Introduction to Database

Associate Prof. Dhiraj Kumar Das
Department of Computer Science and Engineering
Nepal Engineering College, Changunarayan, Bhaktapur
Email: dhirajkd@nec.edu.np

Objective of this course

- to understand relational database model, declarative and procedural query languages (relational algebra and calculus)
- to learn how to use SQL to query and update relational databases, and how to use SQL together with a programming language
- to understand the concept of database design, and Learn how to design good databases
- to understand how a database management system operates, and how it processes queries and transactions

References

- Abraham Silberchatz, Henry F. Korth, S. Sudarshan; Database System Concepts. McGraw Hill, 4th edition
- Date C. J.; An Introduction to Database System. Addison Wesley, 8th edition
- A complete references of **ORACLE 9i**

Basic Concept

- At a Conference in 1990, Silberschatz quoted on database systems- “ The history of database system research is one of exceptional productivity and starting economic impact.
- Barely 20 years old as a basic science research field, database research has fueled an information service industry estimated at $10 billion per year in U.S. alone.
- Achievements in database research underpin fundamental advances in the communication system, transportation and logistics, financial management, knowledge based systems (KBS), accessibility to scientific literature and a host of other civilian and defense application.
- The roots of DBMS lie in file-based systems.

Three generations of DBMS

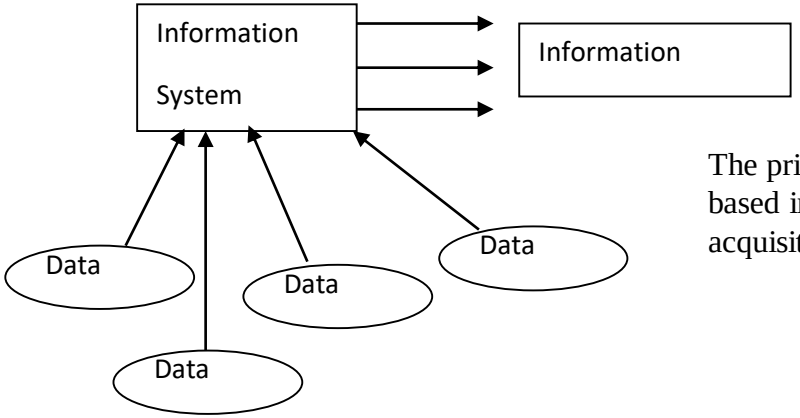
- 1st Generation : The hierarchical and CODASYL systems represent the first generation of DBMS. The hierarchical model is typified by IMS (Information Management System) and the Network or CODASYL model by IDS (Integrated Data Store). Both of these models were developed in mid-1960s.
- 2nd Generation : In 1970, Dr. E.F. Codd represented a research paper at an international conference where he proposed a relational model. It represents the second generation of DBMS.
- 3rd Generation : It is represented by the Object-Relational DBMS and the Object-Oriented DBMS.

Introduction

- A **database management system** (**DBMS**) is a computer program (or more typically, a suite of them) designed to manage a **database**, a large set of structured data, and run operations on the data requested by numerous users.
  - Typical examples of DBMS use include accounting, human resources and customer support systems.
- A database is the solution to store, reference and view the information in various forms.
- It provides various tools and objects to view the information stored as tables.
- It provides more efficient way to manage the information compared to an ordinary files system.
- DBMS provides organization, with the capability to easily integrate and leverage the massive amounts of operational data into meaningful information.
- Database technology has been described as one of the most rapidly growing areas of computer business and information of science.
- Defining a database involves specifying the data types, structures and constraints for the data to stored in the database.
- The DBMS is a general purpose software system that facilitates the processes of defining, constructing and manipulating database for various application.

What is Data?

- Database is a collection of data. It describes the attributes of any real world object, which explain the feature, behavior of that object.
- Data is the most important component in any work that we do. We either use existing data or generate data.
- Data is the plural of datum, which means a single piece of information.
- Data is defined as a known fact that can be recorded and that have implicit meaning.
- Actually, data are raw or isolated facts from which the required information is produced. When this data is gathered and analyzed, it yields information.
  - For example, Exam section gathers the number of student who are scored the grade 'A' in the respective subject. ***Here, the grade constitutes the data.***



The primary objective of any computer-based information handling system is the acquisition and processing of raw data.

What is Data?

- Effective data management enables raw *data* to become useful *information*. ***Thus, information helps us to foresee and plan events.***
- The data are always termed as the raw materials for information. A meaningful information is only possible after processing such a raw data.
- Data must be atomic. The atomic data cannot be broken into pieces.
  - **ANSI definition**, Data is defined as follows: " *Data is a representation of facts, concept or instructions in a formalized manner suitable for communication, interoperation, or processing by humans or by automatic means.*"

What is Information?

- It is defined as a collection of related data that when put together, communicate meaningful and useful message to recipient who uses it, to make decision or to interpret the data to get the meaning.
- Actually, it is data that has been converted into a more useful or intelligible form.
- For example, marks obtained by students and their roll numbers form data where their marksheet is the information.

Difference between Data and Information?

