

**UNIVERSIDAD DEL VALLE DE GUATEMALA**

**CC3089 - Bases de Datos 2**

**Sección 10**

**Ing. Mario Francisco Barrientos**



## **Laboratorio 3**

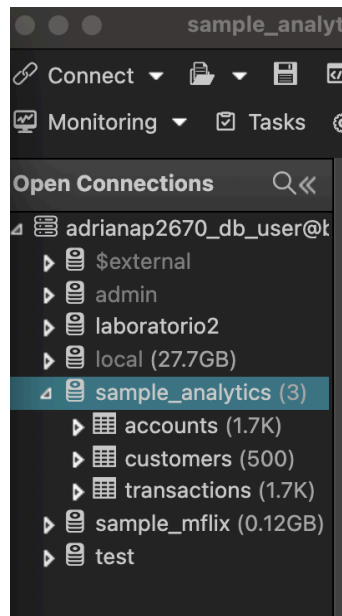
**Adriana Sophia Palacios Contreras 23044**

**GUATEMALA**

**14 de febrero de 2026**

## Ejercicio 1 – Preparación de ambiente

Para este ejercicio deberá usar Atlas, ya que deberá cargar la base de datos de ejemplo de sample\_analytics que MongoDB proporciona por medio de su implementación en la nube. Adicionalmente, realice la exploración de los datos creados en las distintas colecciones. Para cada colección, indique que tipo de información aparece y si comente el tipo de estructura de datos que tiene.



- accounts
  - \_id: es un ObjectId generado por MongoDB para identificar de forma única cada documento.
  - account\_id: Es un número entero que funciona como identificador de la cuenta.
  - limit: Es un número entero que representa un límite de crédito de la cuenta.
  - products: Es un array que representa los productos o servicios de esa cuenta.
- customers
  - id: Es un identificador único generado por MongoDB
  - username: Es el nombre de usuario del cliente en el sistema, tipo string.
  - name: nombre completo del cliente se guarda como tipo String.
  - address: dirección del cliente en formato de texto como String.
  - birthdate: fecha de cumpleaños del cliente como date.
  - email: correo de cliente de tipo String.
  - active: Es un valor booleano, el cual indica si el cliente está activo en el sistema.
  - accounts: Es un array el cual hace referencia a los Id de las cuentas que tienen los clientes.
  - tier\_and\_details: Es un documento embebido que guarda información sobre el nivel o categoría del cliente dentro del sistema, junto con los beneficios que tiene por pertenecer a ese nivel.

- transactions
  - \_id: identificador único generado por MongoDB.
  - account\_id: Es un número entero que identifica la cuenta asociada a esas transacciones.
  - transaction\_count: número total de transacciones que se han realizado en la cuenta.
  - bucket\_start\_date" y "bucket\_end\_date: es el rango de fechas en que se realizaron las transacciones.
  - Transactions: es un array de documentos JSON donde cada elemento representa una transacción financiera realizada por una cuenta. En este arreglo se almacenan datos como la fecha de la operación, la cantidad de unidades compradas o vendidas, el tipo de transacción (por ejemplo, compra o venta), el símbolo del activo financiero, el precio por unidad y el valor total de la operación.

**2.1 Calcule para cada cliente el número total de transacciones y el monto promedio de cada una. Muestre el nombre completo, ciudad y los resultados ordenados por número de transacciones descendente.**

The screenshot shows the NoSQLBooster for MongoDB interface. The query editor contains the following aggregate query:

```

15 use("sample_analytics")
16
17 db.customers.aggregate([
18   {
19     $lookup: {
20       from: "transactions",
21       localField: "accounts",
22       foreignField: "account_id",
23       as: "Totaltransactions"
24     }
25   },
26   {
27     $unwind: "$Totaltransactions"
28   },
29 ])

```

The results are displayed in a table with the following columns: \_id, nombre, ciudad, transacc, and monto. The results are ordered by the number of transactions (transacc) in descending order.

	_id	nombre	ciudad	transacc	monto
1	5ca4bbcea2dd94ee58162c0	John William	USCGC Park	443	172,671.4,11\$
2	5ca4bbcea2dd94ee58162a8	Phillip Obrien	PSC 5639, B	421	737,966.4,75
3	5ca4bbcea2dd94ee58162ac	Travis White	61749 Adam	409	870,523.8,07
4	5ca4bbcea2dd94ee58162c1	Aaron Perez	55375 Malon	404	933,114.0,78
5	5ca4bbcea2dd94ee58162bc	Stacey Mccal	3567 Gonzal	403	606,021.3,54
6	5ca4bbcea2dd94ee58162b0	Angela Leblai	977 Brown J	401	139,047.6,70
7	5ca4bbcea2dd94ee58162bb	Megan Tanne	57817 Julie T	401	133,750.2,21
8	5ca4bbcea2dd94ee58162b9	Ashley Rodrí	94038 Luis C	395	598,133.0,6\$

Para esta consulta, primero utilicé \$lookup para relacionar los clientes con sus transacciones, ya que los clientes tienen un arreglo con los IDs de sus cuentas. Luego usé \$unwind para separar cada transacción individual y así poder contarlas correctamente. Después convertí el campo total a un valor numérico usando \$convert, porque en la base de

datos está guardado como texto y eso puede generar errores al calcular el promedio. Finalmente, agrupé los resultados por cliente para obtener el nombre, la dirección, el número total de transacciones y el monto promedio, y ordené los resultados de forma descendente según la cantidad de transacciones.

**2.2 Clasifique a los clientes en tres categorías según su balance total, debe mostrar el nombre y categoría de estos.**

**a. Bajo (<5,000)**

**b. Medio (5,000–20,000)**

**c. Alto (>20,000)**

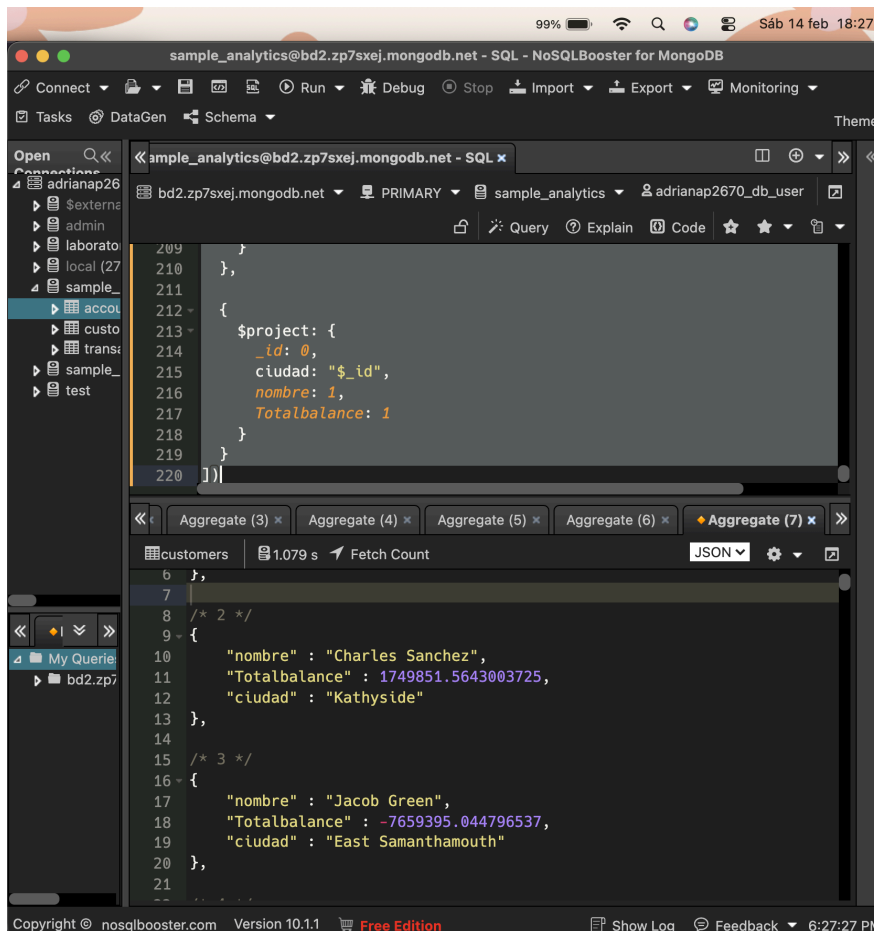
The screenshot shows the NoSQLBooster for MongoDB interface. The query editor displays an aggregate query for the 'customers' collection, using a \$lookup to join with the 'transactions' collection. The results are shown in a table with columns: nombre, Totalbalance, and categoria. The results are ordered by Totalbalance in descending order.

```
58 //2.2
59 use("sample_analytics")
60
61 db.customers.aggregate([
62   {
63     $lookup: {
64       from: "transactions",
65       localField: "accounts",
66       foreignField: "account_id",
67       as: "transacciones"
68     }
69   }
70 ])
```

	nombre	Totalbalance	categoria
1	Danielle Jenkins	3,945,217.624 (3.9M)	Alto
2	Thomas Zimmerman	220,552.0738 (0.22M)	Alto
3	Fred Wang	2,029,076.3889 (2.0M)	Alto
4	Crystal Walker	-14,847,903.1905	Bajo
5	John Williams	-4,845,514.0818	Bajo
6	Travis White	-9,467,625.9624	Bajo
7	Raymond Hudson	-2,216,381.1011	Bajo
8	Erika Davies	-1,924,269.6184	Bajo
9	Gary Nichols	2,903,480.8561 (2.9M)	Alto

Aquí usé transactions para sacar un balance neto: primero uní clientes con sus transacciones y separé cada transacción con \$unwind. Convertí el total a número y luego creé un valor “con signo” (signedTotal), donde si la transacción es sell se suma porque entra dinero y si es buy se resta porque es dinero que salió. Después agrupé por cliente para sumar ese neto y obtener Totalbalance. Al final usé \$switch con los rangos para clasificar en Bajo, Medio o Alto y mostré nombre, Totalbalance y categoría.

**2.3 Para cada ciudad, identifique el cliente con el mayor balance total, sumando todas sus cuentas. Muestre únicamente el nombre, ciudad y total.**



The screenshot shows the NoSQLBooster for MongoDB interface. The left sidebar displays a database structure with collections like 'customers', 'transactions', and 'test'. The main editor shows an aggregate query for the 'customers' collection. The query uses a pipeline with several stages: a \$lookup stage to join transactions, a \$unwind stage to flatten the transactions array, a \$project stage to select fields, a \$convert stage to convert the total balance to a number, a \$signedTotal stage to calculate the net balance, a \$group stage to aggregate by city, and a \$sort stage to sort by total balance. The results pane shows the output of the query, displaying the name, city, and total balance for the top two customers in each city.

```
209 }
210 },
211 {
212   $project: {
213     _id: 0,
214     ciudad: "$_id",
215     nombre: 1,
216     Totalbalance: 1
217   }
218 }
219 }
220 }
```

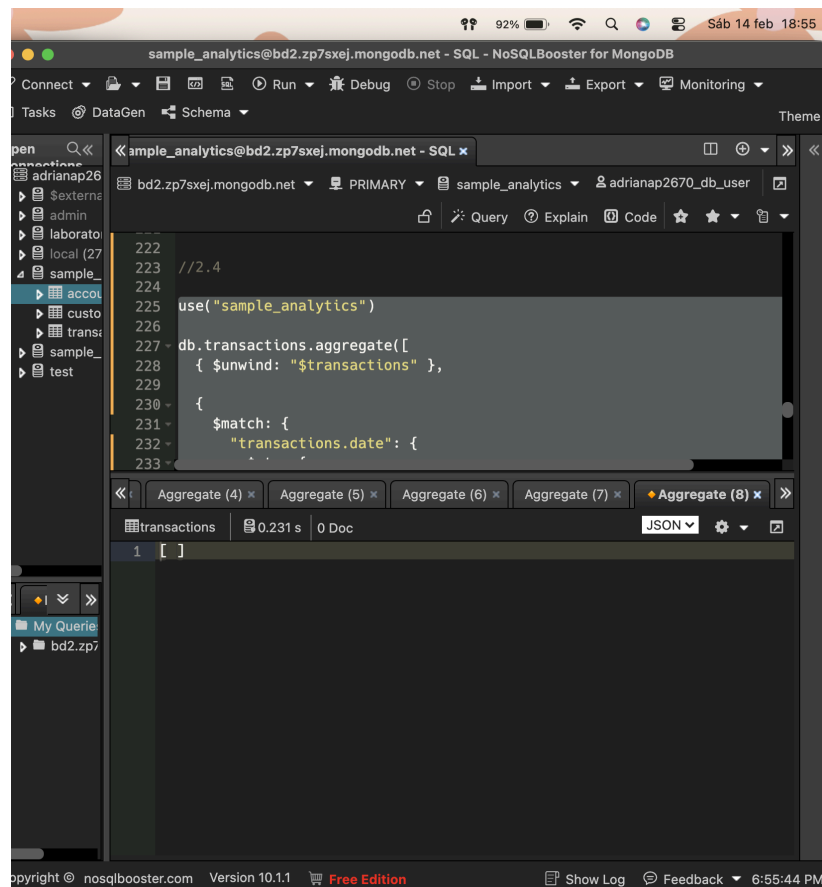
customers | 1.079 s | Fetch Count | JSON

```
6 {
7   /* 2 */
8   {
9     "nombre" : "Charles Sanchez",
10    "Totalbalance" : 1749851.5643003725,
11    "ciudad" : "Kathyside"
12  },
13 },
14 /* 3 */
15 {
16   "nombre" : "Jacob Green",
17   "Totalbalance" : -7659395.044796537,
18   "ciudad" : "East Samanthamouth"
19 },
20 },
21 }
```

Copyright © nosqlbooster.com Version 10.1.1 Free Edition Show Log Feedback 6:27:27 PM

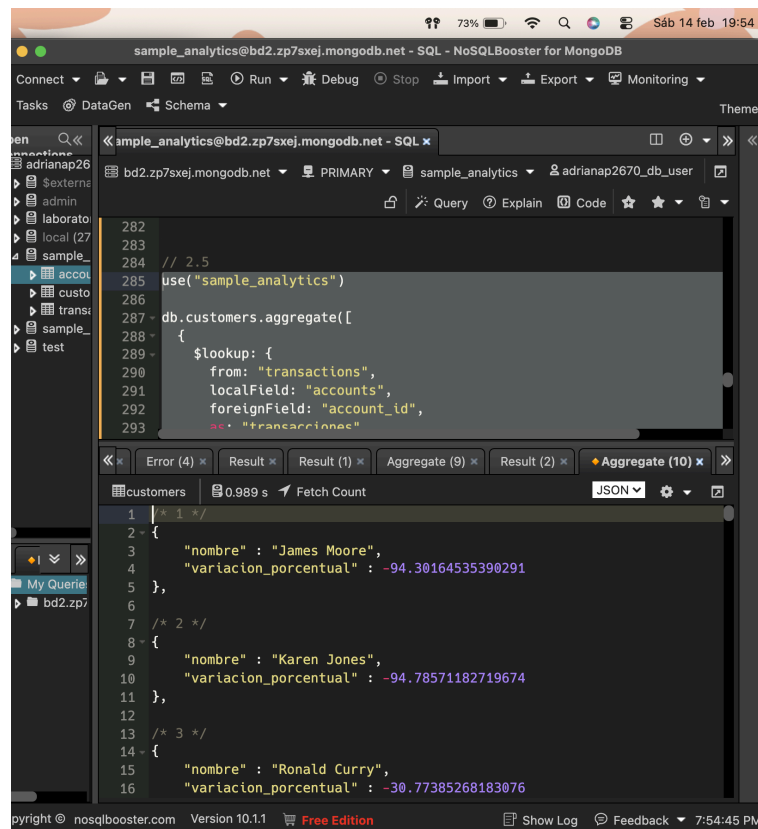
En esta consulta primero relacioné la colección customers con transactions usando \$lookup, porque cada cliente tiene un arreglo con los IDs de sus cuentas y así puedo traer todas las transacciones de esas cuentas. Después utilicé \$unwind dos veces para separar primero cada documento de transacciones y luego cada transacción individual, ya que vienen dentro de un arreglo. Luego saqué la ciudad desde el campo address (tomando la segunda línea y el texto antes de la coma) para poder agrupar por ciudad. También convertí el campo total a número con \$convert porque viene como texto y necesito hacer sumas. Después creé signedTotal, donde si la transacción es sell se suma y si es buy se resta, para obtener un balance neto. Luego agrupé por cliente para calcular su Totalbalance. Finalmente ordené por ciudad y por Totalbalance de mayor a menor, y volví a agrupar por ciudad para quedarme con el primer cliente, que ya es el que tiene el mayor balance en esa ciudad. Al final muestro únicamente el nombre, la ciudad y el Totalbalance.

## 2.4 Extraiga las 10 transacciones más altas realizadas en los últimos 6 meses y enriquezca el resultado con información del cliente asociado.



Para este ejercicio, primero utilicé \$unwind para separar cada transacción individual, ya que están dentro de un arreglo. Luego filtré las transacciones de los últimos 6 meses usando \$match con la fecha actual. Después convertí el campo total a número porque está guardado como texto y así poder ordenarlo correctamente. Luego ordené las transacciones de mayor a menor y limité el resultado a las 10 más altas. Finalmente, utilicé \$lookup para relacionar cada transacción con el cliente correspondiente según la cuenta y mostrar la información enriquecida.

**2.5 Para cada cliente, calcule la variación porcentual entre su transacción más reciente y la más antigua. Solo incluya clientes con al menos dos transacciones.**



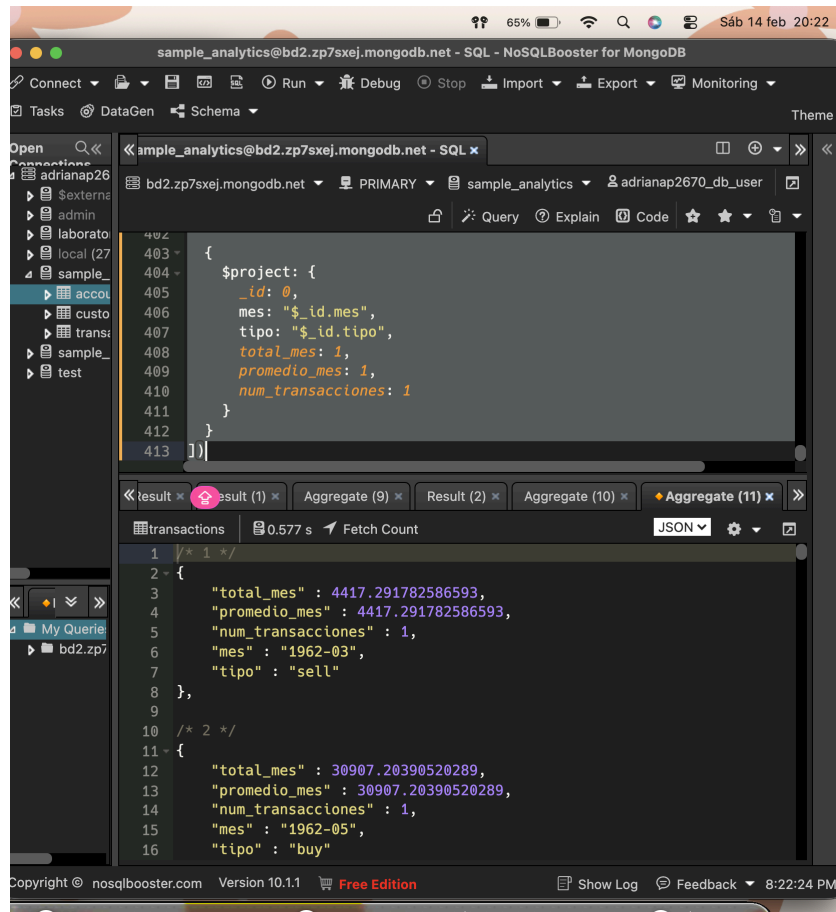
```
282
283
284 // 2.5
285 use("sample_analytics")
286
287 db.customers.aggregate([
288   {
289     $lookup: {
290       from: "transactions",
291       localField: "accounts",
292       foreignField: "account_id",
293       as: "transactions"
294     }
295   },
296   {
297     $unwind: "$transactions"
298   },
299   {
300     $group: {
301       _id: "$_id",
302       total_reciente: { $sum: "$transactions.total" },
303       total_antigua: { $sum: "$transactions.total" },
304       variacion_porcentual: { $divide: [ "$total_reciente", "$total_antigua" ] }
305     }
306   },
307   {
308     $filter: {
309       $gt: [ "$variacion_porcentual", 0 ]
310     }
311   }
312 ])
```

customers 0.989 s Fetch Count JSON

```
1 /* 1 */
2 {
3   "nombre": "James Moore",
4   "variacion_porcentual": -94.30164535390291
5 },
6
7 /* 2 */
8 {
9   "nombre": "Karen Jones",
10  "variacion_porcentual": -94.78571182719674
11 },
12
13 /* 3 */
14 {
15   "nombre": "Ronald Curry",
16   "variacion_porcentual": -30.77385268183076
17 }
```

Yo uní customers con transactions para obtener todas las transacciones de las cuentas del cliente. Separé cada transacción con \$unwind, saqué la fecha y convertí el total a número. En lugar de ordenar todo con \$sort, dentro del \$group usé \$bottom para obtener el total de la transacción más antigua (fecha menor) y \$top para obtener el total de la transacción más reciente (fecha mayor). Luego filtré solo clientes con al menos dos transacciones y calculé la variación porcentual entre el total reciente y el antiguo.

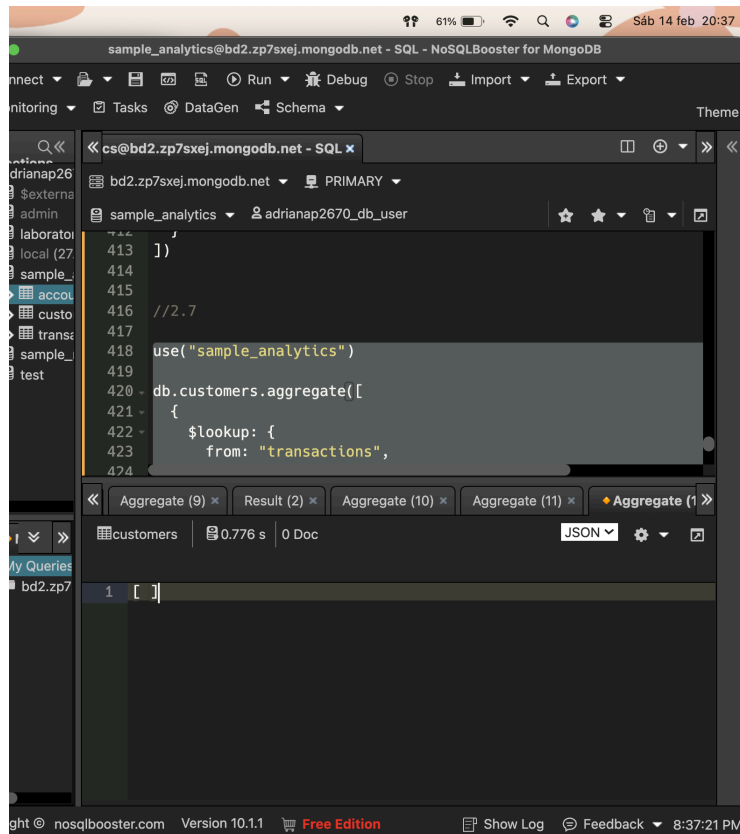
**2.6 Agrupe las transacciones por mes y tipo de transacción. Calcule el total y el promedio por grupo.**



Primero separé cada transacción con \$unwind porque vienen dentro de un arreglo. Luego creé el campo mes usando \$dateToString para quedarme con año-mes y también tomé el tipo de transacción. Convertí el total a número porque estaba como texto. Después agrupé por mes y tipo para calcular el total sumado y el promedio de las transacciones en cada grupo, y al final ordené el resultado para que se vea por mes y por tipo.

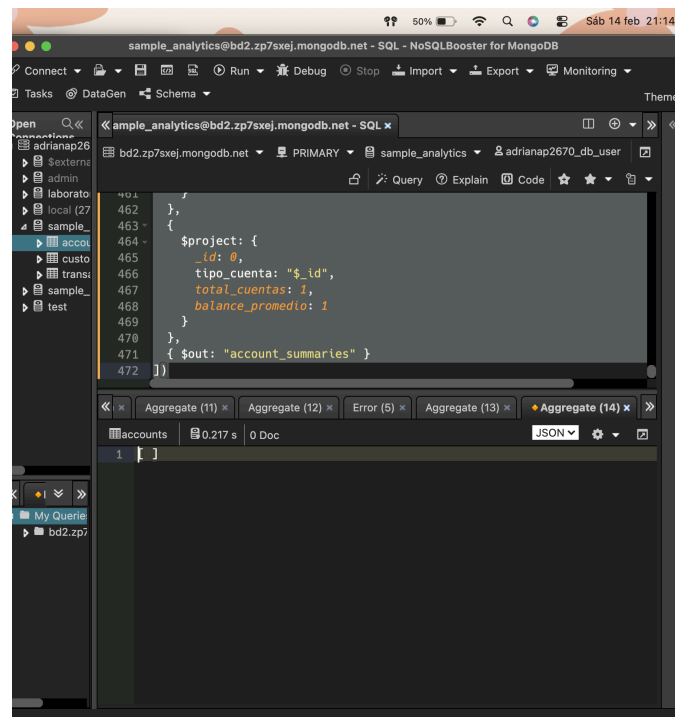
**2.7 Identifique clientes que no han realizado ninguna transacción y guárdelos en una nueva colección llamada inactive\_customers.**





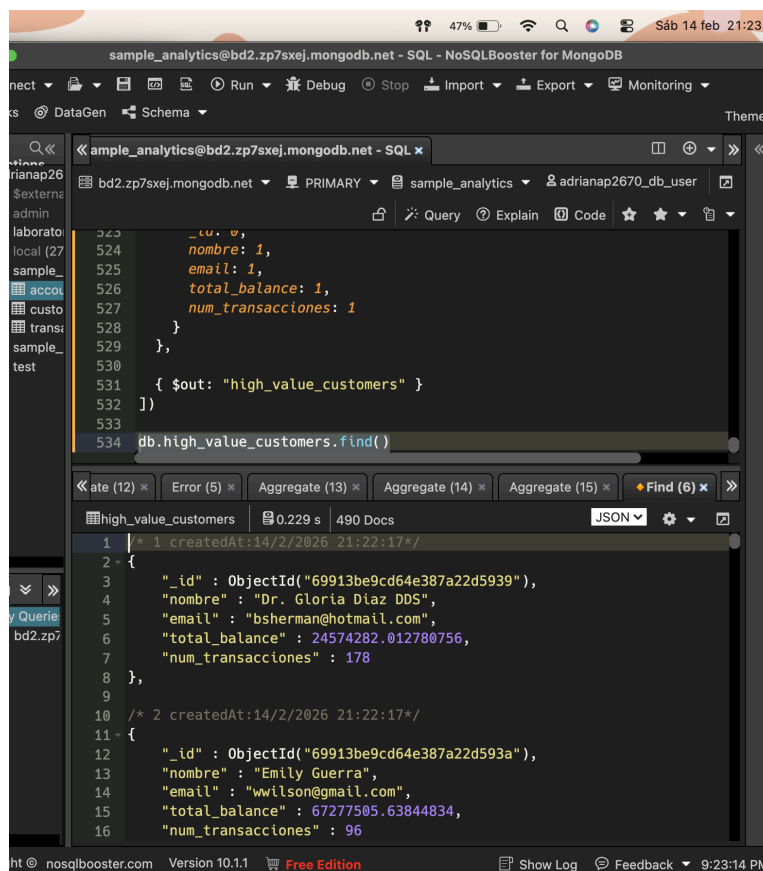
Primero relacioné la colección customers con transactions usando \$lookup para saber qué clientes tienen transacciones asociadas a sus cuentas. Luego filtré solo los clientes que no tienen ninguna transacción, verificando que el arreglo esté vacío. Después seleccioné los campos importantes y finalmente utilicé \$out para guardar el resultado en una nueva colección llamada inactive\_customers.

**2.8 Para cada tipo de cuenta, actualice un documento resumen en una colección account\_summaries que contenga el total de cuentas y el balance promedio por tipo.**



Primero usé \$unwind en products porque ahí vienen los tipos de cuenta en un arreglo, entonces necesito separarlos para contarlos por tipo. Luego con \$group agrupé por cada tipo (\$products) para calcular cuántas cuentas hay (total\_cuentas) y el promedio del balance usando el campo limit como referencia (balance\_promedio). Después usé \$project para darle nombres más claros a los campos y quitar el \_id. Al final utilicé \$out para guardar el resultado en una nueva colección llamada account\_summaries.

**2.9 Cree una colección high\_value\_customers que contenga los clientes cuyo balance total sea mayor a 30,000 y que hayan realizado más de 5 transacciones. Incluya su nombre, email, total\_balance y número de transacciones.**



The screenshot shows the NoSQLBooster for MongoDB interface. The aggregation pipeline is defined in the main editor:

```
1 { $unwind: '$products' }
2 { $group: { _id: '$products', total_balance: { $sum: '$products.balance' }, num_transacciones: { $count: '*' } } }
3 { $project: { _id: 0, nombre: '$_id.nombre', email: '$_id.email', total_balance: '$total_balance', num_transacciones: '$num_transacciones' } }
4 { $out: 'high_value_customers' }
5 }
```

The results pane shows the output of the aggregation, displaying two documents in JSON format:

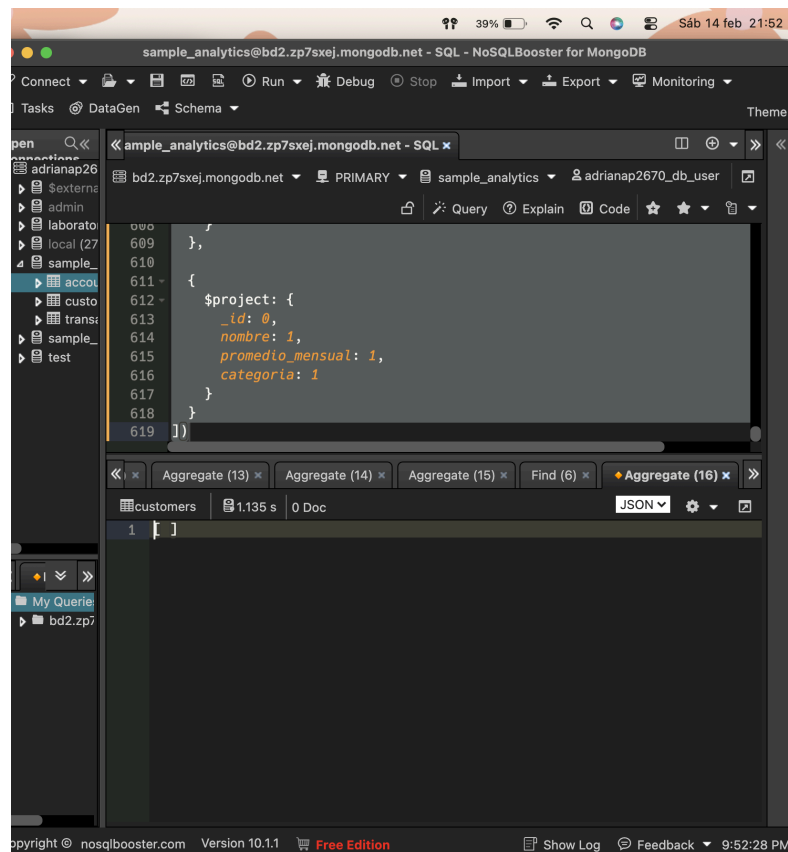
```
1 { "_id": "69913be9cd64e387a22d5939",
2   "nombre": "Dr. Gloria Diaz DDS",
3   "email": "bsherman@hotmail.com",
4   "total_balance": 24574282.012780756,
5   "num_transacciones": 178 }
6
7
8
9
10 /* 2 createdAt:14/2/2026 21:22:17*/
11 { "_id": "69913be9cd64e387a22d593a",
12   "nombre": "Emily Guerra",
13   "email": "wwilson@gmail.com",
14   "total_balance": 67277505.63844834,
15   "num_transacciones": 96 }
```

Primero relacioné customers con transactions usando \$lookup para obtener todas las transacciones de las cuentas de cada cliente. Luego separé cada transacción con \$unwind y convertí el campo total a número porque estaba como texto. Después agrupé por cliente para calcular el balance total sumando todas sus transacciones y conté cuántas transacciones ha realizado cada uno. Luego filtré solo los clientes cuyo balance total fuera mayor a 30,000 y que tuvieran más de 5 transacciones. Finalmente proyecté los campos solicitados y utilicé \$out para guardar el resultado en una nueva colección llamada high\_value\_customers.

2.10 Para cada cliente, calcule el promedio mensual de transacciones en el último año y clasifíquelo según el siguiente criterio:

- 'infrequent' si tiene menos de 2 transacciones por mes en promedio
- 'regular' si tiene entre 2 y 5
- 'frequent' si tiene más de 5.

Muestra el nombre completo, el promedio mensual redondeado a un decimal, y la categoría.



En este ejercicio primero uní la colección de clientes con la de transacciones usando un lookup, para poder ver todas las operaciones que ha hecho cada cliente. Luego separé las transacciones para analizarlas una por una y filtré solo las del último año. Después conté cuántas transacciones hizo cada cliente en ese período y calculé el promedio mensual dividiendo ese total entre 12 meses. Ese resultado lo redondeé a un decimal para que fuera más claro. Finalmente, clasifiqué a cada cliente según su actividad: si tiene menos de 2 transacciones por mes es infrequent, entre 2 y 5 es regular y si tiene más de 5 es frequent. Así se puede identificar fácilmente qué tan activo es cada cliente en el último año.