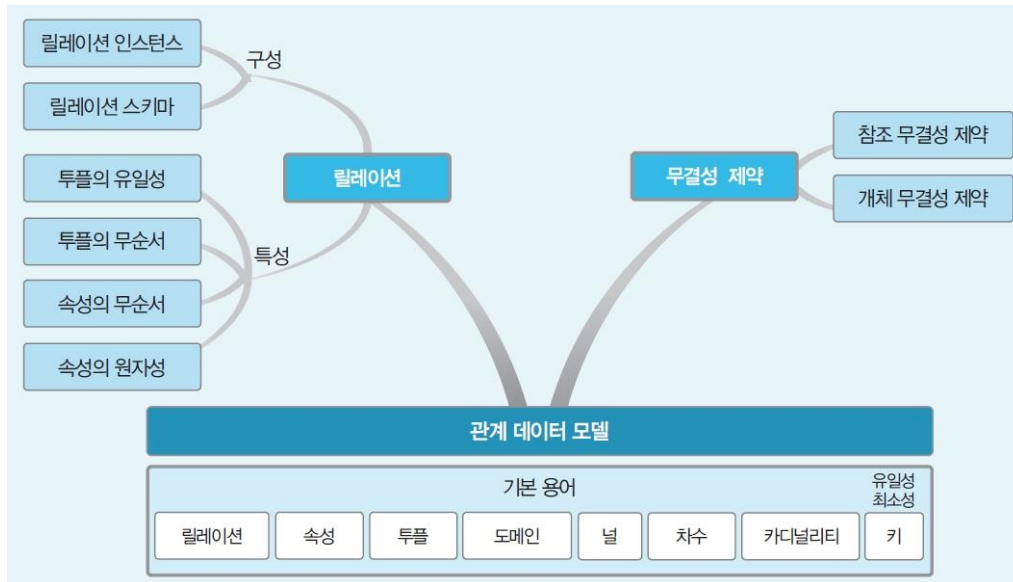


## 데이터베이스 실습 3주차 강의



릴레이션 스키마를 만들어야 한다.

인스턴스 -> 메모리 (저장된 값)

관계형 데이터베이스 -> 테이블 형식으로 저장되는 데이터

데이터베이스 -> 트리 구조

### 중간고사! 관계 데이터 모델 기본용어 정의

릴레이션=테이블 : 데이터를 2차원 테이블의 구조로 저장한 것

속성=필드(열) [ex] 학번] → 속성의 무순서, 원자성

튜플=레코드(행) → 튜플의 유일성, 무순서

도메인 : 하나의 속성이 가질 수 있는 모든 값의 집합 [ex] int, char → 데이터 타입]

널 : 아직 값이 정해지지 않았다.

차수 : 필드의 개수(열)

카디널리티 : 저장된 값의 개수(행)

키 : 릴레이션에서 튜플들을 유일하게 구별하는 속성 또는 속성들의 집합

열(속성, 애트리뷰트)					
고객아이디	고객이름	나이	등급	직업	적립금
CHAR(20)	CHAR(20)	INT	CHAR(10)	CHAR(10)	INT
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	학생	0

도메인

행(튜플)

그림 5-1 릴레이션 예 : 고객 릴레이션

키 : 고객아이디

차수 : 6개

카디널리티 : 4개

## 릴레이션의 구성

중간고사! 릴레이션 스키마, 인스턴스의 정의, 특징

릴레이션 스키마 : 릴레이션의 논리적 구조

특징 → 자주 변하지 않는 정적인 특징

릴레이션 인스턴스 : 어느 한 시점에 릴레이션에 존재하는 튜플(행)들의 집합

특징 → 튜플의 삽입·삭제·수정이 자주 발생하는 동적인 특징

고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	학생	0

릴레이션 스키마

릴레이션 인스턴스

그림 5-2 릴레이션 구성 예 : 고객 릴레이션

## 데이터베이스의 구성

데이터베이스 스키마 : 데이터베이스의 전체 구조 (폴더의 구조)

