Homework 3: MongoDB

Studenti: Antimo Barbato
Matricola: M63/1079

1: Introduzione ad MongoDB

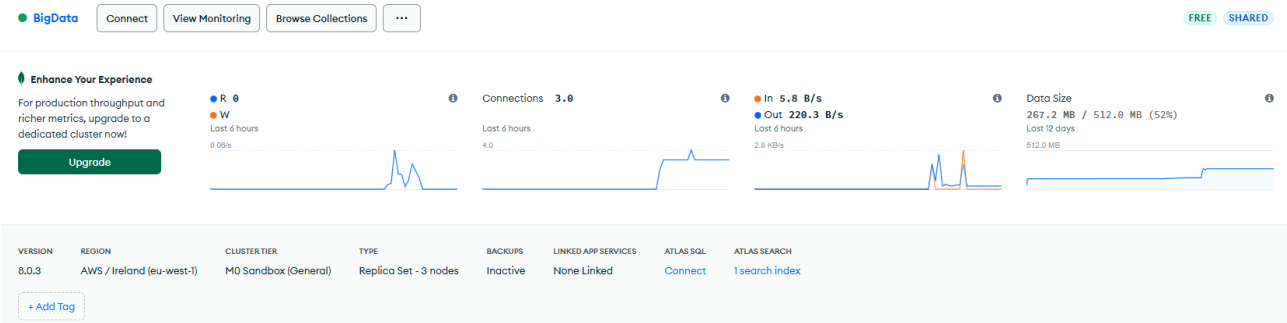
MongoDB è un database NoSQL che memorizza i dati in un formato di documenti JSON (JavaScript Object Notation) altamente flessibile e scalabile. A differenza dei tradizionali database relazionali, che utilizzano tabelle e righe per organizzare i dati, MongoDB utilizza documenti e collezioni, che lo rendono particolarmente adatto a gestire dati non strutturati o semi-strutturati.
MongoDB supporta due tecniche principali per l'elaborazione dei dati complessi: **MapReduce** e **Aggregation**. Entrambe sono utilizzate per eseguire operazioni di aggregazione sui dati, ma con differenze significative nelle prestazioni e nelle modalità di utilizzo. Di solito l'aggregation risulta essere più semplice e performante.
Le operazioni di aggregazione vengono eseguite in **pipeline**, dove i dati passano attraverso diverse fasi che possono includere:

- **$match**: Filtrare i documenti.
- **$group**: Raggruppare i documenti per una o più chiavi e calcolare valori aggregati (ad esempio somma, media, conteggio).
- **$sort**: Ordinare i risultati.
- **$project**: Selezionare i campi che vogliamo mantenere.
- **$limit**: Limitare il numero di documenti restituiti.
- **$unwind**: Decomporre un array in documenti separati.

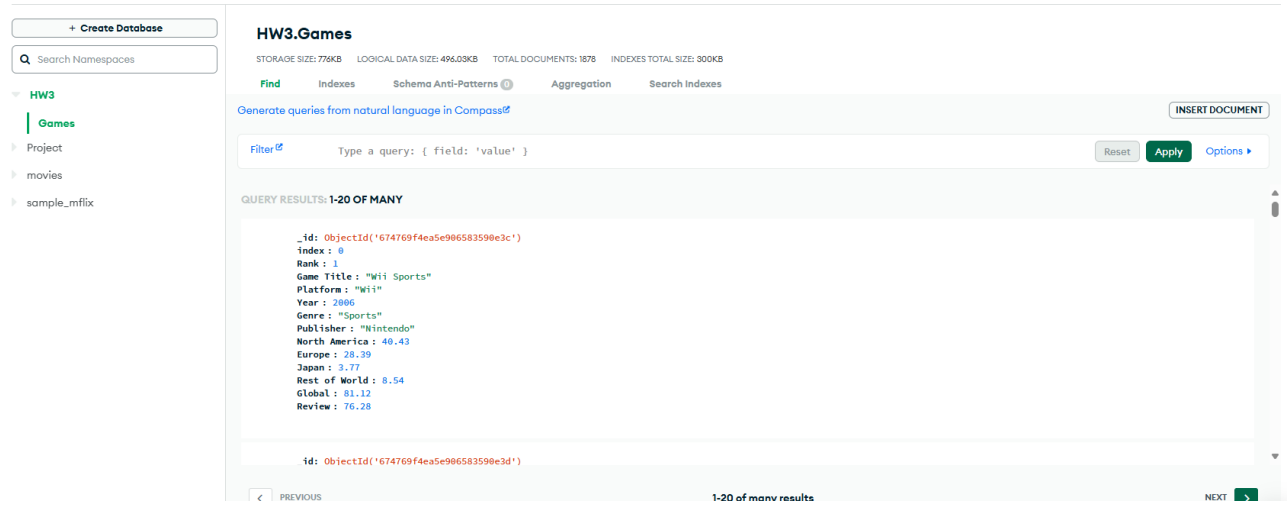
Per l'homework in questione è stato utilizzato nello specifico MongoDB Atlas.

