

2. 데이터베이스 모델링

1. 데이터베이스 모델링의 이해 [34]

개요

프로그램에서 요구되는 현실 세계의 정보를 모두 반영하면서 데이터에 대한 오류 혹은 왜곡이 존재하지 않아야 하며, 응용 프로그래머, 데이터베이스 설계자, 의뢰인 등과 상호 의사소통을 위한 단계적 절차를 제공하는 데이터베이스 모델링 과정이 필요함
실세계의 정보를 사용하여 여러 프로그램을 구축하기 위해 데이터베이스의 사용이 요구되는 경우, 해당 정보를 데이터 모델의 표기법으로 변환하는 정보의 추상화 과정, 즉 데이터 모델링이 선행되어야 함

데이터베이스 모델링 과정

개념적 데이터 모델링

- 데이터에 초점을 맞추어 개별적 데이터의 특징을 분리

논리적 데이터 모델링

- 특정 상용 DBMS에 맞추어 데이터를 표현

프로그램 개발

- 데이터베이스와 관련되는 프로그램이 개발

1. 사용자 요구사항 분석 단계

사용자 요구사항(user requirements) 분석

- 실제 데이터베이스 애플리케이션을 사용할 실무자와 문서를 교환하거나 인터뷰를 통하여 데이터에 대한 상세한 요구사항을 수집하고 결정

명세서(specification)

- 데이터를 활용하는 시스템의 목표를 수립하고 그 목표를 달성하기 위한 문서화 작업이 동반됨

- 업무 및 업무 간의 데이터 흐름을 가능한한 상세하고 내용적으로 완벽하게 기술하는 것을 목표로 함

세부 단계

1. 요구사항 도출 단계

- 데이터베이스를 구축하기 위한 목표, 범위를 기준으로 조사범위를 결정

- 요구사항 명세서(requirement specification)

2. 요구사항 분석 단계

- 도출된 요구사항이 명확한지, 완전한지, 모호하지 않은지 판단

- 요구사항 정의서(requirement definition)

- 도출된 요구사항들을 기능별로 세분화하고 서로 중복되거나 유사한 사항이 있다면 하나로 통합하는 등 기능별로 요구를 상세화

3. 요구사항 기록 단계

- 요구사항 목록을 정리하여 문서화 작업을 한 다음 관리자의 승인을 받고 프로젝트가 종료될 때까지 각 요구사항의 반영 여부를 지속적으로 확인 및 관리한다.

2. 데이터 모델링 단계

데이터 모델링(data modeling)

- 데이터에 대한 요구사항을 정의하고 분석하여 추상화하는 과정

- 업무에 필요한 데이터를 시스템 구축 방법론을 사용하여 분석하고 이를 DB 시스템에 반영할 수 있는 형태로 표현

- 표기법(notation)

표현대상

데이터 관점

- 업무가 어떤 데이터와 관련이 있는지, 데이터 간의 관계는 어떠한지에 대해 표현

프로세스 관점

- 업무를 토옴 어떤 일(연산)을 처리하는지, 무엇을 해야 하는지를 표현

데이터-프로세스 상관 관점

- 업무에서 일을 처리하는 방법에 따라 어떤 데이터가 어떻게 영향을 받는지 표현

- 사용자 요구사항에 대한 명세서는 데이터 흐름도, 순서 다이어그램, 시나리오 등 다양한 형태로 기술되고 정리

데이터베이스 모델링 과정 요약도

- 개념적 데이터 모델링을 시작으로 현실 세계의 업무를 데이터와 데이터 간의 관계로 정의하고 DBMS에서 사용하는 데이터 모델에 맞게 대응시킨 후, 마지막으로 물리적으로 데이터베이스를 모델링하는 작업이 이루어진다.

