

## 데이터 모델링 ( Data Modeling )

데이터 모델링은 데이터가 데이터베이스 내에서 어떻게 저장되고 조직될지를 계획하고 시각화하는 과정이다. 마치 건물을 설계하려는 건축가가 건축도면을 작성하는 것과 비슷하다. 건축도면은 건물이 실제로 지어지기 전에 어떻게 생겼는지, 어떤 재료를 사용할지, 각 방의 기능은 무엇인지 등을 구체적으로 계획하고 시각화하는 데 도움을 준다.

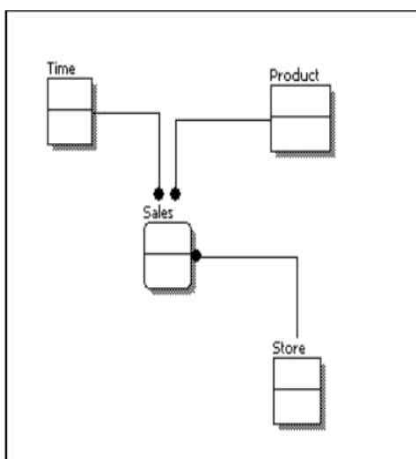
그러므로 데이터 모델링은 비즈니스 팀과 기술 팀 간의 요구 사항을 전달하므로 비즈니스 맥락에서 측정항목과 데이터를 나타내는 훌륭한 도구가 된다. 비즈니스 요구 사항을 정의하는 담당자는 시각적 표현을 사용하여 데이터의 구조와 내용을 쉽게 표현할 수 있다. 이러한 요소는 애플리케이션에 포함되는 방법과 데이터베이스에서 처리, 저장 및 관리되는 방법을 결정한다.

또한 구조화된 데이터 저장, 데이터의 일관성과 정확성 보장, 성능 최적화, 복잡성 관리, 재사용성과 유지보수성 향상, 응용 프로그램 개발 지원, 비즈니스 요구 사항 이해, 데이터베이스의 확장성, 위험 감소, 규정 준수 등의 다양한 이점을 활용할 수 있어 현대 애플리케이션 개발에 필수적으로 사용되어지고 있다.

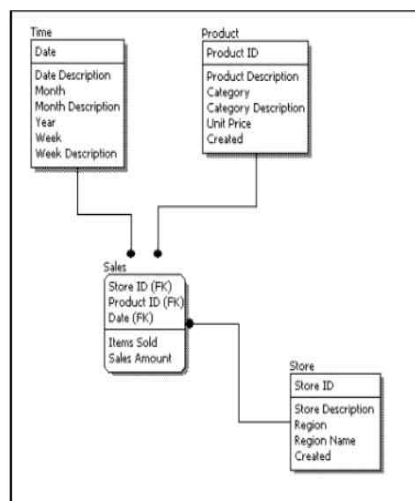
### 1. 개념적 모델링, 논리적 모델링, 물리적 모델링

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

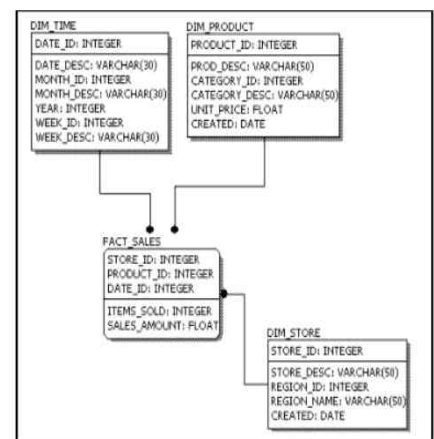
#### Conceptual Model Design



#### Logical Model Design



#### Physical Model Design



이미지출처) <https://i.stack.imgur.com/GYDcP.png>

#### 1) 개념적 모델링 (Conceptual Modeling)

개념적 데이터 모델은 데이터 모델의 가장 추상적인 형태이며 이 단계에서는 시스템이 무엇을 해야 하는지 (What the system should do)를 결정한다. 세부적인 데이터 유형이나 기술과 같은 내용을 조사하지 않고도 애플리케이션이 어떻게 작동하는지 혹은 비즈니스 요구사항을 충족하는지 확인할 수 있다.

## 2) 논리적 모델링 (Logical Modeling)

개념적 모델을 바탕으로 데이터 모델을 좀 더 구체화하여 어떻게 데이터베이스 시스템 내에서 구현될 것인지 (How the system should be implemented)를 결정한다. 아직은 어떤 기술에도 연결되어 있지 않지만 데이터 타입, 기본 키, 외래 키 및 기타 제약 조건을 포함하는 논리적 요소가 포함되어 있다. 기술 팀은 논리적 데이터 모델을 사용하여 비즈니스 요구 사항을 애플리케이션 및 데이터베이스 설계로 변환한다.

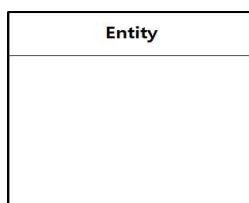
## 3) 물리적 모델링 (Physical Modeling)

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

## 2. ERD

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

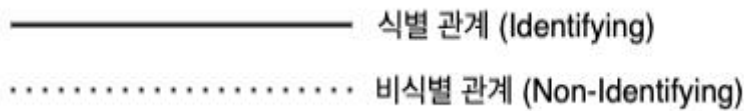
1) 엔티티(Entity): 데이터베이스에 저장되는 항목이나 객체를 말한다. 예를 들어, '고객', '주문', '상품' 등이 엔티티가 될 수 있다. ERD에서 엔티티는 보통 사각형으로 표시된다.



2-2) 속성(Attribute): 엔티티의 특성이거나 세부 사항이다. 예를 들어, '고객' 엔티티의 속성으로는 '고객명', '주소', '전화번호' 등이 있을 수 있다. 속성은 엔티티 내부에 원이나 타원으로 표시된다.

Entity
PK
고객명
주소
전화번호

2-3) **관계(Relationship)**: 엔티티 간의 연결을 나타낸다. 예를 들어, '고객'과 '주문' 사이에는 '주문한다'는 관계가 있을 수 있다. 관계는 엔티티들을 연결하는 선으로 표시되며, 관계의 이름을 선 위에 적어 나타내기도 한다.



- 식별 관계(Identifying Relationship)

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

- 비식별 관계(Non-Identifying Relationship)

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

4) **카디널리티(Cardinality)**: 엔티티 간 관계의 수를 나타낸다. 일대일(1:1), 일대다(1:N), 다대다(M:N)가 일반적인 카디널리티 유형이다.

