

Dia 2: Treball amb MongoDB.

Update, delete i
Aggregation Framework

Xavier Sala Pujolar

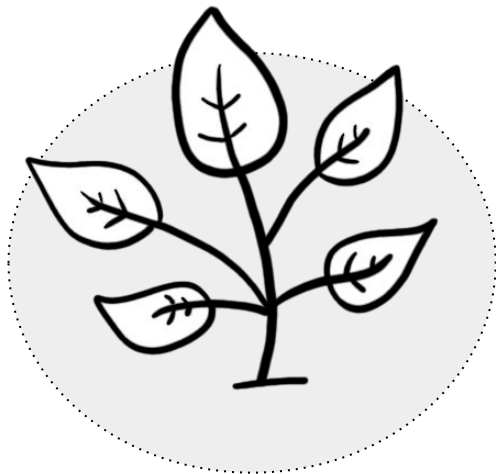


MongoDB

{
Universitat
de Girona
}

Abril 2021

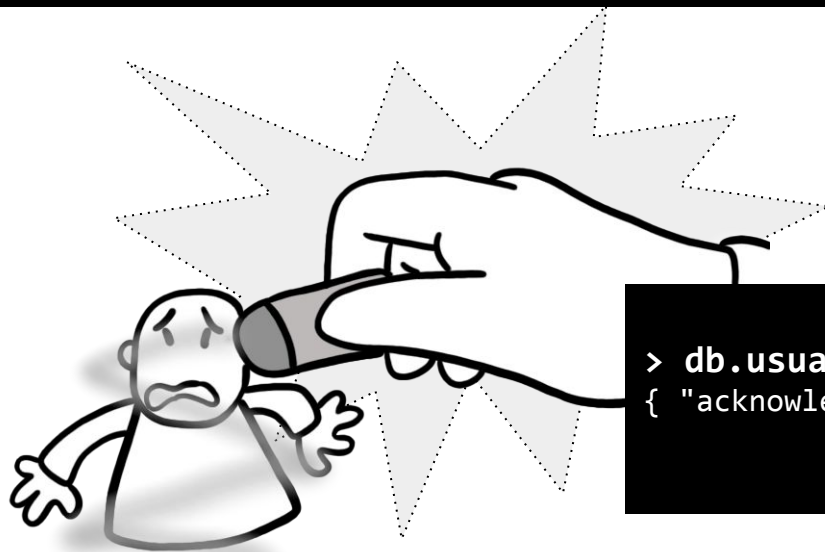
Operacions U i D



Els programes necessiten més coses a part de recerques i insercions.

Les operacions d'actualització i esborrat són una part fonamental en sistemes de base de dades

Esborrat

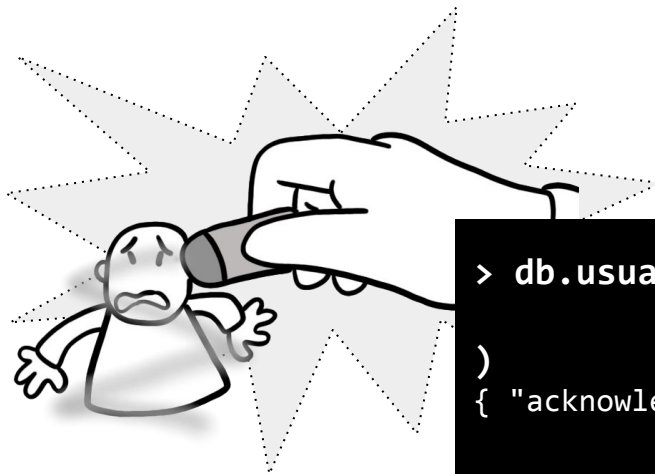


Esborrar només
requereix un
filtre de cerca



```
> db.usuaris.deleteMany({})  
{ "acknowledged":true, "deletedCount": 15824 }
```

```
DELETE * FROM usuaris
```



