Part 1: Database Schema and Fragmentation

Q 1.1 Write an SQL query to solve the following problem: Find the top 5 books with the highest ratings, and 5 books that have the lowest rating, return their ranks (sorted in descending order), titles, publishers and number of pages.

Answer 1.1

After Reading the csv file as dataframe and using Pysqldf Library to run sql statements on the pandas dataframe. The results are **sorted by SalesRank in descending order. The values where RatingValue is not null is returned.**

SQL -

select SalesRank, Title, Publisher, Pages from (

select * from (select * from book3 order by RatingValue desc limit 5)

UNION

select * from (select * from book3 where RatingValue IS NOT NULL order by RatingValue asc limit 5))

order by SalesRank desc

```
pysqldf("""select SalesRank, Title, Publisher, Pages  from (
select * from (select * from book3 order by RatingValue desc limit 5)
UNION
select * from (select * from book3 where RatingValue IS NOT NULL order by RatingValue asc limit 5))
order by SalesRank desc""")
```

	SalesRank	Title	Publisher	Pages
0	2899	The Autobiography of Malcolm X (MaxNotes)	Research & Education Association	104.0
1	2888	The Story of a Soul: The Autobiography of St. ...	CreateSpace Publishing	280.0
2	2058	Freeway Rick Ross: The Untold Autobiography	CreateSpace Publishing	298.0
3	1718	The Autobiography of Mark Twain	HarperCollins Publishers	560.0
4	1257	Mayor: An Autobiography	Simon & Schuster	256.0
5	1100	Angela Davis: An Autobiography	International Publishers Company, Incorporated	400.0
6	1042	The Collected Autobiographies of Maya Angelou	Random House Publishing Group	1184.0
7	401	Winning Is Not Enough: The Autobiography	Headline Book Publishing, Limited	576.0
8	342	An Autobiography of a Nobody	AuthorHouse	512.0
9	100	Autobiografia de un Yogui (Autobiography of a ...	Self-Realization Fellowship	742.0

Q 1.2 Which table schema(s) is/are used to answer the above query?

Answer 1.2

Schema for Book3 is used as it only has the SalesRank column for which rank can be used-

Book3 (ID, Title, Author1, Author2, Author3, Publisher, ISBN13, Date, Pages, ProductDimensions, SalesRank, RatingsCount, RatingValue, PaperbackPrice, HardcoverPrice, EbookPrice, AudiobookPrice)

Q 2.1 If the goal of database A is to handle each query via a dedicated local site (with no information needed from the other site(s)), which fragmentation strategy should be used to fragment the Book3 table? If only two fragments are generated, write their schemas (if vertically fragmented) or predicates (if horizontally fragmented), respectively.

Answer 2.1

Vertical fragmentation will be used for fragmenting the schema. While using vertical fragmentation, the columns of the table will be grouped into 2 different fragments. **Each fragment will contain " ID" column as the primary key of the fragment as it carries the unique value for Book3.**

Fragment 1 - { ID, Title, Author1, Author2, Author3, Publisher}

Fragment 2 - { ID, ISBN13, Date, Pages, ProductDimensions, SalesRank, RatingsCount, RatingValue, PaperbackPrice, HardcoverPrice, EbookPrice, AudiobookPrice}

Q2.2 Assume that we horizontally fragment the table into three fragments based on the following predicates: ◦ Fragment 1: RatingsCount ≤ 25 ◦ Fragment 2: 25 < RatingsCount ≤ 120 ◦ Fragment 3: RatingsCount > 125

Is this set of predicates valid?

Answer 2.2

The fragment given above are incorrect as the 3 properties of fragmentation are not followed.

Completeness, Disjointness, reconstructability - when all the fragments are joined, records 120 < RatingsCount <=125 are missed.

