UNIVERSITÉ DE NANTES

Rapport de Base de données évoluées

Aniss BENTEBIB, Camille-Amaury JUGE, Aya HAITI, Clément ANTHEAUME

Respectivement M1 Informatique ATAL et DS, ALMA et DS.

Table des matières

1. Introduction

Rappelons brièvement le contexte, ce projet a pour but de créer un entrepôt de données qui doit permettre par la suite de répondre à un contexte fixé par les étudiants. Pour cela, le dataset choisi doit évidemment répondre à des contraintes réalistes et s'adapter aux nouvelles bases de données sur le marché.

Dans ce cadre, nous avons donc choisi d'étudier les accidents liés au trafic routier aux Etats-Unis entre 2016 et décembre 2019. Ce dataset est fourni par des chercheurs [1] et possède un peu moins de 3 millions de lignes. Ainsi, dans la limite de nos machines personnelles, cela représente une base de données correctement fournie et représentative des base de données des entreprises.

Nous nous sommes alors mis dans le contexte d'analystes de données, de chercheurs afin de comprendre quels sont les différents axes, heure de la journée et autres facteurs pouvant influer sur les accidents et comment peut-on prétendre à les réduire une fois que nous avons la connaissance de ces données.

Pour cela, nous vous joignons le lien du github qui vous permettra d'obtenir la procédure d'installation en local de notre environnement. Nous avons choisi de stocker nos données dans un entrepôt MongoDB [2] avec l'interface d'administration NoSQLBooster[8] et Studio 3T [3]. Pour la partie interprétation et visualisation, nous utiliserons un outil de Microsoft Platform: le Microsoft Power BI Desktop [9]. Pour plus de détails, référez-vous au dépôt Github suivant :

https://github.com/camilleAmaury/BDD_Evoluees

2. La base de données en profondeur

On remarque que notre dataset est composé de 49 colonnes avec des types très variables (booléens, Chaînes de caractères catégoriques, nombres flottants, nombres entiers, …). De plus, nous possédons des données d'indication géographique (latitude, longitude, ville, pays, …), des données valuées (température, longueur de route, humidité, …) qui vont nous permettre d'effectuer des regroupements et des calculs d'indicateurs.

Ainsi, maintenant que nous avons pris connaissance de la base en elle-même, nous allons justifier notre choix concernant l'utilisation de MongoDB. Etant donné que ce Système de Gestion de Base de Données (SGBD) est spécialisé pour une approche document (Le plus possible dans un unique objet appelé le document), notre collection se prête plutôt bien à cette méthode de traiter les données puisque initialement celles-ci sont déjà présentées sous la forme d'un CSV comprenant un accident et tout ses détails par ligne. Tout peut être regroupé dans un unique document

référencé comme étant un accident. MongoDB a la capacité de gérer efficacement ce type de représentation en dénormalisant les schémas SQL classiques.

Par exemple, le lieu (ville et pays) de l'accident aurait pu appartenir à une table indépendante étant donné que ce même lieu puisse se retrouver dans plusieurs accidents. Mais les jointures coûtent particulièrement plus cher en MongoDB qu'en SQL (même si elles ont un coût non négligeable en SQL aussi). Ainsi, il est plus avantageux de tout réunir sous un même document.
Un problème que cela peut poser : si il n'y a pas de table référençant les lieux, on peut imaginer des fautes de saisie sur des enregistrements qui mèneront par la suite à ne pas avoir toutes les données souhaitées.

Finalement, nous avons choisi une approche NoSQL au vu du nombre de données, même si les gros SGBD comme ORACLE ou PostGreSQL aurait pu faire l'affaire pour 3 millions d'enregistrements, il est plus viable niveau performance de préférer l'approche MongoDB. Voici donc la table décrivant nos données. Celle-ci est unique car nos données reposent principalement sur des agrégats, et non sur une structure complexe, chaque document contient une valeur associé à chaque clé dans cette table. Toutes les clés sont au même niveau dans le document.

3. Requêtes NoSQL pertinentes

Dans les requêtes suivantes, nous utilisons la fonction "aggregate ()" qui permet de spécifier des chaînes d'opérations connu sous le nom de pipeline d'agrégation. On a utilisé un outil de Business Intelligence afin de présenter le résultat des requêtes.

Les requêtes suivantes sont faciles à examiner avec moins de détails à partir de la page:

En ce qui concerne les requêtes réalisées sur NoSQLBooster for MongoDB pour celles-ci ainsi que leur résultat en format json sont consultables à partir du lien suivant: https://github.com/camilleAmaury/BDD_Evoluees/tree/master/Requests, tandis que les requêtes réalisées sur Studio 3T sont consultable sur la page dont le lien est le suivant: https://github.com/camilleAmaury/BDD_Evoluees/blob/master/requests.js.

Requête 1 :

L'Objectif:

L'objectif de cette requête est de recenser le nombre d'accidents enregistrés par état dans l'intervalle de temps du Dataset (entre Février 2016 et Décembre 2019).

La Requête:

```
1  db.getCollection("US_Accidents_Dec19").aggregate([{$group: { _id: "$State", countA: { $sum: 1}}},{$sort:{'countA':-1}}]);
```

Le Résultat:

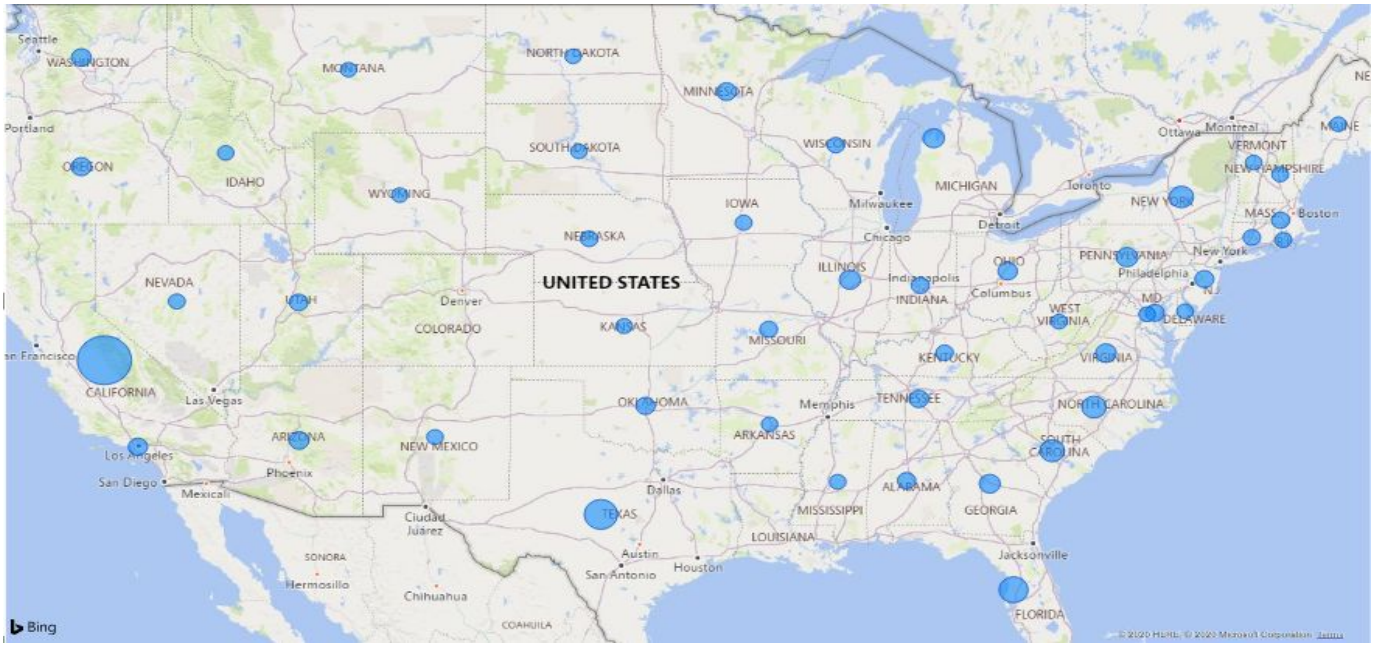
US_Accidents_Dec19 | 13.571 s | 49 Docs | 200 | p. 1

Key	Value	Type
(1) CA	{ _id : "CA", countA : 663204 }	Document
(2) TX	{ _id : "TX", countA : 298062 }	Document
(3) FL	{ _id : "FL", countA : 223746 }	Document
(4) SC	{ _id : "SC", countA : 146689 }	Document
(5) NC	{ _id : "NC", countA : 142460 }	Document
(6) NY	{ _id : "NY", countA : 137799 }	Document
(7) PA	{ _id : "PA", countA : 90395 }	Document
(8) MI	{ _id : "MI", countA : 88694 }	Document
(9) IL	{ _id : "IL", countA : 86390 }	Document
(10) GA	{ _id : "GA", countA : 83620 }	Document
(11) VA	{ _id : "VA", countA : 79957 }	Document
(12) OR	{ _id : "OR", countA : 70840 }	Document
(13) MN	{ _id : "MN", countA : 62727 }	Document
(14) AZ	{ _id : "AZ", countA : 62330 }	Document
(15) WA	{ _id : "WA", countA : 61367 }	Document
(16) TN	{ _id : "TN", countA : 58289 }	Document
(17) OH	{ _id : "OH", countA : 55863 }	Document
(18) LA	{ _id : "LA", countA : 52481 }	Document

La Représentation du résultat:

Afin de représenter le résultat de cette requête, on a choisit une Map qui reflète la différence

parfois assez remarquable entre le nombre d'accidents qu'a pu connaître certains état par rapport à d'autres.



