In [1]:
```
# Import all the required packages
import pandas as pd
from dateutil.relativedelta import relativedelta
import datetime
import time
from datetime import timedelta
from datetime import datetime
import json
import requests
import edgedb

import warnings

#dependencies for milvus
import openai
from pymilvus import Milvus


warnings.filterwarnings('ignore')
```

In [2]:
```
key = 'kdYYXHrus6XKyHIGxfR7sw(('

UNIT_TESTING = False
```

In [3]:
```
# Set the 'fromdate' to the timestamp of 10/5/2023
fromdate = str(int(datetime.timestamp(datetime(2022, 10, 5))))

# Set the 'todate' to the timestamp of 11/5/2023
todate = str(int(datetime.timestamp(datetime(2023, 11, 5))))

print(fromdate, todate)
```

```
1664946000 1699160400
```

In [4]:
```
# #fetching questions by using post_id's
tags_to_check = ["React", "Selenium", "Python", "Keras", "OpenAI", "Docker", "
questionsData = []
#https://api.stackexchange.com/2.3/questions?fromdate=1696118400&todate=169879
def fetchQuestionsData():
    try:
        for tag in tags_to_check:
            page=1
            while True:
                url='https://api.stackexchange.com/2.3/questions?order=desc&so
                response = requests.get(url)
                if not response.ok or len(response.json()) == 0 or page==15:
                    break
                response_json = json.loads(response.text)
                questionsData.append(response_json)
                page+=1
        return questionsData
    except json.decoder.JSONDecodeError:
        # Handle the case where the response is not valid JSON
        print("Invalid JSON response")
        return None
```

In [5]:
```
questionsData = fetchQuestionsData()
print(len(questionsData))
```

```
98
```

In [6]:
```
#fetching answers by using post_id's
def fetchAnswersData(post_id):
    try:
        url='https://api.stackexchange.com/2.3/questions/'+ str(post_id) + '/a
        response = requests.get(url)
        response_json = json.loads(response.text)
        return response_json
    except json.decoder.JSONDecodeError:
        # Handle the case where the response is not valid JSON
        print("Invalid JSON response for post_id:", post_id)
        return None
```

In [7]:
```
questionsDataModified=[]
for data in questionsData:
    for item in data["items"]:
        questionsDataModified.append(item)
```

In [8]:
```
answers=[]
count=0;
ids=""
for question in questionsDataModified:
    ids+=str(question["question_id"])+";"
    count+=1
    if count==100:
        answers.append(fetchAnswersData(ids[:-1]))
        ids=""
        count=0
```

In [9]:
```
answersDataModified=[]
for data in answers:
    for item in data["items"]:
        answersDataModified.append(item)
print(len(answersDataModified))
```

```
2520
```

In [10]:
```
# Create an EdgeDB client for database interactions
client = edgedb.create_client()
```

In [11]:
```
# Insert questions data into EdgeDB.

def insertQuestionsData(questionsData):
    for question in questionsData:
        owner = question.get("owner", {})
        for tag in question['tags']:
            tag_name = tag
            client.query("""
                INSERT QuestionsRelatedTagsInfo {
                    question_id:=<str>$question_id,
                    tag:=<str>$tag,
```

```
                    tagInformation := (
                        select Tags
                        filter
                            .name = <str>$tag
                            limit 1
                        )

                    }
                    """,
                    question_id = str(question.get('question_id', 0)),
                    tag = tag_name
                );
        client.query("""
            INSERT Question {
                post_id :=<int64>$post_id,
                user_id:=<int32>$user_id,
                title :=<str>$title,
                question_id :=<str>$question_id,
                creation_date :=<str>$creation_date,
                is_answered :=<str>$is_answered,
                body:=<str>$body,
                score:=<int32>$score,
                question_by_user := (
                    select User
                    filter
                        .user_id = <int32>$user_id
                    limit 1
                ),
                has_Tags :=(
                    select QuestionsRelatedTagsInfo
                    filter
                        .question_id = <str>$question_id
                    )
            }
                """,
                post_id = question.get('post_id', 0),
                user_id = owner.get("user_id", 0),
                title = question.get('title', 'Unknown Title'),
                body = question.get("body",""),
                question_id = str(question.get('question_id', '0')),
                creation_date = str(question.get('creation_date', '0')),
                is_answered = str(question.get('is_answered', 'False')),
                score = question.get('score', 0)

            );
```

