데이터베이스 보안 실습

Saerom Park

secure.PSR@gmail.com

데이터베이스 보안 실습

❖ Instructor

- 박새롬
- Office: room 503
- Email: secure.psr@gmail.com/ psr6275@sungshin.ac.kr
- GitHub: github.com/psr6275

- Office hour: 월 14:00-15:00

❖ Lecture

- 시간: 월 789 (15시~18시)
- Textbook
  - 전반부: 데이터베이스 시스템 (7판) - Elmasri Navathe, 황규용, 홍의경, 음두헌 옮김, 홍릉과학출판 - [황]
  - 후반부: 데이터베이스 보안: 디자인 및 구현 - 조은백, 생농출판 - [조]

데이터베이스 보안 실습

❖ Homeworks

- 총 2회
- Due date: 숙제 공지날로부터 일주일 후 (수업시간 전까지 -> 15:00)
- Late penalty: 하루에 20% 씩 감점
  - 1일: 숙제점수 = 채점점수 * 0.8 | 2일: 숙제점수 = 채점점수 * 0.6
  - 3일: 숙제점수 = 채점점수 * 0.4 | 4일: 숙제점수 = 채점점수* 0.5
  - 5일 이후: 점수 없음
- Copy 시: 점수 없음.

❖ 시험

- 중간고사: 10/21 월 15:00 ~ 17:00
- 기말고사: 12/9 월 15:00 ~ 17:00

❖ 평가

- 중간고사 (35%), 기말고사 (35%), 과제물 (20%), 출석 (10%)

Class Schedule

날짜	내용	참고	비고
9/2	강의소개 및 오리엔테이션/ 데이터베이스 개요		
9/9	데이터베이스 시스템 개념과 아키텍처	Ch 2	
9/16	엔터티-관계 모델/관계 데이터 모델	Ch 3, 4	
9/23	SQL 개요	Ch 6	
9/30	SQL 기본 연산	Ch 7	
10/7	SQL 프로그래밍	Ch 9,10	숙제 1 공지
10/14	관계형 데이터 베이스의 설계	Ch 8	숙제 1 제출
10/21	**중간고사 15:00-17:00**		
10/28	데이터 베이스 보안 기본/ 엑세스 제어	Ch 20, 21, 27	
11/4	애플리케이션 보안 기본/ SQL 인젝션	Sec 1, 2	
11/11	절차형 언어 인젝션/ 웹 공격	Sec 2	
11/18	MySQL 서버 보안	Sec 3, 4	숙제2 공지
11/25	유저 및 권한 관리	Sec 5	숙제2 제출
12/2	데이터 암호화	Sec 9, Ch 27	
12/9	**기말고사 15:00-17:00**		

Q & A

데이터베이스 개요

Introduction

❖ 데이터베이스 관리 시스템 (DBMS)

- 서로 관계있는 데이터들의 모임과 그 데이터에 접근하기 위한 프로그램의 집합
  - 데이터베이스 (database): 데이터들의 모임
  - 정보를 관리: 저장 구조 정의/ 저장된 정보 조작
  - 데이터 베이스 보안
    - 저장된 정보를 시스켐 고장이나 모든 불법적인 액세스 등으로 부터 안전하게 보호
    - 예기치 않은 이상 결과 방지

❖ 데이터 베이스의 등장

- 사용자들의 직접 접근이 일반화
- 사용자들의 직접 조작이 증가

➡ 데이터 베이스 응용 프로그램의 필요성 증대

파일 처리 시스템의 단점 [1/2]

❖ 데이터 중복과 비일관성

- 통일되지 않은 파일 및 응용 프로그램

❖ 데이터 엑세스 시의 난점

- 필요한 데이터를 편리하고 효율적으로 검색하기 힘듦

❖ 데이터의 고립

- 파편화된 저장 및 호환되지 않는 파일 형식 가능

❖ 무결성 (integrity) 문제

- 일관성 (inconsistency) 제약조건

파일 처리 시스템의 단점 [2/2]

❖ 원자성 문제

- 일련의 과정 전체가 수행 되든지 아니면 아무것도 수행되지 않아야 함
- 예) 예금이체

❖ 동시 엑세스 문제

- 데이터 동시 갱신 시 비일관성 야기 가능

❖ 보안 문제

- 엑세스 관리 필요

데이터의 관점 (Database) [1/2]