Analyse du résultat:

A partir du résultat qu'on a obtenu, on constate, bien que le nombre des accidents reste élevé dans chaque état pendant cette période d'à peu près trois ans, un état se distingue et compte plus de 663204 accidents. C'est donc l'état de Californie qui a connu le plus grand nombre d'accidents avec près de 22% du nombre total d'accidents qu'à connu le pays pendant ces trois ans.

Requête 2 :

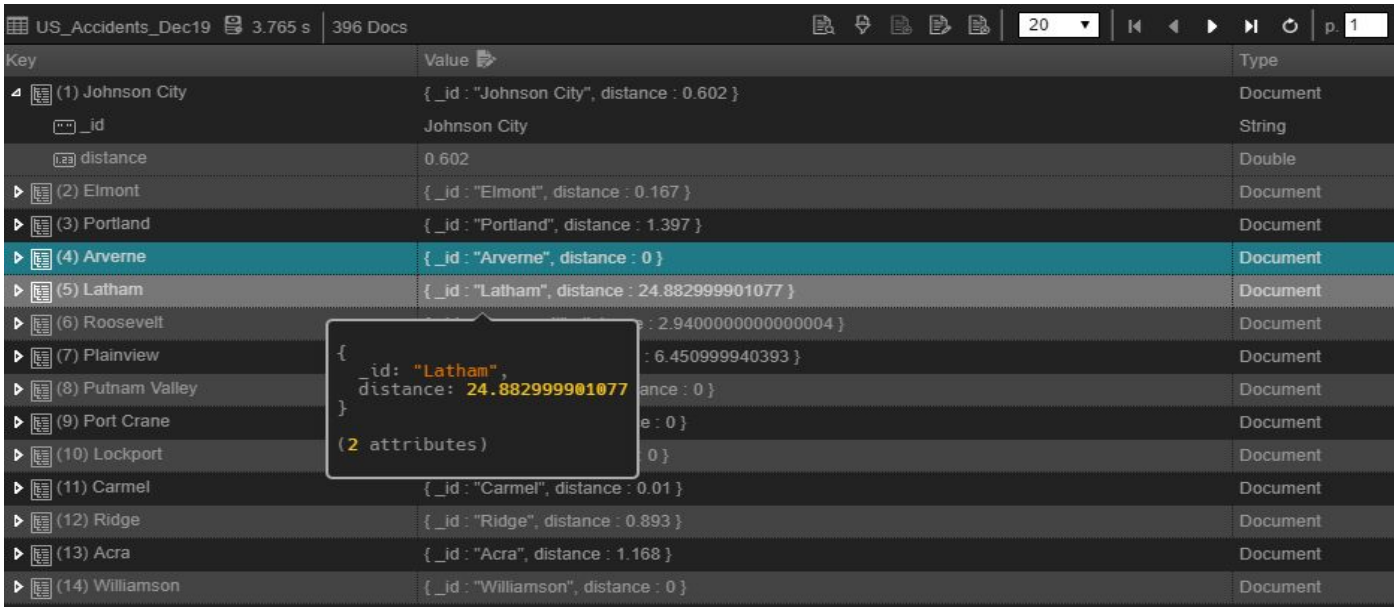
L'Objectif:

Recenser les différents accidents qui ont eu lieu à l'état de New York avec un vent très fort (>9 mph) allant du nord vers le sud, avec la somme des distances des accidents dans chacune de ses villes.

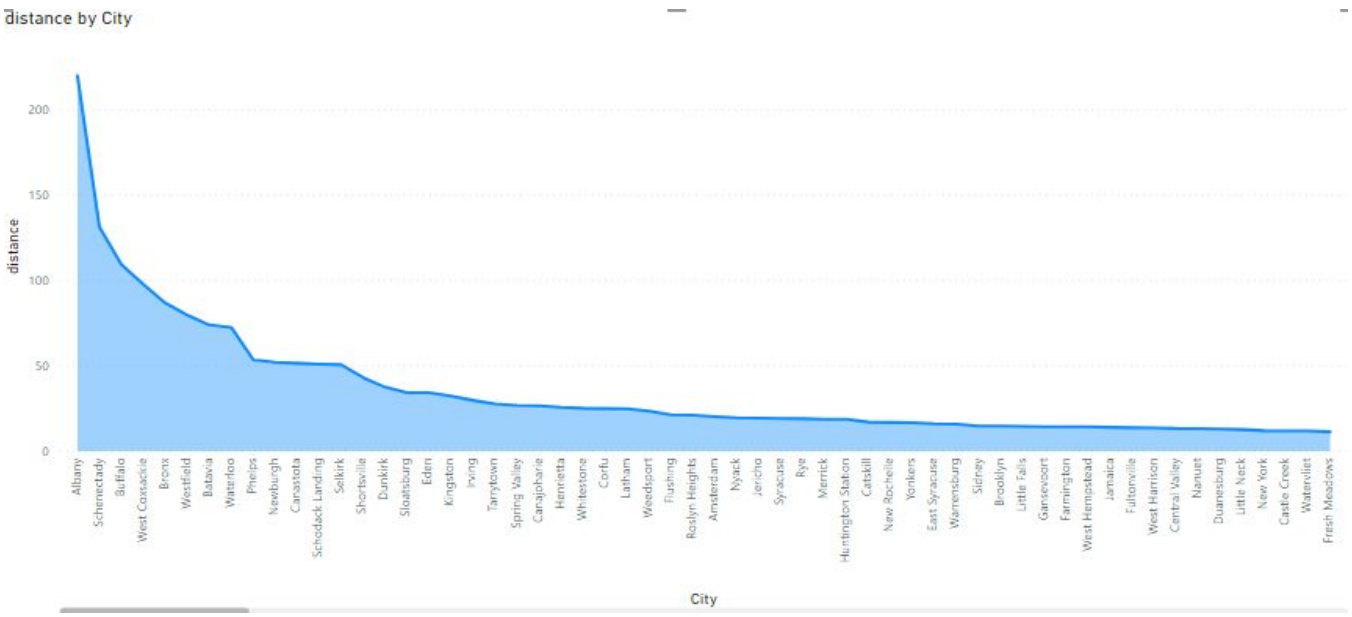
La Requête:

```
db.getCollection("US_Accidents_Dec19").aggregate([{$match : {$and: [
                                                    {"State": "NY"},
                                                    {"Wind_Speed(mph)":{"$gte": 9}},
                                                    {"Wind_Direction":"South"}
                                                    ]
                                                }
                                            },
                                            {$group: { _id: "$City", distance: {$sum: "$Distance(mi)"}
]);
```

Le Résultat:



La Représentation du résultat:



Analyse du résultat:

Dans la figure ci-dessus est représentée chaque ville de l'état de New York en fonction du total de la distance des accidents qui y ont eu lieu. La requête a permis dans la présentation d'observer uniquement les accidents où le vent est très fort (>9 mph) et allant du nord vers le sud de l'état afin d'analyser l'impact de telles circonstances sur la distance, qui est la longueur de l'étendue de la route touchée, des accidents.

Requête 3 :

L'Objectif:

Voir si un temps arbitrairement jugé difficile pour la conduite augmente les risques d'accident.

