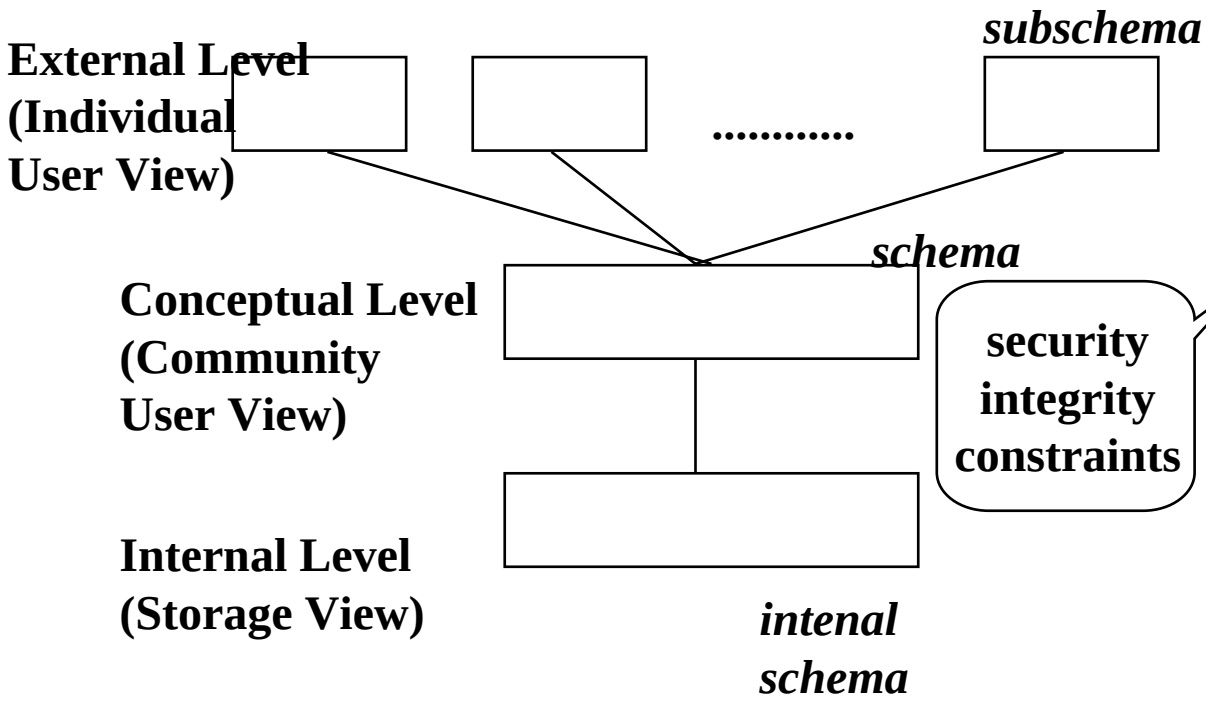
An Introduction to Database Systems

chapter 2. Database System Architecture

spark@dblab.sogang.ac.kr

2.1 Introduction

- **3-levels of data representation**



External Schema :
한 응용에 관심있는 **data object**의 **view**

Conceptual Schema :
한 조직의 **DB**를 전체적인 입장에서 보는것
(all of entities, properties, relationships)

Internal Schema :
Physical Storage Structure (flat file, hierarchical/network/inverted file)

spark@dblab.sogang.ac.kr

2.2 The Three Levels of the Architecture

- **3-levels of data representation**
  - **purpose** ***: data independence***
    - **conceptual schema**를 효율성과 응용 데이타 요구사항에 관계없이 상대적으로 안정된 것으로 유지

