Task No: 12 A Date:	Retrieve Analytic Information given from MongoDB i. For each “place_type”, Find total favorite_count ii. For each "country_code", find total "retweet_count"	CO5 K3

AIM:

To Retrieve Analytic Information given from MongoDB, for each Place Type, Country code, retweet count.

PROCEDURE:

1. Install pymongo in Pycharm IDE
2. Import the pymongo module in the analytics program
3. Configure the mongodb server using mongodb compass
4. Upload the country dataset in to Mongodb
5. Get the MongoDB URI, Database name and Collection Name
6. Establish the MongoDB connection using MongoDB URI, Database name and Collection Name
7. Aggregation pipeline to calculate the total favorite_count, retweet_count
8. Generate the results using aggregate functions.
9. Print the results, total favorite_count, retweet_count

PROGRAM:

```
import pymongo
mongo_uri = "mongodb://localhost:27017/"
database_name = "Country"
collection_name = "Tweets"
client = pymongo.MongoClient(mongo_uri)
db = client[database_name]
collection = db[collection_name]

pipeline = [
    {
        "$group": {
            "_id": "$country_code",
            "total_favorite_count": {"$sum": "$favorite_count"},
            "total_retweet_count": {"$sum": "$retweet_count"}
        }
    }
]

result = list(collection.aggregate(pipeline))

if result:
    total_favorite_count = result[0]["total_favorite_count"]
    total_retweet_count = result[1]["total_retweet_count"]
    print(f"Total favorite_count: {total_favorite_count}")
    print(f"Total_retweet_count: {total_retweet_count}")
else:
    print("No tweets found in the collection.")

# Close the MongoDB connection
client.close()
```

SAMPLE DATASET:

country_code	**latitude**	**longitude**	**country_name**	**place_type**	**favorite_count**	**retweet_count**
AE	23.424076	53.847818	United Arab Emirates	Residential	7	4
AE	23.424076	53.847818	United Arab Emirates	Commercial	5	2
AR	-38.416097	-63.616672	Argentina	Educational	4	6
AR	-38.416097	-63.616672	Argentina	Cultural	5	5
IN	20.593684	78.96288	India	Religious	80	3
IN	20.593684	78.96288	India	Historical	100	6
TH	15.870032	100.992541	Thailand	Shopping	2	5
IN	20.593684	78.96288	India	Sports	55	3
AO	-11.202692	17.873887	Angola	Residential	2	6
AQ	-75.250973	-0.071389	Antarctica	Commercial	4	4
AR	-38.416097	-63.616672	Argentina	Educational	5	9
AS	-14.270972	-170.132217	American Samoa	Cultural	3	6
BD	23.684994	90.356331	Bangladesh	Religious	0	4
AU	-25.274398	133.775136	Australia	Historical	8	4
AW	12.52111	-69.968338	Aruba	Shopping	2	8
AZ	40.143105	47.576927	Azerbaijan	Sports	4	5
EG	26.820553	30.802498	Egypt	Tourist	2	2
IN	20.593684	78.96288	India	Tourist	8	5
BD	23.684994	90.356331	Bangladesh	Religious	9	2

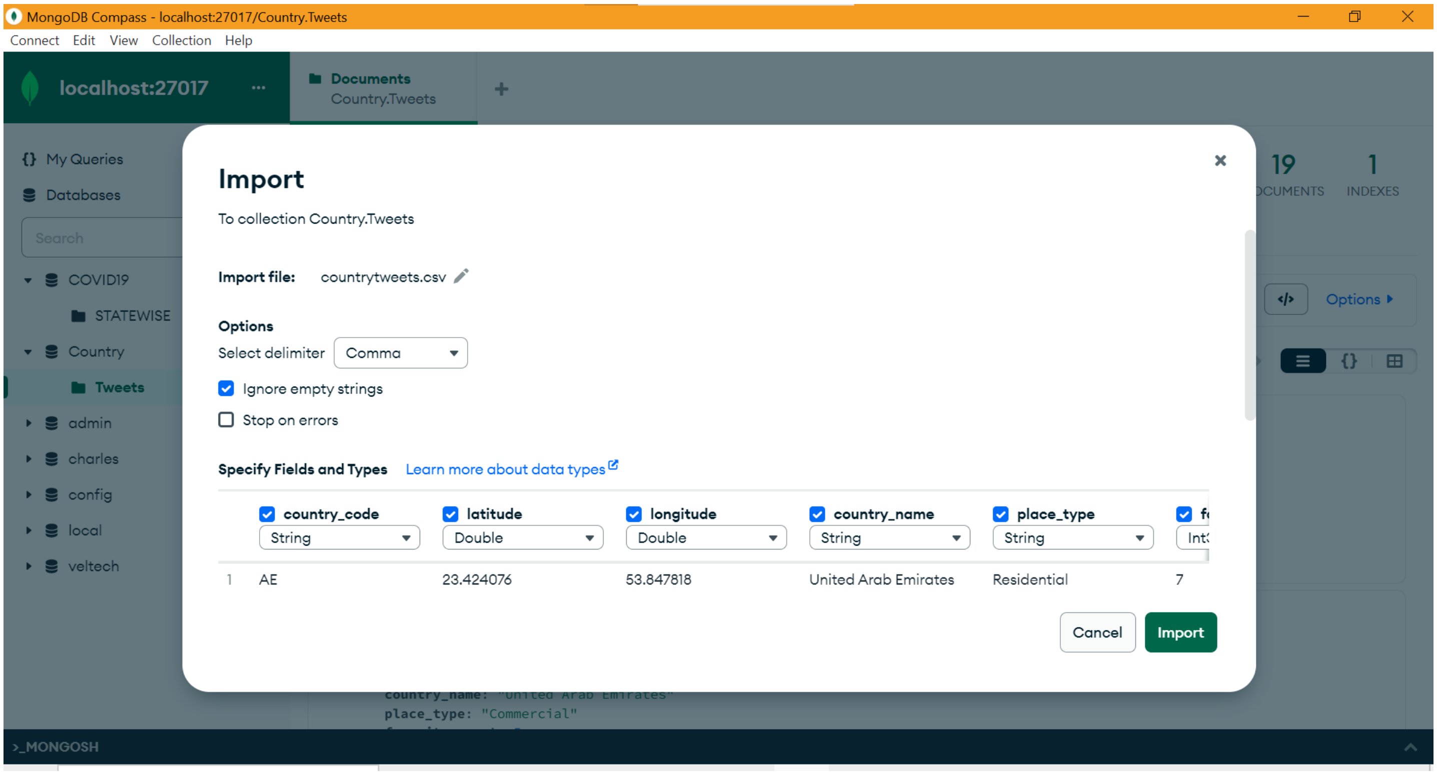
OUTPUT:

```
Total favorite_count: 3

Total_retweet_count: 6
```

RESULT:

Thus the program to Retrieve Analytic Information given from MongoDB, for each Place Type, Country code, retweet count was executed successfully.