SN	Data	Information
1	Data is raw fact and figures. For example, 'A' is data.	Data when stored in some form like Grade: 'A', then it becomes information
2	Data is not significant to a business.	Information is significant to a business.
3	Data are atomic level piece of the information.	Information is a collection of data.
4	Data does not help in decision-making.	Information helps in decision-making.
5	Data is generally in unorganized form.	Information is in organized form.
6	Data is collected form the source directly and hence is not dependent on information.	Information is dependent on the data that is gathered.
	For example, if we say that 89,95,90,93,98 are the numbers then it becomes data.	For example, if we say that 89,95,90,93,98 are the marks of five different subjects, then it becomes information.

Role of Data in Business

- In the world of business, predicting an event and planning for it could save money and time for an organization.
- Let's consider an example, A company manufacturing high quality food grade plastic containers is planning its annual purchase of a certain raw material. With data on the sales of these containers for the last 10 years, the company heads can find out the percentage increase in the consumption of the product. Let's assume they find a percentage increase in the range of 22 to 30% annually. Based on their findings, a production plan can be prepared, which will cater to a production increase of 30%.
- Thereby the company will now place an order only for that much quantity of raw material, thus sawing precious foreign exchange, warehouse space and other related resources.

What is Database?

"A Database is a self-describing collection of integrated files"

- **David M. Kroenke**

"**Self-describing**" - The database contains, within itself, a description of its structure.

"**Integrated files**" - A database is a collection of files and relationships among records in those files.

"A database is a repository for stored data. In general, it is both integrated and shared."

- **C. J. Date**

Integrated - "A database may be thought of as a unification of several otherwise distinct data files, with any redundancy … partially or wholly removed."

Shared - "Individual pieces of data may be shared among several different users"

What is Database?

"… a collection of interrelated data stored together without harmful or unnecessary redundancy to serve multiple applications; the data are stored independent of programs which use the data; a common and controlled approach is used when accessing data held within the database …"

• **James Martin**

• The data in a database should have the following features:

i) Shared: Data in a database is shared among different uses and applications.
ii) Persistence: Data in a database exists permanently.
iii) Validity: Data should be correct with respect to the real world entity that they represent.
iv) Security: Data should be protected from unauthorized access.
v) Consistency: Whenever more then one data element in a database represents related real world values should be consistent with respect to the relationship.
vi) Non-redundancy: No two data items in a database should represent the same real world entity.
vii) Independence: Data should be abstract i.e. it should be independent of each other at different levels. Changes at one level should not affect the other level.

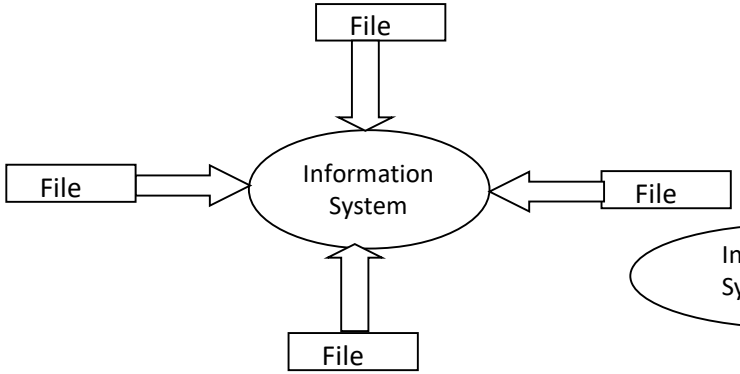
File Processing System

- Before the development of the DBMS, file processing system is used. The actual reason for the introduction of DBMS is various flaws in the conventional file processing system.
- A file is a collection of data about a single entity i.e. files are collection of similar records. Files are typically designed to meet needs of a particular department or user group. Files are also typically designed to be part of a particular computer application.
- Data storages is build around the corresponding application that uses the files.
- The typical file processing system is supported by a conventional operating system. In file processing system, permanent records are stored in various files, and a number of application programs are written to extract records form and add records to the appropriate files.
- Users of file processing systems are almost always at the mercy of the Information Systems department to write programs that manipulate stored data and produce needed information such as printed reports and screen displays.

