UNIT-1

Introduction to Database, Applications of Database, Purpose of Database, View of Data, Data Independence, Data Models, Users of Database, DBA, Query Processor, Storage Manager, Database Architecture.

Introduction to Database

The database is a collection of inter-related data which is used to retrieve, insert and delete the data efficiently. It is also used to organize the data in the form of a table, schema, views, and reports, etc.

For example: The college Database organizes the data about the admin, staff, students and faculty etc.

Using the database, you can easily retrieve, insert, and delete the information.

Database Management System:

- Database management system is a software which is used to manage the database. For example: MySQL, Oracle, etc are a very popular commercial database which is used in different applications.
- DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, creating a table in the database and a lot more.
- It provides protection and security to the database. In the case of multiple users, it also maintains data consistency.

Characteristics of DBMS

- It uses a digital repository established on a server to store and manage the information.
- It can provide a clear and logical view of the process that manipulates data.
- DBMS contains automatic backup and recovery procedures.
- It contains ACID properties which maintain data in a healthy state in case of failure.
- It can reduce the complex relationship between data.
- It is used to support manipulation and processing of data.
- It is used to provide security of data.

- It can view the database from different viewpoints according to the requirements of the user.

Purpose of Database

The purpose of database systems is to make the database user-friendly and do easy operations. Users can easily insert, update, and delete. Actually, the main purpose is to have more control of the data.

The purpose of database systems is to manage the following insecurities:

- data redundancy and inconsistency,
- difficulty in accessing data,
- data isolation,
- atomicity of updates,
- concurrent access,
- security problems, and
- supports multiple views of data.

Avoid data redundancy and inconsistency:

If there are multiple copies of the same data, it just avoids it. It just maintains data in a single repository. Also, the purpose of database systems is to make the database consistent.

Difficulty in accessing data:

A database system can easily manage to access data. Through different queries, it can access data from the database.

Data isolation:

Data are isolated in several fields in the same database.

Atomicity of updates:

In case of power failure, the database might lose data. So, this feature will automatically prevent data loss.

Concurrent access:

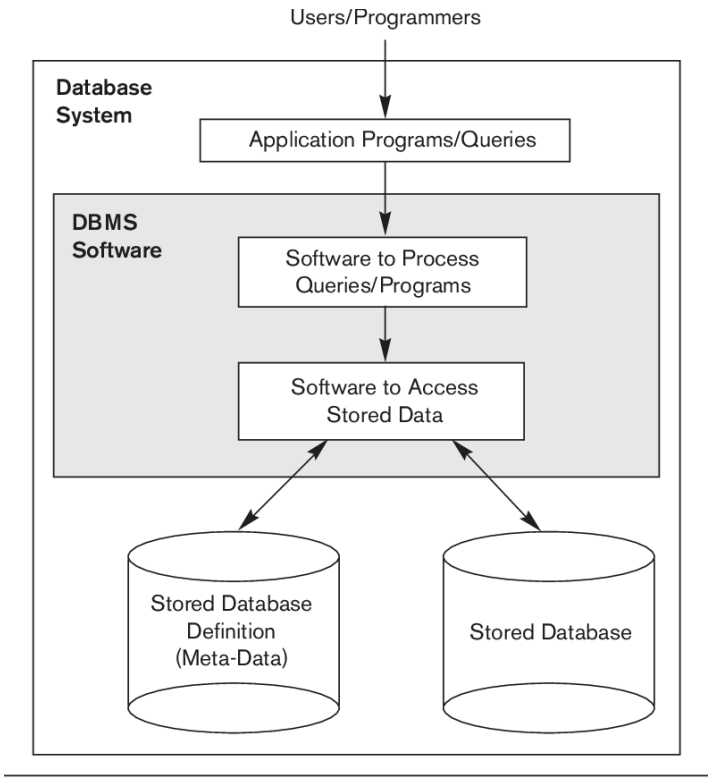
Users can have multiple access to the database at the same time.

Security problems:

Database systems will make the restricted access. So, the data will not be vulnerable.

Supports multiple views of data:

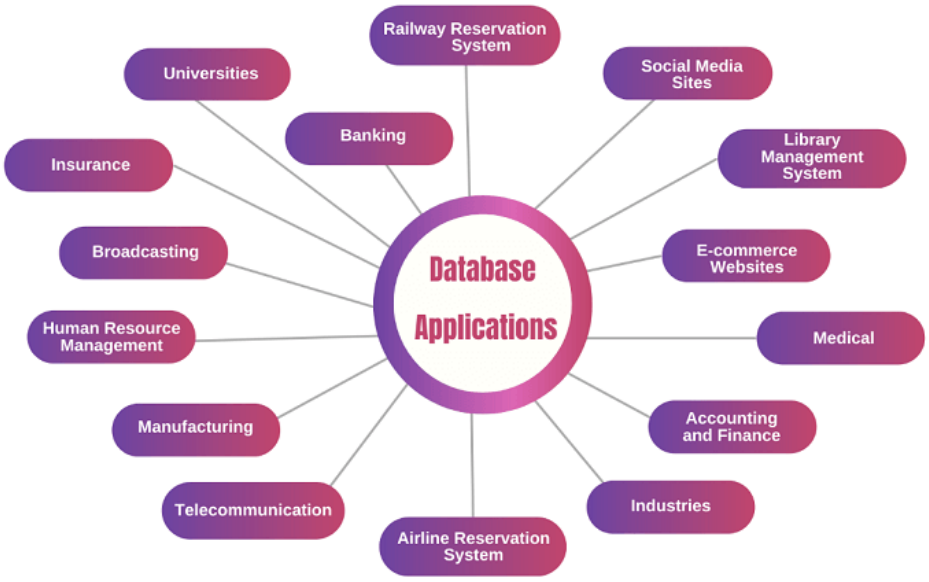
It can support multiple views of data to give the required view as their needs. Only database admins can have a complete view of the database. We cannot allow the end-users to have a view of developers.



Advantages of DBMS

- **Controls database redundancy:** It can control data redundancy because it stores all the data in a single database file and that recorded data is placed in the database.
- **Data sharing:** In DBMS, the authorized users of an organization can share the data among multiple users.
- **Easily Maintenance:** It can be easily maintainable due to the centralized nature of the database system.
- **Reduce time:** It reduces development time and maintenance need.
- **Backup:** It provides backup and recovery subsystems which create automatic backup of data from hardware and software failures and restores the data if required.
- **multiple user interface:** It provides different types of user interfaces like graphical user interfaces, application program interfaces.

Applications of Data Base



1. Universities:

Universities have so much data which can be stored in the database, such as student information, teacher information, non-teaching staff information, course information, section information, grade report information, and many more. University information is kept safe and secure in the database.

