

# Digital Content retrieval - mod B

FEDERICO CIGNOLI ,Mat. 544952

March 2024

## Contents

<b>1</b>	<b>DCR-B part 1</b>	<b>2</b>
1.1	Introduction . . . . .	2
1.2	Tools used . . . . .	2
1.3	Project specification . . . . .	3
1.4	Project delivery . . . . .	5
1.4.1	The schema of the database, in SQL ddl . . . . .	5
1.4.2	The developed code for loading the database . . . . .	6
1.4.3	The developed code for the search engine . . . . .	8
1.4.4	A listing of the whole DCRB directory and sub-directories	11
1.4.5	A snapshot of the search for a “non existing” string . . .	14
1.4.6	A snapshot of the search for a string matching at least two file names, but not found in any searchable file . . .	14
1.4.7	A snapshot of the search for a string matching at least one file name and contained in at least one searchable file, with the counts of the occurrences in the file(s) . . .	16
1.4.8	A snapshot of the search for a string that does not match a file name but is found in at least one searchable file, with the counts of the occurrences in the file(s) . . . . .	18
1.5	Conclusion part1 . . . . .	21

# **1 DCR-B part 1**

## **1.1 Introduction**

In this section, I will briefly discuss the contents of this report.

This report covers the first part of the Digital Content Retrieval - mod B project. The overall task was to develop a search feature for a specific subtree within a local file system.

Additionally, this university project was created during laboratory hours and beyond, following step-by-step all the requirements given during the lectures.

## **1.2 Tools used**

In this project I used lots of tools, software, online services, operating system:

- MySQL Workbench;
- Visual Studio Code;
- GitHub;
- LucidChart;
- Wikipedia;
- Windows 11 pro.

### 1.3 Project specification

In this project, I developed a comprehensive search function for a specific subtree within a local file system. The main task was to create a search field that accepted a string of characters. A match could be found if the string matched the name of a file, the name of a folder (either partially or completely), or a string contained within the file itself.

The target subtree had to meet certain criteria: it needed to have a depth of at least six levels and contain a specific subtree named “DCRB” with a minimum depth of four levels. Within the DCRB subtree, I ensured that it hosted at least fifty files downloaded from Wikipedia, distributed over at least four levels with a minimum of one file per level. My subtree contains around 100 files (mostly HTML) about food around the world. The “food” folder includes two main subfolders named “east food” and “west food,” each with two more levels of depth. All of this is described in more detail in the next section of this report.

Regarding table creation and data insertion, I developed a Python script to analyze a directory, gather information about files and folders, and store it in a MySQL database. The `list_files_and_folders` function scans the specified root directory, creating a list with the path, type (folder or file), size, name, and extension of each file. Once the connection to the MySQL database is established, the `file_info` table is created to store these attributes, and the collected data is inserted.

Next, the `secondary` table is created to contain the textual content of the files. The `read_file_content` function reads the content of each file, which is then inserted into the `secondary` table linked by an ID. Foreign key checks are managed to maintain referential integrity. Finally, the changes are committed, and the connection is closed.

For the search functionality, I developed a Python script (`search.py`) that returns several key pieces of information. The search result for a string includes:

- the name of the file containing the input string;
- the full file path;
- the file ID;
- the number of occurrences of the word in the file text.

Since I didn’t clean the HTML files (i.e., keeping only the text part), I modified my Python code to count only the occurrences of the string present as text in the file. Occurrences within HTML tags (e.g., `<link>`) are not counted. If the searched string is present in the file but not as text, the file will still appear in the terminal output but with “0” occurrences. This helps identify that the searched word is present in that file, but not as part of the text. A separate counter prints the number of occurrences of the word as a file or folder name, along with the details of the corresponding files and folders.

For data persistence, I chose a storage engine (MySQL Workbench) that allowed indexing of critical fields such as the full file path, file name, type,

and additional attributes like file size in bytes. The choice of the persistence engine was guided by the need to facilitate efficient search operations within the files. My choice of MySQL was also due to its familiarity, convenience, and user-friendly interface.

Finally, I created indexes in both tables. Indexes in MySQL Workbench tables are essential for improving the speed and efficiency of string searches. Indexes allow the database to quickly find matching rows without having to examine every row, reducing server workload and improving query performance, especially in large tables. Additionally, indexes facilitate sorting and grouping operations and help maintain high performance even as data volume increases.

