

Aggregation Pipelines

Les opérations d'agrégation aide à regrouper, trier, faire des calculs avancés, analyser les données.

Syntaxe

```
Db.NomCollection.aggregate([
    { $match : { Conditions de filtre } },
    { $project : { champs à afficher } },
    { $sort : { champs de tri } },
    { $limit : 3 }
])
```

Créer la base de données entreprise

Créer la collection emp qui va contenir la liste suivante

Nom	prenom	Fonction	age	dept	salaire
aajli	abdellatif	developpeur	"25"	NTIC	5000
ait ahmed	wafa	directeur	"26"	COMMERCIAL	6000
ait elkadi	halima	graphiste	"24"	NTIC	7200
alakma	soufiane	developpeur	"20"	NTIC	5200
baladda	hassan	developpeur	"28"	COMMERCIAL	3800
BARODI	BOUBAKER	developpeur	"30"	COMMERCIAL	7200
bazaz	yassine	graphiste	"28"	NTIC	4800
BOULAHKA	MOHAMED	graphiste	"32"	NTIC	6200
bouthaim	afafe	developpeur	"20"	NTIC	7000
dakouk	ikram	directeur	"22"	TECHNIQUE	2500

EL GANDI	TAHA	directeur	"21"	NTIC	3500
El hayani Maarouf	Imad	vendeur	"20"	NTIC	3500
el khald	abdellah	developpeur	"23"	TECHNIQUE	6200
el khattaby	lahoucine	vendeur	"24"	TECHNIQUE	3500
el mouaouine	abdelilah	directeur	"25"	NTIC	4500
EL-KABAIR	HASSAN	developpeur	"26"	NTIC	2800
ESSEQALLY	Meriem	vendeur	"20"	NTIC	3200
lagrini	zakaria	developpeur	"28"	TECHNIQUE	6500
lahfaouti	ilyas	graphiste	"20"	COMMERCIAL	2800
MERKHT	ACHRAF	vendeur	"28"	COMMERCIAL	6000
mofdi	maryam	developpeur	"20"	COMMERCIAL	9000
namat	abderhmane	developpeur	"28"	NTIC	6500
oubakasse	eh hassan	vendeur	"20"	NTIC	2700
shabi	zakariae	directeur	"28"	TECHNIQUE	6500
Tamraoui	Abdelhak	developpeur	"29"	NTIC	4800

Cr��er une bd bd203	<code>use bd203</code>
Cr��er une collection emp	<code>db.createCollection("emp")</code>
Afficher les collections	<code>show collections</code>
Supprimer la bd	<code>db.dropDatabase()</code>
Supprime la collection	<code>db.coll201.drop()</code>

Lister les employés	<pre>db.emp.aggregate([])</pre>	<pre>db.emp.find()</pre>
Lister les nom ,prenom ,salaire	<pre>db.emp.aggregate([{ \$project : { "_id":0 , "Nom":1 , "prenom":1 , "salaire":1 } }])</pre>	<pre>db.emp.find({}, { "_id":0 , "Nom":1 , "prenom" : 1 , "salaire":1})</pre>
Lister les nom ,prenom ,salaire trié par salaire croissant	<pre>db.emp.aggregate([{ \$project : { "_id":0 , "Nom":1 , "prenom" : 1 , "salaire" : 1} }, { \$sort : { "salaire" : -1 } }])</pre>	<pre>db.emp.find({}, { "_id":0 , "Nom":1 , "prenom" : 1 , "salaire":1}},).sort({ "salaire" : -1})</pre>
Afficher l'employé le plus payé	<pre>db.emp.aggregate([{ \$project : { "_id":0 , "Nom":1 , "prenom" : 1 , "salaire" : 1} }, { \$sort : { "salaire" : -1 } }, { \$limit : 1 }])</pre>	<pre>db.emp.find({}, { "_id":0 , "Nom":1 , "prenom" : 1 , "salaire":1}},).sort({ "salaire" : -1}).limit(1)</pre>
Afficher l'employé le moins payé	<pre>db.emp.aggregate([{ \$project : { "_id":0 , "Nom":1 , "prenom" : 1 , "salaire" : 1} },</pre>	

