

Progetto Basi di Dati e di Conoscenza:

**“Stagionalità Cibi”**

**Relazione MongoDB**



Zagaria Simone

0282174

A.A. 2021/2022

Docente: Loredana Vigliano

Creazione e Inserimento Dati

Per inserire i dati all’interno di MongoDB è stato scritto un codice Python che generasse un elevato numero di tuple e le inserisse direttamente nel database in MySql (Guardare il file “genera\_tuple\_StagionalitàCibi.py” allegato).

In seguito, è stata creata una view di un ramo del database considerando le tabelle Persona e EffettuataDa, attraverso:

CREATE VIEW Persona\_Effettua AS

SELECT P.CF,P.nome,P.cognome,P.data\_nascita,P.indirizzo,P.telefono, P.email,ED.id\_dieta,ED.inizio\_dieta,ED.fine\_dieta

FROM Persona P LEFT JOIN EffettuataDa ED ON P.CF = ED.CF;

In seguito, a partire da questa view è stato generato un file .json di questi dati, attraverso le istruzioni:

SELECT JSON\_ARRAYAGG(JSON\_OBJECT('CF',CF,'nome',nome,'cognome',cognome, 'data\_nascita',data\_nascita,'indirizzo',indirizzo,'telefono',telefono,'email',email,'id\_dieta', id\_dieta,'inizio\_dieta',inizio\_dieta,'fine\_dieta',fine\_dieta))

FROM Persona\_Effettua

INTO OUTFILE 'C:\Program Files\MySQL\MySQL Server 8.0\Uploads\inserimenti\_persona\_effettua.json';

In tutto, la collezione persona\_effettua contiene circa 273.000 documenti.

Query Statistiche

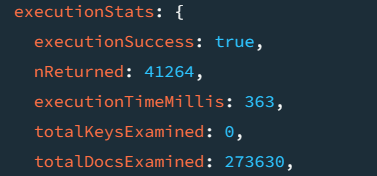
Sono state eseguite diverse query per confrontare i due DBSM e verificarne le differenze:

## Query 1

Si vuole sapere tutte le persone nate prima del 1960, ordinate in base al cognome e nome

MongoDB

db.persona\_effettua.find({data\_nascita: {$lt: "1960-01-01"}}).sort({cognome:1,nome:1});



MySql

SELECT \*

FROM persona\_effettua

WHERE data\_nascita < "1960-01-01"

ORDER BY cognome,nome;

Immagine che contiene testo

Descrizione generata automaticamente

Confronto tempi di esecuzione

MySql ⇒ 1.28 sec

MongoDB ⇒ 0.363 sec

## Query 2

Per ogni dieta, si vuole sapere il numero di volte che è stata effettuata

MongoDB

db.persona\_effettua.aggregate([{"$group": {"\_id":"$id\_dieta", "count":{"$sum":1}}}]);

Immagine che contiene testo

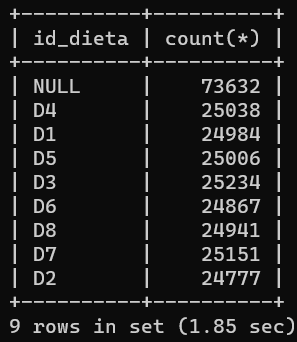
Descrizione generata automaticamente

MySql

SELECT id\_dieta,count(\*)

FROM persona\_effettua

GROUP BY id\_dieta;



Confronto tempi di esecuzione

MySql ⇒ 1.85 sec

MongoDB ⇒ 0.285 sec

## Query 3

Si vogliono sapere le persone che almeno una volta hanno il nome che inizia per "Dar" e hanno effettuato la dieta “D1”

MongoDB

db.persona\_effettua.find({nome: /^Dar/ ,id\_dieta: "D1" });

Immagine che contiene testo

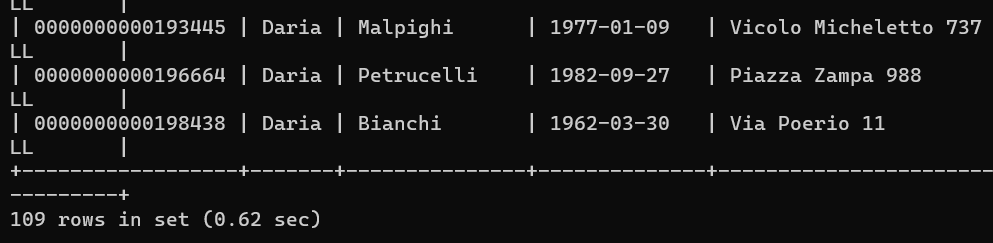
Descrizione generata automaticamente

MySql

SELECT \*

FROM persona\_effettua

WHERE nome LIKE "Dar%" AND id\_dieta = "D1";



Confronto tempi di esecuzione

MySql ⇒ 0.62 sec

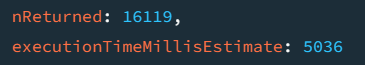
MongoDB ⇒ 0,516 sec

## Query 4

Si vogliono sapere il CF, il nome, il cognome e il numero di diete iniziate di tutte le persone che hanno iniziato 3 o più diete

MongoDB

db.persona\_effettua.aggregate([{"$group": {"\_id": {"CF": "$CF","nome": "$nome","cognome": "$cognome"}, "count":{"$sum":1}}},{"$match": {"count": {"$gte":3}}}]);



MySql

SELECT CF,nome,cognome,count(inizio\_dieta)

FROM persona\_effettua

GROUP BY CF,nome,cognome

HAVING count(inizio\_dieta) >= 3;

Immagine che contiene testo

Descrizione generata automaticamente

Confronto tempi di esecuzione

MySql ⇒ 11.38 sec

MongoDB ⇒ 5.036 sec

Dai confronti effettuati si può concludere che, a fronte di una grande quantità di dati, le prestazioni di MongoDB superano di molto quelle di MySql.