Proposed solution =

	OPTION 1	**OPTION 2**
FRAGMENT 1	RatingsCount ≤ 25	RatingsCount ≤ 25
FRAGMENT 2	25 < RatingsCount ≤ 120	25 < RatingsCount ≤ 125
FRAGMENT 3	RatingsCount > 120	RatingsCount > 125

We will move forward with option 1

Predicates =

RatingsCount <= 25, RatingsCount > 25

25 < RatingsCount <= 120, 25 >= RatingsCount > 120

RatingsCount > 120, RatingsCount <= 120

Minterm predicates = Now we have 3 predicates. so there will be 2^3 = 8 minterm predicates.

(RatingsCount <= 25) ∧ (25 < RatingsCount <= 120) ∧ (RatingsCount > 120)

(RatingsCount <= 25) ∧ (25 < RatingsCount <= 120) ∧ (RatingsCount <= 120)

(RatingsCount <= 25) ∧ (25 >= RatingsCount > 120) ∧ (RatingsCount > 120)

(RatingsCount <= 25) ∧ (25 >= RatingsCount > 120) ∧ (RatingsCount <= 120)

(RatingsCount > 25) ∧ (25 < RatingsCount <= 120) ∧ (RatingsCount > 120)

(RatingsCount > 25) ∧ (25 < RatingsCount <= 120) ∧ (RatingsCount <= 120)

(RatingsCount > 25) ∧ (25 >= RatingsCount > 120) ∧ (RatingsCount > 120)

(RatingsCount > 25) ∧ (25 >= RatingsCount > 120) ∧ (RatingsCount <= 120)

Correct Minterms =

(RatingsCount <= 25) ∧ (25 < RatingsCount <= 120) ∧ (RatingsCount > 120)

(RatingsCount <= 25) ∧ (25 < RatingsCount <= 120) ∧ (RatingsCount <= 120) :: (RatingsCount <= 25)

(RatingsCount <= 25) ∧ (25 >= RatingsCount > 120) ∧ (RatingsCount <= 120) :: (RatingsCount <= 25)

(RatingsCount > 25) ∧ (25 < RatingsCount <= 120) ∧ (RatingsCount <= 120) :: (25 < RatingsCount <= 120)

(RatingsCount > 25) ∧ (25 >= RatingsCount > 120) ∧ (RatingsCount > 120) :: (RatingsCount > 120)

Incorrect Minterms =

(RatingsCount <= 25) ∧ (25 >= RatingsCount > 120) ∧ (RatingsCount > 120) ---------- (incorrect)

(RatingsCount > 25) ∧ (25 < RatingsCount <= 120) ∧ (RatingsCount > 120) ---------- (incorrect)

(RatingsCount > 25) ∧ (25 >= RatingsCount > 120) ∧ (RatingsCount <= 120) ---------- (incorrect)

