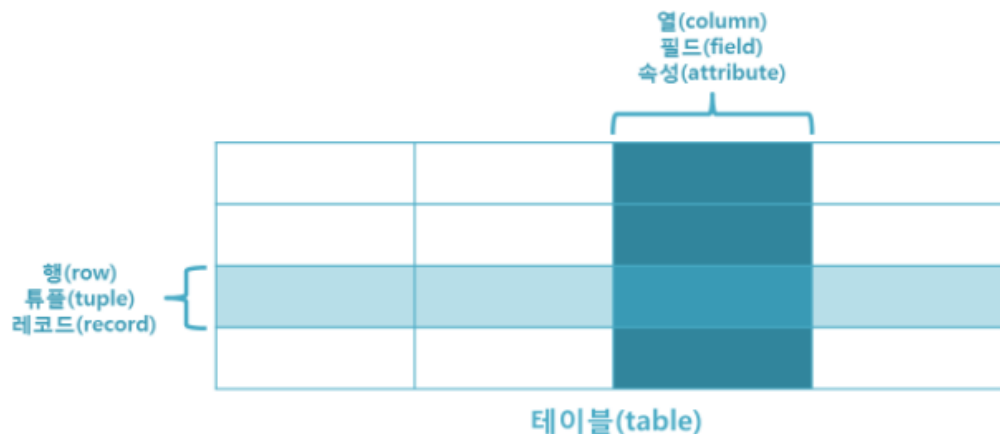


# 3과목. 데이터베이스 구축

## ▼ 관계형 데이터베이스의 Relation 구조

### 1) 관계형 데이터베이스의 Relation 구조 ★★

- 1970년 IBM에 근무하던 코드(E. F. Codd)에 의해 처음 제안됨



출처: [http://www.tcpschool.com/mysql/mysql\\_intro\\_relationalDB](http://www.tcpschool.com/mysql/mysql_intro_relationalDB)

#### ▶ 튜플(Tuple), 행(Row), 레코드(Record)

-속성의 모임으로 구성됨

-파일 구조상 레코드(실제 데이터)와 같은 의미

-튜플의 수 = 카디널리티(Cardinality) 또는 기수, 대응수 ★

#### ▶ 속성(Attribute), 열(Column), 필드(Field)

-데이터베이스를 구성하는 가장 작은 논리적 단위

-파일 구조상의 데이터 항목 또는 데이터 필드에 해당

-개체의 특성을 기술

-속성의 수 = 디그리(Degree) 또는 차수 ★

#### ▶ 도메인(Domain) ★ \_\_ 20년 1, 2, 3회 기출문제

-하나의 속성(Attribute, 애트리뷰트)이 가질 수 있는 같은 타입 원자(Atomic)값들의 집합

ex) 성별 속성(Attribute)의 도메인은 '남', '여'로 그 외의 값은 입력될 수 없음(일반적)

## ▼ 무결성

데이터베이스에 저장된 데이터 값과 그것이 표현하는 현실 세계의 실제 값이 일치하는 정확성을 의미

▼ 개체 무결성

- Entity Integrity, 실체 무결성
- 테이블의 기본키를 구성하는 어떤 속성(Attribute)도 널(Null) 값이나 중복 값을 가질 수 없음
- 기본키의 속성 값이 널(Null) 값이 아닌 원자 값을 갖는 성질

▼ 도메인 무결성

- Domain Integrity, 영역 무결성
- 릴레이션 내의 튜플들이 각 속성(Attribute)의 도메인에 지정된 값만을 가져야 함

▼ 참조 무결성

- Referential Integrity
- 외래키 값은 Null이거나 참조 릴레이션의 기본키 값과 동일해야 함
- 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다는 규정

▼ 사용자 정의 무결성

- 속성 값들이 사용자가 정의한 제약 조건에 만족해야 함

▼ 데이터 무결성 강화

▼ 관계대수

- 관계형 데이터베이스에서 원하는 정보와 그 정보를 검색하기 위해서 어떻게(How) 유도하는가를 기술하는 절차적인 언어
  - ▼ 순수 관계 연산자 (#셀프조디)

