



PROYECTO FINAL Garantías (Finanzas)

Introducción a Bases de Datos 2020
data-analysis-gdl-20-04

Mauricio Millán Solano

Contenido

Introducción	1
Diagrama MySQL.....	1
Definición de tablas.....	2
AGENTE	2
Configuración	2
Empresas	2
Oficinas.....	3
Polizas.....	3
Consultas MYSQL Y Mongo DB.....	5
1. ¿cuáles son las polizas con la garantía 1?	5
MYSQL	5
MongoDB.....	5
2. ¿Cuál es la poliza con el mayor monto del 2020?	5
MYSQL	5
MongoDB.....	6
3. cuáles son las pólizas que tienen fecha en Ultimo estado y tenga fecha fin	6
MYSQL	6
MongoDB.....	7
4. Obtener las polizas que tengan en TipoUltimoEstado 'Auditada o dicataminada'.....	8
MYSQL	8
MongoDB.....	8
5. ¿Cual es el porcentaje del monto retenido? (se divide el monto afianzado entre el monto retenido y se visualiza en porcentaje) (MontoAfianzado/MontoRetenido)/100	8
MYSQL	8
MongoDB.....	9
6. Obtener el top 5 de vendedores con mayor n° de polizas en 2020.....	10
MYSQL	10
MongoDB.....	10
7. ¿Cuántas pólizas se emitieron en el 2020?	11
MYSQL	11
MongoDB.....	12
8. Buscar el top 5 de empresas con mayor monto Afianzado	12

MySQL	12
MongoDB.....	13
9. Buscar el top 5 de empresas con mayor monto Retenido	13
MySQL	14
MongoDB.....	14
10. Obtener las obligaciones de las pólizas y su montocubierto con fecha ultimoestado ordenados por ultimoestado;	14
MySQL	15
MongoDB.....	15
11. Empresas con oficinas en REFORMA, SUC.MEXICO CENTRO , 'SUC.MEXICO CENTRO' ...	16
MySQL	16
MongoDB.....	16
12. Pólizas que pertenecen al ramo administrativo con sus respectivos montos	17
MySQL	17
MongoDB.....	17
13. Obtener los montos ordenados por ultimoEstado de fecha más reciente a más antigua de cada empresa por oficina	18
MySQL	18
MongoDB.....	19
14. Obtener MontoAfianzado total, MontoCubierto total y PorcentajeRetenido agrupado por Oficina por el SubRamo 'DE Obra'	20
MySQL	20
MongoDB.....	20
15. Obtener MontoAfianzado total, MontoCubierto total y PorcentajeRetenido en un top 10 de menor a mayor por empresa pertenecientes al RAMO 'Crédito'	21
MySQL	21
MongoDB.....	22
Vistas	24
MySQL	24
MongoDB	26
Conclusiones	28

Introducción

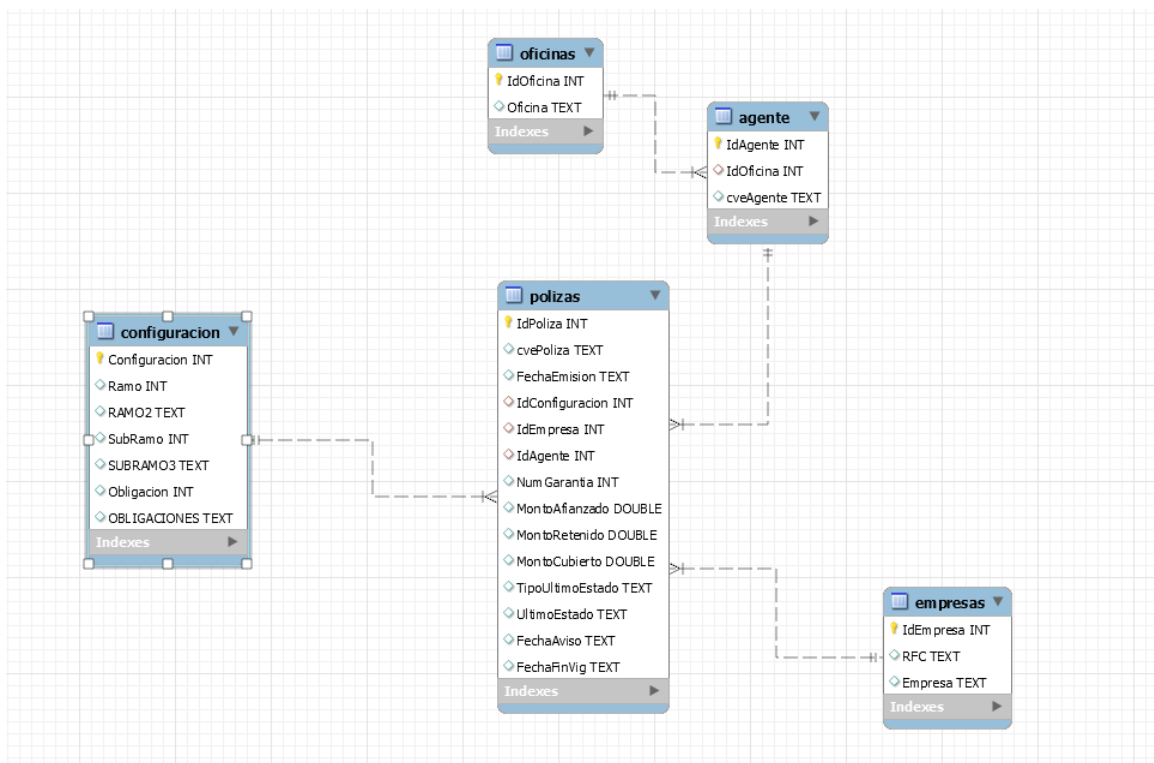
El actual documento se realizó con fines de practica a partir de una base de datos ajena al curso, dicha base de datos elegida para este proyecto fue de Garantías (Finanzas) utilizando como última fecha de información el mes de junio.

Elegí esta base de datos puesto que es un fragmento de la información que utiliza mi papá para su trabajo. El fin de hacer este proyecto con dicha información es poder demostrar que el utilizar las consultas desde un gestor de datos, se puede facilitar el obtener la información que se requiere para hacer un análisis.

Antes de subir la base de datos a MySQL y a MongoDB, se realizó un tratamiento de datos para estructurar la información en tablas relacionales para MySQL y reemplazar datos sensibles por datos "Dummy". Fue necesario realizar dicho proceso puesto que la información proviene de un archivo de Excel (una tabla tabular).

Los datos que se reemplazaron fueron las empresas con su correspondiente RFC. Las empresas se generaron con un número incremental "Empresa1, Empresa2...Empresa(n)", el RFC se realizó por medio de una función Random (), con máximo 13 caracteres.

