Database Systems

Lecture 2

Instructor: Bilal Khalid Dar

Agenda

- Database Approach - DBMS
- Main Characteristics of the Database Approach
- Components of DBMS Environment
- History of Database Systems
- Database Users
- Actors on the scene and Workers behind the scene
- Advantages of the Database Approach
- Advantages of using a DBMS
- Disadvantages of DBMSs
- Implications of the database approach
- When not to use a DBMS

Database Approach (SOLUTION)

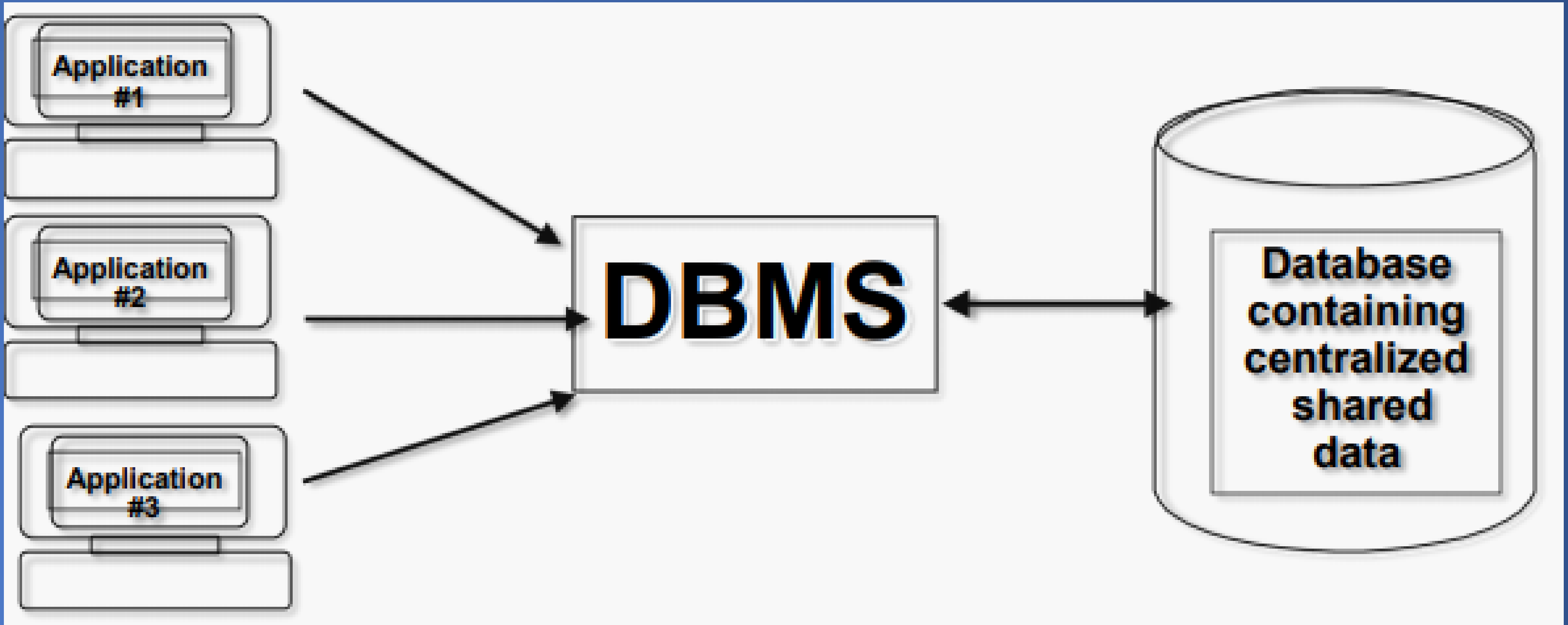
- Arose because:
- Definition of data was embedded in application programs, rather than being stored separately and independently.
- No control over access and manipulation of data beyond that imposed by application programs.
- Central repository of shared data
- Data is Stored in a standardized, convenient form
- Result:
- The database and Database Management System (DBMS).

