



# Digital Content Retrieval part 1

*Vito Giacalone (546646)*

May 21, 2024



UNIVERSITÀ DI PAVIA

# Repository Content

## Schema of the Database

The schema of the database is provided in **SQL DDL**. The following code defines the structure of the database:

```
1 CREATE TABLE file_info (
2     ID INT NOT NULL AUTO_INCREMENT,
3     Name VARCHAR(255) NOT NULL,
4     Path VARCHAR(" " + str(maxsize) + " "),
5     SizeInBytes BIGINT,
6     Type VARCHAR(" " + str(maxsize) + " "),
7     CreationTime DATETIME,
8     LastModifiedTime DATETIME,
9     Txt MEDIUMTEXT,
10    PRIMARY KEY(id)
11 )
```

Listing 1: DDL of the schema

The red characters contains **str(maxsize)**, because the query is submitted using the **mysql.connector** library of python, is an integer value that corresponds to the maximum size of the string treated by the operating system. I preferred to keep the value parametric in order to be adaptive to each system that runs the program.

## Developed Code for Loading the Database

The code for loading the database from source files is written using **Python** and the **mysql.connector** library in order to use **MySQL** syntax inside Python functions. An important strategy used to populate the database is the creation of batches used to change the commit mode of the DBMS. Instead to perform 1 insertion and 1 commit, thanks to the batch strategy the commit is done every **MAX\_BATCH\_SIZE** tuples, in order to speed up the bulk process. The batch solution has been also used to avoid the saturation of the RAM, for example when the script is used to index all the elements of the filesystem. Below the code:

```
1 #!/usr/bin/python3
2
3 ##### Vito Giacalone (546646) #####
4 ##### Digital content retrieval project #####
5 #####
6 #####
7
8
9
10 #####
11 ##### Database Bulk script #####
12 #####
13
14
15 #####
16 ##### Libraries #####
17 #####
18 #####
19 import os
20 import datetime
21 import mysql.connector
22
23
24 #####
25 ##### Database connection #####
26
27 def connect_db(host, user, pwd, db, auth):
28     db_conn = mysql.connector.connect(
29         host=host,
30         user=user,
31         password=pwd,
32         database=db,
33         auth_plugin=auth
34     )
35     return db_conn
36
37
38 ##### SQL DDL #####
39 ##### Definition and creation of the tables #####
40 # Drop off the table if it exists yet
41 # Setting of the charset for the table
42 # Commit
43 #####
44
45 def create_tables(db_connection, maxsize):
46     db_reference = db_connection.cursor()
47     db_reference.execute("DROP TABLE IF EXISTS file_info")
```

```

48     db_reference.execute("""
49         CREATE TABLE file_info (
50             ID INT NOT NULL AUTO_INCREMENT,
51             Name VARCHAR(255) NOT NULL,
52             Path VARCHAR(" + str(maxsize) + ""),
53             SizeInBytes BIGINT,
54             Type VARCHAR(" + str(maxsize) + ""),
55             CreationTime DATETIME,
56             LastModifiedTime DATETIME,
57             Txt MEDIUMTEXT,
58             PRIMARY KEY(id)
59         )
60     """
61     db_reference.execute("SET NAMES 'UTF8MB4'")
62     db_reference.execute("SET CHARACTER SET UTF8MB4")
63     db_connection.commit()
64
65
66
67 ##### Insertion of tuples into the table file_info #####
68 # This table contains informations about the considered file
69 # A list of tuples is given as input (info)
70 # Each tuple contains the necessary to fill the table
71 # Commit
72 #####
73
74 def insert_info_tuple(db_reference, info):
75     info_insert_query = """
76         INSERT INTO `file_info` (Name, Path, SizeInBytes, Type, CreationTime, LastModifiedTime, Txt)
77         VALUES (%s, %s, %s, %s, %s, %s, %s)
78     """
79     db_reference.executemany(info_insert_query, info)
80     db_connection.commit()
81
82
83 ##### Creation of index on column of file_content #####
84 # The index is created on Txt field of file_content table
85 # Commit
86 #####
87
88 def create_index(db_connection):
89     db_reference = db_connection.cursor()
90     db_reference.execute("CREATE INDEX idx_name ON file_info (Name);")
91     db_connection.commit()
92
93
94
95 ##### Remove char from string #####
96 # This function filters "special" characters from a string
97 # The string is typically the name of the file
98 #####
99
100 def remove_char_from_string(string):
101     for c in "#?$/%!\\"$*,:_-#":
102         string = string.replace(c, " ")
103     return string
104
105
106
107 ##### Extraction of information about a file from path #####
108 # The function returns:
109 # file_size
110 # file_name
111 # extension (mp3, html,...)
112 # creation_time
113 # last_modified_time
114 #####
115
116 def _stats_(path):
117     file_size = os.stat(path).st_size
118     file_name = remove_char_from_string(path.split("/")[-1]).lower()
119     extension = path.split(".")[-1].lower()
120     creation_time = datetime.datetime.fromtimestamp(os.path.getctime(path)).strftime('%Y-%m-%d %H:%M:%S')
121     last_modified_time = datetime.datetime.fromtimestamp(os.path.getmtime(path)).strftime('%Y-%m-%d %H:%M:%S')
122     return file_size, file_name, extension, creation_time, last_modified_time
123
124
125
126 ##### Function to check if the batch is full #####
127 # The function returns:
128 # an empty list if the batch is full

