



Chapter 1: 개요

Database System Concepts, 6th Ed.

©Silberschatz, Korth and Sudarshan

See www.db-book.com for conditions on re-use



데이터베이스 관리 시스템 (DBMS)

- DBMS는 특정 분야 관련 정보를 저장/관리한다
 - 상호 관련 있는 데이터의 모임
 - 데이터를 액세스하기 위한 프로그램의 집합
 - 사용하기 편하고 효율적인 환경을 제공
- Database 응용분야:
 - Banking: transactions
 - Airlines: reservations, schedules
 - Universities: registration, grades
 - Sales: customers, products, purchases
 - Online retailers: order tracking, customized recommendations
 - Manufacturing: production, inventory, orders, supply chain
 - Human resources: employee records, salaries, tax deductions
- 데이터베이스는 일반적으로 그 크기가 매우 크다고 가정한다.



대학 데이터베이스예제

- 응용 예제
 - 새로운 학생, 강사(instructor), 코스(course) 입력하기
 - 각 코스에 학생 등록하고, 클래스 명단 생성하기
 - 학생에게 학점(grade) 부여하고, 평점(grade point average, GPA) 계산하고, 성적 증명서(transcript) 생성하기



DB시스템의 목적

DBMS는 전통적인 OS 가 지원하는 파일 처리 시스템의 아래와 같은 문제점을 처리하기 위해 개발 되었음.

- 데이터의 중복과 불일치
- 데이터 액세스상의 어려운 점
- 데이터의 고립성 - 여러 파일과 포맷
- 무결성 문제
- 갱신의 원자성
- 여러 사용자에게 의한 동시 액세스
- 보안 문제



데이터 모델

- 아래 사항을 기술하는 도구들의 모임
 - Data
 - Data 관련성(relationships)
 - Data 의미(semantics)
 - Data 제약조건(constraints)
- 릴레이션 모델 (Relational model)
- E-R 모델 (Entity-Relationship data model): DB 설계에 사용됨
- 객체기반 모델 (Object-based data models, Object-oriented and Object-relational)
- Semistructured data model (XML)
- 기타 :
 - 네트워크 모델 (Network model)
 - 계층 모델 (Hierarchical model)



릴레이션 모델

- Relational model (Chapter 2)
- Example of tabular data in the relational model

Columns

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Rows

(a) The *instructor* table



A Sample Relational Database

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

<i>dept_name</i>	<i>building</i>	<i>budget</i>
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table



Database Design?

- Is there any problem with this design?

<i>ID</i>	<i>name</i>	<i>salary</i>	<i>dept_name</i>	<i>building</i>	<i>budget</i>
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000



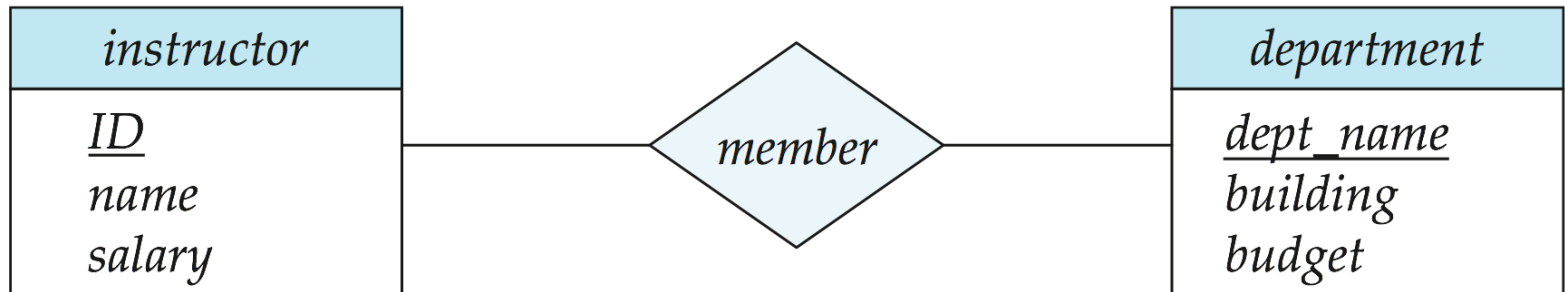
Design Approaches

- 정규화 이론 (Chapter 8)
 - Formalize what designs are bad, and test for them
- E-R 모델 (Chapter 7)
 - Models an enterprise as a collection of *entities* and *relationships*
 - ▶ Entity: a “thing” or “object” in the enterprise that is distinguishable from other objects
 - Described by a set of *attributes*
 - ▶ Relationship: an association among several entities
 - Represented diagrammatically by an *entity-relationship diagram*:



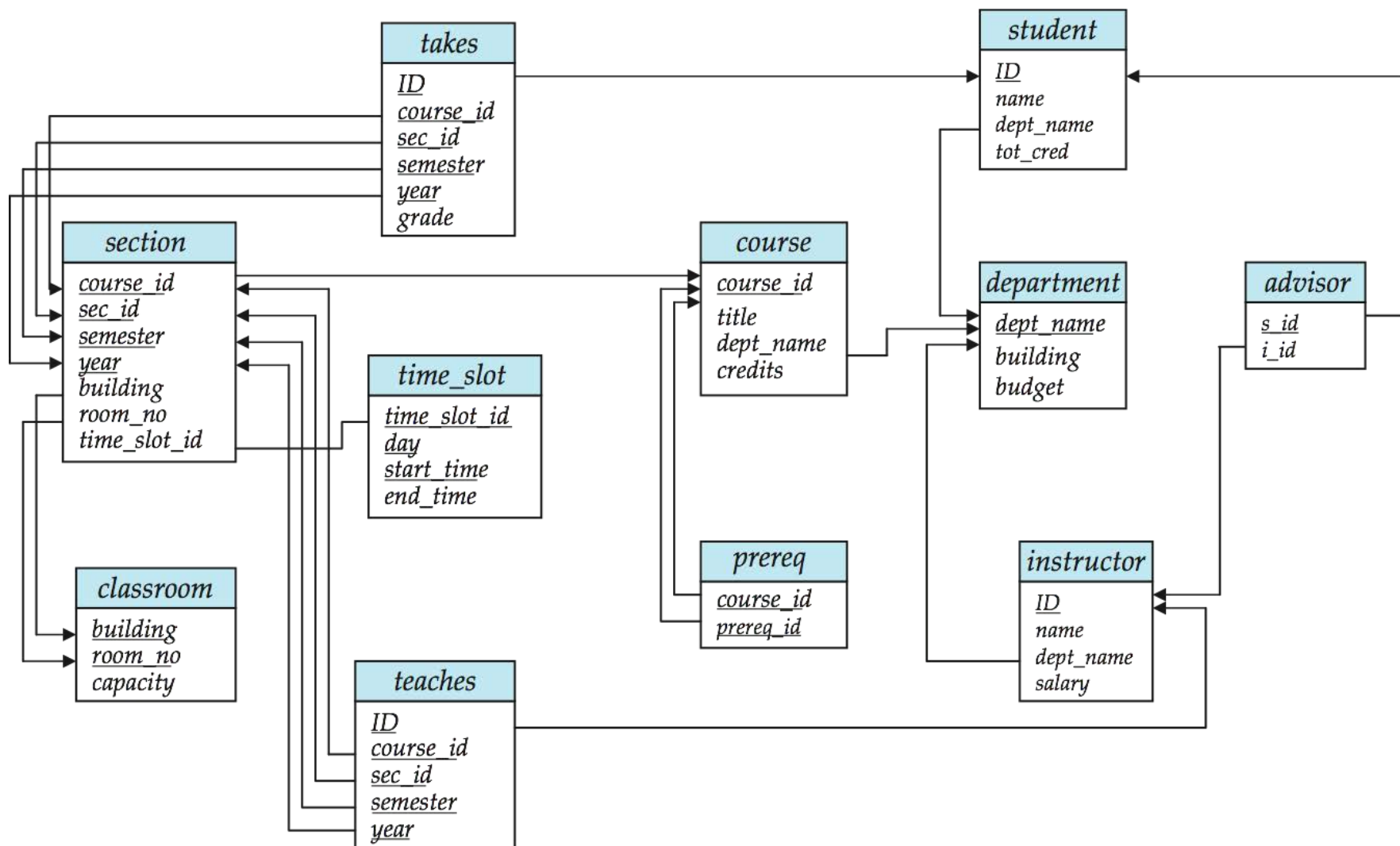
개체-관계 (Entity-Relationship, E-R) 모델

- 개체와 관계를 이용하여 데이터베이스를 모델링 한다.
 - 개체 (Entity): a “thing” or “object” in the enterprise that is distinguishable from other objects
 - ▶ 속성 집합에 의하여 표현된다.
 - 관계 (Relationship): an association among several entities
- *E-R* 다이어그램으로 표현할 수 있다:





Schema Diagram for University Database





데이터 정의를어 (Data Definition Language, DDL)

- DB 스키마를 정의하기 위한 명세 표기
 - Example: **create table** *instructor* (

<i>ID</i>	char (5),
<i>name</i>	varchar (20),
<i>dept_name</i>	varchar (20),
<i>salary</i>	numeric (8,2))
- DDL 컴파일러는 데이터 사전 (***data dictionary***) 에 저장되는 테이블의 집합을 생성
- 데이터 사전에는 메타 데이터가 저장된다 (즉, 데이터에 관한 데이터)
 - Database schema
 - Integrity constraints
 - ▶ Primary key (ID uniquely identifies instructors)
 - ▶ Referential integrity (**references** constraint in SQL)
 - e.g. *dept_name* value in any *instructor* tuple must appear in *department* relation
 - Authorization



