**정보처리기사 실기 2장. 데이터 입출력 구현**

***최종수정일 – 2023/04/01***

**중요사항(암기사항)**은 **볼드체**

**시나공 정보처리기사 실기2023 요약.**

**29. 데이터베이스 개요**

정보를 컴퓨터에서 사용하기 위해 논리적 구조로 조직화하고, 다시 그 논리적 구조를 물리적으로 조직화해 물리적 저장장소에 저장한 게 데이터베이스이다.

**데이터베이스**

여러 사람들이 **공동**으로 사용할 **정보**에서 **중복을 제거**하고 **통합**한뒤 **저장장치**에 저장해 항상 사용할 수 있도록 **운영**하는 시스템이다.

다음과 같이 구분된 정의를 가진다.

* **통합된 데이터(Integrated Data): 중복을 제거**한 **데이터 모임**
* **저장된 데이터(Stored Data): 컴퓨터로 접근**할 수 있는 **저장 매체**에 저장된 자료
* **운영 데이터(Operational Data): 조직을 운영**하는데 있어 **반드시 필요한** 자료
* **공용 데이터(Shared Data): 여러 시스템**들이 **공동으로 소유한** 자료

**DBMS(Database Management System; 데이터베이스 관리 시스템)**

기존의 파일 시스템이 갖는 데이터의 종속성과 중복성의 문제를 해결하기 위해 제안된 시스템으로 사용자가 바라는 대로 데이터베이스에 정보를 생성하고 이를 관리해주는 시스템이다. **DBMS**로 활용되기 위해서는 다음과 같은 필수 기능 세 가지(**DMC**)를 갖춰야 한다.

**D**efinition(정의 기능): 데이터의 type과 구조에 대한 정의, 이용 방식, 제약 조건 등을 명시

**M**anipulation(조작 기능): 데이터 검색, 갱신, 삽입, 삭제를 위한 인터페이스 제공

**C**ontrol(제어 기능): 데이터의 무결성, 보안, 권한 검사, 병행 제어를 제공하는 기능

**DMC(정의, 조작, 제어) 기억하자.**

**데이터의 독립성**

독립성 🡨----🡪 종속성, 논리적 독립성과 물리적 독립성이 있다.

**논리적 독립성 –** 응용프로그램과 데이터베이스의 분리를 말한다. 이렇게 하면 데이터베이스의 논리적 구조를 변경해도 응용프로그램에 영향을 미치지 않는다.

**물리적 독립성 –** 응용프로그램과 보조기억장치 같은 물리적 장치를 독립시켜 디스크를 추가/변경해도 응용 프로그램은 영향을 받지 않는다.

**스키마(Schema)**

데이터베이스의 구조와 제약조건에 관한 전반적인 명세를 기술한 것이다.

**종류.**

**외부 스키마:** 사용자나 응용 프로그래머가 각 개인의 입장에서 필요로 하는 데이터 베이스의 논리적 구조를 정의한 것

**개념 스키마:** 데이터베이스의 논리적 구조, 응용 프로그램과 사용자들이 필요로 하는 데이터를 종합한 조직 전체의 데이터베이스이다. 단 하나 뿐이다.

**내부 스키마:** 물리적 저장장치 관점에서 데이터베이스, 실제 저장될 레코드 형식, 저장 데이터 항목 표현 방법, 내부 레코드의 물리적 순서 표시

**데이터베이스 설계**

소프트웨어의 요구 공학처럼 데이터베이스도 사용자의 요구를 분석해 컴퓨터에 저장할 데이터베이스의 구조에 맞게 변형한 후 DBMS로 데이터베이스를 구현한다.

**데이터베이스 설계 시 고려사항**

* **무결성 :** 삽입, 삭제, 갱신 등 연산 후에도 데이터가 정해진 제약 조건을 항상 만족해야 한다.
* **일관성 :** 특정 데이터에 접근하거나 질의를 할 때 응답에 변함 없어야 한다.
* **회복 :** 시스템장애가 발생하면 장애 이전 상태로 복구 가능해야 한다.
* **보안 :** 허가되지 않은 데이터의 노출, 변경, 손실로부터 데이터를 보호할 수 있어야 한다.
* **효율성 :** 응답시간 단축, 시스템 생산성, 저장 공간의 최적화가 가능해야 한다.
* **데이터베이스 확장 :** 데이터베이스 운영에 영향을 주지 않으면서 지속적으로 데이터를 추가할 수 있어야 한다.

