ORACLE'

FUSION MIDDLEWARE BUSINESS ACTIVITY MONITORING

PATTERN MATCHING CAPABILITIES IN ORACLE BAM 12.1.3



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

Disclaimer	1
Introduction	3
Pattern Matching Use Case Scenario:	3
Preparing the Project	4
Importing the Project	5
Running a PatternMatch Dashboard	9
To run a PatternMatch dashboard, you must run through the following steps:	9
Creating Continuous Queries Using CQL Templates	10
Data Objects	10
Suggested Best Practice	12
Creating the PatternMatch Project	13
Continuous queries	14
Designing a KPI Alert Template	15
Designing a Trending Design Template	18
Designing a Missing Event Template	21
Designing a Monitor Count Template	25
Designing a Moving Aggregation Template	28
Designing a Top N Template	30
Designing a Duplicate Detection Template	33
Continuous Queries Monitoring	36
Designing a Fired KPI Alert Events Query	36
Designing Fired KPI Alert Output Events	37
Understanding Dashboards	38
PatternMatch Dashboard	38

	StrPatternMatch Dashboard	40
	You can create a strPatternMatch Dashboard using the procedure outlined for	
	creating the PatternMatch Dashboard, and add the following views to it.	40
Tr	oubleshooting	41
Be	est Practices	42

Introduction

This whitepaper provides information on defining continuous queries and building dashboards to actively monitor outputs using advanced Pattern Matching capabilities provided by Oracle BAM 12c. Continues Queries differ from traditional queries in that once a query registers, it runs until it is de-registered. This makes continuous queries ideal for addressing real time monitoring with temporal (the last moving 10 minutes, for example) and PatternMatch (e.g. trending, missing event, etc.) capabilities.

Oracle BAM has seven most commonly used Out Of the Box (OOTB) Continuous Query templates, which can be used to define complex PatternMatch scenarios without writing any CQL statements. These templates are parameters-driven and CQL are generated and registered to the BAM server, based on specific parameters. When a pattern is detected by the CQ, you can associate an alert with it, which triggers a list of actions. For example, the outputs from these queries can be daisy-chained to drive the dashboards so you can continuously monitor the number of alerts generated by each of these queries.

For more information on Oracle CQL, see: Understanding Oracle CQL.

Oracle BAM uses OOTB templates for creating CQs, and supports both stream and relational data objects. Stream Data is a type of data which always generates a 'plus' event; for example, GPS data or stock price data that shows the current value of the event. This means, you can insert data into a DO, but you cannot modify or delete them. Archived Relation Data is like traditional relational data where CRUD operations can be performed. Archived Relation obtains the initial state of the query from the database and then performs incremental computations on top of it. Oracle BAM supports the following templates. Note that the Duplicate Detection Template' can only support a stream DO.

- 1. KPI Alert Template.
- 2. Duplicate Detection Template (supports only Stream DOs)
- 3. Trending Detection Template
- 4. Monitor Count Template
- 5. Moving Aggregation Template
- 6. Missing Event Template
- 7. Top N Template

As an example, you can use the KPI Alert Template to generate a query which fires each time the KPI values cross a certain threshold. You can choose to display this using a line chart which shows alert counts for each minute, or any other time value, for the last 10 minutes, or any other time range.

Pattern Matching Use Case Scenario:

Consider a situation where you are monitoring product sales efficiency for a company, using metadata from their in-house call center to detect emerging patterns or trends that may hamper or enhance your sales process. This requires that you:

- ✓ Detect when the average call wait time for all the calls 'closed' in the last 2 minutes is greater than a predefined threshold.
- ✓ Detect if the callProcessingTime is increasing by more than 10% for two consecutive intervals.
- ✓ Detect when a call "Suspend" action is not directly followed by "Resume".
- ✓ Identify if more than one event is detected in the past 2 minutes for the same customerLocationId and productId pair.
- ✓ Continuously monitor the moving average of callProcessingtime in the last 2 minutes for each productId.
- ✓ Continuously update the top 5 products with maximum call Processing Time in the last 2 minutes.
- ✓ Detect if more than one event arrives within 2 minutes of each other with the same CustomerLocationId, productId and callStatus.

Apart from this, you must continuously monitor the count of these anomalies within the past 10 minutes. This document explains how the OOTB templates available in BAM 12c are used to detect these patterns or trends and how this information is displayed on a dashboard so you can track patterns in real time.

Preparing the Project

To start, you must ensure that the project environment is set up correctly as follows.

- 1. Copy the "samples" directory to ORACLE_HOME/soa/bam
- 2. Update ../common/setEnv.sh with proper info.
- Update ../../bin/BAMCommandConfig.xml and add the following parameters (make sure to replace **password** with valid password):
 <password>**password**</password>
- 4. Set environment variable JAVA_HOME.
- 5. If there's a previous installed project, clear all continuous queries. If you are importing a PatternMatch project for the first time, you can ignore these steps:
 - I. Open BAM composer: <u>http://<hostname>:<port>/bam/composer</u>
 - II. Choose 'Administrator'
 - III. Click on 'Continuous Queries Monitoring'
 - IV. Under the 'Project' dropdown list, select 'PatternMatch'
 - V. Check on the 'select all' checkbox
 - VI. Click on 'Deactivate Query'
 - VII. Click on 'Drop Query'

Importing the Project

To import the project, execute: importPatternMatch.sh from ORACLE_HOME/soa/bam/samples/bam-103-pattern-match. Then, check if all project artifacts are ordered correctly, as shown in Figure 1.

PatternMatch 💌
A 🔁 Data Ohianta
Data Objects Image: A constraint of the second s
⊿ 🛅 callcenter
PATTERNMATCH
⊿ 🚞 writeback
BasedWriteBackDO
MonitorCntOutputDDO
TopNOutputDDO
IrendingDetectionOuputDDO
AverageCallWaitingTimeGreateThanThreshold
FiredKPIAlertEventsQuery
EiredTopNEventsQuery
EiredTrendingEventsQuery
StrAvgCallWaitTimeGreaterThanThreshold
strFiredKPIAlertEventsQuery
strFiredMissingEventsQuery
strFiredMonitorCountEventsQuery
strFiredMovingAggregationEventsQuery
CallProcessingTimeTrendingUp 10Percent
MissingResumeCallDetection
MonitorDuplicatedEvents
StrCallProcessingTimeTrendUp 10PC
StrDuplicateDetection
StrMisResumCallDetection
StrMonitDupEvents
StrTop5MaxCallProcessTime
Top 5MaximumCallProcessingTime
Business Views
 FiredKpiAlertOutputEvents
FiredMissingEvents
FiredMonitorCountEvents
FiredMovingAggregationEvents
 FiredTrendingEvents
strFiredDuplicateDetectionEvents
strFiredMovingAggregationEvents
strFiredTopNEvents
StrFiredTrendingEvents
e strPatternMatchDashboard
Alerts
AverageCallWaitingTimeGreateThanThreshold_Output_Alert
CallProcessingTimeTrendingUp10Percent_Output_Alert MissingResumeCallDetection_Output_Alert
MovingAverageForCallProcessingTime_Output_Alert
StrAvgCallWaitTimeGreaterThanThreshold_Output_Alert
StrCallProcessingTimeTrendUp10PC_Output_Alert
StrMisResumCallDetection_Output_Alert
StrMonitDupEvents_Output_Alert
StrMovinAvgCallProcessTime_Output_Alert
StrTop5MaxCallProcessTime_Output_Alert
I oppmaximumCaliProcessingTime_Output_Alert

Figure 1 – Example PatternMatch Project

The following Data objects must be present under Data Objects Oracle \rightarrow callcenter

- 1. PATTERNMATCH_FACT (Archived Relation)
- 2. PATTERNMATCH_DIMENSION (Archived Relation)
- 3. PATTERNMATCH (Logical DO a combination of PATTERNMATCH_FACT and PATTERNMATCH_DIMENSION)
- 4. PATTERNMATCH_FACT_STREAM (Stream DO)

The project also contains the following pre-seeded DOs under oracle \rightarrow writeback

- 1. BasedWriteBackDO
- 2. DupDetectionOutputDDO
- 3. KPIAlertOutputDDO
- 4. MissingEventOutputDDO
- 5. MonitorCntOutputDDO
- 6. MovAggrOutputDDO
- 7. TopNOutputDDO
- 8. TrendingDetectionOuputDDO

In the PatternMatch project, there are six continuous queries over logical DOs and seven continuous queries over stream DOs.

- 1. KPI Alert Template
 - 1. AverageCallWaitingTimeGreateThanThreshold over the logical DO PATTERNMATCH
 - StrAvgCallWaitTimeGreaterThanThreshold over stream DO PATTERNMATCH_FACT_STREAM Both these queries have an alert action configured to write back the output to KPIAlertOutputDDO.
- 2. Duplicate Detection Template (supports only Stream DOs)
 - StrDuplicateDetection over stream DO PATTERNMATCH_FACT_STREAM This query has an alert action configured to write back the output to DupDetectionOutputDDO
- 3. Trending Detection Template
 - 1. CallProcessingTimeTrendingUp10Percent over the logical DO PATTERNMATCH
 - StrCallProcessingTimeTrendUp10PC over stream DO PATTERNMATCH_FACT_STREAM Both these queries have an alert action configured to write back the output to TrendingDetectionOuputDDO
- 4. Monitor Count Template
 - 1. MonitorDuplicatedEvents over the logical DO PATTERNMATCH
 - StrMonitDupEvents over stream DO PATTERNMATCH_FACT_STREAM Both these queries have an alert action configured to write back the output to MonitorCntOutputDDO

- 5. Moving Aggregation Template
 - 1. MovingAverageForCallProcessingTime over the logical DO PATTERNMATCH
 - StrMovinAvgCallProcessTime over stream DO PATTERNMATCH_FACT_STREAM Both these queries have an alert action configured to write back the output to MovAggrOutputDDO
- 6. Missing Event Template
 - 1. MissingResumeCallDetection over the logical DO PATTERNMATCH
 - 2. StrMisResumCallDetection over stream DO PATTERNMATCH_FACT_STREAM Both these queries have an alert action configured to write back the output to MissingEventOutputDDO
- 7. Top N template
 - 1. Top5MaximumCallProcessingTime over the logical DO PATTERNMATCH
 - StrTop5MaxCallProcessTime over stream DO PATTERNMATCH_FACT_STREAM Both these queries have an alert action configured to write back the output to TopNOutputDDO

The following Group Queries are defined over writeback DOs so they can use the output of the continuous queries as event sources:

- 1. FiredKPIAlertEventsQuery & strFiredKPIAlertEventsQuery over KPIAlertOutputDDO
- 2. strFiredDupDtectionEventsQuery over DupDetectionOutputDDO over DupDetectionOutputDDO
- 3. FiredTrendingEventsQuery and strFiredTrendingEventsQuery over TrendingDetectionOuputDDO
- 4. FiredMonitorCountEventsQuery and strFiredMonitorCountEventsQuery over MonitorCntOutputDDO
- 5. FiredMovingAggregationEventsQuery and strFiredMovingAggregationEventsQuery over MovAggrOutputDDO
- 6. FiredMissingEventsQuery and strFiredMissingEventsQuery over MissingEventOutputDDO
- 7. FiredTopNEventsQuery and strFiredTopNEventsQuery over TopNOutputDDO

Ensure that the following business views are present:

- 1. FiredKpiAlertOutputEvents & strFiredKpiAlertOutputEvents
- 1. strFiredDuplicateDetectionEvents
- 2. FiredTrendingEvents& strFiredTrendingEvents
- 3. FiredMonitorCountEvents & strFiredMonitorCountEvents

- 4. FiredMovingAggregationEvents & strFiredMovingAggregationEvents
- 5. FiredMissingEvents & strFiredMissingEvents
- 6. FiredTopNEvents & strFiredTopNEvents

Two dashboards PatternMatch and strPatternMatch should be present with their corresponding business views.

For the Patternmatch dashboard, the following views must be present:

- 1. FiredKpiAlertOutputEvents
- 2. FiredTrendingEvents
- 3. FiredMonitorCountEvents
- 4. FiredMovingAggregationEvents
- 5. FiredMissingEvents
- 6. FiredTopNEvents

For the strPatternMatch dashboard, the following views must be present:

- 1. strFiredKpiAlertOutputEvents
- 2. strFiredDuplicateDetectionEvents
- 3. strFiredTrendingEvents
- 4. strFiredMonitorCountEvents
- 5. strFiredMovingAggregationEvents
- 6. strFiredMissingEvents
- 7. strFiredTopNEvents

Running a PatternMatch Dashboard

To run a PatternMatch dashboard, you must run through the following steps:

 Populate data: Go to ORACLE_HOME/soa/bam/samples/bam-103-pattern-match and execute: startPatternMatch.sh <wls password>

This script will start populating both PATTERNMATCH (logical DO) and PATTERNMATCH_FACT_STREAM (stream DO) which will in turn drive all the views in PatternMatch strPatternMatch dashboards.

- 2. Open the Patternmatch dashboard using the following url: <u>http://<host>:<port>/bam/composer/faces/proxypage?project=PatternMatch&dashboard=</u> <u>PatternMatchDashboard</u>
- Open the strPatternmatch dashboard using the following URL: <u>http://<host>:<port>/bam/composer/faces/proxypage?project=PatternMatch&dashboard=</u> <u>strPatternMatchDashboard</u>
- 4. Once you are done, to stop data population, go to ORACLE_HOME/soa/bam/samples/bam-103-pattern-match and execute: stopPatternMatch.sh



Figure 2 - Sample PatternMatch dashboard

Creating Continuous Queries Using CQL Templates

This section outlines how each of the CQL templates is used to create continuous queries and how outputs from the queries are used to drive real time dashboards.

Data Objects

PATTERNMATCH_FACT is an archived relation DO with the following columns. The contents of an archived relation DO are saved in the database and CRUD operations are allowed.

Column	Column Type	Data Type
customerLocationId	DIMENSION	VARCHAR
productId	DIMENSION	VARCHAR
customerStatus	DIMENSION	VARCHAR
callPriority	ATTRIBUTE	INT
callProcessingTime	MEASURE	INT
callWaitTime	MEASURE	INT
callStatus	ATTRIBUTE	VARCHAR
callClosedTime	DIMENSION	DATETIME
callCreatedTime	DIMENSION	DATETIME

PATTERNMATCH_DIMENSION is also an archived relation DO with the following columns:

Column	Column Type	Data Type
customerLocationId	ATTRIBUTE	VARCHAR
customerLocationName	ATTRIBUTE	VARCHAR

Patternmatch is a Logical DO Logical DOs do not have underlying persistent stores. They are always merges between existing DataObjects. PatternMatch is a merge between PATTERNMATCH_FACT and PATTERNMATCH_DIMENSION using matching 'customerLocationId' as the join condition. It has the following columns.

