

Chapter 01

독일 하루 후기 구독하기



복습 문제

1.1

데이터: 알려진 사실로서 의미를 가지고 있고, 기록될 수 있는 정보.

데이터베이스: 특정한 의미를 가지고 특정한 목적을 위해서 설계, 구축, 운용되는 서로 연관 있는 데이터의 집합. 실세계의 일부분을 표현해야 하고, 변화가 데이터베이스에 반영된다.

DBMS: Database Management System. 데이터베이스 관리 시스템. 사용자가 데이터베이스를 생성하고 관리할 수 있도록 하는 컴퓨터화된 시스템. 데이터베이스의 정의, 구축, 조작, 공유할 수 있는 기능을 가지고 있다.

데이터베이스 시스템: 데이터베이스 + DBMS

데이터베이스 카탈로그: 데이터베이스를 정의하고 설명하는 정보가 저장된 곳. 이를테면 "이름"은 문자열이고, 10자.

프로그램-데이터 독립성: 데이터 파일의 구조가 응용 프로그램과 분리되어 DBMS 카탈로그에 저장된다. 따라서 데이터 파일의 구조가 변경되어도 응용 프로그램은 거의 변하지 않는다.

사용자 뷰: 가상 테이블. 데이터베이스의 일부 또는 데이터베이스로부터 유도되는 가상 데이터. 데이터베이스에 저장되지 않는다.

데이터베이스 관리자: 데이터베이스에 대한 접근 감독, 자원의 사용을 모니터링하고 조정. 필요한 경우 소프트웨어나 하드웨어 자원을 구입. 데이터베이스의 보안이나 시스템의 성능 문제 책임

최종 사용자: 데이터베이스에 질의하고 변경하고 보고서를 작성하는 사람. 대부분 초보 사용자.

미리 작성된 트랜잭션: 기작성 트랜잭션. 미리 프로그램되고 테스트가 된 트랜잭션. 예를 들어 은행 고객과 직원이 계좌 잔액 조회.

[독일 하루 후기 구독하기](#)

연역 데이터베이스 시스템: 데이터베이스에 저장되어 있는 사실로부터 새로운 정보를 추론하는 연역적 규칙을 정의할 수 있다.

지속성 객체: 데이터베이스에 저장된 객체. 프로그램의 수행이 끝난 후에도 데이터베이스에 영구적으로 남아 있어 필요하면 언제든지 다른 프로그램들에서 접근 가능

메타데이터: 데이터베이스의 정의 또는 데이터베이스를 설명하는 정보는 카탈로그나 사전의 형태로 저장되는데 이를 메타데이터라고 한다. 즉, 카탈로그에 저장된 정보.

트랜잭션 처리 응용: 트랜잭션은 데이터베이스를 한 번 이상 읽거나 쓰는 것을 의미하며 하나의 프로세스를 수행하는 것이다. 트랜잭션 처리에 있어서 ACID 규칙을 지켜야 한다. 원자성, 일관성, 고립성, 내구성을 의미한다.

1.2

데이터베이스 관리: 1.1

데이터베이스 설계: 보통 데이터 채우기 전에 작업. 데이터베이스에 저장될 데이터 선정, 구조 정의. 사용자 그룹과 미팅을 통해 요구사항 및 필요한 뷰 개발.

최종 사용자: 캐주얼/초보/전문/독자적인 사용자

시스템 분석가 및 응용 프로그래머: 시스템 분석가는 사용자의 요구사항을 분석하여 기작성 트랜잭션 정의, 응용 프로그래머는 트랜잭션의 정의를 프로그램으로 구현, 유지, 관리



1.3

데이터베이스 시스템의 자기 기술성(메타데이터를 카탈로그 형태로 가지고 있다.) - 기존에는 데이터의 정의가 응용 프로그램의 일부. 따라서 하나의 특정된 데이터만 사용. 지금은 다양하게 사용 가능

프로그램과 데이터의 격리 및 데이터 추상화(데이터 파일의 구조가 응용 프로그램이 아닌 카탈로그에 저장. 파일 구조가 바뀌어도 응용 프로그램은 거의 변하지 않음)

다중 뷰(다수 사용자에게 각자 필요한 데이터만 제공 가능) - 기존에는 데이터 전부 다 불러와야 했음

데이터의 공유와 다수 사용자 트랜잭션 처리(데이터의 일관성 보장. ACID.) - 예전에는 중복 가능성 존재

1.4

관리자는 1.1, 설계자는 1.2

1.5

캐주얼 사용자: 데이터베이스에는 가끔 접근하지만 매번 다른 정보를 찾는다. 대개 중상급의 관리자로 정교한 데이터베이스 질의어 인터페이스를 사용하여 데이터를 요구한다.