데이터베이스 인스턴스 : 데이터베이스를 구성하는 릴레이션 인스턴스의 모음

### 중간고사! 키

키 : 튜플들을 유일하게 구별하는 속성(=필드) 또는 속성들의 집합

→ 단일 속성 사용(복합속성이 불가능한 것은 아님)

키의 특성 : 유일성, 최소성

레코드의 유일하게 구별 -->

슈퍼키(super key) : 유일성을 만족

후보키(candidate key) : 유일성과 최소성을 만족 [선택할 수 있는 거]

기본키(primary key) : 후보키 중에서 최종적으로 선택한 키

대체키(alternate key) : 기본키로 선택되지 못한 후보키

외래키(foreign key) : 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합

외래키는 널값을 가지는 것이 가능하다!



그림 5-9 고객 릴레이션과 주문 릴레이션의 스키마

주문고객이 외래키

## 키의 특성과 종류

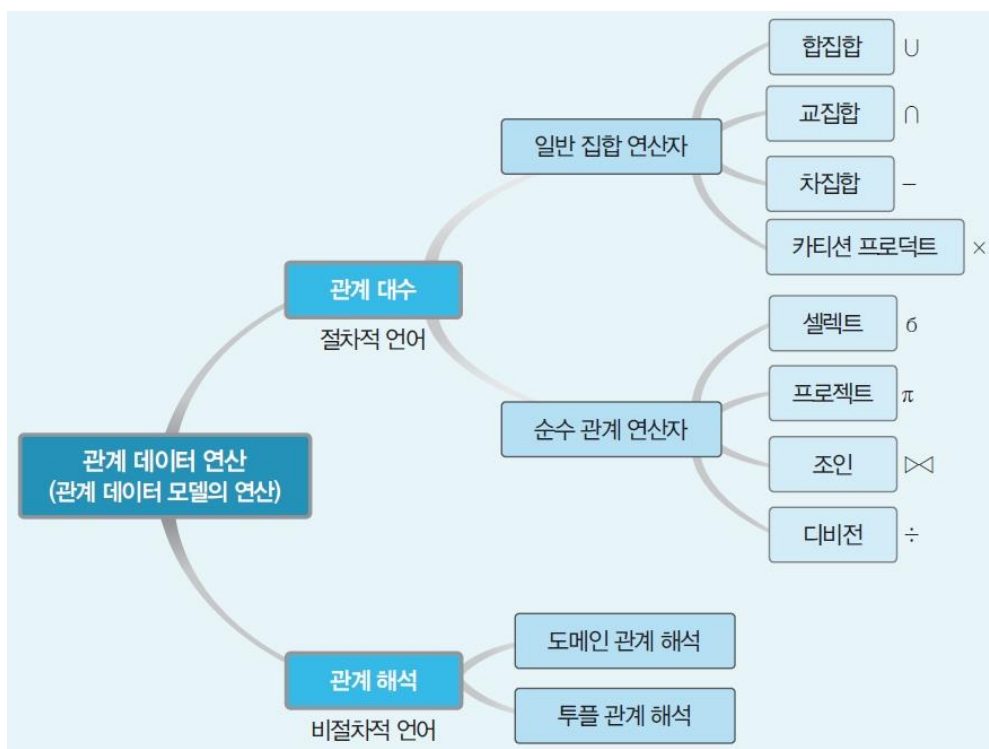
특성	<ul style="list-style-type: none"> <li>• 유일성 : 한 릴레이션에서 모든 튜플은 서로 다른 키 값을 가져야 함</li> <li>• 최소성 : 꼭 필요한 최소한의 속성들로만 키를 구성</li> </ul>
종류	<ul style="list-style-type: none"> <li>• 수퍼키 : 유일성을 만족하는 속성 또는 속성들의 집합</li> <li>• 후보키 : 유일성과 최소성을 만족하는 속성 또는 속성들의 집합</li> <li>• 기본키 : 후보키 중에서 기본적으로 사용하기 위해 선택한 키</li> <li>• 대체키 : 기본키로 선택되지 못한 후보키</li> <li>• 외래키 : 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합</li> </ul>

