

Bases de Dades no Relacionals

Grau en Enginyeria de Dades

Projecte MongoDB (dadesGICESXIX)

Informe final

Gerard Santacatalina Rubio – 1534002

Joel Soler Huix – 1531139

Introducció

En aquest document presentem la primera part del projecte de l'assignatura de BDnR, tant pel que fa al disseny de les diferents col·leccions que hem pensat per recopilar les dades que tenim al fitxer "dadesGICESXIX.xlsx" considerant sempre un disseny el més coherent possible pel que fa als patrons de disseny que hem vist a teoria com també les consultes que se'ns ha demanat resoldre amb la base de dades implementada.

Repartiment de la feina i problemes que han sorgit

Quan vam començar el projecte erem 3 persones al nostre grup tot i que als pocs dies un dels nostres companys ens va adreçar la seva impossibilitat de treballar de forma conjunta amb nosaltres per portar endavant aquest projecte.

Això va suposar que tota la feina passés a ser responsabilitat nostra i la càrrega de feina es va fer molt gran si ja de per si ens faltava un company (originalment erem 3).

No obstant això tant vam començar el projecte amb prou antelació com podem veure al nostre repositori de GitHub amb els diferents updates que hem anat fent per poder tenir-lo acabat el millor possible i que no es notés la falta de dues persones més al nostre grup que certament ens hagués ajudat molt. Així i tot amb diferents pràctiques d'altres assignatures i entregables i els exàmens parcials ha estat impossible fer-ho tot i algunes de les consultes no ens ha donat temps per poder resoldre-les tot i que creiem que la implementació dels scripts per crear la base de dades es bastant robusta i també hem tingut molta cura amb el disseny de les col·leccions.

Cal remarcar que per ser dues persones nosaltres pensem que la feina que hem fet és bastant més elevada francament que la que haguéssim fet de mitjana en cas de haver tingut dos companys més. Així doncs pensem que els nostres resultats, la nostra implementació i en general tot el nostre treball s'ha de jutjar tenint això molt en compte. Tots dos li hem dedicat un gran nombre d'hores al projecte i s'ha fet dur al llarg del seu desenvolupament.

Pel que fa al repartiment de la feina la veritat és que ens hem complementat molt bé. El disseny de les col·leccions autors, cuentos i revistes, la meitat de les consultes que s'han fet, la redacció d'aquest informe i la implementació dels scripts de les col·leccions revistes i autors així com el script "comprova_data" que controla les inconsistències de les dates a les diferents col·leccions i la comunicació que hi ha hagut per part del grup amb el professor responsable de l'assignatura (Oriol Ramos) ho ha dut a terme majoritàriament el Gerard.

La implementació de tota la resta de scripts per totes les altres col·leccions, la meitat de les consultes i per un altra banda la resta de col·leccions que presentem en aquest document i la creació del repositori GitHub ho ha fet el Joel, tot i que ens hem compartit sempre el que anavem fent i anavem fent canvis allà on no hi estàvem d'acord.

Prova d'això són les diferents actualitzacions que hem anat fent al GitHub on podem veure que tots dos hem anat actualitzant sobre la marxa el que ens semblava millorable o que no era del tot adient, comunicant-nos constantment i debatint les nostres idees. Per ser només dues persones no era del tot senzill fer una partició de la feina com segurament haguéssim fet en cas de ser el grup més gran, nosaltres ens hem anat solapant l'un amb l'altre i en tot moment hem estat en contacte per fer aquest projecte, i per aquesta raó tot i que en algunes parts un de nosaltres ha tingut un pes més rellevant al llarg del seu desenvolupament, en totes les parts d'aquest projecte sempre hi han hagut aportacions per part de l'altre.

Al llarg de les darreres setmanes la metodologia de treball ha estat bastant cooperativa i ens ficavem en trucada per Discord on també ens compartíem pantalla per explicar les diferents implementacions que havíem fet cadascún de nosaltres en les parts que ens havíem assignat i després si detectàvem algun error en la inserció dels documents en alguna col·lecció en concret doncs revisàvem amb molta cura el script encarregat de crear aquella col·lecció i controlàvem aquelles inconsistències que no havíem detectat prèviament.

