CS 4250 – Assignment #2
Maximum Points: 100 pts.

Bronco ID: |__|__|__|__|__|__|__|__|__|

Last Name: _____

First Name: _____

Note 1: Your submission header must have the format as shown in the above-enclosed rounded rectangle.
Note 2: Homework is to be done individually. You may discuss the homework problems with your fellow students, but you are NOT allowed to copy – either in part or in whole – anyone else's answers.
Note 3: Your deliverable should be a .pdf file submitted through Gradescope until the deadline. Do not forget to assign a page to each of your answers when making a submission. In addition, source code (.py files) should be added to an online repository (e.g., github) to be downloaded and executed later.
Note 4: All submitted materials must be legible. Figures/diagrams must have good quality.
Note 5: Please use and check the Canvas discussion for further instructions, questions, answers, and hints. The bold words/sentences provide information for a complete or accurate answer.

1. [15 points]. Based on the requirements below, **design** your database (**logical view**) following the **relational model**. Hint: we are simulating an inverted index.

   REQ_01 - Documents Registration

   Rationale: DOCUMENTS are identified by a *doc* number and have the attributes *text*, *title*, *num_chars*, and *date*.

   REQ_02 - Categories Registration

   Rationale: CATEGORIES are identified by an *id* and have the attribute *name*.

   REQ_03 - Relationship between Document and Category

   Rationale: Each DOCUMENT must belong to a unique CATEGORY, but more than one DOCUMENT can belong to the same CATEGORY.

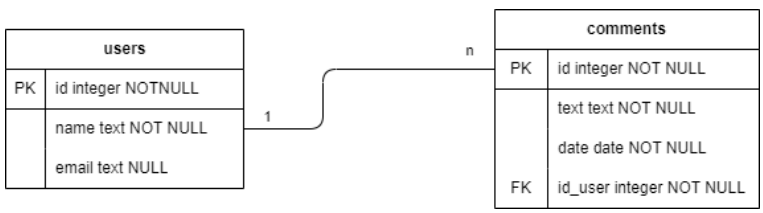
   REQ_04 - Terms Registration

   Rationale: TERMS are identified by a *term* and have the attribute *num_chars*.

   REQ_05 - Relationship between Document and TERM

   Rationale: A DOCUMENT might have multiple TERMS and a TERM might be in multiple DOCUMENTS. An attribute *count* informs how many times a TERM occurs in a DOCUMENT.

   Solution example:



   *In case you have a field that is both (PK and FK), use PFK notation.

2. [20 points]. Based on the database logical model created in question 1, provide a **single SQL query** to retrieve the requested information. Use the documents and terms informed below to guide you.

Documents:

text	title	num_chars*	date	category
Baseball is played during summer months.	Exercise	34	2023-10-03	Sports
Summer is the time for picnics here. Picnics time!	California	40	2023-10-03	Sports
Months, months, months later we found out why.	Discovery	36	2023-10-03	Seasons
Why is summer so hot here? So hot!	Arizona	25	2023-10-03	Seasons

*The number of characters does not include spaces and punctuation marks.

Terms:

baseball	is	played	during	summer	months
the	time	for	picnics	here	later
we	found	out	why	so	hot

a. [2 point]. How many (or what) the documents are in D?
b. [2 points]. How many (or what) documents are in D from the category "Sports"? Requirement: query the tables by using the category name.
c. [2 points]. How many (or what) distinct terms are in D?
d. [2 points]. How many terms (considering repetitions) are in the document “Arizona”? Requirement: query the tables by using the document title.
e. [4 points]. How many terms (considering repetitions) are linked to the category "Seasons"? Requirement: query the tables by using the category name.
f. [2 points]. How many times (considering repetitions) does the term "months" occur in D?
g. [3 points]. What is the largest document (more terms - considering repetitions)? Requirement: output the document title and its corresponding number of terms.
h. [3 points]. What is the most frequent term in D (considering the number of distinct documents)? Requirement: output the term and its corresponding occurrences.

3. [15 points]. Complete the Python program (db_connection.py) that will use the **corpus** database tables created in question 1 to manage an inverted index. Requirements: 1) use the driver program index.py to trigger the operations (**do not change it**) and 2) create your tables in your Python program or provide the database schema in a separate file. Add the link to an online repository as the answer to this question.

a) Create a category.

Input: {*id, name*}

b) Create a document.

Input: {*id, text*, *title*, *date,* and *category*}

c) Update a document.

Input: {*id, text*, *title*, *date,* and *category*}

d) Delete a document.

Input: {*id*}

e) Output the inverted index ordered by term.

Output: {‘*term’: ‘document title: count’*}

Sample output: Input data in red.

######### Menu ##############

#a - Create a category.

#b - Create a document

#c - Update a document

#d - Delete a document.

#e - Output the inverted index.

#q - Quit

Enter a menu choice: e

{}

Enter a menu choice: a

Enter the ID of the category: 1

Enter the name of the category: Sports

Enter a menu choice: a

Enter the ID of the category: 2

Enter the name of the category: Seasons

Enter a menu choice: b

Enter the ID of the document: 1

Enter the text of the document: Baseball is played during summer months.

