Oracle Financial Services Crime and Compliance Studio

User Guide

Release 8.0.8.0.0

September 2020

E91246-01

OFS Crime and Compliance Studio User Guide

Document Control

Version Number	Revision Date	Change Log
Version 80800	August 2020	User Guide Revamp

Table of Contents

1 Preface

This guide provides information related to end-user tasks in the Oracle Financial Services (OFS) Crime and Compliance Studio (FCC Studio) application.

1.1 Audience

This guide is intended for Data Analysts and Data Scientists to create notebooks for analysis and models to be run against FCCM and other data for a number of different use case using these langauges and creating notebooks including graph paragraphs and templates for non-technical users. The basic knowledge of the following is recommended:

- Database concepts
- Big Data
- Python
- Scala
- Spark
- Oracle R
- SQL
- PGQL or PGX-Java
- Groovy
- Markdown

1.2 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website.

1.3 Related Documents

You can access the following additional documents related to the OFS Crime and Compliance Studio application from the Oracle Help Center (OHC) Documentation Library:

- Oracle Financial Services Crime and Compliance Studio Installation Guide
- Oracle Financial Services Crime and Compliance Studio Deployment Guide (Using Kubernetes)
- Oracle Financial Services Crime and Compliance Studio Administration Guide
- Oracle Financial Services Crime and Compliance Studio Data Model Guides
- Oracle Financial Services Crime and Compliance Studio Release Notes and Readme

1.4 Abbreviations

The following table lists the abbreviations used in this document.

Table 1: Abbreviations

Abbreviation	Meaning
OFS	Oracle Financial Services
FCC Studio	Financial Crime and Compliance Studio
OFSAA	Oracle Financial Services Analytical Application
PGX	Parallel Graph AnalytiX
AML	Anti-money Laundering
FCDM	Financial Crime Data Model
BD	Behavior Detection
OOB	Out-of-the-Box

2 Introduction to Financial Crime and Compliance Studio

The Financial Crime and Compliance (FCC) Studio uses the latest technology to harness the power of Graph Analytics to give Financial Institutions the ability to effectively monitor anti-money laundering and anti-fraud programs in financial institutions.

By using the FCC Studio, financial institutions can deploy advanced machine learning algorithms and artificial intelligence to discover unknown criminals, streamline compliance operations and have unprecedented control over their Financial Crime program.

FCC Studio helps financial institutions to overcome the following challenges:

- Fraud programs in financial institutions.
- Identify and adapt to the changing patterns of financial crime.
- Ability to discover new and emerging criminal behavioral patterns.
- Ability of institutions to gain insights on the financial crime data with new and emerging financial crime patterns and trends.

2.1 Solutions offered by FCC Studio

The FCC Studio is a platform that can be used by Data Scientists and Data Analysts for perform the FCC analysis.

The following are the solutions offered by FCC Studio:

- **Machine Learning**

  Out-of-the-box notebooks which are language agnostics and supports SparkML, ORE, R, Python, Deep Learning, On Graph, Case Similarity, Predictive Behavioral Models, AAS, and so on for your Machine Learning requirements.

- **Graph Analytics**

  Provides graph analytics such as, Pattern Matching, Graph Algorithm, Case Creation, Graph Based Investigation, PGX, and Spark.

- **Data Discovery**

  Provides out of the box notebooks that can be used for discovering the data. Also, supports Graph Pattern Discovery, Data Visualization, and Graph Questions.

- **Rapid Development**

  Provides REST interface, widgets notebooks in batch, and consumerization data science.

- **Modeling**

  Provides you the flexibility to create and import graph models and visualize the outputs.

- **Explainability**

  The intuitive Interface of the FCC Studio helps you to learn and adapt to the Interface with less training and explaination.

- **Leverage fragmented data**

  Replicates the Oracle Financial Crime Data Model (Graph and Relational) which are on the data lake and the ability to connect with any of the data set.

- **Collaboration**

  Provides an ability to collaborate and share notebooks to improve the productivity.

2.2 Benefits of FCC Studio

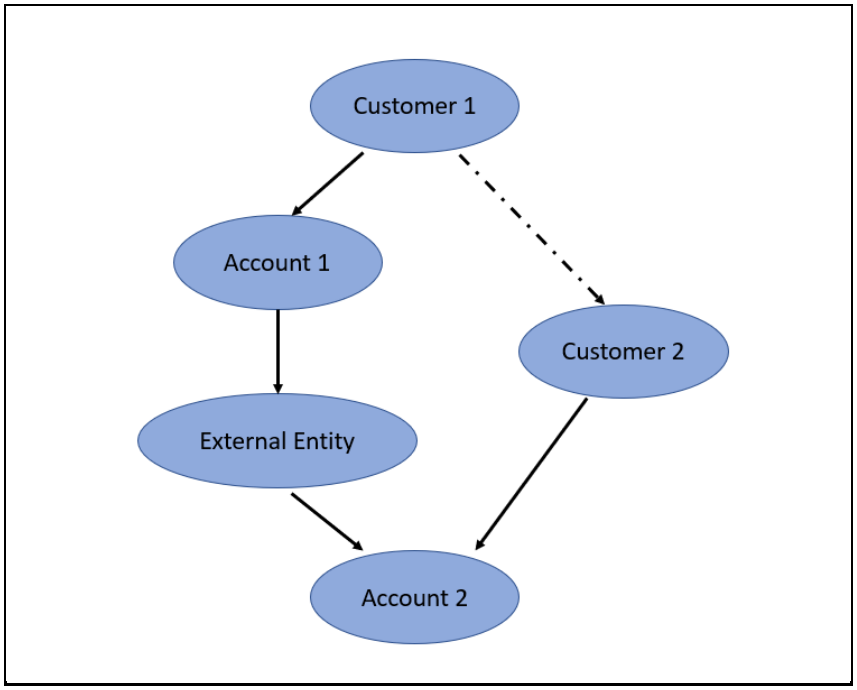
The Financial information stored in tables in the relational database are in tables and provides the users with many tools to access, manipulate, and combine the information. With graphs, data can be managed in more intuitive ways closer to how you can organize your thoughts on a whiteboard.

FCC Studio takes the advantage of parallel processing and huge amount of information and memory available in the modern servers, this allows studio to directly model the relationships between all the data and process these data into graphs or network that can be visualized in FCC Studio and can consume these graphs to look for alerts that are generated by financial systems and see how it can be combined together into sets of cases and then prosecuted more efficiently.

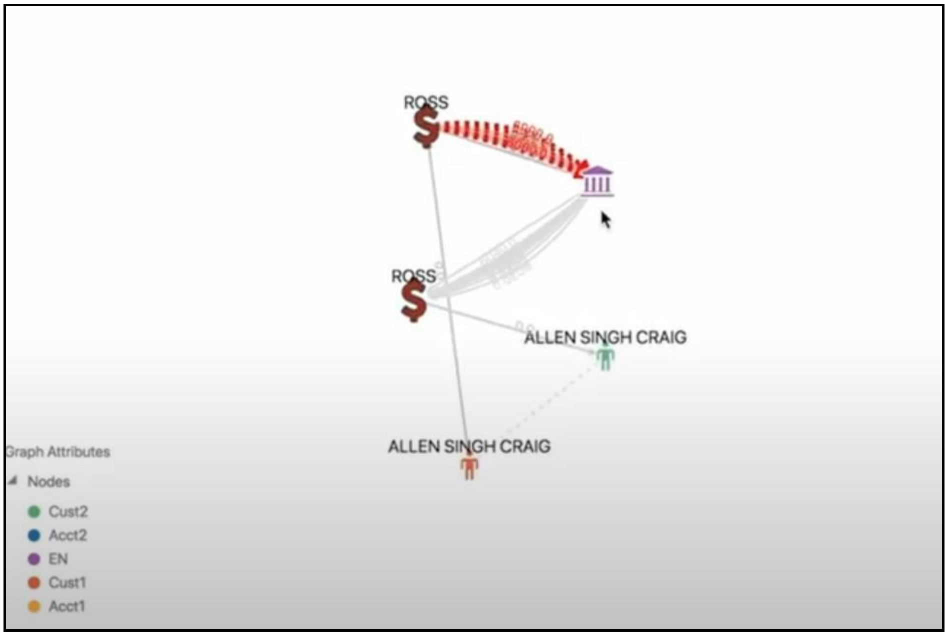
2.2.1.1 Example

Let us take an example of a customer, where the customer wires money through an external entity from one account to another account which belongs to another customer. However, two customers are in fact related in a chain of shared information. Where Customer A share the same e-mail address with Customer B who shares password with Customer C who shares ground address with Customer D, and so on.

The following figure illustrates such a graph pattern. In the figure, the dotted line represents a path (Account and the Customer are related to the common email address).



This pattern can be processed, and the results can be visualized. The following figure shows the graph pattern that is obtained when running the Property Graph Query Language (PGQL) for the previously illustrated scenario.



The insights that are provided by the graph enables you to look at the alerts that are generated by financial systems that banks could have and see how it can be combined to sets of cases and then prosecuted more efficiently. The Investigator, instead of looking at each alert or case individually, can combine the insights into a small network then investigate.

2.3 Key Features

The key features of FCC Studio include the following:

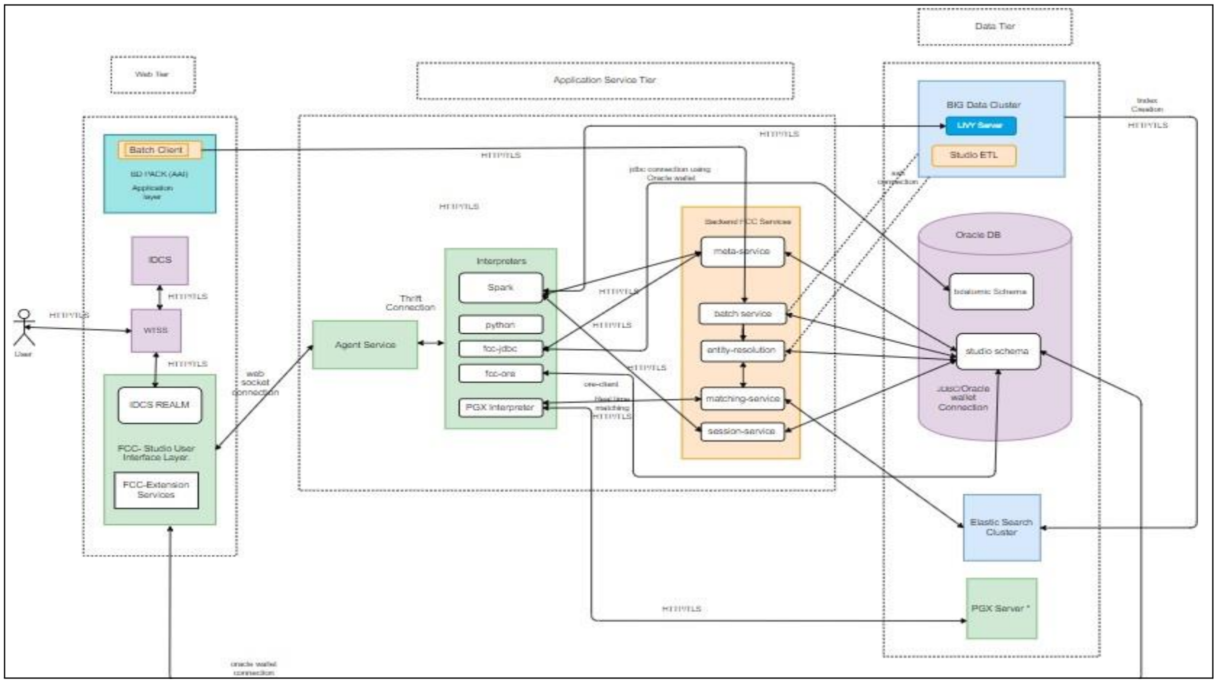
- Provides an integrated and comprehensive analytics toolkit designed to rapidly discover and model new financial crime patterns.
- Interacts with the database, processes the data, and generates patterns in various formats using interpreters.
- Provides secure access to an institution's financial crime data with predefined scenarios, out-of-the-box graph queries, and visualizations.
- Uses Graph Analytics and Graph Query methods to analyze historic data available in the database, and forecast the generated patterns using various interpreters.
- Uses Machine Learning Algorithms to gain insights from historical alert data to prioritize the alerts generated by the detection engines.
- Offers a unified tool for Graph Analytics, Data Visualization, Machine Learning, Scenario Authoring, Pattern Discovery, Data Mashups. and testing for financial crime data.
- Works with Apache Spark, the most prevalent analytics engine on Big Data.
- Works with Apache Zeppelin, a web-based notebook that enables interactive data analysis.
- Supports Polyglot Scenario Authoring to author new scenarios in SQL, Scala, Python, or R language.
- Embedded with a highly scalable in-memory Graph Analytics Engine (PGX).
- Enterprise-ready with underlying OFSAA framework.
- Works with earlier 8.x releases of Oracle Financial Crime and Compliance Management Anti Money Laundering (AML) and Fraud applications.

- Integrates with Oracle Financial Crime Application Data and readily usable across the enterprise financial crime data lake. This can automatically load Oracle AML and Fraud data into the data lake and mashup FCC Studio data with third-party data for discovery and modeling.

2.4 The Architecture of FCC Studio

The following diagram depicts the architecture of the FCC Studio application.

Figure 1: FCC Studio Architecture



The Architecture of the FCC Studio is built upon a secured Behavior Detection (BD) or Financial Crime and Compliance Management (FCCM) platform to offer the ability to build notebooks based on the data available on the platform and run these notebooks to built analytical model for your investigation.

FCC studio is used with oracle's behavior detection or other FCC product there are pre-built integrations for scenario authoring and creation of events in our enterprise case management but FCC studio can be used with any financial crime platform. It offers an enriched graph based analytics service with the capability of intergration with Oracle PGX. The data that is loaded to the plaform is moved to Studio through ETL and this data is sent to the PGX servers to compile and return a graph based analytics.

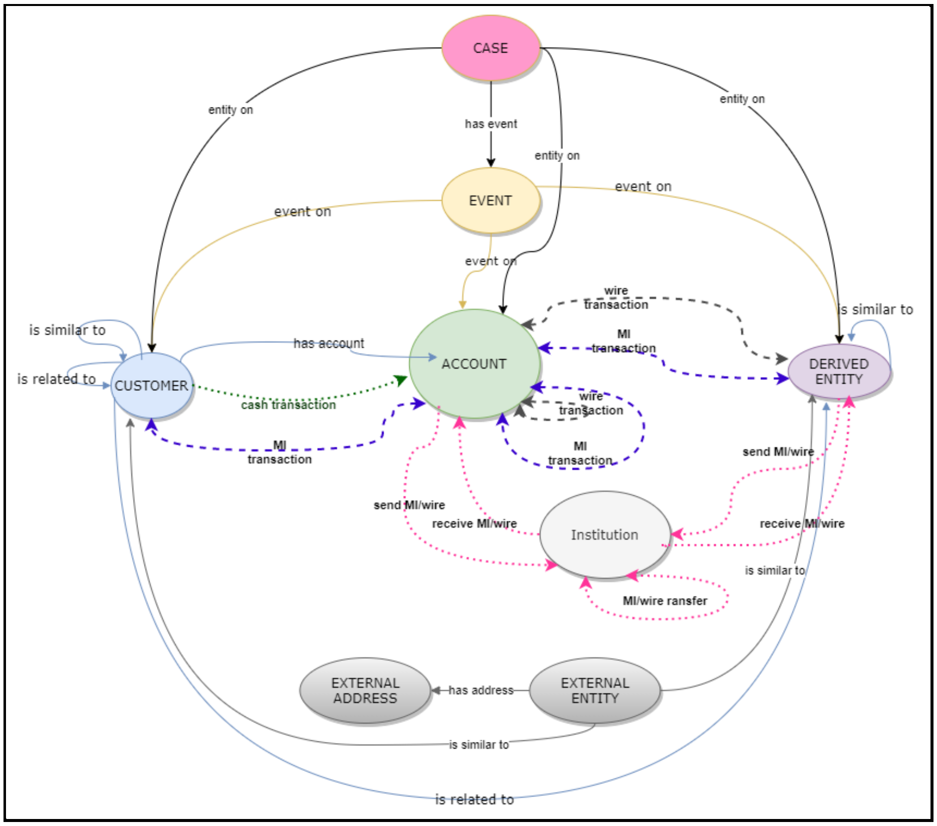
2.5 Oracle Financial Crime Graph Model

The Oracle Financial Crime Graph Model serves as a window into the financial crimes data lake. It collates disparate data sets into an enterprise-wide global graph, enabling a whole new set of financial crime use cases. The Graph model enables to accelerate financial crime investigation use cases.

For information on the Graph Data Model, see Graph Data Model.

For information on the node and edge properties of the Oracle Financial Crime Graph Model, see the Data Model Guides.

Figure 2: Oracle Financial Crime Graph Model



NOTE

The Case node in this Financial Crime Graph Model is loaded only when you load the FCDM data from Enterprise Case Management (ECM). When data is loaded from ECM, the graph includes “CASE” nodes and “has event” edges.

3 Getting Started

The web-based notebook in FCC Studio provides users the ability to combine live code collaboration in multiple programming languages with graph analytics and rich, interactive visualizations. Data Studio accelerates the process of exploring and gaining insights from your data.

Data can be imported to the FCC Studio from various sources (from HDFS or Spark, databases, or files) and analyzed with interpreter environments for a range of programming languages (Python, R, Shell, Spark, and so on).

For financial transactions graph data, FCC Studio comes packaged with Oracle Labs' graph analytics tool (PGX) and property graph query language (PGQL), adding an interactive visual layer that supports options such as:

- Filtering graphs
- Highlighting elements
- Visualizing geographical data, and expanding or contracting the view
- Enabling users to explore large graphs intuitively

The FCC Data Studio components form a reusable base for enterprise software products tailored to specific industries. Example use cases include financial crime detection and compliance, machine learning for health sciences, and market segmentation for retail. It provides you the ability to build notebook to analyze you use cases that are associated with the financial transactions.

To get started using the FCC Studio, see the following topics:

- Accessing the FCC Studio Application
- FCC Studio Workspace
- About Notebooks
- About Graphs
- About Templates
- About Interpreters
- About Tasks
- About Permissions
- About Ruleset
- About Manual Decisioning

To know about the common use cases in FCC Studio and how you can build your use case based on the available use cases, see the Use Cases chapter.

3.1 Accessing the FCC Studio Application

To access the FCC Studio application, follow these steps:

1. Specify the URL in the following format in the web browser:

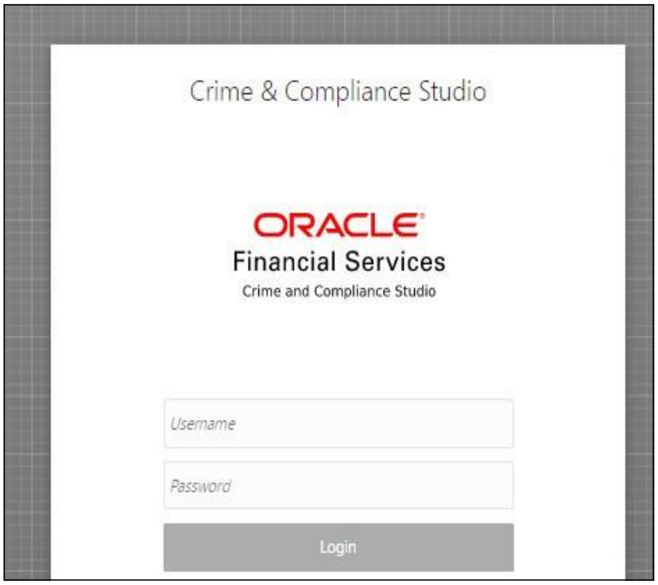
   ```
   https://<Host_Name>:<Port_Number>
   ```

   Use the port number as follows:

- 7008 for FCC Studio installed on-premise.
- 30078 for FCC Studio deployed on the Kubernetes cluster.

The FCC Studio application login page is displayed.

Figure 3: FCC Studio Login Page



Crime & Compliance Studio

ORACLE
Financial Services
Crime and Compliance Studio

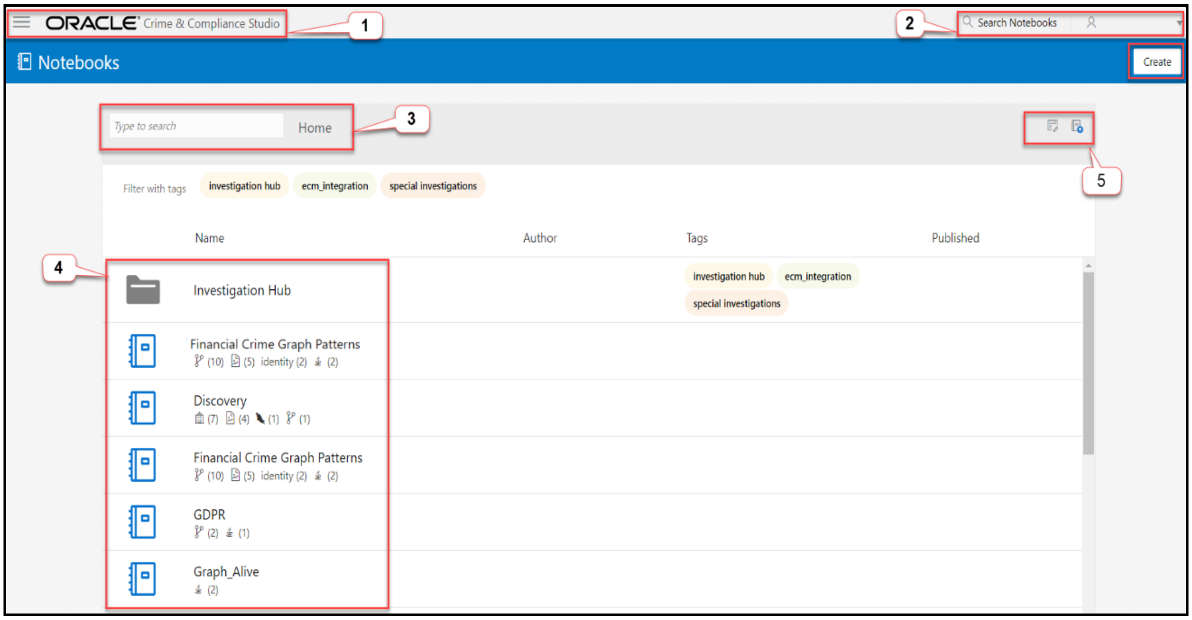
2. Log in with the Username and Password.
3. Click **Login**.The FCC Studio Workspace is displayed.

3.2 FCC Studio Workspace

The FCC Studio Workspace displays the Out of the Box Notebooks for your financial use cases and analysis. It also provides you functionalities such as, Import, Export, Create, and Delete Notebooks.

In the Workspace, you can specify the details to filter and search for Notebooks and then perform the related actions such as, Import, Export, Create, and Delete.

Figure 4: FCC Studio Workspace



Name	Author	Tags	Published
Investigation Hub		investigation hub ecm_integration special investigations	
Financial Crime Graph Patterns (10) (5) identity (2) (2)			
Discovery (7) (4) (1) (1)			
Financial Crime Graph Patterns (10) (5) identity (2) (2)			
GDPR (2) (1)			
Graph_Alive (2)			

The following table describes the call-outs that are illustrated in the Workspace figure.

Table 1: Out-of-the-Box Notebooks in FCC Studio

Call-outs	Description
1	**Home and Menu Navigation icon**: When you click the Home and Menu Navigation options, the available menu items are listed as follows: • Notebooks • Graphs • Templates • Interpreters • Tasks • Permissions • Ruleset • Manual Decisioning **NOTE**: If you are not an Administration user, you will not have the items such as, **Interpreters**, **Tasks**, **Permission**, and **Ruleset**.
2	**Search Notebooks**: To quickly search your notebook, specify the details in the Search Notebooks field. For more information, see Searching your Notebook on the Workspace.

3	**Filter Notebook**: You can do a quick filter and search your notebook that are available in the FCC Studio Workspace. For more information, see Filtering your Notebook
4	**Notebooks**: The list of notebooks that are available out of the box and the notebooks that you have created or imported are also listed. For more information, see About Notebooks.
5	**Quick Actions**: To quickly do a batch, import, export, and delete operations, you can choose the quick actions in the Workspace. For more information, see Quick Actions

3.2.1 Searching your Notebook on the Workspace

The notebooks that are available out of the box and the ones that are created and managed by you, can be quickly searched and drilled down to access the notebook and start your analysis.

The notebooks can be searched by using the following options:

- **Search Notebooks**

  Specify the details of your notebook in this field to quickly access the notebook.

- **Type to Search**

  The **Type to Search** works based on the filters that you have used when creating your notebook. You can specify the filter detials to access the notebook. For more information about filtering, see Filtering your Notebook.

3.2.2 Filtering your Notebook

You can do a quick filter and search your notebook that are available in the FCC Studio Workspace. Following are the ways you can filter and search your notebooks:

- **Filter by Name**

  Using a text input field labeled as Type to search on the left of the main workspace toolbar, you can search for notebooks by name.

- **Filter by Tags**

  If you have added any tags to the notebook, while creating or updating it, you can use the tags to narrow the filter by clicking to one or more colored tags, above notebook list.

3.2.3 Quick Actions

The quick actions available in the Workspace are two functionalities, Select Notebooks and Import Notebooks.

3.2.3.1 Select Notebooks

To select notebooks and perform actions on them, click **Select Notebooks** on the top of the workspace-view.

Notebooks and directories can be selected and actions, such as Export Notebooks and Delete Notebooks can be applied.

- **Export Notebooks**

  Executing this option exports the notebooks to the specified location.

- **Delete Notebooks**

  Executing this option deletes the selected notebooks.

> **NOTE** If a directory is deleted, then all the notebooks that are related to the directory are also deleted.

3.2.3.2 Import Notebooks

If you like to import some notebooks into your workspace, click **Import Notebooks** and drag and drop or select FCC Studio Notebook, Zeppelin or Jupyter format files from your computer.

3.2.3.3 Export All Notebooks

You can also save all Notebooks to your computer and import them later or on another instance. To do so, click **Export All** Notebook button.

3.2.3.4 Open as an iframe Notebook¶

From the workspace view, notebooks can be directly opened in an iframe view. Hover over a notebook displays an Open as iframe option. You can click the option, which will further open a window with the following iframe settings, where you can customize your notebook view:

- Show Notebook toolbar
- Show add-paragraph actions
- Show paragraph actions
- Show paragraph code

By default, these settings are enabled and saved locally, and are applied each time the iframe settings dialog is opened. This behavior can be disabled by deselecting the Save Settings checkbox in the iframe window.

3.3 About Notebooks

Notebooks are used by data scientists and data analysts to quickly explore the scenario and take actions based on the graphical insights. It offers several advantages over any local scripts or tools. When properly set up by the organization, a notebook offers direct connections to all necessary sources of data, without additional effort from the user.

A notebook segments a computation in individual steps called paragraphs. These paragraphs contain an input and an output section. Each paragraph executes separately and modifies the global state of the notebook.

3.3.1 Advantages of Using the Notebooks

The following are the advantages of using the notebooks:

- Notebooks increase productivity, by facilitating incremental improvement. It is easy to use, rerun, and modify only the relevant paragraph.
- Users can develop a notebook and then iterate the notebooks based on their needs.
- Notebooks data are stored in the working memory.
- Notebooks offer advanced interactive visualizations. These range from simple line charts, bar graphs to maps, and custom visualizations that are useful for the data scientists to consume the insights.

3.3.2 Out of the Box Notebook in FCC Studio

The FCC Studio has a collection of Out-of-the-Box (OOB) Notebooks that are packaged along with the license. Following are the Notebooks that are available in FCC Studio:

- Finacial Crime Graph Patterns
- HRG Scenario – AC Focus
- RMF Account (SQL)
- RMF Scenario – AC Focus
- RMF Scenario – CUS Focus

3.3.3 Managing your Notebooks

A notebook is a collection of paragraphs and acts as a container to hold one or more paragraphs. Each notebook is a collection of documentation and snippets of executable code. The notebook allows large scripts to be broken into a modular collection of executable code with tailored results. Different languages, such as Groovy, Scala, Python, and Oracle's own property graph query language (PGQL), can be combined into one notebook. Each notebook is mapped to the role of the logged-in user.

You can view the following topics to manage your notebooks:

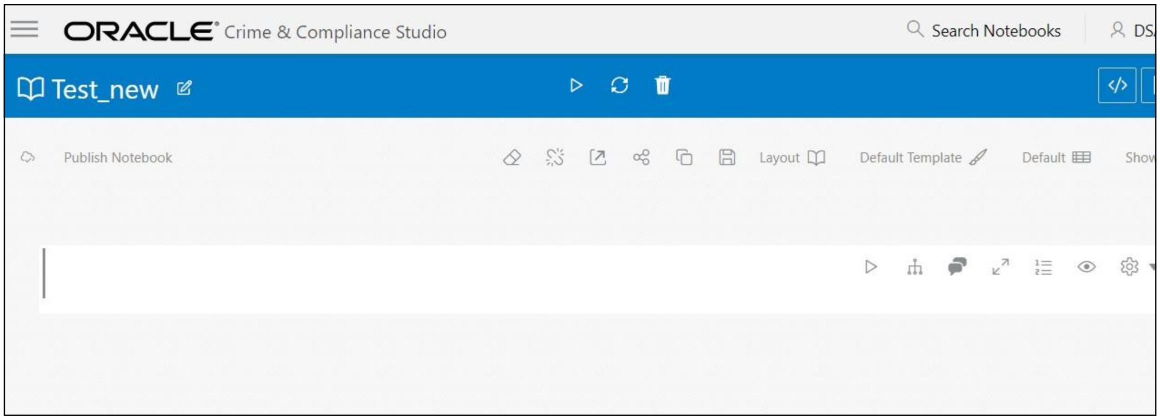
- Creating a Notebook
- Importing a Notebook
- Exporting a Notebook
- Deleting a Notebook

- Common Screen Elements in a Notebook
- Publishing your Notebooks
- Versioning your Notebook

3.3.3.1 Creating a Notebook

To get started with your analysis, you need to create a notebook. It allows large scripts to be broken into a modular collection of executable code with tailored results. To create a notebook, follow these steps:

1. Navigate to the FCC Studio workspace.
2. Click **Create** in the upper-right corner. The **Create Notebook** dialog box is displayed.
3. Specify the notebook details, such as:
   - **Name**
   - **Description**
   - **Tags**
   - **Type**
4. Click **Create**. A new notebook is created and listed in the FCC Studio Workspace.



To know more about the actions that can be performed in a notebook, see Common Screen Elements in a Notebook.

After creating a notebook, you can create Paragraphs in a notebook. For information on creating paragraphs, see Creating a Paragraph.

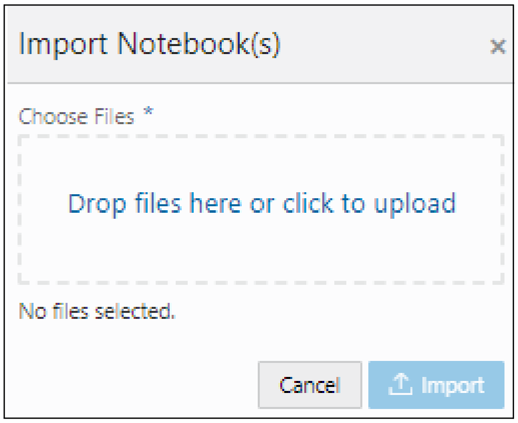
3.3.3.2 Importing a Notebook

The Import Notebook feature enables you to import notebooks into your FCC Studio workspace. The file format supported for import is Data Studio Notebook (*.dsnb).

To import notebook (or notebooks) into the FCC Studio workspace, follow these steps:

1. Navigate to the FCC Studio Workspace.

2. Click **Import Notebooks** on the upper-right corner. The **Import Notebook(s)** dialog box is displayed.



No files selected.

3. Click **Choose Files** to browse and select the files from your local machine or drag and drop the required files to upload.

   A message is displayed at the bottom of the Import Notebook dialog box to indicate if the selected files are valid for import.

4. Click **Import**. The notebooks are imported into the FCC Studio workspace.

3.3.3.3 Exporting a Notebook

The Export Notebook feature enables you to export notebooks from the FCC Studio workspace to your local machine. Notebooks are exported in the Data Studio Notebook (*.dsnb) format and are copied to the Downloads directory in the local machine. These files can be shared or imported.

You can either choose to export all the notebooks or export individual notebooks.

- Exporting All Notebooks
- Exporting Individual Notebooks

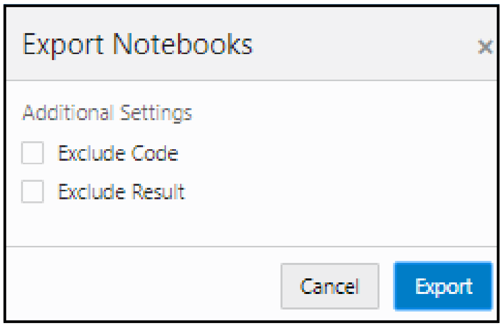
3.3.3.3.1 Exporting All Notebooks

You can export all your notebooks on a single go by using the **Export All Notebooks** option from the FCC Studio workspace to your local machine.

To export all the notebooks, follow these steps:

1. Navigate to the FCC Studio workspace.
2. Click **Export Notebooks** on the upper-right corner. The **Export Notebooks** dialog box is displayed.

Figure 18: Export Notebooks



Export Notebooks

Additional Settings

□ Exclude Code

□ Exclude Result

3. Select the required **Additional Settings** in the **Export Notebooks** dialog box and click **Export**. All the notebooks in the FCC Studio workspace are exported to your computer.

3.3.3.3.2 Exporting Individual Notebook

You can export your notebook by using the Export Notebook option from the FCC Studio workspace to your local machine.

To export an individual notebook, follow these steps:

1. Navigate to the FCC Studio workspace.
2. Select the required notebook to export. The selected notebook is opened.
3. Click **Export Notebook**. The notebook is exported to your computer.

3.3.3.4 Deleting a Notebook

You can delete the notebooks that are no longer required for your analysis. To delete a notebook, follow these steps:

1. Navigate to the FCC Studio workspace.
2. Click **Select Notebooks**.The check boxes are displayed for each notebook.
3. Select the required notebook (or notebooks) and click **Delete**. The selected notebook (notebooks) are deleted.

3.3.3.5 Common Screen Elements in a Notebook

The following table describes the common screen elements in a notebook that can be used to perform quick actions when preparing and executing the notebooks.

Table: Common Screen Elements in a Notebook

Icon	Action or Description
✎	Modify Notebook: Modifies the details of a notebook, such as the name, description, and (or) tags.
</>	Hide Code: Hides or shows the Code Section in all the paragraphs in a notebook.

🗎	Hide Result: Hides or shows the Results Section in all the paragraphs in a notebook.
🔓	Read-Only: Sets the notebook to read-only mode. **NOTE**:The notebook is protected from edit, clear result, delete, share, reset session, and run paragraphs in Read-only mode.
🔒	Write: Sets the notebook to write mode.
▷	Run Paragraphs: Executes all the paragraphs in a notebook in sequential order. For more information, see Run All Paragraphs.
🔄	Reset Session: Resets any connection or code executed in a notebook.
🗑	Delete Notebook: Deletes a notebook.
⌫	Clear Result: Clears results for all the paragraphs in a notebook. **NOTE**: This action clears all the results. You must run the paragraphs again to view the results.
⛓	Clear Paragraph Dependencies: Remove all defined paragraph dependencies.
↗	Open as an iframe: Opens a notebook in an Iframe. This allows a notebook to be embedded inside another webpage.
⧉	Clone Notebook: Creates a copy of a notebook. All paragraphs in the current notebook are replicated in the new notebook. The cloned notebook is created with the default name, Copy of <Current Notebook Name>.
💾	Save Notebook: Saves the notebook to your computer as a DNSB file. For more information, see Exporting Individual Notebooks.
Layout 🕮	Layout: Sets the preferred layout: Zeppelin or Jupyter.
Default Template ✎	Default Template: Applies the overall look and feel of the notebook using the default template.
Default ▤	Default View: Switches between Default, Simple, and Report views.

	Show Panel: Shows or hides the Paragraph Settings Bar Commands, Results Toolbar, and Settings Dialog for a selected paragraph in a panel to the right of the notebook.
	Versioning: You can use this option to create versions for your notebook, which helps you analyze the changes based on the version control.

This table describes the common screen elements in a notebook that can be used to perform various actions on a notebook such as modify notebook details, hide or show code, hide or show the result, set the notebook to read-only or write mode, delete a notebook, and so on.

3.3.3.6 Publishing your Notebooks

Notebooks can be published from the Notebook interface. Only one notebook can be in a published state for a given notebook-ID. If another version of a notebook is published, the previous published version is replaced.

When a notebook is published:

- The original notebook is cloned, and a published notebook is created.
- Any changes made to the original notebook will have no impact on the published notebook.
- Whenever the original notebook is republished, a new version of the published notebook is created.
- The published notebook is in a read-only format.
- The published notebook can be run in a batch pipeline.

The notebook toolbar is extended by following conditional buttons after publishing the notebook:

- **Publish**

  Clicking this button will publish the given notebook. The published notebook is automatically loaded.

- **Published Notebook**

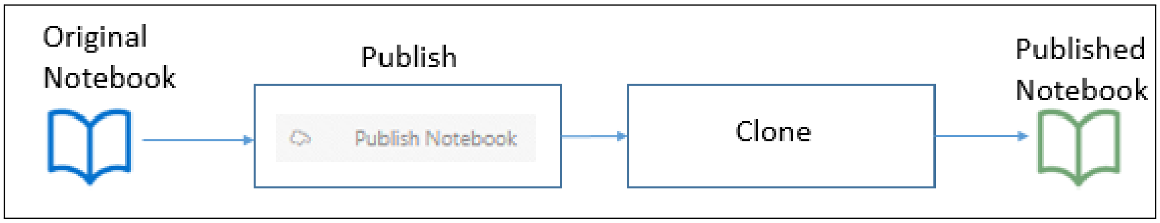
  If the notebook has a published version, this button appears and on click, will redirect you to the published notebook.

The workspace View shows an additional Published column that reflects whether a notebook is published.

> **NOTE** Published notebooks do not appear in the overview. Published notebooks can only be seen by either knowing the direct link or navigating from a notebook that has a published notebook.

3.3.3.6.1 Publishing a Non-scenario Notebook

The non-scenario notebook is of Notebook Type: Default or Jupyter. Upon publishing an original notebook, you must select the user role or group to which the notebook will be published. After publishing, the original notebook is cloned, and a published notebook is created.



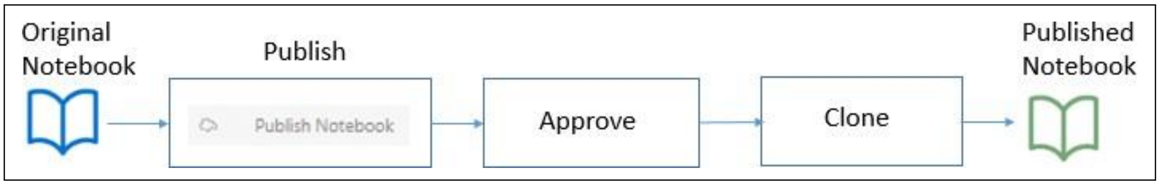
To publish a non-scenario notebook, follow these steps:

1. Navigate to a **Scenario Notebook** details page.
2. Click **Publish Notebook** on the top left corner. The **Publish Notebook** dialog box is displayed.
3. Click **Publish**.

   The notebook is published, and a confirmation message is displayed to indicate that the notebook is published.

3.3.3.6.2 Publishing a Scenario Notebook

The scenario notebook is of Notebook Type: Scenario. For scenario notebooks, the publish functionality provides four eyes approval process.



When a scenario notebook is published, the published notebook is shared with the user who is mapped to the **DSBATCHGRP** group for approval.

When a notebook is approved by the batch user:

- The original notebook is cloned, and a published notebook is created.
- Scenario metadata is created.
- A threshold set is created with values for all the parameters that you have provided in the notebook.

The approved scenario notebook can be executed with any threshold set which has been created while the notebook is being approved by the batch user. Whenever a user wants to create a new threshold set, the notebook must be republished with different threshold values. After a notebook is approved, a new threshold set is created for the same scenario.

To perform batch execution of the published scenario notebook, follow these steps:

1. Navigate to the `<Studio_home>/ficdb/bin` directory.
2. Run from the terminal using the following:

```
./FCCM_Studio_NotebookExecution.sh "notebookID" "null" "scenarioID"
"thresstoresetID" "null"
```

This section includes the following:

- Publishing a Notebook
- Approving a Notebook

3.3.3.6.3 Publishing a Notebook

After the notebook is apporved, you can publish that notebook for your organziation to use it.To publish a scenario notebook, follow these steps:

1. Navigate to a **Scenario Notebook** page.
2. Click **Publish Notebook** on the top left corner.

   The **Publish for Approval** dialog box is displayed with the Parameter Keys and the corresponding Parameter values added to the paragraphs.

> **NOTE** Ensure that the parameters with the same name must have the same values.

3. Click **Publish**.

   The Scenario Notebook is published for approval and listed in the Notebooks page with the *For Approval* tag. The published scenario notebook is shared to the user mapped to the DSBATCHGRP group in OFSAA, with the *For Approva*l tag.

3.3.3.6.4 Approving a Notebook

The notebook that is published must be approved by the group administrator.To approve a notebook, follow these steps:

1. Log in to the FCC Studio application as a DSBATCHGRP user.
2. Navigate to the **Scenario Notebook** page, that you want to approve.
3. Click **Approve Notebook** on the top left corner. The Approve Notebook dialog box is displayed.
4. Click **Approve**.

   A confirmation message is displayed to indicate that the notebook is approved. The following scenario and threshold tables are updated in the BD Atomic Schema: *KDD_SCNRO.*

For an approved notebook, a scenario record is created in the KDD_SCNRO table and the columns scnro_id and cntry_id are updated with the values: ML and customer focus value (113000004), respectively.

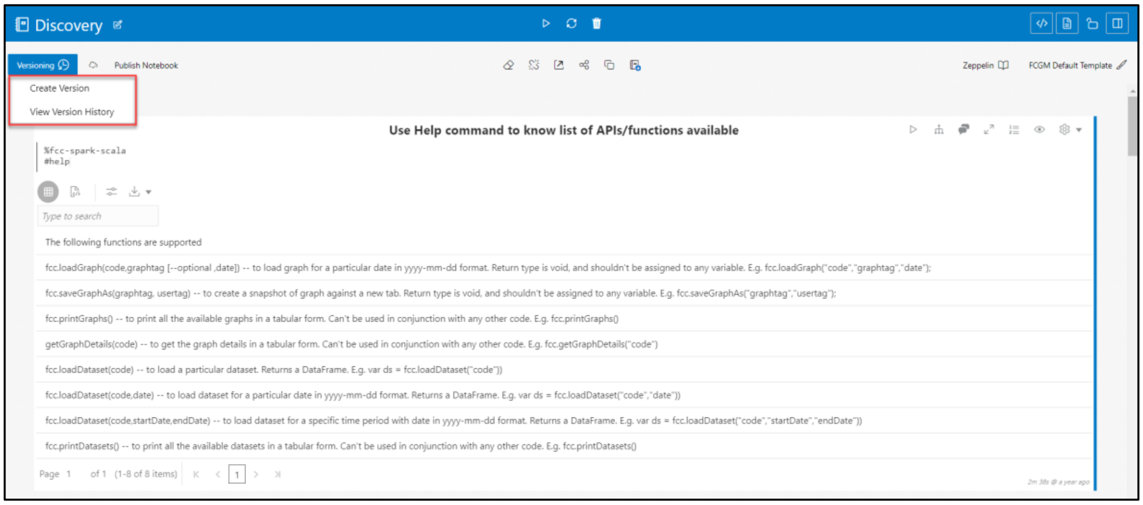
- *KDD_TSHLD_SET*
- *KDD_TSHLD*

For an approved notebook, the parameters are captured in the *KDD_TSHLD* table. Publishing multiple notebooks creates multiple threshold sets in the *KDD_TSHLD* table: *SCNRO_NB_PUBLISH*.

3.3.3.7 Versioning your Notebook

Data Studio provides a notebook versioning system. Notebook versions represent snapshots of the state of a notebook. Notebook versions can be associated with metadata such as a name and a description.

Restoring a notebook version resets the current notebook to the state represented by the version. This may affect the existing paragraphs, the notebook description, the selected layout, the selected style and the selected template.



Use Help command to know list of APIs/functions available

```
%fcc-spark-scala
#help
```

The following functions are supported

fcc.loadGraph(code,graphtag [--optional ,date]) -- to load graph for a particular date in yyyy-mm-dd format. Return type is void, and shouldn't be assigned to any variable. E.g. fcc.loadGraph("code","graphtag","date");

fcc.saveGraphAs(graphtag, usertag) -- to create a snapshot of graph against a new tab. Return type is void, and shouldn't be assigned to any variable. E.g. fcc.saveGraphAs("graphtag","usertag");

fcc.printGraphs() -- to print all the available graphs in a tabular form. Can't be used in conjunction with any other code. E.g. fcc.printGraphs()

getGraphDetails(code) -- to get the graph details in a tabular form. Can't be used in conjunction with any other code. E.g. fcc.getGraphDetails("code")

fcc.loadDataset(code) -- to load a particular dataset. Returns a DataFrame. E.g. var ds = fcc.loadDataset("code"))

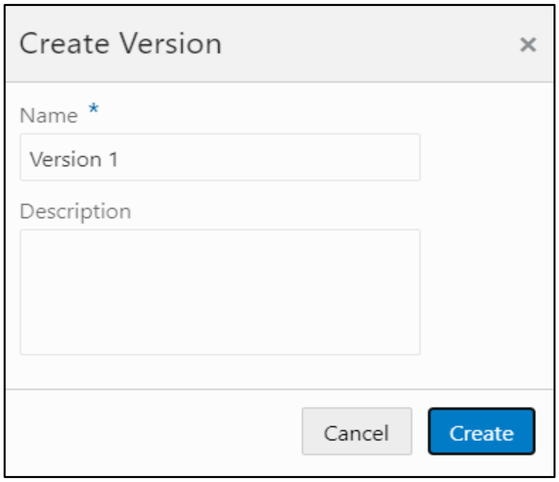
fcc.loadDataset(code,date) -- to load dataset for a particular date in yyyy-mm-dd format. Returns a DataFrame. E.g. var ds = fcc.loadDataset("code","date"))

fcc.loadDataset(code,startDate,endDate) -- to load dataset for a specific time period with date in yyyy-mm-dd format. Returns a DataFrame. E.g. var ds = fcc.loadDataset("code","startDate","endDate"))

fcc.printDatasets() -- to print all the available datasets in a tabular form. Can't be used in conjunction with any other code. E.g. fcc.printDatasets()

3.3.3.7.1 Creating Version

To create a version, use the **Versioning** button in the upper left corner. A version name is suggested, which contains a version number that is based on the existing versions. Optionally, a version description can be added.



Create Version

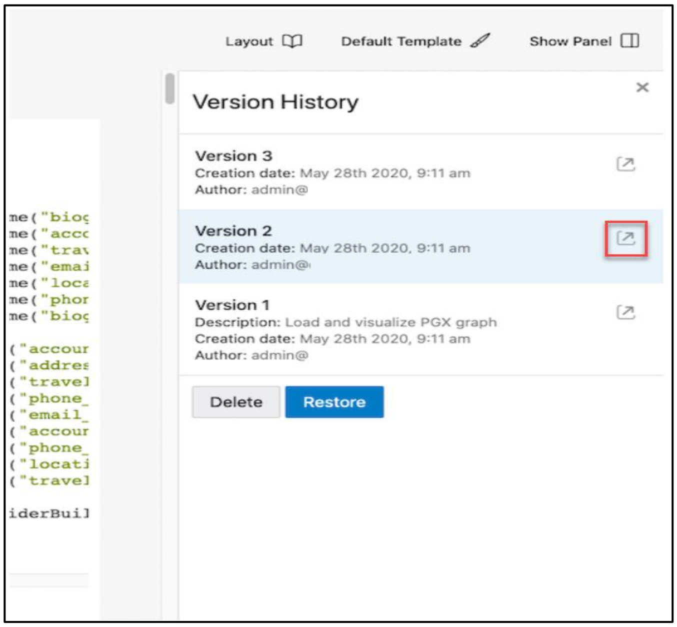
Name *

Version 1

Description

3.3.3.7.2 Viewing Version History

To restore a version, first open the version history panel using the **Versioning** button in the upper left corner. Then select the version that you want to restore and click **Restore.** Versions can also be opened in a new tab by clicking on the **External Link** icon on a version item.