La Requête:

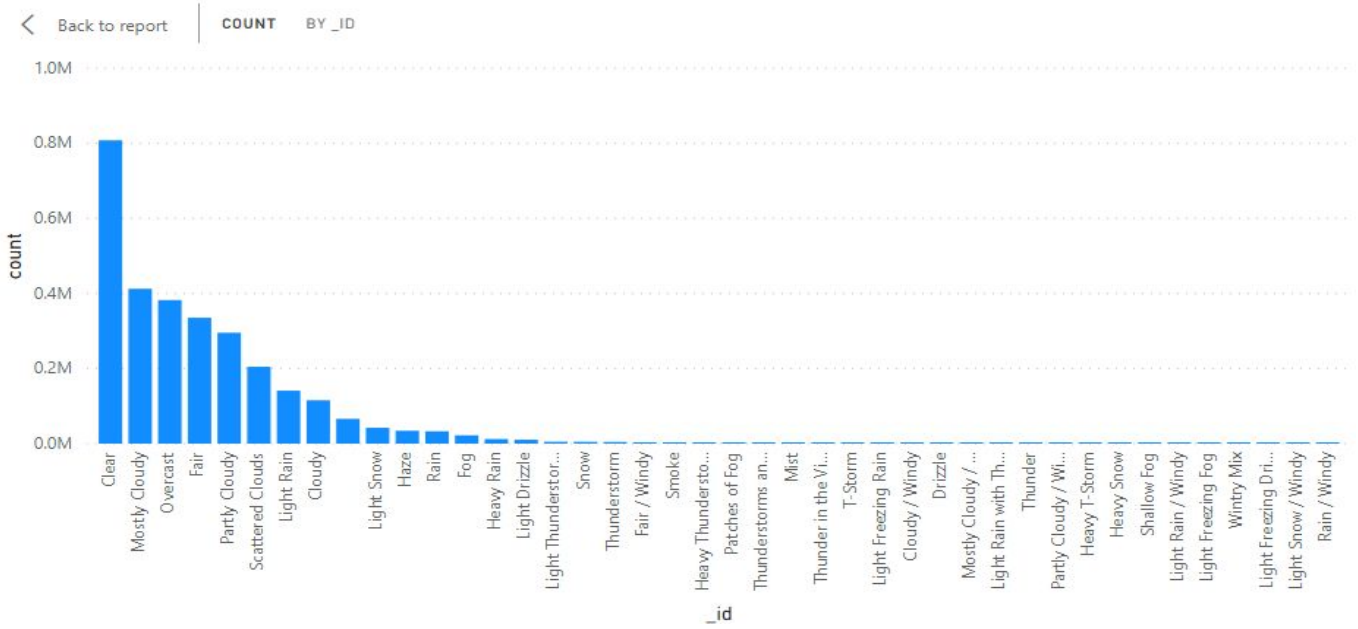
```
1
2  db.getCollection("US_Accidents_Dec19").aggregate([{$group: {_id: "$Weather_Condition", count: { $sum: 1}}},{ $sort:{'count':-1}}])
```

Le Résultat:

US_Accidents_Dec19 | 6.844 s | 121 Docs

Key	Value	Type
(1) Clear	{ _id : "Clear", count : 808171 }	Document
(2) Mostly Cloudy	count : 412528 }	Document
(3) Overcast	t : 382480 }	Document
(4) Fair	5289 }	Document
(5) Partly Cloudy	count : 295439 }	Document
(6) Scattered Clouds	ds", count : 204662 }	Document
(7) Light Rain	{ _id : "Light Rain", count : 141073 }	Document
(8) Cloudy	{ _id : "Cloudy", count : 115496 }	Document
(9)	{ _id : "", count : 65932 }	Document
(10) Light Snow	{ _id : "Light Snow", count : 42123 }	Document
(11) Haze	{ _id : "Haze", count : 34315 }	Document
(12) Rain	{ _id : "Rain", count : 32826 }	Document
(13) Fog	{ _id : "Fog", count : 22138 }	Document

La Représentation du résultat:



Analyse du résultat:

On remarque que, contrairement aux idées reçues, un temps clair ou simplement nuageux, qu'on pourrait juger propices à une conduite confortable, n'empêche pas de nombreux accidents d'arriver. Cette analyse semble correcte si et seulement si nous acceptons l'hypothèse que la distribution des météorologies est équiprobable. Autrement dit dans les faits cette hypothèse n'est pas vérifiée, il conviendrait de nuancer le propos en disant qu'il est aussi possible qu'au long de l'année il y a eu plus de jour ayant un temps clair et que par implication logique, la fréquence des accidents survenu un jour clair est plus importante.

Requête 4 :

L'Objectif:

Nombre d'accidents par Bins de valeurs sur différents champs : (Temperature(F), Wind_Chill(F), Humidity(%), Pressure(in), Visibility(mi), Wind_Direction, Wind_Speed(mph), Precipitation(in), Weather_Condition). Le résultat des requêtes permettra de montrer l'impact de ces facteurs sur les accidents

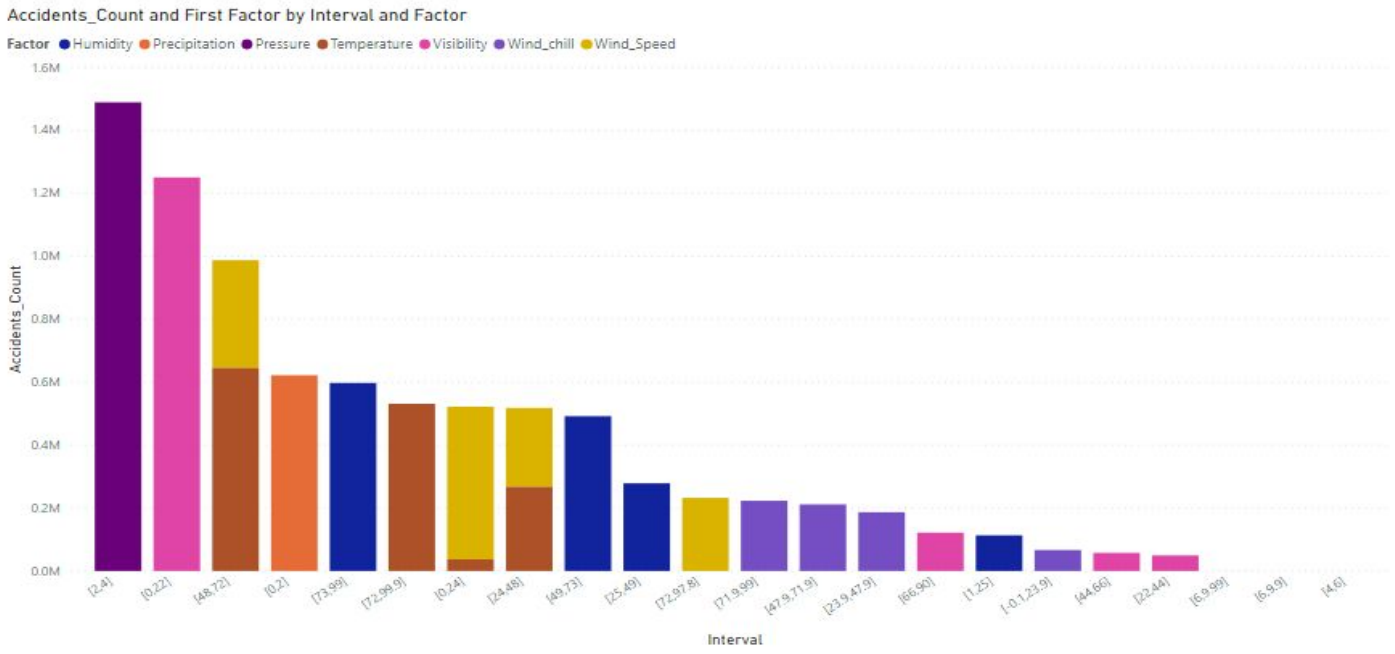
La Requête:

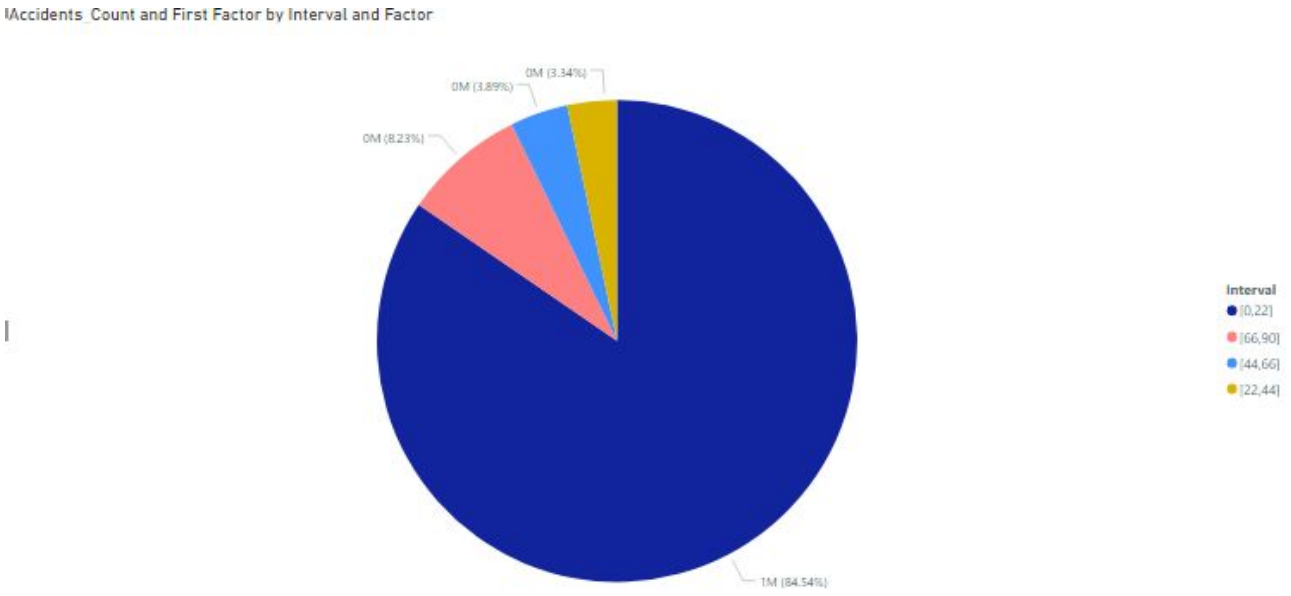
```
var resultat = "req4 = [";
var properties = [
    "Temperature(F)", "Wind_Chill(F)", "Humidity(%)", "Pressure(in)", "Visibility(mi)", "Wind_Speed(mph)", "Precipitation(in)"
];
for(var j = 0; j < properties.length; j++){
    var name = "$" + properties[j];
    resultat += "['" + properties[j] + "', ";
    var res = db.getCollection("US_Accidents_Dec19").aggregate({ $group : { _id: null, max: { $max : name }, min: { $min : name }}});
    var min = res["_batch"][0]["min"];
    var max = res["_batch"][0]["max"];
    resultat += min + ", " + max;
    var range = parseFloat(max) - parseFloat(min);
    var rangeInterval = Math.trunc(range/4);
    var reste = (range - 4 * rangeInterval).toFixed(2);
    var tab = [];
    var tab2 = [];
    resultat += ", [";
    for(var i = 0; i < 4; i++){
        if(i <= 2){
            tab.push([(parseFloat(min) + i * rangeInterval).toString(), (parseFloat(min) + (i+1) * rangeInterval).toString()]);
            resultat += "['" + (parseFloat(min) + i * rangeInterval).toString() + ", " + (parseFloat(min) + (i+1) * rangeInterval).toString() +
        }
        else{
            tab.push([(parseFloat(min) + i * rangeInterval).toString(), (parseFloat(min) + (i+1) * rangeInterval+ parseFloat(reste)).toString()]
            resultat += "['" + (parseFloat(min) + i * rangeInterval).toString() + ", " + (parseFloat(min) + (i+1) * rangeInterval+ parseFloat(re
        }
    }
    resultat += "], [";
    for(var i = 0; i < tab.length; i++){
        var z = {};
        z[properties[j]] = {$gte:tab[i][0], $lte:tab[i][1]};
        resultat += db.getCollection("US_Accidents_Dec19").find(z).count() + ((i !== tab.length - 1) ? ", " : "]");
    }
    var x = {};
    x[properties[j]] = 1.0;
    var y = {};
    y[properties[j]] = {$ne:null};
    resultat += "]" + ((j !== properties.length - 1) ? ", " : "]");
}
print(resultat);
```

Le Résultat:

```
Results :
req4 = [['Temperature', -0.0, 99.9, [['0, 24']['24, 48']['48, 72']['72, 99.9']], [36786, 268534, 646308, 531243]], ['Wind_Chill', -0.1, 99.0, [['-0.1, 23.9']['23.9, 47.9']['47.
```

La Représentation du résultat:





Analyse du résultat:

Pour la plupart des facteurs le nombre des accidents varient sur les quatres intervalles sauf la pression. La deuxième représentation ne prend en compte que la pression comme facteur et montre qu'une pression faible dans l'intervalle: [0, ((maxPressure-minPressure)/4)] compte le plus grand nombre d'accidents par rapport aux autres intervalles. On pourrait peut-être en conclure que la pression de l'air n'est pas un facteur important ou peut être l'est mais uniquement sous l'impact d'autres facteurs.

Requête 5 :

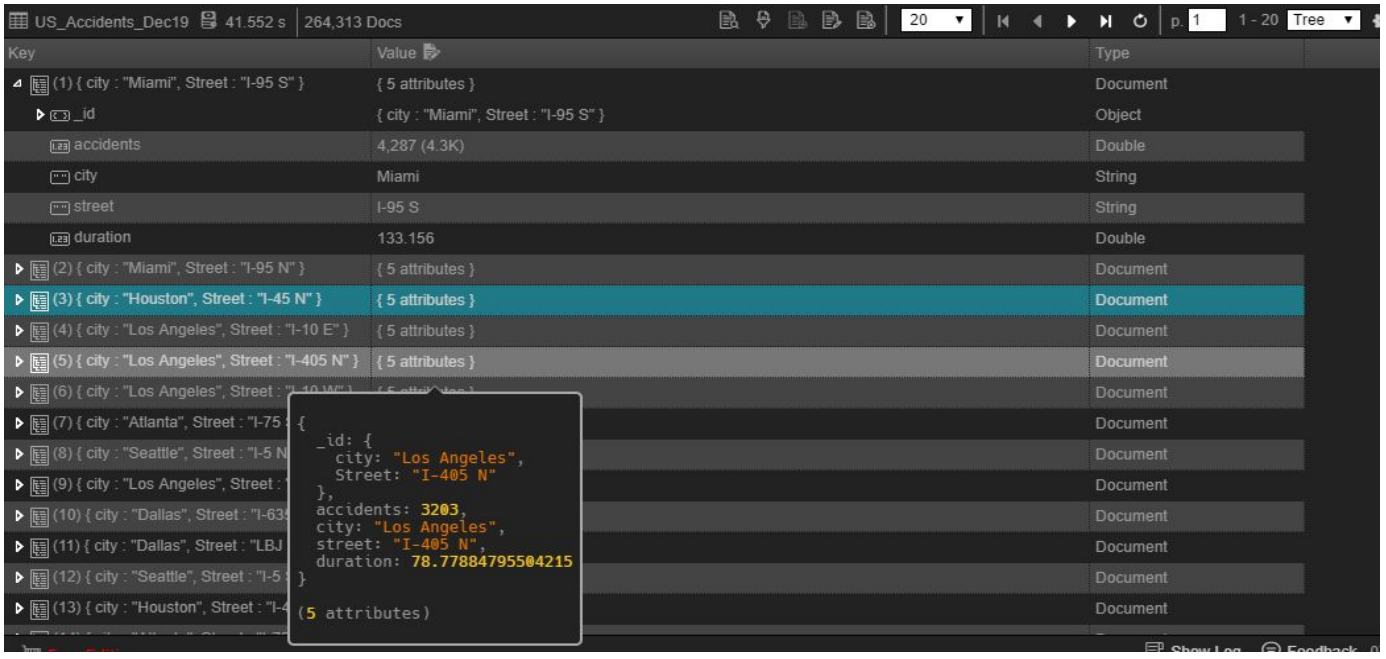
L'Objectif:

Le but est de déterminer le nombre d'accidents et la moyenne des durées de l'ensemble des accidents pour chaque ville.

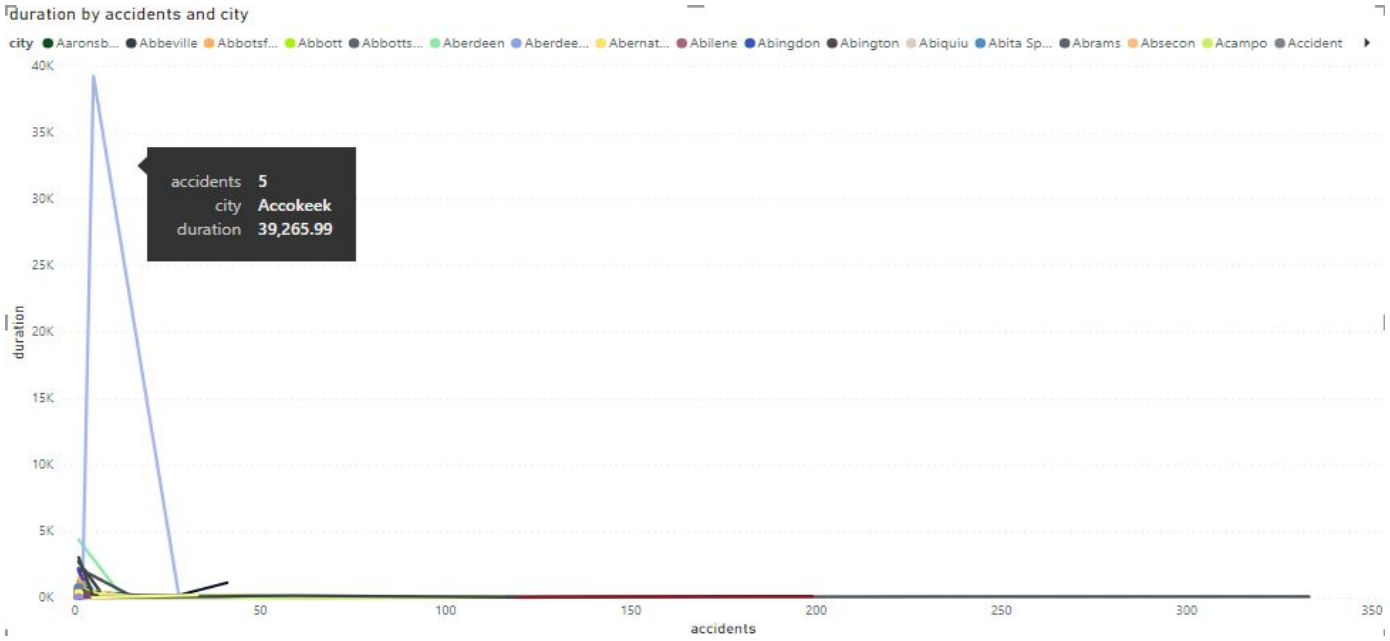
La Requête:

```
db.US_Accidents_Dec19.aggregate([
    {$match:{"City":{$ne:""}}},
    {$group: { _id: {"city":"$City","Street":"$Street"},
               accidents:{$sum: 1},
               city: {$first: "$City"},
               street: {$first: "$Street"},
               avgDurationInMilliSeconds: {$avg:{$subtract: [{$toDate: "$End_Time"},{$toDate:"$Start_Time"}]}}
    }},
    {$project: {duration: {$divide: ["$avgDurationInMilliSeconds",60000 ]}, "accidents": 1,"city": 1,"street": 1}},
    {$sort: {accidents:-1}},
    ], {"allowDiskUse": true});
```

