Experiment 1: Study of Advanced DBMS Architecture

Prerequisites

Basic knowledge of DBMS concepts and database components

Lab Objective

To study and understand the architecture of advanced Database Management Systems (DBMS) and analyze the internal components responsible for data storage, query processing, transaction management, and recovery.

Learning Outcomes

By completing this experiment, students will be able to:

- Understand the overall architecture of a DBMS.
- Explain the three-level architecture of database systems.
- Identify core DBMS components and their functions.
- Understand the roles of query processor and storage manager.
- Analyze how transaction and recovery components interact within the system.

Introduction

An Advanced DBMS is a complex software system designed to manage large volumes of structured data efficiently, securely, and reliably. It provides mechanisms for data storage, retrieval, concurrency control, recovery, and security.

Understanding DBMS architecture is essential for analyzing how queries are processed, how data is stored, and how system performance is optimized.

Types of DBMS Architecture

1. Three-Level Architecture (ANSI-SPARC Model)

- **External Level:** User view of the database
- **Conceptual Level:** Logical structure of the entire database
- **Internal Level:** Physical storage details

This architecture provides **data independence**, allowing changes at one level without affecting others.

---

2. Client-Server Architecture

- **Two-Tier Architecture:** Client communicates directly with database server.
- **Three-Tier Architecture:** Client → Application Server → Database Server.

Common in enterprise applications and web-based systems.

Core Components of Advanced DBMS

1. Query Processor

- Parses SQL queries
- Optimizes query execution
- Generates execution plan

Subcomponents:

- DDL Interpreter
- DML Compiler
- Query Optimizer

---

2. Storage Manager

- Manages data files and indexes
- Handles disk space allocation
- Maintains buffer management

Subcomponents:

- File Manager
- Buffer Manager
- Index Manager

---

3. Transaction Management

- Ensures ACID properties
- Handles concurrency control
- Manages locks and isolation

---

4. Recovery Manager

- Maintains transaction logs
- Performs Undo and Redo operations
- Ensures database consistency after failure

---

5. Authorization & Integrity Manager

- Manages user access rights
- Enforces constraints and security policies

Experiment Steps

---

1. Study DBMS Architecture Diagram

- Draw and label the DBMS architecture.
- Identify main components and explain their roles.

---

2. Analyze Query Processing Flow

Trace the steps followed when a SQL query is executed:

1. Query parsing
2. Query optimization
3. Execution plan generation
4. Data retrieval
5. Result delivery

---

3. Study Storage Mechanism

- Understand how data is stored on disk.
- Study buffer management concept.
- Observe role of indexes in storage structure.

---

4. Study Transaction & Recovery Interaction

- Understand how transactions interact with storage manager.
- Study role of log files in crash recovery.

---

5. Compare Architecture Types

Create a comparison between:

- Two-tier vs Three-tier
- Centralized vs Distributed DBMS

How It Works

- The DBMS receives a query from the user.
- Query processor parses and optimizes the query.
- Storage manager retrieves data from disk or memory.
- Transaction manager ensures safe execution.
- Recovery manager logs operations for fault tolerance.

All components work together to ensure efficient, secure, and reliable data management.

Submission Instructions

Students must submit:

- Neatly drawn and labeled DBMS architecture diagram.
- Explanation of each major component.
- Flow explanation of query processing.
- Comparison table of architecture types.

Conclusion

Understanding Advanced DBMS Architecture provides a foundation for studying query optimization, transaction management, recovery techniques, and distributed database systems. It helps students analyze how database systems function internally and maintain performance, consistency, and reliability.