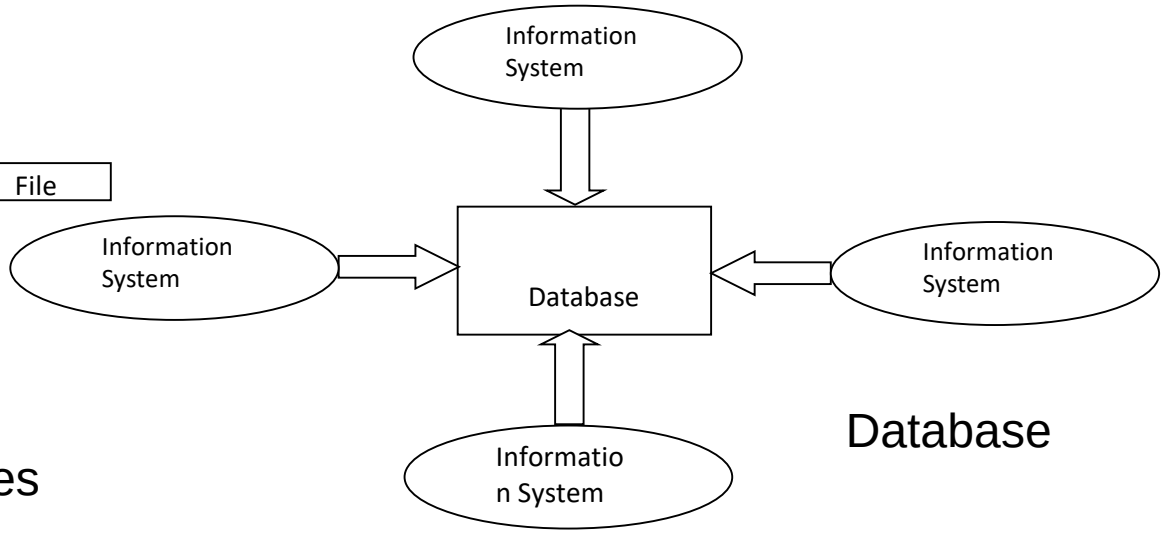
Disadvantage File Processing System

File Processing System has a number of disadvantages, they are:

1. Data redundancy and Inconsistency
   - Data redundancy means repetition of same data in the database.
   - Data Inconsistency means not a similar data in database for a particular time.
   - In conventional data systems, an organization often builds a collection of application programs often created by different programmers and requiring different components of the operational data of the organization.
   - The data in conventional data systems is often not centralized.
   - Some applications may require data to be combined from several systems.
   - These several system could well have data that is redundant as well as inconsistent; i.e. different copies of the same data may have different values.



Conventional Files



Database

Disadvantage File Processing System

2. Difficulty in Accessing Data
    - Suppose the bank officers need to find the names of all customer who live in Lalitpur, Patan. The officer asks the data processing departments to generate such a list. Since there is no such program to fulfill the needs, hence officer has two choices either get the list of customers and get required information manually or write the application program. Hence there is difficulty in accessing data.
3. Data isolation
    - Since data is scattered in various field may be in different formats, it is difficult to write new application programs to retrieve the appropriate data.
4. Security Problem
    - Not every user of the database system should be able to access all the data.
5. Integrity Problems
    - The data values stored in the data must satisfy certain type of consistency constraints. For example, the balance of the bank account may never fall below the prescribed amount (say Rs. 500.00). These constraints are reinforced in system by adding appropriate code in various application.

Database System Versus File Processing System

DBMS	File Processing System
Data redundancy problem is not found.	Here redundancy problem exist.
Data inconsistency does not exist.	It also exists in this.
Accessing database is easier	Accessing database is comparatively difficult
The problem of data isolation is not found in it	Data are scattered in various files in different formats. Writing new program to retrieve appropriate data is difficult.
Atomicity and integrating are not found	Atomicity and integrating problems are found
Security of data is in database	Here is not good.

Database Management System (DBMS)

A **DATABASE SYSTEM**, also known as a Database Management System (DBMS), are large complex pieces of software designed specifically for the efficient management of data.

i.e. It is a software system that provides facilities for storing and retrieving data quickly and in a format that matches user needs.

Database + DBMS software = Database System

Examples:

Oracle (Oracle Corporation)

Ingres (Computer Associates)

SQL Server (Microsoft Corporation)

Access (Microsoft Corporation)

IMS, DB2 (IBM)

And many more…

Database Management System (DBMS) contd .....

Databases system typically store data on a persistent or non-volatile medium such as a disk which does not lose the data when the power is shut off.

Databases are used everywhere including in corporations, ecommerce, telecommunications, government agencies, pharmaceutical research, universities, and personal applications.

Database systems are the underpinning of the information society. Without them, it would be difficult to conduct just about any business activity, especially those involving large amounts of data.

Database system Provide

- persistent storage
- data definition
- Data manipulation (queries for reading, updating, inserting, ad data deleting)
- allow multiple users to simultaneously access the database without interference
- fast access (via indexes)
- alerts
- data consistency maintenance
- data security (preventing unauthorized users from accessing data)
- bulk loading (fast loading of large amounts of data)
- backup and
- recovery from software and hardware crashes, and from media failure.

Database system Provide (Cont....)

Specialized database system provide some additional facilities or put restrictions on these facilities to provide some additional features or put restrictions on these facilities to provide very fast access. For example, these are:

- ✓ single user database systems which do not have the overhead of supporting multiple simultaneous users,
- ✓ Data warehouses, which are database system that support data analysis, read-only queries, and so forth, which are capabilities that do not need "transaction" support,
- ✓ Main memory database systems, which keep all data in memory.

Database Applications

- Banking: all transactions
- Airlines: reservations, schedules
- Universities: registration, grades
- Sales: customers, products, purchases
- Manufacturing: production, inventory, orders, supply chain
- Human resources: employee records, salaries, tax deductions

Databases touch all aspects of our lives

Purpose of Database System

- In the early days, database applications were built on top of file systems
- Drawbacks of using file systems to store data:
  - Data redundancy and inconsistency
    - Multiple file formats, duplication of information in different files
  - Difficulty in accessing data
    - Need to write a new program to carry out each new task
  - Data isolation — multiple files and formats
  - Integrity problems
    - Integrity constraints (e.g. account balance > 0) become part of program code
    - Hard to add new constraints or change existing ones

Purpose of Database System (Cont….)