초보 사용자: 기작성 트랜잭션을 사용하여 데이터베이스를 검색하고 변경하는 대부분의 최종 사용자. 비행기, 호텔, 렌터카 예약 직원과 고객은 원하는 예

약이 가능한지 살펴보고 예약한다. 이때 살펴보는 기능과 예약 기능은 미리 작성된 트랜잭션을 기반으로 한다.

독일 하루 후기 구독하기



전문 사용자: 전문적인 요구사항을 충족하기 위해 DBMS의 고급 기능을 이용하여 응용 프로그램을 구현한다.

독자적인 사용자: 개인 데이터베이스를 유지하는 사용자

1.6

정의: 데이터베이스에 저장되는 데이터에 대한 데이터 타입, 구조, 제약 조건들을 명세(분명하고 자세하게 함)하는 과정. 이런 정보가 카탈로그 형태로 저장되며 그 저장된 데이터를 메타데이터라고 한다.

생성(구축): DBMS가 관리하는 기억 장치에 데이터를 저장하는 과정.

조작: 특정한 데이터를 검색하기 위한 질의, 데이터베이스의 갱신, 데이터로부터 리포트를 생성하는 기능 등을 포함.

공유: 여러 사용자와 프로그램이 데이터베이스에 동시 접속 가능하게 하는 기능

중복성 제거: 동일한 정보를 여러 번 저장하지 않고 한 번만 저장. - 메모리 낭비와 데이터 불일치 예방 가능

보안/권한: 접근의 유형을 검색과 갱신으로 구분하여 제어

지속성 기억 공간 제공: 프로그램의 수행이 끝나도 데이터베이스에 영구적으로 남아 필요할 때 언제든지 다른 프로그램들에게 접근할 수 있는 지속성 객체를 데이터베이스에 저장

효율적인 질의 처리를 위한 저장 구조와 탐색 기법 제공: 인덱스처럼 데이터베이스 퍼포먼스 향상

독일 하루 후기 구독하기



백업과 회복: Data Recovery

GUI: 그래픽 인터페이스 제공

데이터 간의 관계 표현

무결성 제약 조건의 시행: 데이터에 결함이 없는 상태로 데이터가 정확하고 유효하다는 의미(데이터의 일관성). 예를 들어 기본키를 구성하는 모든 속성은 Null 값을 가질 수 없고, 유일한 값을 가져야 한다. 또 다른 예로는 section 테이블은 반드시 하나의 course 테이블 값과 연결이 되어야 한다고 제약조건을 걸었다면 course 값이 없는 section 테이블의 값이 있으면 안 된다.

기타: 데이터베이스를 보호하는 것과 이를 오랫동안 유지보수. Data Recovery(Back & Restore)

규칙을 사용한 추론과 수행: 연역 데이터베이스 시스템 - 이미 알려진 사실과 규칙을 기반으로 사용자의 질의에 답하는 것을 의미한다.

기타: 표준 강화, 응용 개발 시간 단축, 융통성, 최신 정보의 가용성, 규모의 경제성

1.7

데이터베이스 시스템: 데이터가 테이블 형태로 저장되어 있으며 각 테이블은 ACID 규칙에 따라 트랜잭션의 방해 없이 읽고, 쓰여 진다. 따라서 정보의 안정성이 우선시된다.

정보 검색 시스템: 축적된 정보에서 주어진 키를 가지고 요구에 적합한 정보를 꺼내는 시스템. NOSQL이 여기에 해당할 수도 있다. 사진, 영상과 같은 기

존과 다른 다양한 형태의 데이터를 저장할 수 있고, 빠르게 정보를 불러올 수 있다. 다만 정보의 중복이나 일관성은 떨어질 수 있다.

