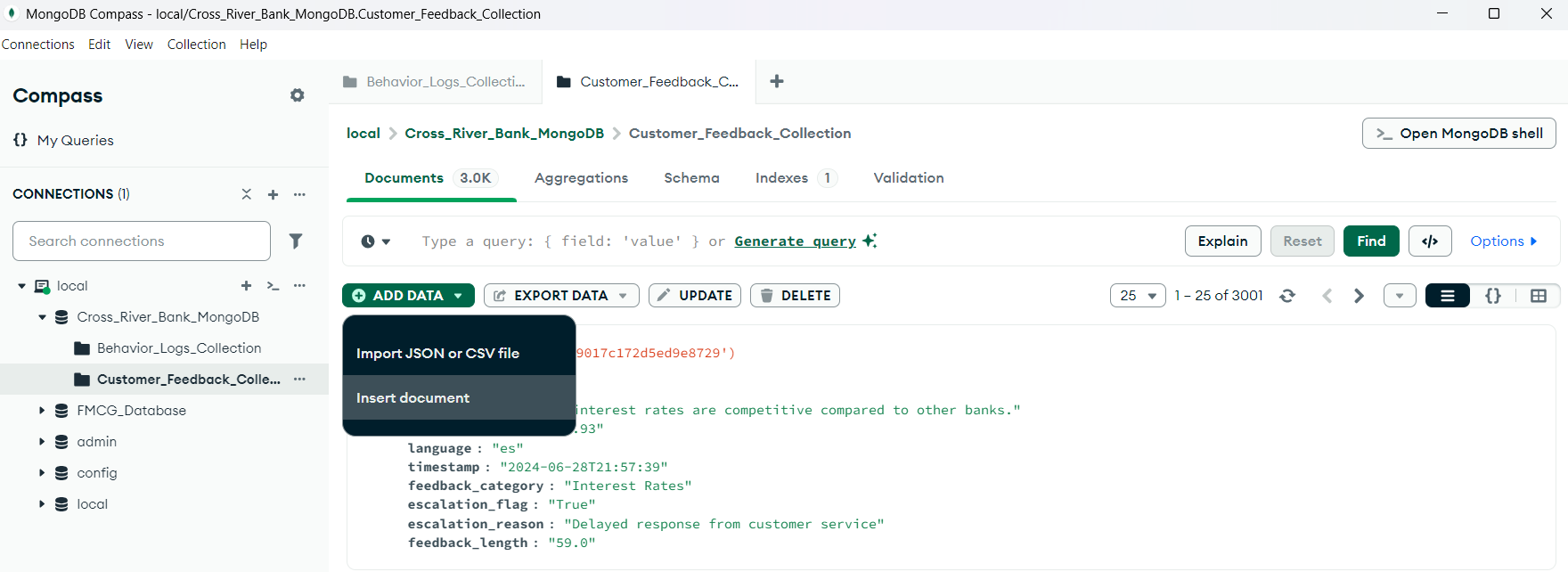
**CRUD Operations:**

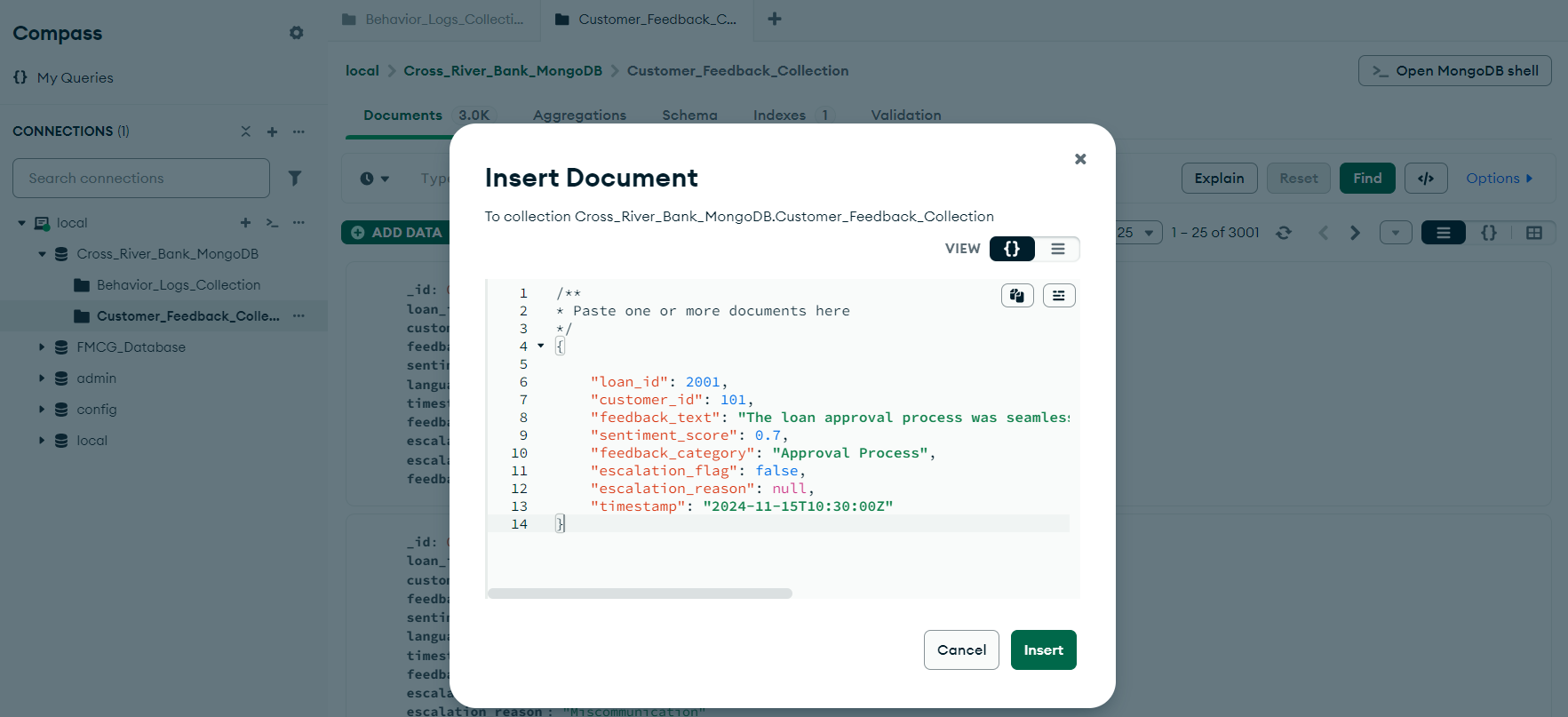
**Task 1 Insert New Feedback:**

Input: Adding feedback for a customer and loan, capturing sentiments and details about the loan process.

