Unit IV — Database Implementation & Management

Database Management System | BMB IT 03 / KMBN IT 03

Unit IV: Database Implementation and Management

Exam Blueprint (2022–2025 papers)

Topic	Years asked	Marks
Transactions and transaction states	2024, 2025	2 & 10
ACID properties	2022 (isolation), 2025	2 & 7
Concurrency control	2023, 2024	2 & 10
Deadlock — detection and recovery	2022, 2024	2 & 10
Two-Phase Locking protocol and variants	2025	7
Serializability (serial vs non-serial schedules)	2023	10
Log-based recovery / transaction log	2022, 2023	2 & 10
Indexing and B-trees	2022, 2023	2

> YE QUESTION BAAR-BAAR AAYA HAI
>
> **ACID properties** aur **Two-Phase Locking** — ye do sabse safe bets hain. 2025 me dono aaye. ACID ke chaar naam aur unka matlab ek line me ratt lo, aur 2PL ke do phases zaroor yaad rakho.

1. Database Storage and Physical Structures

1.1 Storage Hierarchy

```
FASTEST, COSTLIEST, SMALLEST
     |  Cache / CPU registers      |  Primary
     |  Main memory (RAM)          |  (volatile)
     |-----------------------------|
     |  Flash / SSD                |  Secondary
     |  Magnetic disk (HDD)        |  (non-volatile)
     |-----------------------------|
     |  Optical disk, Magnetic tape|  Tertiary
SLOWEST, CHEAPEST, LARGEST          (archival)
```

Volatile storage loses contents on power failure (RAM). **Non-volatile storage** survives (disk, SSD). Databases live on non-volatile storage because of **durability**.

1.2 File Organisation

Type	Description	Best for
Heap (unordered)	Records placed wherever there is space	Fast insertion; slow search
Sequential (ordered)	Records stored sorted on a key	Range queries, ordered retrieval
Hashing	A hash function on the key decides the block address	Very fast **equality** search
Clustered	Related records of different tables stored together	Frequent joins on those tables

1.3 Indexing

> **DEFINITION - RATTA MAARO**
>
> An **index** is an auxiliary data structure that speeds up retrieval, storing (search-key, pointer) pairs — like the index of a book.

Type	Description
Primary index	On the **ordering key** field of a sequentially ordered file
Clustering index	On a **non-key ordering** field
Secondary index	On a **non-ordering** field; a table may have many
Dense index	An index entry for **every** search-key value
Sparse index	An index entry for **some** values only (needs an ordered file); smaller but slower
Multilevel index	An index on the index, to reduce search levels

Trade-off: indexes make much faster but **slow down** , and , and consume extra storage.

1.4 B-Trees and B+ Trees

> **DEFINITION - RATTA MAARO**
>
> A **B-tree** is a balanced multi-way search tree in which all leaf nodes are at the **same level**, keeping search, insertion and deletion cost logarithmic. It is the standard database index structure.

Properties of a B-tree of order *n*:

- Every node has at most *n* children and at least ?n/2? children (except the root)
- All leaves appear at the **same level** — the tree is always **balanced**
- Keys within a node are kept **sorted**
- Search, insert and delete are all **O(log n)**

B+ Tree — the variant actually used in databases:

Basis	B-Tree	B+ Tree
Where data pointers are stored	In **both** internal and leaf nodes	**Only in leaf nodes**
Leaf nodes	Not linked	**Linked as a sequential list**
Range queries	Inefficient	**Very efficient** (walk the leaf chain)