독일 하루 후기 구독하기



연습문제

1.8 체크 필요

```

/*1.8*/
/*Smith 성적을 검색하라*/
Select S.Name, S.Student_number, G.Grade
From Student as S
JOIN Grade_Report as G ON G.Student_number = S.Student_number;

/*2008 년 가을 학기에 개설된 데이터베이스 과목의 강사를 수강한 학생들의 이름과 성적을 검색하라*/
Select S.Name, G.Grade
From Student as S
JOIN Grade_Report as G ON G.Student_number = S.Student_number
JOIN Section as SE ON SE.Section_identifier = G.Section_identifier
Where SE.Semester = "Fall";

/* 데이터베이스 과목의 선수과목을 검색하라*/
Select C.Course_name
From Course as C
JOIN Prerequisite as P ON P.Course_number = C.Course_number
Where C.Course_name = "Database";

/*Smith를 2년생으로 변경하라*/
Update Student
SET Class = "2"
Where Name = "Smith";

/* 이번 학기에 데이터베이스 과목의 새로운 강사를 개설하라*/
Insert Into Section (Section_identifier, Course_number, Semester, Year, Instructor)
Values (200, "CS3380", "Winter", 21, "Lee"); /* 만약 section_identifier 번호가 generator에 의해서 정해지는 것이면 200 삭제*/

/* 지난 학기 데이터베이스 강좌에서 Smith의 성적을 A로 하라*/
Insert Into Grade_Report (Student_number, Section_identifier, Grade)
Values (17, 200, "A");

```

테이블이 미리 되어 있으면 자동완성이 될텐데 오타가 있을 가능성 있다. 이 책은 코드를 입력하라고 하는데 막상 입력하면 안된다고 하고, 관리자한테 연락하라고 하고... 심지어 그 코드가 적힌 종이는 어디있는지 더이상 안보이고... 코드를 다시 돈 주고 사야하고... 그렇다고 답이 있는 것도 아니고... 휴우,,,ㅎㅎ 총체적 난국이지만 반드시 해낼 것이다!

1.9

기본적으로 중복성이 제거되면 데이터 불일치를 예방하여 데이터의 일관성에 도움이 되고, 쓸데없는 메모리 낭비를 줄일 수 있다. 또한 중복된 데이터가 많으면 한번 수정할 때 모든 데이터를 수정해야 하는 번거로움이 있다. 따라서 데이터베이스에서는 데이터를 한 번만 저장하도록 하는데 이를 정규화라고 한다.

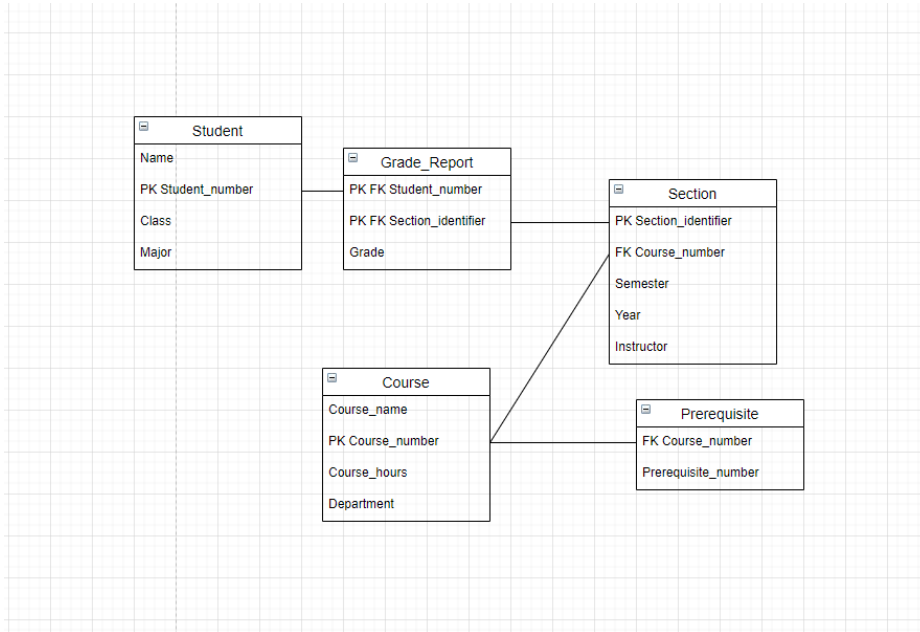
일부 자주 사용하는 항목에 한하여 검색 비용 측면에서 중복을 허용하는데 이를 반정규화라고 한다. 반정규화라고 하더라도 데이터의 일관성을 위한 제어

장치를 마련해야 하는데 이는 외래키를 이용해서 할 수 있다.