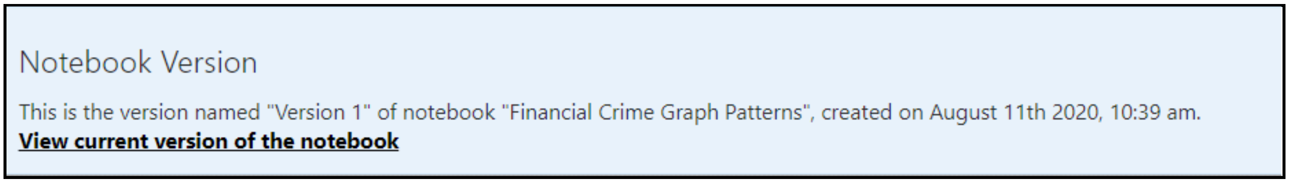
Version History

Version 3
Creation date: May 28th 2020, 9:11 am
Author: admin@

Version 2
Creation date: May 28th 2020, 9:11 am
Author: admin@

Version 1
Description: Load and visualize PGX graph
Creation date: May 28th 2020, 9:11 am
Author: admin@

The opened versions of the notebook has a banner at the top as a visual distinction to regular notebooks.



Notebook Version

This is the version named "Version 1" of notebook "Financial Crime Graph Patterns", created on August 11th 2020, 10:39 am.
View current version of the notebook

3.3.4 About Paragraphs

A paragraph is a piece of code that can be executed to obtain the result. Paragraph offers a workbench to author code or query using interpreter-friendly scripting languages supported in FCC Studio.

Data scientists and analysts can use paragraphs to interactively explore financial crime data by the authoring queries and applying these queries on the financial crime data lake to fetch results in various formats. This enables them to rapidly discover and model financial crime patterns.

Paragraphs are like the pages of the Notebook. A notebook can be developed with multiple types of paragraphs and executed to visualize the results. The FCC Studio notebooks provides you the flexibility to create paragraphs as many as you can by using a normal paragraph option or the paragraph with interpreters. In FCC Studio, you can add the following interpreter-based paragraphs:

- JDBC Interpreter paragraph
- PGX-Java paragraph
- PGX Algorithm paragraph
- PySpark paragraph
- PGQL paragraph
- Spark paragraph

For more information about Interpreters that are used, see About Interpreters.

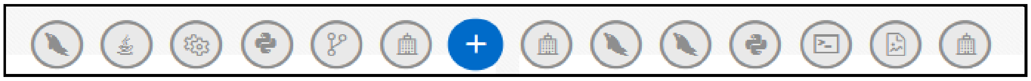
3.3.4.1 Creating a Paragraph

Paragraphs can be created using interpreters that are supported in FCC Studio.

To create a paragraph in a notebook, follow these steps:

1. Navigate to the Notebook page.
2. Hover above or below a paragraph and click + to add a paragraph to your Notebook.

   To create a paragraph based on the interpreter, you can choose the respective icons displayed for the interpreters as shown. You can hover over the icons to view the name of the interpreter-based paragraph.



The added paragraph can be used to write queries based on your requirement to plot and visualize your data. For more information about data visulaization, see Data Visualization.

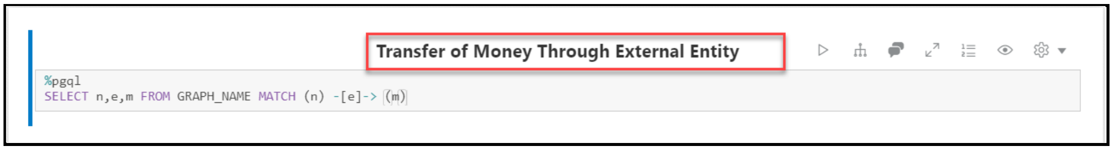
3.3.4.2 Writing a Sample paragraph

After creating the paragraph using the available interpreter, you can write your query in the paragraph to execute and display the visuals or analysis as results.

Let us take an example to write a **PGQL Paragraph** using **PGQL Interpreter** and quering for the graph plots.

To create and write a PGQL query in the paragraph, follow these steps:

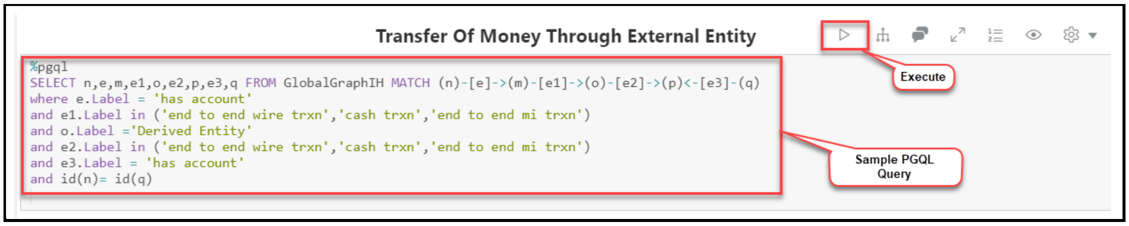
1. Hover above or below a paragraph and click **Add PGQL Paragraph**.
2. Specify the title for your paragraph. In this case, **Transfer of Money Through External Entity.**



Transfer of Money Through External Entity

```
%pgql
SELECT n,e,m FROM GRAPH_NAME MATCH (n) -[e]-> (m)
```

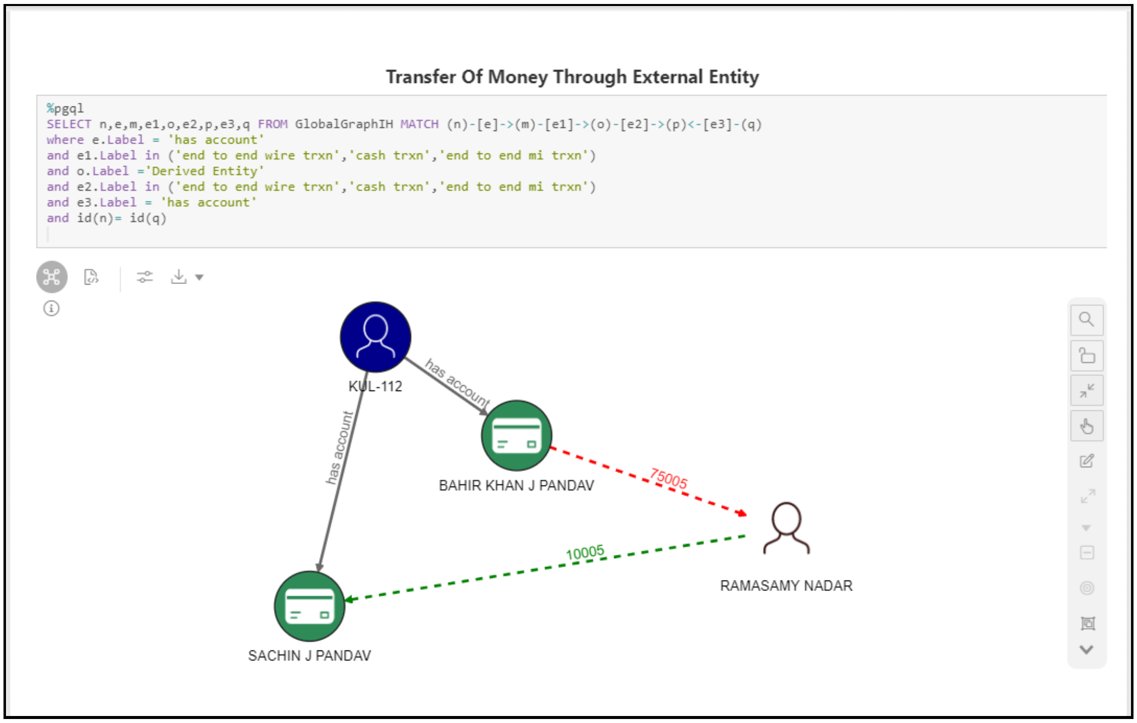
3. Speciy the PGQL query that you want to execute from the Global Graph and click **Execute**.



Transfer Of Money Through External Entity

```
%pgql
SELECT n,e,m,e1,o,e2,p,e3,q FROM GlobalGraphIH MATCH (n)-[e]->(m)-[e1]->(o)-[e2]->(p)<-[e3]-(q)
where e.Label = 'has account'
and e1.Label in ('end to end wire trxn','cash trxn','end to end mi trxn')
and o.Label ='Derived Entity'
and e2.Label in ('end to end wire trxn','cash trxn','end to end mi trxn')
and e3.Label = 'has account'
and id(n)= id(q)
```

The graph results are displayed as shown.



Transfer Of Money Through External Entity

```
%pgql
SELECT n,e,m,e1,o,e2,p,e3,q FROM GlobalGraphIH MATCH (n)-[e]->(m)-[e1]->(o)-[e2]->(p)<-[e3]-(q)
where e.Label = 'has account'
and e1.Label in ('end to end wire trxn','cash trxn','end to end mi trxn')
and o.Label ='Derived Entity'
and e2.Label in ('end to end wire trxn','cash trxn','end to end mi trxn')
and e3.Label = 'has account'
and id(n)= id(q)
```

Similarly, you can us other interpreters to create your paragraph, write your queries and execute to visulaize the data. For more information about data visulaization, see Data Visualization.

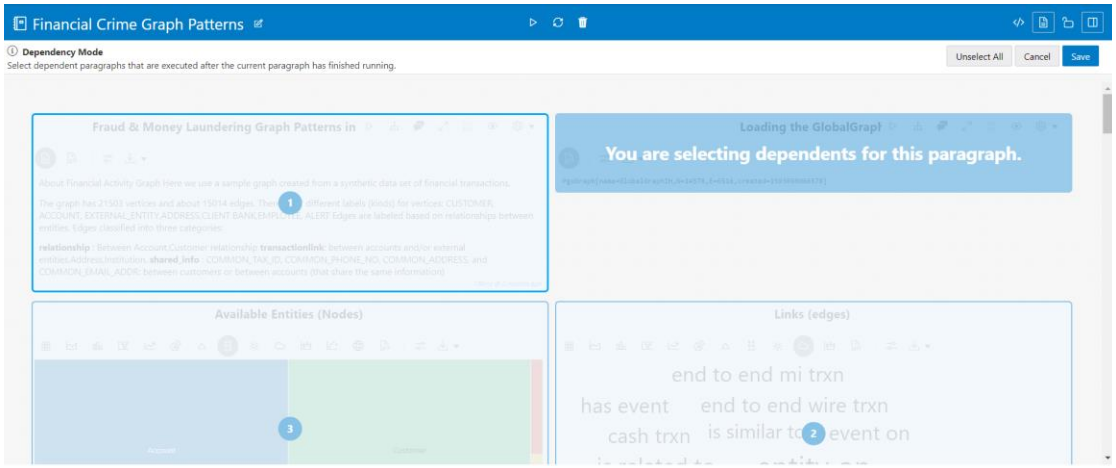
3.3.4.3 Paragraph Dependencies

The Paragraph Dependencies helps you to add dependencies between paragraphs. The dependents of a paragraph are automatically executed after the original paragraph itself or any graph manipulation on the original paragraph is executed.

To create paragraph dependencies, follow these steps:

1. Click Dependency on the **Paragraph Settings Bar** of a paragraph. The **Dependency Mode** window is displayed.
2. Select the paragraphs to add as dependents.

   The order in which the paragraphs are selected appears as a number over the selected paragraphs. The number indicates the order in which the dependent paragraphs will be executed.



Financial Crime Graph Patterns

Dependency Mode

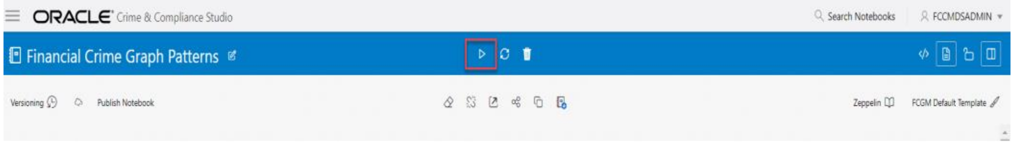
Select dependent paragraphs that are executed after the current paragraph has finished running.

You are selecting dependents for this paragraph.

3. Click **Save**. The changes are saved. Every time a paragraph is executed, or graph actions are applied, its dependent paragraphs will be executed automatically.

3.3.4.4 Run All Paragraphs

You can run all the paragrahs in the notebook by clicking ▷ **Run Paragraphs** in the notebook toolbar to execute the complete notebooks and view the data in a single go.



When you execute all the paragraphs at once, the paragraphs are executed in top to bottom order. If a paragraph was deleted during the Run Paragraphs job execution, it is ignored, and paragraph execution continues for the rest of paragraphs.

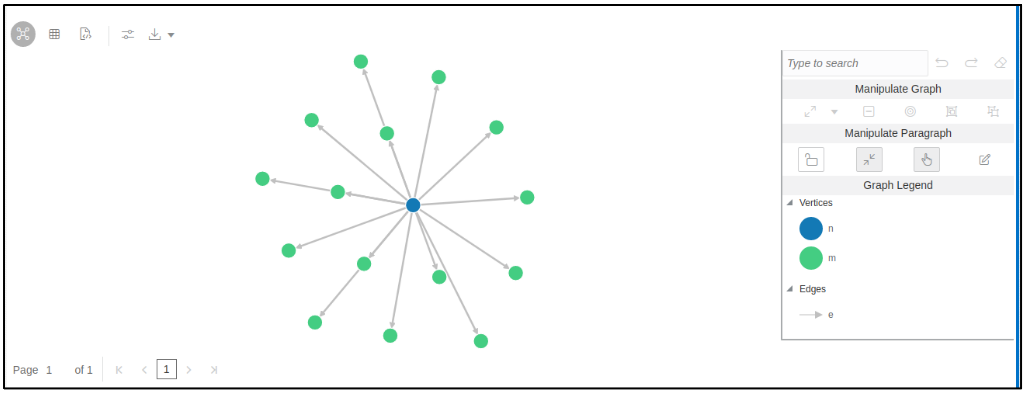
3.3.5 Data Visualization

The notebooks provide you flexibility to visualize and customize your data. Following are the data visualizations that are available in FCC Studio:

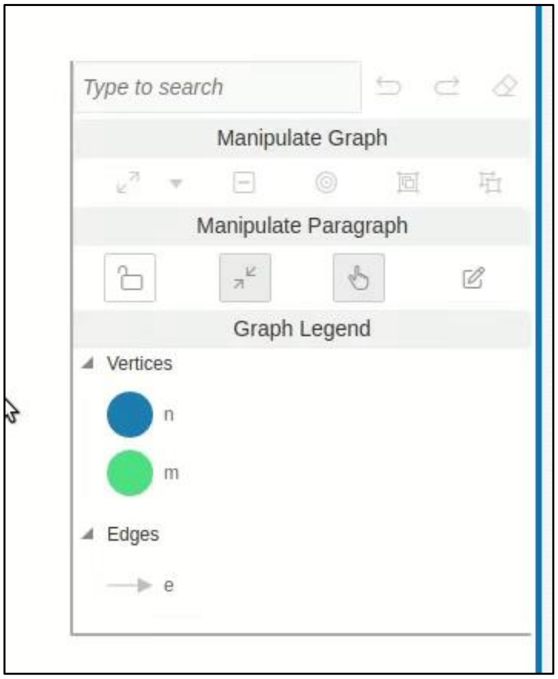
- Graph Visualization
- Table
- HTML or Markdown
- Text
- Area Chart
- Bar Chart
- Funnel Chart
- Line Chart
- Pie Chart
- Pyramid Chart
- Treemap Diagram
- Sunburst Diagram
- Tag Cloud
- Box Plot
- Scatter Plot
- Map Visualizer

3.3.5.1 Graph Visualization

The FCC Studio allows to visualize data in form of a Graph Visualization. All actions of the graph visualization are collected inside a panel.

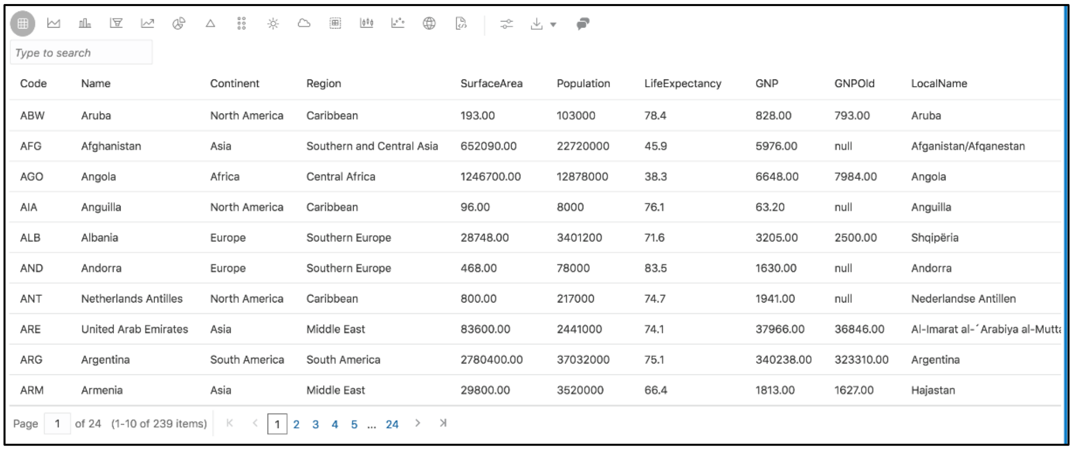


The panel can be resized and includes two states, default and minimized panel. These panels enables you to manipulate paragraphs and graphs; and provides options to customize your graph.



3.3.5.2 Table

The FCC Data Studio allows you to visualize your data in the form of a Table Diagram. The table can be sorted by column in ascending or descending order. Additionally, the table can be filtered for a specific search term. Rows that do not contain this term are hidden from view and the remaining rows highlight the location of the search term within the row.

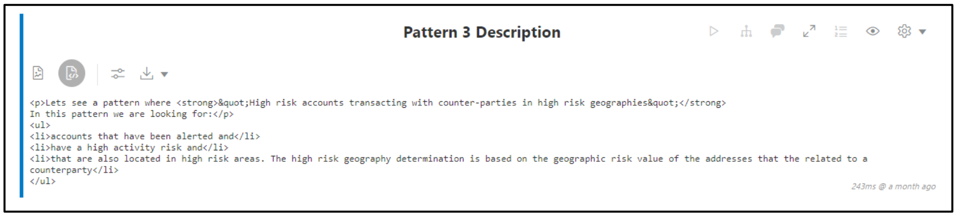


Code	Name	Continent	Region	SurfaceArea	Population	LifeExpectancy	GNP	GNPOld	LocalName
ABW	Aruba	North America	Caribbean	193.00	103000	78.4	828.00	793.00	Aruba
AFG	Afghanistan	Asia	Southern and Central Asia	652090.00	22720000	45.9	5976.00	null	Afganistan/Afqanestan
AGO	Angola	Africa	Central Africa	1246700.00	12878000	38.3	6648.00	7984.00	Angola
AIA	Anguilla	North America	Caribbean	96.00	8000	76.1	63.20	null	Anguilla
ALB	Albania	Europe	Southern Europe	28748.00	3401200	71.6	3205.00	2500.00	Shqipëria
AND	Andorra	Europe	Southern Europe	468.00	78000	83.5	1630.00	null	Andorra
ANT	Netherlands Antilles	North America	Caribbean	800.00	217000	74.7	1941.00	null	Nederlandse Antillen
ARE	United Arab Emirates	Asia	Middle East	83600.00	2441000	74.1	37966.00	36846.00	Al-Imarat al-´Arabiya al-Mutt
ARG	Argentina	South America	South America	2780400.00	37032000	75.1	340238.00	323310.00	Argentina
ARM	Armenia	Asia	Middle East	29800.00	3520000	66.4	1813.00	1627.00	Hajastan

Page 1 of 24 (1-10 of 239 items) 1 2 3 4 5 ... 24

3.3.5.3 HTML or Markdown

The FCC Data Studio allows you to visualize your data in the form of HTML or Markdown text.

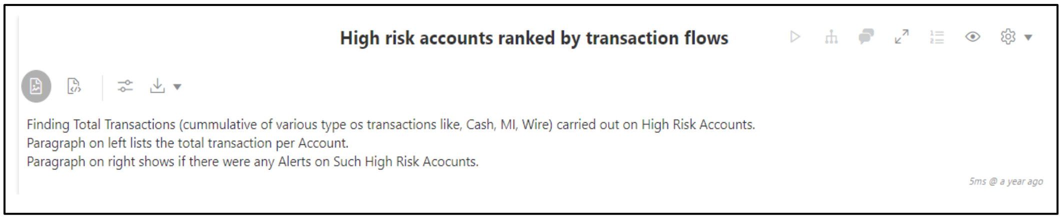

Pattern 3 Description

```
<p>Lets see a pattern where <strong>&quot;High risk accounts transacting with counter-parties in high risk geographies&quot;</strong>
In this pattern we are looking for:</p>
<ul>
<li>accounts that have been alerted and</li>
<li>have a high activity risk and</li>
<li>that are also located in high risk areas. The high risk geography determination is based on the geographic risk value of the addresses that the related to a
counterparty</li>
</ul>
```

243ms @ a month ago

3.3.5.4 Text

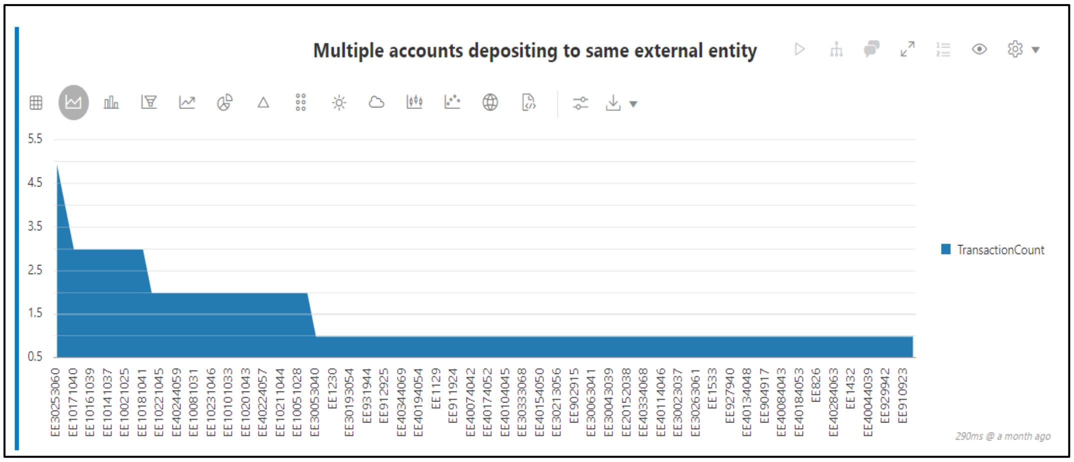
The FCC Studio allows you to visualize your data in the form of a Text that are structured and presented for you to customize the data.


High risk accounts ranked by transaction flows

Finding Total Transactions (cummulative of various type os transactions like, Cash, MI, Wire) carried out on High Risk Accounts.
Paragraph on left lists the total transaction per Account.
Paragraph on right shows if there were any Alerts on Such High Risk Accocunts.

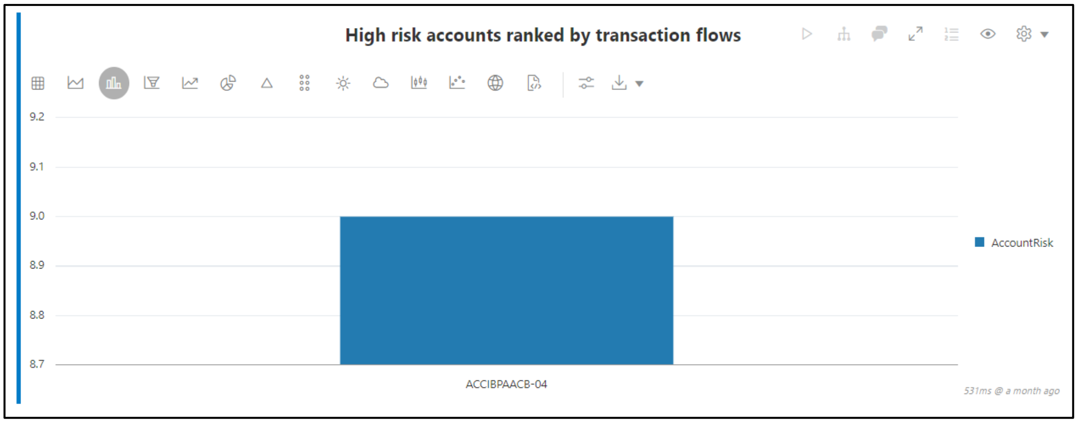
3.3.5.5 Area Chart

The FCC Studio allows you to visualize your data in the form of an Area Chart.



