JDBM Database Scheme of the Indexer

1. PageID – URL Mapping Tables:

PageIDtoURL(PageID, URL)

A mapping table for the interchange of PageID and URL. The key is PageID. The PageID is an index provided by the Spider, starting from 0.

Key	Data Type	Description
PageID	String	The PageID of a web page
Value	Data Type	Description
URL	String	The URL of a web page

URLtoPageID(URL, PageID)

A mapping table for the interchange of PageID and URL. The key is URL.

Key	Data Type	Description
URL	String	The URL of a web page
Value	Data Type	Description
PageID	String	The PageID of a web page

For simplicity of storing the URL in other databases, an ID is given to the page and recorded on this mapping table. Both directions are recorded, for efficiency of retrieval.

2. Table for last modified time of each web page:

PageIDtoTime(PageID, LastModifiedTime)

A table for recording the last modified time of a Page when fetching a page. The key is PageID. If the page is not fetched yet, the initial value of last modified time would be (Thu Jan 01 08:00:00 HKT 1970).

Key	Data Type	Description
PageID	String	The PageID of a web page
Value	Data Type	Description
LastModifiedTime	String	The last modified time of a web page Format: EEE MMM dd HH:mm:ss zzz yyyy Example: Thu Jun 16 16:47:39 HKT 2022

This table is for recording the last modified time of the pages fetched. When the Spider has fetched that page before, and the page is not modified after that, the last modified time from the page and from the table would be the same. Then the Spider will skip this page this time, as it is not modified. This can help handle the cyclic links.

3. WordID – Word Mapping Tables:

WordIDtoWord(WordID, Word)

A mapping table for the interchange of WordID and Word. The key is WordID. The WordID is an index provided by the Spider, staring from 0.

Key	Data Type	Description
WordID	String	The WordID of a word
Value	Data Type	Description
Word	String	The word

WordtoWordID(Word, WordID)

A mapping table for the interchange of WordID and Word. The key is Word.

Key	Data Type	Description
Word	String	The word
Value	Data Type	Description
WordID	String	The WordID of a word

For simplicity of storing the word in other databases, an ID is given to the word and recorded on this mapping table. Both directions are recorded, for efficiency of retrieval.

4. Forward and Inverted Indexes for storing Parent – Child relationship:

ParenttoChild(ParentPageID, ChildPageID)

A forward index for storing the Parent-Child relationship. The key is Parent's PageID.

Key	Data Type	Description
Parent's PageID	String	The PageID of the parent page
Value	Data Type	Description
Child's PageID	String	The PageID of the child page Format: ChildPageID1;ChildPageID2;ChildPageID3;…

ChildtoParent(ChildPageID, ParentPageID)

A backward index for storing the Parent-Child relationship. The key is Child's PageID.

Key	Data Type	Description
Child's PageID	String	The PageID of the child page
Value	Data Type	Description
Parent's PageID	String	The PageID of the parent page Format: ParentPageID1;ParentPageID2;ParentPageID3;…

These tables store the Parent – Child relationships. Both directions of retrieval are recorded, for efficiency.

5. Forward and Inverted Indexes for page and word on page:

PageIDtoWordID(PageID, WordID)

A forward index for storing words crawled from the page. The key is PageID. The frequency of each word is also recorded.

Key	Data Type	Description
PageID	String	The PageID of a web page
Value	Data Type	Description
WordID	String	The WordID of a word on the page Format: word1 freq1;word2 freq2;word3 freq3;…

WordIDtoPageID(WordID, PageID)

An inverted index for storing pages that contains the word. The key is WordID. The frequency of each word on that page is also recorded.

Key	Data Type	Description
WordID	String	The WordID of a word
Value	Data Type	Description
PageID	String	The PageID of a web page that contain the word Format: page1 freq1;page2 freq2;page3 freq3;…

These tables store the words crawled from the web pages. Both directions of retrieval are recorded, for efficiency.

Others:

1. If the title cannot be extracted, “No Title” would be shown, dealing with the certification error.
2. If the last modified time cannot be extracted, the time would be the current time.
3. As most of the content-length cannot be extracted (null), the number of characters of the HTML string would be used also.
4. If the HTML length cannot be extracted also (null), the number of words extracted from the page can be considered.
5. If the page is visited this round (one execution of the program), it will not be visited again, no matter what the new last modified time is, for preventing the cyclic link problem.
6. Only pages fetched this round (this execution) will be shown by the program. Pages stored in the database but not updated this round will not be shown.