S'especifica la
condició en un
document



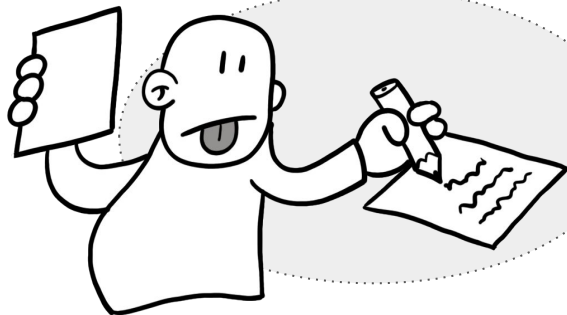
```
> db.usuaris.deleteOne(  
    {"nom":"xavier"}  
)  
{ "acknowledged":true, "deletedCount":1 }
```

```
DELETE FROM usuaris  
WHERE nom="xavier"
```

Actualització

L'actualització
rep dos documents
com a paràmetres

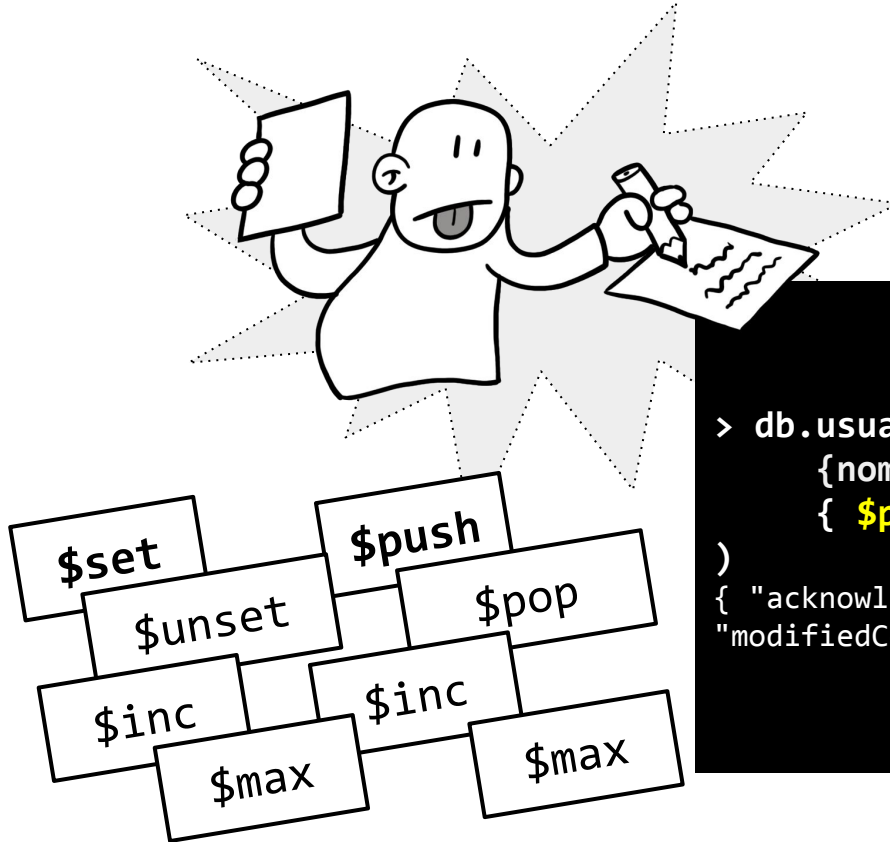
Operadors de canvi



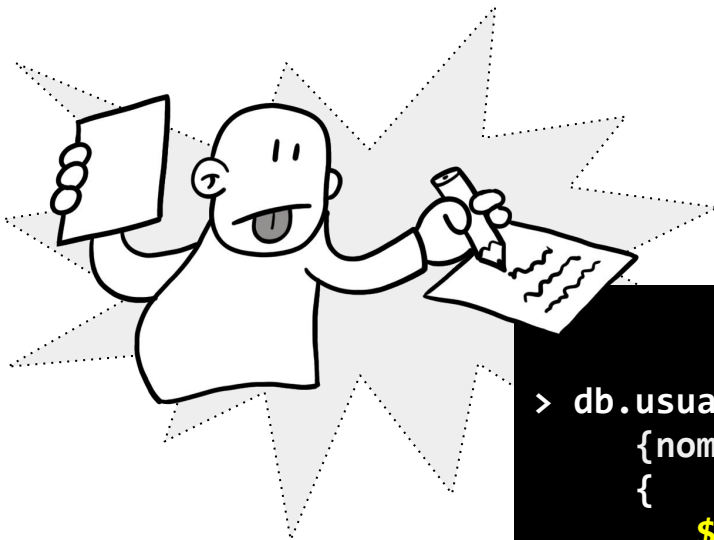
```
> db.usuaris.updateOne(  
  {nom:"Pere"},  
  {$set: nom:"Joan"}  
)  
{ "acknowledged" : true, "matchedCount" : 1,  
  "modifiedCount" : 1 }
```

Criteri de
cerca

Hi ha operacions
específiques per taules



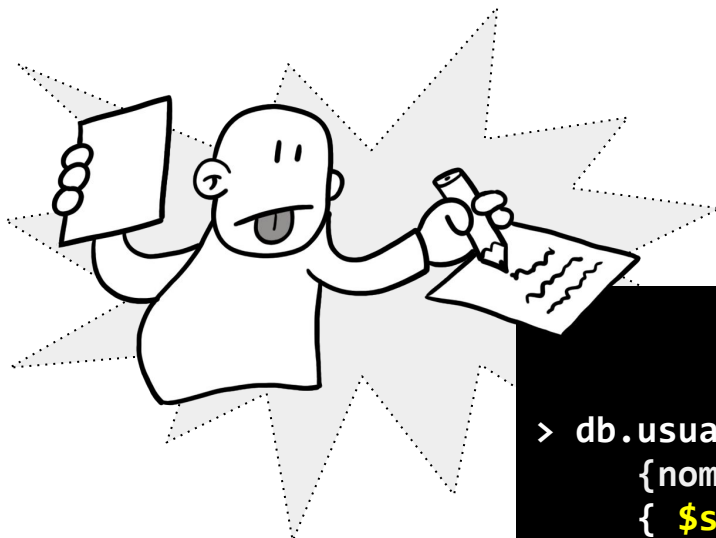
```
> db.usuaris.updateOne(  
  {nom:"Pere"},  
  { $push: {telefons:"232323"} }  
)  
{ "acknowledged" : true, "matchedCount" : 1,  
  "modifiedCount" : 1 }