Diagrama MySQL



Definición de tablas

AGENTE

```
CREATE TABLE `agente` (  
  `IdAgente` int NOT NULL,  
  `IdOficina` int DEFAULT NULL,  
  `cveAgente` text,  
  PRIMARY KEY (`IdAgente`),  
  KEY `IdOficina` (`IdOficina`),  
  CONSTRAINT `agente_ibfk_1` FOREIGN KEY (`IdOficina`) REFERENCES `oficinas` (`IdOficina`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Configuración

```
CREATE TABLE `configuracion` (  
  `Configuracion` int NOT NULL,  
  `Ramo` int DEFAULT NULL,  
  `RAMO2` text,  
  `SubRamo` int DEFAULT NULL,  
  `SUBRAMO3` text,  
  `Obligacion` int DEFAULT NULL,  
  `OBLIGACIONES` text,  
  PRIMARY KEY (`Configuracion`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Empresas

```
CREATE TABLE `empresas` (  
  `IdEmpresa` int NOT NULL,  
  `RFC` text,  
  `Empresa` text,
```

```
PRIMARY KEY (`IdEmpresa`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Oficinas

```
CREATE TABLE `oficinas` (  
  `IdOficina` int NOT NULL,  
  `Oficina` text,  
  PRIMARY KEY (`IdOficina`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Polizas

```
CREATE TABLE `polizas` (  
  `IdPoliza` int NOT NULL,  
  `cvePoliza` text,  
  `FechaEmision` datetime DEFAULT NULL,  
  `IdConfiguracion` int DEFAULT NULL,  
  `IdEmpresa` int DEFAULT NULL,  
  `IdAgente` int DEFAULT NULL,  
  `NumGarantia` int DEFAULT NULL,  
  `MontoAfianzado` double DEFAULT NULL,  
  `MontoRetenido` double DEFAULT NULL,  
  `MontoCubierto` double DEFAULT NULL,  
  `TipoUltimoEstado` text,  
  `UltimoEstado` datetime DEFAULT NULL,  
  `FechaAviso` datetime DEFAULT NULL,  
  `FechaFinVig` datetime DEFAULT NULL,  
  PRIMARY KEY (`IdPoliza`),  
  KEY `IdAgente` (`IdAgente`),  
  KEY `IdEmpresa` (`IdEmpresa`),
```

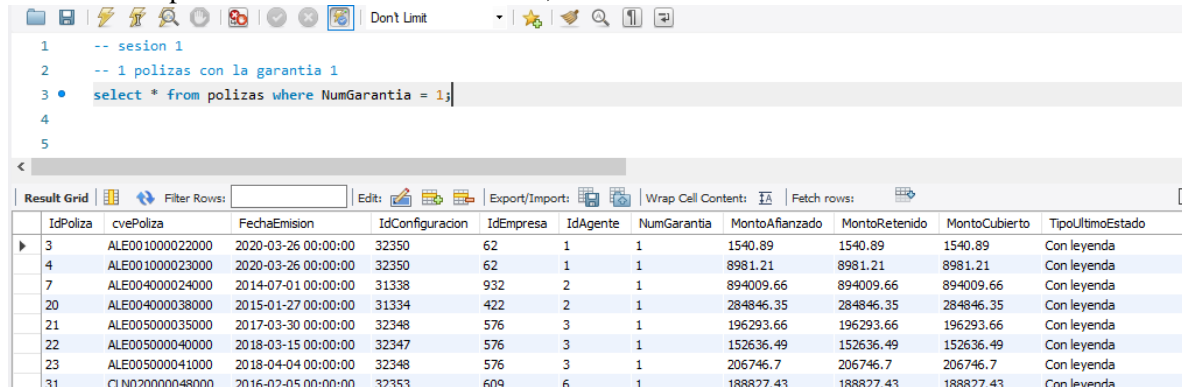
```
CONSTRAINT `polizas_ibfk_1` FOREIGN KEY (`IdAgente`) REFERENCES `agente`  
(`IdAgente`),  
CONSTRAINT `polizas_ibfk_2` FOREIGN KEY (`IdEmpresa`) REFERENCES `empresas`  
(`IdEmpresa`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Consultas MYSQL Y Mongo DB

1. ¿cuáles son las polizas con la garantía 1?

MYSQL

```
select * from polizas where NumGarantia = 1;
```



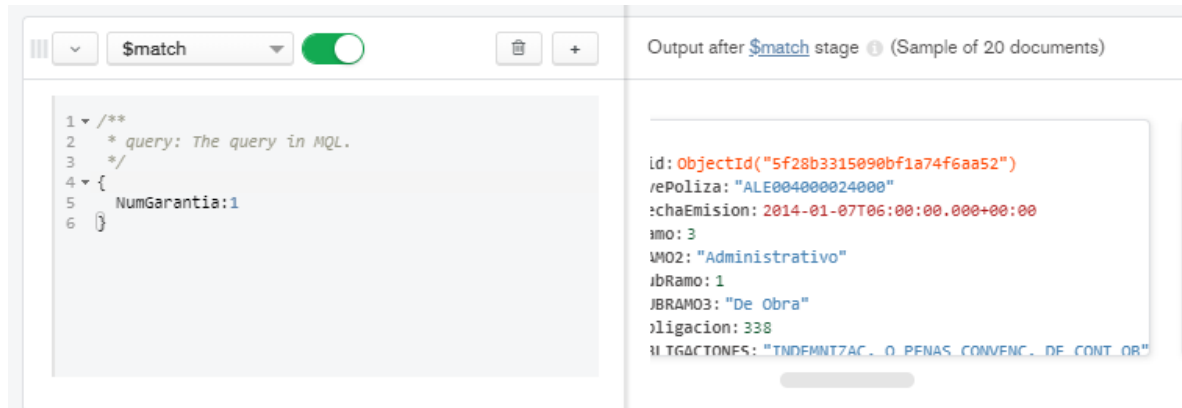
	IdPoliza	cvePoliza	FechaEmision	IdConfiguracion	IdEmpresa	IdAgente	NumGarantia	MontoAfianzado	MontoRetenido	MontoCubierto	TipoUltimoEstado
3	ALE001000022000	2020-03-26 00:00:00	32350	62	1	1	1	1540.89	1540.89	1540.89	Con leyenda
4	ALE001000023000	2020-03-26 00:00:00	32350	62	1	1	1	8981.21	8981.21	8981.21	Con leyenda
7	ALE004000024000	2014-07-01 00:00:00	31338	932	2	1	1	894009.66	894009.66	894009.66	Con leyenda
20	ALE004000038000	2015-01-27 00:00:00	31334	422	2	1	1	284846.35	284846.35	284846.35	Con leyenda
21	ALE005000035000	2017-03-30 00:00:00	32348	576	3	1	1	196293.66	196293.66	196293.66	Con leyenda
22	ALE005000040000	2018-03-15 00:00:00	32347	576	3	1	1	152636.49	152636.49	152636.49	Con leyenda
23	ALE005000041000	2018-04-04 00:00:00	32348	576	3	1	1	206746.7	206746.7	206746.7	Con leyenda
31	CLN02000048000	2016-02-05 00:00:00	32353	609	6	1	1	188827.43	188827.43	188827.43	Con leyenda

MongoDB

```
{{ $match: {
```

```
  NumGarantia:1
```

```
}}}
```



Output after \$match stage (Sample of 20 documents)

```
{
  "id": "ObjectId('5f28b3315090bf1a74f6aa52')",
  "rePoliza": "ALE004000024000",
  "fechaEmision": "2014-01-07T06:00:00.000+00:00",
  "nmo": 3,
  "nmo2": "Administrativo",
  "jbramo": 1,
  "jbramo3": "De Obra",
  "obligacion": 338,
  "obligaciones": "INDEMNIZACION DE PENAS CONVENCIONALES DE CONT. OB"
}
```

2. ¿Cuál es la poliza con el mayor monto del 2020?

MYSQL

```
select * from polizas where year(FechaEmision) = year(curdate()) order by MontoAfianzado desc limit 1;
```



```

5
6 -- 2 cual es la poliza con el mayor monto del 2020
7 • select * from polizas where year(FechaEmision) = year(curdate())
8   order by MontoAfianzado desc limit 1;
9

```

IdPoliza	cvePoliza	FechaEmision	IdConfiguracion	IdEmpresa	IdAgente	NumGarantia	MontoAfianzado	MontoRetenido	MontoCubierto	TipoUltimoEstado
915	DFR001500002530	2020-05-22 00:00:00	41410	202	7	1	171000000	1655271.95	171000000	Sin leyenda

MongoDB

```

[{$project: {
  _id:0,
  cvePoliza:1,
  FechaEmision:1,
  MontoAfianzado:1
}}, {$match: {
  FechaEmision :
    {$gte: ISODate("2020-01-01T00:00:00.000+00:00")}
}}, {$sort: {
  MontoAfianzado: -1
}}, {$limit: 1}]

```

Output after **\$limit** stage (Sample of 1 document)

