

데이터베이스 및 설계

Chap 3. 데이터베이스 시스템 구성



2013.03.14.

오 병 우

컴퓨터공학과

Database System

● Data

- ◆ Facts collected from observations and measurements

● Database

- ◆ A collection of persistent data that is used by the application systems of some given enterprise

● Database Management System

- ◆ A layer of software between the physical database and the user

● Database System

- ◆ A computerized system whose overall purpose is to maintain information and to make that information available on demand

“데이터를 데이터베이스로 저장 관리하여 정보를 생성하는 컴퓨터 중심 시스템”

Purpose

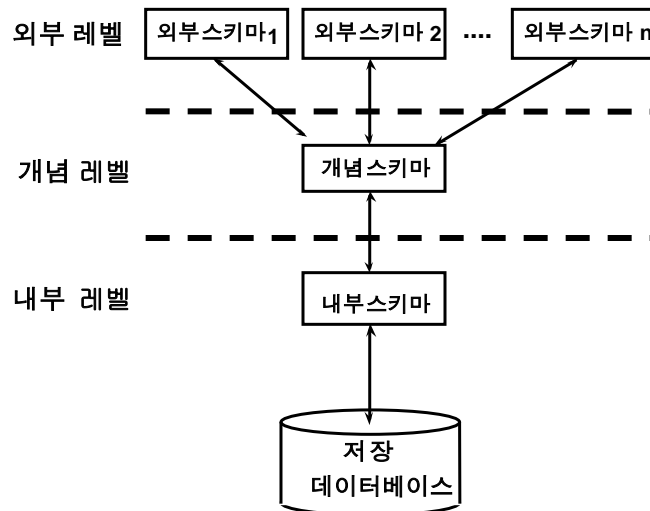
- Architecture for a database system → framework
 - ◆ For describing general database concepts
 - ◆ For explaining the structure of specific database systems

- ANSI/SPARC Study Group on Database Management System

The Three Levels of The Architecture

ANSI/SPARC three level architecture (3 레벨 구조)

- ◆ External level (외부 레벨) → individual user view
 - The one concerned with the way the data is viewed by individual users
- ◆ Conceptual level (개념 레벨) → community view
 - A level of indirection between the other two
- ◆ Internal level (내부 레벨) → storage view
 - The one concerned with the way the data is physically stored



Views

- 데이터베이스에 대한 관점(View)
- Three views
 - ◆ External view (many distinct): 개인
 - More or less abstract representation of some portion of the total database
 - ◆ Conceptual view (one): 기관 전체
 - Abstract representation of the database in its entirety
 - ◆ Internal view: 저장 장치
 - Total database as physically stored

Schema

- 각 레벨에서의 스키마

- ◆ 데이터베이스의 구조(데이터 객체, 관계)와 제약 조건의 명세

- 외부 스키마(external schema)

- ◆ 개개 사용자 관점에서의 데이터베이스 정의
- ◆ 전체 데이터베이스의 한 논리적인 부분
 - Sub Schema

- 개념 스키마(conceptual schema)

- ◆ 범 기관적인 관점에서의 데이터베이스 정의
- ◆ 모든 응용에 대한 전체적인 통합된 데이터 구조
 - 일반적인 Schema

- 내부 스키마(internal schema)

- ◆ 저장 장치 관점에서의 데이터베이스 정의
- ◆ 개념 스키마에 대한 저장구조를 정의

학생 데이터베이스 예제

외부 스키마 1
(학적과)

ST	
Sn	INT
Name	CHAR(10)
Grade	INT
Dept	CHAR(5)

외부 스키마 2
(학생과)

STUDENT	
Sno	PIC 9(4)
Sname	PIC X(10)
Year	PIC 9(2)
Addr	PIC X(44)

개념 스키마

STUDENT	
Snumber	INTEGER
Name	CAHR(10)
Year	SMALLINT
Grade	SMALLINT
Dept	CHAR(5)
Address	CHAR(44)

