

Los resultados se escriben en la sintaxis de Mongo Shell.

Siempre que se pueda se intentará resolver en primera instancia con un **find**. En caso de no ser posible o no poder hacerlo, se hará con una agregación. Algunas consultas se muestran tanto con una agregación y con find, ya que aunque sean posible realizarse con un find, hacerlo de primeras usando find no es nada trivial por lo que es más intuitivo plantearla en primera instancia como una agregación y luego transformar los resultados para que encajen en el find.

Show the owners with more than one property

He supuesto que se “host_total_list_count” es el valor scrapeado de las propiedades obtenidas.

```
{host_total_listings_count: { $gt: 2 } }
```

▼

{ host_total_listings_count: { \$gt: 2 } }

Generate query ↗

Explain

Reset

Find

</>

Options ▶

ADD DATA ▼

EXPORT DATA ▼

UPDATE

DELETE

100 ▼

1 - 100 of 8191

↺

↻

↷

⌵

⌶

⌷

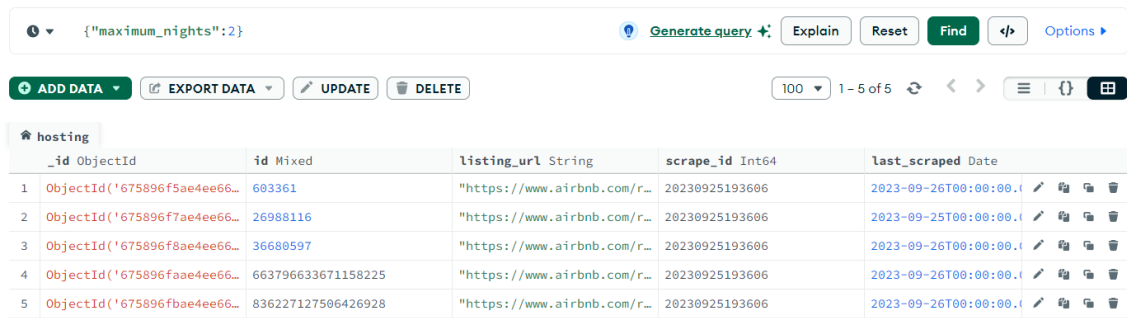
★ hosting						
	_id ObjectId	id Int32	listing_url String	scrape_id Int64	last_scraped Date	
1	ObjectId('675896f5ae4ee66...	96033	"https://www.airbnb.com/r...	20230925193606	2023-09-25T00:00:00	✎ 📄 🗑
2	ObjectId('675896f5ae4ee66...	166473	"https://www.airbnb.com/r...	20230925193606	2023-09-25T00:00:00	✎ 📄 🗑
3	ObjectId('675896f5ae4ee66...	212845	"https://www.airbnb.com/r...	20230925193606	2023-09-25T00:00:00	✎ 📄 🗑
4	ObjectId('675896f5ae4ee66...	330760	"https://www.airbnb.com/r...	20230925193606	2023-09-25T00:00:00	✎ 📄 🗑
5	ObjectId('675896f5ae4ee66...	340024	"https://www.airbnb.com/r...	20230925193606	2023-09-26T00:00:00	✎ 📄 🗑
6	ObjectId('675896f5ae4ee66...	358541	"https://www.airbnb.com/r...	20230925193606	2023-09-26T00:00:00	✎ 📄 🗑
7	ObjectId('675896f5ae4ee66...	412798	"https://www.airbnb.com/r...	20230925193606	2023-09-26T00:00:00	✎ 📄 🗑
8	ObjectId('675896f5ae4ee66...	413063	"https://www.airbnb.com/r...	20230925193606	2023-09-26T00:00:00	✎ 📄 🗑
9	ObjectId('675896f5ae4ee66...	421077	"https://www.airbnb.com/r...	20230925193606	2023-09-26T00:00:00	✎ 📄 🗑
10	ObjectId('675896f5ae4ee66...	427860	"https://www.airbnb.com/r...	20230925193606	2023-09-25T00:00:00	✎ 📄 🗑
11	ObiectId('675896f5ae4ee66...	505688	"httos://www.airbnb.com/r...	20230925193606	2023-09-26T00:00:00	✎ 📄 🗑

Figura 1 Consulta y resultado en MongoDBCompass de la primera consulta

Show the apartments where we can stay for just 2 nights

Para enseñar los apartamentos en el que te puedes quedar 2 noches, nos tendremos que asegurar que “maximum_nights” “2”. No tendré en cuenta el campo “maximum_maximum_nights”, lo consideraré como un dato redundante y en caso de diferir tomaré máximum nights como el valor correcto.

```
{“maximum_nights”: 2}
```



The screenshot shows the MongoDB Compass interface. At the top, a query filter is set to `{"maximum_nights":2}`. Below the query bar, there are buttons for `ADD DATA`, `EXPORT DATA`, `UPDATE`, and `DELETE`. The results are displayed in a table with 5 rows. The columns are: `_id` (ObjectId), `id` (Mixed), `listing_url` (String), `scrape_id` (Int64), and `last_scraped` (Date). Each row has a set of action icons on the right.

	_id ObjectId	id Mixed	listing_url String	scrape_id Int64	last_scraped Date
1	ObjectId('675896f5ae4ee66...	603361	"https://www.airbnb.com/r...	20230925193606	2023-09-26T00:00:00.0...
2	ObjectId('675896f7ae4ee66...	26988116	"https://www.airbnb.com/r...	20230925193606	2023-09-25T00:00:00.0...
3	ObjectId('675896f8ae4ee66...	36680597	"https://www.airbnb.com/r...	20230925193606	2023-09-26T00:00:00.0...
4	ObjectId('675896faae4ee66...	663796633671158225	"https://www.airbnb.com/r...	20230925193606	2023-09-26T00:00:00.0...
5	ObjectId('675896fbae4ee66...	836227127506426928	"https://www.airbnb.com/r...	20230925193606	2023-09-26T00:00:00.0...