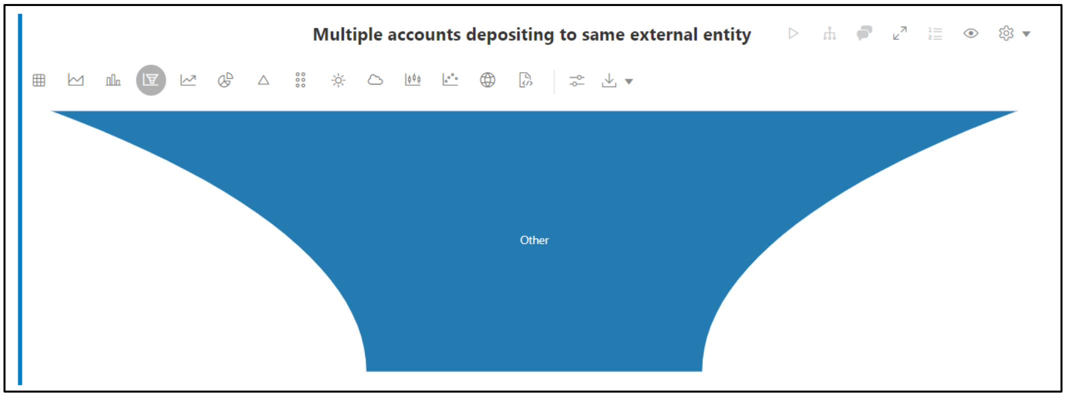
3.3.5.6 Bar Chart

The FCC Studio allows you to visualize your data in the form of a Bar Chart.



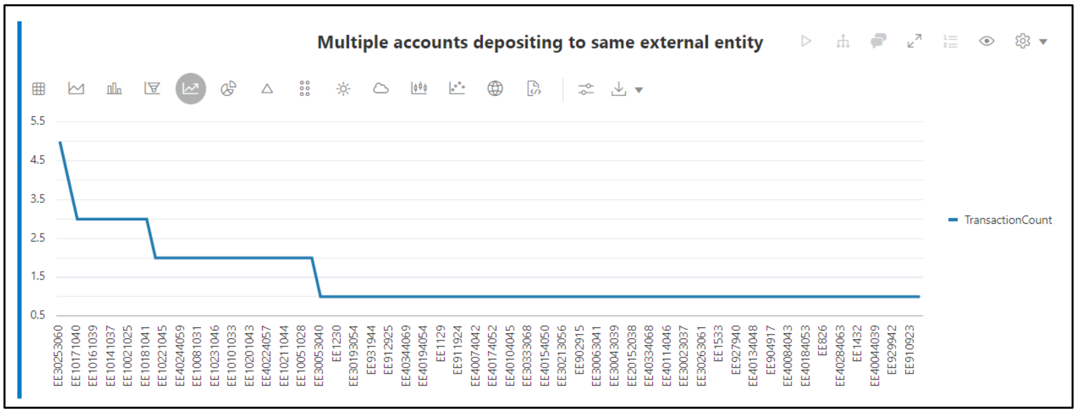
3.3.5.7 Funnel Chart

The FCC Studio allows you to visualize your data in the form of a Funnel Chart.



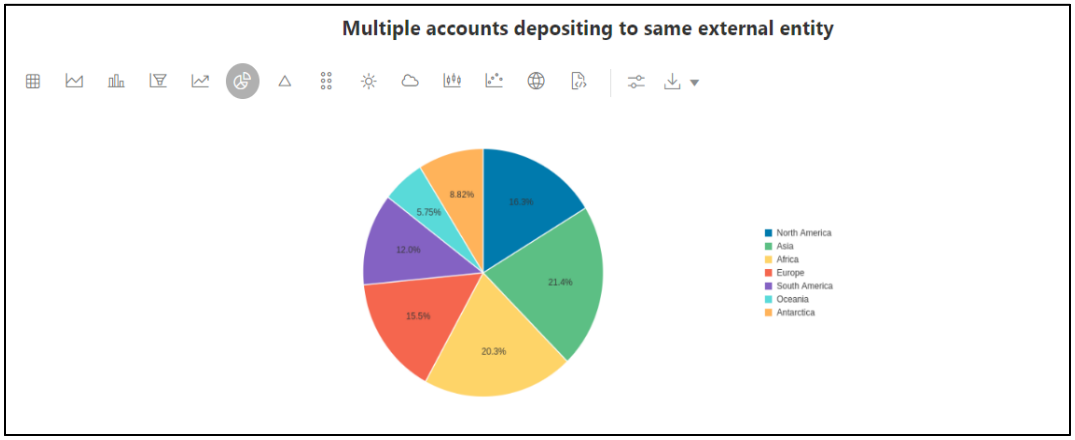
3.3.5.8 Line Chart

The FCC Studio allows you to visualize your data in the form of a Line Chart.



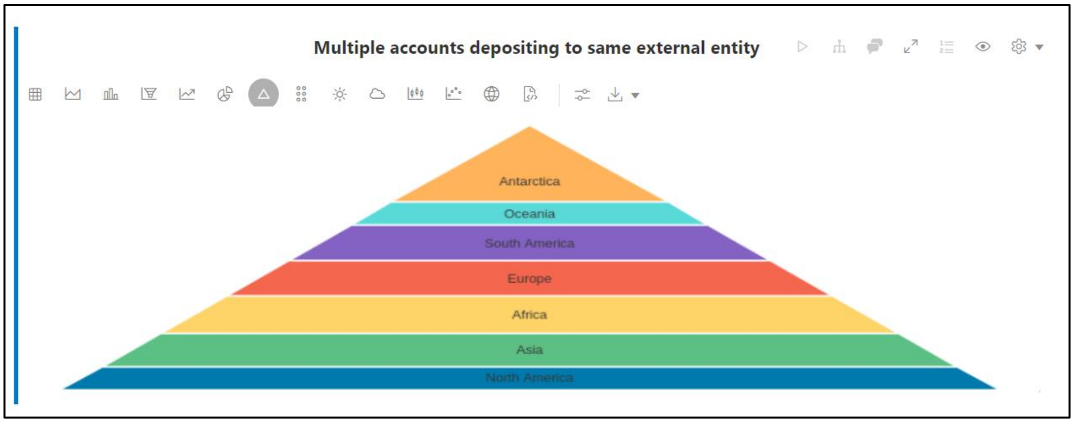
3.3.5.9 Pie Chart

The FCC Studio allows you to visualize your data in the form of a Pie Chart.



3.3.5.10 Pyramid Chart

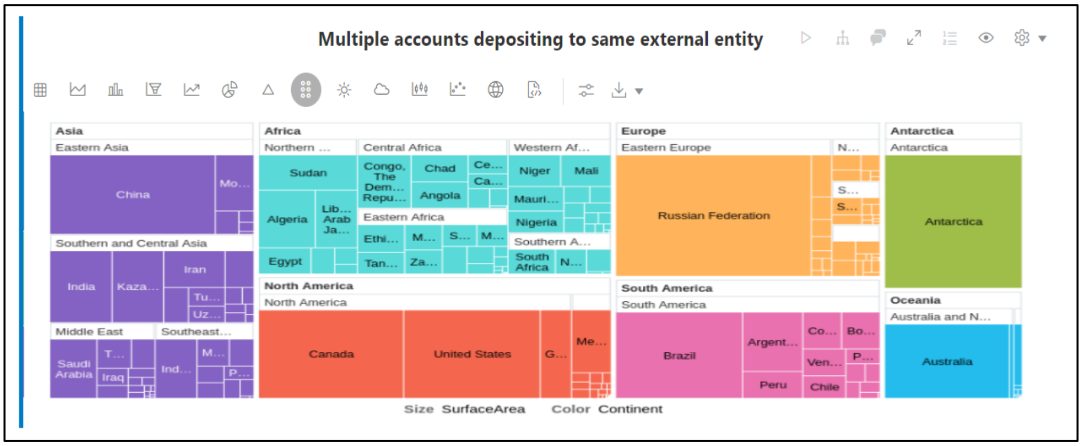
The FCC Studio allows you to visualize your data in the form of a Pyramid Chart.



Multiple accounts depositing to same external entity

3.3.5.11 TreeMap Diagram

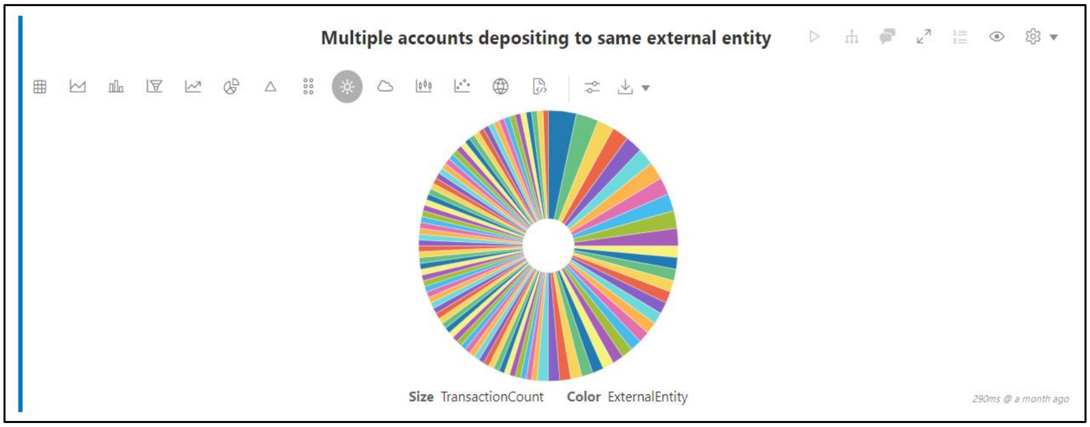
The FCC Studio allows you to visualize your data in the form of a TreeMap Diagram.



Multiple accounts depositing to same external entity

3.3.5.12 Sunburst Diagram

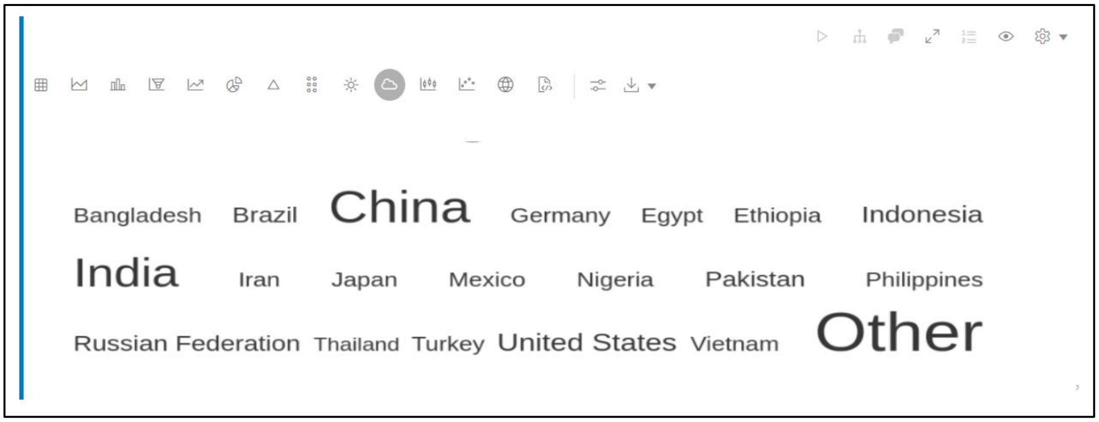
The FCC Studio allows you to visualize your data in the form of a Sunburst Diagram.



Multiple accounts depositing to same external entity

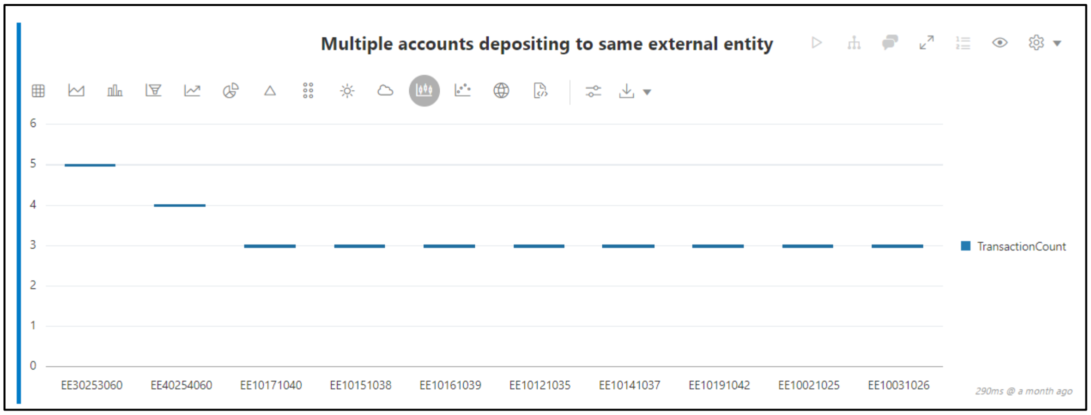
3.3.5.13 Tag Cloud

The FCC Studio allows you to visualize your data in the form of tags.The tag cloud operation is used to identity the spots were there are more flags.



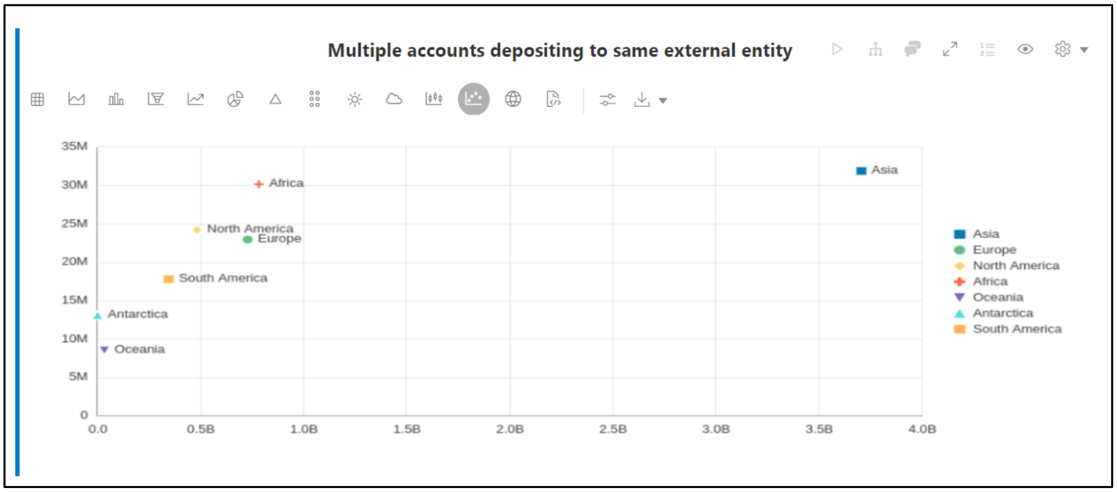
3.3.5.14 Box Plot

The FCC Studio allows you to visualize your data in the form of a Box Plot.



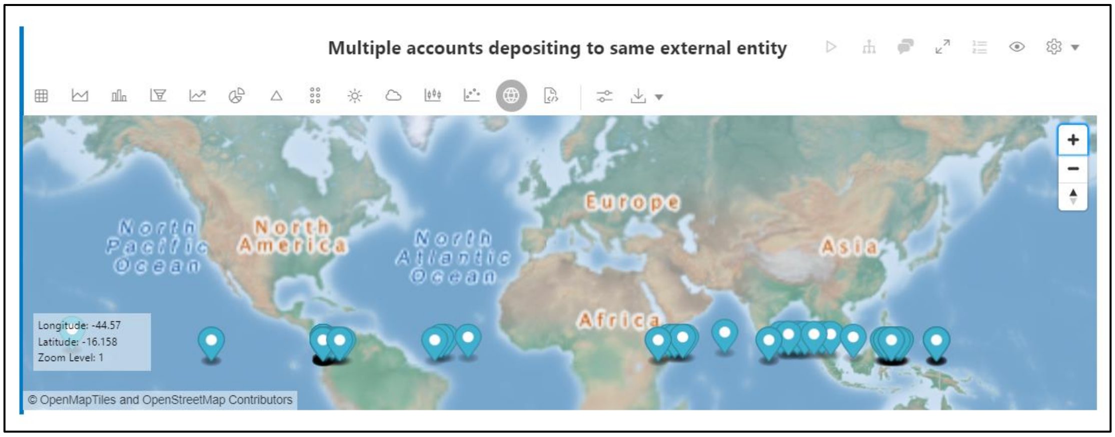
3.3.5.15 Scatter Plot

The FCC Studio allows you to visualize your data in the form of a Scatter Plot.



3.3.5.16 Map Visualizer

The FCC Studio allows you to visualize your data on top of a Map.



Multiple accounts depositing to same external entity

Longitude: -44.57
Latitude: -16.158
Zoom Level: 1

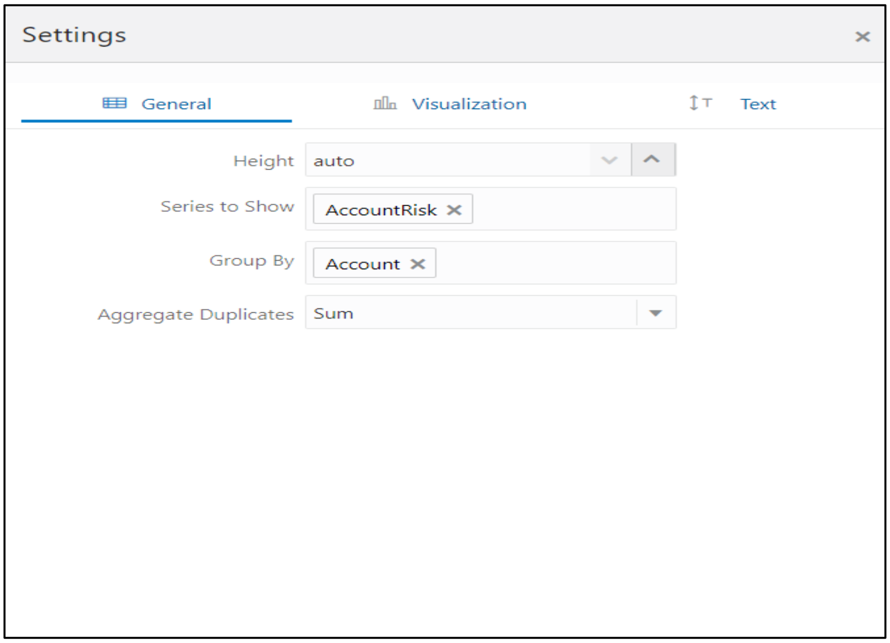
3.3.5.17 Setting up the Visualizations

The visualization settings offer several ways of configuration, which are split into three tabs: General, Visualization, and Text. To set the configuration for your table view, click **Settings.**

These tabs in the Settings window allow you to customize the properties of the text, graphs, visuals, and viewing the visuals in various orientation. For example, you can view the Bar Chart to change the colors, type of bar, and 360-degree view of the visuals.

The Settings window is displayed and contains the following category:

- General
- Visualization
- Text



Settings

Height: auto

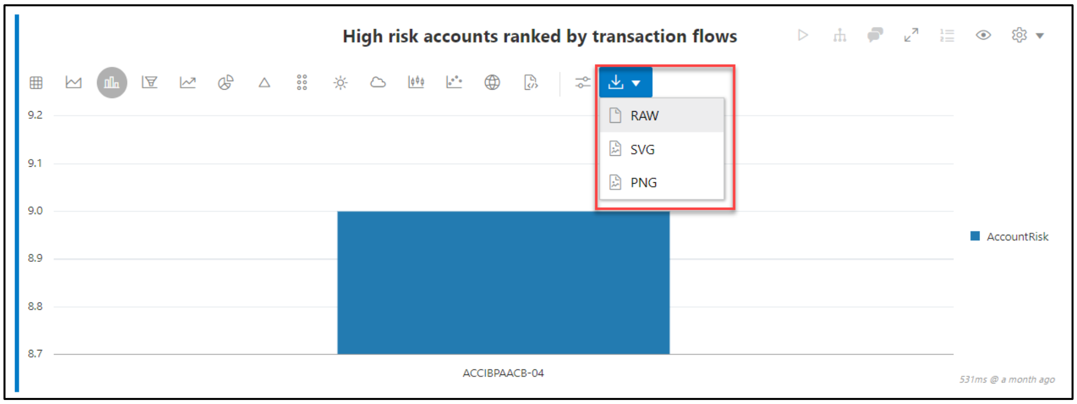
Series to Show: AccountRisk

Group By: Account

Aggregate Duplicates: Sum

3.3.5.18 Download As

The visuals that are displayed can be downloaded using the **Download As** option. The available download formats are, RAW, SVG, and PNG.

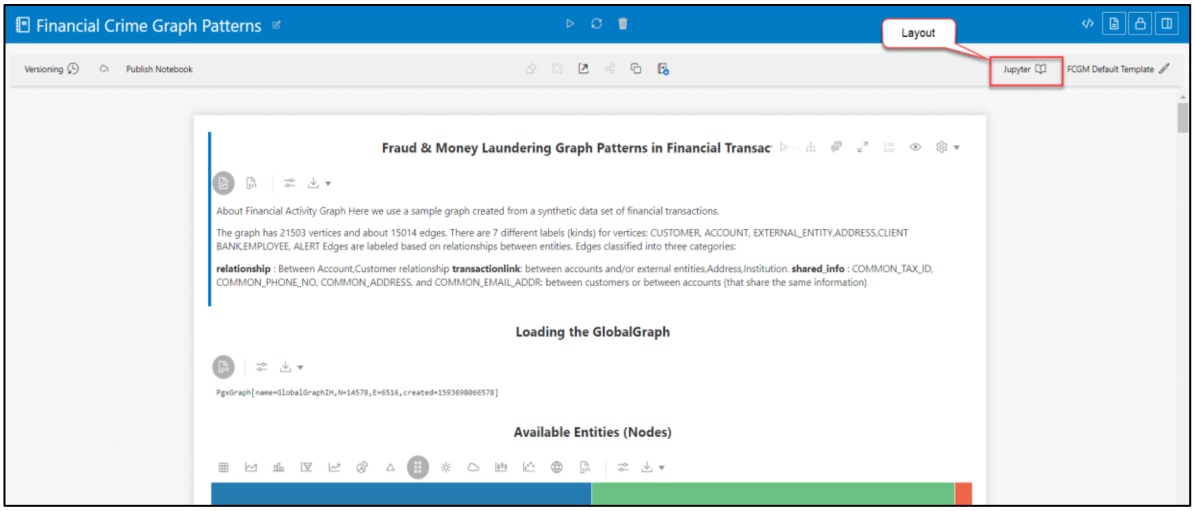


3.3.6 Layouts

Data Studio supports multiple layouts to give the resemblance of familiar systems such as Project Jupyter and Apache Zeppelin.

3.3.6.1 Jupyter Layout

The Jupyter Layout is a single column layout that is something that the traditional documents look like. To get the Jupyter-Experience, paragraphs can hide certain aspects, such as Input, Selection and Result. For example, this allows to show only HTML when a Markdown paragraph is executed.



Fraud & Money Laundering Graph Patterns in Financial Transac

About Financial Activity Graph Here we use a sample graph created from a synthetic data set of financial transactions.

The graph has 21503 vertices and about 15014 edges. There are 7 different labels (kinds) for vertices: CUSTOMER, ACCOUNT, EXTERNAL_ENTITY,ADDRESS,CLIENT BANK,EMPLOYEE, ALERT Edges are labeled based on relationships between entities. Edges classified into three categories:

relationship : Between Account,Customer relationship **transactionlink**: between accounts and/or external entities,Address,Institution. **shared_info** : COMMON_TAX_ID, COMMON_PHONE_NO, COMMON_ADDRESS, and COMMON_EMAIL_ADDR: between customers or between accounts (that share the same information)

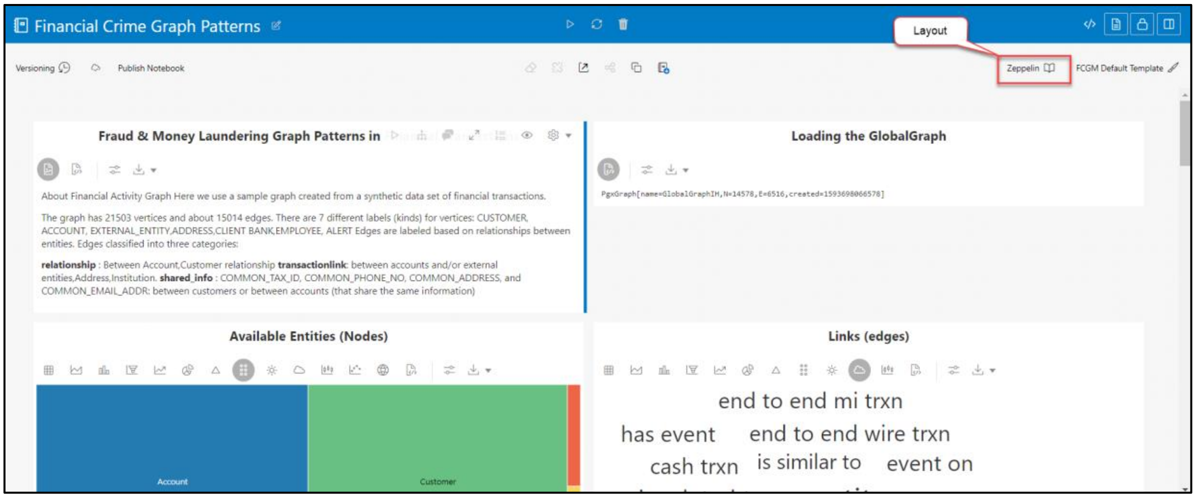
Loading the GlobalGraph

```
PgxGraph[name=GlobalGraphIH,N=14578,E=6516,created=1593698066578]
```

Available Entities (Nodes)

3.3.6.2 Apache Zeppelin Layout

The Zeppelin Layout allows you to resize paragraphs and have multiple paragraphs next to each other.



