DATABASE ARCHITECTURE

Asst. Prof. Taranum Shaikh
(BSC-IT, MCA, MBA-IT, PGDBA, PGDEM,
NTA UGC NET & SET IN COMP. SCI. & APP.
PGT IN COMP. SCI.)

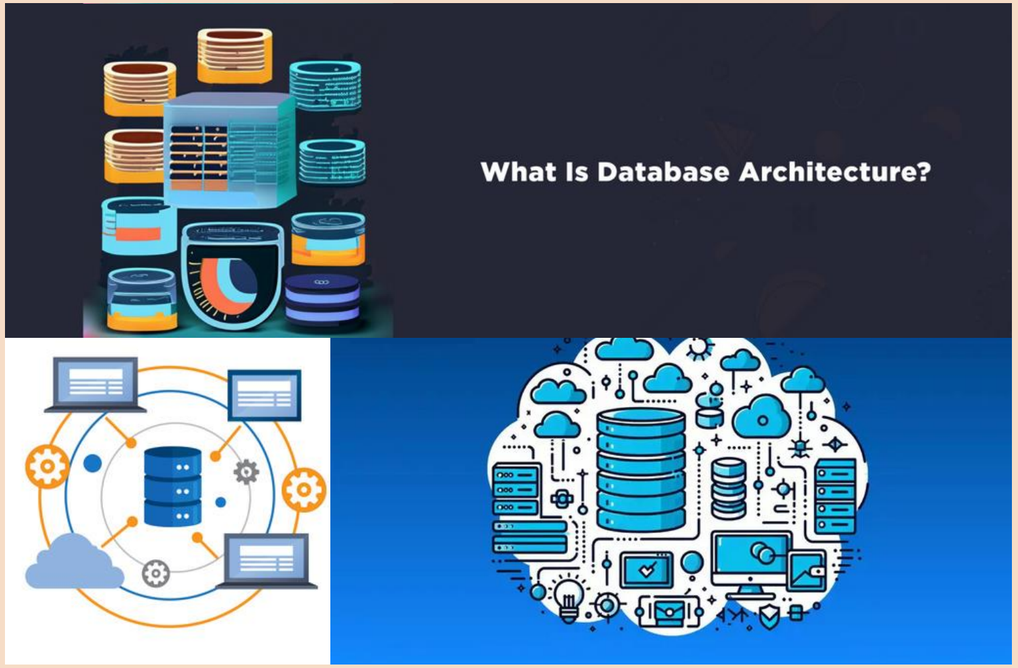
Sub : ADVANCED DATABASE MANAGEMENT SYSTEM

Class: MSC DS PART-I

INTRODUCTION

Database Architecture refers to the **structural design** and **components** that work in harmony to manage and maintain databases effectively.

DBMS architecture consists of **five** key functional components that work together to ensure efficient and reliable data management.



layers of DBMS

External Layer:

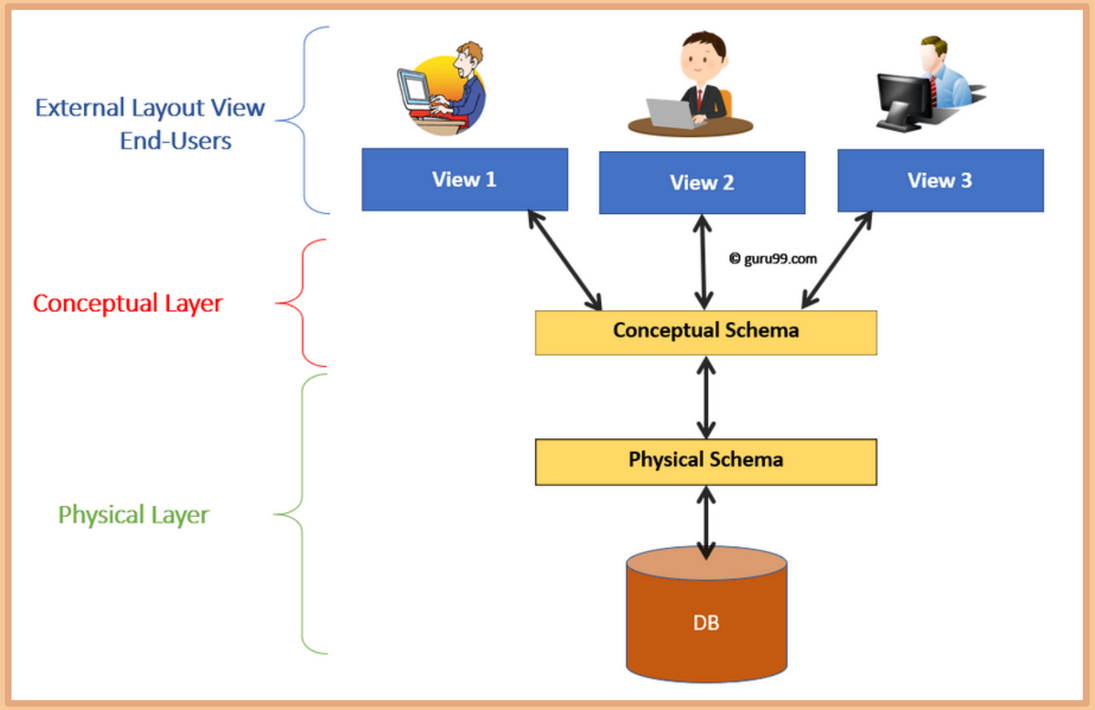
Represents the **user's view** of the database. It deals with how users access the data in the database. It allows users to view data in a way that makes sense to them, without worrying about the underlying implementation details.

Conceptual Layer:

Represents the **logical view** of the database. It deals with the overall organization of data in the database and the relationships between them. It defines the **data schema**, which includes tables, attributes, and their relationships.

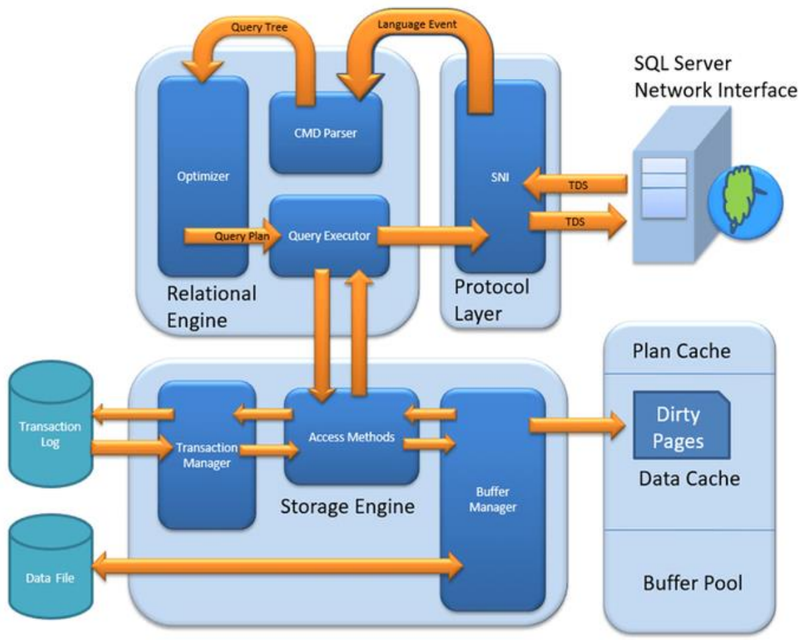
Physical Layer:

Represents the **physical storage** of data in the database. It is responsible for storing and retrieving data from the storage devices, such as **hard drives or solid-state drives.** It deals with low-level implementation details such as data compression, indexing, and storage allocation.