Overall, the project required meticulous planning and execution to ensure all specified requirements were met, resulting in a robust and efficient search functionality for the local file system.

## 1.4 Project delivery

### 1.4.1 The schema of the database, in SQL ddl

```
-- "primary table"
CREATE TABLE IF NOT EXISTS file_info (
    id INT AUTO_INCREMENT PRIMARY KEY, -- Auto-incremented primary key
    path VARCHAR(255),                 -- File / Folder path
    type VARCHAR(255),                 -- Type of item (file or folder)
    size BIGINT,                       -- Size of the file (NULL for folders)
    file_name VARCHAR(255),             -- File name
    extension VARCHAR(255)              -- extension of the file (null if it is a folder)
);
```

Figure 1:

```
-- "secondary table"
CREATE TABLE IF NOT EXISTS secondary (
    file_id INT,                       -- Foreign key referencing the id column in the "file_info" table
    text LONGTEXT,                     -- Content of the file
    FOREIGN KEY (file_id) REFERENCES file_info(id) -- Foreign key constraint linking to "file_info" table
);
```

Figure 2:

Figure 1 and Figure 2 represents the two tables within the respective attributes.

According to my project perspective, implementing two tables with primary keys and foreign keys on MySQL offers various advantages: it ensures data consistency, effectively organizes data structure, establishes relationships between them, optimizes queries, and preserves data consistency. Ultimately, using these keys helps maintain reliable and well-structured database data.

### 1.4.2 The developed code for loading the database

```
1 # script.py
2 import os
3 import mysql.connector
4
5 def list_files_and_folders(root_dir):
6     data = []
7     for root, dirs, files in os.walk(root_dir):
8         for directory in dirs:
9             folder_path = os.path.join(root, directory)
10            data.append([folder_path, "Folder", None, None, None
11                        ])
12            for file in files:
13                file_path = os.path.join(root, file)
14                size = os.path.getsize(file_path)
15                file_name, file_extension = os.path.splitext(file)
16                data.append([file_path, "File", size, file_name,
17                            file_extension])
18
19     return data
20
21 def create_database(cursor, db_name):
22     cursor.execute(f"CREATE DATABASE IF NOT EXISTS {db_name}")
23
24 def create_table(cursor):
25     cursor.execute("""
26         CREATE TABLE IF NOT EXISTS file_info (
27             id INT AUTO_INCREMENT PRIMARY KEY,
28             path VARCHAR(255),
29             type VARCHAR(255),
30             size BIGINT,
31             file_name VARCHAR(255),
32             extension VARCHAR(255)
33         )
34     """)
35
36 def create_secondary_table(cursor):
37     cursor.execute("""
38         CREATE TABLE IF NOT EXISTS secondary (
39             file_id INT,
40             text LONGTEXT,
41             FOREIGN KEY (file_id) REFERENCES file_info(id)
42         )
43     """)
44
45 def insert_data(cursor, data):
46     sql = "INSERT INTO file_info (path, type, size, file_name,
47         extension) VALUES (%s, %s, %s, %s, %s)"
48     cursor.executemany(sql, data)
49
50 def read_file_content(file_path):
51     with open(file_path, 'r', encoding='latin-1') as file:
52         return file.read()
53
54 def connect_to_mysql():
55     return mysql.connector.connect(
56         host="localhost",
```

```

53         user="root",
54         password="Fedemysql99"
55     )
56
57 def connect_to_db(db_name):
58     return mysql.connector.connect(
59         host="localhost",
60         user="root",
61         password="Fedemysql99",
62         database=db_name
63     )
64
65 def populate_secondary_table(cursor, data):
66     file_count = 0
67     for row in data:
68         if row[1] == 'File':
69             try:
70                 file_content = read_file_content(row[0])
71                 cursor.execute("INSERT INTO secondary (file_id,
72                                     text) VALUES ((SELECT id FROM file_info
73                                     WHERE path = %s), %s)", (row[0],
74                                     file_content))
75                 file_count += 1
76                 if file_count % 10 == 0:
77                     connection.commit() # Commit ogni 10 file
78             except Exception as e:
79                 print(f"Errore durante la lettura del file {row
80                     [0]}: {e}")
81
82 # Connessione al server MySQL
83 try:
84     connection = connect_to_mysql()
85
86     if connection.is_connected():
87         cursor = connection.cursor()
88         print("Connessione al server MySQL riuscita")
89
90         # Creazione del database se non esiste
91         create_database(cursor, "dcrb")
92
93         # Connessione al database appena creato
94         connection.database = "dcrb"
95
96         # Crea le tabelle da zero
97         create_table(cursor)
98         create_secondary_table(cursor)
99
100        # Popola la tabella file_info
101        root_directory = r'C:\xampp\htdocs\DCRB project'
102        data = list_files_and_folders(root_directory)
103
104        insert_data(cursor, data)
105        connection.commit()
106
107        # Popola la colonna text nella tabella secondary
108        populate_secondary_table(cursor, data)
109        connection.commit()

