

Experiment No. : 8

Title: Implement Dense Index structure to search particular record.

Objectives:

1. To learn basic indexing technique.
2. To learn to implement dense index on table.

Key Concepts: Indexing, Indexed table, dense index.

Theory:

Indexing

A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of slower writes and the use of more storage space. Indices can be created using one or more columns of a database table, providing the basis for both rapid random lookups and efficient access of ordered records.

In a relational database, an index is a copy of one part of a table. Some databases extend the power of indexing by allowing indices to be created on functions or expressions. For example, an index could be created on upper(last_name), which would only store the upper case versions of the last_name field in the index. Another option sometimes supported is the use of "filtered" indices, where index entries are created only for those records that satisfy some conditional expression.

There are two basic kinds of indices:

- **Ordered indices:**

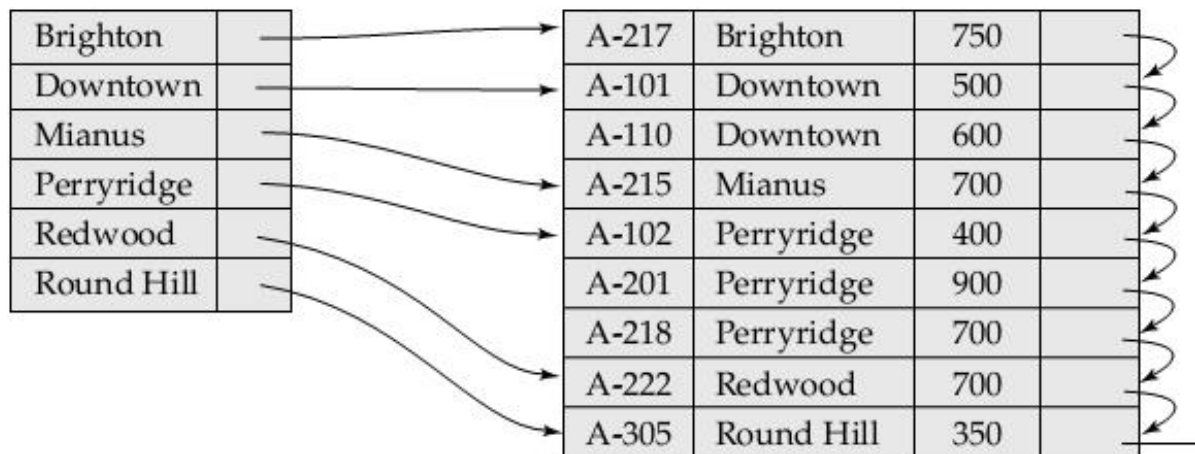
These are based on a sorted ordering of the values.

- **Hash indices:**

These are based on a uniform distribution of values across a range of buckets. The bucket to which a value is assigned is determined by a function, called a hash function

Dense index:

An index record appears for every search-key value in the file. In a dense primary index, the index record contains the search-key value and a pointer to the first data record with that search-key value. The rest of the records with the same search key-value would be stored sequentially after the first record, since, because the index is a primary one, records are sorted on the same search key. Dense index implementations may store a list of pointers to all records with the same search-key value; doing so is not essential for primary indices.



Dense index.

Algorithm:

1. Start
2. Create a file index.dat to store index terms and record number to direct towards actual records in records.dat file
3. Create a file records.dat to store actual records
4. Create index – Scan records.dat file and select all distinct records to enter search key value in index.dat file
5. Write search keys and respective record number in index.dat file
6. Ask user to enter data to search for.
7. Check for that search key in index.dat file
8. If search key is found in index.dat file retrieve it's data from records.dat file and display.
9. Else display message “No Record Found”.
10. Stop.