Figura 2 Consulta y resultado en MongoDBCompass de la segunda consulta, en este caso parece haber 5 resultados.

Show the apartments with a first review in 2023, more than 10 reviews, with at least 2 beds and (private room or entire home).

```
{
  first_review: {
    $gte: ISODate("2023-01-01T00:00:00.000Z")
  },
  number_of_reviews: { $gt: 10 },
  beds: { $gte: 2 },
  $or: [
    { room_type: { $regex: /Entire home/i } },
    { room_type: { $regex: /private room/i } }
  ]
}
```

En este caso se ha usado una regex para ver que el resultado de room_type contenga esas palabras. Otra forma sería utilizar un índice de texto para room_type y utilizar un \$search.

Se ha usado ISODate para comparar con las fechas y se ha puesto el instante 0 del 2023.

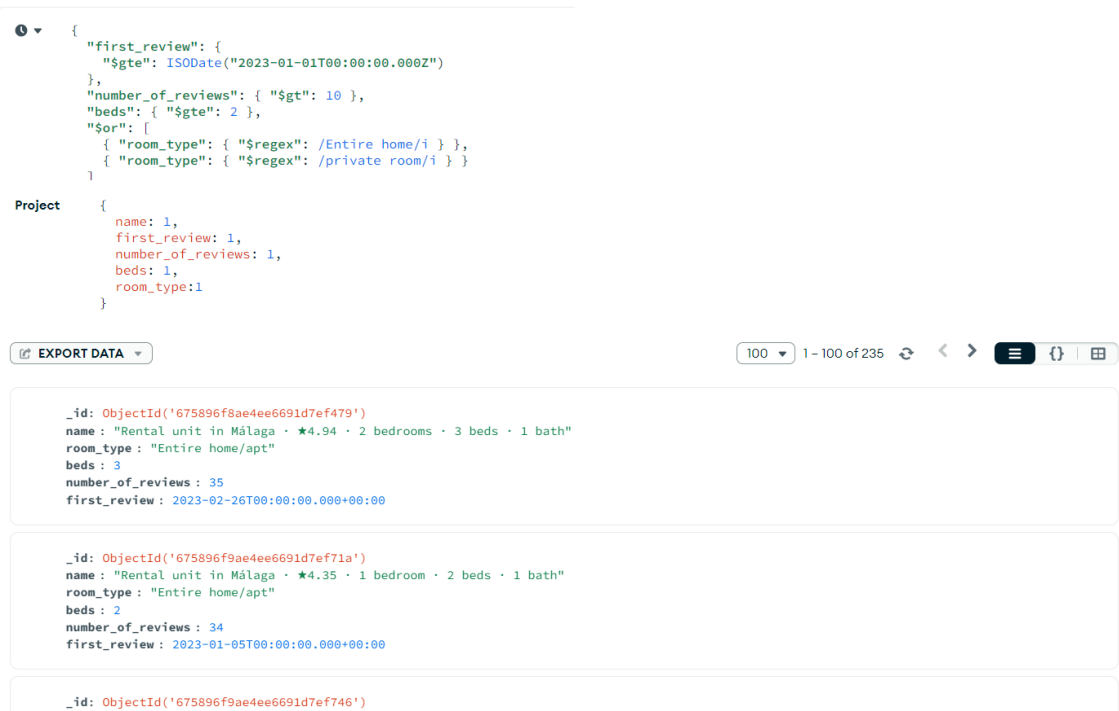


Figura 3 Se obtienen 235 resultados que coinciden con la query.

Show the apartments having wifi, coffee maker and beach access.

```
{
  "$and": [
    { "amenities": "/coffee maker/i" },
    { "amenities": "/beach access/i" },
    { "amenities": "/wifi/i" }
  ]
}
```

Al igual que en el anterior ejercicio sería posible usar un índice de texto y buscarlo con una consulta parecida a algo así: `{ $text: { $search: "coffee maker" } }`.

Pero como no se especifica nada, la hago utilizando regex.

The screenshot shows a MongoDB query builder interface. The query is:

```
{
  $and: [
    { amenities: RegExp("coffee maker", "i") },
    { amenities: RegExp("beach access", "i") },
    { amenities: RegExp("wifi", "i") }
  ]
}
```

Below the query, there are buttons: ADD DATA, EXPORT DATA, UPDATE, and DELETE. To the right, there are buttons: Generate query, Explain, Reset, Find, and Options. Below these buttons, there is a table with 1704 results. The table has columns: _id, id, listing_url, scrape_id, last_scraped, and source. The first 12 rows are shown, and the table is paginated to show 100 of 1604 results.

