

1. 데이터베이스의 등장 배경

컴퓨터가 과학·기술분야 뿐만 아니라 회계관리, 인사관리, 생산관리, 판매관리 등에 이용되기 시작하면서 애플리케이션 프로그램에서 데이터의 중복이 많고, 주어진 데이터 처리 위주로 하는 파일 처리 시스템에서

- 정보 검색과 복잡한 데이터 관리의 불편 해소
- 다양한 정보를 신속하게 수집하여 가공 저장하고, 쉽게 검색할 수 있는 방법의 필요성

에 대한 사회적 요구와

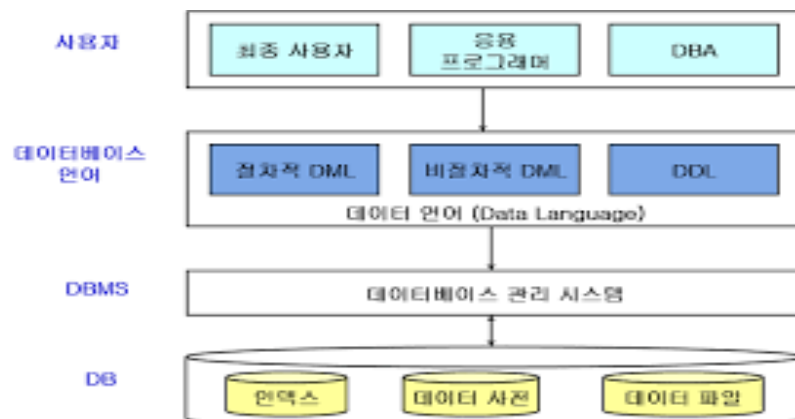
- 하드웨어의 소형화와 가격의 저렴화
- 신뢰성 향상
- 대용량의 기억장치와 자기 디스크의 실용화
- 고속의 연산장치
- 데이터통신 기술의 발달 등

컴퓨터의 하드웨어기술이 발달되면서, 데이터베이스라는 새로운 개념의 처리 방법이 탄생되었다.

2. 데이터베이스 시스템

데이터를 중앙에서 통제할 수 있게 하여 데이터의 중복이나 불일치성을 없애고, 데이터를 여러 사용자가 공동 이용하게 하며, 데이터에 접근하는데 비밀유지가 되게 하는 시스템을 일컫는다.

여기에는 데이터베이스가 저장되는 하드웨어, 데이터베이스와 데이터베이스 관리 시스템, 사용자, 데이터베이스 언어, 데이터베이스 관리자를 포함한다.



2.1. 데이터베이스 관리 시스템

‘사용자가 데이터베이스의 생성, 유지 및 사용을 제어하는 프로그램의 집합’, 즉 사용자와 데이터베이스를 연결시켜주는 소프트웨어 시스템을 말한다.

☞ 데이터 모델(Data Model)

데이터베이스를 설계하고 관리할 때 필요한 기법의 기초를 제공하기 위해 구성되는 것으로, 데이터와 데이터 간의 관계를 어떻게 표현하고 저장하는가에 따라 계층형 데이터 모델, 망형 데이터 모델, 관계형 데이터 모델, 객체지향형 데이터 모델로 구분한다.

데이터베이스관리시스템(DBMS)은 하나의 데이터 모델을 적용하여 소프트웨어를 개발한다.

2.2. 데이터베이스

‘어느 특정 기업체 조직의 응용 시스템에서 사용하기 위해 저장된 데이터의 집합’이다.

- 동일한 데이터들이 원칙적으로 중복되지 않는 통합된 데이터이다.
- 컴퓨터가 액세스하여 처리할 수 있는 저장 장치에 수록된 저장된 데이터이다.
- 그 기능을 수행하는데 반드시 유지해야 할 데이터베이스에 저장된 운영 데이터이다.
- 여러 응용 시스템들이 공동으로 소유하고 유지하며 이용하는 공용 데이터이다.

2.3. 데이터베이스 언어

DBMS와 사용자(User)간의 대화 언어이며, 데이터베이스를 정의하고 접근하기 위한 시스템관의 통신수단을 말한다. 데이터베이스 언어는 DBMS에 따라 사용언어가 다르며, 처리 기능에 따라 데이터 정의어, 데이터 조작어, 데이터 제어어로 구분한다.

2.4 사용자

데이터베이스 언어나 데이터베이스 개발도구 등으로 데이터베이스에 접근하는 최종 사용자와 언어 구사능력을 갖춘 응용 프로그래머, 개체(entity)나 관계들을 정의하고, 데이터베이스를 구축하는 데이터베이스 설계자(Database Designer), 시스템의 관리, 운영등을 총괄하는 데이터베이스 관리자가 있다.

3. 관계형 데이터베이스 시스템

3.1 관계형 데이터 모델

관계형 데이터 모델의 원칙은 관계형 데이터베이스를 이루는 요소로 릴레이션(relation), 속성(attribute), 도메인(domain), 관계 연산자, 논리데이터 독립성, 참조 무결성과 데이터 독립성, 특수 쿼리(query)의 세 가지 특성을 가진 데이터 저장 시스템으로 이런 관계형 데이터 모델을 적용한 관계형 데이터베이스 관리 시스템이 출현하면서 이해하기 쉽고, 명확하게 데이터 조작을 할 수 있게 되었다.

3.2 관계형 데이터베이스 시스템

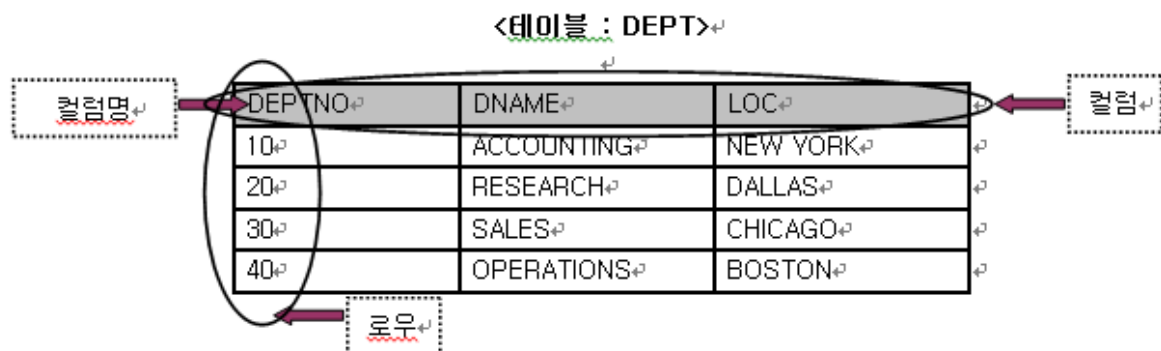
관계형 데이터베이스 관리 시스템(RDBMS), 관계형 데이터베이스(RDB), 관계형 데이터베이스 언어, 사용자, 데이터베이스 관리자를 말하며, 현재 운용되고 있는 대부분의 데이터베이스 시스템 환경이다.

☞ 관계형 데이터베이스 관리 시스템

많은 상업용 관계형 데이터베이스 관리 시스템(RDBMS) 제품들이 있는데 이 제품들은 범용(enterprise), 소규모용(workgroup), PC용으로 구분되며, 대표적인 범용의 제품으로 Oracle[오라클], SQL Server[Microsoft] 등이 있으며, PC용 제품으로 MS Access[Microsoft], PC Oracle[오라클] 등이 있다. 특히, MySQL, mSQL 등은 공개 소프트웨어로 웹(web)상에서 무료로 다운로드 받아 사용할 수 있는 RDBMS 제품들이다.

☞ 관계형 데이터베이스

관계형 데이터베이스 관리 시스템(RDBMS)에서 생성된 데이터베이스를 관계형 데이터베이스(RDB)라 부른다. 관계형 데이터베이스는 사용자에게 논리적인 2차원 테이블(행, 열)의 데이터베이스 구조로 표현한다. 따라서 정보모델링의 모든 엔티티(entity)와 관계(relationship)들은 모두 동일하게 테이블(table)로 표현한다.



<관계형 데이터베이스 특징>

- ✓ 2차 행렬의 테이블 집합으로 표현
- ✓ 각 테이블은 행(row)과 열(column)로 구성
- ✓ 테이블에서 각 행은 유일하게 구별
- ✓ 각 컬럼들은 데이터 무결성(data integrity) 보장
- ✓ 사용자는 테이블에 대한 데이터 정의, 데이터 조작 및 데이터 검색을 SQL을 이용하여 수행

☞ 관계형 데이터베이스 언어

RDBMS 소프트웨어 시스템과 사용자 간의 대화 언어를 관계형 데이터베이스 언어라 한다. 데이터베이스 언어로는 SQL, PL/SQL 등의 텍스트 중심 언어와 Oracle Develop, Delphi, PowerBuilder, Visual Basic 등의 GUI(Graphical User Interface) 데이터베이스 개발 도구들이 있다.

4. 관계형 데이터 모델 이론

정보의 표현은 개념적 데이터로 개체(entity), 속성(attribute), 인스턴스(instance), 관계(relationship)들을 ER Diagram으로 표현하고, 관계형 모델에 의해 표현되는 논리적 스키마(schema)는 테이블(table)이나 뷰(view), 인덱스(index)등으로 변환된다.

☞ 관계형 데이터베이스의 이론적인 개념

- ✓ 모든 개체(entity)는 개체를 설명해 주는 속성(attribute)의 집합을 갖는다.
- ✓ 행(row)은 속성 값 중 하나의 인스턴스(instance)이다.
- ✓ 개체의 속성 중 어떤 속성은 개체의 각 행을 유일하게 식별한다.
- ✓ 기본 키(primary key)를 구성하는 속성은 널(null)이 될 수 없다.
- ✓ 개체는 서로 관계(relationship)가 있다.
- ✓ 개체의 행은 순서가 임의적(random)이다.
- ✓ 개체의 속성은 순서가 임의적(random)이다.

4.1 관계형 데이터베이스

관계형 데이터베이스에서는 논리적으로 모든 데이터를 2차원 테이블로 표현한다. 2차원 테이블의 논리적인 구조로 데이터를 표현하기 때문에 누구나 이해하기가 쉽다.

☞ 테이블

RDMBS의 기본적인 데이터의 저장구조로, 1개 이상의 칼럼과 0개 이상의 행으로 구성된다. 각 칼럼은 테이블에서 단일 종류의 데이터를 나타내며, 이름, 성별, 사원번호, 급여액 등과 같은 속성(attribute)이다. 행은 하나 이상의 칼럼으로 구성되는 집합의 정보로 레코드(record)와 같은 하나의 인스턴스(instance)이다.

☞ 칼럼

테이블 상에서 단일 종류의 데이터를 나타내며, 각 칼럼은 데이터타입(datatype)과 크기(size)를 갖고 있다. 칼럼 순서는 데이터를 저장할 때는 무의미하지만, 조회할 때는 칼럼의 순서를 지정할 수 있다. 각 칼럼명은 한 테이블 내에서 유일하며, 각 테이블명은 한 스키마(schema) 내에서 유일하다.

☞ 행

칼럼들의 값의 조합이며, 레코드라고도 부른다. 각 행은 기본 키(primary key)에 의해 구분되고, 기본 키는 중복된 값을 허용하지 않으며, 반드시 데이터를 입력되어야 하는 필수(not null) 칼럼이다. 행의 순서는 중요하지 않으며, 기본적으로 데이터는 입력되는 순서대로 저장된다.

☞ 키

테이블에는 여러 개의 칼럼이 있다. 테이블의 칼럼들 중에서 특별한 의미를 갖는 칼럼을 키(key)라한다. 관계형 데이터베이스에는 기본 키와 후보 키, 복합키, 외부 키가 있다.

✓ **기본 키** : 한 테이블의 각각의 행을 유일하게 식별해 주는 한 칼럼 또는 칼럼의 조합을 말한다. 중복된 값이나 널(null)값을 허용하지 않는다 기본 키는 테이블에 대하여 단 하나의 개만 지정할 수 있다.

✓ **후보 키** : 테이블의 칼럼들 중에서 기본 키가 될 수 있는 칼럼을 후보 키 또는 보조키라 부른다. 후보 키의 칼럼 중에서 기본 키가 되고, 기본 키로 지정되지 않은 후보 키를 대체키라고도 부른다.

✓ **복합 키** : 테이블의 칼럼들 중에서 한 개의 칼럼으로 행을 식별하지 못할 때, 여러 개의 칼럼을 조합하여 식별하는데 이런 키를 복합 키라 부른다.