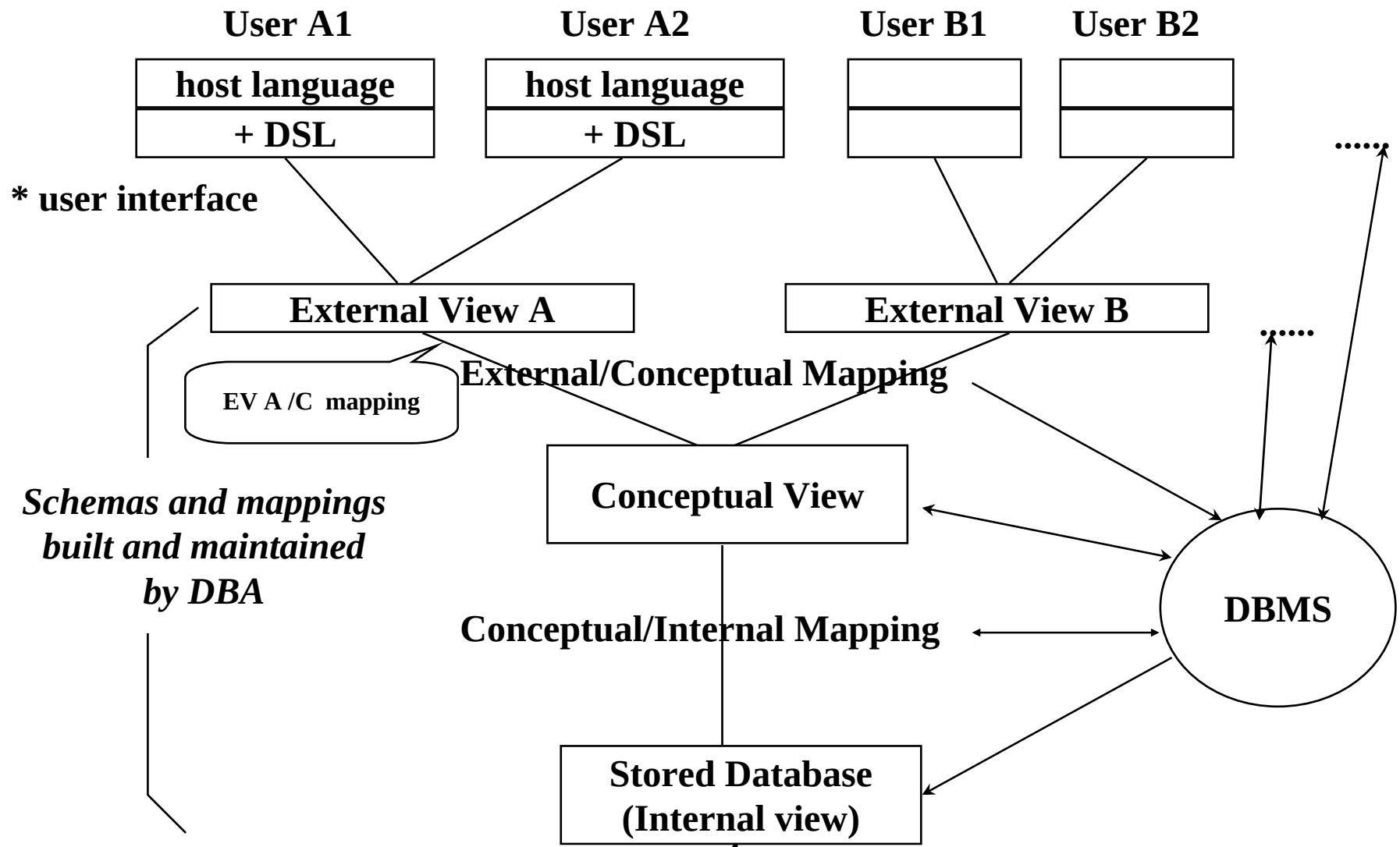
사용자가 보는 **view**와 데이타가 저장되는 방법에서의 **flexibility**와 **adaptability**

external (PL/I)	**DCL 1 EMPP,** **2 EMP# CHAR(6),** **2 SAL FIXED BIN(31) ;**	**01 EMPC** **02 EMPNO PIC X(6)** **02 DEPTNO PIC X(4)**	***external (COBOL)***
conceptual	**EMPLOYEE** **EMPLOYEE_NUMBER CHARACTER (6)** **DEPARTMENT_NUMBER CHARACTER (4)** **SALARY NUMERIC (5)**		
internal	**STORED_EMP LENGTH=20** **PREFIX TYPE=BYTE(6), OFFSET=0** **EMP# TYPE=BYTE(6), OFFSET=6, INDEX=EMPX** **DEPT# TYPE=BYTE(4), OFFSET=12** **PAY TYPE=FULLWORD, OFFSET=16**		