Le Résultat:



La Représentation du résultat:



Analyse du résultat:

On constate que la ville de Accokeek se distingue par la plus longue durée d'accidents de 39265 minutes(équivalent à plus de 654 heures).Un résultat assez étonnant, vu que la ville qui se retrouve au deuxième rang est la ville de Aberdeen avec 4383 minutes. Une telle différence peut être liée à plusieurs facteurs uniques qui caractérisent la ville.

Requête 6 :

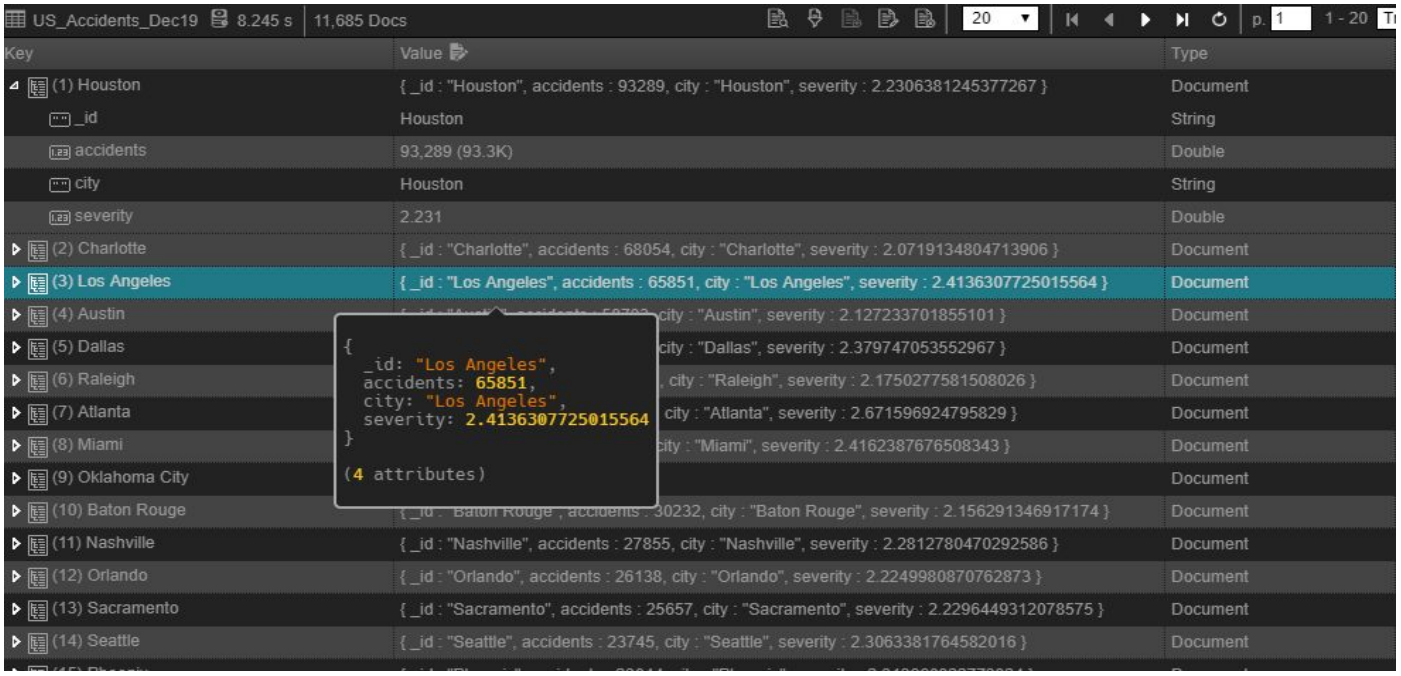
L'Objectif:

Après avoir comparé les villes en fonction de leur durée moyenne des accidents qui s'y sont produits, dans cette requête l'objectif est de les comparer en fonction de la moyenne de sévérité [la sévérité prend comme valeurs: 1, 2, 3 et 4] pour chaque ville.

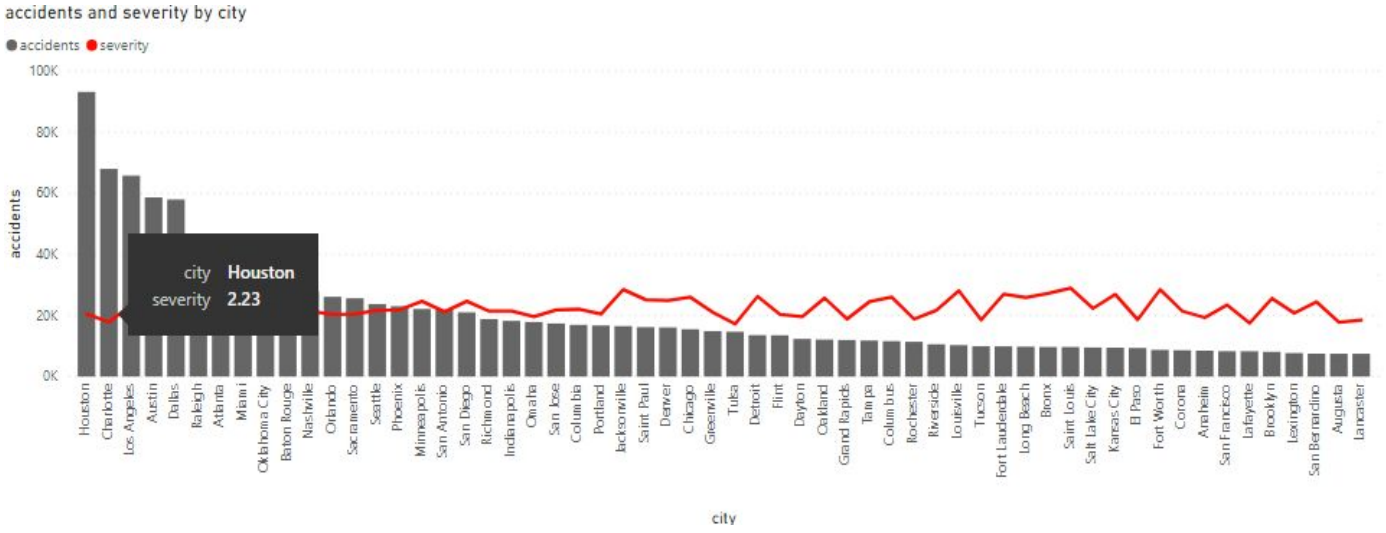
La Requête:

```
db.US_Accidents_Dec19.aggregate([
    {$match:{"City":{$ne:""}}},
    {$group: { _id: "$City",
               accidents:{$sum: 1},
               city: {$first: "$City"},
               severity: {$avg: "$Severity"}
    }},
    {$project: { "accidents": 1,"city": 1,"severity": 1}},
    {$sort: {accidents:-1}},
    ], {"allowDiskUse": true});
```

Le Résultat:



La Représentation du résultat:



Analyse du résultat:

La ville de Houston est la ville qui compte le plus grand nombre d'accidents, certes, mais garde une sévérité moyenne de 2.23/4. Le principe que l'on peut retirer est que le lien de la sévérité par rapport au nombre d'accidents par ville s'avère assez faible dû au cas traité.

Requête 7 :

L'Objectif:

Cette requête se base sur 3 paramètres principaux: les 6 états qui ont connu les plus grand nombre d'accidents, quatre plages horaires qu'on a prédéfini dans la requête[(minuit, avant 6h):Night, (6h, avant midi): Morning, (midi, avant 18h): AfterNoon, (18h, avant minuit): Evening]

ainsi que le nombre d'accidents enregistré par état.