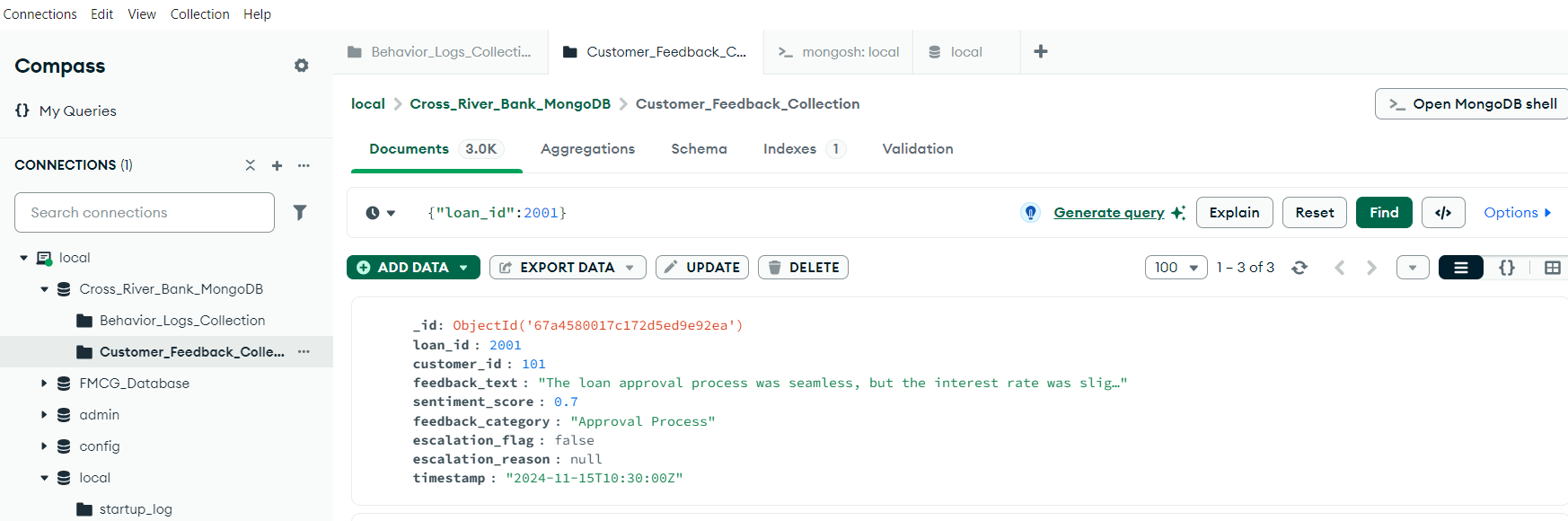
**Step 1:** In order to insert a collection, we need to go to **ADD** **DATA** tab under it we have option of **Insert** **document** click on it.



**Step 2:** It will open a blank data where we have input the documents as what we want to insert.



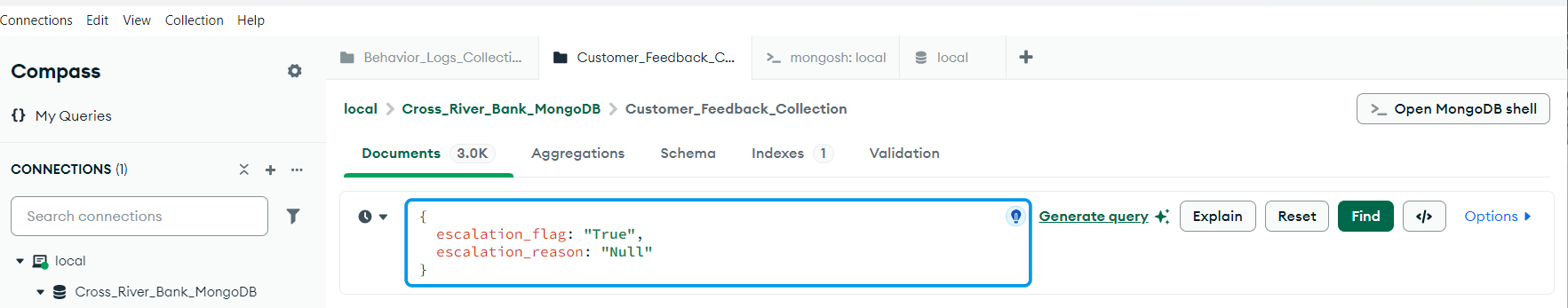
**Output**:



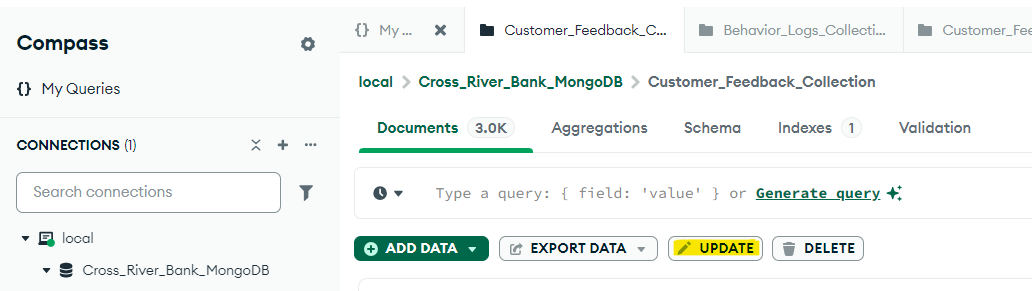
**Task 2 Update Escalation Flags: No Completed**

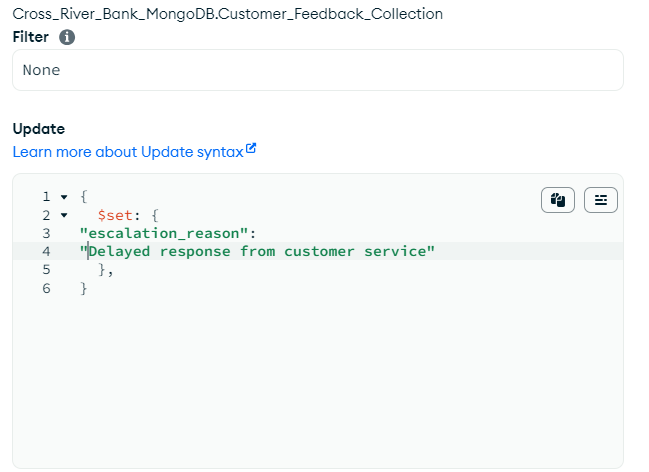
Update escalation flags in feedback to include specific reasons for unresolved customer complaints.

**Step 1:**  In order to update the escalation flags, firstly we need to identify the customers where escalation flag is True and escalation reason is null.