Anyone who needs information about the student, teacher, or course can easily retrieve it from the database. Everything needs to be maintained because even after ten years, information may be required, and the information may be useful, so maintaining complete information is the primary responsibility of any university or educational institution.

2. Banking:

Banks have a huge amount of data as millions of people have accounts that need to be maintained properly. The database keeps the record of each user in a systematic manner. Banking databases store a lot of information about account holders. It stores customer details, asset details, banking

transactions, balance sheets, credit card and debit card details, loans, fixed deposits, and much more. Everything is maintained with the help of a database.

3. Railway Reservation System:

It is an inevitable area of application of databases. They store information such as passenger name, mobile number, booking status, reservation details, train schedule, employee information, account details, seating arrangement, route & alternate route details, etc. All the information needs to be maintained, so railways use a database management system for their efficient storage and retrieval purpose.

4. Social Media Sites:

Nowadays, everyone has a smartphone and accounts on various social media sites like Facebook, LinkedIn, Pinterest, Twitter, Instagram, etc. People can chat with their friends and family and make new friends from all over the world. Social media has millions of accounts, which means they have a huge amount of data that needs to be stored and maintained. Social media sites use databases to store information about users, images, videos, chats, etc.

5. Library Management System:

There are hundreds and thousands of books in the library, so it is not easy to maintain the records of the books in a register or diary, so a database management system is used which maintains the information of the library efficiently. The library database stores information like book name, issue date, author name, book availability, book issuer name, book return details, etc.

6. E-commerce Websites:

E-commerce websites are one of the prominent applications of the database. Websites such as Flipkart, Myntra, Amazon, Nykaa, Snapdeal, Shopify, and many more, are online shopping websites where people buy items online. These websites have so much data. These websites use databases to securely store and maintain customer details, product details, dealer details, purchase details, bank & card details, transactions details, invoice details, etc. You can analyze the sales and maintain the inventory with the help of a database.

7. Medical:

There is a lot of important data collection in the medical field, so it is necessary to use the database to store data related to the medical field, such as patient details, medicine details, practitioner details, surgeon details, appointment details, doctor schedule, patient discharge details, payment

detail, invoices, and other medical records. The database management system is a boon for the medical field because it helps doctors to monitor their patients and provide better care.

8. Accounting and Finance:

When there is big data regarding accounting and finance, there is a need to maintain a large amount of data, which is done with the help of a database. The database stores data such as accounting details, bank details, purchases of stocks, invoice details, sales records, asset details, etc. Accounting and finance database helps in maintaining and analyzing historical data.

9. Industries:

The database management system is the main priority of industries because they need to store huge amounts of data. The industry database stores customer details, sales records, product lists, transactions, etc. All the information is kept secure and maintained by the database.

10. Airline Reservation System:

It is one of the applications of database management systems that contain data such as passenger name, passenger check-in, passenger departure, flight schedule, number of flights, distance from source to destination, reservation information, pilot details, accounting detail, route detail, etc. The database provides maintenance and security to airline data.

11. Telecommunication:

We cannot deny that telecommunication has brought a remarkable revolution worldwide. The Telecom field has huge data, and it is very difficult to manage big data without a database; that is why a telecom database is required, which stores data such as customer names, phone numbers, calling details, prepaid & post-paid connection records, network usage, bill details, balance details, etc.

12. Manufacturing:

In the manufacturing field, a lot of data needs to be maintained regarding supply chain management, so the database maintains the data such as product details, customer information, order details, purchase details, payment info, worker's details, invoice, etc. Manufacturing companies produce and supply products every day, so it is important to use a database.

13. Human Resource Management:

Any organization will definitely have employees, and if there are a large number of employees, then it becomes essential to store data in a database as it maintains and securely saves the data, which can be retrieved and accessed when required. The human resource database stores data such

as employee name, joining details, designation, salary details, tax information, benefits & goodies details, etc.

14. Broadcasting:

Broadcasting is distributing video and audio content to a dispersed audience by television, radio, or other means. Broadcasting database stores data such as subscriber information, event recordings, event schedules, etc., so it becomes important to store broadcasting data in the database.

15. Insurance:

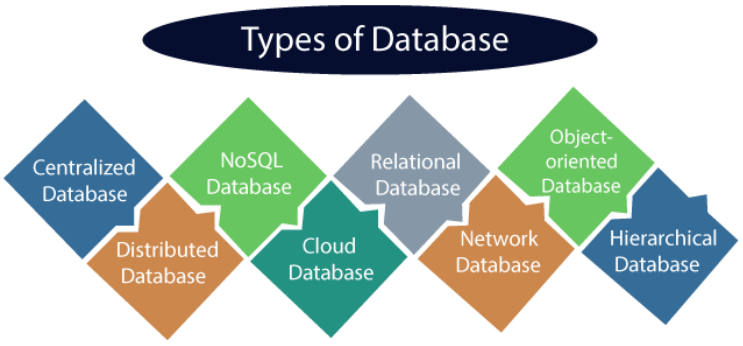
An insurance company needs a database to store large amounts of data. Insurance database stores data such as policy details, user details, buyer details, payment details, nominee details, address details, etc.

Disadvantages of DBMS

- **Cost of Hardware and Software:** It requires a high speed of data processor and large memory size to run DBMS software.
- **Size:** It occupies a large space of disks and large memory to run them efficiently.
- **Complexity:** Database system creates additional complexity and requirements.
- **Higher impact of failure:** Failure is highly impacted the database because in most of the organization, all the data stored in a single database and if the database is damaged due to electric failure or database corruption then the data may be lost forever.

Types of Data Base

Here are various types of databases used for storing different varieties of data:



Relational Database

This database is based on the relational data model, which stores data in the form of rows(tuple) and columns(attributes), and together forms a table(relation). A relational database uses SQL for storing, manipulating, as well as maintaining the data. E.F. Codd invented the database in 1970. Each table in the database carries a key that makes the data unique from others.

Examples : MySQL, Microsoft SQL Server, Oracle, etc.

Properties of Relational Database

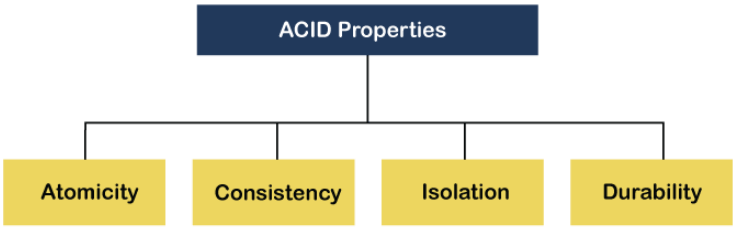
There are following four commonly known properties of a relational model known as **ACID properties**

ACID Properties:

DBMS is the management of data that should remain integrated when any changes are done in it. It is because if the integrity of the data is affected, whole data will get disturbed and corrupted. Therefore, to maintain the integrity of the data, there are four properties described in the database management system, which are known as the **ACID** properties. The ACID properties are meant for the transaction that goes through a different group of tasks, and there we come to see the role of the ACID properties.

