Megha
590017642.
BCA/B6.

PART-A.

Ans 1) Characteristics of secondary storage device

i) Non-volocity → Data is permanantly stored (unlike RAM)
ii) Large capacity → Can store a huge amount of data (GB's to BT's)
iii) Slower than primary memory → Accessing data takes more time composed to RAM).
iv) Cost-effective → Much cheaper per GB compared to RAM
v) portability → Many devices (USB's, external HDDS/SSDS) can be carried easily.

Comparisson b/w HDD vs SSD.

feature	HDD	SSD
Acess time	Slower (because it has moving mechanical parts like spinning disks and read/write heads). Typical: 5-10 hrs.	Much faster (no moving parts, uses flash memory). Typical: 0-1 ms or less
Capacity	generally higher upto 20 TB + av	lower compared to HDDS (common sizes 256 GB-4TB, higher is costly).
Durability	less durable (can be damaged by drops/shocks since its has moving parts.	More durable (no moving parts, resistance to shock/vibration).

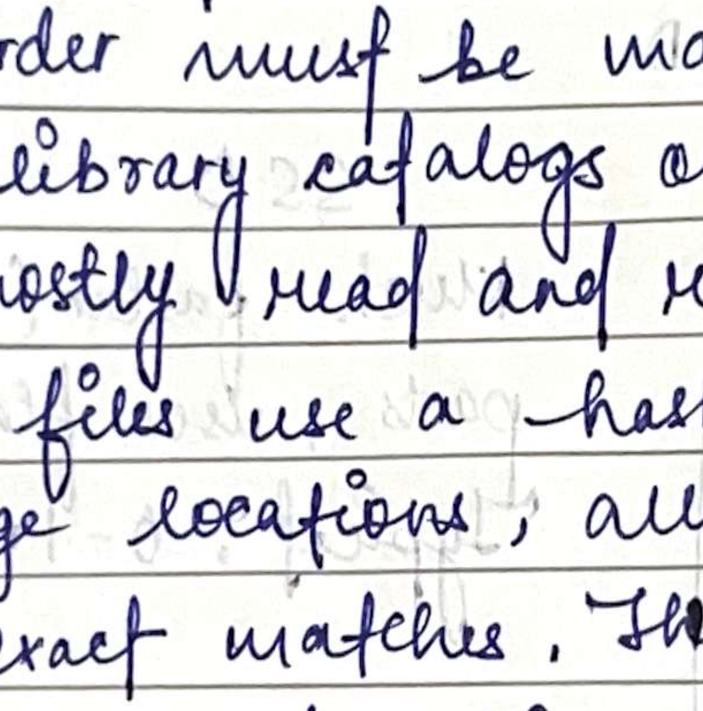
Megha
590017642
BCA/B6.

Ans 2 In DBMS, file organization determines how records are stored accessed, Heap files store records in the order they arrive, without any sorting or indexing. This makes insertion very fast but searching slow, as the entire file may need to be scanned, Heap files are suitable for applications like log storage or backup where records are written sequentially and retrieval is infrequent.

Sorted files store records in a specific order based on a key, enabling efficient searches, especially for range queries. However, inserting new records is slower because the order must be maintained. This method is ideal for library catalogs or student databases, where data is mostly read and rarely updated.

Hash files use a hash function to map keys to storage locations, allowing extremely fast direct access for exact matches. They are best for banking systems, where account details are frequently retrieved using account numbers.

Ans 3



P_1, P_2, P_3, P_4 are the representation of diff. processor user queries or system operation that need to access of data.

Actual storage of database.
Data is organized into blocks
Each blocks holds specific part of database.

Megha
39801764 2
BCA/B6

Database blocks organisations On Disk:

- logical vs physical organisation - users see data organised into tables (like table A & table B) the database physically stores this data in blocks on disk.
- System Catalog: Block 0 is special block that holds systems meta data, It is like table of content for the database, containing info about where tables, their schemes and other vital details.

Table data: The rows of table (eg. Table A or B) are stored in consecutive blocks. If a table is too large for one block, its data is spread across multiple blocks.

Role of Buffer Management

A crucial component that acts as a bridge b/w the fast main memory (RAM) and the slow disk storage. Its ~~storage~~ primary role is to minimize the no. of slow disk I/O operations.

Core functions:

- Data request handling: When a user or application needs to access data, the request is first sent to the buffer manager, The manager checks if required data block already exist in buffer pool.
- Buffer list: If the block is in the block pool, its a "hit" The buffer, and data is accused very quickly without needing its go to the disk.
- Buffer Miss: If block is not buffer pool, its a "miss". The buffer manager must then fetch the block from the disk and load it into an available frame in buffer pool.

Megha
59801764 2
BCA/B6

Ans 4 RAID (Redundant Array of independent Disks) a technology that combines multiples physical disks into one logical unit for improved reliability and/or performance by using striping, mirroring, or parity techniques.

→ RAID 0 → Data is stripped across disks, no redundancy.

→ Advantages.
High read/write speed, full capacity usable.

