

- ★ 데이터베이스(Database) : 데이터들의 체계적인 집합체, 통합 레코드들의 집합
 - ① 통합된 데이터 ② 저장된 데이터 ③ 운영 데이터 ④ 공용 데이터
- ★ 정보(Information) : 사용자의 용도에 맞게 가공한 지식
- ★ 분산 데이터베이스 : 여러 사이트에 물리적으로 분산된 데이터들의 집합
- ★ 객체 지향형 데이터베이스(OODB : Object-Oriented DataBase)
 - 객체 지향 기술과 데이터베이스 기술의 결합체
- ▶ 캡슐화 : 객체 지향 데이터베이스에서 연관된 자료구조와 함수를 한 테두리로 묶는 것
- ★ 데이터 마이닝(Data Mining)
 - 기존 시스템의 축적된 데이터를 공통의 형식으로 변환하여 일원적으로 관리하는 데이터베이스
- ★ 데이터베이스 관리 시스템(DBMS)
 - 데이터베이스 시스템을 운영 관리하기 위한 소프트웨어 집합체
 - ① 정의 기능 ② 조작 기능 ③ 제어 기능
- ▶ DBMS는 데이터베이스의 회복을 위해서 백업, 로깅, 체크포인트 등을 수행한다.
 - ① 백업은 데이터베이스를 주기적으로 자기 테이프 등에 복사하는 것이다.
 - ② 로깅은 현재 수행 중인 트랜잭션의 갱신사항을 기록한다.
 - ③ 체크포인트는 시스템이 붕괴된 후 재가동되었을 때, 재수행하거나 취소해야 하는 트랜잭션 수를 줄여준다.
- ★ 스토어드 프로시저(Stored Procedures)
 - 데이터베이스 관리 시스템에서 데이터베이스 서버와 함께 저장되어 있는 연산
- ★ 스키마(Schema)
 - 데이터베이스에 저장되는 데이터의 구조 및 유형을 정의
 - 내부 스키마(Internal Schema) : 데이터베이스 구조에 대한 논리적 정의
 - 개념 스키마(Conceptual Schema)
 - 범기관적 입장에서 데이터베이스를 정의, 데이터베이스의 전체적인 논리적 구조를 정의
 - 외부 스키마(External Schema)
 - 사용자 요구사항을 도출하는 과정에 대한 정의, 서브 스키마(Sub-Schema)라고도 함
- ★ 시스템 카탈로그 : 데이터베이스에 사용되는 정보와 정보들 간에 관계를 저장
 - ① 자체가 하나의 작은 시스템 데이터베이스
 - ② 데이터 사전(Data Dictionary)이라고도 한다.
 - ③ 테이블, 데이터베이스, 뷰, 접근 권한 등의 객체를 포함
 - ④ 메타 데이터를 모아놓는 저장소

★ 메타 데이터(Meta Data)

어떤 데이터를 분석, 분류하고 부가적 정보를 추가하기 위해 데이터 뒤에 따라가는 정보
데이터를 위한 데이터

① 데이터 사전 → 메타 데이터

② 스키마가 일종의 메타 데이터가 된다.

★ 데이터 정의어(DDL)

내부 스키마를 데이터베이스에 적용하여 물리적인 데이터베이스를 정의하고 관리하는 언어
생성(CREATE), 구조변경(ALTER), 제거(DROP)

★ 데이터 제어어(DCL)

데이터 언어 중 데이터베이스를 공유하기 위한 데이터 제어를 정의하고 기술하는 언어
권한부여(GRANT), 권한제거(REVOKE), 실행결과 반영(COMMIT), 실행 중인 작업 철회(ROLLBACK)

★ 데이터 조작어(DML)

데이터 처리를 위하여 응용 프로그램과 DBMS 사이의 인터페이스 제공,
데이터 개체를 처리하고 데이터를 조작하기 위한 언어
데이터의 검색, 추가, 삭제, 수정
검색(SELECT), 갱신(UPDATE), 삽입(INSERT), 삭제(DELETE)

★ 데이터베이스 관리자(DBA : DataBase Administrator)

데이터 정의 언어를 사용하여 데이터베이스를 DBMS에 표현하고 관리하는 목적
데이터베이스를 접근하여 데이터베이스 시스템의 관리 운영에 대한 책임을 지고 있는 사람

★ 데이터베이스 설계자

데이터베이스를 사용할 사용자와 면담을 통하여 요구사항을 도출 후,
데이터베이스에 저장될 데이터를 분류 선정하고 저장 구조를 설계하며,
데이터베이스 뷰(View)를 개발하는 역할을 담당하는 사람

