ORACLE

FUSION MIDDLEWARE

BUSINESS ACTIVITY MONITORING

Event Stream Analytics in Oracle BAM 12.1.3

ORACLE WHITE PAPER | SEPTEMBER 2014

ORACLE

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Table of Contents

Introduction

This document provides information on using the Oracle BAM Event Stream Analytics sample project in Oracle BAM 12c by elaborating on the BAM Tree Model Query, Active Data Service (ADS) and Continuous Query features. There is some sample data provided to simulate the twitter feeds, which Oracle BAM monitors real time. This data is well-suited for the continuous query to detect patterns from streaming events.

This project depicts a scenario where a user uses Oracle BAM to monitor customer sentiments toward various products by real time monitoring the tweets in real time. The process is as follows:

1. Customers tweet their sentiments toward various products.
2. Oracle Enterprise R is used to perform text mining to look for certain key words.
3. Based on the results from 2, sentiment score (negative or positive) is assigned for a given tweet.
4. Oracle BAM monitors these tweets and looks for patterns from various customer responses in two ways –
    a. Monitor customer sentiment toward various products, visualized by 'Tree Map' visualization.
    b. Check for irate customers by looking for negative tweets from the same customer within a given time interval.

Using the Oracle BAM Tree Model Query and Business Views, you can present feedback for the eight products using color-coded areas. With the dashboard, you can gauge customer sentiments toward various products.

Accessing the Project

This project uses Oracle BAM Monitor Count Template (this monitors streaming events coming into Oracle BAM and looks for 3 negative/positive tweets for a given customer within a period of time) and Oracle BAM Tree Model Query. Monitor Count Templates and Tree Model Query are new features in Oracle BAM 12c. Monitor Count Template uses CQL and the embedded Oracle OEP Engine to support the query. Tree Model Query uses Oracle ADF technology.

Prerequisites

1. Copy BAM "samples" directory to ORACLE_HOME/soa/bam.
2. Set environment variable JAVA_HOME.
3. Update../../bin/ BAMCommandConfig.xml and add the following parameters:

    (make sure to replace **password** with valid password):
    <password>**password**</password>
4. If there's a previous project BAMEventStreamAnalytics installed, clear the continuous queries or delete the project. If you are importing the BAMEventStreamAnalytics project for the first time, you can ignore the following steps.
    a. Open Oracle BAM composer: http://<hostname>:<port>/bam/composer

b. Choose 'Administrator'

c. Click 'Continuous Queries Monitoring'

d. Under the 'Project' dropdown list, select BAMEventStreamAnalytics

e. Check the 'select all' checkbox

f. Click 'Deactivate Query'

g. Click 'Drop Query'

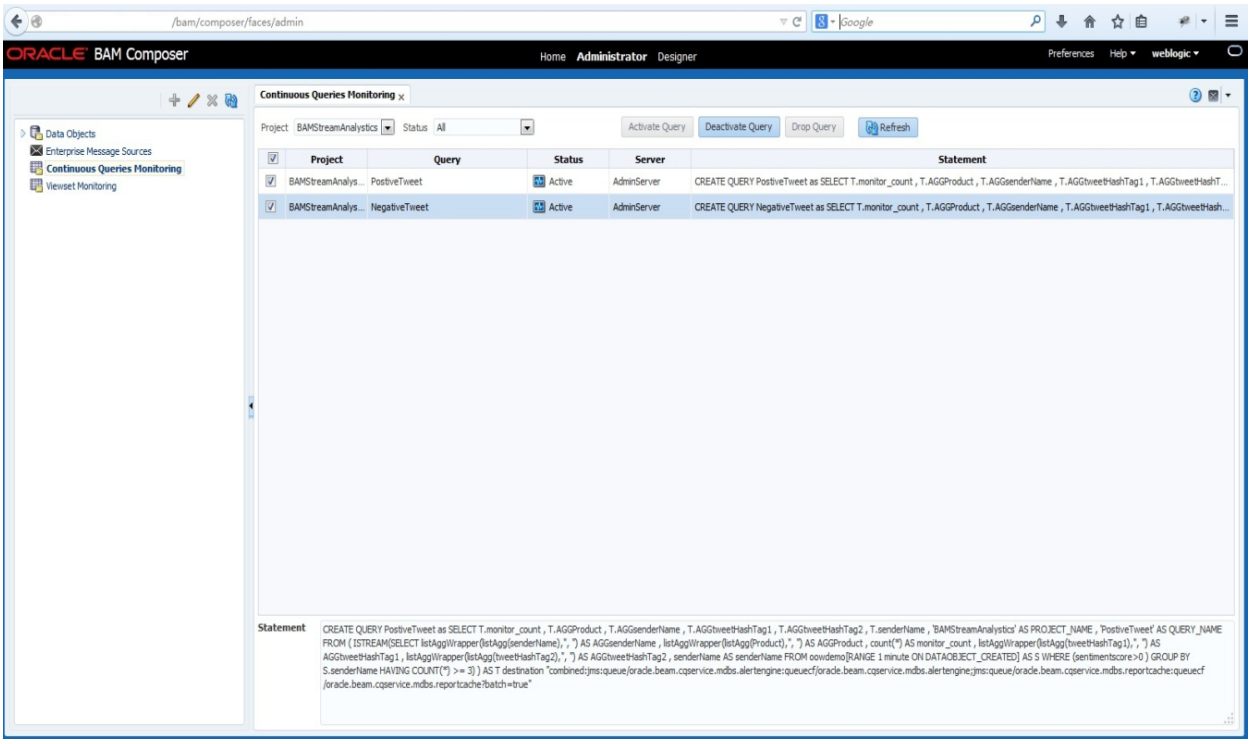


Figure 1 – Continuous Query Monitoring in Oracle BAM Composer

Importing the Project

1. Enter bam-105-twitter-feed directory.

2. Execute importTwitterFeed.sh <wls password>

After the import is complete, you can open the project and view all components. Oracle BAM uses **bamcommand** to import projects into an Oracle BAM 12c runtime environment. If you only have the archived project file, you can use **bamcommand** to import the project to a running Oracle BAM 12c environment. The following is a sample usage for **bamcommand** to import this project.

$ORACLE_HOME/soa/bam /bin/bamcommand -username weblogic -host localhost -port 7001 -cmd import -type project -mode update -file TwitterFeed.zip

Running the Project

This project displays as Oracle BAM Event Stream Analytics, as shown in Figure 2, and includes two data objects, five business queries, five business views, and two dashboards.

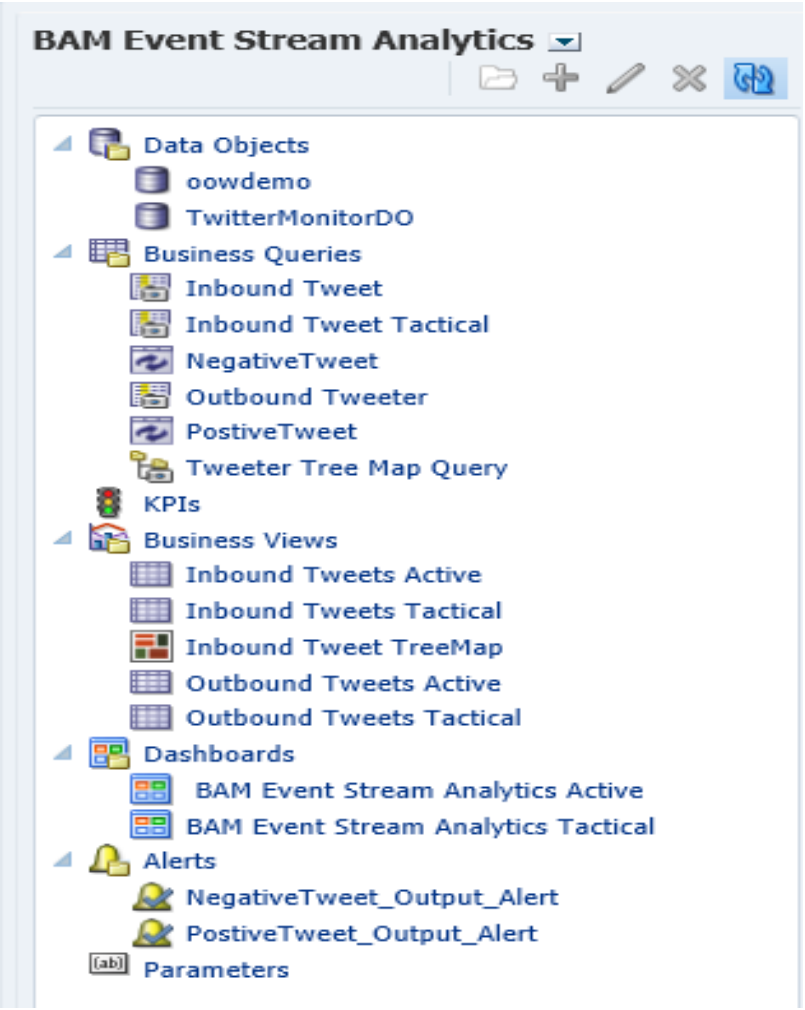


Figure 2 – Oracle BAM Event Stream Analytics Project

Data Objects

Simple Data Object: oowdemo

oowdemo is an **archived relation** simple data object. It is used for collecting twitter feedback data in real time from the ***Negative Tweet*** and ***Positive Tweet*** continuous queries. The structure of the data object is as follows. In this sample, the simulation data is populated by **bamloadgen**.

Column Name	Column Type	Data Type	Description
tweetText	DIMENSION	VARCHAR	The feedback message
tweetDate	ATTRIBUTE	DATETIME	The date for the feedback
tweetSource	ATTRIBUTE	VARCHAR	The source of the message
tweetHashTag1	ATTRIBUTE	VARCHAR	The tag of the message
tweetHashTag2	ATTRIBUTE	VARCHAR	The tag of the message
senderName	DIMENSION	VARCHAR	The owner to send the message
senderFriendsCount	DIMENSION	INT	The sender's friend count
senderFollowersCount	ATTRIBUTE	INT	The sender's follower count
senderLanguage	ATTRIBUTE	VARCHAR	The language for the sender

senderLocation	ATTRIBUTE	VARCHAR	The location for the sender
sentimentsscore	DIMENSION	INT	The score for the sentiment
Product	ATTRIBUTE	INT	The evolution's product
SentimentDislay	MEASURE	Calculated Field	If(sentimentscore>0)Then(Concat(Concat(""))Else(If(sentimentscore<1)Then(Concat(Concat(""))Else(Concat(Concat("")))
SentimentScore	MEASURE	Calculated Field	sum(sentimentscore)

Simple Data Object: TwitterMonitorDO

TwitterMonitorDO is a simple data object as well. The outputs of the ***Negative Tweet*** and ***Positive Tweet*** continuous queries *can be written back to this data object*. A Tree Model Query presents all ***TwitterMonitorDO*** data.

Column Name	Column Type	Data Type	Description
tweetHashTag1	ATTRIBUTE	VARCHAR	The tag of the message
tweetHashTag2	ATTRIBUTE	VARCHAR	The tag of the message
senderName	ATTRIBUTE	VARCHAR	The name of the sender
monitorcount	ATTRIBUTE	INT	The evolution count for every sender
ProjectName	ATTRIBUTE	VARCHAR	The project name for the CQL query
QueryName	ATTRIBUTE	VARCHAR	The CQL query name
Product	ATTRIBUTE	VARCHAR	The product name for the evolution

Consider a scenario where you are monitoring product sales efficiency for a company. Your primary goal is that you must monitor sales cost profits. To monitor sales costs and associated profits, you create a SalesCostProfit analysis dashboard that satisfies common business needs like analyzing sales data to identify trends, gauge sales performance, and study sales costs.

These graphical reports allow sales teams to access minute-to-minute information to achieve projected targets and resolve operational challenges. The SalesCostProfit analysis dashboard shows you a snapshot of sales activity. You can analyze metrics like overall sales against the incurred cost, key loss or profit-making transactions, geographical sales tracking, and so on. This process mandates that you make the following considerations.

Business Queries

Flat SQL Query: Inbound Tweet

The Inbound Tweet query retrieves the Product, senderName, Sentiment Graphic and tweethashTag2 information from the **oowdemo** data object.

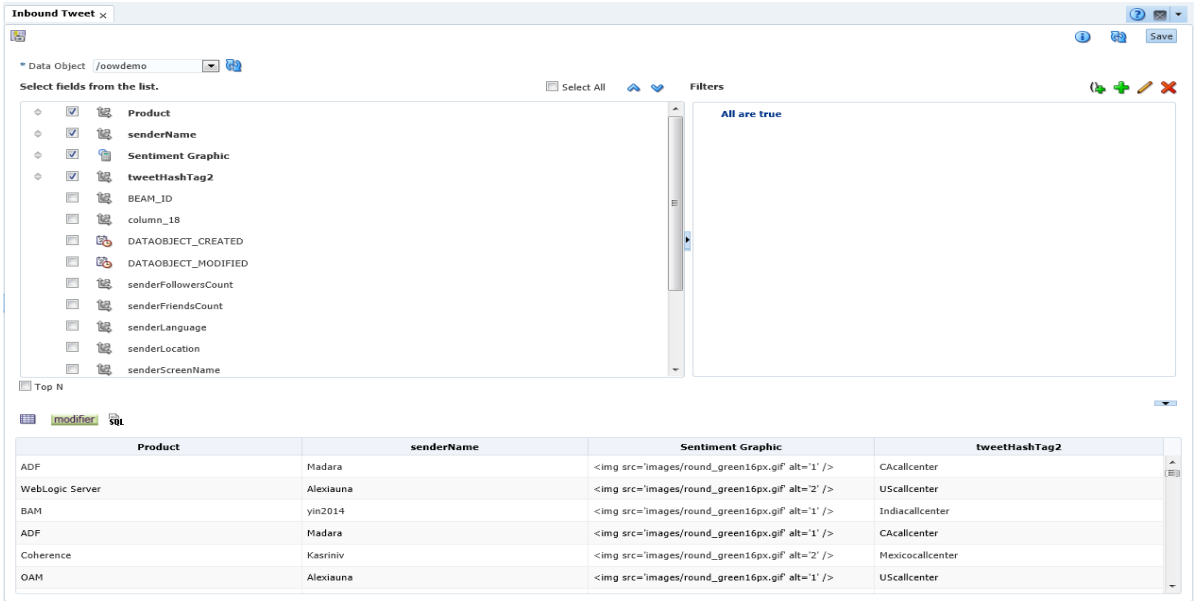


Figure 3 – Inbound Tweet Query Page

Flat SQL Query: Inbound Tweet Tactical

The Inbound Tweet Tactical query retrieves the Product, senderName, Sentiment Graphic and tweethashTag2 information from the **oowdemo** data object. It is ordered by the **DATAOBJECT_CREATED** date time column.

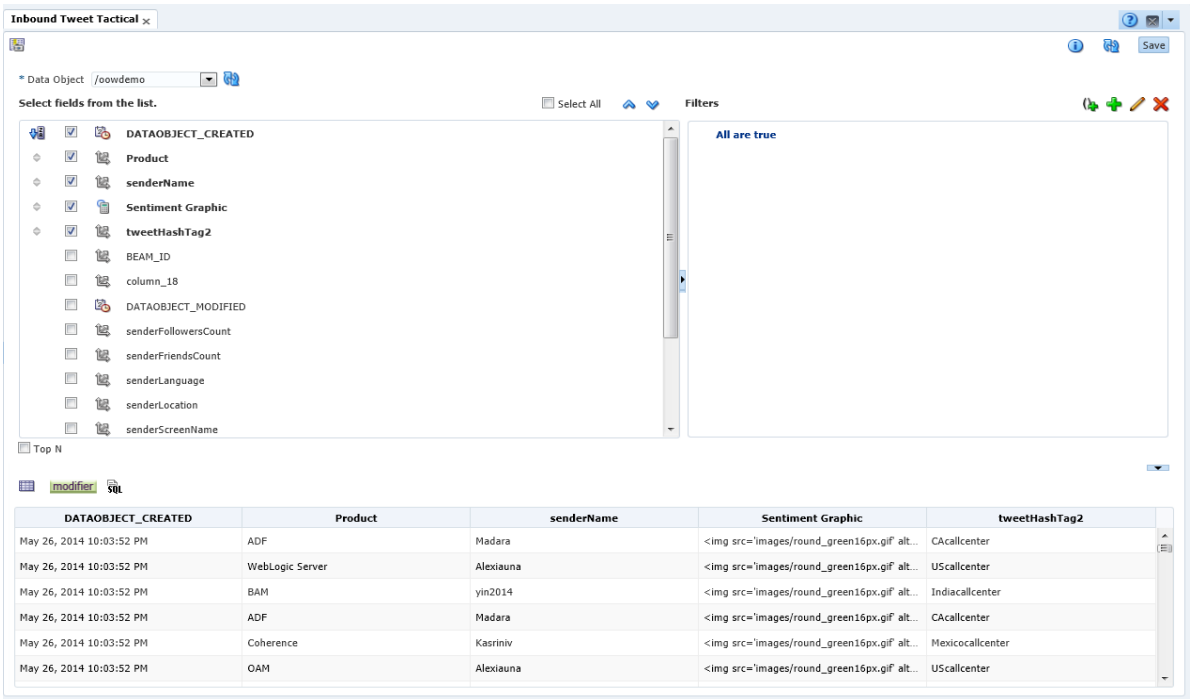


Figure 4 – Inbound Tweet Tactical Query Page

Flat SQL Query: Outbound Tweeter

This query is used for retrieving the monitorcount, QueryName, senderName and tweetHashTag2 information from the ***TwitterMonitorDO*** data object with a filter QueryName field that is equal to the '**NegativeTweet' query**.

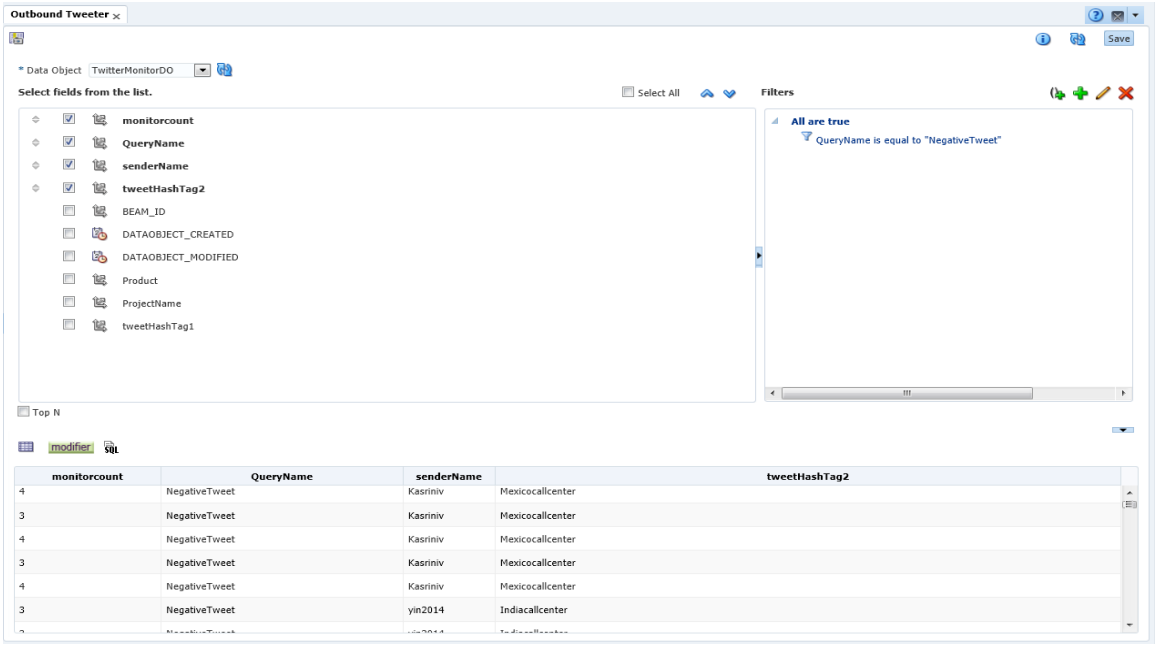


Figure 5 – Outbound Tweeter Query Page

Continuous Query: Negative Tweet

The Negative Tweet query is created with the ***Monitor Count Template*** and is triggered when the volume of bad sentiment scores is greater than three for the same user, within the rolling window of one minute. In this query, it is selected with the Using Rolling Window option, and the range length is set to one minute. The query's output is written back to the **TwitterMonitorDO** data object with the Oracle BAM alert.

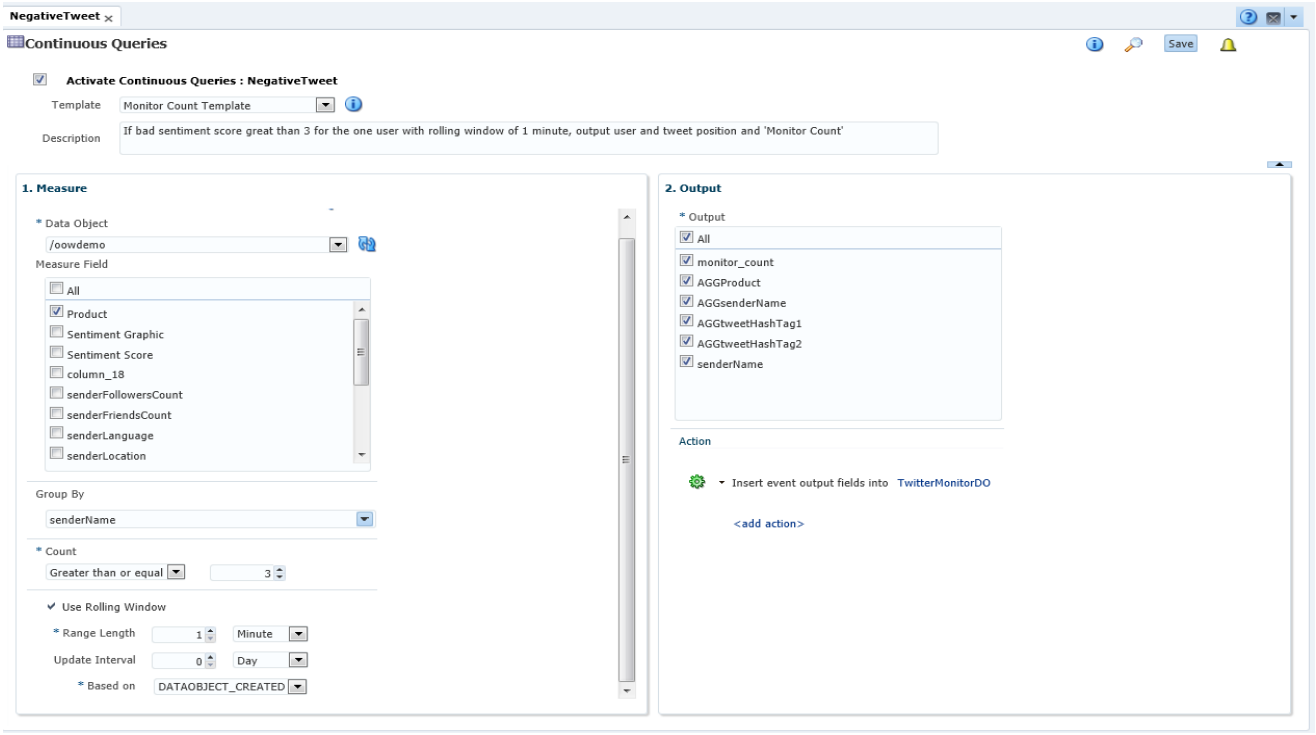


Figure 6 – Negative Tweet Query Page

By clicking the **Filter** button in the UI, you see that the query's filter is configured with a **sentimentscore less than 0**, which means that the query only retrieves and handles negative feedback data. It will form the query's 'where' clause in the CQL.

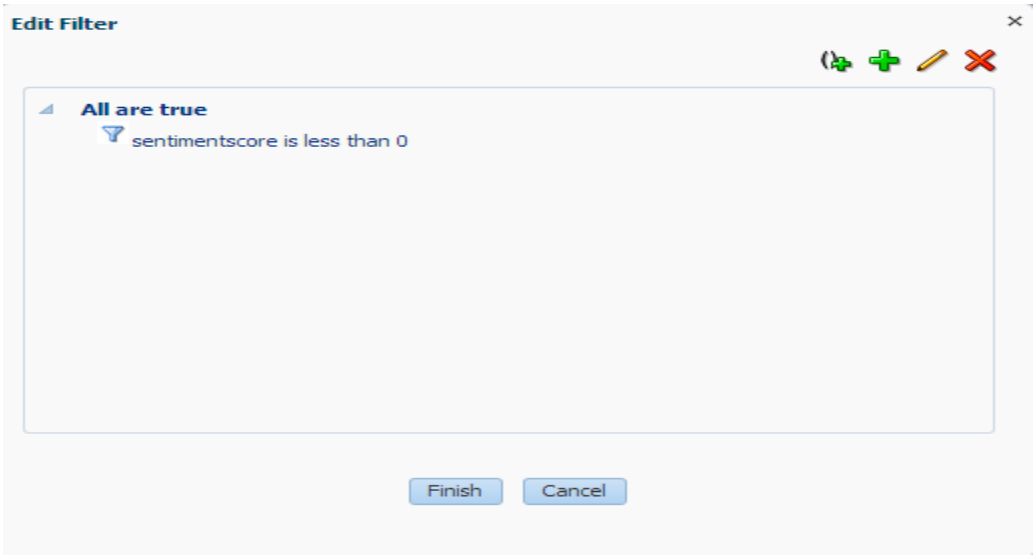


Figure 7 – Edit Filter Page

By clicking the **'TwitterMonitorDO'** link in the **Action menu**, you can review the query's related alert.

Figure 8 – Action Menu Accessed Through the Edit Filter Page

This alert aims to put the query's output write back to the **TwitterMonitorDO** data object.

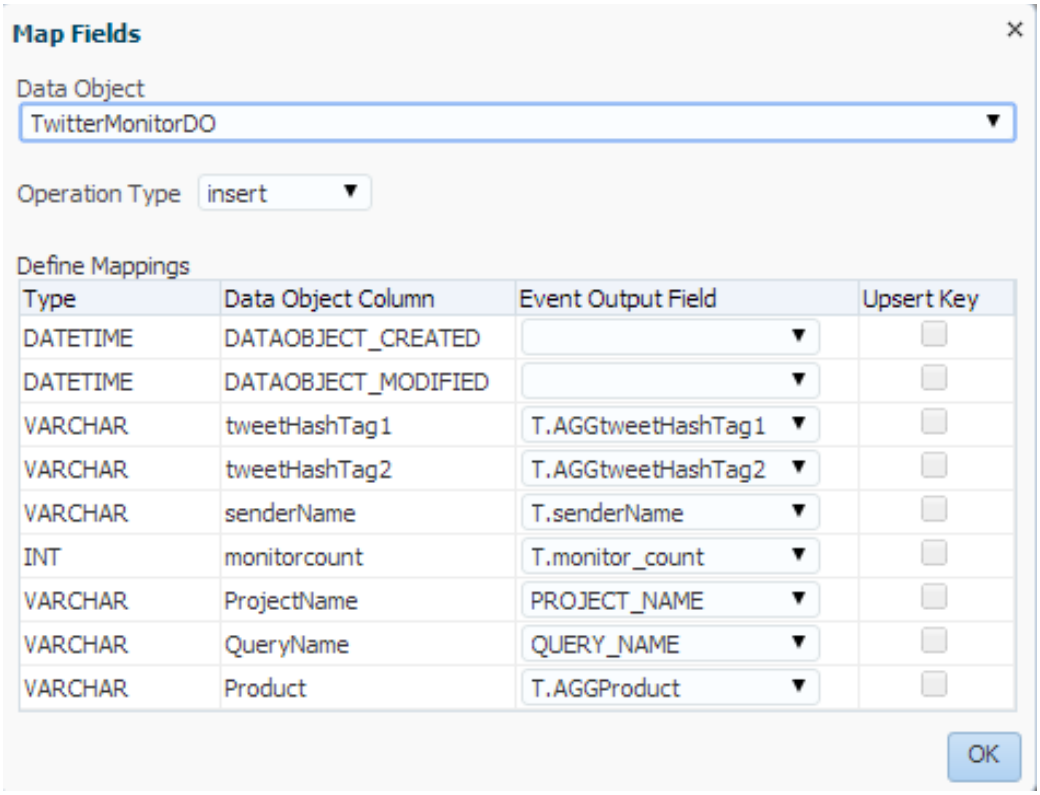


Figure 9 – Map Fields Edit Page

Continuous Query: Positive Tweet

The Positive Tweet query is created with the Monitor Count Template and is triggered while the volume of good sentiment scores is greater than three for the same user, with the rolling window of one minute. In this query, it is selected with the Using Rolling Window option, and the range length is set to one minute. The query's output is written back to the data TwitterMonitorDO object.

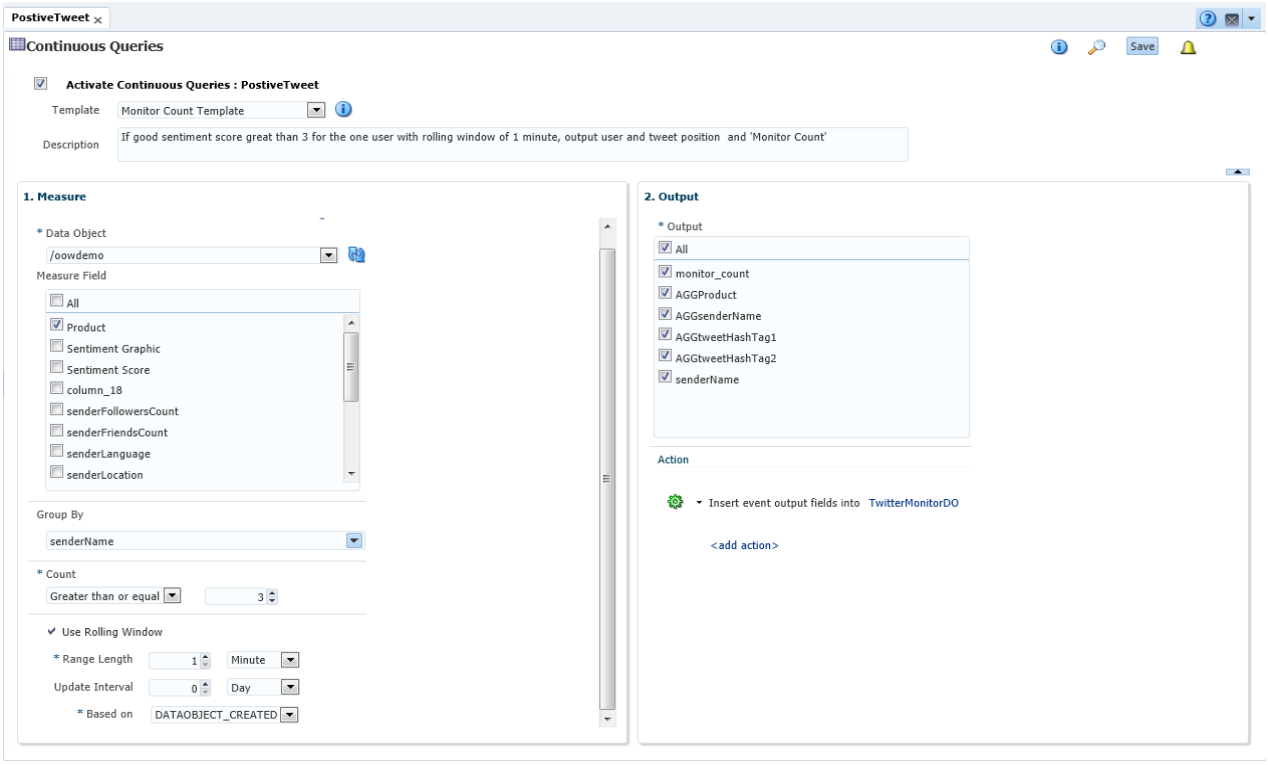


Figure 10 – Positive Tweet Query Page

By clicking the Filter button in the UI, you see that the filter is configured with sentimentscore greater than 0, which means that the query only handles negative feedback data. It forms the query's 'where' clause in the CQL.

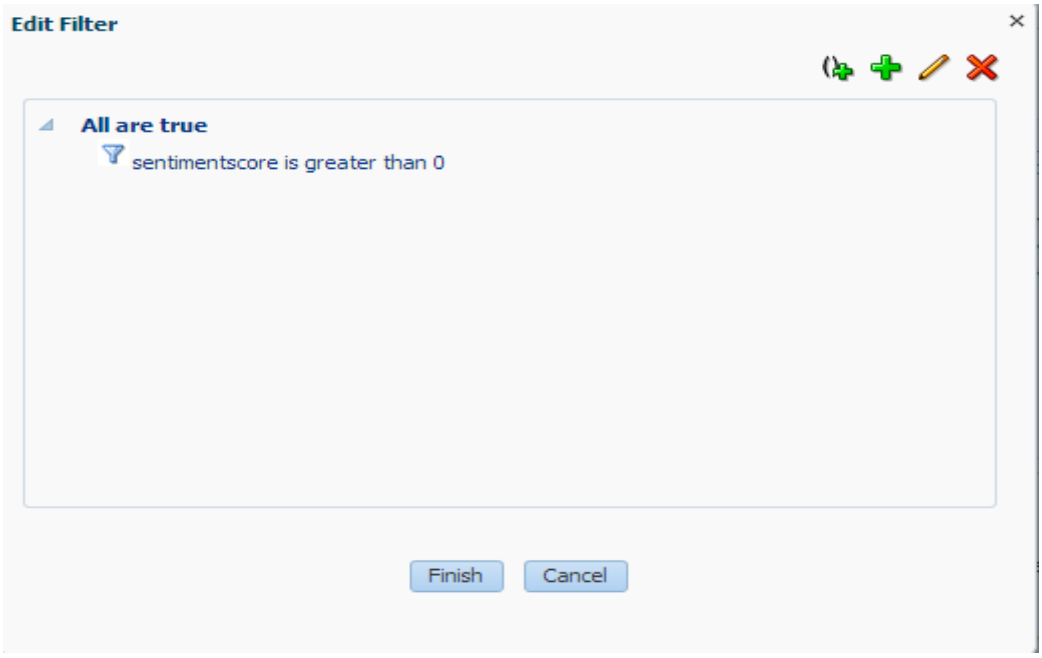


Figure 11 – Edit Filter page showing the 'sentimentscore is greater than 0' filter.

By clicking the ‘TwitterMonitorDO’ link in the Action menu, you can review the query’s related alert.

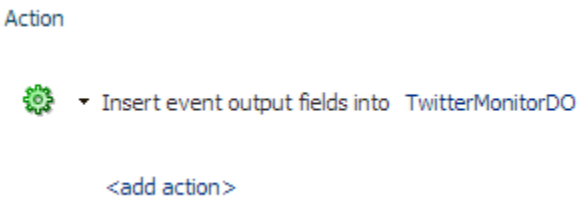


Figure 12 – Utilizing the TwitterMonitorDO Link iun the Actions Menu.

The alert aims to put the query’s output write back to the TwitterMonitorDO data object.

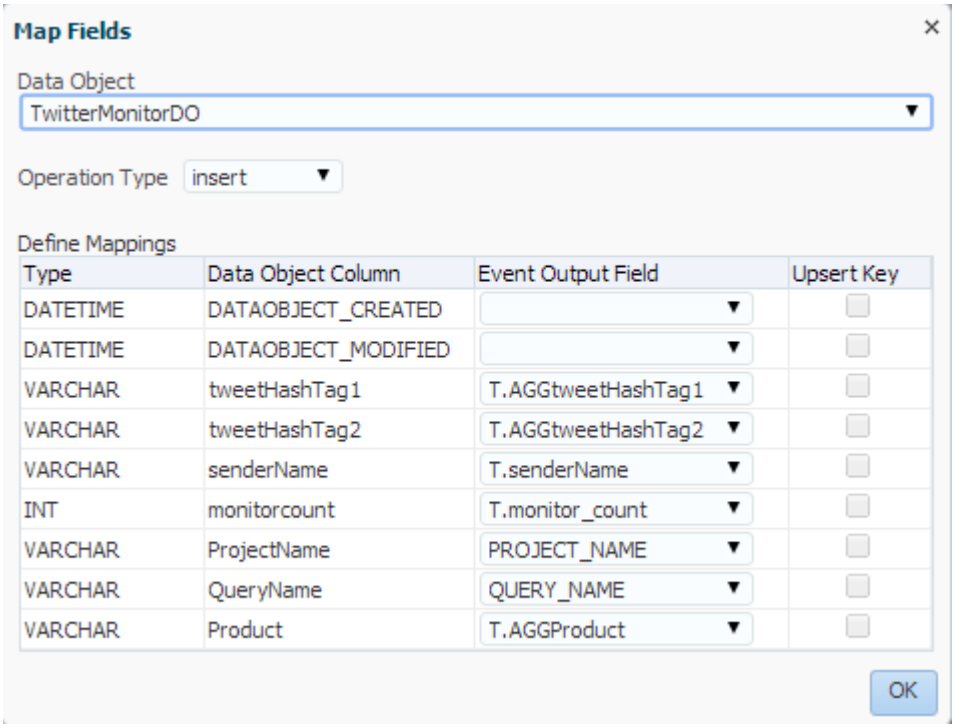


Figure 13 – The TwitterMonitorDo Data Object

For a continuous query, the same TwitterMonitorDO data object is used to write back the Negative Tweet and Positive Tweet query outputs. By adding a filter by QueryName, you can divide the two outputs. For more information on applying the QueryName filter, see the Outbound Tweeter Flat SQL Query section.

Tree Model Query: Tweeter Tree Map Query

In the Measure panel, select the Sentiment Score field (It is a calculated field of sum(sentimentscore)), and in the Hierarchy, select TwoTier hierarchy which is created within the oowdemo data object.

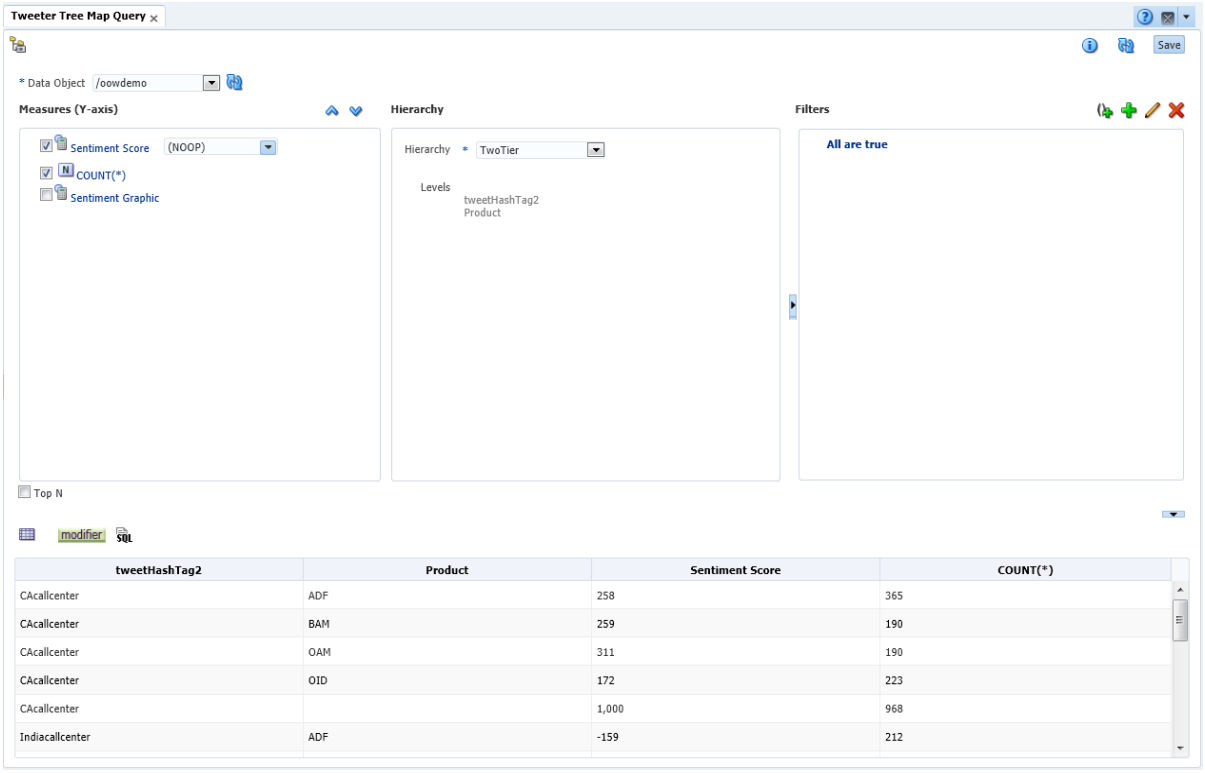


Figure 14 – The Tweeter Tree Map Query Page

The hierarchy is defined in the Hierarchies Tab in the data object before you create the query.

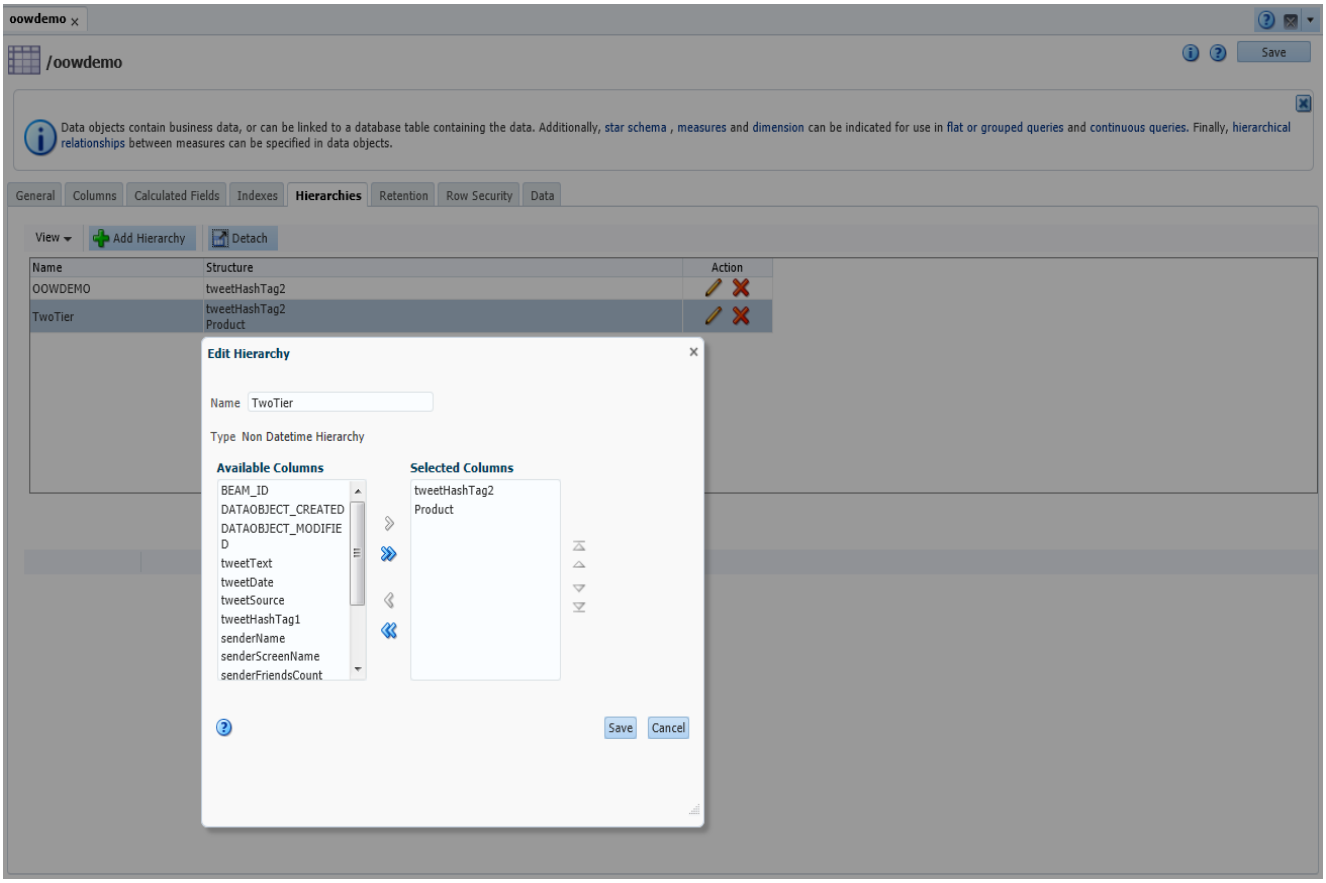


Figure 15 – The Edit Hierarchy Window

Business Views

Business View: Inbound Tweets Active

This is a **List** type view from the **Table** category business view. It is bound with the ***Inbound Tweet*** query, and is configured with the ***Runtime-Interaction*** option.

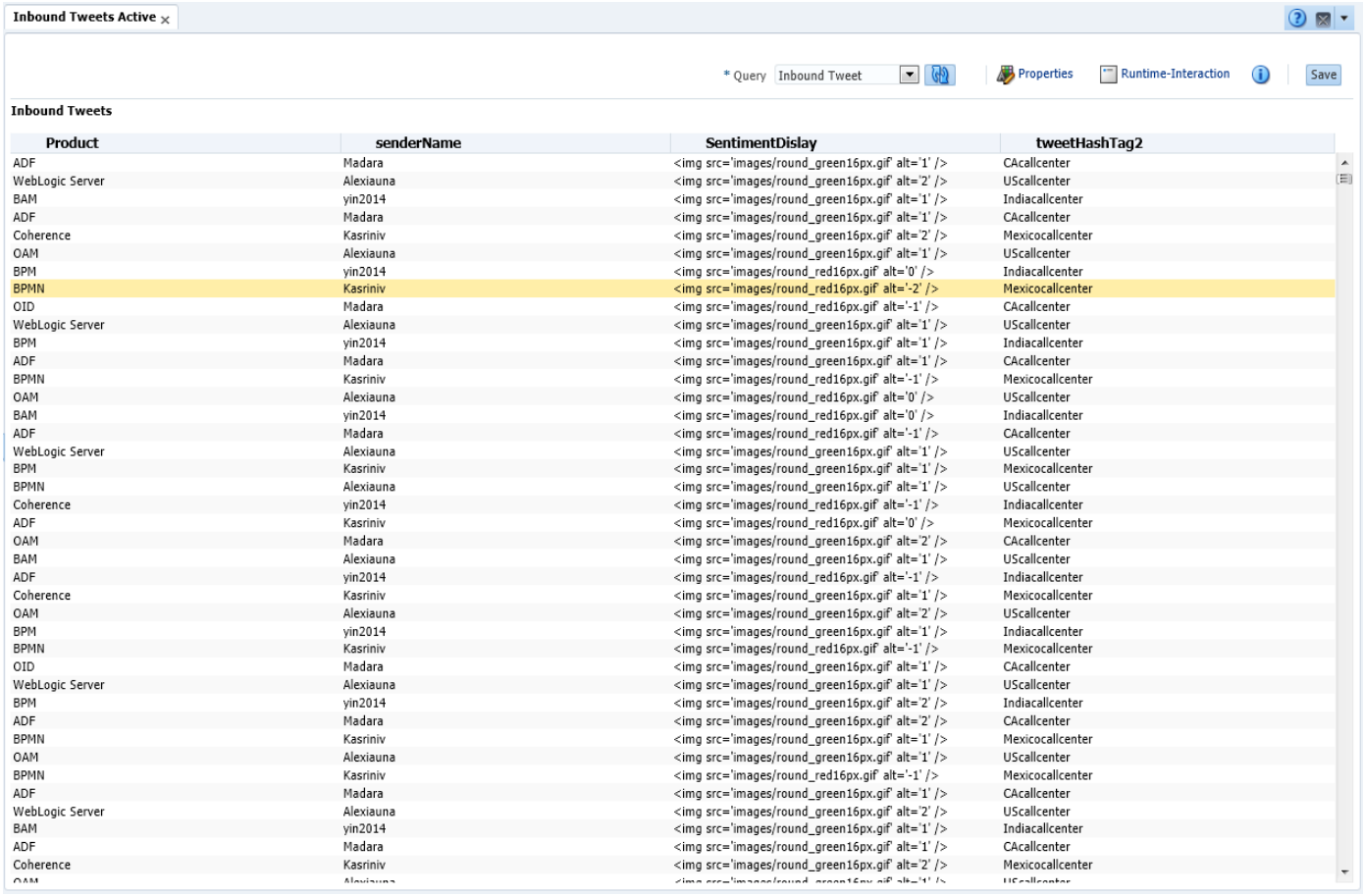


Figure 16 – The Inbound Tweets Active Page

Click ***Runtime-Interaction***, enter into **Active Data** tab, select ***Turn this query into a continuous query***, and set the ***Interval*** to 5 seconds. With this configuration, the query is refreshed automatically every 5 seconds.

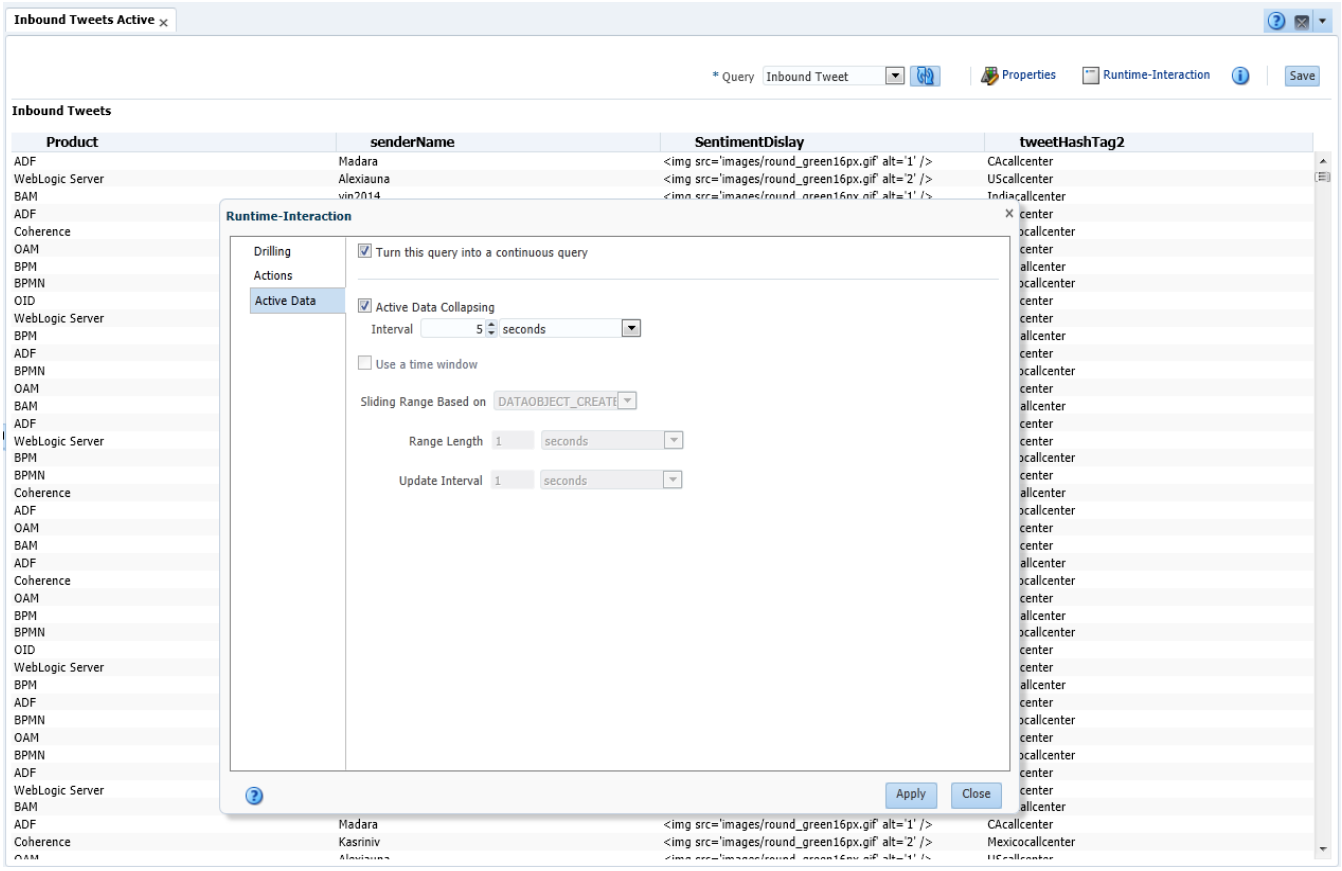
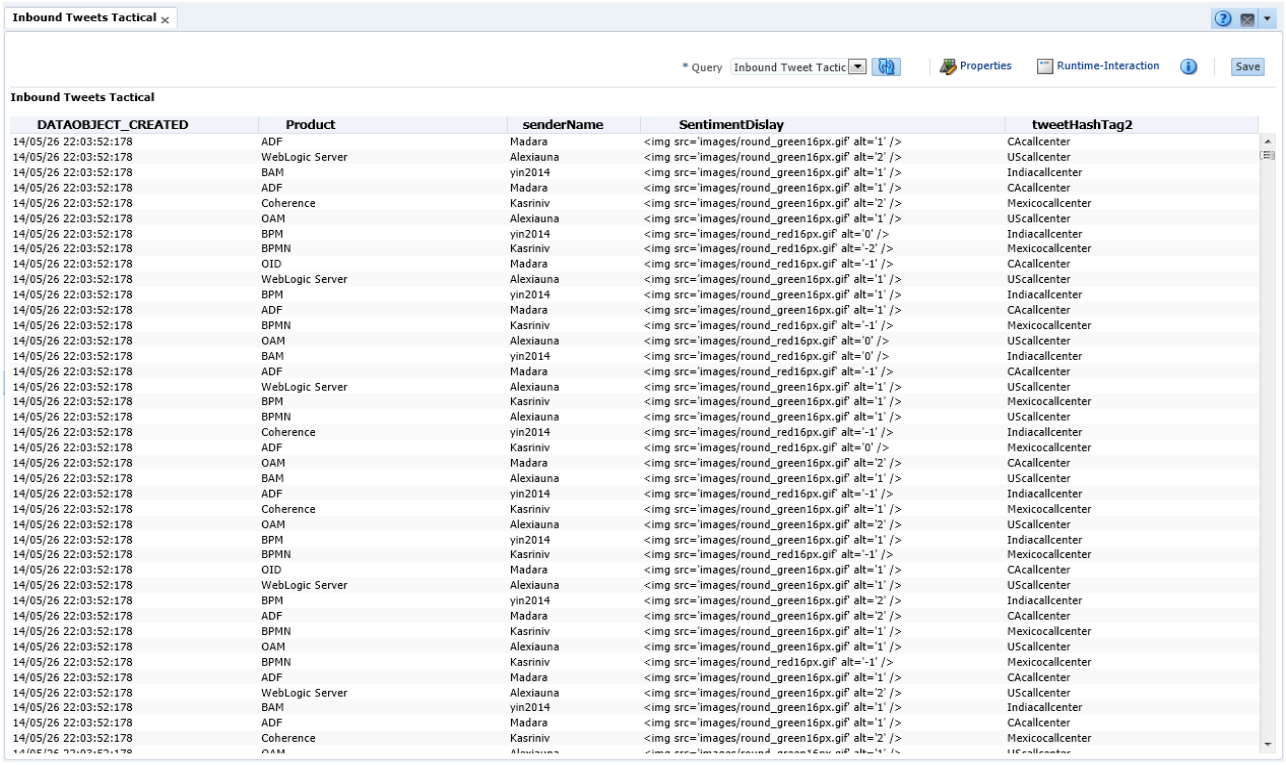


Figure 17 – Turning a Query into a Continuous Query

Business View: Inbound Tweets Tactical

This view is bound with the ***Inbound Tweet Tactical*** query.



DATAOBJECT_CREATED	Product	senderName	SentimentDislay	tweetHashTag2
14/05/26 22:03:52:178	ADF	Madara		CAcallcenter
14/05/26 22:03:52:178	WebLogic Server	Alexiauna		UScallcenter
14/05/26 22:03:52:178	BAM	yin2014		Indiacallcenter
14/05/26 22:03:52:178	ADF	Madara		CAcallcenter
14/05/26 22:03:52:178	Coherence	Kasriniv		Mexicocallcenter
14/05/26 22:03:52:178	OAM	Alexiauna		UScallcenter
14/05/26 22:03:52:178	BPM	yin2014		Indiacallcenter
14/05/26 22:03:52:178	BPMN	Kasriniv		Mexicocallcenter
14/05/26 22:03:52:178	OID	Madara		CAcallcenter
14/05/26 22:03:52:178	WebLogic Server	Alexiauna		UScallcenter
14/05/26 22:03:52:178	BPM	yin2014		Indiacallcenter
14/05/26 22:03:52:178	ADF	Madara		CAcallcenter
14/05/26 22:03:52:178	BPMN	Kasriniv		Mexicocallcenter
14/05/26 22:03:52:178	OAM	Alexiauna		UScallcenter
14/05/26 22:03:52:178	BAM	yin2014		Indiacallcenter
14/05/26 22:03:52:178	ADF	Madara		CAcallcenter
14/05/26 22:03:52:178	WebLogic Server	Alexiauna		UScallcenter
14/05/26 22:03:52:178	BPM	Kasriniv		Mexicocallcenter
14/05/26 22:03:52:178	BPMN	Alexiauna		UScallcenter
14/05/26 22:03:52:178	Coherence	yin2014		Indiacallcenter
14/05/26 22:03:52:178	ADF	Kasriniv		Mexicocallcenter
14/05/26 22:03:52:178	OAM	Madara		CAcallcenter
14/05/26 22:03:52:178	BAM	Alexiauna		UScallcenter
14/05/26 22:03:52:178	ADF	yin2014		Indiacallcenter
14/05/26 22:03:52:178	Coherence	Kasriniv		Mexicocallcenter
14/05/26 22:03:52:178	OAM	Alexiauna		UScallcenter
14/05/26 22:03:52:178	BPM	yin2014		Indiacallcenter
14/05/26 22:03:52:178	BPMN	Kasriniv		Mexicocallcenter
14/05/26 22:03:52:178	OID	Madara		CAcallcenter
14/05/26 22:03:52:178	WebLogic Server	Alexiauna		UScallcenter
14/05/26 22:03:52:178	BPM	yin2014		Indiacallcenter
14/05/26 22:03:52:178	ADF	Madara		CAcallcenter
14/05/26 22:03:52:178	BPMN	Kasriniv		Mexicocallcenter
14/05/26 22:03:52:178	OAM	Alexiauna		UScallcenter
14/05/26 22:03:52:178	BPMN	Kasriniv		Mexicocallcenter
14/05/26 22:03:52:178	ADF	Madara		CAcallcenter
14/05/26 22:03:52:178	WebLogic Server	Alexiauna		UScallcenter
14/05/26 22:03:52:178	BAM	yin2014		Indiacallcenter
14/05/26 22:03:52:178	ADF	Madara		CAcallcenter
14/05/26 22:03:52:178	Coherence	Kasriniv		Mexicocallcenter

Figure 18 – The Inbound Tweets Tactical View Page

Business View: Inbound Tweet TreeMap

This view is bound with the ***Tweeter Tree Map Query***. According to the ***TwoTier*** hierarchy (tweetHashtag2, Product), you can divide the panel into different areas. The first group is divided by tweetHashtag2. They are: CAcallcenter, Indiacallcenter, UScallcenter and Mexicocallcenter. For every tweetHashTag2, the new group is Product. They are: ADF, ODI, OAM, BAM, BPM, Weblogic Server, BPMN, and Coherence.

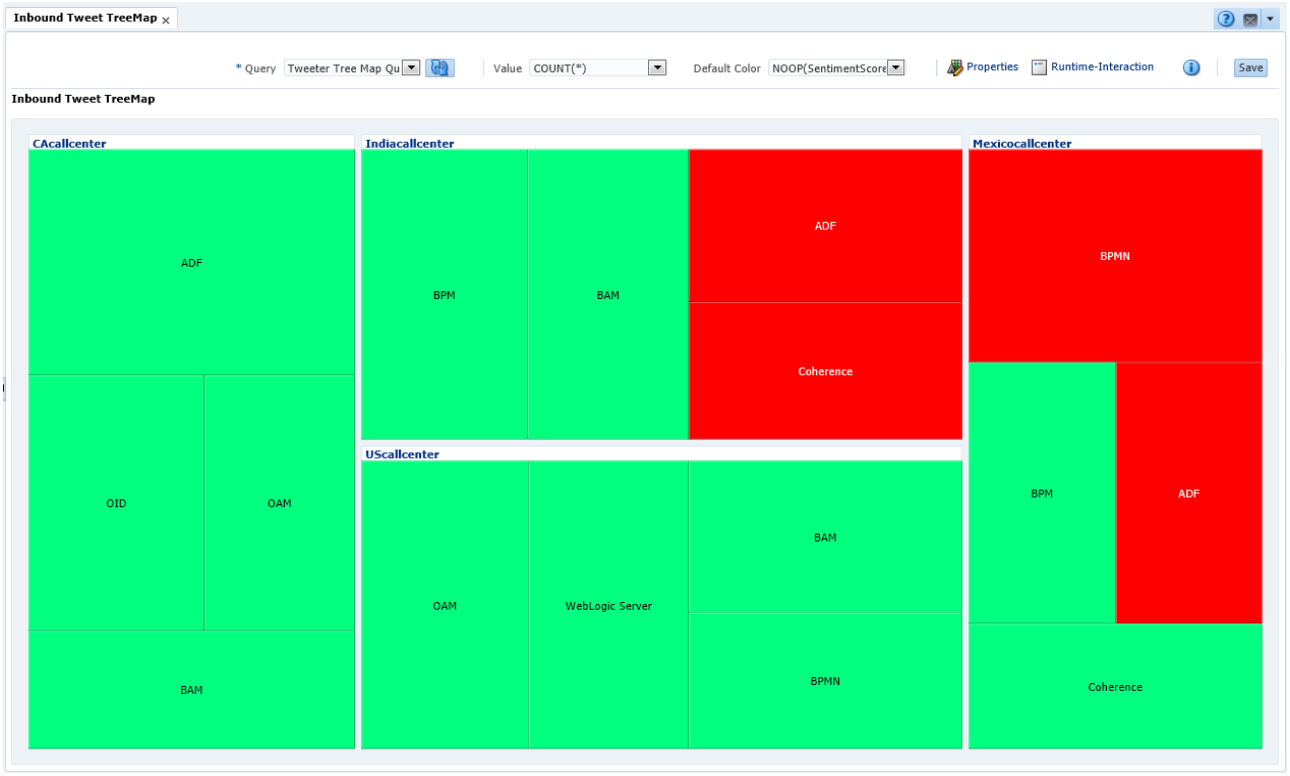


Figure 19 – The Inbound Tweet Tree Map

In order to display different areas with different colors, you must define Thresholds in the Properties tab.

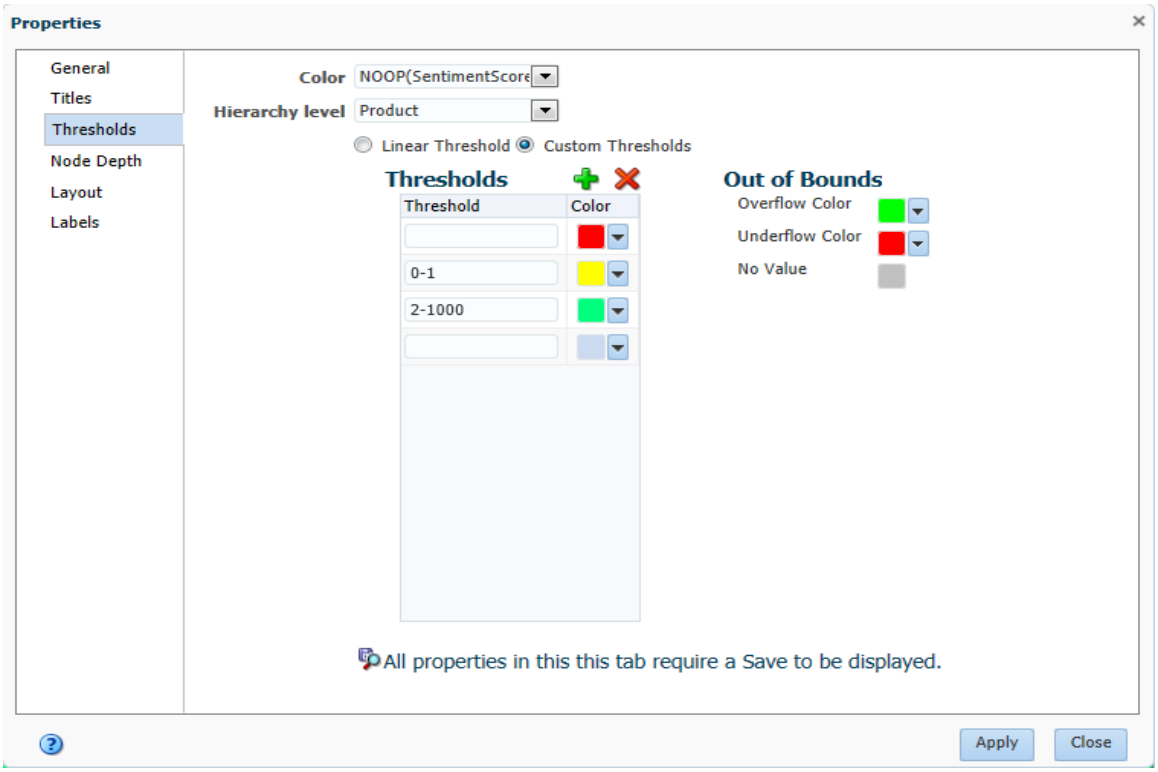


Figure 20 – Defining Thresholds in the Properties Tab

Business View: Outbound Tweets Active

This view is bound with the Outbound Tweeter query and is configured with the Runtime-Interaction option. Click Runtime-Interaction, enter into Active Data tab, select Turn this query into a continuous query, and set the Interval as 5 seconds. With this configuration, the query will be refreshed automatically, every 5 seconds.

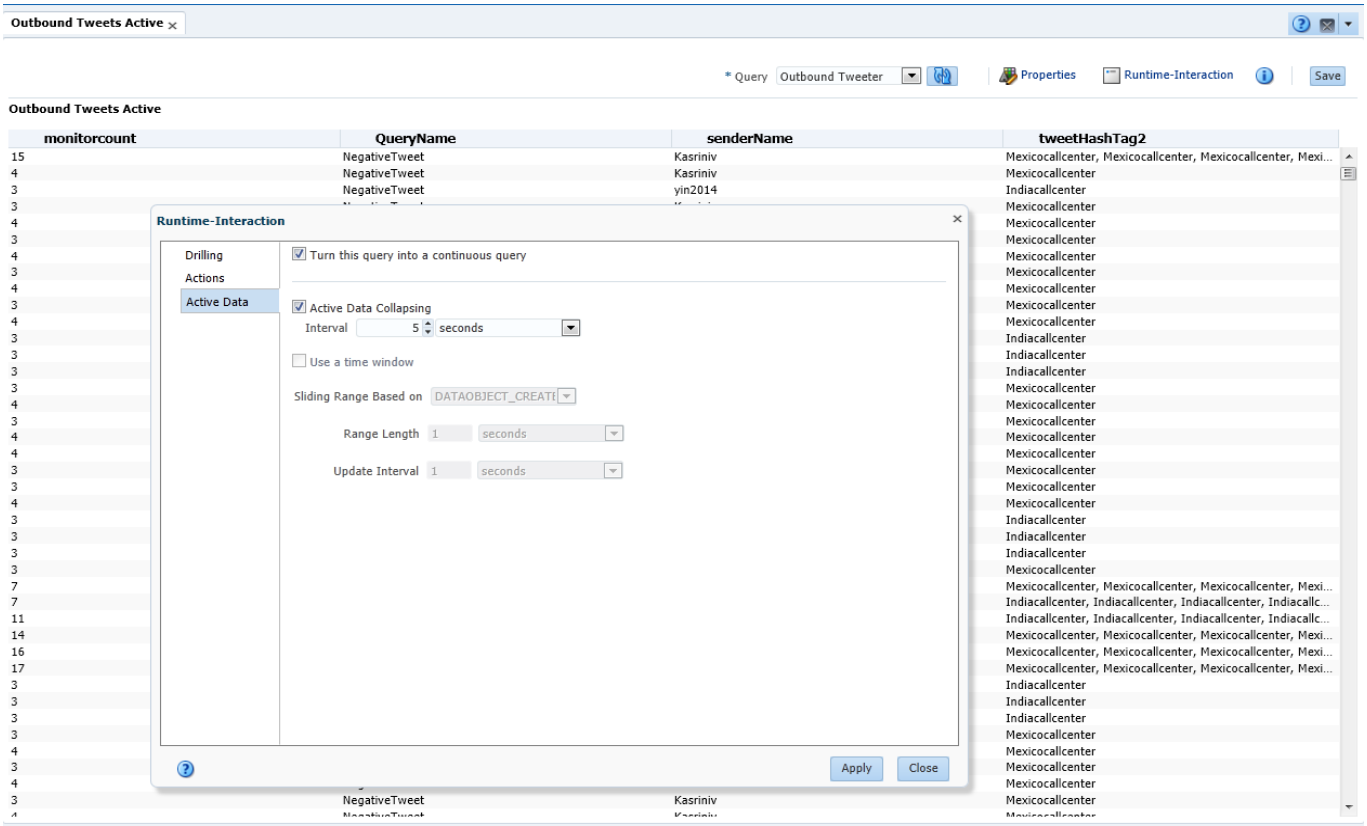


Figure 21 – Turning the Outbound Tweets Active Query into a Continuous Query

Business View: Outbound Tweets Tactical

This view is bound with the Outbound Tweeter query.

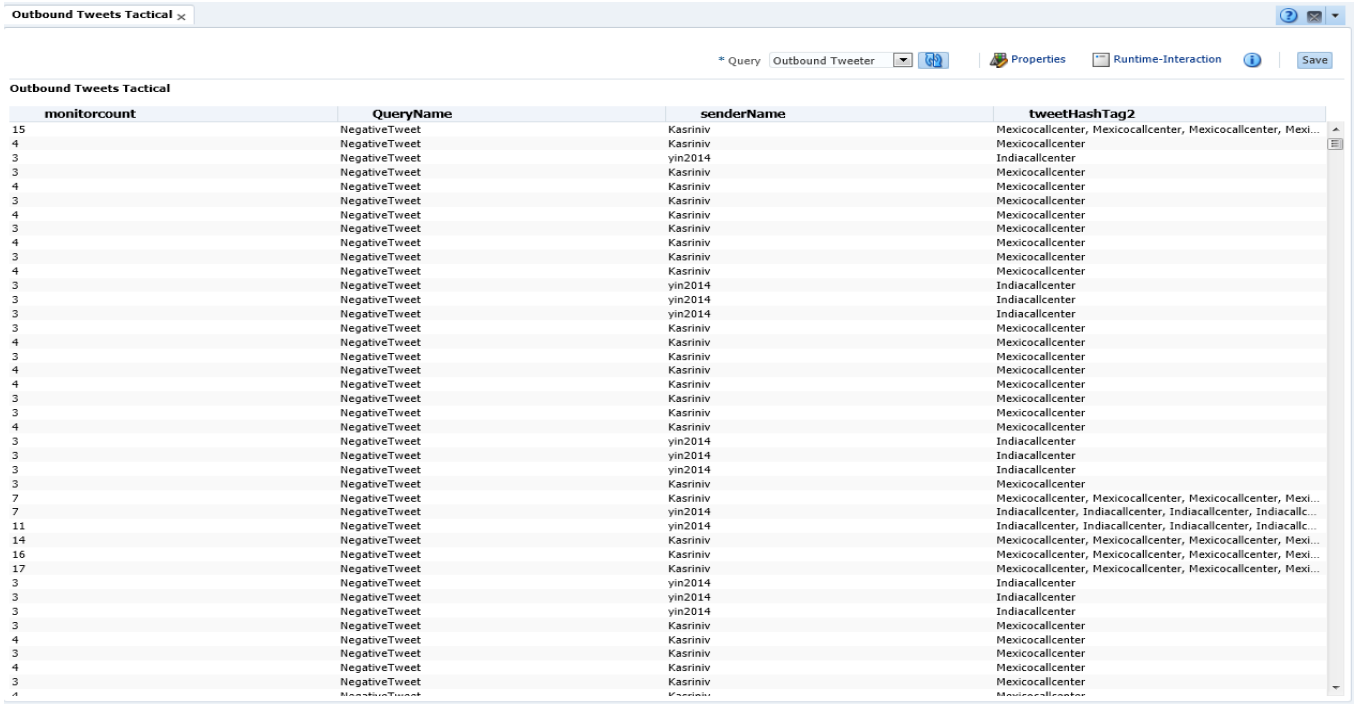


Figure 22 – The Outbound Tweets Tactical View Page

Alerts

NegativeTweet_Output_Alert

This alert is created during the <add action> in NegativeTweet query.

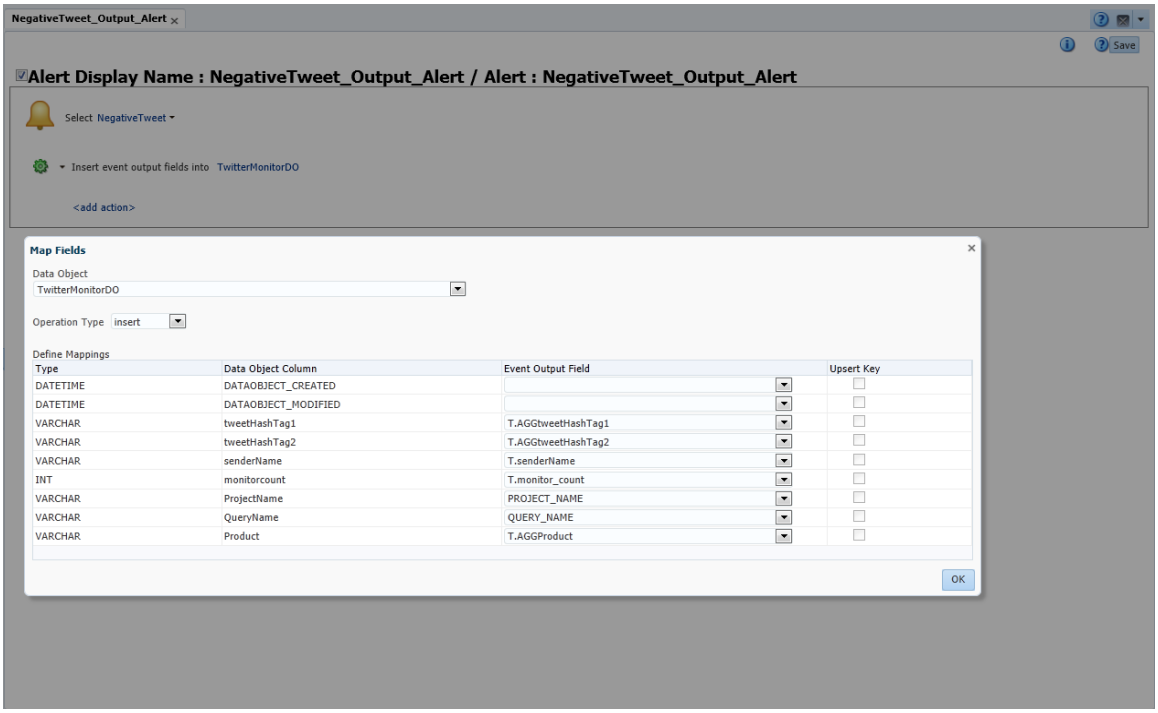


Figure 23 – NegativeTweet_Output_Alert

Postive Tweet_Output_Alert

This alert is created during the <add action> in Positive Tweet query UI. This alert uses the alert engine to write back the CQL result to one data object. You must configure the mapping between CQL output fields with the data object column.

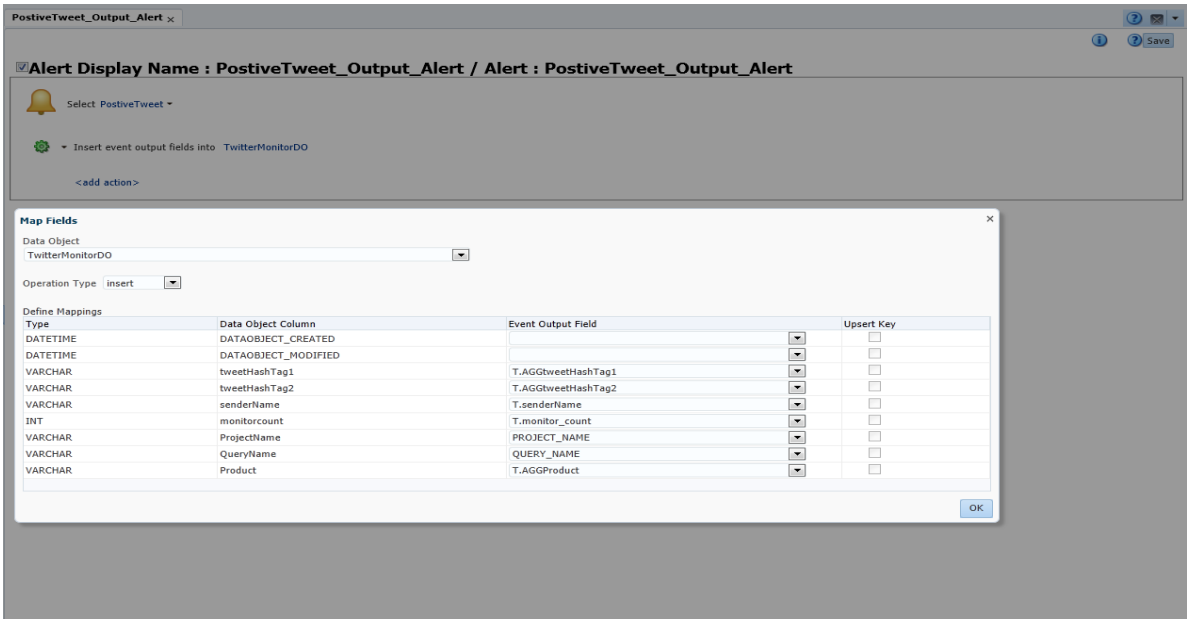


Figure 24 – Postive Tweet_Output_Alert

Dashboards

Oracle BAM Event Stream Analytics Active

This dashboard has three views: Inbound Tweet TreeMap on the top, Inbound Tweets Active at the bottom left, and Outbound Tweets Active at the bottom right. The tree map does not support active data services but the list views are active views.

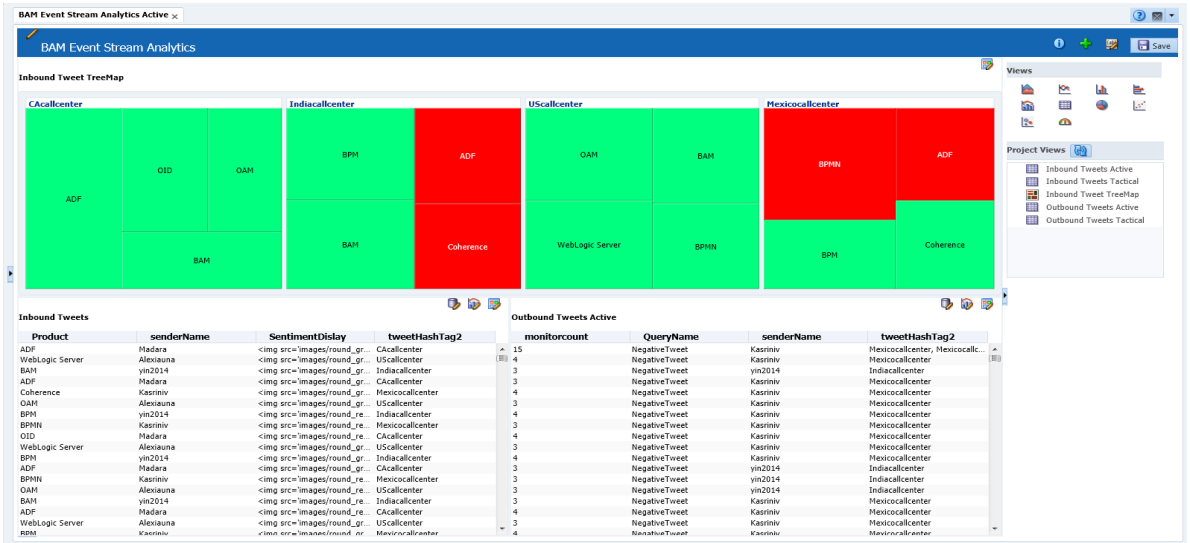


Figure 25 – The Oracle BAM Event Stream Analytics Active Dashboard

Oracle BAM Event Stream Analytics Tactical

This dashboard has three views: Inbound Tweet TreeMap at the top, Inbound Tweets Tactical at the bottom left, and Outbound Tweets Tactical at the bottom right. These two tactical views cannot be refreshed automatically. You must refresh the browser to see the latest chart.

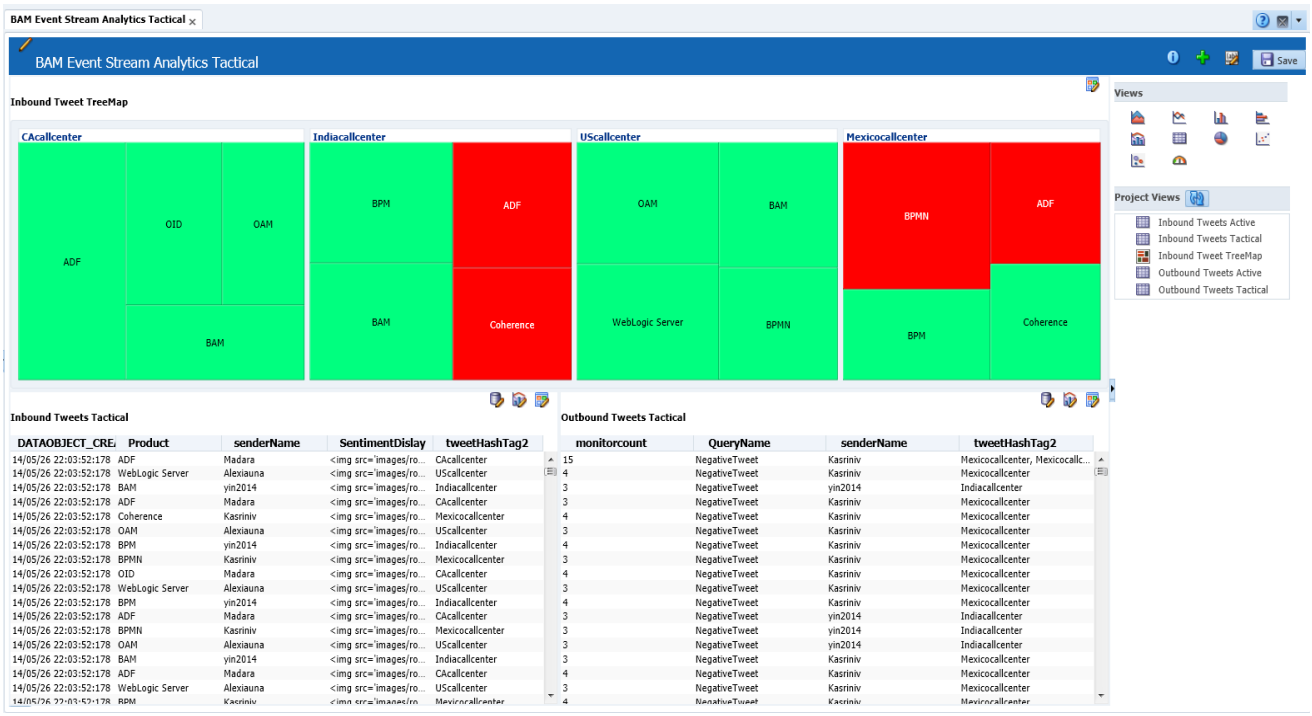


Figure 26 – The Oracle BAM Event Stream Analytics Tactical Dashboard

Creating a Project

This section shows you how to create a similar Oracle BAM Event Stream Analytics project.

Creating a Data Object

Open Oracle BAM composer, click the 'Administrator' link, choose the 'Data Object' node and click '+', and select 'Simple Data Object'. In this case project, we have selected the Archived and Continuous Query Type with RELATION, and have filled the data object name with oowdemo or TwitterMonitorDO. However, because TwitterMonitorDO is a write back data object, you must add the correct columns to adapt with the continuous query's output.

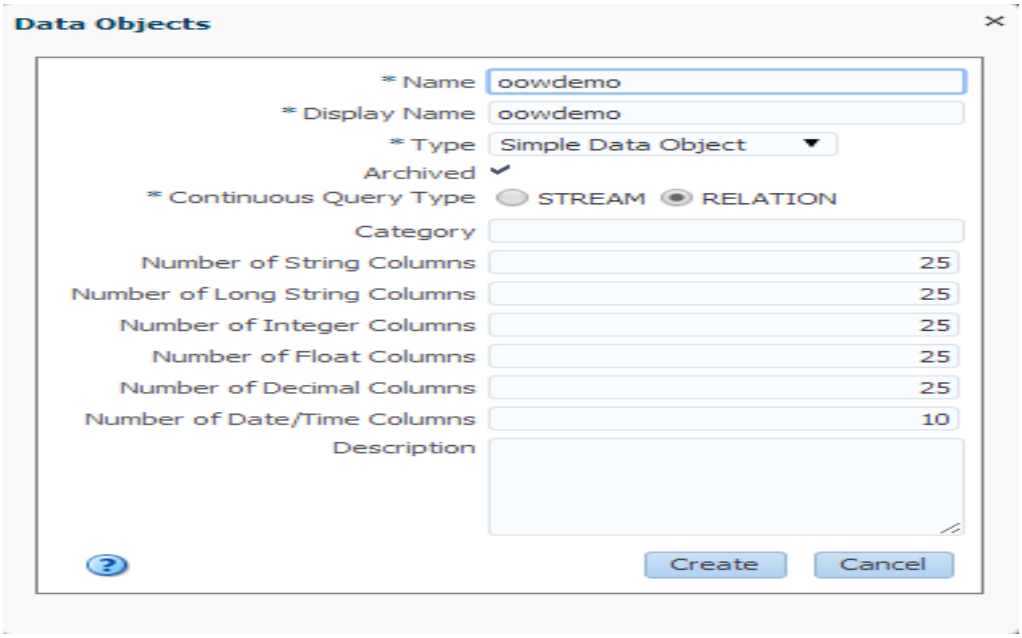


Figure 27 – Creating a Data Object

In order to define the write back data object, you must know the output fields of your continuous queries. You can create the continuous query before defining this data object. After saving the query, the output is clear. You can then create this data object according to the continuous query's output. For example, consider that you want to create a data object to write back the output of two continuous queries: Negative Tweet and Positive Tweet. First save the queries. This ensures that the following fields are listed in the output panel: monitor_count, AGGProduct, AGGsenderName, AGGtweetHashTag1, AGGtweetHashTag2 and senderName. You can now add appropriate columns in the data object and use one data object to write back many of queries' outputs. In order to distinguish all the data in the data object, you can create two columns in the data object: ProjectName and QueryName. For all continuous queries created by templates, these two fields will be output impliedly.

To create a new project, open Oracle BAM composer, click the 'Designer' link, and choose 'create' under the dropdown list to create a project. Fill out the Name and Display Name fields and click the 'Create' button, as shown in Figure 28.

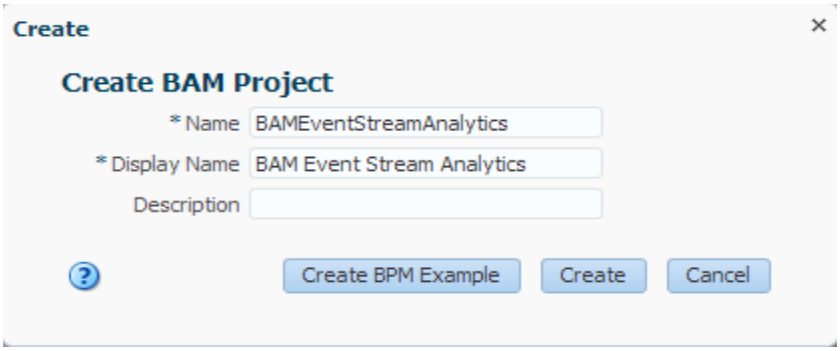


Figure 28 – Creating a new Oracle BAM Project

If you want to modify the project display name, you can choose ‘Rename’ under the dropdown list, edit the Display Name and click the ‘Save’ button, as shown in Figure 29.

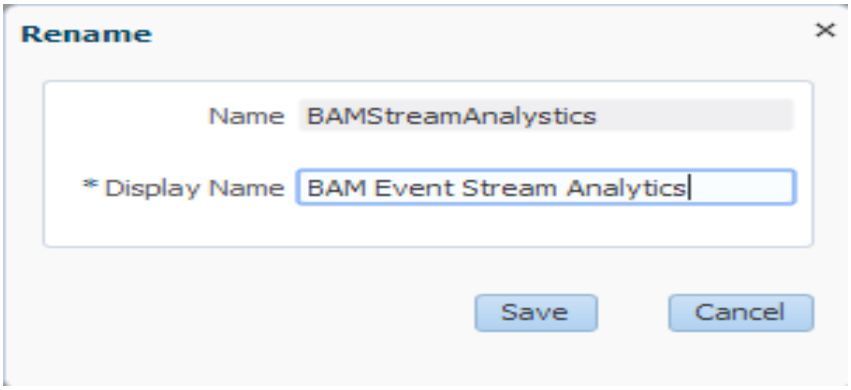


Figure 29 – Renaming a project’s display name.

After creating the data objects, you must add them to the project. Click ‘Design’ to open the project, then click the ‘Data Objects’ node. Click the ‘+’ button to open the ‘Data Objects’ window. Select the data objects you want to add, and click the ‘Add’ button to add them to the project.

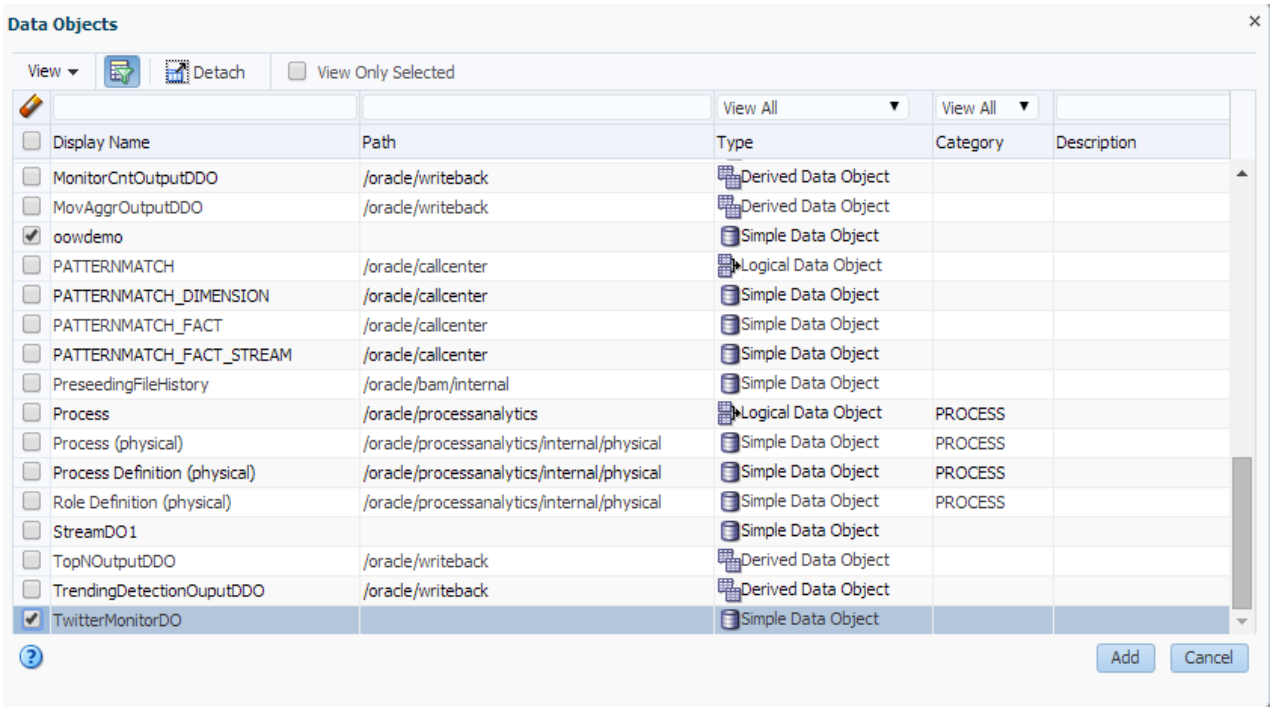


Figure 30 – Adding Data Objects to a Project

While creating a project, the project name and display name are validated by the following rules. To ensure that the project is created correctly, here are some suggested best practices:

1. The project name and display must be unique in the Oracle BAM system.
2. The project name must begin with capital or lower case letters only, i.e., a-z, or A-Z.
3. Only letters, numbers (0-9) and the underscore are allowed in the project name.

4. The project name length must be less than 128 characters.

Business Queries

This section provides information on the business queries supported in Oracle BAM. In addition to outlining their setups, you can also look at some of the suggested best practices while creating each of these business queries.

Flat SQL Query

Open a project, then choose ‘Business Queries’. Click ‘+’, and select ‘FLAT SQL Query’ to create a Flat SQL Query.

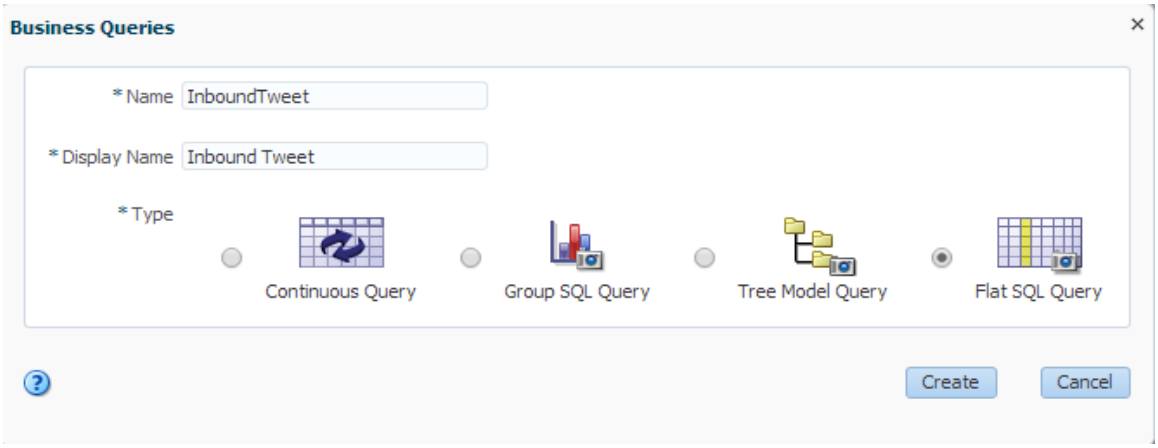


Figure 31 – Creating a Flat DQL Business Query

Open the query edit page, and fill the data object field, select the output fields from the list. You can add a filter to the query by clicking the ‘+’ button in Filter panel, which opens the filter editor.

Suggested best practice:

The Flat SQL Query is a plain query, and it cannot retrieve data which has the ‘grouped by’ clause. To retrieve data with groups or different dimensions, select **Type** with ‘**Group SQL Query’** in the Create Business Queries UI.

Continuous Query

To create a Continuous Query, open a project, choose ‘Business Queries’, click ‘+’, and select ‘Continuous Query’.

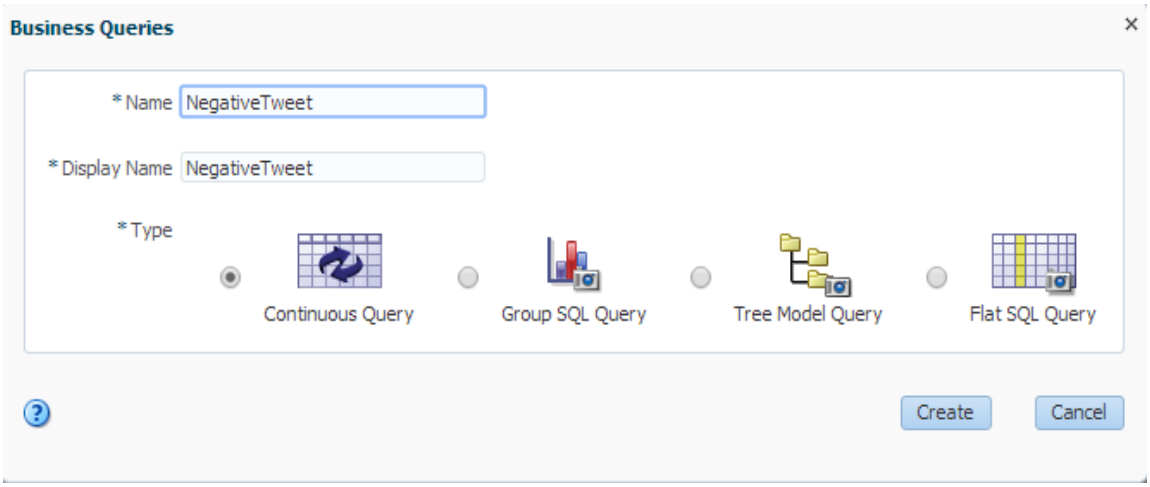


Figure 32 – Creating a Continuous Query

Open the continuous query edit page and select one of templates from the dropdown list. Fill all the required fields and save the query.

Click the Filter button in the UI to add the query's filter. If the filter has been configured with a sentimentscore less than 0, the query only retrieves and handles negative feedback data. It forms the query's 'where' clause in the CQL.

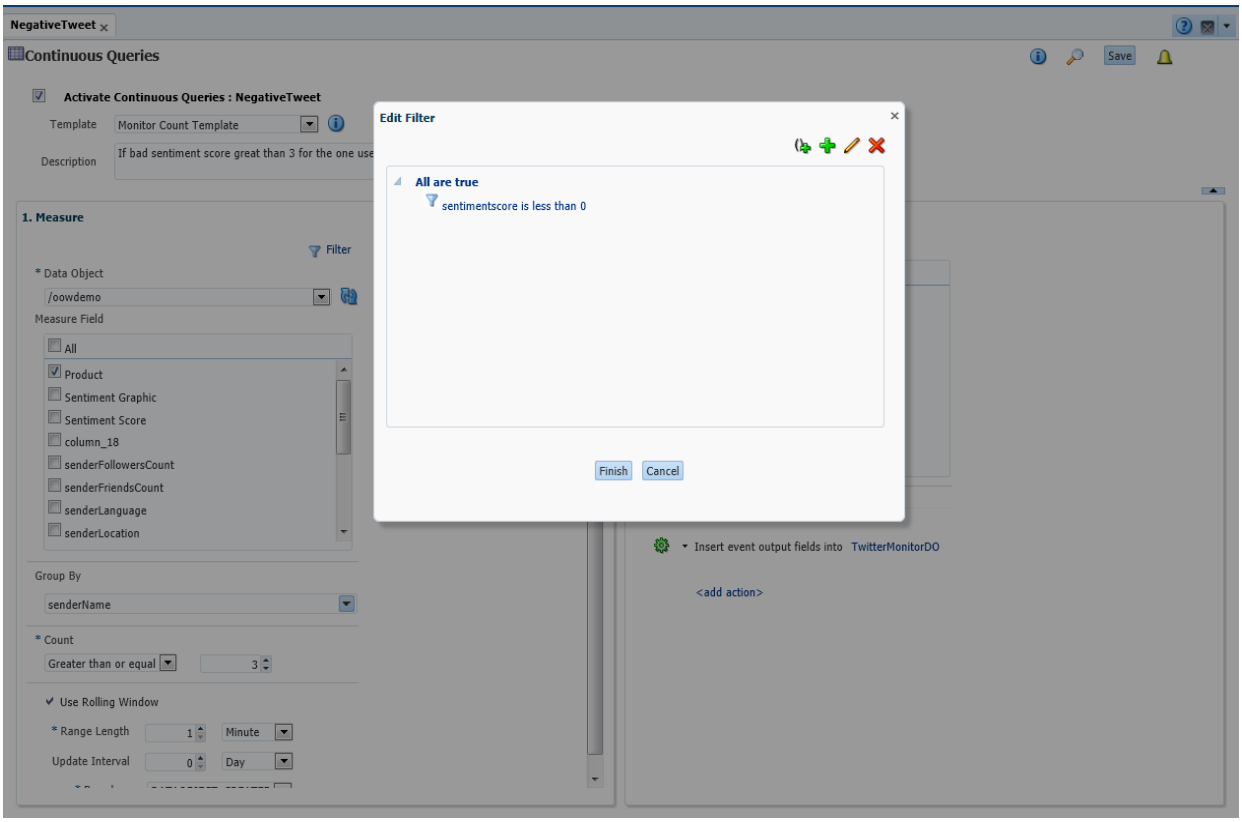


Figure 33 – Adding a Filter to the Query

To write back the query's output to a data object, click the <add action> link to create an action. Then select the 'Insert values into data object' option in the 'Select an Action' UI, as shown in Figure 34.

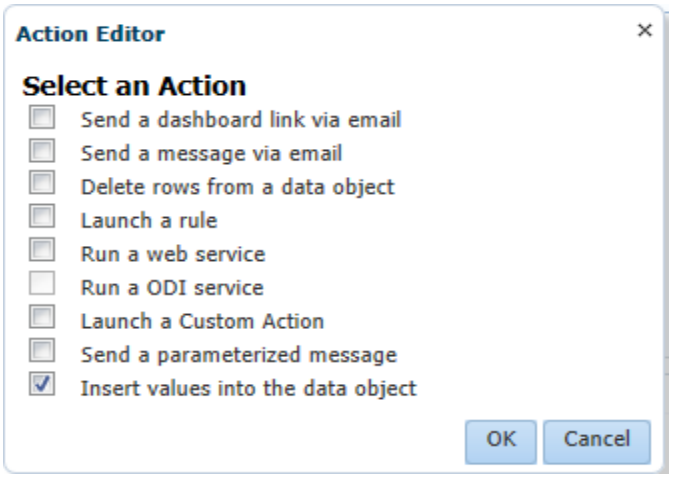


Figure 34 – Using the Action Editor to Add Values to a Data Object

Click the <select data object> link, and open the 'Map Fields' UI. Select TwitterMonitorDO as the data object, then select the correct Event Output Field Map with the data object's columns, as shown in Figure 35.

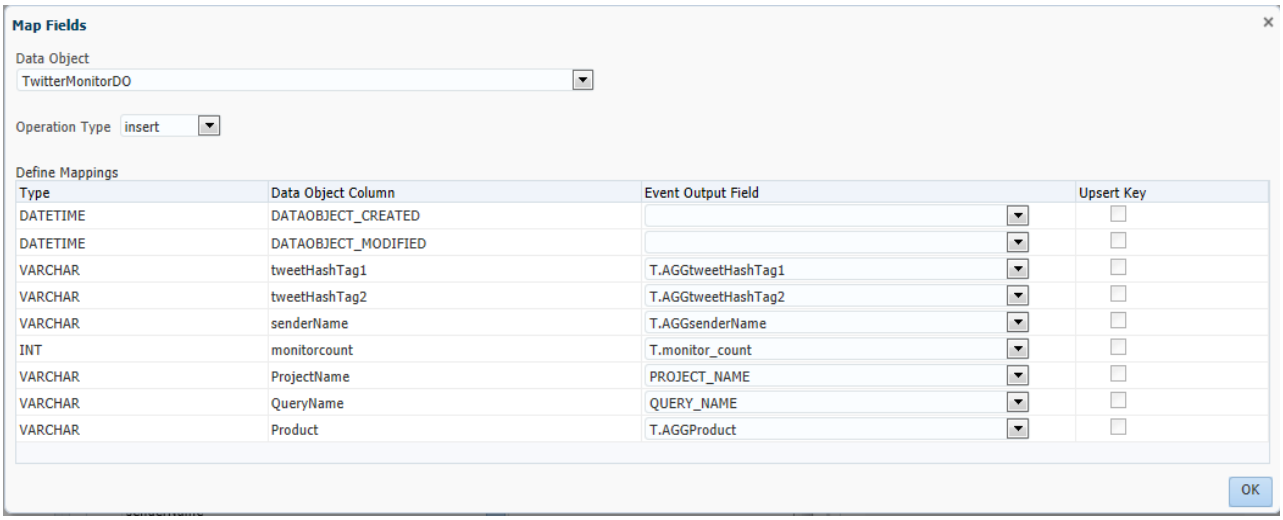


Figure 35 – Selecting the Appropriate Field Mappings for a Data Object

Click the button to save and start the alert action to 'listening' for the query's outputs. If the query has been triggered with the output, you can see output data from the data object.

Here is a suggested best practice:

The continuous query is different from the group and flat query in Oracle BAM 12c. In order to create a correct continuous query, you must know and understand your requirement. You must also understand which templates can be used for your cases. For the 7 template usages, refer to the Oracle BAM user guide documents. Select the correct template and fill all the required fields in the UI to get your query.

To validate whether the query meets your requirements, you can preview the CQL by clicking the icon at the top right corner of the UI. You can also populate some test data using Oracle BAM **loadgen** or insert data from the data object's data tab in the UI, and write back the query's output to one data object. Finally, you can check the output to see whether the query is correct.

Figure 36 shows the relation between a CQL and the UI. It helps you understand the template and your requirements.

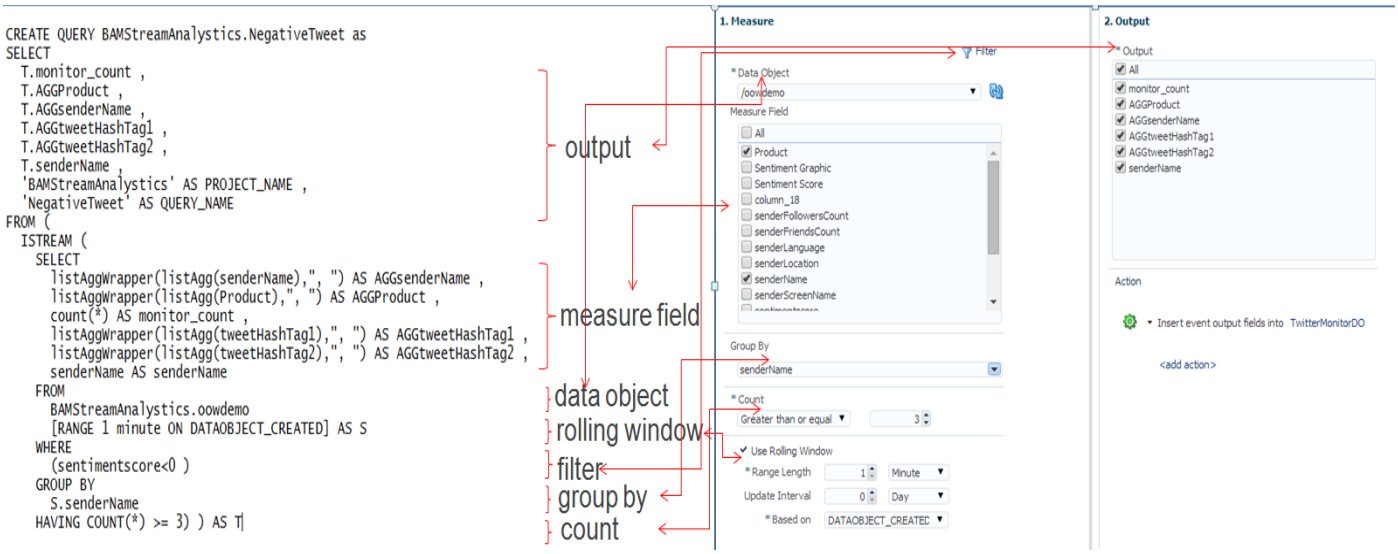


Figure 36 – A CQL Comparison with its Output in the UI

Tree Model Query

To create a tree model query, open a project, choose 'Business Queries', click '+', and select 'Tree Model Query'.

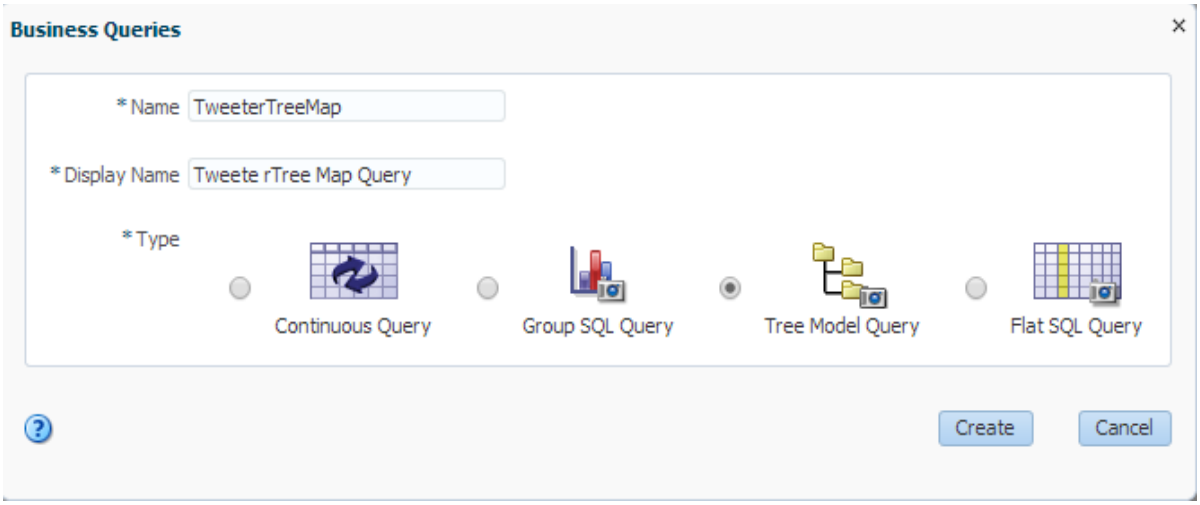


Figure 37 – Creating a Tree Model Query

Suggested best practice:

A Tree Model Query needs ‘hierarchy’ as a demission to divide data to different tree nodes or levels. If you want to create this query, you must define the Hierarchy property in the data object. You must choose columns that have some fixed value sets as Hierarchy selected columns.

Business Views

This section provides information on the different business views supported in Oracle BAM.

Table Type View

To create a Table Type View, open a project, choose ‘Business Views’, click ‘+’, select the List type and Table Category, and select List from View Types.

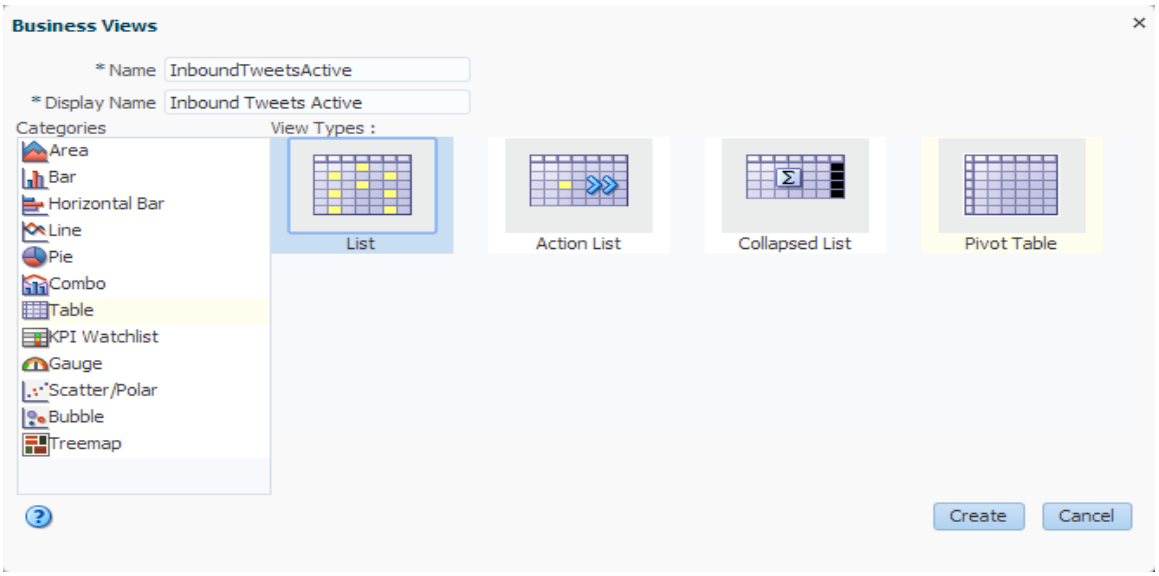


Figure 38 – Creating a Table Type View

Tree Map Type View

To create a Tree Map Type View, open a project, choose ‘Business Views’ and click ‘+’, select ‘Treemap’ in the Categories list and select ‘Treemap’ from View Types.

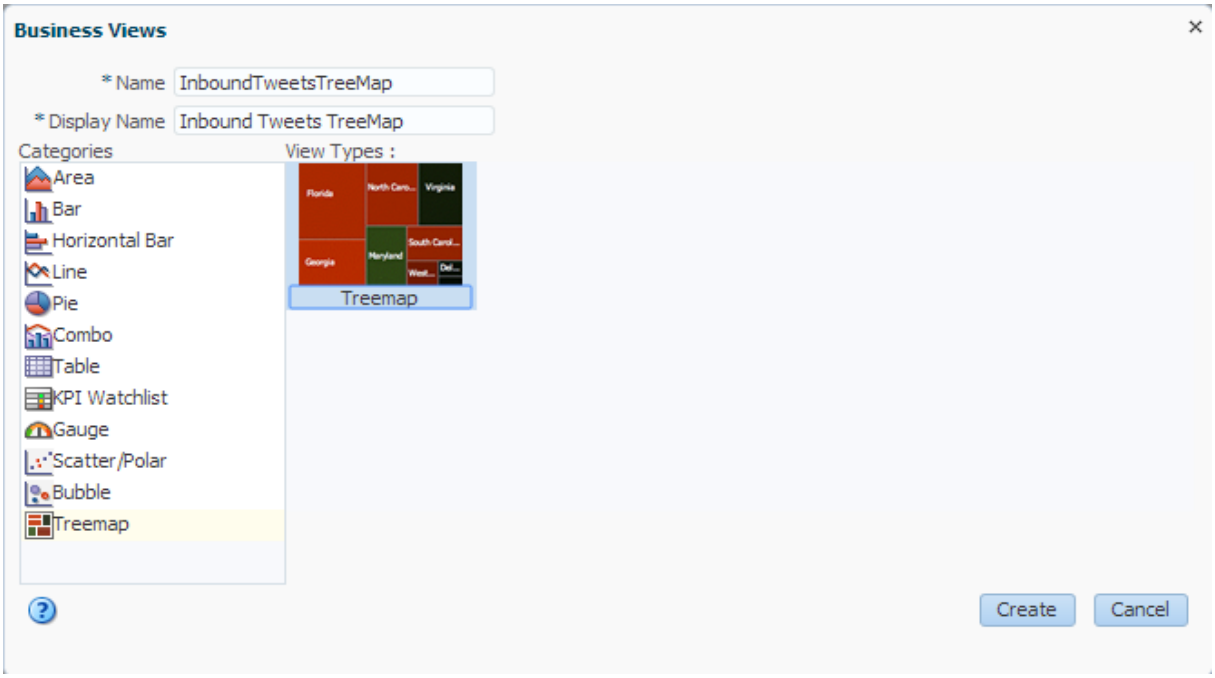


Figure 39 – Creating a Tree Map Type View

Dashboards

Open a project, then choose ‘Dashboards’ and click ‘+’, select ‘Type2’ from ‘Select the style template’ to create a dashboard. Open the dashboard edit page, and drag the specific business views from the appropriate list to the correct area.

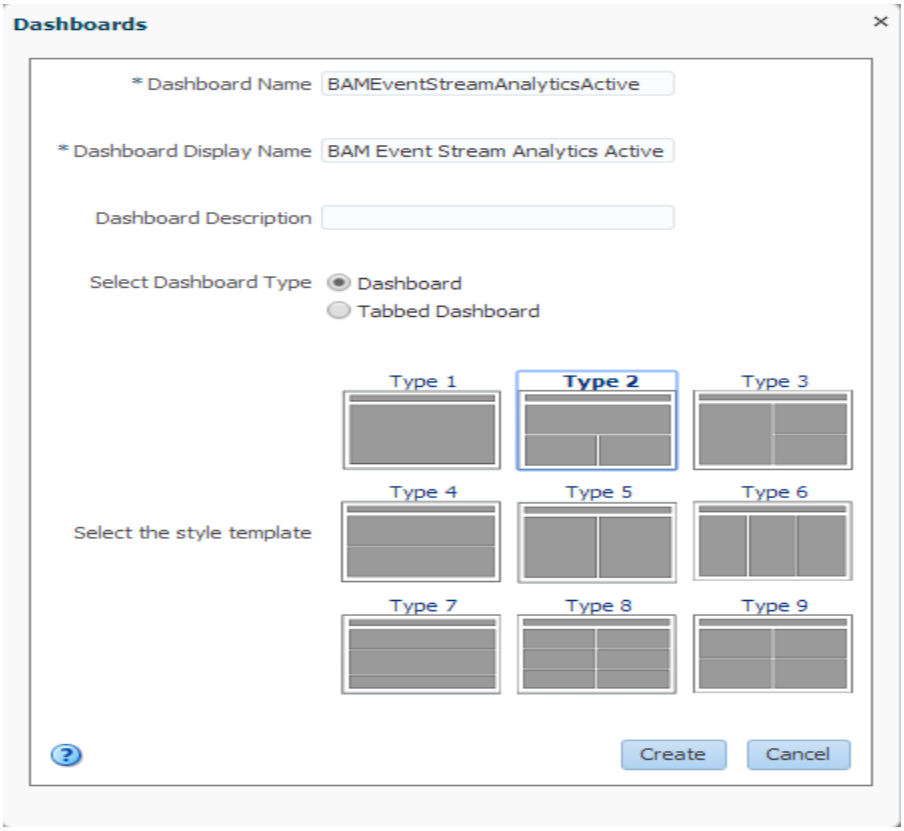


Figure 40 – Creating a Dashboard

Accessing the Dashboard

1. Open Oracle BAM dashboard:

a. 'BAM Event Stream Analytics Active' - Twitter dashboard for active data:

http://<hostname>:<port>/bam/composer/faces/proxypage?project=BAMStreamAnalytics&dashboard=BAMEventStreamAnalytics

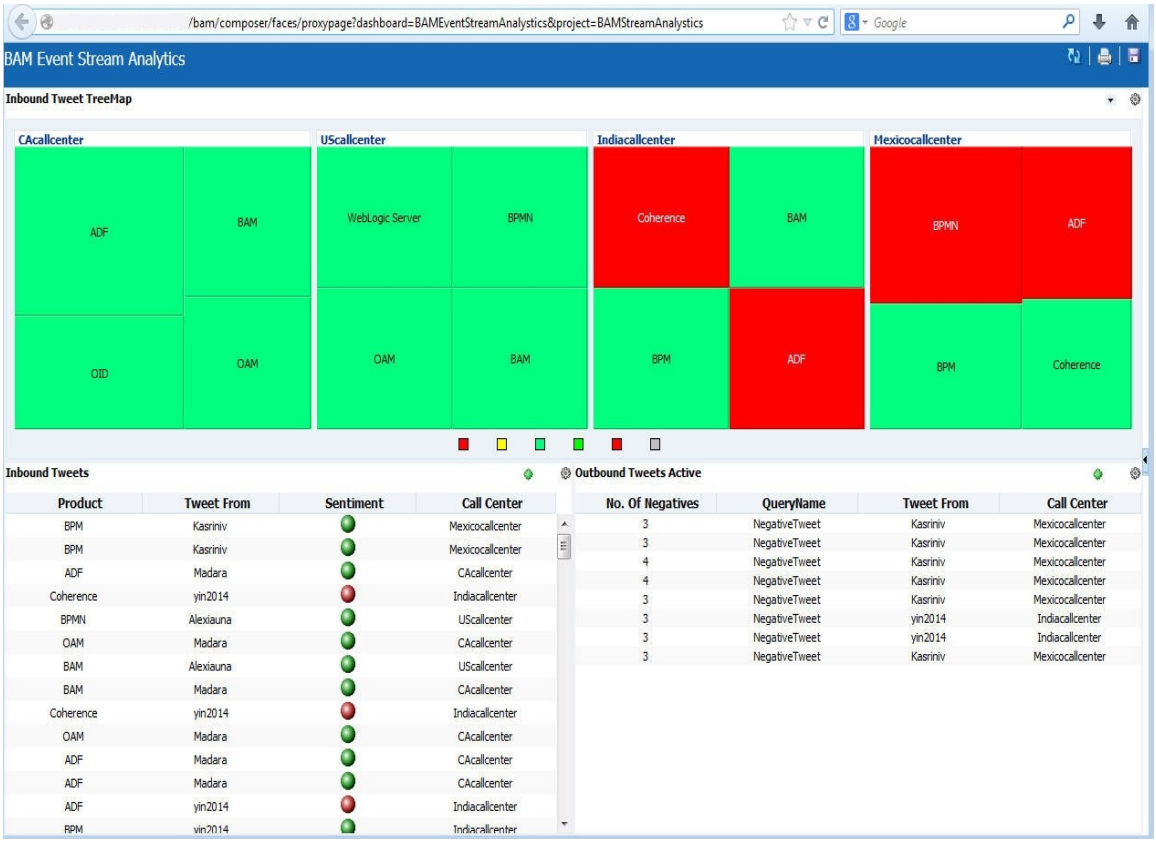


Figure 41 – Oracle BAM Event Stream Analytics Dashboard

b. 'BAM Event Stream Analytics Tactical' - Twitter dashboard for tactical data:

http://<hostname>:<port>/bam/composer/faces/proxypage?project=BAMStreamAnalytics&dashboard=BAMEventStreamAnalyticsTactical

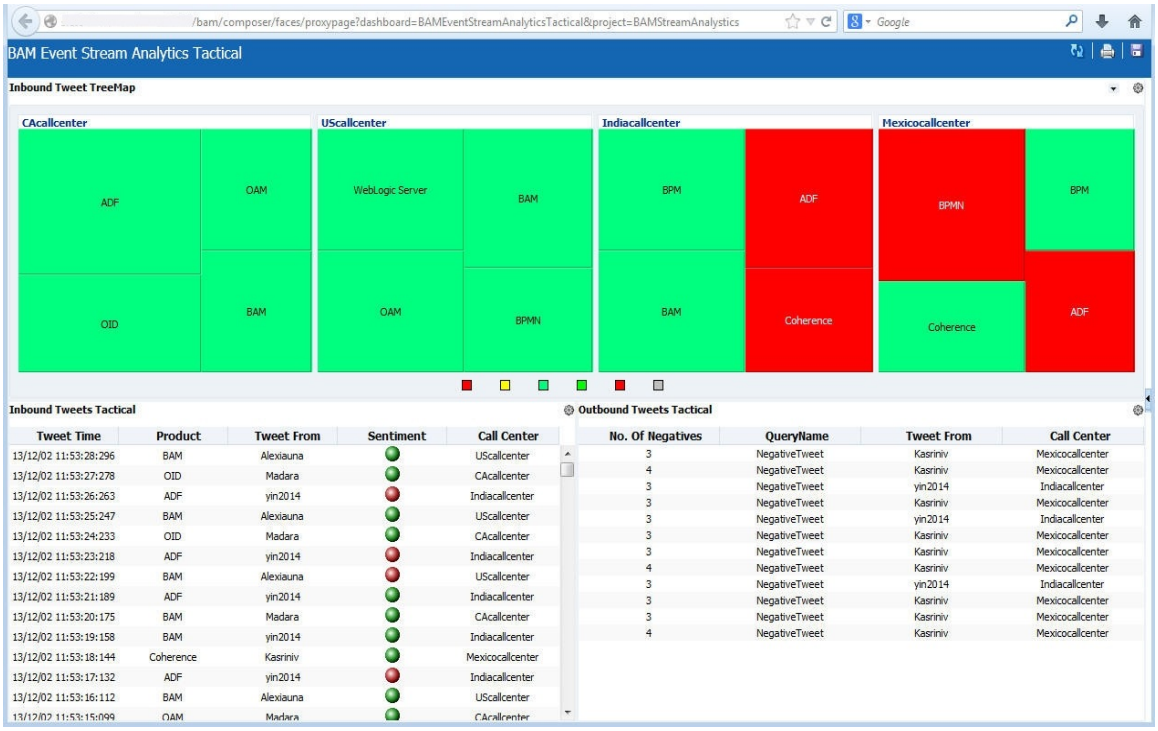


Figure 42 – Oracle BAM Event Stream Analytics Tactical

2. Execute startTwitterFeed.sh <wls password>

3. To stop the data population, execute stopTwitterFeed.sh

Note: Please refer to the following default values:

• WLS password: weblogic1

• Hostname: localhost

• Port: 7004

For populating data to the BAM data object, use the bamloadgen command. The following is a sample usage for bamloadgen to populate data to the data object oowdemo with 1 row per second frequency. In this command, we set the duration to 0, so it populates data forever. You can also specify a value for the duration parameter and allow data population for a limited time period.

$ORACLE_HOME/soa/bam /bin/bamloadgen -XMLFile data/oowdemo.xml -duration 0 -frequency 1 -username weblogic -password welcome1 -host localhost -port 7001

In order to populate the data using bamloadgen, you need to prepare an xml data file. The fields in the xml file need to be the same as the data object column names. This is a sample xml for populating data to the oowdemo data object. You can use the $currentTime token to populate a system at the current date. You can also set a fixed date with the date time field in the xml.

```
<dataObjectOperations>
   <operations>
      <insert dataobject="oowdemo">
         <insert-columns>
            <string value="#oowdemo #CAcallcenter is absolutely lovely pleasure working test 1" name="tweetText"/>
            <string value="web" name="tweetSource"/>
            <string value="oowdemo" name="tweetHashTag1"/>
            <string value="CAcallcenter" name="tweetHashTag2"/>
            <string value="Madara" name="senderName"/>
            <string value="Madara" name="senderScreenName"/>
            <integer value="20" name="senderFriendsCount"/>
            <integer value="2" name="senderFollowersCount"/>
            <string value="en" name="senderLanguage"/>
            <string value="testsource" name="senderLocation"/>
            <string value="BAM" name="Product"/>
            <integer value="2" name="sentimentscore"/>
            <date value="$currentTime" name="tweetDate"/>
         </insert-columns>
      </insert>
   </operations>
</dataObjectOperations>
```

Figure 43 – Populating an XML Data File

Troubleshooting

This section provides information on some scenarios which might require you to run troubleshooting. In this project, Tree Model Query and Continuous Query are used to demonstrate how the event sinks from one flow (the write back DO from the continuous queries) to be the event source of another flow (the data source for dashboard data population).

Project Import Issues

1. You cannot import the project using bamcommand

   The bamcommand script is put into the directory $ORACLE_HOME/soa/bam/bin. Firstly, ensure that you have set the correct $JAVA_HOME. Then, ensure that the Oracle BAM server is running and that the Oracle BAM Composer can be accessed.

2. You cannot see the project in the Oracle BAM Composer after completing the import.

   Ensure that you have set the correct host address and have used the correct and zip file. Then, ensure that you have used the correct '–mode' parameter. If you specify 'append', the project cannot be imported if it has existed on the Oracle BAM server. If you specify 'update', the project will override the previous instance on the Oracle BAM server.

Business Query Issues

1. Continuous queries are not active.

   The continuous query is not active automatically. If you want to activate (register and activate) the query, you need to select the 'Active' check box, then click the 'Save' button. If there isn't an exception while saving the query, it should be active successfully. You can close and reopen the query to check whether the check box is still active. Another way to check the continuous query status is by going to 'Continuous Queries Monitoring' in the 'Administrator' mode, and selecting your project from the Project dropdown list. All the continuous queries in the project are listed on the page. You can activate or deactivate the query by using the toggle button.

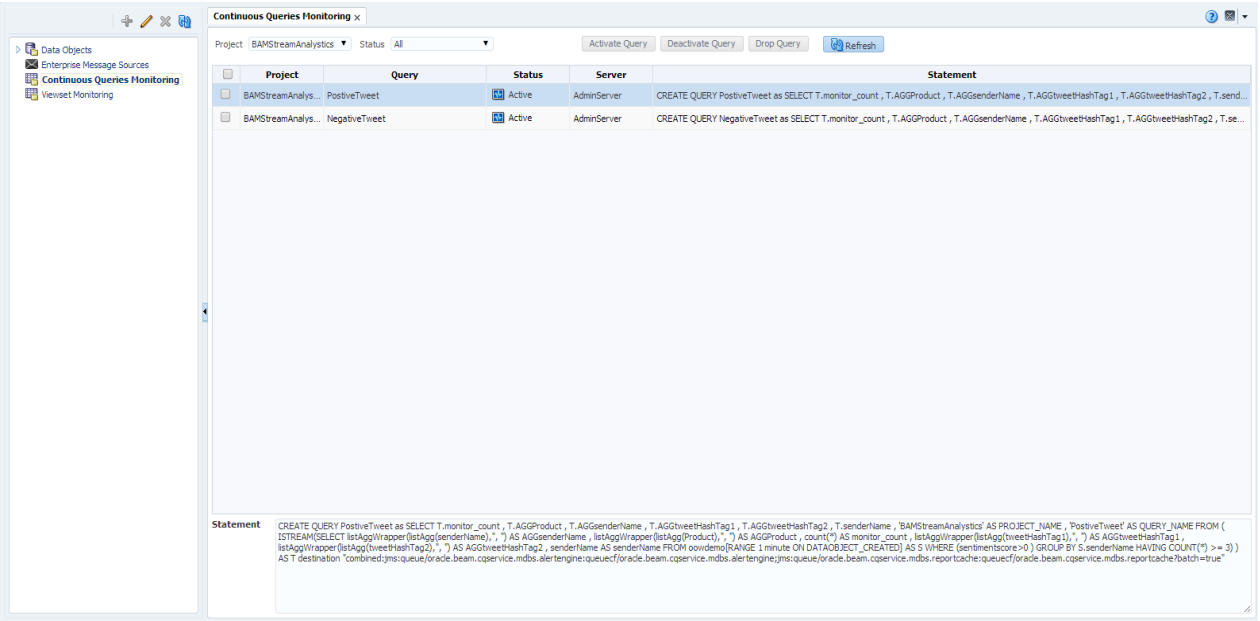


Figure 44 – Making a Continuous Query Active

2. Tree model query cannot be created.

   The Tree model query is a hierarchy group query. Ensure that you have defined some hierarchical relation in the data object and that the relation is valid. If the relation cannot group the data within each part, the query is invalid.

3. Cannot retrieve correct data from a Flat SQL Query

   For the all the queries in Oracle BAM 12c, the data can be configured with row level security filter. This means that users cannot see all the data in a specific data object even if they can access it. Ensure that your query is correct and retrieves data from the data object. Then, review your query's SQL from the UI. You can check whether there is an additional 'where' clause appended to the query's SQL, in which case, you must obtain the runtime security filter permission for your current user role from the administrator. Alternatively, you can modify the security filter for the role, or try to login with another user role's credentials.

Business View Issues

1. Cannot see the data in the business view being refreshed automatically.

   Ensure that the business view is configured with 'Active Data'. Only the query has to be configured with the active data service; the data into the view can be refreshed automatically. Then, go to 'Continuous Queries Monitoring' in 'Administrator'. Select your project from the Project dropdown list. All the business views with active data service in the project are listed on the page. You can check whether the view's active data is running.

2. Cannot create tree map views

   If you have created a tree map view, you must set the correct 'Value' and 'Default Color'. Ensure that the 'Hierarchy Level' and the Thresholds have been set to correct values.

After saving the query, if there isn't data that can be retrieved with the query, the view must be blank. The Tree map must now be working and retrieving data.

Dashboard Issues

1. Cannot open the dashboard

   If you want to view the dashboard, click the 'Open' menu in the dashboard node. If you cannot see a new popup window, ensure that the browser has not blocked it. You can set browser permissions to allow the window to open pop ups for a particular website.

2. Cannot see the data being refreshed automatically

   If you have configured some views with active data, the view should be refreshed automatically. If not, ensure that the query can retrieve the data correctly, and that the active data is running on the view. Also, check if the session hasn't expired.

Data Population Issues

1. Cannot populate data into the specific data object

   The bamloadgen shell script must be put into the directory:

   $ORACLE_HOME/soa/bam/bin. Ensure that you have set the correct $JAVA_HOME. Then, ensure that the Oracle BAM server is running and that the Oracle BAM Composer can be accessed. Check if the data object name in the xml and the data formatter are correct.

2. Cannot set the correct date for the data time field

   The data time in the xml file must be the correct formatter if you don't want to use the $currentTime token. bamloadgen uses "yyyy-MM-dd HH:mm:ss.SSS z". Ensure that the appropriate date formatter has been used in your date value.

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

blogs.oracle.com/oracle
facebook.com/oracle
twitter.com/oracle
oracle.com

Hardware and Software, Engineered to Work Together