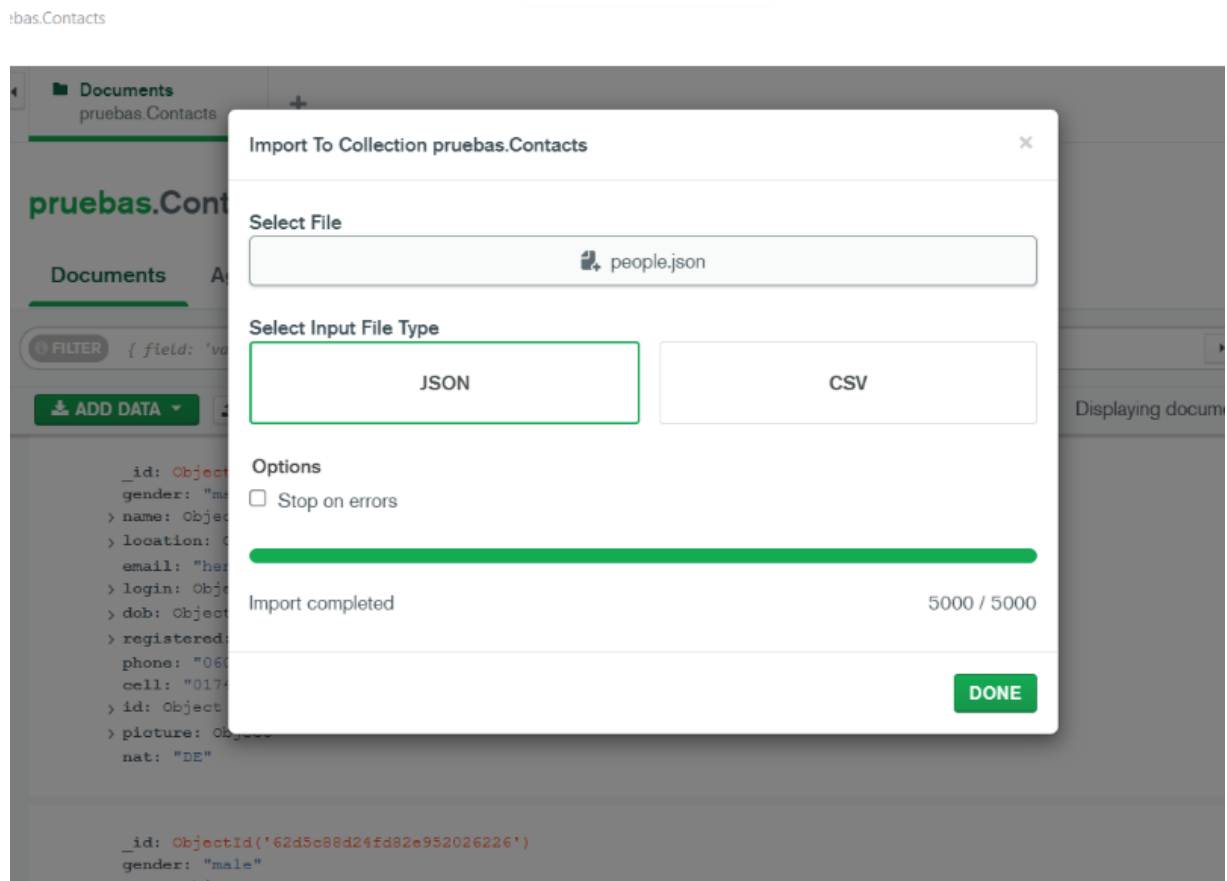


Ejercicios

- Crea una base de datos en Mongo con Mongo Compass llamada “**Pruebas**”
- Importa el JSON a una colección llamada **Contacts**



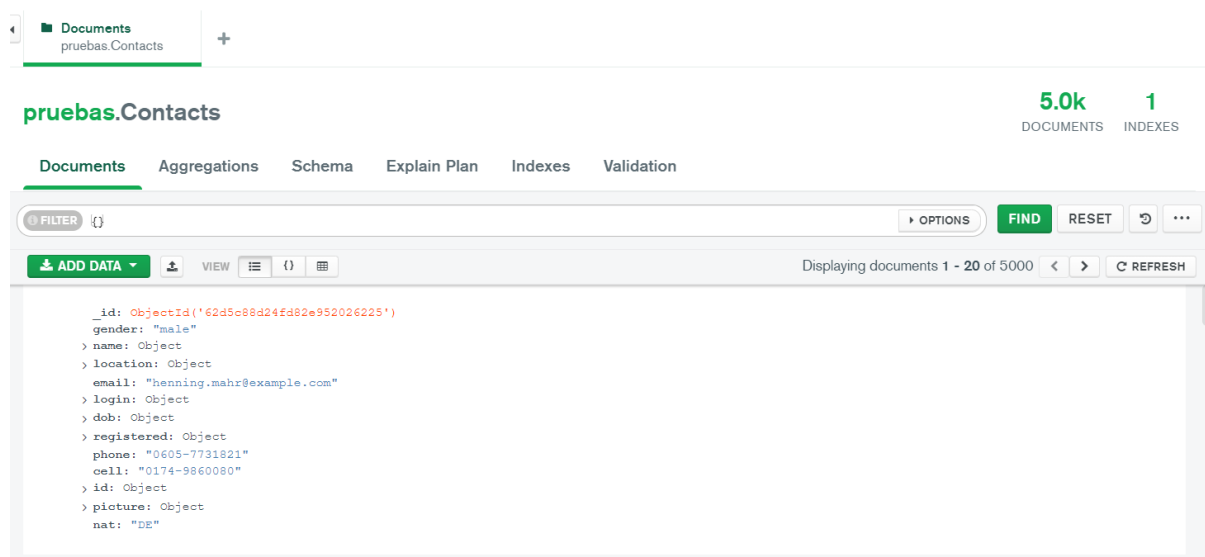
Hacer peticiones al servidor de Mongo desde Mongo Compass & Mongo Shell para hacer los siguientes ejercicios:

- Listar todos los contactos.

Mongo Shell

```
>_MONGOSH
pruebas> db.Contacts.find({}) // Finds all contacts
```

Mongo Compass



- Busca el primer contacto que sea de Alemania (Germany).

Mongo Shell

```
> _MONGOSH
> db.Contacts.findOne({"location.country": "Germany"})
< { _id: ObjectId("62d5c88d24fd82e952026225"),
  gender: 'male',
  name: { title: 'Mr', first: 'Henning', last: 'Mahr' },
  location:
    { city: 'Weißensee',
      state: 'Rheinland-Pfalz',
      country: 'Germany',
      postcode: 10088,
      coordinates: { latitude: '-42.1479', longitude: '-96.5328' },
      timezone: { offset: '+6:00', description: 'Almaty, Dhaka, Colombo' } },
  email: 'henning.mahr@example.com',
  login:
    { uuid: '78c45fa0-1f81-424b-bed3-22e5b18cb52d',
      username: 'ticklishdog227',
      password: 'underwear',
      salt: 'f0FehqUT',
      md5: 'f09dd45293e9bf0811cc90d228815f52',
```

Mongo Compass

pruebas.Contacts

5.0k DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER {"location.country": "Germany"} PROJECT { field: 0 } SORT { field: -1 } or [['field', -1]] COLLATION { Locale: 'simple' } MAX TIME MS 60000 SKIP 0 LIMIT 1

ADD DATA VIEW {}

Displaying documents 1 - 1 of 1 REFRESH

```
{
  "_id": ObjectId('62d5c88d24fd82e952026225'),
  "gender": "male",
  "name": Object,
  "location": Object,
  "email": "henning.mahr@example.com",
  "login": Object,
  "dob": Object,
  "registered": Object,
  "phone": "0605-7731821",
  "cell": "0174-9860080",
  "id": Object,
  "picture": Object,
  "nat": "DE"
}
```

- Busca todos los contactos que tengan Blake como nombre (first).

Mongo Shell

```
> _MONGOSH

pruebas> db.Contacts.find({"name.first": "Blake"}) // Finds all contacts with the first name of Blake
```

Mongo Compass

Documents pruebas.Contacts

pruebas.Contacts

5.0k DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER {"name.first": "Blake"} PROJECT { field: 0 } SORT { field: -1 } or [['field', -1]] COLLATION { Locale: 'simple' } MAX TIME MS 60000 SKIP 0 LIMIT 1

ADD DATA VIEW {}

Displaying documents 1 - 4 of 4 REFRESH

```
{
  "_id": ObjectId('62d5c88d24fd82e952026226'),
  "gender": "male",
  "name": Object,
  "location": Object,
  "email": "henning.mahr@example.com",
  "login": Object,
  "dob": Object,
  "registered": Object,
  "phone": "0605-7731821",
  "cell": "0174-9860080",
  "id": Object,
  "picture": Object,
  "nat": "DE"
}
```

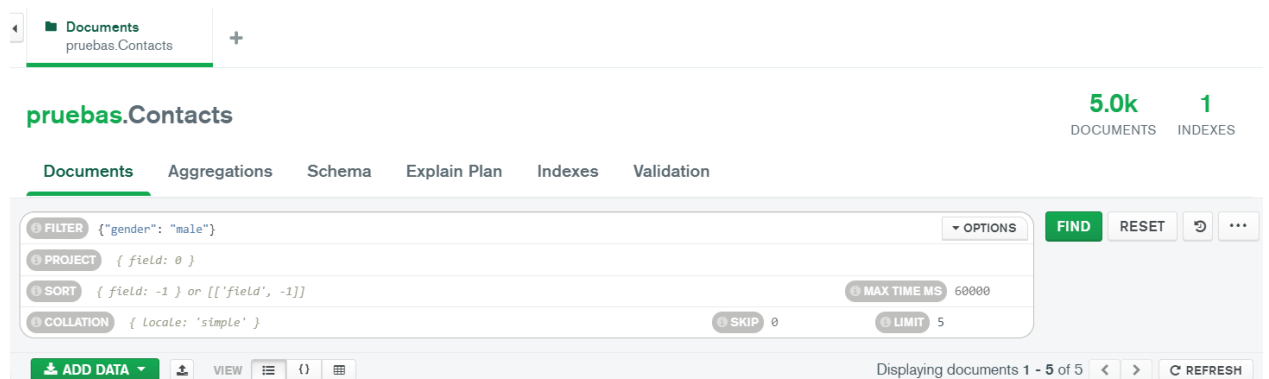
- Busca los primeros 5 contactos que tengan como género (gender) hombre (male)

Mongo Shell

```
>_MONGOSH

pruebas> db.Contacts.find({"gender": "male"}).limit(5) // Finds first 5 male results
```

Mongo Compass



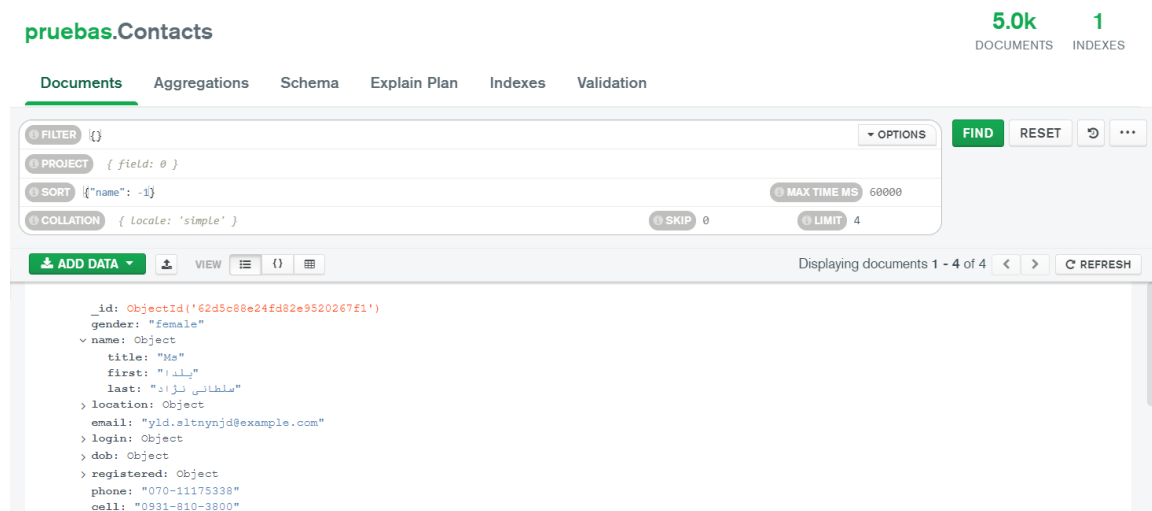
- Devuelve los 4 primeros contactos ordenados por nombre (name) de manera descendente.

Mongo Shell

```
>_MONGOSH

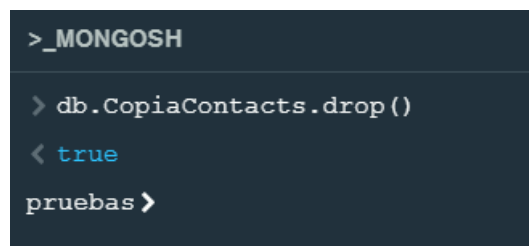
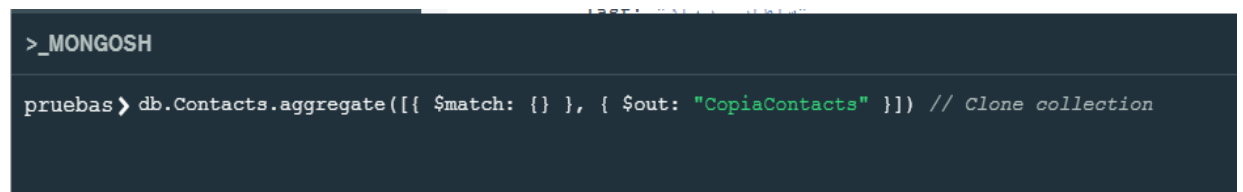
> db.Contacts.find().sort({"name": -1}).limit(4)
< { _id: ObjectId("62d5c88e24fd82e9520267f1"),
  gender: 'female',
  name: { title: 'Ms', first: 'یلدا', last: 'سلطانی نژاد' },
  location:
    { city: 'ساری',
      state: 'قم',
      country: 'Iran',
      postcode: 34101,
      coordinates: { latitude: '51.5272', longitude: '-175.7653' },
      timezone:
        { offset: '+10:00',
          description: 'Eastern Australia, Guam, Vladivostok' } },
  email: 'yld.sltnynjd@example.com',
```

Mongo Compass



- Clona la colección de Contacts a CopiaContacts y luego bórrala.

Mongo Shell



Mongo Compass

Aggregations
pruebas.Contacts

pruebas.Contacts

5.0k DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Pipeline \$out

Explain Export Run More Options

Untitled - modified SAVE CREATE NEW EXPORT TO LANGUAGE

AUTO PREVIEW

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

location: Object
email: "henning.mahr@example.com"
login: Object
dob: Object
registered: Object

location: Object
email: "blake.tremblay@example.com"
login: Object
dob: Object
registered: Object

\$out

Documents will be saved to the collection: 'CopiaContacts'

The \$out operator will cause the pipeline to persist the results to the specified location (collection, S3, or Atlas). If the collection exists it will be replaced.

ADD STAGE

pruebas

Contacts

CopiaContacts

Open in New Tab

Drop Collection

Drop Collection

To drop CopiaContacts type the collection name CopiaContacts.

CopiaContacts

Cancel Drop Collection

- Renombra el campo de name por nombre.

```
>_MONGOSH
> db.Contacts.updateMany({}, {$rename: {name: "nombre"}})
< { acknowledged: true,
    insertedId: null,
    matchedCount: 5000,
    modifiedCount: 5000,
    upsertedCount: 0 }
pruebas>
```

- Borra todos los contactos que tengan como estado (state) Florida.

```
>_MONGOSH
> db.Contacts.deleteMany({"location.state": "Florida"})
< { acknowledged: true, deletedCount: 6 }
pruebas>
```