Fraud & Money Laundering Graph Patterns in

About Financial Activity Graph Here we use a sample graph created from a synthetic data set of financial transactions.

The graph has 21503 vertices and about 15014 edges. There are 7 different labels (kinds) for vertices: CUSTOMER, ACCOUNT, EXTERNAL_ENTITY,ADDRESS,CLIENT BANK,EMPLOYEE, ALERT Edges are labeled based on relationships between entities. Edges classified into three categories:

relationship : Between Account,Customer relationship **transactionlink**: between accounts and/or external entities,Address,Institution. **shared_info** : COMMON_TAX_ID, COMMON_PHONE_NO, COMMON_ADDRESS, and COMMON_EMAIL_ADDR: between customers or between accounts (that share the same information)

Loading the GlobalGraph

```
PgxGraph[name=GlobalGraphIH,N=14578,E=6516,created=1593698066578]
```

3.3.7 Comments

You can add comments on the paragraph to highlight changes or suggestion to the visuals. Comments can be added from the user interface and they can delete their own comments.

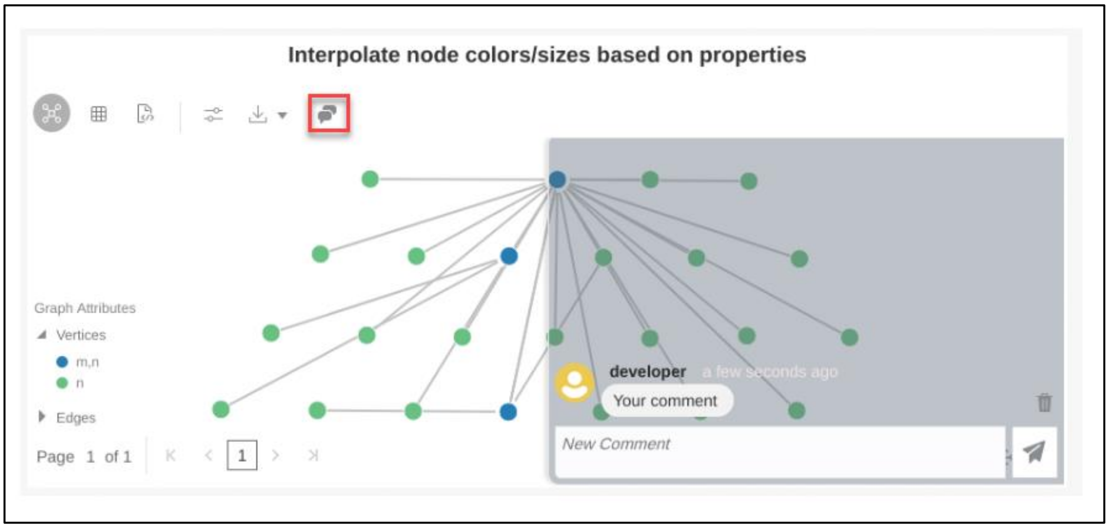
You can click **Comments** in the paragraph toolbar. The Comments view shows all paragraph comments, such as:

- **Add Comment**

  Using a text area field labeled as New Comment on the bottom of comments.

- **Delete Comment**

  Using a bin icon on the right of comments (you can only delete your own comments).



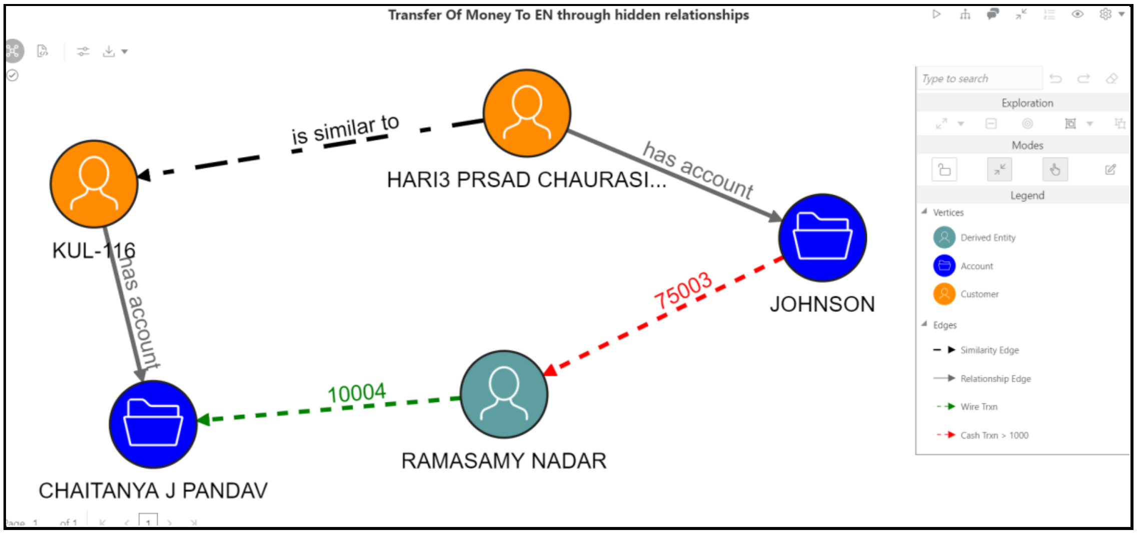
3.4 About Graphs

The FCC Studio provides pre-defined Financial crime graph that can be pre-loaded for most users and then state the ways they can load different graphs. It also provides an intuitive way for creating graphs used in notebooks, where you can load graphs from external sources or create custom graphs. Using

PGX, you can load multiple graphs into a notebook and create PGQL queries against different graphs. The result obtained from running a paragraph in a notebook can be used as an input to other paragraphs in the notebook. The results of analytics algorithms are stored as transient properties of nodes and edges in the graph. Pattern matching can then be used against these properties.

The pre-defined graphs within the notebook can be accessed and used for the analysis. To access the pre-defined graphs within the notebook, select any of the out-of-the-box notebook, make the necessary changes for the PGQL code within the paragraph for the analysis that you require and run the paragraph the query results are displayed as graph.

For example, to view the Transfer of Money through hidden relationships, use the Financial Crime Graph Patterns notebook and execute the paragraph by providing the information about entity, customer, and account.



The graph configuration can be defined through an UI based configurator or a JSON configurator. Graph configurations give you an easy access to graphs using PGX-ALGORITHM, PGX-JAVA and PGQL interpreters.

For information on configuring data sources for graphs, see the Configuring Data Sources for Graph section in the OFS Crime and Compliance Studio Administration and Configuration Guide.

The graph can be created based on the following options:

- Creating Graphs by using the UI Based Configurator
- Creating Graphs by using the JSON Configurator

3.4.1 Accessing the Graphs page

To create graphs based on the configurator, you need to acces the Graphs page in the FCC studio.

To access the Graphs page, follow these steps:

1. In the FCC Studio Workspace, click **Navigation Menu** on upper-left corner. The menu items are listed.
2. Click **Graphs**. The Graphs page is displayed.
3. In the **Graphs** page, click **Create**.

The **Create new Graph Configuration** dialog box is displayed. In the dialog box, you can click:

- **Wizard** to configure the graphs based on the UI configurator.
- **Plain** to configure the graphs based on the JSON configurator.

3.4.2 Creating Graphs by using the UI Based Configurator

To create the Graphs by using the **UI Based Configurator**, follow these steps:

1. In the **Create new Graph Configuration** dialog box, click **Wizard** and specify the following information:
   - **Name**

     Define a unique graph name for this graph.
   - **Format**

     Data studio provides creating a graph from multiple formats, such as: *Two tables*, *PGB*, *Edge list*, *Adjacency list*, *Flat file*, *GraphML*, *PG***,** and *RDF*.
   - **Vertex Properties**

     Define the properties and types that your vertices must have in the graph. For example a Patient Vertex can have a **Name (string)** property.
   - **Edge Properties**

     Define the properties and types that your edges must have in the graph. For example a Patient_Medication Edge can have a **Dose (String)** property.
   - **Datastore**

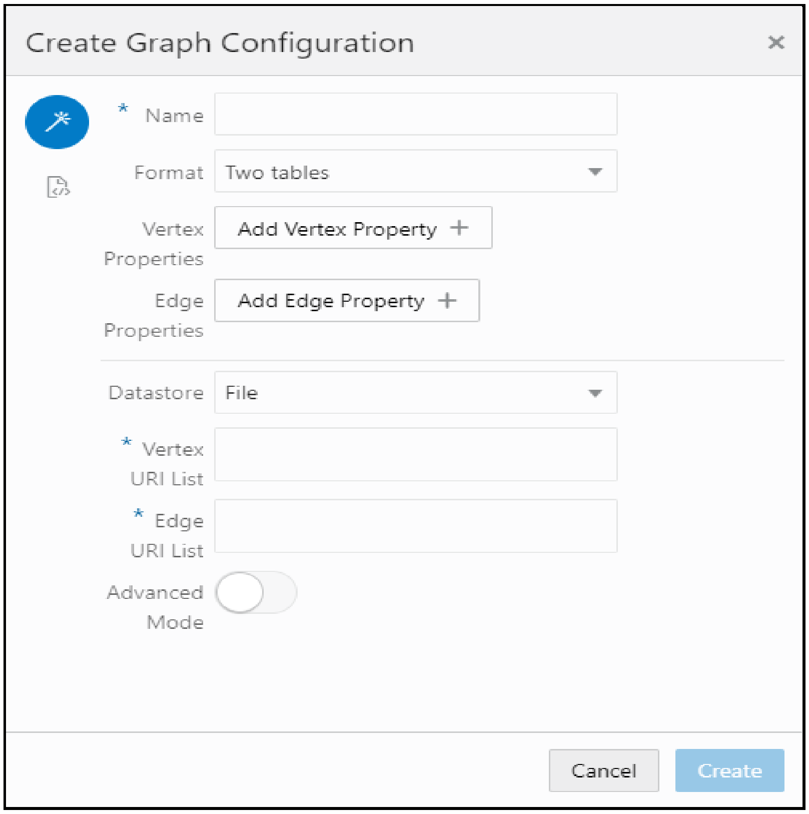
     Choose the data source for the vertices and edges of the graph. Depending on the option you have to provide different information:
     - **File**

       Provide the Vertex URL List and the Edge URL List.
     - **Spark**

       Provide the Nodes and the Edges Table Name.
     - **RDBMS**

       Provide the username, password, number of connections, Nodes and Edges Key Columns and the From and To Node ID Columns.
     - **Advance Mode**

       Click **Advance Mode** to add more properties related to Vertex and Edge.



Create Graph Configuration

* Name

Format: Two tables

Vertex Properties: Add Vertex Property +

Edge Properties: Add Edge Property +

Datastore: File

* Vertex URI List

* Edge URI List

Advanced Mode

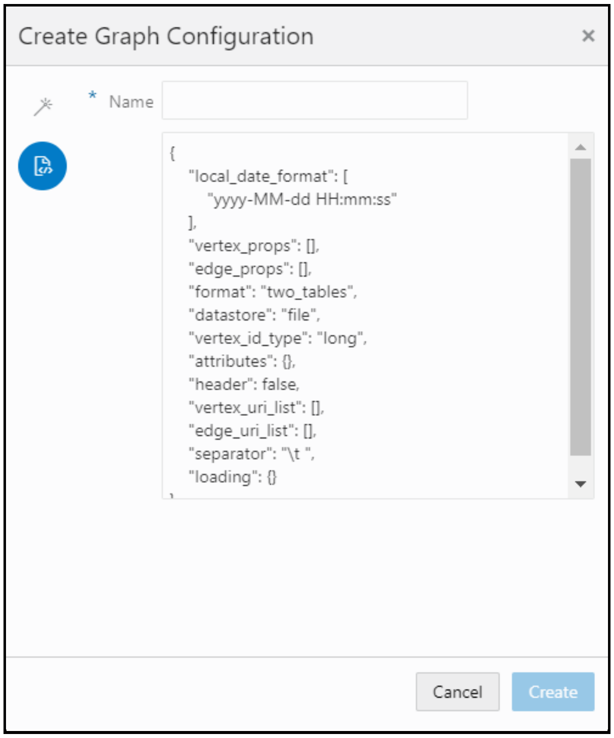
2. Click **Create**. The graph is created.

3.4.3 Creating Graphs by using the JSON Configurator

You can write a JSON configuration file based on the PGX JSON fields defined in the PGX graph configuration documentation to confirgure your graph.

To create the Graphs by using the **JSON Configurator**, follow these steps:

1. In the **Create new Graph Configuration** dialog box, click **Plain**
2. Specify the name of your graph and the **JSON** code in the code field.



```
{
  "local_date_format": [
    "yyyy-MM-dd HH:mm:ss"
  ],
  "vertex_props": [],
  "edge_props": [],
  "format": "two_tables",
  "datastore": "file",
  "vertex_id_type": "long",
  "attributes": {},
  "header": false,
  "vertex_uri_list": [],
  "edge_uri_list": [],
  "separator": "\t ",
  "loading": {}
}
```

3. Click **Create**. The graph is created. Sample code snippet is as follws:

```
{
"vertex_labels": false, "edge_label": true, "loading": {
"load_edge_label": true
},
"date_format": "yyyy-MM-dd HH:mm:ss", "vertex_props": [
{
"type": "string",
"name": "type"
},
{
"type": "string", "name": "religion"
},
{
"type": "string",
"name": "company"
},
{
"type": "string", "name": "musicGenre"
},
{
```

```
"type": "string",
"name": "show"
},
{
"type": "string",
"name": "name"
},
{
"type": "string",
"name": "country"
},
{
"type": "string",
"name": "team"
},
{
"type": "string",
"name": "genre"
},
{
"type": "string", "name": "occupation"
},
{
"type": "string",
"name": "role"
}
],
"edge_props": [
{
"type": "float",
"name": "weight"
}
],
"format": "edge_list", "vertex_id_type": "integer", "attributes": {},
"header": "",
"uri": "http://pgx.us.oracle.com/graphs/connections.edge_list",
"separator": "\t "
}
```

This newly created graph configuration can be used in a notebook for the following:

- Load a graph using the PGX interpreter.

  To load a graph, create a notebook and a paragraph with a PGX interpreter. Enter the code format given in the following example:

  ```
  graph = session.readGraphWithProperties(dataSourceName, 'graphName')
  ```

  Here, `dataSourceName` refers to the graph name that you have created.

- Query a graph using the PGQL interpreter.

  To query a graph, create a notebook and create a paragraph with a PGQL interpreter. Specify the code format given in the following example:

  ```
  "SELECT n,e,m FROM GRAPH_NAME MATCH (n) -[e]-> (m)"
  ```

Here `GRAPH_NAME` refers to the graph name that you have created.

3.4.4 Viewing the Details in a Graph

The graphs that are created and executed can be repositioned, collapsed, expanded, and then edges and vertex can be cusomtized.

3.4.4.1 Repositioning Nodes

The Network Graph page allows you to move nodes around the screen, using the drag and drop feature, to reposition them.

To reposition nodes, follow these steps:

1. Navigate to the **Network Graph** in a notebook.
2. Select a node to reposition and click it.
3. Drag and Drop the node to the required position.

> **NOTE** The graph only uses a specific portion of the browser window to display the graph. Dragging a node beyond a certain point towards the right side of the browser hides the portion of the graph dragged beyond that point. However, you can use the Zoom Out feature on the Graph Toolbar to view the hidden portion again.

3.4.4.2 Dropping and Expanding Nodes

You can drop the nodes to hide all outgoing links and nodes to which these outgoing links are connected from the node being collapsed. You also expand the the nodes. The dropped node can be regenerated again by executing the graph. To drop a node, follow these steps.

1. Navigate to the **Network Graph** in a notebook.
2. Select a node to collapse and right-click the node. An option menu is displayed.
3. Select the **Drop** option from the menu. The node and its outgoing links are dropped.

NOTE If any child node has at least one incoming link from any other node, the child node and its child network are not dropped. But the link from the dropeed node to the child node is hidden and the icon of the dropped node changes to indicate that the node is in a dropped state.

On the Node menu of a dropped node, the drop option changes to Expand. If the user drops a node but there is no impact on the graph (that is, if no part of the graph is hidden), the Node menu remains unchanged. There is no restriction on how many nodes can be dropped on a graph.

4. To expand the node, select **Expand** from the menu. The outgoing links are then restored on the page.

3.4.4.3 Viewing the Node Details

You can view the current information associated with the selected node. This is the same information that is displayed on the **Entity Summary Historical Report** paragraph for this entity.

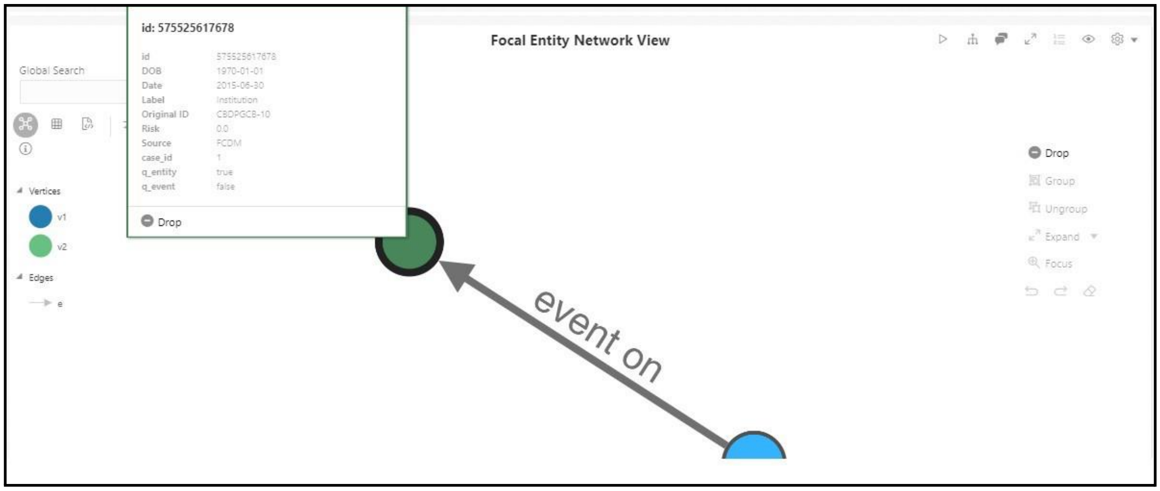
To view the node details, follow these steps:

1. Navigate to the **Network Graph** in a notebook.
2. Select a node and right-click. An option menu is displayed. The **Node Details** window is displayed with the current information associated with the selected node. This includes the Properties and Risk details of the node.

3.4.4.4 Deleting a Node

If you want to delete a node from a graph which is not required for your analysis, you can remove the unwanted node and view. To delete a node, follw these steps:

1. Navigate to the **Network Graph** in a notebook.
2. Right click on any node as shown in the following figure and click **Drop**.



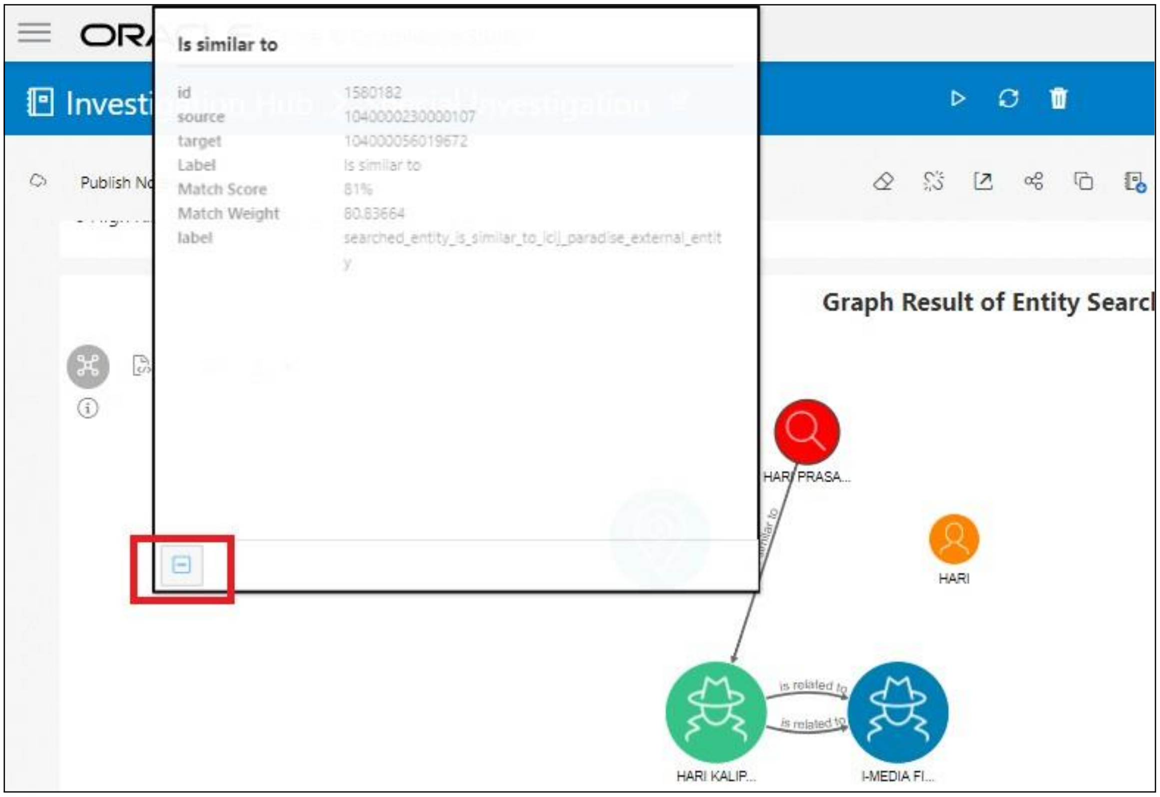
id: 575525617678

id	575525617678
DOB	1970-01-01
Date	2015-06-30
Label	Institution
Original ID	CBDPGCB-10
Risk	0.0
Source	FCDM
case_id	1
q_entity	true
q_event	false

3.4.5 Removing an Edge

You can remove an edge from a network graph to view the result in On-screen data. To remove an edge, follow these steps:

1. Navigate to the **Network Graph** in a notebook.
2. Right- click on any edge and click **Drop - Delete** selected vertices.