****

**Step2:** Now, we can update the documents as if found to update the reason for unresolved customer complaints.





**Task 3 Remove Duplicate Behavior Logs:**

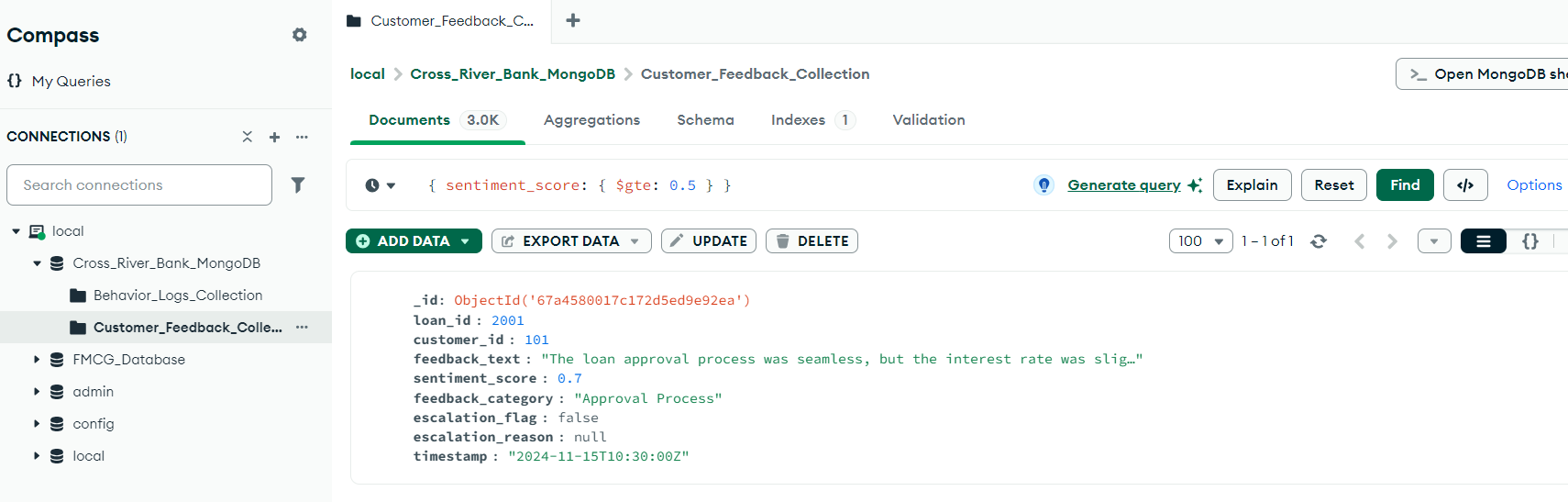
Remove duplicate entries with the same customer\_id and timestamp.

There is no duplicate in behavior logs data to remove.

**4. Retrieve Positive Feedback**

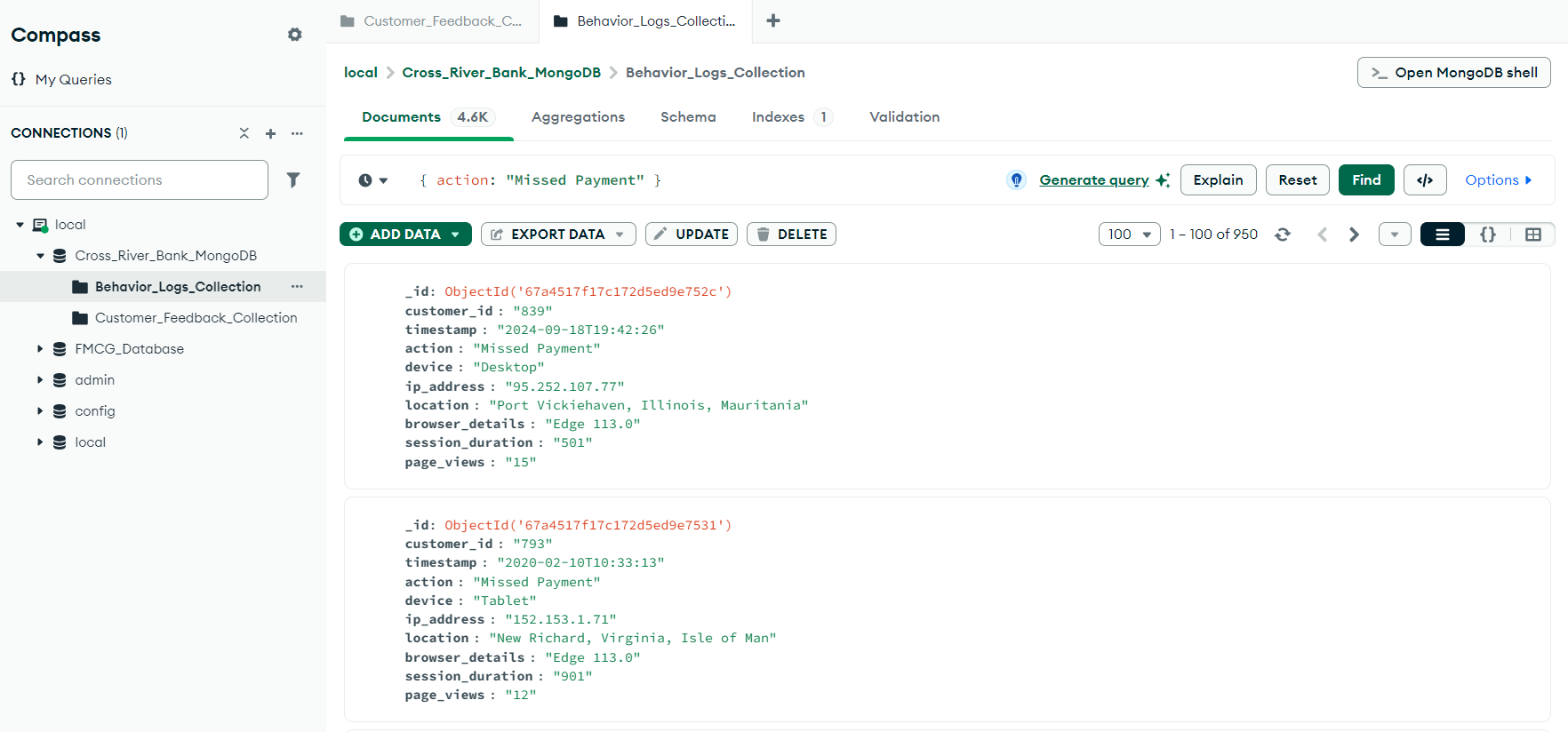
Retrieve feedback entries with sentiment scores greater than 0.5

Output



**Task 5. Fetch Logs for 'Missed Payment' Actions:**

 Retrieve behavior logs where the action is "Missed Payment."



