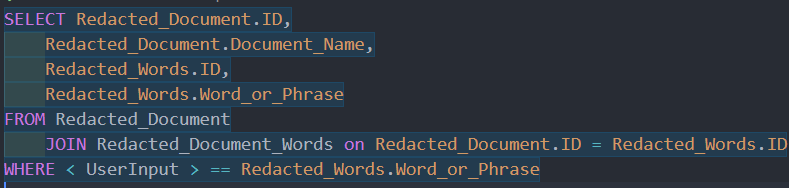


**How the data should be stored:**

The data should be stored in the MySQL database with the format diagramed in the above image with the following tables. The first table should contain the redacted documents. This table would contain all the paths for each document and their assigned id. There should also be a table in the database that contains the redacted words along with an assigned id. This table would have all the words and phrases being hashed, which would ensure security. This would mean that in the event of a data leak or hack that the document cannot have their redacted words guessed based on the information stored in the database. The final table would need to be the join table for storing both the assigned ids mention previously for the redacted documents and redacted words tables. This table would be the relationship between the document and the words.

**How the data should be made searchable:**

The Database should be searchable with SQL commands, with similar structure to the pseudo format in the image below. This pseudo is similar what we want the SQL command to look like where the command would return all the documents that contain the redacted words the user input.   


**How the data should be exposed to eternal consumers:**

* This database should be used as a sub system to a larger system where the CIA will be able to monitor and control access to the files based on credentials and clearance level it might be important to add another column to the table that has a clearance level for each redacted with that column being a foreign key to a table that has the keys stored.
* If this system is intended to be a stand-alone system without user validation to the database, then some other type of security should be implemented. The user should use a program, or a webpage and the server should block outside network connections or only allow connections from a VPN or some other type of security.
* The program used by the consumer should send the keywords from the user over a secure connection either HTTPS or some other secure connection protocol. When the server receives the request, it should then hash the keyword from there it can run the search command to gather all the document names from the database some data manipulation will be involved here either on the user side or server side to separate the path from the name of the file so that the file names can be displayed to the user. All these requests should be done over an API or multiple API’s. once the user decides on the document, they want to view they should be able to click on them and the server should use the path and send back the document they want to view.
* The UI that should be used is more difficult to figure out however I believe a good approach would be to use a textbox that allows you to free type any keyword and phrases the user wants to search by once the user has inputted one keyword or phrase then they should be allowed to hit enter this should create a box on the page with an ‘X’ on it the box should contain the keyword they want to search by with the ‘X’ allowing them to remove the textbox clearing the corresponding keyword from the search. Once they have finished adding all the keywords then they can click a search, and this will send a request to the API for the information matching the parameters.