	_id	id	listing_url	scrape_id	last_scraped	source
1	ObjectId('675896f5ae4ee66...')	96833	"https://www.airbnb.com/r/...	20230925193606	2023-09-25T00:00:00.000+0...	"city sc
2	ObjectId('675896f5ae4ee66...')	212845	"https://www.airbnb.com/r/...	20230925193606	2023-09-25T00:00:00.000+0...	"city sc
3	ObjectId('675896f5ae4ee66...')	340024	"https://www.airbnb.com/r/...	20230925193606	2023-09-26T00:00:00.000+0...	"city sc
4	ObjectId('675896f5ae4ee66...')	427869	"https://www.airbnb.com/r/...	20230925193606	2023-09-25T00:00:00.000+0...	"city sc
5	ObjectId('675896f5ae4ee66...')	505688	"https://www.airbnb.com/r/...	20230925193606	2023-09-26T00:00:00.000+0...	"city sc
6	ObjectId('675896f5ae4ee66...')	507961	"https://www.airbnb.com/r/...	20230925193606	2023-09-26T00:00:00.000+0...	"city sc
7	ObjectId('675896f5ae4ee66...')	508787	"https://www.airbnb.com/r/...	20230925193606	2023-09-26T00:00:00.000+0...	"city sc
8	ObjectId('675896f5ae4ee66...')	509073	"https://www.airbnb.com/r/...	20230925193606	2023-09-26T00:00:00.000+0...	"city sc
9	ObjectId('675896f5ae4ee66...')	558804	"https://www.airbnb.com/r/...	20230925193606	2023-09-26T00:00:00.000+0...	"city sc
10	ObjectId('675896f5ae4ee66...')	634046	"https://www.airbnb.com/r/...	20230925193606	2023-09-25T00:00:00.000+0...	"city sc
11	ObjectId('675896f5ae4ee66...')	655701	"https://www.airbnb.com/r/...	20230925193606	2023-09-25T00:00:00.000+0...	"city sc
12	ObjectId('675896f5ae4ee66...')	718943	"https://www.airbnb.com/r/...	20230925193606	2023-09-25T00:00:00.000+0...	"city sc

Figura 4 Obtenemos 1704 resultados en la cuarta consulta.

Show the apartments with an indication of the price in Euros (€) in the description with a host response rate of more than 80%.

El resultado de la consulta es:

```
[
  {
    $match: { description: /€/i }
  },
  {
    $addFields:
    {
      host_response_rate_number: {
        $convert: {
          input: {
            $substr: [
              "$host_response_rate",
              0,
              {
                $subtract: [
                  {
                    $strLenCP:
                      "$host_response_rate"
                  },
                  1
                ]
              }
            ]
          },
          to: "double",
          onError: null,
          onNull: null
        }
      }
    }
  },
  {
    $match:
    {
      host_response_rate_number: {
        $gt: 80
      }
    }
  }
]
```

La consulta la diseñé con una agregación en 3 etapas. El pipeline es el siguiente:

1. Primero se filtra el parámetro de descripción. Que la indicación del precio sea en euros (€) es algo ambiguo como enunciado, ya que al ser un texto largo donde puede aparecer varias veces mencionado el precio sin una estructura clara hace que sea difícil encontrar el resultado. Por lo que yo he interpretado que para esta etapa en alguna parte de la descripción debe aparecer el símbolo “€”. Es interesante filtrar siempre al principio todo lo posible para no tener que hacer el resto de las etapas en campos innecesarios.

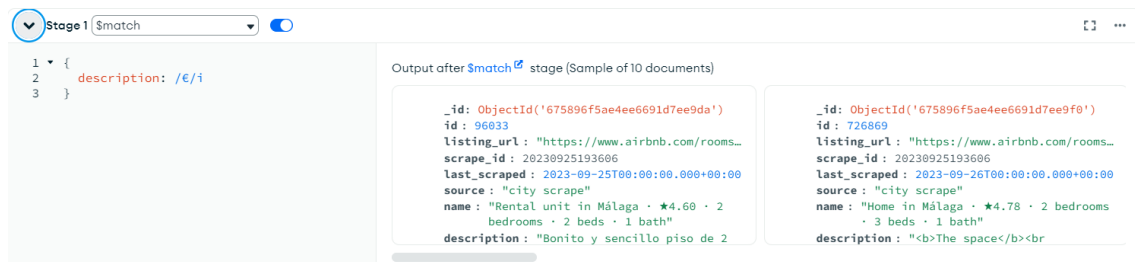


Figura 5 Etapa 1 del pipeline que filtra la descripción.

2. En la segunda etapa se añade un nuevo campo llamado “host_response_rate_number”. Se parte del “host_response_rate” que es un string con un número y terminado por “%”. Para convertir este string a un número en el que luego podamos comparar con otro número, se usa el convert para pasar de cadena de texto a número. Otra opción sería usar el “\$todouble”. Antes de poder transformar la cadena de texto a un número hay que quitar el último carácter “%”. Para ello se usa el “\$substr”, el substr obtiene una sub-cadena de texto nos pide hasta que carácter. La sub-cadena de texto que queremos va desde el índice 0 hasta la longitud del “host_response_rate”-1. Para obtener la longitud utilizamos el operador “\$strLenCP” y usamos “\$subtract” para restarle uno.



Figura 6 Segunda etapa del pipeline con ejemplos de resultado

3. Para la última etapa simplemente filtramos por aquellos que el nuevo field “host_response_rate_number” sea mayor que 80.

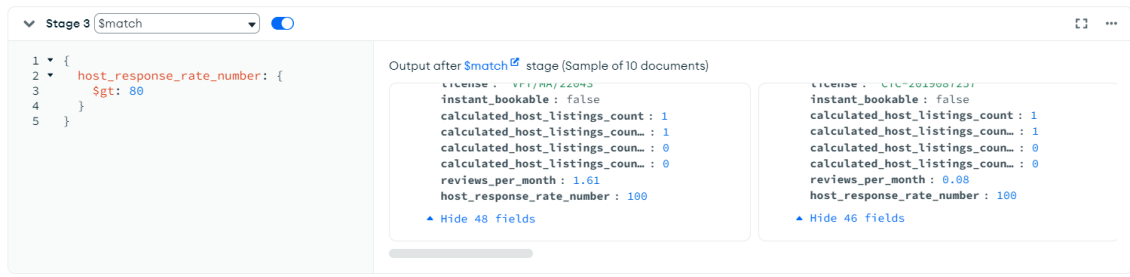


Figura 7 Tercera etapa

Si se quisiera hacer con un único find se puede usar el operador **\$expr** que permite usar operadores de agregación en find:

```
{
  $expr: {
    $and: [
      {
        $regexMatch: {
          input: "$description",
          regex: "€",
          options: "i"
        }
      }
    ],
    $gt: [
      {
        $convert: {
          input: {
            $substr: [
              "$host_response_rate",
              0,
              {
                $subtract: [
                  {
                    $strLenCP:
                      "$host_response_rate"
                  },
                  1
                ]
              }
            ]
          },
          to: "double",
          onError: null,
          onNull: null
        }
      }
    ],
    80
  ]
}
```

