

1. 데이터베이스(DATABASE) 개요

1-1. 자료(Data)와 정보(Information)

“자료(Data)” 는 현실 세계에서 관찰이나 측정을 통해 수집한 단순한 사실이나 결과 값으로 가공되지 않은 상태를 말하며, “정보(Information)” 는 의사 결정에 도움을 줄 수 있는 유용한 형태로 자료를 가공(처리)해서 얻을 수 있는 결과이다.

1-2. 데이터베이스의 정의

데이터베이스(DATABASE, DB)는 체계화된 데이터의 모임이다. 특정 조직의 업무를 수행하는 데 필요한 상호 관련된 데이터들의 모임으로 여러 응용 시스템들의 통합된 정보들을 저장하여 운영할 수 있는 공용 데이터들의 묶음이다.

데이터의 구분

- 가. 통합된 데이터(Integrated Data): 자료의 중복을 배제한 데이터의 모임이다.
- 나. 저장된 데이터(Stored Data): 컴퓨터가 접근할 수 있는 저장 매체에 저장된 자료이다.
- 다. 운영 데이터(Operational Data): 조직의 고유한 업무를 수행하는 데 존재 가치가 확실하고 없어서는 안 될 반드시 필요한 자료이다.
- 라. 공용 데이터(Shared Data): 여러 응용 시스템들이 공동으로 소유하고 유지하는 자료이다.

1-3. 데이터베이스의 특징

- 가. 실시간 접근성(Real-Time Accessibility): 수시적이고 비정형적인 질의(조회)에 대하여 실시간 처리에 의한 응답이 가능해야 한다.
- 나. 계속적인 변화(Continuous Evolution): 데이터베이스의 상태는 동적이다. 즉 새로운 데이터의 “삽입(Insertion)” 하고, 불필요한 데이터는 “삭제(Deletion)” 하며, 변경된 데이터를 “갱신(Update)” 하여 항상 최신의 데이터를 유지해야 한다.
- 다. 동시 공유(Concurrent Sharing): 데이터베이스는 서로 다른 목적을 가진 여러 사용자들을 위한 것이므로 다수의 사용자가 동시에 같은 내용의 데이터를 이용할 수 있어야 한다.
- 라. 내용에 대한 참조(Content Reference): 데이터베이스에 있는 데이터를 참조할 때 데이터 레코드의 주소나 위치에 의해서가 아니라, 사용자가 요구하는 데이터 내용으로 데이터를 찾는다.

1-4. 데이터베이스 시스템의 구성요소

데이터베이스 시스템의 구성요소는 데이터베이스, 스키마, DBMS(데이터베이스 관리 시스템), 데이터베이스 언어, 데이터베이스 컴퓨터, 데이터베이스 사용자이다.

가. 스키마(Schema)

데이터베이스 구조와 제약 조건에 관한 전반적인 명세를 기술한 메타데이터의 집합이다.

나. DBMS(DataBase Management System)

사용자와 데이터베이스 사이에서 사용자의 요구에 따라 정보를 생성해 주고, 데이터베이스를 관리해 주는 소프트웨어이다.

다. 데이터베이스 언어(Database Language)

데이터베이스를 구축하고 이용하기 위한 데이터베이스 시스템과의 통신 수단이다.

- **데이터 정의 언어(DDL: Data Definition Language)** : DB 구조, 데이터 형식, 접근 방식 등 DB를 구축하거나 수정할 목적으로 사용하는 언어이다.
- **데이터 조작 언어(DML: Data Manipulation Language)** : 사용자로 하여금 데이터를 처리할 수 있게 하는 도구로서, 사용자(응용 프로그램)와 DBMS 간의 인터페이스를 제공한다. 응용 프로그램을 통하여 사용자가 DB의 데이터를 실질적으로 조작할 수 있도록 하기 위해 사용하는 언어이다.
- **데이터 제어 언어(DCL: Data Control Language)** : 무결성, 보안 및 권한 제어, 회복 등을 하기 위한 언어이다. 데이터를 보호하고 데이터를 관리하는 목적으로 사용된다.

2. 데이터베이스의 종류

데이터베이스는 계층형 데이터베이스 관리 시스템(HDBMS), 네트워크형 데이터베이스 관리 시스템(NDBMS), 관계형 데이터베이스 관리 시스템(RDBMS), 객체지향형 데이터베이스 관리 시스템(OODBMS), 객체관계형 데이터베이스 관리 시스템(ORDBMS)이 있다.

2-1. 계층형 데이터베이스 관리 시스템(HDBMS)

계층형 데이터 모델은 트리 구조를 이용해서 데이터의 상호관계를 계층적으로 정의한 구조이다. 데이터의 논리적 구조도가 트리 형태이며, 개체(레코드)가 트리를 구성하는 노드 역할을 한다. 개체 간의 관계를 부모와 자식 간의 관계로 표현한다.