```

```

129 # the batch if is not already full
130 # the function also insert the tuples in the DB if the batch is full
131 # This type of implementation has been designed to avoid that the complete filling of the RAM memory
132 # The batch implementation provides that a constant number of tuples are in RAM
133 ######
134
135 def check_and_submit(batch, MAX_BATCH_SIZE, db_reference):
136     if len(batch) == MAX_BATCH_SIZE:
137         insert_info_tuple(db_reference, batch)
138         return []
139     else:
140         return batch
141
142
143
144 ######
145 # This function populate the database
146 # The function focus on the filetype "html"
147 # Each file is analyzed, but only html files are inserted into the DB
148 # The insertion of the data is done tuple by tuple: this type of insertion has been used
149 # because it requires a minimum quantity of memory.
150 # An eventual population of the DB starting from the root may saturate the RAM because
151 # the data structures become too big (In the case without batch)
152 # Thanks to the batch insertion a constant quantitative of ram is allocated
153 # This parameter is fixed inside the code
154 # A future update may provide to read the size of the batch from command line
155 #####
156
157 def populate_db(startpath, db_connection, MAX_BATCH_SIZE):
158     batch = []
159     supported_formats = ["html"]
160     db_reference = db_connection.cursor()
161     for root, dirs, files in os.walk(startpath):
162         for file in files:
163             if len(batch) < MAX_BATCH_SIZE:
164                 try:
165                     file_path = os.path.join(root, file)
166                     file_size, file_name, extension, creation_time, last_modified_time = _stats_(file_path)
167                     #print(file_size, file_name, extension, creation_time, last_modified_time)
168                     txt = ""
169                     if extension in supported_formats:
170                         with open(file_path, 'r', encoding='utf-8') as file:
171                             txt = str(file.read()).lower()
172                         batch.append((file_name, file_path, file_size, extension, creation_time,
173                           last_modified_time, txt))
174                     else:
175                         batch.append((file_name, file_path, file_size, "File", creation_time,
176                           last_modified_time, None))
177                 batch = check_and_submit(batch, MAX_BATCH_SIZE, db_reference)
178             except Exception as e:
179                 continue
180
181             for dir in dirs:
182                 if len(batch) < MAX_BATCH_SIZE:
183                     try:
184                         dir_path = os.path.join(root, dir)
185                         _, dir_name, _, creation_time, last_modified_time = _stats_(dir_path)
186                         batch.append((dir_name, dir_path, None, "Directory", creation_time,
187                           last_modified_time, None))
188                     batch = check_and_submit(batch, MAX_BATCH_SIZE, db_reference)
189                 except Exception as e:
190                     continue
191
192
193
194 #####
195 ######
196 # Please, use the character "/" if you are using this program on Linux or Macos system.
197 # Use "\\" otherwise
198 #####
199
200 db_connection = connect_db("localhost", "root", "GCLVTI99P27F061Y", "DCR", "mysql_native_password")
201 create_tables(db_connection, os.pathconf('/', os.pathconf_names['PC_PATH_MAX']))
202 MAX_BATCH_SIZE = 10
203 populate_db('/', db_connection, MAX_BATCH_SIZE)
204 create_index(db_connection)
205 db_connection.close()