Enter the title of the document: Exercise

Enter the date of the document: 2023-10-03

Enter the category of the document: Sports

Enter a menu choice: e

{'baseball': 'Exercise:1', 'during': 'Exercise:1', 'is': 'Exercise:1', 'months': 'Exercise:1', 'played': 'Exercise:1', 'summer': 'Exercise:1'}

Enter a menu choice: b

Enter the ID of the document: 2

Enter the text of the document: Summer is the time for picnics here. Picnics time!

Enter the title of the document: California

Enter the date of the document: 2023-10-03

Enter the category of the document: Sports

Enter a menu choice: e

{'baseball': 'Exercise:1', 'during': 'Exercise:1', 'for': 'California:1', 'here': 'California:1', 'is': 'Exercise:1, California:1', 'months': 'Exercise:1', 'picnics': 'California:2', 'played': 'Exercise:1', 'summer': 'California:1, Exercise:1', 'the': 'California:1', 'time': 'California:2'}

Enter a menu choice: b

Enter the ID of the document: 3

Enter the text of the document: Months, months, months later we found out why.

Enter the title of the document: Discovery

Enter the date of the document: 2023-10-03

Enter the category of the document: Seasons

Enter a menu choice: e

{'baseball': 'Exercise:1', 'during': 'Exercise:1', 'for': 'California:1', 'found': 'Discovery:1', 'here': 'California:1', 'is': 'Exercise:1, California:1', 'later': 'Discovery:1', 'months': 'Exercise:1, Discovery:3', 'out': 'Discovery:1', 'picnics': 'California:2', 'played': 'Exercise:1', 'summer': 'Exercise:1, California:1', 'the': 'California:1', 'time': 'California:2', 'we': 'Discovery:1', 'why': 'Discovery:1'}

Enter a menu choice: b

Enter the ID of the document: 4

Enter the text of the document: Why is summer so hot here? So hot!

Enter the title of the document: Arizona

Enter the date of the document: 2023-10-03

Enter the category of the document: Seasons

Enter a menu choice: e

{'baseball': 'Exercise:1', 'during': 'Exercise:1', 'for': 'California:1', 'found': 'Discovery:1', 'here': 'Arizona:1, California:1', 'hot': 'Arizona:2', 'is': 'Arizona:1, Exercise:1, California:1', 'later': 'Discovery:1', 'months': 'Exercise:1, Discovery:3', 'out': 'Discovery:1', 'picnics': 'California:2', 'played': 'Exercise:1', 'so': 'Arizona:2', 'summer': 'California:1, Arizona:1, Exercise:1', 'the': 'California:1', 'time': 'California:2', 'we': 'Discovery:1', 'why': 'Arizona:1, Discovery:1'}

Enter a menu choice: d

Enter the document id to be deleted: 3

Enter a menu choice: e

{'baseball': 'Exercise:1', 'during': 'Exercise:1', 'for': 'California:1', 'here': 'Arizona:1, California:1', 'hot': 'Arizona:2', 'is': 'Exercise:1, Arizona:1, California:1', 'months': 'Exercise:1', 'picnics': 'California:2', 'played': 'Exercise:1', 'so': 'Arizona:2', 'summer': 'California:1, Arizona:1, Exercise:1', 'the': 'California:1', 'time': 'California:2', 'why': 'Arizona:1'}

Enter a menu choice: c

Enter the ID of the document: 4

Enter the text of the document: Why is summer so hot here? This is a bad time!

Enter the title of the document: Arizona

Enter the date of the document: 2023-10-03

Enter the category of the document: Seasons

Enter a menu choice: e

{'a': 'Arizona:1', 'bad': 'Arizona:1', 'baseball': 'Exercise:1', 'during': 'Exercise:1', 'for': 'California:1', 'here': 'Arizona:1, California:1', 'hot': 'Arizona:1', 'is': 'California:1, Exercise:1, Arizona:2', 'months': 'Exercise:1', 'picnics': 'California:2', 'played': 'Exercise:1', 'so': 'Arizona:1', 'summer': 'Arizona:1, California:1, Exercise:1', 'the': 'California:1', 'this': 'Arizona:1', 'time': 'Arizona:1, California:2', 'why': 'Arizona:1'}

4. [15 points]. Repeat the process conducted in question 1 here by using the **document model**. You **must** use **embedding** to accomplish this task. If needed, you can omit some of the identifiers (keys). The answer here should be your flexible document schema, for instance:

   {

   "field1": data_type,

   "field2": [data_type],

   "field3": {field31:data_type, field32: datatype},

   "field4": [{field41:data_type, field42: datatype}, {field43:data_type, field44: datatype}]

   . . .

   }

5. [20 points]. Repeat the process conducted in question 2 here by **only using MongoDB queries**, or more formally, MongoDB Query Language (MQL). You can export the queries from Compass if needed.

6. [15 points]. Similar to question 3, complete the Python program (db_connection_mongo.py) by using PyMongo. Requirements: 1) use the driver program index_mongo.py to trigger the operations (**do not change it**). You can skip option a) from the menu since categories will not be created separately from documents, as done with SQL. Use the provided sample output, skipping the menu choices a), to validate your implementation.

Important Note: Answers to all questions should be written clearly, concisely, and unmistakably delineated. You may resubmit multiple times until the deadline (the last submission will be considered).

NO LATE ASSIGNMENTS WILL BE ACCEPTED. ALWAYS SUBMIT WHATEVER YOU HAVE COMPLETED FOR PARTIAL CREDIT BEFORE THE DEADLINE!