**Phase 1 RocksDB Database Schema**

Group 19

|  |  |
| --- | --- |
| **RocksDB Name** | **Mapping** |
| WordToID | word -> wordID   * word: String * wordID: int * A keyword is converted into the corresponding wordID by the order of appearance from the indexer.   Explanation/Notes:   * Before a sentence is processed, it will first be tokenized to get a list of words so that phrases can be supported. * The word inserted into this database is not the original keyword in the document but the stemmed version. * The same tokenizer and stemming algorithm will be used for queries to ensure finding correct wordID. |
| IDToWord | wordID -> word   * wordID: int * word: String * Given a specific wordID, the corresponding word is returned. |
| URLToID | URL -> pageID   * URL: String * pageID: int * A URL is converted into the corresponding pageID by the order of appearance from the indexer. |
| IDToPageInfo | pageID -> page<page title, URL, last modified date, size>   * pageID: int * page: PageInfo   + PageInfo is a class which contains:   + title: String   + url: String   + modDate: Date   + size: int * Given a pageID, the corresponding PageInfo is returned.   Explanation/Notes:   * The PageInfo class is used to store the information of a web page. Accessing these information directly from pageID is easy by using this database. |
| PageInfoToID | page<page title, URL, last modified date, size> -> pageID   * page: PageInfo   + PageInfo is a class which contains:   + title: String   + url: String   + modDate: Date   + size: int * pageID: int * Given a specific PageInfo, the corresponding pageID is returned. |
| InvertedIndex | wordID -> {pageID, <word position>}   * wordID: int * pageMap: HashMap<Integer, String>   + pageMap uses pageID as the key. The value corresponding to the key is the positions of the wordID in this specific pageID. <word position> is represented using a String containing the positions separated by comma. * Given a specific wordID, the corresponding hashmap of pages and their positions contained in the page is returned. |
| ForwardIndex | pageID -> {wordID, <word position>}   * pageID: int * wordMap: HashMap<Integer, String>   + wordMap uses wordID as the key. The value corresponding to the key is the positions of the specific wordID in this pageID. <word position> is represented using a String containing the positions separated by comma. * Given a specific pageID, the corresponding hashmap of words and their positions contained in the page is returned.   Explanation/Notes:  Store the words appearing in a page along with their positions so that searching for phrases will be easier. |