

## Chapter 1

# Introduction

### 1.1 INTRODUCTION TO DBMS

---

In modern society Databases and database systems play a vital role. Most of our day to day activities involve the use of databases. Bank application, ticket reservations for any sort of applications, computerized library etc can be cited as instances for use of databases.

**Data:** Known facts that can be recorded that have implicit meaning.

**Information:** When data is processed, organized, structured or presented in a given context it becomes information.

**Database:** Collection of data.

**DBMS:** DBMS is a collection of interrelated data and a set of program to access those data. The primary goal of a DBMS is to provide a way to store and retrieve database information that is both *convenient and efficient*.

The DBMS manages incoming data, organizes it, and provides ways for the data to be modified or extracted by users or other programs.

**Database System:** Database and DBMS collectively known as database system. Database system is a computerized record keeping system. It is a repository or container for a collection of computerized data files. Users of the system can perform or request the system to perform a variety of operations such as adding new files to the database, inserting data into existing files, retrieving or deleting data from existing files, modifying data in existing file or removing existing files from the database.

#### 1.1.1 Structure of DBMS

An overview of the structure of database management system is shown in Fig. 1.1. A DBMS is a software package, which translates data from its logical representation to its physical representation and vice versa.

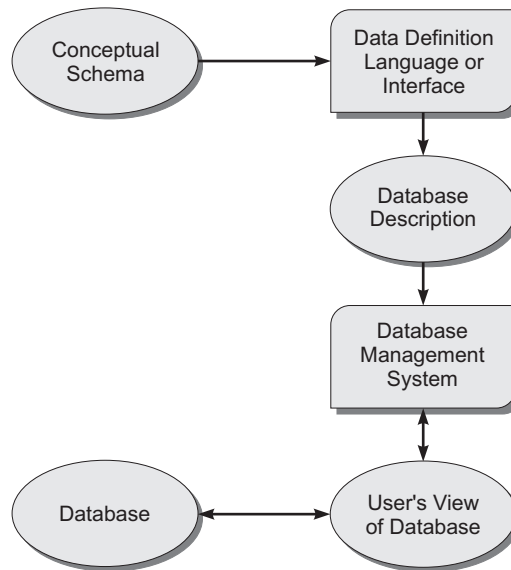
The DBMS uses an application specific database description to define this translation. The database description is generated by a database designer from his or her conceptual view of the database, which is called the Conceptual Schema. The translation from the conceptual schema to

the database description is performed using a Data Definition Language (DDL) or a graphical or textual design interface.

### 1.1.2 Applications of Database

- **Banking:** for all sorts of bank transactions like withdrawal, deposit etc.
- **Airlines:** for reserving tickets and to prepare schedules
- **Universities:** for student information, registration, grades
- **Sales:** customers, products, purchases
- **Online retailers:** order tracking, customized recommendations
- **Manufacturing:** production, inventory, orders, supply chain
- **Human resources:** employee records, salaries, tax deductions

Databases touch all aspects of our lives.



**Fig. 1.1.** Structure of database management system.

### 1.1.3 Classification of Database Management System

The database management system can be broadly classified into:

1. Passive Database Management System
2. Active Database Management System

1. **Passive Database Management System:** Passive Database Management Systems are program-driven. In passive database management system the users query the current state of database and retrieve the information currently available in the database. Traditional DBMS is passive. Applications send requests for operations to be performed by the DBMS and wait for the DBMS to confirm and return any possible answers.

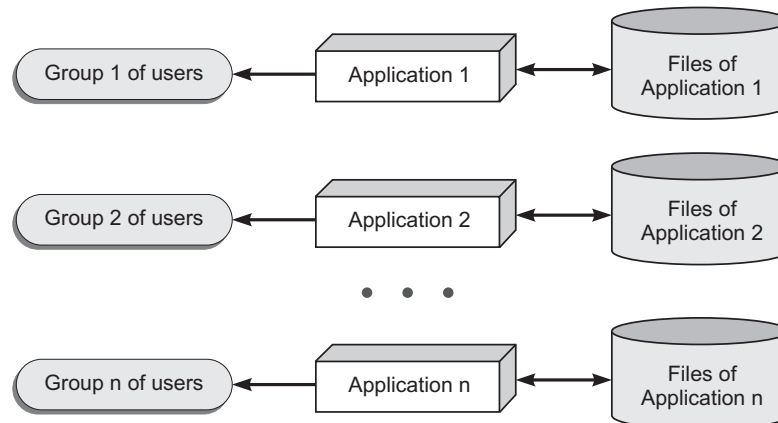
- 2. Active Database Management System:** Active Database Management Systems are data-driven or event-driven systems. In active database management system, the users specify to the DBMS the information they need. If the information is currently available, the DBMS actively monitors the arrival of the desired information and provides it to the relevant users. The scope of a query in a passive DBMS is limited to the past and present data, whereas the scope of a query in an active DBMS additionally includes future data.

