TalkWithTed

Mehmood Zakria - Tecnologie Cloud & Mobile 2024

Homework 2

Job Pyspark WatchNext

- Operazioni:
  - Lettura da file: «related_videos»
  - Cancellazione duplicati
  - Raggruppamento per id di una collect list di una struct
  - Join con il dataset principale

```
## WATCH NEXT
watch_next_path = "s3://tedx-data-mz/related_videos.csv"
watch_next = spark.read \
    .option("header","true") \
    .option("quote", "\"") \
    .option("escape", "\"") \
    .csv(watch_next_path)

##print(f"Number Watch_Next items (RAW): {watch_next_dataset.count()}")

watch_next = watch_next.dropDuplicates()

# ADD WATCH_NEXT TO AGGREGATE MODEL
watch_next_agg = watch_next.groupBy(col("id").alias("id_ref")).agg(collect_list(struct(col("related_id").alias("watch_next"),
                                                                                       col("presenterDisplayName").alias("speaker"))))

watch_next_agg.printSchema()

# AND JOIN WITH THE MAIN TABLE
tedx_dataset_main = tedx_dataset.join(watch_next_agg, tedx_dataset.id == watch_next_agg.id_ref, "left") \
    .drop("id_ref")

tedx_dataset_main.printSchema()
```

MongoDB

Esempio **visualizzazione** dati

```python
if len(r.json()[0]) > 0 and r.json()[0]["data"] is not None and r.json()[0]["data"]["video"] is not None:
  details.append({
          "id": video["id"],
          "slug": video["slug"],
          "interalId": r.json()[0]['data']["video"]["id"],
          "description": r.json()[0]['data']["video"]["description"],
          "duration": r.json()[0]['data']["video"]["duration"],
          "socialDescription": r.json()[0]['data']["video"]["socialDescription"],
          "presenterDisplayName": r.json()[0]['data']["video"]["presenterDisplayName"],
          "publishedAt": r.json()[0]['data']["video"]["publishedAt"],
          "publishedAt": r.json()[0]['data']["video"]["publishedAt"]
      }
  )
if len(r.json()[0]["data"]["video"]["presenterDisplayName"]) > 0:
  for presenter in r.json()[0]["data"]["video"]["presenterDisplayName"]:
    speaker.append({
        "id": video["id"],
        "presenterDisplayName": r.json()[0]['data']["video"]["presenterDisplayName"],
        "duration": r.json()[0]['data']["video"]["duration"],
    })
if len(r.json()[0]["data"]["video"]["primaryImageSet"]) > 0:
```

```python
[ ]  df = pd.DataFrame.from_dict(images)
     df = df.to_csv('tags.csv', index=False)

[ ]  df = pd.DataFrame.from_dict(final_list)
     df = df.to_csv('final_list.csv', index=False)

[ ]  df = pd.DataFrame.from_dict(speaker)
     df = df.to_csv('speaker.csv', index=False)
```

```python
[ ]  final_list = []
     for talk in final:
       slug = talk["slug"]
       final_list.append({
           'id': talk["objectID"],
           'slug': talk["slug"],
           'speakers': talk["speakers"],
           'title': talk["title"],
           "url": f'https://www.ted.com/talks/{slug}',
           'duration': talk["duration"]
       })

[ ]  final_list[0]
```

WebScraping

```python
[ ]  details = []
     images = []
     tags = []
     related_videos = []
     ready = []
     speaker = []
```

