Cas d’ús MongoDB

**Treball en equip**

En el següent informe es presenta un cas pràctic de l’ús de MongoDB. A continuació, s’exposa la repartició de tasques entre els components de l’equip:

1. Tasca 1: Lucía
2. Tasca 2: Aránzazu i Jan
3. Tasca 3: Cristian

L’evolució del treball es pot visualitzar al nostre gitHub: [BDnR\_ProjecteMongoDB](https://github.com/luciarevaliente/BDnR_ProjecteMongoDB.git).

# **Tasca 1**

En el primer exercici, hem aplicat patrons de disseny (incrustació, referència i referència extesa) per convertir el model Entitat-Relació proporcionat als requeriments de dades a un conjunt de col·leccions. Al llarg d’aquest apartat, explicarem i argumentarem les decisions fetes.

En primer lloc, hem analitzat les entitats i relacions de la base de dades. Cada entitat l’hem associat a una col·lecció i observant les relacions, les hem mapejat. En altres paraules, hem realitzat canvis fins a obtenir la següent base de dades.

La primera col·lecció que hem creat ha sigut **Clients**. Cada instància de l’entitat client es representa com un document en la col·lecció. Per tant, cada atribut de l’entitat es converteix en un camp dins del document. No només això, hem realitzat una incrustació, és a dir, hem encastat en aquesta l’entitat Cotxe. Cada instància de cotxe és un element en la llista de l’atribut “cotxes” del Client. Un dels principals motius és perquè en les consultes sempre s’accedeix de forma conjunta, es demanen les dades del client i del cotxe. Tot i que les dades dels vehicles no estan altament relacionades amb les persones, el nostre objectiu és reduir les operacions join. Aquestes poden afectar al rendiment de la base de dades si es realitzen unions complexes. D’altra banda, augmenten la complexitat de les consultes, afectant al manteniment del codi.

La segona col·lecció és **Productes**. Cada instància de l’entitat producte es representa com un document en la col·lecció. D’igual forma, cada atribut de l’entitat es converteix en un camp dins del document. La relació “pack” del diagrama E-R està representada amb el camp “productes\_inclosos”, que és una llista de productes. Si el producte principal està format per un conjunt de productes, aquests s’emmagatzemen de forma individual en la llista mencionada. Així doncs, el número de subproductes equival a la longitud del camp “productes\_inclosos”.

La tercera col·lecció és **Tiquets**. Cada instància de l’entitat tiquet es representa com un document en la col·lecció. D’igual forma, cada atribut de l’entitat es converteix en un camp dins del document. En aquesta col·lecció, hem referenciat l’id del client per saber qui realitza la compra. També hem fet una referència extesa als productes que es compren. Hem escollit aquest patró perquè necessitem saber dades del producte, com el preu per unitat, entre d’altres. També perquè hem afegit atributs nous, com la quantitat del producte afegit a la cistella.

La quarta col·lecció és **Estades**. Cada instància de l’entitat estada es representa com un document en la col·lecció. D’igual forma, cada atribut de l’entitat es converteix en un camp dins del document. És important destacar que aquesta entitat és feble, per tant, depèn de Cotxe i Parking. Com en les consultes es demana la informació de les estades juntament amb la del parking, hem incrustat les places amb la informació a la col·lecció estades. A més, per poder donar informació sobre els vehicles i del conductor, hem referenciat l’id del cotxe i del client. Gràcies a fer aquesta encastació, estalviem espai en la Base de Dades perquè no hem de crear dues col·leccions i estalviem temps al no fer el join de Estada-Parking.

Cadascuna de les col·leccions plantejades tenen un document de referència a la carpeta “ex1/” del gitHub. En aquesta hi ha quatre fitxers.txt amb documents model. En el següent exercici els transformarem a fitxers JSON.