→ Disadvantages.
No fault-tolerance - disk failure loses all data.

→ RAID 1 → Mirroring: same data on 2 disks

→ Advantages.
Data redundancy, high read speed.

→ Disadvantages.
Only 50% storage efficiency, costlier.

→ RAID 5 → Data striped with distributed parity.

→ Advantages.
fault tolerance, efficient storage, good speed.

→ Disadvantages.
parity update slows writes, needs 3+ disks.

Ans 5 Hashing is a file organizing method using a hash function to compute the address of a data record, enabling fast direct access.

- Static Hashing: Hash function and bucket set are fixed. Ex - if hash function is $h(K) = K \bmod 10$, Keys 19 and 29 both go to bucket 9. Buckets may overflow if too many records hash to the same slot.
- Dynamic Hashing: Bucket set grows/shrinks as needed. Ex: extendible hashing uses a directory that expands when buckets overflow, minimizing collisions and storage waste.

Megha
3760176 42
BCA/B6.

Ans 6: • Primary index: Built on an ordered data files primary key. ex → index on StudentID in a sorted student file — fast, direct access.
• Secondary index: Built on a non-unique, non-ordering field. ex → index on "Department" in employee file — supports queries on non-key fields.
• Clustering index: Built on a non-key field where data is physically grouped. ex → employees stored by department index points to blocks of department.

Ans 7 → B-Tree: Balanced multi-way tree with keys and records stored at all nodes; enables fast searches, insertions, deletions
→ B+ Tree: Variant of B-Tree; Only leaf nodes store records & and leaves are linked for efficient range queries.

feature	B-Tree	B+ Tree
Data Storage	All nodes	Only leaf nodes
Range query	less efficient	Very efficient
Index size	larger	Smaller.

B+ Tree are preffered for DB indexing because all records are found at leaves, internal nodes use only keys (smaller, faster, traversal, and leaf-linking enables fast range queries.

Ans 8 Multilevel indexing creates a hierarchy of indexing, with a primary index pointing to secondary index blocks, continued iteratively until the final index points to data blocks. This reduces each index's size. and the number of disk accesses required for lookup, greatly increasing search speed over large files compared to single-level indices.

Megha
5900137642
BCA/B6

Ans 9 → Data records are placed in blocks. The blocking fac
is the number of records per block (Block size / Record si
→ for fixed size records, fill each disk block till full.
for variable - length, may span block or not.

- Steps:
Compute blocking factor.
↓
groups records into blocks.
↓
writes blocks to sequentially disk addresses.

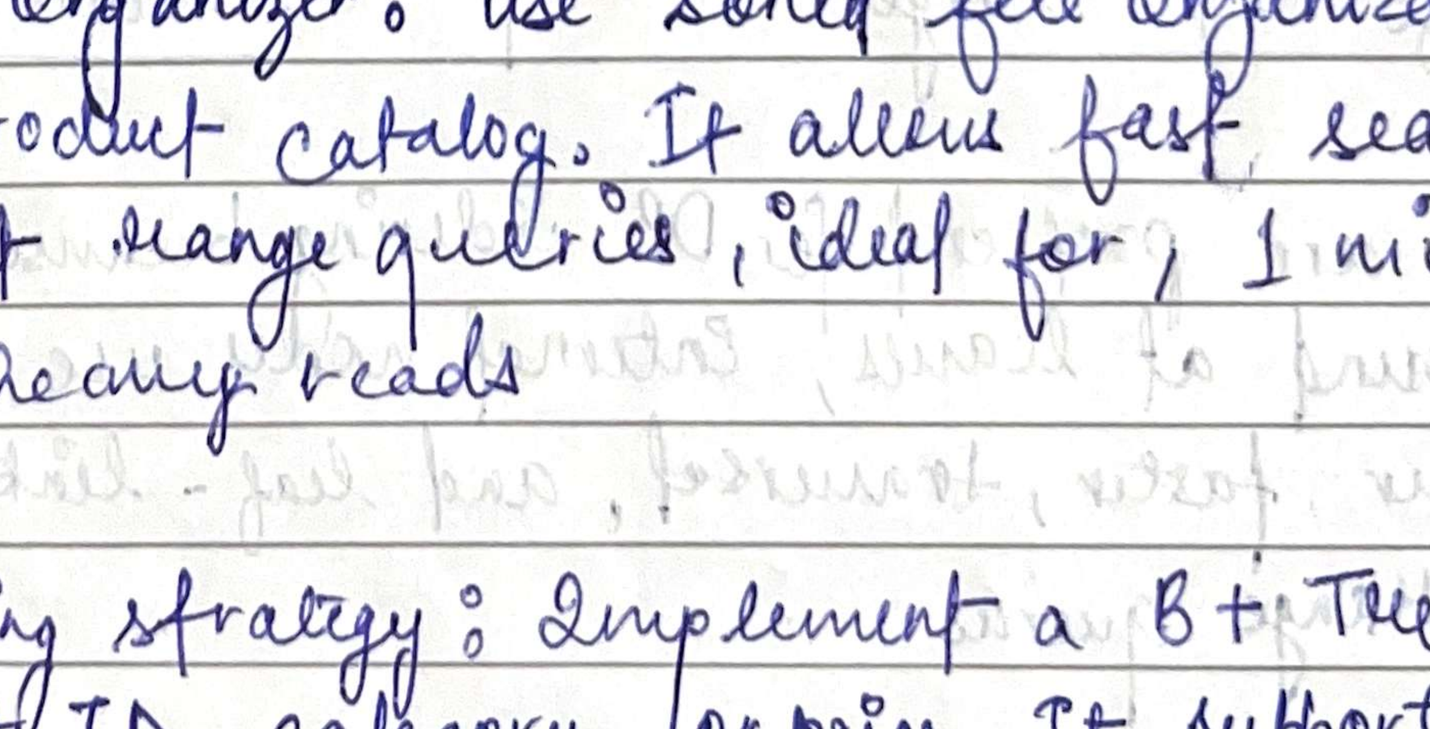
Ans 10 → Single-key indexing - index built using one attribute
→ Multi-key indexing - index built using a combination of attributes. Composite indexes speed up queries filtering on multiple columns together.