❖ 데이터베이스 시스템의 목적

- 사용자에게 데이터에 관한 추상적인 관점 제공
- 데이터가 실제로 어떻게 저장되고 유지되는지에 관한 세부 사항은 은폐

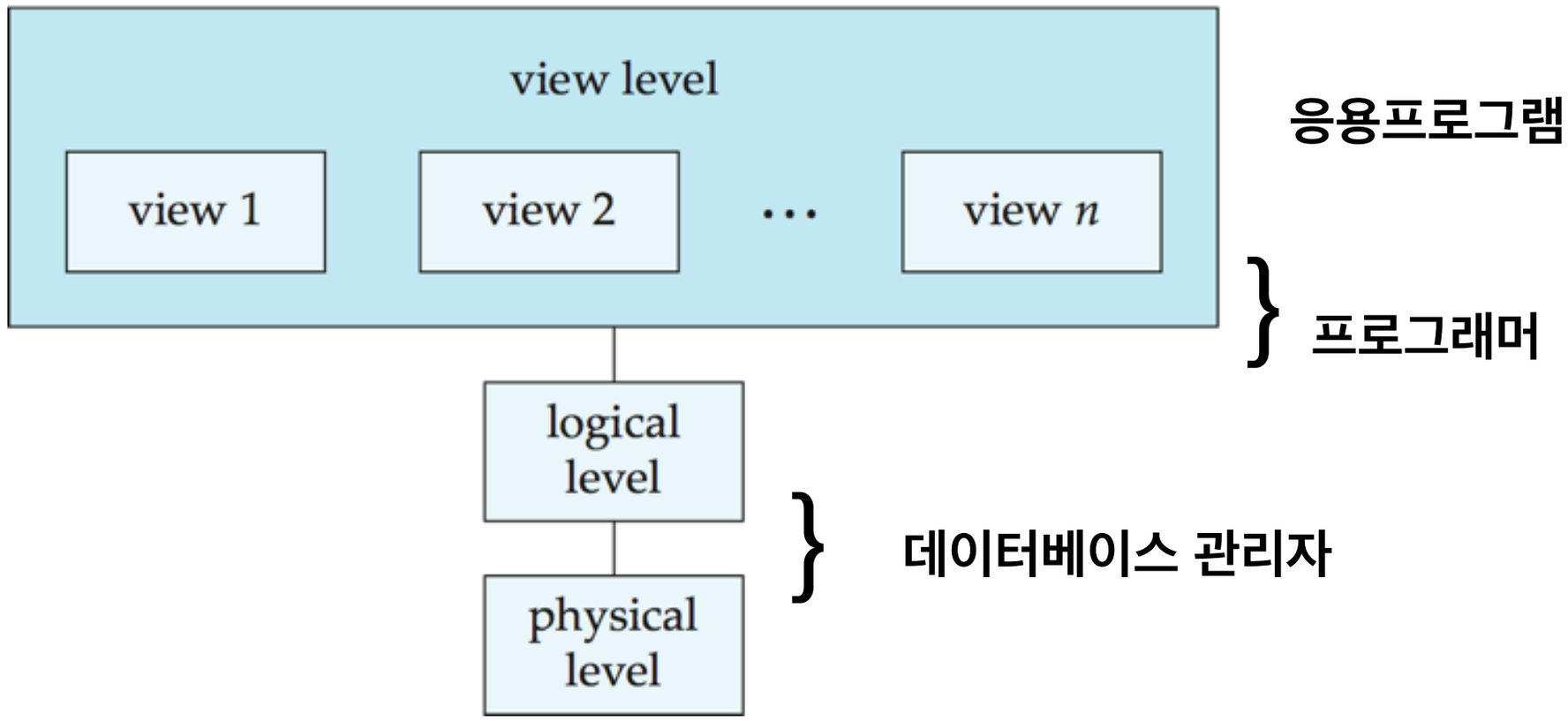
❖ 데이터의 추상화

- 물리적 단계 (physical level)
  - 데이터가 실제로 어떻게 저장되는지 기술
- 논리적 단계 (logical level)
  - 어떤 데이터가 저장되었는지, 데이터들 사이에 어떤 관계가 있는지를 기술
  - 물리적 데이터 독립성 (physical data independence)
  - 데이터 베이스 관리자 (database administrator: DBA)
- 뷰 단계 (view level)
  - 추상화의 최상위 단계. 요청에 따라 데이터 베이스의 일부분만을 기술
  - 응용 프로그램 사용

