Sairam INSTITUTIONS

Sri SAI RAM ENGINEERING COLLEGE
INSTITUTE OF TECHNOLOGY
West Tambaram, Chennai - 44

YEAR	SEM
II	IV

CS 8492

DATABASE MANAGEMENT SYSTEMS
(Common to CSE & IT)

UNIT NO. 1

1.3 Database System Architecture

Version 1.0

4 QUALITY EDUCATION

NBA NATIONAL BOARD OF ACCREDITATION | NAAC | NAAC A+ | nirf NATIONAL INSTITUTIONAL RANKING FRAMEWORK | BUREAU VERITAS Certification

1.3.1 Database System Architecture

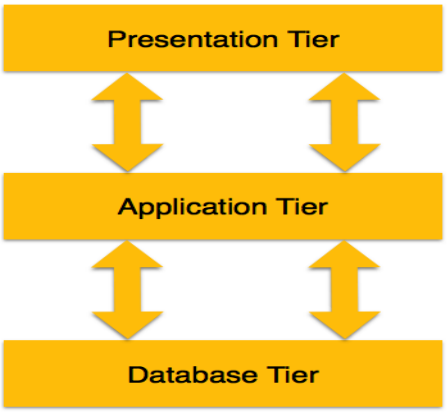
The design of a DBMS depends on its architecture. It can be centralized or decentralized or hierarchical. The architecture of a DBMS can be seen as either single tier or multi-tier. An n-tier architecture divides the whole system into related but independent n modules, which can be independently modified, altered, changed, or replaced.

In 1-tier architecture, the DBMS is the only entity where the user directly sits on the DBMS and uses it. Any changes done here will directly be done on the DBMS itself. It does not provide handy tools for end-users. Database designers and programmers normally prefer to use single-tier architecture.

If the architecture of DBMS is 2-tier, then it must have an application through which the DBMS can be accessed. Programmers use 2-tier architecture where they access the DBMS by means of an application. Here the application tier is entirely independent of the database in terms of operation, design, and programming.

3-tier Architecture

A 3-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. It is the most widely used architecture to design a DBMS.

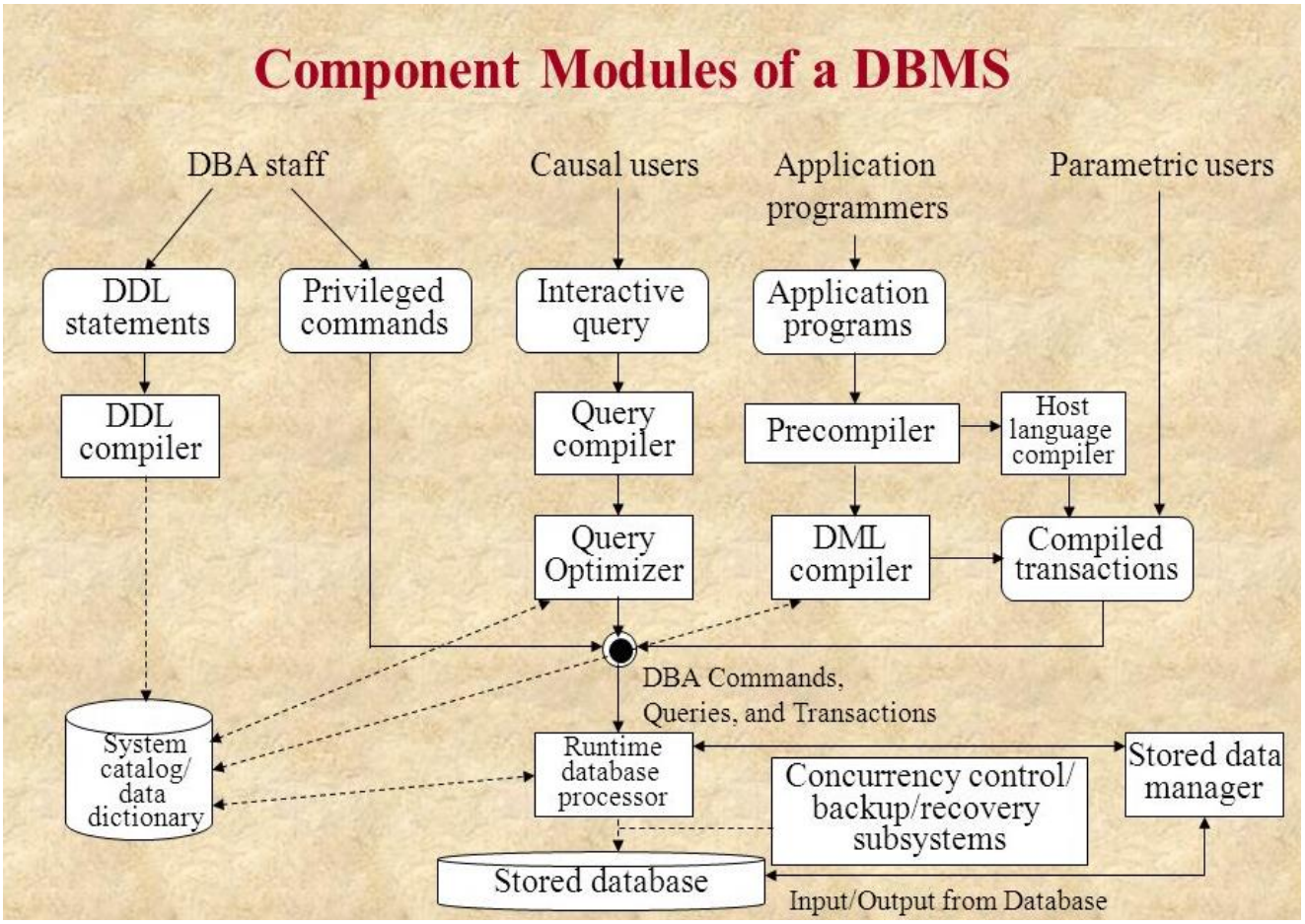


- Database (Data) Tier − At this tier, the database resides along with its query processing languages. We also have the relations that define the data and their constraints at this level.
- Application (Middle) Tier − At this tier reside the application server and the programs that access the database. For a user, this application tier presents an abstracted view of the database.

End-users are unaware of any existence of the database beyond the application. At the other end, the database tier is not aware of any other user beyond the application tier. Hence, the application layer sits in the middle and acts as a mediator between the end-user and the database.

- User (Presentation) Tier − End-users operate on this tier and they know nothing about any existence of the database beyond this layer. At this layer, multiple views of the database can be provided by the application. All views are generated by applications that reside in the application tier.

1.3.1.2 **DBMS Component Modules:**



The Database Users fall into several categories:

- **Casual users** and persons with occasional need for information from the database interact using some form of interface, called the interactive query interface. These queries are parsed and

validated for correctness of the query syntax, the names of files and data elements, and so on by a query compiler that compiles them into an internal form. This internal query is subjected to query optimization. Casual users occasionally access the database, but they may need different information each time. They use a sophisticated database query language to specify their requests and are typically middle- or high-level managers or other occasional browsers.

- **Application programmers** write programs in host languages such as Java, C, or C++ that are submitted to a precompiler. The precompiler extracts DML commands from an application program written in a host programming language. These commands are sent to the DML compiler for compilation into object code for database access. The rest of the program is sent to the host language compiler. The object codes for the DML commands and the rest of the program are linked, forming a canned transaction whose executable code includes calls to the runtime database processor. Canned transactions are executed repeatedly by parametric users, who simply supply the parameters to the transactions. Each execution is considered to be a separate transaction. An example is a bank withdrawal transaction where the account number and the amount may be supplied as parameters.
- **Naive or parametric** end users make up a sizable portion of database end users. Their main job function revolves around constantly querying and updating the database, using standard types of queries and updates-called **Compiled or Canned transactions**-that have been carefully programmed and tested. The tasks that such users perform are varied:

❖ Bank tellers check account balances and post withdrawals and deposits.
❖ Reservation clerks fur airlines, hotels, and
❖ car rental companies check availability for a given request and make reservations.
❖ Clerks at receiving stations for courier mail enter package identifications to update a central database of received and in-transit packages.

