Aggregation Pipelines

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

Syntaxe

Db.NomCollection.aggreagate([  
 { $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 |

|  |  |
| --- | --- |
| Créer une bd bd203 | use bd203 |
| Créer une collection emp | db.createCollection("emp") |
| Afficher les collections | show collections |
| Supprimer la bd | db.dropDatabase() |
| Supprime la collection | db.coll201.drop() |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| Lister les employés | db.emp.aggregate([  ]) | db.emp.find( ) |
| Lister les nom ,prenom ,salaire | db.emp.aggregate([  { $project : {"\_id":0 , "Nom":1 , "prenom":1 , "salaire":1 } }  ]) | db.emp.find({},  {"\_id":0 , "Nom":1 , "prenom" : 1 , "salaire":1}  ) |
| Lister les nom ,prenom ,salaire trié par salaire croissant | db.emp.aggregate([  { $project : { "\_id":0 , "Nom":1 , "prenom" : 1 , "salaire" : 1} },  { $sort : { "salaire" : -1 } }  ]) | db.emp.find({} ,  {"\_id":0 , "Nom":1 , "prenom" : 1 , "salaire":1}},  ).sort({{"salaire" : -1}) |
| Afficher l’employé le plus payé | db.emp.aggregate([  { $project : { "\_id":0 , "Nom":1 , "prenom" : 1 , "salaire" : 1} },  { $sort : { "salaire" : -1 } },  { $limit : 1 }  ]) | db.emp.find({} ,  {"\_id":0 , "Nom":1 , "prenom" : 1 , "salaire":1}},  ).sort({{"salaire" : -1}).limit(1) |
| Afficher l’employé le moins payé | db.emp.aggregate([  { $project : { "\_id":0 , "Nom":1 , "prenom" : 1 , "salaire" : 1} },  { $sort : { "salaire" : 1 } },  { $limit : 1 }  ]) |  |
| Lister les nom ,prenom ,salaire des employés , trié par âge croissant | db.emp.aggregate([  { $project : { "\_id" : 0 , "Nom" : 1 ,"prenom" : 1 , "salaire" : 1, "age":1} },  { $sort : { "age" : 1} }  ] , {collation : {"locale":"fr" , numericOrdering : true } } ) | db.emp.find({},  { "\_id":0 , "Nom":1 , "prenom" : 1 , "age":1 , "salaire":1} )  . sort( {"age" : -1})  .collation({"locale" : "fr" , numericOrdering : true })  ] |
| Lister les nom ,prenom ,salaire  Des developpeurs, triés par salaire décroissant | db.emp.aggregate([  { $match : {"Fonction" : "developpeur"}},  { $project : {"\_id" : 0 , "Nom":1 ,"prenom":1 , "Fonction":1 , "salaire" : 1 } },  { $sort : { "salaire" : -1} }  ]) |  |
| Lister les nom ,prenom ,salaire des developpeur et ayant salaire >5000 | db.emp.aggregate([  { $match :{ $and : [{"Fonction" : "developpeur"} , {"salaire" : {$gt :5000}} ] } },  { $project : {"\_id" : 0 , "Nom":1 ,"prenom":1 , "Fonction":1 , "salaire" : 1 } }  ]) | db.emp.find(  { $and : [{"Fonction" : "developpeur"} , {"salaire" : {$gt :5000}} ] },  { id" : 0 , "Nom":1 ,"prenom":1 , "Fonction":1 , "salaire" : 1 }  ]) |
| Lister les nom ,prenom ,salaire des developpeur ou ayant salaire >6000 | db.emp.aggregate([  { $match : { $or:[ {"Fonction": "developpeur"},{"salaire" : {$gt :6000} } ] } },  {$project : { "\_id":0 , "Nom" : 1 , "prenom" : 1, "Fonction" : 1 , "salaire" :1} }  ]) | Filtre={ $or:[ {"Fonction": "developpeur"},{"salaire" : {$gt :6000} } ] } project = { "\_id":0 , "Nom" : 1 , "prenom" : 1, "Fonction" : 1 , "salaire" :1} db.emp.find( filtre , project) |
| Fonctions et regroupements | | |
| Calculer le nombre des employés ~~par …~~ | db.emp.aggregate([  { $group : {"\_id" : 1 , 'Nbr emp' : { $count :{ } } }}  ]) | Db.emp.find().count() |
| Calculer le nombre des employés par département | db.emp.aggregate([  { $group : { "\_id" : "$dept" , 'Nbr emp par dept': { $count :{} } } }  ]) | On ne peut pas le faire par find |
| Calculer le nombre des employés par fonction | db.emp.aggregate([  { $group : { "\_id" : "$Fonction" , 'Nbr emp par dept': { $count :{} } } }  ]) |  |
| Calculer la somme des salaires des employés par…. "\_id" :1 | db.emp.aggregate([  { $group : { "\_id":1 , 'somme des salaires' : { $sum : "$salaire"} } }  ]) | On ne peut pas le faire par find |
| Calculer la somme des salaires des employés par fonction | db.emp.aggregate([  { $group : { "\_id":"$Fonction" , 'somme des salaires par fonction' : { $sum : "$salaire"} } }  ]) | On ne peut pas le faire par find |
| La moyenne des salaires par département | db.emp.aggregate([  { $group : {"\_id" : "$dept" , "Moyenne des salaires par dept" : {$avg :"$salaire" } } }  ]) | On ne peut pas le faire par find |
| Afficher le salaire max ~~par~~ | db.emp.aggregate([  { $group : { "\_id" : 1 , "salaire maximal" : { $max : "$salaire" } } }  ]) |  |
| Afficher le salaire max par fonction | db.emp.aggregate([  { $group : { "\_id" : "$Fonction" , "salaire maximal" : { $max : "$salaire" } } }  ]) | On ne peut pas le faire par find |
| Afficher le salaire max et min | db.emp.aggregate([  { $group : { "\_id" :1 , "max des salaires" : { $max : "$salaire"} , "min des salaire" : { $min : "$salaire" } } }  ]) |  |
| Afficher le salaire max et min par fonction | db.emp.aggregate([  { $group : { "\_id" :"$Fonction" , "max des salaires" : { $max : "$salaire"} , "min des salaire" : { $min : "$salaire" } } }  ]) | On ne peut pas le faire par find |
| Lister les départements sans répétition | db.emp.distinct("dept") |  |
| Lister les fonctions sans répétition | db.emp.distinct("Fonction") |  |
|  |  |  |