```

```

106
107         print("Tabelle 'file_info' e 'secondary' create e dati
            inseriti correttamente")
108
109 except mysql.connector.Error as e:
110     print("Errore durante la connessione al server MySQL:", e)
111
112 finally:
113     if connection.is_connected():
114         cursor.close()
115         connection.close()

```

This Python code performs various operations: it explores a specified directory and collects information about the files and folders within it, creates a MySQL database and its related tables, inserts the collected data into the database, reads the content of a file, handles foreign key checks, drops an existing table, and manages the connection to the MySQL server and the specified database.

The data insertion is done in batches rather than all at once to reduce the load on the server, improve transaction handling, and optimize resources.

### 1.4.3 The developed code for the search engine

```

1 # search.py
2 import mysql.connector
3
4 def search_word_in_files(word):
5     try:
6         connection = mysql.connector.connect(
7             host="localhost",
8             user="root",
9             password="Fedemysql99",
10            database="dcrb"
11        )
12
13        if connection.is_connected():
14            cursor = connection.cursor()
15
16            # Query per cercare la parola nei testi dei file
17            text_query = """
18                SELECT file_info.file_name, file_info.extension,
19                    file_info.id, file_info.path,
20                    LENGTH(secondary.text) - LENGTH(REPLACE(
21                        secondary.text, %s, '')) AS
22                    text_word_count
23                FROM secondary
24                INNER JOIN file_info ON secondary.file_id =
25                    file_info.id
26                WHERE secondary.text LIKE %s
27            """
28            search_word = '%' + word + '%'
29            cursor.execute(text_query, (word, search_word))
30
31            text_results = cursor.fetchall()

```



```

28
29     # Query per cercare la parola nei nomi dei file e
        nei percorsi
30     name_query = """
31     SELECT file_name, path
32     FROM file_info
33     WHERE file_name LIKE %s OR path LIKE %s
34     """
35     name_search_word = '%' + word + '%'
36     cursor.execute(name_query, (name_search_word,
        name_search_word))
37
38     name_results = cursor.fetchall()
39
40     if text_results:
41         print("Risultati della ricerca per '{}'".format(
            word))
42         for result in text_results:
43             print("Nome file:", result[0] + result[1])
44             # Concatena nome del file ed estensione
45             print("Percorso:", result[3]) # Stampa il
                percorso del file
46             print("ID file:", result[2])
47             print("Numero di occorrenze della parola nel
                testo del file:", result[4])
48             print("-" * 50)
49             print("Numero di occorrenze della parola come
                nome di file o cartella:", len(name_results)
                )
50             print("Dettagli dei file e delle cartelle
                corrispondenti:")
51             for name_result in name_results:
52                 print("Nome file/cartella:", name_result[0])
53                 print("Percorso:", name_result[1])
54                 print("-" * 50)
55             else:
56                 print("Nessun risultato trovato per '{}'".format(
                    word))
57
58     except mysql.connector.Error as e:
59         print("Errore durante la connessione al database MySQL:"
            , e)
60
61     finally:
62         if connection.is_connected():
63             cursor.close()
64             connection.close()
65             print("Connessione al database chiusa.")
66
67 # Esegui la funzione di ricerca
68 parola_da_cercare = input("Inserisci la parola da cercare nei
    file: ")
search_word_in_files(parola_da_cercare)

```

This Python script searches for a specified word within files stored in a MySQL database. It connects to the MySQL database, performs two queries: one to search for the word within the text content of files, and another to search for the word within file names and paths. It then prints the results, including the file name, path, file ID, and the number of occurrences of the word within the file text. Additionally, it prints details of files and folders containing the word in their names. If no results are found, it notifies the user accordingly.

The script handles exceptions related to database connection errors and ensures the connection is closed properly after execution. Finally, it prompts the user to input the word to search for and executes the search function.