(1) 개념적 데이터 모델링

- 사용자 요구사항을 바탕으로 고수준 모델링 단계를 진행, 공통된 표기법을 사용한 추상화(abstraction) 기법이 사용됨

- 추상화

- 실세계의 데이터들을 개념적으로 일반화시킨 데이터 타입, 속성, 관계, 제약조건들을 이끌어 내는 과정
 - ER(개체-관계) 모델
 - 시스템이 어떤 데이터를 포함하며, 어떤 업무가 발생했는지를 추상화하는 모델로, 개체와 개체 간의 관계를 통하여 표현
 - 데이터의 구조를 표현하는 고수준의 추상적 모델
 - (2) 논리적 데이터 모델링
 - 특정 상용 DBMS의 데이터베이스로 작성하기 위해서 DBMS에 맞는 구현 데이터 모델인 스키마(schema)로 변환하는 작업이 요구됨
 - 관계형 모델, 객체지향형 모델, Key/Value 모델 등
 - 최종적으로 목표 DBMS의 데이터 정의 언어(DDL)로 기술된 논리 스키마가 생성됨
 - 데이터들의 논리적 구성만을 명시
 - (3) 물리적 데이터 모델링
 - 데이터베이스 파일의 물리적 저장방식을 결정하는 물리적 데이터 모델링 과정을 거쳐야 완전한 DDL문으로 완성
 - DB 파일의 내부 저장구조, 파일 구성, 인덱스, 접근 경로 등을 결정
 - 레코드 저장방법, 데이터 타입 및 저장공간 등을 계산하여 설계
- ## 2. 사용자 요구사항 분석 과정 [38]
- 사용자 요구사항 분석 과정
 - 제안 요청서(RFP: Request for Proposal)
 - 데이터베이스 시스템의 설계, 개발, 도입 과정은 사업의 목표, 사용자의 요구기능, 관리하는 데이터의 보안 수준 등 시스템의 기능과 성능에 대한 다양한 요구사항들을 기술
 - 실제 시스템을 개발하는 수탁 업체에서는 요구사항 도출 전 RFP를 토대로 인터뷰 및 분석을 통하여 개발 관점에서의 요구사항을 새롭게 도출
 - 수탁 업체는 사업 내용을 숙지하여 제안을 요청한 K대학교가 학사 시스템을 필요로 하게 된 배경, 목적 및 원하는 시스템의 개괄적인 범위에 대하여 파악
 - 제안 요청 내용은 시스템 개발과정에서 사용자의 요구에 대한 기능적/기술적 기능을 해당하는 분류 및 코드에 맞춰 작성
 - 요구사항 목록에서 사용되는 코드와 내용은 표와 같다 [41]
 - SFR
 - 소프트웨어 기능 등 시스템 개발에 필요한 요구사항
 - FUR 코드
 - 기능 요구사항, 학생증에 계좌를 추가하여 학생증으로 결제하거나 학생증에 연동된 계좌에 입출금이 가능한 기능적인 측면이 명시
 - 요구사항 정의서
 - 사용자와 개발자가 협의하에 분석된 요구사항을 명확하고 정확하게 기록하는 문서
 - 요구사항 정의서 작성단계에서 다음의 세 가지 사항이 이루어짐
 - 현재 운용되고 있는 시스템의 사용자 지침서나 인터뷰 결과와 같은 요구사항 관련 문서를 토대로 기술서 개요를 작성
 - 프로젝트의 목적, 기능 및 비기능 요구사항, 가정과 위험 요소, 용어와 데이터 정의 및 사용자 인터페이스 식별 등을 명세화
 - 사용자의 내용을 합의하고 하나의 업무 단위로 가치를 가지고 수행될 수 있는 단위로서의 업무를 도출하여 업무 내용을 기술
 - FUR-001 요구사항의 학생정보 기입
 - 구체적으로 학생번호, 학생이름, 생일, 계좌번호 등 학생증에 기입될 정보를 명확히 파악하여 사용자 요구사항 정의서에 해결방안으로 기록
- ## 3. ER 모델 [45]
- ER(Entity-Relationship) 모델
 - 1976 첸(P. Chen) 박사가 제안한 것, 전체적인 데이터베이스의 논리적 구조를 표현
 - 실세계의 조직 업무의 의미와 상호작용을 개념적으로 나타내는 데 매우 유용
 - 1. 개체 집합
 - 개체(entity)
 - 실세계에 존재하는 다른 모든 개체와 구별되는 유/무형의 대상(사물, 사람, 사건 등)을 표현
 - 무형적 개체
 - 학사관리 시스템에서 과목, 계좌와 같이 눈에 보이지 않는 개체
 - 유형적 개체
 - 학생, 교수와 같이 눈에 보이는 유형적 개체