```



Es poden fer servir
múltiples operadors en
una sola operació



```
> db.usuaris.updateOne(  
  {nom:"Pere"},  
  {  
    $push: {telefons:"232323"},  
    $set: { nom: "Pau" }  
  }  
)
```

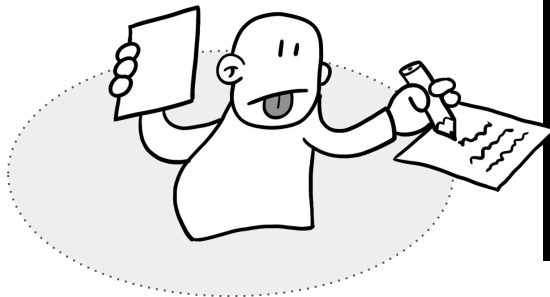


Actualitza tots
els resultats



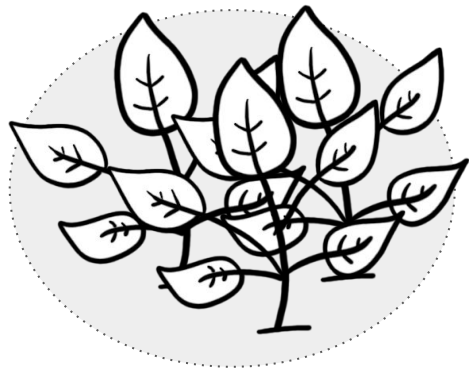
```
> db.usuaris.updateMany(  
  {nom:"Pere"},  
  { $set: {nota: 10 } }  
)  
{ "acknowledged" : true, "matchedCount" : 2,  
  "modifiedCount" : 2 }
```


`replace` canvia
completament el
document



```
> db.usuaris.replaceOne(  
  { nom:"Pere" },  
  { nom:"Pere", cognoms:"Pi"},  
)  
{ "acknowledged" : true, "matchedCount" : 1,  
  "modifiedCount" : 1 }
```

Aggregation Framework



De la mateixa forma que es fa en les bases de dades relacionals, en Mongo també es poden fer operacions estadístiques.