- Drawbacks of using file systems (cont.)
  - ✓ Atomicity of updates
    - ➢ Failures may leave database in an inconsistent state with partial updates carried out
    - ➢ E.g. transfer of funds from one account to another should either complete or not happen at all
  - ✓ Concurrent access by multiple users
    - ➢ Concurrent accessed needed for performance
    - ➢ Uncontrolled concurrent accesses can lead to inconsistencies
      - – E.g. two people reading a balance and updating it at the same time
  - ✓ Security problems

Database systems offer solutions to all the above problems

Brief History of DBMS

- Pre-1960s
  - ✓ Data was stored in files.
  - ✓ Each file was described in the program code of every program that used it.
  - ✓ Each application system was, in effect, a special-purpose Data Base Management System.

- Emergence of the “database” concept
  - ✓ In the early 1960s, the term “database” began to be applied to the *integrated collection of files* each organization, or section within an organization, needed to maintain.
  - ✓ There was a need perceived for general-purpose “Database Management Systems” to be supplied by software vendors to enable data bases to be built and used effectively.

Advantages of Database Processing

- Extracting Information from Data
- Sharing of Data
- Standards
- Controlled Redundancy
- Integrity Control
- Security
- Economy of Scale
- Data Independence

1. Extracting Information from Data

- ✓ Storing data in a central shared database enables users to efficiently extract information.
- ✓ This information may include analyses of patterns, trends and derived characteristics of the data.
- ✓ Database query languages and report generators provided by most DBMS packages enable a common and controlled approach to information extraction.

2. Sharing of Data

- ✓ Data sharing refers to the capacity to make data simultaneously accessible by many users without mutual interference.
- ✓ Sharing of data reduces redundancy (no need to store multiple copies of data for different users).
- ✓ Security of access is easier to control and data integrity is easier to ensure.
- ✓ DBMS package needs to provide some measure of concurrency control to ensure no mutual interference.

3. Standards

- Centralization of data storage requires that design standards and methodologies be developed and adhered to across the business enterprise.

- It is one of the roles of the DBA (Database Administrator) to ensure that standards be observed. Eg. Data naming standards, enforcement of data integrity rules, coding standards – error handling, data access standards

4. Controlled Redundancy

- Pre-database applications employed dedicated files. That is, the Payroll application, say, “owned” its own files, as did say, the Projects application. Often, the same data, e.g. an Employee file, was held by more than one application.

- Redundant data was often not easily identifiable since the only place the files were described was in the programs accessing them.

- Problems associated with redundant data:
  - wastage of storage space
  - wastage of processing time on multiple updates
  - possibility of data inconsistency

5. Integrity Control

- ✓ Data integrity ensures that data does not become inconsistent.
- ✓ Integrity constraints (rules) can be defined to the DBMS in order to protect the integrity of data.
- ✓ Most DBMS software will allow various degrees of specification and enforcement of integrity constraints.

6. Security

- Security is important because of data integration - some of the data in the central database may be of a sensitive and/or critical nature.
- Access to data may be restricted only to authorized users.
- Different levels of access may be defined depending on the role of the user in the organization.
- Database administrator can define and police security policies and backup / recovery routines.

7. Economy of Scale

- Data is combined into a single database.
- Multiple applications can be standardized to interact with this one data source.
- Therefore, cost savings may be realized because there is no need for individual departmental data management budgets.

8. Data Independence

- Description of the data is stored in one central place (the *schema*). Therefore, applications do not have to be recompiled when the format of the data changes. Hence lower maintenance overheads.
  - Logical data independence.
  - Physical data independence.

Disadvantages of Database Processing

- Extra amount will be required for providing security and recovery
- High preliminary investment and Probable need for extra hardware
- Data entry cost
- Cost of Training of using the database
- Cost of maintaining database
- Larger file size
- Increased complexity
- Greater impact of failure
- More difficult recovery
- Complex, Difficult and Time consuming
- Additional hardware costs
- Conversion
- costs

Function of DBMS

- The functions performed by a typical DBMS are the following:
  - Data Definition

    The DBMS provides functions to define the structure of the data in the application. These include defining and modifying the record structure, the type and size of fields and the various constraints/conditions to be satisfied by the data in each field.
  - Data Manipulation
  - Data Security and Integrity
  - Data recovery and Concurrency
  - Data Dictionary Maintenance
  - Performance

Levels of Abstraction

For the system to be usable, it must be retrieve data efficiently. The need for efficiency has led to designers to use complex data structures to represent data in the database. Since many database systems users are not computer trained, developers hide the complexity from the users through several levels of abstraction, to simply users interactions with the system:

1. Physical level
2. Logical level
3. View level

Levels of Abstraction (cont.)

- Physical level: it is lowest level of abstraction and it describes **how** a record (e.g., customer) are actually stored. At the physical level, complex level data structures are described in details.(e.g. index, B-tree, hashing)
- Logical level: New higher level of abstraction describes what data are actually stored and relationship that exit among the data.