- 개체는 속성들의 집합
- 속성(attribute)
 - 개체를 특정 지을 수 있는 다수의 속성들로 정의
 - 학생의 '학생번호', '학생이름', '성별', '나이'
- 개체 집합(entity set)
 - 개체들의 모임으로, 같은 속성을 공유하는 개체들로 구성
- 값(value)
 - '201934-021216', '유관순', '17'
- 하나의 데이터베이스는 다수의 개체 집합으로 이루어진다.
- UML(Unified Modeling Language, 통합 모델링 언어)
 - 키(key)
 - UML 형식에서는 모든 개체를 고유하게 구분하는 키(key)가 존재
 - 도메인(domain)
 - 속성에는 속성이 가질 수 있는 모든 가능한 값들의 범위를 의미
- ERD
- 2. 관계 집합
 - 관계(relationship)
 - 여러 개체들은 그들 사이에 연관성을 가진다.
 - 예) '유관순 학생', '컴퓨터 과학과'를 연결시키는 '전공' 관계
 - 관계 인스턴스(relationship instance)
 - (유관순 학생, 컴퓨터과학과)를 연결시키는 특정 관계
 - 관계 집합(relationship set)
 - 같은 유형의 관계 인스턴스들의 집합으로, 2개 이상의 개체 집합 간의 수학적 연결 관계
 - 한 개체 집합이 다른 개체 집합과 연관성을 가지는 것
 - 관계 집합에 참가(participate)한다.
 - 한 개체 집합이 다른 개체 집합과 연관성을 가지는 것
 - '전공 관계 집합' [48]
 - 관계 집합의 차수(degree)
 - 관계 집합에 참여하는 개체 집합의 수
 - 전공 관계 집합의 차수는 2가 된다.
 - 이항관계 집합
 - 관계 집합의 차수가 2일 경우 이항관계 집합
 - 역할(role)
 - 관계에 참가하는 개체의 기능
 - 재귀적 관계 집합
 - 어떤 개체 집합이 자기 자신과 관계를 형성하는 관계 집합
 - 동일한 개체 집합이 재귀적(recursive)으로 서로 다른 역할로 참가하여 생성되는 재귀적 관계 집합의 경우 역할을 명시해주어야 함.
 - '과목'과 '선수 과목' 관계 집합
 - 관계는 개체의 참가 정보 외에 관계가 맺어질 때 새롭게 생성되는 데이터에 대한 추가적인 설명 속성을 가질 수도 있다.
- 3. ER 모델(2) [50]
- 3. 속성
 - 속성(attribute)
 - 개체를 구체적으로 설명하는 특성
 - (1) 단순 속성과 복합 속성
 - 단순 속성(simple attribute)
 - 더 이상 작은 구성요소 나눌 수 없는 속성
 - '학생' 개체 집합의 학생번호, 학생이름, 성별, 나이와 같은 속성의 값을 나눌 경우 의미가 소실되기 때문에 나누어질 수 없다.
 - 복합 속성(composite attribute)
 - 더 작은 의미단위로 나누어질 수 있는 속성

- └ '생일'은 년, 월, 일의 조합으로 이루어짐
 - └ 복합 속성을 모델링상에 중복하여 포함시키기도 한다.
 - └ 들여쓰기 형태로 표현

(2) 단일값 속성과 다중값 속성

- └ 단일값 속성(single valued attribute)
 - └ 한 개체가 특정 속성에 단 하나의 값을 가질 경우
- └ 다중값 속성(multivalued attribute)
 - └ 속성값으로 여러 개의 값을 가질 경우 이러한 속성을 다중값 속성이라고 한다.
 - └ 집합기호 '{ }'

(3) 유도 속성과 저장 속성

- └ 다른 속성의 값으로부터 연산 과정 통해 새롭게 생성
- └ 유도 속성(derived attribute)
 - └ 다른 속성값으로부터 유도되어 결정되는 속성값
 - └ 함수라는 의미의 괄호 '(')'를 사용하여 표현한다.
- └ 저장 속성(stored attribute)
 - └ 실제 값을 저장해야만 의미가 유지되는 속성

4. 제약조건

관계 집합

- └ 한 개체 집합에서 다른 개체 집합으로 개체 간 대응성, 곧 사상을 의미

ER 모델링은 관계 집합에 대한 상세 표현을 위해 데이터베이스가 준수해야 되는 제약조건(constraints)를 부가할 수 있다.

(1) 사상수(mapping cardinality)

- └ 관계 집합에 참가한 개체 집합들에 대해 각각의 개체가 얼마만큼의 관계를 맺을 수 있는지를 명시

유형

일대일(1:1, one-to-one) 관계

- └ A에 포함된 한 개체가 최대 B에 포함된 한 개체와 관계하며, B에 포함된 한 개체가 최대 A에 포함된 한 개체와 관계한다.

- └ ex) 학사 시스템에서 학생증에 계좌정보를 연동한 경우 한 학생마다 오직 하나의 계좌 정보를 '보유'하고, 그 한 계좌는 오직 학생에게만 '보유'된다.

