

بسمه تعالی

تمرین نهم

کلان داده‌ها

الاستیک سرچ – کاساندرا – کیبانا

عارف یزدخواستی

4013644018

## گام اول:

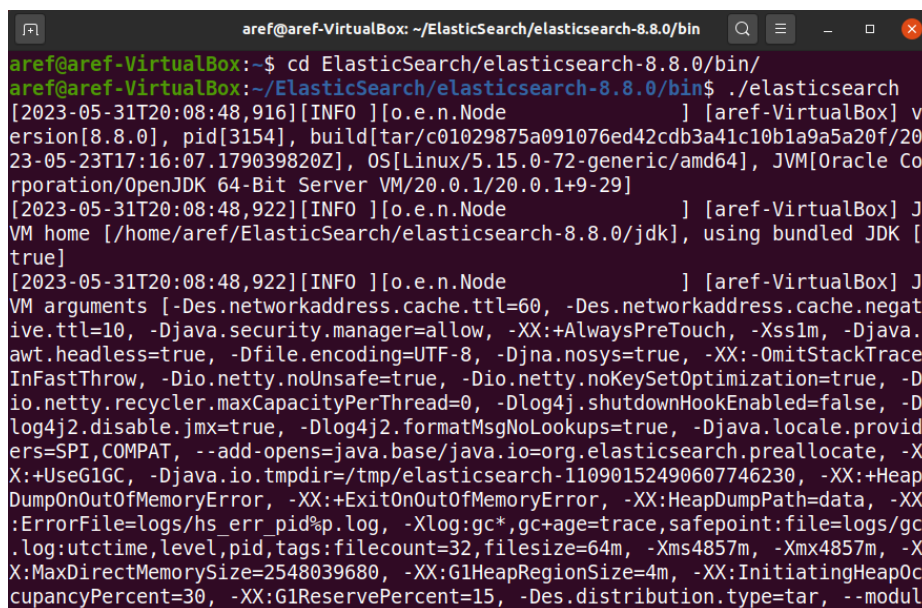
پس از دانلود و اجرای الاستیک سرچ، داده‌های موجود در مجموعه داده در دسترس را در الاستیک سرچ ذخیره کنید. سپس به سوالات زیر پاسخ دهید:

در ابتدا نیاز به نصب ابزارهای کاساندر، کیبانا و الاستیک سرچ داریم. این ابزارها هر کدام با استفاده از دانلود سورسشان از وبسایت رسمی خود قابل اجرا هستند.

کانفیگ خاصی برای هر کدام از این ابزار نیاز نبود تنها به دلیل آنکه برای اجرای داشبورد کیبانا، نیاز به توکن خاص یا یک سری لاگ این با مواردی که خود الاستیک سرچ فراهم میکرد داشتیم، موارد امنیتی را در فایل `/config/elasticsearch.yml` را غیرفعال کردیم. این کار صرفاً برای آسان‌تر شدن مراحل ورود به داشبورد کیبانا بود اما در کل کار صحیحی نمی‌باشد.

```
90
91 # Enable security features
92 xpack.security.enabled: false
93
94 xpack.security.enrollment.enabled: false
95
96 # Enable encryption for HTTP API client connections, such as Kibana, Logstash, and Agents
97 xpack.security.http.ssl:
98   enabled: false
99   keystore.path: certs/http.p12
100
101 # Enable encryption and mutual authentication between cluster nodes
102 xpack.security.transport.ssl:
103   enabled: false
104   verification_mode: certificate
105   keystore.path: certs/transport.p12
106   truststore.path: certs/transport.p12
107 # Create a new cluster with the current node only
108 # Additional nodes can still join the cluster later
109 cluster.initial_master_nodes: ["aref-VirtualBox"]
110
111 # Allow HTTP API connections from anywhere
112 # Connections are encrypted and require user authentication
113 http.host: 0.0.0.0
114
```

حال باید سرویس‌های موردنظر را اجرا کنیم. برای اجرای الاستیک سرچ، دستور `./bin/elasticsearch` را اجرا می‌کنیم.



```
aref@aref-VirtualBox: ~/ElasticSearch/elasticsearch-8.8.0/bin
aref@aref-VirtualBox:~/ElasticSearch/elasticsearch-8.8.0/bin$ ./elasticsearch
[2023-05-31T20:08:48,916][INFO ][o.e.n.Node               ] [aref-VirtualBox] v
ersion[8.8.0], pid[3154], build[tar/c01029875a091076ed42cdb3a41c10b1a9a5a20f/20
23-05-23T17:16:07.179039820Z], OS[Linux/5.15.0-72-generic/amd64], JVM[Oracle Co
rporation/OpenJDK 64-Bit Server VM/20.0.1/20.0.1+9-29]
[2023-05-31T20:08:48,922][INFO ][o.e.n.Node               ] [aref-VirtualBox] J
VM home [/home/aref/ElasticSearch/elasticsearch-8.8.0/jdk], using bundled JDK [
true]
[2023-05-31T20:08:48,922][INFO ][o.e.n.Node               ] [aref-VirtualBox] J
VM arguments [-Des.networkaddress.cache.ttl=60, -Des.networkaddress.cache.negoti
ative.ttl=10, -Djava.security.manager=allow, -XX:+AlwaysPreTouch, -Xss1m, -Djava.
awt.headless=true, -Dfile.encoding=UTF-8, -Djna.nosys=true, -XX:-OmitStackTrace
InFastThrow, -Dio.netty.noUnsafe=true, -Dio.netty.noKeySetOptimization=true, -D
io.netty.recycler.maxCapacityPerThread=0, -Dlog4j.shutdownHookEnabled=false, -D
log4j2.disable.jmx=true, -Dlog4j2.formatMsgNoLookups=true, -Djava.locale.providers=SPI,COMPAT, --add-opens=java.base/java.io=org.elasticsearch.preallocate, -X
X:+UseG1GC, -Djava.io.tmpdir=/tmp/elasticsearch-11090152490607746230, -XX:+Heap
DumpOnOutOfMemoryError, -XX:+ExitOnOutOfMemoryError, -XX:HeapDumpPath=data, -XX
:ErrorFile=logs/hs_err_pid%p.log, -Xlog:gc*,gc+age=trace,safepoint:file=logs/gc
.log:utctime,level,pid,tags:filecount=32,filesize=64m, -Xms4857m, -Xmx4857m, -X
X:MaxDirectMemorySize=2548039680, -XX:G1HeapRegionSize=4m, -XX:InitiatingHeapOc
cupancyPercent=30, -XX:G1ReservePercent=15, -Des.distribution.type=tar, --modul
```