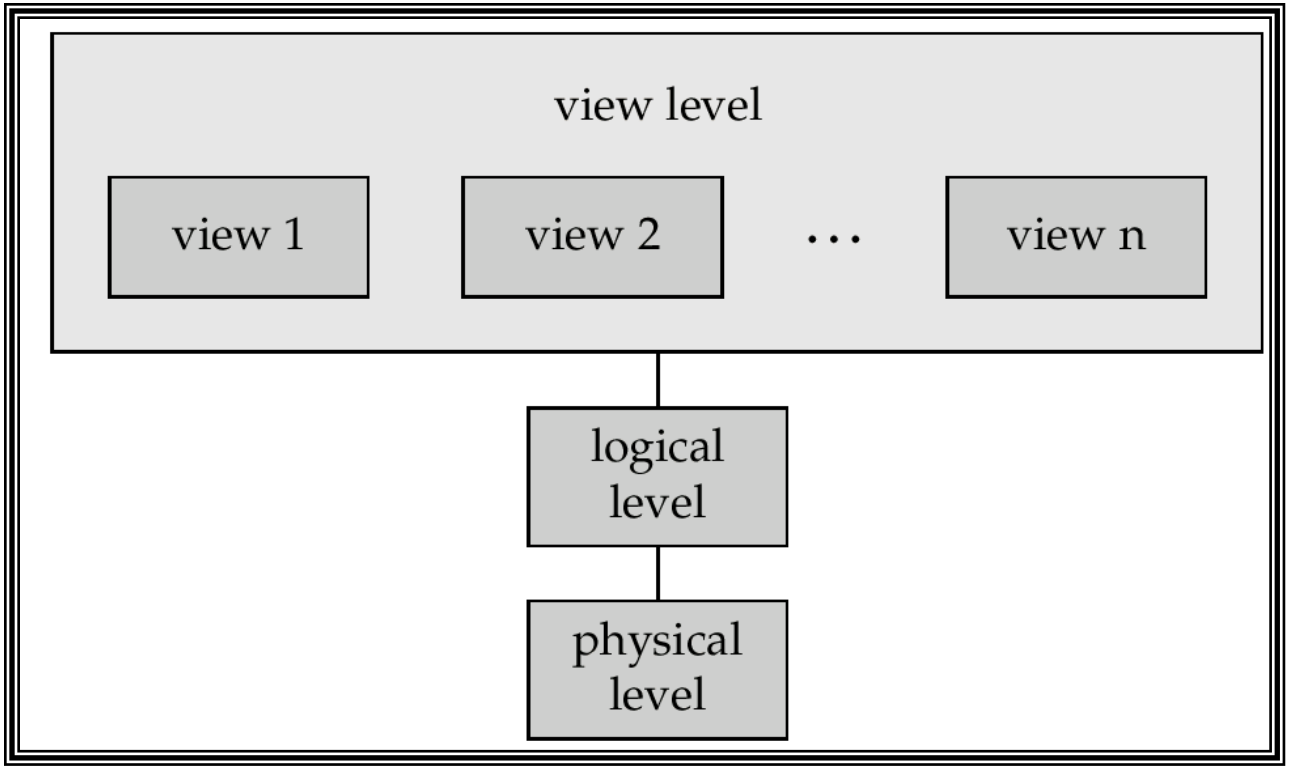
```
type customer = record
                    name : string;
                    street : string;
                    city : integer;
                end;
```

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

View of Data

A database system is a collection of interrelated files and a set of program that allow a users to access and modify these files. A major purpose of database system is to provide user users with an abstract view of data. That is, the system hides certain details of how the data are stored and maintained.

An architecture for a database system



Schema and Instances

Schema

- Overall design of the database is called the database schema.
- Database schemas have various schemas, partitioned according to the levels of data abstraction that we discussed.
- Three level of schema
  i) External Schema
  ii) Conceptual Schema
  iii) Internal Schema

Instances

- The collection of information stored in the database at a particular moment is called an instance of the database.
  – Analogous to the value of a variable

Data Independence

- The three schema architecture can be used for explaining the concept of data independence, which can be defined as the capacity to change the schema at one level of a database system without having to change the schema at the next higher level.
- Two types of data independence
  - i) Logical data independence
  - – Logical data independence is the capacity to change the conceptual schema without having to change external schemas or application programs.
  - – We may change the conceptual schema to expand the database (adding of new records or data) , or to reduce the database (removing records or data).
  - i) Physical data independence
  - – Physical data independence is the capacity to change the internal schema without having to change the conceptual schemas.

Benefits of Three Schema Architecture

- The capacity to change the conceptual schema without having to change external schema or application program. (Logical data independence)

- The capacity to change the internal schema without having to change the conceptual schema. (Physical data independence)

Database Languages

- A database system provides a data definition language to specify the database schema and a data manipulation language to express queries and updates.
- In practice , the DDL and DML are not two separate languages; instead they simply form parts of a single database language, such as the widely used SQL language.

Data Definition Language (DDL)

- Specification notation for defining the database schema
  - E.g.

    create table *account* (
    account-number **char**(10),
    balance **integer**)

- DDL compiler generates a set of tables stored in a *data dictionary*
- Data dictionary contains metadata (i.e., data about data)
  - database schema
  - Data *storage and definition* language
    - language in which the storage structure and access methods used by the database system are specified
    - Usually an extension of the data definition language

Data Manipulation Language (DML)

- Language for accessing and manipulating the data organized by the appropriate data model
  - DML also known as query language
- 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
- SQL is the most widely used query language