#### 1.4.4 A listing of the whole DCRB directory and sub-directories

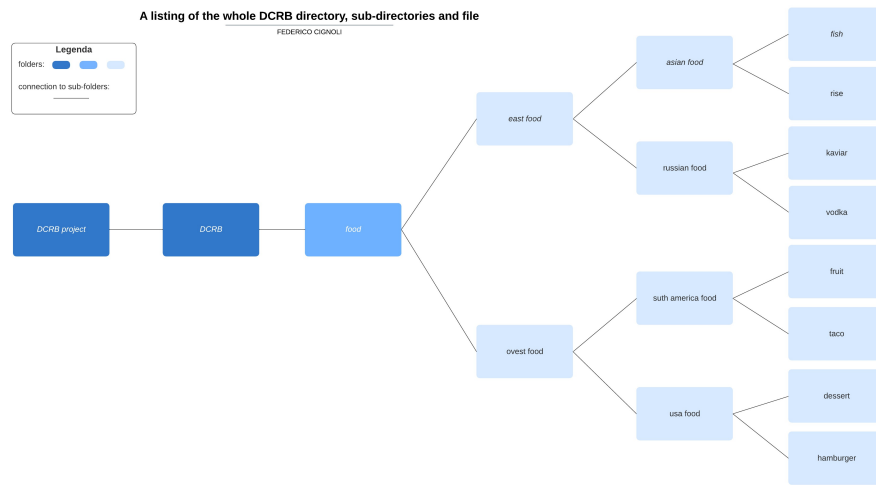


Figure 3:

Figure 3 shows all the tree that there are under DCRB.

```

1 Current directory: C:\xampp\htdocs\DCRB project
2 Subdirectories: ['.vscode', 'DCRB']
3 Files: ['cleaner.php', 'cleaner.py', 'converter.py', '
  preprocessing.py', 'script.py', 'script2.py', 'script3-1.py',
  'script3.py', 'script4.py', 'search.php', 'search.py', '
  tree-scanner.py']
4 -----
5 Current directory: C:\xampp\htdocs\DCRB project\.vscode
6 Subdirectories: []
7 Files: ['launch.json']
8 -----
9 Current directory: C:\xampp\htdocs\DCRB project\DCRB
10 Subdirectories: ['food']
11 Files: ['attr.txt', 'docs.txt', 'file_info.csv', 'output.txt', '
  puntoG.txt', 'puntoH.txt']
12 -----
13 Current directory: C:\xampp\htdocs\DCRB project\DCRB\food
14 Subdirectories: ['east food', 'ovest food']
  
```

```

15 Files: ['Food history.html', 'Food history2.html', 'Food
    history3.html']
16 -----
17 Current directory: C:\xampp\htdocs\DCRB project\DCRB\food\east
    food
18 Subdirectories: ['asian food', 'russian food']
19 Files: ['ListofMiddleEasternDishes.html', 'MiddleEasterncuisine.
    html']
20 -----
21 Current directory: C:\xampp\htdocs\DCRB project\DCRB\food\east
    food\asian food
22 Subdirectories: ['fish', 'rise']
23 Files: ['asianfood.html']
24 -----
25 Current directory: C:\xampp\htdocs\DCRB project\DCRB\food\east
    food\asian food\fish
26 Subdirectories: []
27 Files: ['Asian carp.html', 'Eurasian carp.html', 'farmingfish.
    html', 'Fish.html', 'Grass carp.html', 'Japanese amberjack.
    html', 'Japanese anchovy.html', 'Japanese angelfish.html', '
    Japanese angelshark.html', 'Japanese catshark.html', '
    Japanese dragonet.html', 'Japanese eel.html', 'Japanese
    fluvial sculpin.html', 'Japanese gissu.html', 'Japanese jack
    mackerel.html', 'Japanese pugnose grenadier.html', '
    Japanese roughshark.html', 'Japanese sawshark.html', '
    Japanese sea bass.html', 'japanesefish.html', 'saucfish.
    html']
28 -----
29 Current directory: C:\xampp\htdocs\DCRB project\DCRB\food\east
    food\asian food\rise
30 Subdirectories: []
31 Files: ['glutinousrice.html', 'japanerice.html']
32 -----
33 Current directory: C:\xampp\htdocs\DCRB project\DCRB\food\east
    food\russian food
34 Subdirectories: ['kaviar', 'vodka']
35 Files: ['peasant.html', 'russianfood.html']
36 -----
37 Current directory: C:\xampp\htdocs\DCRB project\DCRB\food\east
    food\russian food\kaviar
38 Subdirectories: []
39 Files: ['Belugacaviar.html', 'Caviar.html', 'Kalleskaviar.html',
    'Redcaviar.html', 'Russiancaviarhouse.html']
40 -----
41 Current directory: C:\xampp\htdocs\DCRB project\DCRB\food\east
    food\russian food\vodka
42 Subdirectories: []
43 Files: ['Alcohol.html', 'Chopin.html', 'Prohibition.html', '
    vodka.html', 'Wyborowa.html', 'Zubrowka.html']
44 -----
45 Current directory: C:\xampp\htdocs\DCRB project\DCRB\food\ovest
    food
46 Subdirectories: ['suth america food', 'usa food']
47 Files: []
48 -----
49 Current directory: C:\xampp\htdocs\DCRB project\DCRB\food\ovest
    food\suth america food

