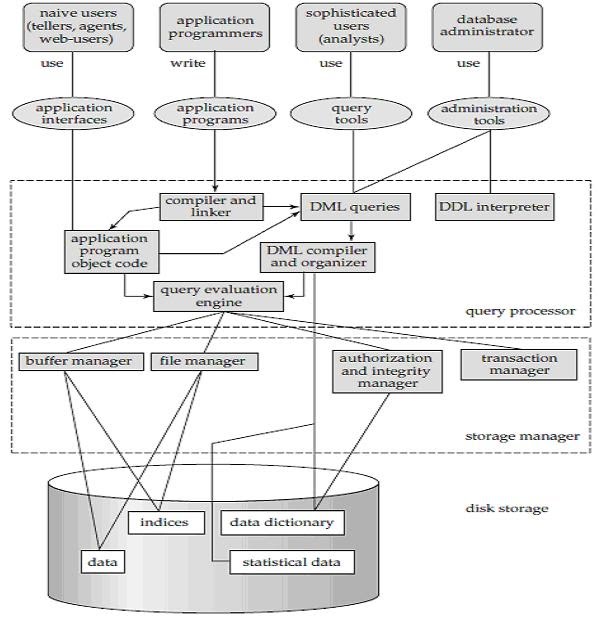
**Overall system structure of a database**

**Or**

**Database architecture**

The database system is divided into three components: Query Processor, Storage Manager, and Disk Storage. These are explained as following below.



Definition from the top:

1. Users

Applications: – It can be considered as a user-friendly web page where the user enters the requests. Here he simply enters the details that he needs and presses buttons to get the data.

End User: – They are the real users of the database. They can be developers, designers, administrators, or the actual users of the database.

1. Query Processor

* 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 usually be translated into any of a number of alternative evaluation plans that all give the same result. The DML compiler also performs query optimization, that is, it picks the lowest cost evaluation plan from among the alternatives.
* Query evaluation engine, which executes low-level instructions generated by the DML compiler.

1. Storage Manager

Storage Manager is a program that provides an interface between the data stored in the database and the queries received. It is also known as Database Control System. It maintains the consistency and integrity of the database by applying the constraints and executes the DCL statements. It is responsible for updating, storing, deleting, and retrieving data in the database.

It contains the following components –

Authorization Manager –

It ensures role-based access control, i.e. checks whether the particular person is privileged to perform the requested operation or not.

Integrity Manager –

It checks the integrity constraints when the database is modified.

Transaction Manager –

It controls concurrent access by performing the operations in a scheduled way that it receives the transaction. Thus, it ensures that the database remains in the consistent state before and after the execution of a transaction.

File Manager –

It manages the file space and the data structure used to represent information in the database.

Buffer Manager –

It is responsible for cache memory and the transfer of data between the secondary storage and main memory.

1. Disk Storage

Data Files – It stores the data.

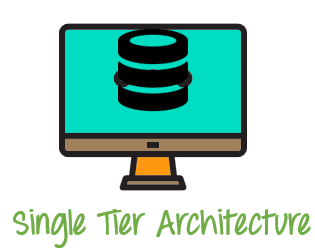
Data Dictionary – It contains the information about the structure of any database object. It is the repository of information that governs the metadata.

Indices – It provides faster retrieval of data item.

**Client/Server architecture or Application architecture of DBMS**

**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.



Advantages

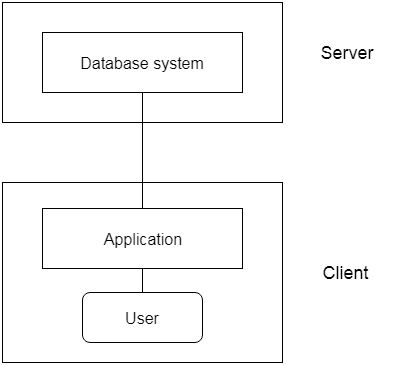
1. Easy to implement and optimize performance.
2. Do not have compatibility or Context switching issues.
3. The cost of deployment is less eg - development and management cost.

Disadvantages

1. Do not support remote/ distributed access for data resources.
2. Monolithic manner of the code causes higher maintenance.
3. The cost of the central mainframe is high.