In [12]:
```
insertQuestionsData(questionsDataModified)
```

In [13]:
```
# Insert answers data into EdgeDB.

def insertAnswersData(answersData):
    for answer in answersData:
        owner = answer.get("owner", {})
        client.query("""
            INSERT Answer {
                user_id:=<int32>$user_id,
                post_id :=<int64>$post_id,
                answer_id := <str>$answer_id,
                body:= <str>$body,
```

```
                question_id :=<str>$question_id,
                account_id := <str>$account_id,
                reputation := <str>$reputation,
                score :=<int32>$score,
                creation_date :=<str>$creation_date,
                to_Question := (
                    select Question
                    filter
                        .question_id = <str>$question_id
                        limit 1
                ),
                by_user := (
                    select User
                    filter
                        .user_id = <int32>$user_id
                    limit 1
                )
            }
                """,
                user_id = owner.get("user_id", 0),
                post_id = answer.get("post_id", 0),
                account_id = str(owner.get("account_id", None)),
                reputation = str(owner.get("reputation", None)),
                body = answer.get("body"),
                score = answer.get("score", 0),
                creation_date = str(answer.get("creation_date", None)),
                answer_id = str(answer.get("answer_id", None)),
                question_id = str(answer.get("question_id", None))

            );
```

In [14]:
```
insertAnswersData(answersDataModified)
```

In [15]:
```
posts=[]
for question in questionsDataModified:
    questionObject={
        "postid":question["question_id"],
        "score":question["score"],
        "creation_date" : question["creation_date"],
        "last_activity_date":question["last_activity_date"],
        "post_type" : "question",
        "user_id": question["owner"].get("user_id"),
    }
    posts.append(questionObject)
for answer in answersDataModified:
    answerObject={
        "postid":answer["answer_id"],
        "score":answer["score"],
        "creation_date" : question["creation_date"],
        "last_activity_date":answer["last_activity_date"],
        "post_type" : "answer",
        "user_id": answer["owner"].get("user_id"),
    }
    posts.append(answerObject)
```

In [16]:
```
def insertPostData(data):
    for post_data in data:
        post_id = post_data.get("postid")
        score = post_data.get("score", 0)
```

```
        last_activity_date = post_data.get("last_activity_date", 0)
        creation_date = post_data.get("creation_date", 0)
        post_type = post_data.get("post_type", "unknown")
        #user_data = post_data.get("owner")
        user_id = post_data.get("user_id")

        #print(user_id)
        if user_id is not None and user_id != 0:
            client.query("""
                    INSERT Post {
                        post_id := <str>$post_id,
                        score := <int32>$score,
                        last_activity_date := <int64>$last_activity_date,
                        creation_date := <int64>$creation_date,
                        post_type := <str>$post_type,
                        user_id := <int32>$user_id,
                        to_Question := (
                                select Question
                                filter
                                    .question_id = <str>$post_id
                                    limit 1
                            ),
                        to_Answer := (
                                select Answer
                                filter
                                    .answer_id = <str>$post_id
                                    limit 1
                            ),
                        };
                    """,
                     post_id=str(post_id),
                     score=score,
                     last_activity_date=last_activity_date,
                     creation_date=creation_date,
                     post_type=post_type,
                     user_id=user_id

                );
```

In [17]:
```
insertPostData(posts)
```

In [18]:
```
#Add your code for this requirement in this cell
import networkx as nx
import matplotlib.pyplot as plt

# Create a graph object
G = nx.DiGraph()

# Add nodes for each object
G.add_node('Tags')
G.add_node('Question')
G.add_node('Answer')
G.add_node('User')
G.add_node('Post')
G.add_node('Comments')
G.add_node('Badge')
G.add_node('TagSynonyms')
G.add_node('UserRelatedTagInfo')
```