```

```

50 Subdirectories: ['fruit', 'taco']
51 Files: ['Chiquita.html', 'United Fruit Company.html']
52 -----
53 Current directory: C:\xampp\htdocs\DCRB project\DCRB\food\ovest
   food\suth america food\fruit
54 Subdirectories: []
55 Files: ['Coconut.html', 'Durian.html', 'Guava.html', 'Mammea
   americana.html', 'Mango.html', 'Morisonia speciosa.html', '
   Myrciaria floribunda.html', 'Passiflora edulis.html', '
   Peanut.html', 'Pineapple.html', 'Tomato.html']
56 -----
57 Current directory: C:\xampp\htdocs\DCRB project\DCRB\food\ovest
   food\suth america food\taco
58 Subdirectories: []
59 Files: ['Burrito.html', 'Taco.html', 'Taquito.html', 'Tortilla.
   html']
60 -----
61 Current directory: C:\xampp\htdocs\DCRB project\DCRB\food\ovest
   food\usa food
62 Subdirectories: ['dessert', 'hamburger']
63 Files: ['American cuisine.html']
64 -----
65 Current directory: C:\xampp\htdocs\DCRB project\DCRB\food\ovest
   food\usa food\dessert
66 Subdirectories: []
67 Files: ['Angel food cake.html', 'Apple crisp.html', 'Banana
   split.html', 'Bananas Foster.html', 'Blackout cake.html', '
   Boston cream pie.html', 'Chocolate brownie.html', 'Coconut
   cake.html', 'Cupcake.html', 'List of American desserts.html'
   ]
68 -----
69 Current directory: C:\xampp\htdocs\DCRB project\DCRB\food\ovest
   food\usa food\hamburger
70 Subdirectories: []
71 Files: ['50_50 hamburger.html', 'Buffalo burger.html', '
   Cheeseburger.html', 'Chili burger.html', 'Crown Burgers.html
   ', 'Hamburger.html', 'History of the hamburger.html', 'Patty
   melt.html', 'Slider.html', 'Veggie burger.html']
72 -----

```

Here is the output showing the current directory, the subdirectories that can be navigated, and the files in the current directory.

#### 1.4.5 A snapshot of the search for a “non existing” string

```
1 $ C:/Users/feder/AppData/Local/Programs/Python/Python312/  
python.exe "c:/xampp/htdocs/DCRB project/search.py"  
2 Inserisci la parola da cercare nei file: tongs  
3 Nessun risultato trovato per 'tongs'.  
4  
5 Connessione al database chiusa.
```

In line 2 we can see that the string given in input is "tongs" but the search give to us no results.