ACID Properties

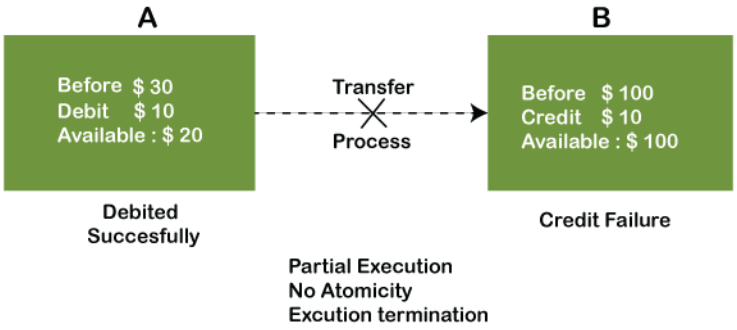
The expansion of the term ACID defines for:



1) Atomicity

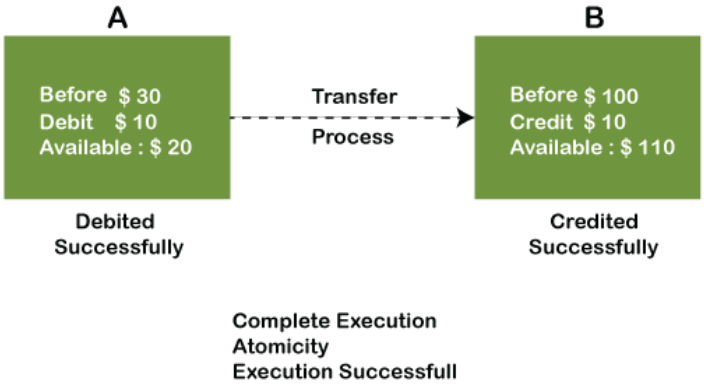
The term atomicity defines that the data remains atomic. It means if any operation is performed on the data, either it should be performed or executed completely or should not be executed at all. It further means that the operation should not break in between or execute partially. In the case of executing operations on the transaction, the operation should be completely executed and not partially.

Example: If Remo has account A having $30 in his account from which he wishes to send $10 to Sheero's account, which is B. In account B, a sum of $ 100 is already present. When $10 will be transferred to account B, the sum will become $110. Now, there will be two operations that will take place. One is the amount of $10 that Remo wants to transfer will be debited from his account A, and the same amount will get credited to account B, i.e., into Sheero's account. Now, what happens - the first operation of debit executes successfully, but the credit operation, however, fails. Thus, in Remo's account A, the value becomes $20, and to that of Sheero's account, it remains $100 as it was previously present.



In the above diagram, it can be seen that after crediting $10, the amount is still $100 in account B. So, it is not an atomic transaction.

The below image shows that both debit and credit operations are done successfully. Thus the transaction is atomic.

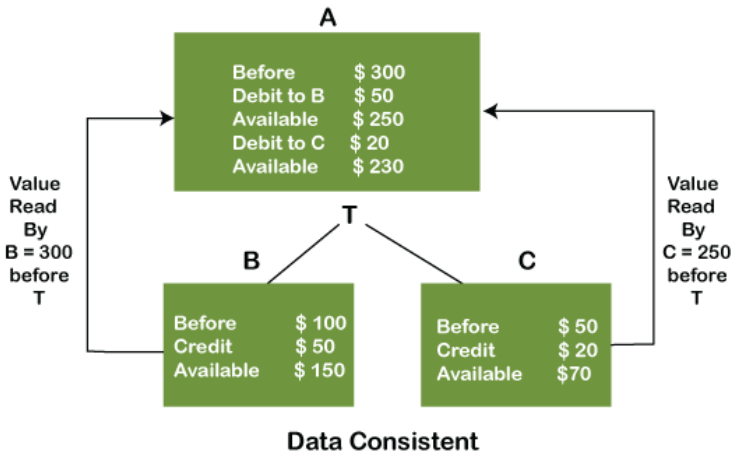


Thus, when the amount loses atomicity, then in the bank systems, this becomes a huge issue, and so the atomicity is the main focus in the bank systems.

2) Consistency

The word **consistency** means that the value should remain preserved always. In DBMS, the integrity of the data should be maintained, which means if a change in the database is made, it should remain preserved always. In the case of transactions, the integrity of the data is very essential so that the database remains consistent before and after the transaction. The data should always be correct.

Example:



Data Consistent

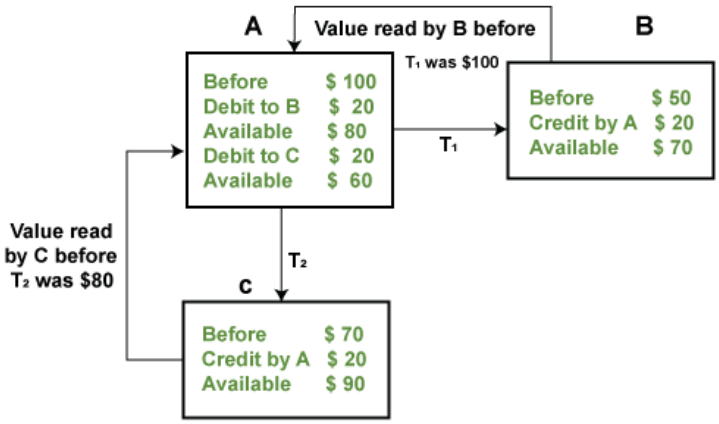
In the above figure, there are three accounts, A, B, and C, where A is making a transaction T one by one to both B & C. There are two operations that take place, i.e., Debit and Credit. Account A firstly debits $50 to account B, and the amount in account A is read $300 by B before the transaction. After the successful transaction T, the available amount in B becomes $150. Now, A debits $20 to account C, and that time, the value read by C is $250 (that is correct as a debit of $50 has been successfully done to B). The debit and credit operation from account A to C has been done successfully. We can see that the transaction is done successfully, and the value is also read correctly. Thus, the data is consistent. In case the value read by B and C is $300, which means that data is inconsistent because when the debit operation executes, it will not be consistent.

