CS562: Database Management Systems II Notes

Steven DeFalco

Spring 2024

Contents

1 Indexing and Hashing **2**
1.1 Index Updates 3
1.1.1 Deletion 3
1.1.2 Insetion 3

1 Indexing and Hashing

Indexing methods are used to speed up access to desired data. A ***search key*** is an attribute used to look up records in a file. An ***index file*** consists of records (called **index entries**) of the form. Index files are typically much smaller than the original file. There are two basic kinds of indices:

- **Ordered indices**: search keys are stored in sorted order
- **Hash indices**: search keys are distributed uniformly across *buckets* using a *hash function*

Remark When *optimizing* in database queries, the goal is to have a small search space.

In an ***ordered index***, index entries are stored sorted on the search key value. In a sequentially ordered file, the ***primary index*** is the index whose search key specifies the sequential order of the file; the search key of a primary index is usually but not necessarily the primary key. The ***secondary index*** is an index whose key specifies an order different from the sequential order of the file (also called non-clustering index). An ***index-sequential file*** is an ordered sequential file with a primary index.
A ***dense index*** is where an index record appears for every search-key value in the file. ***Sparse index*** contains index records for only some search-key values. To locate a record with search-key value K we:

1. Find index record with largest search-key value $< K$
2. Search file sequentially starting at the record to which the index record points

Remark Sparse index must be primary index

Compared to dense indices, sparse indices have less space and less maintenance overhead for insertions and deletions. However, they are generally slower than dense index for locating records. a record with search-key value K we:

1. Find index record with largest search-key value $< K$
2. Search file sequentially starting at the record to which the index record points

Remark Sparse index must be primary index

Compared to dense indices, sparse indices have less space and less maintenance overhead for insertions and deletions. However, they are generally slower than dense index for locating records.

If primary index does not fit in memory, access becomes expensive. A solution to this is to treat primary index kept on disk as a sequential file and construct

a sparse index on it. The outer index is a sparse index of primary index. The inner index is the primary index file. If even out index is too large to fit in main memory, yet another level of index can be created and so on. Indices at all levels must be updated on insertion or deletion from the file.

1.1 Index Updates

1.1.1 Deletion

If the deleted record was the only record in the file with its particular search-key value, the seark-key is deleted from the index also. For singl e-level index deletion.

- Dense indices—deleetion of search-key
- Sparse indices—
  - If an entry for the search key exists in the index, it is deleted by replacing the entry in teh index with the next search-key value in the file
  - If the next search-key value already has an index entry, the entry is deleted instead of being replaced

1.1.2 Insetion

In single-level index insertion:

- Perform a lookup using the searh-key value appearing in the record to be inserted.
- Dense indices—if the search-key value does not appear in the index, insert it
- sparse indices—if index stores an entry for each block of the file, no chnage needs to be made to the index unless a new block is created. If a new block is created, the first search-key value appearing in the new block is inserted into the index.

In multilevel insertion, algorithms are simple extensiosn of the single-level algorithms.
Indices offer substantial benefits when searching for records, but updating indices imposes overhead on database modification—when a file is modified, every index on the file must be updated. Sequential scan using primary index is efficient, but a sequential scan using a second index is expensive.