SQL

- SQL: widely used non-procedural language
  - E.g. find the name of the customer with customer-id 192-83-7465

    select *customer.customer-name*
    from *customer*
    where *customer.customer-id* = '192-83-7465'
  - E.g. find the balances of all accounts held by the customer with customer-id 192-83-7465

    select *account.balance*
    from *depositor*, *account*
    where *depositor.customer-id* = '192-83-7465' **and**
    depositor.account-number = account.account-number
- Application programs generally access databases through one of
  - Language extensions to allow embedded SQL
  - Application program interface (e.g. ODBC/JDBC) which allow SQL queries to be sent to a database

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
- Naïve users – invoke one of the permanent application programs that have been written previously
  - E.g. people accessing database over the web, bank tellers, clerical staff

Database Administrator (DBA)

- Database administrator is a persons having central control of both the data and program accessing that data.
- 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 (DBA) Responsibilities

- The responsibility of a database administrator vary and depend on the job description, corporate and Information Technology (IT) policies and the technical features and capabilities of the DBMS being administrated.
- 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

DBMS Components

- Database engine
  - Storage
  - Retrieval
  - Update
- Query processor: The query processor subsystem compiles and executes DDL and DML statements.
- Data dictionary: The data dictionary contains METADATA. The schema of a table is an example of metadata.
- Utilities
- Security
- Report writer
- Forms generator(input screens)
- Application generator
- Communications

