Database (데이터베이스)

사전 발표 준비 및 전체적인 이론 학습

데이터베이스

데이터베이스

다양한 사용자가 동시에 사용하는 데이터의 대규모 저장소

↓

DBMS

DBMS

김혜란

- 데이터베이스를 관리하고 운영하는 소프트웨어

- MySQL, 오라클(Oracle), SQL 서버, MariaDB 등

- 계층형(Hierarchical), 망형(Network), 관계형(Relational), 객체지향형(Object-Oriented), 객체관계형(Object-Relational)

엔티티 (Entity)
데이터베이스에서 표현하려는 객체

릴레이션(Relation) / 테이블
2차원 테이블

1. 학생(Student) Entity: 학생 개체는 학번, 이름, 학년, 전공등의 속성이 있을 수 있음
2. 학생(Student) Relation: 학생 Entity의 속성을 저장하는 테이블,
   테이블의 열은 학번, 이름, 학년, 전공 등의 속성을 나타내고 각 행은 개별 학생의 정보를 담고 있음

relation은 진짜 데이터가 들어가있는 테이블의 느낌이고, entity는 설계할 때 필요한 개념상의 테이블

애트리뷰트 / 필드 / 열
릴레이션에서 이름을 가진 열

도메인
한 애트리뷰트에 나타날 수 있는 값들의 집합

레코드/튜플/행
릴레이션 각 행

차수
릴레이션에 들어있는 애트리뷰트 수

카디날리티
릴레이션 튜플수

EMPLOYEE ← 릴레이션 이름

기본 키 ↓ EMPNO

열 (애트리뷰트) ↓

행 (투플) →

EMPNO	EMPNAME	TITLE	DNO	SALARY
2106	김창섭	대리	2	2,000,000
3426	박영권	과장	3	2,500,000
3011	이수민	부장	1	3,000,000
1003	조민희	대리	1	2,000,000

카디날리티 (행들의 수)

차수 (열들의 수)

Index

EmpnoIndex	Pointer
1003	
2106	
3011	
3426	

EMPLOYEE

EMPNO	EMPNAME	TITLE	DNO	SALARY
2106	김창섭	대리	2	2,000,000
3426	박영권	과장	3	2,500,000
3011	이수민	부장	1	3,000,000
1003	조민희	대리	1	2,000,000

인덱스
DBMS에서 실제 데이터 레코드를 더 빠르게 찾아가기 위해서, 데이터 레코드를 참조하는 인덱스를 만들기

스키마(Schema)

EMPLOYEE (EMPNO, EMPNAME, TITLE, DNO, SALARY)

데이터베이스의 구조 → 변경이 자주 (발생된다. / 발생되지 않는다.) → 확장성 고려

= 내포(Intension)

상태(State)

EMPLOYEE

EMPNO	EMPNAME	TITLE	DNO	SALARY
2106	김창섭	대리	2	2,000,000
3426	박영권	과장	3	2,500,000
3011	이수민	부장	1	3,000,000
1003	조민희	대리	1	2,000,000

특정 시점의 데이터베이스의 내용 → 변경이 자주 (발생된다. / 발생되지 않는다.)

= 외연(Extension)

Key

문강민

데이터베이스에서 조건에 만족하는 튜플을 찾거나 순서대로 정렬할 때 다른 튜플과 구별될 수 있는 유일한 기준이 되는 속성이다.

Key의 특성

유일성 - 하나의 릴레이션에서 모든 튜플은 서로 다른 키 값을 가져야 한다.
최소성 - 꼭 필요한 최소한의 속성들로만 키를 구성한다.

Super Key

유일성을 만족하는 속성 또는 속성들의 집합

Candidate Key

유일성과 최소성을 만족하는 속성 또는 속성들의 집합

Primary Key

후보 키 중에서 기본적으로 사용하기 위해 선택한 키

Alternate Key

후보 키 중에서 기본키를 제외한 것들

Foreign Key

다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합

학번	주민번호	성명	성별
0001	1111	A	남
0002	2222	B	남
0003	3333	C	여
0004	4444	D	여

학번	과목명
0001	영어
0001	과학
0002	영어
0003	수학
0004	영어
0004	과학

Super key
- 학생 릴레이션에서는 학번, 주민번호, 학번+주민번호, 학번+주민번호+성명 등으로 슈퍼키를 구성 가능.

Candidate Key
- 학생 릴레이션에서는 학번, 주민번호가 후보키를 구성.