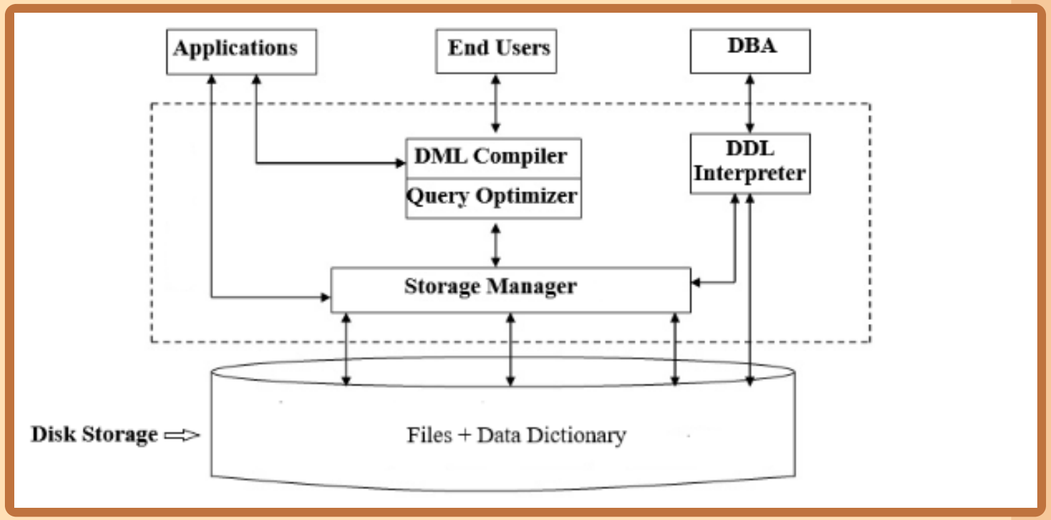


1 Database Engine

The database engine is the central component of a DBMS. It is responsible for **interpreting and executing user requests**, managing **data storage and retrieval,** and ensuring **data integrity and security**. It employs various algorithms and data structures to optimize query processing and data access. The database engine also **handles concurrent access** from multiple users, **implementing locking mechanisms** to prevent data conflicts. The engine's efficiency directly impacts the overall performance of the database system. It utilizes indexing techniques to speed up data retrieval.



2 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 contains the following components –

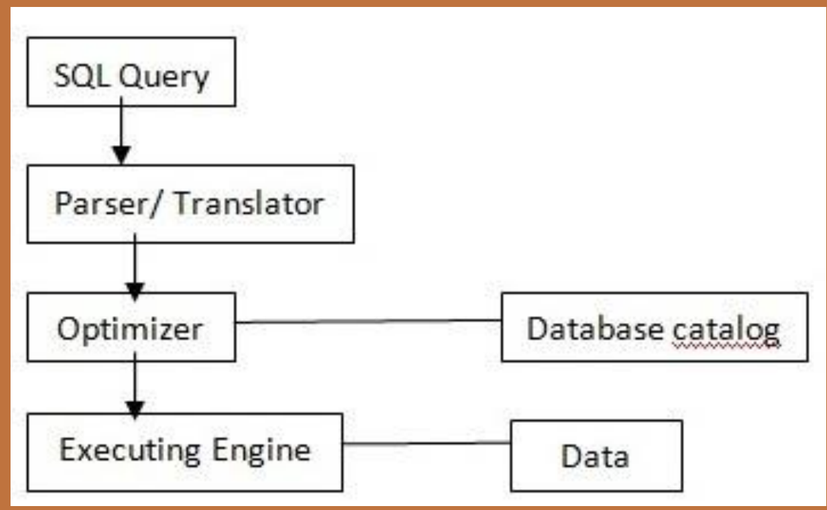
- **Authorization Manager**: It ensures role-based access control, i.e,. checks whether the particular person is privileged to perform the requested operation or not.
- **Integrity Manager**: It checks the integrity constraints when the database is modified.
- **Transaction Manager**: It controls concurrent 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.

3 Query Processor

It interprets the requests (queries) received from end user via an application program into instructions.