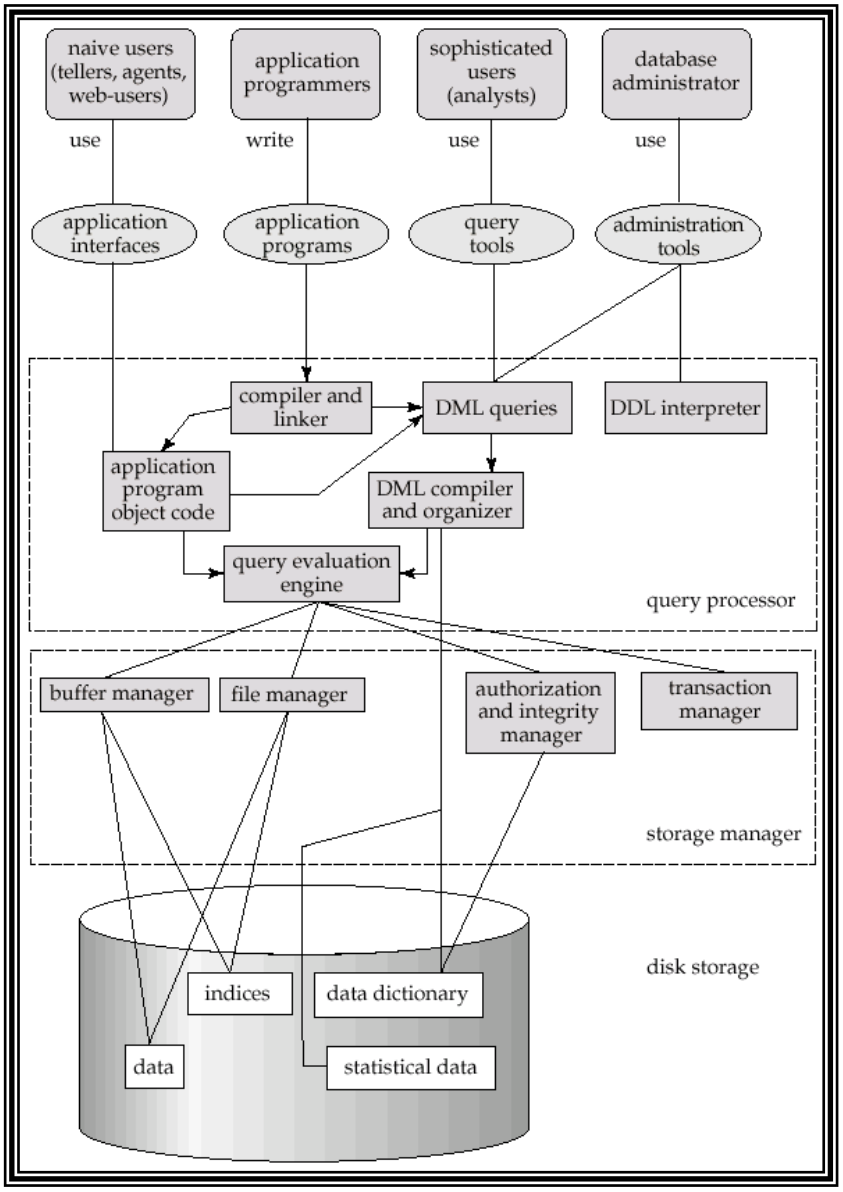
Transaction Management

- 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

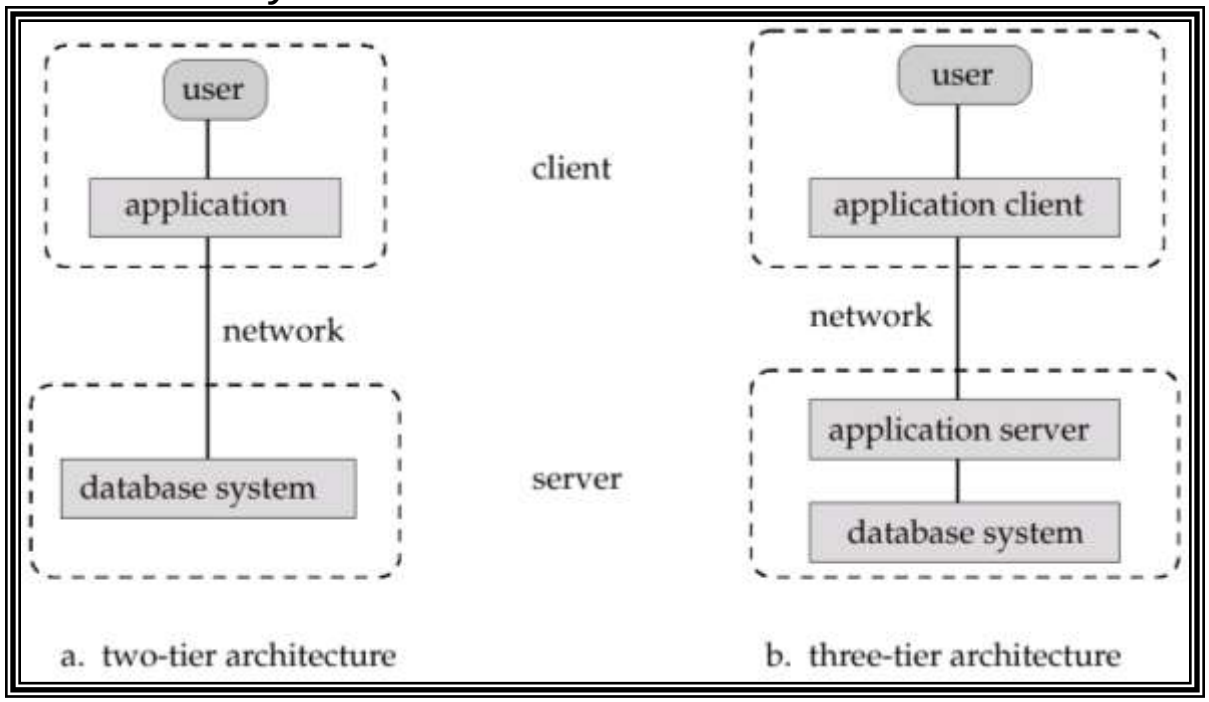
- Storage manager is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.
- The storage manager is responsible to the following tasks:
  - interaction with the file manager
  - efficient storing,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