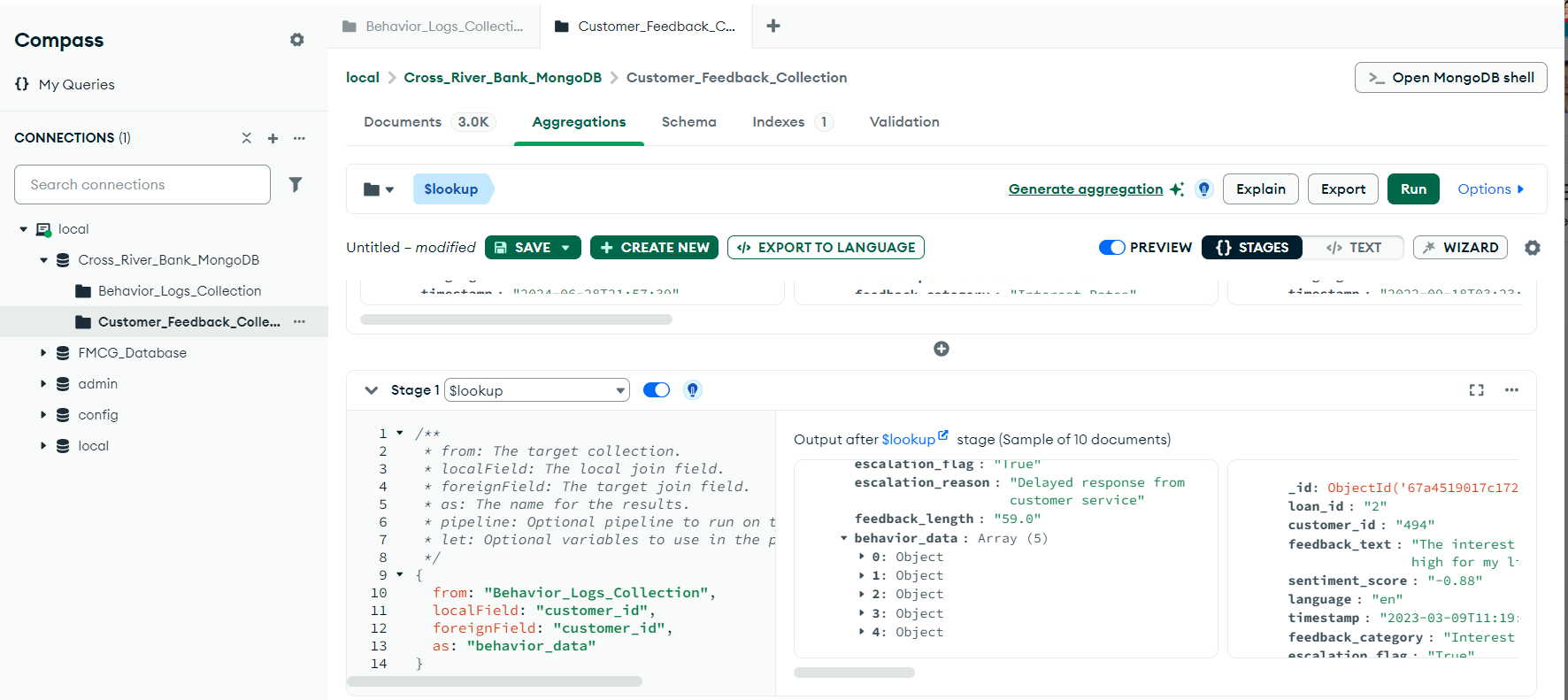
**Aggregate Pipeline task:**

1. **Join Feedback and Behavior Log - In Customer\_Feedback**

Use the $lookup stage to combine feedback from the customer\_feedback collection with behavior logs from the behavior\_logs collection, matching on customer\_id.

\*\* It will join 2 collection based on one common field.

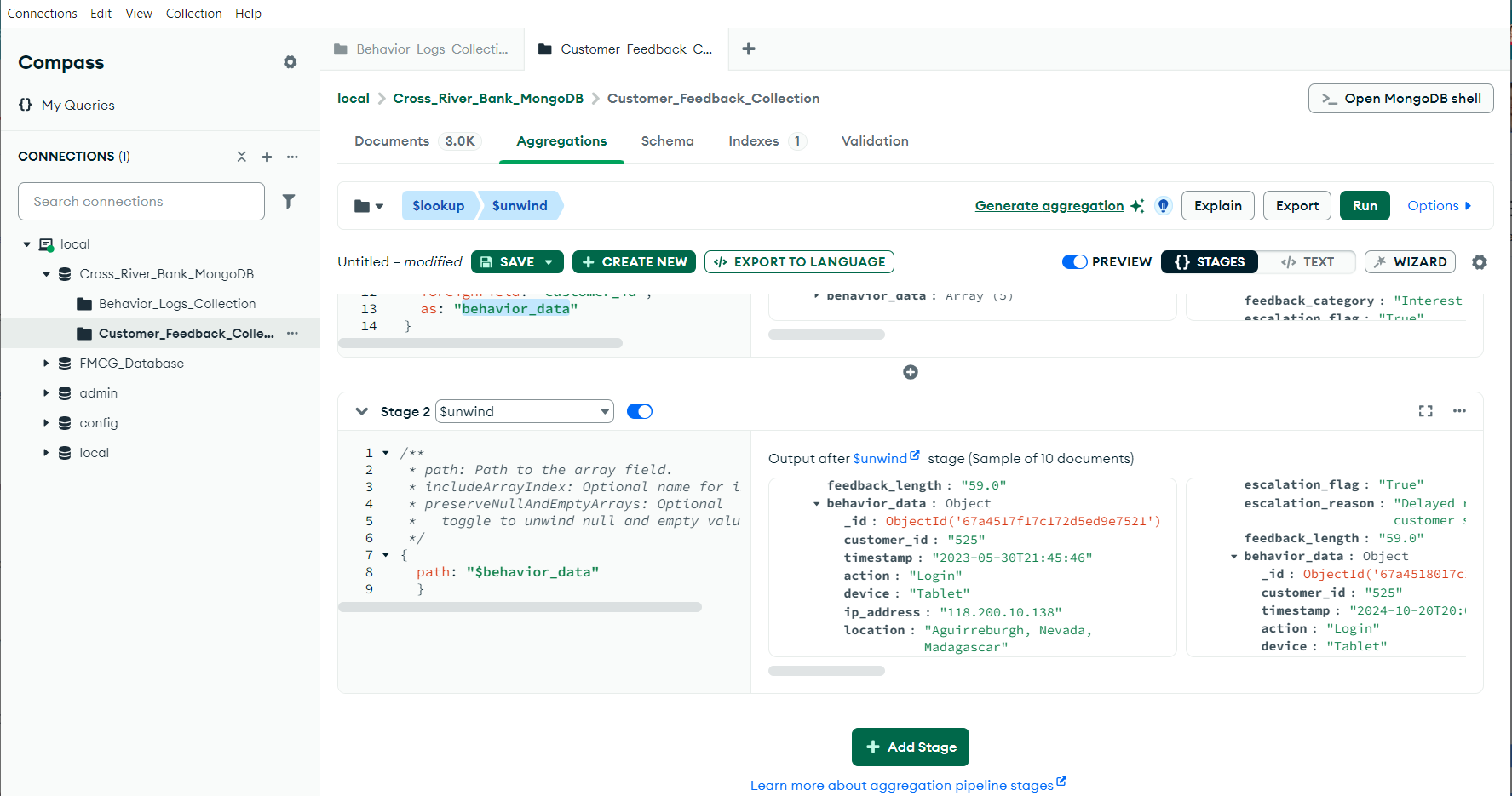
**Output:**



**2. Flatten Joined Data - In Customer\_Feedback**

Apply $unwind to flatten the nested array created by $lookup. This ensures a clean structure for subsequent aggregation stages.

\*\* It will convert the array into individual document to reaqd properly



**3. Aggregate Feedback and Behavioral Metrics - In Customer\_Feedback**

Group the combined data by customer\_id to compute:

* Average sentiment scores.
* Total missed payments.
* Total session durations.
* Count of unresolved escalations

**Codes: {**

\_id: "$behavior\_data.customer\_id",

avg\_sentiment\_score:

{"$avg": {"$toDouble":"$sentiment\_score"} },

Total\_missed\_payments:{"$sum":{"$cond":

[{"$eq":["$behavior\_data.action", "Missed Payment"] }, 1, 0]}},

Total\_session\_durations:{"$sum":

{$toDouble:"$behavior\_data.session\_duration"}},

Total\_unresponsive\_escalations: {"$sum":{

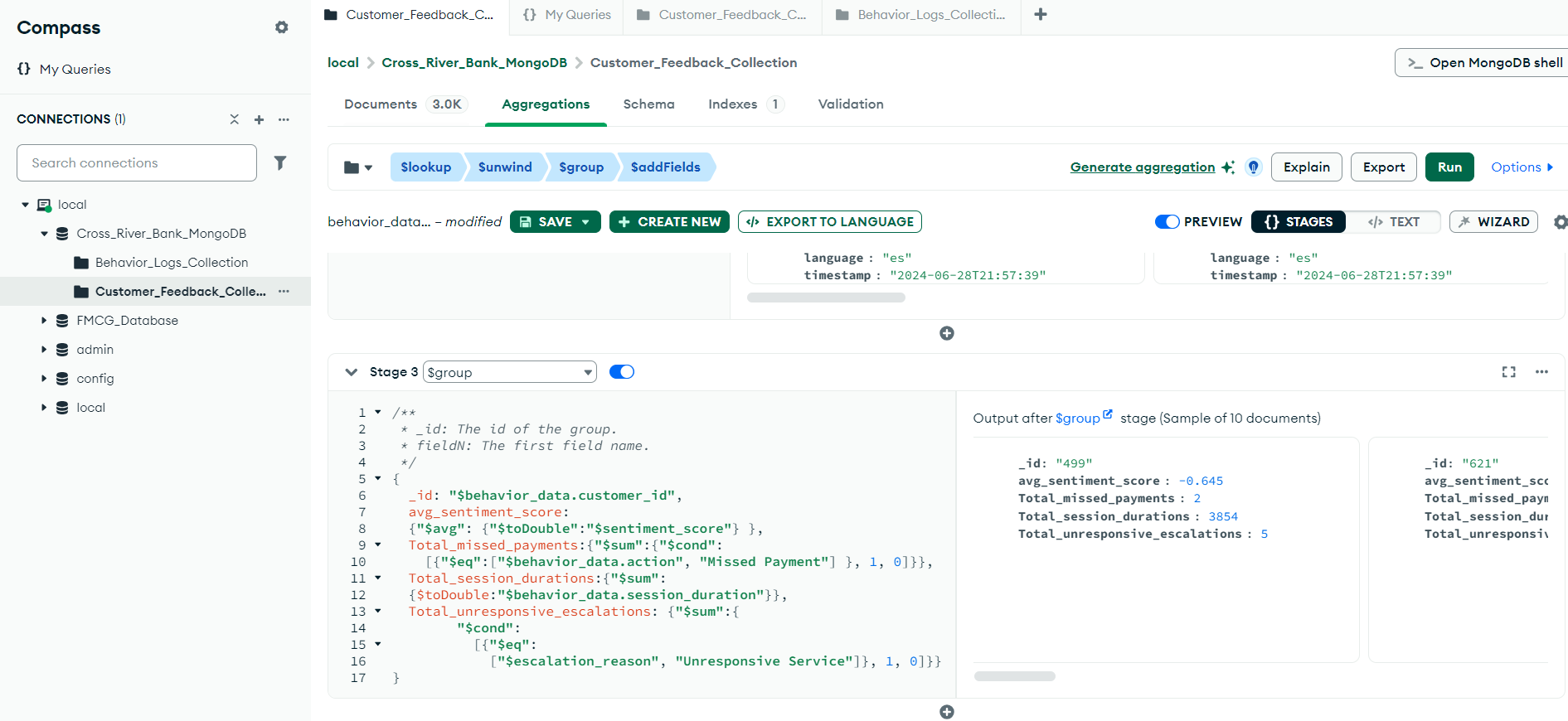
"$cond":

[{"$eq":

["$escalation\_reason", "Unresponsive Service"]}, 1, 0]}}

}

**Output:**



**4. Calculate Risk Scores – In Customer\_Feedback**

Add a composite risk score using $addFields, incorporating:

* Missed payments (weighted heavily as an indicator of risk).
* Negative sentiment (calculated as 1 - avgSentiment).
* Total session durations (scaled if necessary).
* Escalation counts.

**Codes:**

{

"composite\_risk\_score": {

$add: [

{$multiply: [{"$ifNull": ["$Total\_missed\_payments", 0] }, 5] },

{$multiply: [{$subtract: [1, {"$ifNull": ["$avg\_sentiment\_score", 1] }] },

3]},

{$multiply: [{$divide: [{"$ifNull": ["$Total\_session\_durations", 0] }, 1000] },