spark@dblab.sogang.ac.kr

2.2 The Three Levels of the Architecture

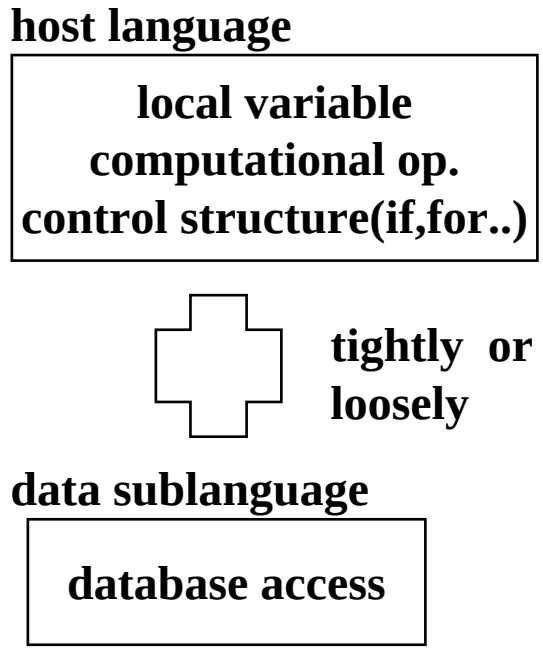
DBMS Architecture



spark@dblab.sogang.ac.kr

2.3 External Level

- **External Level - USER**
  - **application programmer**
    - **PL/I, C++, Java + DSL**
    - **전문적인 language (4GL)**
  - **end-user (on-line terminal user)**
    - **query language**
    - **forms- or menu-based**
  - **DBA**



spark@dblab.sogang.ac.kr

2.3External level

- data language
  - sublanguage
    - DDL(Data Definition/Description Language)
      - definition or description of database objects
      - schema, subschema, mapping
    - DML(Data Manipulation Language)
      - manipulation or processing of those objects
- external view
  - contents of database as perceived by a certain user
  - multiple occurrence of multiple types of external record

    (DML - retrieve an external record)
  - defined by external schema

spark@dblab.sogang.ac.kr

2.4-5 Conceptual and Internal Level

- **Conceptual level**
  - **a representation of the entire information content of the database**
  - **conceptual view**
    - **multiple occurrence of multiple types of conceptual record**
  - **security and integrity checks**
- **Internal level**
  - **internal schema**
    - **define the various types of stored records**
    - **specify**
      - **what indexes exist**
      - **how stored fields are represented**
      - **what physical sequence the stored record are in**
  - **internal view**
    - **multiple occurrence of multiple types of internal (stored) record**

spark@dblab.sogang.ac.kr

2.6 Mapping

- **Conceptual/Internal mapping**
  - **defines the correspondence between the conceptual view and the stored database**
  - **how conceptual records and fields are represented at the internal level**
  - ***If the structure of the stored database is changed, then the conceptual/internal mapping must be changed accordingly***
- **External/Conceptual mapping**
  - **defines the correspondence between a particular external view and the conceptual view**
- **External/External mapping**
  - **definition of one external view to be expressed *in terms of others***

spark@dblab.sogang.ac.kr

2.7 DBA(Database Administrator) : responsibilities

- deciding the information content of the database
  - what information → entities, relationships
  - conceptual schema(using conceptual DDL)
    - object form : used by DBMS
    - source form : reference document for the users
- deciding the storage structure and access strategy
  - how the data is to be represented in database → storage structure definition
  - associated mapping between the internal/conceptual schema
- Liaising with Users
  - external schema
  - associated mapping between external schema and the conceptual schema
- defining security and integrity checks
- defining a strategy for backup and recovery
- monitoring performance and responding to changing requirement

spark@dblab.sogang.ac.kr

2.7 DBA(Database Administrator) : utilities

- load routines
  - to create the initial version of database
- dump/restore routines
  - recovery
- reorganization routines
  - to rearrange the data in the database for various performance reasons
    (cluster, data reclaim space)
- statistics routine
  - to compute various performance statistics(file size, data value distribution)
- analysis routines

