

# Introducción a bases de datos

## Sesión 7

Adrian González Pardo

1 de junio de 2021

## 1. Ejemplos

### 1.1. Ejemplo 1

sample\_airbnb.listingsAndReviews

DOCUMENTS 5.6k TOTAL SIZE 90.0MB AVG. SIZE 16.6KB INDEXES 4

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION: Untitled-Modified SAVE SAMPLE MODE

5555 Documents in the Collection Preview of Documents in the Collection

**\$match** Output after \$match stage (Sample of 20 documents)

```
1 {
2   property_type: "House"
3 }
```

**\$match** Output after \$match stage (Sample of 20 documents)

```
1 {
2   property_type: "House",
3   bedrooms: { $gte: 1 }
4 }
```

**\$addFields** Output after \$addFields stage (Sample of 20 documents)

```
1 {
2   costo_recamara: { $divide: [ "$price", "$bedrooms" ] }
3 }
```

## sample\_airbnb.listingsAndReviews

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled- Modified SAVE

\$addFields Output after \$addFields stage (Sample of 20 documents)

```
1 {
2   costo_recamara: { $divide: ["$price", "$bedrooms"] }
3 }
```

```
{
  _id: "16253247"
  listing_url: "https://www.airbnb.com/rooms/16253247"
  name: "Gorgeous Remodeled Modern Home w/ Beach Across St."
  summary: "Newly remodeled! Gorgeous home with fully remodeled master bedroom + b..."
  space: "Guests love: - Mid Century Modern Main house upgraded king bed, newly..."
  description: "Newly remodeled! Gorgeous home with..."
}
```

\$group Output after \$group stage (Sample of 1 document)

```
1 {
2   id: null
3   recamaras: {
4     $sum: 1
5   },
6   total: {
7     $sum: "$costo_recamara"
8   }
9 }
10
```

```
{
  _id: null
  recamaras: 586
  total: 84695.41428571428571428571428571430
}
```

\$addFields Output after \$addFields stage (Sample of 1 document)

```
1 {
2   costo_promedio: {
3     $divide: ["$total", "$recamaras"]
4   }
5 }
```

```
{
  _id: null
  recamaras: 586
  total: 84695.41428571428571428571428571430
  costo_promedio: 144.531423695758166747927840078010
}
```

*Ejemplos de agrupamiento*

## 1.2. Ejemplo 2

### sample\_mflix.comments

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled- Modified SAVE

50303 Documents in the Collection Preview of Documents in the Collection

Select an operator to construct expressions used in the aggregation pipeline stages. [Learn more](#)

\$lookup Output after \$lookup stage (Sample of 20 documents)

```
1 {
2   from: "movies",
3   localField: "movie_id",
4   foreignField: "_id",
5   as: "pelicula"
6 }
```

```
{
  _id: ObjectId("5a9427648b0beeb09579cc")
  name: "Andrea Le"
  email: "andrea_le@fakegmail.com"
  movie_id: ObjectId("573a1390f29313caabdc418c")
  text: "Rem officiiis eaque repellendus amet eos doloribus. Porro dolor volupta..."
  date: 2012-03-26T23:20:16.000+00:00
}
```

```
{
  _id: ObjectId("5a9427648b0beeb09579cf")
  name: "Greg Powell"
  email: "greg_powell@fakegmail.com"
  movie_id: ObjectId("573a1390f29313caabdc41b1")
  text: "Teneatur dolorum molestiae ea. Eligendi praesentium unde quod porro. Co..."
  date: 1987-02-10T00:29:36.000+00:00
}
```

\$addFields Output after \$addFields stage (Sample of 20 documents)

```
1 {
2   pelicula_objeto: { $arrayElemAt: ["$pelicula", 0] }
3 }
```

```
{
  text: "Eius veritatis vero facilis quaeat fuga temporibus. Praesentium exped..."
  date: 2002-08-18T04:56:07.000+00:00
  pelicula: Array
    _id: ObjectId("573a1390f29313caabdc4323")
    plot: "A young boy, oppressed by his mother, goes on an outing in the country ..."
    genres: Array
    runtime: 14
  _id: ObjectId("5a9427648b0beeb09579ea")
  name: "Cameron Duran"
  email: "cameron_duran@fakegmail.com"
  movie_id: ObjectId("573a1390f29313caabdc433d")
  text: "Ad asperiores mollitia aperiam non incidunt. Totam fugiat cumque praes..."
  date: 1989-01-21T14:20:47.000+00:00
  pelicula: Array
}
```

