

# 20240501 수업 정리

## 데이터 모델링 기본 개념과 관계 데이터 모델에 대한 설명

- 도메인과 제약 조건을 설정하여 데이터 모델링을 시작하고, 릴레이션에 대한 이해가 중요

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

- 데이터 모델의 논리적 표현에서 **릴레이션(테이블)**은 **고객 정보**와 **속성(attribute)**을 담고 있다.
- 속성들은 컬럼으로 표현되며, 이에 대한 **값(튜플)**이 행으로 입력된다.
- 데이터 모델의 개념을 이해하면, **고객 릴레이션**을 통해 **속성**을 이해할 수 있다.

## 데이터베이스 속성과 도메인의 개념

- 도메인은 속성 값의 집합과 특성을 고려한 데이터 타입으로 정의된다.
- 도메인은 입력 수정 시 적합성 판단의 기준이 되며, 속성 값의 특성을 정의하여 그 값들이 가질 수 있는 조건을 설정한다.
- 고객 아이디를 예시로 도메인 개념을 설명하면, 고객 아이디의 속성 값은 길이, 유니크 여부, 중복 여부 등의 제약 조건을 가지며, 해당 속성의 집합을 도메인으로 정의한다.
- 도메인을 이해하면 속성의 값들이 가질 수 있는 조건을 결정하여 데이터 타입을 정의할 수 있게 된다.
- 도메인은 제약과 정의의 중요성이 강조되며, 값을 살펴보고 세분화할 때의 가이드라인이 된다.

## 데이터 테이블 관련 용어 설명

- 널(null)
  - 속성 값을 아직 모르거나 해당되는 값이 없음을 표현
- 차수(degree)
  - 하나의 릴레이션에서 속성의 전체 개수
- 카디널리티(cardinality)
  - 하나의 릴레이션에서 튜플의 전체 개수

cf)

- 0을 **null**로 표현해도 무방하거나 도메인에 따라 0으로 표기할 수도 있음.

- **null**로 연산 불가능하므로 가끔 데이터 타입 문제로 0으로 대체될 수 있음.
- 데이터를 확인하고 차수는 릴레이션에 적어도 하나 이상 유지되어야 함.  
릴레이션에는 최소한 하나의 차수가 필요함.
- **차수**는 속성 수 및 레코드의 개수와 관련되며, 예를 들어 6차수를 가질 경우 6개의 컬럼을 가진다.
- 데이터 차수는 변하지 않지만, 데이터 마트나 웨어하우스에서 생성 시 변경 가능성이 존재하며, 비즈니스적 맥락에 따라 달라짐.
- 카디널리티는 릴레이션 내에 튜플 연결 수를 의미하며,  
행 자체가 없는 경우도 발생할 수 있고 값의 존재 여부를 나타냄.
- 고객 데이터에서 미입력 상태나 다른 테이블에서 값이 없는 경우, 카디널리티의 변화와 **null** 발생 가능.

## 릴레이션의 구성

- 릴레이션 논리적 구조, 스키마와 인스턴스 개념 등을 설명.
- 스키마는 속성 이름과 값의 정의를 포함하며, 인스턴스는 실제 데이터 집합.
- 스키마는 데이터 속성 필요성을 정리하는데 사용되고, 관계 데이터모델의 중심 부분을 이룸.
- 인스턴스는 데이터베이스가 어떻게 구성되어 있는지 보여주며, 관계 데이터모델의 기본 개념을 포함함.

cf) 데이터베이스 구성 예시



그림 5-3 데이터베이스 구성 예 : 인터넷 쇼핑몰 데이터베이스

- 고객, 주문, 임직원, 상품과 같은 다양한 릴레이션들이 결합되어 데이터베이스가 형성된다.
- 데이터베이스의 구성은 고객, 상품, 주문 릴레이션들의 스키마로 이루어진다.
- 고객의 릴레이션 스키마에는 고객 아이디, 이름, 번호 등이 포함되고,  
상품의 릴레이션 스키마는 상품 번호, 명칭, 원자재 등으로 정의된다.
- 위의 릴레이션들이 데이터베이스 스키마를 형성하며,  
데이터베이스 스키마는 모든 릴레이션들의 스키마를 포함한다.