일대다(1:N, one-to-many) 관계

- └ A에 포함된 한 개체가 B에 포함된 한 개 이상의 개체와 관계하며, B에 포함된 개체는 최대 A에 포함된 한 개체와 관계한다.

다대일(N:1, many-to-one) 관계

- └ A에 포함된 한 개체가 최대 B에 포함된 한 개체와 관계하는 반면, B에 포함된 한 개체는 A에 포함된 한 개 이상의 개체와 관계한다.

다대다(M:N, many-to-many) 관계

- └ A에 포함된 한 개체가 B에 포함된 한 개 이상의 개체와 관계하며, B에 포함된 한 개체는 A에 포함된 한 개 이상의 개체와 관계한다.

(2) 참가 제약조건 (participation constraint)

- └ 관계에 참여하는 개체 집합의 범위

전체적(total) 참가

- └ 어떤 개체집합이 관계 집합에 참여할 때, 개체 집합의 모든 개체가 관계에 참가
- └ 이중 실선

부분적(partial) 참가

- └ 일부만 참가해도 된다면 부분적 참가
- └ 단일 실선

(3) 키 속성

키(key)

- └ 개체 집합의 각 개체마다 서로 다른 값을 가지는 속성의 집합
- └ 어떤 개체의 키 값: 각각의 개체들을 서로 구별하는 데 사용되는 개체 집합에서 유일한 값
 - └ 개체 당 여러개의 키를 가질 수도 있음
- └ 밑줄로 표기
- └ 역할

- 개체 찾기
- 관계 집합의 특정한 관계 찾기

표현

- 관계 집합
 - 다이아몬드
- 속성
 - 직사각형 (하위 부분)
- 화살표가 있는 실선
 - 일(1)
- 화살표가 없는 실선
 - 다(N)
- 부분 참여
 - 하나의 실선
- 전체 참여
 - 이중 실선

4. ER 모델의 기호 [57]

ER 다이어그램(ERD: ER diagram)

데이터베이스 전체의 논리적 구조를 ERD로 표현

주요 요소

- 이분 직사각형
 - 개체 집합을 표현하며, 위부분은 개체 집합의 이름을 포함하고, 아랫부분은 개체 집합을 구성하는 속성들의 이름을 포함한다. 기본키는 밑줄로 표현된다.
- 이중 이분 직사각형
 - 약한 개체 집합을 나타낸다.
- 다이아몬드
 - 관계 집합을 나타낸다.
- 단일 직사각형
 - 관계 집합의 속성들을 나타낸다.
- 실선
 - 개체 집합과 관계 집합을 연결한다. 또한 부분 참여를 의미한다.
- 점선
 - 관계 집합의 속성들을 관계 집합에 연결한다.
- 이중선
 - 관계 집합 내에 개체의 전체 참여를 나타낸다.
- 이중 다이아몬드
 - 약한 관계 집합을 나타낸다.

예시 [58]

- 각 개체 집합에서 학생번호와 과목코드는 키
- 수강은 학생과 과목 개체 집합 사이의 관계 집합이며, 추가적인 신청시각 속성
- 학생과 과목은 다대다 관계
- 참가제약 조건 학생은 수강 관계에 부분참여(단선), 과목은 수강 관계에 부분참여(단선)

재귀적 관계 집합(recursive relation set)

- 한 개체 집합과 관계 집합 사이에 두 개의 실선으로 표현하며, 역할을 기술해준다.
- 선수과목 관계 집합
 - 학생은 보유한 학생증에 금융기능을 추가하여, 한 학생당 하나의 계좌를 부여
- 식별 개체(identifying entity)
 - 학생은 보유한 학생증에 금융기능을 추가하여, 한 학생당 하나의 계좌를 부여한다.
- 약한 개체(weak entity)
 - 계좌
- 강한 개체(strong entity)
 - 상대적으로 지금까지 다루었던 일반적 객체

학생 개체가 삭제되면, 그 개체와 연관된 약한 개체 집합 쪽의 계좌 개체는 더 이상 유지될 이유가 없기 때문에 같이 삭제된다.

약한 개체 집합

일대다 관계 사상수를 가지며, 이를 통해 강한 개체 집합과 연결되어야 한다.

약한 개체 집합의 개체를 구별하는 키는 강한 개체 집합의 키와 결합하여 구성하며, 실선이 아닌 점선으로 밑줄을 삽입한다.

약한 개체집합과 연결된 관계 집합은 '이중선'으로 표기한다.