연산자	기호	의미
Select (선택)	$\sigma$	<u>조건(Predicate)을 만족하는 튜플들의 부분 집합</u> (수평 연산) ★
Project (추출)	$\pi$	<u>속성들의 부분 집합, 중복은 제거됨</u> (수직 연산) ★
Join (조인)	$\bowtie$ ex) $R \bowtie S$	두 개의 릴레이션이 <u>공통으로 가지고 있는 속성을 이용하여 두 개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 연산</u> ★ 두 개의 릴레이션의 <u>연관된 튜플들을 결합</u>
Division (나누기)	$\div$ ex) $R \div S$	R릴레이션에서 S릴레이션의 속성 <u>도메인 값과 일치하는 R릴레이션의 튜플들을 찾아내는 연산</u> ★

▼ 일반집합 연산자 (#합교차카)

연산자	기호	의미
Union (합집합)	$\cup$ ex) $R \cup S$	두 개의 릴레이션의 <u>합이 추출되고, 중복은 제거됨</u>
Intersection (교집합)	$\cap$ ex) $R \cap S$	R릴레이션과 S릴레이션의 <u>중복되는 값들만 추출</u>
Difference (차집합)	$-$ ex) $R - S$	R릴레이션에서 S릴레이션에 <u>중복되지 않는 값들만 추출</u>
Cartesian Product (교차곱)	$\times$ ex) $R \times S$	두 릴레이션의 가능한 모든 튜플들의 집합, <u>차수(Degree)는 더하고, 카디널리티(Cardinality)는 곱해서 값을 구함</u> ★

▼ 카탈로그

▼ 시스템 카탈로그의 의미

- 사용자를 포함해 DBMS에서 지원하는 모든 데이터 객체에 대한 정의나 명세에 관한 정보를 유지관리하는 시스템 테이블
- 좁은 의미로는 카탈로그를 데이터 사전(Data Dictionary)이라고도 함
- 시스템 카탈로그에 저장된 정보 = 메타 데이터(Meta-Data)

#### ▼ 카탈로그의 특징

- 카탈로그는 DBMS가 스스로 생성하고 유지함
- DBMS에 따라 상이한 구조를 가짐
- 사용자가 SQL문을 실행시켜 변화를 주면 시스템이 자동으로 갱신함
- 단, insert, delete, update문으로 카탈로그를 갱신할 수 없음.
- 대신 일반 이용자도 SQL을 이용하여 내용을 검색할 수 있음.

#### ▼ 데이터 디렉터리(Data Directory, 사전 관리기)

- 데이터사전(Dictionary)에 수록된 데이터를 실제로 접근하는데 필요한 정보를 관리 및 유지하는 시스템이고
- 시스템만 접근할 수 있다.

cf. 카탈로그(Dictionary): 사용자와 시스템 모두 접근 가능

#### ▼ 트랜잭션

##### ▼ 트랜잭션의 정의 - commit, rollback, savepoint

##### 1) 트랜잭션의 정의 ★ — 20년 3회 기출문제

- 데이터베이스의 상태를 변환시키는 하나의 논리적 기능을 수행하기 위한 작업의 단위
- 한꺼번에 모두 수행되어야 할 일련의 연산들

▶ **COMMIT**: 트랜잭션 처리가 정상적으로 종료되어 수행한 변경 내용을 DB에 반영하는 명령어

▶ **ROLLBACK**: 트랜잭션 처리가 비정상적으로 종료되어 DB의 일관성이 깨졌을 때 트랜잭션이 행한 모든 변경 작업을 취소하고 이전 상태로 되돌리는 연산

\* COMMIT과 ROLLBACK 명령어에 의해 보장 받는 트랜잭션 특징 = 원자성 ★

▶ **SAVEPOINT(=CHECKPOINT)**: 트랜잭션 내에서 ROLLBACK할 위치인 저장점을 지정하는 명령어, 여러 개의 SAVEPOINT 지정 가능

▼ 트랜잭션의 특성 - 원자성, 일관성, 독립성, 영속성(ACID)

원리	특징
원자성 (Atomicity)	트랜잭션 연산을 데이터베이스 <u>모두에 반영되든지 아니면 전혀 반영되지 않아야 함</u> (All or Nothing) ★
일관성 (Consistency)	트랜잭션이 실행을 성공적으로 완료할 시 <u>일관성 있는</u> 데이터베이스 상태를 유지
독립성 (Isolation, 격리성)	둘 이상 트랜잭션 동시 실행 시 <u>한 개의 트랜잭션만 접근이 가능</u> 하여 간섭 불가
영속성 (Durability)	성공적으로 완료된 트랜잭션 결과는 <u>영구적으로 반영</u> 됨
#ACID	