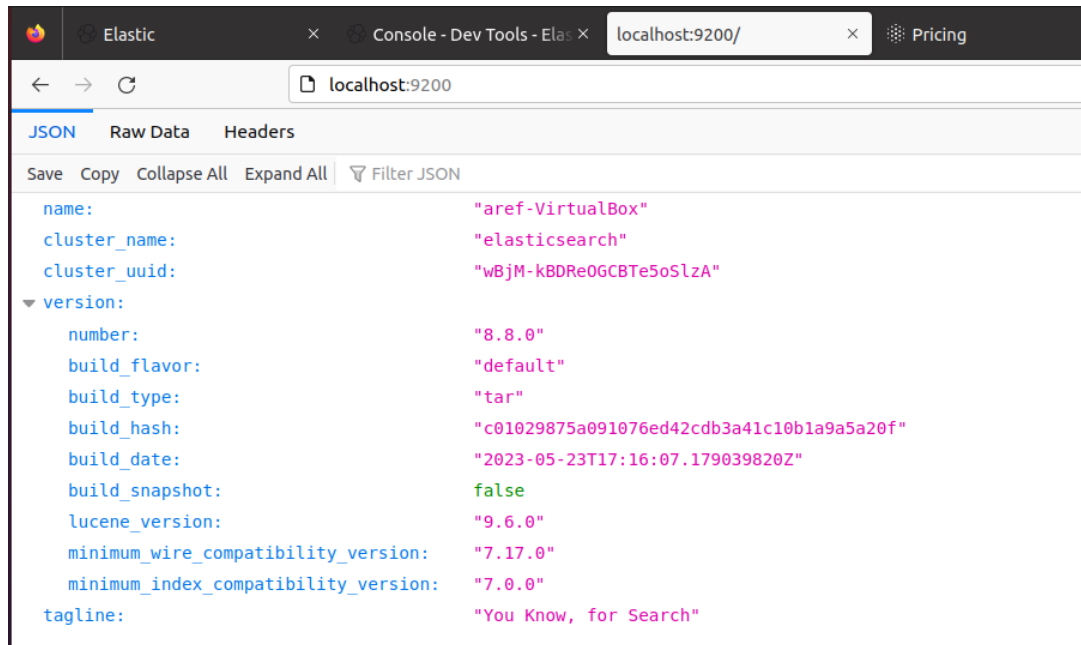
برای اجرای کاساندار، دستور `./bin/cassandra` را اجرا می‌کنیم.

```
aref@aref-VirtualBox: ~/Cassandra/apache-cassandra-4.1.1/bin
aref@aref-VirtualBox:~/Cassandra/apache-cassandra-4.1.1/bin$ ./cassandra
aref@aref-VirtualBox:~/Cassandra/apache-cassandra-4.1.1/bin$ OpenJDK 64-Bit Serv
er VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will
likely be removed in a future release.
CompileCommand: dontinline org/apache/cassandra/db/Columns$Serializer.deserializ
eLargeSubset(Lorg/apache/cassandra/io/util/DataInputPlus;Lorg/apache/cassandra/d
b/Columns;I)Lorg/apache/cassandra/db/Columns;
CompileCommand: dontinline org/apache/cassandra/db/Columns$Serializer.serializeL
argeSubset(Ljava/util/Collection;ILorg/apache/cassandra/db/Columns;ILorg/apache/
cassandra/io/util/DataOutputPlus;V
CompileCommand: dontinline org/apache/cassandra/db/Columns$Serializer.serializeL
argeSubsetSize(Ljava/util/Collection;ILorg/apache/cassandra/db/Columns;I)I
CompileCommand: dontinline org/apache/cassandra/db/commitlog/AbstractCommitLogSe
gmentManager.advanceAllocatingFrom(Lorg/apache/cassandra/db/commitlog/CommitLogS
egment;)V
CompileCommand: dontinline org/apache/cassandra/db/transform/BaseIterator.tryGet
MoreContents()Z
CompileCommand: dontinline org/apache/cassandra/db/transform/StoppingTransformat
ion.stop()V
CompileCommand: dontinline org/apache/cassandra/db/transform/StoppingTransformat
ion.stopInPartition()V
CompileCommand: dontinline org/apache/cassandra/io/util/BufferedDataOutputStream
Plus.doFlush(I)V
```

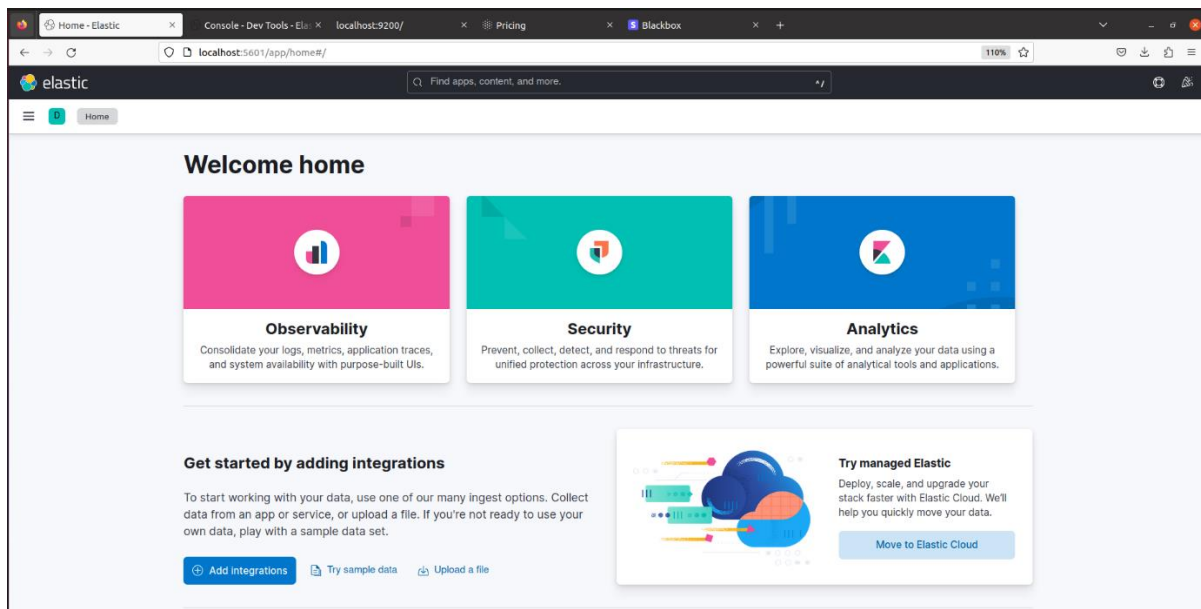
برای اجرای کیبانا، دستور `./bin/kibana` را اجرا می‌کنیم.

```
aref@aref-VirtualBox: ~/kibana/kibana-8.8.0/bin
aref@aref-VirtualBox:~/kibana/kibana-8.8.0/bin$ ./kibana
[2023-05-31T20:09:49.772+03:30][INFO ][node] Kibana process configured with role s: [background tasks, ui]
[2023-05-31T20:10:04.784+03:30][INFO ][plugins-service] Plugin "cloudChat" is di
sabled.
[2023-05-31T20:10:04.786+03:30][INFO ][plugins-service] Plugin "cloudExperiments
" is disabled.
[2023-05-31T20:10:04.786+03:30][INFO ][plugins-service] Plugin "cloudFullStory"
is disabled.
[2023-05-31T20:10:04.787+03:30][INFO ][plugins-service] Plugin "cloudGainsight"
is disabled.
[2023-05-31T20:10:04.806+03:30][INFO ][plugins-service] Plugin "profiling" is di
sabled.
[2023-05-31T20:10:04.864+03:30][INFO ][http.server.Preboot] http server running
at http://localhost:5601
[2023-05-31T20:10:04.963+03:30][INFO ][plugins-system.preboot] Setting up [1] pl
ugins: [interactiveSetup]
[2023-05-31T20:10:04.965+03:30][INFO ][preboot] "interactiveSetup" plugin is hol
ding setup: Validating Elasticsearch connection configuration...
[2023-05-31T20:10:04.985+03:30][INFO ][root] Holding setup until preboot stage i
s completed.
[2023-05-31T20:10:05.329+03:30][WARN ][config.deprecation] The default mechanism
for Reporting privileges will work differently in future versions, which will a
```

