Grau en Matemàtica Computacional i Analítica de Dades

Pràctica MongoDB

Bases de Dades no Relacionals

Judit Yebra Valencia (1603614)
Pau Fuentes Hernández (1600158)
Andrea González Aguilera (1603921)
Xavier Seminario Monllaó (1603853)

Índex

1 Distribució de les tasques

Per la realització del treball, no es va establir un repartiment clar de les tasques. S'ha anat fent a estones conjuntament en trucada. Així doncs, de manera més 'esquemàtica', les tasques s'han repartit de la següent manera:

- El primer exercici, que es basa en decidir els patrons de disseny implementats per cadascuna de les relacions, es va duur a terme de manera conjunta, discutint i argumentant les diverses opinions dels membres del grup.
- El segon exercici es divideix en dues parts essencials, la modificació del document Excel per una manipulació de les dades més correcte i realitzar l'script per carregar les dades donades a la nostra base de dades. Aquest exercici va ser, de nou, duut a terme conjuntament.
- L'informe d'aquests dos exercicis es va realitzar entre l'Andrea i la Judit.
- Per últim, l'exercici 3 el van fer en Pau i en Xavi, tant la realització de les consultes com la seva redacció a l'informe.

2 Introducció

En aquest projecte es treballarà el disseny, la implementació i la consulta a una base de dades en MongoDB. A partir d'uns requisits i material relacionat amb la base de dades, s'haurà d'implementar un script en Python que processi i insereixi les dades en una base de dades de MongoDB. Posteriorment, s'haurà d'implementar en el disseny ja creat deu consultes de les quals es tenen els resultats, fent així un joc de proves.

3 Exercici 1

Apliqueu patrons de disseny per convertir el model Entitat-Relació a un conjunt de col·leccions. Considereu tant el disseny E-R com els enunciats de les consultes a fer i definides en el Joc de Proves. Expliqueu i argumenteu les decisions fetes.

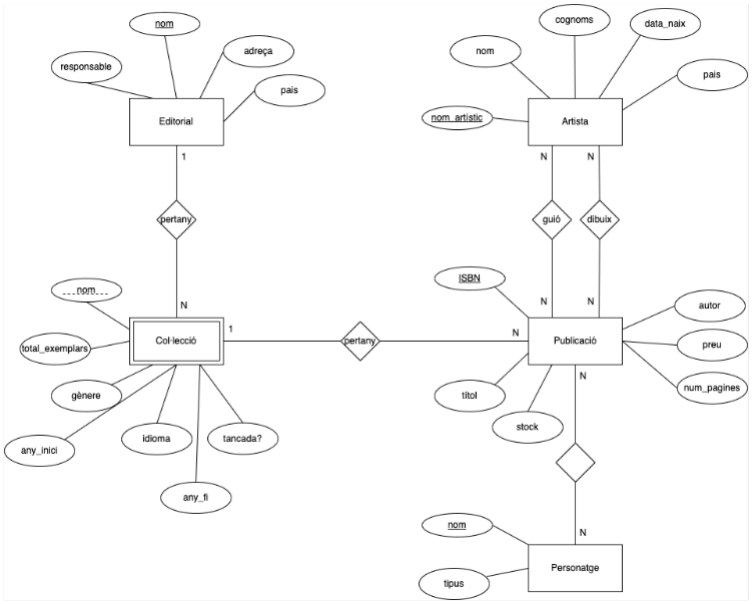


Figura 1: Model Entitat-Relació sobre el qual es treballa

L'objectiu principal d'aquest exercici és transformar el model Entitat-Relació, en un conjunt de col·leccions usant els diferents patrons de disseny apresos a classe.

En aquesta base de dades se'ns planteja un conjunt de relacions entre diverses entitats. Per determinar el patró de disseny que podem usar per cadascuna d'elles tindrem en compte el següent:

- El patró de disseny *embedded* s'utilitza per representar relacions entre entitats que són altament dependents entre si.
- En canvi, El patró de disseny de referència es fa servir usualment en relacions de cardinalitat N-1 per millorar el rendiment i l'eficiència de la base de dades.

3.1 Relació PUBLICACIÓ - PERSONATGE

En aquesta primera relació entre publicació i personatge en una botiga de còmics, la cardinalitat és N-N, cosa que significa que una publicació pot tenir diversos personatges i un personatge pot aparèixer a diverses publicacions.

Aquesta relació pot ser representada mitjançant un patró de disseny *embedded* que permeti emmagatzemar les dades de les publicacions dins dels personatges.