Primary Key
- 학생 릴레이션에서는 학번, 주민번호가 기본키가 될 수 있으며, 수강 릴레이션에서는 학번 + 과목명으로 기본키 구성 가능.

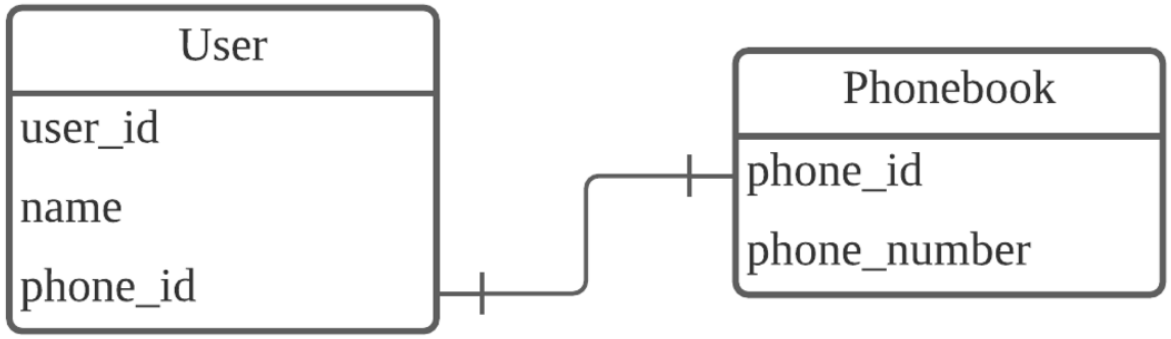
Alternate Key
- 학생 릴레이션에서 학번을 기본키로 정의한다면, 주민번호가 대체키가 될 수 있다.

Foreign Key
- 수강 릴레이션이 학생 릴레이션을 참고하고 있으므로, 학생 릴레이션의 학번은 기본키고, 수강 릴레이션의 학번의 외래키가 된다.

Relation

문강민

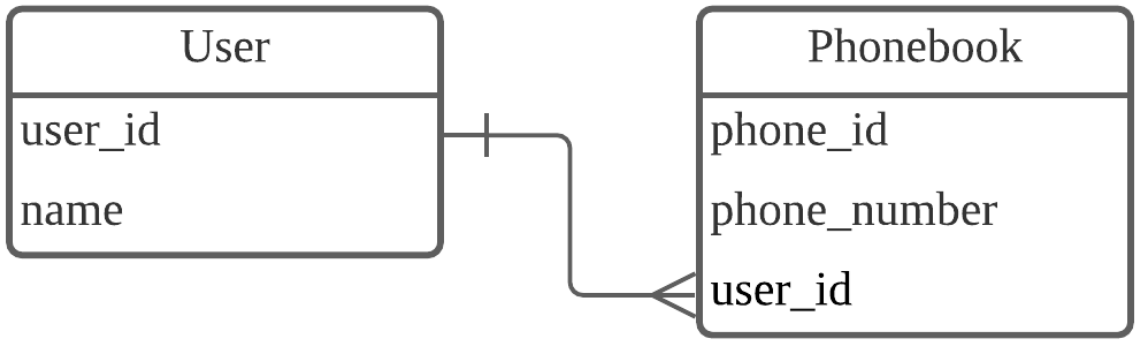
1:1 관계



한 테이블의 한 행이 다른 테이블의 한 행과 연결되는 관계

User table의 phone id 행이 phonebook의 phone id 행과 연결됨을 알 수 있다.