3) Isolation

The term 'isolation' means separation. In DBMS, Isolation is the property of a database where no data should affect the other one and may occur concurrently. In short, the operation on one database should begin when the operation on the first database gets complete. It means if two operations are being performed on two different databases, they may not affect the value of one another. In the case of transactions, when two or more transactions occur simultaneously, the consistency

should remain maintained. Any changes that occur in any particular transaction will not be seen by other transactions until the change is not committed in the memory.

Example: If two operations are concurrently running on two different accounts, then the value of both accounts should not get affected. The value should remain persistent. As you can see in the below diagram, account A is making T1 and T2 transactions to account B and C, but both are executing independently without affecting each other. It is known as Isolation.



Isolation - Independent execution of T_1 & T_2 by A

4) Durability

Durability ensures the permanency of something. In DBMS, the term durability ensures that the data after the successful execution of the operation becomes permanent in the database. The durability of the data should be so perfect that even if the system fails or leads to a crash, the database still survives. However, if gets lost, it becomes the responsibility of the recovery manager for ensuring the durability of the database. For committing the values, the COMMIT command must be used every time we make changes.

RDBMS (Relational Database Management System):

RDBMS stands for *Relational Database Management System.*

All modern database management systems like SQL, MS SQL Server, IBM DB2, ORACLE, My-SQL, and Microsoft Access are based on RDBMS.

It is called Relational Database Management System (RDBMS) because it is based on the relational model introduced by E.F. Codd.

How it works?

Data is represented in terms of tuples (rows) in RDBMS.

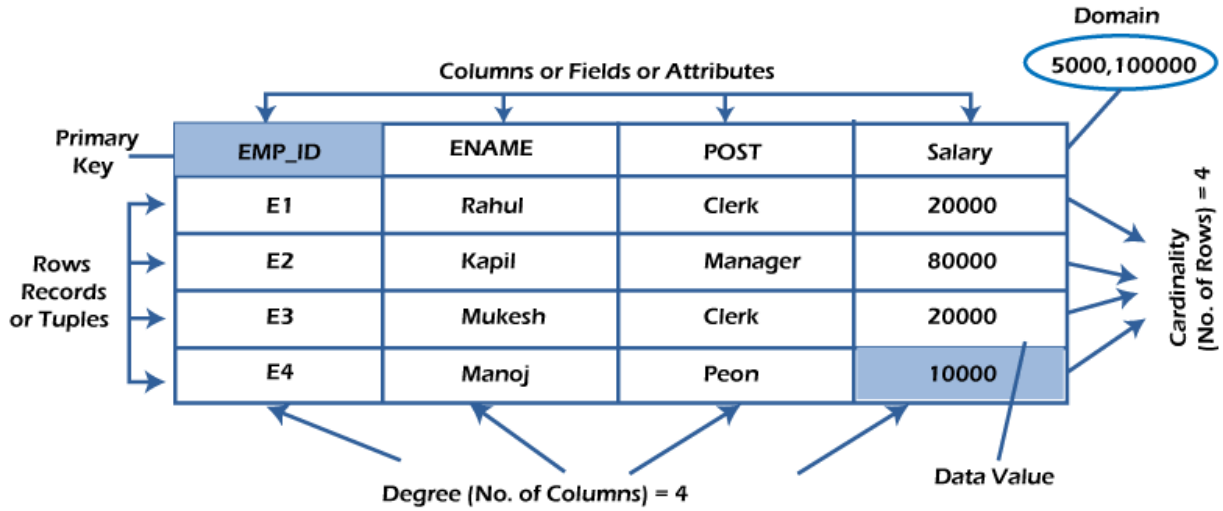
A relational database is the most commonly used database. It contains several tables, and each table has its primary key.

Due to a collection of an organized set of tables, data can be accessed easily in RDBMS.

From 1970 to 1972, E.F. Codd published a paper to propose using a relational database model.

RDBMS is originally based on E.F. Codd's relational model invention.

Following are the various terminologies of RDBMS:



Difference between DBMS and RDBMS

No.	DBMS	RDBMS
1)	DBMS applications store **data as file**.	RDBMS applications store **data in a tabular form**.
2)	In DBMS, data is generally stored in either a hierarchical form or a navigational form.	In RDBMS, the tables have an identifier called primary key and the data values are stored in the form of tables.
3)	**Normalization is not** present in DBMS.	**Normalization is** present in RDBMS.

4)	DBMS does **not apply any security** with regards to data manipulation.	RDBMS **defines the integrity constraint** for the purpose of ACID (Atomocity, Consistency, Isolation and Durability) property.
5)	DBMS uses file system to store data, so there will be **no relation between the tables**.	in RDBMS, data values are stored in the form of tables, so a **relationship** between these data values will be stored in the form of a table as well.
6)	DBMS has to provide some uniform methods to access the stored information.	RDBMS system supports a tabular structure of the data and a relationship between them to access the stored information.
7)	DBMS **does not support distributed database**.	RDBMS **supports distributed database**.
8)	DBMS is meant to be for small organization and **deal with small data**. it supports **single user**.	RDBMS is designed to **handle large amount of data**. it supports **multiple users**.
9)	Examples of DBMS are file systems, **xml** etc.	Example of RDBMS are **mysql**, **postgre**, **sql server**, **oracle** etc.

After observing the differences between DBMS and RDBMS, you can say that RDBMS is an extension of DBMS. There are many software products in the market today who are compatible for both DBMS and RDBMS. Means today a RDBMS application is DBMS application and vice-versa.

Difference between File System and DBMS

Basis	DBMS Approach	File System Approach

Meaning	DBMS is a collection of data. In DBMS, the user is not required to write the procedures.	The file system is a collection of data. In this system, the user has to write the procedures for managing the database.
Sharing of data	Due to the centralized approach, data sharing is easy.	Data is distributed in many files, and it may be of different formats, so it isn't easy to share data.
Data Abstraction	DBMS gives an abstract view of data that hides the details.	The file system provides the detail of the data representation and storage of data.
Security and Protection	DBMS provides a good protection mechanism.	It isn't easy to protect a file under the file system.
Recovery Mechanism	DBMS provides a crash recovery mechanism, i.e., DBMS protects the user from system failure.	The file system doesn't have a crash mechanism, i.e., if the system crashes while entering some data, then the content of the file will be lost.
Manipulation Techniques	DBMS contains a wide variety of sophisticated techniques to store and retrieve the data.	The file system can't efficiently store and retrieve the data.
Concurrency Problems	DBMS takes care of Concurrent access of data using some form of locking.	In the File system, concurrent access has many problems like redirecting the file while deleting some information or updating some information.