#### 1.4.6 A snapshot of the search for a string matching at least two file names, but not found in any searchable file

```
1 feder@workstation MINGW64 /c/xampp/htdocs/DCRB project  
2 $ C:/Users/feder/AppData/Local/Programs/Python/Python312/python.  
exe "c:/xampp/htdocs/DCRB project/search.py"  
3 Inserisci la parola da cercare nei file: food history  
4 Risultati della ricerca per 'food history':  
5 Nome file: docs.txt  
6 Percorso: C:\xampp\htdocs\DCRB\docs.txt  
7 ID file: 7  
8 Numero di occorrenze della parola nel testo del file: 0  
9 -----  
10 Nome file: Food history.html  
11 Percorso: C:\xampp\htdocs\DCRB\progetto\food\Food history.html  
12 ID file: 22  
13 Numero di occorrenze della parola nel testo del file: 0  
14 -----  
15 Nome file: Food history2.html  
16 Percorso: C:\xampp\htdocs\DCRB\progetto\food\Food history2.html  
17 ID file: 23  
18 Numero di occorrenze della parola nel testo del file: 0  
19 -----  
20 Nome file: ListofMiddleEasternDishes.html  
21 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\  
ListofMiddleEasternDishes.html  
22 ID file: 27  
23 Numero di occorrenze della parola nel testo del file: 0  
24 -----  
25 Nome file: MiddleEasterncuisine.html  
26 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\  
MiddleEasterncuisine.html  
27 ID file: 28  
28 Numero di occorrenze della parola nel testo del file: 0  
29 -----  
30 Nome file: asianfood.html  
31 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian  
food\asianfood.html  
32 ID file: 31  
33 Numero di occorrenze della parola nel testo del file: 0  
34 -----  
35 Nome file: peasant.html  
36 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\russian  
food\peasant.html
```

```

37 ID file: 57
38 Numero di occorrenze della parola nel testo del file: 0
39 -----
40 Nome file: russianfood.html
41 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\russian
    food\russianfood.html
42 ID file: 58
43 Numero di occorrenze della parola nel testo del file: 0
44 -----
45 Nome file: American cuisine.html
46 Percorso: C:\xampp\htdocs\DCRB\progetto\food\ovest food\usa food
    \American cuisine.html
47 ID file: 93
48 Numero di occorrenze della parola nel testo del file: 0
49 -----
50 Nome file: Angel food cake.html
51 Percorso: C:\xampp\htdocs\DCRB\progetto\food\ovest food\usa food
    \dessert\Angel food cake.html
52 ID file: 94
53 Numero di occorrenze della parola nel testo del file: 0
54 -----
55 Nome file: History of the hamburger.html
56 Percorso: C:\xampp\htdocs\DCRB\progetto\food\ovest food\usa food
    \hamburger\History of the hamburger.html
57 ID file: 110
58 Numero di occorrenze della parola nel testo del file: 0
59 -----
60 Numero di occorrenze della parola come nome di file o cartella:
    2
61 Dettagli dei file e delle cartelle corrispondenti:
62 Nome file/cartella: Food history
63 Percorso: C:\xampp\htdocs\DCRB\progetto\food\Food history.html
64 -----
65 Nome file/cartella: Food history2
66 Percorso: C:\xampp\htdocs\DCRB\progetto\food\Food history2.html
67 -----
68 Connessione al database chiusa.

```

In line 2, we notice the execution of the file ‘search.py’. Following this, we see a list of files, and it is important to highlight that: in these files, the string “food history” is present, but not as plain text. It is likely within some specific HTML tag, which is irrelevant to us. Consequently, we see the file in the output, but the number of occurrences is zero. In line 60, we observe that the searched string appears as a “file name,” immediately followed by various details related to the match.