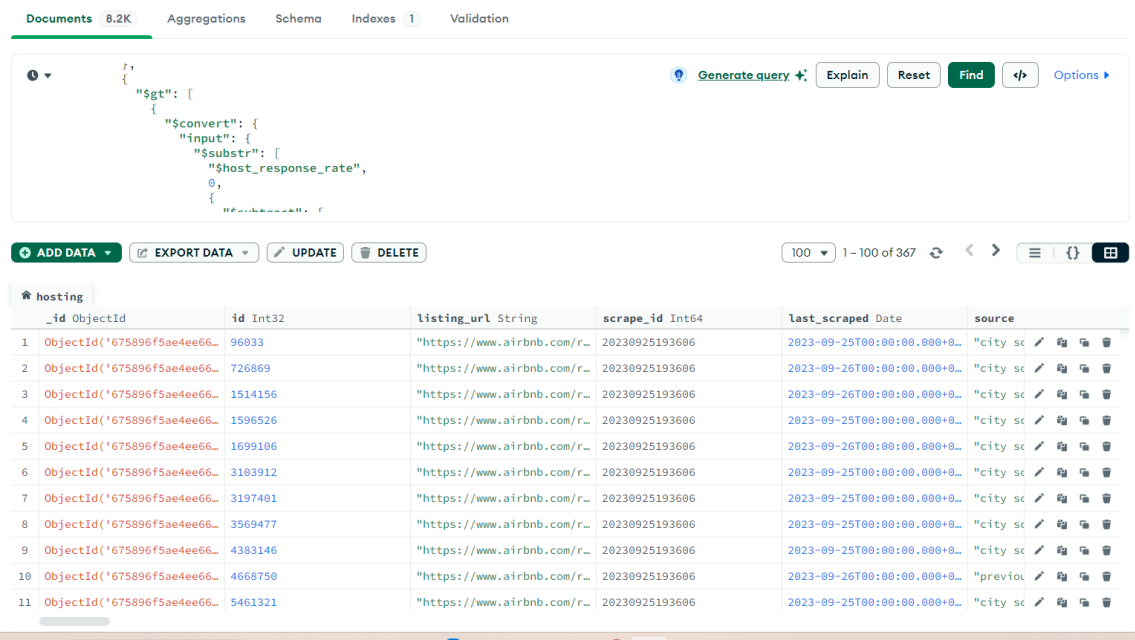



Figura 8 Se obtienen 367 resultados del find

Show the apartments cheaper than 50€ with verified email and work email (use the change the day of solving the exercise)

```
[
  {
    $match: {
      $and: [
        {host_verifications: {$regex: "\\bwork_email\\b", $options: "i"}},
        {host_verifications: {$regex: "\\bemail\\b", $options: "i"}}
      ]
    }
  },
  {
    $addFields: {
      price_double: {
        $convert: {
          input: {$substr: ["$price",1, {$strLenCP: "$price"} ] },
          to: "double",
          onError: null,
          onNull: null
        }
      }
    }
  },
  {
    $match:
    {
      price_double: {
        $lt: 52.5
      }
    }
  },
  {
    $project: {
      name: 1,
      price: 1,
      price_double: 1,
      host_verifications: 1
    }
  }
]
```

Este ejercicio tiene una lógica muy similar al ejercicio anterior y también lo he resuelto con una agregación dividida en 3 etapas.

1. La primera etapa filtra los apartamentos cuyo host tienen verificado el email y email de trabajo, esto para que los siguientes pasos se realicen en solo aquellos campos que nos interesan. En este caso encontramos las verificaciones en un string llamado `host_verifications`. Usamos un `$and` para obtenerlos junto a una expresión regular. La expresión regular `\\bwork_email\\b` busca la palabra exacta `work_email` dentro del string (delimitada por bordes de palabra, `\b`):

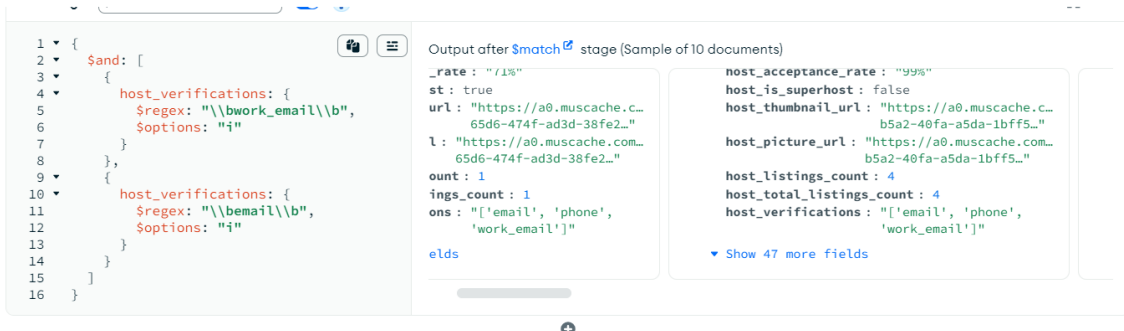


Figura 9 Primera etapa filtrar por correos.