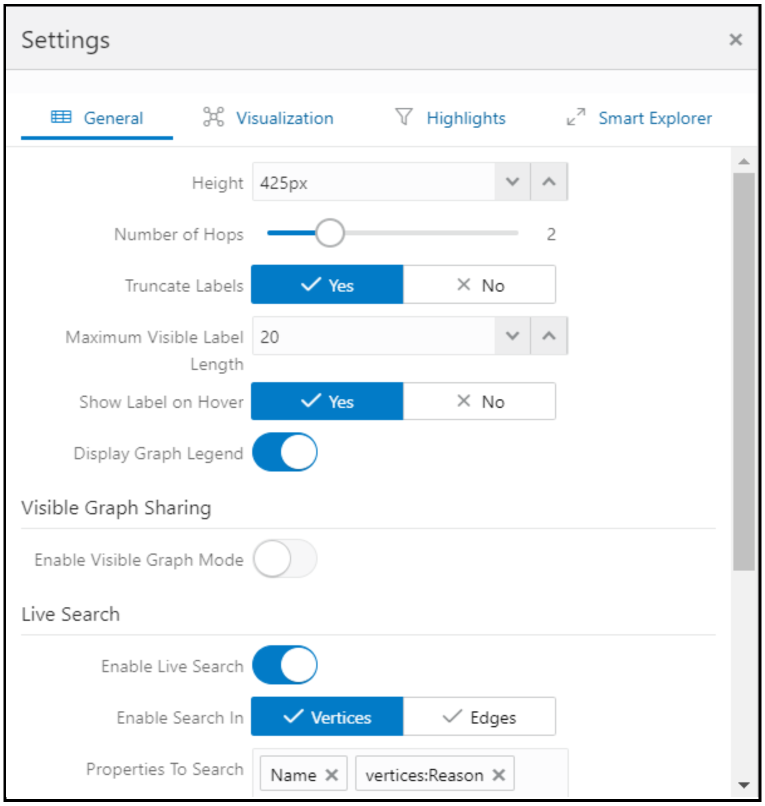
Is similar to

id	1580182
source	1040000230000107
target	10400056019672
Label	Is similar to
Match Score	81%
Match Weight	80.83664
label	searched_entity_is_similar_to_icij_paradise_external_entity

3.4.6 Graph Customization and Settings

The graph can be customized based on your visual needs, the graph customization and settings are based on the configuration in the graph settings, such as **General**, **Visualization**, **Highlights**, and **Smart Explorer**.

The graph settings can be accessed by clicking ⚙ **Settings** in the paragraph. You can specify the properites in the **Settings** window to customize your graphs.



Height: 425px

Number of Hops: 2

Truncate Labels: [✓ Yes] No

Maximum Visible Label Length: 20

Show Label on Hover: [✓ Yes] No

Display Graph Legend: ☑

Visible Graph Sharing

Enable Visible Graph Mode: ☐

Live Search

Enable Live Search: ☑

Enable Search In: [✓ Vertices] Edges

Properties To Search: Name ×, vertices:Reason ×

3.5 About Templates

The FCC Studio offers different formats to view the result after the execution of a Paragraph. Templates enable you to define parameters to customize the result formats. You can customize the visualization of the result by defining parameters in a template and then applying that template to a notebook.

> **NOTE**
>
> - FCC studio comes with a default template but users can customize this at template level but can also override any global template settings in a given notebook paragraph
> - It is recommened to use the template that is available from out of the box FCC Studio.

The customized parameters in the template are applied to the result format in the notebook. You can perform the following actions in the **Templates** page.

- Creating a Template
- Impoting and Exporting a Template
- Updating an Existing Template
- Applying Template to a Notebook

3.5.1 Accessing the Template page

Templates can be created and used across paragraphs in a Notebook based on your visulaization requirements. To create and perform various actions on the template page, you need to access the **Templates** page.

To access the Template page, follow these steps:

1. In the FCC Studio Workspace, click **Navigation Menu** on upper-left corner. The menu items are listed.
2. Click **Templates**. The **Templates** page is displayed.

3.5.2 Creating a Template

You can create a template that you can use for the notebook paragraphs.

To create a template, follow these steps:

1. In the **Template** page, click **Create**. The **Create Template** dialog box is displayed.
2. Specify the **Name** for the template and click **Create**.

   A new template is created and listed on the LHS. The details of the template are displayed on the RHS.
3. Select the required result format icon on the RHS to define parameter values for that format.
4. Specify the required values. For every result format in a template, you can define values for

   General, **Visualization**, and **Text parameters**.
5. Click **Update**. A template is created with the defined parameters.

3.5.3 Importing and Exporting a Template

You can create template by performing template import to use them in your notebooks and template exports to store and use it elsewhere.

1. To import a template, in the **Templates** page, click **Create**.
2. In the **Import Template (s)** dialog box, drap and drop the template that you want to import and click **Import**.
3. To export a template, in the **Template** page, open a template that is already created and click **Export**, below the page next to the **Update** option.
4. Specify the location where you want to export the template and click **Export**.

3.5.4 Updating an Existing Template

To update an existing template, follow these steps:

1. In the **Templates** page, select the template that you want to update from the list displayed on the LHS. The details of the selected template are displayed on the RHS.
2. Modify the required values for **General**, **Visualization**, and **Text parameters**.
3. Click **Update**. The template is updated.

3.5.5 Applying Template to a Notebook

To apply a template to a Notebook, follow these steps:

1. Navigate to the **Notebook** page you want to apply a template.
2. Click the icon in the upper-right corner. The available templates are listed.
3. Select the required template that you want to apply to the notebook. The selected template is applied to the notebook.

3.6 About Interpreters

Data Studio consists of interpreters, which can execute code in different languages. Each interpreter has a set of properties that can be adjusted and will be applied across all notebooks.

For example, by using the *python-interpreter*, it is possible to change between versions, whereas the *jdbc-interpreter* offers to customize the url, schema, or credentials.

To navigate to the **Interpreters** page, in the FCC Studio Workspace, click **Navigation Menu** on the upper-left corner, and then click **Interpreters**.

As multiple users need different settings, the concept of variants is introduced. These variants are linking user-defined settings to an interpreter and can be used independent from each other. It can be used in a paragraph by postfixing the interpreter name such as: *interpreter.variant-name*. It is possible to use multiple variants in the same notebook.

For more information about Interpreters and its configuration see, OFS Crime and Compliance Studio Administration and Configuration Guide.

3.7 About Tasks

The **Tasks** page lists all tasks of FCC Studio and displays detials such as, notebook, paragraph, interpreter, and user associated with the task.

> **NOTE** **Task** is available to the users based on the permissions in FCC Studio.

To navigate to the **Tasks** page, in the FCC Studio Workspace, click **Navigation Menu** on upper-left corner, and then click **Tasks**.

Following are the detials that are available in the Tasks page:

- **Task Status**

  A task can have one of seven possible statuses such as, *created*, *queued*, *running*, *rejected*, *success*, *cancelled*, and *error*. The tasks page displays the satus of the task that are associated with the notebook.

- **Task Coulmn**

  The available columns in the **Tasks** page are, **Notebook**, **Paragraph**, **Interpreter**, **Status**, **Creation Time**, **Queue Time**, **Run Time**, and **User.**

- **Tasks Filter**

  The filters that can be used to filter the results. The availble filters are based on the **Task Status** and **Date and Time**.

For more information about tasks and its configuration see the *Managing Tasks* section in OFS Crime and Compliance Studio Administration and Configuration Guide.

3.8 About Ruleset

A Ruleset is a set of rules that are applied to the defined source and target entities, compares the attributes of the entities to derive a match. The ruleset is provided for resolving entities and is used for creating the graph. This rule set enables you to pull the required data based on the threshold that you specify when configuring the ruleset.

> **NOTE** **Ruleset** can be configured only by the Administrator.

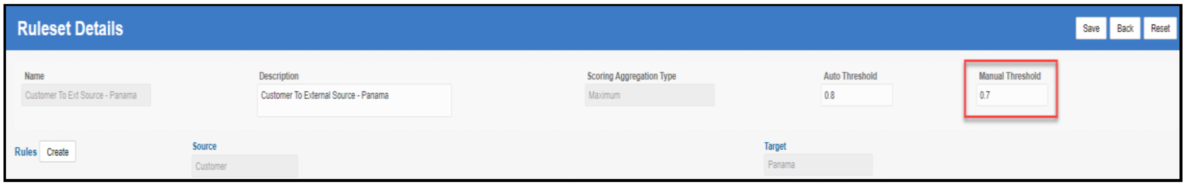
To navigate to the **Ruleset** page, in the FCC Studio Workspace, click **Navigation Menu** on the upper-left corner, and then click **Ruleset**. For more information about creating rulesets, on matching rulesets, see the *Managing Rulesets* in OFS Crime and Compliance Studio Administration and Configuration Guide.

You can also see the Financial Crime Graph Model Matching Guide to know the specific parameters that you can set when applying the ruleset.

3.9 About Manual Decisioning

The **Manual Decisioning** in FCC Studio provides you the ability to make manual decisions on similarity edges which can be marked as required for your graph analysis apart from the threshold that is set in the **Ruleset** page. This flexibilty provides an Ad-hoc experience to improve the graph analytic capability and identify the match to analze risk and take required actions.

You can specify your **Manual Threshold** in the **Ruletset** page for Manual Decisioning.

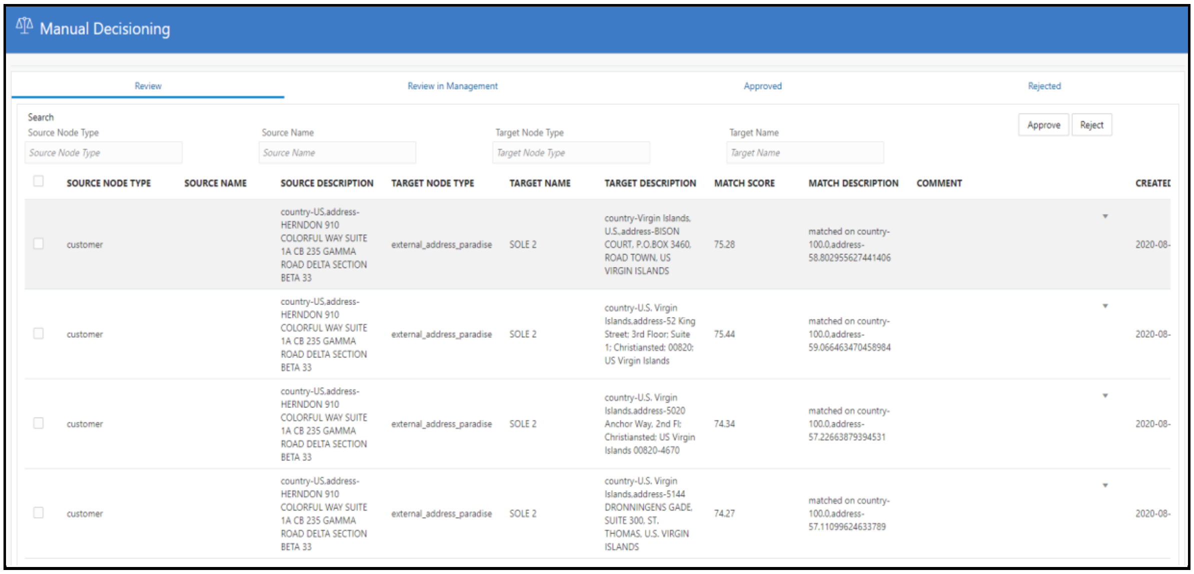


Ruleset Details

Name: Customer To Ext Source - Panama
Description: Customer To External Source - Panama
Scoring Aggregation Type: Maximum
Auto Threshold: 0.8
Manual Threshold: 0.7

Source: Customer
Target: Panama

To navigate to the **Manual Decisioning** page, in the FCC Studio Workspace, click **Navigation Menu** on the upper-left corner, and then click **Manual Decisioning**.



□	SOURCE NODE TYPE	SOURCE NAME	SOURCE DESCRIPTION	TARGET NODE TYPE	TARGET NAME	TARGET DESCRIPTION	MATCH SCORE	MATCH DESCRIPTION	COMMENT	CREATEI
□	customer		country-US.address-HERNDON 910 COLORFUL WAY SUITE 1A CB 235 GAMMA ROAD DELTA SECTION BETA 33	external_address_paradise	SOLE 2	country-Virgin Islands, U.S.,address-BISON COURT, P.O.BOX 3460, ROAD TOWN, US VIRGIN ISLANDS	75.28	matched on country-100.0,address-58.802955627441406	▾	2020-08-
□	customer		country-US.address-HERNDON 910 COLORFUL WAY SUITE 1A CB 235 GAMMA ROAD DELTA SECTION BETA 33	external_address_paradise	SOLE 2	country-U.S. Virgin Islands,address-52 King Street; 3rd Floor; Suite 1; Christiansted; 00820; US Virgin Islands	75.44	matched on country-100.0,address-59.066463470458984	▾	2020-08-
□	customer		country-US.address-HERNDON 910 COLORFUL WAY SUITE 1A CB 235 GAMMA ROAD DELTA SECTION BETA 33	external_address_paradise	SOLE 2	country-U.S. Virgin Islands,address-5020 Anchor Way, 2nd Fl; Christiansted; US Virgin Islands 00820-4670	74.34	matched on country-100.0,address-57.22663879394531	▾	2020-08-
□	customer		country-US.address-HERNDON 910 COLORFUL WAY SUITE 1A CB 235 GAMMA ROAD DELTA SECTION BETA 33	external_address_paradise	SOLE 2	country-U.S. Virgin Islands,address-5144 DRONNINGENS GADE, SUITE 300, ST. THOMAS, U.S. VIRGIN ISLANDS	74.27	matched on country-100.0,address-57.11099624633789	▾	2020-08-

The **Manual Decisioning** page is displayed in a tabular format, where the manual edges source information is available for reviewing and adding to the graph for analysis. It has the following tabs:

- **Review**

  The **Review** tab displays the information, that can be Approved or Rejected.

- **Review in Management**

  The **Review in Management** tab displays the sources that are approved and sent by the user. The Administrators can review these details and further approve or reject.

  This tab has the following tabs that catergorizes the sources based on approval and rejection:

  - **Review for Acceptance**

    In this tab, the sources that are submitted by the users are further validated and if it the sources are accepted, then they are move to the **Accepted** tab. If the sources are rejected, then they are moved to the **Review for Rejection** tab.

  - **Review for Rejection**

    In this tab, the sources that are rejected from the **Review for Acceptance** tab are displayed. You can further validate these source to either approve or review the sources again.

  > **NOTE** The **Review in Management** tab is available only for the users with the `request_approve` and `review_request` permissions.

- **Approved**

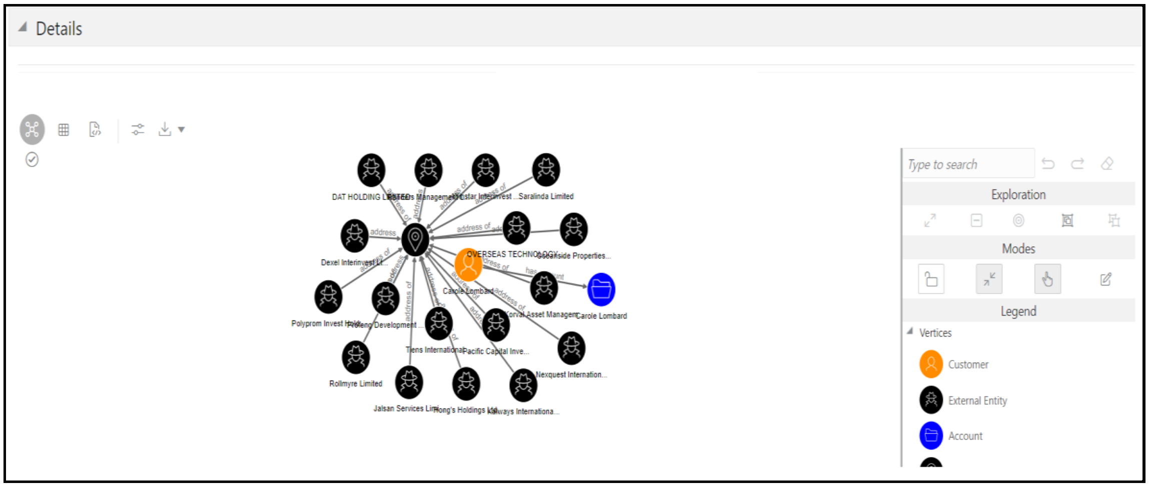
  The **Approved** tab displays the sources that are approved.

- **Rejected**

  The **Rejected** tab displays the sources that are rejected.

3.9.1 Viewing the Edges from the Source

In the **Manual Decisioning** page, when you select the source from the **Review** tab. The **Details** pane is populated below the screen. In the **Details** pane, you can view the edge and validate it to include in your graph.

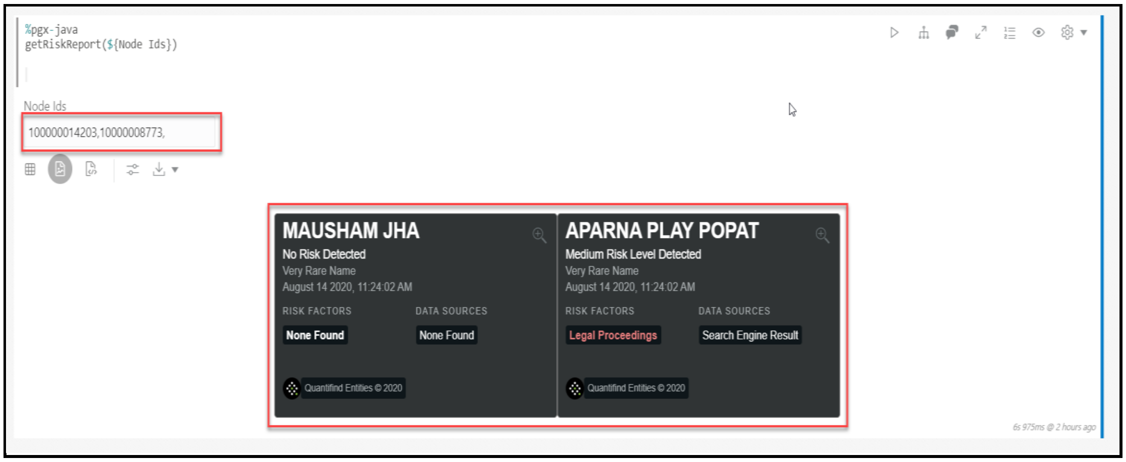


3.10 Quantifind Risk Report

Quantifind integration for real-time risk reports in FCC Studio enables the Financial institutions to discover signals of revenue drivers and risk, including fraud and money.

> **NOTE** These reports are only available for FCC Studio integration with ECM.

In the ECM notebook, by using the java interpreter you can run the risk report query as shown in the figure.



```
%pgx-java
getRiskReport(${Node Ids})
```

Node Ids: 100000014203,10000008773,

MAUSHAM JHA
No Risk Detected
Very Rare Name
August 14 2020, 11:24:02 AM
RISK FACTORS: None Found
DATA SOURCES: None Found
Quantifind Entities © 2020

APARNA PLAY POPAT
Medium Risk Level Detected
Very Rare Name
August 14 2020, 11:24:02 AM
RISK FACTORS: Legal Proceedings
DATA SOURCES: Search Engine Result
Quantifind Entities © 2020

6s 975ms @ 2 hours ago

The results are displayed as a card, which displays the risk status of the identified node details. Based on the risk you can perform the required measures for the risk analysis.

3.11 Dynamic Forms

Dynamic Forms in FCC Studio are the dynamic input fields that can be created by you for the paragraphs. When the pragraphs with the dynamic input fields are exectued, the existing parapraph properites are replaced with these dynamic values and the results are displayed.

The following are the supported input fields in FCC Studio:

- Textbox
- Select
- Slider
- Checkbox
- Date Picker
- Time Picker
- DateTime Picker

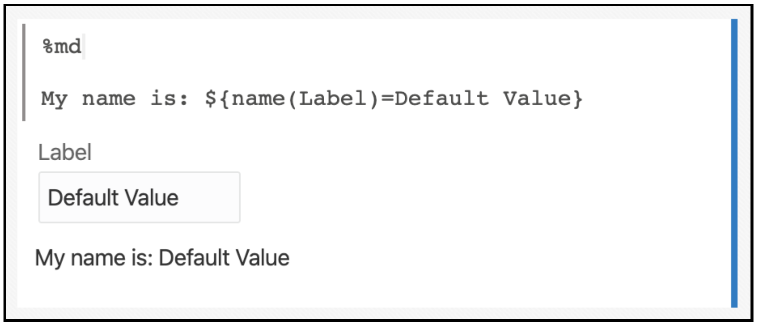
The following sections in this chapter provides the details about how you can specify the dynamic parameters for the exisitng fields.

3.11.1 Textbox

The textbox offers a simple user input that accepts any characters.

- The name of the user-input can be defined using `${name}`.
- A customized label can be set using `${name(Label)}`.
- A default value can be assigned using the equals operator `${name(label)=Default Value}` or `${name=Default Value}`.

For example, `${name(Label)=Default Value}`



```
%md

My name is: ${name(Label)=Default Value}
```

My name is: Default Value

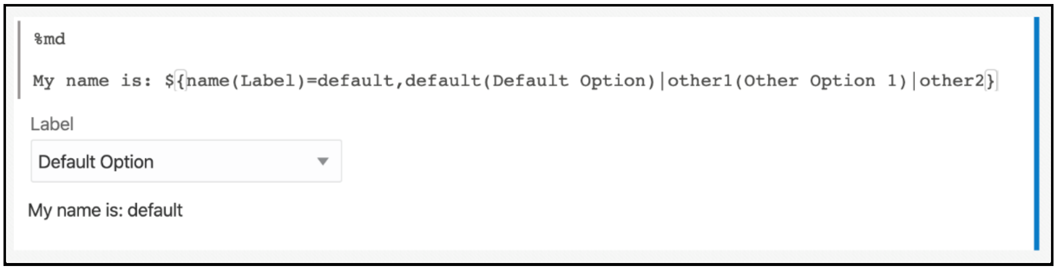
3.11.2 Select

A select-field allows you to choose from a given selection.

- The name, label and default value can be set similar to the textbox.
- Options follow the "`, (comma)`" after the default assignment `name=default,default|other1|other2` and so on.

- The label of an option can be added in the brackets following the option `other1 (Other Option 1)`.

For example: `${name(Label)=default,default(Default Option)|other1(Other Option 1)|other2}`



```
%md
My name is: ${name(Label)=default,default(Default Option)|other1(Other Option 1)|other2}
```

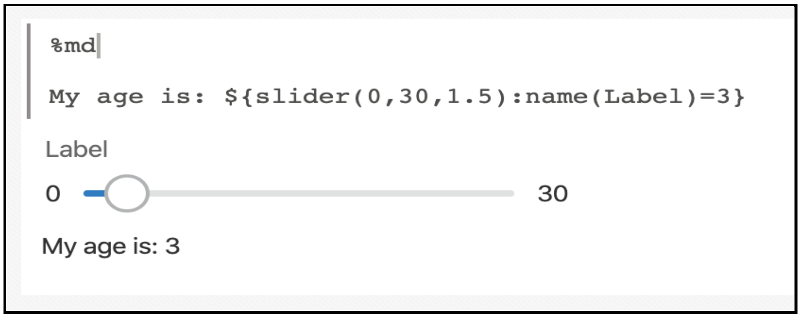
My name is: default

3.11.3 Slider

A slider allows you to choose from a given range.

- The first part of the slider dynamic form is the keyword `slider`, followed by brackets containing the values for minimum, maximum and step size: `slider(<min>,<max>,<step size>)`.
- Then, after a `:` (colon), the name, label, and default value can be set similar to the textbox.

For example: `${slider(0,30,1.5):name(Label)=3}`



```
%md
My age is: ${slider(0,30,1.5):name(Label)=3}
```

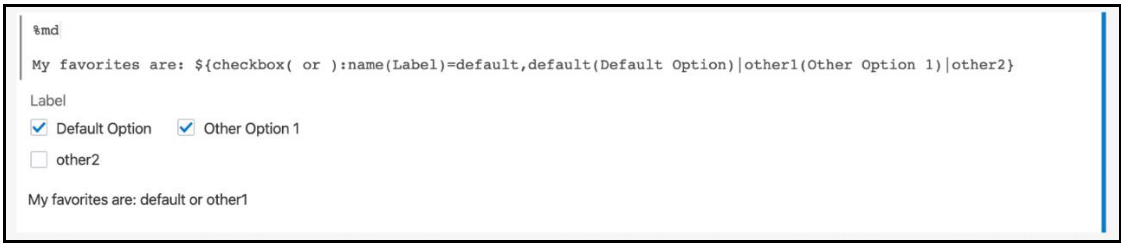
My age is: 3

3.11.4 Checkbox

A checkbox allows you to choose from a given selection.

- The first part of the checkbox dynamic form is the keyword `checkbox`, followed by brackets containing the values join parameter of the options: `slider(<join parameter>)`.
- The join parameter is used between all selections and only shows up if at least two elements are selected.
- Then, after a `:` (colon), the name, label, default value, and options can be set similar to the select dynamic form.