Where to use	Database approach used in large systems which interrelate many files.	File system approach used in large systems which interrelate many files.
Cost	The database system is expensive to design.	The file system approach is cheaper to design.
Data Redundancy and Inconsistency	Due to the centralization of the database, the problems of data redundancy and inconsistency are controlled.	In this, the files and application programs are created by different programmers so that there exists a lot of duplication of data which may lead to inconsistency.
Structure	The database structure is complex to design.	The file system approach has a simple structure.
Data Independence	In this system, Data Independence exists, and it can be of two types. o Logical Data Independence o Physical Data Independence	In the File system approach, there exists no Data Independence.
Integrity Constraints	Integrity Constraints are easy to apply.	Integrity Constraints are difficult to implement in file system.
Data Models	In the database approach, 3 types of data models exist: o Hierarchal data models o Network data models o Relational data models	In the file system approach, there is no concept of data models exists.

Flexibility	Changes are often a necessity to the content of the data stored in any system, and these changes are more easily with a database approach.	The flexibility of the system is less as compared to the DBMS approach.
Examples	Oracle, SQL Server, Sybase etc.	Cobol, C++ etc.

View of Data

View of data in DBMS narrate how the data is visualized at each level of data abstraction?

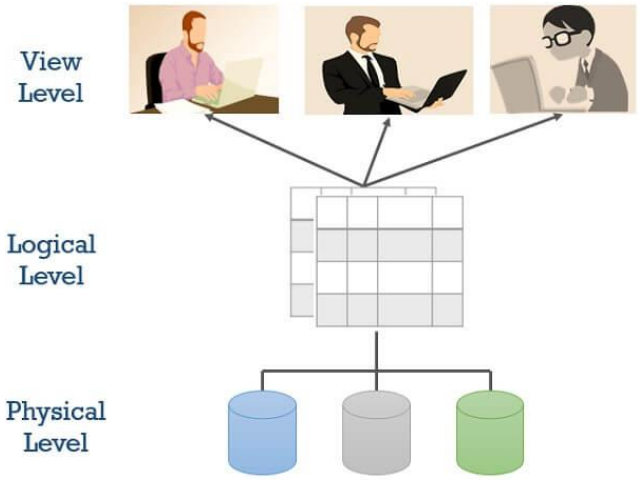
Data Abstraction:

Data abstraction is **hiding the complex data structure** in order to **simplify the user's interface** of the system. It is done because many of the users interacting with the database system are not that much computer trained to understand the complex data structures of the database system.

To achieve data abstraction, we will discuss a **Three-Schema architecture** which abstracts the database at three levels discussed below:

Three-Schema Architecture:

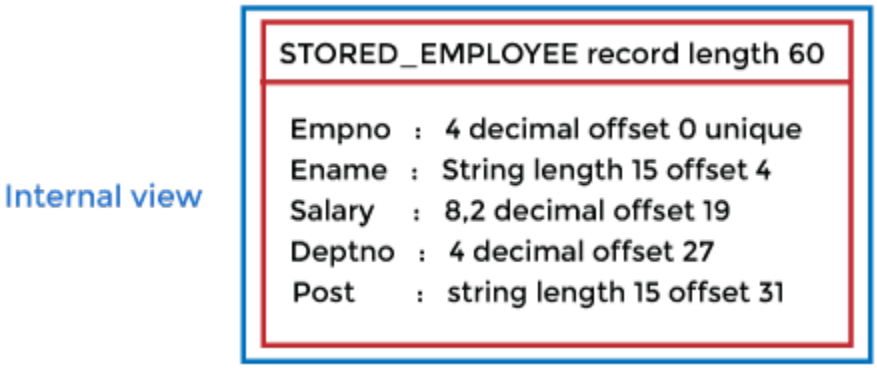
- This framework is used to describe the structure of a specific database system.
- The three schema architecture is also used to separate the user applications and physical database.
- The three schema architecture contains three-levels. It breaks the database down into three different categories.



Three-Schema Architecture

The main objective of three level architecture is to enable multiple users to access the same data with a personalized view while storing the underlying data only once. Thus it separates the user's view from the physical structure of the database.

1. Internal Level



- The internal level has an internal schema which describes the physical storage structure of the database.
- The internal schema is also known as a physical schema.
- It uses the physical data model. It is used to define that how the data will be stored in a block.
- The physical level is used to describe complex low-level data structures in detail.

The internal level is generally is concerned with the following activities:

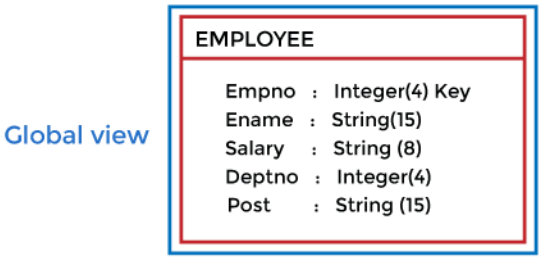
- Storage space allocations.

  For Example: B-Trees, Hashing etc.

- Access paths.

  For Example: Specification of primary and secondary keys, indexes, pointers and sequencing.
- Data compression and encryption techniques.
- Optimization of internal structures.
- Representation of stored fields.

2. Conceptual Level



- The conceptual schema describes the design of a database at the conceptual level. Conceptual level is also known as logical level.
- The conceptual schema describes the structure of the whole database.
- The conceptual level describes what data are to be stored in the database and also describes what relationship exists among those data.
- In the conceptual level, internal details such as an implementation of the data structure are hidden.
- Programmers and database administrators work at this level.

3. External Level

- At the external level, a database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database.
- An external schema is also known as view schema.
- Each view schema describes the database part that a particular user group is interested and hides the remaining database from that user group.
- The view schema describes the end user interaction with database systems.

Data Independence

- Data independence can be explained using the three-schema architecture.
- Data independence refers characteristic of being able to modify the schema at one level of the database system without altering the schema at the next higher level.

There are two types of data independence:

1. Logical Data Independence

- Logical data independence refers characteristic of being able to change the conceptual schema without having to change the external schema.
- Logical data independence is used to separate the external level from the conceptual view.
- If we do any changes in the conceptual view of the data, then the user view of the data would not be affected.
- Logical data independence occurs at the user interface level.

2. Physical Data Independence

- Physical data independence can be defined as the capacity to change the internal schema without having to change the conceptual schema.
- If we do any changes in the storage size of the database system server, then the Conceptual structure of the database will not be affected.
- Physical data independence is used to separate conceptual levels from the internal levels.
- Physical data independence occurs at the logical interface level.