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled - Modified SAVE

Output after \$addFields stage (Sample of 20 documents)

```
1 * {
2   pelricula_objeto: {$arrayElemAt: ["$pelricula",0]}
3 }
```

text: "Eius veritatis vero facilis quaerat fuga temporibus. Praesentium exped..."  
date: 2002-08-18T04:56:07.000+00:00  
pelricula: Array  
 \_id: ObjectId("573a1300f29313caabdc4323")  
 plot: "A young boy, oppressed by his mother, goes on an outing in the country ..."  
 genres: Array  
 runtime: 14

\_id: ObjectId("5a9427648b0beebe69579ea")  
name: "Cameron Duran"  
email: "cameron\_duran@fakegmail.com"  
movie\_id: ObjectId("573a1300f29313caabdc433d")  
text: "Ad asperiores mollitia aperiam non incidunt Totam fugiat cumque praes..."  
date: 1989-01-21T14:20:47.000+00:00  
pelricula: Array

Output after \$addFields stage (Sample of 20 documents)

```
1 * {
2   pelricula_nombre: "$pelricula_objeto.title"
3 }
```

\_id: ObjectId("5a9427648b0beebe69579f3")  
name: "Jorah Mormont"  
email: "lain\_glen@gameofthron.es"  
movie\_id: ObjectId("573a1300f29313caabdc44d3")  
text: "Minus sequi incidunt cum magnam. Quam voluptatum vitae ab voluptatum c..."  
date: 1994-02-18T18:52:31.000+00:00  
pelricula: Array

name: "John Bishop"  
email: "john.bishop@fakegmail.com"  
movie\_id: ObjectId("573a1300f29313caabdc44ef")  
text: "Id error ab at molestias dolorum incidunt. Non deserunt praesentium do..."  
date: 1975-01-21T00:31:22.000+00:00  
pelricula: Array  
pelricula\_objeto: Object  
 pelricula\_nombre: "A Corner in Wheat"

Output after \$project stage (Sample of 20 documents)

```
1 * {
2   _id:0,
3   pelricula_nombre:1,
4   name:1,
5   text:1
6 }
```

name: "Mercedes Tyler"  
text: "Eius veritatis vero facilis quaerat fuga temporibus. Praesentium exped..."  
pelricula\_nombre: "The Land Beyond the Sunset"

name: "Cameron Duran"  
text: "Ad asperiores mollitia aperiam non incidunt Totam fugiat cumque praes..."

## Ejemplos de asociación

## 1.3. Ejemplo 3

sample\_mflix.comentario\_pelricula (view on: sample\_mflix.comments) MODIFY SOURCE

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

VIEW

name: "Greg Powell"  
text: "Tenetur dolorum molestiae ea. Eligendi praesentium unde quod porro. Co..."

name: "Talisa Maegyr"  
text: "Rem itaque ad sit rem voluptatibus. Ad fugiat maxime illum optio iure ..."

name: "Cameron Duran"  
text: "Quasi dicta culpa asperiores quaerat perferendis neque. Est animi pari..."

name: "Petyr Baelish"  
text: "Quo deserunt ipsam ipsum. Tenetur eos nemo nam sint praesentium minus ..."

name: "Taylor Hill"  
text: "Neque repudiandae laborum earum ipsam facilis blanditiis voluptate. Al..."

name: "Olly"  
text: "Perspiciatis sit pariatur quas. Perferendis officia harum ipsum deleni..."

name: "Joshua Kent"  
text: "Corporis pariatur rem autem accusamus debitis. Eaque aspernatur quae a..."

name: "Mercedes Tyler"  
text: "Eius veritatis vero facilis quaerat fuga temporibus. Praesentium exped..."  
pelricula\_nombre: "The Land Beyond the Sunset"

name: "Cameron Duran"  
text: "Ad asperiores mollitia aperiam non incidunt. Totam fugiat cumque praes..."