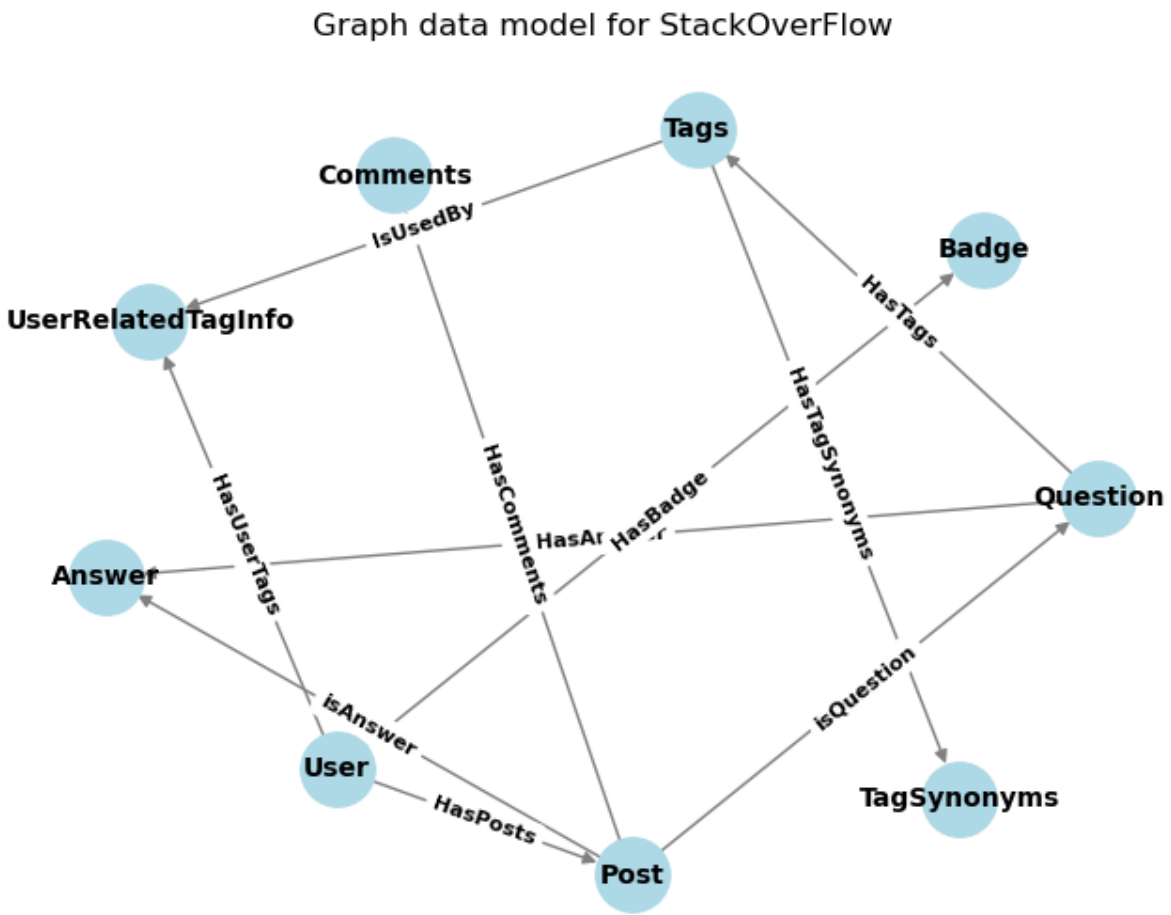
```
# Add edges for each link
G.add_edge('Question', 'Tags', label='HasTags')
G.add_edge('Question', 'Answer', label='HasAnswer')
G.add_edge('User', 'Post', label = 'HasPosts')
G.add_edge('User', 'Badge', label = 'HasBadge')
G.add_edge('Post', 'Question', label = 'isQuestion')
G.add_edge('Post', 'Answer', label = 'isAnswer')
G.add_edge('Post', 'Comments', label = 'HasComments')
G.add_edge('Tags', 'TagSynonyms', label = 'HasTagSynonyms')
G.add_edge('User', 'UserRelatedTagInfo', label = 'HasUserTags')
G.add_edge('Tags', 'UserRelatedTagInfo', label = 'IsUsedBy')




# Draw the graph
pos = nx.spring_layout(G, k = 20, seed=20)
nx.draw(G, pos, with_labels=True, font_size=10, node_size=800, node_color='ligl

# Add edge labels
edge_labels = nx.get_edge_attributes(G, 'label')
nx.draw_networkx_edge_labels(G, pos, edge_labels, font_size=8, font_weight='bo

plt.title("Graph data model for StackOverFlow")
plt.show()
```