En aquest cas, es justifica l'ús d'un patró de disseny *embedded* que permeti emmagatzemar les dades de les publicacions dins dels personatges perquè és el costat més consultat. En una botiga de còmics, els clients solen buscar informació sobre personatges específics, com ara les seves aparicions en diferents publicacions. Emmagatzemar les dades de les publicacions dins dels personatges permet una consulta més eficient de la informació, ja que no cal fer múltiples consultes per recuperar la informació sobre les publicacions en què apareix un personatge.

A més, d'aquesta manera aconseguim reduir la complexitat de la base de dades en evitar la necessitat de fer múltiples consultes per recuperar la informació sobre les publicacions de cada personatge. En resum, l'ús d'un patró de disseny *embedded* és apropiat a causa de la naturalesa de la relació i la manera com interactuen els usuaris amb la base de dades.

3.2 Relació PUBLICACIÓ - ARTISTA

La relació entre publicació i artista en una base de dades d'una botiga de còmic és de cardinalitat N-N, cosa que significa que una publicació pot tenir diversos artistes i que un artista pot treballar en diverses publicacions. Per representar eficaçment aquesta relació a la base de dades, es pot utilitzar un patró de disseny de referència estesa.

En aquest patró de disseny, es crea una entitat addicional que actua com a unió que conté informació addicional sobre la relació entre les dues entitats principals (publicació i artista).

L'ús d'aquest patró de disseny permet més flexibilitat en la representació de la relació. A més, també permet una major eficiència en la recuperació de dades, ja que és possible fer consultes i cerques més precises i específiques utilitzant la informació addicional emmagatzemada a la taula intermèdia.

En resum, l'ús del patró de disseny de referència estesa per a la relació entre publicació i artista en una base de dades d'una botiga de còmic és apropiat a causa de la relació donada i la necessitat d'emmagatzemar-hi informació addicional.

En aquest cas, se'ns presentava una doble relació amb artista, aquestes dues feien referència als artistes guionistes i als dibuixants. Hem decidit ajuntar aquestes dues en una ja que simplement afegint un parell d'atributs a la relació era el mateix.

3.3 Relació PUBLICACIÓ - COL·LECCIÓ

En aquest cas, la relació entre publicació i col·leccions és de cardinalitat N-1, cosa que significa que una publicació pot a només una col·lecció, però una col·lecció pot contenir més d'una publicació.

Encara que es faci un baix nombre de consultes en aquesta relació, és important utilitzar el patró de disseny de referència per evitar la duplicació de dades innecessàries i reduir la mida de la base de dades. Si no s'utilitza aquest patró, cada vegada que s'insereixi una nova publicació en una col·lecció, s'hauria d'actualitzar la informació de la publicació a totes les col·leccions on apareix, cosa que comporta un consum més gran de recursos i temps.

En utilitzar el patró de disseny de referència, es crea una taula de referència que conté les claus primàries d'ambdues taules (publicació i col·lecció), cosa que permet una relació més eficient i sense duplicació de dades. A més, en fer consultes en aquesta relació, es poden obtenir els resultats de manera més ràpida i eficient, ja que la taula de referència actua com un índex per a totes dues taules.

En resum, , és important utilitzar el patró de disseny de referència per millorar l'eficiència de la base de dades i evitar la duplicació de dades innecessàries.

3.4 Relació COL·LECCIÓ - EDITORIAL

La relació entre col·leccions i editorial és de cardinalitat N-1, cosa que significa que una editorial pot publicar més d'una col·lecció, però una col·lecció només pot estar publicada per una editorial. En aquest cas, és recomanable utilitzar el patró de disseny de referència, encara que el nombre de consultes sigui baix i els atributs siguin canviants.

El patró de disseny de referència permet evitar la duplicació de dades innecessàries i millorar l'eficiència de la base de dades en relacions de cardinalitat N-1.

A més, encara que els atributs siguin canviant, és important tenir en compte que la taula de referència actua com un índex per a les dues taules, cosa que permet accedir a les dades de manera més ràpida i eficient. Si no s'utilitza el patró de disseny de referència, s'hauria d'actualitzar la informació de l'editorial a totes les col·leccions que ha publicat, cosa que comporta un consum més gran de recursos i temps.

En resum, , és recomanable utilitzar el patró de disseny de referència per millorar l'eficiència de la base de dades i evitar la duplicació de dades innecessàries.