구교재 p.44
바로잡을 것

내부 스키마

STORED-STUDENT			LENGTH = 71	
prefix	BYTE(4)	OFFSET = 0		
Sno	BYTE(4)	OFFSET = 4		INDEX = STINDX
Sname	BYTE(10)	OFFSET = 8		
Syear	BYTE(2)	OFFSET = 18		
Sgrade	BYTE(2)	OFFSET = 20		
Sdept	BYTE(5)	OFFSET = 22		
Saddr	BYTE(44)	OFFSET = 27		

Mapping

Two levels of mapping

◆ Conceptual / internal mapping

- Correspondence between the conceptual view and the stored database (i.e., how conceptual records and fields are represented at the internal level)
 - Structure of the stored database : change (i.e., change in the storage structure definition)
 - Conceptual / internal mapping : change (by DBA) (∵ conceptual schema : invariant)
- ↓
- } for “data independence”

관련성

External / conceptual mapping

◆ Correspondence between a particular external view and the conceptual view

- Field can have different data types
- Field and record names can be changed
- Multiple conceptual fields can be combined into a single (virtual) external field

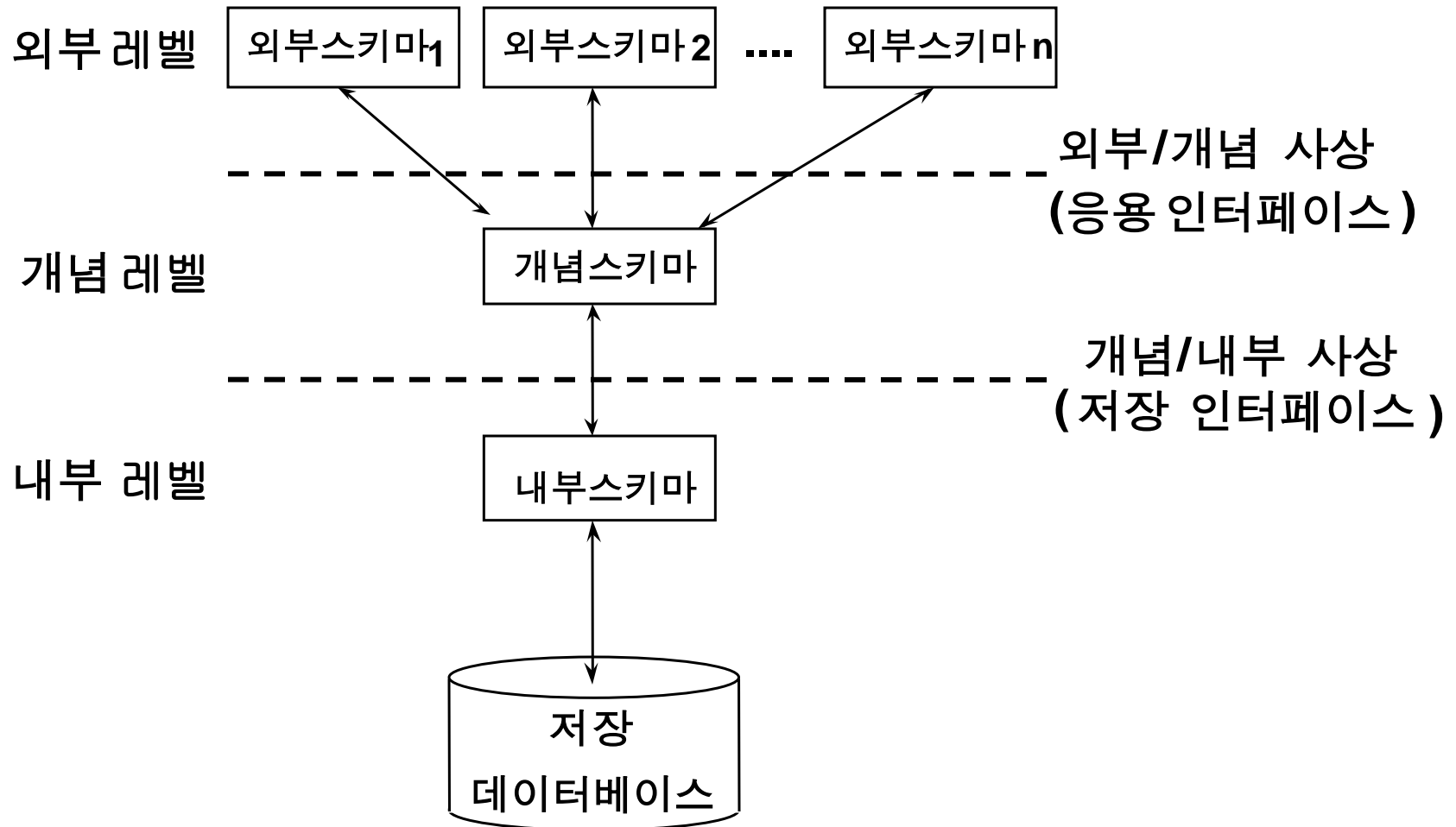
Mapping

- 외부/개념 단계간의 사상
 - ◆ 외부 스키마와 개념 스키마간의 대응 관계 정의
 - ◆ 응용 인터페이스 (application interface)
 - ◆ 논리적 데이터 독립성을 제공
- 개념/내부 단계간의 사상
 - ◆ 개념 스키마와 내부 스키마간의 대응 관계 정의
 - ◆ 저장 인터페이스(storage interface)
 - ◆ 물리적 데이터 독립성을 제공

☞ Note

- ◆ 3단계간의 사상
 - ⇒ 데이터 독립성(data independency)의 구현방법
- ◆ 스키마는 컴파일되어 data dictionary (데이터 사전, 카탈로그)에 저장
- ◆ 스키마 다이어그램
 - ⇒ 스키마의 그래픽 형태로 표현

Mapping



The External Level

- External view : some portion & abstract
 - ◆ The content of the database as it is seen by some particular user
 - ◆ Subset of conceptual view
 - ◆ Multiple occurrences of multiple types of external record (=logical record)
 - ◆ DML
- External schema
 - ◆ Definitions of each of the various types of external record
 - ◆ (external) DDL