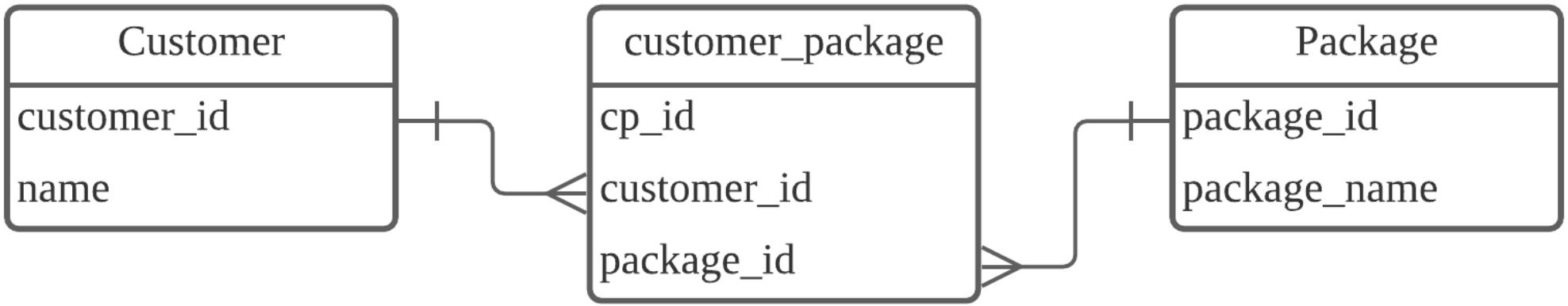
1:N 관계



테이블의 한 행이 다른 테이블의 여러 행과 연결된 관계

한 명의 user는 여러 개의 전화번호를 가질 수 있으나, 여러 명의 유저가 하나의 전화번호를 가질 수는 없는 구조이다.

N:M 관계



한 테이블의 여러 행이 다른 테이블의 여러 행과 연결된 관계

한 명의 고객은 여러 개의 여행 패키지를 구매할 수 있고, 여행 패키지 하나는 여러 명의 고객이 구매할 수 있다.

N:M 관계를 표현 시 새로운 테이블을 하나 더 만들어 1:N 관계 두개로 표현한다.

제1 정규화

테이블의 칼럼이 하나의 값을 가지도록 테이블을 분해하는 것

고객취미들(이름, 취미들)

이름	취미들
김연아	인터넷
추신수	영화, 음악
박세리	음악, 쇼핑
장미란	음악
박지성	게임

고객취미(이름, 취미)

이름	취미
김연아	인터넷
추신수	영화
추신수	음악
박세리	음악
박세리	쇼핑
장미란	음악
박지성	게임

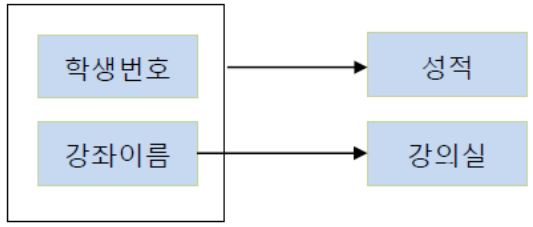
제2 정규화

제1 정규화를 진행한 테이블에 대해 완전 함수 종속을 만족하도록 테이블을 분해하는 것

완전 함수 종속이란 기본키의 부분집합이 결정자가 되어서는 안된다는 것

수강강좌

학생번호	강좌이름	강의실	성적
501	데이터베이스	공학관 110	3.5
401	데이터베이스	공학관 110	4.0
402	스포츠경영학	체육관 103	3.5
502	자료구조	공학관 111	4.0
501	자료구조	공학관 111	3.5

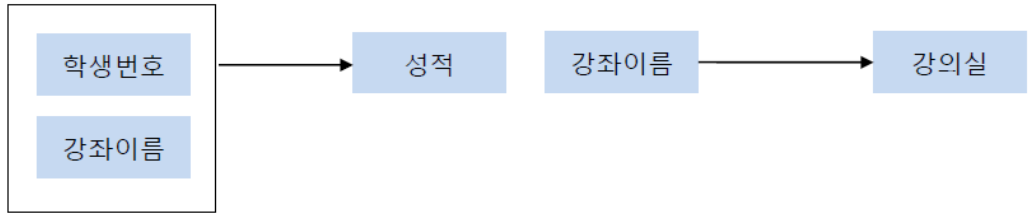


수강

학생번호	강좌이름	성적
501	데이터베이스	3.5
401	데이터베이스	4.0
402	스포츠경영학	3.5
502	자료구조	4.0
501	자료구조	3.5