4 Exercici 2

Implementeu un script en Python main.py -f dades.xlsx que prengui com argument el nom del fitxer Excel amb les dades

Abans d'implementar l'script en Python s'ha editat l'Excel de forma que cada finestra del document sigui una entitat amb les seves dades corresponents. El motiu pel qual això s'ha dut a terme ha estat perquè quan s'utilitzava l'Excel original hi havia camps repetits, com les editorials, per exemple, l'editorial Jupyter té diversos llibres en l'Excel i sortia repetida diverses vegades. Tot i que a les bases de dades no relacionals es permet la duplicitat de dades, no es recomanable, ja que pot donar informació errònia a l'hora de fer cerques, donat que es té la mateixa, en aquest cas, editorial com si fossin diferents.

L'script s'encarrega de carregar l'Excel com un pandas, a continuació llegir-lo, i, posteriorment carregar-lo a la base de dades de Còmics. Al carregar l'Excel com un pandas no va ser necesssari implementar l'script amb el nom del fitxer com argument.

4.1 Imports

A continuació es mostren les llibreries que s'han hagut d'importar per la gestió i creació de la base de dades.

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

4.2 Establir connexió amb el port assignat i escollir la base de dades *Còmic*

```
Host = 'dcccluster.uab.es' # localhost per connexions a la màquina main
Port = 8222
client = MongoClient("mongodb://{}:{}".format(Host,Port))

db = client['Comics']
```

4.3 Creació de del col·leccions

Per cadascuna de les col·leccions, comprovarem si existeix. Si és el cas, sobreescriurem la informació i si no és així, la crearem. Això ho podem implementar amb el codi següent:

```
if 'nom_colleccio' in db.list_collection_names():
coll = db['nom_colleccio']
coll.drop()

coll = db.create_collection('nom_colleccio')
```

4.4 Lectura del fitxer .csv

En aquest cas, com s'ha comentat anteriorment, haurem de realitzar una lectura per cadascuna de les pàgines que conté el fitxer Excel.

```
xls = pd.ExcelFile('Dades (1).xlsx')
Editorial = pd.read_excel(xls,'Editorial')
Colleccio = pd.read_excel(xls,'Colleccions')
Publicacio = pd.read_excel(xls,'Publicacions')
Personatges = pd.read_excel(xls,'Personatges')
Artistes = pd.read_excel(xls,'Artistes')
```

Com podem observar, primer realitzem una conversió del fitxer .xlsx a .xls per poder realitzar cadascuna de les lectures 'individuals' de les entitats.

4.5 De Dataframe a llista de diccionaris

Per poder manipular la informació i fer consultes, caldrà que cada entitat sigui una llista de diccionaris (JSON). Per això, realitzarem un bucle (per cada col·lecció) que el que farà serà separar la entitat en els 'atributs' del model Entitat-Relació i afegir-hi la informació corresponent.

Per comprovar que el pas anterior s'hagi realitzat de manera correcta, farem un *print* de cada entitat.
També s'ha fet una separació de totes les entrades que tenien format de llista, ja que al descarregar-les des de l'Excel el python ho interpreta tot com un string en comptes d'una llista.

4.6 Insertar els documents a la col·lecció

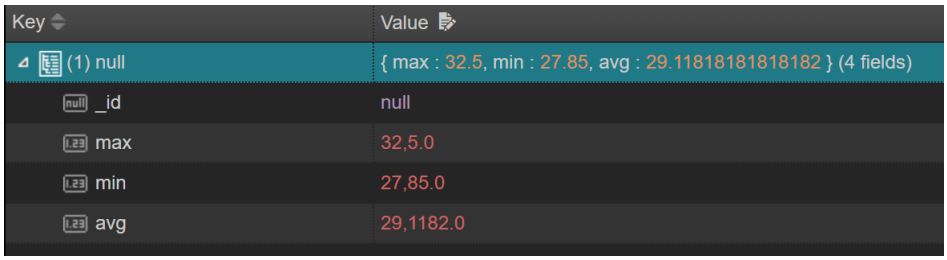
Finalment s'ha fet un *insert_ many()* dels diccionaris a la seva col·lecció corresponent de la base de dades *Comics*.

```
collection = db['Editorial']
collection.insert_many(Editorial)

collection = db['Publicacio']
collection.insert_many(Publicacio)

collection = db['Personatge']
collection.insert_many(Personatges)

collection = db['Colleccio']
collection.insert_many(Colleccio)

collection = db['Artista']
collection.insert_many(Artistes)
```