پس از اجرا کردن سرویس‌های بالا با ورود به آدرس <https://localhost:9200> از اجرای صحیح الاستیک سرچ اطمینان حاصل می‌کنیم.



همچنین برای ورود به داشبورد کیبانا به آدرس <https://localhost:5601> مراجعه می‌کنیم.



قبل از شروع به پاسخ به سوالات مطرح شده، نیاز است درباره روش برخورد با داده‌ها توضیح داده شود.

## Date -

داده‌های ستون سال، در دو فرمت در دیتاست موجود می‌باشند. نوع اول به صورت تک سالی است. به عنوان مثال فیلد Date در یک کتاب تنها شامل یک عدد چهاررقمی به فرمت "yyyy" می‌باشد. در بعضی ستون‌هایی دیگر، فیلد Date به فرمت "yyyy - yyyy" است که یک بازه را نشان می‌دهد. برای سوال‌های گام چهارم چون کتاب‌های قبل از سال 2000 را می‌خواهد، تنها به اولین سال از بازه داده‌هایی که دارای یک رنج از سال‌ها هستند است. برای این منظور تنها سال شروع را در فایل CSV مجزا ذخیره کرده ایم.

|    | A  | B                                      | C                  | D    |
|----|--|--|--------------------|------|
| 1  | title  | url                                    | contributors       | date |
| 2  | Goliath Joe, fisherman / by Charles Thackeray (Wobbe       | https://trove.nla.gov.au/work/10013347 | Thackeray, Charl   | 1900 |
| 3  | Grammar of the Narrinyeri tribe of Australian Aborigines   | https://trove.nla.gov.au/work/10013347 | Taplin, George     | 1878 |
| 4  | The works of the Rev. Sydney Smith                         | https://trove.nla.gov.au/work/10013347 | Smith, Sydney, 1   | 1839 |
| 5  | Nellie Doran : a story of Australian home and school life  | https://trove.nla.gov.au/work/10013347 | Miriam Agatha      | 1914 |
| 6  | Lastkraftwagen 3 t Ford : Baumuster V 3000 S : Geratb      | https://trove.nla.gov.au/work/10013347 | Germany. Heer. I   | 1942 |
| 7  | Trefoil : the story of a girls' society / by M.P. Macdonak | https://trove.nla.gov.au/work/10013347 | Macdonald, M. P    | 1900 |
| 8  | Military report on the province of Chiang-su (south of th  | https://trove.nla.gov.au/work/10068876 |                    | 1909 |
| 9  | Le Siege de Berlin : Drame en un Acte / Charles Hellem     | https://trove.nla.gov.au/work/10069391 |                    | 1915 |
| 10 | The Aborigines of Central Australia : with a vocabulary    | https://trove.nla.gov.au/work/10013347 | Willshire, W. H. ( | 1888 |
| 11 | Astronomia : Deutsch astronomei                            | https://trove.nla.gov.au/work/10013347 | Jacob, Cyriacus,   | 1545 |

اما برای سوال دوم گام اول، به داده‌های هر سال نیاز بود. به عنوان مثال اگر بازه یک کتاب سال 2000 الی 2002 باشد، این کتاب در سال 2001 نیز بوده است. پس یک دیتاست و فایل CSV مجزا به صورتی ایجاد کرده‌ایم که برای فیلدهایی که سال آن‌ها یک رنج است، یک ستون کپی‌شده از آن ردیف، هرکدام به همراه یکی از سال‌های داخل آن رنج ایجاد کرده ایم.

```
1 title,url,contributors,date,format,fulltext_url,trove_id,language,rights,pages,form,volume,parent,children,text downloaded,
2 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1900,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
3 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1901,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
4 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1902,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
5 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1903,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
6 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1904,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
7 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1905,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
8 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1906,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
9 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1907,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
10 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1908,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
11 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1909,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
12 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1910,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
13 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1911,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
14 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1912,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
15 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1913,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
16 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1914,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
17 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1915,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
18 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1916,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
19 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1917,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
20 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1918,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
21 "Goliath Joe, fisherman / by Charles Thackeray (Wobbe",https://trove.nla.gov.au/work/10013347,"Thackeray, Charles",1919,
nla.obj-2831231419,nla.obj-2831231419,English,Out of Copyright,http://rightsstatements.org/vocab/NKC/1.0/,130,Book,,,,,True,g
```

## Format -

داده‌های ستون فرمت، به صورت، چندین فرمت مجزا که با " | " از هم جدا شده‌اند.  
برای سوال اول گام دوم، نیاز به جداسازی این فرمت‌ها بود تا تعداد کتاب‌های هر فرمت به صورت مجزا محاسبه شود. پس یک دیتاست دیگر در فرمت CSV به صورتی ایجاد کرده‌ایم که برای هر داده، یک ستون کپی‌شده از آن ردیف به همراه یکی از فرمت‌های موجود در فیلد فرمت هایش ایجاد کرده ایم.

داده‌های ستون فرمت در فایل ابتدایی به صورت زیر بود.

|   | A                                      | B            | C           | D         | E                                       |   |
|---|--|--------------|-------------|-----------|---|---|
| 1 | title                                  | url          | contributor | date      | format                                  | f |
| 2 | Goliath Joe, fisherman / by Charles T  | https://trov | Thackeray,  | 1900-1919 | Book Book Illustrated                   | t |
| 3 | Grammar of the Narrinyeri tribe of A   | https://trov | Taplin, Ge  | 1878-1880 | Book Government publication             | t |
| 4 | The works of the Rev. Sydney Smith     | https://trov | Smith, Syd  | 1839-1900 | Book Book Illustrated Microform         | t |
| 5 | Nellie Doran : a story of Australian h | https://trov | Miriam Ag   | 1914-1923 | Book                                    | t |
| 6 | Lastkraftwagen 3 t Ford : Baumuster    | https://trov | Germany. I  | 1942      | Book Book Illustrated Government public | t |

پس از تبدیل به صورت زیر درآمد:

|    | A            | B            | C           | D    | E                      |   |
|----|--------------|--------------|-------------|------|------------------------|---|
| 1  | title        | url          | contributor | date | format                 | f |
| 2  | Goliath Joe  | https://trov | Thackeray,  | 1900 | Book                   | t |
| 3  | Goliath Joe  | https://trov | Thackeray,  | 1900 | Book Illustrated       | t |
| 4  | Grammar of   | https://trov | Taplin, Ge  | 1878 | Book                   | t |
| 5  | Grammar of   | https://trov | Taplin, Ge  | 1878 | Government publication | t |
| 6  | The works    | https://trov | Smith, Syd  | 1839 | Book                   | t |
| 7  | The works    | https://trov | Smith, Syd  | 1839 | Book Illustrated       | t |
| 8  | The works    | https://trov | Smith, Syd  | 1839 | Microform              | t |
| 9  | Nellie Doran | https://trov | Miriam Ag   | 1914 | Book                   | t |
| 10 | Lastkraftw   | https://trov | Germany. I  | 1942 | Book                   | t |
| 11 | Lastkraftw   | https://trov | Germany. I  | 1942 | Book Illustrated       | t |
| 12 | Lastkraftw   | https://trov | Germany. I  | 1942 | Government publication | t |

کدهای اجرا شده برای اجرای عملیات‌های بالا به صورت زیر است.



```

import pandas as pd
import csv

# task 1 -> separate date range into several rows
# Open the input and output CSV files
with open('books_info.csv', 'r') as input_file, open('books_info_updated.csv', 'w', newline='') as output_file:
    reader = csv.DictReader(input_file)
    writer = csv.DictWriter(output_file, fieldnames=reader.fieldnames)
    writer.writeheader()

    # Iterate over each row in the input CSV file
    for row in reader:
        # Check if the "date" column contains a range of years separated by "-"
        if '-' in row['date']:
            # Split the range of years into two separate years
            start_year, end_year = row['date'].split('-')
            # Iterate over each year in the range and create a new row for each year
            for year in range(int(start_year), int(end_year)+1):
                # Create a new row with the same data as the original row, except for the "date" column
                new_row = row.copy()
                new_row['date'] = str(year)
                # Write the new row to the output CSV file
                writer.writerow(new_row)
            # row['date'] = str(start_year)
            # writer.writerow(row)
        else:
            # If the "date" column does not contain a range of years, write the original row to the output CSV file
            writer.writerow(row)

# Read the input CSV file into a pandas DataFrame
df = pd.read_csv('books_info_updated.csv')

# task 2 -> drop nulls in date
# Drop all rows with Null values in date col
df = df.dropna(subset=['date'])

# task 3 -> reformat date to "yyyy" instead of "yyyyy.0"
# Remove the last two characters from the 'date' column
df['date'] = df['date'].astype(str).str[:-2]

# task 3 -> remove wrong date
list= ['18','19','180','181','182','183','184','185','186','187','188','189',
      '190','191','192','193','194','195','196','197','198','199']

# Create a boolean mask for rows where the date column does not match the list
mask = ~df['date'].isin(list)

# Get the non-matching rows
new_df = df[mask]

# Create a new CSV file with the non-matching rows
new_df.to_csv('final.csv', index=False)

print(f"{len(df) - len(new_df)} matching rows were removed and {len(new_df)} rows were written to 'final.csv' file.")

```

```

import csv

with open('final_2.csv', 'r') as input_file, open('final_2_formatted.csv', 'w', newline='') as output_file:
    reader = csv.DictReader(input_file)
    writer = csv.DictWriter(output_file, fieldnames=reader.fieldnames)
    writer.writeheader()

    # Iterate over each row in the input CSV file
    for row in reader:
        # Check if the "format" column contains a range of years separated by "-"
        if '|' in row['format']:
            # Split the range of years into two separate years
            formats = row['format'].split('|')

            for format in formats:
                # Create a new row with the same format as the original row, except for the "format" column
                new_row = row.copy()
                new_row['format'] = str(format)
                # Write the new row to the output CSV file
                writer.writerow(new_row)
        else:
            # If the "format" column does not contain a range of years, write the original row to the output CSV file
            writer.writerow(row)

```

حال نیاز است داده‌ها به الاستیک سرچ بی‌افزاییم.

همانطور که در بالا توضیح داده شد 3 نوع دیتاست داریم و برای همین منظور سه نوع ایندکس ایجاد کرده ایم.

`books_index` در واقع داده‌های شامل همه رنج سال‌های موجود را شامل می‌شود

`books_index_2` در واقع داده‌های صرفاً شامل سال شروع هر کتاب را شامل می‌شود.

`books_index_format_date` داده‌های شامل فرمت جداشده ستون فرمت را شامل می‌شود.

با استفاده از کد زیر داده‌های مختلف از فایل CSV خوانده شده به ایندکس موردنظر فرستاده می‌شود.

```
import csv
from elasticsearch import Elasticsearch

# create a connection to Elasticsearch
es = Elasticsearch(
    hosts=['http://localhost:9200']
)

# create an index for the books
index_name = 'books_index'
if not es.indices.exists(index=index_name):
    es.indices.create(index=index_name)

# create a mapping for the index
mapping = {
    "properties": {
        "title": {"type": "text"},
        "url": {"type": "keyword"},
        "contributors": {"type": "text"},
        "date": {
            "type": "date",
            "format": "yyyy"
        },
        "format": {"type": "keyword"},
        "full_text_url": {"type": "keyword"},
        "trove_id": {"type": "keyword"},
        "language": {"type": "keyword"},
        "rights": {"type": "keyword"},
        "pages": {"type": "integer"},
        "form": {"type": "keyword"},
        "volume": {"type": "keyword"},
        "parent": {"type": "keyword"},
        "children": {"type": "keyword"},
        "text_downloaded": {"type": "keyword"},
        "text_file": {"type": "keyword"}
    }
}

es.indices.put_mapping(index=index_name, body=mapping)

# read the data from the csv file and index it
with open('final.csv') as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
        es.index(index=index_name, body=row)
```



## 1- بررسی کنید چند کتاب کودک در این مجموعه جمع آوری شده است و سپس میانگین تعداد صفحات کتاب‌های کودکان را بدست آورید.

برای بررسی کتاب‌های کودکان، کتاب‌هایی را از الاستیک سرچ دریافت کرده ایم که فیلد `parent` در آن‌ها مقدار نال نداشته باشد. در واقع اینگونه به کتاب کودکان نگاه شد که آنهایی که والد پدر آن‌ها مقدار دارد یعنی دارای پدر هستند. نمی‌دانم این دید چقدر درست باشد ولی نحوه بررسی من به این گونه بوده است. در ادامه نیست با استفاده از یک `aggregation`، متوسط تعداد صفحات گرفته شده و در خروجی نمایش داده شده است. خروجی در ادامه سوال دوم نشان داده شده است. برای این سوال، از ایندکس `books_index_2` که توضیح داده شد استفاده شده است.

```
from elasticsearch import Elasticsearch

es = Elasticsearch(
    hosts=['http://localhost:9200']
)
index_name = 'books_index_2'

# Get the number of children's books and their average page count
query = {
    "query": {
        "bool": {
            "must_not": {
                "term": {"parent": ""}
            }
        }
    },
    "aggs": {
        "avg_page_count": {"avg": {"field": "pages"}}
    }
}
result = es.search(index=index_name, body=query)

num_children_books = result["hits"]["total"]["value"]
avg_page_count = result["aggregations"]["avg_page_count"]["value"]

print(f"Number of children's books: {num_children_books}")
print(f"Average page count of children's books: {avg_page_count:.2f}")
```

## 2- در چه سالی بیشترین تعداد کتاب چاپ شده است؟

برای بررسی بیشترین سال کتاب چاپ شده، اینبار از ایندکس `books_index` که توضیح داده شد استفاده شده است.

```
query = {
    "size": 0,
    "aggs": {
        "publication_year": {
            "terms": {
                "field": "date",
                "format": "yyyy"
            },
            "aggs": {
                "doc_count": {
                    "value_count": {
                        "field": "trove_id"
                    }
                }
            }
        }
    }
}

res = es.search(index='books_index', body=query)
largest_num_books_year = res['aggregations']['publication_year']['buckets'][0]['key_as_string']

print(f"The year with the largest number of books published is {largest_num_books_year}.")
```

خروجی نهایی گام اول در زیر قابل مشاهده است. بیشترین کتاب در سال 2018 چاپ شده است. تعداد 1422 کتاب کودک داریم که 298.66 میانگین تعداد صفحات این کتابها است.

```
/bin/python3 /home/aref/BigData/Tarmin9/Step1.py
• aref@aref-VirtualBox:~/BigData/Tarmin9$ /bin/python3 /home/aref/
/usr/lib/python3/dist-packages/requests/__init__.py:89: Request
ion!
    warnings.warn("urllib3 ({}), or chardet ({}), doesn't match a s
/home/aref/BigData/Tarmin9/Step1.py:21: DeprecationWarning: The
dividual parameters.
    result = es.search(index=index_name, body=query)
Number of children's books: 1422
Average page count of children's books: 298.66
/home/aref/BigData/Tarmin9/Step1.py:47: DeprecationWarning: The
dividual parameters.
    res = es.search(index='books_index', body=query)
The year with the largest number of books published is 2018.
• aref@aref-VirtualBox:~/BigData/Tarmin9$
```

## گام دوم:

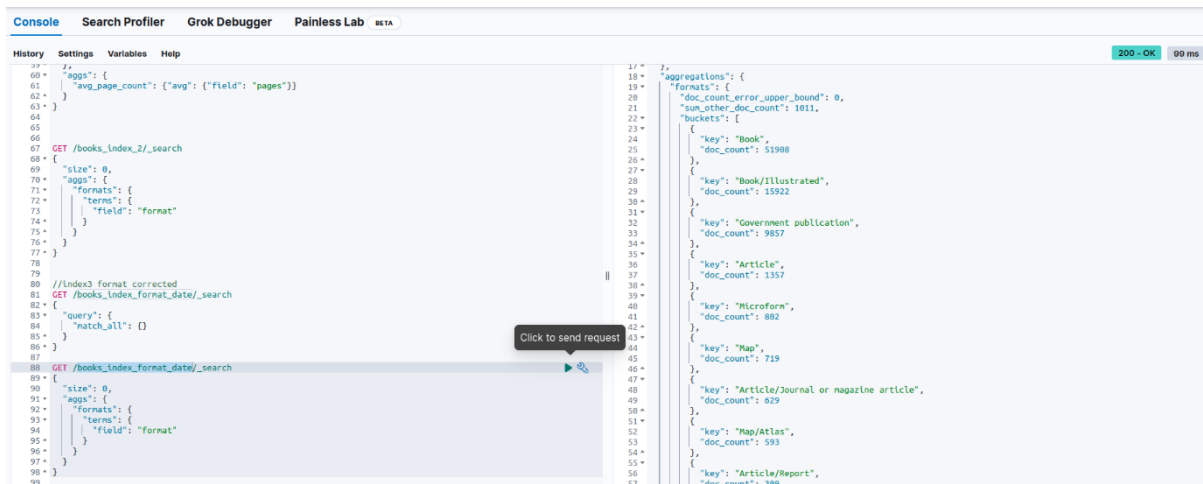
الاستیک سرچ و کیبانا با یکدیگر نصب و اجرا میشوند. بنابراین داشبوردی در کیبانا طراحی کنید که موارد خواسته شده در زیر را به شما نمایش دهد:

1- از هر فرمت چند کتاب در این مجموعه داده جمعآوری شده است؟

برای این سوال از داشبورد کیبانا استفاده شده است. برای این سوال، از ایندکس `books_index_format_date` که توضیح داده شد استفاده شده است.

```
87
88 GET /books_index_format_date/_search
89 {
90   "size": 0,
91   "aggs": {
92     "formats": {
93       "terms": {
94         "field": "format"
95       }
96     }
97   }
98 }
99
```

خروجی نهایی پرسوجو فوق به صورت زیر است.



```

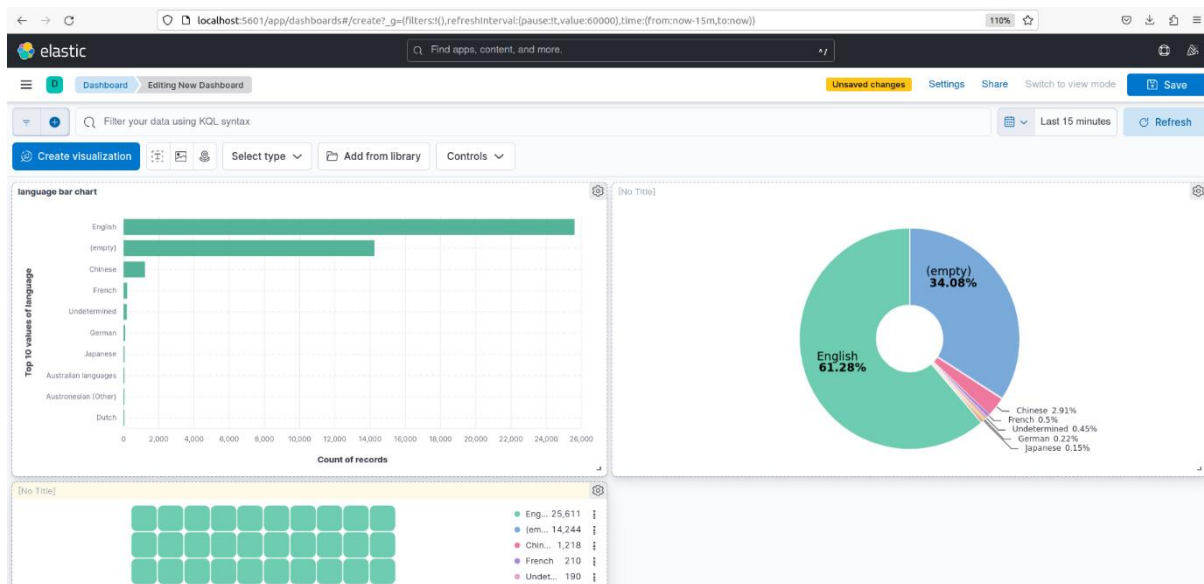
"aggregations": {
  "formats": {
    "doc_count_error_upper_bound": 0,
    "sum_other_doc_count": 1011,
    "buckets": [
      {
        "key": "Book",
        "doc_count": 51908
      },
      {
        "key": "Book/Illustrated",
        "doc_count": 15922
      },
      {
        "key": "Government publication",
        "doc_count": 9857
      },
      {
        "key": "Article",
        "doc_count": 1357
      },
      {
        "key": "Microform",
        "doc_count": 802
      },
      {
        "key": "Map",
        "doc_count": 719
      },
      {
        "key": "Article/Journal or magazine article",
        "doc_count": 629
      },
      {
        "key": "Map/Atlas",
        "doc_count": 593
      },
      {
        "key": "Article/Report",
        "doc_count": 309
      },
      {
        "key": "Published",
        "doc_count": 200
      }
    ]
  }
}

