Roberto Carlos Ávila Crespo
1º ASIR

Proyecto Base de Datos 2º Trimestre

- *Explicación de la base de datos:*

Esta base de datos trata de una empresa en la que existen dos colecciones, la primera colección tiene almacenados los datos de los trabajadores y en la segunda se guardan los datos de los distintos departamentos que existen dentro de la empresa.

1º) Primera Colección:
Esta colección tiene primeramente un identificador que establezco para no dejar que se ponga automáticamente, luego tiene el nombre del trabajador, los apellidos, la fecha de contratación del empleado, el teléfono, el cargo que tiene dentro de la empresa, dentro de el cargo el empleado puede tener más de un cargo en la empresa, es decir, se trata de un array y para terminar la colección, el último dato es un booleano, y lo que recoge es si el empleado posee un coche de empresa o no.

2º) Segunda Colección:
Esta segunda colección lo que guarda es los datos de cada departamento.
En primer lugar tenemos el identificador igual que en la primera consulta, luego el segundo dato es el nombre del departamento, luego PrecioHora se refiere a cuanto se cobraría la hora trabajada en dicho departamento, la Jornada, son las horas que se trabajan en este departamento al día y los días trabajados son todos los días laborables que tiene cada departamento por mes.

- *Explicación de los tipos de consultas:*

En mi proyecto se usan la mayoría o todos los operadores estudiados tanto en clases presenciales como en la semana de trabajo para casa, hay dos tipos de consultas, el primer tipo son aquellas llamadas *aggregate,* que son las que solo usan una única colección y en las que se hace digamos un "filtrado" rápido en el que también podemos introducir operaciones matemáticas entre otras cosas.

El segundo tipo de consulta viene a ser lo mismo que la anterior pero con una diferencia muy notable, en esta consulta podremos unir las dos colecciones que tenemos en nuestra base de datos con el operador $lookup (que explicaré más tarde en el apartado de comandos), con esto lo que nos facilita es el poder hacer consultas con datos de ambas colecciones a la misma vez y así de este modo manejar mucha más información.

- *Explicación de los operadores usados:*

→ $lookup:

El operador $lookup es aquel que nos permite unir dos colecciones introduciendo los nombres de cada una y luego un localfield y un foreingfield.

→ $group:

El operador $group agrupa los documentos de entrada por la _id especificada y para cada agrupación distinta, genera un documento. El _id de cada documento de salida contiene el grupo único por valor. Los documentos de salida también pueden contener campos calculados

→ $project:

El operador $project nos permitirá mostrar los datos que nosotros queramos dentro de nuestras colecciones además de permitirnos introducir campos nuevos, los campos que queramos visualizar se representarán con un 1 y los que no con un 0.

→ $sort:

El operador $sort nos permite organizar de forma ascendente o descente, también representado con un 1 (ascendente) y con un -1 (descentente)

→ $match:

Filtra los documentos para solo dejar pasar aquellos que cumplen con las especificaciones indicadas en el .

- *Explicación de las consultas:*

1º Consulta:

En la primera consulta, uno las dos colecciones de mi base de datos y me muestra primero una serie de datos de la primera colección y luego dentro de departamentos, me da dos datos más como son el nombre del departamento y el precio al que se paga la hora en el.

```
switched to db proyecto2trim
> db.empleados2.aggregate(
...     [
...         {
...             $lookup: {
...                 from: "departamentos2",
...                 localField: "Cargo",
...                 foreignField: "Departamento",
...                 as: "departamentos"
...             }
...         },
...         {
...             $project: {
...                 _id: 1,
...                 Nombre: 1,
...                 Apellidos: 1,
...                 Coche_empresa: 1,
...                 departamentos: {
...                     Departamento: 1,
...                     PrecioHora: 1,
...                 },
...             }
...         },
...         {
...             $sort: { Nombre: 1 }
...         }
...
...     ]
... ).pretty()
{
        "_id" : 4,
        "Nombre" : "Augusto",
        "Apellidos" : "Figueras Coronell",
        "Coche_empresa" : false,
        "departamentos" : [
                {
                        "Departamento" : "Camionero",
                        "PrecioHora" : 9
                }
        ]
}
{
        "_id" : 3,
        "Nombre" : "Daniel",
        "Apellidos" : "Guisado Oliva",
        "Coche_empresa" : false,
        "departamentos" : [
                {
                        "Departamento" : "Camionero",
                        "PrecioHora" : 9
                }
        ]
}
{
        "_id" : 7,
        "Nombre" : "Francisco",
        "Apellidos" : "Montes Leon",
        "Coche_empresa" : false,
        "departamentos" : [
                {
                        "Departamento" : "Montador",
                        "PrecioHora" : 8
```

2º Consulta:

En la segunda consulta busco tener ordenados por cada departamento, saber cual es la jornada diaria en cada departamento, cuántos días se trabaja a la semana y el total de horas que el trabajador realiza en un mes en un departamento concreto.
Esto lo haré siempre y cuando el trabajador esté contratado como mínimo 5 horas es decir, no esté contratado a media jornada.
Para acabar, ordenaremos según el número de horas trabajadas al mes de forma ascendente.

```
> db.departamentos2.aggregate([
...     {
...          $project:{
...            _id:0,
...            Departamento: "$Departamento",
...            Jornada: "$Jornada",
...            DiasTrabajados: "$DiasTrabajados",
...            TotalHorasalMes:{$multiply:["$Jornada", "$DiasTrabajados"]},
...      }
...      },
...      {
...           $match: { $expr: { $gte: [ "$Jornada" , 5 ] }},
...
...      },
...      {
...          $sort: { TotalHorasalMes: 1 }
...      }
... ]).pretty()
{
        "Departamento" : "Direccion",
        "Jornada" : 8,
        "DiasTrabajados" : 20,
        "TotalHorasalMes" : 160
}
{
        "Departamento" : "Comercial",
        "Jornada" : 8,
        "DiasTrabajados" : 20,
        "TotalHorasalMes" : 160
}
{
        "Departamento" : "Montador",
        "Jornada" : 8,
        "DiasTrabajados" : 24,
        "TotalHorasalMes" : 192
}
{
        "Departamento" : "Camionero",
        "Jornada" : 8,
        "DiasTrabajados" : 24,
        "TotalHorasalMes" : 192
}
>
```

3º Consulta:

En la tercera consulta, quiero saber, de los empleados contratados como camioneros, cuantos años llevan trabajando en dicha empresa para lo que realizaré una operación que automáticamente me calculará desde la fecha de contratación hasta la de hoy en dia cuantos son los años transcurridos.
Una vez se sepan esos años los ordenaremos según la antigüedad en la empresa en orden descendente.

```
> db.empleados2.aggregate(
...     [ { $match: {"Cargo":{$eq:"Camionero"}}},
...         {
...         $project:
...           { _id:"$Nombre",
...             "añostrabajados": {"$divide":[
...                 { $subtract:
...                  [ new Date(), "$Fecha_contratacion" ]  }, 1000 * 60 * 60 * 24 * 365]},
...         }
...       },
...       { $sort : { añostrabajados : -1 } }
...     ]
...  ).pretty()
{ "_id" : "Daniel", "añostrabajados" : 6.911528186992643 }
{ "_id" : "Miguel", "añostrabajados" : 5.8320761321981225 }
{ "_id" : "Augusto", "añostrabajados" : 3.1553638034309994 }
{ "_id" : "Jeronimo", "añostrabajados" : 2.9142679130200406 }
>
```