

Chapter 06

모델링 실습



데이터베이스 응용

□ 모델링 실습

● Oracle SQL Developer Data Modeler

- * 오라클에서 개발한 **graphical data modeling tool**
- * **DB 설계자, 관리자, 응용 프로그램 개발자, 사용자** 사이에 모델링 결과를 공유하게 함
- * 데이터 모델링 과정을 단순화

● SQL Developer Data Modeler의 기능

- * **logical, relational, physical, multi-dimensional, data type** 모델들의 생성, 열람, 편집
- * 데이터 모델링 결과로 **DDL** 스크립트 생성 : 다양한 **DBMS**에 **export** 가능

● Integrated Models

✱ **logical, relational, physical** 등의 세 가지 모델링 과정 제공

☑ **logical model** : 엔티티, 애트리뷰트, 관계성 등 표준 **ER modeling** 기반의 구성 요소를 제공

✍ 구현-독립적인 관점을 지원

☑ **relational model** : **logical**과 **physical model** 사이의 중간자적 모델

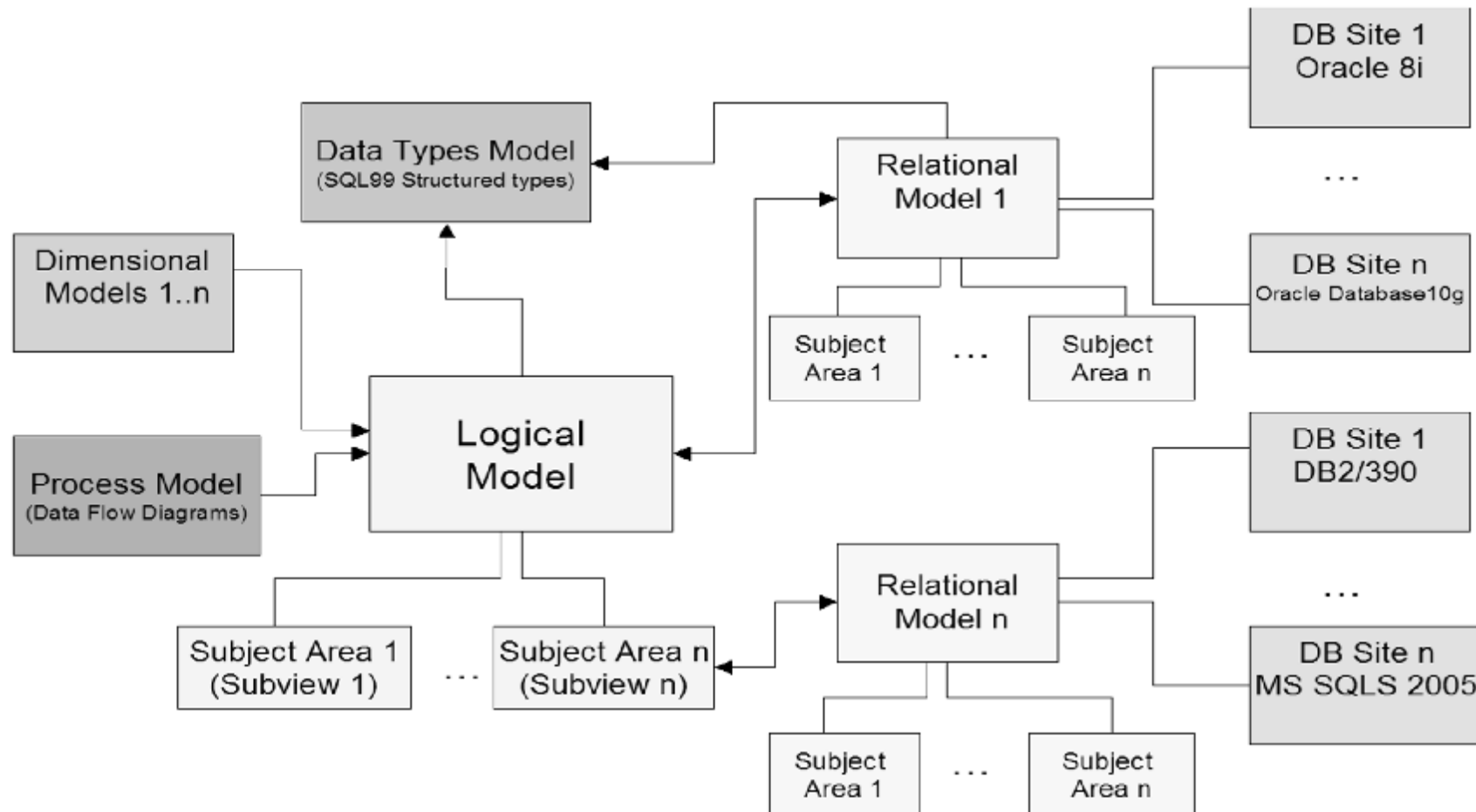
✍ 목표 **DBMS** 플랫폼과 독립적인 관계형 데이터 모델링 관점을 제공

☑ **physical model** : 특정 **DBMS** 위주의 **physical object**를 지원

✍ 목표 **DBMS**를 고려한 다양한 스키마 객체들을 제공

□ 모델링 실습

● SQL Developer Data Modeler : Model Structure



□ 모델링 실습

● 모델링 과정

* 논리 모델(logical model) 설계

- ① 도메인 추가 : 각 애트리뷰트에 적합한 타입을 정의
- ② 엔티티 생성 : 엔티티 이름, 애트리뷰트 이름과 타입, 주키 등을 정의
- ③ 관계성 생성 : 엔티티들 사이의 관계(**relation**)를 설정
 - 1:1, 1:N, M:N 등의 관계성 선택
 - 외래키 제약과 직접 연관됨
 - 엔티티 **A**의 외래키가 엔티티 **B**의 주키를 참조하는 관계성 생성 방법 :
 - 1:N 관계성을 선택 : 먼저 엔티티 **B**를 클릭한 후 엔티티 **A**를 클릭

* 관계형 모델(relational model) 생성

- ① **Design->Engineer to Relational Model**을 클릭하여 관계형 모델 생성
- ② 각 테이블의 외래키 이름을 적당하게 수정

* 물리 모델(physical model) 과정

- ① 목표 **DBMS**의 선택 : **Oracle 10g, Oracle 11g, SQL Server** 등등
- ② **DDL** 스크립트 생성 및 파일로 저장