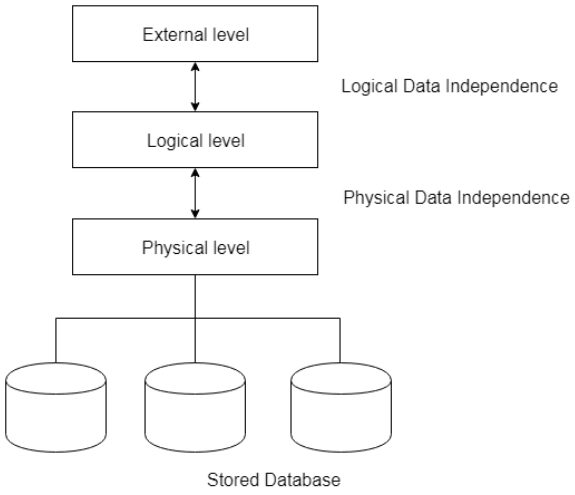
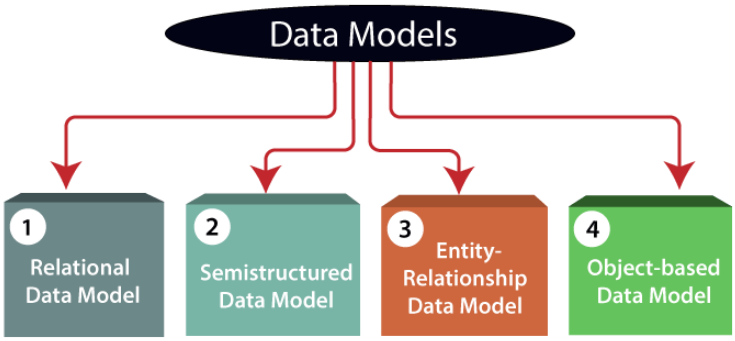


Fig: Data Independence

Data Models

Data Model is the modeling of the data description, data semantics, and consistency constraints of the data. It provides the conceptual tools for describing the design of a database at each level of data abstraction.



1) Relational Data Model: This type of model designs the data in the form of rows and columns within a table. Thus, a relational model uses tables for representing data and in-between relationships. Tables are also called relations. This model was initially described by Edgar F. Codd, in 1969. The relational data model is the widely used model which is primarily used by commercial data processing applications.

2) Entity-Relationship Data Model: An ER model is the logical representation of data as objects and relationships among them. These objects are known as entities, and relationship is an association among these entities. This model was designed by Peter Chen and published in 1976 papers. It was widely used in database designing. A set of attributes describe the entities. For example, student_name, student_id describes the 'student' entity. A set of the same type of entities is known as an 'Entity set', and the set of the same type of relationships is known as 'relationship set'.

3) Object-based Data Model: An extension of the ER model with notions of functions, encapsulation, and object identity, as well. This model supports a rich type system that includes structured and collection types. Thus, in 1980s, various database systems following the object-oriented approach were developed. Here, the objects are nothing but the data carrying its properties.

4) Semi-structured Data Model: This type of data model is different from the other three data models (explained above). The semi-structured data model allows the data specifications at places where the individual data items of the same type may have different attributes sets. The Extensible Markup Language, also known as XML, is widely used for representing the semi-structured data. Although XML was initially designed for including the markup information to the text document, it gains importance because of its application in the exchange of data.

Users of Database

Database users are categorized based up on their interaction with the database. These are seven types of database users in DBMS.

1. **Database Administrator (DBA) :** Database Administrator (DBA) is a person/team who defines the schema and also controls the 3 levels of database. The DBA will then create a new account id and password for the user if he/she need to access the database. DBA is also responsible for providing security to the database and he allows only the authorized users to access/modify the data base. DBA is responsible for the problems such as security breaches and poor system response time.
    - DBA also monitors the recovery and backup and provide technical support.
    - The DBA has a DBA account in the DBMS which called a system or superuser account.
    - DBA repairs damage caused due to hardware and/or software failures.
    - DBA is the one having privileges to perform DCL (Data Control Language) operations such as GRANT and REVOKE, to allow/restrict a particular user from accessing the database.
2. **Naive / Parametric End Users :** Parametric End Users are the unsophisticated who don't have any DBMS knowledge but they frequently use the database applications in their daily life to get the desired results. For examples, Railway's ticket booking users are naive users. Clerks in any bank is a naive user because they don't have any DBMS knowledge but they still use the database and perform their given task.
3. **System Analyst :** System Analyst is a user who analyzes the requirements of parametric end users. They check whether all the requirements of end users are satisfied.

4. **Sophisticated Users :** Sophisticated users can be engineers, scientists, business analyst, who are familiar with the database. They can develop their own database applications according to their requirement. They don't write the program code but they interact the database by writing SQL queries directly through the query processor.
5. **Database Designers :** Data Base Designers are the users who design the structure of database which includes tables, indexes, views, triggers, stored procedures and constraints which are usually enforced before the database is created or populated with data. He/she controls what data must be stored and how the data items to be related. It is responsibility of Database Designers to understand the requirements of different user groups and then create a design which satisfies the need of all the user groups.
6. **Application Programmers :** Application Programmers also referred as System Analysts or simply Software Engineers, are the back-end programmers who writes the code for the application programs. They are the computer professionals. These programs could be written in Programming languages such as Visual Basic, Developer, C, FORTRAN, COBOL etc. Application programmers design, debug, test, and maintain set of programs called "canned transactions" for the Naive (parametric) users in order to interact with database.
7. **Casual Users / Temporary Users :** Casual Users are the users who occasionally use/access the database but each time when they access the database they require the new information, for example, Middle or higher level manager.

DBA

A database administrator (DBA) is a person or group in charge of implementing **DBMS** in an organization. The DBA job requires a high degree of technical expertise. DBA consists of a team of people rather than just one person.

The primary role of Database administrator is as follows −

- Database design
- Performance issues
- Database accessibility
- Capacity issues
- Data replication
- Table Maintenance

Responsibilities of DBA:

The responsibilities of DBA are as follows −

- Makes the decision concerning the content of the database.
- Plans the storage structure and access strategy.
- Provides the support to the users.
- Defines the security and integrity checks.
- Interpreter backup and recovery strategies.
- Monitoring the performance and responding to the changes in the requirements.

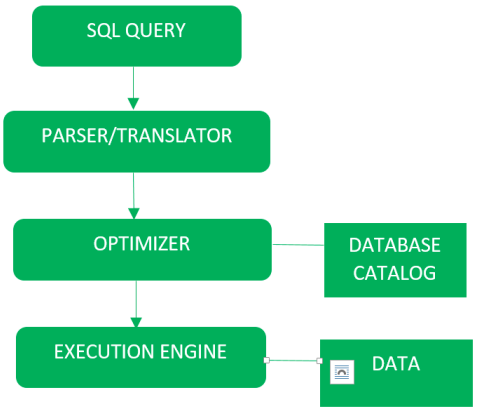
Skills required for DBA:

The skills required to be a successful DBA are as follows −

- Database designing.
- Knowledge of Structured Query Language (SQL).
- Know about distributed architecture.
- Knowledge on different operating servers.
- Idea on Relational Database Management System (RDBMS).
- Ready to face challenges and solve the problems quickly.