장 점	단 점
구조가 간단하고, 판독이 용이하다. 구현, 수정, 검색이 용이하다. 데이터의 독립성이 보장된다.	데이터 상호 간의 유연성이 부족하다. 검색 경로가 한정되어 있다. 데이터 삽입과 삭제 연산이 매우 복잡하다.

2-2. 망(네트워크)형 데이터베이스 관리 시스템(NDBMS)

망형 데이터 모델은 레코드 타입과 링크들의 집합으로 구성되며, 상위 하나의 레코드에 대하여 하위의 레코드가 복수 대응하고, 하위 하나의 레코드에 대해 상위 레코드도 복수 대응한다. 데이터 모델링이 복잡하여 사용이 일반화되지 않았다.

2-3. 관계형 데이터베이스 관리 시스템(RDBMS)

가. 관계형 데이터베이스 관리 시스템

1970년 IBM에 근무하던 코드(E.F.Codd)에 의해 처음 제안되었으며, 일반적으로 데이터베이스 관리 시스템은 RDBMS(Realational Database Management System)를 사용한다. 관계형 데이터베이스 관리 시스템에는 Oracle, MySQL, MS-SQL, MariaDB 등이 있다. Oracle은 가장 유명한 유료 RDBMS로 미국의 대표적인 RDBMS 회사인 오라클의 데이터베이스 관리 시스템이다. 엔터프라이즈 DB 시장의 절반 이상을 항상 차지하고 있다. MySQL은 가장 많이 사용하는 무료 RDBMS이다. 커뮤니티 버전은 오픈소스이다. 오라클이 오픈소스에 호의적이지 않기 때문에 MySQL 사용자들 사이에서 불안감이 커졌고 그로 인해 대체재로 등장한 것이 바로 MariaDB이다. MariaDB는 MySQL이 Sun Microsystems로 넘어가면서 당시 MySQL AB 출신들이 따로 나와서 MySQL을 기반으로 한 오픈 소스 RDBMS를 만들었다. MySQL을 기반으로 만들었기 때문에 거의 완벽하게 호환된다.

장 점	단 점
간결하고 보기에 편리하다. 다른 데이터베이스로의 변환이 용이하다.	성능이 다소 떨어진다.

나. 관계형 데이터베이스의 Relation 구조

RDBMS는 관계형 데이터베이스를 구성하는 개체(Entity)나 관계(Relationship)를 모두 릴레이션(Relation)이라는 표(Table)로 표현한다.

릴레이션(Relation)

데이터들을 표(Table)의 형태로 표현한 것으로 구조를 나타내는 릴레이션 스키마와 실제 값들인 릴레이션 인스턴스로 구성된다.

튜플(Tuple)

- 릴레이션을 구성하는 각각의 행을 말한다.
- 파일 구조에서 레코드(행)과 같은 의미이다.

속성(Attribute)

- 개체의 특성을 기술하며, 데이터베이스를 구성하는 가장 작은 논리적 단위이다.
- 파일 구조에서 데이터 항목 또는 데이터 필드(열)에 해당된다.

도메인(Domain)

- 하나의 속성이 취할 수 있는 같은 타입의 원자(Atomic)값들의 집합이다.
- 파일 구조에서 레코드(행)과 같은 의미이다.

	속성	속성	속성	속성	속성	
	학번	이름	학년	신장	학과	→ 릴레이션 스키마
튜플 →	89001	홍길동	2	170	CD	
튜플 →	89002	이몽룡	1	169	CD	
튜플 →	87012	성춘향	2	180	ID	→ 릴레이션 인스턴스
튜플 →	86032	장보고	4	174	ED	

↑ 학년 도메인

〈학생〉 릴레이션

다. 릴레이션(Relation)의 특징

같은 릴레이션에는 똑같은 튜플이 포함될 수 없으므로 릴레이션에 포함된 튜플들은 모두 상이하다.

- ※ 〈학생〉 릴레이션을 구성하는 홍길동 튜플은 홍길동에 대한 학적 사항을 나타내는 것으로 〈학생〉 릴레이션 내에서는 유일하다.

같은 릴레이션에 포함된 튜플 사이에는 순서가 없다.

- ※ 〈학생〉 릴레이션에서 홍길동 튜플과 임꺽정 튜플의 위치가 바뀌어도 상관없다.

튜플들의 삽입, 삭제 등의 작업으로 인해 릴레이션은 시간에 따라 변한다.

릴레이션 스키마를 구성하는 속성들 간의 순서는 중요하지 않다.