▼ view

▼ 뷰의 개요 및 속성

1) 뷰의 개요 및 특징

- 기본 테이블로부터 유도된, 이름을 가지는 가상 테이블로 기본 테이블과 같은 형태의 구조를 사용하며, 조작도 기본 테이블과 거의 같음
- 가상 테이블이기 때문에 물리적으로 구현되어 있지 않지만 사용자에게 있는 것처럼 간주됨 → 저장장치 내에 논리적으로 존재
- 정의된 뷰로 다른 뷰를 정의할 수 있음
- 뷰가 정의된 기본 테이블이나 뷰를 삭제하면 그 테이블이나 뷰를 기초로 정의된 다른 뷰도 자동으로 삭제됨

속성	설명
REPLACE	뷰가 이미 존재하는 경우 <u>재생성</u>
FORCE	본 테이블의 <u>존재 여부에 관계 없이</u> 뷰 생성
NOFORCE	기본 테이블이 존재할 때만 뷰 생성
WITH CHECK OPTION	서브 쿼리 내의 조건을 만족하는 행만 변경
WITH READ ONLY	데이터 조작어(DML) 작업 불가

▼ 뷰의 장단점

## 2) 뷰의 장, 단점 ★ — 20년 1, 2, 3회 기출문제

### ▶ 장점

- 논리적 데이터 독립성 제공
- 접근 제어를 통한 자동 보안 제공
- 사용자 데이터 관리 용이

### ▶ 단점

- 독립적인 인덱스를 가질 수 없음
- 뷰의 정의를 ALTER로 변경할 수 없음 → DROP하고 새로 CREATE해야 함
- 뷰로 구성된 내용에 대한 삽입, 삭제, 갱신, 연산에 제약이 따름

## ▼ 클러스터: 데이터의 효율적인 저장

### 1) 클러스터의 개요 및 특징 ★

- 데이터 저장 시 데이터 액세스 효율을 향상시키기 위해 동일한 성격의 데이터를 동일한 데이터 블록에 저장하는 물리적 저장 방법
- 인덱스의 단점을 해결한 기법 → 분포도(Selectivity)가 넓을수록 오히려 유리함
- 분포도가 넓은 "테이블"의 클러스터링은 저장 공간의 절약이 가능
- 대량의 범위를 자주 액세스(조회)하는 경우 적용
- 인덱스를 사용한 처리 부담이 되는 넓은 분포도에 활용

### 2) 클러스터의 선정기준 및 고려사항

#### ▶ 클러스터 테이블 선정

- 수정이 빈번하지 않는 "테이블"
- ORDER BY, GROUP BY, UNION이 빈번한 "테이블"
- 처리 범위가 넓어 문제가 발생하는 경우 단일 테이블 클러스터링 사용
- 조인이 많아 문제가 발생하는 경우는 다중 테이블 클러스터링 사용

#### ▶ 설계 시 고려사항

- 클러스터링 된 테이블은 조회 속도를 향상시켜주지만 입력, 수정, 삭제 시 성능이 저하됨(부하가 증가)
- 대용량을 처리하는 트랜잭션은 전체 테이블을 스캔하는 일이 자주 발생하므로 클러스터링을 하지 않는 것이 좋음
- 클러스터링 된 테이블에 클러스터드 인덱스를 생성하면 접근 성능이 향상됨

## ▼ 데이터베이스 보안스토리지

### ▼ 데이터베이스 보안이란?

데이터베이스 일부분 또는 전체에 대해서 권한이 없는 사용자가 액세스 하는 것을 금지하기 위해 사용되는 기술

### ▼ 암호화(Encryption)

- 암호화 과정 = 평문 → 암호문
  - # 개인키 암호방식 (대칭키)
  - # 공개키 암호방식 (비대칭키)
- 복호화(Decryption) 과정 = 암호문 → 평문

### ▼ 암호화 방식

방식	특징	종류
개인키 암호 방식 (Private Key Encryption, 비밀키 암호 방식, 대칭키)	<u>동일한 키</u> 로 데이터를 암호화하고 복호화 함, 비밀키는 DB 사용 권한이 있는 사용자만 나눠 가짐	DES, AES, SEED, ARIA
공개키 암호방식 (Public Key Encryption, 비대칭키)	데이터를 <u>암호화할 때 사용하는 키(공개키)</u> 는 DB 사용자에게 공개하고, <u>복호화할 때의 키</u> ( <u>비밀키</u> )는 관리자가 관리하는 방법	RSA (Rivest Shamir Adleman) Diffie Hellman Algorithm