In [ ]:

In [19]:
```
# ## Installing Dependencies
import openai
```

In [20]:
```
HOST = 'localhost'
PORT = 19530
COLLECTION_NAME = 'postData'
DIMENSION = 1536
OPENAI_ENGINE = 'text-embedding-ada-002'
openai.api_key = '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'

index_params = {
    'index_type': 'IVF_FLAT',
    'metric_type': 'L2',
    'params': {'nlist': 1024}
}

QUERY_PARAM = {
    "metric_type": "L2",
    "params": {"ef": 64},
}

BATCH_SIZE = 1000
```

In [21]:
```
import edgedb
```

In [22]:
```
# Connect to EdgeDB
client = edgedb.create_client()

posts_query = """
SELECT Post {
  creation_date,
  last_activity_date,
  post_type,
  score,
  user_id,
  post_id,
  to_Question: {
    title,
    body
  },
  to_Answer: {
    body,
    reputation,
    to_Question: {
      body,
      title
    }
  }
}"""
# Execute the query and fetch the results
result = client.query(posts_query)

# Print or process the result as needed
print (len(result))
```

```
10918
```

In [23]:
```
from pymilvus import (
    connections,
    utility,
    FieldSchema,
    CollectionSchema,
    DataType,
    Collection,
)

# Connect to Milvus Database
connections.connect(host="localhost", port="19530",timeout=60)  # Adjust the t
print("Connection Successfull")

# Remove collection if it already exists
if utility.has_collection(COLLECTION_NAME):
    utility.drop_collection(COLLECTION_NAME)
```

```
Connection Successfull
```

In [24]:
```
# Create collection which includes the id, title, and embedding.
fields = [
    FieldSchema(name="post_id", dtype=DataType.VARCHAR, is_primary=True, max_l
    FieldSchema(name="post_type", dtype=DataType.VARCHAR, max_length=64000),
    FieldSchema(name="user_id", dtype=DataType.INT64),
    FieldSchema(name='title', dtype=DataType.VARCHAR, max_length=64000),
    FieldSchema(name='body', dtype=DataType.VARCHAR, max_length=64000),
    FieldSchema(name="creation_date", dtype=DataType.VARCHAR, max_length=64000
    FieldSchema(name="last_activity_date", dtype=DataType.VARCHAR, max_length=
    FieldSchema(name="score", dtype=DataType.INT64),
    FieldSchema(name="reputation", dtype=DataType.INT64),
    FieldSchema(name="embedding", dtype=DataType.FLOAT_VECTOR, dim=1536)
]


schema = CollectionSchema(fields=fields,description='Posts Data')
collection = Collection(name=COLLECTION_NAME, schema=schema)
```

In [ ]:
```

```

In [25]:
```
# Create the index on the collection and load it.
collection.create_index(field_name="embedding", index_params=index_params)
collection.load()
```

In [ ]:
```

```

In [26]:
```
# Fetch embeddings for each Review to insert into Milvus
import openai
from openai import OpenAI
openAiClient = OpenAI(api_key='sk-EBbbUsNBSrtBk5iS9aLdT3BlbkFJB79Xw8NzRfUp5d4Y
def embed(text_to_embed):
    # Embed a line of text
    response = openAiClient.embeddings.create(input=text_to_embed, model=OPENA

    # Return the list of embeddings
    return response.data[0].embedding
```