2. En la segunda etapa es muy parecido al anterior ejercicio. Se usa también un `convert` para pasar de cadena de texto a número y he usado los operadores `$substr` y `$strLenCP` ya que el atributo `price` guarda una cadena de texto que comienza siempre con un `$`. (El cual he interpretado que siempre estará guardado en dólares). Por lo que esta vez el `$substr` irá desde la posición 1 hasta el final de la longitud de la cadena de texto:

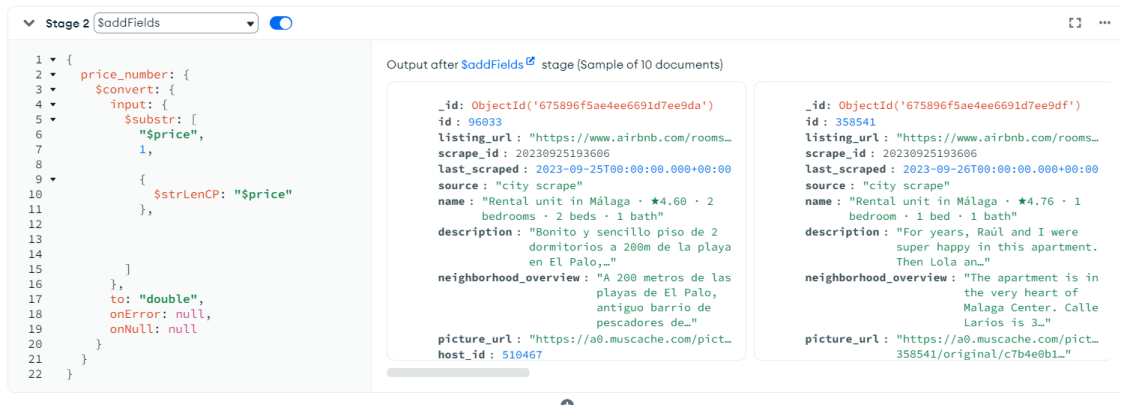


Figura 10 Segunda etapa del pipeline

3. Para la última etapa se tiene en cuenta que a día de 14 de diciembre de 2024 un euro equivale a 1.05 dólares y se debe filtrar por el campo creado. Por lo que la consulta quedaría como se demuestra en la Figura

11. Otra forma alternativa sería por filtrar multiplicado por el valor del dólar (Véase la Figura 12).

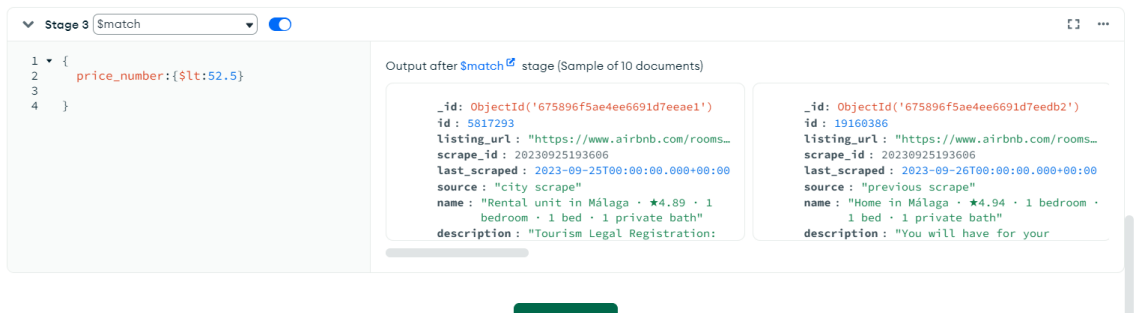


Figura 11 Tercera etapa

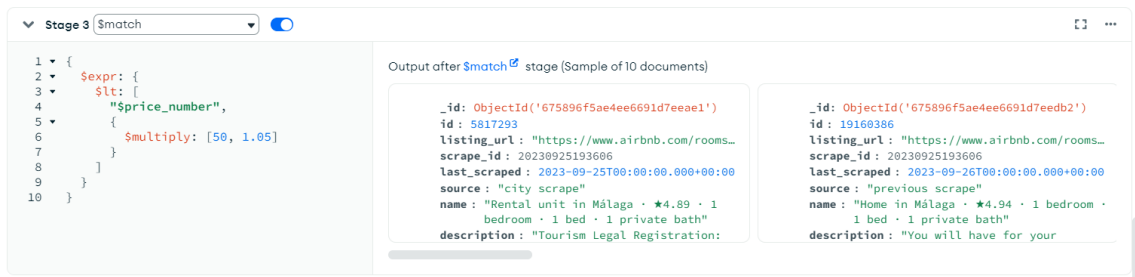


Figura 12 Ejemplo de tercera etapa usando \$multiply



Figura 13 Valor del dólar

Por último, esta consulta también podría hacerse con un find utilizando el operador \$expr, quedaría de la siguiente forma:

```
{
  $and: [
    {
      host_verifications: {
        $regex: "\\bwork_email\\b",
        $options: "i"
      }
    },
    {
      host_verifications: {
        $regex: "\\bemail\\b",
        $options: "i"
      }
    },
    {
      $expr: {
        $lt: [
          {
            $convert: {
              input: {
                $substr: [
                  "$price",
                  1,
                  { $strlenCP: "$price" }
                ]
              },
              to: "double",
              onError: null,
              onNull: null
            }
          ],
          52.5
        ]
      }
    }
  ]
}
```

Documents 8.2K Aggregations Schema Indexes 1 Validation

```

    $substr: [
      "$price",
      1,
      {
        $strLenCP: "$price"
      }
    ],
    to: "double",
    onError: null.
  
```

Generate query Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE 100 1 - 100 of 993

★ hosting	_id ObjectId	id Int32	listing_url String	scrape_id Int64	last_scraped Date	source
1	ObjectId('675896f5ae4ee66...	166473	"https://www.airbnb.com/r...	20230925193606	2023-09-25T00:00:00.000+0...	"city sc /
2	ObjectId('675896f5ae4ee66...	212845	"https://www.airbnb.com/r...	20230925193606	2023-09-25T00:00:00.000+0...	"city sc /
3	ObjectId('675896f5ae4ee66...	603361	"https://www.airbnb.com/r...	20230925193606	2023-09-26T00:00:00.000+0...	"city sc /
4	ObjectId('675896f5ae4ee66...	730493	"https://www.airbnb.com/r...	20230925193606	2023-09-25T00:00:00.000+0...	"city sc /
5	ObjectId('675896f5ae4ee66...	787704	"https://www.airbnb.com/r...	20230925193606	2023-09-25T00:00:00.000+0...	"city sc /
6	ObjectId('675896f5ae4ee66...	1048083	"https://www.airbnb.com/r...	20230925193606	2023-09-25T00:00:00.000+0...	"city sc /
7	ObjectId('675896f5ae4ee66...	1164851	"https://www.airbnb.com/r...	20230925193606	2023-09-25T00:00:00.000+0...	"city sc /
8	ObjectId('675896f5ae4ee66...	1249797	"https://www.airbnb.com/r...	20230925193606	2023-09-26T00:00:00.000+0...	"previo /
9	ObjectId('675896f5ae4ee66...	1265238	"https://www.airbnb.com/r...	20230925193606	2023-09-26T00:00:00.000+0...	"previo /
10	ObjectId('675896f5ae4ee66...	1274221	"https://www.airbnb.com/r...	20230925193606	2023-09-26T00:00:00.000+0...	"previo /
11	ObjectId('675896f5ae4ee66...	1274222	"https://www.airbnb.com/r...	20230925193606	2023-09-26T00:00:00.000+0...	"previo /

Figura 14 El resultado del find, parece encontrar 993 resultados.

How many Rafael are superhost and speak English?

```

[
  {
    $match: {
      host_name: "Rafael",
      host_is_superhost: true,
      host_about:/inglés/i
    }
  },
  {
    $group: {
      _id: "$host_id" // Agrupa por host_id para obtener IDs únicos
    }
  },
  {
    $count: "Nº Rafael:"
  }
]
  
```

He planteado esta consulta como una agregación de 3 etapas.

1. He filtrado en primer lugar por los hosts llamados Rafael que son superhost. Al acceder a su perfil se puede observar que existen hosts llamados Rafael superhost que hablan inglés (Figura 16) pero parece ser que el atributo que guarda el valor del idioma no se ha scrapeado corretamente o por lo menos no lo encuentro. Por lo que he intentado

filtrar por si dice algo en su biografía del inglés lo interpreto como válido.
En este hay un solo Rafael que cumple estas condiciones:

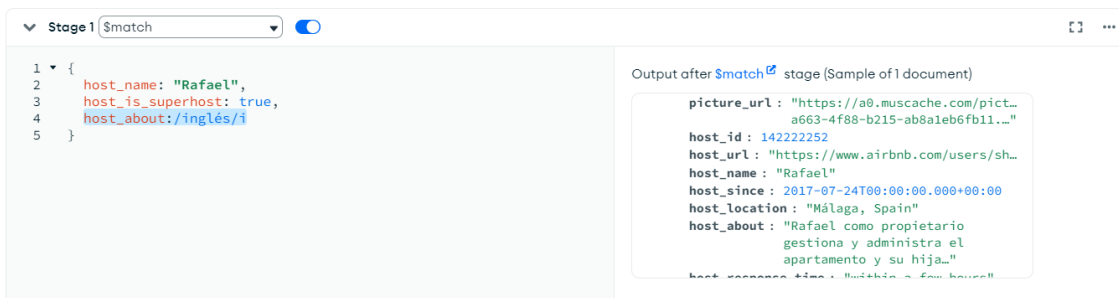


Figura 15 Resultado del primer filtro

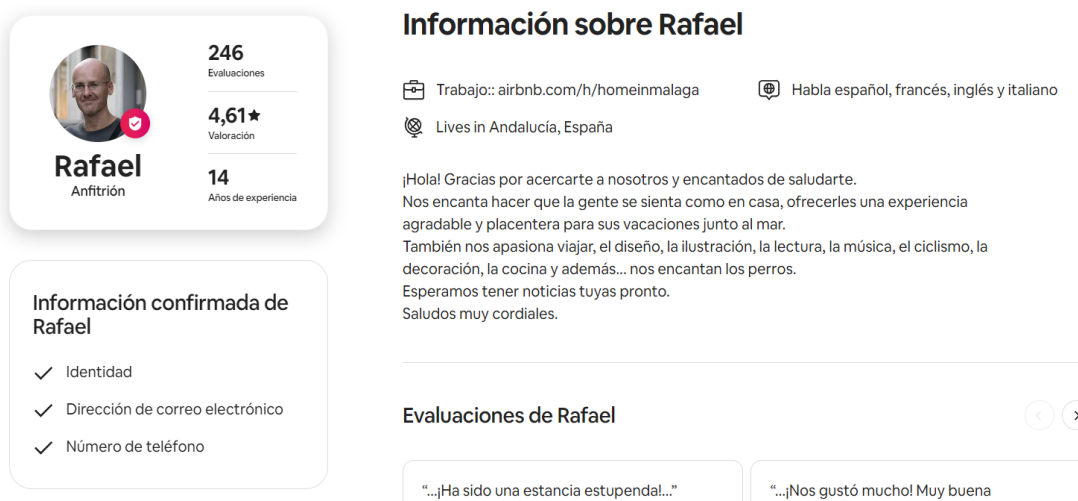


Figura 16 Rafael Super host que habla inglés.

2. La siguiente etapa sería una agrupación por host_id:

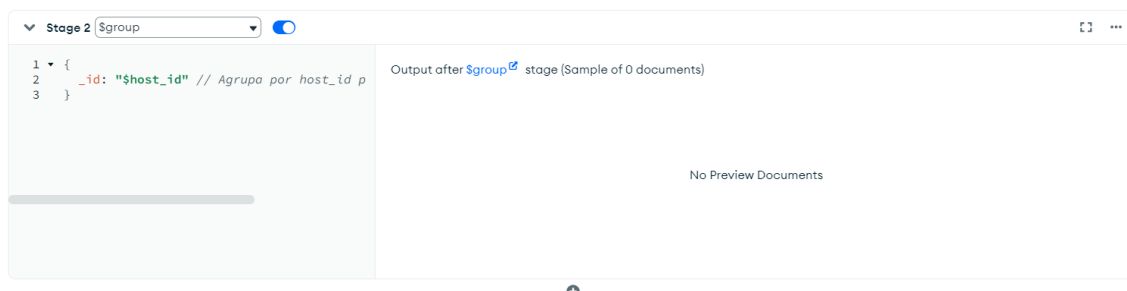


Figura 17 Etapa 2 mostrando el agrupamiento

3. En la última etapa, simplemente haríamos un “count”. El resultado sería como se ve en la Figura 18.

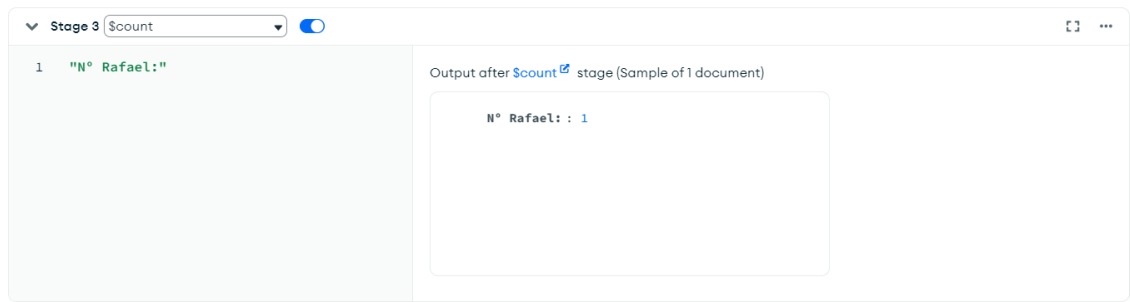


Figura 18 N° de hosts llamados Rafael que son superhost y mencionan el inglés en su biografía.

Is there any owner with a photograph in PNG format?

```
{
  $or: [
    { host_thumbnail_url: /png/i },
    { host_picture_url: /png/i }
  ]
}
```

Interpreto que las fotografías de los hosts son "host_thubmnail_url" y "host_picture_url". Se obtienen 228 resultados.

What is the best response rate in the database?

```
[
  {
    $addFields: {
      host_response_rate_number: {
        $convert: {
          input: {
            $substr: [
              "$host_response_rate",
              0,
              {
                $subtract: [
                  {
                    $strLenCP:
                      "$host_response_rate"
                  },
                  1
                ]
              }
            ]
          },
          to: "double",
          onError: null,
          onNull: null
        }
      }
    }
  },
  {
    $sort: {
      host_response_rate_number: -1
    }
  },
  {
    $limit: 1
  },
  {
    $project: {
      _id: 0,
      host_id: 1,
      host_name: 1,
      host_response_rate: 1
    }
  }
]
```

Esta consulta la he pensado también como una agregación con las siguientes etapas:

1. Al igual que antes convertimos `host_response_rate` a double.

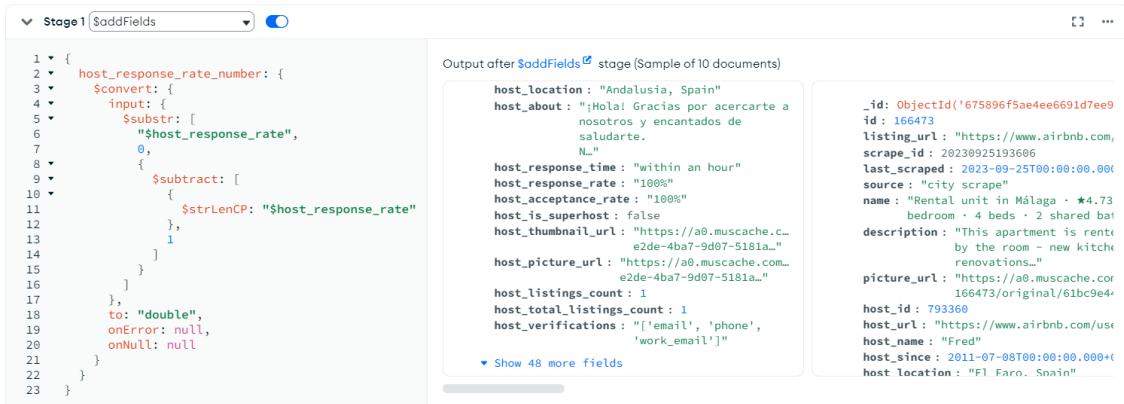


Figura 19 Primera etapa del pipeline

2. Se realiza una ordenación de mayor a menor según el valor recién calculado.

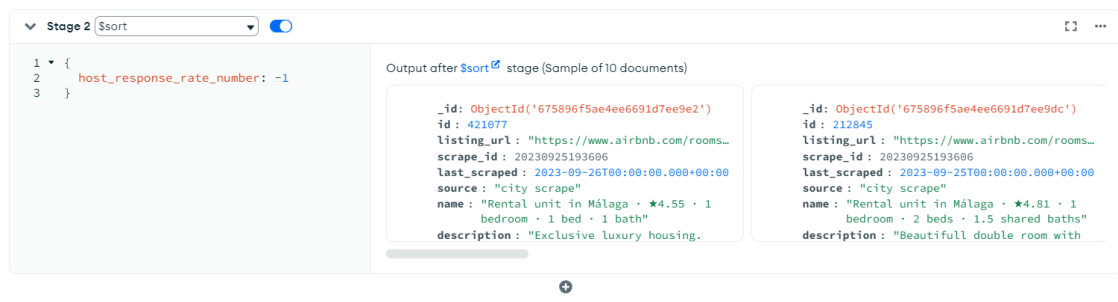


Figura 20 Se ordenan de forma descendente según `response_rate`

3. Se limitan los resultados a uno para obtener el apartamento/vivienda con el host con `response_rate` más alto.



Figura 21 Etapa 3 limitando los resultados a 1.

4. Para terminar, nos falta mostrar tanto el porcentaje del `response_rate` como el `host` que lo tiene.

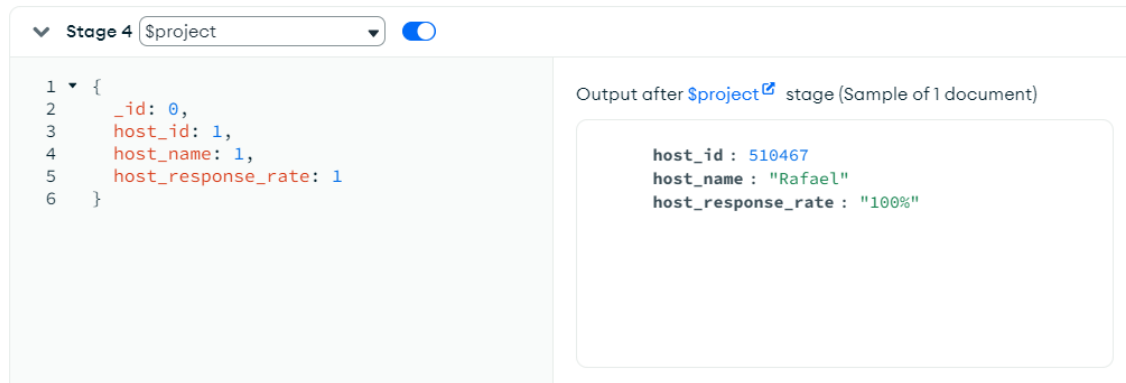


Figura 22 Proyección para ver el `host` y su `response rate`.

Show apartments without a license of Junta de Andalucía

Hay que tener en cuenta que el formato de las licencias de la junta de Andalucía es:

- **VFT**: Siglas de "Vivienda con Fines Turísticos".
- **XX**: Código de la provincia donde se ubica la vivienda.
- **NNNNN**: Número secuencial asignado a la vivienda.
- Por tanto, tenemos que buscar las licencias que empiecen por:
- **VFT/AL**: Almería
- **VFT/CA**: Cádiz
- **VFT/CO**: Córdoba
- **VFT/GR**: Granada
- **VFT/HU**: Huelva
- **VFT/JA**: Jaén
- **VFT/MA**: Málaga
- **VFT/SE**: Sevilla

Para encontrar aquellas que tienen la licencia, he decidido hacerlo con una expresión regular.

```
{
  "license": {
    "$regex": "^VFT/(AL|CA|CO|GR|HU|JA|MA|SE)",
    "$options": "i"
  }
}
```

Ahora bien, si quisiéramos aquellas que no cumplen el formato, el resultado sería añadiéndole un \$not delante:

```
{
  "license": {
    "$not": {
      "$regex": "^VFT/(AL|CA|CO|GR|HU|JA|MA|SE)",
      "$options": "i"
    }
  }
}
```

The screenshot shows a web application interface for querying data. At the top, there is a JSON query editor with a dropdown menu on the left and buttons for 'Explain', 'Reset', 'Find', and 'Options' on the right. The query is:

```
{
  "license": {
    "$not": {
      "$regex": "^VFT/(AL|CA|CO|GR|HU|JA|MA|SE)",
      "$options": "i"
    }
  }
}
```

Below the query editor, there is a toolbar with buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. To the right of these buttons, there is a status bar showing '100' and '1 - 100 of 3368'. Below the toolbar, there is a list of results. The first result is expanded, showing the following fields:

```
review_scores_checkin: 4.81
review_scores_communication: 4.91
review_scores_location: 4.8
review_scores_value: 4.69
license: "CTC/2018118027"
instant_bookable: false
calculated_host_listings_count: 4
calculated_host_listings_count_entire_homes: 0
calculated_host_listings_count_private_rooms: 4
calculated_host_listings_count_shared_rooms: 0
reviews_per_month: 0.62
```

Below the expanded result, there is a link to 'Hide 44 fields'. At the bottom, there is a section for the document's metadata:

```
_id: ObjectId('675896f5ae4ee6691d7ee9e1')
id: 413063
listing_url: "https://www.airbnb.com/rooms/413063"
scrape_id: 20230925193606
last_scraped: 2023-09-26T00:00:00.000+00:00
```

Figura 23 Se obtienen 3368 viviendas registradas sin licencia de la junta de Andalucía.