TedxSpeaker

- Import della libreria functions
- Conteggio del numero dei video per speaker

```
4  import sys
5  import json
6  import pyspark
7  from pyspark.sql.functions import col, collect_list, array_join, struct
8  import pyspark.sql.functions as func
9
10 from awsglue.transforms import *
11 from awsglue.utils import getResolvedOptions
12 from pyspark.context import SparkContext
13 from awsglue.context import GlueContext
14 from awsglue.job import Job
15
```

```
##IL MAIN è SPEAKER POI AGGIUNGERE COME SOTTO CAMPI ID E ALTRE INFO DETTAGLIATE COME LINK PAR VIDEO ETC...
## READ THE SPEAKER----------------------------------------------------------------------

speaker_dataset = speaker_dataset.dropDuplicates(['id'])
#speaker_dataset.printSchema()

tedx_dataset_main = speaker_dataset.groupBy('presenterDisplayName').count().select(func.col("count").alias("nrVideos"),
                                                                                   func.col("presenterDisplayName"))
```

TedxSpeaker

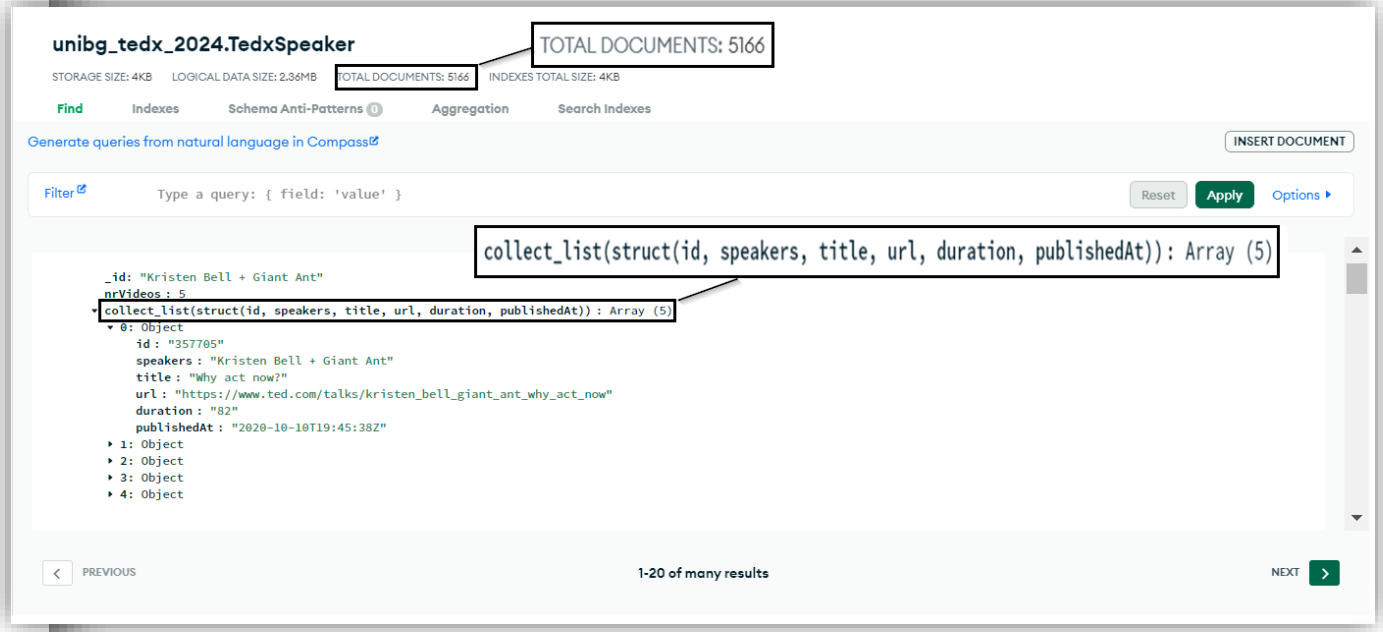
- Lettura colonne selezionate da final_list.csv
- Letura colonne selezionate da details.csv
- Unione colonne final_list + details
- Collect list di struct del risultato precedente

```
## LETTURA FINAL_LIST.CSV (ID | SPEAKERS | TITLE | URL)
tedx_path = "s3://tedx-data-mz/final_list.csv"
tedx_dataset = spark.read.option("header","true").csv(tedx_path)

tedx_dataset = tedx_dataset.select(col("id"),
                                   col("speakers"),
                                   col("title"),
                                   col("url"))

##  LETTURA DA DETAILS.CSV (ID | DURATION | publishedAt)
detail_path = "s3://tedx-data-mz/details.csv"
detail_dataset = spark.read.option("header","true").csv(detail_path)

detail_dataset = detail_dataset.select(col("id").alias("id_ref"),
                                   col("duration"),
                                   col("publishedAt"))

##  UNIONE COLONNE FINAL_LIST.CSV E DETAILS.CSV
tedx_dataset = tedx_dataset.join(detail_dataset, tedx_dataset.id == detail_dataset.id_ref, "left") \
    .drop("id_ref")
tedx_dataset.printSchema()

# CREATE THE AGGREGATE MODEL
tedx_dataset_agg = tedx_dataset.groupBy(col("speakers")).agg(collect_list(struct(col("id"), col("speakers"), col("title"),
                                                                                 col("url"), col("duration"), col("publishedAt"))))
tedx_dataset_agg.printSchema()
```

```
tedx_dataset_main = tedx_dataset_main.join(tedx_dataset_agg, tedx_dataset_main.presenterDisplayName == tedx_dataset_agg.speakers, "left") \
    .drop("speakers") \
    .select(col("presenterDisplayName").alias("_id"), col("*")) \
    .drop("presenterDisplayName") \
    .drop("id") \

tedx_dataset_main.printSchema()
```