Hence the groups will be =

Fragment 1: RatingsCount ≤ 25

Fragment 2: 25 < RatingsCount ≤ 120

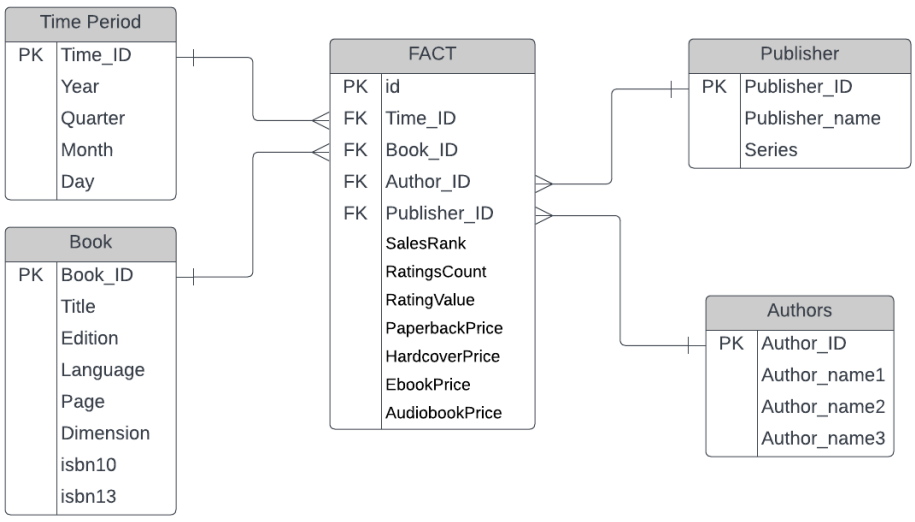
Fragment 3: RatingsCount > 120

INSERTING Record into new Fragments - After the new Fragments are created, when a new record needs to be inserted, we check if the predicates (column used for divisions in horizontal fragment) are defined as the primary key or not. If the predicate or fragment is defined as the primary key, we check all the fragments to make sure that this new value which will be primary key does not exist. Then, the value of new "RatingsCount" will be checked with the predicates defined and insert it into the right fragment. If the value already exists and it's a primary key, we need to update the record. The existing record will be searched using where clause and the record/tuple will be sent to the right fragment.

Part 2: Data Warehousing

Q 3.1 Show the schema and point out the dimensions and fact table. Given that we have a dimension table for each dimension and there are 4000 records in the fact table; Among all dimension tables and the fact table, which table has the most records? Why? Explain your answer.

Answer 3.1



Dimension table – Time Period, Book, Publisher, Authors

Fact table – FACT

The fact table is always larger than the dimension table. The fact table contains foreign keys, measurements and metrics on which calculations are performed. The fact table grows vertically and have more number of records and fewer attributes when compared with dimension table.

Q 4.1 What are the advantages of building a bitmap index? Which type of column is not suitable for a bitmap index? Why?

Answer 4.1

Bitmap Indexing is an indexing technique that uses bitmaps (0,1) for its columns. This technique is used for data warehouses with large amount of data. This is applied to columns with low level of cardinality(number of individual distinct elements), where these columns are most frequently used for querying the data rather than issueing updates. The columns can also be considered categorical/Nominal columns.

Advantages:

- Faster retrieval of records and a reduction in response time for large classes of ad hoc queries
- A substantial reduction of space usage
- Dramatic performance gains even on hardware with low resources
- Increase in efficiency in terms of insertion, deletion and updation of data.

Bitmap index is not created for columns that have unique values in the row or act as the primary key. It should also not be created for the columns that are transactional in nature, whose values keep on updating. They should also not be created for the columns with a high level of cardinality as bitmap indexing is time consuming and hard to maintain if updates are common.

Q 4.2 Suppose the Publisher column only contains four distinct values and Language only contains two, which are all shown in the above example. Please create bitmap indices for both Publisher and Language.

Answer 4.2

Original Table

Date	Publisher	Language	Sales
07/15/1984	AAAI Press	English	11
05/05/1990	Springer International Publishing	English	23
06/04/1995	Springer London	English	15
12/11/2000	IEEE Computer Society Press	English	30
04/03/2004	AAAI Press	Japanese	2
05/01/2008	Springer International Publishing	Japenese	13
11/19/2012	Springer London	Japanese	5
08/06/2014	IEEE Computer Society Press	Japanese	22

BITMAP INDICES on Publisher Column

RecID	AAAI Press	Springer International Publishing	Springer London	IEEE Computer Society Press
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1	0	0	0
6	0	1	0	0
7	0	0	1	0
8	0	0	0	1

BITMAP INDICES on Language Column

RecID	English	Japanese
1	1	0
2	1	0
3	1	0
4	1	0
5	0	1
6	0	1
7	0	1
8	0	1

Q4.3 Explain how to use the bitmap indices to find the total sales of Japanese books published by AAAI Press.

Answer 4.3

B(AAAI Press)	10001000
B(Springer International Publishing)	01000100
B(Springer London)	00100010
B(IEEE Computer Society Press)	00010001
B(English)	11110000
B(Japanese)	00001111

B(Japanese) ∧ B(AAAI Press)	0	0	0	0	1	1	1	1
	1	0	0	0	1	0	0	0
Result	0	0	0	0	1	0	0	0

