File Organization

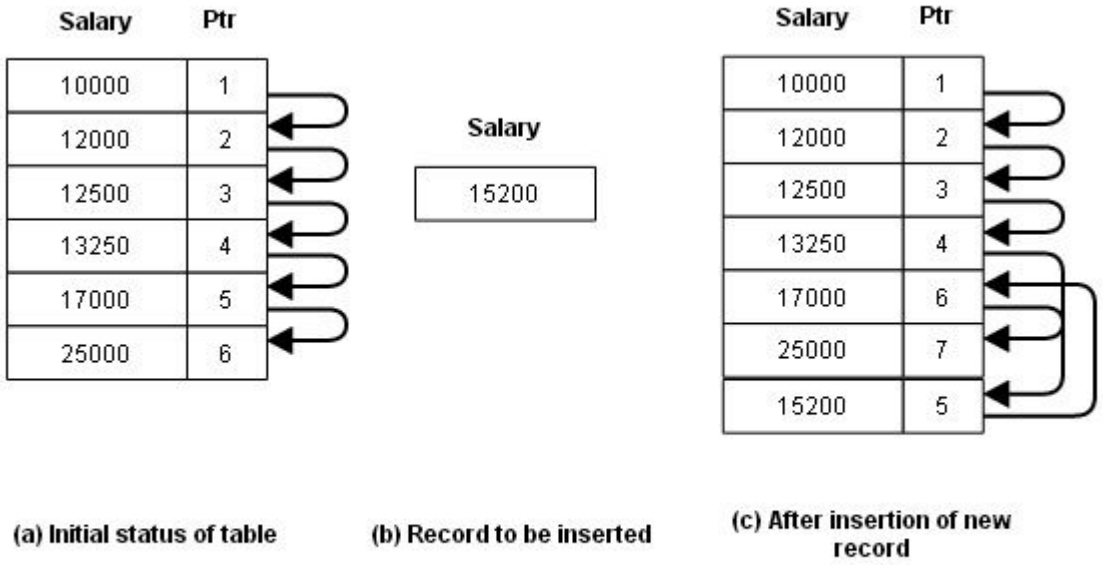
And Hashing

How do you physically store the data ?

Objectives of file organization

1. Records should be accessed as fast as possible
2. Any insert, update or delete transaction on records should be easy, quick and should not harm other records.
3. No duplicate records should be induced as a result of insert, update or delete
4. Records should be stored efficiently so that cost of storage is minimal.

1. Sequential File Organization



Sequential (sorted) file organization - Record insertion

Insertion:-

1. Insert the record at the end of the file
2. Set appropriate pointer for the inserting record
3. Update other record pointers accordingly
4. Reorganize all the records to arrange the records in sorting order of the ordering attribute

Deletion:-

1. Locate the record to be deleted
2. Update all other records' pointer

Advantages:

1. Efficient query retrieval if the query uses sorting attribute as the search key

Disadvantages:

1. Insertion and deletion are expensive
2. Updating the sorting attribute values is expensive
3. Retrieval of records on non-ordering attributes is not efficient

2. Heap file organization (unordered)

Simplest type of file organization

Insertion:

1. New records are inserted at the end of the file
2. No order is required in storing the database

Deletion:

1. Frequent deletion of records would create much free space, thus needs periodic reorganization of the file

Advantages:

1. Simple file organization
2. Insertion is easy
3. Best method for bulk loading of data

Disadvantages:

1. Because of no particular ordering, finding a record would require to do a linear search only.
2. Frequent deletion would required periodic file reorganization

3. Hash file organization

It is a file organization technique where a hash function is used to compute the address of a record. It uses value of an attribute or set of attributes and gives the location where the record could be stored

Points:

1. If bucket(s) is/are full, then overflow buckets can be used to store more records.
2. The attribute(s) that is frequently used for data manipulation can be chosen as the input for the hash function.
3. Same hash function that was used to store the records has to be used for deletion, modification or selection of records.