```

Listing 2: Bulk Database Script Using Batch Loading

## Developed Code for the Search Engine

The search engine is **case insensitive** and implemented querying the database and returning results based on user input from standard input. Also here, a batches solution is used in order to avoid to fill completely the RAM. The produced output is saved into a new file, named **out.txt**. Here is the Python code for the search functionality:

```
1 #!/usr/bin/python3
2
3 ##### Vito Giacalone (546646) #####
4 ##### Digital content retrival project #####
5 #####
6 #####
7
8
9 #####
10 ##### Retrieval script #####
11 #####
12 #####
13
14
15 #####
16 ##### Libraries #####
17 #####
18 #####
19 import mysql.connector
20
21
22 #####
23 ##### Database connection #####
24
25 def connect_db(host, user, pwd, db, auth):
26     db_connection = mysql.connector.connect(
27         host=host,
28         user=user,
29         password=pwd,
30         database=db,
31         auth_plugin=auth
32     )
33     return db_connection
34
35
36
37 ##### Search into the content of the file and into the nam field of the file #####
38 # The parameter "keyword" is the word to retrieve inside the Txt field (html)
39 # and inside the name field
40 # Execute the query
41 #####
42
43 def search_query(db_reference, keyword):
44     query = """
45         SELECT ID, file_info.Path, file_info.Type, SUM(count)
46         FROM (SELECT DISTINCT ID, count
47                 FROM (SELECT ID, count
48                         FROM (SELECT ID, Name, ROUND ((LENGTH(Txt) - LENGTH( REPLACE ( Txt, \"\"\"+str(
49             keyword)+"""\", """)) / LENGTH(\""""+str(keyword)+"""\") ) AS count
50                         FROM file_info) AS A
51                         WHERE count > 0
52
53                         UNION
54
55                         SELECT ID, 0 AS count
56                         FROM file_info FORCE INDEX (idx_name)
57                         WHERE Name LIKE \"%\""+str(keyword)+"%\") AS B) AS C
58             NATURAL JOIN file_info
59             GROUP BY file_info.ID, file_info.NAME
60
61     db_reference.execute(query)
62     return db_reference
63
64
65 ##### Process the results of the DB and print them #####
66 # The parameter "keyword" is the word to retrieve inside the Txt field (html)
67 # and inside the name field
68 # The tuples are retrieved in batches from the cursor
69 # Print the results
70 # Close the reference to DB
71 #####
72
73 def search_name_and_occurrences(db_connection, keyword, batch_size):
74     db_reference = db_connection.cursor()
75     search_query(db_reference, keyword)
76     file_out = open("./out.txt", "w")
```

```

77     print("ID", "Path", "Type", "Occurrences Inside text", file=file_out)
78     flag = 0
79     while True:
80         out = db_reference.fetchmany(batch_size)
81         if not out:
82             if flag == 0:
83                 print("Keyword \'", keyword, "\' not found")
84                 break
85             break
86         for _list_ in out:
87             flag = 1
88             print('%d\t%s\t%s\t%d' % (_list_[0], _list_[1], _list_[2], _list_[3]), file=file_out)
89     file_out.close()
90     db_reference.close()
91
92
93
94 ##### Main #####
95 #####
96
97 db_connection = connect_db("localhost", "root", "GCLVTI99P27F061Y", "dcr", "mysql_native_password")
98 keyword = input("Insert the name or partial name to retrive: ")
99 MAX_BATCH_SIZE = 10
100 search_name_and_occurrences(db_connection, keyword, MAX_BATCH_SIZE)
101 db_connection.close()