```

1 //**
2 * Provide the number of documents to limit.
3 */
4 1

```

```

cvePoliza: "DFR001500002530"
FechaEmision: 2020-05-22T00:00:00.000+00:00
MontoAfianzado: 171000000

```

3. cuáles son las pólizas que tienen fecha en Ultimo estado y tenga fecha fin

MYSQL

```
select * from polizas where UltimoEstado is not null and FechaFinVig is not null;
```

```

9
10
11 -- 3 cuales son las polizas que tienen fecha en Ultimo estado y tenga fecha fin
12 • select * from polizas where UltimoEstado is not null and FechaFinVig is not null;
13

```

	IdPoliza	cvePoliza	FechaEmision	IdConfiguracion	IdEmpresa	IdAgente	NumGarantia	MontoAfianzado	MontoRetenido	MontoCub
▶	915	DFR001500002530	2020-05-22 00:00:00	41410	202	7	1	171000000	1655271.95	171000000
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

MongoDB

```

[{$project: {
  _id:0,
  cvePoliza:1,
  MontoAfianzado:1,
  FechaEmision:1,
  UltimoEstado:1
}}, {$match: {
  FechaEmision: {
    $gt: ISODate('1970-01-01T00:00:00.000+00:00')
  },
  UltimoEstado:{
    $gt: ISODate('1970-01-01T00:00:00.000+00:00')
  }
}}

```

Output after `$match` stage (Sample of 20 documents)

```

1 {
2   FechaEmision: {
3     $gt: ISODate('1970-01-01T00:00:00.000+00:00')
4   },
5   UltimoEstado:{
6     $gt: ISODate('1970-01-01T00:00:00.000+00:00')
7   }
8 }
9
10

```

```

cvePoliza: "ALE001000022000"
FechaEmision: 2020-03-26T00:00:00.000+00:00
MontoAfianzado: 1540.89
UltimoEstado: 2019-12-31T00:00:00.000+00:00

```

4. Obtener las polizas que tengan en TipoUltimoEstado 'Auditada o dictaminada'

MYSQL

select * from polizas where TipoUltimoEstado = 'Auditada o dictaminada';

```

13
14
15 -- 4 obtener las polizas que tengan en TipoUltimoEstado 'Auditada o dictaminada'
16 • select * from polizas where TipoUltimoEstado = 'Auditada o dictaminada';
17

```

	IdPoliza	cvePoliza	FechaEmision	IdConfiguracion	IdEmpresa	IdAgente	NumGarantia	MontoAfianzado	MontoRetenido	MontoCubierto	TipoUltimoEstado
▶	26	CLN001000449000	2004-01-30 00:00:00	31335	830	4	2	15015139.98	7220925.98	15015139.98	Auditada o dictaminada
	36	DFR001005820000	2003-10-31 00:00:00	31335	830	4	0	9000000	6558129	9000000	Auditada o dictaminada
	37	DFR001005821000	2003-10-31 00:00:00	31334	830	4	0	6000000	6000000	6000000	Auditada o dictaminada
	47	DFR001010287000	2010-05-31 00:00:00	22214	525	7	2	6270	6270	6270	Auditada o dictaminada
	51	DFR001011184000	2011-12-22 00:00:00	32348	421	7	3	19922761.2	16420104.29	19922761.2	Auditada o dictaminada
	55	DFR001011484000	2012-06-27 00:00:00	33309	245	7	2	2506428.03	2506428.03	2506428.03	Auditada o dictaminada
	56	DFR001011498000	2012-07-16 00:00:00	32348	421	7	3	3473560.43	3473560.43	3473560.43	Auditada o dictaminada
	60	DFR001011650000	2012-10-02 00:00:00	22213	184	7	2	667261.6	667261.6	667261.6	Auditada o dictaminada
	136	DFR001012612000	2014-09-01 00:00:00	22216	525	7	2	121122	121122	121122	Auditada o dictaminada
	137	DFR001012613000	2014-09-01 00:00:00	22216	525	7	2	121122	121122	121122	Auditada o dictaminada
	143	DFR001012693000	2014-10-23 00:00:00	33309	418	28	1	84785.4	84785.4	84785.4	Auditada o dictaminada
	146	DFR001012710000	2014-11-05 00:00:00	33309	418	28	1	90403.76	90403.76	90403.76	Auditada o dictaminada
	164	DFR001012874000	2015-02-17 00:00:00	32348	942	7	1	2333331	2333331	2333331	Auditada o dictaminada
	165	DFR001012897000	2015-03-12 00:00:00	32353	942	7	1	2900	2900	2900	Auditada o dictaminada
	170	DFR001012923000	2015-04-08 00:00:00	32353	942	7	1	20000	20000	20000	Auditada o dictaminada
	173	DFR001012940000	2015-04-13 00:00:00	32348	942	7	1	667419.88	667419.88	667419.88	Auditada o dictaminada
	175	DFR001012943000	2015-04-15 00:00:00	32348	942	7	1	14351.82	14351.82	14351.82	Auditada o dictaminada

MongoDB

```

{{
  $match: {
    TipoUltimoEstado: "Auditada o dictaminada"
  }
}}

```

Output after \$match stage (Sample of 20 documents)

<pre> 1 //** 2 * query: The query in MQL. 3 */ 4 { 5 TipoUltimoEstado:"Auditada o dictaminada" 6 } </pre>	<pre> Oficina:"CULIACAN" NumGarantia:2 MontoAfianzado:15015139.98 MontoRetenido:7220925.98 MontoCubierto:15015139.98 TipoUltimoEstado:"Auditada o dictaminada" UltimoEstado:2007-12-31T00:00:00.000+00:00 FechaAviso:2019-12-31T00:00:00.000+00:00 FechaFinVig:2016-07-12T00:00:00.000+00:00 </pre>	<pre> Oficina:"CULIACAN" NumGarantia:0 MontoAfianzado:9000000 MontoRetenido:6558129 MontoCubierto:9000000 TipoUltimoEstado:"Auditada o dictaminada" UltimoEstado:1978-01-01T00:00:00.000+00:00 FechaAviso:1978-01-01T00:00:00.000+00:00 FechaFinVig:2018-01-14T00:00:00.000+00:00 </pre>
---	---	--

5. ¿Cual es el porcentaje del monto retenido? (se divide el monto afianzado entre el monto retenido y se visualiza en porcentaje) (MontoAfianzado/MontoRetenido)/100

MYSQL

select cvePoliza, MontoRetenido, MontoAfianzado, concat(round((MontoRetenido/MontoAfianzado * 100),2),'%') as PerMontoRetenido from polizas order by MontoRetenido/MontoAfianzado * 100) asc;

```

18 -- 5 Cual es el porcentaje del monto retenido
19 -- (se divide el monto afianzado entre el monto retenido y se visualiza en porcentaje)
20 -- (MontoAfianzado/MontoRetenido)/100
21 • select cvePoliza, MontoRetenido, MontoAfianzado,
22 concat(round(( MontoRetenido/MontoAfianzado * 100 ),2),'%') as PerMontoRetenido
23 from polizas order by (MontoRetenido/MontoAfianzado * 100) asc;

```

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content: Fetch rows:				
	cvePoliza	MontoRetenido	MontoAfianzado	PerMontoRetenido
▶	DFR001500002530	1655271.95	171000000	0.97%
	DFR001014205000	1349875.08	17356192.13	7.78%
	DFR001014206000	1349875.08	17356186.38	7.78%
	QRO050000016000	1778386.39	21993164.15	8.09%
	DFR001300100185	1693729.33	20000000	8.47%
	DFR001300100184	1693729.32	17500000	9.68%
	DFR001500002373	1636884.53	16161616.16	10.13%
	DFS001001779000	1655271.93	13000000	12.73%
	VER002003904000	1655271.95	13000000	12.73%
	DFR134000003000	1693729.33	13125000	12.90%
	DFR001014396000	1693729.31	13000000	13.03%
	DFR001500002491	1693729.33	12424242.42	13.63%
	GDL006001208000	1655271.95	12100000	13.68%
	DFR001500002467	1693729.3	12000000	14.11%

MongoDB

```
{ $addFields: {
```

```
  PorcentajeRetenido: { $multiply: [
```

```
    { $divide: [ "$MontoRetenido", "$MontoAfianzado" ] }, 100 ]
```

```
  } }, { $sort: {
```

```
    PorcentajeRetenido: 1
```

```
  } }, { $project: {
```

```
    _id: 0,
```

```
    cvePoliza: 1,
```

```
    MontoAfianzado: 1,
```

```
    MontoRetenido: 1,
```

```
    Porcentaje:
```

```
    { $concat: [
```

```
      { $toString: [
```

```
        { $round: [
```

```
          "$PorcentajeRetenido", 2 ] } ] ] }
```

```

    ,"%"]}}
  }}

```

The screenshot shows the MongoDB Atlas pipeline editor. On the left, the configuration for the '\$project' stage is shown, with a toggle switch turned on. The configuration includes a comment and a document structure: `{_id:0, cvePoliza:1, MontoAfianzado:1, MontoRetenido:1, Porcentaje:{$concat:[$toString:{$round:[$PorcentajeRetenido,2]}},"%"]}}`. On the right, the output after the '\$project' stage is displayed as a sample of 20 documents. The output document is: `{cvePoliza:"DFR001500002530", MontoAfianzado:171000000, MontoRetenido:1655271.95, Porcentaje:"0.97%"}`.

6. Obtener el top 5 de vendedores con mayor n° de polizas en 2020

MYSQL

select p.IdAgente, (select cveAgente from agente as a where a.IdAgente = p.IdAgente) as ClaveAgente, year(FechaEmision) as Año, count(*) as totalPolizas from polizas as p where FechaEmision > '2020/01/01' group by IdAgente order by totalPolizas desc limit 5 ;

```

27 -- 6 el vendedor con mayor n° de polizas en 2020
28 • select p.IdAgente, (select cveAgente from agente as a
29   where a.IdAgente = p.IdAgente) as ClaveAgente, year(FechaEmision) as Año,
30   from polizas as p where FechaEmision > '2020/01/01'
31   group by IdAgente order by totalPolizas desc limit 5 ;
32

```

	IdAgente	ClaveAgente	Año	totalPolizas
▶	75	QRO001	2020	86
	7	DFR001	2020	50
	28	DFS001	2020	23
	54	MTY001	2020	22
	48	HER001	2020	21

MongoDB

```

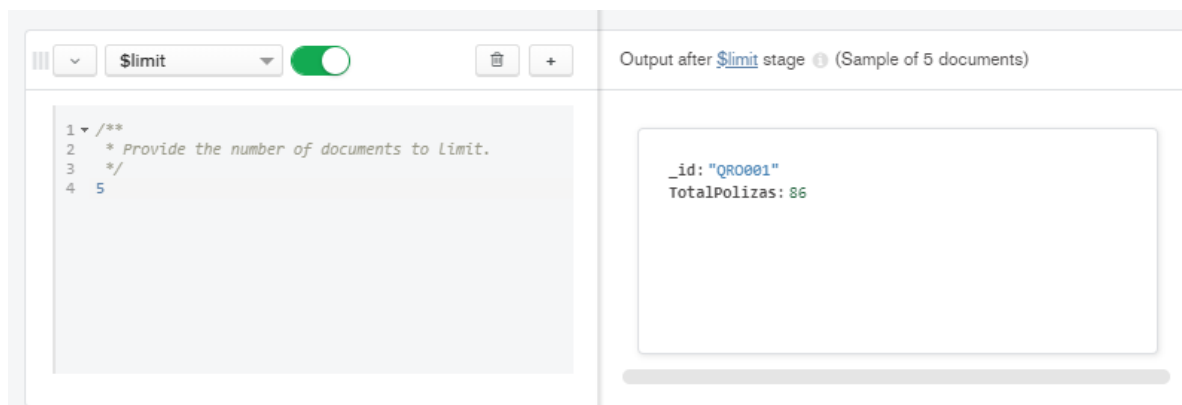
[{$match: {
  FechaEmision: {
    $gte: ISODate('2020-01-01T00:00:00.000+00:00')
  }
}

```

```

}}, {$project: {
  _id:0,
  cveAgente:1,
  FechaEmision:1
}}, {$group: {
  _id: "$cveAgente",
  TotalPolizas: {
    $sum:1
  }
}}, {$sort: {
  TotalPolizas: -1
}}, {$limit: 5}]

```



7. ¿Cuántas pólizas se emitieron en el 2020?

MYSQL

```
select count(*) as totalPolizas, year(FechaEmision) Año from polizas
```

```
where FechaEmision > '2020/01/01' group by Año order by totalPolizas desc;
```

```

33      -- 7 ¿Cuántas polizas se emitieron en el 2020?
34      • select count(*) as totalPolizas, year(FechaEmision) Año from polizas
35      where FechaEmision > '2020/01/01' group by Año order by totalPolizas desc;
36

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
totalPolizas	Año			
376	2020			

MongoDB

```
[{$match: {  
  FechaEmision: {  
    $gte: ISODate('2020-01-01T00:00:00.000+00:00')  
  }  
}}, {$project: {  
  _id:0,  
  cvePoliza:1,  
  year: {$year:"$FechaEmision"}  
}}, {$group: {  
  _id: "$year",  
  TotalPolizas: {  
    $sum:1  
  }  
}}]
```

The screenshot shows the MongoDB Compass interface. On the left, the aggregation pipeline is defined with a \$group stage. The right pane shows the output after the \$group stage, displaying a single document for the year 2020 with a total of 376 policies.

```
1 //**  
2 * _id: The id of the group.  
3 * fieldN: The first field name.  
4 */  
5 {  
6   _id: "$year",  
7   TotalPolizas: {  
8     $sum:1  
9   }  
10 }
```




Output after \$group stage (Sample of 1 document)

```
{  
  "_id": 2020,  
  "TotalPolizas": 376  
}
```

8. Buscar el top 5 de empresas con mayor monto Afianzado

MYSQL

```
select IdEmpresa, sum(MontoAfianzado) TotalAfianzado from polizas group by IdEmpresa order  
by TotalAfianzado desc limit 5;
```

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

	IdEmpresa	TotalAfianzado
▶	202	171000000
	656	167062324.47
	947	115190318.87
	123	78109325.79
	664	71000000

```
[{$project: {
  _id:0,
  MontoAfianzado:1,
  cveFiado:1
}}, {$group: {
  _id:"$cveFiado"
,   TotalAfianzado: {
    $sum:"$MontoAfianzado"
  }
}}, {$sort: {
  TotalAfianzado: -1
}}, {$limit: 5}]
```

The screenshot shows the MongoDB Compass interface. On the left, the 'Filter' tab is active, displaying a JSON filter: `{ "limit": 5 }`. The 'Output after \$limit stage' (Sample of 5 documents) is shown on the right. The output consists of two documents:

```
{ "_id": "CON160815AJ3",  
  "TotalAfianzado": 171000000 }  
{ "_id": "NIN910514BE0",  
  "TotalAfianzado": 167062324.47 }
```

13

MYSQL

select IdEmpresa, sum(MontoRetenido) TotalRetenido from polizas group by IdEmpresa order by TotalRetenido desc limit 5;

```
41
42 -- 9 buscar el top 5 de empresas con mayor monto Retenido
43 • select IdEmpresa, sum(MontoRetenido) TotalRetenido
44   from polizas group by IdEmpresa order by TotalRetenido desc limit 5;
45
```

IdEmpresa	TotalRetenido
656	73727317.39999999
123	66107629.8
947	55817823.00000001
387	55592229.839999996

MongoDB

```
{{ $project: {
  _id: 0,
  MontoRetenido: 1,
  cveFiado: 1
}}, { $group: {
  _id: "$cveFiado"
, TotalAfianzado: {
  $sum: "$MontoRetenido"
}
}}, { $sort: {
  TotalAfianzado: -1
}}, { $limit: 5 }}
```

<div>1 //** 2 * Provide the number of documents to limit. 3 */ 4 5</div>	<div>Output after \$limit stage (Sample of 5 documents)</div> <div><div><div>_id: "NIN9105148E0" TotalAfianzado: 73727317.40</div><div>_id: "CBB820227AC3" TotalAfianzado: 66107629.80</div></div></div>
--	--

10. Obtener las obligaciones de las polizas y su montocubierto con fecha ultimoestado ordenados por ultimoestado;

MYSQL

select IdPoliza, cvePoliza, (select OBLIGACIONES from configuracion as C where .Configuracion = P.IdConfiguracion) as Obligaciones, MontoCubierto, UltimoEstado from polizas as P where UltimoEstado is not null order by UltimoEstado desc;

```
46 -- 10 obtener las obligaciones de las polizas y su montocubierto con fecha ultimoestado ordenados por ultimoestado;
47 • select IdPoliza, cvePoliza,
48 (select OBLIGACIONES from configuracion as C where C.Configuracion = P.IdConfiguracion) as Obligaciones
49 , MontoCubierto, UltimoEstado from polizas as P where UltimoEstado is not null order by UltimoEstado desc;
50
```

IdPoliza	cvePoliza	Obligaciones	MontoCubierto	UltimoEstado
590	DFR001014395000	SUMINISTRO NO PEMEX (ESTACION DE SERVICIO)	7200000	2020-04-30 00:00:00
591	DFR001014396000	SUMINISTRO NO PEMEX (ESTACION DE SERVICIO)	13000000	2020-04-30 00:00:00
603	DFR001014409000	SUMINISTRO NO PEMEX (ESTACION DE SERVICIO)	6000000	2020-04-30 00:00:00
672	DFR001014510000	SUMINISTRO NO PEMEX (ESTACION DE SERVICIO)	12000000	2020-04-30 00:00:00
3386	VER002003865000	CUMPLIMIENTO PROVEEDORES	127219.79	2020-04-30 00:00:00
3402	VER002003903000	SUMINISTRO NO PEMEX (ESTACION DE SERVICIO)	7200000	2020-04-30 00:00:00
3403	VER002003904000	SUMINISTRO NO PEMEX (ESTACION DE SERVICIO)	13000000	2020-04-30 00:00:00
142	DFR001012692000	CUMPLIMIENTO PROVEEDORES	422450	2020-03-31 00:00:00
259	DFR001013370000	CUMPLIMIENTO PROVEEDORES	250000	2020-03-31 00:00:00
275	DFR001013446000	DE AMPARO	81000	2020-03-31 00:00:00
283	DFR001013496000	CUMPLIMIENTO PROVEEDORES	25000	2020-03-31 00:00:00
284	DFR001013497000	CUMPLIMIENTO PROVEEDORES	25000	2020-03-31 00:00:00
285	DFR001013498000	CUMPLIMIENTO PROVEEDORES	25000	2020-03-31 00:00:00

MongoDB

```
{{$project: {
  _id:0,
  cvePoliza:1,
  OBLIGACIONES:1,
  MontoCubierto:1,
  UltimoEstado:1
}}, {$match: {
  UltimoEstado:{
    $gt: ISODate('1970-01-01T00:00:00.000+00:00')
  }
}}, {$sort: {
  UltimoEstado: -1,
  MontoCubierto:-1
}}}
```

The screenshot shows the MongoDB Compass interface. On the left, the 'Query' stage is selected, and the query is `{}`. The 'Output after \$sort stage' (Sample of 20 documents) is displayed on the right. The output shows a single document with the following fields: `cvePoliza: "VER002003904000"`, `OBLIGACIONES: "SUMINISTRO NO PEMEX (ESTACION DE SERVICIO)"`, `MontoCubierto: 13000000`, and `UltimoEstado: 2020-04-30T00:00:00.000+00:00`.

11. Empresas con oficinas en REFORMA, SUC.MEXICO CENTRO , 'SUC.MEXICO CENTRO'

MYSQL

Result Grid
Filter Rows:

Export:
Wrap Cell Content:

	total	oficina
▶	342	SUC.MEXICO CENTRO

\$group

+

Output after \$group stage (Sample of 20 documents)

```

1 //**
2 * _id: The id of the group.
3 * fieldN: The first field name.
4 */
5 {
6   _id: {Oficina:'$Oficina',Agente:'$cveAgente'
7     ,Garantia:'$NumGarantia'},
8   Total: {
9     $sum:1
10  }
11 }

```

_id: Object

Oficina: "SUC.MEXICO CENTRO"

Agente: "DFS002"

Garantia: 2

Total: 5

12. Pólizas que pertenecen al ramo administrativo con sus respectivos montos

MYSQL

select cvePoliza, C.RAMO2 as Ramo, MontoAfianzado, MontoCubierto, MontoRetenido from Polizas as P inner join configuracion as C ON C.Configuracion = P.IdConfiguracion where RAMO2 = 'Administrativo';

```

59  -- 12 polizas que pertenecen al ramo administrativo con sus respectivos montos
60  • select cvePoliza, C.RAMO2 as Ramo, MontoAfianzado,
61    MontoCubierto, MontoRetenido from polizas as P
62    inner join configuracion as C ON C.Configuracion = P.IdConfiguracion
63    where RAMO2 = 'Administrativo';

```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Fetch rows:

	cvePoliza	Ramo	MontoAfianzado	MontoCubierto	MontoRetenido
▶	ALE001000020000	Administrativo	10125.87	10125.87	10125.87
	ALE001000021000	Administrativo	193992.31	193992.31	193992.31
	ALE001000022000	Administrativo	1540.89	1540.89	1540.89
	ALE001000023000	Administrativo	8981.21	8981.21	8981.21
	ALE001000024000	Administrativo	154117.51	154117.51	154117.51
	ALE004000023000	Administrativo	2827850.74	2827850.74	2827850.74
	ALE004000024000	Administrativo	894009.66	894009.66	894009.66
	ALE004000025000	Administrativo	1961499.15	1961499.15	1961499.15
	ALE004000026000	Administrativo	894009.66	894009.66	894009.66
	ALE004000027000	Administrativo	59394.71	59394.71	59394.71
	ALE004000028000	Administrativo	154545.84	154545.84	154545.84
	ALE004000029000	Administrativo	59394.71	59394.71	59394.71

MongoDB

```

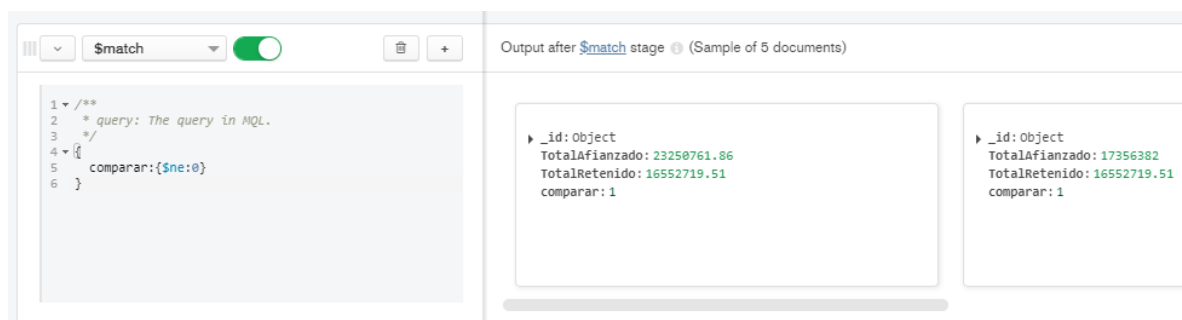
[{$match: {
  RAMO2:{$eq:"Administrativo"},
  FechaEmision: {
    $gte: ISODate('2020-01-01T00:00:00.000+00:00')}
}}, {$group: {