- cf. (비교, 참조)
 - ◆ View: DB 내용
 - A set of occurrences
 - ◆ Schema: DDL로 정의한 것
 - Definition of a view
 - ◆ Model: Relation (table), Network model, etc.
 - Structure, operation, constraint

존재하는 것

Language of the External Level

● Language

◆ Data sublanguage (DSL)

- A subset of the total language that is concerned specially with database objects and operations
- Embedded within the corresponding host language

◆ Host language

- Responsible for providing various non-database facilities
- e.g., local variables, computational operations, if-then-else logic, etc.

● Two types

◆ Tightly coupled (ODBC, OLE/DB): 많이 사용됨

- If the two are indistinguishable, or if they can be separated only with difficulty

◆ Loosely coupled (Embedded SQL): 최근에는 잘 사용되지 않음

- If the two are clearly and easily separable

Language of the External Level (cont'd)

● Data sublanguage

- ◆ Data definition language (DDL)
 - For the definition or declaration of database objects
- ◆ Data manipulation language(DML)
 - For the manipulation or processing of such objects
- ◆ Data control language (DCL)
 - Security, integrity, trigger, etc.

The Conceptual Level

Conceptual view: abstract

- ◆ A representation of the entire information content of the database
- ◆ Multiple occurrences of multiple types of conceptual record

Conceptual schema

- ◆ Definitions of each of the various types of conceptual record
- ◆ (conceptual) DDL
- ◆ No considerations of $\left\{ \begin{array}{l} \text{storage structure or} \\ \text{access strategy} \end{array} \right\}$ → For “data independence”
- ◆ No reference to stored field representations, stored record sequence, indexing, hash-addressing, pointers, etc.
- ◆ Additional features: security checks, integrity checks, etc.