데이터의 관점 (Database) [2/2]

❖ 데이터의 추상화

- 물리적 단계 (physical level)
- 논리적 단계 (logical level)
- 뷰 단계 (view level)



데이터 베이스 기본

❖ 인스턴스 스키마

- 데이터 베이스는 정보가 추가되고 삭제됨에 따라 변함
- 인스턴스 (instance): 어느 특정한 순간에 데이터베이스에 저장되어 있는 정보의 모임
- 스키마 (schema): 데이터 베이스의 전체적인 설계
  - 물리적 스키마 (physical schema): 물리적 단계에서 데이터베이스 설계
  - 논리적 스키마 (logical schema): 논리적 단계에서 데이터 베이스 설계
  - 서브 스키마 (subschema): 필요에 따라 서로 다른 뷰를 기술하는 여러개의 뷰 단계의 스키마

❖ 데이터 모델

- 데이터, 데이터 사이의 관계, 데이터의 의미, 일관성 제약조건 등을 기술하기 위한 개념적 표현들의 집합
- 물리적, 논리적, 뷰 단계에서 데이터 베이스 설계하는 방법 제공

➡ 관계형 모델 (relational model), 개체-관계 모델 (entity-relationship model), 객체-기반 데이터 모델 (object-based data model), 반구조형 데이터 모델 (semistructured data model)

데이터베이스 언어 [1/2]

❖ 데이터 베이스 언어 종류

- 데이터 조작 언어 (data manipulation language: DML)
  - 데이터베이스의 질의 및 갱신을 표현
- 데이터 정의 언어 (data definition language: DDL)
  - 데이터베이스 스키마를 기술
- 경계가 명확히 구분되어 있지는 않음. 예) SQL

❖ 데이터 조작 언어 (DML)

- 사용자가 적절한 데이터 모델로 구성된 데이터를 접근하거나 조작할 수 있도록 하는 언어
  - 저장된 정보 검색 | 새로운 정보 삽입 | 정보 삭제 | 저장된 데이터 수정
- 절차식 DML: 어떤 데이터가 필요하며 그 데이터를 어떻게 구할지 지정
- 선언적 DML: 필요한 데이터를 어떻게 구할지 명시 없이 어떠한 데이터가 필요한지만 지정
- 질의 (query): 정보의 검색을 요청하는 문장
- 질의어 (query language): 데이터 조작 언어에서 정보의 검색을 담당하는 부분

데이터베이스 언어 [2/2]

❖ 데이터 정의 언어 (DDL)

- 데이터 베이스의 스키마의 정의를 표현하는데에 사용 되는 언어이며, 데이터의 추가적 특성을 표현하는 데에도 사용.
  - 데이터 저장 및 정의 언어 (data storage and definition language): 저장 구조 storage structure)와 엑세스 방법을 지정함. 스키마 구현상의 세부 사항 정의
  - 명령문 (statement) → 메타데이터 (metadata: 데이터를 위한 데이터)
  - 데이터 사전 (data dictionary): 메타 데이터를 수록, 데이터베이스 시스템에 의해서만 접근되고 갱신될 수 있음.
- 무결성 제약조건
  - 도메인 제약조건 (domain constraints): 지정된 도메인 중 하나의 타입을 가져야 함
  - 참조 무결성 (referential integrity): 주어진 속성들의 집합에 대한 릴레이션의 한 값이 다른 릴레이션에 대한 속성 집합의 값으로 반드시 나타냐아 할 경우가 있음.
  - 주장 (assertions): 데이터 베이스가 항상 만족시켜 주어야 하는 조건
  - 권한 (authorization): 다양한 데이터들에 대해 사용자들마다 접근을 다르게 할 경우