강의실

강좌이름	강의실
데이터베이스	공학관 110
스포츠경영학	체육관 103
자료구조	공학관 111



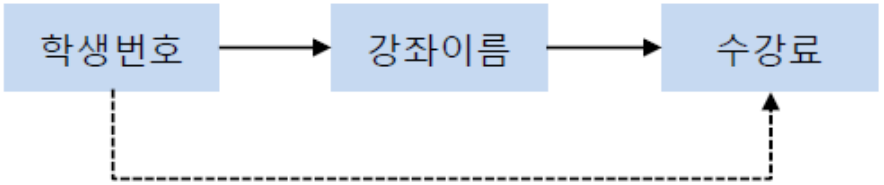
제3 정규화

제2 정규화를 진행한 테이블에 대해 이행적 종속을 없애도록 테이블을 분해하는 것

이행적 종속이란 A -> B, B -> C가 성립할때 A -> C가 성립하는 것을 의미

계절학기

학생번호	강좌이름	수강료
501	데이터베이스	20000
401	데이터베이스	20000
402	스포츠경영학	15000
502	자료구조	25000



계절수강

학생번호	강좌이름
501	데이터베이스
401	데이터베이스
402	스포츠경영학
502	자료구조

수강료

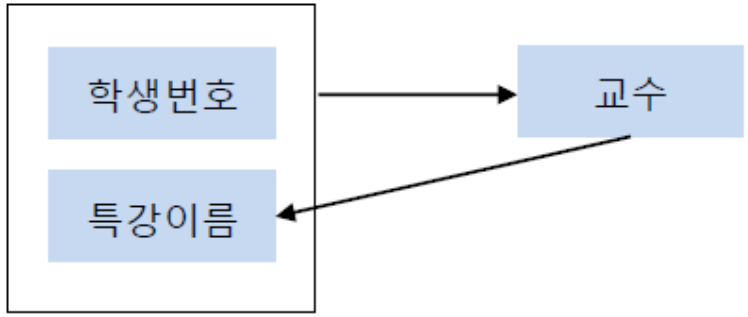
강좌이름	수강료
데이터베이스	20000
스포츠경영학	15000
자료구조	25000

BCNF 정규화

제3 정규화를 진행한 테이블에 대해 모든 결정자가 후보키가 되도록 테이블을 분해하는 것

특강수강

학생번호	특강이름	교수
501	소셜네트워크	김교수
401	소셜네트워크	김교수
402	인간과 동물	승교수
502	창업전략	박교수
501	창업전략	홍교수

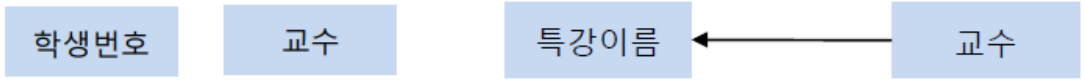


특강신청

학생번호	교수
501	김교수
401	김교수
402	승교수
502	박교수
501	홍교수

특강교수

특강이름	교수
소셜네트워크	김교수
인간과 동물	승교수
창업전략	박교수
창업전략	홍교수



정규화 (Normalize) 릴레이션 분해

학번	이름	학과	학과 전화번호
1111	김00	CSE	111-1111
2222	이00	AI	222-2222
3333	박00	SW	333-3333

→

학번	이름	학과
1111	김00	CSE
2222	이00	AI
3333	박00	SW

학과	학과 전화번호
CSE	111-1111
AI	222-2222
SW	333-3333

Q. 학생 이름으로 학과 전화번호 검색하기 : join을 통해 질의해야함

자주 검색되는 상황

정규화	릴레이션 분해	Join
역정규화	릴레이션 통합	관리비용

RDBMS

김소명

Relational Data-Base Management System : 관계형 데이터베이스 관리 시스템

관계형 데이터베이스 : 데이터를 테이블(표) 형식으로 저장 + 데이터 간의 관계를 정의하는 데이터베이스

- 테이블 기반 구조 : 각 테이블은 특정 엔티티를 나타내고, 각 행은 개별 인스턴스를 나타냄
- 관계성 : 테이블 간의 관계를 정의(1:1, 1:M, M:N) -> Primary Key, Foreign Key를 통해 설정
- SQL(Structured Query Language) : 데이터를 관리하는 언어. 검색, 삽입, 수정, 삭제
- 트랜잭션 처리에서 ACID를 보장
  - Atomicity(원자성) : 트랜잭션은 모두 성공하거나 모두 실패해야 함
  - Consistency(일관성) : 트랜잭션이 완료되면 데이터베이스는 일관된 상태여야 함
  - Isolation(고립성) : 동시에 실행되는 트랜잭션은 서로 간섭하지 않아야 함
  - Durability(지속성) : 트랜잭션이 완료되면 그 결과는 영구적으로 반영되어야 함

데이터 무결성, 정규화, …

MySQL : 오픈소스 RDBMS로, 대규모 웹 애플리케이션에서 많이 사용
Oracle : 상용 RDBMS로, 대규모 기업에서 많이 사용. 높은 안정성과 확장성
PostgreSQL : MySQL보다 많은 기능을 제공. 대용량 처리에 적합

