능은 여전히 위력적, 따라서 기업전체의 데이터를 관리하는 Mainframe용 DBMS와 각 부문단위의 데이터를 관리하는 Unix용 DBMS가 협력 처리하는 시스템환경의 구축필요성 강조

DBMS의 기술도 Downsizing과 Open화 과제 해결에 맞춰 변화, 발전

2.3 DBMS 시장 판도 변화

- Mainframe용 DBMS의 성장세가 둔화되고 Unix용 DBMS가 급속도로 신장
- Unix용 DBMS 업체가 대거 등장, 기존 Unix용 DBMS 업체에 신규업체가 참여하고 또한 Mainframe용 DBMS업체가 Unix로 방향 전환

2.4 Downsizing과 Open화에 따른 DBMS 기술발전 추세

- Client/Server Architecture 적용
- 국제 표준 데이터베이스 언어 SQL 지원

1) Client/Server Architecture

- Network상의 복수의 분산처리시스템의 일종으로 User Interface부분과 DB 처리 부분을 나누어 처리하여 효율성을 높임.
- Unix와 Mainframe, PC와 Unix의 연결등에 효율적이므로 점차 DBMS의 표준적인 이용방법으로 정착

기술변화

① 초기

- Client - User Interface 부분
- Server - Presentation+Integrity Logic +Data Access부분

② 중기

- Client-User Interface+Prsentation+Integrity Logic
- Server-Data Access

③ 현재

- Client-User Interface+Presentation
- Server-Integrity Logic+Data Access

2) SQL 지원

- 국제 표준 데이터베이스 언어 SQL
1986. 11. ANSI에서 표준으로 채택
1987. 6. ISO에서 표준으로 채택
- 각 DBMS 업체들이 적극적으로 SQL을 지원하여 DBMS 측면에서의 Open화 촉진

기술변화

- ① 대화형 SQL에서 3GL과 함께 사용하는 Embedded SQL 추가
- ② Integrity Enhancement Feature 도입
- ③ Stored Procedure, Trigger 기능도입 검토

3)기타 주요기술

- ① 4GL - Mainframe DBMS에서 주력하던 부분으로 자사 DBMS의 효율적인 사용 및 Application개발 생산성을 높이기 위해 개발
- ② 의사결정지원 Tool, 질의어 - End User Computing의 확산으로 전산지식이 없는 사용자가 SQL지식 없이도 대화형식으로 원하는 데이터를 얻거나 그래프나 표를 간단히 화면에 표시 가능

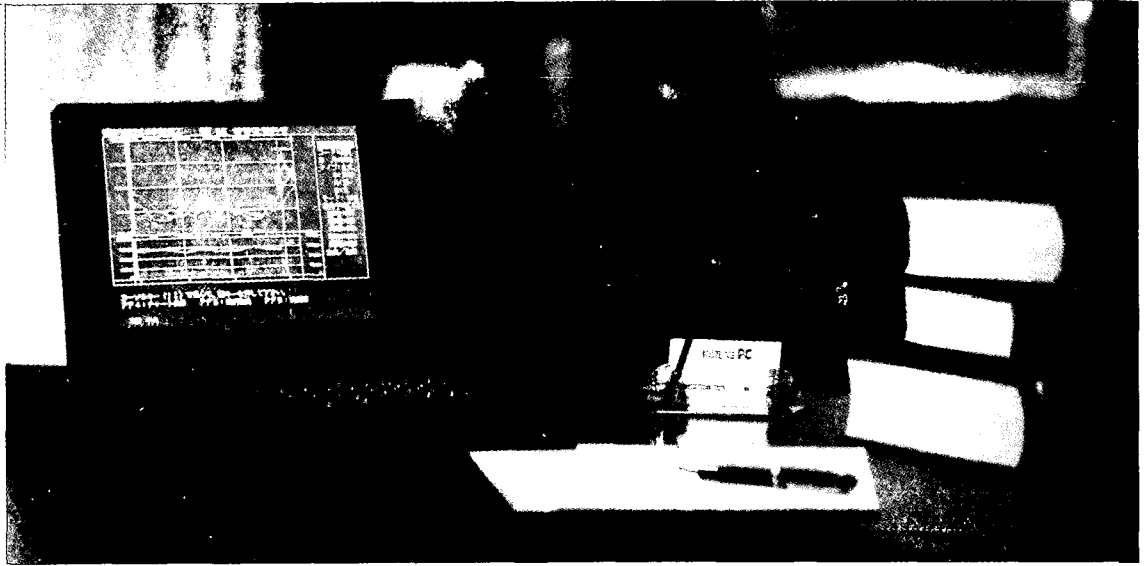
기술변화

Business Application개발에서 사용하던 4GL들이 Unix용으로 대거 전환하고 Client./Server의 확산에 따라 PC에서 사용가능한 4GL들이 등장

3. Unix 관계형 DBMS의 기술동향

3.1 분산처리기술

처리루틴을 분리해 물리적으로 각기 다



른 컴퓨터에서 수행

1) 분산액세스, 분산조회

사용자 측면에서 위치적으로 투명한 액세스가 가능하며, 하나의 SQL로 분산된 데이터를 조회 가능해야 함.

2) 분산트랜잭션

하나의 트랜잭션으로 분산된 데이터를 갱신가능해야 함

필요기술

- Location Transparency, Two-Phase Commit, Data Replication 등이 지원

3.2 분산데이터베이스 기술

물리적으로 떨어진 위치에 있는 DBMS 하의 데이터베이스 (논리적으로 하나의 데이터베이스로 취급)

1) 분산데이터베이스 표준화 추진

ISO등의 기구에서 RDA(Remote Data Access) 검토

2) 분산된 이기종간의 여러 DB간의 접속 중요

필요기술

- 다양한 Network Protocol을 지원하는 S/W 필요
- SQL의 차이를 변환하는 Gateway 필요

3.3 온라인 트랜잭션처리(OLTP) 표준화 추진

X/OPEN에서 분산 트랜잭션처리(Distributed Transaction Processing) 검토

필요기술

- 신뢰성 보장/장애대책
- 고성능 실현
· I/O 감소 (공유메모리 활용, 비동기 I/O, Group Commit 등)
· DBMS 향상(Stored Procedure, Trigger 채택)

3.4 PC Package와 DBMS Interface

현재 널리 사용중인 Spreadsheet, 통계 S/W, DTP, 4GL등과의 연결 활발 