L'objectiu del Aggregation Framework és obtenir resultats calculats a partir d'un grup de resultats.

Comptar



Condicció



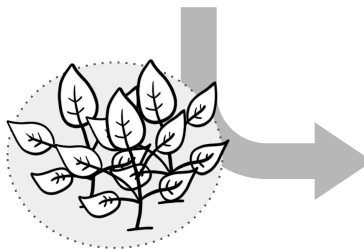
```
> db.usuaris.count({nom:"Pere"})
```

```
1
```

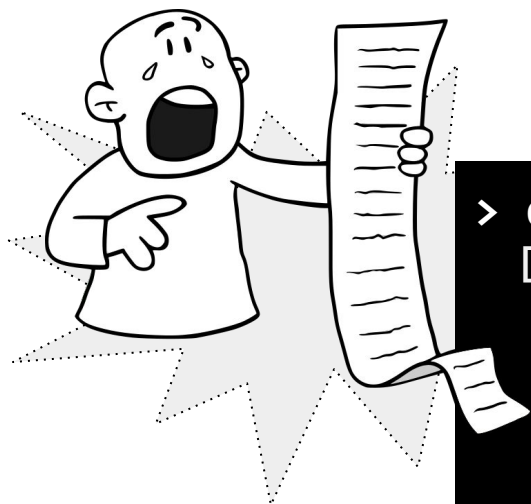
Aggregate

Agreggration
framework permet
fer coses més
complexes

```
{ "_id" : ObjectId("54ba7c122"),  
  "nom" : "globus",  
  "color": "verd" }  
{ "_id" : ObjectId("54ba7c123"),  
  "nom" : "llapis",  
  "color": "verd" }  
{ "_id" : ObjectId("54ba7c124"),  
  "nom" : "cotxe",  
  "color": "vermell" }
```



```
{ "color" : "verd",  
  "quantitat": 2 }  
{ "color": "vermell",  
  "quantitat": 1 }
```



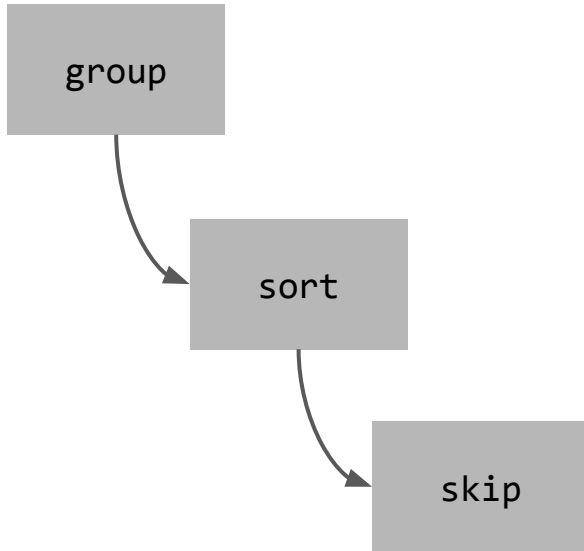
Una llista
d'operacions

Es fa servir la
operació **aggregate**



```
> db.usuaris.aggregate(  
  [  
    { $group:{  
      "_id":"$color",  
      "quantitat" : {$sum: 1}  
    }  
  },  
  { $sort: { "quantitat": 1 } },  
  { $skip: 2 }  
  ]  
)
```