장점
- 데이터의 중복을 줄이고, 일관성을 유지할 수 있음
- SQL을 통해 복잡한 쿼리와 데이터 조작이 가능, 표준화된 인터페이스 제공
- ACID 속성을 통해 트랜잭션 처리의 신뢰성과 안정성을 보장

단점
- 고정된 스키마 사용 -> 스키마 변경이 어렵고 유연성이 떨어짐
- 수직 확장을 통해 성능을 높이면 하드웨어 비용이 증가

NoSQL

김소명

Not Only SQL : 비관계형 데이터베이스

비관계형 데이터베이스 : 관계형 데이터베이스의 한계를 극복하고 다양한 모델을 지원하기 위해 등장한 모든 데이터베이스 시스템을 포괄

- 비관계형 구조 : 테이블 뿐만 아니라 Document, Key-Value, Column-Family, Graph 등을 지원
- 스키마 유연성 : 고정된 스키마를 요구하지 않아서 데이터 구조를 유연하게 변경하고, 다양한 형식의 데이터를 저장 가능
- 수평 확장성 : 여러 대의 서버에 데이터를 분산하여 처리할 수 있음(Scale-Out)
- CAP 이론 : Consistency(일관성), Availability(가용성), Partition-Tolerance(파티션 허용성) 중 두가지 특성을 보장해야 함.
  CP 시스템 : 데이터의 정확성을 유지하면서 네트워크가 복구될 때까지 일부 요청에 대한 응답 포기 (MongoDB)
  AP 시스템 : 네트워크가 분할된 상태에서도 시스템은 계속 응답하지만, 데이터는 일시적으로 일관되지 않을 수 있음 (Cassandra)
  CA 시스템 : 네트워크 분할을 피할 수 없어 현실적으로 어려움

소셜 미디어, 로그 데이터, IoT 등에서 생성되는 빅데이터를 효율적으로 관리할 수 있음. 대규모 데이터 처리에 적합

MongoDB : 문서지향 DB, JSON 형식으로 데이터를 저장. 복잡한 계층적 데이터를 저장할 때 유용. 웹 애플리케이션에서 많이 사용
Cassandra, HBase : Column 기반, 특정 Column에 대핸 빠른 조회가 가능. 분산 환경에서 높은 가용성과 확장성을 제공
Redis : Key-Value, 간단한 구조로 속도가 매우 빠름. 캐시나 세션 관리에 자주 사용
Neo4j : Graph, Node-Edge 기반의 복잡한 관계(네트워크, 추천 알고리즘 등)를 효율적으로 탐색하고 분석하는데 적합

장점
- 관계형 데이터베이스보다 더 빠른 읽기/쓰기 성능 제공하는 경우가 많고, 다양한 데이터를 저장하고 관리할 수 있음

단점
- SQL처럼 복잡한 쿼리를 지원하지 않는 경우가 많음
- 정규화 작업을 거치지 않아 데이터 중복이 발생, 관리 복잡성을 증가시킬 수 있음

NoSQL vs RDBMS

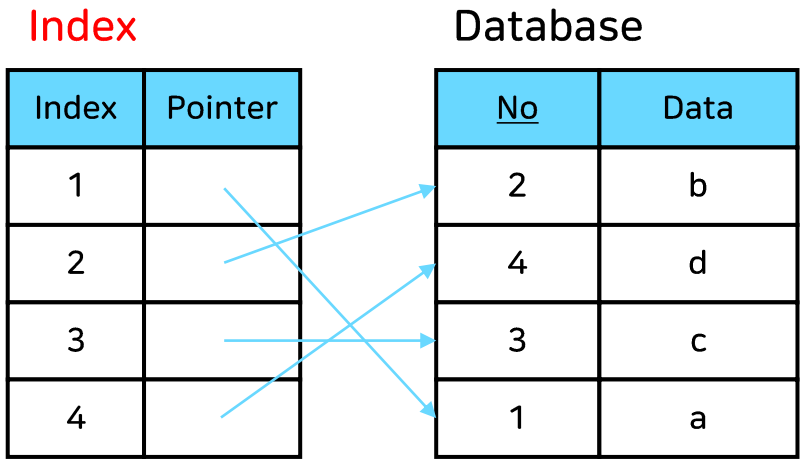
특성	NoSQL	RDBMS
데이터 모델	비정형 또는 반정형(문서, 키-값 등)	정형 (테이블 기반)
스키마	유연함 (스키마 없음 또는 동적 스키마)	고정된 스키마
확장성	수평 확장 (Scale-out)	주로 수직 확장 (Scale-up)
일관성	CAP 이론에 따라 일관성을 희생할 수 있음	ACID 속성을 보장
쿼리 언어	SQL이 아닌 전용 API 또는 언어 사용	SQL 표준 언어 사용
사용 사례	빅데이터, 실시간 분석	전통적인 트랙잭션 기반 애플리케이션

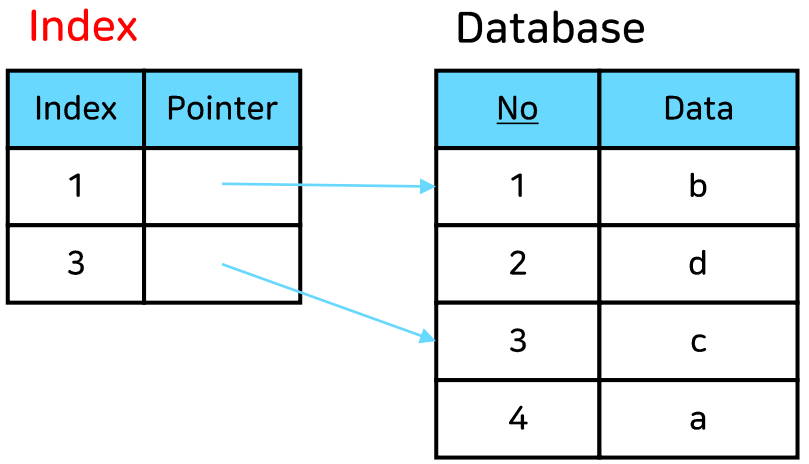
데이터베이스의 특징

- (효율적인) 데이터 접근
- 데이터 일관성
- 백업과 회복
- 데이터 무결성
- 질의어
- 다양한 인터페이스
- 데이터 보호/보안

효율적인 데이터 접근

단일 단계 인덱스

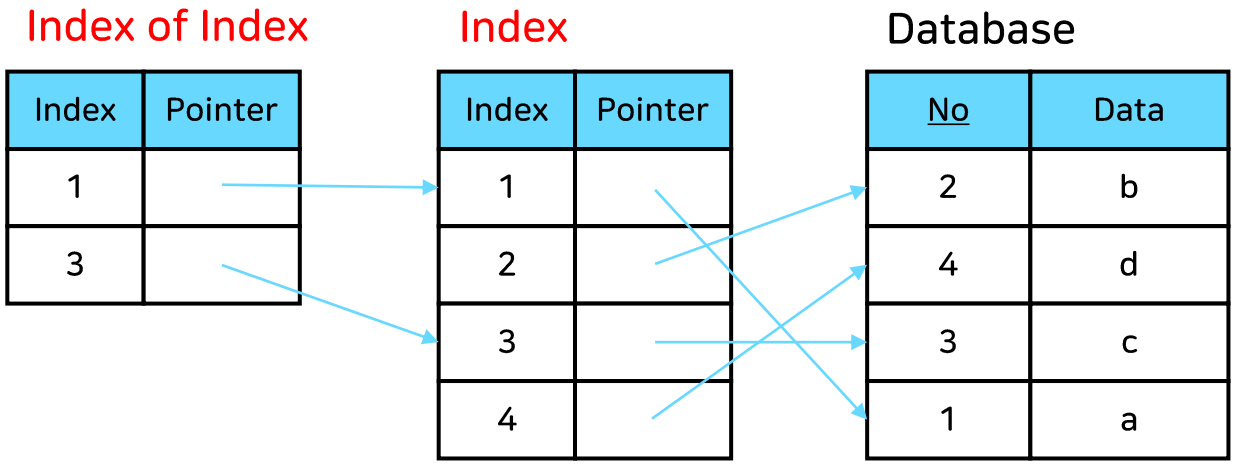




기본 인덱스
보조 인덱스

밀집 인덱스
희소 인덱스

다단계 인덱스

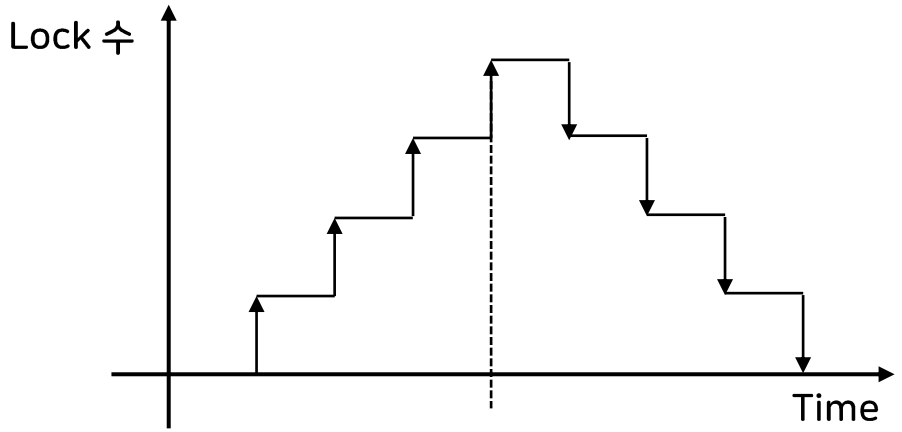


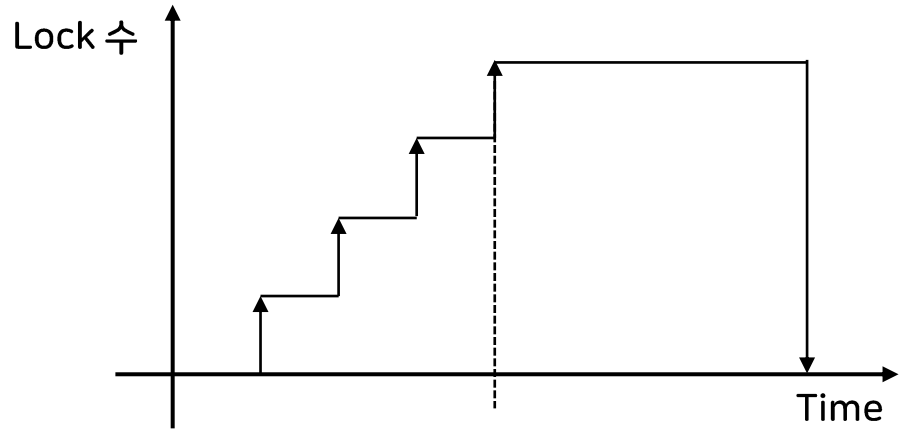
데이터 일관성

데이터를 동시에 접근 → 동시성 제어

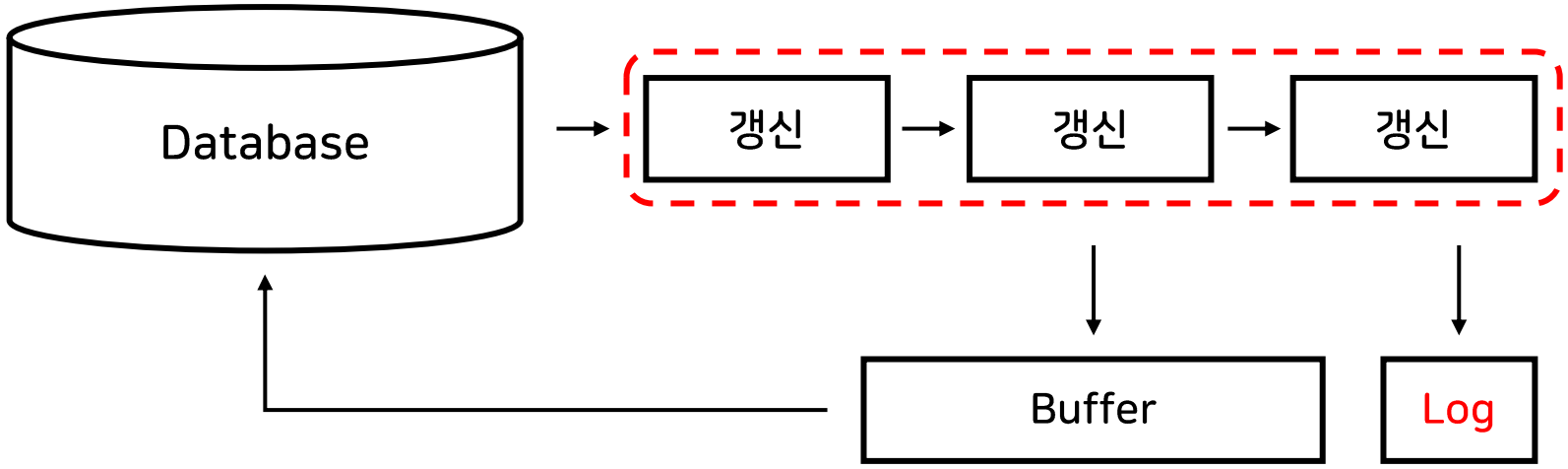
Locking → Deadlock 발생 가능

2단계 Locking : 확장 단계(Lock Only) + Lock Point + 수축 단계(Unlock Only)