Import

To collection Country.Tweets

Import file: countrytweets.csv

Options

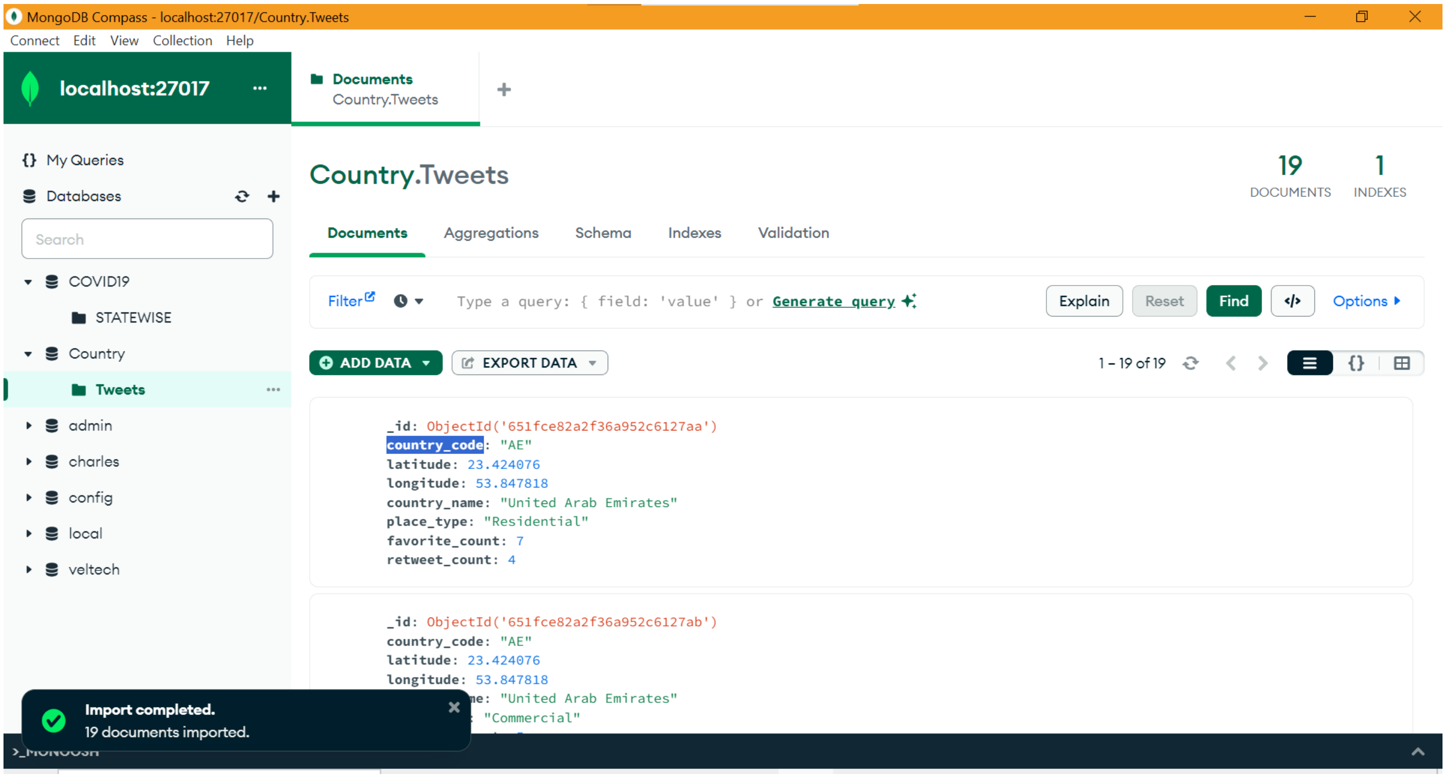
Select delimiter: Comma

☑ Ignore empty strings

☐ Stop on errors

Specify Fields and Types Learn more about data types

	☑ country_code	☑ latitude	☑ longitude	☑ country_name	☑ place_type	☑ fa
	String	Double	Double	String	String	Int3
1	AE	23.424076	53.847818	United Arab Emirates	Residential	7



```
_id: ObjectId('651fce82a2f36a952c6127aa')
country_code: "AE"
latitude: 23.424076
longitude: 53.847818
country_name: "United Arab Emirates"
place_type: "Residential"
favorite_count: 7
retweet_count: 4

_id: ObjectId('651fce82a2f36a952c6127ab')
country_code: "AE"
latitude: 23.424076
longitude: 53.847818
...ne: "United Arab Emirates"
...: "Commercial"
```

Import completed.
19 documents imported.

Task No: 12 B Date:	Find out top 10 most frequent topic words of the entire tweet message texts of your collection after lemmatization/stemming and removing all the Stop Words.	CO5 K3

AIM:

To find the top 10 most frequent topic words of the entire tweet message texts after lemmatization, stemming and removing stop words.

PROCEDURE:

1. Install NLTK data in the PyCharm IDE
2. Import the NLTK libraries WordNetLemmatizer, PorterStemmer, stopwords
3. Read the sample tweet message from Mongodb or assign tweet = ‘sample message’
4. Tokenize the tweet into words
5. Initialize lemmatizer and stemmer
6. Lemmatize and stem each word
7. Remove stop words
8. Define the list of English stop words
9. Join the filtered words back into a sentence
10. Print the results

PROGRAM:

```
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.stem import PorterStemmer
from nltk.corpus import stopwords

nltk.download('punkt')
nltk.download('wordnet')
nltk.download('stopwords')

tweet = "The quick brown foxes are jumping over the lazy dogs' bones."

words = nltk.word_tokenize(tweet)
lemmatizer = WordNetLemmatizer()
stemmer = PorterStemmer()
lemmatized_words = [lemmatizer.lemmatize(word) for word in words]
stemmed_words = [stemmer.stem(word) for word in words]
words = nltk.word_tokenize(tweet)

stop_words = set(stopwords.words('english'))
filtered_words = [word for word in words if word.lower() not in stop_words]
filtered_text = ' '.join(filtered_words)
print("Original words:", words)
print("Lemmatized words:", lemmatized_words)
print("Stemmed words:", stemmed_words)
print("Tweet without stop words:", filtered_text)
```

OUTPUT:

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Original words: ['The', 'quick', 'brown', 'foxes', 'are', 'jumping', 'over', 'the', 'lazy', 'dogs', "'", 'bones', '.']
Lemmatized words: ['The', 'quick', 'brown', 'fox', 'are', 'jumping', 'over', 'the', 'lazy', 'dog', "'", 'bone', '.']
Stemmed words: ['the', 'quick', 'brown', 'fox', 'are', 'jump', 'over', 'the', 'lazi', 'dog', "'", 'bone', '.']
Tweet without stop words: quick brown foxes jumping lazy dogs  bones .
```

RESULTS:

Thus the Program to find the top 10 most frequent topic words of the entire tweet message texts after lemmatization, stemming and removing stop words was executed successfully.