In [27]:
```
from tqdm import tqdm
from datetime import datetime
```

```
import openai

BATCH_SIZE = 100

data = [[],[],[],[],[],[],[],[],[],[]]

for i in tqdm(range(0,len(result))):
    data[0].append(result[i].post_id or '')
    data[1].append(result[i].post_type or '')
    data[2].append(result[i].user_id or -1)

    if(result[i].post_type == "question"):
        data[3].append(result[i].to_Question.title or '')
        data[4].append(result[i].to_Question.body or '')
    else:
        data[3].append('') #answers do not have a title
        data[4].append(result[i].to_Answer.body or '')


    created_timestamp = int(result[i].creation_date)
    creation_date = datetime.utcfromtimestamp(created_timestamp).strftime("%A,
    data[5].append(creation_date or '')

    last_activity_timestamp = int(result[i].last_activity_date)
    last_activity_date = datetime.utcfromtimestamp(last_activity_timestamp).st
    data[6].append(last_activity_date)


    data[7].append(result[i].score or 0)
    data[8].append(result[i].score or 0)

    dataToEmbed = str(data[3][-1] + " " + data[4][-1])

    if len(dataToEmbed) > 8000:
        dataToEmbed = dataToEmbed[:8000]

    embeddings = embed(dataToEmbed)

    data[9].append(embeddings)


    if len(data[0]) % BATCH_SIZE == 0:
        collection.insert(data)
        data = [[],[],[],[],[],[],[],[],[],[]]

# Embed and insert the remainder

if len(data[0]) != 0:
    collection.insert(data)
    data = [[],[],[],[],[],[],[],[],[],[]]

print("Insertion Success!")
```

```
100%|██████████████████████████████████████| 10918/10918 [35:52<00:00,  5.07it/
s]
Insertion Success!
```

In [ ]:

In [84]:
```
import textwrap
from datetime import datetime

def query(query, top_k = 5):
    text, expr= query
    res = collection.search([embed(text)], anns_field='embedding', expr=expr, |
    for i, hit in enumerate(res):
        print(f'\nDisplaying Top {top_k} Results for query "{text}"')
        for ii, hits in enumerate(hit):
            print('\t' + 'Rank:', ii + 1, '| Score:', hits.score)
            print('\t\t' + '  Title:', hits.entity.get('title'))
            print('\t\t' + 'Post id:', hits.entity.get('post_id'))
            print('\t\t' + 'Post type:', hits.entity.get('post_type'))
#             creation_date = datetime.strptime(hits.entity.get('creation_date
            print('\t\t' + 'Creation Date:',creation_date)
#             last_activity_date = datetime.strptime(hits.entity.get('last_acti
            print('\t\t' + 'Last Activity Date:', last_activity_date)
            print('\t\t' + 'User id:', hits.entity.get('user_id'))
            print()
            print("\n")
```

In [89]:
```
#Query 1:
query(('Keras LSTM state', 'creation_date >= 2023-10-5 and creation_date <= "2(
```

```
Displaying Top 5 Results for query "Keras LSTM state"
        Rank: 1 | Score: 0.3569783866405487
                  Title: Difficulties training an LSTM for Straightforward Tim
e Series Predictions
                Post id: 77231180
                Post type: question
                Creation Date: Monday, November 21, 2022 03:08 PM
                Last Activity Date: Sunday, October 08, 2023 12:08 AM
                User id: 5314329



        Rank: 2 | Score: 0.372653603553772
                  Title: LSTM sequence2sequence translation using keras
                Post id: 76436106
                Post type: question
                Creation Date: Monday, November 21, 2022 03:08 PM
                Last Activity Date: Sunday, October 08, 2023 12:08 AM
                User id: 14636579



        Rank: 3 | Score: 0.3736841082572937
                  Title: What is the inputs to the each layer of the stacked L
STM?
                Post id: 77424342
                Post type: question
                Creation Date: Monday, November 21, 2022 03:08 PM
                Last Activity Date: Sunday, October 08, 2023 12:08 AM
                User id: 22531503



        Rank: 4 | Score: 0.37758201360702515
                  Title: Shape problems in LSTM; Configuring LSTM properly in
Keras
                Post id: 77376219
                Post type: question
                Creation Date: Monday, November 21, 2022 03:08 PM
                Last Activity Date: Sunday, October 08, 2023 12:08 AM
                User id: 22812618
```