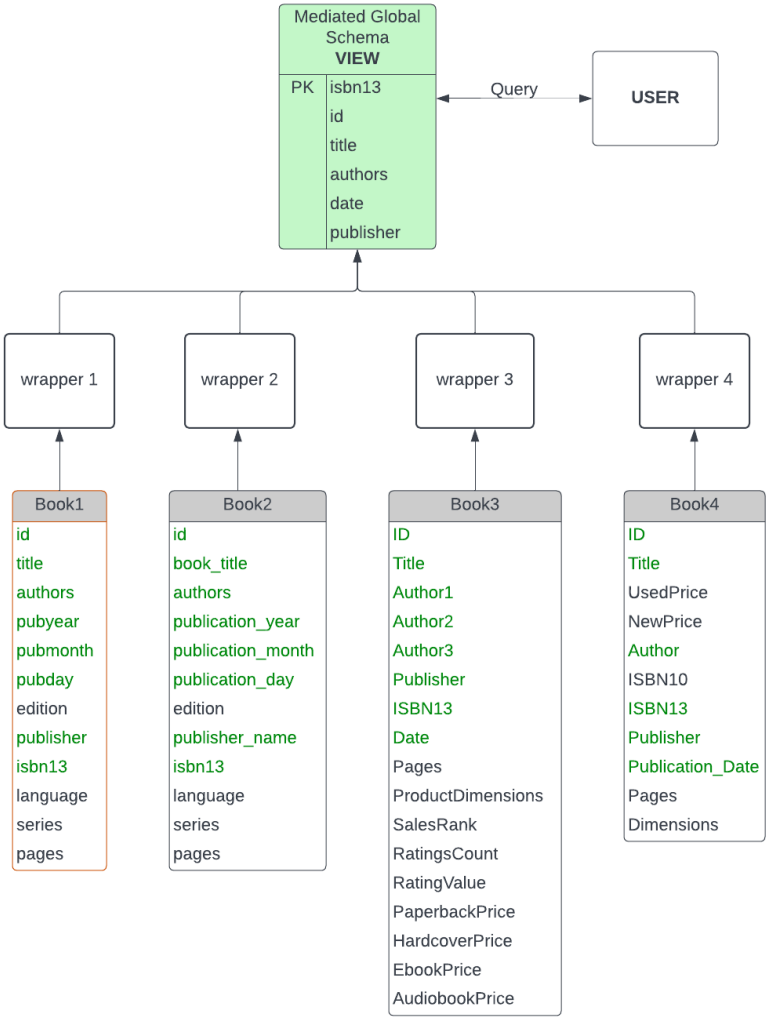
Hence, only 5th row is the result of the query. i.e. sales is 2

04/03/2004	AAAI Press	Japanese	2

Part 3: Data Integration

Q5.1 Design a global schema which will combine the common attributes from each schema together. Your design should include any information that is represented in all four schemas. If an attribute cannot be found or derived in the given schemas, then it should be left out of your global schema

Answer 5.1



Query to make Global View -

CREATE VIEW Global_view AS

select * from (select id, title as title, authors as authors, cast(pubday as integer) || '/' || cast(pubmonth as integer) || '/' || cast(pubyear as integer) as date, publisher as publisher, isbn13 as isbn13 from book1

union

select id, book_title as title, authors as authors, cast(publication_day as integer) || '/' || cast(publication_month as integer) || '/' || cast(publication_year as integer) as date, publisher_name as publisher, isbn13 as isbn13 from book2

union

select ID as id, Title as title, Author1 || ',' || Author2 || ',' || Author3 as authors, Date as date, Publisher as publisher, ISBN13 as isbn13 from book3

union

select ID as id, Title as title, Author as authors, Publication_Date as date, Publisher as publisher, ISBN13 as isbn13 from book4)

Q 5.2 Identify any structural or semantic heterogeneity issues that may occur during your integration. Using the data, provide a concrete example of each, if applicable, and outline a possible resolution.