Column	Source DO	Column	Data Type
		Туре	
customerLocationId	PATTERNMATCH_FACT	DIMENSION	VARCHAR
productId	PATTERNMATCH_FACT	DIMENSION	VARCHAR
customerStatus	PATTERNMATCH_FACT	DIMENSION	VARCHAR
callPriority	PATTERNMATCH_FACT	MEASURE	INT
callProcessingTime	PATTERNMATCH_FACT	MEASURE	INT
callWaitTime	PATTERNMATCH_FACT	MEASURE	INT
callStatus	PATTERNMATCH_FACT	DIMENSION	VARCHAR
callClosedTime	PATTERNMATCH_FACT	ATTRIBUTE	DATETIME
callCreatedTime	PATTERNMATCH_FACT	ATTRIBUTE	DATETIME
customerLocationName	PATTERNMATCH_DIMENSION	DIMENSION	VARCHAR

Oracle BAM also supports the concept of Derived Data Objects. You can use an existing DataObject and extend it by incorporating additional fields. Any data operation done on the derived DO will be applied to the base DO as well. PATTERNMATCH_FACT_STREAM is a stream DO. You can create it with either the 'Archived'=true or the Archived=false condition. If Archive=true, then the data is saved in the database. In this case,

PATTERNMATCH_FACT_STREAM has 'Archived'=false enabled. It has the following columns:

Column	Column Type	Data Type
customerLocationId	DIMENSION	VARCHAR
productId	DIMENSION	VARCHAR
customerStatus	DIMENSION	VARCHAR
callPriority	ATTRIBUTE	INT
callProcessingTime	MEASURE	INT
callWaitTime	MEASURE	INT
callStatus	ATTRIBUTE	VARCHAR
callClosedTime	DIMENSION	DATETIME
callCreatedTime	DIMENSION	DATETIME
customerLocationNa me	ATTRIBUTE	VARCHAR

Suggested Best Practice

If you have Dimension DO where you expect the data to change rarely, then consider marking it a 'Slow Changing Dimension'. If a DO is marked as a Slow Changing Dimension, the CQL query will be optimized to not keep the fact DO tuples in memory. This can give you a considerable performance advantage and reduce memory use. The trade off is that CQ Service restarts the query each time it detects any event on the Slow Changing Dimension DO. Note that this optimization kicks in only when a logical DO based on the Slow Changing Dimension DO is used in the query, and not when the same DO is used directly, outside of a query. This option should ideally be exercised only when you are sure that the changes to your Dimension DO will be very infrequent, because frequent changes to the Dimension DO will result in frequent query restarts and it can worsen performance.

View Name	
	BEAM_VIEW_103
Туре	Simple Data Object
Slow Changing Dimension	
Archived	V
Continuous Query Type	RELATION
Name	PATTERNMATCH_DIMENSION
Display Name	/oracle/callcenter/PATTERNMATCH_DIM
Alias	
Category	
Number of String Columns	25
Number of Long String Columns	25
Number of Integer Columns	25
Number of Float Columns	25
Number of Decimal Columns	25
Number of Date/Time Columns	10
Description	

Figure 3 – Enabling the Slow Changing Dimension option

Creating the PatternMatch Project

The project is a logical organization of work. You can create the project from Oracle BAM Composer and add all the required Data Objects to the project.



Figure 4 - The 'Create Project' list item in BAM Composer

Create				×
Create BAM P	roject			
* Name	PatternMatch			
* Display Name	PatternMatch			
Description				
3	Create BPM Example	Create	Cancel	

Figure 5 – The 'Create BAM Project' start screen

Once the project is created, click on the Data Objects link and add the required Data Objects into the project.

Continuous queries

This section outlines each OOTB Template in detail. Here are some CQL terms used in this document with their respective explanations:

- 1. Stream DO: Stream represents an event stream, which only supports an insert operation. Events arriving on a stream DO are processed by any Continuous Queries over the stream DO currently registered with BAM, but the data is not saved.
- 2. Archived Stream: Archived Stream DO is essentially a stream DO but the data is saved in the database. The amount of data replayed to a newly registering or restarting continuous query is based on settings in the DO.
- 3. Archived Relation: An Archived Relation obtains the initial state of the query from the database and allows incremental computation on top of it.
- 4. Rolling Window: A Rolling Window of N minutes implies that the query will consider only events that have arrived in the last N minutes. When time progresses, events that arrived before N minutes will 'fall off' the window, i.e., they will no longer affect the output of the query. The "Use rolling window" option is made mandatory for all CQL Template queries over Archived Relations. For more details, see Best Practices.
- 5. Update Interval: An update interval can be used to throttle the output of the query. If you do not specify one, the query will give you an output each time there is a change. If you specify an update interval, the output is provided only at those intervals.

To create a continuous Query using a CQL Template, open the project and click on the "Business Queries" link. Enter a name and display name; make sure "Continuous Query" is selected by default and click "Create".

Business Queries * Name * Name * Display Name * Type Image: Continuous Query Group SQL Query Tree Model Query Create Cancel

Figure 6 – Creating a continuous query

Steps specific to each template are detailed below. Ensure that the check box "Activate Continuous Queries :<Query_Name>" is checked before you save the query.

Designing a KPI Alert Template

The goal is to design a query which will, for each product ID, fire an alert when the average call wait time for all the calls 'closed' in the last two minutes is greater than 100(Norm)+10(Allowed deviation). This is achieved using a KPI Alert Template. This template supports all types of DOs.

Over Logical DO (Query: AverageCallWaitingTimeGreateThanThreshold)

Data Object: PatternMatch

- Filter
 - ✓ Branch: all are true callStatus is equal to "CLOSED"
- Measure
 - ✓ Measure field: callWaitTime
 - ✓ Aggregation Function: Average
 - ✓ Group By productId
 - ✓ Use rolling window = true with Range Length = 2 minute based on callCreatedTime
- Threshold

Criteria: Measure is greater than (Norm + Deviation) Norm Definition: Constant, 100 Deviation definition: Scalar, 10

Alert Event

Output all fields

Action

Insert Event output into KPIAlertOutputDDO, and use the following mapping:

DataObject column	Event Output Field
ProjectName	PROJECT_NAME
QueryName	QUERY_NAME
GroupFieldString1	productId
Measure	MEASURE
Norm	NORM
AllowedDeviation	ALLOWED_DEVIATION
ActuralDeviation	ACTUAL_DEVIATION

$rageCallWaitingTimeGreateThanThreshold \times$										
ontinuous Queries							i	Þ	Save	Δ
Activate Continuous Queries : AverageCallWaiting	gTimeGreateThanTh	reshold								
Template KPI Alert Template 💽 🧻										
Description If the moving average of call processing time of send an alert event.	over the last 1 minute fi	or all closed(the status is 'CLOS	ED') calls is not within 30	second	average of call proces	ssing time +/- standard deviation,				
Measure		2. Threshold				3. Alert Event				
P Filter	er	* Criteria				* Output				
* Data Object		Measure is greater that	n a (Norm + Deviation)	•		I All				
/oracle/callcenter/PATTERNMATCH	3 <u>1</u> 2					Measure				
* Measure Field		Norm Definition				Norm Norm				
callWaitTime	-	Constant O Histor	prical			Allowed Deviation				
* Aggregation Functions		a. 1 m 1				Actual Deviation				
Average	•	Simple value 100				I productId				
Group By		Deviation Definition								
productId										
		Scalar Value	10							
✓ Use Rolling Window		Percent of Norm	1 🗘 %			Action				
* Range Length 2 🖨 Minute 💌										
Update Interval 0 🗘 Second 💌		Calculation	1 times Standar	d Devia	ation 💌	Insert event output fiel	ds into			
Update Interval 0 Second *Based on callCreatedTime		Calculation	1 times Standar	d Devia	ation 💌	Insert event output fiel KPIAlertOutputDDO	ds into			
Update Interval 0 Second * Based on callCreatedTime	Map Fields	Calculation	1 times Standar	d Devia	ation 💌	 Insert event output fiel KPIAlertOutputDDO 	ds into			
Update Interval 0 Second * Based on callCreatedTime	Map Fields Data Object	Calculation	1 times Standar	d Devia	ation v	Insert event output field KPIAlertOutputDDO <add action=""></add>	ds into			
Update Interval 0 Second x	Map Fields Data Object KPIAlertOutput	Calculation	1 times Standar	d Devia	ation v X	 Insert event output fiel KPIAlertOutputDDO <add action=""></add> 	ds into			
Update Interval 0 Second V *Based on callCreatedTime V	Map Fields Data Object KPIAlertOutput Operation Type	© Calculation	1 times Standar	d Devia	ation v	Insert event output fiel KPIAlertOutputDDO <add action=""></add>	ds into			
Update Interval 0 Second V *Based on callCreatedTime V	Map Fields Data Object KPIAlertOutput Operation Type Define Mappings	Calculation	1 times Standar	d Devia	ation v	Insert event output fiel KPIAlertOutputDDG <add action=""></add>	ds into			
Update Interval 0 Second x	Map Fields Data Object KPIAlertOutput Operation Type Define Mappings Type	Calculation	1 times Standar	d Devia	upsert Key	Insert event output fiel KPIAlertOutputDOO <add action=""></add>	ds into			
Update Interval 0 Second * Based on californatedTime	Map Fields Data Object KPIAlertOutput Operation Type Define Mappings Type DATETIME DATETIME	Calculation	1 times Standar	d Devia	Lupsert Key	Insert event output fié KPIAlertOutputDOO <add action=""></add>	ds into			
Update Interval 0 Second r * Based on callCreatedTime r	Map Fields Data Object KPIAlertOutput Operation Type Define Mappings Type DATETIME DATETIME DATETIME	Caladation	1 times Standar	d Devia	X Vpoert Key	Insert event output field KPIAlertOutputDoel <add action=""></add>	ds into			
Update Interval 0 Second 💌 * Based on calificreatedTime	Map Fields Data Object KP2AlertOutput Operation Type Define Mappings Type DATETIME DATETIME VARCHAR	Calculation	1 tmes Standar	d Devia	Lupsert Key	Insert event output fiel KPIAlertOutputDO <add action=""></add>	ds into			
Update Interval 0 Second x * Based on calCreatedTime x	Map Fields Data Object KPIAlertOutput Operation Type Define Mappinge Type DATETIME VARCHAR VARCHAR VARCHAR	Calculation	1 tmes Standar	d Devia	Upsert Key	Insert event output fiel KPIAlertOutputDDG <add action=""></add>	ds into			
Update Interval 0 Second r * Based on callCreatedTime r	Map Fields Data Object Kr/Jaler Coutput Operation Type Define Mappings Type DATETIME DATETIME DATETIME VARCHAR VARCHAR DECIMAL DECIMAL	Calculation	1 tmes Standar	d Devia	Upsert Key	Insert event output field KPIAlertOutputDOO <add action=""></add>	ds into			
Update Interval	Map Fields Data Object (P7)AlertOutput Operation Type Define Mappings Type DATETIME DATETIME VARCHAR VARCHAR DECIMAL DECIMAL DECIMAL	Caladation	1 times Standar	d Devia	Uppert Key	Insert event output fiel KPIAlertOutputDoel <add action=""></add>	ds into			
Update Interval 0 Second x * Based on californatedTime x	Map Fields Data Object KPIAIertOutput Operation Type Define Mappings Type DATETIME DATETIME VARCHAR VARCHAR VARCHAR VARCHAR	Calculation	1 tmes Standar	d Devia	Upset Key	Insert event output fiel KPIAlertOutputDO <add action=""></add>	ds into			
Update Interval 0 Second x * Based on calCreatedTime x	Map Fields Data Object KPIAlerCoutpul Operation Type Define Mappings Type DATETIME DATETIME VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR	Calculation Control Column Column Control Column Col	1 tmes Standar	d Devia	Upsert Key	 Insert event output fiel KPIAlertOutputDOO <add action=""></add> 	ds into			
Update Interval	Map Fields Data Object IRPIAIer toutput Operation Type Define Mappings Type DATETIME DATETIME DATETIME DATETIME DATETIME VARCHAR VARCHAR VARCHAR VARCHAR INT	Calculation	1 tmes Standar		Upsert Key	Insert event output field (RPIAlertOutputDOO) <add action=""></add>	ds into			
Update Interval * Based on caliCreatedTime	Map Fields Data Object (PTIAlertOutbul Operation Type Define Mappings Type DATETIME DATETIME VARCHAR DECIMAL DECIMAL DECIMAL DECIMAL DECIMAL DECIMAL DECIMAL DECIMAL	Caladation	1 tmes Standar		Uppert Key	Insert event output field kPIAlertOutputDOO <add action=""></add>	ds into			
Update Interval 0 Second x * Based on calCreatedTime x	Map Fields Data Object (P21AlertOutput Operation Type Define Mappings Type DATETIME DATETIME VARCHAR DECIMAL DECIMAL VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR	Calculation	1 tmes Standar		Upeert Key	Insert event output field kPIAlertOutputDool <add action=""></add>	ds into			
Update Interval 0 Second •	Map Fields Data Object In/FiAlertOutput Operation Type Dattertime DATETIME DATETIME DATETIME VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR MIT DECIMAL DECIMAL DECIMAL DECIMAL DECIMAL	Calculation Calculation Control Contro	1 times Standar		Uppert Key	 Insert event output fiel (RPIAlertOutputDOO) <add action=""></add> 	ds into			
Update Interval 0 Second •	Map Fields Data Object (P7)44F0x0pul Operation Type Define Mappings Type DATETIME DATETIME DATETIME DATETIME DATETIME DATETIME DATETIME VARCHAR VARCHAR VARCHAR VARCHAR NTT INT DECIMAL DECIMAL DECIMAL DECIMAL	Calculation Calculation CDO Conservery Data Object Column DATAOBJECT_CREATED DATAOBJECT_CREATED DATAOBJECT_MCDIFIED ProjectName QueryName GroupFieldDecimal1 GroupFieldDecimal2 GroupFieldString1 GroupFieldString1 GroupFieldString2 GroupFieldInteger 1 GroupFieldInteger 1 GroupFieldInteger 2 Morm AllowedDeviation AdverdDeviation	1 tmes Standar		Upsert Key	Insert event output field (kPIAlertOutputDOO) <add action=""></add>	ds into			

Figure 7 – Creating a continuous query using the KPI Alert Template.

Click the preview icon to see what the final query generated by the template looks like. In this case, the CQL statement is as follows:

CREATE QUERY PatternMatch.AverageCallWaitingTimeGreateThanThreshold as ISTREAM(SELECT AVG(callWaitTime) AS MEASURE, 100.0 AS NORM, 10.0 AS ALLOWED_DEVIATION, AVG(callWaitTime) - 100.0 AS ACTUAL_DEVIATION, productId AS productId, 'PatternMatch' AS PROJECT_NAME, 'AverageCallWaitingTimeGreateThanThreshold' AS QUERY_NAME FROM PatternMatch.PATTERNMATCH[RANGE 2 minute ON callCreatedTime] AS T WHERE (callStatus="CLOSED") GROUP BY T.productId HAVING AVG(T.callWaitTime) - 100.0 > 10.0) destination

"combined:jms:queue/oracle.beam.cqservice.mdbs.alertengine:queuecf/oracle.beam.cqservice.m dbs.alertengine;jms:queue/oracle.beam.cqservice.mdbs.reportcache:queuecf/oracle.beam.cqserv ice.mdbs.reportcache:queuecf/oracle.beam.cqserv

Over Stream DO (StrAvgCallWaitTimeGreaterThanThreshold)

Data Object: PATTERNMATCH_FACT_STREAM

To continue, follow the same procedure as the one outlined for the KPI Alert template query over Logical DO.