```

## 2- با کمک نمودارها تعداد کتابهای موجود به هر زبان را نشان دهید.

برای این سوال از داشبورد کیبانا استفاده کرده ایم. باید یک `data_view` با ایندکس خود ساخته و سپس با درگ و دراپ کردن فیلد زبان نمودارهای مختلف را نمایش دهیم.

در ادامه نمونه‌هایی از این نمودارها در داشبورد کیبانا نمایش داده شده است.



## گام سوم:

پس از بررسی داده‌های ذخیره شده در الاستیک سرچ، داده‌ها را بر اساس موارد خواسته شده که در ادامه درج شده، در کاساندرا ذخیره کنید. لازم است توجه داشته باشید که تکرار داده‌ها یک اصل کامل پذیرفته شده بوده و بنابراین به دنبال نرمال سازی داده‌ها نباشید.

در این قسمت دو جدول در کاساندرا ایجاد کنید که بر اساس گروه سنی کتابها این تقسیم بندی انجام شود، و در هر دو جدول از فیلد `trove_id` به عنوان کلید اصلی جداول استفاده کنید.

- سپس یک جدول برای ذخیره سازی اطلاعات کتابهای کودک با استفاده از فیلدهای زیر ایجاد کنید:  
نام کتاب، نام نویسندگان، و آدرس اینترنتی در دسترس برای محتوای الکترونیک و نوع فایل در دسترس کتاب.
- جدول دومی برای ذخیره سازی اطلاعات کتابهای `parent` با استفاده از فیلدهای زیر ایجاد کنید:  
نام کتاب، نام نویسندگان، آدرس اینترنتی در دسترس برای محتوای الکترونیک و نوع فایل در دسترس کتاب.

در زیر کد این قیمت از تمرین نشان داده شده است. در ابتدا با اتصال به الاستیک سرچ، با استفاده از دو پرس - وجو داده هارا دریافت می کنیم. بار اول اجرا و پس از یافتن تعداد دقیق داده‌ها، آنها رو تحت عنوان فیلد "size" به پرس وجو خود می دهیم. الاستیک سرچ به صورت دیفالت، تنها 10 رکورد ابتدایی را برمیگرداند. با این روش و دادن سایز موردنظر همه داده‌ها را به صورت یکجا دریافت می کنیم.

```

from elasticsearch import Elasticsearch
import pandas as pd
from cassandra.cluster import Cluster
import json

es = Elasticsearch(
    hosts=['http://localhost:9200']
)
index_name = 'books_index_2'

# Get the number of children's books and their average page count
query_children = {
    "query": {
        "bool": {
            "must_not": {
                "term": {"parent": ""}
            }
        }
    },
    "size": 1422
}

books_info_children = es.search(index=index_name, body=query_children)
print(f"Number of matching documents: {books_info_children['hits']['total']['value']}")

query_adults = {
    "query": {
        "bool": {
            "must_not": {
                "term": {"children": ""}
            }
        }
    },
    "size": 239
}

books_info_adults = es.search(index=index_name, body=query_adults)
print(f"Number of matching documents: {books_info_adults['hits']['total']['value']}")

# Connect to the Cassandra cluster
cluster = Cluster(['127.0.0.1'])
session = cluster.connect()

# Specify a keyspace
session.execute("CREATE KEYSPACE IF NOT EXISTS elasticSearchKS WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '1'}")

session.execute("DROP TABLE IF EXISTS elasticSearchKS.childrenBooks")
session.execute("DROP TABLE IF EXISTS elasticSearchKS.parentBooks")

# Create table for children's books
session.execute("""
CREATE TABLE IF NOT EXISTS elasticSearchKS.childrenBooks (
    id_trove text PRIMARY KEY,
    title text,
    contributors text,
    full_text_url text,
    format text,
    pages int,
    publication_date int
)
""")

# Create table for parent books
session.execute("""
CREATE TABLE IF NOT EXISTS elasticSearchKS.parentBooks (
    id_trove text PRIMARY KEY,
    title text,
    contributors text,
    full_text_url text,
    format text,
    pages int,
    publication_date int
)
""")

# Insert data into appropriate table based on the age group of the book
index = 0
for source in books_info_children['hits']['hits']:
    source_dict = source['_source']
    index += 1
    session.execute("""
INSERT INTO elasticSearchKS.childrenBooks (id_trove, title, contributors, full_text_url, format, pages, publication_date)
VALUES (%s, %s, %s, %s, %s, %s, %s) IF NOT EXISTS""",
        (source_dict["trove_id"], source_dict["title"],
         source_dict["contributors"], source_dict["fulltext_url"],
         source_dict["format"], int(source_dict["pages"]), int(source_dict["date"])))

print(index)

index = 0
for source in books_info_adults['hits']['hits']:
    source_dict = source['_source']
    index += 1
    session.execute("""
INSERT INTO elasticSearchKS.parentBooks (id_trove, title, contributors, full_text_url, format, pages, publication_date)
VALUES (%s, %s, %s, %s, %s, %s, %s) IF NOT EXISTS""",
        (source_dict["trove_id"], source_dict["title"],
         source_dict["contributors"], source_dict["fulltext_url"],
         source_dict["format"], int(source_dict["pages"]), int(source_dict["date"])))

print(index)

```

در ادامه توضیح بالا، به کلاستر کاساندرا که روی لوکال هاست در حال اجرا است متصل شده و ابتدا یک keyspace جدید و سپس دو جدول یکی برای کتاب‌های کودکان و یکی برای کتاب‌های بزرگسال به همراه فیلدهایی که مورد نیاز است ایجاد می‌کنیم. مدنظر داشته باشید که فیلدهای سال چاپ و تعداد صفحات نیز در این جداول قرار دارد.