Answer 5.2

Structural Heterogeneity: Different schemas have different formats for columns which are used as primary column in there respective schemas. A prime example will be the id or ID column from the 4 different schemas which are used as primary key. While integrating, the id columns are string and int which can cause issue during integration. Here the ID column for book3 is integer while for book4 is string. A better implementation would have been the use of ISBN13 column as the primary column. ISBN13 is a unique identifier for books and if the ISBN13 matches in different schemas, then they refer to the same book title.

```
book3["ID"].head()
```

```
0    1
1    2
2    3
3    4
4    5
Name: ID, dtype: int64
```

```
book4["ID"].head()
```

```
0    HC0001
1    HC0002
2    HC0003
3    HC0004
4    HC0005
Name: ID, dtype: object
```

Semantic Heterogeneity: Different things mean the same thing in different schemas. A good example of this is the title/Book_Title column or the publication_year/pubyear column of Book1 and Book2 schemas. A good approach to solve this issue will be to make a new view with a column Title that will accommodate values for both "title" column from book1 schema and "book_title" column from book2 schema. Book1 and book2 schema has date parts as separate columns for both the schemas. A possible solution for this will be to make a date column that integrates individual values from year, month and day and push the value to a new "Date" column.

```
book1[["title", "pubyear"]].head()
```

	title	pubyear
0	Building the Data Warehouse by Inmon, W. H. Pa...	0.0
1	Joint Application Design: How to Design Qualit...	89.0
2	Oracle Mobile Application Framework Developer ...	14.0
3	OCA/OCP Java SE 7 Programmer I & II Practice E...	15.0
4	JasperReports 3.5 for Java Developers	9.0

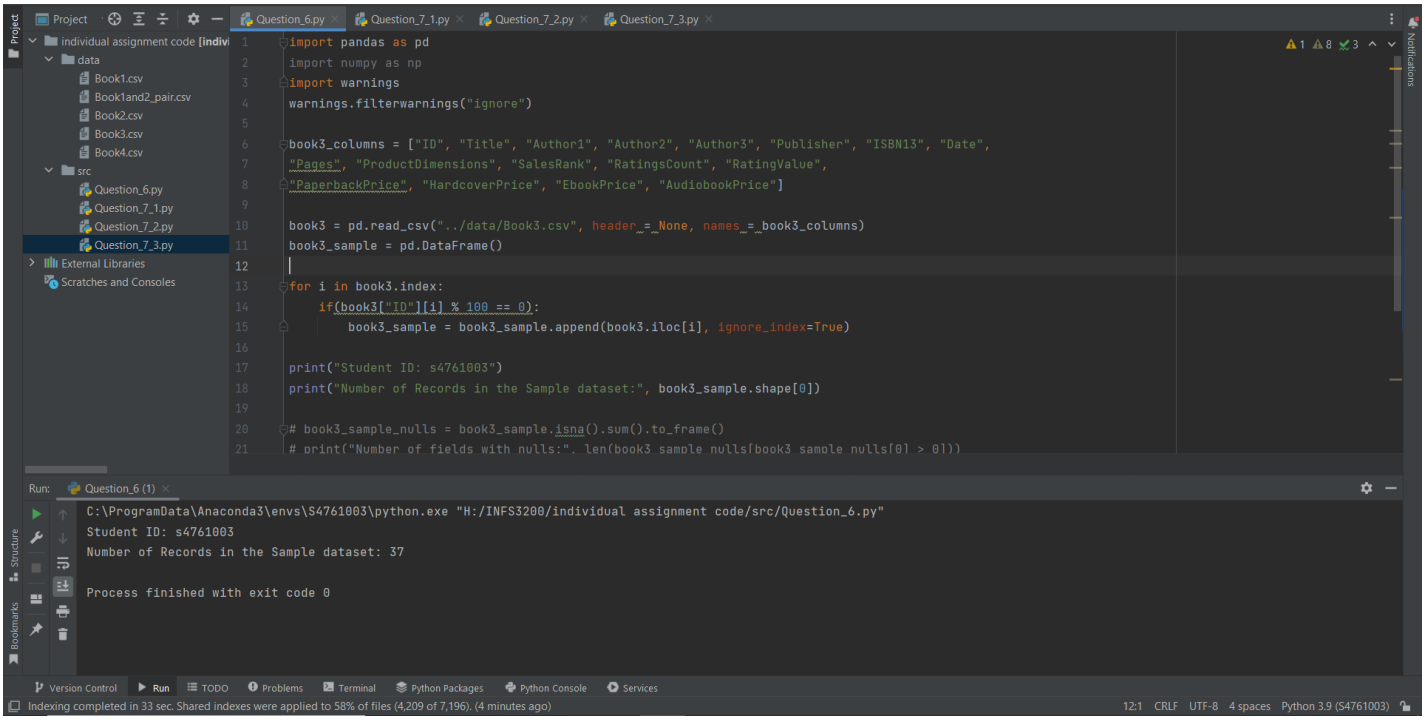
```
book2[["book_title", "publication_year"]].head()
```