	integreater main intestion x										
ontinuous Q	Queries							()	P	Save	Δ
Activate	Continuous Queries : StrAvaCallWaitTimeGrea	aterThanThreshol	d								
Template	KPT Alert Template										
1 cmprotoc		ha laat daalaa ka Kaa)		- 6 1	an Kura A. (a bandrad day la Kar				
Description	send an alert event.	ne last 1 minute for a	all closed (the status is CLOSED) calls is not within 30 secor	d average	or call processin	ng time +/- standard deviation,				
Measure			2. Threshold				3. Alert Event				
	₩ Filter		* Criteria				* Output				
* Data Object			Measure is greater than a	(Norm + Deviation) 💌			II II				
/oracle/callce	enter/PATTERNMATCH_FACT_STREAM 💽 🚻						Measure				
* Measure Field	ld		Norm Definition				Norm Norm				
callWaitTime	•		Constant				Allowed Deviation				
Aggregation I	Functions		Simple Value 100				Actual Deviation				
Average	•						in productio				
Group By			Deviation Definition								
productId											
			Scalar Value	10							
Use Rolling	g Window		Percent of Norm	1 2 %			Action				
* Range Ler	ength 2 Minute										
2											
Update Inte	erval 0 Day		Calculation	1 times Standard Dev	iation 💌		Sos _ Insert event output field	ls into			
Update Inte	erval 0 Day	Map Fields	Calculation	1 times Standard Dev	iation 💌	×	Insert event output field KPIAlertOutputDDO	ls into			
Update Inte	erval 0 Day	Map Fields	Calculation	1 times Standard Dev	iation 💌	×	Insert event output field KPIAlertOutputDDO	ls into			
Update Inte	o Day	Map Fields Data Object KPIAlertOutpo	Calculation	1 times Standard Dev	iation 👻	×	Insert event output field KPIAlertOutputDDO <add action=""></add>	ls into			
Update Inte	erval 0 🐑 Day 💌	Map Fields Data Object KPIAlertOutpu	Calculation	1 times Standard Dev	iation 💌	×	Insert event output field KPIAlertOutputDOO <add action=""></add>	is into			
Update Inte	erval 0 Day 💌	Map Fields Data Object KPIAlertOutpu Operation Type	Calculation	1 times Standard Dev	iation 👻	×	Insert event output field KPIAIertOutputDOO <add action=""></add>	ls into			
Update Inte	erval 0 💭 Day 💌	Map Fields Data Object KPIAlertOutpu Operation Type Define Mapping	Calculation	1 times Standard Dev	iation 💌	×	Insert event output field KPIAlertOutputDOO <add action=""></add>	ls into			
Update Inte	erval 0 Day	Map Fields Data Object KPIAlertOutpu Operation Type Define Mapping Type	© Calculation utDDO e insert • Js Data Object Column	1 times Standard Dev	ups	× v	Insert event output field KPIAlertOutputDOO <add action=""></add>	ls into			
Update Inte	erval 0 💭 Day 💌	Map Fields Data Object KPIAlertOutpu Operation Type Define Mapping Type DATETIME	Calculation	1 times Standard Dev	Ups	× sert Key	Insert event output field KPIAlertOutputDOO <add action=""></add>	ls into			
Update Inte	erval 0 💭 Day 💌	Map Fields Data Object KPIAlertOutpx Operation Type DateTIMe DATETIME DATETIME	© Calculation uDDO Insert Data Object Column DatA Object Column DATAOB.ECT_MODIFIED DataOB.ECT_MODIFIED	1 times Standard Dev	Ups	× sert Key	Insert event output field kPIAlertOutputDOO <add action=""></add>	is into			
Update Inte	erval 0 Day 💌	Map Fields Data Object KPIAlertOutpu Operation Typu Define Mapping Type DATETIME DATETIME VARCHAR	Calculation UEDO a Insert Data Object Column Data Object Column Data Object Column Oatta Object Object Object Oatta Object Object Object Oatta Obj	1 times Standard Dev	Ups	sert Key	Insert event output field KPIAlertOutputDOO <add action=""></add>	is into			
Update Inte	erval 0 💭 Day 💌	Map Fields Data Object KPIAlertOutpu Operation Type Define Mapping Type DATETIME VARCHAR VARCHAR VARCHAR	© Calculation utDOO e insert p Data Object Column DATAOB/ECT_OREATED DATAOB/ECT_MODIFIED ProjectName QueryName Conservational*	1 times Standard Dev Event Output Field PROJECT_NAME QUERY_NAME	Ups	sert Key	Insert event output field KPIAlertOutputDOO <add action=""></add>	is into			
Update Inte	erval 0 Day 💌	Map Fields Data Object KPIAlertOutpx Operation Type DATETIME DATETIME DATETIME VARCHAR VARCHAR DECIMAL DECIMAL	© Calculation utDDO e insert p p p p p p p p p p p p control	1 times Standard Dev	Ups	sert Key	Insert event output field KPIAlertOutputDOO <add action=""></add>	is into			
Update Inte	erval 0 Day 💌	Map Fields Data Object KPIAlertOutpx Operation Type DATETIME DATETIME VARCHAR VARCHAR VARCHAR VARCHAR	Calculation	times Standard Dev Event Output Field PROJECT_NAME QUERY_NAME productId	Ups	x	Insert event output field KPIAertOutputDOO <add action=""></add>	is into			
Update Inte	erval 0 🗊 Day 💌	Map Fields Data Object KPTAlertOutpu Operation Type DatTetTime DATETIME VARCHAR DECIMAL DECIMAL DECIMAL VARCHAR	Calculation Calcul	1 times Standard Dev Event Output Field PROJECT_NAME QUERY_NAME productId	Ups	sert Key	Insert event output field kPIAlertOutputDOO <add action=""></add>	is into			
Update Inte	erval 0 💭 Day 💌	Map Fields Data Object KPIAlertOutpu Operation Type DATETIME DATETIME VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR	© Calculation UEDO e Insert Data Object Column DatA CoBject_COREATED DATA COBJEct_MODIFIED ProjectName QueryName QueryName GroupFieldDecimal1 GroupFieldDecimal1 GroupFieldDecimal1 GroupFieldDecimal2 GroupFieldNting1 GroupFieldNting2 Home ProjectName CompFieldNting2 Home ProjectName Home Proje	1 times Standard Dev Event Output Field PROJECT_NAME QUERY_NAME productId	Ups	×	Insert event output field KPIAlertOutputDOO <add action=""></add>	is into			
Update Inte	erval 0 Day	Map Fields Data Object KPIAlertOutpa Operation Type DatterIme DATETIME DATETIME VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR NT INT	Calculation UEDOO e insert Dat Object Column Dat AOBJECT_CORENTE DATAOBJECT_ORENTED DATAOBJECT_ORENTED DATAOBJECT_ORDIFIED ProjectName QueryName GroupFieldDecimal2 GroupFieldDecimal2 GroupFieldDecimal2 GroupFieldIntegr1 GroupFieldIntegr1	1 times Standard Dev	Ups	×	Insert event output field KPIAIertOutputDOO <add action=""></add>	s into			
Update Inte	erval 0 Day v	Map Fields Data Object KPI AlertOutpx Operation Type DATETIME DATETIME VARCHAR VARCHAR VARCHAR DECIMAL VARCHAR INT INT DECIMAL	Calculation utDOO te insert Data Object Column DatA CoBject Column DatA CoBject Column DatA CoBject Column Otaba Ect _MODIFIED ProjectName QueryName GroupFieldDecmal 1 GroupFieldDecmal 1 GroupFieldDecmal 1 GroupFieldString 1 GroupFieldString 1 GroupFieldString 2 GroupFieldStri	1 times Standard Dev	Ups	x	Insert event output field KPIAertOutputDOO <add action=""></add>	s into			
Update Inte	erval 0 🗊 Day 💌	Map Fields Data Object KPT/AlertOutpu Operation Type DatTetTime DATETIME VARCHAR VARCHAR DECIMAL DECIMAL NT DECIMAL DECIMAL DECIMAL	© Calculation	1 times Standard Dev Event Output Field PROJECT_NAME QUERY_NAME productId MEASURE NORM	Ups Ups V V V V V V V V V V V V V V V V V V V	×	Insert event output field KPIAerOutputDOO <add action=""></add>	s into			
Update Inte	erval 0 Day 💌	Map Fields Data Object KPFIAlertOutpu Operation Type DatTetTime VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR	© Calculation	1 times Standard Dev Event Output Field PROJECT_NAME QUERY_NAME productId MEASURE NORM ALLOWED_DEVIATION	Ups Vps V V V V V V V V V V V V V V V V V	×	Insert event output field KPIAlertOutputDOO <add action=""></add>	s into			

Figure 8 – Creating a continuous query using the KIP Alert Template, over a stream DO.

Click the preview icon to see what the final query generated by the template looks like. In this case, the CQL statement is as follows:

CREATE QUERY PatternMatch.StrAvgCallWaitTimeGreaterThanThreshold as ISTREAM(SELECT AVG(callWaitTime) AS MEASURE , 100.0 AS NORM , 10.0 AS

ALLOWED_DEVIATION , AVG(callWaitTime) - 100.0 AS ACTUAL_DEVIATION , productId AS productId , 'PatternMatch' AS PROJECT_NAME , 'StrAvgCallWaitTimeGreaterThanThreshold' AS QUERY_NAME FROM PatternMatch.PATTERNMATCH_FACT_STREAM[RANGE 2 minute] AS T WHERE (callStatus="CLOSED") GROUP BY T.productId HAVING AVG(T.callWaitTime) - 100.0 > 10.0) destination

"combined:jms:queue/oracle.beam.cqservice.mdbs.alertengine:queuecf/oracle.beam.cqservice.m dbs.alertengine;jms:queue/oracle.beam.cqservice.mdbs.reportcache:queuecf/oracle.beam.cqserv ice.mdbs.reportcache?batch=true"

Designing a Trending Design Template

The goal is to detect if the callProcessingTime is increasing by more than 10% for two consecutive intervals. The trending detection template is used to achieve this. This template works with all types of Dos.

Over Logical DO (Query: CallProcessingTimeTrendingUp10Percent)

Data Object: PatternMatch

- Filter
 - ✓ None: No filters specified for this query.
- Measure
 - ✓ Measure Field: callProcessingTime
 - ✓ Aggregation Function: None The callProcessingTime field is observed for changes, so an aggregation function need not be specified. On the other hand, if you want to detect, say, trends in Sum of callProcessingTime or Average of callProcessingTime, then you must specify appropriate aggregation functions.
 - Partition By: productId
 Trends are identified for each productId independently.
 - ✓ Use rolling window = true with Range Length = 2 minute based on callCreatedTime. In this example, the 'update interval' field isn't used. Generally, if an aggregation function is applied in trend detection, the 'update interval' field can help.
- Trending
 - ✓ Change greater than 10%
 - ✓ Consecutive interval 2

- Output
 - ✓ Select all available fields
- Action

Insert Event output into TrendingDetectionOuputDDO, and use the following mapping:

DataObject column	Event Output Field
ProjectName	PROJECT_NAME
QueryName	QUERY_NAME
GroupFieldString1	T.productID
EndPointTrendingValue	T.endCallProcessingTime
StartPointTrendingDecimalValue	T.callProcessingTime

Contained Queeds Co	CallProcessingTimeTrendingUp10Percent ×						2 🛽
Contract Contractions Contract Contractions Contract C	Continuous Queries						i) 🔎 Save 🛕
<pre>tender petetein "emplate in """""""""""""""""""""""""""""""""""</pre>	Activate Continuous Queries : CallProcessingTimeTrer	ndingUp10Percent	t				
fieldProcessingTime with a rolling window of 1 minutes is the rolling window of 1 m	Template Trending Detection Template 💽 🕕						
I Messure 2 Tending * Output * Ostas Object * Grange 0 % % * Ostas Object * Grange time • 0 0 % % * Output * Measure Field 2 * * Output * Measure Field 2 * * Grange time • 0 0 % % * Measure Field 2 * * Output * Measure Field * Output * Output * Measure ** 0 * Output * Output	Description If callProcessingTime with a rolling window of 1 min	utes is trending up b	y 10% for 2 occurences, outpu	t callProcessingTime and Trer	ding.		
1. Measure							
 Change * Orange * Or	1. Measure		2. Trending			3. Output	
• Oreato Opject Oreater than ■ OD ③ % • Measure Field 2 • Agregation Functions 2 • Partition By 2 • Partition By 0 ③ Second ■ • Vue Rolling Window • Consecutive Interval • Range Length 2 • Use Rolling Window • Consecutive Interval • Range Length 2 • Use Rolling Window • Consecutive Interval • Range Length 2 • Beased on callCreatedTime • Data Object • Based on callCreatedTime • Data Object Column VARCHAR GroupFieldDecmal 2 • OroupFieldDecmal 2 VARCHAR GroupFieldDecmal 2 • OroupFie	Y Filter		* Change			* Output	
Improve Field <td< td=""><td>* Data Object</td><td></td><td>Greater than</td><td>■ 10 \$ %</td><td></td><td>All</td><td></td></td<>	* Data Object		Greater than	■ 10 \$ %		All	
* Measure Field * Appreaden Functions None * Appreaden Functions None * Partition By * Outs Boling Window * Range Length * Based on calCreatedTime VacCHAR Operation Type Note * Based on calCreatedTime VacCHAR Operation Type Note * Based on calCreatedTime VacCHAR Operation Type Note * Based on calCreatedTime VacCHAR Operation Type Note VacCHAR Operation Type VacCHAR	/oracle/callcenter/PATTERNMATCH		* Consecutive Interval			CallProcessingTime	
calProcessingTime • Agrogation Functions None Partition By productid • Use Rolling Window • Agre Length • Based on • Based on calCroatedTime • Based on calCroatedTime • CICIMAL GroupFieldDirege1 VARCHAR GroupFieldDirege1 VARCHAR CorpertionTime VIT GroupFieldDirege1 VIT StartPointTrendingNube. Corput VIT StartPointTrendingNube. VIT StartPointTrendingNube. VIT StartPointTrendingNube. VIT StartPointTrendingNube. VIT VIT VIT VIT VIT VIT VIT VIT VIT VIT VIT VIT VIT <td>* Measure Field</td> <td></td> <td>2 ≑</td> <td></td> <td></td> <td>endcallProcessingTime</td> <td></td>	* Measure Field		2 ≑			endcallProcessingTime	
* Aggregation Functions None Partition By productid • Use Rolling Window * Range Length 2 Minute 0 • Based on callCreatedTime Data ObjectC.colum VURCHAR GroupFieldDireing VACHAR GroupFieldDireing VAT StartPointT	callProcessingTime 💌					roductId	
None Partition By productid v Use Rolling Window * Range Length 2 Minute 0 Second * Based on callCreatedTime Define Mapping: Type Data Object Column Vupdate Interval 0 Second • Based on callCreatedTime Define Mapping: Type Data Object Column VanCHAR QueryName QueryName QueryName QueryName DeCIMAL GroupFieldDroinal1 VARCHAR GroupFieldDroinal2 VARCHAR VA	* Aggregation Functions	Map Fields			×	1	
Partition By productid Partition By productid Productid C partition Type insert Define Mappings Define Mappings DatTETTIME DatABCDByCCT_CREATED DatTETTIME DATADABSECT_CREATED DatTETTIME DATADABSECT_CREATED DatTETTIME DATADABSECT_MODIFIED PROJECT_JNAME DATETTIME DATACHAR CyceryName QUERY_NAME DECIMAL GroupFieldDreimal1 DECIMAL GroupFieldDring2 Int <	None	Data Object					
roductid Use Ralling Window * Range Length 2 Winute Based on allCreatedTime Operation Type insert in Define Mappings Type Data Object Column Vent Output Field Upsert Key DATETIME DATAOBECT_MODIFIED Non-DECT_JAMME OACOABECT_CREATED OACOABECT_UNAME OAC	Partition By	TrendingDete	ctionOuputDDO		•		
Coperation Type Insert ■ ✓ Use Rolling Window • Range Length 2 1 Minute 0 Second • Coperation Type Insert • Define Mappings Type Define Mappings Data Object Column Event Output Field v Update Interval 0 Second • Second • Second • Second • Second • Second • Coperation Type Type Data Object Column Event Output Field • Update Interval 0 Second • VARCHAR ProjectName QUERY_NAME • DECIMAL GroupFieldString1 - rendulProdedting2 - rendulProdedting2 - rendulProdettinger1 - rendulProcessingTime - INT GroupFieldInteger1 - rendulProcessingTime - INT StartPointTrendingInteg	productid						
 ✓ Use Ralling Window * Range Length 2 Minute Define Mappings Type Data Cobject Column Vare Time Vare	producad	Operation Type	e insert 💌				
*Range Length 2 minute Update Interval 0 minute *Based on callCreatedTime *Based on callCreatedTime VARCHAR QueryName QueryName QueryName DECIMAL GroupFieldDecimal1 VARCHAR GroupFieldDecimal2 VARCHAR StartPointTrendingDeces VARCHAR StartPointTrendingDeces	✓ Use Rolling Window	Define Mapping	IS			Action	
Update Interval O is Second ■ * Based on CallCreatedTime DATETIME DATAOBECT_MODIFIED ● * Based on CallCreatedTime VARCHAR ProjectName QUERY_JNAME ● VARCHAR QueryName QUERY_JNAME ● ● - DECIMAL GroupFieldDecimal 1 ● ● ● - - - VARCHAR GroupFieldDecimal 2 ● ● ● - <td>* Range Length 2 Minute</td> <td>Туре</td> <td>Data Object Column</td> <td>Event Output Field</td> <td>Upsert Key</td> <td></td> <td></td>	* Range Length 2 Minute	Туре	Data Object Column	Event Output Field	Upsert Key		
Dobale Interva DatacoBect_MODIFIED DATACOBECT_		DATETIME	DATAOBJECT_CREATED			Insert event output fields into	
* Based on allCreatedTime VARCHAR ProjectName PROJECT_JNAME VARCHAR QueryName QUERY_NAME VARCHAR QueryName QUERY_NAME VARCHAR GroupFieldDecinal1 VARCHAR GroupFieldDecinal2 VARCHAR GroupFieldString1 T_productId VARCHAR GroupFieldString2 INT GroupFieldInteger1 INT GroupFieldInteger1 INT GroupFieldInteger1 DECIMAL EndPointTrendingIndue T_endcalProcessingTime DECIMAL StartPointTrendingInteg VK	opdate Interval 0 Second	DATETIME	DATAOBJECT_MODIFIED			TrendingDetectionOuputDDO	
VARCHAR QueryName QUERY_NAME Image: Compression of the second of t	* Based on callCreatedTime	VARCHAR	ProjectName	PROJECT_NAME			
DECIMAL GroupFieldDecm31 Image: CompFieldDecm32 VARCHAR GroupFieldString1 T.productId VARCHAR GroupFieldString2 Image: CompFieldInteger1 INT GroupFieldInteger1 Image: CompFieldInteger2 INT GroupFieldInteger3 Image: CompFieldInteger3 DECIMAL EndPontTrendingValue T.endcalProcessingTime DECIMAL StartPointTrendingInteg T.calProcessingTime INT StartPointTrendingInteg Image: CompFieldInteger3		VARCHAR	QueryName	QUERY_NAME		<add action=""></add>	
DECIMAL GroupFieldDecimal2 VARCHAR GroupFieldShing1 T.productId VARCHAR GroupFieldShing2 INT GroupFieldInteger1 INT GroupFieldInteger2 DECIMAL EndPointTrendingValue T.endcalProcessingTime DECIMAL StartPointTrendingValue T.endcalProcessingTime INT StartPointTrendingInteg INT StartPointTrendingInteg		DECIMAL	GroupFieldDecimal 1				
VARCHAR GroupFieldSting1 T.productId VARCHAR GroupFieldSting2 Image: CompFieldIntege:		DECIMAL	GroupFieldDecimal2				
VARCHAR GroupFieldString2 GroupFieldInteger 1 GroupFieldInteger 2 GroupFieldInteger 3 DECIMAL EnPointTrendingNue T.endcalProcessingTime DECIMAL StartPointTrendingInteg T.calProcessingTime GroupFieldInteger 3 GroupFieldInteger 4 GroupFieldInteger 4 GroupFieldInteger 5 GroupFieldInteger 4 GroupFieldInteger 5 GroupFieldInteger 5 GroupFieldInteger 5 GroupFieldInteger 5 		VARCHAR	GroupFieldString1	T.productId			
INT GroupFieldInteger1 INT GroupFieldInteger2 DECIMAL EnPontTrendngValue T.endcalProcessingTime INT StartPointTrendingInteg		VARCHAR	GroupFieldString2				
INT GroupFieldInteger2 DECIMAL EndPointTrendingValue T.endcalProcessingTime DECIMAL StartPointTrendingDedum T.calProcessingTime INT StartPointTrendingInteg		INT	GroupFieldInteger1				
DECIMAL EndPointTrendingValue T.endkalProcessingTime DECIMAL StartPointTrendingDecim T.calProcessingTime CIMAL StartPointTrendingInteg COK		INT	GroupFieldInteger2				
DECIMAL StartPointTrendingDecim T.calProcessingTime INT StartPointTrendingInteg K		DECIMAL	EndPointTrendingValue	T.endcallProcessingTime			
INT StartPointTrendingInteg		DECIMAL	StartPointTrendingDecim	T.callProcessingTime			
ОК		INT	StartPointTrendingInteg				
					ОК		

Figure 9 - Create a continuous query using the Trending Detection Template

Click the preview icon to see what the final query generated by the template looks like. In this case, the CQL statement is as follows:

CREATE QUERY PatternMatch.CallProcessingTimeTrendingUp10Percent as SELECT T.callProcessingTime, T.endcallProcessingTime, T.productld, 'PatternMatch' AS PROJECT_NAME, 'CallProcessingTimeTrendingUp10Percent' AS QUERY_NAME FROM (SELECT productId, callCreatedTime, callProcessingTime, endcallProcessingTime FROM (ISTREAM(SELECT productId, callCreatedTime AS callCreatedTime, callProcessingTime AS callProcessingTime , callProcessingTime AS endcallProcessingTime FROM PatternMatch.PATTERNMATCH[RANGE 2 minute ON callCreatedTime] AS S)) AS P ORDER BY callCreatedTime) AS Q MATCH_RECOGNIZE (PARTITION BY productId MEASURES C.callProcessingTime AS endcallProcessingTime,A.callProcessingTime AS callProcessingTime,A.productId AS productId ALL MATCHES PATTERN (A B+ C) DEFINE B AS B.callProcessingTime>1.1*prev(B.callProcessingTime) and count(*) <= 2, C AS C.callProcessingTime>1.1*last(B.callProcessingTime) and count(*) = 3) AS T destination "combined:jms:queue/oracle.beam.cqservice.mdbs.alertengine:queuecf/oracle.beam.cqservice.m dbs.alertengine;jms:queue/oracle.beam.cqservice.mdbs.reportcache:queuecf/oracle.beam.cqserv ice.mdbs.reportcache?batch=true"

Over Stream DO (Query: StrCallProcessingTimeTrendUp10PC)

Data Object: PATTERNMATCH_FACT_STREAM

To continue, follow the same procedure as outlined in the Trending Detection Template Query over Logical DO.

StrCallProcessingTimeTrendUp10PC ×										
Continuous Queries							i	<u>,</u>	Save	Δ
Activate Continuous Queries : StrCallProcessingTimeTre	endUp10PC									
Template Trending Detection Template 💽 🧻										
Description If callProcessingTime with a rolling window of 15 minu	utes is trending up by 3	10% for 2 occurences, output	callProcessingTime and Tre	nding.						
										•
1. Measure	2	2. Trending				3. Output				
🖓 Filter		* Change				* Output				
* Data Object		Greater than 💌	10 2 %			V All				
/oracle/callcenter/PATTERNMATCH_FACT_STREAM		Consecutive Interval				CallProcessingTime				
* Measure Field		2				endcallProcessingTime				
callProcessingTime						v productId				
* Aggregation Functions	Map Fields				×					
None	Data Object									
None 💌	Data Object TrendingDetec	tionOuputDDO								
None Partition By	Data Object TrendingDetec	tionOuputDDO								
None Partition By productId	Data Object TrendingDetec Operation Type	tionOuputDDO								
None Partition By productId Vise Rolling Window	Data Object TrendingDetec Operation Type Define Mappings	insert				Action				
None Partition By productid Vuse Rolling Window	Data Object TrendingDetec Operation Type Define Mappings Type	tionOuputDDO s Data Object Column	Event Output Field		▼ Upsert Key	Action				
None Partition By ProductId Use Rolling Window * Range Length 2 Minute	Data Object TrendingDetec Operation Type Define Mappings Type DATETIME	tionOuputDDO insert Data Object Column DATAOBJECT_CREATED	Event Output Field		Upsert Key	Action				
None Partition By productid Vise Rolling Window Range Length Update Interval 0 Day	Data Object TrendingDetec Operation Type Define Mappings Type DATETIME DATETIME	tionOuputDDO insert Data Object Column DATAOBJECT_CREATED DATAOBJECT_MODIFIED	Event Output Field		Upsert Key	Action Insert event output fields into TrendingDetectionOuputDDO				
None Partition By productid Vuse Rolling Window Range Length Qudate Interval	Data Object TrendingDetec Operation Type Define Mappings Type DATETIME DATETIME VARCHAR	tionOupuIDDO insert Data Object Column DATAOBJECT_OREATED DATAOBJECT_MODIFIED ProjectName	Event Output Field		Upsert Key	Action Conserve event output fields into TrendingDetectionOuputDDO				
None Partition By productid Vuse Rolling Window Range Length 2 Minute Update Interval 0 Day	Data Object TrendingDetec Operation Type DateTIME DATETIME VARCHAR VARCHAR	tionOuputDDO insert DataObject Column DatAc082ECT_CREATED DATA082ECT_MODIFIED ProjectName QueryName	Event Output Field PROJECT_NAME QUERY_NAME		Upsert Key	Action Insert event output fields into TrendingDetectionOuputDDO				
None Partition By Productid State Rolling Window Range Length 2 Minute Update Interval 0 Day	Data Object TrendingDetec Operation Type Define Mappings Type DATETIME VARCHAR VARCHAR DECIMAL	tonOuputDDO Insert Data Object Column DATAOB ECT_CREATED DATAOB ECT_CREATED DATAOB ECT_CREATED COLUMNE QueryName GroupFieldDecimal 1	Event Output Field PROJECT_NAME QUERY_NAME	•	Upsert Key	Action Insert event output fields into TrendingDetectionOuputDDO <add action=""></add>				
None Partition By Productid Vise Rolling Window Range Length 2 Day Update Interval 0 Day	Data Object TrendingDetec Operation Type Define Mapping Type DATETIME DATETIME VARCHAR VARCHAR VARCHAR DECIMAL	tionOuputDDO insert Data Object Column Data Object Column DataOstECT_CREATED DATAOStECT_MODIFIED ProjectName QueryName GroupFieldDecimal1 GroupFieldDecimal2	Event Output Field PROJECT_NAME QUERY_NAME	•	Upsert Key	Action Insert event output fields into TrendingDetectionOuputDDD <add action=""></add>				
None Partition By productid Vuse Rolling Window Range Length 2 Minute Update Interval 0 Day	Data Object TrendingDetec Operation Type Define Mapping: Type DATETIME DATETIME VARCHAR VARCHAR VARCHAR	Insert Data Object Column DATAOBJECT_ORATED DATAOBJECT_MODIFIED ProjectName QueryName GroupFieldDecimal2 GroupFieldDecimal2	Event Output Field PROJECT_NAME QUERY_NAME T.productId		Upsert Key	Action Insert event output fields into TrendingDetectionOuputDDO <add action=""></add>				
None Partition By productid Vuse Rolling Window Range Length 2 Minute Update Interval 0 Day	Data Object TrendingDetec Operation Type Define Mapping Type DATETIME VARCHAR VARCHAR VARCHAR VARCHAR	tonOupuEDDO Insert Data Object Column Data Object Column DatAOBJECT_CREATED DATAOBJECT_MODIFIED ProjectName GroupFielDDecimal1 GroupFielDDecimal1 GroupFieldString1 GroupFieldString1	Event Output Field PROJECT_NAME QUERY_NAME T.productId	•	Upsert Key	Action Action TrendingDetectionOuputDDO <add action=""></add>				
None Partition By productid Ø Use Rolling Window Range Length 2 Minute Update Interval 0 Day	Data Object TrendingDetec Operation Type Define Mapping Type DATETIME DATETIME VARCHAR DECIMAL DECIMAL DECIMAL VARCHAR VARCHAR INT	tionOuputDDO Insert Data Object Column DATAOBJECT_CREATED DATAOBJECT_CREATED DATAOBJECT_CREATED CATAOBJECT_MODIFIED ProjectName GroupFieldDecimal 1 GroupFieldDecimal 2 GroupFieldString 2 GroupFieldString 2 GroupFieldString 2 GroupFieldInteger 1	Event Output Field PROJECT_NAME QUERY_NAME T.productId		Upsert Key	Action Insert event output fields into TrendingDetectionOuputDDO <add action=""></add>				
None Partition By Productid State Rolling Window Range Length 2 Minute Update Interval 0 Day	Data Object TrendingDetec Operation Type Define Mapping: Type DATETIME VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR	tionOuputDDO insert Data Object Column DatAOB ECT_CREATED DATAOB ECT_MODIFIED ProjectName QueryName QueryName GroupFieldDecinal1 GroupFieldDecinal2 GroupFieldDecinal2 GroupFieldString1 GroupFieldString1 GroupFieldInteger1	Event Output Field PROJECT_NAME QUERY_NAME T.productId		Upsert Key	Action TrendingDetectionOuputDDO <add action=""></add>				
None Partition By productid Vides Rolling Window Range Length 2 Vides Rolling Window Update Interval 0 Day	Data Object TrendingDetec Operation Type Define Mapping: Type DATETIME VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR NT INT DECIMAL	Insert	Event Output Field PROJECT_NAME QUERY_NAME T.productId T.endcallProcessingTime		Upsert Key	Action Insert event output fields into TrendingDetectionOuputDDO <add action=""></add>				
None Partition By productId Vuse Rolling Window Range Length 2 Minute Update Interval 0 Day	Data Object TrendingDetec Operation Type Define Mapping Type DATETIME DATETIME VARCHAR VARCHAR DECIMAL DECIMAL DECIMAL DECIMAL	tonoUputDDO Insert Data Object Colum Data Object Colum DataOBJECT_CREATED DATAOBJECT_CREATED DATAOBJECT_MODIFIED ProjectName GroupFieldDecimal1 GroupFieldDecimal1 GroupFieldString1 GroupFieldString1 GroupFieldString1 GroupFieldInteger1 GroupFieldInteger1 GroupFieldInteger3 StartPointTrendingValue StartPointTrendingValue	Event Output Field PROJECT_NAME QUERY_NAME T.productId T.endcaliProcessingTime T.caliProcessingTime		Upsert Key	Action				
None Partition By productid Vise Rolling Window Range Length 2 Minute Update Interval 0 Day	Data Object TrendingDetec Operation Type Define Mapping: Type DATETIME DATETIME VARCHAR VARCHAR DECIMAL DECIMAL DECIMAL DECIMAL DECIMAL INT	tionOupuIDDO Insert Data Object Colum DATAOB ECT_CREATED DATAOB ECT_CREATED DATAOB ECT_CREATED DATAOB ECT_CREATED CATAOBECT_MODIFIED ProjectName GroupFieldString1 GroupFieldString1 GroupFieldString2 GroupFieldS	Event Output Field PROJECT_NAME QUERY_NAME T.productId T.endcallProcessingTime T.callProcessingTime		Upsert Key	Action Insert event output fields into TrendingDetectionOuputDDO <add action=""></add>				

Figure 10 – Creating the StrCallProcessingTimeTrendUp10PC query

Click the preview icon to see what the final query generated by the template looks like. In this case the CQL statement is as follows:

CREATE QUERY PatternMatch.StrCallProcessingTimeTrendUp10PC as SELECT T.callProcessingTime , T.endcallProcessingTime , T.productId , 'PatternMatch' AS

PROJECT_NAME , 'StrCallProcessingTimeTrendUp10PC' AS QUERY_NAME FROM (ISTREAM(SELECT productId , callProcessingTime AS callProcessingTime , callProcessingTime AS endcallProcessingTime FROM PatternMatch.PATTERNMATCH_FACT_STREAM[RANGE 2 minute] AS S)) AS Q MATCH_RECOGNIZE (PARTITION BY productId MEASURES C.callProcessingTime AS endcallProcessingTime,A.callProcessingTime AS callProcessingTime,A.productId AS productId ALL MATCHES PATTERN (A B+ C) DEFINE B AS B.callProcessingTime>1.1*prev(B.callProcessingTime) and count(*) <= 2, C AS C.callProcessingTime>1.1*last(B.callProcessingTime) and count(*) = 3) AS T destination "combined:jms:queue/oracle.beam.cqservice.mdbs.alertengine:queuecf/oracle.beam.cqservice.m dbs.alertengine;jms:queue/oracle.beam.cqservice.mdbs.reportcache:queuecf/oracle.beam.cqservice.m

Designing a Missing Event Template

The goal is to design a query that can detect when a call "Suspend" action is not directly followed by "Resume". This is achieved using the Missing Event Template which supports all types of DOs.

Over Logical DO (Query: MissingResumeCallDetection)

Data Object: PatternMatch

- Filter
 - ✓ None No filter specified.
- Measure
 - ✓ productId
 You can select any additional fields that you wish to expose in the query output.
 - ✓ Partition by: productId
 - This implies that the PatternMatching is done independently for each productId. Consider the following order of events productId, callStatus
 - 1. Suspend
 - 2, Open -> This will not trigger an output because productId one and two are tracked separately
 - 1, Resume
 - 2, Suspend
 - 1, Closed->This will not trigger an output because productId one and two are tracked separately

2, On Hold \rightarrow This will trigger an output because for productId=2, 'Suspend' was followed by 'On Hold' instead of 'Resume'

 Use rolling windows = true, Range Length two minutes, based on callCreatedTime.

- Events
 - Pattern: Event A is not followed directly by event B
 The other options available are "Event C is missing between event A and Event
 B" and "Event A is not followed by Event B in specific duration"
- Output

Output all fields

Action

Insert Event output into MissingEventOutputDDO, and use the following mapping

DataObject column	Event Output Field
ProjectName	PROJECT_NAME
QueryName	QUERY_NAME
MeasureString1	T. B_productId
MeasureString2	T.B_callStatus

MissingResumeCallDetection ×									2 🛛
Continuous Queries							i	Save	Δ
Activate Continuous Queries : MissingResumeCallDetection	1								
Template Missing Event Template 💽 🧻									
Description For product ID with a rolling window of 1 minute, if call su	ispend is not followed by re	sume, output product ID and '	Missing Event'.						
	1								-
1. measure	Z. Event					3. Output			
Pilter	Pattern					* Output			
* Data Object	Even	t A is not followed directly by E	vent B			All			
/oracle/callcenter/PATTERNMATCH	Even	t C is missing between event A	and Event B			A_callStatus			
* Measure Field	C LVBI	t A is not followed by Event bi	r specific duration			B_callStatus			
All	Event A					A_productid			
CallPriority	T Call	Status 💌 * Eq	Jai to	ND .		B_productid			
CalProcessingTime	* cal	Status 💌 * Eq	ual to 💌 * RESUM	E					
CalStatus									
CalWart I me	Man Fields				>	1			
	Pate Object								
Customer Status	MissingEventC	utputDDQ				Action			
V productid									
	Operation Type	insert 💌				Insert event output fields into			
Partition By	Define Manning					Missingeventoutputbbo			
productId	Туре	Data Object Column	Event Output Field		Upsert Key	and the officers			
	DATETIME	DATAOBJECT_CREATED		•		<add action=""></add>			
✓ Use Rolling Window	DATETIME	DATAOBJECT_MODIFIED		•					
* Range Length 2 🗘 Minute 💌	VARCHAR	ProjectName	PROJECT_NAME	-					
Update Interval 0 🖨 Second 💌	VARCHAR	QueryName	QUERY_NAME	•					
* Based on callCreatedTime	DECIMAL	GroupFieldDecimal 1		-					
	DECIMAL	GroupFieldDecimal2		-					
	VARCHAR	GroupFieldString1		-					
	VARCHAR	GroupHeidString2		-					
	INT	GroupFieldInteger 1							
	DECIMAL	Groupmeidinteger2		-					
	DECIMAL	MeasureDecimal?		-					
	INT	MeasureInteger 1							
	INT	MeasureInteger2		-					
	VARCHAR	MeasureString 1	T.B_productId	-					
	VARCHAR	MeasureString2	T.B_callStatus	-					
	DECIMAL	PatternFieldDecimal 1		-					
	DECIMAL	PatternFieldDecimal2		-					
	DECIMAL	PatternFieldDecimal3		-					
	INT	PatternFieldInteger 1		-					
	INT	PatternFieldInteger2		•					
	INT	PatternFieldInteger3		-					
	VARCHAR	PatternFieldString1		-					
	VARCHAR	PatternFieldString2		•					
	VARCHAR	PatternFieldString3		-					

Figure 11 – Setting up the MissingResumeCallDetection Query

Click the preview icon to see what the final query generated by the template looks like. In this case, the CQL statement is as follows:

CREATE QUERY PatternMatch.MissingResumeCallDetection as SELECT T.A_callStatus , T.B_callStatus , T.A_productId , T.B_productId , 'PatternMatch' AS PROJECT_NAME , 'MissingResumeCallDetection' AS QUERY_NAME FROM (SELECT callCreatedTime , callStatus , productId FROM (ISTREAM(SELECT callCreatedTime , callStatus , productId FROM PatternMatch.PATTERNMATCH[RANGE 2 minute ON callCreatedTime] AS S)) AS P ORDER BY callCreatedTime) AS Q MATCH_RECOGNIZE (PARTITION BY productId MEASURES A.callStatus AS A_callStatus,NOTB.callStatus AS B_callStatus,A.productId AS A_productId,NOTB.productId AS B_productId ALL MATCHES PATTERN (A NOTB) DEFINE A AS (callStatus='SUSPEND'),NOTB AS (NOT(callStatus='RESUME'))) AS T destination "combined:jms:queue/oracle.beam.cqservice.mdbs.alertengine:queuecf/oracle.beam.cqservice.m dbs.alertengine;jms:queue/oracle.beam.cqservice.mdbs.reportcache:queuecf/oracle.beam.cqserv ice.mdbs.reportcache?batch=true"

Over Stream DO (StrMisResumCallDetection)

Data Object: PATTERNMATCH_FACT_STREAM

To continue, use the same procedure as the one outlined for the Missing Event Template Query over Logical DO.

····· · · · · · · · · · · · · · · · ·								?
ntinuous Queries							i) 🔎	Save
Activate Continuous Queries : StrMisResumCallDetection								
Template Missing Event Template								
For loan ID with a rolling window of 1 hour if loan submit	sion is followed by a	noroval without business review	utput loan ID, loan am	ount and	'Missing Event'			
Description	sion is rollowed by a	pprovar without business review, t	auput loan 10, loan am	ount, and	missing Event.			
easure	2.1	vent				3. Output		
						to her		
Y Hiter	F	attern) Event A is not followed directly b	v Event B					
/orade/calicenter/DATTEDNMATCH_EACT_STDEAM	0	Event C is missing between even	t A and Event B					
	0	Event A is not followed by Event	B in specific duration			✓ A_calStatus		
		* callStatus	Equal to 💌 * 🔍 ISI	PEND		B productId		
CallPriority	E	Event B	203	0.10				
i callProcessingTime		callStatus 🔹 *	Equal to 💌 * RES	JME				
					_			
a ustomeri ocationId	Map Fields				×	1		
customer ocationName	Data Object							
Customer Status	MissingEve	entOutputDDO			•	Action		
✓ productId								
	Operation I	ype insert				Insert event output fields into		
irtition By	Define Map	pings				MissingEventOutputDDO		
productId	Type	Data Object Column	Event Output Field		Upsert Key			
	DATETIME	DATAOBJECT_CREATED		-		<add action=""></add>		
✓ Use Rolling Window	DATETIME	DATAOBJECT_MODIFIED		-				
* Range Length 2 🗘 Minute 💌	VARCHAR	ProjectName	PROJECT_NAME	-	_			
Update Interval 0 Day	VARCHAR	QueryName	QUERY_NAME	-				
	DECIMAL	GroupHeldDecimal1		-				
	DECIMAL	GroupHeidDecimai2		-				
	VARCHAR	GroupFieldString1		-				
	TNT	GroupFieldInteger 1						
	INT	GroupFieldInteger?		-				
	DECIMAL	MeasureDecimal 1		-				
	DECIMAL	MeasureDecimal?		-				
	INT	MeasureIntener1		-				
	INT	MeasureInteger 2						
	VARCHAR	MeasureString 1	T.B. productId					
	VARCHAR	MeasureString2	T.B callStatus	-				
	DECIMAL	PatternFieldDecimal 1		-				
	DECIMAL	PatternFieldDecimal2		-				
	DECIMAL	PatternFieldDecimal3		-				
	INT	PatternFieldInteger 1		-				
	INT	PatternFieldInteger2		-				
	INT	PatternFieldInteger3		-				
	VARCHAR	PatternFieldString1		-				
	THIS GENERAL							
	VARCHAR	PatternFieldString2		-				

Figure 12 – Setting up the StrMisResumCallDetection Query

Click the preview icon to see what the final query generated by the template looks like. In this case, the CQL statement is as follows.

CREATE QUERY PatternMatch.StrMisResumCallDetection as SELECT T.A_callStatus , T.B_callStatus , T.A_productId , T.B_productId , 'PatternMatch' AS PROJECT_NAME ,

'StrMisResumCallDetection' AS QUERY_NAME FROM (ISTREAM(SELECT callStatus, productId FROM PatternMatch.PATTERNMATCH_FACT_STREAM[RANGE 2 minute] AS S)) AS Q MATCH_RECOGNIZE (PARTITION BY productId MEASURES A.callStatus AS A_callStatus,NOTB.callStatus AS B_callStatus,A.productId AS A_productId,NOTB.productId AS B_productId ALL MATCHES PATTERN (A NOTB) DEFINE A AS (callStatus='SUSPEND'),NOTB AS (NOT(callStatus='RESUME'))) AS T destination

"combined:jms:queue/oracle.beam.cqservice.mdbs.alertengine:queuecf/oracle.beam.cqservice.m dbs.alertengine;jms:queue/oracle.beam.cqservice.mdbs.reportcache:queuecf/oracle.beam.cqserv ice.mdbs.reportcache:queuecf/oracle.beam.cqserv

Designing a Monitor Count Template

The goal is to design a query that will indicate if more than one event is detected in the past 2 minutes for the same customerLocationId and productId pair. This is achieved using Monitor Count Template which supports all types of Dos.

Over Logical DO (Query: MonitorDuplicatedEvents)

Data Object: Patternmatch

- Filter
 - ✓ None. No filter specified.

✓ Measure

- Measure Field: Callstatus
 You will get a comma separated list of all call statuses in the output field
 AGGcallStatus.
- ✓ Group By: customerLocationId, productId The count of each of these pairs is monitored separately.
- Count: Greater than 1
 You have the option to specify a variety of options like greater than, less than, equal to, and so on.
- ✓ Use rolling window: yes, Range length two minutes, based on callCreatedTime.
- Output

Output all fields

Action

Insert Event output into MonitorCntOutputDDO, and use the following mapping.

DataObject column	Event Output Field
ProjectName	PROJECT_NAME
QueryName	QUERY_NAME

GroupFieldString1	T. customerLocationId
GroupFieldString2	T. productId
MeasureString2	T. AGGcallStatus

ontinuous	0								-		
	Queries								i) 🍦	Save	•
Activate	e Continuous Queries : MonitorDuplicatedEven	ts									
Template	Monitor Count Template 💽 🚺										
Description	If count great than 1 for product ID with rolling wi	ndow of 1 minute, ou	utput product ID and duplicated	d values							
Measure				2	. Output						
	🖓 Filter				* Output						
* Data Object	t				🔽 All						
/oracle/callo	lcenter/PATTERNMATCH				monitor_c	count					
Measure Field	1				AGGcallSt	tatus					
					customer	LocationId					
CallPrior	rity				roductio	d					
CallProc	cessingTime										
🗹 callStat	tus										
CallWait	tTime										
custome	erLocationId										
Custome	erLocationName				Action						
Custome	ierStatus										
D product	tId				ana Ins	sert event output	fields into				
Group By					Mo Mo	onitorCntOutputD	00				
customerlic	ocationId: productId	Map Fields				×					
customerco		Data Ohiaat									
		Data Opect									
* Count		MonitorCntOu	itputDDO			-					
• Count Greater tha	an 💌 1 ¢	MonitorCntOu	itputDDO			•					
* Count Greater tha	an 💌 1 💭	MonitorCntOu Operation Type	e insert			•					
* Count Greater tha ✓ Use Rolli	an 💌 1 🖢	MonitorCntOu Operation Type	itputDDO			v					
* Count Greater tha ✓ Use Rolli * Range Le	an 1 ing Window ength 2 Minute	Data Object MonitorCntOL Operation Type Define Mapping Type	insert p Data Object Column	Event Output Field	d	▼ Upsert Key					
* Count Greater tha ✓ Use Rolli * Range Le Update Int	an v 1 b Ing Window ength 2 c Minute v Interval 0 c Second v	Data Object MonitorCntOL Operation Type Define Mapping Type DATETIME	itputDDO insert insert J S Data Object Column DATAOBJECT_CREATED	Event Output Field	d	Upsert Key					
* Count Greater tha ✓ Use Rolli * Range Le Update Int * Base	an v 10 Ing Window Length 20 Minute v Iterval 00 Second v sed on calCreatedTime v	Data Object MonitorCntOL Operation Type Define Mapping Type DATETIME DATETIME	ItputDDO insert insert bata Object Column DATAOBJECT_CREATED DATAOBJECT_MOOIFIED	Event Output Field	d	Upsert Key					
* Count Greater tha ✓ Use Rolli * Range Le Update Int * Base	an I I I IIII Ing Window Length 2 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Define Mapping Type DATETIME DATETIME VARCHAR	toputDDO insert	Event Output Field	d v v	Upsert Key					
* Count Greater the V Use Rolli * Range Li Update Int * Base	an v 10 Ing Window ength 20 Minute v terval 0 8 Second v sed on calCreatedTime v	Data Object MonitorCntOL Operation Type Define Mapping Type DATETIME DATETIME VARCHAR VARCHAR	tputDDO insert Data Object Column DATAOBJECT_GREATED DATAOBJECT_GREATED DATAOBJECT_MODIFIED ProjectName QueryName	Event Output Field		Upsert Key					
 Count Greater the ✓ Use Rolli * Range Li Update Int * Base 	an 1 1 and 1	Data Object MonitorCntOL Operation Type DateTime DATETIME VARCHAR VARCHAR DECIMAL DECIMAL	tpuEDO a insert v Data Object Column DATAOBJECT_CREATED DATAOBJECT_MODIFIED ProjectName GroupPiedDecimal 1	Event Output Field		Upsert Key					
 Count Greater the Use Rolli Range Li Update Int * Base 	an v 1 b Ing Window ength 2 minute v terval 0 second v sed on callCreatedTime v	Define Mapping Type Daffine Mapping Type DATETIME DATETIME VARCHAR VARCHAR DECIMAL DECIMAL DECIMAL	tpuEDO e insert Data Object Column DatA Object Column DATAOBJECT_GREATED DATAOBJECT_MODIFIED ProjectName GroupFieldDecimal1 GroupFieldDecimal2	Event Output Field		Upsert Key					
* Count Greater the V Use Rolli * Range Li Update In * Base	an v 1 b Ing Window Length 2 b Minute v Iterval 0 b Second v sed on callCreatedTime v	MonitorCntDL Operation Type Define Mapping Type DATETIME DATETIME VARCHAR VARCHAR DECIMAL DECIMAL VARCHAR	tpuEDO tpuEDO tpuEDO Data Object Colum DATAOBJECT_OREATED DATAOBJECT_OREATED ProjectName QueryName QueryName GroupFieldDecimal1 GroupFieldDecimal2 GroupFieldDecimal	Event Output Field	d V V V V V tonto	Upsert Key					
* Count Greater tha V Use Rolli * Range L Update In * Base	an v 10 Ing Window ength 20 Minute v terval 00 Second v sed on calCreatedTime v	MoniterCritic MoniterCritic Operation Type DateTIME DATETIME VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR	tpuEDO participation participation Data Object Column Data Object Column Data Object Column Data Object Column Data Object Column Data Object Column Data Object Column ProjectName GroupFieldDecimal GroupFieldDecimal GroupFieldDecimal	Event Output Fiel PROJECT_NAME QUERY_NAME T.customerLocatt T.productId	d V V V V V V V V	Upsert Key					
* Count Greater thy Vuse Rolli * Range L Update In * Bas	an v 10 Ing Window Length 20 Minute v terval 00 Second v sed on calCreatedTime v	Data Object MonitorCntOL Operation Type DatTETIME DATETIME VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR INT	tpuEDDO a insert b more than the more than	Event Output Field PROJECT_NAME QUERY_NAME T.customerLocat T.productId	d V V V V V V V	Upset Key					
* Count Greater thy Vuse Rolli * Range Li Update In * Bas	an v 15 Ing Window ength 2 % Minute v o % Second v sed on callCreatedTime v	Antis Color Monitor ChOU Operation Type DateTIME DATETIME DATETIME VARCHAR VARCHAR VARCHAR VARCHAR INT INT DECIMAL DECIMAL	tpuEDDO e insert Data Object Column DatA Object Column DatA Object Column DatA Object Column OatA Object Column OatA Object Column QueryName GroupFieldDecimal1 GroupFieldString1 GroupFieldString2 GroupFieldString3 GroupFieldSt	Event Output Fiel PROJECT_NAME QUERY_NAME T.customerLocat T.productId	d V V V V V ionid V V	Upsert Key					
* Count Greater the Vuse Roll * Range L Update In * Bas	an v 15 ing Window .ength 2 Minute v terval 0 Second v sed on callCreatedTime v	Date Object Monitor ChOU Operation Type Define Mapping Type DATETIME DATETIME DATETIME VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR	tpuEDO tpuEDO tpuEDO Data Object Colum DATAOBJECT_CREATED DATAOBJECT_CREATED DATAOBJECT_MODIFIED ProjectName GroupFieldDecimal 1 GroupFieldDecimal 2 GroupFieldInteger 1 GroupFieldInteger 1 GroupFieldInteger 2 MeasureDecimal 1	Event Output Field	d v v v v v v v v v v	Upset Key					
* Count Greater the Vuse Roll Range L Update In * Bas	an v 10 Ing Window ength 20 Minute v terval 00 Second v sed on calCreatedTime v	Varchar Varchar	tpuEDO Data Object Column DatA Object Column DATAOBJECT_CREATED DATAOBJECT_MODIFIED ProjectName QueryName GroupFieldDecimal 1 GroupFieldString1 GroupFieldString1 GroupFieldInteger 1 GroupFieldInteger 1 MeasureDecimal 1 MeasureDecimal 2 MeasureDecimal 2	Event Output Field	d V V V V V V V V V V V	Upset Key					
* Count Greater thu	an v 10 Ing Window Length 20 Minute v terval 00 Second v sed on calCreatedTime v	Date dojet Monitor ChtOu Operation Typy Define Mapping Type DATETIME DATETI	tpuEDDO a insert v b bata Object Column DATAOBJECT_CREATED DATAOBJECT_MODIFIED ProjectName QueryName GroupFieldDecimal 1 GroupFieldDstring 1 GroupFieldString 1 GroupFieldString 1 GroupFieldString 2 GroupFieldString 2 GroupFieldTecenal 1 MeasureInteger 1 MeasureInteger 2	Event Output Field PROJECT_NAME QUERY_NAME T.customerLocat T.productId	d V V V V V V V V V V	Upsert Key					
* Count Greater thu ✓ Use Roll * Range Lu Update In * Bas	an v 15 Ing Window Length 2 Minute v treval 0 5 Second v and on calCreatedTime v	Data Object Monitor ChOU Operation Type Date Mapping Type DATETIME DATETIME DATETIME DATETIME VARCHAR VARCHAR VARCHAR VARCHAR INT INT DECIMAL DECIMAL DECIMAL INT INT INT VARCHAR	tpuEDDO te insert Data Object Column OntA Object	Event Output Field PROJECT_NAME QUERY_NAME T.customerLocati T.productid	d v v v v v v v v v v v	Upsert Key					
* Count Greater thu Y Use Roll * Range L Update In * Bas	an v 15 Ing Window .ength 2 Minute v o Second v sed on calCreatedTime v	VarcHaR VARCHAR VARCHAR VARCHAR VARCHAR	tpuEDDO a insert b for the second s	Event Output Field	d v v v v v v v v v v v v v v v	Upset Key					

Figure 13 - Setting up the MonitorDuplicatedEvents Query

Click the preview icon to see what the final query generated by the template looks like. In this case, the CQL statement is as follows.

CREATE QUERY PatternMatch.MonitorDuplicatedEvents as SELECT T.monitor_count , T.AGGcallStatus , T.customerLocationId , T.productId , 'PatternMatch' AS PROJECT_NAME , 'MonitorDuplicatedEvents' AS QUERY_NAME FROM (ISTREAM(SELECT count(*) AS monitor_count , listAggWrapper(listAgg(callStatus),", ") AS AGGcallStatus , customerLocationId AS customerLocationId , productId AS productId FROM PatternMatch.PATTERNMATCH[RANGE 2 minute ON callCreatedTime] AS S GROUP BY S.customerLocationId,S.productId HAVING COUNT(*) > 1)) AS T destination

"combined:jms:queue/oracle.beam.cqservice.mdbs.alertengine:queuecf/oracle.beam.cqservice.m

dbs.alertengine;jms:queue/oracle.beam.cqservice.mdbs.reportcache:queuecf/oracle.beam.cqserv ice.mdbs.reportcache?batch=true"

Over Stream DO (StrAvgCallWaitTimeGreaterThanThreshold)

Data Object: PATTERNMATCH_FACT_STREAM

To continue, follow the same procedure as outlined for the Monitor Count Template Query over Logical DO.



Figure 14 - Setting up the StrMonitDupEvents Query

Click the preview icon to see what the final query generated by the template looks like. In this case the CQL statement is as follows.

CREATE QUERY PatternMatch.StrMonitDupEvents as SELECT T.monitor_count , T.AGGcallStatus , T.customerLocationId , T.productId , 'PatternMatch' AS PROJECT_NAME , 'StrMonitDupEvents' AS QUERY_NAME FROM (ISTREAM(SELECT count(*) AS monitor_count , listAggWrapper(listAgg(callStatus),", ") AS AGGcallStatus , customerLocationId AS customerLocationId , productId AS productId FROM PatternMatch.PATTERNMATCH_FACT_STREAM[RANGE 2 minute] AS S GROUP BY S.customerLocationId,S.productId HAVING COUNT(*) > 1)) AS T destination "combined:jms:queue/oracle.beam.cqservice.mdbs.alertengine:queuecf/oracle.beam.cqservice.m dbs.alertengine;jms:queue/oracle.beam.cqservice.mdbs.reportcache:queuecf/oracle.beam.cqserv ice.mdbs.reportcache?batch=true"

Designing a Moving Aggregation Template

The goal is to design a query that will output the moving average of callProcessing time in the last two minutes for each productId. This is achieved using the Moving Aggregation Template, which supports all types of DOs.

Over Logical DO (Query: MovingAverageForCallProcessingTime)

Data Object: PatternMatch

- Filter
 - ✓ None. No filter specified.
- Measure
 - Measure field: callProcessingTime (Average)
 The other options are max, min, count and sum. You can select multiple aggregation functions for the same field.
 - ✓ Group By: productId
 - ✓ Use rolling window = True; Range length -two minutes, based on callCreatedTime.
- Output
 - ✓ All fields.
- Action

Insert Event output into MovAggrOutputDDO, and use the following mapping.

DataObject column	Event Output Field
ProjectName	PROJECT_NAME
QueryName	QUERY_NAME
GroupFieldString1	T.productId
AggregationValue1	T.AVGcallProcessingTime

ngAverageForCallProcessing	Time X											(
ntinuous Queries		r c llo							()	Þ	Save	ſ
Activate Continuous Quer	les : MovingAvera	JeForCaliProcess	inglime									
Template Moving Aggregatio	n Template											
Description Calculate the movi	ng average of callPro	cessingTime with a r	rolling window of 1 minute.	Output moving average, time and	Moving Aggrega	ation.						
easure					2. Output							
		🚏 Filter			* Output							
Data Object					A							
/oracle/callcenter/PATTERNMATC	СН	- 62			AVGcallPro	ocessingTime						
rleasure Field					roductId							
] 📴 callPriority												
🛛 📴 callProcessingTime	Average											
] 📴 callWaitTime												
-					Action							
oup By												
productId		•			🔅 - Inse	ert event output fields into						
✓ Lise Rolling Window					tan Fields	Aggroupubbo			×			
* Dance Length	Marsha				Tap rielus							
	Minute			L	Jata Object MovAggrQutput	tDDO						
Update Interval 0	Second 💌											
* Based on callCreated	Time 💌			c	Operation Type	insert 💌						
					Define Mappings							
					Туре	Data Object Column	Event Output Field	_	Upsert Key			
					DATETIME	DATAOBJECT_CREATED		-				
				1	DATETIME	DATAOBJECT_MODIFIED		-				
					VARCHAR	ProjectName	PROJECT_NAME	-				
					VARCHAR	Queryivame	QUERT_NAME					
					DECIMAL	GroupFieldDecimal?		-				
					VARCHAR	GroupFieldString1	T productId					
					VARCHAR	GroupFieldString2		-				
					INT	GroupFieldInteger 1						
					INT	GroupFieldInteger2		-				
					DECIMAL	AggregationValue 1	T.AVGcallProcessing	Fime 💌				
					DECIMAL	AggregationValue2		-				
					DECIMAL	AggregationValue3		-				
									_			
					DECIMAL	AggregationValue4		-				

Figure 15 – Setting up the MovingAverageforCallProcessingTime Query

Click the preview icon to see what the final query generated by the template looks like. In this case, the CQL statement is as follows.

CREATE QUERY PatternMatch.MovingAverageForCallProcessingTime as SELECT T.AVGcallProcessingTime, T.productld, 'PatternMatch' AS PROJECT_NAME, 'MovingAverageForCallProcessingTime' AS QUERY_NAME FROM (ISTREAM(SELECT AVG(callProcessingTime) AS AVGcallProcessingTime, productId AS productId FROM PatternMatch.PATTERNMATCH[RANGE 2 minute ON callCreatedTime] AS S GROUP BY S.productId)) AS T destination

"combined:jms:queue/oracle.beam.cqservice.mdbs.alertengine:queuecf/oracle.beam.cqservice.m dbs.alertengine;jms:queue/oracle.beam.cqservice.mdbs.reportcache:queuecf/oracle.beam.cqserv ice.mdbs.reportcache:queuecf/oracle.beam.cqserv

Over Stream DO (Query Name: StrMovinAvgCallProcessTime)

Data Object: PATTERNMATCH_FACT_STREAM

To continue, follow the same procedure as outlined for the Moving Aggregation Template Query over Logical DO.

StrMovinAvgCall	ProcessTime ×										? N •
Continuous (Queries							(D 🔑	Save	Δ
Activate	Continuous Queries : StrMovinAvgCallProcessT	me									
Template	Moving Aggregation Template										
Description	Calculate the moving average of dropped calls with a	rolling window of 24	hours. Output moving average	ge, time and Moving A	ggregation						
											-
1. Measure				2.0	utput						
	Pilter				Output						
* Data Object											
/oracle/callo	enter/PATTERNMATCH_FACT_STREAM				AVGcallPr	ocessingTime					
* Measure Hel	d			L.	/ production						
🔽 📴 callProce	essingTime Average										
🔲 📴 callWait	Time										
				Ac	ction						
Group By											
productId					💩 🔸 İns	ert event output	t fields into				
V Use Rolling	Window				MC MC	VAggrOutputDDC					
***		Map Fields				^					
* Range Le	ngth 2 - Minute	Data Object	000								
Update Inte	erval 0 🗘 Day 💌	MovAggrOutput	000			•					
		Operation Type	insert 💌								
		Define Mappings	Data Object Column	Event Output Field		Lincert Key					
		DATETIME	DATAOBIECT CREATED	Event output ried		opservicy					
		DATETIME	DATAOBJECT MODIFIED								
		VARCHAR	ProjectName	PROJECT NAME	-						
		VARCHAR	QueryName	QUERY_NAME	-						
		DECIMAL	GroupFieldDecimal 1		-						
		DECIMAL	GroupFieldDecimal2		-						
		VARCHAR	GroupFieldString1	T.productId	-						
		VARCHAR	GroupFieldString2		-						
		INT	GroupFieldInteger 1		-						
		INT	GroupFieldInteger2		-						
		DECIMAL	AggregationValue 1	T.AVGcallProcessin	gTime 💌						
		DECIMAL	AggregationValue2		-						
		DECIMAL	AggregationValue3		-						
		DECIMAL	AggregationValue4		-						
		DECIMAL	AggregationValue5		-						
		-	-								
						OK					

Figure 16 - Setting up the StrMovinAvgCallProcessTime Query

Click the preview icon to see what the final query generated by the template looks like. In this case the CQL statement is as follows.

CREATE QUERY PatternMatch.StrMovinAvgCallProcessTime as SELECT T.AVGcallProcessingTime , T.productld , 'PatternMatch' AS PROJECT_NAME , 'StrMovinAvgCallProcessTime' AS QUERY_NAME FROM (ISTREAM(SELECT AVG(callProcessingTime) AS AVGcallProcessingTime , productId AS productId FROM PatternMatch.PATTERNMATCH_FACT_STREAM[RANGE 2 minute] AS S GROUP BY S.productId)) AS T destination

"combined:jms:queue/oracle.beam.cqservice.mdbs.alertengine:queuecf/oracle.beam.cqservice.m dbs.alertengine;jms:queue/oracle.beam.cqservice.mdbs.reportcache:queuecf/oracle.beam.cqserv ice.mdbs.reportcache:queuecf/oracle.beam.cqserv

Designing a Top N Template

The goal is to design a query that will indicate the top five products with the maximum callProcessingTime in the last two minutes. This is achieved using Top N Template, which supports all types of DOs.

Over Logical DO (Query: Top5MaximumCallProcessingTime)

Data Object: Patternmatch

- Filter
 - ✓ None. No filter specified.
- Measure
 - ✓ Measure Field: callProcessingTime
 - ✓ Aggregation Function: Max
 - ✓ Group By: productId
 - ✓ Top N: 5
 - Here, you can specify the number of top rows you want in the output.
 - ✓ Use rolling window: True; Range length 2 Minute, based on callCreatedTime
- Output

Output all fields.

• Action

Insert Event output into TopNOutputDDO, and use the following mapping.

DataObject column	Event Output Field
ProjectName	PROJECT_NAME
QueryName	QUERY_NAME
GroupFieldString1	T. productId
TopNValue	T. MAXcallProcessingTime

	fop5MaximumC	allProcessingTime ×										3 🛛
Activation constrained	Continuous	Queries							i	Þ	Save	Δ
Textility Decode Celeclister tite too 5 lienes for calibroance gifting with a maining withow of 1 monte. Output product 15 diret 7 top Values I. Celeclister I Celeclister <td< td=""><td>Activate</td><td>Continuous Queries : Top5MaximumCallProc</td><td>essingTime</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></td<>	Activate	Continuous Queries : Top5MaximumCallProc	essingTime									
Conception Conception <td>Template</td> <td>Top N Template</td> <td></td>	Template	Top N Template										
Leterpton		Calculate the top 5 items for callProcissingTime with	a colling window o	of 1 minute. Output product Id :	and Top Values							
Lessure • Carbon	Description	calculate are up orteino for call rociongrine vita	ra rolling mildorre									
• Data Chypic! • Mackace Freed • Mackace Freed • Agregation Fractions Max • Agregation Fractions • Max • Max • Brook of the fractions • Max • Solutions • Solutions <td>1. Measure</td> <td></td> <td></td> <td></td> <td></td> <td>2. Output</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	1. Measure					2. Output						
* bes of operations in the constraints of the const		😗 Filter				Output						
Ivrade/calkenter PATTERDMATCY Maxer Frid Controls Max Topic Controls Max Topic Controls Max Max Max Topic Controls Max Max Max Max Max Max Max Max Max Max Max Max Max Max Max Max	* Data Object											
* Messare Field callProcessingTime * Agrogation Functions Max * Coop By productid * Top N * Top N * Top N * Lose Rolling Window * Range Length 2 Minute * Based on callCreatedTime * Base	/orade/callo	enter/PATTERNMATCH				MAXCZ	ProcessingTime					
CalifroncessingTime * Aggregation Functions Max Corup By productid * Top N * Los Rolling Window * Los Rolling Window * Los Rolling Window * Los Rolling Window * Based on calificeatedTime © Operation Type InsetT Define Mappings Type Data Object Column Type Data Object Column Type OutputDOO Operation Type InsetT Path Colling VARCHAR ProjectInee PROJECT_INAME PROJECTINAME PROJECTINA PROJECTINAME PROJECTINAME PROJECTINA PROJECTINAME PROJECTINAME PROJECTINAME PROJECTINAME PROJECTINAME PROJECTINAME PROJECTINAME PROJECTINAME PROJECTINAM	* Measure Fie	ld				v produ	tīd					
* Agoregation Functions Max Coup By ordoutdi * Top N * Top N * Use Rolling Window * Use Rolling Window * Use Rolling Window * Based on relificatedTime * Based on relificatedTime * Based on relificatedTime * Based on relificatedTime * Route Type Total Object Top NoutputDDO Operation Type Insert Define Mappings Type Total Object Define Mappings Type Total Object Total Define Mappings Type Total Define Define br>Define Def	callProcessir	ngTime 🔹										
Max Group By productid **Tap H SS * Lus haling Window **Range Lingth * Based on allCreatedTime Operation Type Define Magorige <	* Aggregation	Functions										
Group By * Top N * Top N * Use Rolling Window * Range Length 2 1 Minutive Update Interval Second * Based on caliCreatedTime Define Mappings Type Data Object Data Object Data Object Column Define Mappings Type Data Object Column Update Interval Update	Max											
inductid *To N * To N * Class Roling Window * Range Length 2 Minute Update Interval 0 S Scond * Based on callCreatedTime Define Mappings To Define Mappings Define Mappings VARCHAR ProjectName URENT PROJECT_NAME UPD VARCHAR ProjectName URENT PROJECT_NAME UPD VARCHAR ProjectName URENT ProductId UPD	Group By											
* Top N S C Use Rolling Window * Range Length 2 Minute wassed on calCreatedTime Coperation Type Part Object TopNOLupuUDDO Coperation Type Part Column Type Coperation Type Part Column Coperation Type Column Coperation Type Column Coperation Type Column	productId											
S * Ube Roling Window * Range Length 2 Minute O Second * Based on callCreatedTime Define Mappings Type Data Object Type Data Object Column Operation Type Insert event Output Fields into Type Data Object Column Define Mappings Type Type Data Object Type Data Object Column Operation Type Insert event Output Fields into Type Data Object Column Operation Type Data Object Type Data Object Type Data Object Type Data Type Data Object Ty	* Top N											
* Lose Rolling Window * Range Length 2 * Minute Update Interval 0 * Second * Based on allCreatedTime Operation Type Type Data Object Operation Type Define Mappings Type Data Object Column Vent Output Field Update Interval Operation Type Interval Operation Type Operation Type Operation Type Operation Type Interval Operation Type Operation Type Interval Operation Type Interval	5	\$				Action						
Construing finitude Range Length 2 Minute Winute Pased on calCreatedTime Coperation Type insert Pased on calCreatedTime Coperation Type insert Point	¥ Lice Reli	ag Window										
* kange Lengin 2 Winute 0 * Based on 0 * Based on calCreatedTime Define Mappings Type Data Object Column VARCHAR CoupFieldDecimal 1 DeCIMAL GroupFieldString 1 VARCHAR GroupFieldString 1 VARCHAR GroupFieldString 2 VARCHAR GroupFieldString 2 INT GroupFieldString 2 INT GroupFieldInteger 1 INT GroupFieldInteger 2 INT GroupFieldInteger 2 INT GroupFieldInteger 2	* Ose Rom					@ •	Insert event outp	ut fields into TopNOutputDDO				
Update Interval 0 * Based on * Based on • ealCreatedTime OperationType • Define Mappings • Update Object Column • Define Mappings • Define Mappings • Define Mappings • Update Object Column • Define Mappings • Update Object Column • Define Mappings • Operation Type • Define Mappings • Operation Type • Define Mappings • Operation Type • Operation Type • Operation Type • Operation Type <td> Kange Le </td> <td>2 Minute</td> <td>Map Fields</td> <td></td> <td></td> <td></td> <td>×</td> <td></td> <td></td> <td></td> <td></td> <td></td>	 Kange Le 	2 Minute	Map Fields				×					
* Based on calCreatedTime TopNOUtputDDO Operation Type insert Define Mappings Define Mappings Define Mappings Data Object Column VaRCHAR ProjectName PROJECT_PRATED VARCHAR ProjectName PROJECT_NAME VARCHAR ProjectName PROJECT_NAME VARCHAR GroupFieldDecimal2 VARCHAR GroupFieldString1 T.productId VARCHAR GroupFieldString1 T.productId VARCHAR FOroupFieldString2 VARCHAR	Update Int	terval 0 Second 💌	Data Object									
Operation Type insert Define Mappings Type Data Object Column Event Output Field Uppert Key DATETIME DATAOBJECT_CREATED Image: Column <li< td=""><td>* Base</td><td>ed on callCreatedTime</td><td>TopNOutputD</td><td>DO</td><td></td><td></td><td>•</td><td></td><td></td><td></td><td></td><td></td></li<>	* Base	ed on callCreatedTime	TopNOutputD	DO			•					
Define Mappings Type Data Object Column Event Output Field Uppert Key DATETIME DATAOBJECT_CREATED DATETIME DATAOBJECT_MODIFIED VARCHAR QueryName QUERY_NAME I DECIMAL GroupFieldDecimal1 I INT GroupFieldString1 INT GroupFieldInteger1 INT GroupFieldInteger2 INT CorupFieldInteger2 INT CorupFieldInteger2 INT CorupFieldInteger2 INT CorupFieldInteger3 INTAQPFieldInteger4 INTAQPFieldInteger4 INTAQPFieldInteger4 INTAQPFieldInteger4 INTAQPFieldInteger5 INTAQPFieldInteger4 INTAQPFieldInteger5 INTAQPFieldInteg			Operation Type	e insert 👻								
Define Mappings Type Data Object Colum Event Output Field Upset Key DATETIME DATAOB/ECT_CREATED Image: Column Column Image: Column Image: Column Column<												
Injee Uata Uger Cultum Loper Leve DATETIME DATAOBECT_CREATED Image: Control of			Define Mapping	js	5							
DATEINE DATAGBECT_MODIFIED DATETINE DATAGBECT_MODIFIED VARCHAR ProjectName PROJECT_NAME VARCHAR QueryName QUERY_NAME DECIMAL GroupFieldDecimal1 DECIMAL GroupFieldDecimal2 VARCHAR GroupFieldDirtng2 VARCHAR GroupFieldTirtng2 VARCHAR GROUPFIElD V			DATETIME	Data Object Column	Event Output He	DIE	Upsert Key					
VARCHAR QueryName PROJECT_NAME VARCHAR QueryName QUERY_NAME DECIMAL GroupFieldbechmal1 DECIMAL GroupFieldbechmal2 VARCHAR GroupFieldbechmal2 VARCHAR GroupFieldbring2 INT GroupFieldbring2 INT GroupFieldbring2 INT GroupFieldhring2 INT			DATETIME	DATAOBJECT_OREATED								
VARCHAR QueryName QUERY_NAME QUER			VARCHAR	DrojectName	PROJECT NAME							
DECIMAL GroupFieldDecimal2 UARCHAR GroupFieldDecimal2 UARCHAR GroupFieldDecimal2 UARCHAR GroupFieldString1 T.productId UARCHAR GroupFieldString2 UARCHAR GroupFieldString2 UARCHAR GroupFieldString2 UARCHAR GroupFieldInteger UARCHAR GROUPFIEL UARC			VARCHAR	Querchisme	OLIEDY, NAME							
DECIMAL GroupFieldString2 VARCHAR GroupFieldString2 VARCHAR GroupFieldString2 INT GroupFieldString2 INT GroupFieldInteger1 INT GroupFieldInteger2 DECIMAL TopNValue T.MAXcalProcessingTm			DECIMAL	GroupFieldDecimal 1	Quartification							
VARCHAR GroupFieldString1 T.productId C VARCHAR GroupFieldString2 C INT GroupFieldString2 C INT GroupFieldInteger1 C INT GroupFieldInteger2 C DECIMAL TopNValue T.MAXcallProcessingTm C			DECIMAL	GroupFieldDecimal2								
VARCHAR GroupFieldString2 INT GroupFieldInteger1 INT GroupFieldInteger2 DECIMAL TopNvalue T.MAXcallProcessingTm			VARCHAR	GroupFieldString 1	T.oroductId							
INT GroupFieldInteger 1 INT GroupFieldInteger 2 DECIMAL TopNValue T.MAXcallProcessingTime			VARCHAR	GroupFieldString2	. iproduced							
INT GroupFieldInteger 2 DECIMAL TopWalue T.MAXcaliProcessingTm			INT	GroupFieldInteger 1								
DECIMAL TopWalue T.MAXcallProcessingTm			INT	GroupFieldInteger2								
			DECIMAL	TopNValue	T.MAXcalProces	ssingTim						

Figure 17 – Setting up the Top5MaximumCallProcessingTime Query

Click the preview icon to see what the final query generated by the template looks like. In this case the CQL statement is as follows.

CREATE QUERY PatternMatch.Top5MaximumCallProcessingTime as SELECT T.MAXcallProcessingTime , T.productld , 'PatternMatch' AS PROJECT_NAME , 'Top5MaximumCallProcessingTime' AS QUERY_NAME FROM (RSTREAM(SELECT S.productld , S.MAXcallProcessingTime FROM (SELECT productld , MAX(callProcessingTime) AS MAXcallProcessingTime FROM PatternMatch.PATTERNMATCH[RANGE 2 minute ON callCreatedTime] AS Q GROUP BY Q.productld) AS S ORDER BY S.MAXcallProcessingTime DESC ROWS 5)) AS T destination

"combined:jms:queue/oracle.beam.cqservice.mdbs.alertengine:queuecf/oracle.beam.cqservice.m dbs.alertengine;jms:queue/oracle.beam.cqservice.mdbs.reportcache:queuecf/oracle.beam.cqserv ice.mdbs.reportcache:queuecf/oracle.beam.cqserv

Over Stream DO (Query Name: StrTop5MaxCallProcessTime)

Data Object: PATTERNMATCH_FACT_STREAM To continue, follow the same procedure as the one outlined for the Top N template Query over Logical DO.

StrTop5MaxCallPr	rocessTime X									? 🛛 🔻
Continuous Q	ueries							1	Save	Δ
								-		_
Activate C	ontinuous Queries : Str i opsmaxcaiiProcess II	me								
Template	Top N Template									
Description	Calculate the top 3 items for totalSale with a rolling w	indow of 24 hours.	Output UPC, itemName, totalSi	ale and Monitor Co	ount.					
										-
1. Measure				2	. Output					
	😗 Filter				Output					
* Data Object					I Al					
/oracle/callcer	nter/PATTERNMATCH_FACT_STREAM 🖃 🔞				MAXcalPr	rocessingTime				
* Measure Field					roductIo	- 1				
callProcessing	Time 💌									
* Aggregation F	unctions									
Max	•									
Group By										
productId										
* Top N										
5 🗘					Action					
Use Rolling	window				🎲 🔸 Ins	sert event outpu	t fields into TopNOutputDDO			
* Range Len	gth 2 🗘 Minute 💌	Map Fields					×			
Update Inter	rval 0 💭 Day 💌	Data Object								
		TopNOutputDD	00			•	J			
		Operation Type	insert 💌							
		Define Mappings	Data Object Column	Event Output Ei	ald	Lincert Key	1			
		DATETIME	DATAOBJECT_CREATED	Evencoupurn	eiu 👻	opsertikey				
		DATETIME	DATAOBJECT_MODIFIED							
		VARCHAR	ProjectName	PROJECT_NAM	E 💌					
		VARCHAR	QueryName	QUERY_NAME						
		DECIMAL	GroupFieldDecimal 1							
		DECIMAL	GroupFieldDecimal2		-					
		VARCHAR	GroupFieldString1	T.productId						
		VARCHAR	GroupFieldString2		•					
		INT	GroupFieldInteger1		-					
		INT	GroupFieldInteger2		•					
		DECIMAL	TopNValue	T.MAXcallProce	ssingTim					
						OK				

Figure 18 – Setting up the StrTop5MaxCallProcessTime Query

Click the preview icon to see what the final query generated by the template looks like. In this case the CQL statement is as follows.

CREATE QUERY PatternMatch.StrTop5MaxCallProcessTime as SELECT T.MAXcallProcessingTime , T.productld , 'PatternMatch' AS PROJECT_NAME , 'StrTop5MaxCallProcessTime' AS QUERY_NAME FROM (RSTREAM(SELECT S.productld , S.MAXcallProcessingTime FROM (SELECT productld , MAX(callProcessingTime) AS MAXcallProcessingTime FROM PatternMatch.PATTERNMATCH_FACT_STREAM[RANGE 2 minute] AS Q GROUP BY Q.productld) AS S ORDER BY S.MAXcallProcessingTime DESC ROWS 5)) AS T destination

"combined:jms:queue/oracle.beam.cqservice.mdbs.alertengine:queuecf/oracle.beam.cqservice.m dbs.alertengine;jms:queue/oracle.beam.cqservice.mdbs.reportcache:queuecf/oracle.beam.cqserv ice.mdbs.reportcache:queuecf/oracle.beam.cqserv

Designing a Duplicate Detection Template

The goal is to design a query that will provide an output if more than one event arrives within two minutes of each other with the same CustomerLocationId, productId and callStatus. This is achieved using the Duplicate Detection Template. This template supports only Stream DOs and

does not currently support filters. The use of the rolling window is mandatory and the update interval field is not supported.

Over Stream DO (StrDuplicateDetection)

Data Object: PATTERNMATCH_FACT_STREAM

- Measure
 - ✓ Measure fields: customerLocationId, productId, callStatus
 - ✓ Use Rolling window: True; Range length: two minutes

Output

- ✓ Output All fields
- Action

Insert Event output into DupDetectionOutputDDO, and use the following mapping.

DataObject column	Event Output Field
ProjectName	PROJECT_NAME
QueryName	QUERY_NAME
GroupFieldString1	T.customerLocationId
GroupFieldString2	T.productId
MeasureString1	T.callStatus

StrDuplicateDetection v 2 🛛 🗸 Continuous Queries Save Λ Activate Continuous Queries : StrDupli ateDetect Template Duplicate Detection Template Description If duplicate values are detected for supplier ID and request ID with a rolling window of 60 seconds, output supplier ID, reqest ID and 'Duplicate Detection' 1. Measure 2. Output * Output 🔽 All * Data Object /orade/callcenter/PATTERNMATCH_FACT_STREAM 💽 🝓 ✓ callStatus * Measure Field customerLocationId CallPriority ✓ productId CallProcessingTime CallStatus CallWaitTime customerLocationId CustomerLocationName CustomerStatus Action productId Insert event output fields into DupDetectionOutputDDO **83** -✓ Use Rolling Window Map Fields × * Range Length 2 🗘 Minute 💌 Data Object -DupDetectionOutputDDO Operation Type insert -Define Mappings
Turne Data Object Column
COREAL Event Output Field Upsert Key DATETIME DATAOBJECT_CREATED • DATETIME DATAOBJECT_MODIFIED • VARCHAR ProjectName PROJECT NAME • VARCHAR QueryName QUERY NAME DECIMAL GroupFieldDecimal1 DECIMAL GroupFieldDecimal2 -VARCHAR GroupFieldString1 T.customerLocationId • T.productId VARCHAR GroupFieldString2 • INT GroupFieldInteger 1 -INT GroupFieldInteger2 • • • DECIMAL MeasureDecimal 1 DECIMAL MeasureDecimal2 T.callStatus VARCHAR MeasureString1 MeasureString2 -VARCHAR -INT MeasureInteger 1 INT MeasureInteger2 -ОК

Figure 19 – Setting up the StrDuplicateDetection Query

Click the preview icon to see what the final query generated by the template looks like. In this case the CQL statement is as follows.

CREATE QUERY PatternMatch.StrDuplicateDetection as SELECT T.callStatus ,

T.customerLocationId , T.productId , 'PatternMatch' AS PROJECT_NAME ,

'StrDuplicateDetection' AS QUERY_NAME FROM (ISTREAM(SELECT S.callStatus,

S.customerLocationId , S.productId FROM

PatternMatch.PATTERNMATCH_FACT_STREAM[NOW] AS

D,PatternMatch.PATTERNMATCH_FACT_STREAM[RANGE 2 minute] AS S WHERE

D.callStatus=S.callStatus AND D.customerLocationId=S.customerLocationId AND

D.productId=S.productId GROUP BY S.callStatus,S.customerLocationId,S.productId HAVING COUNT(*)>1) AS T destination

"combined:jms:queue/oracle.beam.cqservice.mdbs.alertengine:queuecf/oracle.beam.cqservice.m dbs.alertengine;jms:queue/oracle.beam.cqservice.mdbs.reportcache:queuecf/oracle.beam.cqserv ice.mdbs.reportcache?batch=true"

Continuous Queries Monitoring

You can use the Continuous Queries Monitoring screen to see Active queries. This is accessed from the Continuous Queries Monitoring link on the Administrator panel. This page gives you the following details:

- ✓ Name of the query
- ✓ Status
- ✓ Server on which the query is currently running and the complete CQL Statement.

Conti	ontinuous Queries Honitoring x							
Projec	t All	▼ Status All	•	Activate Query	Deactivate Query Drop Query @Refresh			
	Project	Query	Status	Server	Statement			
	PatternMatch	StrAvgCallWaitTimeGreaterThanThresh	Active	bam_server1	CREATE QUERY StrAvgCallWaitTimeGreaterThanThreshold as ISTREAM(SELECT AVG(callWaitTime) AS MEASURE , 100.0 AS NORM , 10.0 AS ALLOWED_DEVIATION , AVG			
	PatternMatch	StrTop5MaxCallProcessTime	Mactive	bam_server1	CREATE QUERY StrTop5MaxCallProcessTime as SELECT T.MAXcallProcessingTime , T.productId , 'PatternMatch' AS PROJECT_NAME , 'StrTop5MaxCallProcessTime' AS QU			
	PatternMatch	Top5MaximumCallProcessingTime	Active	bam_server1	CREATE QUERY Top5MaximumCallProcessingTime as SELECT T.MAXcallProcessingTime , T.productId , 'PatternMatch' AS PROJECT_NAME , 'Top5MaximumCallProcessingTi			
	PatternMatch	StrMisResumCallDetection	Mactive	bam_server1	CREATE QUERY StrMisResumCalDetection as SELECT T.A_callStatus , T.B_callStatus , T.A_productId , T.B_productId , 'PatternMatch' AS PROJECT_NAME , 'StrMisResum			
	PatternMatch	CallProcessingTimeTrendingUp10Percent	Mactive	bam_server1	CREATE QUERY CallProcessingTimeTrendingUp10Percent as SELECT T.callProcessingTime , T.endcallProcessingTime , T.productId , 'PatternMatch' AS PROJECT_NAME , 'C			
	PatternMatch	MissingResumeCallDetection	Mactive	bam_server1	CREATE QUERY MissingResumeCalDetection as SELECT T.A_calStatus , T.B_calStatus , T.A_productId , T.B_productId , 'PatternMatch' AS PROJECT_NAME , 'MissingResumeCalDetection as SELECT T.A_calStatus , T.B_calStatus , T.A_productId , T.B_productId , 'PatternMatch' AS PROJECT_NAME , 'MissingResumeCalDetection as SELECT T.A_calStatus , T.B_calStatus , T.A_productId , T.B_productId , 'PatternMatch' AS PROJECT_NAME , 'MissingResumeCalDetection as SELECT T.A_calStatus , T.B_calStatus , T.A_productId , T.B_productId , 'PatternMatch' AS PROJECT_NAME , 'MissingResumeCalDetection as SELECT T.A_calStatus , T.B_calStatus , T.A_productId , T.B_productId , 'PatternMatch' AS PROJECT_NAME , 'MissingResumeCalDetection as SELECT T.A_calStatus , T.B_calStatus , T.A_productId , T.B_productId , 'PatternMatch' AS PROJECT_NAME , 'MissingResumeCalDetection as SELECT T.A_calStatus , T.B_calStatus , T.A_productId , T.B_productId , 'PatternMatch' AS PROJECT_NAME , 'MissingResumeCalDetection as SELECT T.A_calStatus , T.B_calStatus , T.A_productId , T.B_productId , 'PatternMatch' AS PROJECT_NAME , 'MissingResumeCalDetection as SELECT T.A_calStatus , T.B_calStatus , T.A_productId , T.B_productId , 'PatternMatch' AS PROJECT_NAME , 'MissingResumeCalDetection as SELECT T.A_calStatus , T.B_calStatus , T.A_productId , T.B_productId , 'PatternMatch' AS PROJECT_NAME , 'MissingResumeCalDetection as SELECT T.A_calStatus , T.B_calStatus , T.A_productId , T.B_productId , 'PatternMatch' AS PROJECT_NAME , 'MissingResumeCalDetection as SELECT T.A_calStatus , T.B_calStatus , T.B_calStat			
	PatternMatch	AverageCallWaitingTimeGreateThanTh	Mactive	bam_server1	CREATE QUERY AverageCallWaitingTimeGreateThanThreshold as ISTREAM(SELECT AVG(callWaitTime) AS MEASURE , 100.0 AS NORM , 10.0 AS ALLOWED_DEVIATION ,			
	PatternMatch	StrMonitDupEvents	Mactive	bam_server1	CREATE QUERY StrMonitDupEvents as SELECT T.monitor_count , T.AGGcallStatus , T.customerLocationId , T.productId , 'PatternMatch' AS PROJECT_NAME , 'StrMonitDu			
	PatternMatch	StrMovinAvgCallProcessTime	Mactive	bam_server1	CREATE QUERY StrMovinAvgCallProcessTime as SELECT T.AVGCallProcessingTime , T.productid , 'PatternMatch' AS PROJECT_NAME , 'StrMovinAvgCallProcessTime' AS Q			
	PatternMatch	MovingAverageForCallProcessingTime	Mactive	bam_server1	CREATE QUERY MovingAverageForCalProcessingTime as SELECT T.AVGcalProcessingTime , T.productId , 'PatternMatch' AS PROJECT_NAME , 'MovingAverageForCalPro			
	PatternMatch	MonitorDuplicatedEvents	Mactive	bam_server1	CREATE QUERY MonitorDuplicatedEvents as SELECT T.monitor_count , T.AGGcallStatus , T.customerLocationId , T.productId , 'PatternMatch' AS PROJECT_NAME , 'Monit			
	PatternMatch	StrCallProcessingTimeTrendUp10PC	Mactive	bam_server1	CREATE QUERY StrCallProcessingTimeTrendUp10PC as SELECT T.callProcessingTime , T.endcallProcessingTime , T.productId , 'PatternMatch' AS PROJECT_NAME , 'StrCall			
	PatternMatch	StrDuplicateDetection	Mattive	bam_server1	CREATE QUERY StrDuplicateDetection as SELECT T.callStatus , T.customerLocationId , T.productId , 'PatternMatch' AS PROJECT_NAME , 'StrDuplicateDetection' AS QUE			
State	atement CREATE QUERY StrAvgCalWaitTimeGreaterThanThreshold as ISTREAM(SELECT AVG(calWaitTime) AS MEASURE ; 100.0 AS NORM, 10.0 AS ALOWED_DEVIATION , AVG(calWaitTime) - 100.0 AS ACTUAL_DEVIATION , productid AS productid , VatterMatch' AS PROJECT JVAVE ; StrAvgCalWaitTimeGreaterThanThreshold as ISTREAM(SELECT AVG(calWaitTime) AS MEASURE ; 100.0 AS ALOWED_DEVIATION , AVG(calWaitTime) - 100.0 AS ACTUAL_DEVIATION , productid AS productid AS productid , VatterMatch' AS PROJECT JVAVE ; StrAvgCalWaitTimeGreaterThanThreshold as QUERY jVAVE FROM PATTERNMATCH ; FACT_STREAM(RANCE 2 minute) AS T TVHERE (calIStatus="CLOSED") GROUP BY T_productid HAVING AVG(T.calIWaitTime) - 100.0 > 10.0) destination "combined;ins:queue/orade.beam.cqeervice.mdbs.reportcache:queuecf/orade.beam.cqueuecf/orade.beam.cqueuecf/orade.beam.cqueuecf/orade.beam.cqueuecf/orade.beam.cqueuecf/orade.beam.cqueuecf/orade.beam.cqueuecf/orade.beam.cqueuecf/orade.beam.cqueuecf/orade.beam.cqueuecf/orade.beam.cqueuecf/orade.beam.cqueuecf/orade.beam.cqueuecf/orade.beam.cqueuecf/orade.beam.cqueuecf/orade.beam.cqueuecf/orade.beam.cqueuecf/orade.beam.cqueuecf/orade							

Figure 20 – Continuous Queries Monitoring Screen.

Designing a Fired KPI Alert Events Query

The goal is to design a group query which will give the count of events fired by the CQL 'AverageCallWaitingTimeGreateThanThreshold' for each minute. The continuous query's output is constantly written back to KPIAlertOutputDDO, so this is achieved by creating a group query on KPIAlertOutputDDO with adequate filters on ProjectName and QueryName. The auto populated field 'DATAOBJECT_CREATED' is present in every DO and it will always contain the time at which the record was created.

Data Object: /oracle/writeback/KPIAlertOutputDDO

Measure (Y-axis)

• Norm(Count)

Dimensions (X-axis)

- DATAOBJECT_CREATED
 - ✓ Time Grouping: Use Time Series
 - ✓ Time Unit: Minute
 - ✓ Continuous time series: true
 - ✓ Quantity: 1

Filters

- Branch :All Are True
 - ✓ ProjectName is equal to 'PatternMatch'
 - ✓ QueryName is equal to 'AverageCallWaitingTimeGreateThanThreshold'

FiredKPIAlertEventsQuery ×					(2) 🕅 ·
lik (i 🕅 Save
* Data Object 🛛 /orade/writeback/KPIA 💌 🚻					
Measures (Y-axis)	~ v	Dimensions (X-axis)	/ & &	Filters	(); 🕂 🦯 🗶
		Image: State		All are true ProjectName is equal to 'Path QueryName is equal to 'Avera'	ernMatch" geCallWatingTimeGreateThanThreshold"
	Time Grouping Use Time Series * Time Unit Min * Quantity	Continuous Time Series	×		
	3		OK Cancel		
				8	

Figure 21 – Time Grouping Feature

You can design the other group queries by repeating the procedure outlined for the previous queries. Ensure that you select the appropriate Data Object and set the correct Query name in the filter condition "QueryName is equal to".

Designing Fired KPI Alert Output Events

The goal is to create an Active business view using a line chart based on the query FiredKPIAlertEventsQuery. The View must be an active view, which means it must show the data for the last 10 minutes, or any other specified time value, and keep updating automatically.

Business Query: FiredKpiAlertOutputEvents

Category: Line, View Type: Line

• Run Time-interaction

- ✓ Click on "Run time interaction" and click on Active Data tab.
- ✓ Check "Turn this query into a continuous query"
- ✓ Check "Use time window"
- ✓ Sliding Range based on DATAOBJECT_CREATED
- ✓ Range Length: 10 minutes
- ✓ Update interval: 1 second

This converts the Group SQL query into a continuous query with the range being 10 minutes and slide being one second.

FiredKpiAlertOutputEvents	×						
			* Query FiredKPIAlertEventsQu	િય	🗠 Chart Switcher	Properties	Runtime-Interaction
FiredKpiAlertOutputEvents							
8	Runtime-Interactio	20		×			
7	Drilling Actions Active Data	Turn this query into a continuous query Carton and the second s					
6		Vuse a time window Sliding Range Based on DATAOBJECT_CREATE					
5		Range Length 10 minutes Update Interval 1 seconds					
4							
3							
2	2		Apply	Close			

Figure 22 – Setting up the FiredKpiAlertOutputEvents Query

You can design all business views using the aforementioned procedure. Ensure that you select appropriate query names from the drop down.

Understanding Dashboards

PatternMatch Dashboard

The goal is to create a dashboard with six views to show the output rate of all Continuous Queries based on the PatternMatch Logical DO. From the PatternMatch Project, create a Dashboard using the "Type 8" Style template as shown in Figure 23.



Figure 23 – Creating a PatternMatch Dashboard

Then, drag and drop the following views into one of the cells in the dashboard as shown in Figure 24.

- FiredKpiAlertOutputEvents
- FiredMissingEvents
- FiredMonitorCountEvents
- FiredMovingAggregationEvents
- FiredTrendingEvents
- FiredTopNEvents



Figure 24 – PatternMatch Dashboard with Various Business Views

The business views are now added to PatternMatch Dashboard.

StrPatternMatch Dashboard

You can create a strPatternMatch Dashboard using the procedure outlined for creating the PatternMatch Dashboard, and add the following views to it.

- ✓ strFiredKpiAlertOutputEvents
- ✓ strFiredMissingEvents
- ✓ strFiredMonitorCountEvents
- ✓ strFiredMovingAggregationEvents
- ✓ strFiredTopNEvents
- ✓ strFiredTrendingEvents

Add an extra cell in the bottom row to accommodate the additional strFiredDuplicateDetectionEvents

strFiredTopNEvents	i> i> i> strFiredTrendingEvents	Image: Second
8,	1.2	Add New Cell Right
7		Add New Cell Below
6	1.0	Add new Cell Up
5	0.8	🖽 Change Width
4	0.6	I Change Height
2	0.0	🥔 Clear
5	0.4	X Delete the cell
	0.2	
0		

Figure 25 – Adding a New Cell to accommodate the additional strFiredDuplicateDetectionEvents.



Figure 26 – The strPatternMatch Dashboard

Troubleshooting

- 1. If your artifacts are not imported:
 - Verify that samples/common/setEnv.sh has correct values for all parameters
 - Verify that the correct password is specified in <password></password> tag of bin/ BAMCommandConfig.xml
 - Check for any specific errors in console or in log files that are generated in the bin directory, for example, bamcommand.log.*
- 2. If you get errors while populating data:

- Verify that samples/common/setEnv.sh has correct values for all parameters
- 3. If the dashboard is not showing any data:
 - In case of PatternMatch Dashboard not showing any data, check if the PATTERNMATCH_DIMENSION DO has the following data:
 - customerlocationId,customerLocationName US, United States CA, CANADA

The PATTERNMATCH_FACT DO is populated with data to ensure that the dimension side is not empty. This ensures that the PatternMatch Logical DO is populated with data, thereby satisfying the join condition. If the two aforementioned rows are not present in the dimension DO, you must add them manually and try again.

• Ensure that the writeback DOs are populated with new data every minute. If not, open the Continuous Queries Monitoring screen and check if all the 13 CQL Template queries are Active. If not, navigate to the designer, open each query individually, and save it again with the "Activate Continuous Query" checkbox checked

Best Practices

For queries over Archived Relation DOs (or Derived or Logical DOs built on top of Archived Relations), depending on your expected input event rates, ensure that you specify an appropriate "Range Length". If you specify a large value, those many events must be fetched from the database which can result in serious performance implications. In such cases, the "use rolling window" option is checked by default and you cannot change it. This avoids an overload in fetching records from the database while starting the query.

blogs.oracle.com/oracle

facebook.com/oracle

twitter.com/



CONNECT WITH US

B blogs.oracle.com/oracle

facebook.com/oracle

twitter.com/oracle

oracle.com

Oracle Corporation, World Headquarters 500 Oracle Parkway Redwood Shores, CA 94065, USA Worldwide Inquiries Phone: +1.650.506.7000 Fax: +1.650.506.7200

Hardware and Software, Engineered to Work Together

Copyright © 2014, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0914

igsimed igsimed