백업과 회복



Redo(재수행) : 트랜잭션 완료 but 미반영

Undo(재수행) : 트랜잭션 미완료 but 반영

무결성

데이터의 정확성 / 유효성 검사 (문제가 없는지)

ex) 기본키에 null 값이 들어감

DBMS가 자동으로 검사

도메인 제약조건 (Domain Constraint)

애트리뷰트가 원자값이고 type에 맞는지

키 제약조건 (Key Constraint)

키 애트리뷰트는 UNIQUE 해야함

기본 키와 엔티티 무결성 제약조건 (Entity Integrity Constraint)

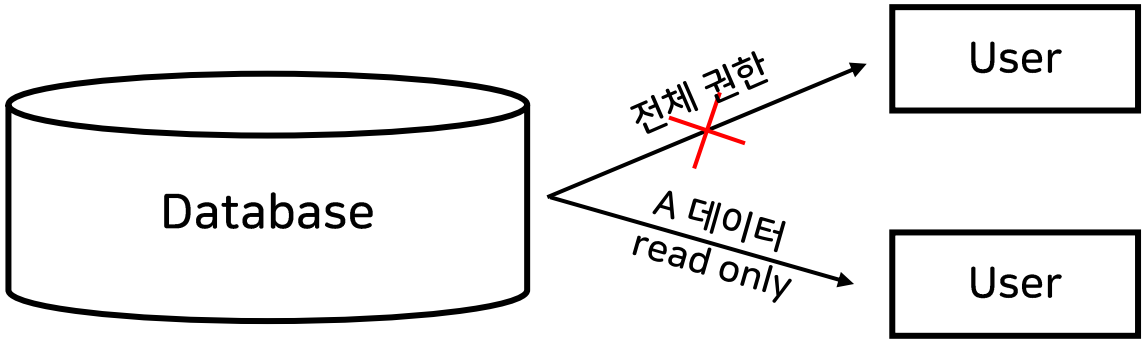
기본 키는 NULL값을 가질 수 없음

외래 키와 참조 무결성 제약조건 (Referential Integrity Constraint)

연관된 튜플 사이의 일관성 유지

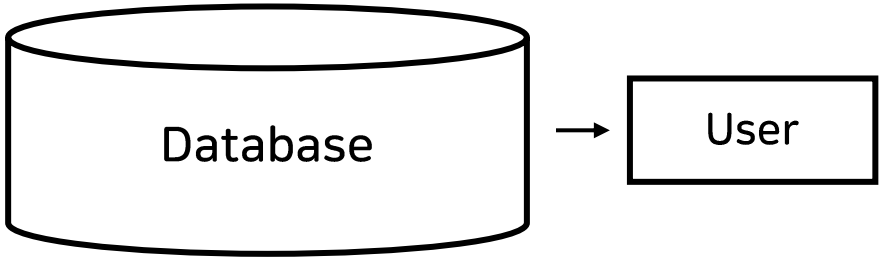
데이터 보호/보안

권한을 분리해서 최소한으로 줄 것



View : 가상 릴레이션

Data_2가 10인 경우만 접근가능



No	Data_1	Data_2
1	a	10
2	d	20
3	c	10
4	d	15

User : No가 3 미만인 것.

DBMS : 자동적으로 질의를 변환
return (1, a, 10)

Task

Desing Pattern / Security 문제 풀기 (upload : 11/22 (금) ~ deadline : 11/26 (화))

Real Problem Solving 문제 풀기 (deadline : 12/04 (화))

Curriculum

1주차 : Orientation	(9/4)
2주차 : Data Structure	(9/11)
3주차 : Algorithm 1	(9/25)
4주차 : Algorithm 2	(10/2)
5주차 : Computer Architecture	(10/16)
6주차 : Operating System	(11/6)
7주차 : Computer Network	(11/13)
8주차 : DataBase	(11/20)
9주차 : Design Pattern & Security	(11/27)
10주차 : Real Problems	(12/4)