La Requête:

```
db.US_Accidents_Dec19.aggregate([
      {$match: {"State": {$in: ["NC","CA","TX","FL","SC"]}}},
      {$addFields: { timeOfTheDay: {$cond: { if: { $and: [{$lt: [{$hour: {$toDate: "$Start_Time"}},12]}, {$gte:[{$hour: {$toDate: "$Start_Time"}},6]}]}, then: "Morning",
                                    else: { $cond: {if: { $and: [{$gte: [{$hour: {$toDate: "$Start_Time"}},12]}, {$lt:[{$hour: {$toDate: "$Start_Time"}},18]}]}, then: "Aftern
                                    else: { $cond: {if: { $and: [{$gte: [{$hour: {$toDate: "$Start_Time"}},18]}, {$lt:[{$hour: {$toDate: "$Start_Time"}},23]}]}, then: "Eveni
                                    else: "Night"}
                                    }}}}
      }},
      {$project: {"_id":1,"State":1,"timeOfTheDay":1}}
   ])
```

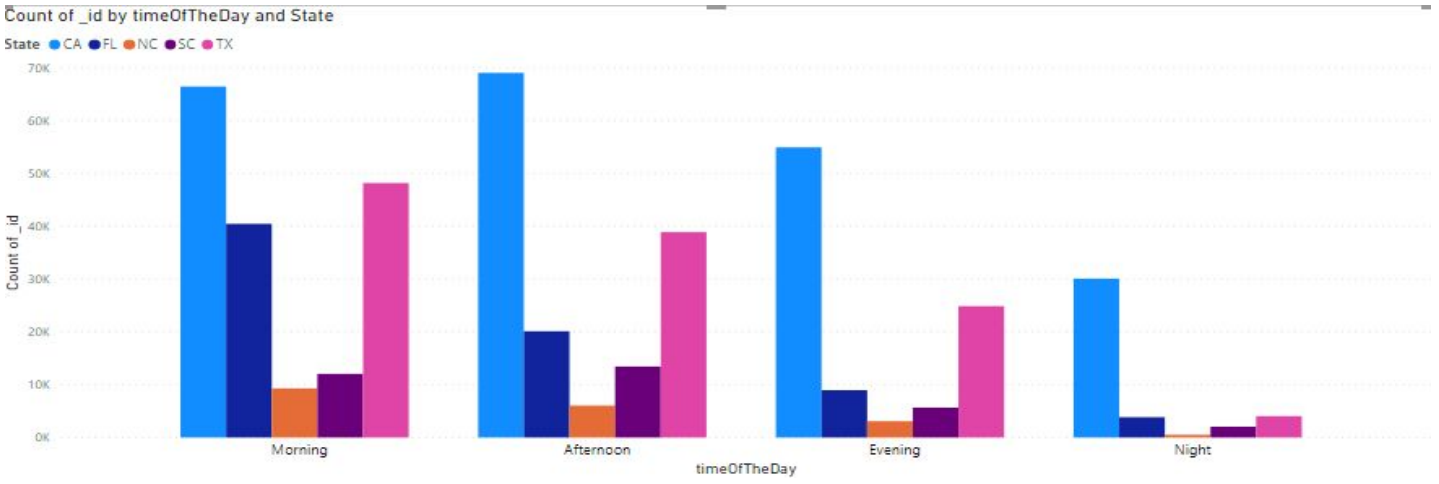
Le Résultat:

US_Accidents_Dec19 | 1.605 s | 1,474,161 Docs — 20 — p. 1

Key	Value	Type
(1) ObjectId("5e5d829e5c3e8257fa6a8596")	{ 3 attributes }	Document
_id	ObjectId("5e5d829e5c3e8257fa6a8596")	ObjectId
State	CA	String
timeOfTheDay	Morning	String
(2) ObjectId("5e5d829e5c3e8257fa6a8597")	{ 3 attributes }	Document
(3) ObjectId("5e5d829e5c3e8257fa6a8598")	{ 3 attributes }	Document
(4) ObjectId("5e5d829e5c3e8257fa6a8599")	{ 3 attributes }	Document
(5) ObjectId("5e5d829e5c3e8257fa6a859a")	{ 3 attributes }	Document
(6) ObjectId("5e5d829e5c3e8257fa6a859b")	{ 3 attributes }	Document
(7) ObjectId("5e5d829e5c3e8257fa6a859c")	{ 3 attributes }	Document
(8) ObjectId("5e5d829		Document
(9) ObjectId("5e5d829		Document
(10) ObjectId("5e5d82		Document
(11) ObjectId("5e5d82		Document
(12) ObjectId("5e5d82		Document
(13) ObjectId("5e5d829e5c3e8257fa6a85a2")	{ 3 attributes }	Document
(14) ObjectId("5e5d829e5c3e8257fa6a85a3")	{ 3 attributes }	Document
(15) ObjectId("5e5d829e5c3e8257fa6a85a4")	{ 3 attributes }	Document

```
{
  _id: ObjectId("5e5d829e5c3e8257fa6a859b"),
  State: "CA",
  timeOfTheDay: "Morning"
}
(3 attributes)
```

La Représentation du résultat:



Analyse du résultat:

L'évolution du nombre d'accidents est une fonction décroissante partant de la plage horaire 'Morning' [de 6h à 12h[vers 'Night' qui,elle, s'étend sur l'interval de temps de [de 00h à 6h[. Ceci est vrai pour 3 états, sauf pour les deux états: CA :"La Californie" et SC:"La Caroline du Sud" dont la

courbe monte vers la deuxième plage horaire ‘Afternoon’ [de 12h à 18h[pour redescendre par la suite. De plus, pour les 3 autres états, la première plage horaire connaît le plus grand nombre d’accidents dans la journée.

Requête 8 :

L'Objectif:

Les accidents de la route sont une source importante de décès, de blessures et de nombreux dommages matériels qui sont modélisés, dans notre dataset, par la sévérité de l'accident. Les accidents sont également une cause majeure d'embouteillages et de retards dans la circulation. La question qui se pose est à quelle point la sévérité de l'accident peut-elle affecter sa durée?

La Requête:

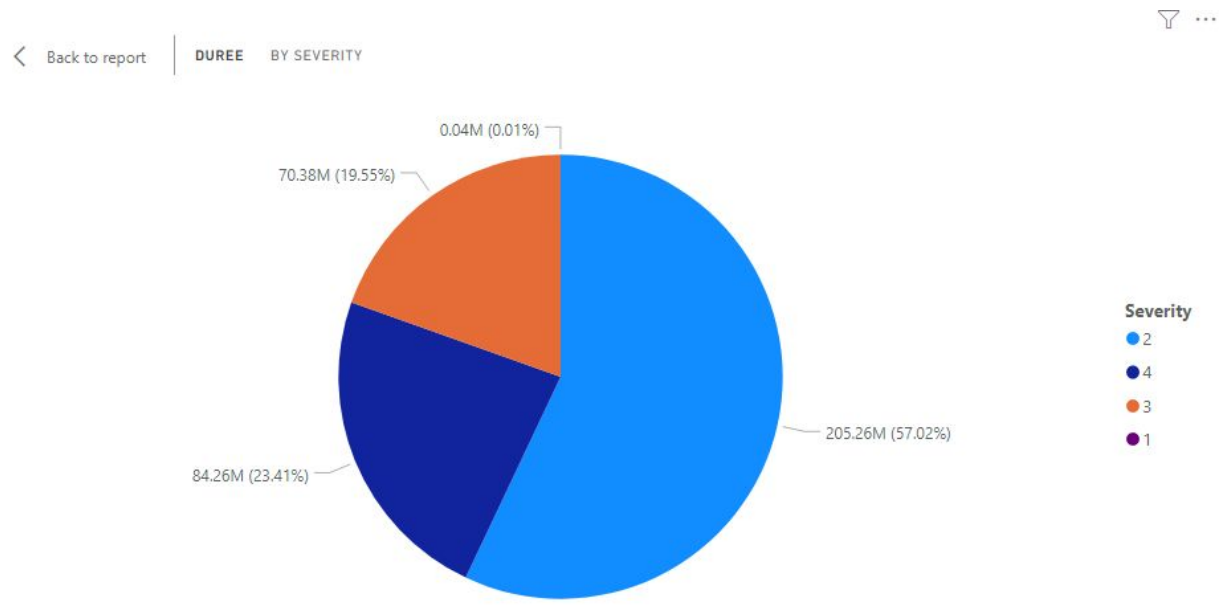
```
db.US_Accidents_Dec19.aggregate([
    {$match: {}},
    {$project: {"duree":{ $divide: [{$subtract: [{$toDate: "$End_Time"},{$toDate: "$Start_Time"}]},60000]},
                "_id":0,
                "Severity": 1,
                "Start_Time": 1,
                "End_Time": 1
    }},
    {$sort: {"Severity": 1}}
], {"allowDiskUse" : true})
```

Le Résultat:

US_Accidents_Dec19 31.339 s Fetch Count 20 p. 1

Key	Value	Type
(1)	{ 4 attributes }	Object
Severity	1	Int32
Start_Time	2016-02-15 17:22:10	String
End_Time	2016-02-15 18:07:10	String
duree	45	Int32
(2)	{ 4 attributes }	Object
(3)	{ 4 attributes }	Object
(4)	{ 4 attributes }	Object
(5)	{ 4 attributes }	Object
(6)	{ 4 attributes }	Object
(7)	{ 4 attributes }	Object
(8)	{ 4 attributes }	Object

La Représentation du résultat:



Analyse du résultat:

La durée totale des accidents de sévérité ‘1’ est négligeable par rapport à celle des autres sévérités, un constat qui est bien conforme à nos prédictions. Ce qui reste intéressant et contrairement à nos attentes, est le fait que les accidents de sévérité ‘2’, que l’on peut considérer comme étant une sévérité moyenne, et non ceux de sévérité ‘4’, sont eux qui ont occupé plus de 57% de la durée totale des accidents traités.