무결성 제약조건 : 데이터의 무결성을 보장하고 일관된 상태로 유지하기 위한 규칙

개체 무결성 제약조건 : 기본키를 구성하는 모든 속성은 널 값을 가질 수 없는 규칙

참조 무결성 제약조건 : 외래키는 참조할 수 없는 값을 가질 수 없는 규칙

## 관계 데이터 연산



관계 데이터 연산

관계 대수 : 데이터의 처리 과정을 순서대로 기술 (절차 언어)

관계 해석 : 원하는 데이터가 무엇인지만 기술 (비절차 언어)

## !중간고사! 관계 대수의 연산자 특징

### 일반 집합 연산자

연산자	기호	표현	의미
합집합	$\cup$	$R \cup S$	릴레이션 R과 S의 합집합을 반환
교집합	$\cap$	$R \cap S$	릴레이션 R과 S의 교집합을 반환
차집합	$-$	$R - S$	릴레이션 R과 S의 차집합을 반환
카티션 프로덕트	$\times$	$R \times S$	릴레이션 R의 각 튜플과 릴레이션 S의 각 튜플을 모두 연결하여 만든 새로운 튜플을 반환

합집합, 교집합, 차집합 -> 합병조건이 필요

**합병조건** : 테이블 구조가 같아야한다. 도메인이 같아야 한다.

고객 릴레이션

고객번호	고객이름	나이
INT	CHAR(20)	INT
100	정소화	20
200	김선우	35
300	고명석	24

직원 릴레이션

직원번호	직원이름	직위
INT	CHAR(20)	CHAR(20)
10	김용욱	부장
20	채광주	과장
30	김수진	대리

그림 6-6 합병이 불가능한 예

고객 릴레이션

고객번호	고객이름	나이
INT	CHAR(20)	INT
100	정소화	20
200	김선우	35
300	고명석	24

직원 릴레이션

직원번호	직원이름	나이
INT	CHAR(20)	INT
10	김용욱	40
20	채광주	32
30	김수진	28

그림 6-7 합병이 가능한 예

합집합(union)  $R \cup S$  : 카디널리티는 릴레이션 R과 S의 카디널리티를 더한 것과 같거나 적어짐

교집합(intersection)  $R \cap S$  : 공통적으로 가지고 있는 레코드를 결과로 구성

카디널리티는 릴레이션 R과 S의 어떤 카디널리티보다 크지 않음(같거나 적음)

차집합(difference)  $R - S$  :  $R - S$ 의 카디널리티는 릴레이션 R의 카디널리티와 같거나 적음

$S - R$ 의 카디널리티는 릴레이션 S의 카디널리티와 같거나 적음

카티션 프로덕트(cartesian product)  $R \times S$  : R에 속한 각 튜플과 릴레이션 S에 속한 각 튜플을 모두 연결

**! 중간고사 ! 관계대수는 정의와 특징만 나옴, 코딩X**

순수 관계 연산자

연산자	기호	표현	의미
선택	$\sigma$	$\sigma_{조건}(R)$	릴레이션 R에서 조건을 만족하는 튜플들을 반환
프로젝트	$\pi$	$\pi_{속성리스트}(R)$	릴레이션 R에서 주어진 속성들의 값으로만 구성된 튜플들을 반환
조인	$\bowtie$	$R \bowtie S$	공통 속성을 이용해 릴레이션 R과 S의 튜플들을 연결하여 만든 새로운 튜플들을 반환
디비전	$\div$	$R \div S$	릴레이션 S의 모든 튜플과 관련이 있는 릴레이션 R의 튜플들을 반환

선택(select) : 테이블에서 조건을 만족하는 튜플(행)을 반환해주는 명령어

프로젝트(project) : 테이블에서 선택한 속성의 값으로 결과를 반환

수직적 연산자 : 결과 릴레이션은 연산 대상 릴레이션의 수직적 부분집합