Chapter 1: Introduction

- Purpose of Database Systems
- View of Data
- Data Models
- Data Definition Language
- Data Manipulation Language
- Transaction Management
- Storage Management
- Database Administrator
- Database Users
- Overall System Structure

Database Management System (DBMS)

- Collection of interrelated data
- Set of programs to access the data
- DBMS contains information about a particular enterprise
- DBMS provides an environment that it both convenient and efficient to use
Purpose of Database Systems

Database management systems were developed to handle the following difficulties of typical file-processing systems supported by conventional operating systems.

- Data redundancy and inconsistency
- Difficulty in accessing data
- Data isolation – multiple files and formats
- Integrity problems
- Atomicity of updates
- Concurrent access by multiple users
- Security problems

View of Data

An architecture for a database system

```
view level

view 1   view 2   ...   view n

logical level

physical level
```
Levels of Abstraction

- Physical level: describes how a record (e.g., customer) is stored.
- Logical level: describes data stored in database, and the relationships among the data.

```type customer = record
    name : string;
    street : string;
    city : integer;
end;```

- View level: application programs hide details of data types. Views can also hide information (e.g. salary) for security purposes.

Instances and Schemas

- Similar to types and variables in programming languages
- Schema – the logical structure of the database (e.g., set of customers and accounts and the relationship between them)
- Instance – the actual content of the database at a particular point in time
Data Independence

- Ability to modify a schema definition in one level without affecting a schema definition in the next higher level.
- The interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others.
- Two levels of data independence:
  - Physical data independence
  - Logical data independence

Data Models

- A collection of tools for describing:
  - data
  - data relationships
  - data semantics
  - data constraints
- Object-based logical models
  - entity-relationship model
  - object-oriented model
  - semantic model
  - functional model
- Record-based logical models
  - relational model (e.g., SQL/DS, DB2)
  - network model
  - hierarchical model (e.g., IMS)
**Entity-Relationship Model**

Example of entity-relationship model

- social-security
- customer-street
- customer-name
- customer-city
- account-number
- balance

**Relational Model**

Example of tabular data in the relational model:

<table>
<thead>
<tr>
<th>customer-name</th>
<th>social-security</th>
<th>customer-street</th>
<th>customer-city</th>
<th>account-number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Johnson</td>
<td>192-83-7465</td>
<td>Alma</td>
<td>Palo Alto</td>
<td>A-101</td>
</tr>
<tr>
<td>Smith</td>
<td>019-28-3746</td>
<td>North</td>
<td>Rye</td>
<td>A-215</td>
</tr>
<tr>
<td>Johnson</td>
<td>192-83-7465</td>
<td>Alma</td>
<td>Palo Alto</td>
<td>A-201</td>
</tr>
<tr>
<td>Jones</td>
<td>321-12-3123</td>
<td>Main</td>
<td>Harrison</td>
<td>A-217</td>
</tr>
<tr>
<td>Smith</td>
<td>019-28-3746</td>
<td>North</td>
<td>Rye</td>
<td>A-201</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>account-number</th>
<th>balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-101</td>
<td>500</td>
</tr>
<tr>
<td>A-201</td>
<td>900</td>
</tr>
<tr>
<td>A-215</td>
<td>700</td>
</tr>
<tr>
<td>A-217</td>
<td>750</td>
</tr>
</tbody>
</table>
Data Definition Language (DDL)

- Specification notation for defining the database schema
- DDL compiler generates a set of tables stored in a *data dictionary*
- Data dictionary contains metadata (i.e., data about data)
- *Data storage and definition* language – special type of DDL in which the storage structure and access methods used by the database system are specified

Data Manipulation Language (DML)

- Language for accessing and manipulating the data organized by the appropriate data model
- Two classes of languages
  - Procedural – user specifies what data is required and how to get those data
  - Nonprocedural – user specifies what data is required without specifying how to get those data
Transaction Management

- A *transaction* is a collection of operations that performs a single logical function in a database application.
- Transaction-management component ensures that the database remains in a consistent (correct) state despite system failures (e.g. power failures and operating system crashes) and transaction failures.
- Concurrency-control manager controls the interaction among the concurrent transactions, to ensure the consistency of the database.

Storage Management

- A storage manager is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.
- The storage manager is responsible for the following tasks:
  - interaction with the file manager
  - efficient storing, retrieving, and updating of data
Database Administrator

- Coordinates all the activities of the database system; the database administrator has a good understanding of the enterprise's information resources and needs.
- Database administrator's duties include:
  - Schema definition
  - Storage structure and access method definition
  - Schema and physical organization modification
  - Granting user authority to access the database
  - Specifying integrity constraints
  - Acting as liaison with users
  - Monitoring performance and responding to changes in requirements

Database Users

- Users are differentiated by the way they expect to interact with the system
- Application programmers – interact with system through DML calls
- Sophisticated users – form requests in a database query language
- Specialized users – write specialized database applications that do not fit into the traditional data processing framework
- Naive users – invoke one of the permanent application programs that have been written previously
Overall System Structure

- Application Interfaces
- Application Programs
- Database Management System
- Storage Manager
- Transaction Manager
- File Manager
- Query Processor
- Query Evaluation Engine
- DML Precompiler
- DML Compiler
- DDL Interpreter
- Query Processor

- Naive Users
- Application Sophisticated Users
- Administrator

- Database System
- Application Programs
- User Interfaces
- Database Management System
- Application Programs
- Object Code
- Embedded DML
- DML Precompiler
- DML Compiler
- DDL Interpreter
- Query Processor

- Database Scheme
- Query
- Database Administration
- Application Programs
- Application Interfaces

- Storage Manager
- Disk Storage
- Indices
- Statistical Data
- Data Files
- Data Dictionary