Advantages:

1. Quick access to records in terms of selection. [If queried on the attribute that was used for hashing]
2. Easy to insert, delete, or update a record.

Disadvantages:

1. Records are randomly stored in scattered locations. May waste a lot of space in case of small files.
2. Not efficient for range based queries
3. If querying attribute is not the hashed attribute, you may need to scan the entire table for retrieval.

Two types of hashing

1. Static Hashing
2. Dynamic Hashing

Static hashing

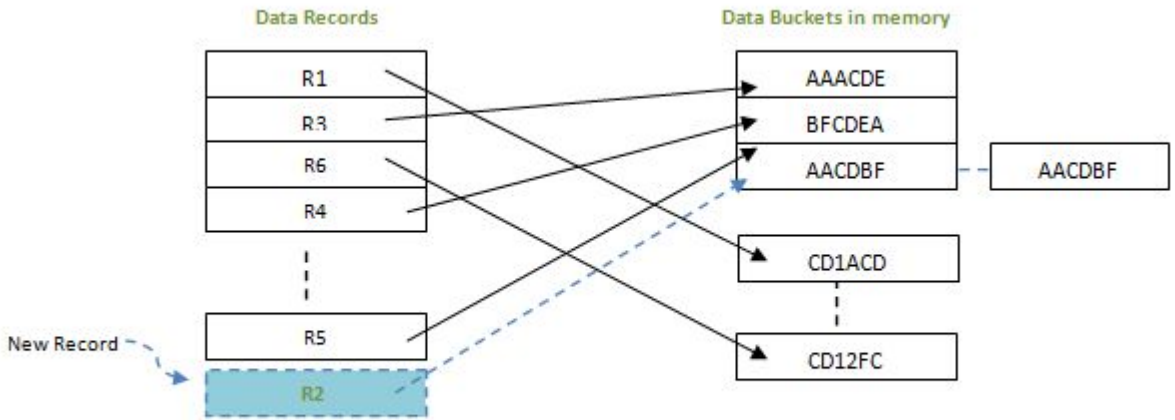
In this method, the resultant data address bucket will always be the same.

Problem: **Bucket overflow**

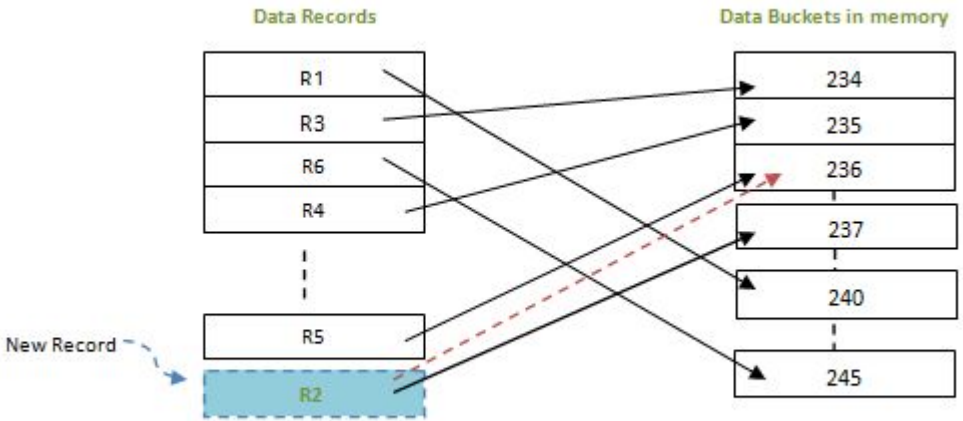
What if the data bucket address generated by the hash function is already full?

Some ways to resolve it.

Closed hashing (overflow chaining)



Open hashing (linear probing)



Dynamic Hashing (extendible hashing)

In this method of hashing, data buckets grows or shrinks as the records increases or decreases.

Continuing with an example.