속성의 유일한 식별을 위해 속성의 명칭은 유일해야 하지만, 속성 값은 동일한 값이 있을 수 있다.

릴레이션을 구성하는 튜플을 유일하게 식별하기 위해 속성들의 부분집합을 키(Key)로 설정한다.

- ※ 〈학생〉 릴레이션에서는 학번이나 이름 튜플들을 구분하는 유일한 값인 키가 될 수 있다.

속성의 값은 논리적으로 더 이상 쪼갤 수 없는 원자 값만을 저장한다.

라. 관계형 데이터베이스의 무결성(Integrity) 제약조건

무결성이란 데이터베이스에 저장된 데이터 값과 그것이 표현하는 현실 세계의 실제 값이 일치하는 정확성을 의미하며, 무결성 제약 조건이란 데이터베이스에 저장되는 데이터의 정확성(일관성)을 보장하기 위해 부정확한 자료가 데이터베이스 내에 저장되는 것을 방지하기 위한 제약조건을 말한다. 이를 위해 키를 이용하여 입력되는 데이터에 제한을 주는 것으로 개체 무결성 제약, 참조 무결성 제약 등이 해당된다.

개체 무결성 : 릴레이션에서 기본키를 구성하는 속성은 널(NULL) 값이나 중복 값을 가질 수 없다.

참조 무결성 : 외래키 값은 NULL이거나 참조 릴레이션의 기본키 값과 동일해야 하며, 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다.

- ※ **기본키** : 릴레이션에서 특정 튜플을 유일하게 구별할 수 있는 속성으로 NULL 값과 중복되는 값을 가질 수 없다.

- ※ **외래키** : 참조되는 릴레이션의 기본키와 대응되어 릴레이션 간에 참조 관계를 표현하는데 중요한 도구이다.

3. 데이터베이스 관리 시스템(DBMS)의 장단점

장 점	단 점
<p>① 데이터 중복의 최소화</p> <p>파일 시스템에서는 각 응용 프로그램마다 자신의 파일이 개별적으로 관리 유지되기 때문에 전체적으로 저장되는 데이터의 입장에서 보면 상당히 많은 데이터가 같은 내용을 표현 하면서 중복적으로 저장되고 있다. 그러나, 데이터베이스는 데이터를 통합하여 구성함으로써 이러한 중복을 사전에 통제할 수 있다. 물론 데이터의 중복을 완전히 배제한다는 것을 의미하는 것은 아니다. 왜냐하면 통합 데이터베이스 환경속에서도 성능 향상의 이유로 데이터의 중복이 불가피할 때가 있기 때문이다.</p> <p>② 데이터의 공유</p> <p>같은 내용의 데이터를 여러 가지 구조로 지원해 줄 수 있는 DBMS의 복잡하고도 정교한 기법 때문에 데이터베이스의 데이터 공유가 가능하다.</p> <p>③ 데이터의 일관성 유지</p> <p>데이터 변경 시 하나의 데이터만이 변경되고 다른 하나는 변경되지 않는다면 데이터간의 불일치성, 즉 모순성을 갖게 된다. 모순성을 내포한 데이터베이스는 서로 상충되는 정보를 제공하게 되고 데이터베이스의 유용성을 저해하게 된다. 데이터베이스는 관리 시스템은 바로 이 데이터의 중복을 제어하고 중앙 집중식 통제를 통해 데이터의 일관성을 유지할 수 있다.</p>	<p>① 운영비의 증대</p> <p>DBMS는 가격으로 보아 고가의 제품이고 컴퓨터 시스템의 지원을 많이 사용한다. 특히, 주기억장치를 많이 차지하기 때문에 DBMS를 운영하기 위해서는 메모리 용량이 더 필요하게 되고, 더 빠른 CPU를 요구하게 된다. 결과적으로 시스템 운영비의 오버헤드를 가중시키게 된다.</p> <p>② 자료 처리의 복잡화</p> <p>데이터 베이스에는 상이한 여러 타입의 데이터가 서로 관련되어 있다. 응용 프로그램은 이러한 상황속에서 여러 가지 제한점을 가지고 작성되고 수행될지도 모른다. 따라서, 응용 시스템은 설계 시간이 길어지게 되고 보다 전문적, 기술적이 되어야 하기 때문에 고급 프로그래머가 필요하게 된다.</p> <p>③ 복잡한 예비와 회복</p> <p>데이터베이스는 그 구조가 복잡하고 여러 사용자가 동시에 공유하기 때문에 문제가 일어났을 때 정확한 이유나 상태를 파악하기 어려울 뿐만 아니라 여기에 대한 예비(Backup)조치나 사후회복(Recovery)기법을 수립해 놓는 것도 매우 어렵다.</p> <p>④ 시스템의 취약성</p> <p>데이터베이스 시스템은 통합된 시스템이기 때문에 그 일부의 고장이 전체 시스템을 정지시켜 시스템 신뢰성과 가용성을 저해할 수 있다. 이것은 특히 데이터베이스에 의존도가 높은 환경에서는 아주 치명적인 약점이 아닐 수 없다.</p>