For example: `${checkbox( or ):name(Label)=default,default(Default Option)|other1(Other Option 1)|other2}`



```
%md
My favorites are: ${checkbox( or ):name(Label)=default,default(Default Option)|other1(Other Option 1)|other2}
```

Label

☑ Default Option ☑ Other Option 1

□ other2

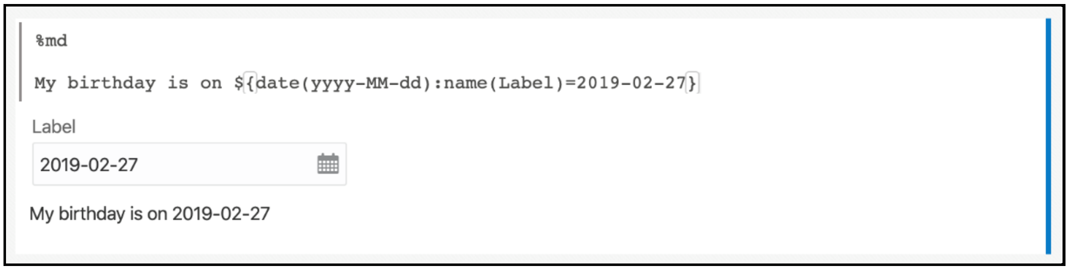
My favorites are: default or other1

3.11.5 Date Picker

A date-picker allows you to choose from a given date.

- The first part of the checkbox dynamic form is the keyword `date`, followed by brackets containing the output format, that is `date(day)` or `date(yyyy-MM-dd)`.
- The name, label, and default value can be set similar to the textbox.

For example: `${date(yyyy-MM-dd):name(Label)=2019-02-27}`



```
%md
My birthday is on ${date(yyyy-MM-dd):name(Label)=2019-02-27}
```

Label

2019-02-27

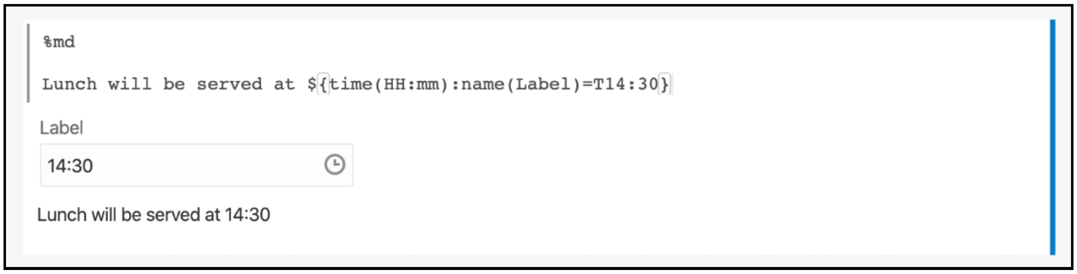
My birthday is on 2019-02-27

3.11.6 Time Picker

A time picker allows you to choose from a given time.

- The first part of the checkbox dynamic form is the keyword `time`, followed by brackets containing the output format, that is `time(HH:mm)` or `time(HH)`.
- The name, label, and default value can be set similar to the textbox.

For example: `${time(HH:mm):name(Label)=T13:30}`.



```
%md
Lunch will be served at ${time(HH:mm):name(Label)=T14:30}
```

Label

14:30

Lunch will be served at 14:30

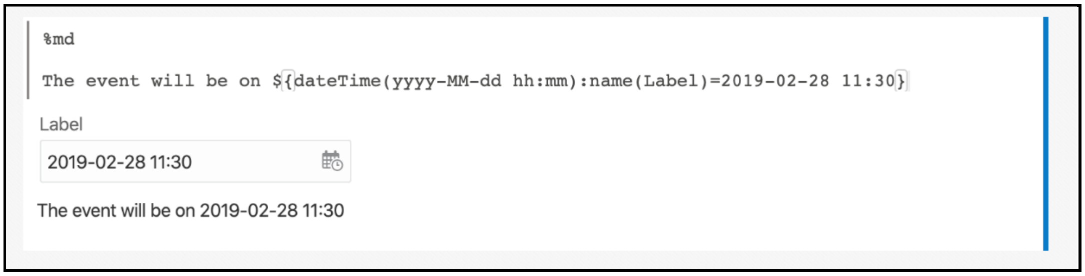
3.11.7 DateTime Picker

A datetime picker allows you to choose from a given date and time.

- The first part of the checkbox dynamic form is the keyword `dateTime`, followed by brackets containing the output format, that is `dateTime (YYYY-MM-dd hh:mm)`.

- The name, label, and default value can be set similar to the textbox.

For example: `${dateTime(yyyy-MM-dd hh:mm):name(Label)=2019-02-28 11:30}`.



```
%md

The event will be on ${dateTime(yyyy-MM-dd hh:mm):name(Label)=2019-02-28 11:30}
```

Label

2019-02-28 11:30

The event will be on 2019-02-28 11:30

3.11.8 Setting Values Through a Notebook URL

In the web interface, dynamic form values can also be set by passing a name-value pair as parameter of a notebook URL. For example, to set the value of the dynamic form named `myDynamicForm` to `FCCStudio`, the parameter `&form_myDynamicForm=FCCStudio` must be added to the URL. If there are multiple dynamic forms with the same name in the same notebook, all of them will be set.

> **NOTE** It is recommend not to pass sensitive information via the URL. If you have sensitive information, you can use the REST API to change the values of a dynamic form.

4 Scenario Notebooks

The FCC Studio has a collection of Out-of-the-Box (OOB) Notebooks that are packaged along with the license. Following are the Notebooks that are available in FCC Studio:

- Financial Crime Graph Patterns
- Example Scenario Notebooks

4.1 Financial Crime Graph Patterns

Finacial Crime Graph Pattern notebook provides sample graph that are created from a synthetic data set of financial transactions. There are 7 different label vertices such as, CUSTOMER, ACCOUNT, DERIVED ENTITY, EXTERNAL ENTITY, EXTERNAL ADDRESS, CASE, EVENT, and INSTITUTION. The graph edges are classified into following categories:

- **has event**
- **entity on**
- **event on**
- **is similar to**
- **is related to**
- **has account**
- **cash transaction**
- **MI transaction**
- **wire transaction**
- **send MI/wire**
- **receive MI/wire**
- **MI/wire transfer**
- **has address**

This notebook provides patterns such as Transfer of money through an external entity, Transfer of money to external entity, and High risk accounts transacting with counter-parties in high risk geographies.

For example, the and High risk accounts transacting with counter-parties in high risk geographies pattern identifies the following:

- Accounts that are alerted
- Accounts that have a high activity risk
- Accounts that are also located in high risk areas.

The high risk geography determination is based on the geographic risk value of the addresses that the related to a counterparty.

4.2 Example Scenario Notebook

The scenario notebooks such as, HRG and RMF are available for you to create notebooks which are can be baselined based on these notebooks as an example and create different AML scenarios using different interpreters and be able to push events to ECM.

4.2.1 HRG Scenario – AC Focus

The notebook contains the High-Risk Geography scenario logic defined in Scala and Spark SQL. You can use this notebook to visualize the financial data flow in high risk geographical area.

4.2.2 RMF Account (SQL)

The notebook contains the Rapid Movements of Fund (RMF) - Account scenario. The scenario logic is defined in the Oracle SQL query. You can run the RMF Account scenario using the JDBC Interpreter. This interpreter uses Oracle (RDBMS) as the source and target database to create the required temporary tables.Temporary tables are used in the scenario development process.

4.2.3 RMF Scenario – AC Focus

The notebook contains the Rapid Movements of Fund (RMF) - Account Focus scenario. The scenario logic is defined in the Oracle SQL query.

4.2.4 RMF Scenario – CUS Focus

The notebook contains the Rapid Movements Of Fund (RMF) - Customer Focus scenario. The scenario logic is defined in Scala and Spark SQL.

5 Use Cases

In real world, it is difficult to not just detect but discover the new patterns that the criminals are executing everyday. These patterns are not just new, but this could be used for years and are difficult to identify with traditional techniques.

To execute your use cases and scenarios, you can create a notebook and assume the notebook that is created is your use case at a granular level.

The scenarios in this chapter will walk you through to quickly build your notebook by using the FCC Studio based on the common scenarios that are explained in the following sections and get a quick insight.

5.1 Anti-Money Laundering Pattern Data Discovery

Anti-money laundering (AML) refers to the laws, regulations and procedures intended to prevent criminals from disguising illegally obtained funds as legitimate income. The AML Pattern Data Discovery is a scenario that helps you to see new financial crime patterns. These patterns enables you to easily explore any relationship among entities in real time to extract valuable insights from your data.

This data discover pattern quickly uncover emerging, complex money laundering and terrorist financing threats with network and entity generation processes that automatically build network diagrams and reveal hidden relationships. The advanced graph analytics enables entity resolution by looking at multiple data sources and references to a customer, then accounting for inconsistencies, errors, abbreviations and incomplete records to help determine whether they relate to the same entity.

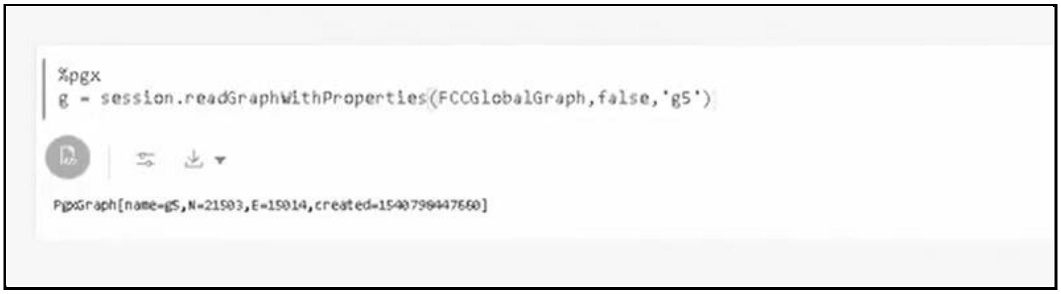
The scenario includes the following actions that you can perform for the data discovery:

- Creating a Notebook for Your Financial Crime Discovery
- Loading a Financial Crime Graph
- Insight to Customize and Arrange Data in a Graph
- Customizing the Nodes and Edges of the Graph

5.1.1 Creating a Notebook for Your Financial Crime Discovery

The first step in any scenario is to create a notebook. In this use case, let's create a notebook to discover your financial crime. To create a new notebook, follow these steps:

1. Create a Notebook and specify the notebook details.
2. After the notebook is created, create a Paragraph using the **PGX Interprter**.

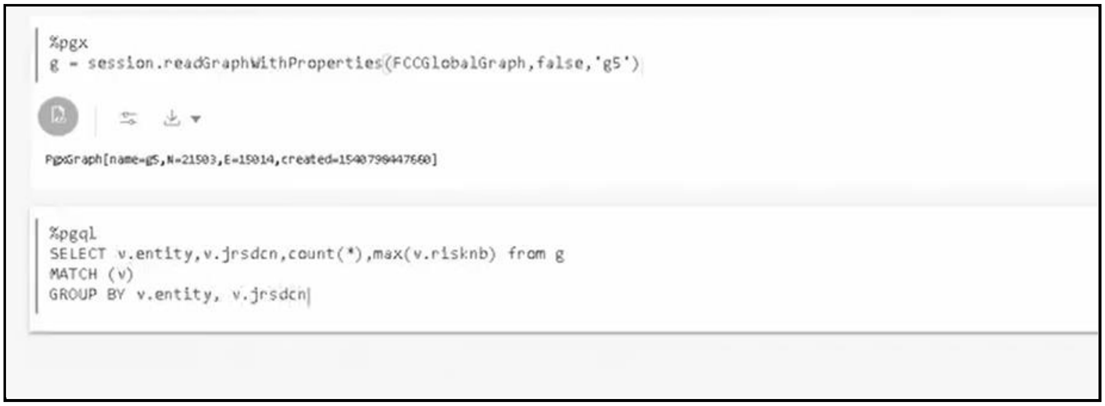


```
%pgx
g = session.readGraphWithProperties(FCCGlobalGraph,false,'g5')
```

```
PgxGraph[name=g5,N=21503,E=15014,created=1540798447660]
```

In this example, the **PGX Interpreter** is used to create a **PGX query** to load the graph. You must specify the session information that you want to retreive the data. The query is executed and the graph details such as, graph name, edges, and nodes or vertices are displayed. Further to understand the pattern detection, you must understand the details of the node, vertices, and edges within the displayed data.

3. For instance, you can specify the matrices such as, what are the vertices and edges that exists within the data.



```
%pgx
g - session.readGraphWithProperties(FCCGlobalGraph,false,'g5')
```

```
PgxGraph[name=g5,N=21503,E=15014,created=1540798447660]
```

```
%pgql
SELECT v.entity,v.jrsdcn,count(*),max(v.risknb) from g
MATCH (v)
GROUP BY v.entity, v.jrsdcn
```

In this example, the parameters included are:

- SELECT

  The `code v.entity,v.jrsdcn,count(*),max(v.rishnb) from g`, fetches the jurisdiction count and risk numbers within the graph **g**, that is built from the previous paragraph.

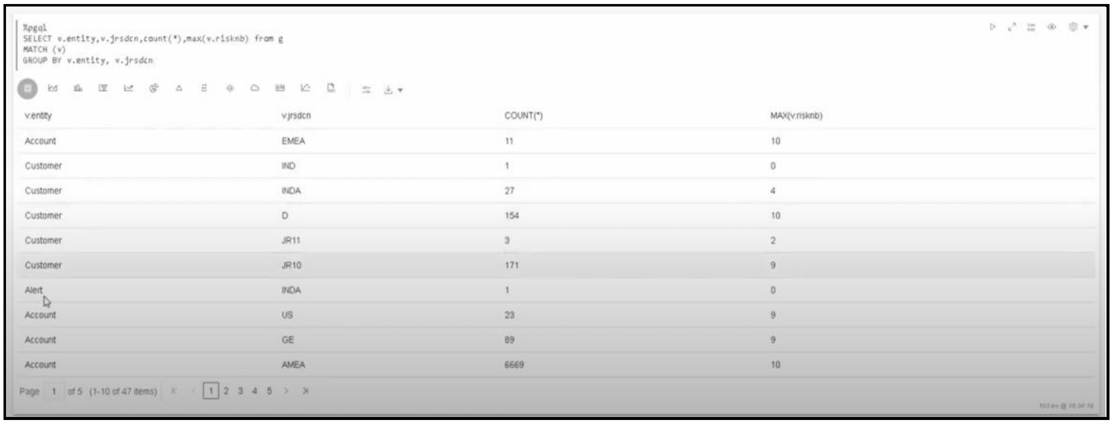
- MATCH

  Matching the vertices.

- GROUP BY

  Grouping the entities by vertices and jurisdiction.

4. Click **Execute Paragraph**. The entities data such as, the jurisdiction, country, risk numbers are displayed.



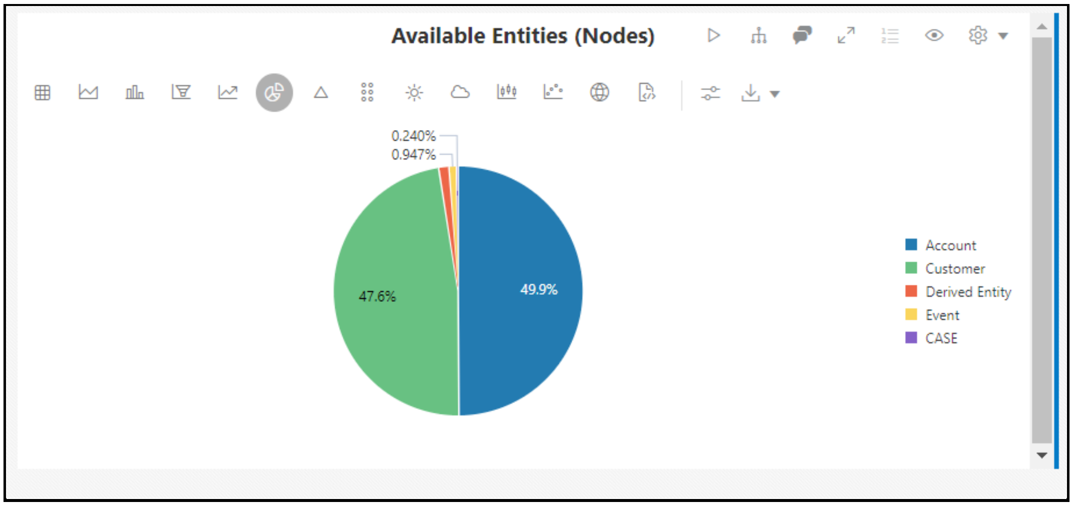
```
%pgql
SELECT v.entity,v.jrsdcn,count(*),max(v.risknb) from g
MATCH (v)
GROUP BY v.entity, v.jrsdcn
```

v.entity	v.jrsdcn	COUNT(*)	MAX(v.risknb)
Account	EMEA	11	10
Customer	IND	1	0
Customer	INDA	27	4
Customer	D	154	10
Customer	JR11	3	2
Customer	JR10	171	9
Alert	INDA	1	0
Account	US	23	9
Account	GE	89	9
Account	AMEA	6669	10

Page 1 of 5 (1-10 of 47 items) 1 2 3 4 5

5. You can click the visualization icons within the paragraph, to view the details in the visual form to make it more consumable for you.

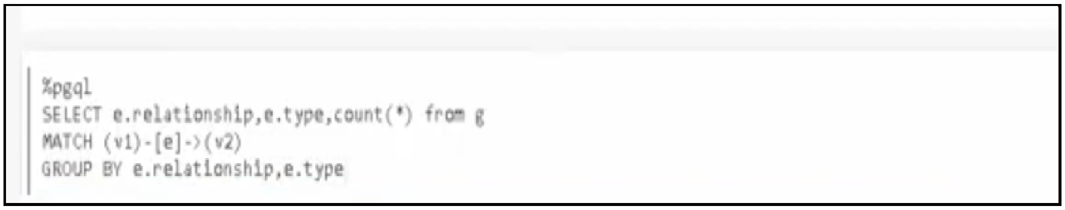
6. You can hide the code and add a title to the data that is fetched to make it presentable, consumable, and sharable with others.



The visualizations that are displayed are easy to consume providing you the flexibility to choose the type of visual. The visualization in this case is a **Pie Chart**, which enables you to see the entites associated by its type based on the percentage.

You can use the other visulaization option to see various types of visuals and different visualization formats.

7. After the first level execution, execute further to understand the relationships that exists between the edges by writing the relationship query.



```
%pgql
SELECT e.relationship,e.type,count(*) from g
MATCH (v1)-[e]->(v2)
GROUP BY e.relationship,e.type
```

In this example, the parameters included are:

- SELECT

  The `code e.relationship,e.type,coun(*)from g`, fetches the relationship count within the graph **g,** that is build from the first paragraph.

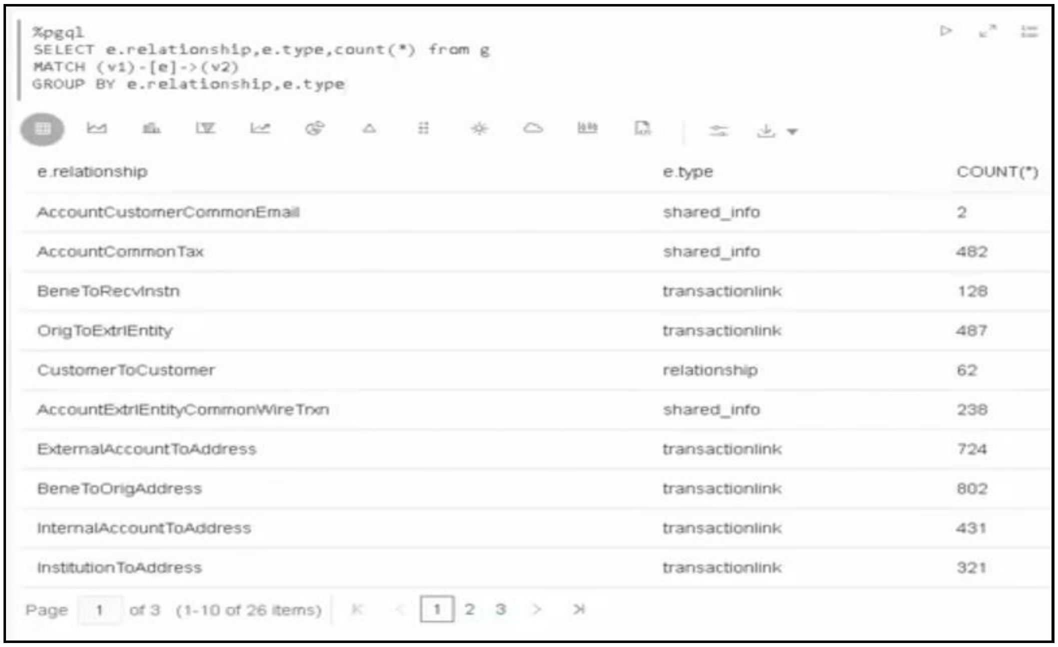
- MATCH

  Matching the vertices "v1" and "v2" and the edge is "e". You must write the vertices within parenthesis and edges within brackets to fetch the PGQL query.

- GROUP BY

  Grouping the entities by relationship and type.

8. Click **Execute Paragraph**. The entities data such as, the jurisdiction, country, risk numbers are displayed.

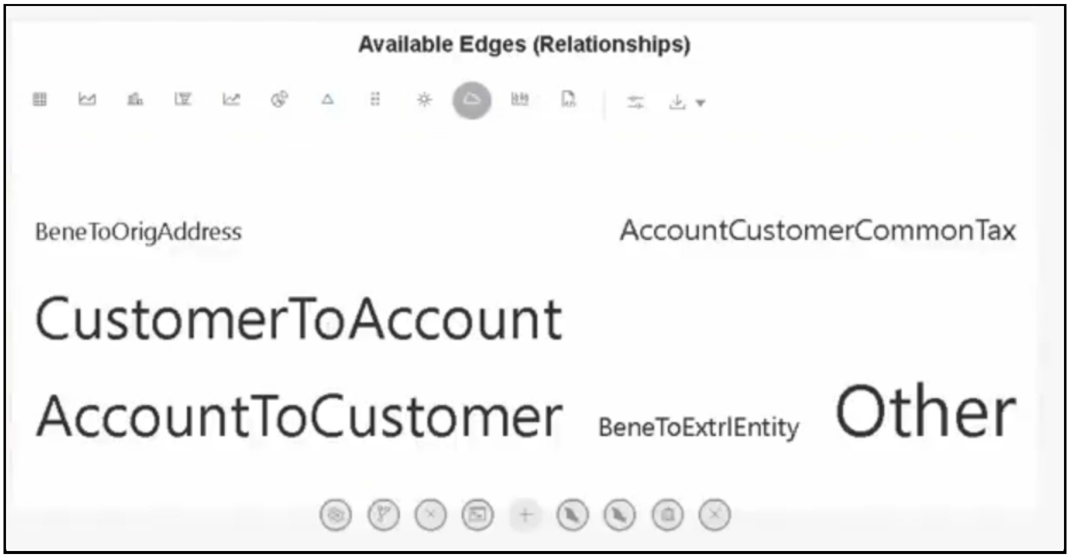


```
%pgql
SELECT e.relationship,e.type,count(*) from g
MATCH (v1)-[e]->(v2)
GROUP BY e.relationship,e.type
```

e.relationship	e.type	COUNT(*)
AccountCustomerCommonEmail	shared_info	2
AccountCommonTax	shared_info	482
BeneToRecvInstn	transactionlink	128
OrigToExtrlEntity	transactionlink	487
CustomerToCustomer	relationship	62
AccountExtrlEntityCommonWireTrxn	shared_info	238
ExternalAccountToAddress	transactionlink	724
BeneToOrigAddress	transactionlink	802
InternalAccountToAddress	transactionlink	431
InstitutionToAddress	transactionlink	321

Page 1 of 3 (1-10 of 26 items) 1 2 3

9. You can click the visualization icons within the paragraph, to view the details in the visual form to make it more consumable for you. Each paragraph that is executed is a rest services which provides you a flexibility to run the executing based on the updated service. You can hide the code and add a title to the data that is fetched to make it presentable, consumable, and sharable with others.



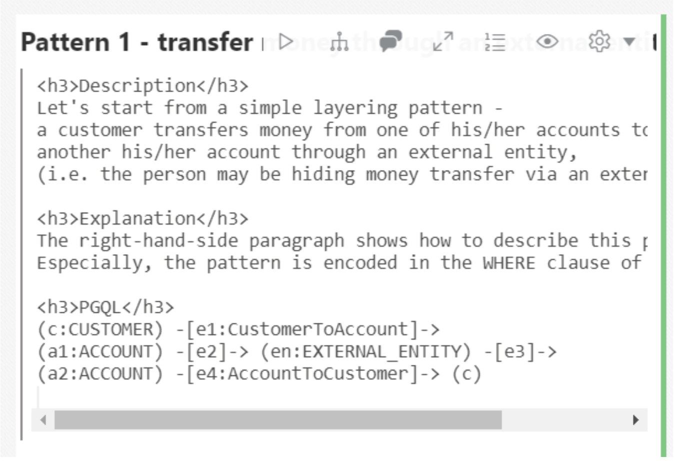
Available Edges (Relationships)

5.1.2 Loading a Financial Crime Graph

To make your pattern even more readable you can write a plain text in a paragraph and then simultanously write the query in another paragraph and seemlesly produce a readable graph insight. This pattern can be a pattern that an investor or an analyst has provided.

