Module-5

Hashing: Can search for a data in O(1) time.

- → Hashing refers to a technique used to quickly locate a specific record or data item within a database using a hash function.
- → Commonly used terms: {k mod n, Mid Square, Folding method}
  - → Hash function: used to generate hash code based on the key.
  - → Hash table: The hash code is then used as an index into a data structure called hash table. A hash-table is an array like structure, pointing towards actual records in the DB.
  - → Collision: A situation where different keys producing same hash code, need to be stored in same index of table.
  - → Collision resolution: Handling Collisions.
    - → Chaining (Open Hashing) {linked list way}
    - → Open Addressing (closed Hashing)
      - → Linear Probing (putting in next free space)
      - → Quadratic Probing (Quadratic Eqn decides new hash code)
      - → Double hashing

Types of Hashing:

Static Hashing

- → Division hashing
- → Multiplication hashing
- → Modulo hashing

Dynamic Hashing

- → Extendible Hashing
- → Linear Hashing

⇒ Static Hashing:

when a search key value provided, the hash function always computes the same address

Ex: mod(4) has only 4 values : 0,1,2,3

⇒ Dynamic Hashing:

The drawback of static hashing is that it does not expand or shrink dynamically as the size of DB grows/shrinks.

In dynamic hashing, data bucket grows or shrinks (added or removed dynamically) as records increase or decrease.

→ Extendible hashing: (Minimizing collisions & efficient data retrieval)

→ directories & buckets are used to hash data, instead of hash function.

→ An aggressively flexible method in which the hash function experiences dynamic changes.

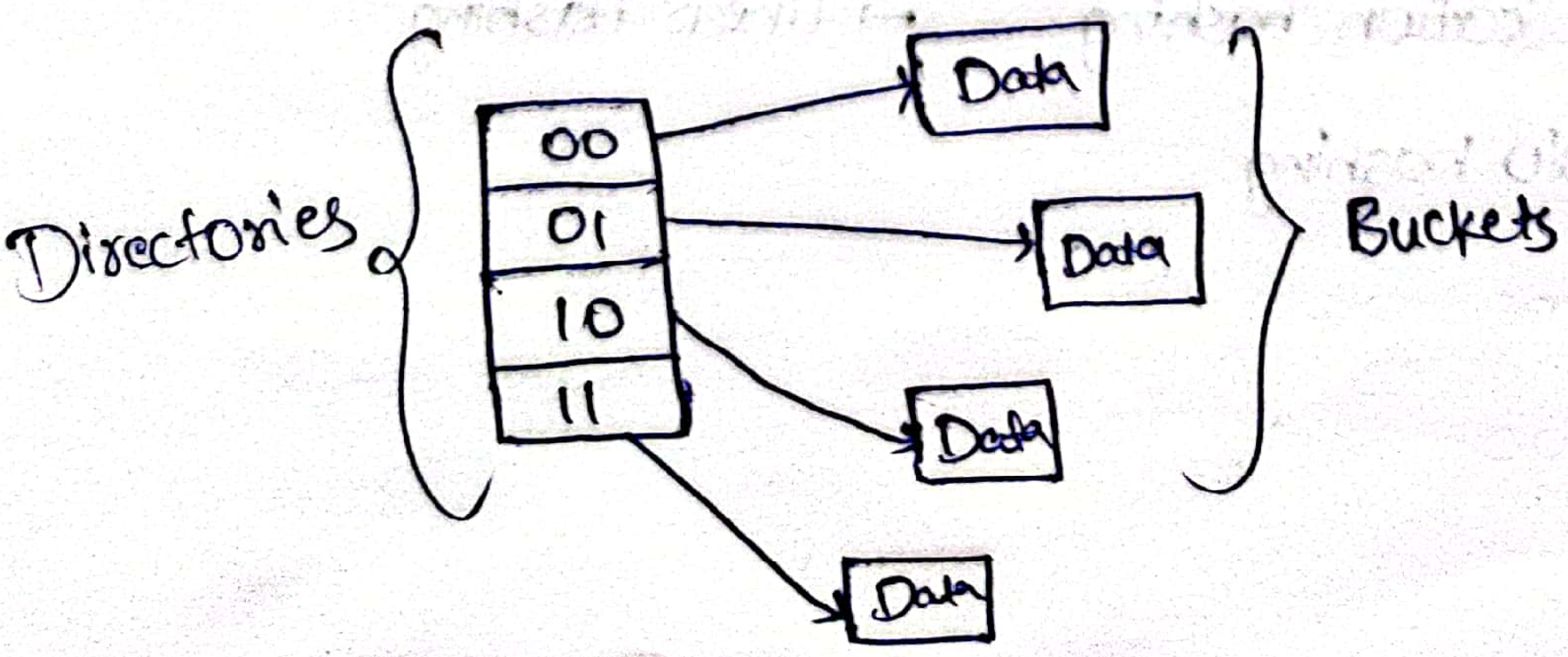
→ Directories : store addresses of buckets in pointers. An ID is assigned to each directory which may change each time when directory expands

