Universitat Autònoma de Barcelona

Facultat de Ciències

Projecte final MongoDB

Bases de dades no relacionals

Francesc Albareda Civit - 1603751
Alba Fernández Coronado - 1600123
Marina Palomar González - 1605547
Guillem Paz García - 1598850

Abril 2023

Contents

1 Introducció

Aquest treball final tracta sobre el disseny, la implementació i la consulta a una base de dades en MongoDB implementada manualment a través d'un script de Python.

L'objectiu és acabar d'assolir els conceptes teòrics de l'assignatura mitjançant la realització del projecte en grups.

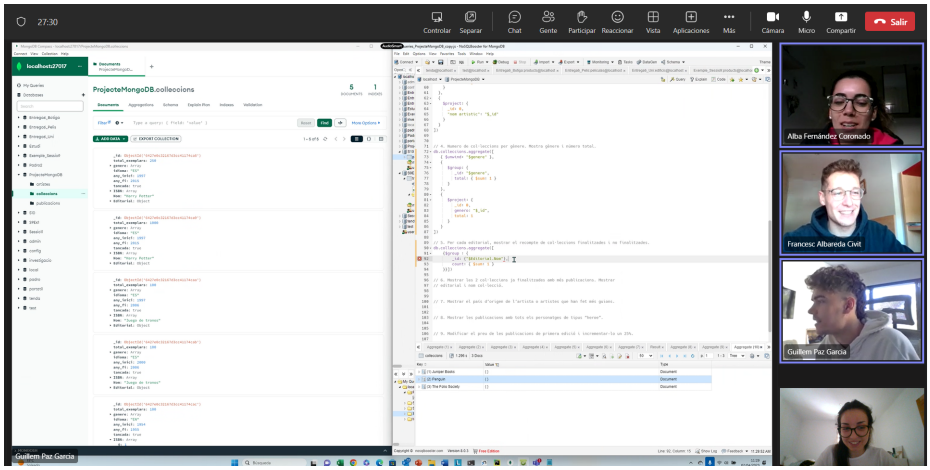
2 Repartició de la feina

El grup de treball està format per quatre integrants: Francesc Albareda, Alba Fernández, Marina Palomar i Guillem Paz.

Des de l'inici del projecte vam distribuir la feina per tal d'esprémer al màxim les habilitats de cada membre i, tot i que la majoria hem treballat en tots els exercicis del projecte, hem tractat d'assignar qui s'encarregaria de liderar-lo i realitzar els *commits* al *Github* necessaris.

L'Alba va ser l'encarregada de liderar l'exercici 1, d'apuntar i verificar les idees proposades per tots els membres del grup sobre quins patrons de disseny aplicaríem a la base de dades per tal de fer-la més compacta i eficient. En Francesc s'ha encarregat de dirigir la part més interna del projecte, el disseny i implementació del codi en Python per tal de plasmar l'esquema generat anteriorment a la base de dades real, juntament amb l'ajuda col·laborativa de tots els membres per tal de no deixar passar cap mínima errada. Finalment, tant el Guillem com la Marina s'han encarregat de portat tota la part més externa del projecte, les consultes a la base de dades. Mitjançant reunions dels quatre components, s'anaven posant idees sobre la taula fins que la *query* generada complia tots els requisits de l'enunciat demanat.

Com bé s'ha explicat anteriorment, malgrat que cadascú ha tingut el seu rol específic el qual liderar, s'ha treballat en col·laboració per tal d'aconseguir els objectius del projecte. A causa de la distància d'on vivim, hem usat diàriament la plataforma *Teams* per comunicar-nos i compartir informació.



3 Exercici 1

Després de fer una anàlisi previ de les dades, la seva estructura i el seu ús teòric, s'ha decidit aplicar els següents patrons:

Per a relacionar personatges amb les publicacions s'ha aplicat un embedded, ja que tot i ser de cardinalitat N-N, normalment no es repeteixen gaire (només per col·leccions) i aquests simplement contenen dos atributs (nom i tipus).

Pel que fa als artistes, s'ha utilitzat una referència a publicacions, perquè aquests contenen 5 atributs i la relació és N-N. A més, un artista pot ser un guionista en una publicació i dibuixant en una altra. D'aquesta manera tindrem dos atributs de tipus array a les publicacions, un per als guionistes i l'altra per als dibuixants, totes dues amb l'identificador únic de l'artitsta (nom atrístic).

Per determinar a quina col·lecció pertany cada publicació es va optar per una referència on el document col·lecció contingués un array de totes les seves publicacions (ja que és una relació 1-N) i amb bastants atributs. Aquesta referència es fa amb l'ISBN, identificador únic de cada publicació.