	<pre>{ \$sort : { "salaire" : 1 } }, { \$limit : 1 }])</pre>	
Lister les nom ,prenom ,salaire des employés , trié par âge croissant	<pre>db.emp.aggregate([{ \$project : { "_id" : 0 , "Nom" : 1 ,"prenom" : 1 , "salaire" : 1, "age":1} }, { \$sort : { "age" : 1} }] , {collation : {"locale":"fr" , numericOrdering : true } })</pre>	<pre>db.emp.find({}, { "_id":0 , "Nom":1 , "prenom" : 1 , "age":1 , "salaire":1}) . sort({"age" : -1}) .collation({"locale" : "fr" , numericOrdering : true })</pre>
Lister les nom ,prenom ,salaire Des developpeurs, triés par salaire décroissant	<pre>db.emp.aggregate([{ \$match : {"Fonction" : "developpeur"}}, { \$project : {"_id" : 0 , "Nom":1 ,"prenom":1 , "Fonction":1 , "salaire" : 1 } }, { \$sort : { "salaire" : -1} }])</pre>	
Lister les nom ,prenom ,salaire des developpeur et ayant salaire >5000	<pre>db.emp.aggregate([{ \$match :{ \$and : [{"Fonction" : "developpeur"} , {"salaire" : {\$gt :5000}}] } }, { \$project : {"_id" : 0 , "Nom":1 ,"prenom":1 , "Fonction":1 , "salaire" : 1 } }])</pre>	<pre>db.emp.find({ \$and : [{"Fonction" : "developpeur"} , {"salaire" : {\$gt :5000}}] }, { _id : 0 , "Nom":1 ,"prenom":1 , "Fonction":1 , "salaire" : 1 }])</pre>
Lister les nom ,prenom ,salaire des developpeur ou ayant salaire >6000	<pre>db.emp.aggregate([{ \$match : { \$or:[{"Fonction": "developpeur"}, {"salaire" : {\$gt :6000} }] } }, { \$project : { "_id":0 , "Nom" : 1 , "prenom" : 1, "Fonction" : 1 , "salaire" :1} }])</pre>	<pre>Filtre={ \$or:[{"Fonction": "developpeur"}, {"salaire" : {\$gt :6000} }] } project = { "_id":0 , "Nom" : 1 , "prenom" : 1, "Fonction" : 1 , "salaire" :1} db.emp.find(filtre , project)</pre>

Fonctions et regroupements

Calculer le nombre des employés par ...	<pre>db.emp.aggregate([{ \$group : { "_id" : 1 , 'Nbr emp' : { \$count : { } } } }])</pre>	Db.emp.find().count()
Calculer le nombre des employés par département	<pre>db.emp.aggregate([{ \$group : { "_id" : "\$dept" , 'Nbr emp par dept': { \$count : { } } } }])</pre>	On ne peut pas le faire par find
Calculer le nombre des employés par fonction	<pre>db.emp.aggregate([{ \$group : { "_id" : "\$Fonction" , 'Nbr emp par dept': { \$count : { } } } }])</pre>	
Calculer la somme des salaires des employés par "_id":1	<pre>db.emp.aggregate([{ \$group : { "_id":1 , 'somme des salaires' : { \$sum : "\$salaire" } } }])</pre>	On ne peut pas le faire par find
Calculer la somme des salaires des employés par fonction	<pre>db.emp.aggregate([{ \$group : { "_id":"\$Fonction" , 'somme des salaires par fonction' : { \$sum : "\$salaire" } } }])</pre>	On ne peut pas le faire par find
La moyenne des salaires par département	<pre>db.emp.aggregate([{ \$group : { "_id" : "\$dept" , "Moyenne des salaires par dept" : { \$avg : "\$salaire" } } }])</pre>	On ne peut pas le faire par find
Afficher le salaire max par	<pre>db.emp.aggregate([</pre>	

	<pre>{ \$group : { "_id" : 1 , "salaire maximal" : { \$max : "\$salaire" } } } }]</pre>	
Afficher le salaire max par fonction	<pre>db.emp.aggregate([{ \$group : { "_id" : "\$Fonction" , "salaire maximal" : { \$max : "\$salaire" } } }])</pre>	On ne peut pas le faire par find
Afficher le salaire max et min	<pre>db.emp.aggregate([{ \$group : { "_id" :1 , "max des salaires" : { \$max : "\$salaire"} , "min des salaire" : { \$min : "\$salaire" } } }])</pre>	
Afficher le salaire max et min par fonction	<pre>db.emp.aggregate([{ \$group : { "_id" :"\$Fonction" , "max des salaires" : { \$max : "\$salaire"} , "min des salaire" : { \$min : "\$salaire" } } }])</pre>	On ne peut pas le faire par find
Lister les départements sans répétition	<pre>db.emp.distinct("dept")</pre>	
Lister les fonctions sans répétition	<pre>db.emp.distinct("Fonction")</pre>	