

# UML 클래스 다이어그램을 이용한 관계형 데이터베이스 설계 방법

조도형\*, 주경수\*\*

\*\* \*순천향대학교 컴퓨터학과

e-mail : jhodoxyung@sch.ac.kr

## Relational Database Design Method With UML Class Diagram

\*Do-Hyung Jho, \*\*Kyung-Soo Joo

\*\* \*Dept of Computer Science, SoonChunHyang University

### 요 약

객체지향 설계에 사용하는 모델링 언어인 UML을 이용하여 유스 케이스 다이어그램을 도출합니다. 도출된 유스 케이스 다이어그램을 바탕으로 생성되는 클래스 다이어그램을 관계형 데이터베이스 스키마로 변환 방법을 소개 합니다.

### 1. 소개

UML 객체지향 개념이 데이터베이스에서 프로그래밍 언어로 도입되면서 이제는 분석 및 설계 부분에서도 표준화된 객체지향 설계 방법론이 등장하게 되었다. 객체 지향 설계에서 사용하는 모델링 표기법은 UML(Unified Modeling Language)이라는 표준화된 표기법을 따른다. UML은 소프트웨어 시스템의 구조물을 명세화, 구조화, 시각화, 문서화 뿐만 아니라 비-소프트웨어 시스템 그리고 비즈니스 모델링을 위한 하나의 언어이다[3].

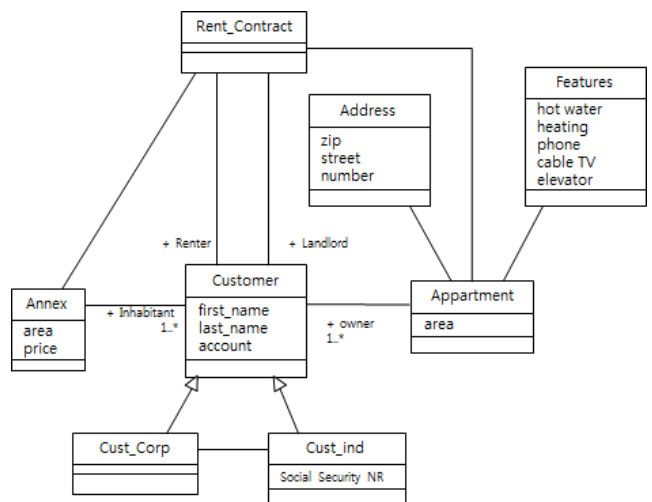
프로그램 개발은 요구사항 분석, 설계, 구현, 테스트 등의 과정을 거치게 된다. 만일 대규모 프로젝트라면 개발 기간 역시 수개월에서 수년이 걸리고, 요구분석가/아키텍처 설계자/DB설계자/프로그래머 등 각 부문에 전문가들이 협력해서 작업을 진행한다. UML은 프로그램 개발 과정을 일관된 관점에서 진행하고, 이들 업무 담당자들 간의 의사소통을 용이하게 하여 프로젝트가 효율적으로 진행될 수 있는 공통된 객체 지향적 설계 언어이다.

UML에서 순차 다이어그램은 응용프로그램에서 사건의 흐름을 분명하게 하는 역할을 하며 글로 풀어 쓴 유스 케이스(Use case)를 메시지로 표현한 그림으로 변환할 수 있다. 또한 그 그림은 항상 더 간결하게 이해되고, 특히 우리가 클래스 다이어그램을 만들 때 작업을 쉽게 해준다. 그러므로 UML을 이용해 유스 케이스에 의한 순차 다이어그램을 도출하여 클래스 다이어그램을 만든 후, 그 클래스 다이어그램에 의해서 관계형 데이터 모델링으로 변환할 수 있다.

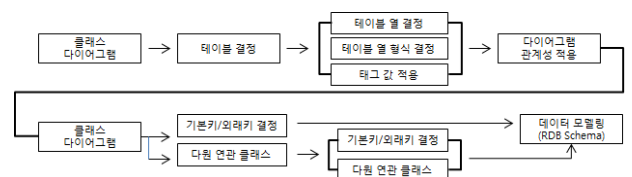
### 2. 데이터 모델링

데이터베이스 스키마 변환 과정에서 정보 구조로부터

논리적 개념을 이용하여 어떤 논리적(데이터) 구조로 표현하는 것이 필요한데, 데이터 모델링이란, 이 변환 과정을 말한다.[4] 그림 1의 클래스 다이어그램을 관계형 데이터베이스 스키마로 변환한다. UML 클래스를 이용한 데이터 모델링 과정의 다이어그램은 그림 1과 같다[8].