5 Exercici 3

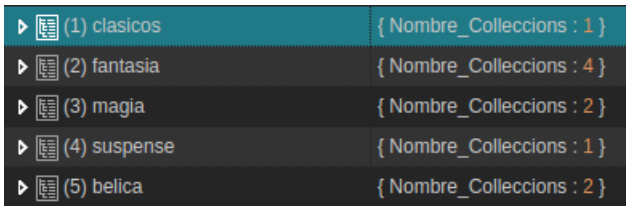
1. Les 5 publicacions amb major preu. Mostrar només el títol i el preu:
Codi: *db.Publicacio.find().sort({preu:-1}).limit(5).project({titol_publicacio:1,preu:1})*

2. Valor màxim, mínim i mitjà del preus de les publicacions de l'editorial Juniper Books:
Codi: *db.Colleccio.aggregate([{$unwind:"$ISBN_publicacions"}, {$lookup:{from:"Publicacio",localField:'ISBN',foreignField:'ISBN_Publicacions',as:"Publi"}}, {$unwind:"$Publi"}, {$match:{"Nom_editorial": "Juniper Books"}}, {$match:{$expr:{$eq:["$ISBN_publicacions",{$toString:"$Publi.ISBN"}]}}}, {$group:{_id:null,max:{$max:"$Publi.preu"},min:{$min:"$Publi.preu"},avg:{$avg:"$Publi.preu"}}}])*



3. Artistes (nom artístic) que participen en més de 5 publicacions com a dibuixant
Codi: *db.Publicacio.aggregate([{$unwind : "$id_dibuixants"}, {$group:{_id:"$id_dibuixants", Nombre_dibuixants:{$sum:1}}}, {$match:{Nombre_dibuixants:{$gt:5}}},{$project:{dibuixants:1, Nombre_dibuixants:1}}])*

4. Número de col·leccions per gènere. Mostra gènere i número total:
Codi: *db.Colleccio.aggregate([{$unwind : "$genere"},{$group:{_id:"$genere", Nombre_Colleccions:{$sum : 1}}}, {$project : {genere:1,Nombre_Colleccions:1}}])*



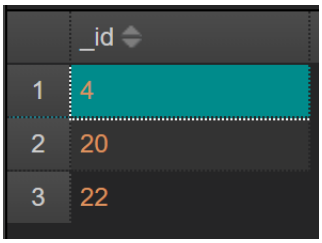
5. Per cada editorial, mostrar el recompte de col·leccions finalitzades i no finalitzades:
Codi: *db.Colleccio.aggregate([{$facet:{ Çol·leccions finalitzades": [{$match:{tancada:true}} ,{$group:{_id:"$Nom_editorial", "Recompte": {$sum:1}}}], Çol·leccions no finalitzades": [{$match:{tancada:false}},{$group:{_id:"$Nom_editorial", "Recompte": {$sum:1}}}] }}])*

6. Mostrar les 2 col·leccions ja finalitzades amb més publicacions. Mostrar editorial i nom col·lecció:
Codi: *db.Colleccio.aggregate([{$addFields:{Total_Publicacions:{$sum:{$size:"$ISBN_publicacions"}}}}, {$sort:{"Total_Publicacions":-1}}, {$match:{"tancada":true}}, {$project:{"_id":0,"Nom_editorial":1,"NomColleccio":1}}, {$limit:2}])*

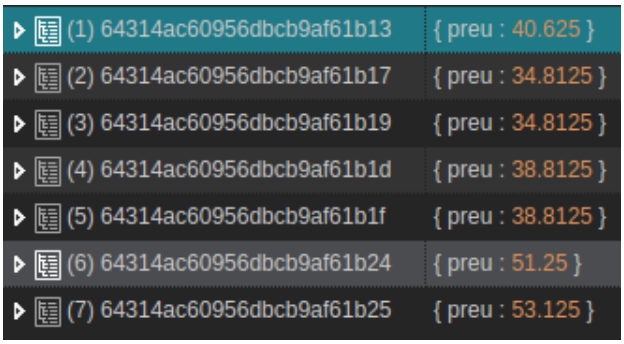
	NomColleccio	Nom_editorial
1	Harry Potter	Penguin
2	Harry Potter	Juniper Books

7. Mostrar el país d'origen de l'artista o artistes que han fet més guions:
Codi: *db.Publicacio.aggregate([{$unwind:"$id_guionistes"}, {$lookup:{from:"Artista", localField:id_guionistes", foreignField:id_artistes", as:"Artistes"}} ,{$group:{_id:"$id_guionistes", Pais:{$addToSet:"$Artistes.pais"},Nombre_guions:{$count:{}}}}, {$project:{Pais:1 , Nombre_guions:1}}, {$sort:{Nombre_guions:-1}}, {$limit:1}])*

