

Compass

My Queries Data Modeling

CONNECTIONS (2)

- cluster0.rqommrg.mongodb.net
- localhost:27017
 - admin
 - config
 - employees
 - local
 - schooldb
 - socialmedia**
- reddit**
- x
- test

localhost:27017 > socialmedia > reddit

Documents 7.2K Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#) + Explain Reset Find Options ↗

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

25 1 - 25 of 7194

```
_id: ObjectId('691029aae6013e37e8fac9dc')
author : "CuriousWanderer567"
likes : 21170
post_score : 21170
down_votes : 0
title: "This tortoise's reaction to the darker colored material of this person..."
 subreddit: "interesting"
url : "https://v.redd.it/lvl2ugaws40g1"
location : "MISC."
```

```
_id: ObjectId('691029aae6013e37e8fac9dd')
author : "flat-Limit5595"
likes : 14176
post_score : 14176
down_votes : 0
title: "Mom with the cat she didn't want"
 subreddit: "cats"
url : "https://www.reddit.com/gallery/1os6mpn"
location : "Humor"
```

```
_id: ObjectId('691029aae6013e37e8fac9de')
author : "montageidiots"
likes : 8620
post_score : 8620
down_votes : 0
title: "[Highlight] De'Andre Hunter breaks Giddey's ankles and then put Patric..."
 subreddit: "nba"
url : "https://streamable.com/dy092f"
location : "Highlight"
```

```
_id: ObjectId('691029aae6013e37e8fac9df')
author : "TrustMeIALawyer"
likes : 16419
```

The screenshot shows a code editor interface with the following details:

- Top Bar:** Includes tabs for "backend.py", "database.py", and "database_adddata.py".
- Search Bar:** Contains "social media analytics".
- Left Sidebar (Explorer):** Shows a tree view of files under "SOCIAL MEDIA ANALYTICS".
 - "backendsrc":
 - __pycache__
 - database_adddata.py
 - database.py
 - logs.txt
 - test.py
 - backend.py
 - logs.txt
 - OUTLINE
 - TIMELINE
- Code Editor Area:** Displays the content of the "database.py" file.
- Bottom Status Bar:** Shows "Ln 9, Col 72" and other system information like "Spaces: 4", "UTF-8", "LF", "Python 3.13.5", "Go Live", and "3.13.5".

```
import mysql.connector
import pymongo
import datetime

class Database_connector:
    def __init__(self):
        self.conn = None
        self.cursor = None
        self.client = pymongo.MongoClient("mongodb://localhost:27017/")
    @staticmethod
    def write_logs(text , error=None):
        file = open('logs.txt','a')
        write = f'{datetime.datetime.now()} {text} {error}\n'
        file.write(write)
        file.close()
    #connecting to mysql database
    def connect_sql_db(self):
        try:
            self.conn = mysql.connector.connect(user='root',password ='harshiljain',database='socialmedia')
            self.cursor = self.conn.cursor()
            Database_connector.write_logs(' successfully connected to database')
        except Exception as e:
            Database_connector.write_logs(' error occurred in connecting to the database: ',e)
    #Connecting to mongodb database
    def connect_mongo_db(self):
        try:
            db = self.client['socialmedia']
            Database_connector.write_logs('Successfully connected to mongodb')
        except Exception as e:
            Database_connector.write_logs('Not able to connect to the mongodb database ',e)
    #function to add documents in mongodb
    def add_data_mongo(self,collection,data):
        try:
            db = self.client['socialmedia']
            col = db[collection]
            col.insert_many(data)
            Database_connector.write_logs('successfully added to mongodb database')
        except Exception as e:
            Database_connector.write_logs('cannot add data to mongodb collection ',e)
    ...
    database = Database_connector()
    database.connect_sql_db()
    database.connect_mongo_db()
    database.add_data_mongo('x')
```

The screenshot shows a code editor interface with several windows open. The main window displays a Python script named `database_adddata.py`. The script is part of a project structure under `SOCIAL MEDIA ANALYTICS`, specifically within the `backendsrc` directory. The code implements a `Data_Loader` class to handle data from Instagram and Reddit. It uses the praw library to interact with Reddit and the `Database_connector` to store data in a MongoDB database. The code includes functions for loading data from both platforms and processing trends and subreddit posts.

```
backend.py    database.py    database_adddata.py < social media analytics
```

```
backendsrc > database_adddata.py > Data_Loader > load_data_instagram
  9  class Data_Loader(Database_connector):
12      def load_data_instagram(self):
13          count = 0
14          for post in posts:
15              if count >= 100:
16                  break
17
18              doc = {
19                  "platform": "instagram",
20                  "id": post.shortcode,
21                  "title": post.caption or "",
22                  "likes": post.likes,
23                  "comments": post.comments,
24                  "author": post.owner_username,
25                  "url": f"https://www.instagram.com/p/{post.shortcode}/",
26                  "created_utc": datetime.utcnow().timestamp(),
27                  "fetched_at": datetime.utcnow()
28              }
29
30              super().add_data_mongo('reddit', doc)
31
32          print(f"Saved {count} Instagram posts")
33      def load_data_reddit(self):
34          reddit = praw.Reddit(client_id='mep06j61ZvyKZImjRqf_mQ', client_secret='hdY0M4NjswAOtUHXdobimlDs9rD7Q', user_agent='social media')
35          ...
36          trend analysis: likes, karma, title(subject detection, locations)
37          ...
38          subreddit = reddit.subreddit('all')
39          data = []
40          for post in subreddit.hot(limit=100000):
41              post_data = {
42                  "author": str(post.author) if post.author else None,
43                  "likes": post.upvotes,
44                  "post_score": post.score,
45                  "down_votes": post.upvotes - post.score,
46                  "title": post.title,
47                  "subreddit": str(post.subreddit),
48                  "url": post.url,
49                  "location": None
50              }
51
52              if hasattr(post, "link_flair_text") and post.link_flair_text:
53                  post_data["location"] = post.link_flair_text
54
55              data.append(post_data)
56
57      if data:
58          trv:
```

The screenshot shows a code editor interface with the following details:

- EXPLORER** sidebar:
 - SOCIAL MEDIA ANALYTICS** folder:
 - backendsrc
 - _pycache_
 - database_adddata.py
 - database.py
 - logs.txt
 - test.py
 - backend.py** (selected)
 - logs.txt
 - OUTLINE
 - TIMELINE
- SEARCH** sidebar:
 - Search (highlighted)
 - Replace
- Code Editor**:
 - File tabs: backend.py, database.py, database_adddata.py
 - Code content:

```
import pymongo
import pandas as pd
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import LatentDirichletAllocation
import matplotlib.pyplot as plt
import datetime

def download_nltk_assets():
    try:
        nltk.data.find('corpora/stopwords')
    except LookupError:
        print("Downloading NLTK 'stopwords'...")
        nltk.download('stopwords')
    try:
        nltk.data.find('corpora/wordnet')
    except LookupError:
        print("Downloading NLTK 'wordnet'...")
        nltk.download('wordnet')
    try:
        nltk.data.find('tokenizers/punkt')
    except LookupError:
        print("Downloading NLTK 'punkt'...")
        nltk.download('punkt')
    try:
        nltk.data.find('tokenizers/punkt_tab')
    except LookupError:
        print("Downloading NLTK 'punkt_tab'...")
        nltk.download('punkt_tab')

def fetch_reddit_data(mongo_uri="mongodb://localhost:27017/", db_name="socialmedia", collection_name="reddit"):
    print(f"Connecting to MongoDB at {mongo_uri}...")
    try:
        client = pymongo.MongoClient(mongo_uri)
        db = client[db_name]
        collection = db[collection_name]

        cursor = collection.find({}, {"title": 1})
        df = pd.DataFrame(list(cursor))
    except Exception as e:
        print(f"Error connecting to MongoDB: {e}")
        return None
```
 - Status bar: Ln 159, Col 55, Spaces: 4, UTF-8, LF, Python, 3.13.5, Go Live