Overall System Structure



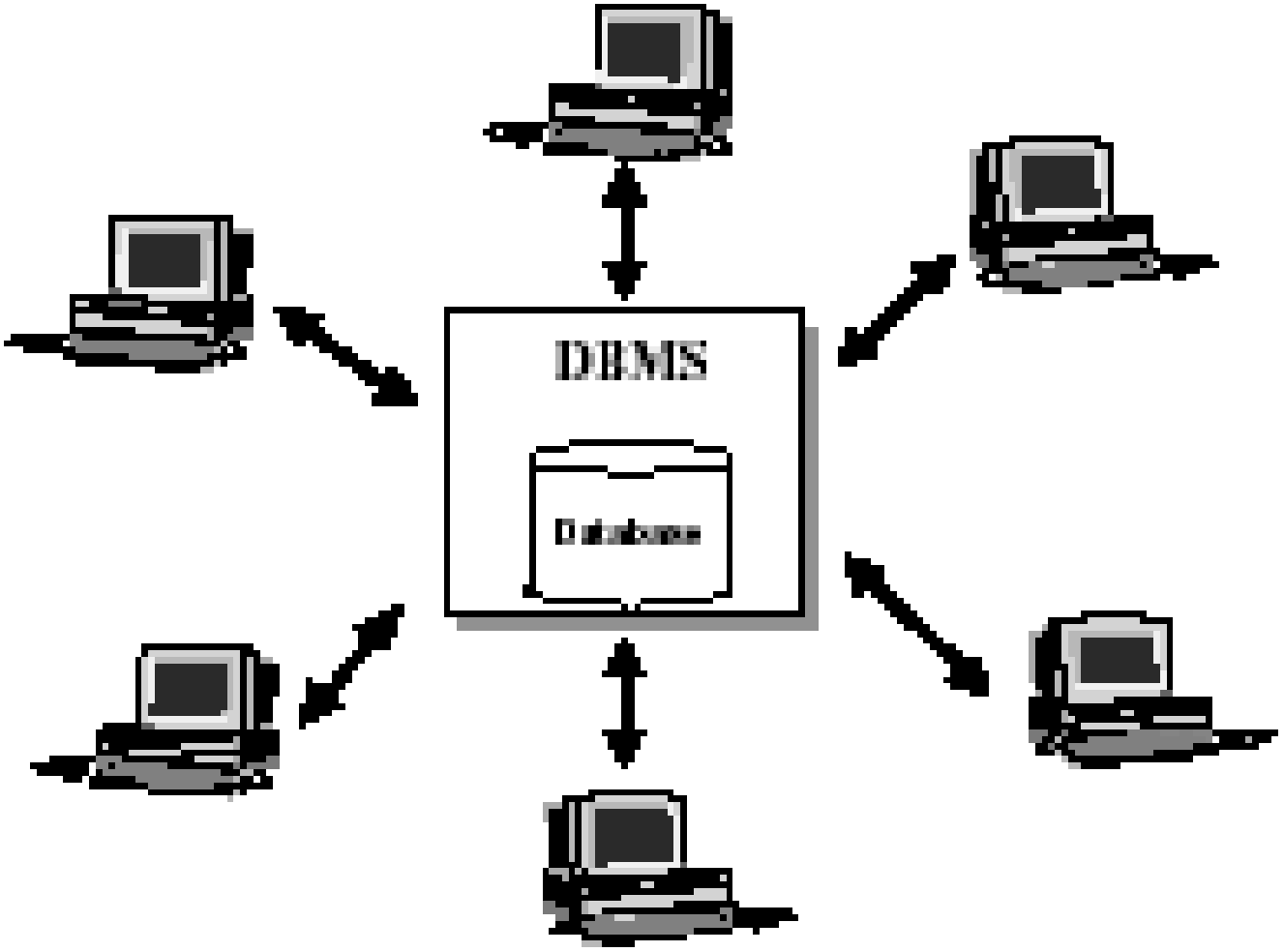
Application Architecture

Most user of a database system today are not present at the sit of the database system, but connect it through network. We can therefore differentiate between client machines, on which remote database user works, and server machines, on which the database system runs

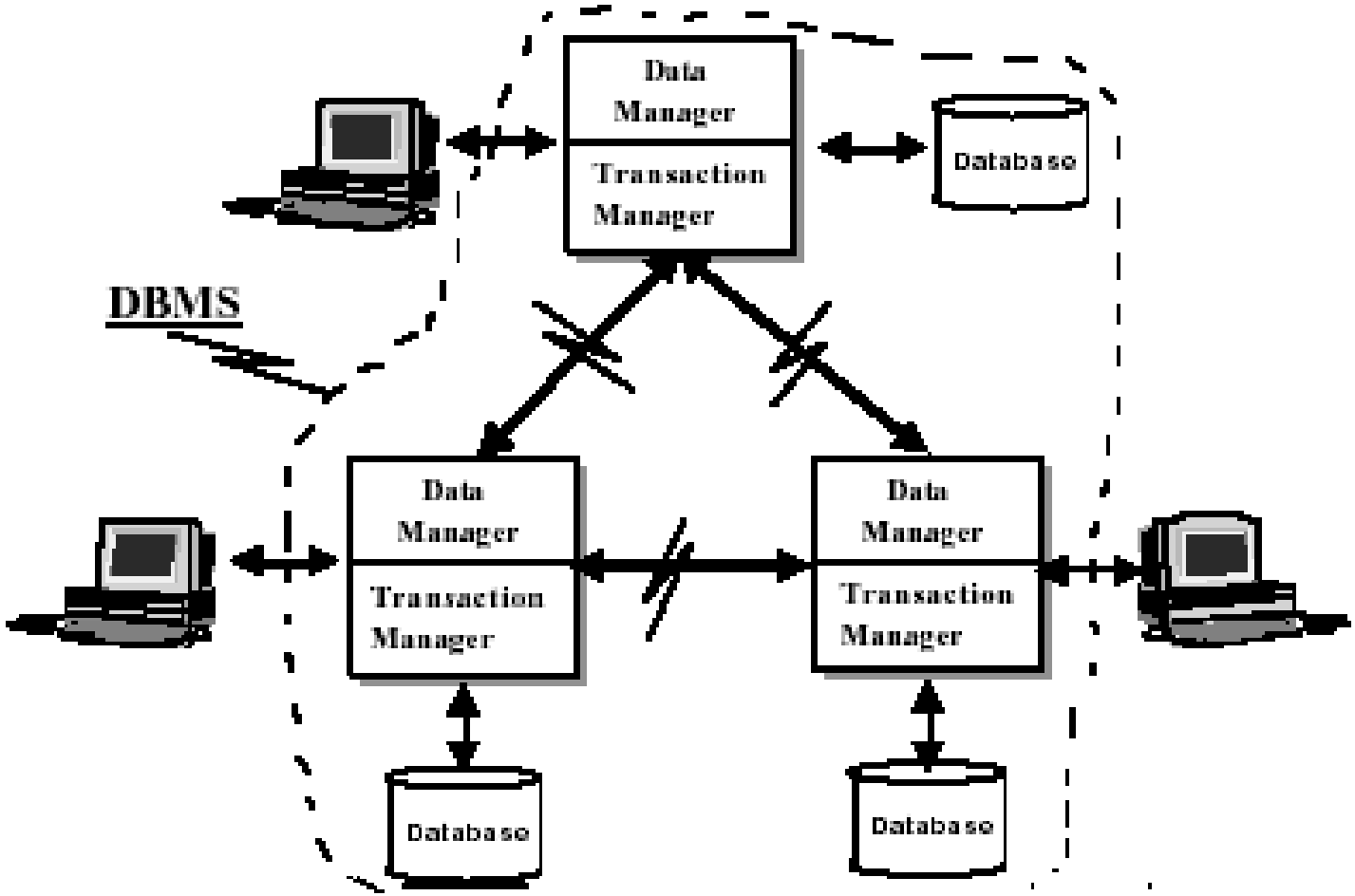


- **Two-tier architecture**: E.g. client programs using ODBC/JDBC to communicate with a database
- **Three-tier architecture**: E.g. web-based applications, and applications built using “middleware”

Centralized Database system



Distributed Database System



Client/Server Architecture