To create a plain text paragraph, perform the following steps:

1. Create a paragraph and write the description of the analysis or your use case. For example, refer the figure.



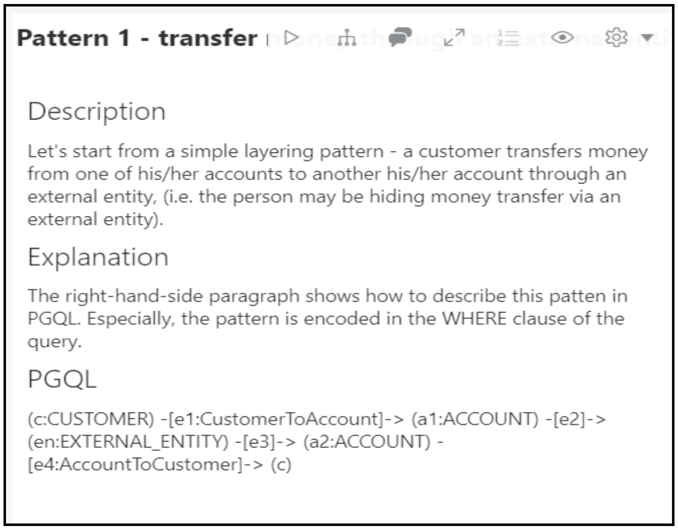
Pattern 1 - transfer

```
<h3>Description</h3>
Let's start from a simple layering pattern -
a customer transfers money from one of his/her accounts tc
another his/her account through an external entity,
(i.e. the person may be hiding money transfer via an exter

<h3>Explanation</h3>
The right-hand-side paragraph shows how to describe this p
Especially, the pattern is encoded in the WHERE clause of

<h3>PGQL</h3>
(c:CUSTOMER) -[e1:CustomerToAccount]->
(a1:ACCOUNT) -[e2]-> (en:EXTERNAL_ENTITY) -[e3]->
(a2:ACCOUNT) -[e4:AccountToCustomer]-> (c)
```

2. After executing the paragraph, the plain text is formatted and displayed as shown.



Pattern 1 - transfer

Description

Let's start from a simple layering pattern - a customer transfers money from one of his/her accounts to another his/her account through an external entity, (i.e. the person may be hiding money transfer via an external entity).

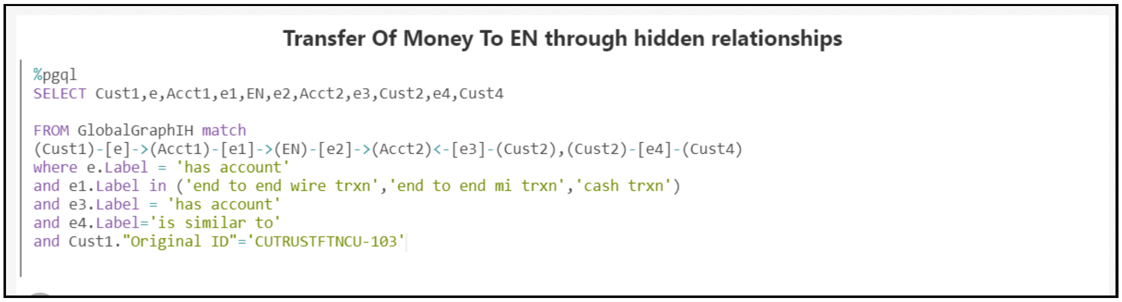
Explanation

The right-hand-side paragraph shows how to describe this patten in PGQL. Especially, the pattern is encoded in the WHERE clause of the query.

PGQL

(c:CUSTOMER) -[e1:CustomerToAccount]-> (a1:ACCOUNT) -[e2]-> (en:EXTERNAL_ENTITY) -[e3]-> (a2:ACCOUNT) -[e4:AccountToCustomer]-> (c)

As your next step, create another paragraph to run a PGQL query based on the pattern that you described in the previous paragraph. The code snippet can be provided as shown in the figure.

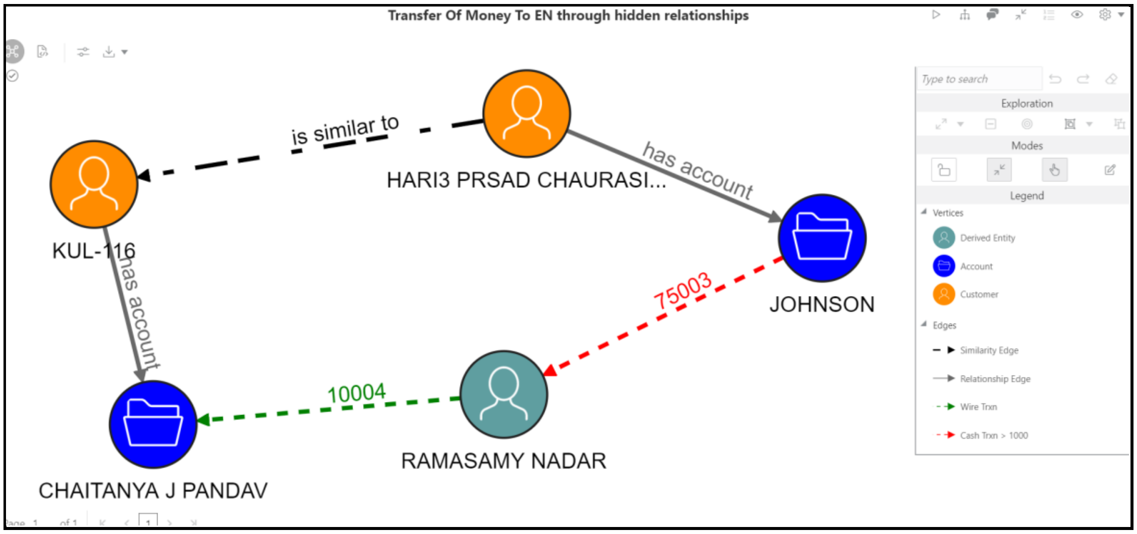


Transfer Of Money To EN through hidden relationships

```
%pgql
SELECT Cust1,e,Acct1,e1,EN,e2,Acct2,e3,Cust2,e4,Cust4

FROM GlobalGraphIH match
(Cust1)-[e]->(Acct1)-[e1]->(EN)-[e2]->(Acct2)<-[e3]-(Cust2),(Cust2)-[e4]-(Cust4)
where e.Label = 'has account'
and e1.Label in ('end to end wire trxn','end to end mi trxn','cash trxn')
and e3.Label = 'has account'
and e4.Label='is similar to'
and Cust1."Original ID"='CUTRUSTFTNCU-103'
```

3. Click **Execute Paragraph** to visualize your pattern.



You can hide the code and add a title to the data that is fetched to make it presentable, consumable, and sharable with others. You can provide more highlights (such as, labels, and vertices) and rearrange the nodes and toggle the graph to enrich your graph, you can open the paragraph in the iframe mode or expand the graph to make it more readable.

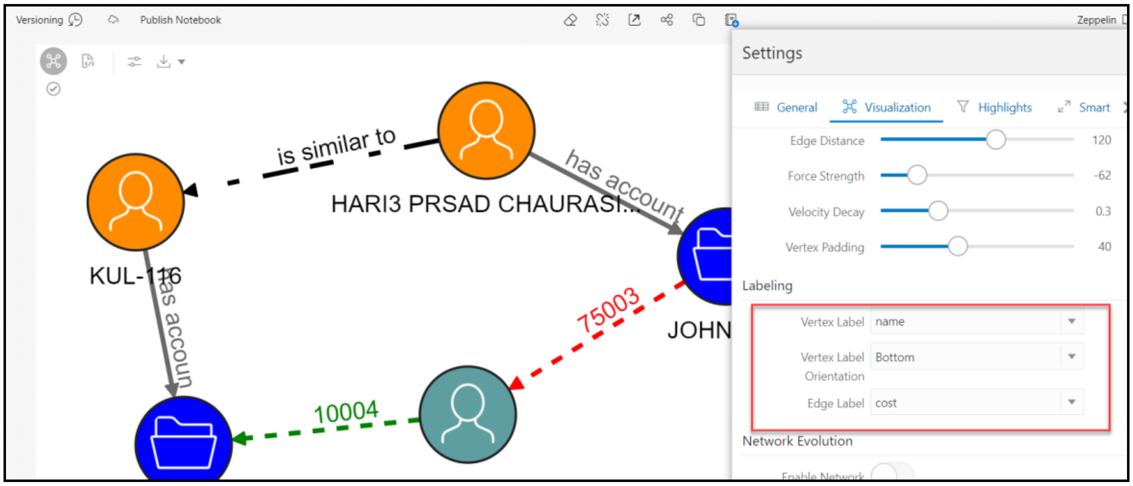
5.1.3 Insight to Customize and Arrange Data in a Graph

To enrich the graph that is executed, you can provide more highlights (such as, labels, and vertices) by using the Graph Customization options.

The graph can be customized based on your visual needs, the graph customization and settings are based on the configuration in the graph settings, such as **General**, **Visualization**, **Highlights**, and **Smart Explorer**. For more information see, Graph Customization and Settings.

The following figure is customized using **Visualization** and **Highlights** in the Graph Settings.

Providing Labeling changes, such as Name Label and Edge Label, the graph is displayed as follows.



Settings

Edge Distance 120

Force Strength -62

Velocity Decay 0.3

Vertex Padding 40

Labeling

Vertex Label: name

Vertex Label Orientation: Bottom

Edge Label: cost

Network Evolution

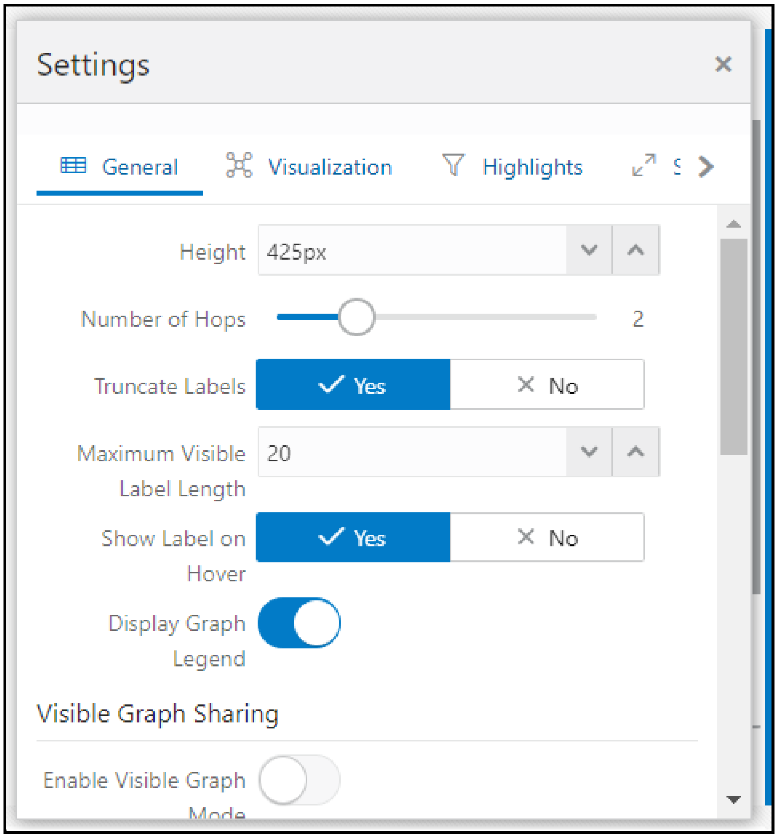
Enable Network

5.1.4 Customizing the Nodes and Edges of the Graph

The Nodes and Edges in your graphs can be customized to provide a better visualization of the data flow between the entities. To customize the Nodes and Edges, you can navigate to the graph setting and add new highlights in the **Highlights** tab.

5.1.4.1 Vertices

Vertices can be highlighted by clicking the **New Highlight**, and specify the parameters for the highights. For example, the figure shows how you can differentiate the vertices with different entities as Customer, Account, and Derived Entity.



Height 425px

Number of Hops 2

Truncate Labels ✓ Yes ✕ No

Maximum Visible Label Length 20

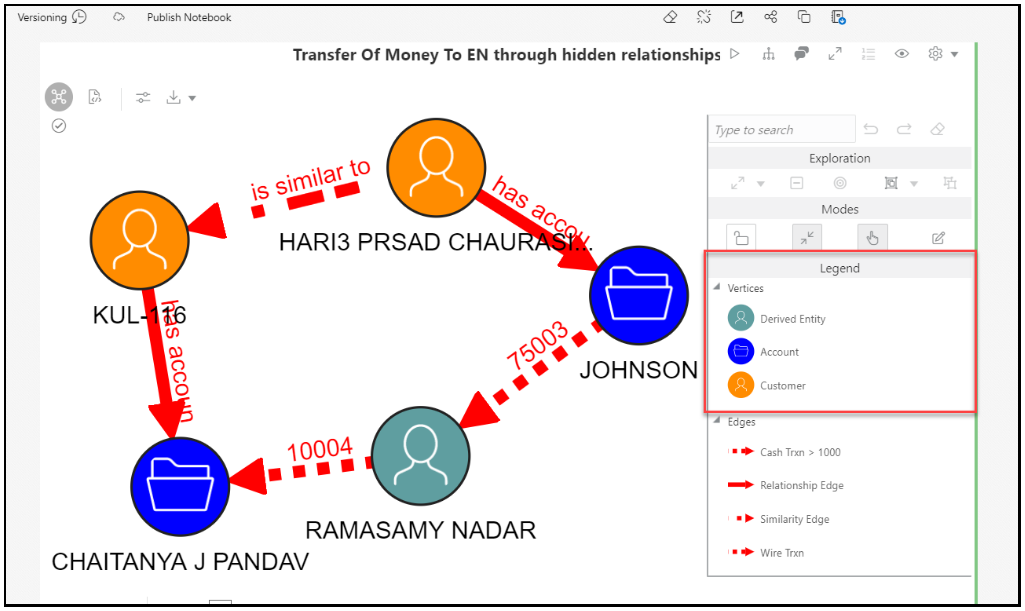
Show Label on Hover ✓ Yes ✕ No

Display Graph Legend ●

Visible Graph Sharing

Enable Visible Graph Mode ○

Based on the applied highlights, the graph changes is displayed as follows.

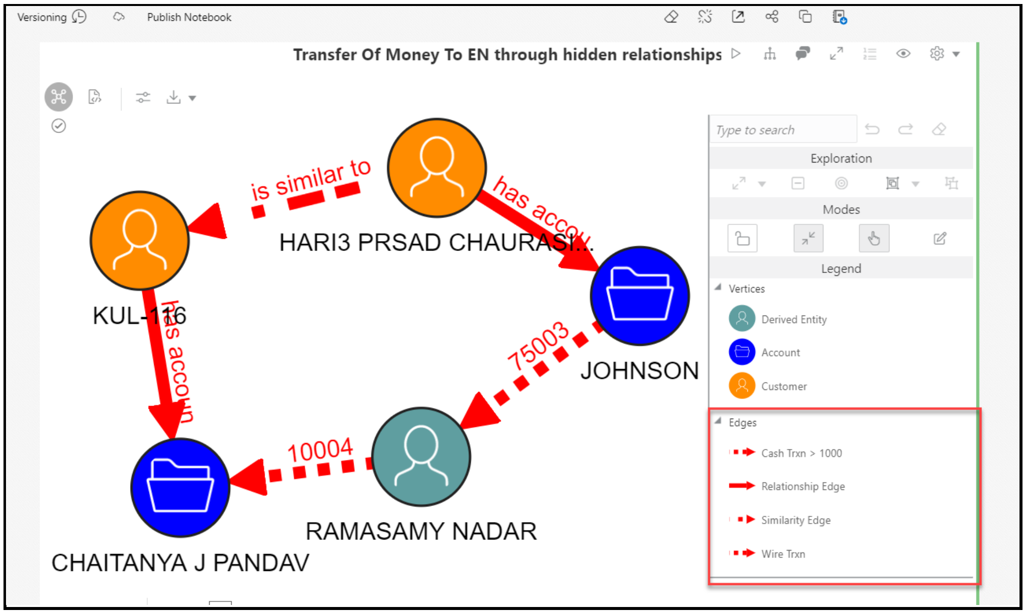


The graphs illustrates a pattern, where there is a money flow from the different entities with a hidden relationship.

5.1.4.2 Edges

Edge can be highlighted by clicking the **New Highlight**, and specify the parameters for the highights. For example, the figure shows how you can differentiate the edges with different color and animations to visulaize the flow of money from the entities.

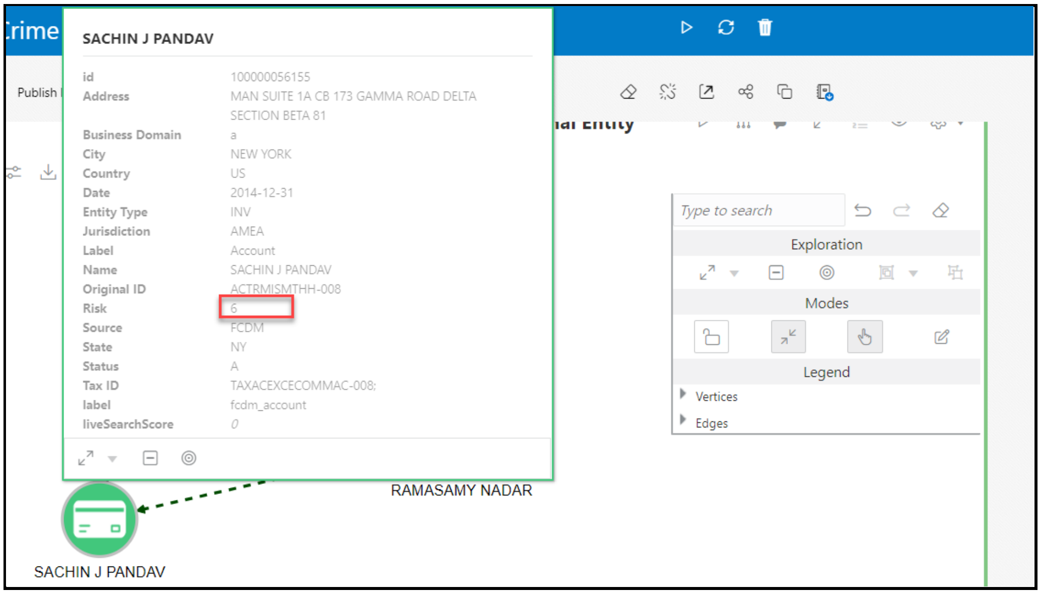
Based on the applied highlights, the graph changes is displayed as follows.



The graphs illustrates a pattern, where there is a money flow from the bank to the different accounts from a customer, with the flow of money highlighted with different colors. Blue represents the money flow from the account to the bank and red represents the money flow from the bank to the account.

You can also specify the risk numbers to this pattern and visualize your risk. For example, if we use the risk parameters as highlights.

The graphs illustrates a pattern, where there is a risk in the account and when you click for details, the risk number is displayed.



SACHIN J PANDAV

id	100000056155
Address	MAN SUITE 1A CB 173 GAMMA ROAD DELTA SECTION BETA 81
Business Domain	a
City	NEW YORK
Country	US
Date	2014-12-31
Entity Type	INV
Jurisdiction	AMEA
Label	Account
Name	SACHIN J PANDAV
Original ID	ACTRMISMTHH-008
Risk	6
Source	FCDM
State	NY
Status	A
Tax ID	TAXACEXCECOMMAC-008;
label	fcdm_account
liveSearchScore	0

By exploring the options available within the graph customization, you can streamline your analysis and present your reports based on your use case.

6 Create Event API

The Create Event API is a collection of DB program units. This API accepts inputs from two tables to generate alerts that are populated in the event tables.

The syntax to call the Create Event API is as follows:

```
SELECT FUNC_CREATE_EVENT(,) FROM DUAL;
```

For information on Table_1 and Table_2, see Table 1 Details and Table 2 Details.

6.1 Table 1 Details

Table 1 contains information related to the direct focus. This information is aggregated for focus. For example, the account number and the total transaction amount for that account will be stored in Table 1. Here, one record is inserted for each focus (account or customer).

Table 1

Column Name	Sample Values	Description
V_FOCUS_TYPE	ACCOUNT	The main focus of the scenario. The value can be obtained from the kdd_dataset_base table.
V_ENTITY_NAME	ANDERSON	Name of the focused entity, that is, account display name and customer display name.
V_DATA_ORIGIN	AMLBIG	Data origin, that is, MAN, DLY, and so on.
V_STATUS_CD	NEW	Event status code. Default value: New.
D_EVENT_CREATED_DAT E	3/11/2019 15:02	Event creation date.
V_JURISDICTION_CD	AMEA	Jurisdiction.
V_BUSINESS_DOMAIN_C D	a	Business domain.
V_COMMENTS		General comment for the scenario.
V_SCENARIO_CLASS	AML	Class to which the scenario belongs to, that is, AML, TC, and BC.
V_FOCUS_CD	ACMLTERR ORFINFAC- 005	Code for focus, that is, account internal ID in case of account.

V_ENTITY_CD	832	Focus sequence ID, that is, account sequence ID in case of account focus.
MAX_DATA_DUMP_DT	12/10/2015	Data Dump date for which business data is involved.
MAX_NOT_EFT_TRANS	0	Specific for an account. This column can be ignored for other scenarios.
TOT_TRXN_AMT	10500	
TOT_TRXN_CNT	2	
EFFECTIVERISK	9	
ACTIVITYRISK	10	
RISK_CATEGORY	HR	
ACCT_TYPE	Seasoned	
TRUSTED_PRCTG	0	
PASS_THRU_PRCTG	0	
LRF_PRCTG	0	
HR_PRCTG	0	
ACT_TOT_CREDIT_TRXN_ AMT	5400	Threshold specific for an account. It can be different for other scenarios.
TH_TOT_MIN_CREDIT_TR XN_AMT	50	
TH_TOT_MAX_CREDIT_T RXN_AMT	10000000	
ACT_TOT_CREDIT_TRXN CT	2	
TH_TOT_MIN_CREDIT_TR XNCT	1	
TH_TOT_MAX_CREDIT_T RXNCT	20	
ACT_TOT_DEBIT_TRXNC T	2	
TH_TOT_MIN DEBIT_TR XNCT	1	
TH_TOT_MAX_DEBIT_TR XNCT	20	
ACT_TOT_DEBIT_TRXN_A MT	5100	
TH_MIN_DEBIT_TRXN_A MT	20	
REMARK		
SCENARIO_CD	10000	
SCENARIO_MN	RMF Account(sql	Name of scenario notebook.

	)	
N_RUN_ID	84	The Create Event API automatically creates this value.
N_EVENT_CD	366	
V_ENTITY_CD	25291	Transaction sequence ID.
V_FOCUS_CD	ACMLNOA AC-302	Account Internal ID in case of account focus.
N_ENTITY_ID	FOMLNOA AC-301	Transaction Internal ID.
TRXN_DATADUMP_DT	12/10/2015	Transaction Data Dump date.
TRXN_PRDCT_TYPE	WIRE_TRXN	Transaction Product type. The value can be obtained from the kdd_dataset_base table.

6.2 Table 2 Details

Table 2 contains information on all the transactions involved in alert generation.

Table 2

Columns	Sample Values	Description
V_ENTITY_C D	25291	Transaction sequence ID.
V_FOCUS_C D	ACMLNOAAC-302	Account Internal ID in case of account focus.
N_ENTITY_ID	FOMLNOAAC-301	Transaction Internal ID.
TRXN_DATA DUMP_DT	12/10/2015	Transaction data dump date
TRXN_PRDC T_TYPE	WIRE_TRXN	Transaction Product type. The value can be taken from the kdd_dataset_base table.

OFSAA Support

Raise a Service Request (SR) in My Oracle Support (MOS) for queries related to the OFSAA applications.

Send Us Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, indicate the title and part number of the documentation along with the chapter/section/page number (if available) and contact the My Oracle Support.

Before sending us your comments, you might like to ensure that you have the latest version of the document wherein any of your concerns have already been addressed. You can access My Oracle Support site that has all the revised/ or recently released documents.