سپس مجموعه داده‌های فچ شده از اسلاستک سرچ را در دو حلقه فور مجزا به جداول موردنظرشان می‌افزاییم.

خروجی نهایی اجرای این فایل به صورت زیر است:

```
aref@aref-VirtualBox:~/BigData/Tarmin9$ /bin/python3 /home/aref/BigData/Tarmin9/Step3_Store.py
/usr/lib/python3/dist-packages/requests/_internal.py:89: RequestsDependencyWarning: urllib3 (1.26.15) or chardet (3.0.4) doesn't match a supported version!
  warnings.warn("urllib3 ({}), or chardet ({}), doesn't match a supported version".format(urllib3.__version__, chardet.__version__))
/home/aref/BigData/Tarmin9/Step3_Store.py:23: DeprecationWarning: The 'body' parameter is deprecated and will be removed in a future version. Instead use individual parameters.
  books_info_children = es.search(index=index_name, body=query_children)
Number of matching documents: 1422
/home/aref/BigData/Tarmin9/Step3_Store.py:37: DeprecationWarning: The 'body' parameter is deprecated and will be removed in a future version. Instead use individual parameters.
  books_info_adults = es.search(index=index_name, body=query_adults)
Number of matching documents: 239
1422
239
aref@aref-VirtualBox:~/BigData/Tarmin9$
```

برای اطمینان از اینکه داده‌ها به درستی به کاساندرا افزوده شده‌اند، در یک فایل مجزا از هر جدول ایجاد کرده، همه داده‌ها به همراه ساینش را دریافت می‌کنیم. اینطور که پیداست، 4 داده از جدول children ها دارای id\_\_trove تکراری بوده‌اند که باعث شده است از 1422 داده موجود در الاستیک تنها 1418 تای آنها به کاساندرا افزوده شود ولی تعداد داده‌های parent کاملاً درست بوده است!

```
from cassandra.cluster import Cluster

# Connect to the Cassandra cluster
cluster = Cluster(['127.0.0.1'])
session = cluster.connect()

# Define a SELECT query to retrieve data from your Cassandra table
query = "SELECT * FROM elasticSearchKS.childrenBooks"

# Execute the query and retrieve the query results
rows = session.execute(query)

# Loop over the rows in the query results and print each row
index = 0
for row in rows:
    index += 1
    # print(row)
print(f"Number of stored children book=> {index}")

# Define a SELECT query to retrieve data from your Cassandra table
query = "SELECT * FROM elasticSearchKS.parentBooks"

# Execute the query and retrieve the query results
rows = session.execute(query)

# Loop over the rows in the query results and print each row
index = 0
for row in rows:
    index += 1
    # print(row)
print(f"Number of stored parent book=> {index}")
```

```
aref@aref-VirtualBox:~/BigData/Tarmin9$ /bin/python3 /home/aref/BigData/Tarmin9/Step3_retrieve.py
Number of stored children book=> 1418
Number of stored parent book=> 237
aref@aref-VirtualBox:~/BigData/Tarmin9$
```



## گام چهارم:

به کوئری‌های زیر بر اساس جداول ذخیره شده در گام سوم پاسخ دهید.

1- میانگین تعداد صفحات کتابهای موجود در هر دو جدول را محاسبه کنید.

2- کتاب‌هایی را نشان دهید که در سالهای قبل از 2000 چاپ شده اند را نمایش دهد.

در گام چهارم و پس از افزودن داده‌ها به کاسندرا و اطمینان از درست افزوده‌شدن آن‌ها، روی آن‌ها پرس‌وجوهای اجرا می‌کنیم.

سوال اول این گام حذف شده است ولی یک راه حل برای آن در اینجا بیان می‌شود ولیکن از صحت آن اطمینان ندارم.

در این روش، تمام داده‌های هر دو جدول از کاسندرا فچ شده است و سپس کتاب‌هایی که هم در دسته کتاب کودک و هم بزرگسال بوده اند استخراج شده اند و در نهایت روی فیلد کتاب آن‌ها متوسط گرفته شده است. خروجی نهایی نشان می‌دهد میانگین صفحات کتاب‌هایی که در هر دو جدول موجود است، 255.6 است.

برای سوال دوم نیز یک پرس‌وجو ساده با فیلتر کردن سال چاپ به قبل از 2000 روی هر دو جدول ایجاد شده و نام رکوردهای منطبق به همراه سال چاپشان در خروجی نمایش داده شده است.

```
from cassandra.cluster import Cluster

# Connect to the Cassandra cluster
cluster = Cluster(['127.0.0.1'])
session = cluster.connect()

# Retrieve all data from childrenBooks table
rows_children = session.execute('SELECT * FROM elasticSearchKS.childrenBooks')
data_children = [(row.title, row.pages) for row in rows_children]

# Retrieve all data from parentBooks table
rows_parent = session.execute('SELECT * FROM elasticSearchKS.parentBooks')
data_parent = [(row.title, row.pages) for row in rows_parent]

# Find books that exist in both tables
common_books = set([book[0] for book in data_children]).intersection(set([book[0] for book in data_parent]))
books_to_avg = [book for book in data_children + data_parent if book[0] in common_books]

# Calculate the average number of pages
avg_pages = sum([int(book[1]) for book in books_to_avg]) / len(books_to_avg)

print(f"The average number of pages for books in both tables is {avg_pages}.")

# Get all children's books published before 2000
children_query = "SELECT * FROM elasticSearchKS.childrenBooks WHERE publication_date < 2000 ALLOW FILTERING"
children_results = session.execute(children_query)

print("Children's books published before 2000:-----")
for index, result in enumerate(children_results):
    print(f"{index} -> {result.publication_date} : {result.title}")

# Get all parent books published before 2000
parent_query = "SELECT * FROM elasticSearchKS.parentBooks WHERE publication_date < 2000 ALLOW FILTERING"
parent_results = session.execute(parent_query)

print("Parent books published before 2000:-----")
for index, result in enumerate(parent_results):
    print(f"{index} -> {result.publication_date} : {result.title}")
```