**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.



**Advantages**

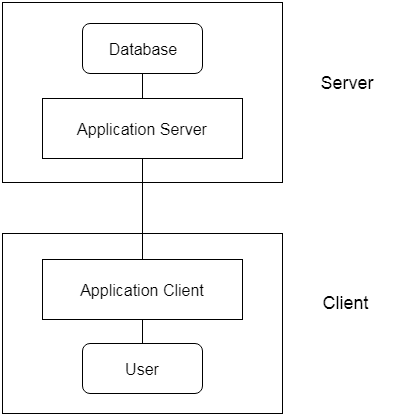
* Since this contains static business rules it’s more applicable for homogenous environments.
* Database server and business logic is physically close, which offers higher performance.
* It is fast and easy to implement
* communication is faster
* It is suitable in an environment where business rules or logic operations are static

Disadvantages

* Heterogeneous environments/Business environments with rapidly changing rules and regulations are not suitable since the database server has to handle the business logic which slows down database performance.
* Since client beholds most of the application logic, problems arise in controlling the software version and re-distributing of new versions.
* Security wise this is complicated as users need to have separate login information for every SQL server.
* It is not easily scalable; thus, performance degrades as users scale.
* Data integrity issue may arise due to the server responding to multiple requests at the same time.

**Three-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.



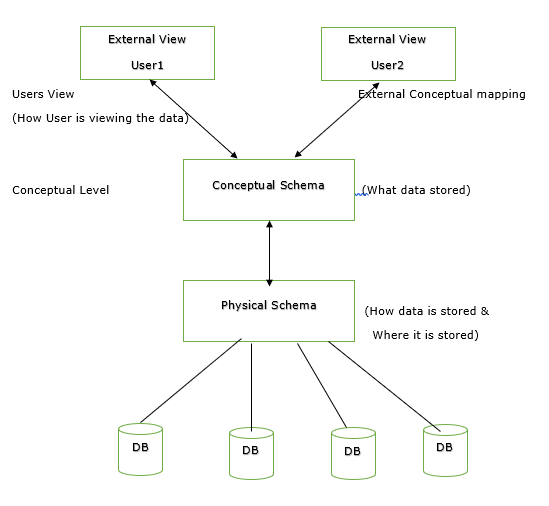
Advantages:

1. Easy to modify without affecting other modules
2. Fast communication
3. Performance will be good in three tier architecture.
4. It has better reusability of the various components as a result of the separation of concerns (interface, logic and data management).
5. Data integrity is improved due to the existence of middleware between the client and the server.
6. Maintenance is simplified because each layer is treated as an independent entity.

Disadvantages:

* 1. complexity is increased.
  2. A separate proxy server may be required.
  3. Network traffic will be increased if a separate proxy server is used.

**Three schema architecture**



The three-schema architecture divides the database into three-level used to create a separation between the physical database and the user application. In simple terms, this architecture hides the details of physical storage from the user.

The database administrator (DBA) responsible is to change the structure of database storage without affecting the user’s view. It deals with the data, the relationship between them and the different access methods implemented on the database. The logical design of database is called a schema

This architecture contains three layers of database management system, which are as follows −

* External level
* Conceptual level
* Internal level

## **External/ View level**

This is the highest level of database abstraction. It includes a number of external schemas or user views. This level provides different views of the same database for a specific user or a group of users. An external view provides a powerful and flexible security mechanism by hiding the parts of the database from a particular user.

## **Conceptual or Logical level**

This level describes the structure of the whole database. It acts as a middle layer between the physical storage and user view. It explains what data to be stored in the database, what the data types are, and what relationship exists among those data. There is only one conceptual schema per database.

This level describes the structure of the whole database. It acts as a middle layer between the physical storage and user view. It explains what data to be stored in the database, what the data types are, and what relationship exists among those data. There is only one conceptual schema per database.

## **Internal or Physical level**

This is the lowest level of database abstraction. It describes how the data is stored in the database and provides the methods to access data from the database. It allows viewing the physical representation of the database on the computer system.

The interface between the conceptual and internal schema identifies how an element in the conceptual schema is stored and how it may be accessed. It is one which is closest to physical storage. The internal schema not only defines different stored record types, but also specifies what indices exist, how stored fields are represented.