spark@dblab.sogang.ac.kr

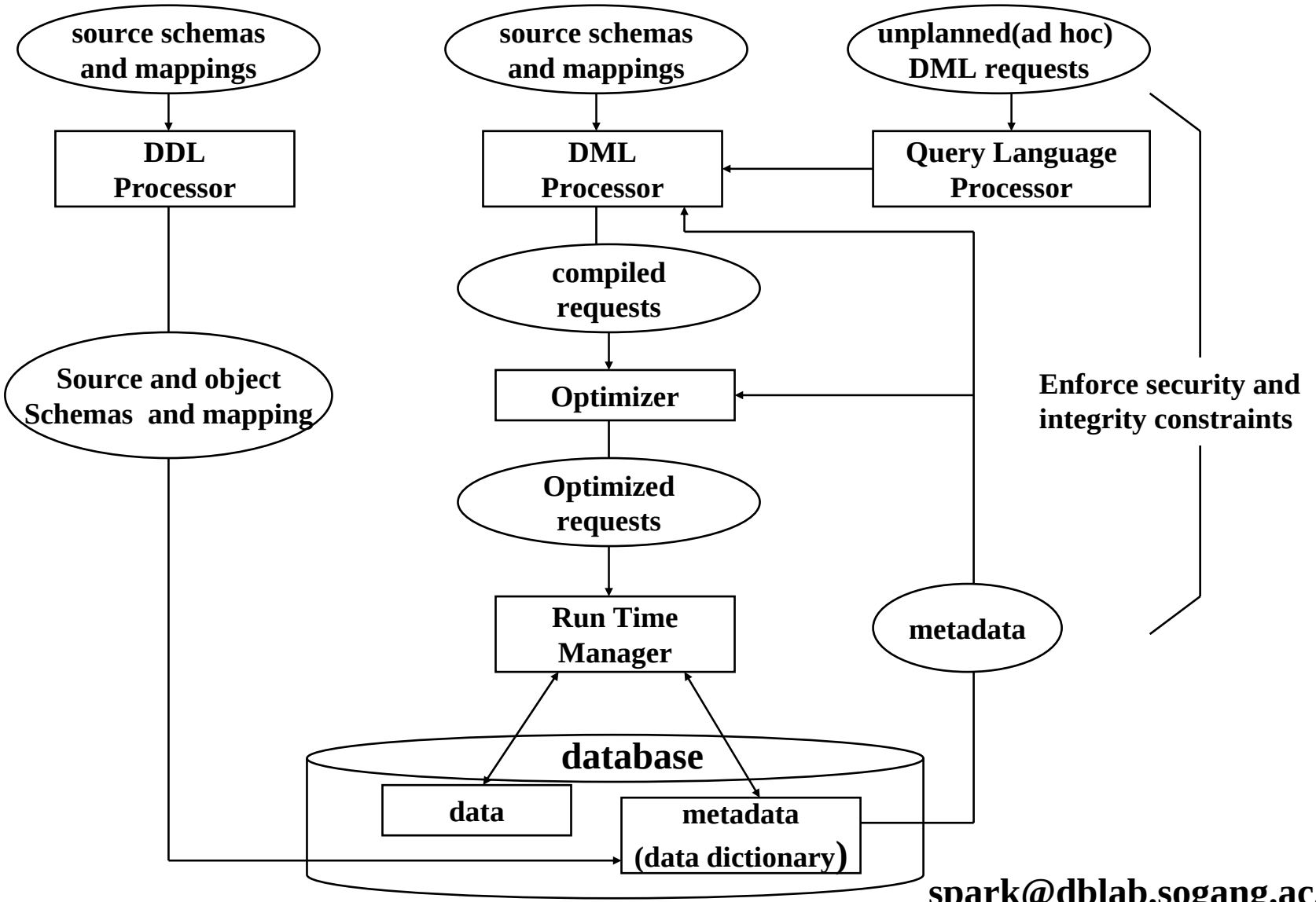
2.8 DBMS

- **Database Management System(DBMS)**
  - **software that handles all access to the database**
    - **A user issues an access request using SQL**
    - **DBMS intercepts that request and analyzes it**
    - **DBMS inspects the external schema for that user**
      - **external/conceptual mapping → the conceptual schema → conceptual/internal mapping → storage structure definition**
    - **DBMS executes the necessary operations on the stored database**
  - ***the entire process is interpretive (poor performance)***
    - ***the process is done at execution time***
  - ***In practice, access requests is compiled in advance of execution time (ex, DB2)***

spark@dblab.sogang.ac.kr

2.8 DBMS

✓ **Major DBMS functions and components**



spark@dblab.sogang.ac.kr

2.8 DBMS

- **the functions of DBMS**
  - **data definition**
    - **to accept data definitions(external schemas, the conceptual schema, the internal schema, and all associated mappings) in source form and to convert them to the appropriate object form**
    - **language processor for DDLs**
  - **data manipulation**
    - **to handle requests from the user to retrieve, update, or delete existing data in the database or to add new data to the database**
    - **DML language processor**
    - **DML requests**
      - **a planned request (operational or production application)**
      - **an unplanned request(decision support application)**

        ad hoc query
  - **Optimization and execution**
    - **Determine an efficient way of implementing the request**
    - **Executed under the control of the runtime manager**

spark@dblab.sogang.ac.kr

2.8 DBMS

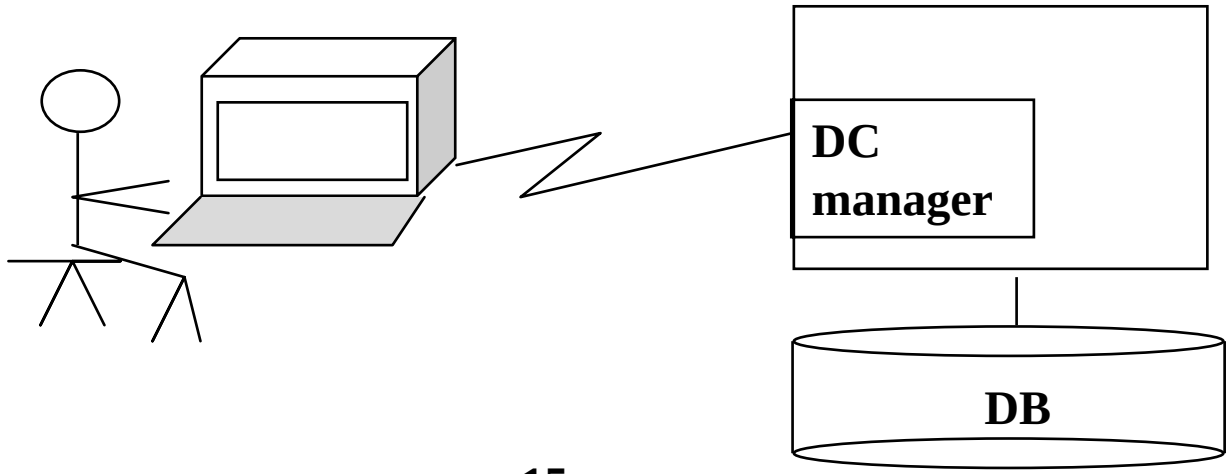
- **functions of DBMS(cont'd)**
  - **data security and integrity**
    - **monitor user requests and reject any attempts to violate the security and integrity rules defined by DBA**
  - **data recovery and concurrency**
    - **transaction manager or TP monitor to enforce recovery and concurrency control**
  - **data dictionary (system catalog)**
    - **system database to contain metadata(data about the data) definitions of other objects in the systems(not raw data)**
    - **all schemas and mappings**
    - **cross-reference information**
    - **integrated into the database(possible to query itself)**
  - **performance**

★ ***provide user interface to DBMS***

spark@dblab.sogang.ac.kr

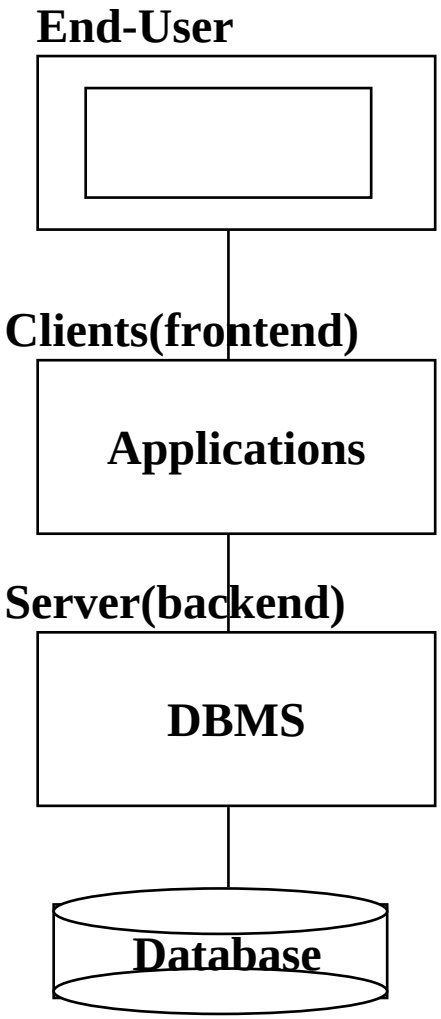
2.9 Data Communications Manager

- database requests from an end user
  (transmitted in the form of communication messages)
- data communications manager(DC manager)
- DB/DC system



spark@dblab.sogang.ac.kr

2.10 Client/Server Architecture



. The server is the DBMS itself
(data definition/manipulation, security, integrity, ...)