### ▼ 접근통제(DAC, MAC)

#### 4) 접근통제 \_\_ 5-86

- 데이터가 저장된 객체와 이를 사용하려는 주체 사이의 정보 흐름을 제한하는 것

- 접근통제 3요소: 접근통제 정책, 접근통제 보안모델, 접근통제 메커니즘

##### #정보커

##### ▶ 임의 접근통제(DAC; Discretionary Access Control)

-데이터에 접근하는 사용자의 신원에 따라 접근 권한 부여

# 접근통제 권한=주체

##### ▶ 강제 접근통제(MAC; Mandatory Access Control)

-주체와 객체의 등급을 비교해 접근 권한 부여

# 접근통제 권한=제3자

#### ▼ 접근통제 정책(DAC, MAC, RBAC)

정책	설명
신분 기반 정책 (DAC)	주체나 그룹의 <u>신분</u> 에 근거해 객체의 접근을 제한하는 방법 # IBP(Individual-Based Policy), GBP(Group-Based Policy)
규칙 기반 정책 (MAC)	주체가 갖는 <u>권한</u> 에 근거해 객체의 접근을 제한하는 방법 # MLP(Multi-Level Policy), CBP(Compartment-Based Policy)
역할 기반 정책 (RBAC)	주체가 맡은 <u>역할</u> 에 근거해 객체의 접근을 제한하는 방법 # 인사 담당자, DBA(Database Administration)

#### ▼ 접근 통제 메커니즘(ACL, CL, Security Label)

#### 6) 접근통제 메커니즘 ★

##### ▶ 접근통제 목록(ACL; Access Control List)

객체를 기준으로 특정 객체에 대해 어떤 주체가 어떤 행위를 할 수 있는지를 기록한 목록

##### ▶ 능력 리스트(CL; Capability List)

주체를 기준으로 주체에게 허가된 자원 및 권한을 기록한 목록

##### ▶ 보안 등급(Security Label), 패스워드, 암호화



▼ 접근 통제 보안 모델 (벨라파둘라 비바 등)

## 7) 접근통제 보안 모델

### ▶ 기밀성 모델

군사적인 목적으로 개발된 최초의 수학적 모델, 기밀성 보장 최우선

# 벨라파둘라 모델: No Read UP(기밀성), No Write Down

### ▶ 무결성 모델

불법적인 정보 변경을 방지하기 위해 무결성을 기반으로 개발된 모델

# 비바 모델: No Read Down, No Write Up(무결성)

### ▶ 접근통제 모델

접근통제 메커니즘을 보안 모델로 발전시킨 것

# 접근통제 행렬(Access Control Matrix): 행=주체, 열=객체

▼ 스토리지(Storage)

종류	설명	장점	단점
DAS (Direct Attached Storage)	서버와 저장장치를 <u>전용 케이블로 직접 연결</u> 하는 방식	속도가 빠르고 <u>설치 및 운영이 쉬움</u> , 초기 구축 및 유지보수 비용 저렴	파일 공유 불가, 확장성 및 유연성이 떨어짐
NAS (Network Attached Storage)	서버와 저장장치를 <u>네트워크를 통해 연결</u> 하는 방식	장소에 구애받지 않고 저장장치에 쉽게 접근, <u>확장성 및 유연성 우수</u>	접속 증가 시 성능 저하
SAN (Storage Area Network)	서버와 저장장치를 연결하는 <u>전용 네트워크를 별도로 구성</u> 한 방식	<u>파이버 채널 스위치</u> 로 네트워크 구성, <u>광케이블로 처리</u> 속도 빠름, 확장성, 유연성, 가용성 뛰어남	설치 비용이 많이 들
# <u>다나쓰</u>			

▼ SQL - DDL

▼ DDL

## 1) SQL(Structured Query Language)의 분류 ★★ \_\_ 20년 1, 2회 기출문제

▶ **DDL**(Data Define Language, 데이터 정의어) \_\_ 3-26

-DOMAIN(도메인), SCHEMA(스키마), TABLE(테이블), VIEW(뷰), INDEX(인덱스)를 정의하거나 변경 또는 삭제할 때 사용하는 언어

