

# Mongo DB

## 1- Encuentra todos los Pokémon de tipo “Electric”.

```
> db.Pokemones.aggregate([
  { $match: { Type1: "Electric" } }]);
< {
  _id: ObjectId('67edde1135bb6d6f6ebed73a'),
  Num: 25,
  Name: 'Pikachu',
  Type1: 'Electric',
  HP: 35,
  Attack: 55,
  Defense: 40,
  SpAtk: 50,
  SpDef: 50,
  Speed: 90,
  Generation: 1,
  Legendary: false
}
{
  _id: ObjectId('67edde1135bb6d6f6ebed73b'),
  Num: 26,
  Name: 'Raichu'.
```

```
{
  _id: ObjectId('67edde1135bb6d6f6ebed73b'),
  Num: 26,
  Name: 'Raichu',
  Type1: 'Electric',
  HP: 60,
  Attack: 90,
  Defense: 55,
  SpAtk: 90,
  SpDef: 80,
  Speed: 110,
  Generation: 1,
  Legendary: false
}
{
  _id: ObjectId('67edde1135bb6d6f6ebed774'),
  Num: 81,
  Name: 'Magnetite',
  Type1: 'Electric',
  Type2: 'Steel',
  HP: 25,
  Attack: 35,
  Defense: 70,
```

**2-Muestra solo los nombres y el ataque de los Pokémon con más de 100 de ataque.**

```
> db.Pokemones.aggregate([
  { $match: { Attack: { $gt: 100 } } }
]);
< {
  _id: ObjectId('67edde1135bb6d6f6ebed723'),
  Num: 6,
  Name: 'CharizardMega Charizard X',
  Type1: 'Fire',
  Type2: 'Dragon',
  HP: 78,
  Attack: 130,
  Defense: 111,
  SpAtk: 130,
  SpDef: 85,
  Speed: 100,
  Generation: 1,
  Legendary: false
}
{
  _id: ObjectId('67edde1135bb6d6f6ebed724'),
  Num: 6,
```

```
  _id: ObjectId('67edde1135bb6d6f6ebed724'),
  Num: 6,
  Name: 'CharizardMega Charizard Y',
  Type1: 'Fire',
  Type2: 'Flying',
  HP: 78,
  Attack: 104,
  Defense: 78,
  SpAtk: 159,
  SpDef: 115,
  Speed: 100,
  Generation: 1,
  Legendary: false
}
{
  _id: ObjectId('67edde1135bb6d6f6ebed728'),
  Num: 9,
  Name: 'BlastoiseMega Blastoise',
  Type1: 'Water',
  HP: 79,
  Attack: 103,
  Defense: 120,
```

### 3- Encuentra los Pokémon cuya defensa esté entre 80 y 100 (inclusive).

```
> db.Pokemones.find(
  { Defense: { $gte: 80, $lte: 100 } }
)
< {
  _id: ObjectId('67edde1135bb6d6f6ebed71e'),
  Num: 3,
  Name: 'Venusaur',
  Type1: 'Grass',
  Type2: 'Poison',
  HP: 80,
  Attack: 82,
  Defense: 83,
  SpAtk: 100,
  SpDef: 100,
  Speed: 80,
  Generation: 1,
  Legendary: false
}
{
  _id: ObjectId('67edde1135bb6d6f6ebed726'),
  Num: 8,
  Name: 'Wartortle',
```

```
  _id: ObjectId('67edde1135bb6d6f6ebed727'),
  Num: 9,
  Name: 'Blastoise',
  Type1: 'Water',
  HP: 79,
  Attack: 83,
  Defense: 100,
  SpAtk: 85,
  SpDef: 105,
  Speed: 78,
  Generation: 1,
  Legendary: false
}
{
  _id: ObjectId('67edde1135bb6d6f6ebed733'),
  Num: 18,
  Name: 'PidgeotMega Pidgeot',
  Type1: 'Normal',
  Type2: 'Flying',
  HP: 83,
  Attack: 80,
  Defense: 80,
```

#### 4- Muestra el promedio de ataque de los Pokémon por tipo (Type1).

```
> db.Pokemones.aggregate([
  { $group: { _id: "$Type1", promedioAttack: { $avg: "$Attack" } } },
  {$sort: { promedioAttack: -1 } }])
K {
  _id: 'Dragon',
  promedioAttack: 112.125
}
{
  _id: 'Fighting',
  promedioAttack: 96.77777777777777
}
{
  _id: 'Ground',
  promedioAttack: 95.75
}
{
  _id: 'Rock',
  promedioAttack: 92.86363636363636
}
{
  _id: 'Steel',
  promedioAttack: 92.70370370370371
```

```
  _id: 'Dark',
  promedioAttack: 88.38709677419355
}
{
  _id: 'Fire',
  promedioAttack: 84.76923076923077
}
{
  _id: 'Flying',
  promedioAttack: 78.75
}
{
  _id: 'Poison',
  promedioAttack: 74.67857142857143
}
{
  _id: 'Water',
  promedioAttack: 74.15178571428571
}
}
```

## 5-Encuentra el Pokémon con más HP de cada tipo.

```
> db.Pokemones.aggregate([
  { $sort: { HP: -1 } },
  {
    $group: {
      _id: "$Type1",
      Nombre: { $first: "$Name" },
      MaxHP: { $first: "$HP" }
    }
  },
  { $sort: { MaxHP: -1 } }
])
< {
  _id: 'Normal',
  Nombre: 'Blissey',
  MaxHP: 255
}
{
  _id: 'Psychic',
  Nombre: 'Wobbuffet',
  MaxHP: 190
```

```
  _id: 'Water',
  Nombre: 'Wailord',
  MaxHP: 170
}
{
  _id: 'Ghost',
  Nombre: 'Drifblim',
  MaxHP: 150
}
{
  _id: 'Fighting',
  Nombre: 'Hariyama',
  MaxHP: 144
}
{
  _id: 'Dark',
  Nombre: 'Yveltal',
  MaxHP: 126
}
{
  _id: 'Fairy',
```

