UNIVERSITÀ DEGLI STUDI DI MILANO

Master of Science in Data Science and Economics

StackSample: Finding similar items

Veronica Astorino

ID number: 942307

TABLE OF CONTENTS

LIST OF FIGURES

Introduction

Stack Overflow is a platform in which user post questions to a large developer community and receive answer about an issue that users have. Sometimes questions are similar and already received some answer, so knowing if the users refer to questions that already have been covered could be useful.

The aim of the project is to find similar questions on the Stack Overflow dataset retrieved from Kaggle. For the purpose of perform a similar items analysis, thus finding the similar questions that appear, I implemented in Spark the locality sensitive hashing (LSH) and approximate Near-Neighbors. This dataset contains a huge amount of data therefore the implementation of Spark is due to that. Furthermore, it is able to solve that problem by scaling up with the size of data.

As first, I applied some preprocessing techniques: the remove of stop words and the tokenization of the sentences: the idea is that questions with many tokens in common have high chances of being about the same topic.

The tokens that are extracted for each row are vectorized according to the HashingTF. The features that have been extracted are used to determine the Inverse document frequencies. Subsequently, I classified the questions into 1000 backets, this is done by using MinHashLSH, this technique is based on hashing techniques which allow finding pairs that are potentially similar.

Finally, I used approximate Near- Neighbors method in order to group the similar questions, this has been done using the TF-IDF feature vectors and a single feature as key.

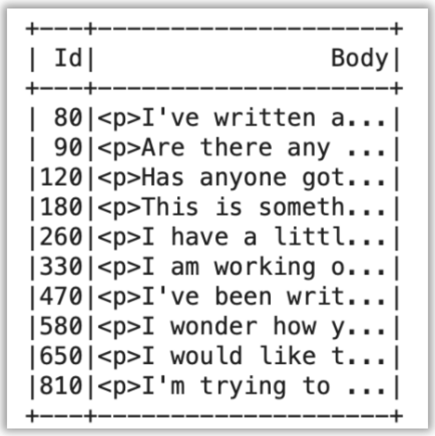
Data

The dataset contains the text of 10% of questions and answers of Stack Overflow website. This dataset is provided as three tables:

- Questions: composed by the title, the body, the date of creation, closed date, the score, and the ID of all non-deleted Stack Overflow questions
- Answers: includes the body, the creation date, the score, and the ID of each answer
- Tags: composed by the tags on each of these questions.

For this project only the Questions is used, and in particular:

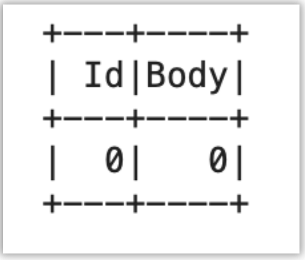
- the Body, the corpus of the questions
- The ID columns, the owner of the questions.



```
+---+--------------------+
| Id|                Body|
+---+--------------------+
| 80|<p>I've written a...|
| 90|<p>Are there any ...|
|120|<p>Has anyone got...|
|180|<p>This is someth...|
|260|<p>I have a littl...|
|330|<p>I am working o...|
|470|<p>I've been writ...|
|580|<p>I wonder how y...|
|650|<p>I would like t...|
|810|<p>I'm trying to ...|
+---+--------------------+
```

FIGURE 1: DATASET

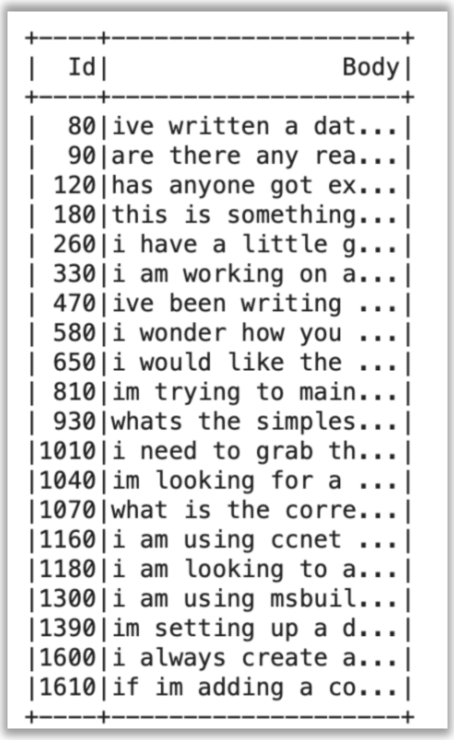
Questions have 1,264,216 observations and as Table 2 shows, in the columns Id and Body there are not missing values.



```
+---+----+
| Id|Body|
+---+----+
|  0|   0|
+---+----+
```

