January 29, 1999

Georgia State University
Department of Mathematics and Computer Science

Dr. Ray Sundaram

DATABASE SYSTEMS
Figure 1.2
Figure 1.1

- Application (update)
- Constructing the database (populating)
- Manipulating the database (query)
- Creating and maintaining a database

Database Management System (DBMS): collection of programs that enables users to:

- (d) manipulate large in size
- (c) to be designed, built and populated with data for
- (b) to create a coherent collection of data (inter-related)
- Changes to the real world are reflected in the database
- (a) it represents some aspect of the real world:

Database: a collection of data with the following properties:

Chapter 1: Databases and Database Users
Support for multiple views of data (Figure 1.4)
- Support for multiple-users and concurrent access (data sharing)
- Backup and Recovery from system crashes
- Enforcing Integrity Constraints
  - Secure access to database; restricting unauthorized access
- Controlling Redundancies and Inconsistencies
- Data Sharing
  - actually stored (Figure 1.3 add Birthdate)
- Data Independence: application programs and queries are independent of how data is
- Self contained nature of database systems (database contains both data and meta-data)
Workers behind the scene:

- System administrators
- Tool developers
- DBMS designers and implementors
- System analysts and application programmers
- Stand-alone users (personal dp, single user, ex. tax package)
- Sophisticated end users (engineers, scientists, analysts)
- Naive (or parametric end users) (bank tellers/clerks)
- Casual users (managers using SQL)

End users:

- Database designers
- Database administrator (DBA)
Backup and recovery

- Ensure integrity constraints
- Complex relationships among data (1-to-many, many-many, 1-to-1)
- Multiple user interfaces (SQL, C-SQL, forms, menus, web)
- Deductive interference of data
- Persistent storage of program objects and data
- Restricting unauthorized access
- Controlling redundancy (Figure 1.5)

Intended uses of a DBMS
Economies of scale (invest in one server, centralized database)

- Up-to-date infomration (e.g. airline reservations)

- Flexibility (changing database structure)

- Reduced application development time (once database is up and running)

- Display formats, report structures

Potential for enforcing standards (name/-formatting convention)

Implications of Database approach
Data Models, Schemas, Instances

- Low-Level or Physical Data Model (access paths)
  - Relational, Object-Oriented, Deductive, Hierarchical, Network
    - Record-based, Value-based/Object-based
  - Representation or Implementation Data Model

- High-Level or Conceptual Data Model (ER)

  a set of operations for specifying updates and queries of a database (abstraction tool); It may also include

  Data Model: set of concepts used to describe the structure
Instances: actual data (occurrences or instances) database state

Figure 2.1

Schema: structure (meta-data, schema diagram, schema constructs)

Schemas (intention and instances extension)
Data Independence: ability to change conceptual or external schema without having to change conceptual or external schema.

Application Programs

Logical Data Independence: ability to change internal schema without having to change external views or physical schema.

Figure 2.2

2.2 DBMS Architecture and Data Independence

Database Systems
Natirical-Language
User Interfaces (Menu-based, GUI, Forms-based,
host language (embedded)
record at a time
- Procedural DML
set at a time
- Query Language (non-procedural/declarative)
DML (for queries and manipulation)
levels are separated by system.
SDL (Storage and VDL (Views) when three
DBMS Languages: DDL (for conceptual level)

2.3 Database Languages and Interfaces
Figure 2.3

2.4 Database System Environment
Centralized, Distributed (Homogeneous/Heterogeneous)
-- Based on number of sites

Single-user, Multi-user
-- Based on number of users

Relational, Network, Hierarchical, Object-oriented, Others
-- Based on data model

Classification of DBMS

Database Systems