<p>④ 데이터의 무결성 유지</p> <p>무결성이란 데이터베이스에 저장된 데이터 값과 그것이 표현하는 실제 값이 일치하는 정확성을 말한다.</p> <p>데이터 중복성이 완전히 제거된다고 하더라도 허용되지 않는 값이나 부정확한 데이터가 여러 가지 경로에 의해 데이터베이스에 잠입될 수도 있다. 데이터베이스 관리 시스템은 데이터베이스가 생성 조작될 때마다 제어 기능을 통해 그 유효성을 검사함으로써 데이터의 무결성을 유지할 수 있다.</p> <p>⑤ 데이터의 보안 보장</p> <p>DBMS는 데이터베이스를 중앙 집중식으로 총괄, 관장함으로써 데이터베이스의 관리 및 접근을 효율적으로 통제할 수 있다. 이것은 DBMS가 정당한 사용자, 허용된 데이터와 연산등을 확인 검사함으로써 모든 데이터에 대해 철저한 보안을 제공한다.</p> <p>⑥ 표준화</p> <p>DBMS의 중앙 통제 기능을 통해 데이터의 기술양식, 내용, 처리방식, 문서화 양식등에 관한 표준화를 범기관적으로 시행할 수 있다.</p> <p>⑦ 전체 데이터 요구의 조정</p> <p>한 기관의 모든 응용 시스템들이 요구하는 데이터들을 전체적으로 수집 분석하고 상충되는 데이터 요구는 조정해서 기관 전체에 가장 유익한 구조로 조직하여 효율적인 정보처리 효과를 얻게 할 수 있다.</p>	
--	--

4. SQL(Structured Query Language)

4-1. SQL의 개요

가. 1974년 IBM 연구소에서 개발한 SEQUEL에서 유래한다.

나. 국제표준 데이터베이스 언어이며, 관계형 데이터베이스(RDB)를 지원하는 언어로 채택하고 있다.

다. 질의어(대화식으로 쉽게 DB를 이용할 수 있도록 하는 비절차어)지만 질의 기능만 있는 것이 아니라 데이터 구조의 정의, 데이터 조작, 데이터 제어 기능을 모두 갖추고 있다.

4-2. SQL의 분류

가. DDL(데이터 정의어, Data Define Language)

SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 정의하거나 변경 또는 삭제할 때 사용하는 언어이다. 논리적 데이터 구조와 물리적 데이터 구조의 사상을 정의한다.

※ 데이터베이스 관리자나 데이터베이스 설계자가 사용한다.

명령어	기 능
CREATE	SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 정의한다.
ALTER	TABLE에 대한 정의를 변경하는 데 사용한다.
DROP	SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 삭제한다.

나. DML(데이터 조작어, DML(Data Manipulation Language))

데이터베이스 사용자가 응용 프로그램이나 질의어를 통하여 저장된 데이터를 실질적으로 처리하는 데 사용되는 언어이다.

명령어	기 능
SELECT	테이블에서 조건에 맞는 튜플을 검색한다.
INSERT	테이블에서 새로운 튜플을 삽입한다.
DELETE	테이블에서 조건에 맞는 튜플을 삭제한다.
UPDATE	테이블에서 조건에 맞는 튜플의 내용을 변경한다.

다. DCL(데이터 제어어, DCL(Data Control Language))

데이터의 보안, 무결성, 회복, 병행 수행 제어 등을 정의하는 데 사용되는 언어이다.

데이터베이스 관리자가 데이터 관리를 목적으로 사용한다.

명령어	기 능
COMMIT	명령에 의해 수행된 결과를 실제 물리적 디스크로 저장하고, 데이터베이스 조작 작업이 정상적으로 완료되었음을 관리자에게 알려준다.
ROLLBACK	데이터베이스 조작 작업이 비정상적으로 종료되었을 때 원래의 상태로 복구한다.
GRANT	데이터베이스 사용자에게 사용 권한을 부여한다.
REVOKE	데이터베이스 사용자의 사용 권한을 취소한다.

4-3. CRUD

CRUD는 대부분의 컴퓨터 소프트웨어가 가지는 기본적인 데이터 처리 기능인 Create(생성), Read(읽기), Update(갱신), Delete(삭제)를 묶어서 일컫는 말이다. 사용자 인터페이스가 갖추어야 할 기능(정보의 참조/검색/갱신)을 가리키는 용어로서도 사용된다.