	book_title	publication_year
0	1-2-3 Database Techniques	89.0
1	10 Minute Guide to Access 97	96.0
2	10 Minute Guide to Lotus Notes	96.0
3	11th International Workshop on Database and Ex...	0.0
4	18th International Conference on Scientific an...	6.0

Part 4: Data Quality

Q6.1 By sampling records whose id is the multiple of 100 (that is: 0, 100, 200, 300, …), how many records are there in the sample set?

Answer 6.1



Q6.2 Among the samples found in the previous question (question 6.1), how many fields containing NULL values are present?

Answer 6.2

```python
warnings.filterwarnings("ignore")

book3_columns = ["ID", "Title", "Author1", "Author2", "Author3", "Publisher", "ISBN13", "Date",
"Pages", "ProductDimensions", "SalesRank", "RatingsCount", "RatingValue",
"PaperbackPrice", "HardcoverPrice", "EbookPrice", "AudiobookPrice"]

book3 = pd.read_csv("../data/Book3.csv", header = None, names = book3_columns)
book3_sample = pd.DataFrame()

for i in book3.index:
    if(book3["ID"][i] % 100 == 0):
        book3_sample = book3_sample.append(book3.iloc[i], ignore_index=True)

print("Student ID: s4761003")
print("Number of Records in the Sample dataset:", book3_sample.shape[0])

book3_sample_nulls = book3_sample.isna().sum().to_frame()
print("Number of columns with nulls:", len(book3_sample_nulls[book3_sample_nulls[0] > 0]))

number_of_nulls = book3_sample_nulls[0].sum()
print("Number of Nulls in the dataset:", number_of_nulls)
```

```
C:\ProgramData\Anaconda3\envs\S4761003\python.exe "H:/INFS3200/individual assignment code/src/Question_6.py"
Student ID: s4761003
Number of Records in the Sample dataset: 37
Number of columns with nulls: 12
Number of Nulls in the dataset: 286

Process finished with exit code 0
```

Q6.3 Calculate the defects per million opportunities (DPMO) according to your samples. You can assume that any NULL value is an error, and that the remaining values are valid.

Answer 6.3

```python
for i in book3.index:
    if(book3["ID"][i] % 100 == 0):
        book3_sample = book3_sample.append(book3.iloc[i], ignore_index=True)

print("Student ID: s4761003")
print("Number of Records in the Sample dataset:", book3_sample.shape[0])

book3_sample_nulls = book3_sample.isna().sum().to_frame()
print("Number of columns with nulls:", len(book3_sample_nulls[book3_sample_nulls[0] > 0]))

number_of_nulls = book3_sample_nulls[0].sum()
print("Number of Nulls in the dataset:", number_of_nulls)

DPMO = (number_of_nulls / (book3_sample.shape[0] * book3_sample.shape[1])) * (10**6)
print("Defects per Million Opportunities:", DPMO)
```

```
C:\ProgramData\Anaconda3\envs\S4761003\python.exe "H:/INFS3200/individual assignment code/src/Question_6.py"
Student ID: s4761003
Number of Records in the Sample dataset: 37
Number of columns with nulls: 12
Number of Nulls in the dataset: 286
Defects per Million Opportunities: 454689.98410174885

Process finished with exit code 0
```

s1 = “Peter Rob, Carlos Coronel”

s2 = “Carlos Coronel;Steven Morris;Peter Rob;”

Q 7.1 Compute the edit distance between s1 and s2 – What is the value?