The Internal Level

Internal view (\equiv stored database)

- ◆ A low-level representation of the entire database
- ◆ Multiple occurrence of multiple types of internal record(=stored record)
- ◆ At one remove from the physical level
 - ∴ not deal in terms of physical records (pages or blocks), nor with any device-specific considerations (e.g., cylinder or track size)
- ◆ Assume an infinite linear address space

동떨어진

Internal schema (\equiv storage structure definition)

- ◆ Define the various types of stored record and specify what indexes exist, how stored fields are represented, what physical sequence the stored records are in, and so on.
- ◆ (internal) DDL

In certain exceptional situations (not recommended)

- ◆ Application program operates directly at the internal level
 - To obtain the required function or performance

Metadata

● 데이터 사전 (data dictionary)

- ◆ 데이터베이스에 저장되어 있는 모든 개체들에 대한 정의나 명세에 관한 정보를 유지/관리
 - 여러 스키마와 이들에 포함된 mapping 정보도 저장/관리
- ◆ 시스템 데이터베이스
- ◆ 시스템 카탈로그(catalog)라고도 함

● Metadata: data about data

- ◆ 시스템 데이터베이스에 저장되어 있는 데이터

● 데이터 디렉토리(data directory)

- ◆ 시스템 데이터베이스에 수록된 데이터를 실제로 접근하는데 필요한 위치 정보를 관리하는 시스템
- ◆ 차이점
 - 데이터 사전: 사용자와 시스템 모두 사용
 - 데이터 디렉토리: 시스템만 사용

DDL

● Data Language

- ◆ 데이터베이스의 정의, 조작, 제어를 위한 시스템과의 통신수단

● DDL : Data Definition Language

- ◆ 데이터베이스의 정의 및 수정

● 정의 내용

- ◆ Logical 데이터 구조의 정의

- 스키마, 외부 스키마의 기술

- ◆ Physical 데이터 구조의 정의

- 내부 스키마 기술

- 데이터 저장 정의 언어 (Data Storage Definition Language)

- ◆ 논리적 데이터 구조와 물리적 데이터 구조 간의 mapping 정의

DML

● DML: Data Manipulation Language

- ◆ 사용자(응용 프로그램)와 DBMS 사이의 통신 수단
- ◆ 데이터 처리 연산의 집합 (데이터의 검색, 삽입, 삭제, 변경 연산)

● 절차적 (procedural) DML

- ◆ what과 how를 명세
- ◆ Host Language (tightly coupled, loosely coupled)
- ◆ 한번에 하나의 레코드만 처리
- ◆ Loosely coupled Language (DSL)
 - 응용 프로그램 속에 삽입(embedded)되어 사용
 - DML Pre-compiler에 의해 처리

Completeness of data language:
 사용자가 원하는 어떤 객체도
 추출할 수 있고 관계도
 표현할 수 있는 능력

● 비절차적 (non-procedural) DML

- ◆ what만 명세 (declarative, descriptive)
- ◆ SQL
- ◆ 한번에 여러 개의 레코드 처리
- ◆ 질의어 (Query Language)
- ◆ 독자적, 대화식 사용 : 커맨드 타입
- ◆ 프로그램의 추상적 표현

DCL

- DCL : Data Control Language
 - ◆ 공용 데이터베이스 관리를 위해 데이터 제어를 정의하고 기술

- 데이터 제어 내용
 - ◆ 데이터 보안(security)
 - ◆ 데이터 무결성(integrity)
 - ◆ 데이터 회복(recovery)
 - ◆ 병행 수행(concurrency)

- 관리 목적으로 데이터베이스 관리자(DBA)가 사용

사용자(User)

- 데이터베이스를 이용하기 위해 접근하는 사람
- 일반 사용자(end user)
 - ◆ 비절차적 DML(질의어)을 통해서 데이터베이스를 접근
 - ◆ menu, form, graphics
 - ◆ 데이터의 삽입, 삭제, 갱신, 검색
- 응용 프로그래머(application programmer)
 - ◆ 호스트 언어 + DML(DSL)을 통해서 데이터베이스를 접근
 - ◆ PL/I, COBOL, PASCAL, C, Java
- 데이터베이스 관리자 (DBA)
 - ◆ DDL과 DCL 을 통해 DB를 정의하고 제어하는 사람