1]},

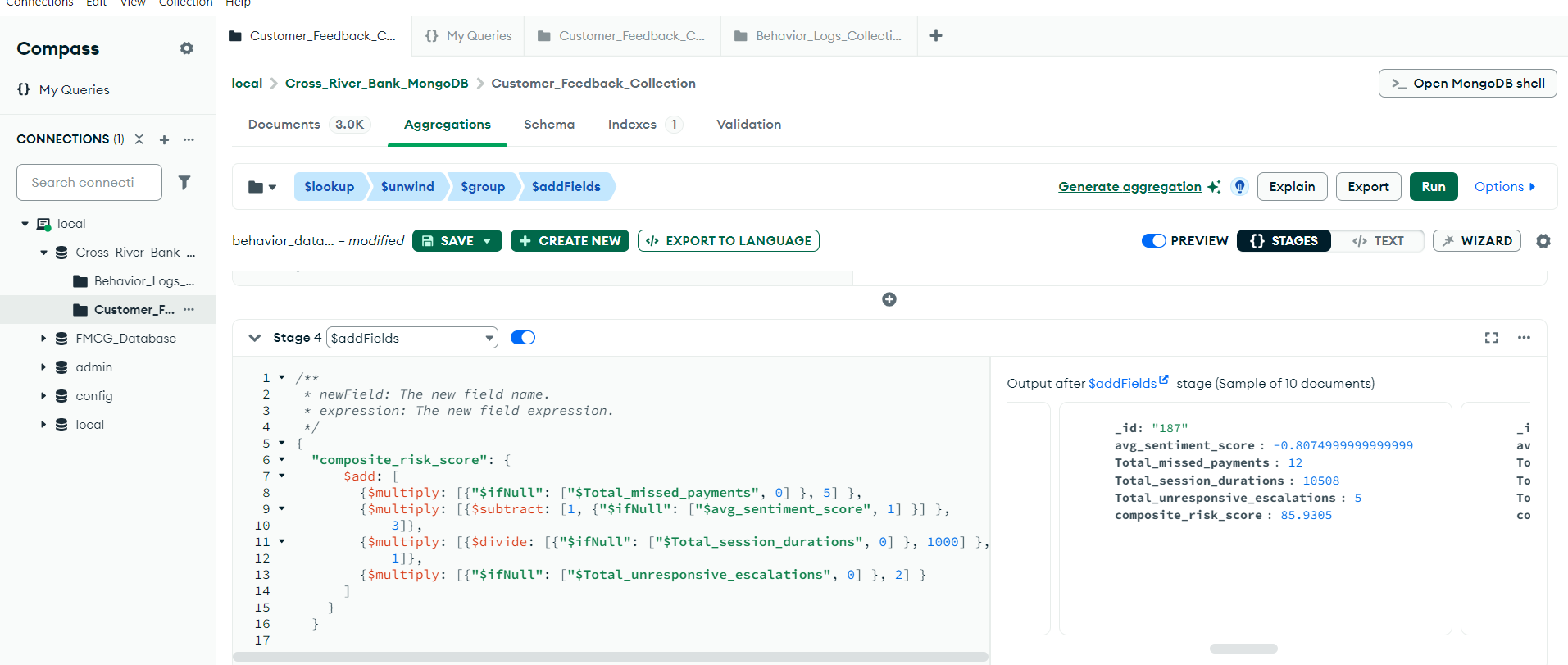
{$multiply: [{"$ifNull": ["$Total\_unresponsive\_escalations", 0] }, 2] }

]

}

}

**Output:**



**5. Identify Device Usage Trends – In Behavior\_Log**

Group behavior logs by device to detect preferences and anomalies in usage patterns, especially for high-risk actions like "Missed Payment."

**Code:**

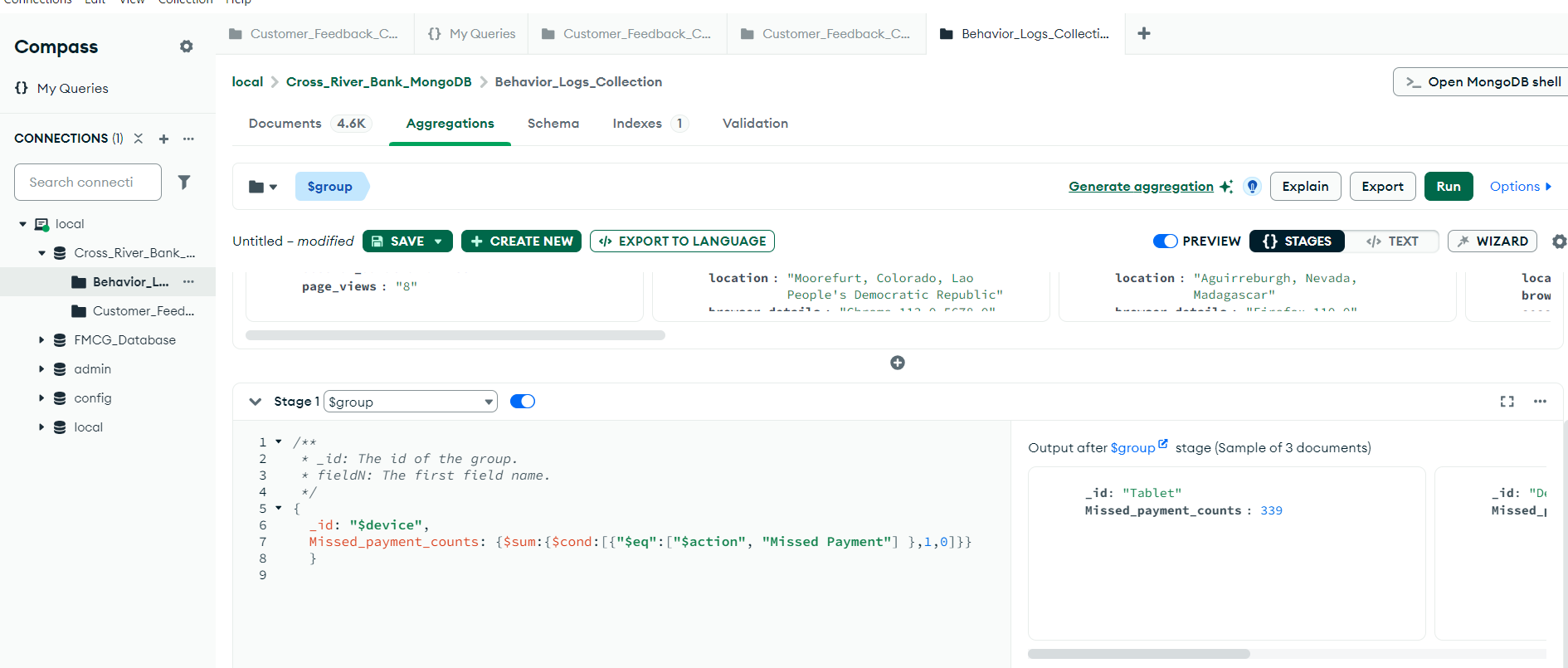
{

\_id: "$device",

Missed\_payment\_counts: {$sum:{$cond:[{"$eq":["$action", "Missed Payment"] },1,0]}}

}

**Output:**





**6. Detect Session Outliers**

Use the $bucket to identify customers with session durations that fall into extreme ranges (e.g., above the 90th percentile or below the 10th percentile).