독일 하루 후기 구독하기



1.10 체크 필요



draw.io 로 작성함. 몇 대 몇 관계를 표시하는 Notation도 해야 하나?

1.11 체크 필요

이 대학교 컴퓨터 사이언스학과(CS)에 입학하고 싶은 학생이 자신이 입학할 연도의 직전 년도의 커리큘럼을 확인하고자 할 때 Section의 Year, Semester, Instructor와 Course에서 Course_name, Prerequisite 테이블을 이용해서 선수과목을 볼 수 있는 뷰를 만들 수 있을 것 같다.

1.12 체크 필요

데이터 타입을 일치시켜야 한다. 예를 들어 Student 테이블의 Name의 값은 30자 이내의 문자여야 한다.

Course 테이블에 Course_number를 PK 설정하여 Null이 없는 유일 값을 가지도록 한다.

독일 하루 후기 구독하기



Section 테이블의 Course_number 값을 Course 테이블의 Course_number 값과 연결시킨다.

1.13

인스타그램, 페이스북 같은 소셜 미디어 웹사이트, 대규모 전자 상거래 기업, 웹 검색 인덱스, 클라우드 저장소 또는 대용량 데이터베이스와 대규모 서버에 저장되는 데이터가 많은 시스템이 필요한 경우 의미가 있다.

테이블이 아닌 다양한 형태로 저장되어야 하는 데이터의 경우 의미가 있다.

1.14 체크 필요

a. Student의 Major, Course의 Course_number, Department, Section의 Course_number, Prerequisite의 Course_number, Prerequisite_number 가 업데이트되어야 한다.

b. 언급된 3개의 테이블은 Course_number 열을 가지고 있다. 따라서 각 테이블을 업데이트하지 않고, 한 번만 업데이트를 해서 3개의 테이블 모두 업데이트 효과를 주려면 Section과 Prerequisite 테이블이 해당 열을 스스로 가지고있는 것이 아니라 Course테이블에서 가져오면 된다.

Course	Prerequisite	Section
Course_name	Prerequisite_number	PK Section_identifier
PK Course_number		Semester
Course_hours		Year
Department		Instructor

정규화 작업을 통해서 애초에 테이블을 이런 식으로 만들고, SQL 질의를 통해서 Course_number를 불러오게 되면 업데이트는 한 번만 해도 세 테이블 모두 갱신된 정보를 불러올 수 있다.

[독일 하루 후기 구독하기](#)

처음으로 연습문제를 단 하나의 도움 없이 혼자 풀었다. 물론 책을 뒤적거리면서 풀긴 했지만 스스로 결과물을 만든 것 같아 뿌듯하다. 왜 연습문제를 풀어야 한다고 했는지 알겠다. 이해하고 생각하고 복습하는데 큰 도움이 된다. 답이 없는 게 아쉽다. 시간 날 때 구글 신에게 물어보고 찾아봐야겠다. 정답을 확인하고 싶다. 데이터베이스에서 조금씩 재미를 찾아가고 있다.

[♡ 공감 000](#)[구독하기](#)

'IT-hugi > Database' 카테고리의 다른 글

[DB] 데이터베이스 시스템 챕터03 연습문제 풀이 (0)	2021.03.01
[DB] 데이터베이스 시스템 챕터03 복습문제 풀이 (0)	2021.02.27
[DB] 데이터베이스 시스템 챕터02 연습 문제 풀이 (0)	2021.02.14
[DB] 데이터베이스 시스템 챕터02 복습 문제 풀이 (0)	2021.02.13
<u>[DB] 데이터베이스 시스템 복습/연습문제 풀이 챕터01 (2)</u>	2021.02.04

태그

[데이터베이스 시스템 챕터1](#)[복습문제풀이](#)[연습문제풀이](#)

관련글