SQL

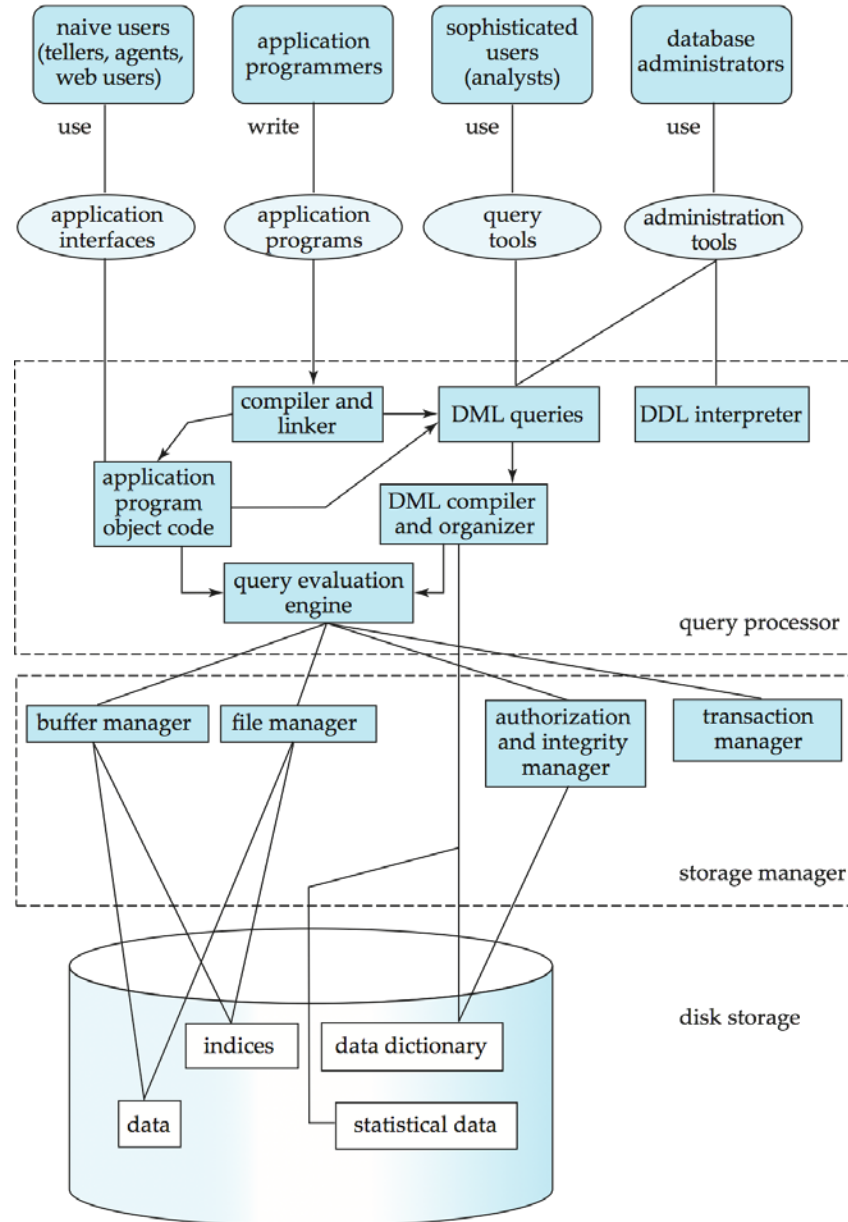
- **SQL**: widely used non-procedural language
 - Example: Find the name of the instructor with ID 22222

```
select   name
from     instructor
where    instructor.ID = '22222'
```
 - Example: Find the ID and building of instructors in the Physics dept.

```
select instructor.ID, department.building
from   instructor, department
where  instructor.dept_name = department.dept_name and
        department.dept_name = 'Physics'
```
- 응용 프로그램은 다음 중 하나의 방법을 이용하여 DB에 액세스한다
 - Language extensions to allow embedded SQL
 - Application program interface (e.g., ODBC/JDBC) which allow SQL queries to be sent to a database
- Chapters 3, 4 and 5