Database Approach

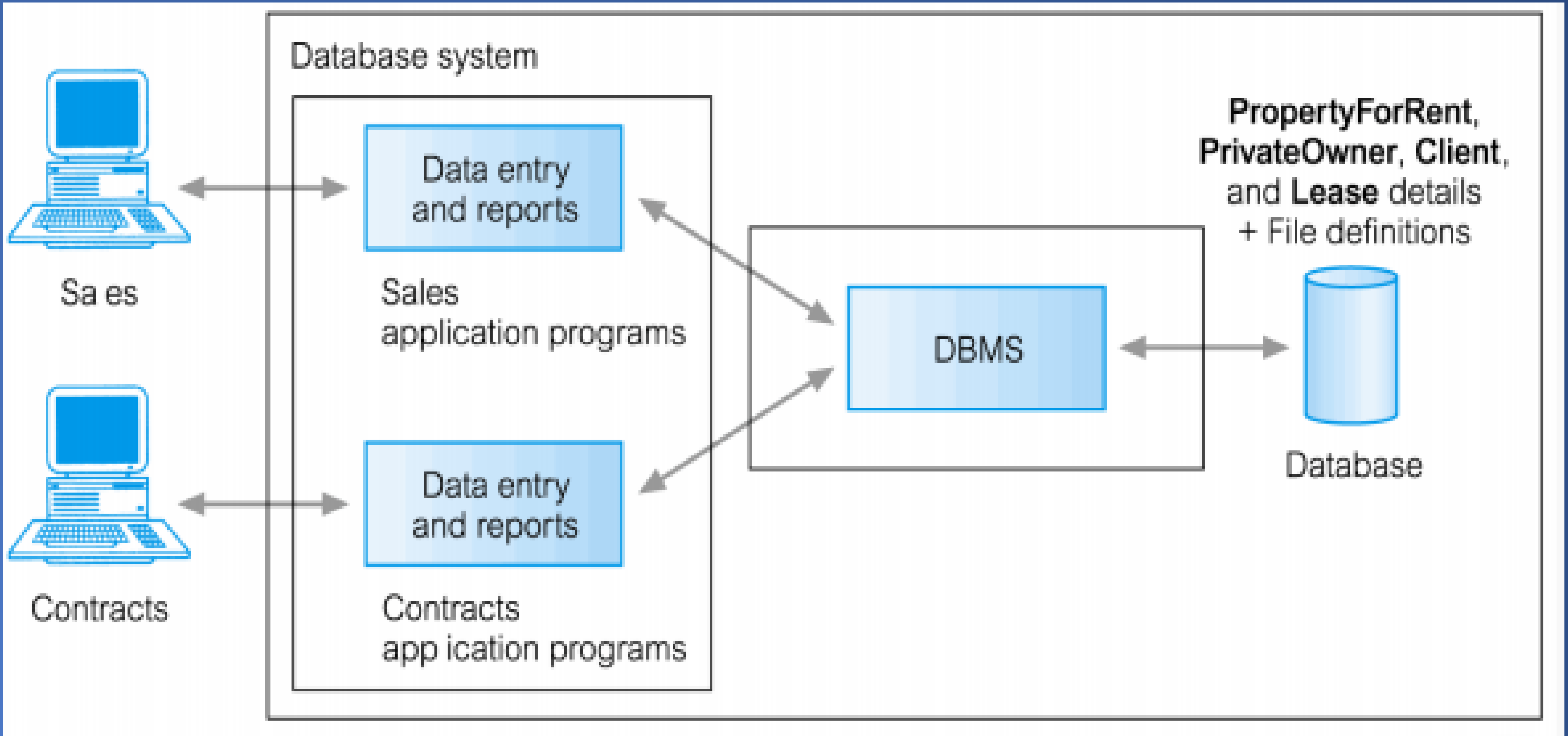
- Controlled access to database may include:
    - a security system
    - an integrity system
    - a concurrency control system
    - a recovery control system
    - a user-accessible catalogue

Database Management System

- A DBMS is a data storage and retrieval system which permits data to be stored non-redundantly while making it appear to the user as if the data is well-integrated.



Database Management System


Database system
Data entry and reports
Sales application programs
Data entry and reports
Contracts app ication programs
Sa es
Contracts
DBMS
PropertyForRent, PrivateOwner, Client, and Lease details + File definitions
Database

Main Characteristics of the Database Approach

1. Self-describing nature of a database system:

- A DBMS catalog stores the description of a particular database (e.g. data structures, types, and constraints). The description is called meta-data
- This allows the DBMS software to work with different database applications.