```

Listing 3: Code of the search engine with batch solution

## DCRB Directory Listing

A listing of the whole DCRB directory and sub-directories is provided below:

```
1 DCR_B
2 |
3 |--- Algebra
4 |   |
5 |   |-- Arithmetic.html
6 |   |-- Axiom Non logical axioms.html
7 |   |-- Congruence relation.html
8 |   |-- Euclidean plane.html
9 |   |-- Identity element Definitions.html
10 |  |-- Modular arithmetic Congruence class.html
11 |  |-- Topological group.html
12 |  |-- Vector space.html
13 |  |-- structures
14 |
15 |    |-- Fields
16 |      |-- Field.html
17 |
18 |    |-- Groups
19 |      |-- AAAAAAA.html
20 |      |-- Abelian groups
21 |        |-- Associative property.html
22 |        |-- Commutative property.html
23 |        |-- Additive identity.html
24 |        |-- Coprime integers.html
25 |        |-- Distributive property.html
26 |        |-- Generating set of a group.html
27 |        |-- Multiplicative group of integers modulo n.html
28 |        |-- Multiplicative inverse.html
29 |        |-- Subgroups
30 |          |-- Homeomorphism.html
31 |          |-- Homotopy Isotopy.html
32 |          |-- Homotopy.html
33 |
34 |    |-- Rings
35 |      |-- Binary operation.html
36 |      |-- Commutative ring.html
37 |      |-- Isomorphism.html
38 |
39 |-- Calculus
40 |
41 |--- Calculus_1
42 |   |-- Real analysis.html
43 |
44 |--- Calculus_2
45 |   |-- Compact open topology.html
46 |   |-- Conservative extension.html
47 |   |-- Curve Definition.html
48 |   |-- Curve orientation.html
49 |   |-- Differential form.html
50 |   |-- Differential structure.html
51 |   |-- Discrete logarithm.html
52 |   |-- Exterior derivative.html
53 |   |-- Line integral.html
54 |   |-- Manifold.html
55 |   |-- Multiple integral.html
56 |   |-- Theorems
57 |     |-- Divergence theorem.html
58 |     |-- Green identities.html
59 |     |-- Green law.html
60 |     |-- Green theorem.html
61 |       |-- Stokes theorem Theorem.html
62 |       |-- Stokes theorem.html
63 |     |-- Topological manifold.html
64 |
65 |--- Calculus_3
66 |   |-- Complex analysis.html
67 |   |-- Complex number.html
68 |   |-- Costas array Welch.html
69 |   |-- Jordan curve theorem.html
70 |   |-- Transforms
71 |     |-- Discrete Fourier transform.html
72 |     |-- Fourier inversion theorem.html
73 |     |-- Fourier transform Fourier transform on Euclidean space.html
74 |     |-- Harmonic analysis.html
75 |
76 |-- listingDCRB.txt
```

## Search Snapshots

The wikipedia pages downloaded for the project are related to mathematical topics. To satisfy the requests some specific inputs have been submitted.

### Search for a Non-existing String

The snapshot below shows the result of searching for a non-existing string ([github](#)):

```
[Svitol>python retrieval.py
Insert the name or partial name to retrieve: github
Keyword " github " not found
```

Figure 1: Search for a Non-existing String

### Search for a String Matching at Least Two File Names, but Not Found in Any Searchable File

The documents used to perform the retrieval have been created manually in order to satisfy the request. I modified the name of the page, adding to the original name of the pages the words "**Digital content**". Since the topic treated in the wikipedia pages are mathematical, it's very likely that they don't contain the words "**Digital content**". The snapshot below illustrates this scenario:

```
[Svitol>python retrieval.py
Insert the name or partial name to retrieve: digital content
[Svitol>more out.txt
ID Path Type Occurrences Inside text
81    /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/Digital content Arithmetic.html html 0
77    /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/Digital content Congruence relation.html html 0
78    /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/Digital content Vector space.html html 0
out.txt (END)
```