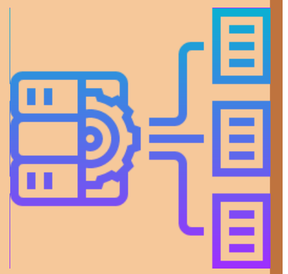
Query Processor contains the following components –

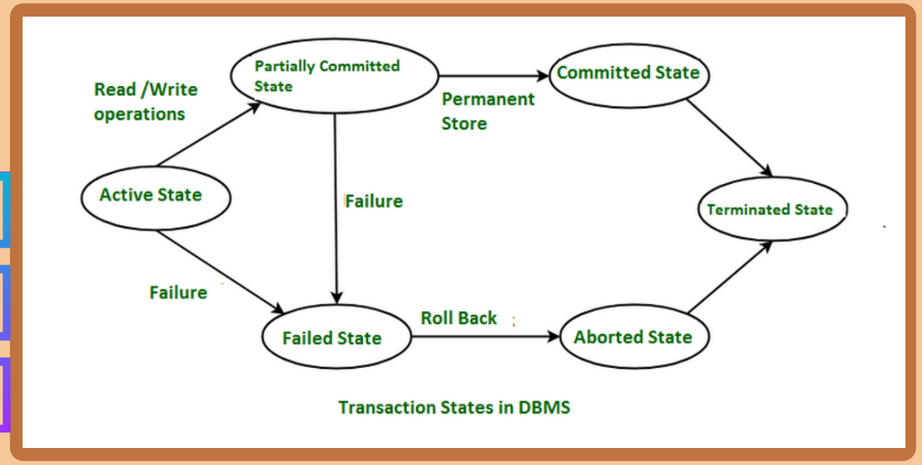
- **DML Compiler**: It processes the DML statements into low level instruction (machine language), so that they can be executed.
- **DDL Interpreter**: It processes the DDL statements into a set of table containing meta data (data about data).
- **Embedded DML Pre-compiler**: It processes DML statements embedded in an application program into procedural calls.
- **Query Optimizer**: It executes the instruction generated by DML Compiler.



4 Transaction Manager

The **transaction manager** is responsible for ensuring the **ACID** (**Atomicity, Consistency, Isolation, Durability**) properties of database transactions.





Transaction States in DBMS

One key function of the transaction manager is to **implement locking mechanisms,** which prevent concurrent access to the same data by multiple transactions. This helps **maintain isolation and consistency**.

In the case of a **system failure**, the transaction manager plays a crucial role in **recovery**. It **maintains transaction logs** that record all changes made during a transaction, allowing the system to **roll back.**

5 Concurrency Control

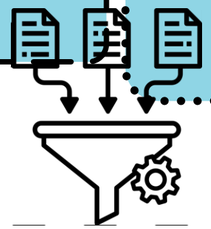
The concurrency control component is responsible for managing simultaneous access to the database by multiple users or processes.

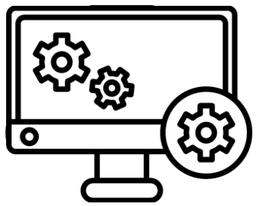
It ensures that concurrent transactions do not interfere with each other and that data integrity is maintained.

Additionally, this component handles deadlock detection and resolution, ensuring that transactions do not become indefinitely blocked due to circular dependencies on resources. **It may employ techniques such as timeout mechanisms or deadlock detection algorithms** to identify and resolve such situations

Key mechanisms :

- **Locking:** It prevents multiple transactions from accessing the same data simultaneously
- **Timestamping:** It assigns unique timestamps to transactions to determine their order of execution
- **Multi-version Concurrency Control (MVCC):** MVCC allows multiple versions of data to coexist, enabling read operations to proceed without blocking write operations.





Conclusion:

A clear understanding of Database Management System (DBMS) architecture is essential for designing and managing databases effectively. Key parts like **data storage, query processing, transaction management, and security play** important roles in ensuring smooth operations.

By knowing how these components work together, one can make **better decisions** during database development, leading to improved **performance, scalability, and security**.

This understanding helps in building strong and reliable data management solutions.

THANK YOU