Example of a simplified database catalog

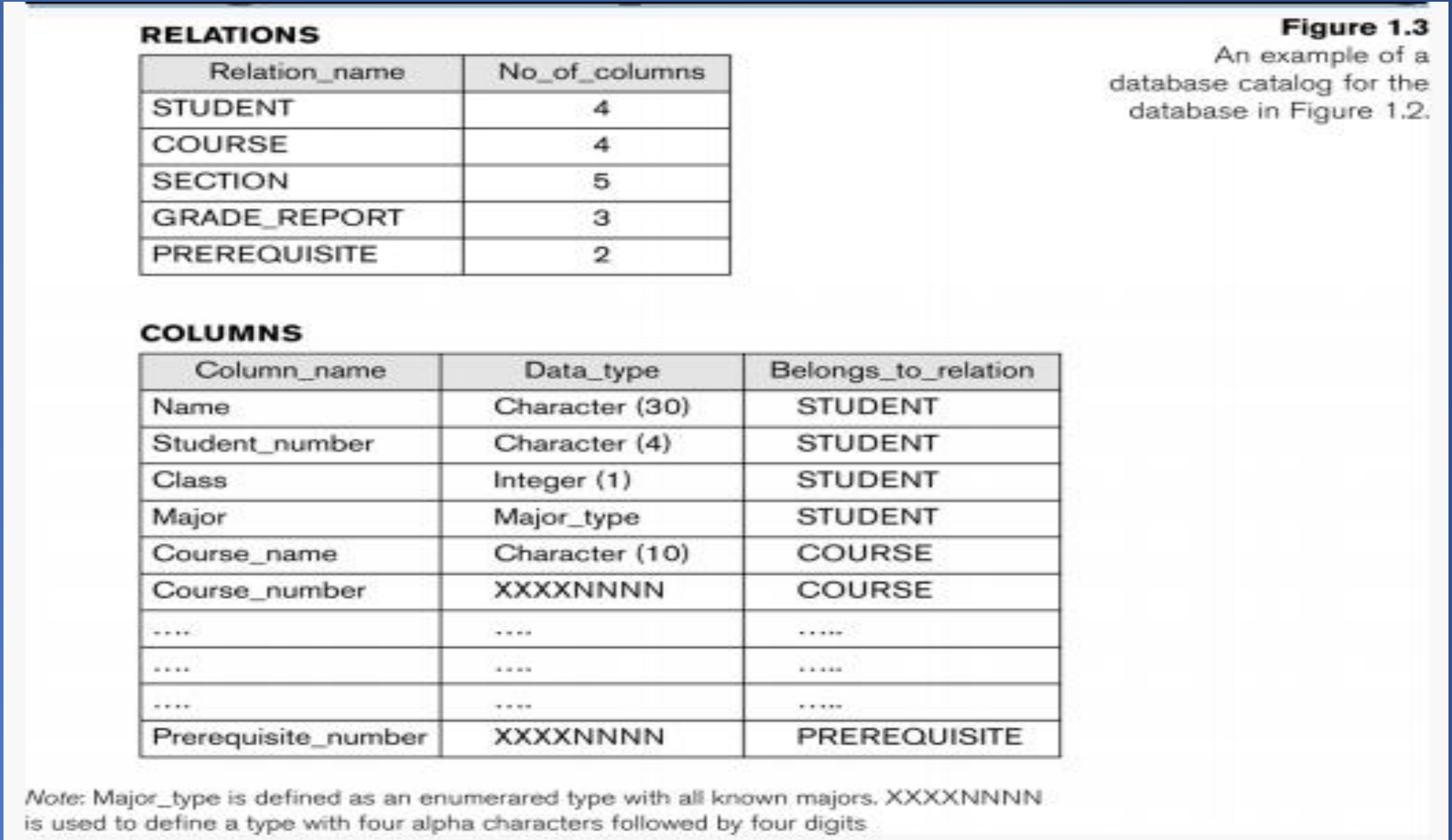


Figure 1.3
An example of a database catalog for the database in Figure 1.2.

RELATIONS

Relation_name	No_of_columns
STUDENT	4
COURSE	4
SECTION	5
GRADE_REPORT	3
PREREQUISITE	2

COLUMNS

Column_name	Data_type	Belongs_to_relation
Name	Character (30)	STUDENT
Student_number	Character (4)	STUDENT
Class	Integer (1)	STUDENT
Major	Major_type	STUDENT
Course_name	Character (10)	COURSE
Course_number	XXXXNNNN	COURSE
....	....	.....
....	....	.....
....	....	.....
Prerequisite_number	XXXXNNNN	PREREQUISITE

Note: Major_type is defined as an enumerared type with all known majors. XXXXNNNN is used to define a type with four alpha characters followed by four digits

Main Characteristics of the Database Approach

2. Insulation between programs and data:

- The structure of data files is stored in the DBMS catalog separately from the access programs and as Called program-data independence.
- Allows changing data structures and storage organization without having to change the DBMS access programs.