1.2: MongoDB Atlas

Per la realizzazione di questo homework si è creato anzitutto un cluster su mongoDB Atlas



Successivamente è stata creata la basedati e la collezione annessa.



1.3: Google Colab

Google Colab (abbreviazione di **Collaboratory**) è un ambiente di sviluppo interattivo basato su cloud, creato da Google, che consente di scrivere ed eseguire codice Python direttamente nel browser. È uno strumento particolarmente utile per la programmazione, l'analisi dei dati, il machine learning, e la creazione di modelli di intelligenza artificiale.
Google Colab è stato utilizzato per la connessione a mongoDB e per eseguire le query.
Vediamo nello specifico la parte relativa alla connessione e il caricamento del dataset su MongoDB:

1) **Prelievo del dataset da Google Drive**

```
from google.colab import drive
drive.mount('/content/drive')
df = pd.read_csv('/content/drive/MyDrive/videogames.csv')
```

2) **Rimozione valori nulli dal dataset (Pre-Processing)**

```
# Rimuovi le righe con valori nulli
df_cleaned = df.dropna()
```

3) **Connessione e caricamento dei dati su MongoDB**

```
import pandas as pd
import pymongo
from pymongo import MongoClient

def get_mongo_client(mongo_uri):
  """Establish connection to the MongoDB."""
  try:
    client = pymongo.MongoClient(mongo_uri)
    print("Connection to MongoDB successful")
    return client
  except pymongo.errors.ConnectionFailure as e:
    print(f"Connection failed: {e}")
    return None

mongo_uri = "URL"

mongo_client = get_mongo_client(mongo_uri)

# Ingest data into MongoDB
db = mongo_client['HW3']
collection = db['Games']

# Delete any existing records in the collection
collection.delete_many({})

#Inserimento del dataframe in mongoDB come dizionario
documents = df_cleaned.to_dict('records')
collection.insert_many(documents)
```

4) **Estrazione dati dalla collezione di mongoDB e conversione in DataFrame di pandas**

```
# Esegui una query per ottenere tutti i documenti dalla collezione
cursor = collection.find({})

# Converti i risultati della query in una lista di dizionari
data_list = list(cursor)

# Chiudi il cursore
cursor.close()
# Converti la lista di dizionari in un DataFrame
df = pd.DataFrame(data_list)
```

1.4: Dataset – Videogames Sales

Il dataset "Video Games Sales" (formato csv) offre una panoramica dettagliata delle vendite globali di videogiochi, includendo informazioni come il titolo del gioco, la piattaforma, il genere, l'editore, le vendite regionali (Nord America, Europa, Giappone e Resto del Mondo), le vendite globali e le recensioni. Questo dataset è utile per analizzare le tendenze del mercato dei videogiochi, confrontare le performance degli editori e studiare le preferenze dei giocatori in diverse regioni.
Vediamo nel dettaglio gli attributi relativi al dataset

- **Rank**: La classifica del videogioco basata sul volume di vendite globali.
- **Game Title**: Il nome del videogioco.
- **Platform**: La piattaforma su cui è disponibile il gioco, come PC, PS4, Xbox One, ecc.
- **Year**: L'anno in cui il gioco è stato rilasciato.
- **Genre**: Il genere del gioco, come azione, avventura, corse, ecc.
- **Publisher**: La società che ha pubblicato il gioco.
- **North America**: Il numero di unità vendute in Nord America, in milioni.
- **Europe**: Il numero di unità vendute in Europa, in milioni.
- **Japan**: Il numero di unità vendute in Giappone, in milioni.
- **Rest of World**: Il numero di unità vendute nel resto del mondo, esclusi Nord America, Europa e Giappone, in milioni.
- **Global**: Il numero totale di unità vendute in tutto il mondo, in milioni.
- **Review**: Il punteggio delle recensioni del gioco, su una scala da 1 a 10.