FIGURE 2: MISSING VALUES

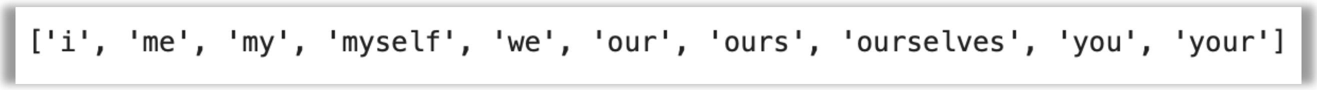
As first, I lowered all the text and I filtered out all the things from the column ‘Body’ that are not useful information for the propose of the project, like "<p>", [^a-zA-Z0-9\\s]". Both operations have been performed by using the library pyspark.sql.functions.



```
+----+--------------------+
|  Id|                Body|
+----+--------------------+
|  80|ive written a dat...|
|  90|are there any rea...|
| 120|has anyone got ex...|
| 180|this is something...|
| 260|i have a little g...|
| 330|i am working on a...|
| 470|ive been writing ...|
| 580|i wonder how you ...|
| 650|i would like the ...|
| 810|im trying to main...|
| 930|whats the simples...|
|1010|i need to grab th...|
|1040|im looking for a ...|
|1070|what is the corre...|
|1160|i am using ccnet ...|
|1180|i am looking to a...|
|1300|i am using msbuil...|
|1390|im setting up a d...|
|1600|i always create a...|
|1610|if im adding a co...|
+----+--------------------+
```

FIGURE 3: CLEAN BODY

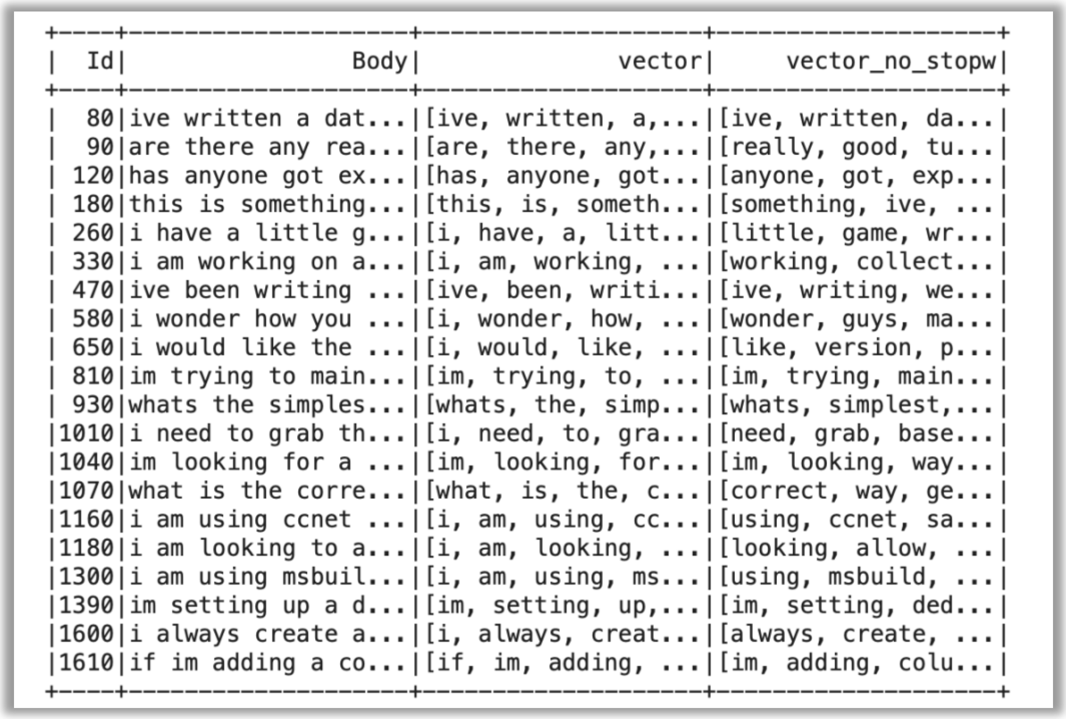
After, I tokenize the Body, this process split text into individual sentences. The previous step was not satisfactory to clean the data, hence I use the StopWordsRemover to drops all the stop words from the input sequences. I decided to remove stop words from the given body so that more attention could be given to the words that really give meaning to the text.



```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', 'your']
```