Database System Internals





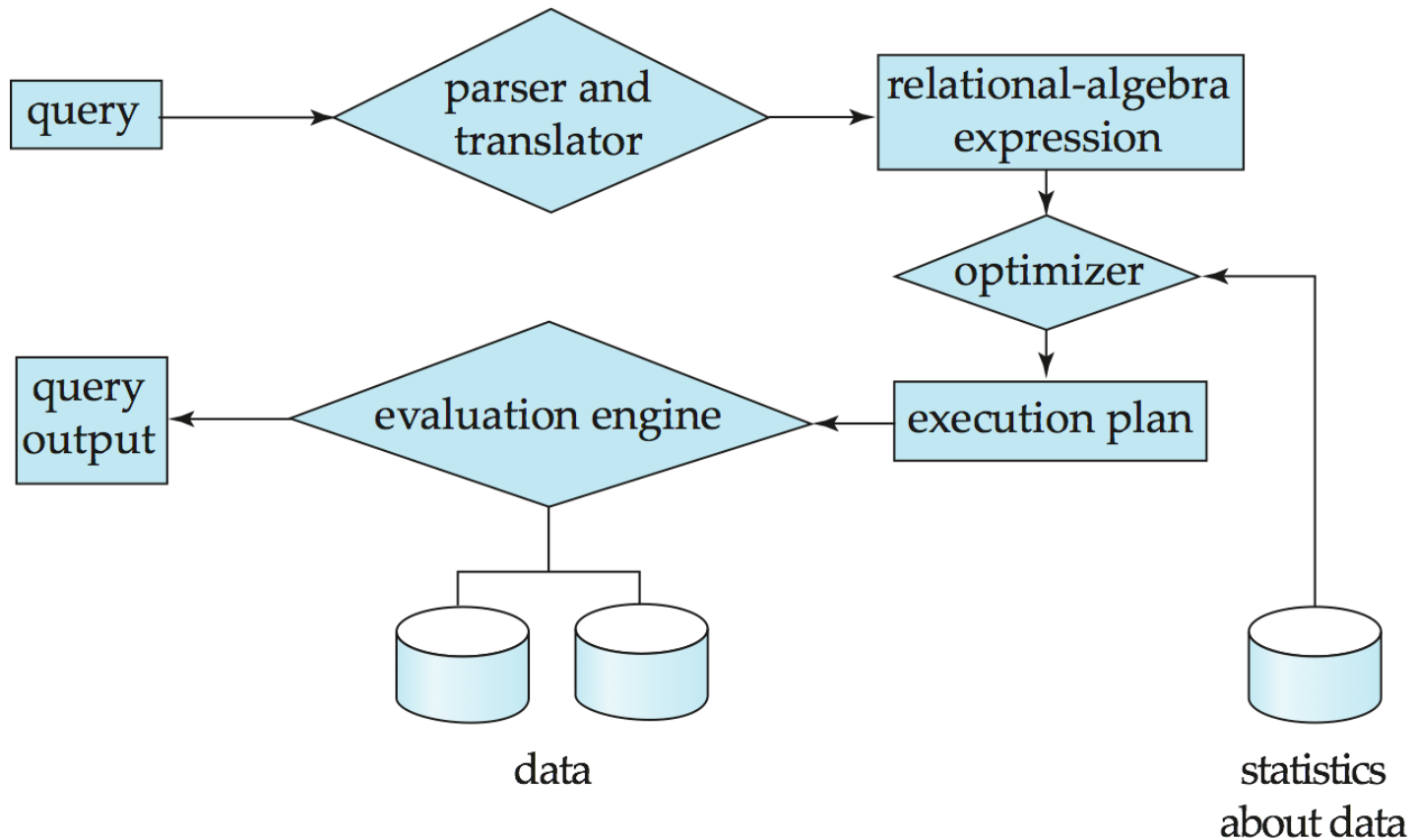
저장 장치 관리 (Storage Management)

- 저장 장치 매니저는 데이터베이스에 저장된 하위 단계 데이터와 시스템에 제기된 어플리케이션 프로그램 및 질의 간에 인터페이스를 제공하는 프로그램 모듈이다.
- 저장 장치 매니저는 다음과 같은 임무를 수행한다.
 - 파일 매니저와의 상호 작용
 - 효율적인 데이터의 저장, 검색 및 갱신
- 문제점 :
 - Storage access
 - File organization
 - Indexing and hashing



질의 처리 (Query Processing)

1. Parsing and translation
2. Optimization
3. Evaluation





트랜잭션 관리 (Transaction Management)

- What if the system fails?
- What if more than one user is concurrently updating the same data?
- 트랜잭션은 DB 어플리케이션 내에서 하나의 논리적 기능을 수행하는 연산들의 모임이다.
- **Transaction-management component** : 트랜잭션 관리 구성 요소는 시스템 고장(정전 및 운영체제 손상)과 트랜잭션의 실패에도 불구하고 데이터베이스가 일관성 있는(정확한) 상태를 유지하도록 보장
- **Concurrency-control manager** : 동시성 제어 매니저는 DB의 일관성을 보장하기 위해 동시 실행 트랜잭션 간의 상호 작용을 통제한다.



End of Chapter 1