Les operations es
fan en l'ordre en
què es defineix en
l'array



```
> db.usuaris.aggregate(  
  [  
    { $group:{  
      "_id":"$color",  
      "quantitat" : {$sum: 1}  
    }  
    },  
    { $sort: { "quantitat": 1 } },  
    { $skip: 2 }  
  ]  
)
```

operacions d'agreggate

`$match`

`$project`

`$group`

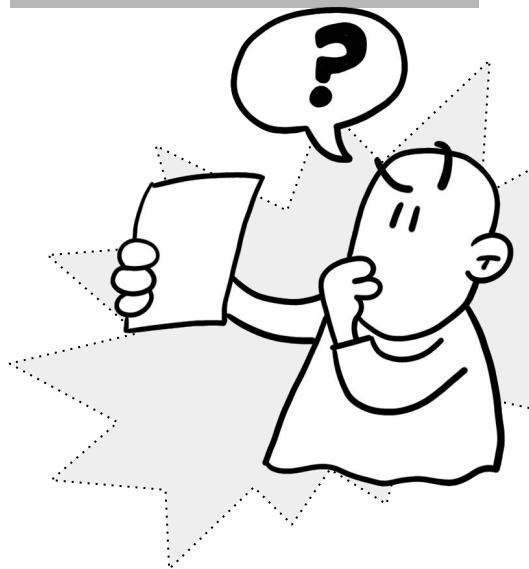
`$sort`

`$limit`

`$skip`

match

`$match` filtra els resultats que rep



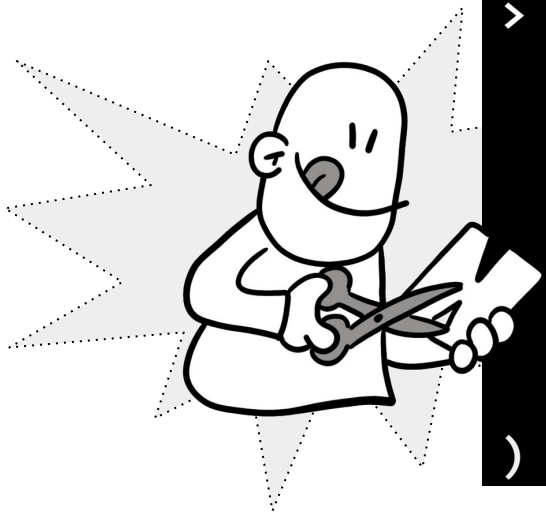
Se li posa una condició com en el `find()`

```
> db.usuaris.aggregate(  
  [  
    { $match:{  
      "nom":"Pere",  
      "edat" : {$gte: 18}  
    }  
  }  
]  
)
```


project

La operació `$project`
pot generar nous
documents a partir
del que rep

S'hi defineix la
nova estructura



```
> db.usuaris.aggregate(  
  [  
    { $project:{  
      "_id": 0,  
      "nom" : 1,  
      "àlies": "$nom"}  
    }  
  ]  
)
```

```
{ _id:1, nom:"Pere", cognom:"Pi", edat: 23 }  
{ _id:2, nom:"Pau", cognom:"Pujol", edat: 31 }  
{ _id:3, nom:"lu", cognom:"Garcia", edat: 12 }  
{ _id:4, nom:"Bernat", cognom:"Puig", edat: 54 }  
{ _id:5, nom:"Joel", cognom:"Boix", edat: 8 }
```

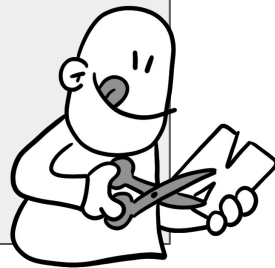
Deixa "nom" tal
com està

Posa el contingut
de "nom" a àlies

```
$project:{ "_id": 0, "nom" : 1, "àlies": "$nom" }
```

Si no es posa hi és
sempre

```
{ nom:"Pere", àlies:"Pere" }  
{ nom:"Pau", àlies:"Pau" }  
{ nom:"lu", àlies:"lu" }  
{ nom:"Bernat", àlies:"Bernat" }  
{ nom:"Joel", àlies:"Joel" }
```