Query Processor

Query Processing includes translations on high level Queries into low level expressions that can be used at physical level of file system, query optimization and actual execution of query to get the actual result.



Step-1:

Parser: During parse call, the database performs the following checks- Syntax check, Semantic check and Shared pool check, after converting the query into relational algebra.

Parser performs the following checks as (refer detailed diagram):

1. **Syntax check –** concludes SQL syntactic validity. Example:

   SELECT * FORM employee;

   Here error of wrong spelling of FROM is given by this check.
2. **Semantic check –** determines whether the statement is meaningful or not. Example: query contains a tablename which does not exist is checked by this check.
3. **Shared Pool check –** Every query possess a hash code during its execution. So, this check determines existence of written hash code in shared pool if code exists in shared pool then database will not take additional steps for optimization and execution.

Hard Parse and Soft Parse

If there is a fresh query and its hash code does not exist in shared pool then that query has to pass through from the additional steps known as hard parsing otherwise if hash code exists then query does not passes through additional steps. It just passes directly to execution engine (refer detailed diagram). This is known as soft parsing.

Hard Parse includes following steps – Optimizer and Row source generation.

Step-2:

Optimizer: During optimization stage, database must perform a hard parse atleast for one unique DML statement and perform optimization during this parse. This database never optimizes DDL unless it includes a DML component such as subquery that require optimization.

It is a process in which multiple query execution plan for satisfying a query are examined and most efficient query plan is satisfied for execution.

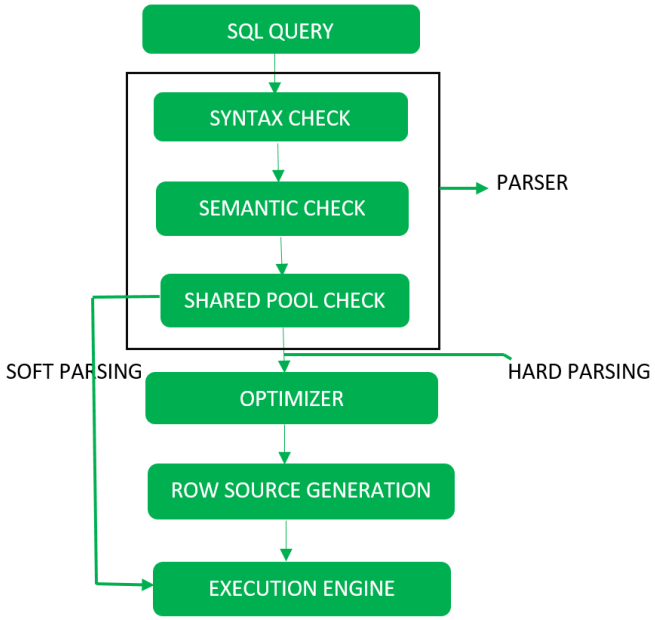
Database catalog stores the execution plans and then optimizer passes the lowest cost plan for execution.

Row Source Generation

The Row Source Generation is a software that receives a optimal execution plan from the optimizer and produces an iterative execution plan that is usable by the rest of the database. the iterative plan is the binary program that when executes by the sql engine produces the result set.

Step-3:

Execution Engine: Finally runs the query and display the required result.



Storage Manager

It serves as an interface between the data stored in the database and the application programs and queries submitted to the system.

It is responsible for:

- interaction with the file manager (OS)
- efficient storage, retrieval and updation of data

It has the following components:

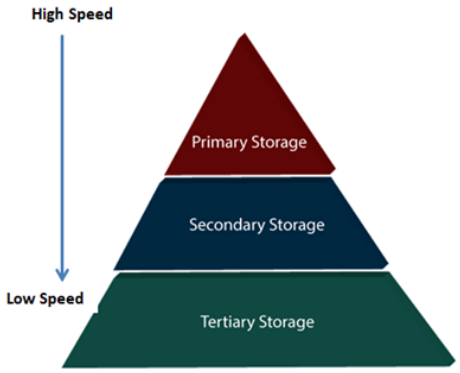
- **Authorization/Integrity Manager**: authenticates users and enforces integrity constraints
- **Transaction Manager**: ensures atomicity of transactions
- **File Manager**: keeps track of logical mapping between data and files
- **Buffer Manager**: uses data buffering policies to speed up data access

It is responsible for storing data, data dictionary (that contains information about all the entities in the database) and indices.

Types of Data Storage

For storing the data, there are different types of storage options available. These storage types differ from one another as per the speed and accessibility.

- Primary Storage
- Secondary Storage
- Tertiary Storage



Primary Storage

It is the primary area that offers quick access to the stored data. We also know the primary storage as volatile storage. It is because this type of memory does not permanently store the data. As soon as the system leads to a power cut or a crash, the data also get lost. Main memory and cache are the types of primary storage.

- **Main Memory:** It is the one that is responsible for operating the data that is available by the storage medium. The main memory handles each instruction of a computer machine. This type of memory can store gigabytes of data on a system but is small enough to carry the entire database. At last, the main memory loses the whole content if the system shuts down because of power failure or other reasons.
- **Cache:** It is one of the costly storage media. On the other hand, it is the fastest one. A cache is a tiny storage media which is maintained by the computer hardware usually. While designing the algorithms and query processors for the data structures, the designers keep concern on the cache effects.

Secondary Storage

Secondary storage is also called as Online storage. It is the storage area that allows the user to save and store data permanently. This type of memory does not lose the data due to any power failure or system crash. That's why we also call it non-volatile storage.