→ Buckets : used to hash the actual data.

→ linear hashing:

→ handles growing datasets efficiently with no need for frequent global restruction, which can lead to performance bottlenecks.

→ can grow or shrink one bucket at a time.



→ file structure adapts itself to the changes in size of file.

→ linear hashing file expands by splitting a predetermined bucket into two & contracts by merging two predetermined buckets Into one.

⇒ Hash based indexing:

→ A technique used In DBMS to quickly locate records or data entries in db using hash function & hash table.

→ This provides fast data retrieval (usually O(1)).

→ consider factors like distribution of keys, quality of hash function & chosen collision resolution strategy to ensure effectiveness of hash based indexing.

⇒ Tree based Indexing:

→ A method to organize and effeciently retrieve data by using a tree data structure.

Ex: B-Tree, BST, etc.

→ best for finding range (in b/w) queries.

→ uses the same concept of key-index where primary key is used to sort the records.

File Organizations:

1. Sequential F.O.:

→ Basic data storage method used In DBMS to store records in sequence of particular order, typically the order they (records) were Inserted Into the file.

→ straight forward & easy to Implement

2. Heap F.O.:

→ Basic & simple data storage method where records are Inserted into the file as they arrive without any specific order and placed wherever there's available space within the file.

→ works well when records are inserted frequently but arent accessed in particular order.

3. Hash / Direct File Organization:

→ Records are organized and accessed based on hash function.

→ Insertion occurs in specific order.

→ Quicker access to records using hash value.

4. Indexed sequential access method:

→ Sequential + Indexed access combination

→ Sequential & random access of records in DB

5. B+ Tree File Organization:

→ widely used in the scenarios of quick access & searching.

→ Suitable for db where datasets are significantly large and need to be stored on disk.

→ Data records are stored in leaf nodes and are at same level.

→ Efficient searching & great for range queries.

6. Cluster File Organization:

→ when two or more records are stored in the same file, it is known as clusters.

→ This refers to the way, files are physically stored on storage medium, such as hard disks, in clusters or blocks

→ clusters are smallest unit of allocation for file storage on most file systems.

→ To efficiently manage & utilize storage space, minimize fragmentation, optimize file access & retrieval.

I/O costs for all File Organizations:

F.O.	Pros	cons
Heap File	storage efficiency fast scan & insertion	slow searching & deletion
Sorted file	storage efficiency searches >>> heap fast	Insertion, deletions are slow.
clustered file	Sorted file pros + insert, delete, search are fast	
unclustered tree, hash indexes	fast searches, insertion, deletion	scan & range searches are slow.

