DBMS Functions & Environment
Data Abstraction

DBMS Functions

- Performs functions that guarantee integrity and consistency of data
  - Data dictionary management
    - defines data elements and their relationships
  - Data storage management
    - stores data and related data entry forms, report definitions, etc.
  - Data transformation and presentation
    - translates logical requests into commands to physically locate and retrieve the requested data
DBMS Functions (continued)

- Security management
  - enforces user security and data privacy within database

- Multi-user access control
  - creates structures that allow multiple users to access the data

- Backup and recovery management
  - provides backup and data recovery procedures

DBMS Functions (continued)

- Data integrity management
  - promotes and enforces integrity rules to eliminate data integrity problems

- Database access languages and application programming interfaces
  - provides data access through a query language

- Database communication interfaces
  - allows database to accept end-user requests within a computer network environment
Disadvantages of DBMS

- Complexity
- Size
- Cost of DBMS
- Additional hardware cost
- Cost of conversion
- Performance
- Higher impact of failure

The Database System Environment

- Database system environment is composed of 5 main parts:
  1. Hardware
  2. Software
  3. Data
  4. People
  5. Policies & Procedures
Roles in Database Environment

- Database Designer
- Application Developer
- Database Administrator (DBA)
- End-user

The Database System Environment (continued)
Database Schema and State

**Database Model/Schema (intension)**
Allowable logical structures of database is known as data model/schema. This gives description of a database for a particular universe of discourse.

**Database Instance (extension)**
The data in the database at a particular moment in time.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>varchar(100)</td>
<td>☐</td>
</tr>
<tr>
<td>Password</td>
<td>varchar(100)</td>
<td>☐</td>
</tr>
<tr>
<td>UserType</td>
<td>char(10)</td>
<td>☑</td>
</tr>
<tr>
<td>RoleID</td>
<td>int</td>
<td>☑</td>
</tr>
<tr>
<td>HospitalID</td>
<td>int</td>
<td>☑</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Username</th>
<th>Password</th>
<th>UserType</th>
<th>RoleID</th>
<th>HospitalID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin</td>
<td>******</td>
<td>Admin</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Manager</td>
<td>******</td>
<td>Manager</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Operator</td>
<td>******</td>
<td>Operator</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
Database Languages

- **Data Definition Language (DDL)**
  Provides set of operations to create or modify the database schema.
  *e.g. Create table, Alter table, Drop table*

- **Data Manipulation Language (DML)**
  Provides a set of operations that support the basic data manipulation operations the data.
  *e.g. Select, Insert, Update, Delete*

- **Data Control Language (DCL)**
  Use to control/configure database configurations and access control.
  *e.g. Grant, Revoke*

Multi-user DBMS Environment

- Two main factors to manage in a multi-user DBMS are:
  - Data Storage
  - Data Processing

- Following are different type of architectures that have been used to manage multi-user DBMS requirement:
  - Teleprocessing
  - File-Server
  - Client-Server
Data Abstraction

Abstraction

- Abstraction is the process of recognizing and focusing on important characteristics of a situation or object and leaving/filtering out the unwanted characteristics of that situation or object.

- Abstraction: A concept or idea not associated with any specific instance.
Degrees of Data Abstraction

- American National Standards Institute/Standards Planning and Requirements Committee (ANSI/SPARC)
  - Classified data models according to their degree of abstraction (1970s):
    - Conceptual
    - External
    - Internal

ANSI/SPARC Three level architecture

External Level

Conceptual Level

Internal Level

User’s view of the database

Community view

Physical representation

Physical data organization
Abstraction levels

- **External Level** - (End-user’s view of data)
  Describes that part of the database that is relevant to a particular user

- **Conceptual Level** - (Community View of data)
  Describes what data is stored in the database and relationships among the data

- **Internal Level** – (DBMS view of data)
  Describes how the data is stored in the database

Data Independence

- **Logical data Independence**
  Immunity of external schema to changes in conceptual schema

- **Physical data independence**
  Immunity of Conceptual schema to changes in Internal schema