## 릴레이션의 특성

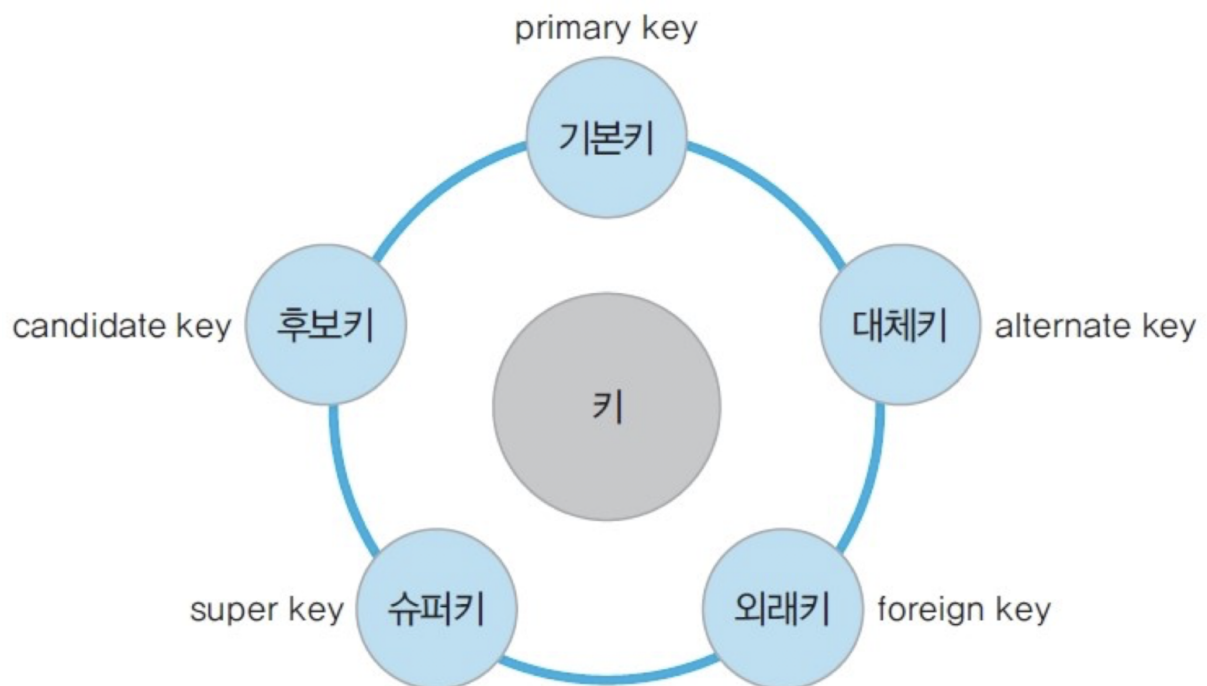
- 튜플의 유일성 : 하나의 릴레이션에는 동일한 튜플이 존재할 수 없다.
- 튜플의 무순서 : 하나의 릴레이션에서 튜플 사이의 순서는 무의미하다.
- 속성의 무순서 : 하나의 릴레이션에서 속성 사이의 순서는 무의미하다. (속성은 단순히 값이 정해진 것)
- 속성의 원자성 : 속성 값으로 원자 값만 사용할 수 있다.

cf) 쿼리 실행 중 데이터 중복에 대한 주의사항

- 쿼리 실행 시 중복된 값을 피해야하며, 조인 시 중요한 전제 조건은 중복된 값이 없어야 한다.
- 데이터베이스의 각 튜플은 중복되지 않아야 한다.
- 튜플 전체에 중복된 값이 발생하지 않도록 조심한다.
- 순서는 랜덤하게 배치되므로 데이터 중복 회피 및 키에 대한 이해가 필요하다.

## 키(key)

: 릴레이션에서 튜플들을 유일하게 구별하는 속성 또는 속성들의 집합

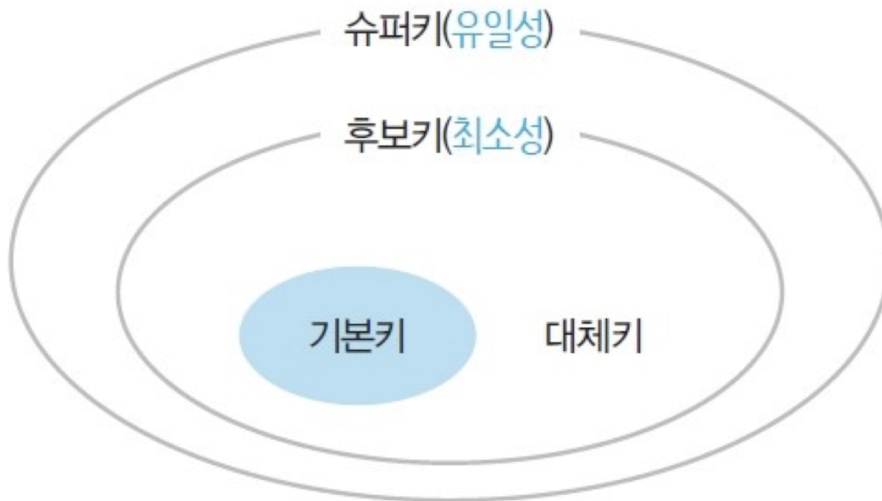


- 키를 통해 데이터를 **구분하고 식별**하여 중복과 잘못된 데이터를 방지한다.
- 고객 아이디는 도메인에 따라 중복될 수 있지만, **유일키(Primary Key)**로 고객을 식별할 수 있다.
- **고유 속성인 키**는 데이터를 의미 있게 만들고 서로 다른 것을 구별한다.
- SQL에서 조인 시 **유일키**를 기준으로 릴레이션을 연결

## 키의 특성

- 유일성(uniqueness) : 하나의 릴레이션에서 모든 튜플은 서로 다른 키 값을 가져야 한다.
- 최소성(minimality) : 꼭 필요한 최소한의 속성들로만 키를 구성함

## 키의 종류



- Super Key(슈퍼키)
  - 유일성을 가진 컬럼 또는 그 집합 ex) 고객아이디, (고객아이디, 고객이름), (고객이름, 주소) 등
  - 우리는 고객에 대해 고객 아이디만으로 식별이 된다고 맹신해서는 안 된다. → 고객 이름, 주소 등도 고려
  - 딱 한 개의 식별 키가 없는 경우도 안 되며, 최소한 하나의 식별 키가 필요하다.
- Candidate Key(후보키)
  - 유일성과 최소성을 만족하는 속성 또는 속성들의 집합 ex) 고객아이디(고객이름, 주소) 등
- Primary Key(기본키)
  - 후보키 중에서 기본적으로 사용하기 위해 선택한 키 ex) 고객아이디
- Alternate Key(대체키)
  - 기본키로 선택되지 못한 후보키 ex) (고객이름, 주소)