2. Query su MongoDB con Google Colab

Si precisa che per avere una visualizzazione dei dati più ottimali, dove possibile, si è usato pandas per ottenere un dataframe in uscita per ottenere una rappresentazione tabellare dei dati risultati dalla query.

Query 1: Conteggio dei giochi con lo stesso nome e ordinati in ordine decrescente

```
pipeline = [
    {"$group": {"_id": "$Game Title", "game_count": {"$sum": 1}}},
    {"$sort": {"game_count": -1}}
]

result = collection.aggregate(pipeline)

# Converti i risultati in un DataFrame pandaas
df = pd.DataFrame(list(result))
# Rinomina il campo _id
df = df.rename(columns={'_id': 'game_name'})
# Stampa il DataFrame in formato tabellare
df.head(10)
```

	game_name	game_count
0	FIFA Soccer 08	6
1	Pro Evolution Soccer 2008	5
2	The Simpsons Game	5
3	LEGO Indiana Jones: The Original Adventures	5
4	WWE SmackDown vs Raw 2008	5
5	Star Wars: The Force Unleashed	5
6	FIFA Soccer 10	5
7	Need for Speed: ProStreet	4
8	The Sims 3	4
9	LEGO Star Wars: The Video Game	4

Query 2: Media vendite globali ordinate in ordine decrescente

```
pipeline = [
    {"$group": {"_id": "$Genre", "Media_Globale": {"$avg": "$Global"}}},
    {"$sort": {"Media_Globale": -1}}
]

result = collection.aggregate(pipeline)

# Converti i risultati in un DataFrame pandaas
df = pd.DataFrame(list(result))

# Rinomina il campo _id
df = df.rename(columns={'_id': 'Genre'})

# Stampa il DataFrame in formato tabellare
df.head(10)
```

	Genre	Media_Globale
0	Platform	3.189677
1	Role-Playing	2.889123
2	Misc	2.726471
3	Shooter	2.713627
4	Puzzle	2.469318
5	Racing	2.438804
6	Action	2.331360
7	Sports	2.281221
8	Simulation	2.229783
9	Adventure	2.191524

Concentriamoci sui primi 3 per la prossima query per vedere quale Publisher ne ha prodotti maggiormente

Query 3: Filtriamo in base ai 3 generi, raggruppiamo in base al publisher e contiamo il numero di giochi prodotti per genere con relativo ordinamento decrescente

```
pipeline = [
    {
        "$match": {
            "Genre": {"$in": ["Platform", "Role-Playing", "Misc"]}
        }
    },
    {
        "$group": {
            "_id": {"Publisher": "$Publisher", "Genre": "$Genre"},
            "game_count": {"$sum": 1}
        }
    },
    {
        "$sort": {
            "game_count": -1
        }
    }
]
result = collection.aggregate(pipeline)

for i in result:
  print(i)
```

```
{'_id': {'Publisher': 'Nintendo', 'Genre': 'Platform'}, 'game_count': 67}
{'_id': {'Publisher': 'Nintendo', 'Genre': 'Role-Playing'}, 'game_count': 38}
{'_id': {'Publisher': 'Nintendo', 'Genre': 'Misc'}, 'game_count': 32}
{'_id': {'Publisher': 'Sega', 'Genre': 'Platform'}, 'game_count': 30}
{'_id': {'Publisher': 'Sony Computer Entertainment', 'Genre': 'Misc'}, 'game_count': 27}
{'_id': {'Publisher': 'Activision', 'Genre': 'Misc'}, 'game_count': 23}
{'_id': {'Publisher': 'Square Enix', 'Genre': 'Role-Playing'}, 'game_count': 23}
{'_id': {'Publisher': 'Sony Computer Entertainment', 'Genre': 'Platform'}, 'game_count': 21}
{'_id': {'Publisher': 'Square', 'Genre': 'Role-Playing'}, 'game_count': 16}
{'_id': {'Publisher': 'THQ', 'Genre': 'Platform'}, 'game_count': 16}
{'_id': {'Publisher': 'Activision', 'Genre': 'Role-Playing'}, 'game_count': 15}
{'_id': {'Publisher': 'Sony Computer Entertainment', 'Genre': 'Role-Playing'}, 'game_count': 13}
{'_id': {'Publisher': 'Ubisoft', 'Genre': 'Misc'}, 'game_count': 13}
{'_id': {'Publisher': 'Activision', 'Genre': 'Platform'}, 'game_count': 10}
{'_id': {'Publisher': 'Capcom', 'Genre': 'Platform'}, 'game_count': 10}
{'_id': {'Publisher': 'Electronic Arts', 'Genre': 'Role-Playing'}, 'game_count': 9}
{'_id': {'Publisher': 'Bethesda Softworks', 'Genre': 'Role-Playing'}, 'game_count': 9}
{'_id': {'Publisher': 'Enix Corporation', 'Genre': 'Role-Playing'}, 'game_count': 8}
```