**Code:**

{

groupBy: "$session\_duration",

boundaries: [0,300, 600,900,1200,1500,1800],

"default": "Outliers",

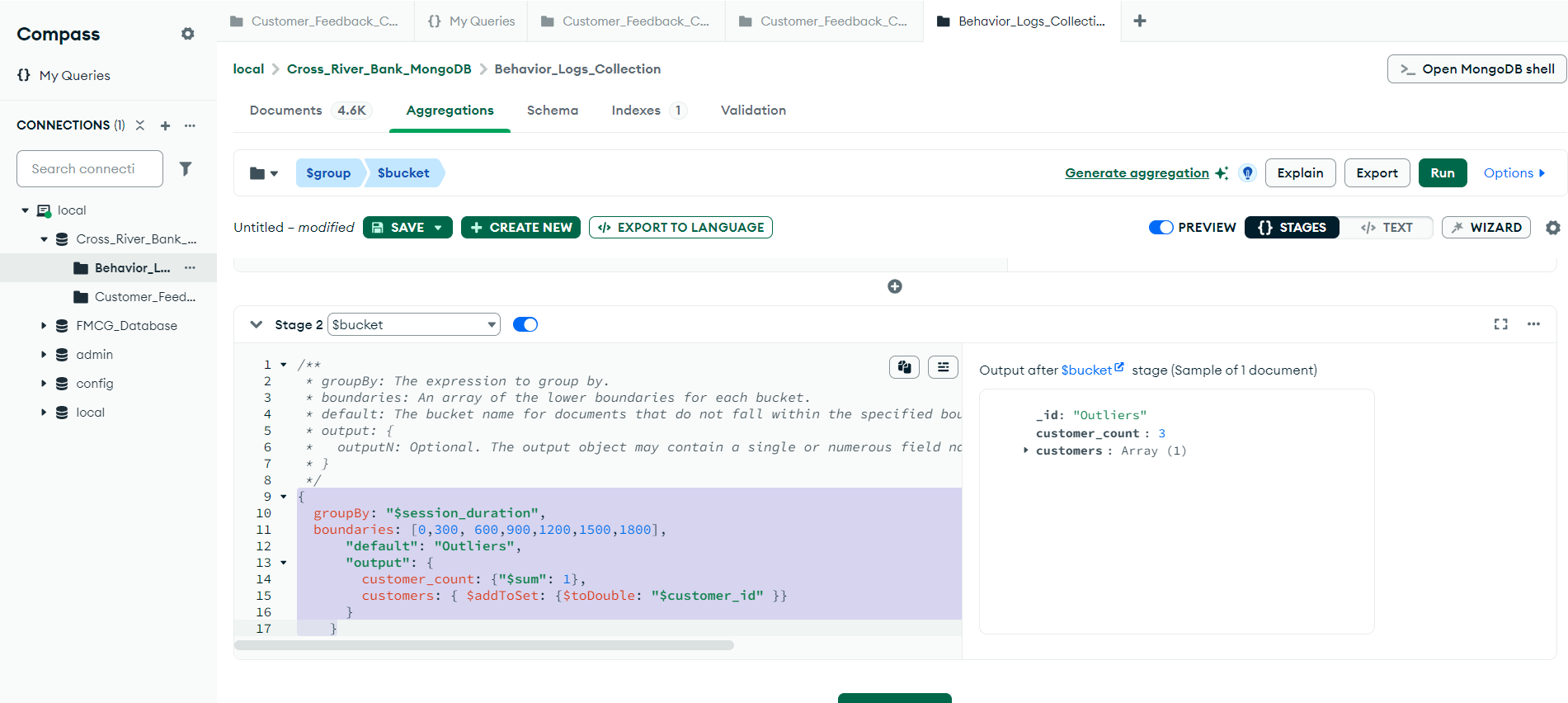
"output": {

customer\_count: {"$sum": 1},

customers: { $addToSet: {$toDouble: "$customer\_id" }}

}

}



**7. Escalation Analysis**

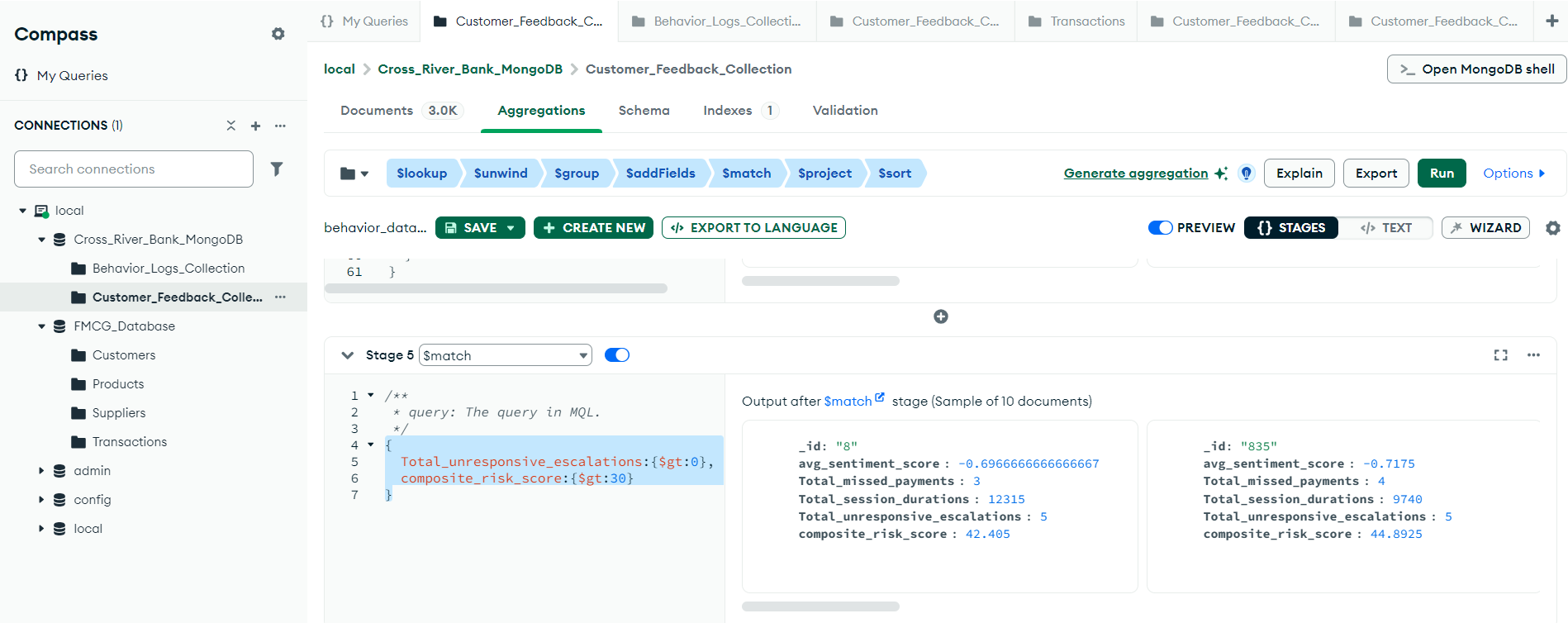
* Filter for customers with:
* Non-zero escalation counts.
* High composite risk scores.
* Highlight these customers for immediate follow-up and prioritization.
* **Code:**
  1. **Match**

{

Total\_unresponsive\_escalations:{$gt:0},

composite\_risk\_score:{$gt:30}

}



* 1. **Project**

{

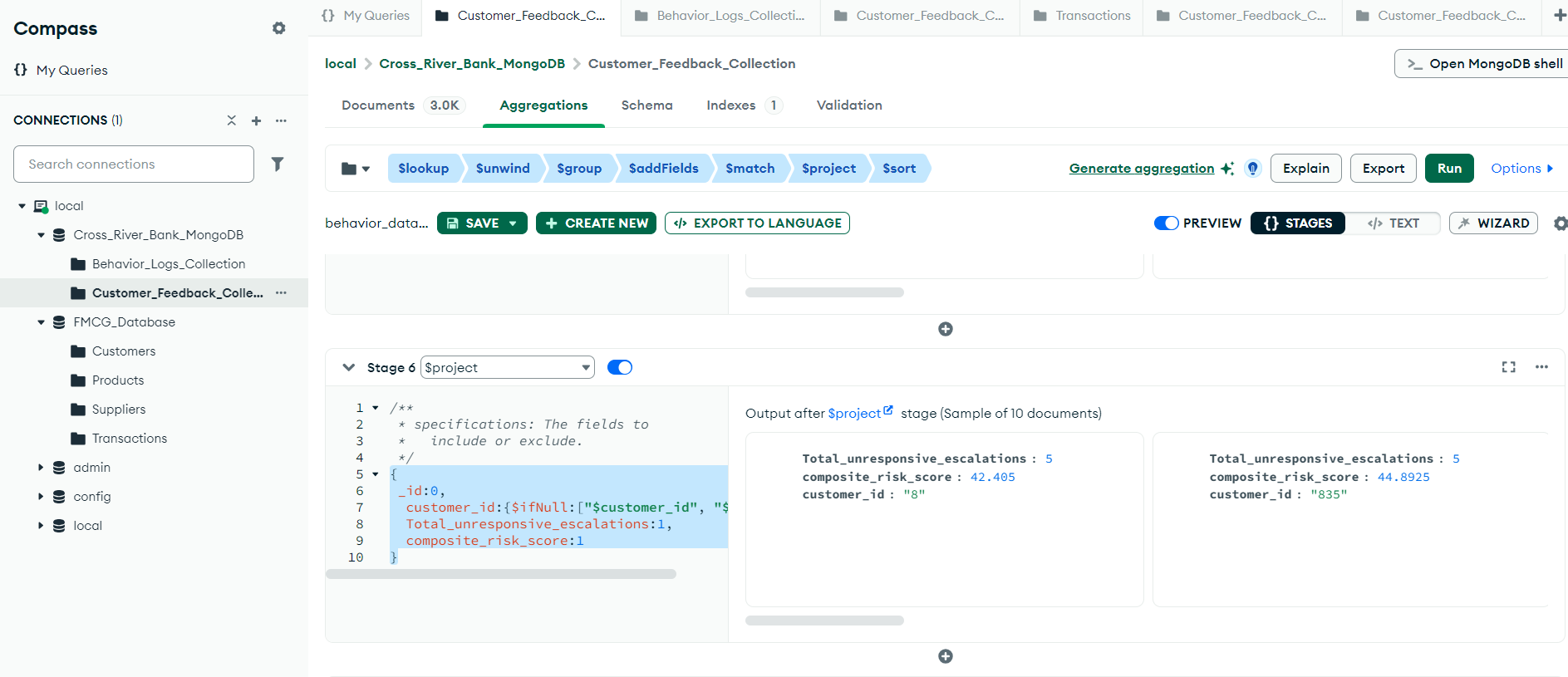
\_id:0,

customer\_id:{$ifNull:["$customer\_id", "$\_id"]},

Total\_unresponsive\_escalations:1,

composite\_risk\_score:1

}

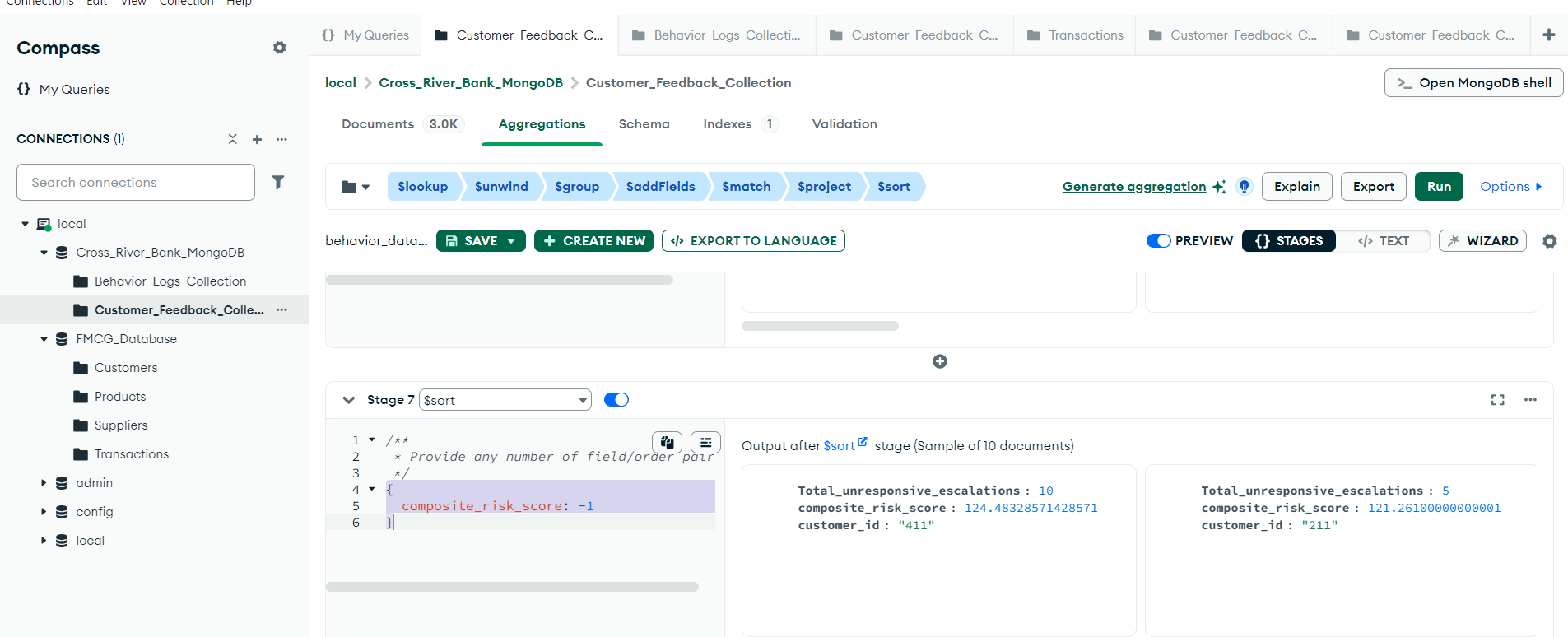


* 1. **Sort**

{

composite\_risk\_score: -1

}



**Final Output of Aggregation Pipeline:**