#### 1.4.7 A snapshot of the search for a string matching at least one file name and contained in at least one searchable file, with the counts of the occurrences in the file(s)

```
1 PS C:\xampp\htdocs\DCRB> & C:/Users/feder/AppData/Local/Programs
2 /Python/Python312/python.exe c:/xampp\htdocs\DCRB/search.py
3 Inserisci la parola da cercare nei file: shark
4 Risultati della ricerca per 'shark':
5 Nome file: docs.txt
6 Percorso: C:\xampp\htdocs\DCRB\docs.txt
7 ID file: 7
8 Numero di occorrenze della parola nel testo del file: 4485
9 -----
10 Nome file: file_info.csv
11 Percorso: C:\xampp\htdocs\DCRB\file_info.csv
12 ID file: 8
13 Numero di occorrenze della parola nel testo del file: 20
14 -----
15 Nome file: output.txt
16 Percorso: C:\xampp\htdocs\DCRB\output.txt
17 ID file: 9
18 Numero di occorrenze della parola nel testo del file: 20
19 -----
20 Nome file: Eurasian carp.html
21 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
   food\fish\Eurasian carp.html
22 ID file: 33
23 Numero di occorrenze della parola nel testo del file: 0
24 -----
25 Nome file: farmingfish.html
26 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
   food\fish\farmingfish.html
27 ID file: 34
28 Numero di occorrenze della parola nel testo del file: 30
29 -----
30 Nome file: Fish.html
31 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
   food\fish\Fish.html
32 ID file: 35
33 Numero di occorrenze della parola nel testo del file: 275
34 -----
35 Nome file: Grass carp.html
36 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
   food\fish\Grass carp.html
37 ID file: 36
38 Numero di occorrenze della parola nel testo del file: 0
39 -----
40 Nome file: Japanese angelshark.html
41 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
   food\fish\Japanese angelshark.html
42 ID file: 40
43 Numero di occorrenze della parola nel testo del file: 870
44 -----
45 Nome file: Japanese catshark.html
46 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
   food\fish\Japanese catshark.html
```



```

47 ID file: 41
48 Numero di occorrenze della parola nel testo del file: 270
49 -----
50 Nome file: Japanese jack mackerel.html
51 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
   food\fish\Japanese jack mackerel.html
52 ID file: 46
53 Numero di occorrenze della parola nel testo del file: 5
54 -----
55 Nome file: Japanese roughshark.html
56 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
   food\fish\Japanese roughshark.html
57 ID file: 48
58 Numero di occorrenze della parola nel testo del file: 6420
59 -----
60 Nome file: Japanese sawshark.html
61 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
   food\fish\Japanese sawshark.html
62 ID file: 49
63 Numero di occorrenze della parola nel testo del file: 6495
64 -----
65 Nome file: saucefish.html
66 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
   food\fish\saucefish.html
67 ID file: 52
68 Numero di occorrenze della parola nel testo del file: 0
69 -----
70 Nome file: Caviar.html
71 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\russian
   food\kaviar\Caviar.html
72 ID file: 60
73 Numero di occorrenze della parola nel testo del file: 0
74 -----
75 Nome file: Coconut.html
76 Percorso: C:\xampp\htdocs\DCRB\progetto\food\ovest food\suth
   america food\fruit\Coconut.html
77 ID file: 76
78 Numero di occorrenze della parola nel testo del file: 20
79 -----
80 Nome file: Hamburger.html
81 Percorso: C:\xampp\htdocs\DCRB\progetto\food\ovest food\usa food
   \hamburger\Hamburger.html
82 ID file: 109
83 Numero di occorrenze della parola nel testo del file: 25
84 -----
85 Nome file: Patty melt.html
86 Percorso: C:\xampp\htdocs\DCRB\progetto\food\ovest food\usa food
   \hamburger\Patty melt.html
87 ID file: 111
88 Numero di occorrenze della parola nel testo del file: 15
89 -----
90 Nome file: Slider.html
91 Percorso: C:\xampp\htdocs\DCRB\progetto\food\ovest food\usa food
   \hamburger\Slider.html
92 ID file: 112
93 Numero di occorrenze della parola nel testo del file: 15
94 -----

```

```

95 Numero di occorrenze della parola come nome di file o cartella:
96     4
97 Dettagli dei file e delle cartelle corrispondenti:
98 Nome file/cartella: Japanese angelshark
99 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
100     food\fish\Japanese angelshark.html
101 -----
102 Nome file/cartella: Japanese catshark
103 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
104     food\fish\Japanese catshark.html
105 -----
106 Nome file/cartella: Japanese roughshark
107 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
108     food\fish\Japanese roughshark.html
109 -----
110 Nome file/cartella: Japanese sawshark
111 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
112     food\fish\Japanese sawshark.html
113 -----
114 Connessione al database chiusa.
115 PS C:\xampp\htdocs\DCRB>

```

In this case, in line 4, the string is "shark" and the result show to us that this word is present in both case, as a string in the searchable file text and in the name of files

#### 1.4.8 A snapshot of the search for a string that does not match a file name but is found in at least one searchable file, with the counts of the occurrences in the file(s)

```

1 PS C:\xampp\htdocs\DCRB> & C:/Users/feder/AppData/Local/
2   Programs/Python/Python312/python.exe c:/xampp/htdocs/
3   DCRB/search.py
4 Inserisci la parola da cercare nei file: tuna
5 Risultati della ricerca per 'tuna':
6 Nome file: docs.txt
7 Percorso: C:\xampp\htdocs\DCRB\docs.txt
8 ID file: 7
9 Numero di occorrenze della parola nel testo del file: 88
10 -----
11 Nome file: Food history.html
12 Percorso: C:\xampp\htdocs\DCRB\progetto\food\Food history.html
13 ID file: 22
14 Numero di occorrenze della parola nel testo del file: 8
15 -----
16 Nome file: Food history2.html
17 Percorso: C:\xampp\htdocs\DCRB\progetto\food\Food history2.html
18 ID file: 23
19 Numero di occorrenze della parola nel testo del file: 0
20 -----
21 Nome file: Food hystory3.html
22 Percorso: C:\xampp\htdocs\DCRB\progetto\food\Food hystory3.html
23 ID file: 24
24 Numero di occorrenze della parola nel testo del file: 0
25 -----