FIGURE 4:STOP WORDS

The list of stop words is given by the StopWords parameter and Figure 3 shows the first ten. Figure 4 clearly exhibits that the column Body, that is the original corpus, the column vector, that is the tokenized corpus and the column 'vector_no_stopw' that is the corpus tokenized and without stop words.



```
+----+--------------------+--------------------+--------------------+
|  Id|                Body|              vector|     vector_no_stopw|
+----+--------------------+--------------------+--------------------+
|  80|ive written a dat...|[ive, written, a,...|[ive, written, da...|
|  90|are there any rea...|[are, there, any,...|[really, good, tu...|
| 120|has anyone got ex...|[has, anyone, got...|[anyone, got, exp...|
| 180|this is something...|[this, is, someth...|[something, ive, ...|
| 260|i have a little g...|[i, have, a, litt...|[little, game, wr...|
| 330|i am working on a...|[i, am, working, ...|[working, collect...|
| 470|ive been writing ...|[ive, been, writi...|[ive, writing, we...|
| 580|i wonder how you ...|[i, wonder, how, ...|[wonder, guys, ma...|
| 650|i would like the ...|[i, would, like, ...|[like, version, p...|
| 810|im trying to main...|[im, trying, to, ...|[im, trying, main...|
| 930|whats the simples...|[whats, the, simp...|[whats, simplest,...|
|1010|i need to grab th...|[i, need, to, gra...|[need, grab, base...|
|1040|im looking for a ...|[im, looking, for...|[im, looking, way...|
|1070|what is the corre...|[what, is, the, c...|[correct, way, ge...|
|1160|i am using ccnet ...|[i, am, using, cc...|[using, ccnet, sa...|
|1180|i am looking to a...|[i, am, looking, ...|[looking, allow, ...|
|1300|i am using msbuil...|[i, am, using, ms...|[using, msbuild, ...|
|1390|im setting up a d...|[im, setting, up,...|[im, setting, ded...|
|1600|i always create a...|[i, always, creat...|[always, create, ...|
|1610|if im adding a co...|[if, im, adding, ...|[im, adding, colu...|
+----+--------------------+--------------------+--------------------+
```

FIGURE 5: CLEAN DATA

Implementation

Term frequency- inverse document frequency is a feature vectorization method used in textual corpus to return the relevance of a word to a corpus. Considering t as term, d as document and D as corpus. Term frequency (t,d) is the number of times that term t is present in document d, whereas document frequency DF(t,D) is the number of documents that includes term t. The TF-IDF measure is the product of TF and IDF:

$$TFIDF(t,d,D) = TF(t,d) \cdot IDF(t,D).$$

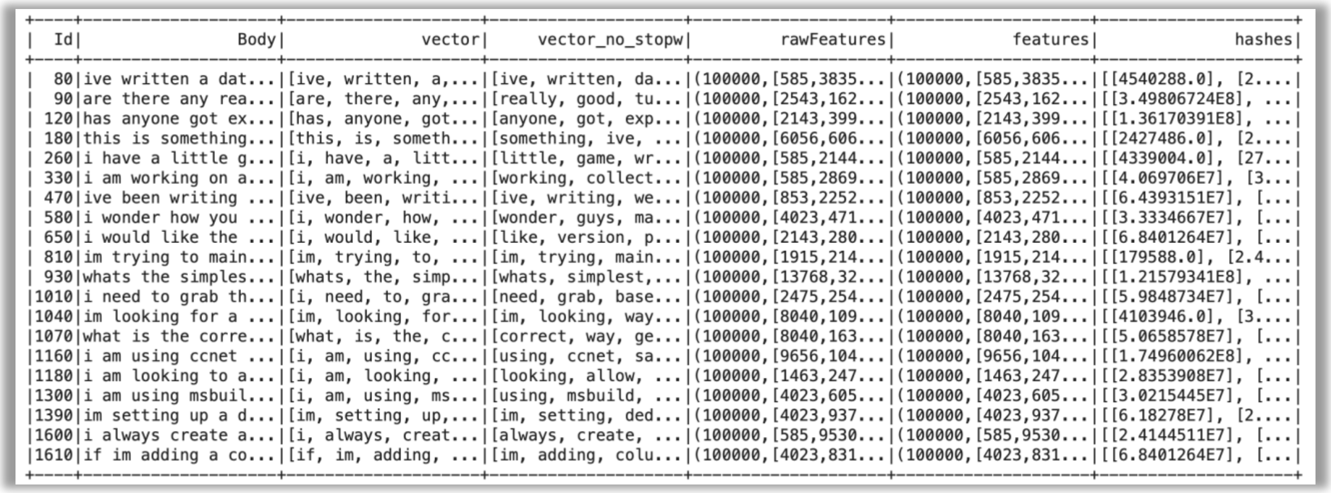
I separated TF and IDF and I have applied a hashing function: a raw feature is mapped into an index, and based on that, the term frequencies are calculated. Given the fact that the dataset is composed by a large corpus, this method escapes from the necessity to compute a global term-to- index map but it could have some hash collision, in which different raw features could became the same term after hashing. With the propose to reduce the number of collisions, I placed the target feature dimension, so the number of buckets of the hash table, equals to 100000. The application of the HashingTF requires only a single pass to the data, but IDF needs two: as first the computation of the IDF vector and secondly the scaled of the term frequencies by IDF.

Finding the similarity among text, especially when the dataset contains a huge amount of data, could be very difficult, in particular because the propose of the project is to find similar items and not items that are equal. Due to that, I decided to use the MinHash algorithm, it is designed to work with the Jaccard Distance, and it falls into the LSH family. Locally Sensitive hashing (LSH) is a hashing technique, that hash items several times, in a manner that similar items are more likely to be hashed to the same backet than the items that are different. So, I implemented the MinHash for Jaccard Distance and I put the total amount of buckets equal to 1000. The Jaccard distance is defined as d(x,y) = 1 – SIM(x,y). This means that the Jaccard distance is 1 minus the ratio of the size of the intersection and union of sets x and y.

$$d(\mathbf{A}, \mathbf{B}) = 1 - \frac{|\mathbf{A} \cap \mathbf{B}|}{|\mathbf{A} \cup \mathbf{B}|}$$

MinHash applies a random hash function *g* to each element in the set and take the minimum of all hashed values:

$$h(\mathbf{A}) = \min_{a \in \mathbf{A}}(g(a))$$



```
+----+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
|  Id|                Body|              vector|     vector_no_stopw|         rawFeatures|            features|              hashes|
+----+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
|  80|ive written a dat...|[ive, written, a,...|[ive, written, da...|(100000,[585,3835...|(100000,[585,3835...|[[4540288.0], [2....|
|  90|are there any rea...|[are, there, any,...|[really, good, tu...|(100000,[2543,162...|(100000,[2543,162...|[[3.49806724E8], ...|
| 120|has anyone got ex...|[has, anyone, got...|[anyone, got, exp...|(100000,[2143,399...|(100000,[2143,399...|[[1.36170391E8], ...|
| 180|this is something...|[this, is, someth...|[something, ive, ...|(100000,[6056,606...|(100000,[6056,606...|[[2427486.0], [2....|
| 260|i have a little g...|[i, have, a, litt...|[little, game, wr...|(100000,[585,2144...|(100000,[585,2144...|[[4339004.0], [27...|
| 330|i am working on a...|[i, am, working, ...|[working, collect...|(100000,[585,2869...|(100000,[585,2869...|[[4.069706E7], [3...|
| 470|ive been writing ...|[ive, been, writi...|[ive, writing, we...|(100000,[853,2252...|(100000,[853,2252...|[[6.4393151E7], [...|
| 580|i wonder how you ...|[i, wonder, how, ...|[wonder, guys, ma...|(100000,[4023,471...|(100000,[4023,471...|[[3.3334667E7], [...|
| 650|i would like the ...|[i, would, like, ...|[like, version, p...|(100000,[2143,280...|(100000,[2143,280...|[[6.8401264E7], [...|
| 810|im trying to main...|[im, trying, to, ...|[im, trying, main...|(100000,[1915,214...|(100000,[1915,214...|[[179588.0], [2.4...|
| 930|whats the simples...|[whats, the, simp...|[whats, simplest,...|(100000,[13768,32...|(100000,[13768,32...|[[1.21579341E8], ...|
|1010|i need to grab th...|[i, need, to, gra...|[need, grab, base...|(100000,[2475,254...|(100000,[2475,254...|[[5.9848734E7], [...|
|1040|im looking for a ...|[im, looking, for...|[im, looking, way...|(100000,[8040,109...|(100000,[8040,109...|[[4103946.0], [3....|
|1070|what is the corre...|[what, is, the, c...|[correct, way, ge...|(100000,[8040,163...|(100000,[8040,163...|[[5.0658578E7], [...|
|1160|i am using ccnet ...|[i, am, using, cc...|[using, ccnet, sa...|(100000,[9656,104...|(100000,[9656,104...|[[1.74960062E8], ...|
|1180|i am looking to a...|[i, am, looking, ...|[looking, allow, ...|(100000,[1463,247...|(100000,[1463,247...|[[2.8353908E7], [...|
|1300|i am using msbuil...|[i, am, using, ms...|[using, msbuild, ...|(100000,[4023,605...|(100000,[4023,605...|[[3.0215445E7], [...|
|1390|im setting up a d...|[im, setting, up,...|[im, setting, ded...|(100000,[4023,937...|(100000,[4023,937...|[[6.18278E7], [2....|
|1600|i always create a...|[i, always, creat...|[always, create, ...|(100000,[585,9530...|(100000,[585,9530...|[[2.4144511E7], [...|
|1610|if im adding a co...|[if, im, adding, ...|[im, adding, colu...|(100000,[4023,831...|(100000,[4023,831...|[[6.8401264E7], [...|
+----+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
```

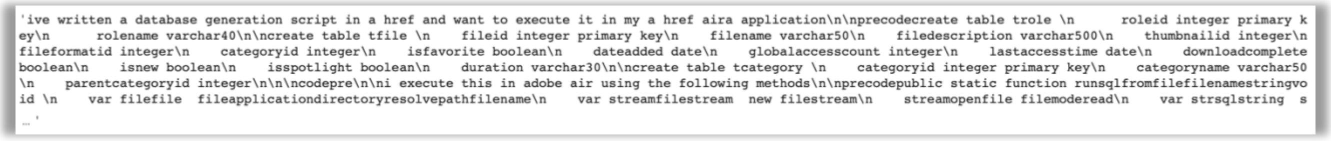
FIGURE 6: TF- IDF AND MIN HASHLSH

Approximate nearest neighbor search takes a dataset of feature vectors and a key, so a single feature vector, and it approximately returns a specified number of rows in the dataset that are closest to the vector. I used approximate nearest neighbor to obtain a good approximation of the group of similar questions. In the output dataset is present a distance column that shows the true distance between each output row and the searched key; each group is composed by 10 questions.