## 1.2 FILE PROCESSING SYSTEM

A file processing system is a collection of files and programs that access or modify these files. New files and programs are added by the programmers when there is a need for,

- Storing new information and
- New ways to access information

File processing system is supported by a conventional operating system. The system stores permanent records in various files. The system needs different application programs to extract records from and add records to the appropriate files. Each program defines and manages its own data.



**Fig. 1.2.** File based system.

### 1.2.1 Drawbacks of Conventional File Processing System

The conventional file processing system has a number of drawbacks.

- (i) **Data Redundancy and Inconsistency:** Since the files and application programs are created by different programmers over a long period of time, the files have different formats and the programs may be written in several programming language. The same piece of information may be duplicated in several files. For instance, in bank application, the address and phone number of particular customer may appear in a file that consists of personal information and in saving account records file too. This redundancy leads to data consistency, that is, the various copies of the same data may no longer agree.

**For example,** a changed customer address may be reflected in personal information file, but not in saving account records file.

This redundancy leads to wastage of storage space, high access cost and data inconsistency.

- (ii) **Difficulty in Accessing Data:** Conventional file processing environments do not allow needed data to be retrieved in a convenient and efficient manner.

*For example*, if a bank officer needs to find out the names of all customers who live within a particular area, he has two choices:

1. Get the list of customers and extract the needed information manually.
2. Ask the data processing department to have a system programmer write the necessary application program. Both alternatives are unsatisfactory.

- (iii) **Data Isolation:** Data is scattered in various files, and as the files may be in different formats, it is difficult to write new application programs to retrieve appropriate data.

- (iv) **Concurrent Access Anomalies:** In order to improve the overall performance of the system and obtain a faster response time many systems allow multiple users to update the data simultaneously. In such environment, interaction of concurrent updates may result in inconsistent data.

Consider bank account A, with ₹ 500. If two customers withdraw funds (say ₹ 50 and ₹ 100 respectively) from account A at the same time, the result of the concurrent executions ₹ 400, rather than ₹ 350. In order to guard against this possibility, some form of supervision must be maintained in the system.

- (v) **Security Problems:** Not every user of the database system should be able to access all the data.

*For example*, in a banking system, pay roll personnel need to be only part of the database that has information about the various bank employees. They do not need access to information about customer accounts. Since application programs are added to the system in an ad-hoc manner, it is difficult to enforce such security constraints.

- (vi) **Integrity Problems**

- The data values stored in the database must satisfy certain types of consistency constraints.

*For example*, the balance of a bank account may never fall below a prescribed amount (say ₹ 100). These constraints are enforced in the system by adding appropriate code in the various application programs. However, when new constraints are added, it is difficult to change the programs to enforce them. The problem is compounded when constraints involve several data items from different files.

- (vii) **Atomicity Problems**

- Like all devices a computer system may also be subjected to failure. If a failure occurs the data existed prior to failure can be restored to the consistent state.
- Consider a program to transfer ₹ 50 from account A to account B. If a system failure occurs during the execution of the program, it is possible that the ₹ 50 is removed from account A but is not credited to account B, resulting in an inconsistent database state.
- The funds transfer must be atomic, it must happen entirety or not at all. It is difficult to ensure atomicity in a conventional file-processing system.

### 1.3 ADVANTAGES OF DATABASE

---

Database is a way to consolidate and control the operational data centrally. The DBMS has a number of advantages as compared to traditional computer file processing approach. The DBA must keep in mind these benefits or capabilities while designing databases, coordinating and monitoring the DBMS. The major advantages of having a centralized database are as follows:

**(i) Controlling Data Redundancy**

- In traditional computer file processing, each application program has its own files. In this case, the duplicated copies of the same data are created at many places. In DBMS, all the data of an organization is integrated into a single database. The data is recorded at only one place in the database and it is not duplicated.
- When they are converted into database, the data is integrated into a single database so that multiple copies of the same data are reduced to single copy.
- In DBMS, the data redundancy can be controlled or reduced but is not removed completely. Sometimes, it is necessary to create duplicate copies of the same data items in order to relate tables with each other.
- By controlling the data redundancy, you can save storage space. Similarly, it is useful for retrieving data from database using queries.