데이터베이스 관리 시스템(DBMS)

- 사용자의 요구에 따라 데이터베이스에 대한 모든 연산을 수행
 1. 사용자의 접근 요구를 접수/분석
 2. 시스템이 이해할 수 있는 형태로 요구를 변환
 3. 외부/개념/내부/저장 구조간의 사상을 수행
 4. 저장 데이터베이스에 대해 연산을 실행

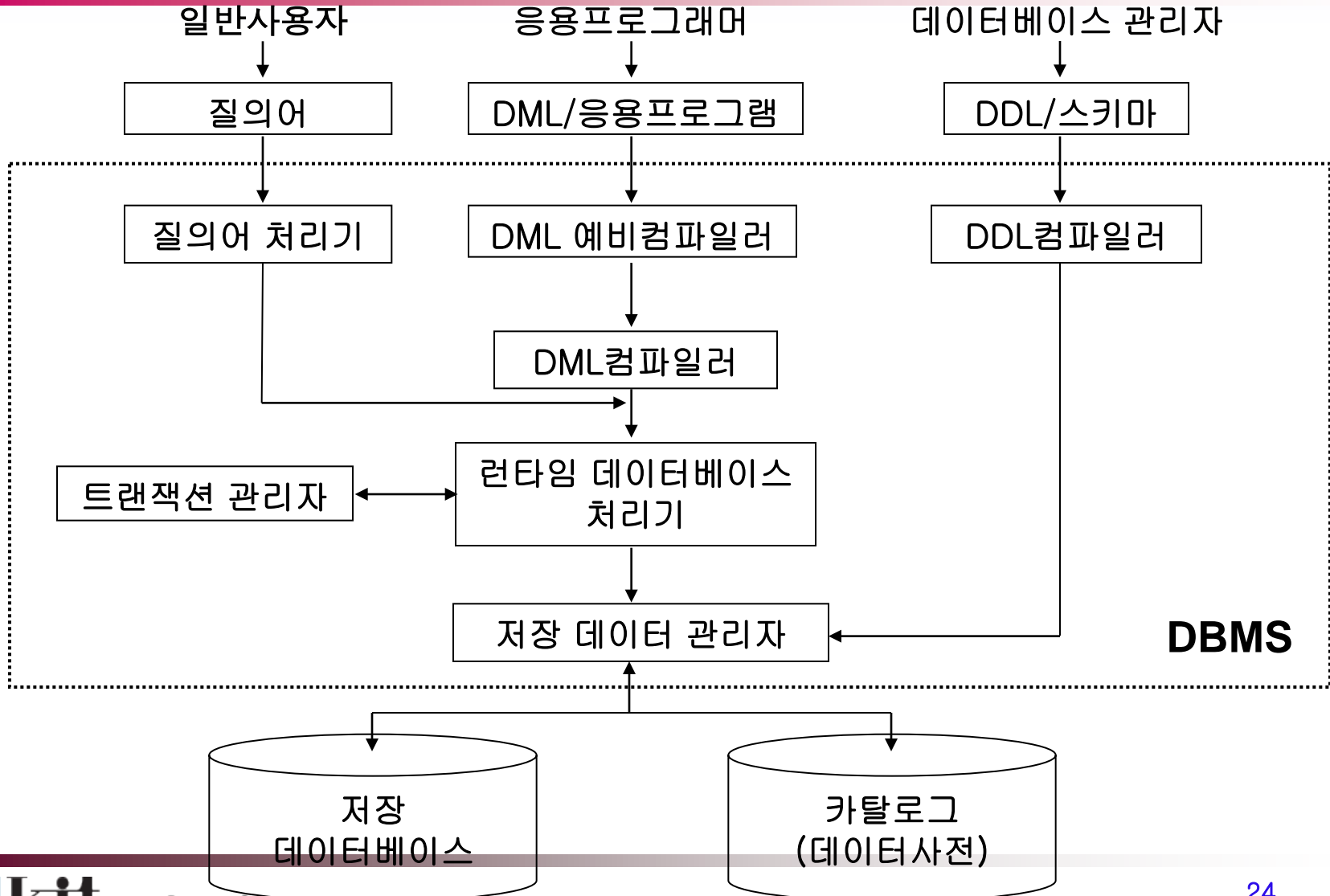
DBMS의 구성요소(1)