El mateix vam fer amb la redacció d'aquest informe, tots dos hem revisat la presentació del treball fet en aquest document i hem estat d'acord en la seva redacció i disseny.

Afegim l'enllaç del nostre repositori a sota on podrem trobar els diferents documents que hem anat fent durant el projecte, tant els scripts Python per poder crear la base de dades i les seves diferents col·leccions, pre-processar, generar i inserir els documents i el fitxer en format javascript on implementem les consultes demanades.

Repositori GitHub: <https://github.com/JoelSolerHuix/BDnR---1531139-1534002>

1) Disseny de les col·leccions creades

En primer lloc el que vam fer va ser familiaritzar-nos una mica amb les dades sobre les que treballarem revisant tant el fitxer “dadesGICESXIX.xlsx” on tenim tots els sheets que recullen les diferents entrades com també el fitxer “JocDeProves.xlsx” on tenim els resultats de les diferents consultes que farem sobre la nostra base de dades que ens ajudaràn una mica també a trobar un disseny adient per cadascuna de les col·leccions que considerem fer per poder disposar d’una accesibilitat el més favorable possible per poder treballar correctament.

També vam pensar en tot moment en tenir sensibilitat a les redundàncies tractant d’evitar-les el màxim possible.

LLavors vam veure la necessitat de fer 6 col·leccions, tot seguit mostrarem el disseny de cadascuna d’aquestes col·leccions:

Col·lecció **autors**: on recollim tant el nom del autor que identifica el document, com un camp booleà on especifiquem si aquest autor és estranger o no i un array amb tots els contes que aquest autor ha publicat. Aquest disseny està basat en la idea del one-to-many on la cardinalitat dels contes no és exageradament elevada, sinó que era una cardinalitat més sutil i doncs vam decidir en afegir un array a la col·lecció autors on afegiem tots els contes que aquell autor concret havia publicat.

```
{
  'Autor': 'Yago, Pedro',
  'Estranger': 'False',
  'Cuentos': ['La independencia', 'Un capricho. Episodio']
}
```

Col·lecció **col·laboracions**: aquí recollim a cada document un camp on s’especifica el títol que identifica la col·laboració i després tenim un segon camp que té com a valor un array on recull tota la resta de informació com la revista on es va publicar l’obra, el tomo, número, data, autor, si va ser traduït o no aquesta obra, si va estar firmada, el seu pseudònim, les pàgines que van ser publicades, la classificació, les diferents notes i versos.

```
{
  'Titol': 'Soneto. A la ciudad de reina de Andalucia'
  'Aparicions': [{'Revista': 'El Mundo Pintoresco (1859)',
    'Tomo': 2,
    'Numero': 39,
    'Data': '25/9/1859',
    'Autor': 'Estébanez Calderón, Serafín',
    'Traducido': Null,
    'Firmat': 'El solitario',
    'Seudònim': 'El solitario',
    'Pàgines': [311],
    'Classificació': 19,
```

```

    'Notes': Null,
    'Versos': 'Casas moriscas, palios con jazmines ...']
}

```

Col·lecció **contes**: en aquesta col·lecció el que recollim són els diferents contes on tenim un camp titol que conté el títol del conte i identifica el document, un camp dedicat al seu títol alternatiu en cas que en tingui, un array temes on recollim tots els temes que tracta l'obra així com un altre array que agafa tots els seus gèneres. A més a més afegim un array publicacions que recull un seguit de documents on cadascun d'aquests especifica la revista, data, pàgines i fiabilitat de les diferents publicacions d'aquell conte. Aquí, de nou, ens vam basar en temes de la cardinalitat de les publicacions de les revistes, vam pensar en que tindria sentit ficar un array amb tots els documents referents a les diferents publicacions d'aquella revista com a camp de cadascun d'aquests documents de la col·lecció cuentos.

```

{
  'Titol': 'Zaida',
  'Titol_Alternatiu': Null,
  'Temes': Null,
  'Gèneres': ['Histórico', 'Legendario'],
  'Publicacions': [{ 'Revista': 'El Mundo Pintoresco',
                      'Data': 12/6/1859,
                      'Pàgines': [188, 190],
                      'Fiabilitat': True}]
}

```

Col·lecció **revistes**: aquí els nostres documents tenen com a camp identificatiu el propi títol de cada revista i després tenim un array que anomenem exemplars on desem subdocuments especificant cadascun dels exemplars que han existit on arrepleguem tant la data, com el número i volum de cadascuna d'aquests exemplars.

```

{
  'Títol': 'Revista Literaria De El Español',
  'Exemplars': [{ 'Data': 5/4/1847,
                  'Numero': 14,
                  'Volum': Null}]
}

```

Col·lecció **traduccions**: en aquesta col·lecció tenim com a camp identificatiu de cada document el títol de l'obra, agafem el nom del traductor al camp traductor, com està signat i el títol original de l'obra traduïda.

```
{  
  'Títol': 'Miollano. Historia veneciana',  
  'Traductor': 'Muñoz y Gaviria, José',  
  'Firmado': 'José Muñoz y Gaviria',  
  'Títol original': 'The Fiery Vault'  
}
```

Col·lecció **volums**: dissenyada per emmagatzemar documents identificats pel títol del volum i la data, on també tenim un camp dedicat a la editorial que el va produir i un array `cuentos` on tenim un seguit de subdocuments que arrepleguen els diferents contes arreplegant com a informació el títol, les pàgines del conte i la fiabilitat d'aquest.

```
{  
  'Títol_Volum': 'Cuentos y fábulas',  
  'Editorial': 'Imp. De M. Rivadeneyra',  
  'Data': '1/1/1862',  
  'Cuentos': [{ 'Cuento': 'Una mártir desconocida o la hermosura por castigo',  
                 'Paginas': '[1,16]',  
                 'Fiabilidad': True}]  
}
```

També mostrem tot seguit una col·lecció de prova que no ha estat útil realment i que potser és redundant però si en un futur vulguéssim inserir dades de diferents editorials per ciutats doncs potser seria interessant seguir el disseny que hem considerat tot i que per desar més informació sobre aquestes editorials el que passariem a tenir seria un array de subdocuments on tinguéssim aquella informació que vulguéssim tenir a la BD en diferents camps.

Col·lecció **editorials**: aquí haviem pensat en emmagatzemar documents identificats per un camp que tingués com a nom el d'una ciutat concreta, com per exemple Barcelona, i com a valor tindriem un array amb un seguit de noms de les editorials de la ciutat.

```
{  
  'Barcelona': ['Biblioteca', 'Librería de Juan y Antonio Bastinos editores',  
               'Imprenta del Diario de Barcelona']  
}
```

2) Implementació dels scripts Python

La implementació dels diferents scripts Python que creen tant la base de dades com les col·leccions i pre-processen i afegeixen les dades la podem veure amb detall al link del repositori que hem fet servir al llarg del desenvolupament del projecte.

Hem tingut un control bastant exhaustiu o almenys això és el que hem intentat sobre les dades que se'ns ha donat al fitxer excel per resoldre totes aquelles inconsistències que ens hem trobat. Algunes inconsistències petites les hem tingut en compte en els propis scripts per crear les diferents col·leccions però les inconsistències referents a les dates que eren les més importants i les que més problemes ens han donat les hem resolt a un script a part ("comprova_data.py") que implementa un codi que realment es fa servir a diferents col·leccions, codi que no és curt i per evitar redundàncies als diferents scripts doncs hem decidit separar-ho en un altre fitxer.

També ens vam adonar d'algunes inconsistències referents a alguns nombres representats en format romà. Vam afegir una petita implementació que ens convertia el nombre romà a un sencer per poder crear els objectes de tipo Date.

Referent al no duplicar o no inserir més d'un cop les dades del fitxer .xlsx que ens passen per crear les col·leccions i afegir totes les dades a la BD vam cercar a la xarxa informació sobre com fer-ho donat que no hi vam veure res d'això a classe.

Després de fer les nostres proves vam pensar en crear indexos únics sobre aquells camps que identificaven als documents a dintre de cada col·lecció de forma que si executavem el main.py no els afegia duplicats perquè amb el índex podem controlar el que el document que intentavem afegir estigués duplicat i no el fiqués a la col·lecció. Ara per exemple si esborrem tots els autors que tenim al fitxer excel des d'on carreguem les dades i ens fiquem a nosaltres mateixos com únic autor i tornem a executar el main.py veurem que a la col·lecció tindrem els mateixos autors que abans +1, això és important perquè ens alliberem de la dependència de tenir a tots els autors al sheet del excel.

En un primer moment nosaltres el que feiem era comprovar el nombre de documents que tenia una col·lecció en concret i si aquest nombre era igual o major que 1 doncs el-eliminavem tots els continguts d'aquella col·lecció concreta i tornavem a carregar totes les dades que teniem al excel.

Clarament això no és una sol·lució manegable per a bases de dades més serioses perquè el nombre de documents a el·liminar i carregar cada cop que vulguem actualitzar la BD doncs seria realment elevat i dispararia la complexitat de la nostra implementació i sobretot en termes d'espai. Si haguéssim d'actualitzar la base de dades en un moment concret hauriem de tenir tots els documents que aquella base de dades contingués al moment d'afegir-ne de nous per no perdre informació. Una dependència important i dolenta que ens hem tret de sobre amb la creació d'indexos únics que fa que tinguem un control de no inserir els documents més d'un cop i que ens permet no haver de mantenir informació ja inserida a la base de dades en diferents fitxers.

Finalment remarcar que hem tingut en compte el que aquest codi està pensat per ser executat per terminal i hem implementat el necessari per poder executar les comandes: **python main.py -f dadesGICESXIX.xlsx** i **python main.py --delete_all --bd dadesGICES** des de la terminal.

3) Implementació de les consultes

Consulta 1: Contes, no anònims, d'autors estrangers.

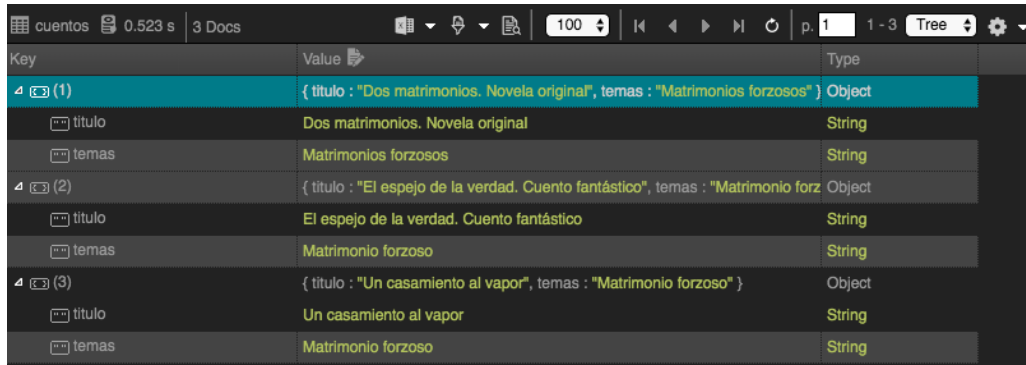
```
db.autors.aggregate([
  {$match: {Extranjero: true, Autor: {$not:{$in:['Anonimo', 'Anonimo -
traducciones']}}}},
  {$unwind:"$Contes"},
  {$project: {Contes: 1, Autor: 1, _id: 0}},
  {$sort: {Autor: 1}}])
```

Resultat: 62 documents. Ens dona el mateix resultat que al professor.

Consulta 2: En quins contes hi ha: “matrimonio(s) forzoso(s)”? (són preguntes sobre el llistat “Temas, tipus y motivos”. Mostrar nom del conte i tema.

```
db.cuentos.aggregate([
  {$unwind:"$temas"},
  {$match:{$or: [{temas: 'Matrimonio forzoso'}, {temas: 'Matrimonios forzosos'}]}},
  {$project:{_id:0, temas:1, titulo:1}}])
```

Resultat: 3 documents. També ens dona el mateix resultat que al professor.
Tot seguit mostrem les sortides donat que el nombre de documents és petit.