★ 응용 프로그래머

데이터 조작어(DML)를 호스트 언어에 포함하여 응용 프로그램을 개발,
일반 사용자에게 대한 인터페이스를 제공할 목적으로
데이터베이스에 접근하는 사용자

★ 일반 사용자(User)

질의어(Query)를 사용하여 응용 프로그램을 통하여 데이터베이스에 접근

★ 데이터 모델(Data Model)

현실 세계의 데이터를 컴퓨터에 표현하기 위한 논리적 모형
구성 요소 : 개체, 속성, 관계

★ 개체(Entity)

데이터베이스에 표현하려고 하는 유형, 무형의 정보

★ 속성(Attribute)

- ① 관계 데이터 모델에서 데이터의 가장 작은 논리적 단위
- ② 파일 구조상의 데이터 항목 또는 데이터 필드와 동일
- ③ 데이터 개체를 구성

특성

- ① 개체의 성질이나 상태를 기술해 주는 역할
- ② 자체만으로는 중요한 의미를 가지지 못하며, 단독으로 존재하기 어려운 특성
- ③ 개체의 성질, 분류, 식별, 수량, 상태 등을 나타내는 세부 정보의 관리 요소

★ 관계

두 개 이상의 개체(Entity)들 사이의 연관성을 결정짓는 의미 있는 연결

- ① 두 개체 사이에서 1 : 1, 1 : N, N : 1, M : N의 카디널리티 비율 제약조건을 가지고 E-R 다이어그램에서 다이아몬드 모양으로 표현한다.
- ② 데이터 테이블에 존재하는 인스턴스의 상태에 따라 차수가 결정
- ③ 다 대 다(M : N)의 경우에는 교차 테이블 이용

★ 개체 - 관계 모델(E - R Model)

데이터베이스를 구성하는 개체(Entity) 타입과 관계(Relation) 타입 간의 구조, 개체를 구성하는 속성(Attribute) 등을 기호를 이용하여 표현함으로써, 데이터베이스의 전반적인 구조를 이해하기 쉽게 표현할 수 있는 방법
현실 세계의 각종 데이터를 논리 데이터로 표현하기 위한 가장 대표적인 방법

★ 관계 데이터 모델

2차원 구조의 표(테이블) 형태로 데이터 상호관계를 표현하는 방법

- ① 데이터베이스의 논리적 설계 단계에서 사용되는 데이터베이스 모델 중 하나
- ② 표 데이터 모델이라고도 함
- ③ 구조가 단순하며 사용이 편리하여 가장 많이 사용하고 있는 형태

★ 관계 해석

릴레이션에서 원하는 결과를 얻기 위한 과정을 표현

코드(E. F. Codd)가 수학의 술어해석(Predicate Calculus)을 기반으로, 관계 데이터베이스를 위해 제안
튜플 변수나 도메인 변수, 한정 애트리뷰트, 원자, 정형식 등으로 구성되는 비절차적 처리
원하는 정보가 무엇이라는 것만 선언
연산자 없이 정의하는 방법을 이용하는 비절차적 언어의 특징을 갖음

★ 관계 대수

원하는 정보와 그 정보를 어떻게 유도하는가를 기술하는 절차적인 방법
릴레이션으로부터 필요한 릴레이션을 만들어내는 연산자의 집합

★ 속성(Attribute)

관계 데이터 모델에서 데이터의 가장 작은 논리적 단위

★ 튜플(Tuple)

관계형 데이터 모델에서 릴레이션을 구성하는 각각의 행(Row)을 의미
파일 구조에서 레코드(Record)와 같은 의미

★ 카디널리티(Cardinality)

개체들 간의 관계를 표현할 때 종속 개체가 가질 수 있는 튜플의 개수

★ 도메인(Domain)

관계 데이터 모델에서 하나의 속성이 취할 수 있는 같은 데이터 타입의 모든 원자 값들의 집합
데이터베이스에 저장되는 데이터 속성들의 데이터 유형, 데이터의 길이, 데이터 허용 범위,
데이터의 기본값, 데이터의 생성규칙 등을 정의

★ 후보키(Candidate Key)

릴레이션에서 튜플을 유일하게 구별해 주는 속성 또는 속성들의 집합
하나의 릴레이션에 속하는 모든 원소(튜플)들은 중복된 값을 가질 수 없으므로
모든 릴레이션은 반드시 하나 이상을 가져야 한다.
후보가 될 수 있는 조건은 유일성과 최소성을 모두 만족해야 한다.