- DDL 컴파일러
 - ◆ DDL로 명세된 스키마를 내부 형태로 변환하여 카탈로그에 저장
 - ◆ 메타 데이터: 파일의 이름과 크기, 데이터 아이템의 이름과 타입, 저장 세부 사항, 스키마 간의 사상 정보, 제약 조건
- 질의어 처리기
 - ◆ 질의문을 파싱, 분석, 컴파일하고 목적 코드를 생성
- 예비 컴파일러 (pre-compiler, pre-processor)
 - ◆ 응용 프로그램에 삽입된 DML(DSL)을 추출하고 그 자리에 procedure call로 대체
 - ◆ 추출된 DML은 DML 컴파일러로 전달
- DML 컴파일러
 - ◆ DML 명령어를 Object 코드로 변환

DBMS의 구성요소(2)

- 런타임 데이터베이스 처리기
 - ◆ 실행 시간에 데이터베이스를 접근
 - ◆ 데이터베이스 연산을 수행
- 트랜잭션 관리자
 - ◆ 무결성과 권한 제어
 - ◆ 병행 제어와 회복 작업
- 저장 데이터관리자
 - ◆ 디스크에 있는 데이터베이스 접근을 제어
 - ◆ 기본 OS 모듈(File 관리자, Disk 관리자)을 이용

DBMS의 구성요소(3)



The Database Administrator

● Data administrator (DA)

- ◆ Person who makes the strategic and policy decisions regarding the data of the enterprise
- ◆ 경영 전략

● Database administrator(DBA)

- ◆ Person who provides the necessary technical support for implementing those decisions
- ◆ Responsible for the overall control of the system at a technical level
- ◆ 데이터베이스 시스템의 관리운영에 대한 모든 책임을 지고 있는 사람
- ◆ DB 구축

The Database Administrator

● Functions of DBA

- ◆ Defining the conceptual schema : logical (or conceptual) database design
- ◆ Defining the internal schema : physical database design
 - Conceptual / internal mapping(by conceptual DDL)
- ◆ Liaising (접촉) with users
 - External schemas and external/conceptual mapping(by external DDL)
- ◆ Defining security and integrity checks
 - As part of the conceptual schema
- ◆ Defining backup and recovery procedures
- ◆ Monitoring performance and responding to changing requirements
 - Best for the enterprise