Requête 9 :

L'Objectif:

A partir de cette requête, on souhaite observer l'impact de la proximité à un point d'intérêt [les attributs: Amenity, Bump, No-Exit, Railway, Roundabout, station, stop, traffic_Calming, traffic signal, Crossing, Give_Way, Junction,.. au moins un de valeur 'true'] sur la sévérité de l'accident. Ce sont les accidents dues à l’affichage routier ou la configuration routière.

La Requête:

Pour cela, on regroupe tout accident ayant eu au moins un point d'intérêt à proximité ainsi que sa sévérité:

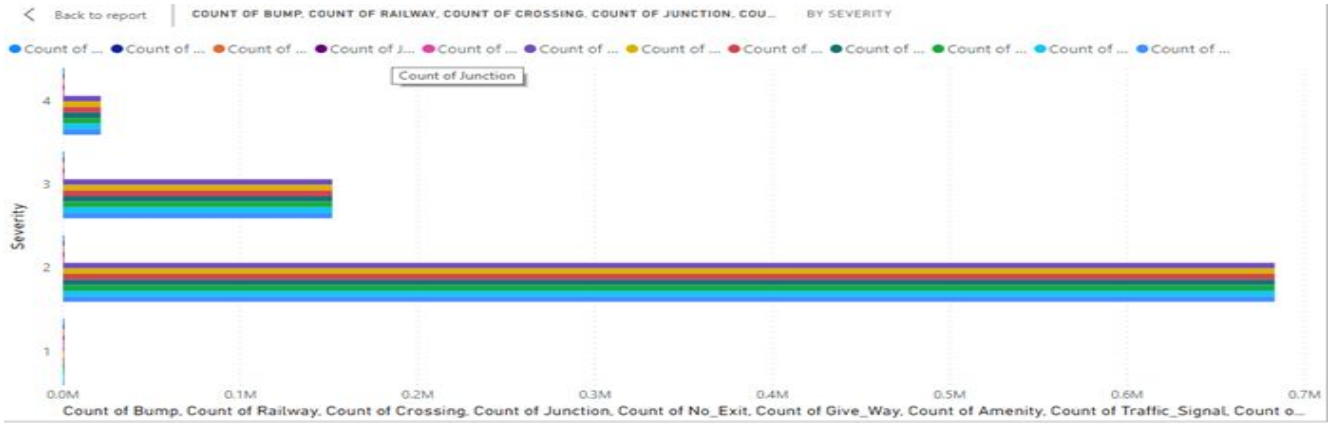
```
db.getCollection("US_Accidents_Dec19").aggregate([{$match: {$or: [{"Bump": "True"}, {"Amenity": "True"}, {"Junction": "True"}, {"No_Exit": "True"}, {"Traffic_Signal": "True"}, {
                                        {$project: {
                                            "_id":0,
                                            "State": 1,
                                            "Severity": 1,
                                            "Bump": 1,
                                            "Crossing":1,
                                            "Give_Way": 1,
                                            "Amenity": 1,
                                            "Junction": 1,
                                            "Railway": 1,
                                            "Roundabout": 1,
                                            "Stop": 1,
                                            "Station": 1,
                                            "Traffic_Calming": 1,
                                            "Traffic_Signal": 1,
                                            "No_Exit": 1
                                        }},
                                        {$sort: { "Severity":1}}
], { "allowDiskUse" : true });
```

Le Résultat:

US_Accidents_Dec19 | 35.991 s | 856,502 Docs

	Severity	State	Amenity	Bump	Crossing	Give_Way	Junction	No_Exit	Railway	Roundabout	Station	Stop	Traffic_Calming	Traffic_Signal
53	1	MD	False	False	False	False	False	False	False	False	False	False	False	True
54	1	MD	False	False	False	False	False	False	False	False	True	False	False	False
55	1	VA	False	False	True	False	False	False	False	False	False	False	False	True
56	1	TX	False	False	False	False	False	False	False	False	False	False	False	True
57	1	TX	False	False	False	False	False	False	False	False	False	False	False	True
58	1	TX	False	False	True	False	False	False	False	False	False	False	False	True
59	1	TX	False	False	False	False	False	False	False	False	False	False	False	True
60	1	WA	True	False	False	False	False	False	False	False	False	False	False	False
61	1	WA	False	False	True	False	False	False	False	False	False	False	False	True
62	1	WA	True	False	False	False	False	False	False	False	False	False	False	False
63	1	WA	False	False	False	False	False	False	False	False	False	False	False	True
64	1	CA	False	False	True	False	False	False	False	False	False	False	False	True
65	1	CA	False	False	False	False	True	False	False	False	False	False	False	False
66	1	IL	False	False	True	False	False	False	False	False	False	False	False	True
67	1	VA	False	False	False	False	False	True	False	False	False	False	False	False
68	1	TX	False	False	False	False	True	False	True	False	False	False	False	False
69	1	TX	False	False	False	False	False	False	False	False	False	True	False	False
70	1	TX	False	False	True	False	False	False	True	False	False	False	False	False
71	1	TX	False	False	False	False	False	False	False	False	True	False	False	False

La Représentation du résultat:



Analyse du résultat:

Dans la figure ci-dessus, on a réalisé la représentation de la distribution du nombre d'objets à proximité des accidents sur leur sévérité. Traffic_Calming, Traffic_Signal, Stop, Crossing, Bump, Junction, Give_Way, No_Exit, Railway, Roundabout, Station et Amenity sont les booléens qui présentent ces informations dans le Dataset. On conclut, alors, que le nombre d'accidents ayant eu un ou plusieurs points d'intérêt à proximité est de 856 502, ce qui représente près de 28% des

accidents étudiés et près de 700000 de ces accidents sont de sévérité 2.

Requête 10 :

L'Objectif:

L'objectif de la requête est de déterminer pour chaque état le nombre total d'accidents par chaque jour de semaine(une valeur qu'on va extraire de "start-time" qui représente la date du début de l'accident), du Dimanche comme jour 1 au Samedi, le jour 7.

La Requête:

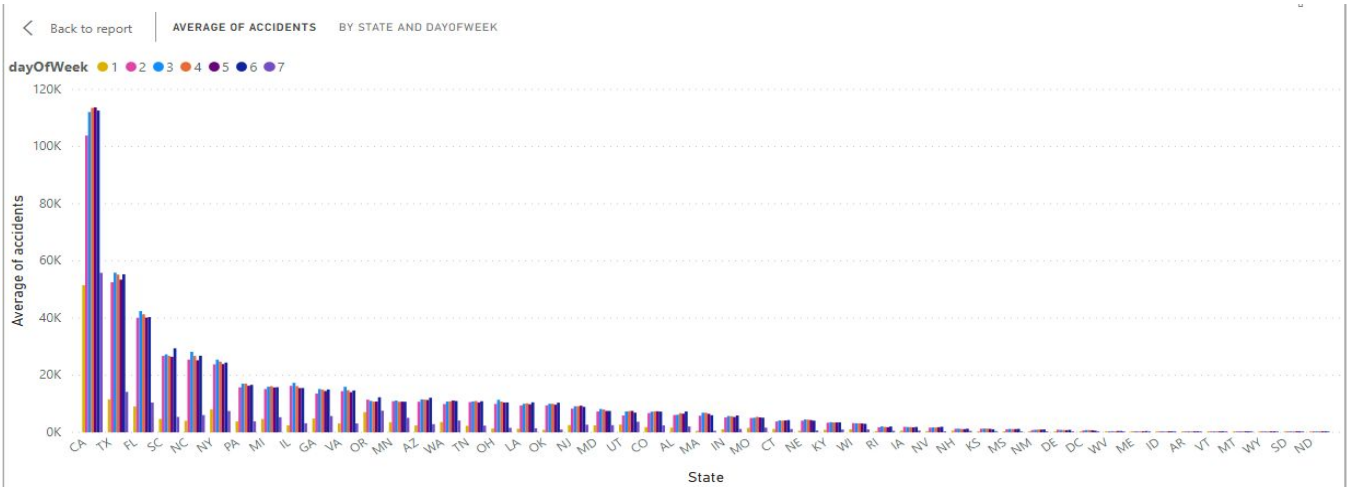
```
db.US_Accidents_Dec19.aggregate([
    {$match:{}},
    {$project:
        {
        day:  { $dayOfWeek: { $toDate: "$Start_Time" }}, "State": 1
        }
    },
    {$group: { _id: {"state": "$State", "day": "$day"}, count: {$sum: 1}}},
    {$sort: {"_id": 1}}

])
```