. The Clients are the various applications that
run on top of the DBMS
- user-written applications
- built-in applications

spark@dblab.sogang.ac.kr

2.10 Client/Server Architecture

- **Applications**
  - **user-written applications**
    - **regular application programs written in C or COBOL + DSL**
  - **vendor-provided applications**
    - **tools to assist in the process of creating and executing other applications**
    - **query language processors for ad hoc queries**
    - **report writers**
    - **business graphics subsystems**
    - **spreadsheets**
    - **natural language processors**
    - **statistical packages**
    - **copy management tools**
    - **application generators(including “4GL” processors)**
    - **other application development tools, including CASE products**
  - **client - server**
    - **different machines ; distributed processing**

spark@dblab.sogang.ac.kr

2.11 Utilities

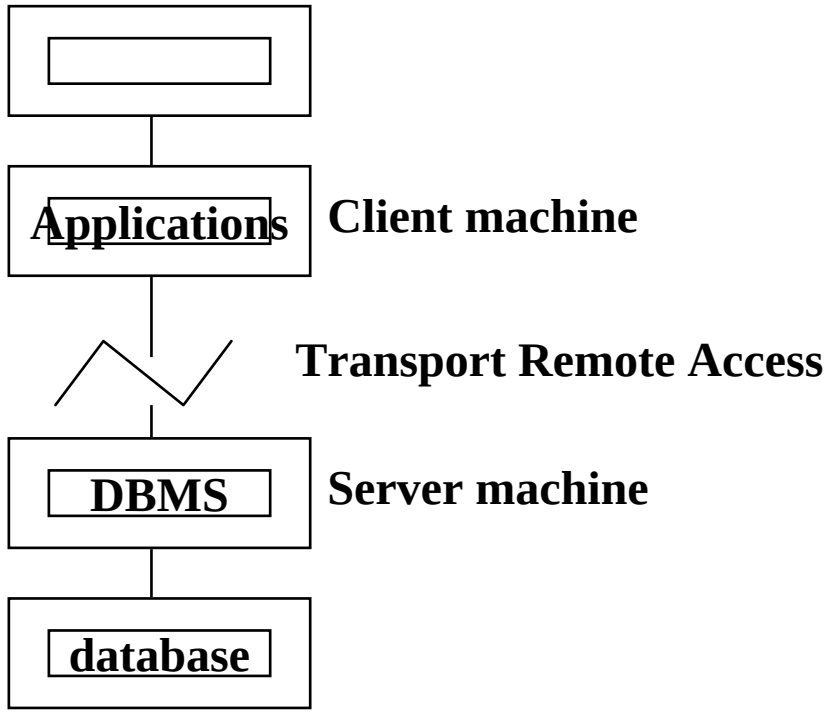
to help the DBA with various administration tasks

- **load routines**
  - **to create the initial version of the database from one or more non-database files**
- **unload/reload routines**
  - **to backup storage for recovery purposes**
- **reorganization routines**
  - **to rearrange the data in the database for various reasons**
- **statistical routines**
  - **to compute various performance statistics such as file sizes or data value distributions or I/O counts, etc.**
- **analysis routines**
  - **to analyze the statistics just mentioned**

spark@dblab.sogang.ac.kr

2.12 Distributed Processing

- *It means that distinct machines can be connected together into a communications network such that a single data processing task can span several machines in the network. (cf. Parallel Processing)*