- In a database environment, the primary resource is the database itself, and the secondary resource is the DBMS and related software. Administering these resources is the responsibility of the **Database Administrator (DBA)**. The DBA is responsible for authorizing access to the database, coordinating and monitoring its use, and acquiring software and hardware resources as needed. The DBA is accountable for problems such as security breaches and poor system response time. In large organizations, the DBA is assisted by a staff that carries out these functions.

One of the main reasons for using DBMSs is to have central control of both the data and the programs that access those data. A person who has such central control over the system is called a database administrator (DBA).

- **The functions of a DBA include:**

➢ 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. The DBA may specify some parameters pertaining to the physical organization of the data and the indices to be created.

➢ Schema and physical-organization modification. The DBA carries out changes to the schema and physical organization to reflect the changing needs of the organization, or to alter the physical organization to improve performance.

➢ 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 a user tries to access the data in the system.

➢ Routine maintenance. Examples of the database administrator's routine maintenance activities are:

★ Periodically backing up the database to remote servers, to prevent loss of data in case of disasters such as flooding.

★ Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required.

★ Monitoring jobs running on the database and ensuring that performance is not degraded by very expensive tasks submitted by some users.

Most database contain **privileged commands** that can be used only by the DBA staff. These include commands for creating accounts, setting system parameters, granting account authorization, changing a schema, and reorganizing the storage structures of a database.

Database System Structure are partitioned into modules for different functions. Some functions (e.g. file systems) may be provided by the operating system. Components include:

- File Manager manages allocation of disk space and data structures used to represent information on disk.
- **Database Manager**: The interface between low-level data and application programs and queries.
- **Query Processor** translates statements in a query language into low-level instructions the database manager understands. (May also attempt to find an equivalent but more efficient form.) The Query Processor simplifies and facilitates access to data. The Query processor includes the following component.

❖ DDL Interpreter
❖ DML Compiler

- **The DDL compiler** processes schema definitions, specified in the DDL, and stores

descriptions of the schemas (meta-data) in the DBMS catalog. The catalog includes information such as the names and sizes of files, names and data types of data items, storage details of each file, mapping information among schemas, and constraints. In addition, the catalog stores many other types of information that are needed by the DBMS modules, which can then look up the catalog information as needed.

- **Query Compiler:** Interactive queries are parsed and validated for correctness of the query syntax, the names of files and data elements, and so on by a query compiler that compiles them into an internal form. This internal query is subjected to query optimization. The query optimizer is concerned with the rearrangement and possible reordering of operations, elimination of redundancies, and use of correct algorithms and indexes during execution. It consults the system catalog for statistical and other physical information about the stored data and generates executable code that performs the necessary operations for the query and makes calls on the runtime processor.

- **Precompiler:** The precompiler extracts DML commands from an application program written in a host programming language. These commands are sent to the DML compiler for compilation into object code for database access. The rest of the program is sent to the **host language compiler.** The object codes for the DML commands and the rest of the program are linked, forming a canned transaction whose executable code includes calls to the runtime database processor. Canned transactions are executed repeatedly by parametric users, who simply supply the parameters to the transactions. Each execution is considered to be a separate transaction. An example is a bank withdrawal transaction where the account number and the amount may be supplied as parameters.

- **The runtime database processor** executes

  (1) the privileged commands,

  (2) the executable query plans, and

  (3) the canned transactions with runtime parameters.