**(ii) Data Consistency**

- By controlling the data redundancy, the data consistency is obtained. If a data item appears only once, any update to its value has to be performed only once and the updated value is immediately available to all users.

**(iii) Data Sharing**

- In DBMS, data can be shared by authorized users of the organization. The DBA manages the data and gives rights to users to access the data. Many users can be authorized to access the same set of information simultaneously.
- The remote users can also share same data. Similarly, the data of same database can be shared between different application programs.

**(iv) Data Integration**

- Data in database is stored in tables. A single database contains multiple tables and relationships can be created between tables. This makes easy to retrieve and update data.

**(v) Integrity Constraints**

- Integrity constraints or consistency rules can be applied to database so that the correct data can be entered into database.
- The constraints may be applied to data item within a single record or they may be applied to relationships between records.

**(vi) Data Security**

- Data security is the protection of the database from unauthorized users. Only the authorized persons are allowed to access the database. Some of the users may be allowed to access only a part of database i.e., the data that is related to them or related to their department.

- Mostly, the DBA or head of a department can access all the data in the database. Some users may be permitted only to retrieve data, whereas others are allowed to retrieve as well as to update data. The database access is controlled by the DBA. He creates the accounts of users and gives rights to access the database. Typically, users or group of users are given usernames protected by passwords.
- The user enters his/her account number or username and password to access the data from database.

*For example*, if you have an account of e-mail in the “gmail.com”, you have to give your correct username and password to access your account of e-mail. Similarly, when you insert your ATM card into the Auto Teller Machine (ATM) in a bank, the machine reads your ID number printed on the card and then asks you to enter your pin code to access your account.

#### (vii) Data Atomicity

- A transaction in commercial databases is referred to as atomic unit of work. For example, when you purchase something from a point of sale terminal, a number of tasks are performed such as;
  - Company stock is updated.
  - Amount is added in company’s account.
  - Sales person’s commission increases etc.
- All these tasks collectively are called an atomic unit of work or transaction. These tasks must be completed in all; otherwise partially completed tasks are rolled back. Thus through DBMS, it is ensured that only consistent data exists within the database.

#### (viii) Development of Application

- The cost and time for developing new applications is reduced. The DBMS provides tools that can be used to develop application programs.
- For example*, some wizards are available to generate Forms and Reports. Stored procedures facility reduces the size of application programs.

#### (ix) Creating Forms

- Form is very important object of DBMS. You can create Forms very easily and quickly in DBMS, once a Form is created, it can be used many times and it can be modified very easily. The created Forms are also saved along with database and behave like a software component.
- A Form provides very easy way (user-friendly interface) to enter data into database, edit data, and display data from database.
- The non-technical users can also perform various operations on databases through Forms without going into the technical details of a database.

#### (x) Report Writers

- Most of the DBMSs provide the report writer tools used to create reports. The users can create reports very easily and quickly.
- Once a report is created, it can be used many times and it can be modified very easily.

- The created reports are also saved along with database and behave like a software component.

**(xi) Control over Concurrency**

- In a computer file-based system, if two users are allowed to access data simultaneously, it is possible that they will interfere with each other.

*For example*, if both users attempt to perform update operation on the same record, then one may overwrite the values recorded by the other. Most DBMSs have sub-systems to control the concurrency so that transactions are always recorded with accuracy.

**(xii) Backup and Recovery Procedures**

- In a computer file-based system, the user creates the backup of data regularly to protect the valuable data from damaging due to failures to the computer system or application program. It is a time consuming method, if volume of data is large.
- Most of the DBMSs provide the 'backup and recovery' sub-systems that automatically create the backup of data and restore data if required.

*For example*, if the computer system fails in the middle or at the end of an update operation of the program, the recovery sub-system is responsible for making sure that the database is restored to the state it was before the program started executing.

**(xiii) Data Independence**

- The separation of data structure of database from the application program that is used to access data from database is called data independence.
- In DBMS, database and application programs are separated from each other. The DBMS sits in between them. You can easily change the structure of database without modifying the application program.

*For example*, you can modify the size or data type of a data items. On the other hand, in computer file-based system, the structure of data items is built into the individual application programs. Thus the data is dependent on the data file and vice versa.

**(xiv) Advanced Capabilities**

- DBMS also provides advance capabilities for online access and reporting of data through Internet.
- Today, most of the database systems are online.
- The database technology is used in conjunction with Internet technology to access data on the web servers.

## 1.4 DISADVANTAGES OF DATABASE

Although there are many advantages the DBMS may also have some minor disadvantages. They are:

**(i) Cost of Hardware and Software**

- A processor with high speed of data processing and memory of large size is required to run the DBMS software.
- It means that you have to upgrade the hardware used for file-based system. Similarly, DBMS software is also very costly.

**(ii) Cost of Data Conversion**

- When a computer file-based system is replaced with a database system, the data stored into data file must be converted to database files. It is difficult and time consuming method to convert data of data files into database.
- It is necessary to hire a DBA or database designer and system designer along with application programmers.

**(iii) Cost of Staff Training**

- Most DBMSs are often complex systems so the training for users to use the DBMS is required.
- Training is required at all levels, including programming, application development, and database administration.
- The organization has to pay a lot of amount on the training of staff to run the DBMS.

**(iv) Appointing Technical Staff**

- The trained technical persons such as database administrator and application programmers are required to handle the DBMS.

**(v) Database Failures**

- In most of the organizations, all data is integrated into a single database.
- If database is corrupted due to power failure or it is corrupted on the storage media, our valuable data may be lost or the whole system stops.

**1.5 VIEWS OF DATA**

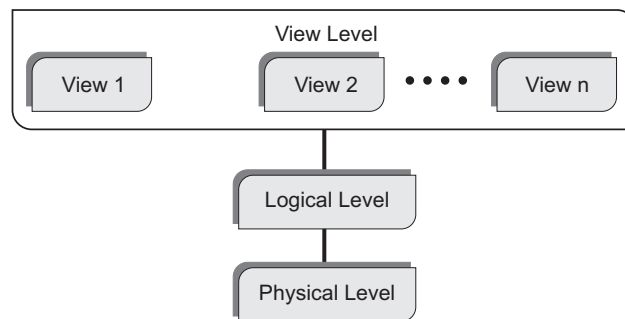
A major purpose of a database system is to provide users with an abstract view of the data. That is, the system hides certain details of how the data are stored and maintained.

**1.5.1 Data Abstraction**

The main objective of DBMS is to store and retrieve information efficiently; all the users should be able to access required data. The designers use complex data structure to represent the data, so that data can be efficiently stored and retrieved, but it is not necessary for the users to know physical database storage details because all database users are not computer trained. The developers hide the complexity from users through several levels of abstraction. The data abstraction helps the users easily interact with the database system. There are three levels of data abstraction:

- (i) Physical level:** It is the lowest level of abstraction that describes how the data are actually stored. The physical level describes complex low-level data structures in details.
- (ii) Logical level:** It is the next higher level of abstraction that describes what data are stored in the database and what relationships exist among those data.
- (iii) View level:** It is the highest level of abstraction that describes only part of the entire database.





**Fig. 1.3.** *The three levels of data abstraction.*

### 1.5.2 Data Independence

The ability to modify a scheme definition in one level without affecting a scheme definition in the next higher level is called data independence.

There are two levels of data independence:

1. **Physical data independence** is the ability to modify the physical scheme without causing application programs to be rewritten. Modifications at the physical level are occasionally necessary in order to improve performance.
2. **Logical data independence** is the ability to modify the conceptual scheme without causing application programs to be rewritten. Modifications at the conceptual level are necessary whenever the logical structure of the database is altered.

Logical data independence is more difficult to achieve than physical data independence since application programs are heavily dependent on the logical structure of the data they access.

### 1.5.3 Instances and Schemas

Database change over time as information is inserted and deleted. The collection of information stored in the database at a particular moment is called an instance of the database. The overall design of the database is called the database schema.

#### Types of database schemas

- (i) **Physical schema:** It describes the database design at the physical level.
- (ii) **Logical schema:** It describes the database design at the logical level.
- (iii) **Subschema:** A database may also have several sub schemas at the view level called as sub schemas that describe different views of the database.

## 1.6 DATABASE SYSTEM STRUCTURE

A database system is partitioned into modules that deal with each of the responsibilities of the overall system. The functional components of a database system can be broadly divided into

- Storage Manager
- Query Processor

## Database System Structure

### (i) Storage Manager

- 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 interaction with the file manager.
- The storage manager translates the various DML statements into low-level file system commands. Thus, the storage manager is responsible for storing, retrieving, and updating data in the database.

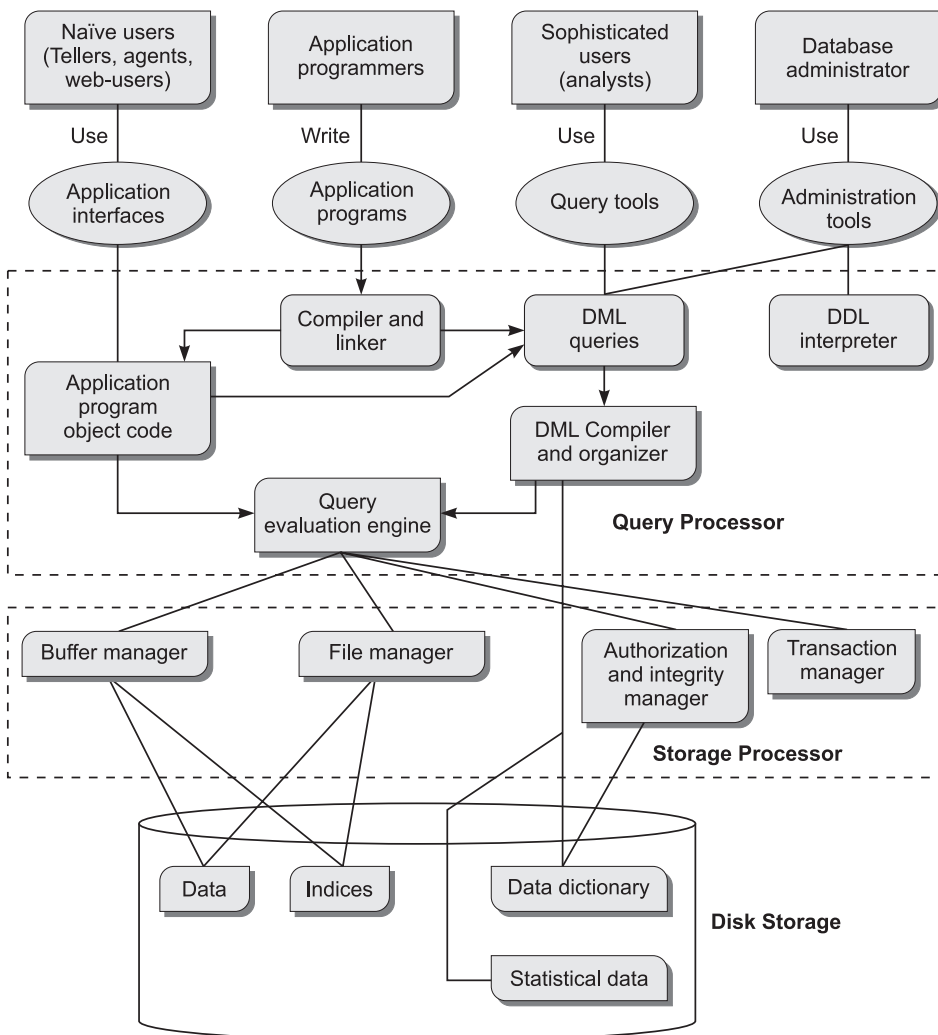


Fig. 1.4. Database system architecture.

### Components of the storage manager are:

- **Authorization and integrity manager:** It tests for satisfaction of various integrity constraints and checks the authority of users accessing the data.
- **Transaction manager:** It ensures that the database remains in a consistent state despite system failures, and concurrent executions proceed without conflicting.
- **File manager:** It manages the allocation of space on disk storage and the data structures used to represent information stored on disk.
- **Buffer manager:** It is responsible for fetching data from disk storage into main memory and deciding what data to cache in main memory.

The storage manager implements several data structures as part of physical system implementation.

- **Data files:** Which store the database itself
- **Data dictionary:** It contains metadata that is data about data. The schema of a table is an example of metadata. A database system consults the data dictionary before reading and modifying actual data.
- **Indices:** Which provide fast access to data items that hold particular values.

### (ii) The Query Processor

The query processor is an important part of the database system. It helps the database system to simplify and facilitate access to data.

The query processor components include:

- **DDL interpreter,** which interprets DDL statements and records the definitions in the data dictionary.
- **DML compiler,** which translates DML statements in a query language into an evaluation plan consisting of low-level instructions that the query evaluation engine understands.

A query can be translated into any number of evaluation plans that give the same result. The DML compiler also performs query optimization, that is, it picks up the lowest cost evaluation plan from among the alternatives.

**Query evaluation engine,** which executes low-level instructions generated by the DML compiler.

## 1.6.1 Database Users and Administrators

People who work with a database can be categorized as:

- Database users
- Database administrators

### 1.6.1.1 Database Users

#### 1. Naïve Users

- Naïve users who interact with the system by invoking one of the application programs that have been written previously.
- Naïve users use a form interface, where the user can fill in appropriate fields of the form. Naïve users may also simply read reports generated from the database.

## 2. Application Programmers

- Application programmers are computer professionals who write application programs.
- Rapid application development (RAD) tools are tools that enable an application programmer to construct forms and reports without writing a program.
- Special types of programming languages that combine control structures with data manipulation language. These languages are sometimes called as fourth-generation languages.

## 3. Sophisticated Users

- Sophisticated users interact with the system without writing programs. Instead, they form their requests in a database query language. They submit each such query to a query processor that the storage manager understands.
- Online analytical processing (OLAP) tools simplify analyst's tasks.
- Another tool for analysts is data mining tools, which help them to find certain kinds of patterns in data.

## 4. Specialized Users

- Specialized users are sophisticated users who write specialized database applications that do not fit into the traditional data-processing framework.
- The applications are computer-aided design systems, knowledge base and expert systems, systems that store data with complex data types

### 1.6.1.2 Database Administrators

A person who has such central control over the system is called a database administrator (DBA).

#### The functions of a DBA

- **Schema definition:** The DBA creates the original database schema by executing a set of data definition statements in the DDL.
- **Storage structure and access-method definition.**
- **Schema and physical-organization modification:** The DBA carries out changes to the schema and physical organization to reflect the changing needs of the organization.
- **Granting of authorization for data access:** By granting different types of authorization, the database administrator can regulate which parts of the database various users can access. The authorization information is kept in a special system structure that the database system consults whenever someone attempts to access the data in the system.
- **Routine maintenance:** Examples of the database administrator's routine maintenance activities are:
  1. Periodically backing up the database
  2. Ensuring that enough free disk space
  3. Monitoring jobs running on the database and ensuring that performance is not degraded by very expensive tasks submitted by some users.

## 1.6.2 Application Architectures

Most users of a database system today are not present at the site of the database system, but connect to it through a network. We can therefore differentiate between **client** machines, on which remote database users work, and server machines, on which the database system runs.

Database applications are usually partitioned into two or three parts, and according to the partition they are named as:

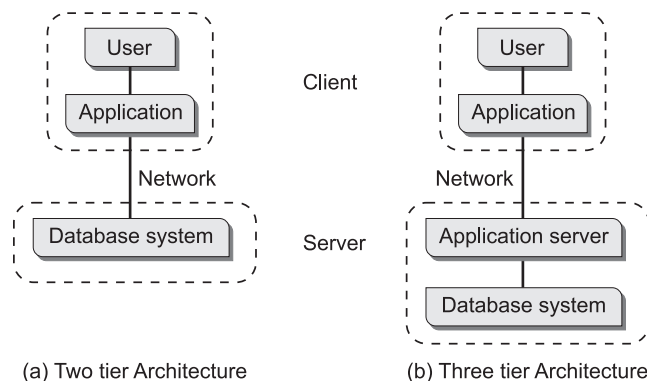
1. Two-tier architecture
2. Three-tier architecture

### 1.6.2.1 Two-tier Architecture

In this architecture, the application is partitioned into a component that resides at the client machine, which invokes database system functionality at the server machine through query language statements. Application program interface standards like ODBC and JDBC are used for interaction between the client and the server.

### 1.6.2.2 Three-tier Architecture

In this architecture, the client machine acts as front end and does not contain any direct database calls. Instead, the client end communicates with an application server, usually through a forms interface. The application server in turn communicates with a database system to access data. Three-tier applications are more appropriate for large applications, and for applications that run on the World Wide Web.



**Fig. 1.5.** Application architecture.

## REVIEW QUESTIONS

1. Define
  - (a) Data
  - (b) Information
  - (c) Database
  - (d) DBMS
  - (e) Database System
2. List out the applications of database.

3. Explain the drawbacks of conventional file processing system over DBMS.
4. Explain the merits and demerits of DBMS.
5. Define data abstraction. How is data abstraction implemented in database system?
6. With a neat sketch explain the different components of Database management system and its functionality.
7. Explain application architecture in details.
8. List out the roles of database users and administrators.