**데이터베이스 설계 순서**

**요구조건 분석 – 개념적 설계 – 논리적 설계 – 물리적 설계 – 구현**

**개념적 설계(정보 모델링, 개념화)**

개념 스키마 모델링과 트랜잭션 모델링을 병행 수행한다. 요구 분석에서 나온 요구 조건 명세를 DBMS에 독립적인 E-R 다이어그램으로 작성함으로 DBMS에 독립적인 개념 스키마를 설계한다.

**\*개념 스키마는 DBMS에 독립적이다.**

**논리적 설계(데이터 모델링)**

현실의 개념을 특정 DBMS에 맞는 논리적 자료 구조로 변환(mapping)하는 과정이다. 데이터를 필드로 기술된 데이터 타입과 데이터 타입 간의 관계로 표현되는 논리적 구조로 모델링한다. 개념적 설계가 개념 스키마를 설계하는 단계라면 논리적 설계는 개념 스키마를 평가, 정제하고 DBMS에 따라 서로 다른 논리적 스키마를 설계하는 단계이다. 트랜잭션의 인터페이스를 설계한다.

**물리적 설계(데이터 구조화)**

논리적 구조로 표현된 데이터를 하드웨어에 저장할 수 있는 물리적 구조의 데이터로 변환하는 과정이다. 파일 저장결로, 액세스 경로 등을 설정한다. 또 저장 레코드의 형식, 순서, 접근 경로, 조회 집중 레코드 등의 정보를 사용해 데이터가 컴퓨터에 어떻게 저장될지 결정한다.

**데이터베이스 구현**

논리 스키마와 물리 스키마를 실제 파일로 생성하는 과정이다. 사용 DBMS의 데이터 질의어(DDL)을 통해 데이터베이스 스키마를 기술하고 컴파일해 데이터베이스 파일을 생성한다. 응용 프로그램을 위한 트랜잭션을 생성하고 데이터베이스에 접근하기 위한 응용 프로그램을 구현한다.

**31. 데이터 모델의 개념**

**데이터 모델**

구성요소: 개체 속성 관계

종류: 개념적 모델, 논리적 모델, 물리적 모델

데이터 모델에 표시할 요소: 구조, 연산, 제약조건

**개념적 데이터 모델**

**정보모델, E-R 다이어그램으로 표현, 결과적으로 데이터를 개체, 속성, 관계로 구분하는 일**

**속성은 행렬에서 열의 구분기준이 되고 개체는 행의 기준이 된다. 관계는 개체의 속성 type과 이 type들 간의 관계로 설명된다.**

**논리적 데이터 모델**

**단순히 데이터 모델**이라고 하면 **논리적 데이터 모델**을 의미한다. **개념적 테이터 모델을 컴퓨팅 논리에 적합하게 변형**시킨 모델이다. **테이블, 레코드, 필드, 엔티티** 등을 통해 **현실을 추상화**한 데이터 모델로 변형시킨 모델이 논리 데이터 모델이다. DBMS에 맞는 논리 데이터 모델 하나를 선정해 사용한다. 데이터 간의 관계를 어떻게 표현하냐에 따라 관계 모델, 계층 모델, 네트워크 모델로 구분한다.

**참고. 테이블=행렬, 레코드=행, 필드=열, 엔티티=현실세계의 존재를 논리 모델로 성형한 것**

**데이터 모델에 표시할 요소**

* **구조:** 논리적 모델링을 거친 개체 타입들 간의 관계를 말한다. 데이터 구조의 정적 성질을 표현한다.
* **연산:** DBMS에서 데이터 처리 방식을 명세화한 것이다. 데이터 조작 도구를 말한다.
* **제약조건:** DBMS에 저장될 수 있는 데이터의 논리적 제약조건이다.