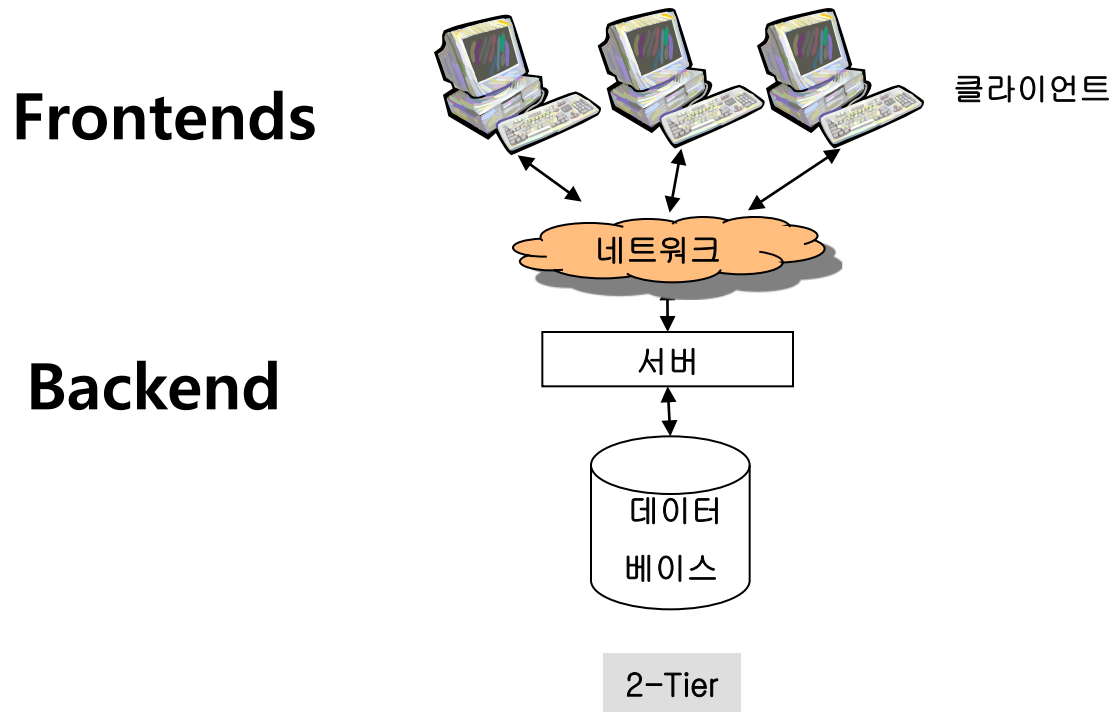
이러함

DBA의 업무

- 데이터베이스 설계와 운영
 - ◆ 데이터베이스의 구성 요소를 결정
 - ◆ 스키마 정의
 - ◆ 저장 구조와 접근 방법 설정
 - ◆ 보안 정책 수립, 권한부여, 유효성 검사
 - ◆ 예비(backup), 회복(recovery) 절차의 수립
 - ◆ 데이터베이스의 무결성 유지
 - ◆ 성능 향상과 새로운 요구에 대응한 데이터베이스의 재구성
 - ◆ 데이터 사전의 유지 관리
- 행정 및 불평 해결
 - ◆ 데이터의 표현과 시스템의 문서화에 표준 설정
 - ◆ 사용자의 요구 및 불평 해결
- 시스템 감시 및 성능 분석
 - ◆ 시스템 성능의 청취
 - ◆ 자원의 이용도, 병목 현상, 장비 성능 등
 - ◆ 사용자 요구의 변화, 데이터 사용 추세, 각종 통계의 종합 분석

Backend vs. Frontends

- Two-part structure of a database system
 - ◆ Backend: DBMS
 - ◆ Frontends: various applications that run on top of the DBMS



Applications

Two categories of applications

◆ User-written applications

- Regular application programs

◆ Vendor-provided applications (“tools”)

- To assist in the process of creating and executing other applications

(1) query language processors

(2) report writer

(3) business graphics subsystems

(4) spreadsheets

(5) natural language processors

(6) statistical package

(7) copy management tools

(8) application generators (including “4GL” processors)

(9) computer-aided software engineering (CASE) products

Utilities

- Utilities for DBA: integral (필수) components of DBMS
 - ◆ Load routines: to create the initial version of the database
 - ◆ Unload/reload routines: for recovery purpose
 - ◆ Reorganization routines: for performance purpose
 - ◆ Statistics routines: to compute various performance statistics
 - ◆ Analysis routines: to analyze the statistics