```

```

24 Nome file: asianfood.html
25 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
   food\asianfood.html
26 ID file: 31
27 Numero di occorrenze della parola nel testo del file: 8
28 -----
29 Nome file: Eurasian carp.html
30 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
   food\fish\Eurasian carp.html
31 ID file: 33
32 Numero di occorrenze della parola nel testo del file: 48
33 -----
34 Nome file: farmingfish.html
35 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
   food\fish\farmingfish.html
36 ID file: 34
37 Numero di occorrenze della parola nel testo del file: 12
38 -----
39 Nome file: Fish.html
40 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
   food\fish\Fish.html
41 ID file: 35
42 Numero di occorrenze della parola nel testo del file: 96
43 -----
44 Nome file: Grass carp.html
45 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
   food\fish\Grass carp.html
46 ID file: 36
47 Numero di occorrenze della parola nel testo del file: 48
48 -----
49 Nome file: Japanese amberjack.html
50 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
   food\fish\Japanese amberjack.html
51 ID file: 37
52 Numero di occorrenze della parola nel testo del file: 8
53 -----
54 Nome file: Japanese jack mackerel.html
55 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
   food\fish\Japanese jack mackerel.html
56 ID file: 46
57 Numero di occorrenze della parola nel testo del file: 8
58 -----
59 Nome file: saucefish.html
60 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\asian
   food\fish\saucefish.html
61 ID file: 52
62 Numero di occorrenze della parola nel testo del file: 12
63 -----
64 Nome file: Caviar.html
65 Percorso: C:\xampp\htdocs\DCRB\progetto\food\east food\russian
   food\kaviar\Caviar.html
66 ID file: 60
67 Numero di occorrenze della parola nel testo del file: 12
68 -----
69 Nome file: Burrito.html
70 Percorso: C:\xampp\htdocs\DCRB\progetto\food\ovest food\south
   america food\taco\Burrito.html

```

```

71 ID file: 87
72 Numero di occorrenze della parola nel testo del file: 4
73 -----
74 Nome file: American cuisine.html
75 Percorso: C:\xampp\htdocs\DCRB\progetto\food\ovest food\usa food
   \American cuisine.html
76 ID file: 93
77 Numero di occorrenze della parola nel testo del file: 8
78 -----
79 Nome file: List of American desserts.html
80 Percorso: C:\xampp\htdocs\DCRB\progetto\food\ovest food\usa food
   \dessert\List of American desserts.html
81 ID file: 103
82 Numero di occorrenze della parola nel testo del file: 8
83 -----
84 Nome file: Hamburger.html
85 Percorso: C:\xampp\htdocs\DCRB\progetto\food\ovest food\usa food
   \hamburger\Hamburger.html
86 ID file: 109
87 Numero di occorrenze della parola nel testo del file: 0
88 -----
89 Nome file: Patty melt.html
90 Percorso: C:\xampp\htdocs\DCRB\progetto\food\ovest food\usa food
   \hamburger\Patty melt.html
91 ID file: 111
92 Numero di occorrenze della parola nel testo del file: 0
93 -----
94 Nome file: Slider.html
95 Percorso: C:\xampp\htdocs\DCRB\progetto\food\ovest food\usa food
   \hamburger\Slider.html
96 ID file: 112
97 Numero di occorrenze della parola nel testo del file: 0
98 -----
99 Numero di occorrenze della parola come nome di file o cartella:
   0
100 Dettagli dei file e delle cartelle corrispondenti:
101 Connessione al database chiusa.
102 PS C:\xampp\htdocs\DCRB>

```

As we can see here, the searched word is "tuna", and consequently, it is present within many HTML files but not as a file or folder name.

## 1.5 Conclusion part1

In conclusion, this project not only achieves the initial objectives but also paves the way for further developments. In fact, I continued a personal experiment, using the entire file system of my PC. I can therefore conclude that the strategy of using two tables proved very favorable in terms of data consistency, effective organization of the data structure, query optimization, etc. Moreover, the method of committing data not all at once, but in batches, was a good plan to ensure that the PC did not experience lag during execution.

Finally, it was an excellent project for experimenting with searching within a personally created database.