



UNIVERSITAT<sub>DE</sub>  
BARCELONA

**Módulo I:**

Tarea – Consultas Básicas a Base de Datos no Relacionales con MongoDB

**Estudiante:**

Heiner Romero Leiva

**Máster en Big Data & Data Science**

Noviembre, 2021

## Descripción de la tarea:

Has sido contratado por una consultora como Data Engineer. Se te da acceso a una base de datos no relacional, la cual está montada en un servidor MongoDB.

Dentro de la base de datos existen varias colecciones:

Colección	Descripción	Documentos
companies	Contiene información sobre las grandes empresas de la actualidad.	18801
mails	Contiene emails (asunto, destinatarios, mensaje, etc.).	5929
neighborhoods	Contiene información geo posicionada de barrios/ciudades del Reino Unido.	195
restaurants	Contiene información sobre restaurantes de ciudades de Reino Unido.	25359
zips	Contiene información sobre códigos postales de EEUU.	29467

Debes crear las consultas necesarias para poder responder a las siguientes preguntas:

1. ¿Cuántos restaurantes pertenecen a la ciudad (borough) de “Queens”?
2. ¿Cuántos restaurantes tienen el código postal 11374?
3. ¿Cuántos restaurantes son de tipo “American” o “Bakery”?
4. ¿Cuántos nombres de restaurantes empiezan por la letra “A”?
5. ¿Cuáles es el nombre de las 10 empresas mas antiguas?
6. ¿Cuántas empresas se fundaron durante o después del año 2000?
7. ¿Cuántas empresas tienen entre 500 y 1000 empleados/as?
8. ¿Cuántas empresas no tienen productos informados?
9. ¿En cuántos mails aparecen las palabras “Wall Street Journal” en el asunto del correo?
10. ¿Cuántos correos tienen como remitente un correo con dominio @enron.com?

### Companies:

Activar Windows  
Ve a Configuración para activar Windows

Activar Windows

```

{ is_past: true,
  title: 'Director of Engineering',
  person:
    { first_name: 'Rajendran',
      last_name: 'Dandapani',
      permalink: 'rajendran-dandapani' } },
{ is_past: true,
  title: 'Market Analyst',
  person:
    { first_name: 'Aravind',
      last_name: 'Natarajan',
      permalink: 'aravind-natarajan' } },
{ is_past: true,
  title: 'Director of Product Management',
  person:
    { first_name: 'Hyther',
      last_name: 'Nizam',
      permalink: 'hyther-nizam' } },
{ is_past: true,
  title: 'Western Regional OEM Sales Manager',
  person: { first_name: 'Ian', last_name: 'Wenig', permalink: 'ian-wenig' } } ],
competitions: [],
providerships:
[ { title: 'DHFH',
  is_past: true,
  provider: { name: 'A Small Orange', permalink: 'a-small-orange' } } ],
total_money_raised: '$0',
funding_rounds: [],
investments: [],
acquisition: null,

```

Activar Windows  
Ve a Configuración para activar Windows

```

provider: { name: 'A Small Orange', permalink: 'a-small-orange' } } ],
total_money_raised: '$0',
funding_rounds: [],
investments: [],
acquisition: null,
acquisitions: [],
offices:
[ { description: 'Headquarters',
  address1: '4900 Hopyard Rd.',
  address2: 'Suite 310',
  zip_code: '94588',
  city: 'Pleasanton',
  state_code: 'CA',
  country_code: 'USA',
  latitude: 37.692934,
  longitude: -121.904945 } ],
milestones: [],
video_embeds: [],
screenshots:
[ { available_sizes:
  [ [ 150, 94 ],
    'assets/images/resized/0004/3400/43400v1-max-150x150.png' ],
  [ 250, 156 ],
    'assets/images/resized/0004/3400/43400v1-max-250x250.png' ],
  [ 450, 282 ],
    'assets/images/resized/0004/3400/43400v1-max-450x450.png' ] ],
  attribution: null } ],
external_links: [],
partners: [] ]

```

Activar Windows  
Ve a Configuración para activar Windows

## Query:

db.companies.findOne()



## Mails:

```
> db.mails.findOne()
< { _id: ObjectId("52af48b5d55148fa0c199645"),
  sender: 'rosalee.fleming@enron.com',
  recipients: [ 'rob.bradley@enron.com' ],
  cc: [],
  text: 'Oh, Rob, I\'m happy to do so!! I hope you are on vacation!!\n\nRosie\n\n\n\n\nRob Bradley\n08/04/2000 01:27 PM\nTo: Kenneth Lay/Corp/Enron@ENRON\ncc: \nSubject: Re: Gathering Place Speaking Invitation \n\nGreat--and I will be inspired with my draft of his remarks.\n\nRosie, could you call the phone number on the invitation, ask for Herb \nBateman (he is the director), and tell him of the commitment?\n\nI am out of town and do not have their number, and they called me wondering \nabout Ken\'s decision last Friday.\n\nThanks,\n\n- Rob\n\n\n\n\nKenneth Lay on 08/04/2000 12:40:45 PM\nSent by: Rosalee Fleming\nTo: Rob Bradley/Corp/Enron@ENRON\ncc: Tori L Wells/HOU/ECT@ECT \n\nSubject: Re: Gathering Place Speaking Invitation \n\nHi Rob -\n\nKen said "If you\'ll prepare the remarks, I\'ll be happy to do this".\n\nRosie\n\n\n\n\nRob Bradley\n07/19/2000 09:51 AM\n\nTo: Kenneth Lay/Corp/Enron@ENRON\ncc: \nSubject: Gathering Place Speaking Invitation\n\nIf you are inclined to accept this invitation, we could put together an \ninteresting talk on the importance of good mental health to business with \nsome national statistics and then segue into Enron\'s emphasis on wellness and \nhaving fun in the workplace. Our strategy and experience will be very \ninteresting to the group since it shows what a corporate leader is doing to \nbecome the employer of choice in the marketplace.\n\nYou could expect a full house (around 400) for the talk. If it fits into \nyour schedule, Jack Bowen said he would be delighted to be honorary chairman \nof the event.\n\n- Rob\n\n\n\n\n\n',
  mid: '20840329.1075840285588.JavaMail.evans@thyme',
  fpath: 'enron_mail_20110402/maildir/lay-k/_sent/100.',
  bcc: [],
  to: [ 'rob.bradley@enron.com' ],
  replyto: null,
  ctype: 'text/plain; charset=us-ascii',
  fname: '100.',
  date: '2000-08-04 08:52:00-07:00',
  folder: '_sent',
  subject: 'Re: Gathering Place Speaking Invitation' }
```

Activar Windows  
Ve a Configuración para activar Windows

## Query:

```
db.mails.findOne()
```

## Neighborhoods:

```
> db.neighborhoods.findOne()
< { _id: ObjectId("55cb9c666c522cafdb059a1b"),
  geometry:
    { coordinates:
      [ [ [ -73.94732672160579, 40.62916656720943 ],
        [ -73.94687439946738, 40.626773668128486 ],
        [ -73.94642266064318, 40.6243855672864 ],
        [ -73.94542442897408, 40.624495609598185 ],
        [ -73.94497031348948, 40.622104512937604 ],
        [ -73.94451833370498, 40.61971364709504 ],
        [ -73.94551663574785, 40.6196056119517 ],
        [ -73.94651373193828, 40.61949401914993 ],
        [ -73.9460669961243, 40.61710373918014 ],
        [ -73.9457635491893, 40.61550406422122 ],
        [ -73.94573691912028, 40.61536340675661 ],
        [ -73.94657366828137, 40.61478913113749 ],
        [ -73.94742608780807, 40.61422460055365 ],
        [ -73.94772682569696, 40.614169716182865 ],
        [ -73.94834517370118, 40.61413649663159 ],
        [ -73.94930527478495, 40.614030734516 ],
        [ -73.95026514882962, 40.613924351797365 ],
        [ -73.94999598103774, 40.612491461663225 ],
        [ -73.94996296695325, 40.61231572784935 ],
        [ -73.94993049319106, 40.612144650893164 ],
        [ -73.94978380450677, 40.61137175047764 ],
        [ -73.95070512926118, 40.611250452224034 ],
        [ -73.9516323504994, 40.6111235398617 ],
        [ -73.95256553090685, 40.610987184733915 ],
        [ -73.95258786948972, 40.61111223958901 ],
        [ -73.94732672160579, 40.62916656720943 ] ] ] ],
    type: 'Polygon' },
  name: 'Midwood' }
```

Activar Windows  
Ve a Configuración para activar Windows

```
> .MONGOSH BETA
[ -73.96414862835245, 40.62973896556123 ],
[ -73.96322149343028, 40.62984106103682 ],
[ -73.9623259936459, 40.62996871673756 ],
[ -73.96197454283188, 40.62999504108035 ],
[ -73.96182099849077, 40.63000487740179 ],
[ -73.96163639940383, 40.63002222301838 ],
[ -73.96136921593188, 40.630058388262405 ],
[ -73.96046280639705, 40.630155661219426 ],
[ -73.96040734342054, 40.630162530576236 ],
[ -73.9602585501078, 40.62941826256536 ],
[ -73.96024902920536, 40.62937062183239 ],
[ -73.96022560057911, 40.62925450129709 ],
[ -73.96014773493862, 40.62891518541703 ],
[ -73.95995565121835, 40.627778672540586 ],
[ -73.95899416297213, 40.62788559454311 ],
[ -73.95805293146621, 40.627971118631955 ],
[ -73.95684697280593, 40.62811987905402 ],
[ -73.95581207194951, 40.62823349190697 ],
[ -73.95488499156933, 40.6283355483374 ],
[ -73.95395983418409, 40.62843938961405 ],
[ -73.95303133100138, 40.62854011509266 ],
[ -73.95206901809604, 40.628645781014725 ],
[ -73.95110641920337, 40.628750467094505 ],
[ -73.95017656950532, 40.62885277202806 ],
[ -73.94925229833628, 40.628954765326526 ],
[ -73.94832386196275, 40.62905666053273 ],
[ -73.94732672160579, 40.62916656720943 ] ] ],
type: 'Polygon' },
name: 'Midwood' }
```

Activar Windows  
Ve a Configuración para activar Windows

## Query:

db.neighborhoods.findOne()

## Restaurants:

```
> db.restaurants.findOne()
< { _id: ObjectId("60c07773e937638198ea23ae"),
  address:
    { building: '7114',
      coord: [ -73.9068506, 40.6199034 ],
      street: 'Avenue U',
      zipcode: '11234' },
  borough: 'Brooklyn',
  cuisine: 'Delicatessen',
  grades:
    [ { date: 2014-05-28T00:00:00.000Z, grade: 'A', score: 10 },
      { date: 2014-01-14T00:00:00.000Z, grade: 'A', score: 10 },
      { date: 2013-08-03T00:00:00.000Z, grade: 'A', score: 8 },
      { date: 2012-07-18T00:00:00.000Z, grade: 'A', score: 10 },
      { date: 2012-03-08T00:00:00.000Z, grade: 'A', score: 13 },
      { date: 2011-10-14T00:00:00.000Z, grade: 'A', score: 9 } ],
  name: 'Wilken\'S Fine Food',
  restaurant_id: '40356483' }
```

Activar Windows  
Ve a Configuración para activar Wind

## Query:

```
db.restaurants.findOne()
```

## Zips:

```
> db.zips.findOne()
< { _id: '01001',
  city: 'AGAWAM',
  loc: [ -72.622739, 42.070206 ],
  pop: 15338,
  state: 'MA' }
```

Activar Windows  
Ve a Configuración para activar Wind

## Query:

```
db.zips.findOne()
```

Una vez hecho esto y habiendo visto cuáles son los nombres de los campos, se puede empezar el ejercicio propuesto.

## 1. ¿Cuántos restaurantes pertenecen a la ciudad (borough) de “Queens”?

Para realizar este ejercicio se utilizan dos tipos diferentes de sentencias, una con comillas y con “\$eq”, y otra estándar sin comillas y ambos dan el mismo resultado, que son **5656 restaurantes que pertenecen a la ciudad de Queens**.

Queries:

```
db.restaurants.find({"borough": {"$eq": "Queens"}}).count()
```

```
> db.restaurants.find({"borough": {"$eq": "Queens"}}).count()
< 5656
>
```

Activar Windows  
Ve a Configuración para activar Wind

```
db.restaurants.find({borough : "Queens"}).count()
```

```
> db.restaurants.find({borough: "Queens"}).count()
< 5656
>
```

Activar Windows  
Ve a Configuración para activar Wind

```
db.restaurants.find({borough : "Queens"}).pretty()
```

```
> db.restaurants.find({borough: "Queens"}).pretty()
< { _id: ObjectId("60c07773e937638198ea23b7"),
  address:
    { building: '265-15',
      coord: [ -73.7032601, 40.7386417 ],
      street: 'Hillside Avenue',
      zipcode: '11004' },
  borough: 'Queens',
  cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
  grades:
    [ { date: 2014-10-28T00:00:00.000Z, grade: 'A', score: 9 },
      { date: 2013-09-18T00:00:00.000Z, grade: 'A', score: 10 },
      { date: 2012-09-20T00:00:00.000Z, grade: 'A', score: 13 } ],
  name: 'Carvel Ice Cream',
  restaurant_id: '40361322' }
{ _id: ObjectId("60c07773e937638198ea23bb"),
  address:
    { building: '129-08',
      coord: [ -73.839297, 40.78147 ],
      street: '20 Avenue',
      zipcode: '11356' },
  borough: 'Queens',
  cuisine: 'Delicatessen',
  grades:
    [ { date: 2014-08-16T00:00:00.000Z, grade: 'A', score: 12 },
      { date: 2013-08-27T00:00:00.000Z, grade: 'A', score: 9 },
      { date: 2012-09-20T00:00:00.000Z, grade: 'A', score: 7 },
      { date: 2011-09-29T00:00:00.000Z, grade: 'A', score: 10 } ],
```



## 2. ¿Cuántos restaurantes tienen el código postal 11374?

Para este caso, también se hace una comprobación usando el comando “\$eq” y sin él, dando el mismo resultado; es decir, **88 restaurantes tienen el código 11374**, luego se hace una comprobación con la consulta para verificar que efectivamente retorna un resultado válido con el código 11374 con el que estamos filtrando.

```
db.restaurants.find({"address.zipcode": "11374"}).count()
```

```
> db.restaurants.find({"address.zipcode": "11374"}).count()
< 88
```

```
db.restaurants.find({"address.zipcode": {"$eq": "11374"}}).count()
```

```
> db.restaurants.find({"address.zipcode": {"$eq": "11374"}}).count()
< 88
```

Activar Windows

Ve a Configuración para activar Wind

```
db.restaurants.find({"address.zipcode": "11374"})
```

```
> db.restaurants.find({"address.zipcode": "11374"})
< { _id: ObjectId("60c07773e937638198ea23c4"),
  address:
    { building: '97-22',
      coord: [ -73.8601152, 40.7311739 ],
      street: '63 Road',
      zipcode: '11374' },
  borough: 'Queens',
  cuisine: 'Jewish/Kosher',
  grades:
    [ { date: 2014-11-24T00:00:00.000Z, grade: 'Z', score: 20 },
      { date: 2013-01-17T00:00:00.000Z, grade: 'A', score: 13 },
      { date: 2012-08-02T00:00:00.000Z, grade: 'A', score: 13 },
      { date: 2011-12-15T00:00:00.000Z, grade: 'B', score: 25 } ],
  name: 'Tov Kosher Kitchen',
  restaurant_id: '40356068' }
{ _id: ObjectId("60c07773e937638198ea23f6"),
  address:
    { building: '96-40',
      coord: [ -73.86137149999999, 40.7293762 ],
      street: 'Queens Boulevard',
      zipcode: '11374' },
  borough: 'Queens',
  cuisine: 'Jewish/Kosher',
  grades:
    [ { date: 2014-03-13T00:00:00.000Z, grade: 'A', score: 11 },
      { date: 2013-09-30T00:00:00.000Z, grade: 'A', score: 9 },
      { date: 2013-04-26T00:00:00.000Z, grade: 'A', score: 8 },
      { date: 2012-09-11T00:00:00.000Z, grade: 'B', score: 24 },
      { date: 2011-09-19T00:00:00.000Z, grade: 'A', score: 10 },
```

Activar Windows

Ve a Configuración para activar Wind

### 3. ¿Cuántos restaurantes son de tipo “American” o “Bakery”?

La cantidad de restaurantes que son de tipo “American” o “Bakery” es de 6874, de igual forma se coteja el resultado para verificar que efectivamente la consulta está tomando en cuenta “American” como “Bakery” tal y como se muestra a continuación.

```
db.restaurants.find({$or: [{cuisine: "American"}, {cuisine: "Bakery"}]}).count()
```

```
> db.restaurants.find({$or: [{"cuisine": "American"}, {"cuisine": "Bakery"}]}).count()
< 6874
```

Activar Windows  
Ve a Configuración para activar Wind

```
db.restaurants.find({$or: [{cuisine: "American"}, {cuisine: "Bakery"}]}).count()
```

```
> db.restaurants.find({$or: [{cuisine: "American"}, {"cuisine": "Bakery"}]}).count()
< 6874
```

Activar Windows  
Ve a Configuración para activar Wind

```
db.restaurants.find({$or: [{cuisine: "American"},
{cuisine: "Bakery"}]}).sort({cuisine: -1}).limit(2)
```

```
> db.restaurants.find({$or: [{cuisine: "American"}, {"cuisine": "Bakery"}]}).sort({cuisine: -1}).limit(2)
< { _id: ObjectId("60c07773e937638198ea23dd"),
  address:
    { building: '120',
      coord: [ -73.9998042, 40.7251256 ],
      street: 'Prince Street',
      zipcode: '10012' },
  borough: 'Manhattan',
  cuisine: 'Bakery',
  grades:
    [ { date: 2014-10-17T00:00:00.000Z, grade: 'A', score: 11 },
      { date: 2013-09-18T00:00:00.000Z, grade: 'A', score: 13 },
      { date: 2013-04-30T00:00:00.000Z, grade: 'A', score: 7 },
      { date: 2012-04-20T00:00:00.000Z, grade: 'A', score: 7 },
      { date: 2011-12-19T00:00:00.000Z, grade: 'A', score: 3 } ],
  name: 'Olive'S',
  restaurant_id: '40363151' }
{ _id: ObjectId("60c07773e937638198ea23c0"),
  address:
    { building: '1007',
      coord: [ -73.856077, 40.848447 ],
      street: 'Morris Park Ave',
      zipcode: '10462' },
  borough: 'Bronx',
  cuisine: 'Bakery',
  grades:
    [ { date: 2014-03-03T00:00:00.000Z, grade: 'A', score: 2 },
      { date: 2013-09-11T00:00:00.000Z, grade: 'A', score: 6 },
      { date: 2013-01-24T00:00:00.000Z, grade: 'A', score: 10 },
```

Activar Windows  
Ve a Configuración para activar Wind

```
db.restaurants.find({$or: [{cuisine: "American"},  
  
  {cuisine: "Bakery"}]}).limit(7)
```

```
> db.restaurants.find({$or: [{cuisine: "American"}, {cuisine: "Bakery"}]}).limit(7)  
< { _id: ObjectId("60c07773e937638198ea23af"),  
  address:  
    { building: '6409',  
      coord: [ -74.00528899999999, 40.628886 ],  
      street: '11 Avenue',  
      zipcode: '11219' },  
  borough: 'Brooklyn',  
  cuisine: 'American',  
  grades:  
    [ { date: 2014-07-18T00:00:00.000Z, grade: 'A', score: 12 },  
      { date: 2013-07-30T00:00:00.000Z, grade: 'A', score: 12 },  
      { date: 2013-02-13T00:00:00.000Z, grade: 'A', score: 11 },  
      { date: 2012-08-16T00:00:00.000Z, grade: 'A', score: 2 },  
      { date: 2011-08-17T00:00:00.000Z, grade: 'A', score: 11 } ],  
  name: 'Regina Caterers',  
  restaurant_id: '40356649' }  
{ _id: ObjectId("60c07773e937638198ea23b1"),  
  address:  
    { building: '2300',  
      coord: [ -73.8786119, 40.8502883 ],  
      street: 'Southern Boulevard',  
      zipcode: '10460' },  
  borough: 'Bronx',  
  cuisine: 'American',  
  grades:  
    [ { date: 2014-05-28T00:00:00.000Z, grade: 'A', score: 11 },  
      { date: 2013-06-19T00:00:00.000Z, grade: 'A', score: 4 },  
      { date: 2012-06-15T00:00:00.000Z, grade: 'B', score: 3 } ],
```

Activar Windows  
Ve a Configuración para activar Wind

#### 4. ¿Cuántos nombres de restaurantes empiezan por la letra “A”?

La cantidad de restaurantes que empiezan con la letra “A” es de 1126. De igual forma se adjunta captura de la consulta de verificación.

```
db.restaurants.find({"name": /^A/}).count()
```

```
> db.restaurants.find({"name": /^A/}).count()
< 1126
```

```
db.restaurants.find({"name": /^A/}, {"name": 1})
```

```
> db.restaurants.find({"name": /^A/}, {"name": 1})
< { _id: ObjectId("60c07773e937638198ea23c7"),
  name: 'Angelika Film Center' }
{ _id: ObjectId("60c07773e937638198ea2413"), name: 'Arturo'S' }
{ _id: ObjectId("60c07773e937638198ea2417"),
  name: 'Angelo Of Mulberry St.' }
{ _id: ObjectId("60c07773e937638198ea2483"),
  name: 'African Terrace' }
{ _id: ObjectId("60c07773e937638198ea2484"),
  name: 'Amore Pizzeria & Restaurant' }
{ _id: ObjectId("60c07773e937638198ea2487"),
  name: 'African Market (Baboon Cafe)' }
{ _id: ObjectId("60c07773e937638198ea24b1"),
  name: 'Armando'S Pizza' }
{ _id: ObjectId("60c07773e937638198ea24fd"),
  name: 'Aunt Bella'S Rest Of Little Neck' }
{ _id: ObjectId("60c07773e937638198ea2514"),
  name: 'Astro Restaurant' }
{ _id: ObjectId("60c07773e937638198ea2544"),
  name: 'Agra Restaurant' }
{ _id: ObjectId("60c07773e937638198ea2545"), name: 'Arturo'S' }
{ _id: ObjectId("60c07773e937638198ea255e"),
  name: 'American Museum Of Natural History Food Court' }
{ _id: ObjectId("60c07773e937638198ea255f"),
  name: 'Amsterdam Restaurant & Tapas Lounge' }
{ _id: ObjectId("60c07773e937638198ea2581"),
  name: 'Andrews Coffee Shop' }
{ _id: ObjectId("60c07773e937638198ea2593"),
  name: 'Aqueduct North' }
```

Activar Windows  
Ve a Configuración para activar Win

## 5. ¿Cuáles es el nombre de las 10 empresas mas antiguas?

Se adjunta captura de pantalla con el nombre de las 10 empresas ordenadas en forma ascendente con respecto al “founded\_year”.

```
db.companies.find({ "founded_year": { "$ne": null } },  
  
    { "name": 1, "founded_year": 1, "_id": 0 }  
  
).sort({ "founded_year": 1 }).limit(10)
```

```
> db.companies.find({"founded_year": {"$ne": null}}, {"name": 1, "founded_year": 1, "_id": 0}).sort({"founded_year": 1}).limit(10)  
< { name: 'SmallWorlds', founded_year: 1800 }  
  { name: 'US Army', founded_year: 1800 }  
  { name: 'Altrasoft', founded_year: 1800 }  
  { name: 'DuPont', founded_year: 1802 }  
  { name: 'Bachmann Industries', founded_year: 1833 }  
  { name: 'McKesson', founded_year: 1833 }  
  { name: 'Bertelsmann', founded_year: 1835 }  
  { name: 'Accuity', founded_year: 1836 }  
  { name: 'CENTRA', founded_year: 1839 }  
  { name: 'WeGame', founded_year: 1840 }  
>
```



## 6. ¿Cuántas empresas se fundaron durante o después del año 2000?

**La cantidad de empresas que se fundaron durante o después del año 2000 asciende a 10742.** Se adjunta captura de la verificación de la consulta tanto a nivel del límite inferior (2000) como del límite superior (más alejadas del año 2000).

```
db.companies.find({"founded_year": {"$gte": 2000}}).count()
```

```
> db.companies.find({"founded_year": {"$gte": 2000}}).count()
< 10742
>
```

```
db.companies.find({"founded_year": {"$gte": 2000}}, {"name": 1, "founded_year": 1, "_id": 0}).sort({"founded_year": -1}).limit(10)
```

```
> db.companies.find({"founded_year": {"$gte": 2000}}, {"name": 1, "founded_year": 1, "_id": 0}).sort({"founded_year": -1}).limit(10)
< { name: 'Wamba', founded_year: 2013 }
  { name: 'Gimigo', founded_year: 2013 }
  { name: 'Clowdy', founded_year: 2013 }
  { name: 'Fixya', founded_year: 2013 }
  { name: 'Fluc', founded_year: 2013 }
  { name: 'SEOGroup', founded_year: 2013 }
  { name: 'Pikk', founded_year: 2013 }
  { name: 'WhosCall', founded_year: 2013 }
  { name: 'iBazar', founded_year: 2013 }
  { name: 'Advaliant', founded_year: 2013 }
>
```

```
db.companies.find({"founded_year": {"$gte": 2000}}, {"name": 1, "founded_year": 1, "_id": 0}).sort({"founded_year": 1}).limit(10)
```

```
> db.companies.find({"founded_year": {"$gte": 2000}}, {"name": 1, "founded_year": 1, "_id": 0}).sort({"founded_year": 1}).limit(10)
< { name: 'Steorn', founded_year: 2000 }
  { name: 'SelectMinds', founded_year: 2000 }
  { name: 'ContextWeb', founded_year: 2000 }
  { name: 'PhotoBox', founded_year: 2000 }
  { name: 'AllofMP3', founded_year: 2000 }
  { name: 'Boonex', founded_year: 2000 }
  { name: 'eHarmony', founded_year: 2000 }
  { name: 'ZoomInfo', founded_year: 2000 }
  { name: 'Sulake', founded_year: 2000 }
  { name: 'MeeVee', founded_year: 2000 }
>
```

## 7. ¿Cuántas empresas tienen entre 500 y 1000 empleados/as?

**La cantidad de empresas asciende a 144.** Se adjuntan capturas del límite inferior (más cercanas o iguales a 500) y del límite superior (más cercanas o iguales a 1000).

```
db.companies.find({"number_of_employees": {"$gte": 500, "$lte": 1000}}).count()
```

```
> db.companies.find({"number_of_employees": {"$gte": 500, "$lte": 1000}}).count()
< 144
>
```

```
db.companies.find({"number_of_employees": {"$gte": 500, "$lte": 1000}}, {"number_of_employees": 1, "name": 1, "_id": 0}).sort({"number_of_employees": 1}).limit(10)
```

```
> db.companies.find({"number_of_employees": {"$gte": 500, "$lte": 1000}}, {"number_of_employees": 1, "name": 1, "_id": 0}).sort({"number_of_employees": 1}).limit(10)
< { name: 'Endeca', number_of_employees: 500 }
  { name: 'XING', number_of_employees: 500 }
  { name: 'Adconion Media Group', number_of_employees: 500 }
  { name: 'Trulia', number_of_employees: 500 }
  { name: 'Alipay', number_of_employees: 500 }
  { name: 'QuinStreet', number_of_employees: 500 }
  { name: 'SilkRoad Technology', number_of_employees: 500 }
  { name: 'Infoblox', number_of_employees: 500 }
  { name: 'Tesla Motors', number_of_employees: 500 }
  { name: 'Splunk', number_of_employees: 500 }
>
```

```
db.companies.find({"number_of_employees": {"$gte": 500, "$lte": 1000}}, {"number_of_employees": 1, "name": 1, "_id": 0}).sort({"number_of_employees": -1}).limit(10)
```

```
> db.companies.find({"number_of_employees": {"$gte": 500, "$lte": 1000}}, {"number_of_employees": 1, "name": 1, "_id": 0}).sort({"number_of_employees": -1}).limit(10)
< { name: 'Telenav', number_of_employees: 1000 }
  { name: 'ERG Transit Systems', number_of_employees: 1000 }
  { name: 'Yatra online', number_of_employees: 1000 }
  { name: 'Sonus Networks', number_of_employees: 1000 }
  { name: 'Yodle', number_of_employees: 1000 }
  { name: 'Antal International', number_of_employees: 1000 }
  { name: 'Gumtree', number_of_employees: 1000 }
  { name: 'SK Net Service Company Ltd',
    number_of_employees: 1000 }
  { name: 'Omniure', number_of_employees: 1000 }
  { name: 'Akamai Technologies', number_of_employees: 1000 }
>
```

## 8. ¿Cuántas empresas no tienen productos informados?

**La cantidad de empresas sin productos informados asciende a 13282.** De igual forma se adjunta captura de pantalla con la comprobación.

```
db.companies.find({"products": { "$eq": [] }}).count()
```

```
> db.companies.find({"products": {"$eq": []}}).count()
< 13282
>
```

```
db.companies.find({"products": [] }).count()
```

```
> db.companies.find({"products": []}).count()
< 13282
>
```

```
db.companies.find({"products": { "$eq": [] }}, {"products": 1, "name": 1, "_id": 0}).limit(10)
```

```
> db.companies.find({"products": {"$eq": []}}, {"products": 1, "name": 1, "_id": 0}).limit(10)
< { name: 'AdventNet', products: [] }
  { name: 'Fox Interactive Media', products: [] }
  { name: 'Scribd', products: [] }
  { name: 'StumbleUpon', products: [] }
  { name: 'Viacom', products: [] }
  { name: 'JotSpot', products: [] }
  { name: 'Gannett', products: [] }
  { name: 'NetRatings', products: [] }
  { name: 'Meetup', products: [] }
  { name: 'YouTube', products: [] }
>
```

**9. ¿En cuántos mails aparecen las palabras “Wall Street Journal” en el asunto del correo?**

**Dicha palabra aparece 4 veces.** Se adjuntan capturas de pantalla con la comprobación; además se adjunta una captura sin el comando “\$regex” para verificar si habían resultados, pero dichas palabras aparecen con la composición “Millennium Forum”, por ello aparece como “0”.

```
db.mails.find({"subject": {"$regex": "Wall Street Journal"}}).count()
```

```
> db.mails.find({"subject": {"$regex": "Wall Street Journal"}}).count()
< 4
>
```

```
db.mails.find({"subject": { "$regex": "Wall Street Journal"}}, {"sender": 1, "subject": 1,
"_id": 0}).limit(10)
```

```
db.mails.find({"subject": {"$regex": "Wall Street Journal"}}, {"sender": 1, "subject": 1, "_id": 0}).limit(10)
```

```
{ sender: 'tori.wells@enron.com',
  subject: 'Wall Street Journal Millennium Forum' }
{ sender: 'tori.wells@enron.com',
  subject: 'Wall Street Journal Millennium Forum' }
{ sender: 'tori.wells@enron.com',
  subject: 'Wall Street Journal Millennium Forum' }
{ sender: 'tori.wells@enron.com',
  subject: 'Wall Street Journal Millennium Forum' }
```

```
db.mails.find({"subject": "Wall Street Journal"}).count()
```

```
> db.mails.find({"subject": "Wall Street Journal"}).count()
< 0
>
```

## 10. ¿Cuántos correos tienen como remitente un correo con dominio @enron.com?

La cantidad de correos asciende a los **2431**. Se adjuntan capturas de pantalla con la verificación y además, en caso de querer saber cuántos de esos correos asociados a remitentes son únicos se debe usar el comando “distinct”.

```
db.mails.find({"sender": {"$regex": "@enron.com"}}).count()
```

```
> db.mails.find({"sender": {"$regex": "@enron.com"}}).count()
< 2431
>
```

```
db.mails.find({"sender": {"$regex": "@enron.com"}}, {"sender": 1, "subject": 1, "_id": 0}).limit(10)
```

```
> db.mails.find({"sender": {"$regex": "@enron.com"}}, {"sender": 1, "subject": 1, "_id": 0}).limit(10)
< { sender: 'rosalee.fleming@enron.com',
  subject: 'Re: Gathering Place Speaking Invitation' }
  { sender: 'tori.wells@enron.com',
  subject: 'Wall Street Journal Millennium Forum' }
  { sender: 'rosalee.fleming@enron.com',
  subject: 'KLLA's e-mail address' }
  { sender: 'rosalee.fleming@enron.com',
  subject: 'Re: EXECUTIVE COMMITTEE MEETINGS - MONDAY, JANUARY 17' }
  { sender: 'tori.wells@enron.com',
  subject: 'Re: Sports Arena Deal' }
  { sender: 'rosalee.fleming@enron.com',
  subject: 'Re: Friday, September 29, 2000' }
  { sender: 'rosalee.fleming@enron.com',
  subject: 'Re: Phone Call' }
  { sender: 'rosalee.fleming@enron.com',
  subject: 'Re: World Economic Forum, Melbourne, Australia' }
  { sender: 'rosalee.fleming@enron.com',
  subject: 'Re: EXECUTIVE COMMITTEE MEETING - MONDAY, AUGUST 21' }
  { sender: 'rosalee.fleming@enron.com',
  subject: 'Re: Committee to Encourage Corporate Philanthropy' }
>
```

```
db.mails.distinct("sender", {"sender": {"$regex": "@enron.com"}})
```

```
> db.mails.distinct("sender", {"sender": {"$regex": "@enron.com"}})
< [ '.elizondo@enron.com',
  '.palmer@enron.com',
  '40enron@enron.com',
  'a..davis@enron.com',
  'a..hughes@enron.com',
  'a..lindholm@enron.com',
  'a..schroeder@enron.com',
  'a..shankman@enron.com',
  'aaron.berutti@enron.com',
  'administration.enron@enron.com',
  'adnan.patel@enron.com',
  'adriana.wynn@enron.com',
  'agatha.tran@enron.com',
  'amanda.day@enron.com',
  'amita.gosalia@enron.com',
  'amy.lee@enron.com',
  'anastasia.acurik@enron.com',
  'andrea.yowman@enron.com',
  'andrew.fastow@enron.com',
  'andrew.henderson@enron.com',
  'andrew.kosnaski@enron.com',
  'andrew.parsons@enron.com',
  'andrew.wu@enron.com',
  'andy.blanchard@enron.com',
```

----- FIN -----