Answer 7.1

```
    edit_distance = len(s1)
else:
    len_source = len(s1) + 1
    len_target = len(s2) + 1

    arr = []
    for i in range(len_target):
        arr.append([0] * len_source)

    for i in range(len_target):
        for j in range(len_source):
            if (i == 0):
                arr[i][j] = j
            elif (j == 0):
                arr[i][j] = i
            elif (s2[i - 1] == s1[j - 1]):
                arr[i][j] = arr[i - 1][j - 1]
            else:
                arr[i][j] = 1 + min(arr[i][j - 1], arr[i - 1][j], arr[i - 1][j - 1])
    edit_distance = arr[len_target - 1][len_source - 1]
```

```
C:\ProgramData\Anaconda3\envs\S4761003\python.exe "H:/INFS3200/individual assignment code/src/Question_7_1.py"
Student ID: s4761003
Edit Distance: 33

Process finished with exit code 0
```

Q7.2 Compute the Jaccard distance between s1 and s2 – What is the value?

Answer 7.2

```
        return list(set(str_tokens))
    else:
        str_tokens = string.split(" ")
        return list(set(str_tokens))


s1 = "Peter Rob, Carlos Coronel"
s2 = "Carlos Coronel;Steven Morris;Peter Rob;"

print("Student ID: s4761003")
for q in range(1,6):
    str1_tokens = tokenize(s1, q)
    str2_tokens = tokenize(s2, q)
    total_tokens = str1_tokens + str2_tokens
    total_tokens = list(set(total_tokens))
    jaccard_distance = 1 - (len(str1_tokens) + len(str2_tokens) - len(total_tokens)) / len(total_tokens)
    print(f"Jaccard Distance with tokenization parameter set as {str(q)}:", jaccard_distance)
```

```
C:\ProgramData\Anaconda3\envs\S4761003\python.exe "H:/INFS3200/individual assignment code/src/Question_7_2.py"
Student ID: s4761003
Jaccard Distance with tokenization parameter set as 1: 0.3157894736842105
Jaccard Distance with tokenization parameter set as 2: 0.4473684210526315
Jaccard Distance with tokenization parameter set as 3: 0.5365853658536586
Jaccard Distance with tokenization parameter set as 4: 0.5853658536585367
Jaccard Distance with tokenization parameter set as 5: 0.6341463414634146

Process finished with exit code 0
```

Q 7.3 Write a program to link the data between Book1 and Book2 using the Jaccard coefficient with 3-gram tokenization as the similarity measure, performing the comparison only on the book title field, and using a matching threshold of 0.75 (that is, only return matches with a similarity of 0.75 or more). Compare your output with the gold-standard dataset and compute the precision, recall and F-measure.

Answer 7.3

```python
        print("no file.")
except:
    print("Error occurred. Check if file exists")

pair_found = []
count = 0
for pair in results:
    if pair in benchmark:
        pair_found.append(pair)
        count = count + 1

print("Student ID: s4761003")
if count == 0:
    print('Precision=0, Recall=0, Fmeasure=0')
else:
    precision = count / len(results)
    recall = count / len(benchmark)
    f_measure = 2 * precision * recall / (precision + recall)

    print("Pairs found in gold-standard dataset: " + str(pair_found))
    print(f'Precision= {precision}, Recall= {recall}, Fmeasure= {f_measure}')
```

```
C:\ProgramData\Anaconda3\envs\S4761003\python.exe "H:/INFS3200/individual assignment code/src/Question_7_3.py"
Student ID: s4761003
Pairs found in gold-standard dataset: ['31,890', '45,195', '50,3093', '63,2522', '64,2251', '88,1377', '95,283', '114,408', '136,2482', '141,3244', '150,2826', '157,1949', '183,2
Precision= 0.1823579304495335, Recall= 0.9267241379310345, Fmeasure= 0.30474840538625086

Process finished with exit code 0
```