## Ejemplos generación de vistas

## 2. Retos

### 2.1. Reto 1

sample\_airbnb.listingsAndReviews

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled- Modified SAVE

5555 Documents in the Collection Preview of Documents in the Collection

**\$match** Output after \$match stage (Sample of 20 documents)

```
1- {
2-   "property type": "House",
3-   "bedrooms": { "$gte": 1 }
4- }
```

**\$addFields** Output after \$addFields stage (Sample of 20 documents)

```
1- {
2-   "costo_recamara": { "$divide": [ "$price", "$bedroom
3-   ]
4- }
```

**\$group** Output after \$group stage (Sample of 9 documents)

```
1- {
2-   "_id": "$address.country",
3-   "recamaras": {
4-     "$sum": 1
5-   },
6-   "total": {
7-     "$sum": "$costo_recamara"
8-   }
9- }
```

sample\_airbnb.listingsAndReviews

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled- Modified SAVE

**\$group** Output after \$group stage (Sample of 9 documents)

```
1- {
2-   "_id": "$address.country",
3-   "recamaras": {
4-     "$sum": 1
5-   },
6-   "total": {
7-     "$sum": "$costo_recamara"
8-   }
9- }
```

**\$addFields** Output after \$addFields stage (Sample of 9 documents)

```
1- {
2-   "pais": "$_id",
3-   "costo_promedio": { "$divide": [ "$total", "$recamar
4-   ]
5- }
```

**\$project** Output after \$project stage (Sample of 9 documents)

```
1- {
2-   "_id": 0,
3-   "pais": 1,
4-   "costo_promedio": 1
5- }
```

Retos de agrupamiento

## 2.2. Reto 2

sample\_mflix.comments

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled- Modified SAVE

50303 Documents in the Collection Preview of Documents in the Collection

Select an operator to construct expressions used in the aggregation pipeline stages. [Learn more](#)

**\$lookup** Output after \$lookup stage (Sample of 20 documents)

```
1 = {
2   "from": "users",
3   "localField": "name",
4   "foreignField": "name",
5   "as": "usuario"
6 }
```

**\$addFields** Output after \$addFields stage (Sample of 20 documents)

```
1 = {
2   "usuario_objeto": { "$arrayElemAt": [ "$usuario", 0 ] }
3 }
```

**\$addFields** Output after \$addFields stage (Sample of 20 documents)

```
1 = {
2   "usuario_objeto": { "$arrayElemAt": [ "$usuario", 0 ] }
3 }
```

**\$project** Output after \$project stage (Sample of 20 documents)

```
1 = {
2   "_id": 0,
3   "name": 1,
4   "email": 1,
5   "usuario_password": 1
6 }
```

*Reto asociación*

## 2.3. Reto 3

sample\_mflix.email\_password\_users (view on: sample\_mflix.comments) [MODIFY SOURCE](#)

Documents Aggregations Schema Explain Plan Indexes Validation

[FILTER](#) { field: 'value' }

[VIEW](#) [JSON](#) [SQL](#) [YAML](#)

