**Database Schema Diagram**

**Explanation of the Design**

1. **Pages Table**
   * **Purpose:** Stores metadata for each crawled page.
   * **Columns:**
     + page\_id: A unique identifier for each page.
     + title: The page title (extracted from the HTML).
     + body: Text content of the page (used for potential full-document display or re-indexing).
     + url: The unique URL of the page; this is critical for preventing duplicate fetches.
     + last\_modified: The last modification date as obtained from the HTTP headers.
     + size: The size of the page in bytes.
2. **Words Table**
   * **Purpose:** Maintains a unique list of words (keywords) extracted from both page titles and bodies.
   * **Columns:**
     + word\_id: A unique identifier for the word.
     + word: The keyword.
3. **Forward Index Tables (Body and Title)**
   * **Purpose:**
     + These tables link each word occurrence to a page, along with a frequency count within that page.
     + Having two separate tables (one for the body and one for the title) allows you to favor title matches during query processing.
   * **Columns:**
     + word\_id: References the word in the *words* table.
     + page\_id: References the page in the *pages* table.
     + frequency: The number of occurrences of that word in the specific context (body or title).
4. **Inverted Index Tables (Body and Title)**
   * **Purpose:**
     + These tables aggregate the total frequency of each word across all pages, which is useful for quickly computing significance during ranking.
   * **Columns:**
     + word\_id: Identifies the word and is the primary key.
     + page\_frequency: The summed frequency from the forward indexes.
5. **Parent-Child Links Table**
   * **Purpose:**
     + Captures the hierarchical relationship among pages by linking parents with their child pages.
     + This structure is utilized both for constructing the navigation graph (or sitemap) and later for search result displays.
   * **Columns:**
     + parent\_id: Corresponds to the parent page identifier.
     + child\_id: Corresponds to the child page identifier.

**Mapping Table for URL ⟺ Page ID**

**Explanation:**

* The **pages** table acts as a mapping table where each unique URL is associated with a unique numeric identifier (page\_id).
* The URL is marked as unique and not null, ensuring that each page is only stored once.
* Additional metadata (title, body, last\_modified, size) can be stored here, but the primary purpose for our mapping is the url and page\_id relationship.

**Mapping Table for Word ⟺ Word ID**

**Explanation:**

* The **words** table is designed to ensure that each unique word is assigned a unique numeric identifier (word\_id).
* The word column is unique and not null, which prevents duplicate entries.
* This mapping enables efficient storage and lookup when building both the forward and inverted indexes.

**Overview of the Functions and Their Requirements**

* **Crawler Function:**  
  The crawler fetches web pages using BFS, extracts content (title, body, and hyperlinks), and captures metadata (e.g., URL, last modification date, size). It must also maintain parent-child relationships between pages to record the link structure.
* **Indexer Function:**  
  The indexer processes page contents to:
  + Remove stopwords.
  + Index words separately for page titles and bodies to support relevance ranking.
  + Build both forward and inverted indexes that record word frequencies per page and across pages.
  + Map human-readable URLs and words to internal IDs for efficient storage and lookup.