There are some commonly described secondary storage media which are available in almost every type of computer system:

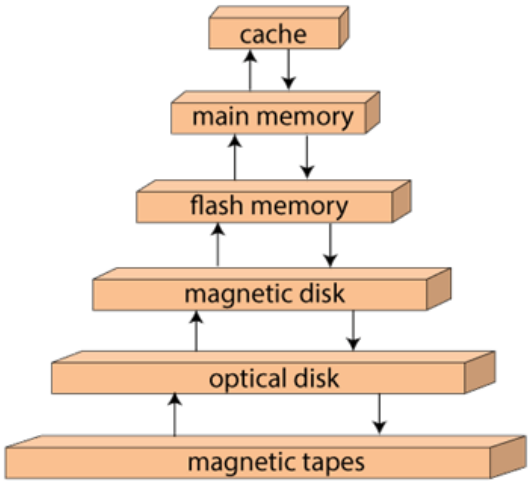
- **Flash Memory:** A flash memory stores data in USB (Universal Serial Bus) keys which are further plugged into the USB slots of a computer system. These USB keys help transfer data to a computer system, but it varies in size limits. Unlike the main memory, it is possible to get back the stored data which may be lost due to a power cut or other reasons. This type of memory storage is most commonly used in the server systems for caching the frequently used data. This leads the systems towards high performance and is capable of storing large amounts of databases than the main memory.
- **Magnetic Disk Storage:** This type of storage media is also known as online storage media. A magnetic disk is used for storing the data for a long time. It is capable of storing an entire database. It is the responsibility of the computer system to make availability of the data from a disk to the main memory for further accessing. Also, if the system performs any operation over the data, the modified data should be written back to the disk. The tremendous capability of a magnetic disk is that it does not affect the data due to a system crash or failure, but a disk failure can easily ruin as well as destroy the stored data.

Tertiary Storage

It is the storage type that is external from the computer system. It has the slowest speed. But it is capable of storing a large amount of data. It is also known as Offline storage. Tertiary storage is generally used for data backup. There are following tertiary storage devices available:

- **Optical Storage:** An optical storage can store megabytes or gigabytes of data. A Compact Disk (CD) can store 700 megabytes of data with a playtime of around 80 minutes. On the other hand, a Digital Video Disk or a DVD can store 4.7 or 8.5 gigabytes of data on each side of the disk.
- **Tape Storage:** It is the cheapest storage medium than disks. Generally, tapes are used for archiving or backing up the data. It provides slow access to data as it accesses data sequentially from the start. Thus, tape storage is also known as sequential-access storage. Disk storage is known as direct-access storage as we can directly access the data from any location on disk.

Storage Hierarchy

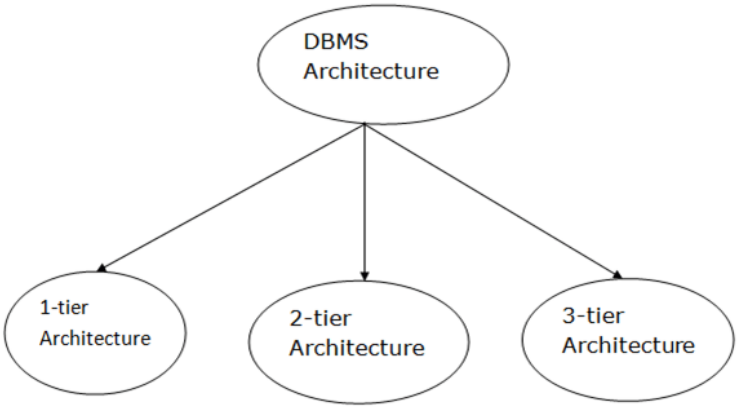


Storage device hierarchy

DBMS Architecture

- The DBMS design depends upon its architecture. The basic client/server architecture is used to deal with a large number of PCs, web servers, database servers and other components that are connected with networks.
- The client/server architecture consists of many PCs and a workstation which are connected via the network.
- DBMS architecture depends upon how users are connected to the database to get their request done.

Types of DBMS Architecture



Database architecture can be seen as a single tier or multi-tier. But logically, database architecture is of two types like: **2-tier architecture** and **3-tier architecture**.

1-Tier Architecture

- In this architecture, the database is directly available to the user. It means the user can directly sit on the DBMS and uses it.
- Any changes done here will directly be done on the database itself. It doesn't provide a handy tool for end users.
- The 1-Tier architecture is used for development of the local application, where programmers can directly communicate with the database for the quick response.

2-Tier Architecture

- The 2-Tier architecture is same as basic client-server. In the two-tier architecture, applications on the client end can directly communicate with the database at the server side. For this interaction, API's like: **ODBC**, **JDBC** are used.
- The user interfaces and application programs are run on the client-side.
- The server side is responsible to provide the functionalities like: query processing and transaction management.
- To communicate with the DBMS, client-side application establishes a connection with the server side.

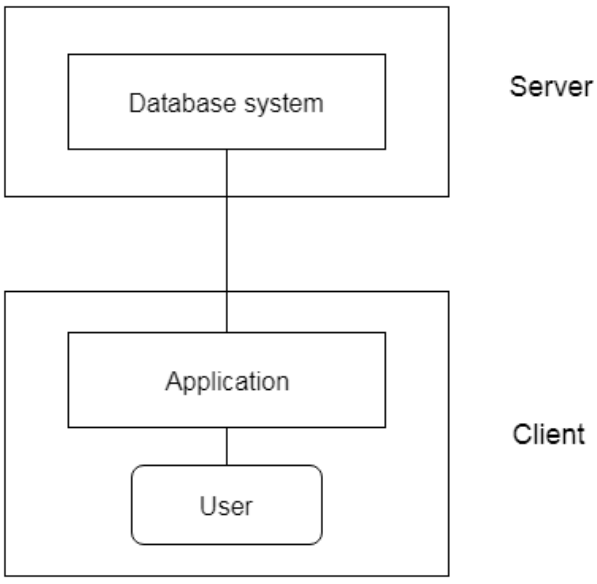


Fig: 2-tier Architecture

3-Tier Architecture

- The 3-Tier architecture contains another layer between the client and server. In this architecture, client can't directly communicate with the server.
- The application on the client-end interacts with an application server which further communicates with the database system.
- End user has no idea about the existence of the database beyond the application server. The database also has no idea about any other user beyond the application.
- The 3-Tier architecture is used in case of large web application.

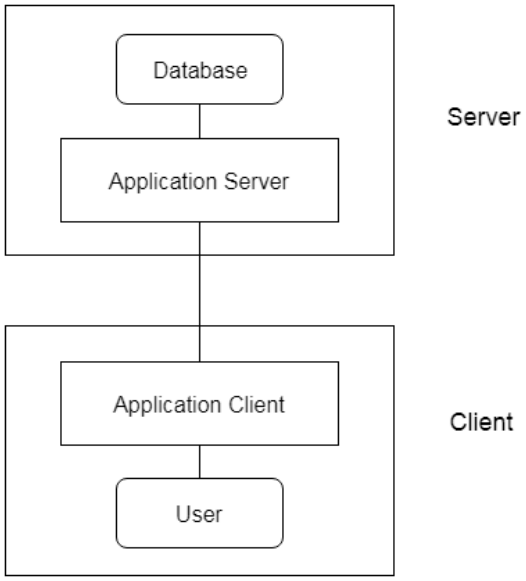


Fig: 3-tier Architecture