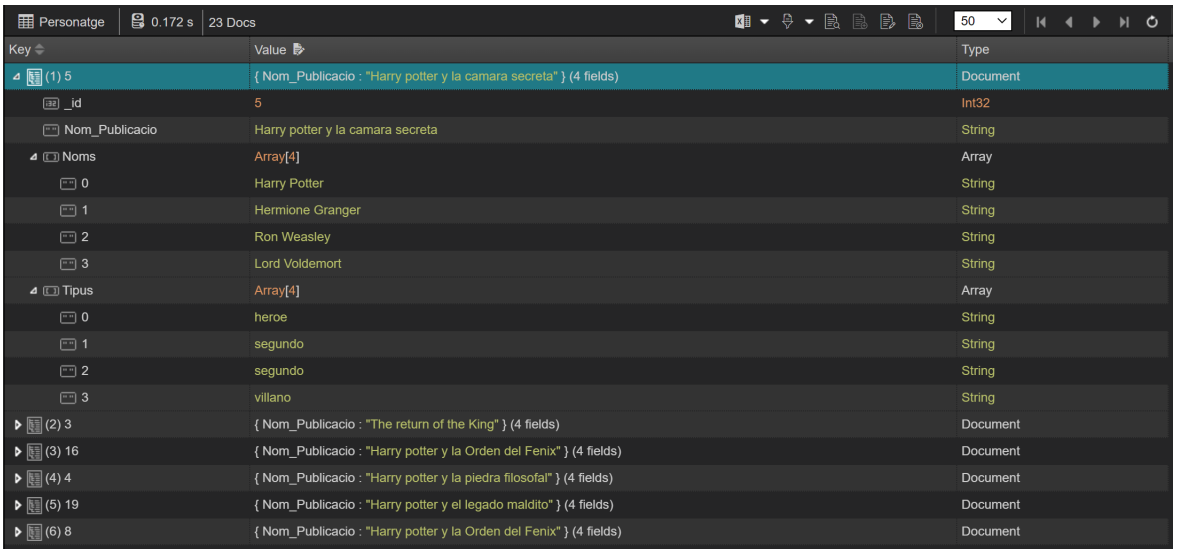
8. Mostrar el país d'origen de l'artista o artistes que han fet més guions:
Codi: *db.Personatge.aggregate([{$group:{_id:"$ISBN",Tipus:{$push:"$tipus"}}}, {$addFields:{Total_Psj:{$sum:{$size:"$Tipus"}}}}, {$unwind:"$Tipus"}, {$match:{"Tipus":"heroe"}}, {$group:{_id:"$_id",Tipus:{$push:"$Tipus"},Total:{$avg:"$Total_Psj"}}}, {$match:{$expr:{$eq:[{$sum:{$size:"$Tipus"}},{$toInt:"$Total"}]}}},{$project:{"_id":1}}])*



9. Modificar el preu de les publicacions conjuntament amb tota la seva informació dels personatges:
Codi: *db.Publicacio.updateMany({stock:{$gt:20}}, {$mul:{preu:1.25}})*



10. Mostrar ISBN i títol de les publicacions conjuntament amb tota la seva informació dels personatges:
Codi: *db.Personatge.aggregate([{$group:{_id:"$ISBN",Nom_Publicacio:{$first:"$titol_publicacio"},Noms:{$push:"$nom"},Tipus:{$push:"$tipus"}}}])*



Personatge | 0.172 s | 23 Docs — 50

Key	Value	Type
(1) 5	{ Nom_Publicacio : "Harry potter y la camara secreta" } (4 fields)	Document
_id	5	Int32
Nom_Publicacio	Harry potter y la camara secreta	String
Noms	Array[4]	Array
0	Harry Potter	String
1	Hermione Granger	String
2	Ron Weasley	String
3	Lord Voldemort	String
Tipus	Array[4]	Array
0	heroe	String
1	segundo	String
2	segundo	String
3	villano	String
(2) 3	{ Nom_Publicacio : "The return of the King" } (4 fields)	Document
(3) 16	{ Nom_Publicacio : "Harry potter y la Orden del Fenix" } (4 fields)	Document
(4) 4	{ Nom_Publicacio : "Harry potter y la piedra filosofal" } (4 fields)	Document
(5) 19	{ Nom_Publicacio : "Harry potter y el legado maldito" } (4 fields)	Document
(6) 8	{ Nom_Publicacio : "Harry potter y la Orden del Fenix" } (4 fields)	Document