Finalment, a les editorials, se'ls ha aplicat un patró embedded afegint-les així a col·leccions, perquè soc pocs atributs i tot i ser una relació de cardinalitat editorial 1-N col·leccions aquest valor no sembla ser gaire gran.

4 Exercici 2

Inicialment, es disposa d'un fitxer Excel (en format *.xsls*) amb tres pestanyes anomenades Artistes, Personatges i Col·leccions-Publicacions. El primer pas ha estat separar manualment les tres pestanyes i guardar-les en format *.csv*, aquests estan adjuntats a l'entrega, ja que són necessaris per a executar el fitxer *.py* amb el tractament de les dades.

Primerament, es realitza la connexió a la base de dades i es comprova l'existència de les tres col·leccions que s'han d'omplir. Si aquestes existeixen s'eliminen per a crear-les de nou i omplir-les. Si no existien simplement es creen.

Cal remarcar que abans de carregar les dades s'han dut a terme dos canvis en les dades eliminant els accents de les paraules Gràcia i Cabutí del fitxer Col·leccions-Publicacions, ja que posteriorment donaven un error de sintaxi amb la codificació UTF-8 utilitzada, a més de modificar també el fitxer Personatges on un camp que hauria de dir "heroe" deia "heror".

Per a fer la part de tractament de dades s'han emprat diccionaris i DataFrames de la llibreria pandas, ja que en tenim un coneixement avançat i es poden passar fàcilment a *json*. El primer pas ha estat carregar els tres fitxers *.csv* en format DataFrame. Si analitzem com estan distribuïdes les dades, podem veure que en el fitxer Col·leccions-Publicacions ja conté dos atributs anomenats guionistes i dibuixants de tipus *array*, on apareixen els noms artístics dels artistes que hi participen. Per tant, la referència proposada pels artistes i les publicacions ja està contemplada amb el format amb el qual hem rebut les dades.

Per crear la col·lecció de Col·leccions (amb l'embedded d'editorial) simplement s'ha de crear un subdataframe amb els atributs necessaris per a aquestes col·leccions i l'ISBN, a través del qual es fa la referència. Un cop fet això es passa a format diccionari i es modifica el format per a ajuntar els atributs d'Editorial en un sol atribut amb nom editorial. Per a fer la referència es crea un nou atribut on hi ha guardat el resultat d'aplicar un groupby per nom de col·lecció, l'identificador d'aquestes. El resultat del grupby ha estat crear una llista amb els valors d'ISBN no repetits per a cada agrupació. D'aquesta manera queda la referència amb els ISBN de cada publicació. Per a publicacions també es fa un subdataframe amb els atributs necessaris.

Pel que fa als personatges, aquests estan relacionats amb les publicacions amb un atribut ISBN per a cada objecte del fitxer personatges. Per a aplicar l'embedded proposat a l'apartat anterior simplement s'ha hagut de recórrer el dataframe personatges fila per fila i efectuar algunes operacions d'assignació. L'atribut ISBN s'utilitza com a clau d'un diccionari, i el valor relacionat amb cada clau és una llista de diccionaris que contenen els atributs de l'embedded (nom i tipus de cada objecte). D'aquesta manera s'ha ajuntat els personatges per publicació transformant la referència rebuda en embedded.

Un cop ja es té el format desitjat per a publicacions i el diccionari amb l'embedded dels personatges simplement s'ha d'ajuntar amb l'atribut ISBN.

Un cop aplicats els patrons a les col·leccions, simplement s'ha de convertir els diccionaris i dataframes resultants a format .json, inserir-ho a la base de dades i comprovar-ne el correcte funcionament.

5 Exercici 3

En aquest apartat es tractarà d'implementar les consultes requerides que s'adequïn als enunciats proporcionats.

Primer, per tal de fer la importació de les dades la base de dades mitjançant el *script* programat anteriorment s'han separat els diferents fitxers exportats prèviament a *csv* i, un cop executat el codi per generar-la, s'ha obert el programa *NoSQLBooster for MongoDB* des d'on s'han realitzat les consultes següents:

1. Les 5 publicacions amb major preu. Mostrar només el títol i preu:

```
db.publicacions.aggregate([
    { $project: { titol: 1, preu: 1, _id: 0 } },
    { $sort: { preu: -1 } },
    { $limit: 5 }
])
```

Key	Value	Type
(1)	{ titol : "Dracula", preu : 125.5 }	Object
titol	Dracula	String
preu	125,5.0	Double
(2)	{ titol : "Tragedias", preu : 85.4 }	Object
(3)	{ titol : "Romances", preu : 72.4 }	Object
(4)	{ titol : "Crimen y castigo", preu : 59.4 }	Object
(5)	{ titol : "En el Este", preu : 43.5 }	Object