spark@dblab.sogang.ac.kr

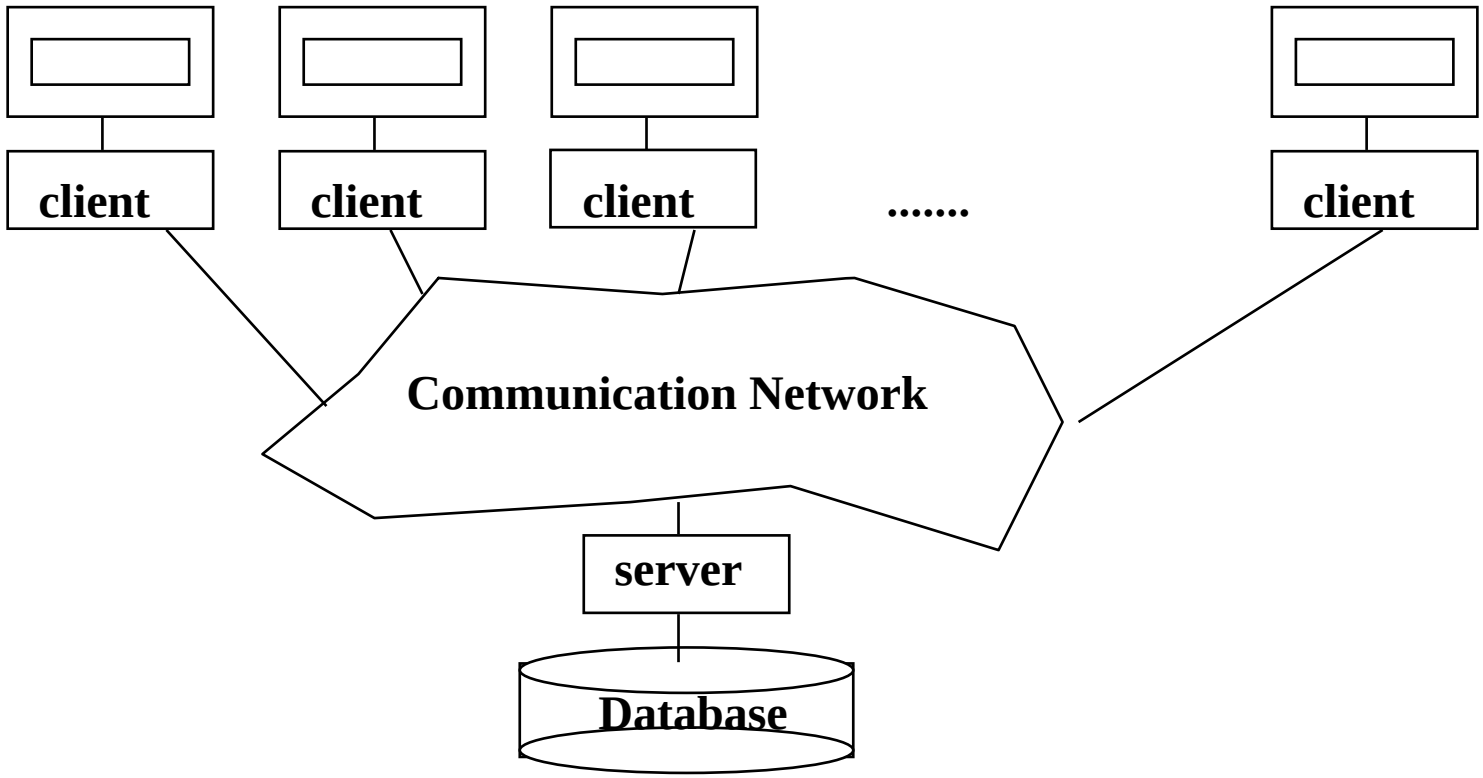
2.12 Distributed Processing

- arguments we prefer to Client/Server architecture
  - clients/server : parallel processing
    - response time, throughput
  - server machine : “a database machine”
    - DBMS performance
  - client machine
    - tailored to the needs of the end-users
    - better interfaces, high availability, fast responses, improved ease of use
  - several different client machines - one server
    - It is common for a single enterprise to operate many computers