```

```

    _id: {Oficina:'$Oficina',Agente:'$cveAgente'
    ,Garantia:'$NumGarantia',
    FechaEmision:'$FechaEmision'
    },
    TotalAfianzado: {
        $sum:"$MontoAfianzado"
    },
    TotalRetenido: {
        $sum:"$MontoRetenido"
    }
  }}, {$project: {
    comparar:
    { $cmp:["$TotalAfianzado","$TotalRetenido"]}
    ,TotalAfianzado:1
    ,TotalRetenido:1
  }}, {$match: {
    comparar:{$ne:0}
  }}

```



13. Obtener los montos ordenados por ultimoEstado de fecha más reciente a más antigua de cada empresa por oficina

MYSQL

```

select Year(UltimoEstado) as Año, monthname(UltimoEstado) as Mes, dayname(UltimoEstado) as Dia, E.Empresa, O.Oficina, MontoAfianzado from polizas as P inner join empresas as E on P.IdEmpresa = E.IdEmpresa inner join agente as A on P.IdAgente = A.IdAgente inner join oficinas as O on A.IdOficina = O.IdOficina order by UltimoEstado desc;

```

```

65 -- 13 obtener los montos ordenados por ultimoEstado de fecha mas reciente a mas antigua
66 -- de cada empresa por oficina
67 • select Year(UltimoEstado) as Año, monthname(UltimoEstado) as Mes, dayname(UltimoEstado) as Dia,
68     E.Empresa, O.Oficina, MontoAfianzado from polizas as P
69     inner join empresas as E on P.IdEmpresa = E.IdEmpresa
70     inner join agente as A on P.IdAgente = A.IdAgente
71     inner join oficinas as O on A.IdOficina = O.IdOficina order by UltimoEstado desc;

```

	Año	Mes	Dia	Empresa	Oficina	MontoAfianzado
▶	2020	April	Thursday	Empresa824	VERACRUZ	7200000
	2020	April	Thursday	Empresa864	VERACRUZ	13000000
	2020	April	Thursday	Empresa315	VERACRUZ	6000000
	2020	April	Thursday	Empresa315	VERACRUZ	12000000
	2020	April	Thursday	Empresa824	VERACRUZ	127219.79
	2020	April	Thursday	Empresa824	VERACRUZ	7200000
	2020	April	Thursday	Empresa864	VERACRUZ	13000000

MongoDB

```
{{ $project: {
```

```
  _id:0,
```

```
  Oficina:1,
```

```
  cveFiado:1,
```

```
  MontoAfianzado:1,
```

```
  year: { $year: "$UltimoEstado" }
```

```
  ,month: { $month: "$UltimoEstado" }
```

```
}}, { $match: {
```

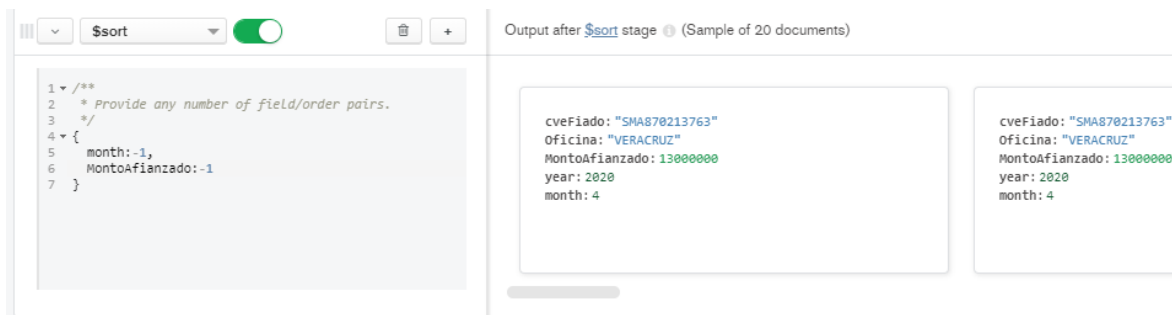
```
  year: { $eq: 2020 }
```

```
}}, { $sort: {
```

```
  month: -1,
```

```
  MontoAfianzado: -1
```

```
}}}
```



14. Obtener MontoAfianzado total, MontoCubierto total y PorcentajeRetenido agrupado por Oficina por el SubRamo 'DE Obra'

MYSQL

select O.Oficina, round(sum(MontoAfianzado),2) as MontoTotalAfianzado,
round(sum(MontoRetenido),2) as MontoTotalRetenido,
concat(round((sum(MontoRetenido)/sum(MontoAfianzado) * 100),2),'%') as PerMontoRetenido,
C.SUBRAMO3 as SubRamo from polizas as P inner join agente as A on P.IdAgente = A.IdAgente
inner join oficinas as O on O.IdOficina = O.IdOficina inner join configuracion as C on
C.Configuracion = P.IdConfiguracion where SUBRAMO3 = 'De Obra' group by O.Oficina;

```
73 -- 14 obtener MontoAfianzado total, MontoCubierto total y PorcentajeRetenido
74 -- agrupado por Oficina por el SubRamo 'DE Obra'
75 • select O.Oficina, round(sum(MontoAfianzado),2) as MontoTotalAfianzado, round(sum(MontoRetenido),2) as MontoTotalRetenido,
76 concat(round(( sum(MontoRetenido)/sum(MontoAfianzado) * 100 ),2),'%') as PerMontoRetenido, C.SUBRAMO3 as SubRamo
77 from polizas as P
78 inner join agente as A on P.IdAgente = A.IdAgente
79 inner join oficinas as O on A.IdOficina = O.IdOficina
80 inner join configuracion as C on C.Configuracion = P.IdConfiguracion
81 where SUBRAMO3 = 'De Obra' group by O.Oficina;
```

Oficina	MontoTotalAfianzado	MontoTotalRetenido	PerMontoRetenido	SubRamo
MONTERREY	176253988.55	168923065.73	95.84%	De Obra
SUC.MEXICO CENTRO	52689833.63	52689833.63	100.00%	De Obra
CULIACAN	30180704.88	19944619.88	66.08%	De Obra
TIJUANA	11054686.77	11054686.77	100.00%	De Obra
QUERETARO	577529276.68	456347509.76	79.02%	De Obra
REFORMA	231393578.03	199507395.06	86.22%	De Obra
VERACRUZ	90277421.92	84372395.53	93.46%	De Obra
HERMOSILLO	3806806.87	3806806.87	100.00%	De Obra
GUADALAJARA	19641151.96	19641151.96	100.00%	De Obra

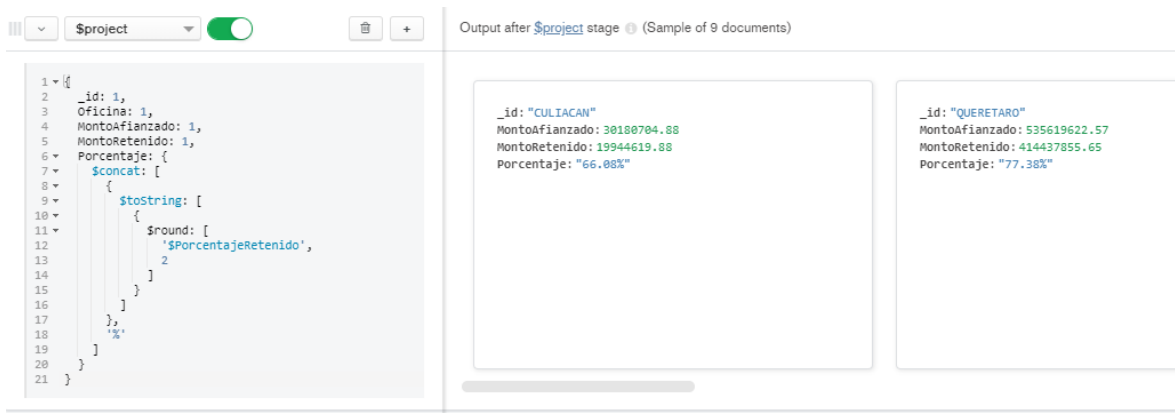
MongoDB

```
{{$match: {
  SUBRAMO3:"De Obra"
}}, {$group: {
  _id: "$Oficina",
  MontoAfianzado: { $sum: "$MontoAfianzado"},
  MontoRetenido: { $sum: "$MontoRetenido"
  }
}}
```

```

    }}, {$addField: {Subramo:"De Obra", PorcentajeRetenido: {$multiply: [ { $divide: [
'$MontoRetenido', '$MontoAfianzado' ]}, 100 ]
    }
}}, {$sort: {
    PorcentajeRetenido: 1
}}, {$project: {
    _id: 1,
    Oficina: 1,
    MontoAfianzado: 1,
    MontoRetenido: 1,
    Porcentaje: { $concat: [ { $toString: [ { $round: ['$PorcentajeRetenido',2] } ]}, '%' ] }
}}

```



15. Obtener MontoAfianzado total, MontoCubierto total y PorcentajeRetenido en un top 10 de menor a mayor por empresa pertenecientes al RAMO 'Crédito'

MYSQL

```

select E.Empresa, round(sum(MontoAfianzado),2) as MontoTotalAfianzado,
round(sum(MontoRetenido),2) as MontoTotalRetenido,
concat(round((sum(MontoRetenido)/sum(MontoAfianzado) * 100 ),2),'%') as PerMontoRetenido
from polizas as P inner join empresas as E on E.IdEmpresa = P.IdEmpresa inner join configuracion
as C on C.Configuracion = P.IdConfiguracion where C.RAMO2 = 'Crédito' group by E.Empresa

```


Order by (sum(MontoRetenido)/sum(MontoAfianzado) * 100) asc limit 10;

```

83 -- 15 obtener MontoAfianzado total, MontoCubierto total y PorcentajeRetenido
84 -- en un top 10 de menor a mayor por empresa pertenecientes al RAMO 'Crédito'
85 • select E.Empresa, round(sum(MontoAfianzado),2) as MontoTotalAfianzado, round(sum(MontoRetenido),2) as MontoTotalRetenido,
86 concat(round(( sum(MontoRetenido)/sum(MontoAfianzado) * 100 ),2),'%') as PerMontoRetenido
87 from polizas as P
88 inner join empresas as E on E.IdEmpresa = P.IdEmpresa
89 inner join configuracion as C on C.Configuracion = P.IdConfiguracion
90 where C.RAMO2 = 'Crédito' group by E.Empresa
91 Order by ( sum(MontoRetenido)/sum(MontoAfianzado) * 100 ) asc limit 10;

```

Empresa	MontoTotalAfianzado	MontoTotalRetenido	PerMontoRetenido
Empresa202	171000000.00	1655271.95	0.97%
Empresa247	37500000.00	3387458.65	9.03%
Empresa224	16161616.16	1636884.53	10.13%
Empresa864	26000000.00	3349001.26	12.88%
Empresa900	13125000.00	1693729.33	12.90%
Empresa762	12424242.42	1693729.33	13.63%
Empresa76	12000000.00	1693729.32	14.11%
Empresa219	11111111.11	1636884.54	14.73%
Empresa315	30000000.00	5081187.94	16.94%
Empresa882	25000000.00	4965815.83	19.86%

MongoDB

[{\$match: {

Ramo:4

}}, {\$group: {

_id: "\$cveFiado",

MontoAfianzado: {

\$sum: "\$MontoAfianzado"

},

MontoRetenido: {

\$sum: "\$MontoRetenido"

}

}}, {\$addFields: {

Subramo:"De Obra",

PorcentajeRetenido: {

\$multiply: [

{

\$divide: [

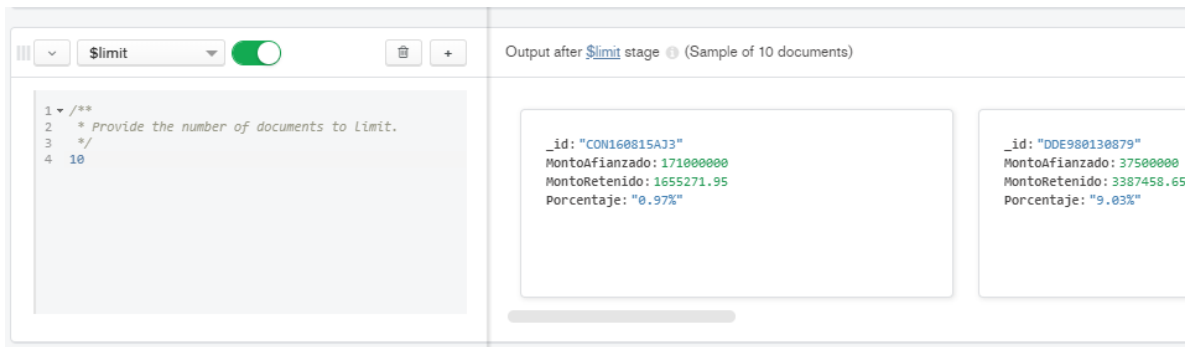
'\$MontoRetenido',

'\$MontoAfianzado'

```

    ]
  },
  100
]
}
}}, {$sort: {
  PorcentajeRetenido: 1
}}, {$project: {
  _id: 1,
  Oficina: 1,
  MontoAfianzado: 1,
  MontoRetenido: 1,
  Porcentaje: {
    $concat: [
      {
        $toString: [
          {
            $round: [
              '$PorcentajeRetenido',
              2
            ]
          }
        ],
      },
      '%'
    ]
  }
}}, {$limit: 10}]

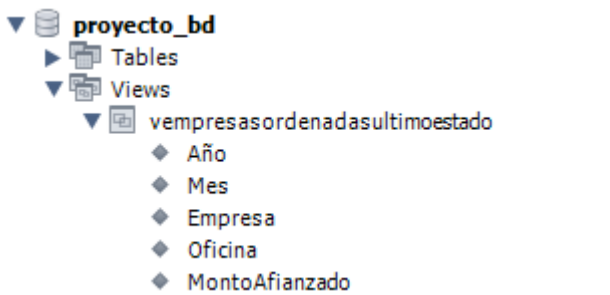
```



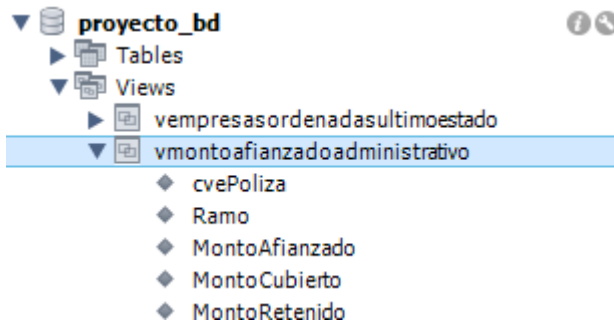
Vistas

MYSQL

```
create view vTotalOficinasMexicoCentro as
select count(*) total, o.oficina from polizas P
inner join agente A on A.IdAgente = P.IdAgente
inner join oficinas O on O.Idoficina = A.Idoficina where O.IdOficina = 3;
```



```
create view vMontoAfianzadoAdministrativo as
select cvePoliza, C.RAMO2 as Ramo, MontoAfianzado,
MontoCubierto, MontoRetenido from polizas as P
inner join configuracion as C ON C.Configuracion = P.IdConfiguracion
where RAMO2 = 'Administrativo';
```



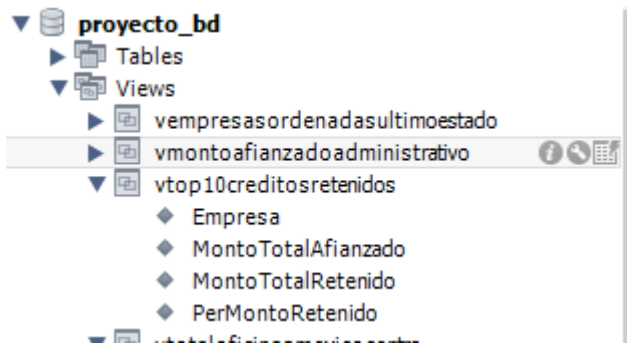
create view vEmpresasOrdenadasUltimoEstado AS

select Year(UltimoEstado) as Año, monthname(UltimoEstado) as Mes, E.Empresa, O.Oficina,
MontoAfianzado from polizas as P

inner join empresas as E on P.IdEmpresa = E.IdEmpresa

inner join agente as A on P.IdAgente = A.IdAgente

inner join oficinas as O on A.IdOficina = O.IdOficina order by UltimoEstado desc;



create view vTotaObraPorOficina as

select O.Oficina, round(sum(MontoAfianzado),2) as MontoTotalAfianzado,
round(sum(MontoRetenido),2) as MontoTotalRetenido,

concat(round((sum(MontoRetenido)/sum(MontoAfianzado) * 100),2),'%') as PerMontoRetenido

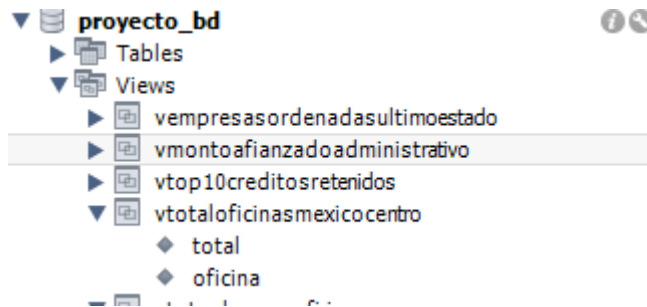
from polizas as P

inner join agente as A on P.IdAgente = A.IdAgente

inner join oficinas as O on A.IdOficina = O.IdOficina

inner join configuracion as C on C.Configuracion = P.IdConfiguracion

where SUBRAMO3 = 'De Obra' group by O.Oficina;



CREATE VIEW vTop10CreditosRetenidos as

select E.Empresa, round(sum(MontoAfianzado),2) as MontoTotalAfianzado,
round(sum(MontoRetenido),2) as MontoTotalRetenido,

concat(round((sum(MontoRetenido)/sum(MontoAfianzado) * 100),2),'%') as PerMontoRetenido

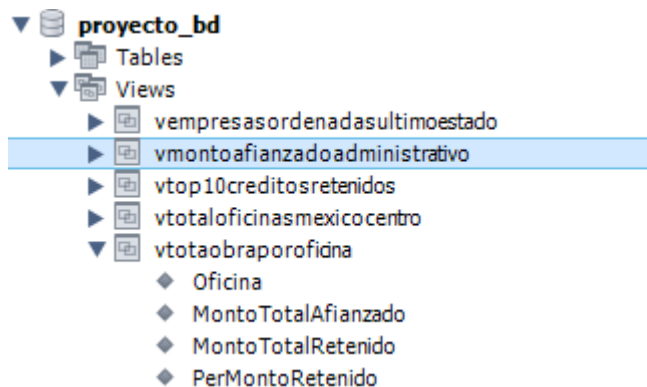
from polizas as P

inner join empresas as E on E.IdEmpresa = P.IdEmpresa

inner join configuracion as C on C.Configuracion = P.IdConfiguracion

where C.RAMO2 = 'Crédito' group by E.Empresa

Order by (sum(MontoRetenido)/sum(MontoAfianzado) * 100) asc limit 10;



MongoDB

Garantias.AgentesConDiferenciasAyR2020 (view on: Garantias.Proyecto)

[MODIFY SOURCE](#)[Documents](#)[Aggregations](#)[Schema](#)[Explain Plan](#)[Indexes](#)[Validation](#)[FILTER](#)[VIEW](#)[Displa](#)

▼ **_id:** Object
Oficina: "GUADALAJARA"
Agente: "GDL001"
Garantia: 1
FechaEmision: 2020-04-22T00:00:00.000+00:00
TotalAfianzado: 13007260.17
TotalRetenido: 3000000
comparar: 1

▼ **_id:** Object
Oficina: "MONTERREY"
Agente: "MTY001"
Garantia: 0
FechaEmision: 2020-01-13T00:00:00.000+00:00
TotalAfianzado: 17356382
TotalRetenido: 16552719.51
comparar: 1

▶ **_id:** Object
TotalAfianzado: 45107135.01
TotalRetenido: 33105439.02
comparar: 1

▼ **_id:** Object
Oficina: "TIJUANA"
Agente: "TIJ054"
Garantia: 3
FechaEmision: 2020-03-04T00:00:00.000+00:00
TotalAfianzado: 23250761.86
TotalRetenido: 16552719.51
comparar: 1

▼ **_id:** Object
Oficina: "VERACRUZ"
Agente: "VER053"
Garantia: 1
FechaEmision: 2020-01-29T00:00:00.000+00:00
TotalAfianzado: 22457745.9
TotalRetenido: 16552719.51
comparar: 1

Documents	Aggregations	Schema	Explain Plan	Indexes	Validation
<div>FILTER</div>					
<div>VIEW</div>					
<div> <div>_id: "CULIACAN"</div> <div>MontoAfianzado: 30180704.88</div> <div>MontoRetenido: 19944619.88</div> <div>Porcentaje: "66.08%"</div> </div>					
<div> <div>_id: "GUADALAJARA"</div> <div>MontoAfianzado: 39708800.05</div> <div>MontoRetenido: 39708800.05</div> <div>Porcentaje: "100.00%"</div> </div>					
<div> <div>_id: "HERMOSILLO"</div> <div>MontoAfianzado: 3806806.87</div> <div>MontoRetenido: 3806806.87</div> <div>Porcentaje: "100.00%"</div> </div>					
<div> <div>_id: "MONTERREY"</div> <div>MontoAfianzado: 176344709.14</div> <div>MontoRetenido: 169013786.32</div> <div>Porcentaje: "95.84%"</div> </div>					
<div> <div>_id: "QUERETARO"</div> <div>MontoAfianzado: 535619622.57</div> <div>MontoRetenido: 414437855.65</div> <div>Porcentaje: "77.38%"</div> </div>					
<div> <div>_id: "REFORMA"</div> <div>MontoAfianzado: 253144855.46</div> <div>MontoRetenido: 221258672.49</div> <div>Porcentaje: "87.40%"</div> </div>					
<div> <div>_id: "SUC.MEXICO CENTRO"</div> <div>MontoAfianzado: 52689833.63</div> <div>MontoRetenido: 52689833.63</div> <div>Porcentaje: "100.00%"</div> </div>					

Conclusiones

Al tener las bases con las cuales se hacen las consultas y el resultado que se desea obtener facilitan el generar consultas que generen valor.

El utilizar las bases de datos relacionales o no relacionales, pueden ayudar al análisis de la información.

Es necesario analizar la información, el nivel de datos a tratar y cual es el resultado que se desea para saber que tipo de base de datos es más eficiente. En el caso particular de los datos de este proyecto, fue un poco mas sencillo hacer las consultas en MongoDB, dado que la fuente era solo un archivo.