<code>name: "Andrea Le"</code> <code>email: "andrea_le@fakegmail.com"</code> <code>usuario_password: "\$2b\$12\$JS87HwUL2y8P1E6kYrcbK0Kx22.w5KEdlT50F734/VKdhdudLM8Ve"</code>
<code>name: "Greg Powell"</code> <code>email: "greg_powell@fakegmail.com"</code> <code>usuario_password: "\$2b\$12\$XpveUB6KIu3zG5aABw260i1tIB7cdBBSUMJA24WDF4XXyWNJ/np76"</code>
<code>name: "Talisa Maegyr"</code> <code>email: "oona_chaplin@gameofthron.es"</code> <code>usuario_password: "\$2b\$12\$B3DKrkTVNJd1gT0xClpcw0JvnKICuSLerTxnFZ1eoqbHIQURsNrXS"</code>
<code>name: "Cameron Duran"</code> <code>email: "cameron_duran@fakegmail.com"</code> <code>usuario_password: "\$2b\$12\$56w2j63AT6mhV0h2rgdjv.w0d9TV0Jb9XK/Anms0FxySVGMf5MwvK"</code>
<code>name: "Petyr Baelish"</code> <code>email: "aidan_gillen@gameofthron.es"</code> <code>usuario_password: "\$2b\$12\$Qm.YvmiekYYY7p7phpK30icbRCDKN7ESWYAnG/o9YnFHC0Mhkmbi"</code>
<code>name: "Taylor Hill"</code> <code>email: "taylor_hill@fakegmail.com"</code> <code>usuario_password: "\$2b\$12\$5NR03TpZt162ZN2rSLxY0ourfMQAUeQ4oiufFfAhcsSqGp4eKF4GI"</code>
<code>name: "Olly"</code> <code>email: "brenock_o'connor@gameofthron.es"</code> <code>usuario_password: "\$2b\$12\$Mp9mG3u/6xuQSe/3/Uj08uTHUDQUicV0Ik142vfAqKLaBdyXhjv."</code>
<code>name: "Joshua Kent"</code> <code>email: "joshua_kent@fakegmail.com"</code> <code>usuario_password: "\$2b\$12\$XsoFV4d1icg1RA9yRtB8BescTvQFusNdde7/yz6AQ8lRKXMTdhrLW"</code>

*Reto generación de vistas*

### 3. Ejercicio

sample\_mflix.comments

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled- Modified SAVE

50303 Documents in the Collection Preview of Documents in the Collection

Select an operator to construct expressions used in the aggregation pipeline stages. [Learn more](#)

**\$lookup** Output after \$lookup stage (Sample of 20 documents)

```
1 {
2   from: 'movies',
3   localField: 'movie_id',
4   foreignField: '_id',
5   as: 'películas'
6 }
```

**\$match** Output after \$match stage (Sample of 20 documents)

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   películas:{$gt:[]}
6 }
```

**\$addFields** Output after \$addFields stage (Sample of 20 documents)

```
1 {
2   película objeto: {
3     $arrayElemAt: [
4       '$películas',
5       0
6     ]
7   }
8 }
```

**\$addFields** Output after \$addFields stage (Sample of 20 documents)

```
1 /**
2  * newField: The new field name.
3  * expression: The new field expression.
4  */
5 {
6   país: '$película objeto.countries',
7   genero: '$película objeto.genres'
8 }
```

**\$unwind** Output after \$unwind stage (Sample of 20 documents)

```
1 {
2   path: '$país'
3 }
```

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled - Modified SAVE SAMPLE MODE

**\$unwind** Output after \$unwind stage (Sample of 20 documents)

```
1 {
2   path: "$genero"
3 }
```

```
email: "mercedes_tyler@fakegmail.com"
movie_id: ObjectId("573a1390f29313caabdc4323")
text: "Eius veritatis vero facilis quaerat fuga temporibus. Praesentium exped..."
date: 2002-08-18T04:56:07.000+00:00
peliculas: Array
pelicula objeto: Object
pais: "USA"
genero: "Short"
```

```
_id: ObjectId("5a9427648b0beebe69579e7")
name: "Mercedes Tyler"
email: "mercedes_tyler@fakegmail.com"
movie_id: ObjectId("573a1390f29313caabdc4323")
text: "Eius veritatis vero facilis quaerat fuga temporibus. Praesentium exped..."
date: 2002-08-18T04:56:07.000+00:00
peliculas: Array
pelicula objeto: Object
```

```
_id: ObjectId("5a9427648b0beebe69579e7")
name: "Mercedes Tyler"
email: "mercedes_tyler@fakegmail.com"
movie_id: ObjectId("573a1390f29313caabdc4323")
text: "Eius veritatis vero facilis quaerat fuga temporibus. Praesentium exped..."
date: 2002-08-18T04:56:07.000+00:00
peliculas: Array
pelicula objeto: Object
```

**\$project** Output after \$project stage (Sample of 20 documents)