## 6- Muestra los 5 Pokémon más rápidos.

```
> db.Pokemones.aggregate([
  { $sort: { Speed: -1 } },
  { $limit: 5 }
]);
< {
  _id: ObjectId('67edde1135bb6d6f6ebed8cb'),
  Num: 386,
  Name: 'DeoxysSpeed Forme',
  Type1: 'Psychic',
  HP: 50,
  Attack: 95,
  Defense: 90,
  SpAtk: 95,
  SpDef: 90,
  Speed: 180,
  Generation: 3,
  Legendary: true
}
{
  _id: ObjectId('67edde1135bb6d6f6ebed857'),
  Num: 291
```

```
  Name: 'Ninjask',
  Type1: 'Bug',
  Type2: 'Flying',
  HP: 61,
  Attack: 90,
  Defense: 45,
  SpAtk: 50,
  SpDef: 50,
  Speed: 160,
  Generation: 3,
  Legendary: false
}
{
  _id: ObjectId('67edde1135bb6d6f6ebed763'),
  Num: 65,
  Name: 'AlakazamMega Alakazam',
  Type1: 'Psychic',
  HP: 55,
  Attack: 50,
  Defense: 65,
  SpAtk: 175,
```

```
  _id: ObjectId('67edde1135bb6d6f6ebed763'),
  Num: 65,
  Name: 'AlakazamMega Alakazam',
  Type1: 'Psychic',
  HP: 55,
  Attack: 50,
  Defense: 65,
  SpAtk: 175,
  SpDef: 95,
  Speed: 150,
  Generation: 1,
  Legendary: false
}
{
```

**7-Muestra el promedio de ataque de los Pokémon tipo “Water” ordenado de mayor a menor.**

```
> db.Pokemones.aggregate([
  { $match: { Type1: "Water" } },
  {
    $group: {
      _id: "$Generation",
      promedioAtaque: { $avg: "$Attack" }
    }
  },
  { $sort: { promedioAtaque: -1 } }
])
< {
  _id: 3,
  promedioAtaque: 80.66666666666667
}
```

```
  _id: 1,
  promedioAtaque: 74.19354838709677
}
{
  _id: 5,
  promedioAtaque: 73.27777777777777
}
{
  _id: 4,
  promedioAtaque: 72.46153846153847
}
{
  _id: 2,
  promedioAtaque: 68.11111111111111
}
{
  _id: 6,
  promedioAtaque: 68
}
```

## 8- Encuentra el Pokémon con más ataque por generación y ordénalos de mayor a menor.

```
> db.Pokemones.aggregate([
  { $group: { _id: "$Generation", maxAtaque: { $max: "$Attack" } } },
  { $sort: { maxAtaque: -1 } }
]);
< {
  _id: 1,
  maxAtaque: 190
}
{
  _id: 2,
  maxAtaque: 185
}
{
  _id: 3,
  maxAtaque: 180
}
{
  _id: 4,
  maxAtaque: 170
}
```

```
}
{
  _id: 4,
  maxAtaque: 170
}
{
  _id: 5,
  maxAtaque: 170
}
{
  _id: 6,
  maxAtaque: 160
}
```



## 9- Crea un índice en el campo Type1.

```
> db.pokemon.createIndex({ "Type1": 1, "attack": -1 })  
< Type1_1_attack_-1
```

## 10-Usa explain () para analizar el rendimiento de una búsqueda:

```
> db.Pokemons.find({ Type1: "Fire" }).explain("executionStats")
```

```
< {  
  explainVersion: '1',  
  queryPlanner: {  
    namespace: 'Pokemon.Pokemons',  
    parsedQuery: {  
      Type1: {  
        '$eq': 'Fire'  
      }  
    },  
    indexFilterSet: false,  
    queryHash: '135BE0C6',  
    planCacheShapeHash: '135BE0C6',  
    planCacheKey: '3B166ACA',  
    optimizationTimeMillis: 26,  
    maxIndexedOrSolutionsReached: false,  
    maxIndexedAndSolutionsReached: false,  
    maxScansToExplodeReached: false,
```

```
    prunedSimilarIndexes: false,  
    winningPlan: {  
      isCached: false,  
      stage: 'FETCH',  
      inputStage: {  
        stage: 'IXSCAN',  
        keyPattern: {  
          Type1: 1  
        },  
        indexName: 'Type1_1',  
        isMultiKey: false,  
        multiKeyPaths: {  
          Type1: []  
        },  
        isUnique: false,  
        isSparse: false,  
        isPartial: false,  
        indexVersion: 2,  
        direction: 'forward',  
        indexBounds: {
```

## 11- Crea un índice compuesto en Type1 y Speed, y analiza una búsqueda:

```
> db.Pokemones.createIndex({ Type1: 1, Speed: -1 })
< Type1_1_Speed_-1
> db.Pokemones.find({ Type1: "Electric" })
  .sort({ Speed: -1 })
  .explain("executionStats")
< {
  explainVersion: '1',
  queryPlanner: {
    namespace: 'Pokemon.Pokemones',
    parsedQuery: {
      Type1: {
        '$eq': 'Electric'
      }
    },
    indexFilterSet: false,
    queryHash: '93BBCE94',
```