Continuiamo analizzando a questo punto le vendite globali di Nintendo per il genere Platform

Query 4: Filtro per il genere “Platform” e Publisher “Nintendo.
Calcolo la media delle vendite globali per il publisher Nintendo per il genere “Platform”

```
pipeline = [
    {
        "$match": {
            "Publisher": "Nintendo",
            "Genre": "Platform"
        }
    },
    {
        "$group": {
            "_id": "$Publisher",
            "Vendite_Medie_Gobali": {"$avg": "$Global"}
        }
    }
]

result = collection.aggregate(pipeline)

# Converti i risultati in un DataFrame pandas
df = pd.DataFrame(list(result))

# Rinomina il campo _id
df = df.rename(columns={'_id': 'Publisher'})

# Mostra il DataFrame in formato tabellare
print(df)
```

```
Publisher  Vendite_Medie_Gobali
 Nintendo              5.740597
```

Successivamente calcolo la media delle vendite annue di nintendo

Query 5: Filtro per il Publisher “Nintendo” e calcolo la media di vendite globali annue di Nintendo

```
pipeline = [
    {
        "$match": {
            "Publisher": "Nintendo"
        }
    },
    {
        "$group": {
            "_id": "$Year",
            "AVG_Annuale": {"$avg": "$Global"}
        }
    },
    {
        "$sort": {
            "AVG_Annuale": -1
        }
    }
]

result = collection.aggregate(pipeline)

# Converti i risultati in un DataFrame pandas
df = pd.DataFrame(list(result))

# Rinomina il campo _id
df = df.rename(columns={'_id': 'Year'})

# Mostra il DataFrame in formato tabellare
print(df)
```

	Year	AVG_Annuale
0	2006.0	11.232353
1	1989.0	9.833333
2	1985.0	9.704000
3	2009.0	8.651429
4	1988.0	6.860000
5	2005.0	6.820588
6	1984.0	6.184286
7	2008.0	6.159231
8	1999.0	5.602727
9	2007.0	5.600000
10	1990.0	5.070000
11	1994.0	4.578000
12	1993.0	4.322500
13	1992.0	4.228889
14	1996.0	4.212222
15	2002.0	4.063636
16	2010.0	4.009231
17	1997.0	3.721667
18	1995.0	3.707500
19	1998.0	3.548462
20	2001.0	3.200000

Query 6: Filtro in base a Review e Global per ottenere i giochi con valutazione maggiore di 8 e con vendite superiori a 20 milioni

```
# Esegui l'aggregazione con ordinamento
pipeline = [
    {
        "$match": {
            "Review": {"$gt": 8},
            "Global": {"$gt": 20}
        }
    },
    #Voglio visualizzare solo alcuni attributi
    {
        "$project": {
            "Game Title": 1,
            "Review": 1,
            "Global": 1,
            "_id" : 0 #Escludo l'id per avere un risultato più pulito
        }
    },
    {
        "$sort": {
            "Global": -1  # Ordina in base a "Global" in ordine decrescente
        }
    }
]

result = collection.aggregate(pipeline)

# Converti i risultati in un DataFrame pandas
df = pd.DataFrame(list(result))

# Mostra il DataFrame in formato tabellare
df.head(10)
```

	Game Title	Global	Review
0	Wii Sports	81.12	76.28
1	Super Mario Bros.	40.24	91.00
2	Mario Kart Wii	33.55	82.07
3	Wii Sports Resort	31.52	82.65
4	Tetris	30.26	88.00
5	New Super Mario Bros.	29.08	90.00
6	Wii Play	28.71	61.64
7	Duck Hunt	28.31	84.00
8	New Super Mario Bros. Wii	26.75	88.18
9	Nintendogs	24.50	85.00

A seguito di tale risultato si è deciso di inserire nella select anche l'attributo relativo all'anno, per verificare il primo gioco a che anno fa riferimento, considerando che nintendo ha le migliori vendite medie annue nel 2006

```
# Esegui l'aggregazione con ordinamento
pipeline = [
    {
        "$match": {
            "Review": {"$gt": 8},
            "Global": {"$gt": 20}
        }
    },
    #Voglio visualizzare solo alcuni attributi
    {
        "$project": {
            "Game Title": 1,
            "Review": 1,
            "Global": 1,
            "Year" : 1, #Aggiungo Year
            "_id" : 0 #Escludo l'id per avere un risultato più pulito
        }
    },
    {
        "$sort": {
            "Global": -1  # Ordina in base a "Global" in ordine decrescente
        }
    }
]

result = collection.aggregate(pipeline)

# Converti i risultati in un DataFrame pandas
df = pd.DataFrame(list(result))

# Mostra il DataFrame in formato tabellare
df.head(10)
```

	Game Title	Year	Global	Review
0	Wii Sports	2006.0	81.12	76.28
1	Super Mario Bros.	1985.0	40.24	91.00
2	Mario Kart Wii	2008.0	33.55	82.07
3	Wii Sports Resort	2009.0	31.52	82.65
4	Tetris	1989.0	30.26	88.00
5	New Super Mario Bros.	2006.0	29.08	90.00
6	Wii Play	2006.0	28.71	61.64
7	Duck Hunt	1984.0	28.31	84.00
8	New Super Mario Bros. Wii	2009.0	26.75	88.18
9	Nintendogs	2005.0	24.50	85.00

Come si può notare nella top 10 rientrano 3 giochi Nintendo con un numero di vendite globali pari a 81 milioni, 29 milioni e 28 milioni per quanto riguarda l'anno 2006.
A questo punto può essere interessante **vedete tutti i giochi venduti da Nintendo nel 2006**.

Query 7: Filtro per Publisher e Anno.
Ordino in base alle vendite globali per ottenere una lista dei giochi più venduti di nintendo nel 2006

```
pipeline = [
    {
        "$match": {
            "Publisher": "Nintendo",
            "Year": 2006
        }
    },
    {
        "$project": {
            "Game Title": 1,
            "Global": 1,
            "_id": 0  # Escludi il campo _id
        }
    },
    {
        "$sort": {
            "Global": -1  # Ordina in base a "Global" in ordine decrescente
        }
    }
]

result = collection.aggregate(pipeline)

# Converti i risultati in un DataFrame pandas
df = pd.DataFrame(list(result))

# Mostra il DataFrame in formato tabellare
print(df)
```

	Game Title	Global
0	Wii Sports	81.12
1	New Super Mario Bros.	29.08
2	Wii Play	28.71
3	Pokémon Diamond / Pearl Version	18.05
4	The Legend of Zelda: Twilight Princess	6.76
5	Clubhouse Games	3.35
6	English Training: Have Fun Improving Your Skills!	3.33
7	Personal Trainer: Cooking	3.09
8	WarioWare: Smooth Moves	2.84
9	Yoshi's Island DS	2.69
10	Pokémon Ranger	2.11
11	Tetris DS	2.07
12	Kirby Squeak Squad	1.91
13	The Legend of Zelda: Twilight Princess	1.59
14	Mario Hoops 3 on 3	1.57
15	Pokémon Battle Revolution	1.52
16	Mario vs. Donkey Kong 2: March of the Minis	1.16