```
1 {
2   _id:0,
3   pais:1,
4   genero:1
5 }
```

```
pais: "USA"
genero: "Short"
```

```
pais: "USA"
genero: "Drama"
```

```
pais: "USA"
genero: "Fantasy"
```

**\$group** Output after \$group stage (Sample of 20 documents)

```
1 /**
2  * id: The id of the group.
3  * fieldN: The first field name.
4  */
5 {
6   id: {
7     pais: "$pais",
8     genero: "$genero"
9   },
10  peliculas: {
11    $sum:1
12  }
13 }
```

```
_id: Object
pais: "Panama"
genero: "Drama"
peliculas: 1
```

```
_id: Object
pais: "Slovenia"
genero: "Crime"
peliculas: 1
```

```
_id: Object
pais: "Norway"
genero: "Drama"
peliculas: 47
```

## sample\_mflix.comments

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled - Modified SAVE

**\$group** Output after \$group stage (Sample of 20 documents)

```
1 /**
2  * id: The id of the group.
3  * fieldN: The first field name.
4  */
5 {
6   id: {
7     pais: "$pais",
8     genero: "$genero"
9   },
10  peliculas: {
11    $sum:1
12  }
13 }
```

```
_id: Object
pais: "Panama"
genero: "Drama"
peliculas: 1
```

```
_id: Object
pais: "Slovenia"
genero: "Crime"
peliculas: 1
```

**\$addFields** Output after \$addFields stage (Sample of 20 documents)

```
1 /**
2  * newField: The new field name.
3  * expression: The new field expression.
4  */
5 {
6   pais: "$_id.pais",
7   genero: "$_id.genero"
8 }
```

```
_id: Object
peliculas: 209
pais: "UK"
genero: "Mystery"
```

```
_id: Object
peliculas: 47
pais: "Spain"
genero: "Thriller"
```

**\$project** Output after \$project stage (Sample of 20 documents)

```
1 /**
2  * specifications: The fields to
3  * include or exclude.
4  */
5 {
6   id:0,
7   pais:1,
8   genero:1,
9   peliculas:1
10 }
11 }
```

```
peliculas: 4
pais: "Czech Republic"
genero: "Family"
```

```
peliculas: 72
pais: "Japan"
genero: "Animation"
```

## Ejercicio de la sesión

Para resolver este reto se tiene el ejemplo del código del json empleado.

```
1 /* Lookup en comments */
2 {
```



```

3   "from": "movies",
4   "localField": "movie_id",
5   "foreignField": "_id",
6   "as": "peliculas"
7 }
8
9 /* Match para eliminar arreglos vacios */
10 {
11   "peliculas":{"$gt":[]}
12 }
13
14 /* AddField */
15 {
16   "pelicula_objeto": {
17     "$arrayElemAt": [
18       "$peliculas",
19       0
20     ]
21   }
22 }
23
24 /* AddField */
25 {
26   "pais":"$pelicula_objeto.countries",
27   "genero":"$pelicula_objeto.genres"
28 }
29
30 /* Unwind para el array de pais y posterior el de genero */
31 {
32   "path": "$pais"
33 }
34 {
35   "path": "$genero"
36 }
37
38 /* Project de los datos que necesitamos */
39 {
40   "_id":0,
41   "pais":1,
42   "genero":1
43 }
44
45 /* Group para los datos, como nota aqui podria terminar el reto */
46 {
47   "_id": {
48     "pais":"$pais",
49     "genero":"$genero"
50   },
51   "peliculas": {
52     "$sum":1
53   }
54 }
55
56 /* En caso de que no se desee ver el json con el _id, addField */
57 {
58   "pais":"$_id.pais",

```

```
59     "genero":"$_id.genero"
60 }
61
62 /* Project */
63 {
64     "_id":0,
65     "pais":1,
66     "genero":1,
67     "peliculas":1
68 }
69 }
```