Key	Value	Type
(1)	{ titulo : "Dos matrimonios. Novela original", temas : "Matrimonios forzosos" }	Object
titulo	Dos matrimonios. Novela original	String
temas	Matrimonios forzosos	String
(2)	{ titulo : "El espejo de la verdad. Cuento fantástico", temas : "Matrimonio forzoso" }	Object
titulo	El espejo de la verdad. Cuento fantástico	String
temas	Matrimonio forzoso	String
(3)	{ titulo : "Un casamiento al vapor", temas : "Matrimonio forzoso" }	Object
titulo	Un casamiento al vapor	String
temas	Matrimonio forzoso	String

Consulta 3: Títol dels contes, revista, data i pàgines que va publicar Sánchez Viedma, José? (en cas que el conte es publiqués per parts, indicar la data i pàgines de la publicació de la 1a part).

Aquí en comptes de fer un \$lookup dintre del aggregate vam voler provar fent servir una variable on emmagatzemavem el resultat de fer un find i passar-lo com a array i fent servir després als camps projectats l'operador \$slice que vam veure al doc de MongoDB, per variar una mica.

```
var x = db.autors.find({$or: [{Autor: 'Sánchez Viedma, José'}, {Autor: 'Sanchez Biedma, Jose'}]}).toArray()
```



```
db.cuentos.aggregate([
  {$match:{$or:{{titulo:x[0]["Contes"][0]}, {titulo: x[0]["Contes"][1]}}}},
  {$project: {'titulo': 1, revista:{$slice: ['$publicaciones.revista', 1]},
data:{$slice:['$publicaciones.data',1]}, pagines:{$slice:['$publicaciones.pagines',1]},
'_id':0}}])
```

Resultat: 2 documents. El mateix que als resultats proporcionats pel professor.

Mostrem la sortida:

Key	Value	Type
(1)	{ titulo : "Aventuras de una silla contadas por ella misma" } (4 fields)	Object
titulo	Aventuras de una silla contadas por ella misma	String
revista	["El Museo Universal"]	Array
data	[ISODate("1868-10-04T09:45:16.000-00:15")]	Array
pagines	[[318, 310]]	Array
(2)	{ titulo : "La hermana del quinto" } (4 fields)	Object
titulo	La hermana del quinto	String
revista	["El Museo Universal"]	Array
data	[ISODate("1869-08-29T09:45:16.000-00:15")]	Array
pagines	[[275, 277]]	Array

Consulta 4: Quin autor (no anònim) ha publicat més contes?

```
db.autors.aggregate([
  {$match:{Autor:{$not:{$in:['Anonimo', 'Anonimo - traducciones']}}}},
  {$addFields:{ "Cont":{$size:"$Contes"} }},
  {$sort:{Cont:-1}},
  {$limit:1},
  {$unwind:"$Autor"},
  {$project:{Autor:"$Autor", '_id':0}}
])
```

Resultat: 1 (Fernandez Iturralde, Enrique).

Ens torna a donar el mateix que al professor.

Key	Value	Type
(1)	{ Autor : "Fernandez Iturralde, Enrique" }	Object
Autor	Fernandez Iturralde, Enrique	String

Consulta 6: Quants contes hi ha entre de 1840 fins 1850 ?

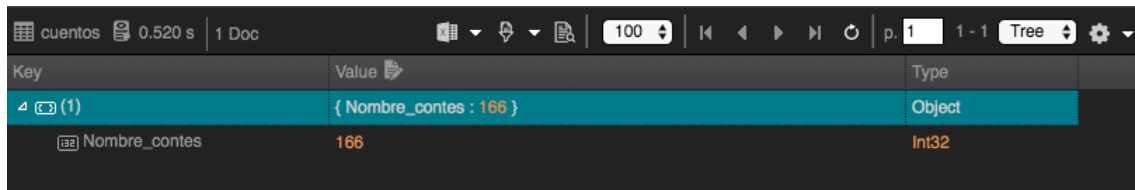
Fent servir el find:

```
db.cuentos.find({"publicaciones.data":{"$gte":ISODate('1840-01-01'),  
'$lte':ISODate('1850-01-01')}}).count()
```

Amb un aggregate:

```
db.cuentos.aggregate([  
  {$match:{"publicaciones.data":{"$gte":ISODate('1840-01-01'), '$lte':ISODate('1850-  
01-01')}}},  
  {$count:"Nombre_contes"}])
```

Resultat: 1 (Nombre_contes: 166). També ens dona el mateix resultat que al professor.



Key	Value	Type
(1)	{ Nombre_contes : 166 }	Object
Nombre_contes	166	Int32