# **Tasca 2**

Un cop vam dissenyar la base de dades exposada a la Tasca 1, vam realitzar els fitxers per tal de crear-la. A partir del disseny de la base de dades afegim els documents de cada col·lecció. Nosaltres hem triat afegir les dades a partir de la creació dels fitxers json de la seva corresponent col·lecció. Vam triar aquesta manera d’implementar-ho ja que a la Tasca 1 el format que vam seguir va ser agafant d’exemple el json. A més, com només hem creat entre 5 i 10 documents per col·lecció vam trobar més senzill implementar en json, ja que al ser tan poques dades, ens seria molt més fàcil operar amb aquest tipus d’estructura de dades (ja indexada, amb claus i valors…). Si més endavant volguéssim ampliar la nostra base de dades podriem considerar fer-ho amb Excel.

Seguidament vam desenvolupar el codi del fitxer main.py. En aquest s’estableix la connexió al cluster de la UAB amb el nostre port corresponent, el 8209. Un cop establerta la connexió creem la nostra base de dades anomenada ‘aplicacio’ i les col·leccions (Clients, Productes, Tiquets, Estades). Amb la connexió establerta vam afegir les dades a partir dels fitxers json.

Quan vam intentar afegir els documents a partir les json vam haver de solucionar alguns errors de sintaxi ja que ens havíem oblidat algunes comes. També, a l’hora d’afegir-les vam controlar els duplicats amb try-except per si la connexió s’interrompia o es trobava a executar el fitxer.

# **Tasca 3**

Després d’implementar un script Python per inserir els fitxers JSON a la Base de Dades no Relacional, vam realitzar les consultes demanades als requeriments de dades. A continuació s’exposen aquestes, juntament amb el resultats.

La primera consulta és: “Nombre de clients atesos durant un mes/any en concret amb tiquet superior a 500€”. La comanda generada és:

*db.tiquets.aggregate([*

*{*

*$match: {total: {$gte: 500}}*

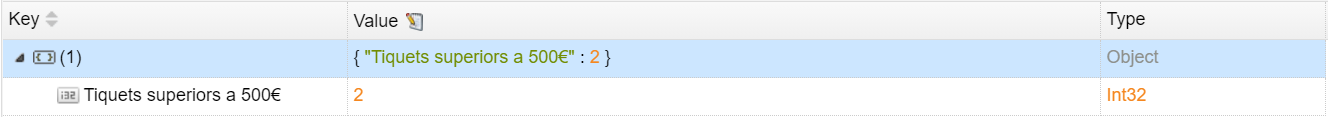
*}, {*

*$count: "Tiquets superiors a 500€"*

*}*

*])*

El resultat que obtenim és:



La segona consulta és: “Mostra la informació del client de Barcelona amb major edat”. La comanda generada és:

*db.clients.aggregate([*

*{*

*$match: {poblacio: "Barcelona"}*

*},{*

*$sort: {edat: -1}*

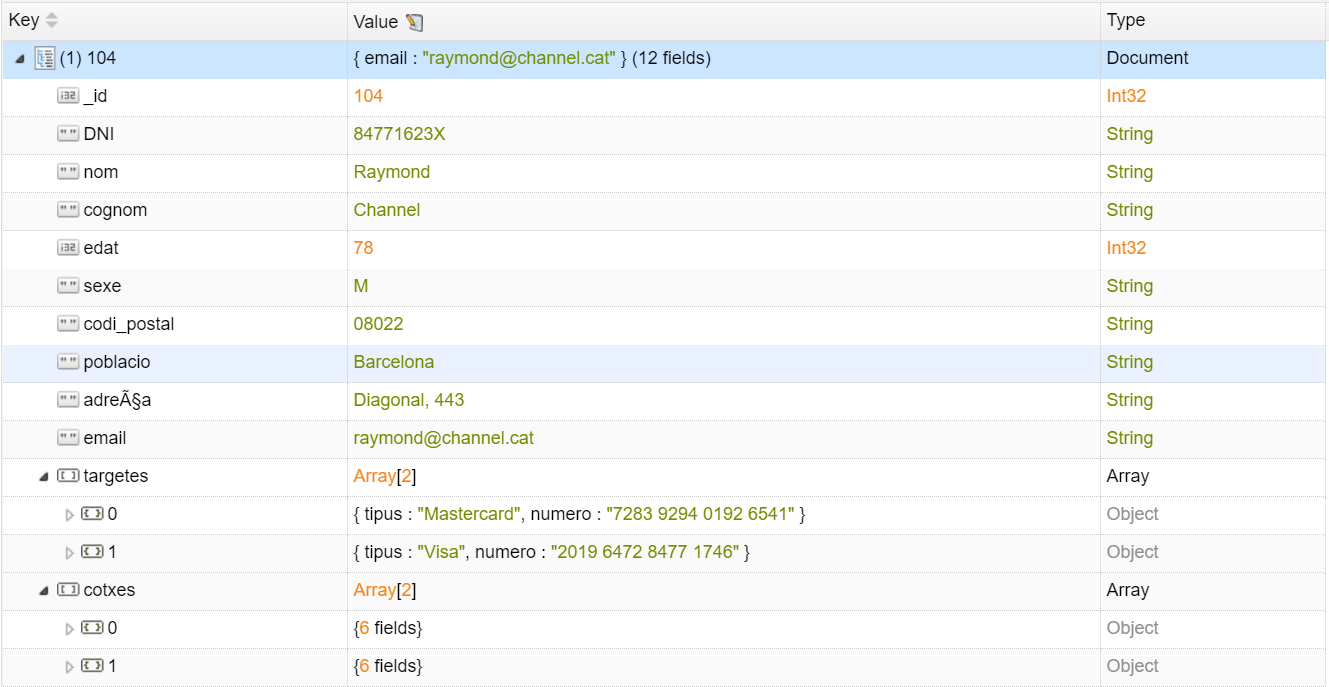
*},{*

*$limit: 1*

*}*

*])*

El resultat que obtenim és:



La tercera consulta és: “Valor màxim, mínim i mitjà dels preus dels productes”. La comanda generada és:

*db.productes.aggregate([*

*{*

*$group: {*

*\_id: null,*

*maxim: {$max: "$preu"},*

*minim: {$min: "$preu"},*

*mitjana: {$avg: "$preu"}*

*}*

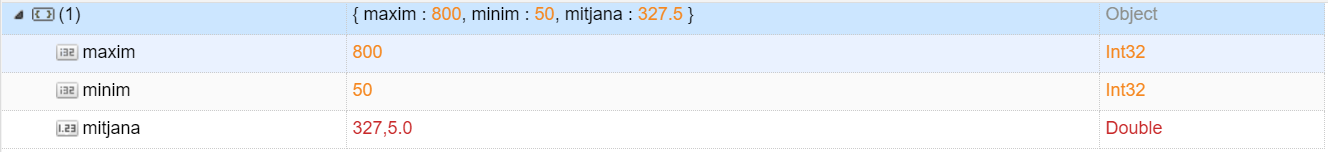
*},{*

*$project: {\_id: 0}*

*}*

*])*

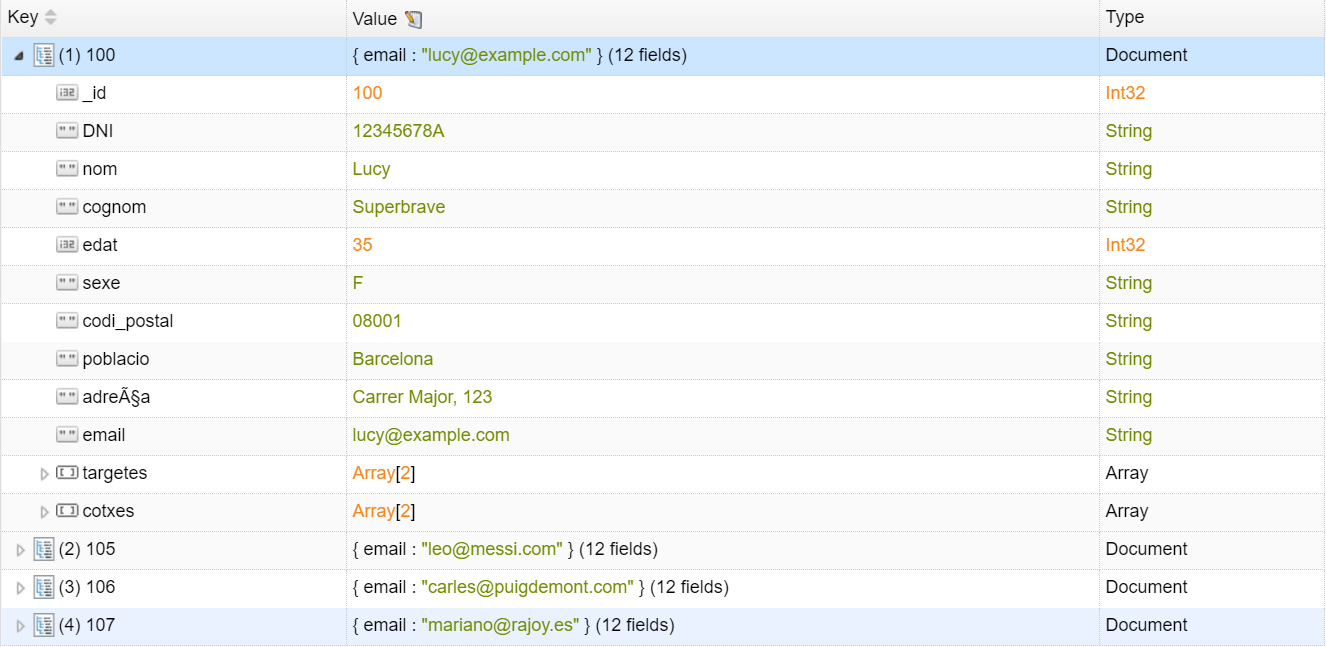
El resultat que obtenim és:



La quarta consulta és: “Mostra 5 clients que mai venen en cotxe a la tenda”. La comanda generada és:

*db.clients.find({\_id: {$nin: db.estades.distinct("id\_client")}}).limit(5)*

El resultat que obtenim és:



La cinquena consulta és: “Foto actual del pàrquing (o en una data concreta). Mostrar plaça, dades del vehicle estacionat i el nom i cognom del client”. La comanda generada és:

*// Fem servir la data 2024-03-30 a l'hora 10:15:00*

*db.estades.aggregate([*

*{*

*$match: {*

*data\_i\_hora\_entrada: {$lte: ISODate("2024-03-30T10:15:00Z")},*

*data\_i\_hora\_sortida: {$gte: ISODate("2024-03-30T10:15:00Z")},*

*}*

*},{*

*$lookup: {*

*from: "clients",*

*localField: "id\_client",*

*foreignField: "\_id",*

*as: "client"*

*}*

*},{*

*$unwind: "$client"*

*},{*

*$project: {*

*\_id: 0,*

*plaça: {*

*$concat: [*

*"Planta: ", "$planta", ". Zona: ", "$zona", ". Numero: ", "$numero"*

*]*

*},*

*nom\_client: "$client.nom",*

*cognom\_client: "$client.cognom",*

*dades\_vehicle: {*

*$arrayElemAt: [*

*{*

*$filter: {*

*input: "$clien.cotxes",*

*as: "coche",*

*cond: { $eq: ["$$coche.matricula", "$matricula\_cotxe"]}*

*}*

*}, 0]*

*}*

*}*

*}*

*])*

El resultat que obtenim és:



La sisena consulta és: “Productes formats per més de 4 subproductes (pack)”. La comanda generada és:

*db.productes.find({$expr: {$gt: [{$size: "$productes\_inclosos"}, 4]}})*

El resultat que obtenim és:



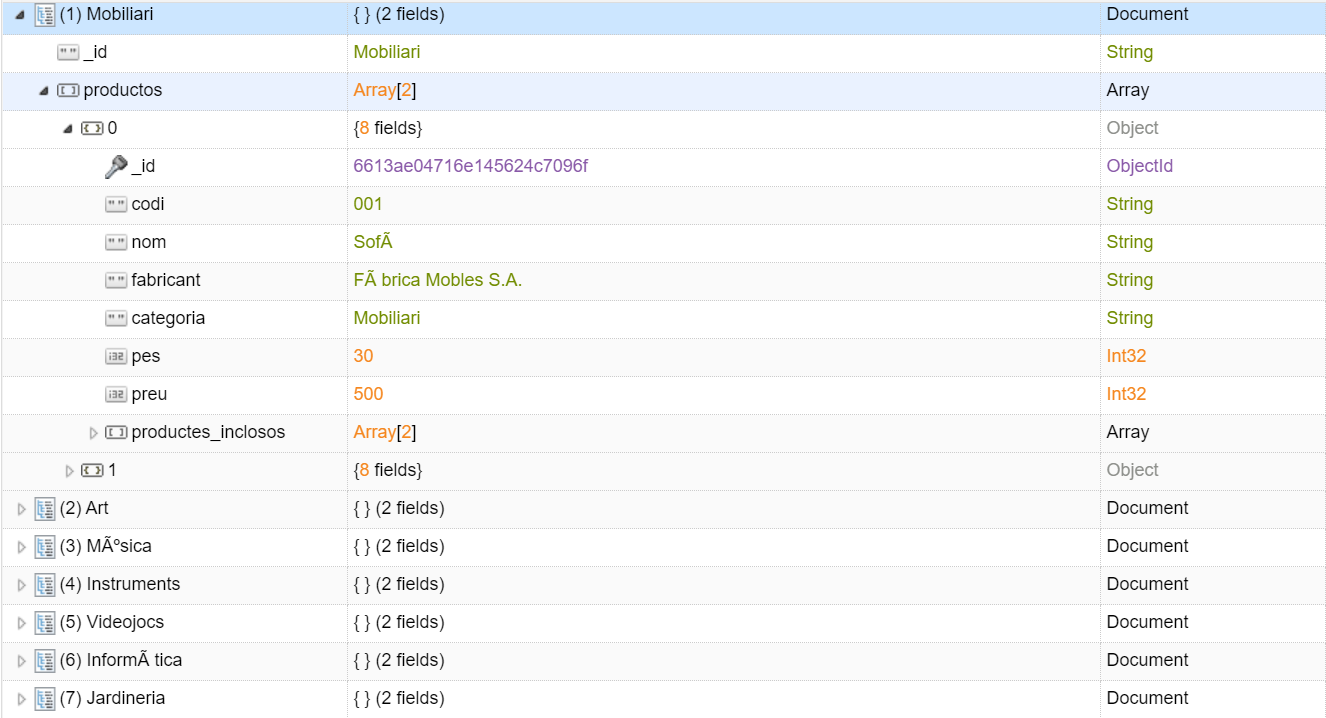
La setena consulta és: “Llistat de productes agrupats per categoria”. La comanda generada és:

*db.productes.aggregate([*

*{$group: {\_id: "$categoria", productos: {$push: "$$ROOT"}}}*

*])*

El resultat que obtenim és:



La vuitena consulta és: “Nombre de tiquets segons el tipus de pagament”. La comanda generada és:

*db.tiquets.aggregate([*

*{$unwind: "$pagament"},*

*{$sortByCount: "$pagament"}*

*])*

El resultat que obtenim és:



La novena consulta és: “Mostrar el nom i cognom dels clients que han comprat un producte en concret (de la vostra elecció) en algun dels seus tiquets (lookup)”. La comanda generada és:

*// Fem servir el codi de producte "001"*

*db.tiquets.aggregate([*

*{*

*$unwind: "$productes"*

*},{*

*$match: {"productes.codi": "001"}*

*}, {*

*$lookup: {*

*from: "clients",*

*localField: "id\_client",*

*foreignField: "\_id",*

*as: "clients"*

*}*

*},{*

*$unwind: "$clients"*

*},{*

*$project: {*

*\_id: 0,*

*producte: "001",*

*nom: "$clients.nom",*

*cognom: "$clients.cognom"*

*}*

*}*

*])*

El resultat que obtenim és:

