

## PREWORK SESIÓN 6

### Introducción

En la sesión anterior construimos las primeras agregaciones y obtuvimos resultados a partir de la generación de pipelines de filtros, proyecciones, ordenamientos, entre otras. En esta sesión profundizaremos en el uso de agregaciones, primero repasando el concepto de agrupamiento y después usando búsquedas (\$lookup) para relacionar colecciones.

Finalmente, aprenderás a generar vistas que almacenen los resultados de un pipeline para que puedas consultarlo como si se tratara de cualquier otro tipo de colección.

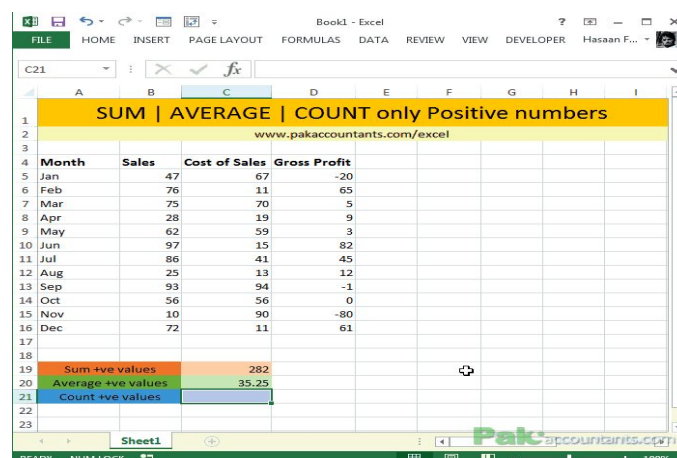
### Objetivos

1. Repasar el concepto de agrupamiento de la sesión anterior.
2. Realizar consultas a dos o más colecciones a la vez mediante agregaciones.
3. Aprender a generar vistas para preservar los resultados de un pipeline.

### Agrupamientos

Como recordarás de las sesiones de MySQL y la última de MongoDB, un agrupamiento nos permite aplicar una función a una columna en el caso de las tablas o al campo de una colección en el caso de los documentos.

Esto es similar a cuando aplicamos una función en las columnas de los documentos en Excel.



Month	Sales	Cost of Sales	Gross Profit
Jan	47	67	-20
Feb	76	11	65
Mar	75	70	5
Apr	28	19	9
May	62	59	3
Jun	97	15	82
Jul	86	41	45
Aug	25	13	12
Sep	93	94	-1
Oct	56	56	0
Nov	10	90	-80
Dec	72	11	61
Sum +ve values		282	
Average +ve values		35.25	
Count +ve values			

La única restricción es el uso de funciones de agrupamiento. En MongoDB los agrupamientos tienen que requerir la siguiente sintaxis:

```
{
  _id: <expresión>, // Campo por el cual agrupar
  <Nuevo campo>: {<acumulador>: <expresión1>},
  ...
}
```

Al igual que en SQL, necesitamos especificar el campo por el cual agrupar, y después indicar uno por uno los acumuladores y a qué columna o valor serán aplicados. Por acumulador nos referimos a las funciones de agrupamiento, en MongoDB son llamadas acumuladores, pero es lo mismo.

Por ejemplo, supongamos que queremos el promedio de las calificaciones de los distintos grupos de un profesor. Para ello usamos una agregación \$group con los siguientes datos:

```
{
  _id: grupo,
  promedio: {$avg: calificacion},
  ...
}
```

De esta forma, por cada grupo asignado al profesor, se calculan los promedios generando documentos con la siguiente estructura:

```
{
  grupo: 401
  promedio: 7.2
},
...

```

Otros acumuladores que se pueden usar con esta agrupación son: \$max, \$min, y \$sum por ejemplo. Puedes consultar más acumuladores y su funcionamiento en la documentación de MongoDB.

<https://docs.mongodb.com/manual/reference/operator/aggregation/group/>

## Consultas a múltiples colecciones

En SQL usamos las distintas variantes de JOIN para relacionar tablas. En MongoDB no existe como tal la operación JOIN, sin embargo, podemos relacionar colecciones por sus campos usando un tipo de agregación llamado \$lookup.

Esta operación requiere la siguiente sintaxis:

```
{
  from: <colección derecha>,
  localField: <campo de la colección izquierda>,
  foreignField: <campo de la colección derecha>,
  as: <arreglo de salida con los resultados>
}
```

## \$lookup



En SQL al hacer un JOIN se obtenía una nueva colección. En MongoDB, se genera un arreglo con los documentos que coinciden en la colección derecha y se agregan como nuevo campo a la colección izquierda.

## Vistas

Las vistas en MongoDB funcionan de la misma manera que en SQL, es decir, se genera una colección *virtual* con los resultados de una consulta. En este caso en lugar de consulta se genera una colección con los resultados del pipeline de una agregación.

Para crear vistas, en la vista de agregaciones seleccionaremos el botón [Save] y daremos clic en la opción [Create a view], damos un nombre a la colección y ¡Listo! tenemos nuestra vista creada.

Puedes visualizarla en el listado que se muestra a la izquierda y abrirla como si se tratara de cualquier colección.