در این خروجی تنها 10 رکورد اول نمایش داده شده است.

```
250 -> 1973 : Private violence and public policy : the needs of battered women and the response of the public services / edited by Jan Pahl
● aref@aref-VirtualBox:~/BigData/Tarmin9$ /bin/python3 /home/aref/BigData/Tarmin9/Step4.py
The average number of pages for books in both tables is 255.61631419939576.
Children's books published before 2000:-----
0 -> 1716 : Kangxi zi dian
1 -> 1716 : Kangxi zi dian
2 -> 1800 : Zhong xiao jie yi er du mei zhuan
3 -> 1716 : Kangxi zi dian
4 -> 1716 : Kangxi zi dian
5 -> 1716 : Kangxi zi dian
6 -> 1716 : Kangxi zi dian
7 -> 1716 : Kangxi zi dian
8 -> 1800 : Bankoku zusetsu fudo bussanshi / Seiyōjin senjutsu yakuhon
9 -> 1800 : Shi ming shu zheng / Liu Xi zhuan ; Bi Yuan shu zheng
Parent books published before 2000:-----
0 -> 1935 : Lists of men-of-war, 1650-1700
1 -> 1889 : [Bessie Campbell, Banjo Queen : a collection of music programmes, ephemera relating to her career]
2 -> 1860 : Report of the proceedings of the Queensland Government schooner "Spitfire" in search of the mouth of the River Burdekin, on the north-eastern coast of Australia : and of the exploration of a portion of that coast extending from Gloucester Island to Halifax Bay
3 -> 1968 : Westminster Abbey : a service of memorial and thanksgiving for the Right Honourable Harold Edward Holt, C.H., M.P., Prime Minister of the Commonwealth of Australia : born 5 August 1908, died 17 December 1967 ; Thursday 18th January, 1968 at 12 noon
4 -> 1901 : Brief sketch of the life of Theophilus Gum, the prophet, author and teacher of the new dispensation : with a criticism of his extraordinary writings / by the author of Popular preachers of our day
5 -> 1880 : Yokohama tenpozu
6 -> 1858 : Be men : a sermon preached in the St. Kilda Congregational Church, on Sunday morning, 11th April, 1858 / by Thomas Binney
7 -> 1838 : Remarks upon the latest official documents relating to New Zealand (ordered by the House of Commons, to be printed, February 7, 1838) with a notice of a pamphlet by Samuel Hinds ... in a letter to a friend / by John Beecham
8 -> 1900 : Catalogue of a series of photographs illustrating the scenery and peoples of the islands in the south and western Pacific / photographed and published by J.W. Beattie
9 -> 1903 : A white Australia impossible : cause of colour in mankind
● aref@aref-VirtualBox:~/BigData/Tarmin9$
```

در این خروجی آخرین رکوردهای قبل از سال 2000 از جدول کتاب کودکان به همراه اولین رکوردهای قبل از سال 2000 از کتاب بزرگسالان آورده شده است.

```
1400 -> 1800 : [Collection of calligraphical copybook]
1401 -> 1885 : Rivals for supremacy in the Pacific : a book for every British subject / by General MacIver
1402 -> 1891 : A chapter from the history of early discovery and colonisation of Australia / by A.C. Macdonald
1403 -> 1935 : [National Anti-Sweating League of Victoria : ephemera material in the Riley Collection]
1404 -> 1903 : A white Australia impossible : cause of colour in mankind
1405 -> 1716 : Kangxi zi dian
1406 -> 1716 : Kangxi zi dian
1407 -> 1971 : Izesuq oko moloko imolaqu
1408 -> 1716 : Kangxi zi dian
1409 -> 1716 : Kangxi zi dian
1410 -> 1889 : [Bessie Campbell, Banjo Queen : a collection of music programmes, ephemera relating to her career]
1411 -> 1716 : Kangxi zi dian
1412 -> 1754 : The story of the stone : a Chinese novel in five volumes / translated by David Hawkes
1413 -> 1716 : Kangxi zi dian
1414 -> 1929 : [Australian grocery shop advertisements of the Depression era]
1415 -> 1971 : Izesuq oko moloko imolaqu
1416 -> 1856 : Sōmoku zusetsu. Inuma Yokusai cho
1417 -> 1716 : Kangxi zi dian
Parent books published before 2000:-----
0 -> 1935 : Lists of men-of-war, 1650-1700
1 -> 1889 : [Bessie Campbell, Banjo Queen : a collection of music programmes, ephemera relating to her career]
2 -> 1860 : Report of the proceedings of the Queensland Government schooner "Spitfire" in search of the mouth of the River Burdekin, on the north-eastern coast of Australia : and of the exploration of a portion of that coast extending from Gloucester Island to Halifax Bay
3 -> 1968 : Westminster Abbey : a service of memorial and thanksgiving for the Right Honourable Harold Edward Holt, C.H., M.P., Prime Minister of the Commonwealth of Australia : born 5 August 1908, died 17 December 1967 ; Thursday 18th January, 1968 at 12 noon
4 -> 1901 : Brief sketch of the life of Theophilus Gum, the prophet, author and teacher of the new dispensation : with a criticism of his extraordinary writings / by the author of Popular preachers of our day
```

در این خروجی آخرین رکوردهای قبل از سال 2000 از جدول کتاب بزرگسالان آورده شده است.

```
223 -> 1716 : Kangxi zi dian / [Zhong xiao jie yi er du mei zhuan]
224 -> 1869 : Hanum sōnsaeng mun'go purok / [Yi Tōk-hyōng chō ; Yi Ki-yang p'yōn]
225 -> 1935 : [National Anti-Sweating League of Victoria : ephemera material in the Riley Collection]
226 -> 1907 : The cyclopedia of South Australia in two volumes : an historical and commercial review, descriptive and biographical, facts, figures and statistics : an epitome of progress / edited by H.T. Burgess
227 -> 1863 : Jin xiang xiao pin
228 -> 1878 : Works on New South Wales : compiled at the Free Public Library, Sydney / under the direction of R.C. Walker
229 -> 1897 : Notes on the Aborigines of Tasmania / extracted from the manuscript journals of George Washington Walker ; with an introductory notice by R.C. Walker
230 -> 1871 : William Caxton : a contribution in commemoration of the festival held in Melbourne, 1871, to celebrate the fourth centenary of his birth in the English language / compiled by John Ferres
231 -> 1910 : Grammar, vocabulary, and notes of the Wanggerriburra Tribe
232 -> 1858 : Sei ryakuron / Gasshin-shi cho ; Kan Mozai dōsen
233 -> 1971 : Izesuq oko moloko imolaqu
234 -> 1941 : Henderson's Australian families : a genealogical and biographical record / compiled and edited by Alexander Henderson
235 -> 1918 : The rabbit pest in Australia : [collection] / by W. Rodier
236 -> 1873 : Private violence and public policy : the needs of battered women and the response of the public services / edited by Jan Pahl
● aref@aref-VirtualBox:~/BigData/Tarmin9$
```

<https://chatbot.theb.ai/>

<https://chat.forefront.ai/>

<https://www.youtube.com/watch?v=XqqC5oigleY>

<https://www.youtube.com/watch?v=XqqC5oigleY>

<https://stackoverflow.com/questions/522563/accessing-the-index-in-for-loops>

<https://logz.io/blog/kibana-tutorial-2/#kibanaaggregations>

<https://phoenixnap.com/kb/kibana-tutorial>

<https://stackoverflow.com/questions/21762599/primary-key-in-cassandra-is-unique>

<https://stackoverflow.com/questions/6077675/why-am-i-seeing-typeerror-string-indices-must-be-integers>

<https://artifacts.elastic.co/downloads/kibana/>

<https://www.elastic.co/downloads/kibana>

<https://www.elastic.co/downloads/elasticsearch>

<https://phoenixnap.com/kb/install-kubernetes-on-ubuntu>

<https://stackoverflow.com/questions/65631801/illegal-instructioncore-dumped-error-on-jetson-nano>