Le Résultat:

US_Accidents_Dec19 | 23.807 s | 343 Docs

Key	Value	Type
(1) { state : "AL", day : 1 }	{ _id : { state : "AL", day : 1 }, count : 1635 }	Document
_id	{ state : "AL", day : 1 }	Object
state	AL	String
day	1	Double
count	1,635 (1.6K)	Double
(2) { state : "AL", day : 2 }	{ _id : { state : "AL", day : 2 }, count : 6053 }	Document
(3) { state : "AL", day : 3 }	{ _id : { state : "AL", day : 3 }, count : 6132 }	Document
(4) { state : "AL", day : 4 }	{ _id : { state : "AL", day : 4 }, count : 6685 }	Document
(5) { state : "AL", day : 5 }	{ _id : { state : "AL", day : 5 }, count : 6522 }	Document
(6) { state : "AL", day : 6 }	{ _id : { state : "AL", day : 6 }, count : 7295 }	Document
(7) { state : "AL", day : 7 }	{ _id : { state : "AL", day : 7 }, count : 2047 }	Document
(8) { state : "AR", day : 1 }	{ _id : { state : "AR", day : 1 }, count : 113 }	Document
(9) { state : "AR", day : 2 }	{ _id : { state : "AR", day : 2 }, count : 201 }	Document
(10) { state : "AR", day : 3 }	{ _id : { state : "AR", day : 3 }, count : 293 }	Document
(11) { state : "AR", day : 4 }	{ _id : { state : "AR", day : 4 }, count : 354 }	Document
(12) { state : "AR", day : 5 }	{ _id : { state : "AR", day : 5 }, count : 345 }	Document
(13) { state : "AR", day : 6 }	{ _id : { state : "AR", day : 6 }, count : 300 }	Document
(14) { state : "AR", day : 7 }	{ _id : { state : "AR", day : 7 }, count : 143 }	Document
(15) { state : "AZ", day : 1 }	{ _id : { state : "AZ", day : 1 }, count : 2414 }	Document

La Représentation du résultat:



Analyse du résultat:

On constate que le jour 1 (le dimanche) suivi du jour 7 (le samedi)sont les jours ayant compté le moins d'accidents dans tous les états.Contrairement aux jours 3 (Mardi), 4 (Mercredi) et 6 (Vendredi) qui enregistre des nombres d'accidents largement plus élevé que ceux pendant les Week-ends.On conclut alors, selon le graphique, que le nombre des accidents varie légèrement entre les jours de la semaine. Toutefois, la distribution de ce nombre pendant le week-end est largement différente de celle des jours de la semaine.

4. Note sur l'organisation et le déroulement du projet

Afin d'organiser notre projet et de permettre un travail efficace en équipe, nous avons décidé de suivre les principes de la méthode AGILE (hiérarchie horizontale et fonctionnement par Sprint) grâce notamment à l'outil Trello [7]. Nous avons pu organiser nos différentes tâches par importance et de manière atomique afin de produire des versions augmentant la qualité et la quantité du projet.

De plus, nous avons choisi Github plutôt que GitLab avec lequel nous sommes plus familier.

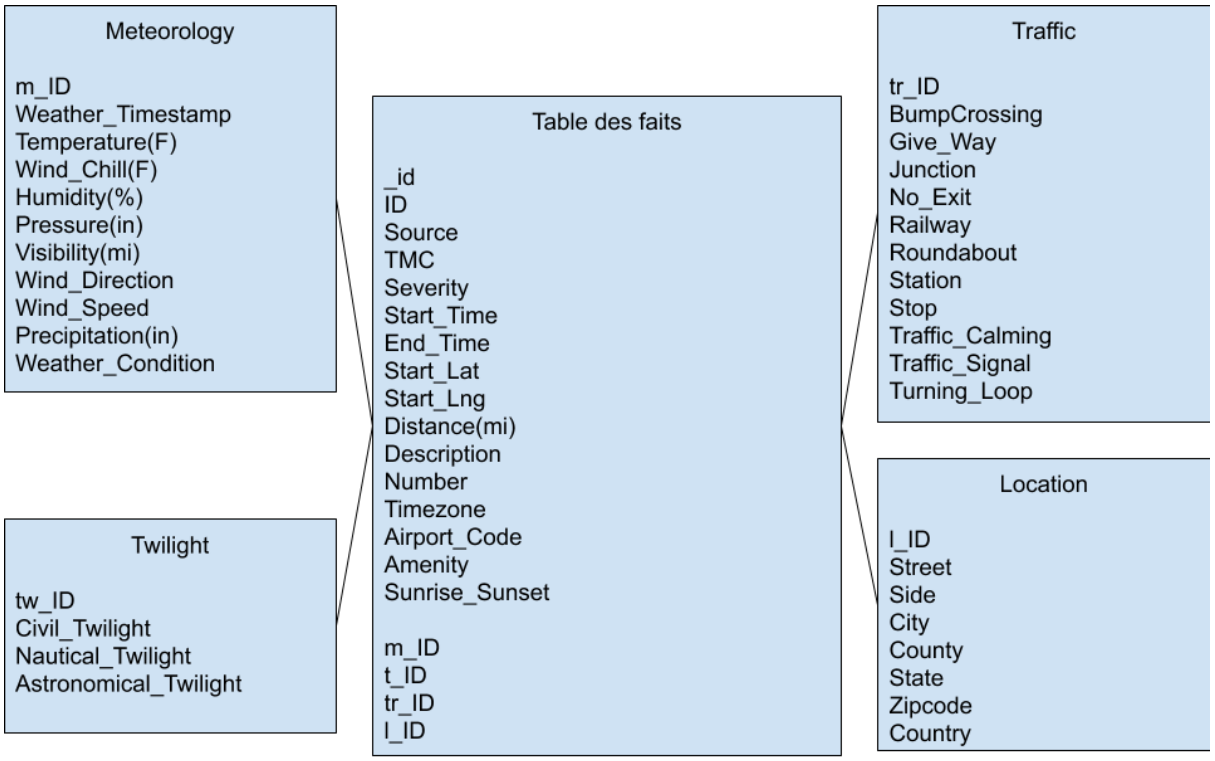
Pour ce qui est de la participation sur les tâches, afin de nous répartir équitablement le travail et de pouvoir tous travailler sur les aspects importants du projet, nous avons tous réalisé des requêtes en MongoDB (Javascript). En revanche, Aniss et Aya se sont occupés plus particulièrement de la visualisation des données et l'interprétation des requêtes. Clément a réalisé une partie de la rédaction du rapport et surtout les slides de présentation ainsi que la gestion du Trello. Quant à Camille, il a effectué la majeure partie de la rédaction du rapport et l'organisation/rédaction du github et la première étape de rédaction du modèle globale des requêtes en javascript.

Concernant le projet en lui-même, nous avons rencontré quelques erreurs notamment lors de l'exécution de certaines requêtes. Une des erreurs est le dépassement de la limite de mémoire avec le code d'erreur 16819 lorsque le tri par agrégation est utilisé. Par défaut, l'agrégation dans MongoDB se produit en mémoire et les étapes du pipeline ont une limite de 100 Mb de RAM. Ayant dépassé ce seuil, pour traiter un grand ensemble de données, il faut activer les étapes du pipeline d'agrégation pour écrire des données dans des fichiers temporaires. En ajoutant la commande suivante { "allowDiskUse" : true }, le problème fut résolu.

5. Conclusion

Ce projet aura été pour nous l'occasion de manipuler un nouvel environnement de gestion de base de données. Se confronter à de nouvelles technologies est toujours un challenge intéressant et enrichissant pour renforcer et élargir nos compétences.

En terme d'amélioration, structurer nos tables en étoile pourrait permettre de gagner de la place. Cependant, au vue des évènements enregistrés et historisés, il n'était pas utile selon nous d'adapter la structure ainsi car le gain aurait été beaucoup plus limité que sur des données qui se manifestent plus régulièrement. Le modèle est décrit par les tables suivantes :



6. Sources

[1] https://www.kaggle.com/sobhanmoosavi/us-accidents, visité le 17/02/2020. Se réfère aux travaux :

- Moosavi, Sobhan, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, and Rajiv Ramnath. "A Countrywide Traffic Accident Dataset.", 2019.
- Moosavi, Sobhan, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, Radu

Teodorescu, and Rajiv Ramnath. "Accident Risk Prediction based on Heterogeneous Sparse Data: New Dataset and Insights." In proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2019.

[2] https://www.mongodb.com/fr, visité le 19/02/2020. Site portail de MongoDB.

[3] https://studio3t.com/, visité le 19/02/2020. Site portail de Studio 3T.

[4] https://trello.com/, visité le 19/02/2020. Site portail de Trello.

[5] https://nosqlbooster.com/, visité le 20/02/2020. Site portail de NoSQLBooster pour MongoDB

[6] https://www.microsoft.com/fr-fr/download/details.aspx?id=58494, le lien de téléchargement de l'outil de Microsoft: Power BI Platform.