하드웨어

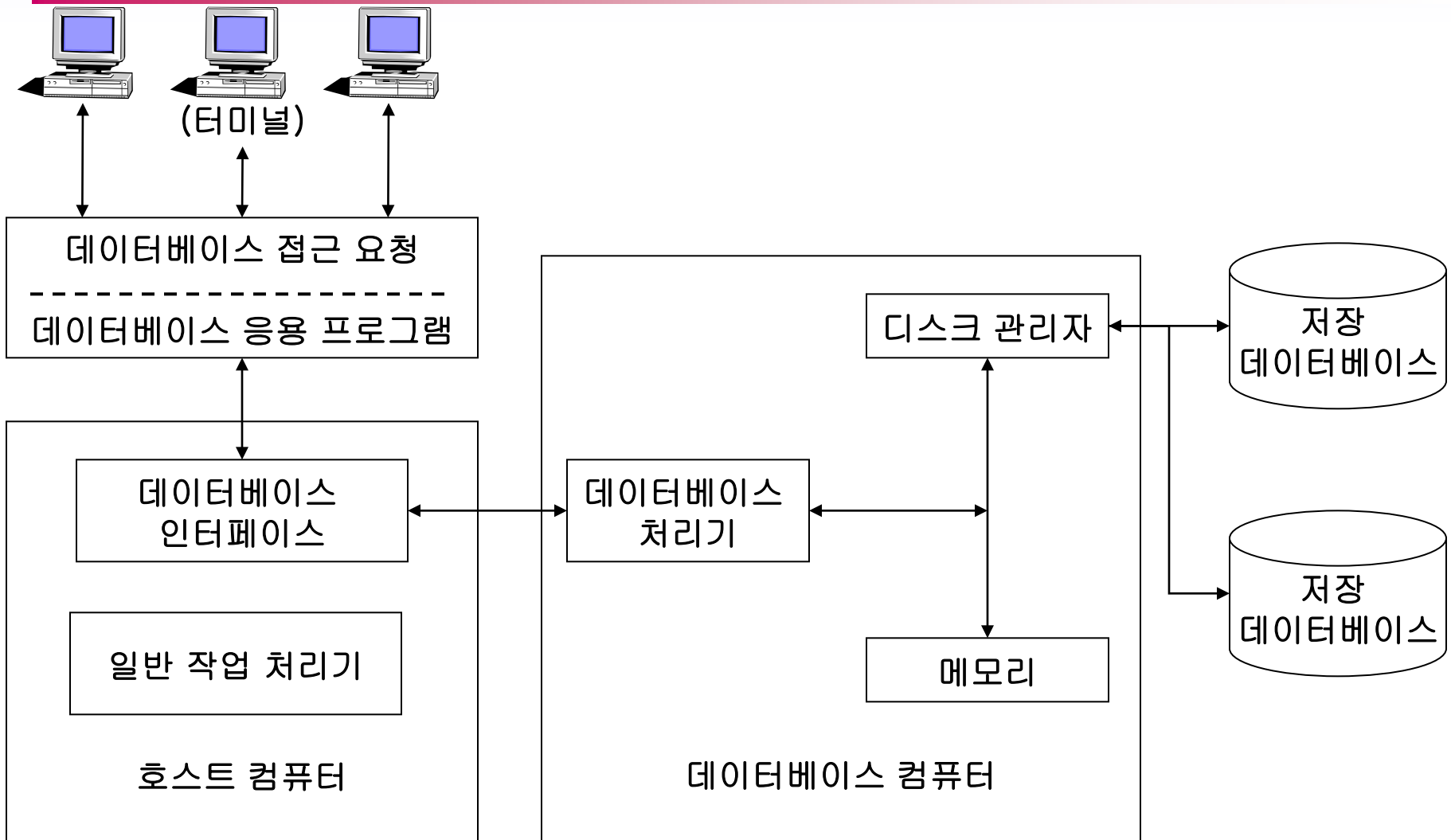
- 대규모 데이터베이스 (VLDB)

- ◆ GB (gigabyte): 1024MB
- ◆ TB (terabyte): 1024GB
- ◆ PB (petabyte): 1024TB
- ◆ EB (exabyte): 1024PB
- ◆ ZB (zettabyte): 1024EB
- ◆ YB (yottabyte): 1024ZB

- 데이터베이스 기능만을 효율적으로 수행하기 위한 특수한 목적의 하드웨어

- ◆ Backend processor
- ◆ 지능형 저장 장치
- ◆ 내용식(associative) 메모리
- ◆ 병렬 처리(parallel processing)

데이터베이스 컴퓨터 구성도



데이터베이스 시스템 구성도