★ 기본키(Primary Key)

한 릴레이션에서 특정 튜플을 유일하게 구별할 수 있는 속성
후보키 중에서 선택한 주키(Main Key)이다.

★ 외래키(Foreign Key)

하나의 테이블에서 필요에 의해 다른 테이블을 참조하기 위해 사용되는 속성
참조 릴레이션의 기본키(Primary Key)와 동일해야 한다.
NULL이 올 수 없다.

★ 널값(Null Value)

관계의 한 속성이 가질 수 있는 특별한 원자값으로 “미정”, “알 수 없음”
또는 “해당 없음”과 같은 뜻을 가지며, 아무 것도 없는 상태를 나타낸다.

★ 색인순차접근방식(ISAM : Indexed Sequential Access Method)

레코드에 순차적으로 접근하거나 색인을 통해 선택적으로 접근하는 방식들 모두를 제공하는
파일 관리 시스템이다.

★ 인덱스(Index)

데이터를 보다 효율적으로 관리하고 빠르게 검색할 수 있도록
데이터의 위치 정보와 관련지어 유지하는 정보

★ 해싱(Hashing)

직접 파일은 저장할 데이터의 키값을 디스크의 물리적 주소로 변환할 수 있는
해싱 함수를 이용하는 파일 구조

★ 클러스터링(Clustering)

검색 효율을 개선하기 위해 관련 있는 Record들을 물리적으로 인접한 장소에 위치시키는 기법

★ 백업(Backup)

회복 기능의 하나, 컴퓨터 시스템의 운용 중에 발생하는 각종 고장에 대해 하드웨어 기기 및 파일이나
데이터베이스의 내용 등을 언제라도 회복시킬 수 있도록 예비해두는 것

▶ 개방형 데이터베이스 접속성(ODBC : Open DataBase Connectivity)

- ① 윈도 응용 프로그램에서 다양한 데이터베이스 관리 시스템에 접근하여 사용할 수 있도록 개발한 것
- ② 공통적인 인터페이스를 통해 서로 다른 데이터베이스 파일을 볼 수 있도록 만들어 준
데이터베이스 표준 접속 규격을 말한다.

★ 무결성(Integrity)

데이터베이스 내에 저장된 데이터 값과 그것이 나타내는 실제 값이 일치하는 정확성을 의미
무결성 규정의 대상으로 도메인, 키, 종속성, 관계성 등이 있다.

★ 개체 무결성

릴레이션(테이블)에서 개체(Entity)를 식별함에 있어 오류가 없도록 하기 위한 제약조건
기본키는 중복된 값이 있을 수 없으며, NULL이 될 수 없을 나타내는 제약조건

★ 참조 무결성

두 릴레이션(테이블)의 참조 관계에서 외래키 값은 참조 릴레이션의 기본키와 같다는 제약조건으로,
릴레이션(테이블 참조 시 오류가 발생하지 않도록 하기 위한 제약조건을 말한다.

▶ 동시성(Concurrency)

- ① 데이터베이스 관리 시스템의 중요 관심사는 어떻게 많은 사용자들이
동시에 같은 데이터를 액세스하는 것을 제어하는가이다.
- ② 충분한 동시성 제어가 없다면, 부적당한 데이터가 업데이트 또는 변경될 수 있고,
데이터 무결성을 손상시킬 수 있다.

★ 정규화(Normalization)

관계형 모델에서 서로 독립적인 관계에 있는 정보들이 별개의 릴레이션으로 표현되도록 하는 작업
정규화의 목적

- ① 자료저장 공간의 최소화 및 데이터 구조의 안정성 최대화
- ② 데이터베이스 내부 자료의 무결성 유지 극대화
- ③ 데이터베이스 사용자의 의도하지 않은 삽입, 삭제, 갱신이 발생하는 이상(Anomaly) 현상을 최소화

★ 함수 종속성

- ① 관계 데이터베이스가 올바르게 설계되었는지 검증할 수 있다.
- ② 정규화 과정에서 필요하다.
- ③ FD라고도 한다.
- ④ $A \rightarrow B$ 형태
- ⑤ 릴레이션의 어떤 속성의 값이 다른 속성의 값을 고유하게 결정하는 것

★ 트랜잭션(Transaction)

데이터베이스에 보관되어 있는 자료를 조작하여 작업을 수행하는 모든 것

- ① 데이터베이스 관련 연산의 가장 기본적인 단위
- ② 원자성, 일관성, 격리성, 영속성 등의 특징을 갖는다.
- ③ 사용자의 데이터베이스 접근 기본 단위이며,
여러 개의 질의(Query)문이 마치 하나의 질의문처럼 처리되는 것을 의미
- ④ 사용자가 제기한 데이터베이스의 작업 수행을 위한 데이터 조작어의 집단을 말한다.
- ⑤ 데이터베이스의 일관적 상태 유지를 위한 병행 수행 제어 및 회복의 기본 단위
- ⑥ 데이터 파일에 영향을 미치는 참조 및 갱신 등의 모든 작업을 의미