- Advantages: Reduces I/O for multi-condition queries and increase query performance with compound search condit

PART B

Ans 11 1) File Organizer: Use sorted file organization for the product catalog. It allows fast searches and efficient range queries, ideal for, 1 million products with heavy reads

2) Indexing strategy: Implement a B+ Tree index on product ID, category, or price. It supports quicks lookup efficient range queries, and sequential access.



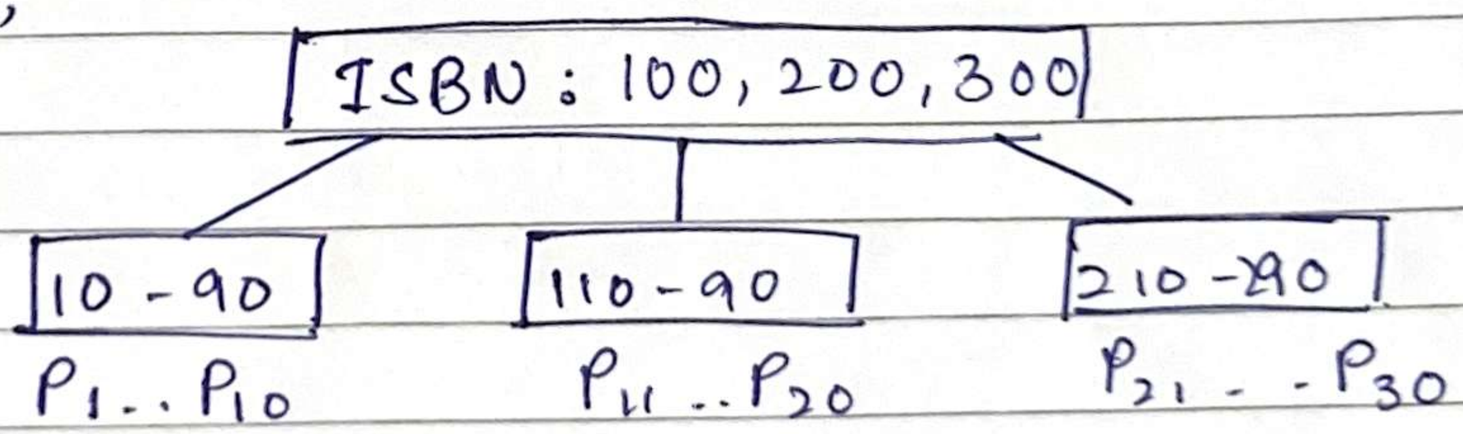
Megha
590017642
BCA/B6

Ans 12 a) file Organisation and Indexing

- Sorted file organisation by ISBN for efficient searches and report generation.
- Reason: Supports fast searches, range queries, and sorted output for reports, while daily insertions are managable with periodic organisation.

b) Critical Index types:

- Primary B+ ~~clustered~~ Tree on ISBN for fast- look up
- Secondary B+ Tree index on title and author to accelerate searches,



Megha
5980176 42
BCA | B6.

Saathi

Date ___ / ___ / _____

Am13A) RAID & Storage Hierarchy

- RAID (eg:- RAID 10): Provides high throughput, fault Tolerance, and Parallel I/O for sub-millisecond queries.

- Storage Hierarchy:- Uses SSDs for hot, real-time data and HDDA/archives for older batch data, balancing speed and cost.

B) Indexing Comparison:-

- Hash-Based: Fast exact-match queries,
- Tree-based (B+ trees): efficient for range queries, sorting and dynamic insertion, suitable for mixed workloads.

C) Buffer and Block Management:-

- Frequently accessed blocks cached in RAM; write-back/write-through policies optimize disk usage.
- Disk blocks allocated sequentially for analytics, clustered for real time across.