Results

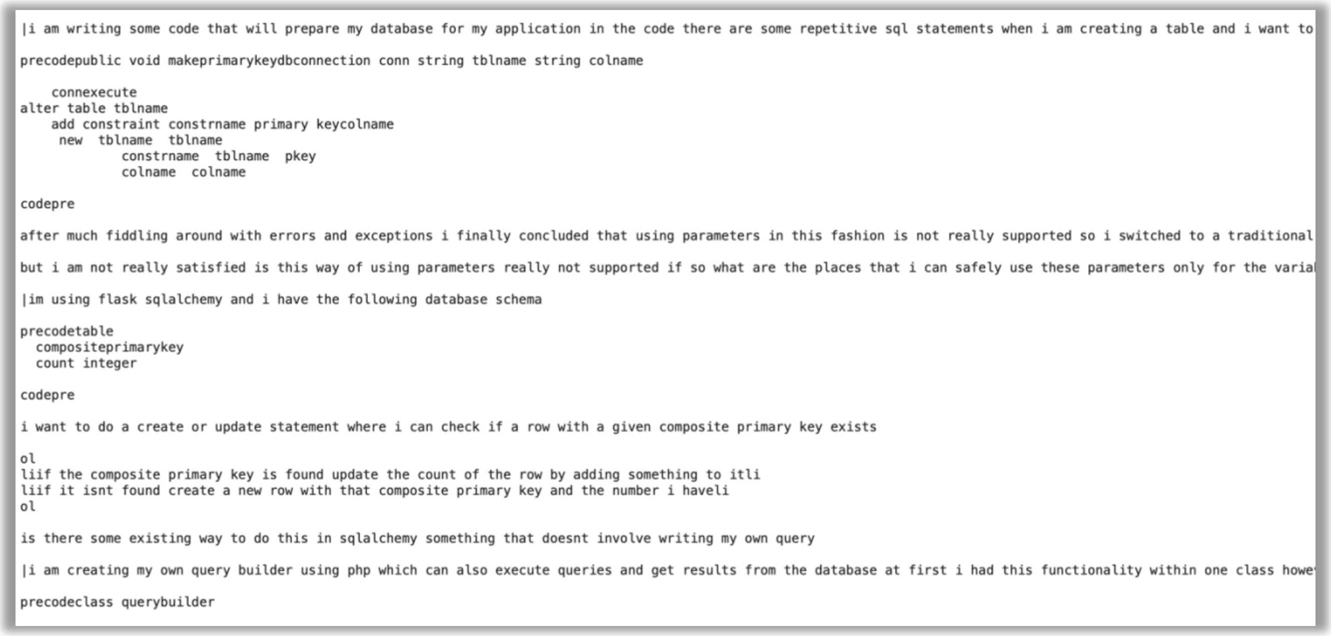
To conclude, the aim of the project was to detect pairs of similar items. The data was imported, cleaned and preprocessing. After that, the HashingTF transformer was used to count the number of tokens present in each document, and the application of MinHashLSH. At least, approximate Near- Neighbors method was implemented to achieve a grouping of similar questions, this has been done using the TF- IDF feature vectors and a single feature as key.

Figure 8 shows a sample of the matched identified two of 10 questions that are similar to the question in Figure 7, based on the assumption that the similarity is based on the presence of the same words in sentences and on the weight of these words computed through the TF-IDF function, all the questions below are related to an SQL database and contain common words in the code.



```
'ive written a database generation script in a href and want to execute it in my a href aira application\n\nprecodecreate table trole \n      roleid integer primary k
ey\n      rolename varchar40\n\ncreate table tfile \n    fileid integer primary key\n    filename varchar50\n    filedescription varchar500\n    thumbnailid integer\n
fileformatid integer\n    categoryid integer\n    isfavorite boolean\n    dateadded date\n    globalaccesscount integer\n    lastaccesstime date\n    downloadcomplete
boolean\n    isnew boolean\n    isspotlight boolean\n    duration varchar30\n\ncreate table tcategory \n    categoryid integer primary key\n    categoryname varchar50
\n    parentcategoryid integer\n\n\ncodepre\n\ni execute this in adobe air using the following methods\n\nprecodepublic static function runsqlfromfilefilenamestringvo
id \n    var filefile  fileapplicationdirectoryresolvepathfilename\n    var streamfilestream  new filestream\n    streamopenfile filemoderead\n    var strsqlstring  s
…'
```

FIGURE 7: QUESTION



```
|i am writing some code that will prepare my database for my application in the code there are some repetitive sql statements when i am creating a table and i want to

precodepublic void makeprimarykeydbconnection conn string tblname string colname

    connexecute
alter table tblname
    add constraint constrname primary keycolname
     new  tblname  tblname
             constrname  tblname  pkey
             colname  colname

codepre

after much fiddling around with errors and exceptions i finally concluded that using parameters in this fashion is not really supported so i switched to a traditional

but i am not really satisfied is this way of using parameters really not supported if so what are the places that i can safely use these parameters only for the varial

|im using flask sqlalchemy and i have the following database schema

precodetable
  compositeprimarykey
  count integer

codepre

i want to do a create or update statement where i can check if a row with a given composite primary key exists

ol
liif the composite primary key is found update the count of the row by adding something to itli
liif it isnt found create a new row with that composite primary key and the number i haveli
ol

is there some existing way to do this in sqlalchemy something that doesnt involve writing my own query

|i am creating my own query builder using php which can also execute queries and get results from the database at first i had this functionality within one class howe

precodeclass querybuilder
```

FIGURE 8: SIMILAR QUESTIONS