MongoDB

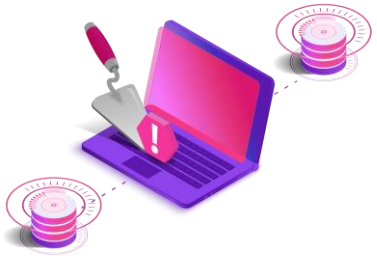

```
_id: "Kristen Bell + Giant Ant"
nrVideos : 5
collect_list(struct(id, speakers, title, url, duration, publishedAt)) : Array (5)
  0: Object
     id : "357705"
     speakers : "Kristen Bell + Giant Ant"
     title : "Why act now?"
     url : "https://www.ted.com/talks/kristen_bell_giant_ant_why_act_now"
     duration : "82"
     publishedAt : "2020-10-10T19:45:38Z"
  1: Object
     id : "357749"
     speakers : "Kristen Bell + Giant Ant"
     title : "Where does all the carbon we release go?"
     url : "https://www.ted.com/talks/kristen_bell_giant_ant_where_does_all_the_ca…"
     duration : "83"
     publishedAt : "2020-10-10T17:35:48Z"
  2: Object
     id : "357703"
     speakers : "Kristen Bell + Giant Ant"
     title : "Why is 1.5 degrees such a big deal?"
     url : "https://www.ted.com/talks/kristen_bell_giant_ant_why_is_1_5_degrees_su…"
     duration : "70"
     publishedAt : "2020-10-10T17:01:05Z"
  3: Object
  4: Object
```

Criticità

- Pulizia dati duplicati

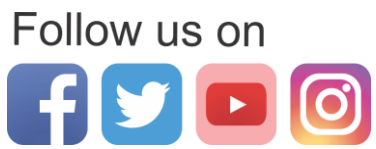
- Web scraping



- Conteggio e aggiunta del numero di video per speaker

Evolizioni

- Web scraping profili facebook, instagram e X



- Aggiunta di contatti per ogni speaker

- Aggiunta di sottotitoli per la generazione di quiz

Grazie

Mehmood Zakria

https://github.com/zakria-mehmood/unibg_cloud_e_mobile_2024

z.mehmood@unibg.it

https://trello.com/invite/b/PhlMCIv5/ATTI53506od83afb7092dd5d9ce5fa9e7331B166BE6F/talkwithted

Trello