spark@dblab.sogang.ac.kr

2.12 Distributed Processing

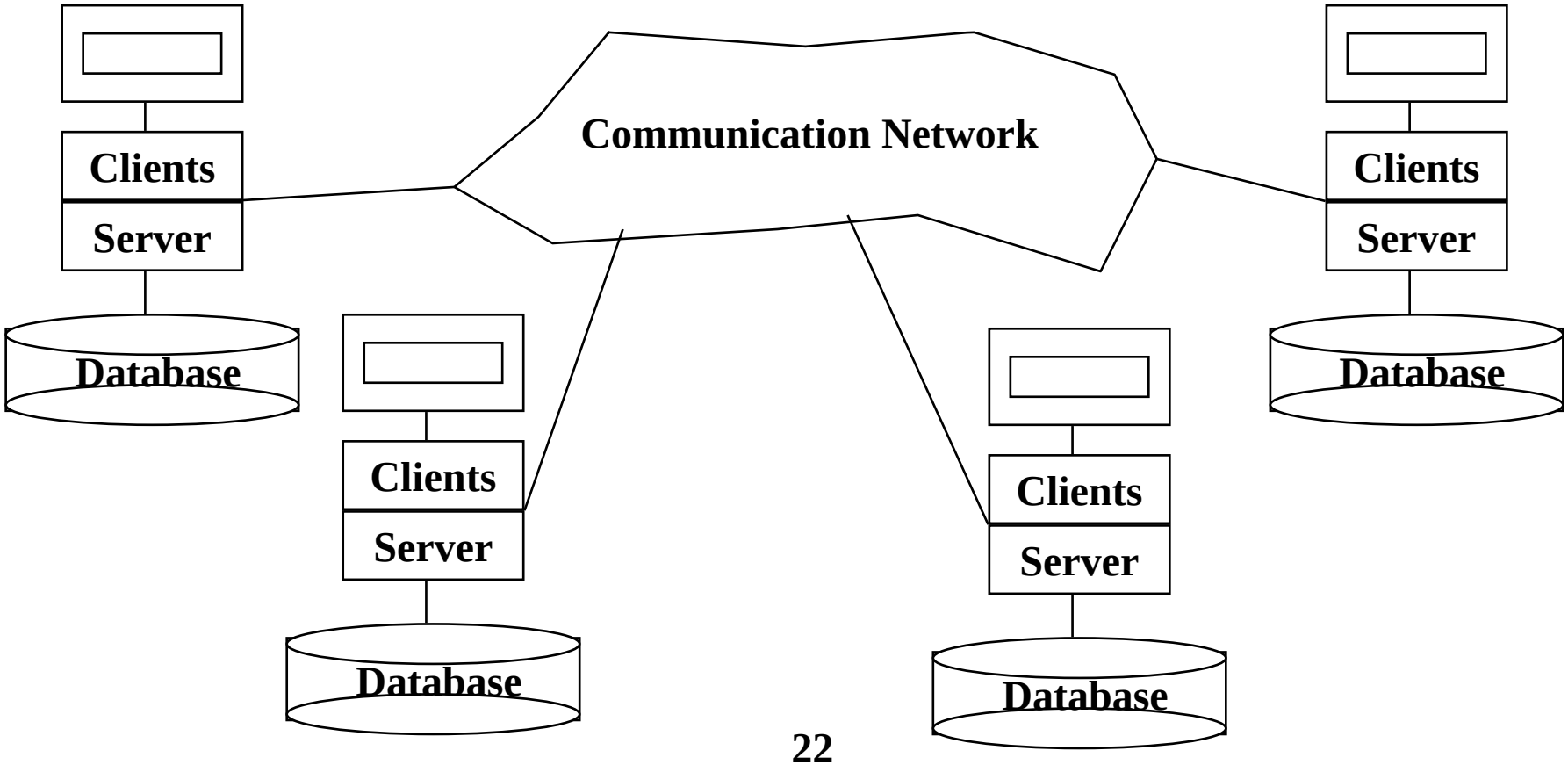
- several different client machines - one server



spark@dblab.sogang.ac.kr

2.12 Distributed Processing

- Each machine is both client and server



spark@dblab.sogang.ac.kr

2.12 Distributed Processing

- **Each machine is both client and server - The Bank Example**
  - **A single client machine might be able to access several different server machines**
    - ***A given client might be able to access any number of servers, but only one at a time* (i.e., each individual database request must be directed to just one server)**
      - **impossible to combine data from two or more servers**
      - **the user has to know which machines hold which pieces of data**
    - ***The client might be able to access many servers simultaneously*(i.e., a single database request might be able to combine data from several servers)**
      - **the servers look to the client as if they were really a single server**
      - **Distributed Database System**

        "Transparency"

spark@dblab.sogang.ac.kr