**32. 데이터 모델의 구성요소**

**개체(Entity)**

현실 속 유, 무형의 대상을 컴퓨팅 환경에서 다루기 위해 개념적으로 만들어낸 모델이다. 유일한 식별자(Unique Identifier)에 의해 구분되며 다른 개체와 하나 이상의 관계에 있다.

개체는 객체가 아님에 유의하자

**속성(Attribute)**

데이터베이스를 구성하는 **가장 작은 논리단위**, **데이터 필드(열)을 구분하는 기준이라고 생각하면 된다.** 속성의 수를 **디그리 또는 차수**라고 한다. 속성은 속성의 특성과 개체 구성 방식에 따라 분류한다.

**속성의 특성에 따른 분류**

**기본 속성: 속성 중 가장 많고 일반적이다. 도메인 영역**에서 뽑아낸 속성이다. **하지만 도메인에서 뽑아냈어도 업무상 코드로 정의된 속성이라면 기본 속성에서 제외된다.**

**설계 속성:** 도메인에 없고 설계, 데이터 모델링 과정에서 나오는 속성.

**파생 속성:** 다른 속성에서 계산, 변경 등 영향 받아 생기는 속성. 적을수록 좋다.

**속성의 개체 구성 방식에 따른 분류**

**기본키:** 개체를 식별할 유일한 속성

**외래키:** 다른 개체와의 관계에서 포함된 속성

**일반 속성:** 개체에는 포함되어 있으나 기본키, 외래키에 포함되지 않는 속성

**관계(Relationship)**

개체와 개체 사이의 논리적 연결

**관계의 형태**

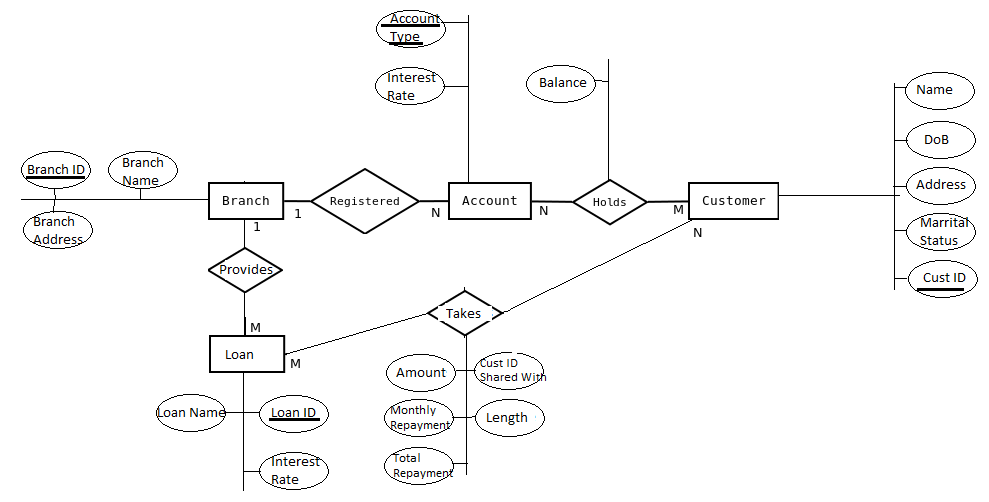
**1:1, 1:N, N:M**

**관계의 종류**

**종속 관계, 중복 관계(없을수록 좋지만 반정규화 등 필요에 의해 쓰기도 함), 재귀 관계, 배타 관계**

**33. E-R(개체-관계) 모델**

엔터티-릴레이션쉽 모델. 현실세계의 무질서를 개체의 관계 속에서 사용 가능한 논리 데이터로 표현한다. 개체, 관계, 속성으로 데이터를 표현하며 1:1, 1:N, N:M 등의 관계 유형을 사용할 수 있다. 개념적 모델의 대표격



알 수 없는 작성자 님의 [이 사진](http://stackoverflow.com/questions/20313095/bank-er-notation-example)에는 [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/) 라이선스가 적용됩니다.

사각형 = 개체

마름모 = 관계

타원 = 속성

이중타원 = 다중값 속성(복합 속성)

밑줄 타원 = 기본키 속성

복수 타원 = 복합 속성

네모-마름모-네모 = 관계

선 = 개체 타입과 속성 타입을 연결