Figure 2: Search for a String Matching File Names but Not in Files

### Search for a String Matching at Least One File Name and Contained in at Least One Searchable File

The submitted word is "**theorem**". The snapshot below shows the result of a search, including the count of occurrences:

```
Svitol>python retrieval.py
Insert the name or partial name to retrieve: theorem
Svitol>more out.txt
ID Path Type Occurrences Inside text
31   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 1/set theory.html html 63
32   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 1/real analysis.html html 63
34   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/topological manifold.html html 26
35   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/curve Definition.html html 12
36   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/curve orientation.html html 3
37   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/line integral.html html 103
38   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/discrete logarithm.html html 6
39   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/Manifold.html html 51
40   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/Differential form.html html 69
42   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/Exterior derivative.html html 114
43   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/Differential structure.html html 32
44   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/Multiple integral.html html 96
45   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/Conservative extension.html html 44
48   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/Theorems/Divergence theorem.html html 270
49   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/Theorems/Green law.html html 3
50   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/Theorems/Green identities.html html 49
51   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/Theorems/Stokes theorem Theorem.html html 203
52   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/Theorems/Stokes theorem.html html 283
53   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/Theorems/Green theorem.html html 227
54   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/complex curves.html html 48
56   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/complex integrals.html html 48
57   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/Complex analysis.html html 45
58   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/Jordan curve theorem.html html 243
60   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/Complex number.html html 48
63   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/Transforms/Dynamical system.html html 64
64   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/Transforms/Fourier transform Fourier transform on Euclidean space.html html 120
65   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/Transforms/Lyapunov equation.html html 3
67   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/Transforms/Hopf bifurcation.html html 9
68   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/Transforms/Discrete Fourier transform.html html 82
69   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/Transforms/Fourier inversion theorem.html html 113
71   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/Transforms/Lorenz system.html html 24
72   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/Transforms/Laplace transform.html html 55
73   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/Transforms/Harmonic analysis.html html 13
74   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/Euclidean plane.html html 30
75   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/Arithmetic.html html 17
79   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/Modular arithmetic Congruence class.html html 38
80   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/Topological group.html html 50
81   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/Congruence relation.html html 6
82   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/Axiom Non logical axioms.html html 59
83   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/Algebraic structures.html html 30
84   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/Vector space.html html 45
85   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/Algebraic cryptography.html html 38
92   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/structures/Groups/Distributive property.html html 4
95   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/structures/Groups/Coprime integers.html html 17
97   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/structures/Groups/Multiplicative group of integers modulo n.html html 20
98   /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/structures/Groups/Generating set of a group.html html 1
102  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/structures/Groups/Subgroups/Homeomorphism.html html 9
103  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/structures/Groups/Subgroups/Homotopy.html html 8
104  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/structures/Groups/Subgroups/Homotopy Isotopy.html html 8
109  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/structures/Fields/Field.html html 67
110  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/structures/Rings/Binary operation.html html 31
112  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/structures/Rings/Commutative ring.html html 60
113  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/structures/Rings/Isomorphism.html html 10
out.txt (END)
```

Figure 3: Search Matching File Name and Content with Occurrence Count

## Search for a String Not Matching a File Name but Found in at Least One Searchable File

The searched word is "poincaré" but no file contains poincaré in the name. The snapshot below demonstrates this case:

```
Svitol>python retrieval.py
Insert the name or partial name to retrieve: poincaré
Svitol>more out.txt
ID Path Type Occurrences Inside text
39  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/Manifold.html html    27
40  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/Differential form.html      html    2
42  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/Exterior derivative.html      html    4
43  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 2/Differential structure.html      html    4
54  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/complex curves.html      html    2
56  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/complex integrals.html      html    2
60  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/Complex number.html      html    2
63  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/transforms/Dynamical system.html      html    29
64  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/transforms/Fourier transform on Euclidean space.html      html    6
67  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/transforms/Hopf bifurcation.html      html    5
71  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/transforms/Lorenz system.html      html    10
73  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Calculus/Calculus 3/transforms/Harmonic analysis.html      html    1
74  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/Euclidean plane.html      html    4
80  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/Topological group.html      html    2
82  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/Axiom Non logical axioms.html      html    2
83  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/algebraic structures.html      html    4
97  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/structures/Groups/Multiplicative group of integers modulo n.html      html    2
102  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/structures/Groups/Subgroups/Homeomorphism.html      html    11
103  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/structures/Groups/Subgroups/Homotopy.html      html    4
104  /Users/svitol/Desktop/Unipv/DCR project/Project/DCR B/Algebra/structures/Groups/Subgroups/Homotopy Isotopy.html      html    4
out.txt (END)
```

Figure 4: Search Not Matching File Name but Found in Content

## Conclusions

With this project, I learned how to build a simple information retrieval system from scratch using indexes in MySQL. I had to adopt several strategies to address challenges, such as managing RAM during bulk database loading and information retrieval. This project has provided valuable insights into database management and optimization techniques.