▶ 로그(Log)

- ① 트랜잭션 수행 중 작성된 데이터베이스의 변경 내용에 대한 정보
- ② Redo와 Undo를 수행할 때 이용되는 것
- ③ 일반적으로 <트랜잭션 ID, 데이터 아이템, 이전 값, 이후 값> 형식을 갖고 있다.

★ 원자성(Atomicity)

트랜잭션의 특성 중 한 트랜잭션의 모든 연산이 데이터베이스에 완전히 수행되거나
수행되지 않음을 의미하는 특성

★ 일관성

실행된 후에도 언제나 일관성 있는 데이터베이스 상태가 되어야 하며,
결과에 모순이 생겨서는 안 되도록 하는 성질

▶ ABORT

- ① 작업 수행 중인 데이터베이스 트랜잭션에 대해 강제로 작업을 종료시킴(비정상적으로 종료하는 것)
- ② 데드락(Deadlock)의 원인이 되기도 한다.

★ 격리성(고립성)

현재 실행 중인 트랜잭션이 완료될 때까지 다른 트랜잭션들이 접근할 수 없도록 하는 성질

★ 영속성(지속성)

트랜잭션이 성공적으로 완료된 후 데이터베이스에 반영한 수행 결과는 영구적으로 반영되어야 하는 성질

★ SQL(Structured Query Language)의 개념

관계 대수와 관계 해석을 기초로 한 고급 데이터 언어

- ① 질의어지만, 질의 기능만 있는 것이 아니라 데이터 구조의 정의, 데이터 조작, 데이터 제어 기능을 모두 갖추고 있다.
- ② 관계 데이터베이스로부터 정보를 요청하기 위해 사용하는 표준화된 질의어이다.
- ③ 비절차형 프로그램 언어로서, 각 명령은 데이터를 독립적으로 처리할 수 있다.

★ NoSQL

- ① 비관계형 데이터 저장소
- ② 데이터 스토어들은 테이블 스키마가 고정되지 않는다.
- ③ 테이블 간 조인(join) 연산을 지원하지 않는다.
- ④ 대체로 수평적 확장이 용이하다.

★ 외부조인

릴레이션에 있는 어떤 튜플에 대해 조인할 상대 릴레이션에 대응하는 튜플이 없을 경우, 이를 제외시키지 않고 상대를 널(Null) 튜플로 만들어 결과 릴레이션에 포함시키는 것으로, 조인 조건을 만족하지 않는 행까지 결과 집합에 포함하여 돌려주는 조인 유형의 한 종류를 말한다.

★ 뷰(VIEW)

SQL에서 사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위해 하나 이상의 물리적인 기본 테이블로부터 유도된 가상 테이블을 말한다.
뷰(VIEW) 생성할 때는 CREATE문을 사용하여 정의한다.

★ 데이터 정의어(DDL)

CREATE, ALTER, DROP, CASCADE, RESTRICT

★ CREATE

테이블, 인덱스, 스키마, 뷰 등을 생성할 때 사용하는 SQL명령이다.
사용법 : CREATE 생성대상 생성대상_이름

★ ALTER

테이블 구조 변경 시 사용하는 데이터 정의 언어
새로운 속성 추가, 속성에 대한 새로운 값을 정의하는 SQL문이다.

★ DROP

스키마, 도메인, 테이블, 뷰, 인덱스의 제거 시 사용되는 SQL명령이다.

★ CASCADE

- ① SQL에서 Main Table의 데이터 삭제 시 각 외래키에 대해 부합되는 모든 데이터를 삭제하는 옵션, 삭제할 요소를 참조하는 다른 모든 개체를 함께 삭제할 때 사용
- ② 참조 무결성 제약조건이 설정된 기본 테이블의 특정 데이터를 삭제할 때, 그 데이터와 관계를 맺고 연관되어 있는 다른 테이블의 데이터들도 연쇄적으로 삭제시킨다.

