Assignment No: 2

Title: Retrieval of documents using inverted files.

Problem Definition:

Implement a program for retrieval of documents using inverted files.

Outcome:

Students will be able to,

1. Apply various tools and techniques for information retrieval and web mining .
2. Evaluate and analyze retrieved information.

Theory:

Introduction of Inverted files.

Inverted files allow fast search for statistics related to the distinct words found in a text. They are projected for using words as the search unit, which restricts their use in applications where words are not clearly defined or in applications where the system does not use words as the search unit. In information retrieval, the inverted file theory, also known as an inverted index or inverted file, is a fundamental concept used to efficiently retrieve documents or records that contain specific words or terms. It's a data structure that maps terms to their corresponding documents or locations. Inverted files are a crucial component of modern information retrieval systems, enabling fast and efficient searching through large document collections, such as web pages, books, or databases. They greatly reduce the time and resources needed to locate specific information within these collections.

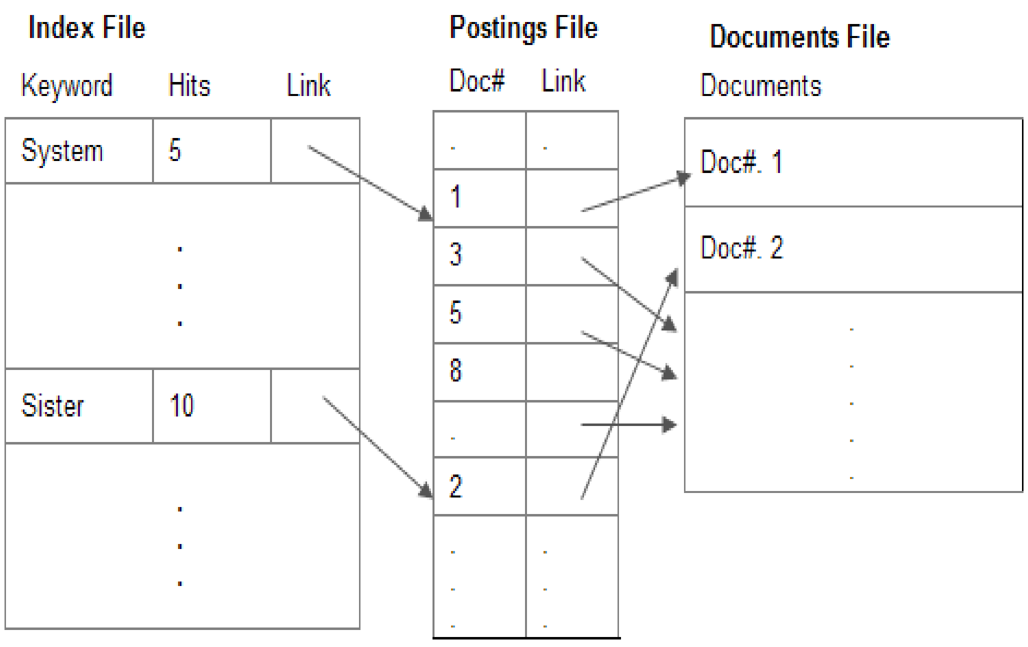

Fig : Inverted Index Files

Here's how it works:

Indexing: During the indexing process, documents are parsed and tokenized into words or terms. For each unique term in the document collection, an entry in the inverted file is created. This entry typically contains information such as the term itself, and a list of document IDs or pointers where the term occurs.

Term Frequencies: In addition to document IDs, inverted files often store information about the frequency of each term within a document. This can be used for ranking search results.

Query Processing: When a user enters a query, it's processed to identify the search terms. The system then looks up these terms in the inverted file, retrieving a list of documents containing each term.

Ranking: After obtaining the lists of documents, the system can rank them using various algorithms (e.g., TF-IDF, BM25) to present the most relevant results to the user.

Create and use an inverted file for information retrieval:

1. Document Collection:

Start with a collection of documents you want to make searchable, such as a set of web pages, books, articles, or any text-based content.

2. Text Preprocessing:

Preprocess the documents to prepare them for indexing. This includes steps like tokenization, stemming, and removing stop words. Tokenization breaks text into individual words or terms, stemming reduces words to their root form (e.g., "running" becomes "run"), and stop words (common words like "the," "and," "in") are removed.

3. Inverted File Construction:

For each term in the preprocessed documents, create an entry in the inverted file.

4. Sorting:

Sort the inverted file by terms. This allows for efficient searching because you can use techniques like binary search to locate terms quickly.

5. Query Processing:

When a user submits a query, the query terms are processed similarly to how the documents were preprocessed. This ensures that the query terms match the terms in the inverted file.

6. Retrieval:

For each query term, look up its entry in the inverted file to retrieve the list of documents where it appears.

Conclusion:-

we have to implemented inverted indexes remain an integral part of information retrieval systems, enabling quick and accurate searching across a wide array of applications, from web search engines to database systems.

Performance & Understanding	**Innovation**	**Timely Completion**	**Total**	**Sign & Date**
3	**1**	**1**	**5**	