

1. 개념적 모델링 , 논리적 모델링 , 물리적 모델링

데이터베이스 설계는 크게 세 단계로 나눌 수 있으며, 각 단계는 데이터베이스의 추상화 수준을 나타냅니다.

1) 개념적 모델링

이 단계에서는 시스템이 무엇을 해야 하는지(What the system should do)에 대한 고차원적인 이해를 구축합니다. 여기서는 데이터의 주요 엔티티(Entity)와 그 엔티티들 간의 관계(Relationship)를 식별합니다. 결과물은 일반적으로 엔티티-관계 다이어그램(ERD)입니다.

2) 논리적 모델링

개념적 모델을 바탕으로 데이터 모델을 좀 더 구체화하여, 어떻게 데이터베이스 시스템 내에서 구현될 것인지(How the system should be implemented)를 결정합니다. 데이터 타입, 기본 키, 외래 키 및 기타 제약 조건을 포함하는 논리적 스키마가 이 단계의 산출물입니다.

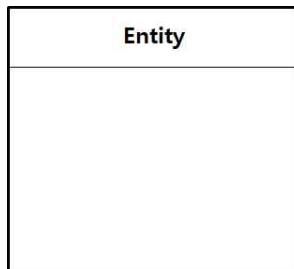
3) 물리적 모델링

물리적 모델링은 논리적 모델을 실제 데이터베이스 스키마로 변환하는 단계입니다. 이 단계에서는 논리적 모델을 구체적인 데이터베이스 시스템의 구현으로 변환합니다. 즉, 실제로 데이터를 저장, 검색, 수정할 수 있는 형태로 만듭니다.

2. ERD

ERD(Entity-Relationship Diagram, 엔티티-관계 다이어그램)는 데이터베이스의 구조를 시각화하는 도구입니다. ERD를 통해 데이터베이스에 저장되는 정보(엔티티), 그 정보들 간의 관계, 그리고 각 정보의 속성을 나타냅니다.

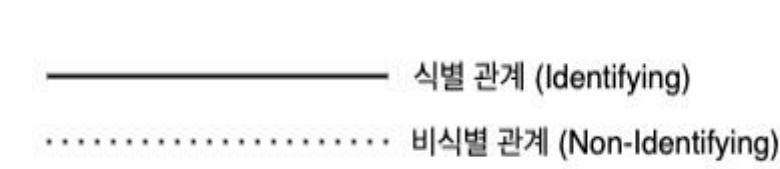
엔티티(Entity): 데이터베이스에 저장되는 항목이나 객체를 말합니다. 예를 들어, '고객', '주문', '상품' 등이 엔티티가 될 수 있습니다. ERD에서 엔티티는 보통 사각형으로 표시됩니다.



속성(Attribute): 엔티티의 특성이나 세부 사항입니다. 예를 들어, '고객' 엔티티의 속성으로는 '고객명', '주소', '전화번호' 등이 있을 수 있습니다. 속성은 엔티티 내부에 원이나 타원으로 표시됩니다.

Entity
PK
고객명
주소
전화번호

관계(Relationship): 엔티티 간의 연결을 나타냅니다. 예를 들어, '고객'과 '주문' 사이에는 '주문한다'는 관계가 있을 수 있습니다. 관계는 엔티티들을 연결하는 선으로 표시되며, 관계의 이름을 선 위에 적어 나타내기도 합니다.









- 식별 관계(Identifying Relationship)

식별 관계에서는 하나의 엔티티가 다른 엔티티의 존재를 식별하는데 필수적입니다. 즉, 자식 엔티티의 기본 키가 부모 엔티티의 기본 키를 포함하게 됩니다. 이런 유형의 관계에서 자식 엔티티는 부모 엔티티 없이는 존재할 수 없으며, 부모 엔티티의 키가 자식 엔티티의 기본 키의 일부가 됩니다.

- 비식별 관계(Non-Identifying Relationship)

비식별 관계에서는 자식 엔티티가 부모 엔티티와 관계를 맺고 있지만, 부모 엔티티의 기본 키를 자식 엔티티의 기본 키에 포함하지 않습니다. 이 경우, 부모 엔티티의 키는 자식 엔티티에서 외래 키로만 사용됩니다. 자식 엔티티는 부모 엔티티와 독립적으로 존재할 수 있습니다.

카디널리티(Cardinality): 엔티티 간 관계의 수를 나타냅니다. 일대일(1:1), 일대다(1:N), 다대다(M:N)가 일반적인 카디널리티 유형입니다.

	One
	Many
	One (and only one)
	Zero or one
	One or many
	Zero or many



하나의 A는 하나의 B로 구성되어 있다.



하나의 A는 0 또는 하나의 B로 구성되어 있다.



하나의 A는 두 개 이상의 B로 구성되어 있다.



하나의 A는 하나 이상의 B로 구성되어 있다.



하나의 A는 0, 1, 또는 그 이상의 B로 구성되어 있다.