#도스테뷰인

명령어	기능
CREATE	DOMAIN, SCHEMA, TABLE, VIEW, INDEX <u>정의</u> → CREATE DOMAIN, SCHEMA, TABLE, VIEW, INDEX 도스테뷰인명;
ALTER	TABLE에 대한 <u>정의 변경</u> → ALTER TABLE 테이블명;
DROP	DOMAIN, SCHEMA, TABLE, VIEW, INDEX <u>삭제</u> → DROP DOMAIN, SCHEMA, TABLE, VIEW, INDEX 도스테뷰인명; * <u>CASCADE</u> : 참조하는 모든 개체 <u>함께 제거</u> ★ * <u>RESTRICTED</u> : 다른 개체가 제거할 요소를 참조 중이면 제거 취소

### ▼ DML

▶ **DML**(Data Manipulation Language, 데이터 조작어) ★ \_\_ 3-13, 1, 2회 기출문제

-데이터베이스 사용자가 응용 프로그램이나 질의어를 통해 저장된 데이터를 실질적으로 처리하는데 사용하는 언어

명령어	기능
SELECT	테이블에서 조건에 맞는 튜플 <u>검색</u> → <u>SELECT FROM</u> 테이블명 [ <u>WHERE</u> 조건];
INSERT	테이블에 새로운 튜플 <u>삽입</u> → <u>INSERT INTO</u> 테이블명 <u>VALUES</u> 데이터;
DELETE	테이블에서 조건에 맞는 튜플 <u>삭제</u> → <u>DELETE FROM</u> 테이블명 [ <u>WHERE</u> 조건];
UPDATE	테이블에서 조건에 맞는 튜플의 <u>내용 갱신(변경)</u> → <u>UPDATE</u> 테이블명 <u>SET</u> 속성명=데이터 [ <u>WHERE</u> 조건];

### ▼ DCL

▶ **DCL(Data Control Language, 데이터 제어어)** \_\_ 3-15, 20년 1, 2, 3회 기출문제

-데이터의 무결성, 보안, 회복, 병행수행 제어 ★ 등을 정의하는 데 사용되는 언어

-데이터베이스 관리자(DBA)가 데이터 관리를 목적으로 사용

명령어	기능
COMMIT	명령에 의해 수행된 결과를 실제 물리적 디스크로 저장하고, 데이터베이스 조작 작업이 <u>정상적으로 완료됐음</u> 을 알려주는 명령어
ROLLBACK	아직 COMMIT되지 않은 변경된 모든 내용들을 <u>취소</u> 하고, 데이터베이스를 <u>이전 상태로 되돌리</u> 는 명령어 *SAVEPOINT: 트랜잭션 내에 ROLLBACK 할 위치인 <u>저장점</u> 을 지정하는 명령어
GRANT	데이터베이스 사용자에게 사용 권한 부여 → GRANT 권한 리스트 ON 개체 TO 사용자 [WITH GRANT OPTION]; *WITH GRANT OPTION: 부여받은 권한을 다른 사용자에게 다시 부여할 수 있는 권한
REVOKE	데이터베이스 사용자의 사용 권한 <u>취소</u> → REVOKE [GRANT OPTION FOR] 권한 리스트 ON 개체 FROM 사용자 [CASCADE]; *GRANT OPTION FOR: 다른 사용자에게 권한을 부여할 수 있는 권한을 취소

## ▼ SELECT

### 2) SELECT \_\_ p.444~453 ★ \_\_ 20년 1, 2, 3회 기출문제

▶ **WHERE**절: 검색할 조건을 기술

▶ **ORDER BY**절: 특정 속성을 기준으로 정렬해 검색할 때 사용

#ASC(오름차순), DESC(내림차순) – 따로 설정이 없을 때는 기본적으로 ASC 사용

▶ **GROUP BY**절: 특정 속성을 기준으로 그룹화해 검색할 때 사용, 일반적으로 그룹 함수와 함께 사용

▶ **HAVING**절: GROUP BY와 함께 사용되며, 그룹에 대한 조건 지정

\* **DISTINCT**: 중복 튜플 제거 ★

▶ **집계/그룹함수**: GROUP BY절에 지정된 그룹별로 속성의 값을 집계할 함수를 기술함