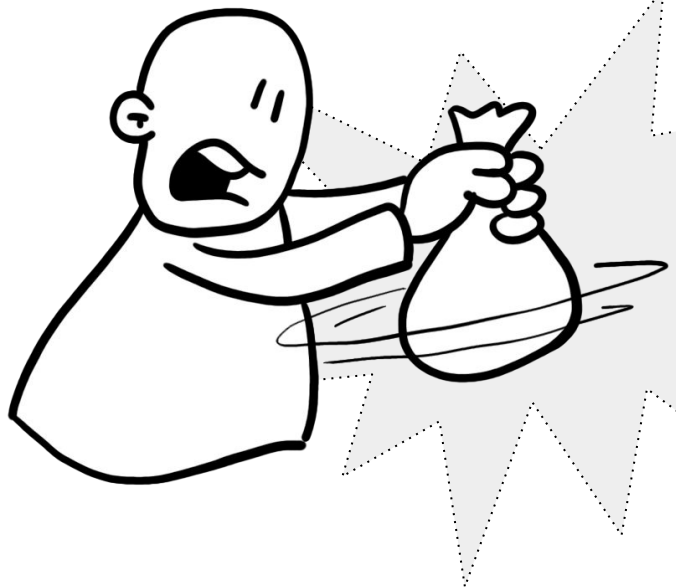


group

Per agrupar valors i fer-hi alguna operació es fa servir \$group

Al camp `_id` s'hi posa el camp d'agrupació

Els altres camps han de ser operadors



```
> db.usuaris.aggregate(  
  [  
    { $group: {  
      _id: "$nom",  
      quants: {$sum: 1 }  
    }  
  ]  
)
```

```
{_id:1, nom:"Pere", cognom:"Pi", edat: 23 }  
{_id:2, nom:"Pau", cognom:"Pujol", edat: 31 }  
{_id:3, nom:"Pere", cognom:"Garcia", edat: 12 }  
{_id:4, nom:"Bernat", cognom:"Puig", edat: 54 }  
{_id:5, nom:"Pere", cognom:"Boix", edat: 8 }
```

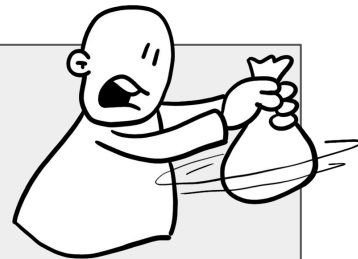
Suma 1 per cada
document

Només operadors

```
$group:{ "_id": "$nom", "quants" : { $sum: 1 } }
```

Genera un nou Id a
partir del nom

```
[  
  {_id: 'Pau', quants: 1 },  
  {_id: 'Bernat', quants: 1 },  
  {_id: 'Pere', quants: 3 }  
]
```



unwind

Permet generar documents a partir de les dades de les taules

```
{ _id: 234, nom: "Pere",  
  telefons: [ "505050", "404040" ]  
}
```

\$unwind: "\$telefons"



```
{ _id: 234, nom: "Pere",  
  telefons: "505050" }
```

```
{ _id: 234, nom: "Pere",  
  telefons: "404040" }
```

Altres

`$set` afegeix un
camp nou "guanyador"

El resultat s'emmagatzema
amb `$out` en una col·lecció
que es diu "resultats"

```
> db.usuaris.aggregate(  
  [  
    { $sort: { "cognoms": 1 } },  
    { $skip: 1 },  
    { $limit: 3 },  
    { $set: { "guanyador": "si" } },  
    { $out: "resultats" }  
  ]  
)
```

SQL Terms, Functions, and Concepts	MongoDB Aggregation Operators
WHERE	<code>\$match</code>
GROUP BY	<code>\$group</code>
HAVING	<code>\$match</code>
SELECT	<code>\$project</code>
ORDER BY	<code>\$sort</code>
LIMIT	<code>\$limit</code>
<code>SUM()</code>	<code>\$sum</code>
<code>COUNT()</code>	<code>\$sum</code>
join	No direct corresponding operator; <i>however</i> , the <code>\$unwind</code> operator allows for somewhat similar functionality, but with fields embedded within the document.

Indexar

L'índex per **Id** es
crea automàticament



Se'n poden
afegir més



```
> db.usuaris.createIndex({"nom":1})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```