Query 8: Raggruppo in base a platform e year e effettuo il conteggio per ottenere il numero di giochi in base all'anno

```
pipeline = [
    {
        "$group": {
            "_id": {"Platform": "$Platform", "Year": "$Year"},
            "game_count": {"$sum": 1}
        }
    },
    {
        "$sort": {
            "game_count": -1
        }
    }
]

result = collection.aggregate(pipeline)

counter = 0

for i in result:
    if counter < 20:
        print(i)
        counter += 1
    else:
        break
```

```
{'_id': {'Platform': 'PS2', 'Year': 2002.0}, 'game_count': 66}
{'_id': {'Platform': 'PS2', 'Year': 2004.0}, 'game_count': 64}
{'_id': {'Platform': 'PS2', 'Year': 2003.0}, 'game_count': 59}
{'_id': {'Platform': 'PS', 'Year': 1998.0}, 'game_count': 55}
{'_id': {'Platform': 'PS2', 'Year': 2005.0}, 'game_count': 48}
{'_id': {'Platform': 'PS', 'Year': 1999.0}, 'game_count': 45}
{'_id': {'Platform': 'X360', 'Year': 2010.0}, 'game_count': 43}
{'_id': {'Platform': 'PS2', 'Year': 2001.0}, 'game_count': 43}
{'_id': {'Platform': 'PS3', 'Year': 2008.0}, 'game_count': 43}
{'_id': {'Platform': 'X360', 'Year': 2008.0}, 'game_count': 42}
{'_id': {'Platform': 'Wii', 'Year': 2008.0}, 'game_count': 40}
{'_id': {'Platform': 'PS', 'Year': 1997.0}, 'game_count': 38}
{'_id': {'Platform': 'PS3', 'Year': 2010.0}, 'game_count': 37}
{'_id': {'Platform': 'DS', 'Year': 2007.0}, 'game_count': 37}
{'_id': {'Platform': 'Wii', 'Year': 2007.0}, 'game_count': 36}
{'_id': {'Platform': 'X360', 'Year': 2011.0}, 'game_count': 36}
{'_id': {'Platform': 'DS', 'Year': 2008.0}, 'game_count': 35}
{'_id': {'Platform': 'PS3', 'Year': 2011.0}, 'game_count': 34}
{'_id': {'Platform': 'PS3', 'Year': 2009.0}, 'game_count': 33}
{'_id': {'Platform': 'Wii', 'Year': 2009.0}, 'game_count': 33}
```

Query 9: Raggruppo per Publisher e effettuo la somma relative alle vendite sia regionali che globali e ordino in base alle globali

```
pipeline = [
    {
        "$group": {
            "_id": "$Publisher",
            "Vendite_NA": {"$sum": "$North America"},
            "Vendite_EU": {"$sum": "$Europe"},
            "Vendite_JP": {"$sum": "$Japan"},
            "Vendite_RW": {"$sum": "$Rest of World"},
            "Vendite_Globali": {"$sum": "$Global"}
        }
    },
    {
        "$sort": {
            "Vendite_Globali": -1
        }
    }
]

result = collection.aggregate(pipeline)

# Converti i risultati in un DataFrame pandas
df = pd.DataFrame(list(result))

# Rinomina il campo _id
df = df.rename(columns={'_id': 'Publisher'})

# Mostra il DataFrame in formato tabellare
df.head(10)
```

	Publisher	Vendite_NA	Vendite_EU	Vendite_JP	Vendite_RW	Vendite_Globali
0	Nintendo	687.79	341.49	338.04	80.54	1447.81
1	Electronic Arts	345.25	203.04	8.72	67.23	624.18
2	Sony Computer Entertainment	167.59	119.19	50.27	40.51	377.61
3	Activision	218.68	112.82	3.66	34.85	369.98
4	Take-Two Interactive	112.64	67.43	3.82	24.50	208.42
5	Ubisoft	103.23	69.61	1.79	21.69	196.32
6	Microsoft Game Studios	110.68	43.22	2.37	13.37	169.73
7	THQ	80.19	47.00	2.58	13.27	142.98
8	Sega	60.53	39.73	11.26	10.27	121.80
9	Capcom	49.42	24.31	32.91	7.76	114.33