In [87]:
```
#Query 2:
query(('Docker nginx', 'creation_date >= 2023-10-5 and creation_date <= 2023-11
# query(my_query2)
```

```
Displaying Top 5 Results for query "Docker nginx"
        Rank: 1 | Score: 0.2811281681060791
                  Title: Nextjs nginx docker configuration
                Post id: 77323678
                Post type: question
                Creation Date: Monday, November 21, 2022 03:08 PM
                Last Activity Date: Sunday, October 08, 2023 12:08 AM
                User id: 9560002



        Rank: 2 | Score: 0.28543519973754883
                  Title: Docker with Nginx php-fpm does not load images, css o
r js
                Post id: 77333362
                Post type: question
                Creation Date: Monday, November 21, 2022 03:08 PM
                Last Activity Date: Sunday, October 08, 2023 12:08 AM
                User id: 21250316



        Rank: 3 | Score: 0.2967311143875122
                  Title: Nginx not working with Docker, Django and Gunicorn
                Post id: 77392858
                Post type: question
                Creation Date: Monday, November 21, 2022 03:08 PM
                Last Activity Date: Sunday, October 08, 2023 12:08 AM
                User id: 4818882



        Rank: 4 | Score: 0.30499452352523804
                  Title: Properly deploying Nginx+React+FastAPI+SQLite+Docker
keeping things secure (protecting the backend etc.)
                Post id: 77423200
                Post type: question
                Creation Date: Monday, November 21, 2022 03:08 PM
                Last Activity Date: Sunday, October 08, 2023 12:08 AM
                User id: 18601287
```

In [88]:
```
#Query 3:
query(('Python OpenAI API Key', 'creation_date >= 2023-10-5 and creation_date <
query(my_query3)
```

```
Displaying Top 5 Results for query "Python OpenAI API Key"
        Rank: 1 | Score: 0.25997409224510193
                  Title:
                Post id: 76594109
                Post type: answer
                Creation Date: Monday, November 21, 2022 03:08 PM
                Last Activity Date: Sunday, October 08, 2023 12:08 AM
                User id: 22158775



        Rank: 2 | Score: 0.29740235209465027
                  Title: How to check the validity of the OpenAI key from pyth
on?
                Post id: 76522693
                Post type: question
                Creation Date: Monday, November 21, 2022 03:08 PM
                Last Activity Date: Sunday, October 08, 2023 12:08 AM
                User id: 2641825



        Rank: 3 | Score: 0.2986624836921692
                  Title: Problem with Azure Api and ChatGPT (python)
                Post id: 76596008
                Post type: question
                Creation Date: Monday, November 21, 2022 03:08 PM
                Last Activity Date: Sunday, October 08, 2023 12:08 AM
                User id: 13390415



        Rank: 4 | Score: 0.30952194333076477
                  Title: OpenAI module keeps giving me an error when I try to
use it with the API key
                Post id: 77003415
                Post type: question
                Creation Date: Monday, November 21, 2022 03:08 PM
                Last Activity Date: Sunday, October 08, 2023 12:08 AM
                User id: 22269214



        Rank: 5 | Score: 0.31298166513442993
                  Title:
                Post id: 76370913
                Post type: answer
                Creation Date: Monday, November 21, 2022 03:08 PM
                Last Activity Date: Sunday, October 08, 2023 12:08 AM
                User id: 2466958



Displaying Top 5 Results for query "Python OpenAI API Key"
        Rank: 1 | Score: 0.2598593235015869
                  Title:
                Post id: 76594109
                Post type: answer
                Creation Date: Monday, November 21, 2022 03:08 PM
```

```
                Last Activity Date: Sunday, October 08, 2023 12:08 AM
                User id: 22158775



        Rank: 2 | Score: 0.29736989736557007
                  Title: How to check the validity of the OpenAI key from pyth
on?
                Post id: 76522693
                Post type: question
                Creation Date: Monday, November 21, 2022 03:08 PM
                Last Activity Date: Sunday, October 08, 2023 12:08 AM
                User id: 2641825



        Rank: 3 | Score: 0.29854780435562134
                  Title: Problem with Azure Api and ChatGPT (python)
                Post id: 76596008
                Post type: question
                Creation Date: Monday, November 21, 2022 03:08 PM
                Last Activity Date: Sunday, October 08, 2023 12:08 AM
                User id: 13390415



        Rank: 4 | Score: 0.30944961309432983
                  Title: OpenAI module keeps giving me an error when I try to
use it with the API key
                Post id: 77003415
                Post type: question
                Creation Date: Monday, November 21, 2022 03:08 PM
                Last Activity Date: Sunday, October 08, 2023 12:08 AM
                User id: 22269214



        Rank: 5 | Score: 0.3129372000694275
                  Title:
                Post id: 76370913
                Post type: answer
                Creation Date: Monday, November 21, 2022 03:08 PM
                Last Activity Date: Sunday, October 08, 2023 12:08 AM
                User id: 2466958
```