2. Valor màxim, mínim i mitjà del preus de les publicacions de l'editorial Juniper Books:

```
db.publicacions.aggregate([
   {
       $lookup: {
           from: "colleccions",
           localField: "ISBN",
           foreignField: "ISBN",
           as: "infocol"
       }},
       {$match: {'infocol.0.Editorial.Nom': "Juniper Books"}},
       {$group: {_id: null, maxim: {$max: "$preu"}, minim: {$min: "$preu"}, mitjana: {$a
       {$project: {_id: 0}}])
```

Key	Value	Type
(1)	{ maxim : 32.5, minim : 27.85, mitjana : 29.118181818181812 }	Object
maxim	32.5.0	Double
minim	27.85.0	Double
mitjana	29.1182.0	Double

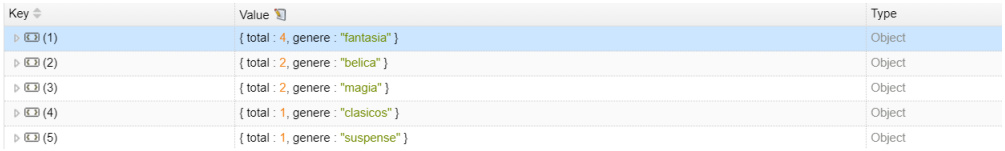
3. Artistes (nom artístic) que participen en més de 5 publicacions com a dibuixant:

```
db.publicacions.aggregate([
    {$match: {"dibuixants": { $exists: true } } },
    {$unwind: "$dibuixants"  },
    {$group: {
        _id: "$dibuixants",
        count: { $sum: 1 } } },
    {$match: {
        count: { $gt: 5 } } },
    {$project: {
        _id: 0,
        "nom_artistic": "$_id"  } }
])
```

Key	Value	Type
(1)	{ "nom artistic" : "Artista1" }	Object
(2)	{ "nom artistic" : "Artista2" }	Object

4. Numero de col·leccions per gènere. Mostra gènere i número total:

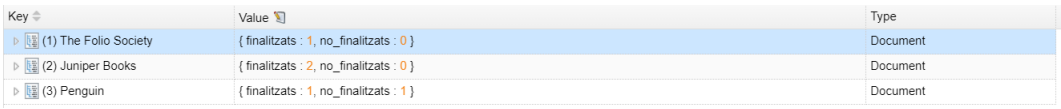
```
db.colleccions.aggregate([
    {$unwind: "$genere" },
    {$group: {
        _id: "$genere",
        total: { $sum: 1 } } },
    {$project: {
        _id: 0,
        genere: "$_id",
        total: 1 } },
    {$sort: {total: -1, genere: 1} }
])
```



Key	Value	Type
(1)	{ total : 4, genere : "fantasia" }	Object
(2)	{ total : 2, genere : "belica" }	Object
(3)	{ total : 2, genere : "magia" }	Object
(4)	{ total : 1, genere : "clasicos" }	Object
(5)	{ total : 1, genere : "suspense" }	Object

5. Per cada editorial, mostrar el recompte de col·leccions finalitzades i no finalitzades:

```
db.colleccions.aggregate([
    {$group: {
        _id: "$Editorial.Nom",
        finalitzats: {
            $sum: {$cond: [{ $eq: ["$tancada", true] }, 1, 0] } },
        no_finalitzats: {
            $sum: {$cond: [{ $eq: ["$tancada", false] }, 1, 0]}} } }
])
```



Key	Value	Type
(1) The Folio Society	{ finalitzats : 1, no_finalitzats : 0 }	Document
(2) Juniper Books	{ finalitzats : 2, no_finalitzats : 0 }	Document
(3) Penguin	{ finalitzats : 1, no_finalitzats : 1 }	Document

6. Mostrar les 2 col·leccions ja finalitzades amb més publicacions. Mostrar editorial i nom col·lecció:

```
db.colleccions.aggregate([
    {$match: {tancada: true}},
    {$group: {
        _id: { nom: "$Nom", editorial: "$Editorial.Nom" },
        total_publicacions: {$sum: {$size: "$ISBN"}}}},
    {$sort: {total_publicacions: -1}},
    {$limit: 2},
    {$project: {
        "_id.nom": 1,
        "_id.editorial": 1}}
])
```