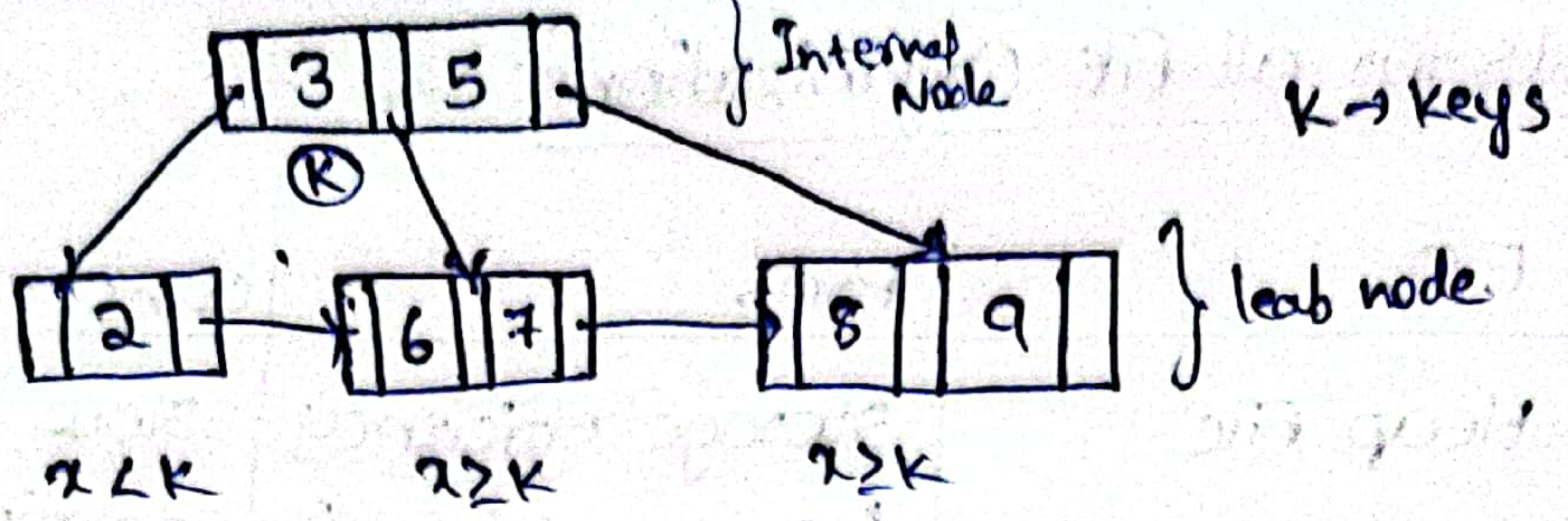
ISAM: Indexed Sequential access Method.

- → developed at IBM
- → creating, maintaining & manipulating files of data to make sequential and random retrievals possible by one or more keys.
- → Indexes of key fields are maintained to achieve for fast retrieval of required files.

B+ Tree: Balanced binary search tree.

- → used to implement db indexes.
- → In B+ Tree, leaf nodes denotes actual data pointers.
- → All leaf nodes must be at same height.
- → leaf nodes are linked using linked list.
  - → supports random access. as well as sequential access.
- → every leaf node is equidistance from root node.
- → Internal nodes just stores keys.

Ex:



Searching in B+ Tree :

1. Start at the root
2. Search in internal nodes.
3. Descend to Leaf nodes
4. Sequential search in leaf node.

Insertion in B+ Tree :

1. Search for leaf node
2. Insertion in leaf node.
3. If full, splitting the leaf node
4. Updating parent nodes.
5. Balancing & Maintenance

Deletion in B+ Tree:

1. Search for the key
2. Delete key in last node
3. Underflow handling
4. Update parent nodes.
5. Root Node update

RAID levels: Redundant Array of Independent disks.

→ combining 2 or more physical ~~devices~~ drives into logical unit presented as single Hard drive to OS.

→ Provide varying degrees of reliability, withstand drive failure. speed of I/O.

→ Six Raid levels: