|  |
| --- |
| rectangle transparent colorérectangle coloréUne image contenant texte  Description générée automatiquementrectangle transparent coloré  RAPPORT  TP ElasticSearch |

|  |
| --- |
| NoSQL - DIA2  MANNAI Hasna ()  CHENIK Yassine ()  BOUCHIBA Emine () |

Table des matières

Importation du ficher json en utilisant Kibana 2

Simple Queries 3

Complex Queries 9

Hard Queries 11

|  |
| --- |
| Importation du json dans MongoDB |

Lancement de Docker :

Lancer Docker et le container ElasticSearch.

Vérification du statut de Kibana et ElasticSearch :

* ElasticSearch :
  + <http://localhost:9200>
* Kibana :
  + <http://localhost:5601>

Importation du fichier JSON :

* Menu/Kibana/Index Patterns
* « Upload File »
* And we drag and drop our file « companies.json ».

|  |
| --- |
| Simple Queries |

1. Get all the companies that have a category\_code=”nanotech” :

* Code :

GET /companies/\_search

{

"query": {

"match": {

"category\_code": "nanotech"

}

}

}

* *Ici, on veut afficher les entreprises qui sont dans le domaine de la nanotechnologie en affichant le "name", le "category\_code" ainsi que l’id.*
* *Output :*

Une image contenant texte, tableau blanc

Description générée automatiquement

|  |
| --- |
| Simple Queries |

1. Get all the Companies that founded in 2008 :

* Code :

GET /companies/\_search

{

"\_source": ["name", "founded\_year", "founded\_month", "founded\_day"],

"query": {

"bool": {

"must": [

{"range": {

"founded\_year": {"gte": 2008, "lte": 2008}

}}

]

}

}

}

* *Ici, on veut afficher les entreprises qui ont été fondés en 2008. en affichant le "name", le "founded\_year", le "founded\_month", le "founded\_day" ainsi que l’id.*
* *Output :*

*Une image contenant texte, tableau blanc

Description générée automatiquement*

|  |
| --- |
| Simple Queries |

1. Get the number of Companies whith more than 100 employees :

* Code :

GET /companies/\_search

{

"\_source": ["name", "number\_of\_employees"],

"query": {

"bool": {

"must": [

{"range": {

"number\_of\_employees": {"gte": 100}

}}

]

}

}

}

* *Ici, on veut afficher les entreprises qui ont plus de 100 employés en affichant le "name", le "number\_of\_employees" ainsi que l’id.*
* *Output :*

*Une image contenant texte

Description générée automatiquement*

|  |
| --- |
| Simple Queries |

1. Get the total amount raised per companies where category\_code= "security" :

* Code :

GET /companies/\_search

{

"query": {

"match": {

"category\_code": "security"

}

},

"aggs": {

"companies": {

"terms": {

"field": "name"

},

"aggs": {

"total\_raised": {

"sum": {

"field": "funding\_rounds.raised\_amount"

}

}

}

}

}

, "size": 1

, "\_source": ["name"]

}

|  |
| --- |
| Simple Queries |

* *Ici nous cherchons à afficher le montant total amassé par les entreprises qui travaillent dans le domaine de la sécurité.*
* *Output :*

*Une image contenant texte, tableau blanc

Description générée automatiquement*

|  |
| --- |
| Simple Queries |

1. Get total money raised for each domain :

* Code :

GET /companies/\_search

{

"aggs": {

"category\_code": {

"terms": {

"field": "category\_code"

},

"aggs": {

"total\_raised": {

"sum": {

"field": "funding\_rounds.raised\_amount"

}

}

}

}

}

, "size": 1

, "\_source": ["category\_code"]

}

|  |
| --- |
| Simple Queries |

* *Ici, nous cherchons à afficher le montant total amassé par les entreprises du même domaine.*
* *Output :*

*Une image contenant texte, tableau blanc

Description générée automatiquement*

|  |
| --- |
| Simple Queries |

1. Get all companies that starts with an “F” :

* Code :

GET /companies/\_search

{

"query": {

"prefix": {

"name": "F"

}

},

"\_source": ["name"]

}

* *Ici, on cherche à afficher toutes les entreprises qui commencent par la lettre F. Pour ce faire, on utilise l’option $regex qui nous permet de reconnaître un motif dans une chaîne de caractères et on lui donne comme argument ‘^F’. De et on affiche les noms des entreprises correspondantes.*
* *Output :*

*Une image contenant texte, tableau blanc

Description générée automatiquement*

|  |
| --- |
| Complex Queries |

1. Get the number of employees of each Company that have as category\_code =”software” :

* Code :

GET /companies/\_search

{

"query": {

"match": {

"category\_code": "software"

}

},

"aggs": {

"total\_raised": {

"sum": {

"field": "funding\_rounds.raised\_amount"

}

},

"group\_by\_name": {

"terms": {

"field": "name.keyword",

"size": 10

},

"aggs": {

"employee\_count": {

"max": {

"field": "number\_of\_employees"

}

}

}

}

},

"\_source": ["name", "number\_of\_employees","category\_code"],

"size": 10

}

|  |
| --- |
| Complex Queries |

* *Ici, on cherche à avoir le nombre d'employés travaillant au sein d’entreprises dans le domaine du Software et on affiche le nom de l’entreprise, sa category\_code et le nombre d’employés.*
* *Output :*

*Une image contenant texte

Description générée automatiquement*

|  |
| --- |
| Complex Queries |

1. Get the number of products grouped by companies name :

* Code :

GET /companies/\_search

{

"aggs": {

"company": {

"terms": {

"field": "name"

},

"aggs": {

"Number\_of\_Products": {

"sum": {

"field": "products"

}

}

}

}

}

, "size": 1

, "\_source": ["category\_code"]

}

|  |
| --- |
| Complex Queries |

* *Ici, on veut afficher le nombre de produits provenant de chacune des entreprises.*
* *Output :*

*Une image contenant texte

Description générée automatiquement*

|  |
| --- |
| Hard Queries |

1. Get the total amount of money raised by companies founded in each year, sorted in descending order of the total amount raised :

* db.TD.aggregate([{"$match":{"founded\_year":{"$ne":null}}},{"$unwind":"$funding\_rounds"},{"$group":{"\_id":{"year":"$founded\_year","name":"$name"},

|  |
| --- |
| Hard Queries |

"total\_raised\_amount":{"$sum":"$funding\_rounds.raised\_amount"}}},{"$group":{"\_id":"$\_id.year","total\_raised\_amount":{"$sum":{"$toDouble":"$total\_raised\_amount"}}}},{"$sort":{"total\_raised\_amount":-1}}])

* *Ici, on cherche à obtenir le montant total d'argent collecté par les entreprises fondées chaque année, trié par ordre décroissant du montant total collecté.*
* *Output :*

*Une image contenant texte

Description générée automatiquement*