(그림 1) 클래스 다이어그램



(그림 2) 데이터 모델링 과정

그림2에서 관계형 데이터베이스 스키마 도출을 위한 데이터 모델링의 변환 방법은 다음의 방법을 사용한다[2].

- ① 클래스는 테이블이 됨.
- ② 클래스의 속성(attribute)은 테이블의 열(column)이 됨.
- ③ 클래스의 속성 타입은 테이블의 열 타입이 됨.
- ④ 일반화가 없는 클래스를 위해서는 integer 기본키를 생성하고, {oid}를 위해서는 기본키 제약 조건에 {oid} 태그 열을 추가함.
- ⑤ 자식 클래스(Subclasses)들은, 각 부모 클래스의 키를 기본키와 외부키 제약 조건에 추가함.
- ⑥ 속성에 {nullable} 태그가 있으면 테이블 속성에 NULL 또는 NOT NULL을 추가함.
- ⑦ 속성이 초기 값을 가지면, 열에 DEFAULT 문을 추가함.
- ⑧ 연관 클래스들은, 각 역할-실행 테이블에 대한 기본키를 기본키와 외부키 제약 조건에 추가함.
- ⑨ 만일 {alternate oid = <n>} 태그이면, UNIQUE 제약 조건에 대한 열을 추가함.
- ⑩ 각 명시된 제약에 대해 CHECK를 추가함.
- ⑪ 0..1, 1..1 규칙의 연관 관계에서 참조하는 테이블에 외부키를 생성함.
- ⑫ 집합 테이블(CASCADE와 같이)의 외부키를 갖는 복합집합을 위해서 기본키를 생성한다 ; 기본키를 위해 추가적인 열(column)을 추가함.
- ⑬ 이진 연관 클래스를 적당한 “N”쪽 테이블로 이동함으로써 최적화 함.
- ⑭ 연관 클래스가 아닌 3원 연관은 N : N에 대한 테이블로 생성함.
- ⑮ N : N, 3원 연관에서 역할-실행 테이블의 키로부터 기본키와 외부키 제약 조건을 생성함.
- ⑯ 연관 클래스 없는 다대다(many-to-many) 연관을 위해 기본키와 외부키를 생성함.

### 3. 데이터 모델링의 예

그림 1에 의한 ‘Rent\_Contract’ 객체는 그림 2의 관계형 데이터베이스 변환 방법 ①과 ②및 ④번의 성질에 따라 ‘Rent\_ContractID’ 객체 타입 속성을 저장하는 ‘Rent\_Contract’테이블로 그림 3과 같이 변환된다.

```
SQL> CREATE TABLE Rent_Contract(
    Rent_ContractID INTEGER PRIMARY KEY )
```

(그림 3) Rent\_Contract 테이블

그림 1에 의한 ‘Annex’객체는 그림 2와 관계형 데이터베이스 변환 방법 ①과 ②및 ④번의 성질에 따라 ‘AnnexID’ 객체 타입 속성을 저장하는 ‘Annex’ 테이블로 그림 4와 같이 변환된다.

```
SQL> CREATE TABLE Annex(
    AnnexID INTEGER PRIMARY KEY
    area string
    price double )
```

(그림 4) Annex 테이블

그림 1에 의한 ‘Address’객체는 그림 2와 관계형 데이터베이스 변환 방법 ①과 ②및 ④번의 성질에 따라 ‘AddressID’ 객체 타입 속성을 저장하는 ‘Address’ 테이블로 그림 5와 같이 변환된다.

```
SQL> CREATE TABLE Address(
    AddressID INTEGER PRIMARY KEY
    zip string
    street string
    number int )
```

(그림 5) Address 테이블

그림 1에 의한 ‘Features’객체는 그림 2와 관계형 데이터베이스 변환 방법 ①과 ②및 ④번의 성질에 따라 ‘FeaturesID’ 객체 타입 속성을 저장하는 ‘Features’ 테이블로 그림 6과 같이 변환된다.

```
SQL> CREATE TABLE Features(
    FeaturesID INTEGER PRIMARY KEY
    hotwater string
    heating string
    phone string
    cableTV string
    elevator string )
```

(그림 6) Features 테이블

그림 1에 의한 ‘Customer’객체는 그림 2와 관계형 데이터베이스 변환 방법 ①과 ②및 ④번의 성질에 따라 ‘CustomerID’ 객체 타입 속성을 저장하는 ‘Customer’ 테이블로 그림 7과 같이 변환된다.

```
SQL> CREATE TABLE Customer(
    CustomerID INTEGER PRIMARY KEY
    first_name string
    last_name string
    account int )
```

(그림 7) Customer 테이블

그림 1에 의한 ‘Apartment’객체는 그림 2와 관계형 데이터베이스 변환 방법 ①과 ②및 ④번의 성질에 따라 ‘Apartment’ 객체 타입 속성을 저장하며 ‘Apartment’는 변환방법 ⑩에 의해서 그림 5와 그림 6에서 정의한 ‘Address’와 ‘Features’를 적용하고 그림 8과 같이 변환된다.

```
SQL> CREATE TABLE Apartment(
ApartmentID INTEGER PRIMARY KEY
area string
AddressID INTEGER REFERENCE Address
FeaturesID INTEGER REFERENCE Features
CONSTRAINT Apartment_PK PRIMARY
KEY(ApartmentID, AddressID FeaturesID) )
```

(그림 8) Apartment 테이블

그림 1에 의한 'Cust\_Corp' 객체는 그림 2와 관계형 데이터베이스 변환 방법 ①과 ② 및 ④번의 성질에 따라 'Cust\_Corp' 객체 타입 속성을 저장하며 'Cust\_Corp' 는 변환방법 ⑤와 ⑪에 의해서 그림 7에서 정의한 'Customer' 를 적용하고 그림 9과 같이 변환한다.

```
SQL> CREATE TABLE Cust_Corp(
Cust_CorpID INTEGER PRIMARY KEY
CustomerID INTEGER REFERENCE Customer
CONSTRAINT Cust_Corp_PK PRIMARY
KEY(Cust_CorpID, CustomerID) )
```

(그림 9) Cust\_Corp 테이블

그림 1에 의한 'Cust\_ind' 객체는 그림 2와 관계형 데이터베이스 변환 방법 ①과 ② 및 ④번의 성질에 따라 'Cust\_ind' 객체 타입 속성을 저장하며 'Cust\_ind' 는 변환방법 ⑤와 ⑪에 의해서 그림 7에서 정의한 'Customer' 를 적용하고 그림 10과 같이 변환한다.

```
SQL> CREATE TABLE Cust_ind(
Cust_indID INTEGER PRIMARY KEY
SocialSecurityNR INTEGER
CustomerID INTEGER REFERENCE Customer
CONSTRAINT Cust_ind_PK PRIMARY
KEY(Cust_indID, CustomerID) )
```

(그림 10) Cust\_ind 테이블

#### 4. 결론 및 향후 과제

지금까지 UML을 통해서 도출된 유스 케이스를 바탕으로 생성되는 클래스 다이어그램을 관계형 데이터베이스 스키마로 변환 시키는 방법을 소개하였습니다. 향후 연구 과제로는 기존의 개체 관계형 모델(ERD)과 같은 관계형 데이터베이스 설계 기법은 데이터의 구조적 관점만을 고려하므로 행위적 관점을 포함하는 객체 관계형 데이터베이스에 관하여 연구를 진행해 나갈 것입니다.

#### 참고문헌

[1] Database Desing for Smarties, Using UML for Data Modeling Robert J.Muller, MOGAN KAUFMANN PUBLISHERS.

[2] 방승윤; 주경수 , “XML Schema 응용을 위한 UML

기반의 설계 방법론 개발” 2002. 순천향대학교 대학원

[3] 임춘봉; 신인철; 신재철 공저, UML 사용자와 지침서, 도서 출판 인터비전, 1999.

[4] 이석호 ; 데이터베이스론, 정익사 1999.

[5] Duckett Jon, Ozu Nik, Williams Kevin, Mohr Stephen, Cagle Jurt, Griffin Oliver, Norton Francis, Stokes-Rees Ian, and Tennison Jeni. Professional XML Schemas, Wrox Pr Inc, 2001.

[6] 방승윤; 주경수 , “데이터베이스 기반의 XML 응용을 위한, UML을 이용한 통합 설계 방법론”, pp.49 - 67, 2002. 6.

[7] 이상태; 주경수 , “전자상거래를 위한 UML 기반의 XML DTD의 ORDB 스키마로의 변환”, 한국데이터베이스 학회 2001년 춘계 Conference, pp.303-310. 2001.5

[8] Database Design with UML : Implementation Issues, University of Ljubljana, Slovenia

[9] 방승윤; 주경수 , “관계형 데이터베이스 기반의 XML 응용을 위한, UML 클래스를 이용한 통합 설계 방법론”, 정보처리학회논문지 D, 제10-D권 제1호, 2003. 2. 28