3. Data Abstraction:

- A data model is used to hide storage details and present the users with a conceptual view of the database.
- „Programs refer to the data model constructs rather than data storage details

Views

- Allows each user to have his or her own view of the database.
- A view is essentially some subset of the database.

Views - Benefits

- Reduce complexity
- Provide a level of security
- Provide a mechanism to customize the appearance of the database
- Present a consistent, unchanging picture of the structure of the database, even if the underlying database is changed

Main Characteristics of the Database Approach

4. Support of multiple views of the data :

Each user may see a different view of the database, which describes only the data of interest to that user.

5.Sharing of data and multi-user transaction processing:

- Allowing a set of concurrent users to retrieve from and to update the database.
- **Concurrency control** within the DBMS guarantees that each transaction is correctly executed or aborted.
- **Recovery subsystem** ensures each completed transaction has its effect permanently recorded in the database
- **OLTP (Online Transaction Processing)** is a major part of database applications This allows hundreds of concurrent database applications. This allows hundreds of concurrent transactions to execute per second.

History of Database Systems

- First-generation
  - Hierarchical and Network
- Second generation
  - Relational
- Third generation
  - Object-Oriented
  - Object-Relational

Historical Development of Database Technology

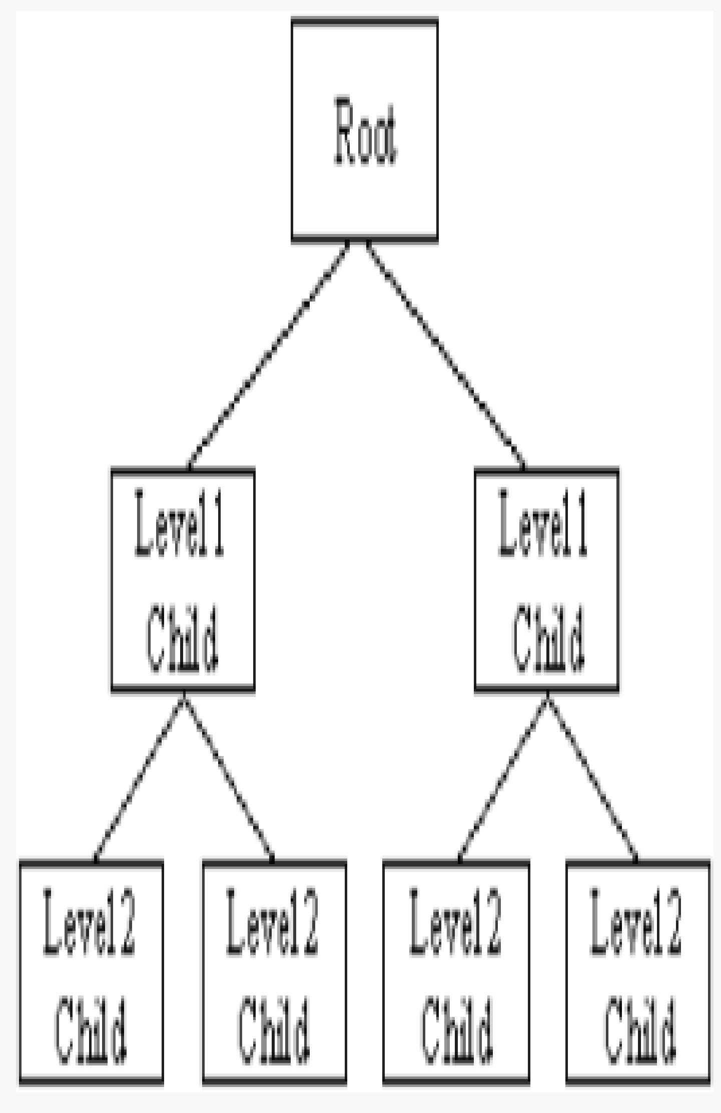
- **Early Database Applications:**
- The Hierarchical and Network Models were introduced in mid 1960s and dominated during the seventies.