Key	Value	Type
(1) { nom : "Harry Potter", editorial :	{ }	Document
_id	{ nom : "Harry Potter", editorial : "Juniper Books" }	Object
(2) { nom : "Harry Potter", editorial :	{ }	Document
_id	{ nom : "Harry Potter", editorial : "Penguin" }	Object

7. Mostrar el país d'origen de l'artista o artistes que han fet més guions:

```
db.artistes.aggregate([
    {$lookup: {
        from: "publicacions",
        localField: "Nom_artistic",
        foreignField: "guionistes",
        as: "guiones"}},
    {$unwind: "$guiones"},
    {$group: {
        _id: {nom: "$Nom_artistic", pais: "$pais"},
        num_guiones: {$sum: 1}}},
    {$sort: {num_guiones: -1}},
    {$limit: 1},
    {$project: {"_id.pais": 1}}
])
```

Key	Value	Type
(1) { nom : "Artista7", pais : "Norueg	{ num_guiones : 9 }	Document
(2) { nom : "Artista6", pais : "Espany	{ num_guiones : 8 }	Document
(3) { nom : "Artista4", pais : "Italia" }	{ num_guiones : 7 }	Document

Key	Value	Type
(1) { pais : "Noruega" }	{ }	Document

8. Mostrar les publicacions amb tots els personatges de tipus "heroe":

```
db.publicacions.find({"personatges": { "$exists": true,
"$not": { "$elemMatch": { "tipus": { "$ne": "heroe" } } } } },
{ "ISBN": 1, "_id": 0 })
```

Key	Value	Type
(1)	{ ISBN : 4 }	Object
(2)	{ ISBN : 20 }	Object
(3)	{ ISBN : 22 }	Object

9. Modificar el preu de les publicacions amb stock superior a 20 exemplars i incrementar-lo un 25%:

```
db.publicacions.updateMany(
    {stock: { $gt: 20 } },
    { $mul: { preu: 1.25 } }
)
db.publicacions.aggregate([
    {$project: {stock: 1, preu:1}}
])
```

Key	Value	Type
(1)	{ acknowledged : true, matchedCount : 7, modifiedCount : 7 }	Object
acknowledged	true	Bool
matchedCount	7	Int32
modifiedCount	7	Int32

Key	Value	Type
(1) 642daa4b4e3848e9fe46c411	{ stock : 20, preu : 32.5 }	Document
(2) 642daa4b4e3848e9fe46c412	{ stock : 5, preu : 32.5 }	Document
(3) 642daa4b4e3848e9fe46c413	{ stock : 50, preu : 40.625 }	Document
(4) 642daa4b4e3848e9fe46c414	{ stock : 7, preu : 27.85 }	Document
(5) 642daa4b4e3848e9fe46c415	{ stock : 6, preu : 27.85 }	Document
(6) 642daa4b4e3848e9fe46c416	{ stock : 2, preu : 27.85 }	Document
(7) 642daa4b4e3848e9fe46c417	{ stock : 22, preu : 34.8125 }	Document
(8) 642daa4b4e3848e9fe46c418	{ stock : 13, preu : 27.85 }	Document
(9) 642daa4b4e3848e9fe46c419	{ stock : 44, preu : 34.8125 }	Document
(10) 642daa4b4e3848e9fe46c41a	{ stock : 7, preu : 27.85 }	Document

10. Mostrar ISBN i títol de les publicacions conjuntament amb tota la seva informació dels personatges:

```
db.publicacions.aggregate([
    {$project: {
        _id:0,
        ISBN:1,
        titol: 1,
        personatges:1 }}
])
```

Key	Value	Type
(1)	{ ISBN : 1, titol : "The fellowship of the ring" } (3 fields)	Object
ISBN	1	Int32
titol	The fellowship of the ring	String
personatges	Array[3]	Array
0	{ nom : "Gandalf", tipus : "mago" }	Object
1	{ nom : "Frodo", tipus : "heroe" }	Object
2	{ nom : "Samsagaz", tipus : "segundo" }	Object
(2)	{ ISBN : 2, titol : "The two towers" } (3 fields)	Object
(3)	{ ISBN : 3, titol : "The return of the King" } (3 fields)	Object
(4)	{ ISBN : 4, titol : "Harry potter y la piedra filosofal" } (3 fields)	Object
(5)	{ ISBN : 5, titol : "Harry potter y la camara secreta" } (3 fields)	Object

6 Conclusions

Durant la realització d'aquest treball en grup s'ha treballat amb una base de dades a la qual se li ha aplicat els patrons de disseny per a la seva correcta organització, per posteriorment realitzar certes querys amb el llenguatge MongoDB. Estem molt satisfets amb treball realitzat per part de tots els membres del grup i dels resultats obtinguts.