It works with the system catalog and may update it with statistics. It also works with the stored data manager, which in turn uses basic operating system services for carrying out low-level input/output (read/write) operations between the disk and main memory. The runtime database processor handles other aspects of data transfer, such as management of buffers in the main memory. Some DBMSs have their own buffer management module while others depend on the OS for buffer management. Concurrency control and backup and recovery systems are integrated into the working of the runtime database processor for purposes of transaction management.

- **Storage Manager**

The storage manager is important because databases typically require a large amount of storage

space. So it is a very important efficient use of storage, and to minimize the movement of data to and from disk .

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 the 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. The storage manager components include the following.

- **Authorization and Integrity Manager**
- **Transaction Manager**
- **File Manager**
- **Buffer Manager**

Authorization and Integrity Manager tests for the satisfaction of integrity constraints and checks the authority of users to access data. Transaction manager ensures that the database remains in a consistent state and allows concurrent transactions to proceed without conflicting.

- The file manager manages the allocation of space on disk storage and the data structures used to represent information stored on disk. The Buffer manager is responsible for fetching the data from disk storage into main memory and deciding what data to cache in main memory.
- The storage manager implements the following data structures as part of the physical system implementation. Data File, Data Dictionary, Indices. Data files store the database itself. The Data dictionary stores Metadata about the structure of the database, in particular the schema of the database. Indices provide fast access to data items.

Functions of Database Administrator

Database Administrator (DBA) is a person or group in charge of implementing DBMS in an organization. Database Administrator's job requires a high degree of technical expertise and the ability to understand and interpret management requirements at a senior level. In practice the DBA may consist of a team of people rather than just one person.

The main responsibilities of DBA are:

Makes decisions concerning the content of the database: It is the DBA's job to decide exactly what information is to be held in the database-in other words, to identify the entities of interest to the

enterprise and to identify information to be recorded about those entities .

Plans storage structures and access strategies: The DBA must also decide how the data is to be represented in the database, and must specify the representation by writing the storage structure definition (using the internal data definition language).

In addition, the associated mapping between the storage structure definition and the conceptual schema must also be specified.

Provides support to users: It is the responsibility of the DBA to provide support to the users, to ensure that the data they require is available, and to write the\ necessary external schemas (using the appropriate external data definition language).

In addition, the mapping between any given external schema and the conceptual schema must also be specified.

Defines security and integrity checks: DBA is responsible for providing the authorization and authentication checks such that no malicious users can access the database and it must remain protected. DBA must also ensure the integrity of the database.

Interprets backup and recovery strategies: In the event of damage to any portion\ of the database-caused by human error, say, or a failure in the hardware or supporting operating system-it is essential to be able to repair the data concerned with a minimum of delay and with as little effect as possible on the rest of the system.

The DBA must define and implement an appropriate recovery strategy to recover the database from all types of failures.

Monitoring performance and responding to changes in requirements: The

DBA is responsible for organizing the system to get the performance that is “best for the enterprise,” and for making the appropriate adjustments as requirements change.

Data Manager

The data manager is the central software component of the DBMS. It is sometimes referred to as the database control system. One of the functions of the data manager is to convert operations in the user’s queries coming directly via the query processor or\ indirectly via an application program from the user’s logical view to a physical file system. The data manager is responsible for interfacing with the file system as shown. In addition, the tasks of enforcing constraints to maintain the consistency and integrity of the data, as well as its security, are also performed by the data manager. It is also the responsibility of the Data. Manager to provide the synchronization in the simultaneous

operations performed by concurrent users and to maintain the backup and recovery operations.

File Manager

Responsibility for the structure of the files and managing the file space rests with the file manager. It is also responsible for locating the block containing the required record, requesting this block from the disk manager, and transmitting the required record to the data manager as shown. The file manager can be implemented using an interface to the existing file subsystem provided by the operating system of the host computer or it can include a file subsystem written especially for the DBMS.

Disk Manager

The disk manager is part of the operating system of the host computer and all physical input and output operations are performed by it. The disk manager transfers the block or page requested by the file manager so that the latter need not be concerned with the physical characteristics of the underlying storage media.