- **Relational Model based Systems:**
- Relational model was originally introduced in 1970, was heavily researched and experimented within IBM Research and several universities.
- Rational DBMS products emerged in the early 1980’s

Historical Development of Database Technology(continued)

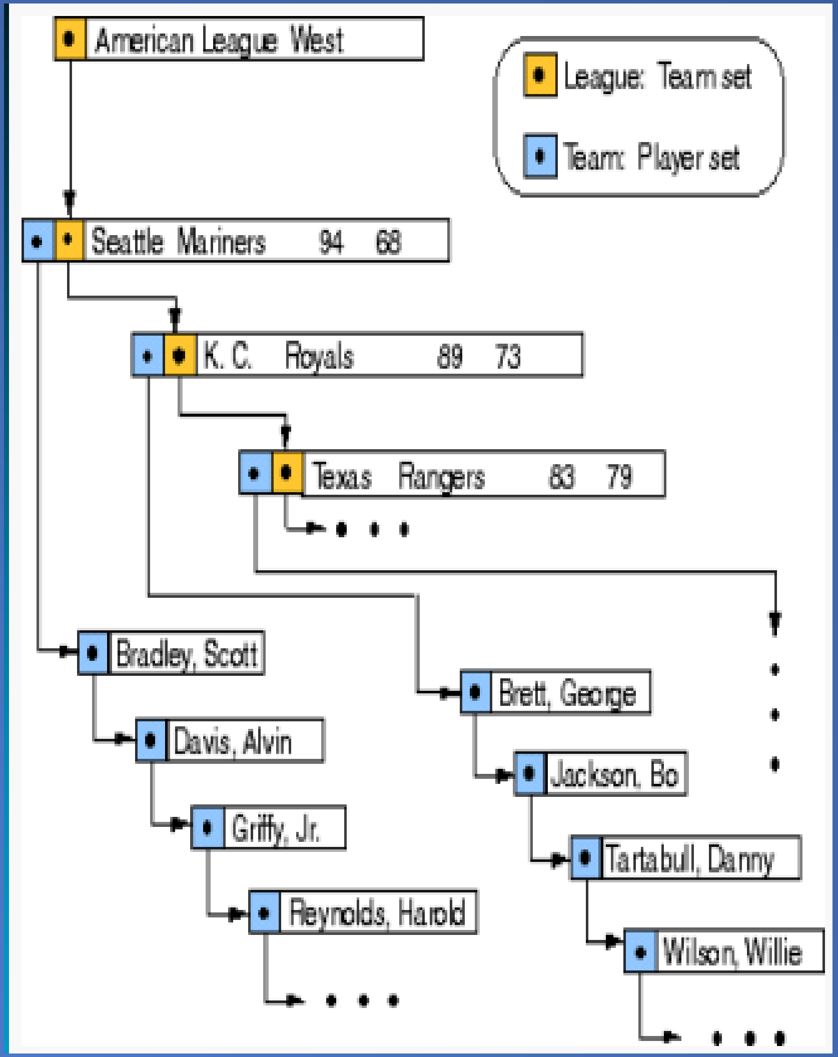
- **Object-oriented :**
- Object-Oriented Database Management Systems (OODBMSs) were introduced in late 1980s and early 1990s to cater to the need of complex data processing in CAD and other applications.
  - Their use has not taken off much.
- **Object-Relational :**
- Many relational DBMSs have incorporated object database concepts, leading to a new category called object-relational DBMSs (ORDBMSs)

DBs Hierarchical 1960s



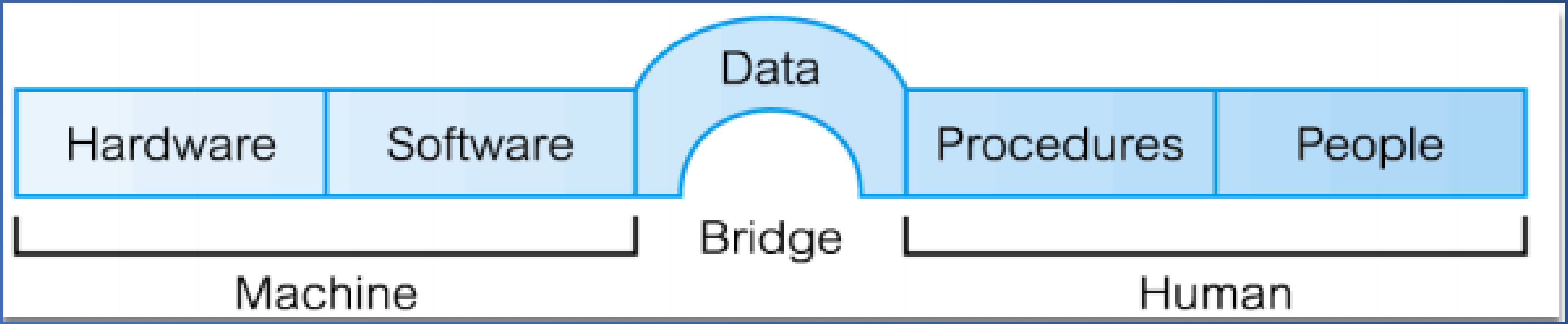
- The hierarchical data model organizes data in a tree structure.
- There is a hierarchy of parent and child data segments.
- This structure implies that a record can have repeating information, generally in the child data segments.
- Data in a series of records, which have a set of field values attached to it. It collects all the instances of a specific record together as a record type.
- These record types are the equivalent of tables in the relational model, and with the individual records being the equivalent of rows.
- To create links between these record types, the hierarchical model uses Parent Child Relationships. These are a 1:N mapping between record types. This is done by using trees,

DBs Network 1970s



- data were more naturally modeled with more than one parent per child.
- So, the network model permitted the modeling of many-to-many relationships in data
- The basic data modeling construct in the network model is the set construct.
- A set consists of an owner record type, a set name, and a member record type. A member record type can have that role in more than one set

Components of DBMS Environment



- Hardware Can range from a PC to a network of computers.
- Software DBMS, operating system, network software (if necessary) and also the application programs.
- Data Used by the organization and a description of this data called the schema

Components of DBMS Environment

- Procedures
- Instructions and rules that should be applied to the design and use of the database and DBMS.
- People
- Data Administrators – personnel responsible for maintaining the database
- System Developers – personnel responsible for designing databases and software
- End Users – people who use the applications and databases

Database Users

- Actors on the scene
- Database administrators:
- Responsible for authorizing access to the database, for coordinating and monitoring its use, acquiring software and hardware resources, controlling its use and monitoring efficiency of operations.
- Database Designers:
- Responsible to define the content, the structure, the constraints, and functions or transactions against the database. They must communicate with the end-user and understand their needs.

Database Users

- Actors on the scene (continued)

End-users: They use the data for queries, reports and some of them update the database content.

- **Categories of End-users**
- End-users can be categorized into:
- Casual: access database occasionally when needed
- Naïve or Parametric: they make up a large section of the end-user population.

Database Users

- Categories of End-users
- They use previously well-defined functions in the form of “canned transactions” against the database.
- Canned transactions are standard types of queries and updates which frequently used by Naive or parametric end users to constantly querying and updating
- Examples are bank-tellers or reservation clerks who do this activity for an entire shift of operations this activity for an entire shift of operations.

Database Users

- Categories of End-users

- Sophisticated: These include business analyst’s scientist’s engineer’s „ and engineers, others thoroughly familiar with the system capabilities.

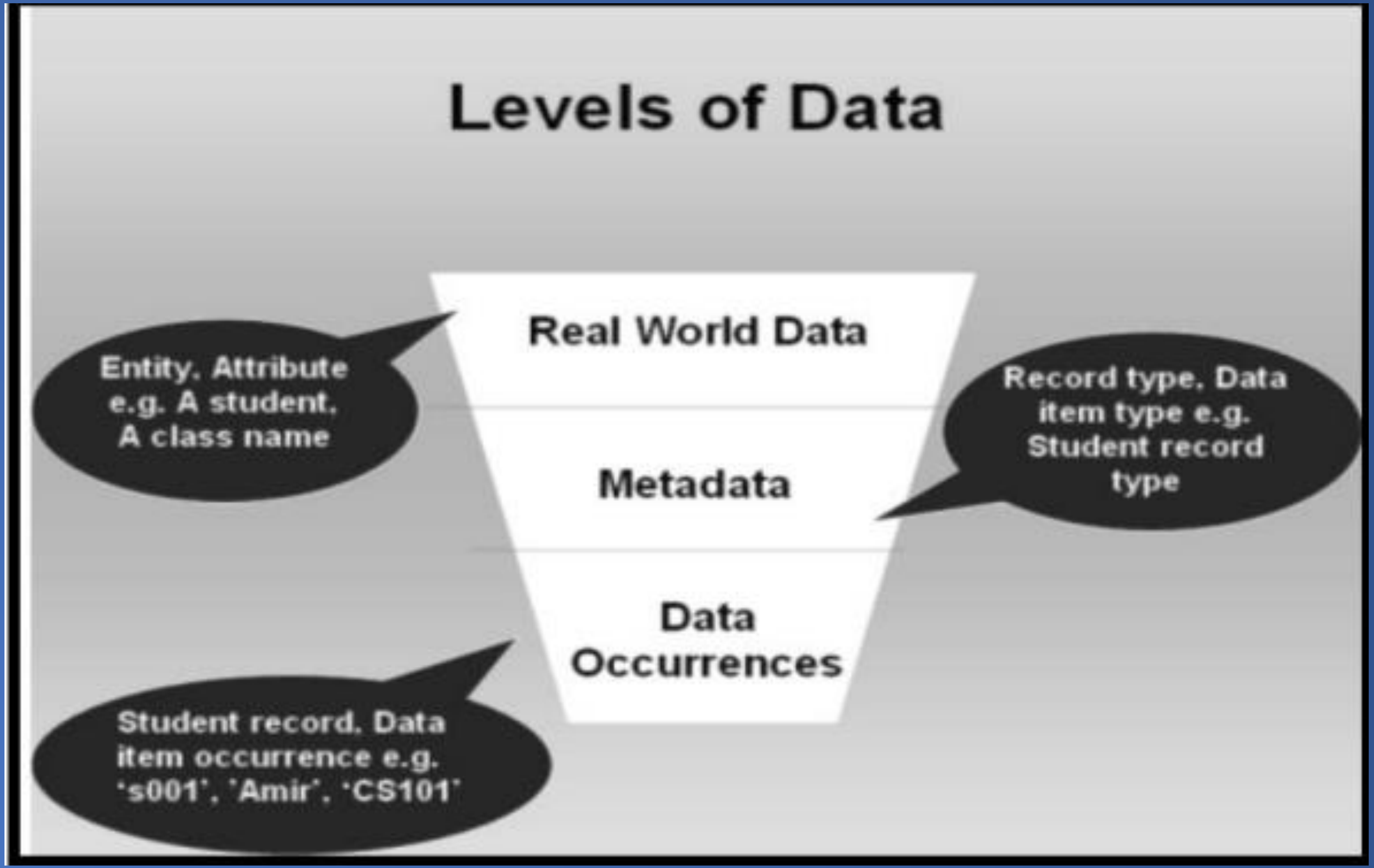
- Many use tools in the form of software packages that work closely with the stored database work closely with the stored database.
- Stand-alone: Mostly maintain personal databases using ready-to-use packaged applications packaged applications.

- An example is a tax program user that creates its own internal database. Another example is a user that maintains an address book

Roles in the Database Environment

- Data Administrator (DA)
- Database Administrator (DBA)
- Database Designers (Logical and Physical)
- Application Programmers
- End Users (naive and sophisticated)

Levels Of Data



Levels of Data

Real World Data

Metadata

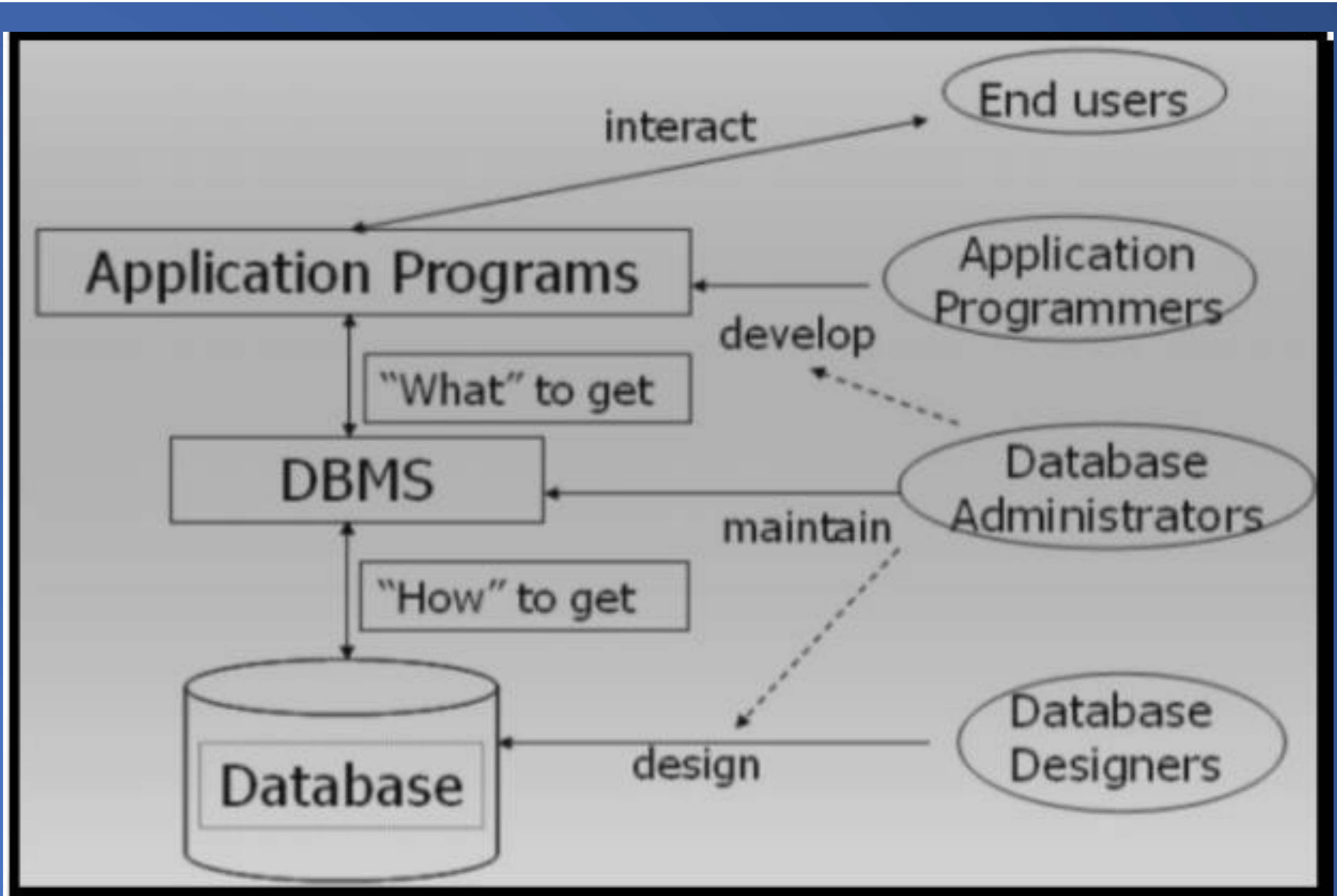
Data Occurrences

Entity, Attribute e.g. A student, A class name

Record type, Data item type e.g. Student record type

Student record, Data item occurrence e.g. 's001', 'Amir', 'CS101'

Database Administration's interaction with other users



Advantages of DBMSs

- Control of data redundancy (data storage, development)
- Sharing of data (among multiple users)
- Improved data integrity(accuracy and consistency of data)
- Improved security
- Enforcement of standards
- Data consistency
- Economy of scale
- More information from the same amount of data

Advantages of DBMSs

- Improved maintenance through data independence
- Increased concurrency
- Improved backup and Recovery Services
- Security mechanism
- Balance conflicting requirements
- Improved data accessibility and responsiveness

Disadvantages of DBMSs

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

Additional Implications of Using the Database Approach

- Potential for enforcing standards:

This is very crucial for the success of database applications in large organizations. Standards refer to data item names, display formats, screens, report, structures, meta-data (description of data), Web page layouts, etc.

- Reduced application development time:

Incremental time to add each new application is reduced

Additional Implications of Using the Database Approach

- Flexibility to change data structures: Database structure may evolve as new requirements are defined.
- Availability of current information: Availability of current information:
- „Extremely important for on-line transaction systems such as airline, hotel, car reservations.
- Economies of scale:
- Wasteful overlap of resources and personnel can be avoided by consolidating data and applications across departments...

When not to use a DBMS

- High initial investment and possible need
    - High initial investment and possible need
    - Overhead for providing generality, security, concurrency control, recovery, and integrity functions.
- When a DBMS may be unnecessary:
    - If the database and applications are simple well defined
    - If there are stringent real-time requirements that may not met because of DBMS overhead.
    - If access to data by multiple users is not required

When not to use a DBMS

- When no DBMS may suffice:
- If the database system is not able to handle the complexity of data because of modelling limitations
- If the database users need special operations not supported by the DBMS

+1 Bonus Mark

- Task – 01: Search, Understand with concepts and then write down about the following:
- Conceptual Design
- Logical Design
- Physical Design

Thank You