✓ **외부 키** : 업무의 성격이나 데이터의 중복을 최소화하기 위해서 데이터를 여러 개의 테이블로 분리하여 저장한다. 분리된 테이블들은 관계(relationship)를 통해 연결한다. 외부 키(foreign key)는 외래 키라고도 부르며, 같은 테이블 또는 다른 테이블의 기본 키를 참조하는 단일 칼럼 또는 칼럼의 조합으로 구성되며, 테이블과 테이블 간의 관계를 정의한다. 외부 키를 정의함으로써 관계형 데이터베이스 설계 규칙을 따를 수 있다. 외부 키는 논리적인 값에 근거를 한 것이지만 물리적으로 존재하는 포인터와는 다르다. 외부 키가 가질 수 있는 값은 참조하는 테이블의 기본 키 값이나 널 값만을 허용한다. 각 테이블에 대하여 여러 개의 외부 키를 지정할 수 있다.

🔗 관계

개체와 개체 간의 연결을 정의하는 것으로, 한 개체가 부모가 되고, 다른 개체가 자식으로 식별된다. 관계는 필수와 선택, 크기에 대한 특징이 있다.



- ✓ 필수 관계 : 자식 개체의 행이 반드시 부모 개체의 행을 가져야 하는 경우
- ✓ 선택 관계 : 자식 개체의 행이 부모 개체의 행을 가지지 않아도 되는 경우
- ✓ 관계 크기 : 부모 개체의 단일 행이 가질 수 있는 자식 개체의 행의 수

4.2. 관계형 데이터 베이스의 객체

객체의 종류	내 용
테이블	행과 열의 조합으로 구성된 데이터베이스의 기본 저장 단위
뷰	하나 또는 그 이상의 테이블로부터 논리적으로 데이터를 추출한 부분 집합으로, 논리적이고 가상적인 테이블
시퀀스	자동으로 고유한 숫자 값을 생성해 주며, 주로 기본 키 값을 생성하기 위해 사용
인덱스	쿼리(query) 속도를 향상시키기 위해 사용
시노님	객체에 대한 또 다른 이름으로 별명(alias) 부여
프로그램 유닛	SQL문과 PL/SQL문으로 작성. 프로시저(procedure), 함수(function), 패키지(package), 트리거(trigger)

5. SQL은 무엇인가?

사용자가 데이터를 정의하고, 데이터를 조작하기 위해 접근하는 비절차적 데이터베이스 언어이다. C, JAVA, JSP, ASP 등의 절차적인 프로그래밍 언어들은 '어떻게 처리할 것인가?'에 중점을 두고 프로그램을 작성하여 실행하며, 각종 프로그램 작성 기법에 관한 알고리즘 중심으로 작성한다. 그러나 SQL은 '무엇을 할 것인가?'에 중점을 두고 처리하는 4세대의 비절차적인 데이터베이스 언어이

다.

SQL은 구조화된 질의어(Structured Query Language)라고 한다. 데이터 정의어(DDL)와 데이터 조작어(DML)를 포함한 데이터베이스용 질의 언어(query language)의 일종이다.

SQL은 관계 사상(relation mapping)을 기초로 한 대표적인 언어로서 입력 릴레이션(테이블)로부터 원하는 출력 릴레이션을 매핑시키는 언어이다.

또한 SQL은 단순한 질의 기능뿐만 아니라 완전한 데이터 정의 기능과 조작 기능을 갖추고 있다. 또 온라인 단말기를 통해 대화식으로 사용할 수도 있고, COBOL이나 PL/I, C 등의 호스트 언어로 된 프로그램에 삽입되어서 사용되기도 한다. 특히 웹 기반의 애플리케이션에 사용되는 자바(JAVA), JSP, ASP, XML, PHP 등과 C#, ASP.NET의 웹 프로그래밍 언어에서 관계형 데이터베이스를 접근하기 위하여 SQL은 필수적으로 반드시 마스터해야 할 데이터베이스 언어이다.

SQL은 장치 독립적이고 액세스 경로에 대해서는 어떠한 참조도 하지 않으며, 개개의 레코드보다는 레코드의 집합인 테이블은 단위로 연산을 수행한다. 또한 SQL은 영어 문장과 비슷한 구문을 갖고 있으므로 초보자들도 비교적 쉽게 사용할 수 있다.

<SQL문의 입력 및 실행>

- 모든 SQL문은 세미콜론(;)으로 끝난다.
- SQL의 명령문은 대소문자를 구분하지 않는다. SQL의 예약어는 대소문자를 구분하지 않으나, 사용자 정의어(객체명, 칼럼명)와 데이터는 대소문자를 구분하여 사용할 수 있다.
- 하나의 SQL문은 명료하고 한 줄 또는 여러 줄로 나누어 입력할 수 있다.

6. DBMS의 종류

Oracle	MySQL	MSSQL	MariaDB
<ul style="list-style-type: none">• 오라클에서 만들어 판매중인 상업용 데이터베이스• 윈도우, 리눅스, 유닉스 등 다양한 운영체제(OS)에서 설치 가능• MySQL, MSSQL보다 대량의 데이터 처리 용이	<ul style="list-style-type: none">• MySQL사에서 개발, 썬마이크로시스템즈를 거쳐 현재 오라클에 인수합병• 윈도우, 리눅스, 유닉스 등 다양한 운영체제(OS)에서 설치 가능• 오픈소스로 이루어져 있는 무료 프로그램(상	<ul style="list-style-type: none">• 마이크로소프트(MS)사에서 개발한 상업용 데이터베이스• 다른 운영체제에서도 사용가능하지만 윈도우에 특화됨• 비공개 소스로 폐쇄적인 운영(리눅스 버전은 오픈 소스)	<ul style="list-style-type: none">• MySQL이 오라클에 인수합병된 후 불확실한 라이선스 문제를 해결하려고 나온 오픈소스 RDBMS• 구현 언어 : C++• MySQL과 동일한 소스 코드 기반• MySQL과 비교해 애

<ul style="list-style-type: none"> • 대기업에서 주로 사용하며, 글로벌 DB 시장 점유율 1위 • 비공개 소스, 폐쇄적인 운영 • 가장 널리 사용되는 RDBMS 	<ul style="list-style-type: none"> 업적 사용 시 비용 발생) • 가격 등의 장점을 앞세워 다수의 중소기업에서 사용 중 • RDBMS 	<ul style="list-style-type: none"> • RDBMS 	<ul style="list-style-type: none"> 플리케이션 부분 속도가 약 4~5천배 정도 빠름
--	--	---	--

7. RDBMS와 Nosql 차이점

RDBMS

1. 우선 SQL(Structured Query Language)은 데이터베이스에서 사용하는 쿼리 언어다. SQL을 사용하여 RDBMS에서 데이터를 검색, 저장, 수정, 삭제 등이 가능하다.

2. RDB(Relational Database)란 관계형 데이터 모델에 기초를 둔 데이터베이스다. 관계형 데이터 모델이란 데이터를 구성하는데 필요한 방법 중 하나로 모든 데이터를 2차원 테이블 형태로 표현해준다.

Nosql

1. 관계형 데이터베이스와 반대되는 방식을 사용하여 데이터간의 관계를 정의하지 않는다.

2. RDBMS에서는 스키마에 맞추어 데이터를 관리하여야 하지만 Nosql은 스키마가 없어 좀 더 자유롭게 데이터를 관리할 수 있다.

3. Nosql에서 테이블과 같은 개념으로 컬렉션이라는 형태로 데이터를 관리한다.

RDBMS 특징

1. Data를 Colum과 Row 형태로 저장한다.

2. 데이터의 분류, 정렬, 탐색 속도가 비교적 빠르다.

3. SQL(Structured Query Language, 구조화 질의어)라는 정교한 검색 query를 통해 데이터를 다룬다.

4. Transaction(작업의 완전성을 보장)

5. 반드시 Schema 규격에 맞춰야 한다.(유연한 데이터 저장 안됨)

6. 부하의 분산이 어렵다.

Nosql의 특징

1. 데이터간의 관계를 정의하지 않는다.(Table간의 join도 불가능)
2. RDBMS의 복잡도와 용량 한계를 극복하기 위한 목적으로 등장한 만큼 RDBMS에 비해 훨씬 더 대용량의 데이터를 저장할 수 있다.
3. 분산형 구조 : 데이터를 여러 대의 서버에 분산해 저장
4. 고정되지 않은 Table Schema(Schema가 없어 다루기 쉬움)
5. Key에 대한 put/get만 지원한다.
6. Schema가 없으니 data에 대한 규격화된 결과 값을 얻기 힘들다.
7. MongoDB, Cassand, Redis 등....