★ RESTRICT

참조 무결성 제약조건이 설정된 기본 테이블의 특정 데이터를 삭제할 때,
그 데이터와 관계를 맺고 연관되어 있는 데이터가 있다면 삭제되지 않도록 하는 옵션

★ 데이터 조작용어(DML)

UPDATE, WHERE, AND, HAVING, DISTINCT, ORDER BY, 집단 함수, GROUP BY

★ UPDATE

데이터의 입력을 잘못하여 올바른 값이 되도록 수정을 할 때 사용하는 명령
사용법 : UPDATE 테이블_이름 SET 변경사항 WHERE 조건;

- ▶ SELECT : 관계 데이터베이스에서 정보를 검색하는 SQL문은 SELECT문이다.
- ▶ DELETE ① 테이블에서 데이터를 삭제할 때 사용하는 명령
② WHERE절을 생략하면 모든 데이터가 삭제되지만, 테이블 자체는 삭제x (빈 테이블)
③ 사용법 : DELETE FROM 테이블_이름 WHERE 조건;
- ▶ INSERT : ① 테이블에 새로운 데이터를 삽입할 때 사용되는 명령
② 사용법 : INSERT INTO 테이블_이름(속성_이름) VALUES (실제_데이터);

★ WHERE

SQL 질의에서 조건을 나타낼 때 사용

사용 예 : [R] 테이블에서 R.A 속성이 R.B 속성보다 큰 값을 모두 검색하려면
SELECT * FROM R WHERE R.A > R.B;

★ AND

조건식에서 사용하는 논리 연산자, 두 개의 조건을 모두 만족할 경우 참이 된다.

사용 예 : [SAWON] 테이블에서 나이가 25살 이상인 모든 여자(성별에서 여자를 'F'로 함)를 찾으려면
SELECT * FROM SAWON WHERE age>24 AND sex = "F";

- ▶ UPPER : 데이터베이스 질의 시에 사용되는 함수로, 대상이 되는 문자열 모두를 대문자로 변환한 문자열 값으로 변환하기 때문에 보통 문자열을 비교할 때 사용한다.

- ★ HAVING : SQL문장의 GROUP BY 절에 선택된 조건검색을 지정하기 위한 구문으로, 테이블에 존재하는 특정 값의 포함여부를 비교한다.

★ DISTINCT

SELECT문장을 이용하여 데이터를 질의할 때 검출되는 중복 값을 제거하기 위해 사용되는 명령
SQL에서 테이블의 특정 항목에 같은 이름이 중복되어 입력되어 있는 경우, 중복을 피하고 값이 다른 행만 나타내고 싶을 경우 사용한다.

★ ORDER BY

- ① SELECT문에서 질의 결과의 순서를 명시하지 않으면 DBMS가 튜플들을 검색한 임의의 순서대로 사용자에게 제시된다. 이에 하나 이상의 애트리뷰트를 사용하여 검색 결과를 정렬하여 표시하게 하는 구문이다.
- ② SELECT문에서 한 개 또는 그 이상의 속성(필드)을 기준으로, 오름차순(ASC) 또는 내림차순(DESC)으로 정렬한다.
정렬 방식은 생략이 가능하며, 생략된 경우 오름차순으로 정렬된다.
- ③ 사용법 : ORDER BY 속성_이름 [ASC or DESC]

★ 집단 함수

대상이 되는 행을 모은 '그룹' 개념으로 사용되는 함수

행 집합에 적용하여 그룹당 하나의 결과를 생성

종류 : SUM, AVG, MAX, MIN, COUNT

- ① COUNT : √, 그룹별로 튜플의 수를 구하는 함수이다.
 - ② MAX : √, 그룹별로 최댓값을 구하는 함수이다.
 - ③ SUM : √, 그룹별로 합계를 구하는 함수이다.
 - ④ MIN : √, 그룹별로 최솟값을 구하는 함수이다.
 - ⑤ AVG : √, 그룹별로 평균값을 구하는 함수이다.
- √ = 데이터베이스 질의 시에 사용되는 그룹 함수의 한 종류로,

★ GROUP BY

릴레이션의 한 열을 요약하거나 집계하는 역할을 한다.

★ 데이터 제어어(DCL)

데이터 제어어의 종류 : GRANT, REVOKE, ROLLBACK, COMMIT

★ GRANT

데이터베이스 관리자가 데이터베이스 사용자에게 권한을 부여하는 명령어

★ REVOKE

부여된 데이터베이스의 사용 권한을 제거하는 명령어

★ ROLLBACK

데이터베이스 조작 작업이 비정상적으로 종료됐을 때 변경된 모든 내용들을 취소하고 데이터베이스를 이전 상태로 되돌리는 명령어

★ COMMIT

수행된 결과를 실제 물리적 디스크로 저장하고, 데이터베이스 조작 작업이 정상적으로 완료되었음을 관리자에게 알려주는 명령어