# **Collect: Data Virtualization Caching - Deeper Dive**

## Lab overview

IBM Cloud Pak for Data (CPD) is a robust, fully integrated platform that will improve business agility and decrease the cost of creating high quality, impactful Analytics.

Data Virtualization (DV), an important component of IBM Cloud Pak for Data, helps integrate data sources across multiple types and locations and turn them into one logical data view. This virtual data lake decreases the amount of time needed to get value out of your data. With data virtualization, you can query data across many source systems without having to copy and replicate the sources into a centralized repository, which saves time and reduces cost. It simplifies your analytics and provides “current state of the business” answers because you’re querying the data at its source with all its respective policies enforced. One can create virtual objects in DV, pointing to the original data sources to be used in queries as if they were local objects. Each query executed against these virtual objects gets efficiently compiled and optimized to work directly with the source data. The query times and performance will be dictated by latency and the workload on the underlying data sources, especially while performing aggregated queries across different data sources.

With Data Virtualization Caching, it is possible to avoid trips to the original source(s) and create a temporary cache of the data. This helps immensely with query performance as the DV optimizer can choose to use the local cached copy instead of querying the remote sources repeatedly.

In this lab, you will create Data Virtualization objects that will be used within a dashboard to perform analysis followed by using the caching capability of DV to experience how query times can be drastically reduced. You will also learn how caches can be refreshed, disabled, enabled and deleted, and the best practices and guidelines to create DV Caches.

In our scenario, the Trade Co. Business Analyst (BA) initially creates a dashboard using data from a DV view, which in turn points to multiple remote data sources underneath. But rendering these dashboards and using them for analysis takes a very long time since the data has to be retrieved every time from those remote data sources. The BA works with the DV Admin, who helps create a cache for the virtualized view to help address the data retrieval performance issues observed while working with the dashboard.

## Personas represented in this lab

The Business Analyst persona along with the Data Engineer and Data Virtualization Administrator personas will perform the exercises in this lab.

|  |  |
| --- | --- |
| **Persona (Role)** | **Capabilities** |
| Business Analyst | Business Analysts deliver value by taking data, using it to answer questions, and communicating the results to help make better business decisions. |
| A close up of a logo  Description generated with high confidence  Data Engineer | Data Engineers build and optimize the systems to allow data scientists and business analysts to perform their work. The Data Engineer ensures that any data is properly received, transformed, stored, and made accessible to other users. |
| A close up of a logo  Description automatically generated  Administrator (DV/CPD) | Administrators set up and maintain the DV module within the CPD environment itself. They are responsible for granting DV access to users and administration tasks such as creating a data cache. |

For convenience in doing this lab, instead of switching between personas, all the required privileges have been provided to the same user. The workbook will refer to the respective personas at different stages to help understand the flow of this task.

## Logging into the CPD web client (if you have not already done so)

1. If you are starting this lab stand-alone (without going through previous labs) do the following:
2. Double-click the desktop icon: Cloud Pak for Data Web Client.

A picture containing object, clock

Description automatically generated

1. The CPD web client GUI displays as shown. Use cpduser and cpdaccess for the *Username* and *Password* and click Sign in.

A screenshot of a cell phone

Description automatically generated

## Reviewing the dashboard: Stock Trading Analysis - Trade Co.

In an attempt to understand stock trading patterns, the business analysis for Trade Co. starts by creating a simple dashboard to find the most popular historically traded stocks. The dashboard shows the number of Shares Sold vs. the number of Daily Trades.

A screen shot of a computer

Description automatically generated

The Business Analyst (BA) works with the Data Engineer (DE) to get virtualized access to the different data sources required for creating the dashboard(s). The DE creates the Data Sources, Virtual Tables and finally a Virtualized View joining all the Virtual Tables and computing the basic aggregations required. The View is then shared with the BA, who can then proceed with creating the dashboard.

However, once the dashboard is initially created, the BA notices delays in rendering it. Since every request from the dashboard has to fetch the data from its original source(s), latency starts to play an important role, slowing down response. The BA works with the Data Virtualization Admin to create a cache for the View, which helps speed up query times significantly and hence rendering of the dashboard(s).

## Virtualizing the remote tables and creating a view

The Data Virtualization process begins by adding data sources to virtualize and is typically done by the Data Engineer.

|  |
| --- |
| **Persona (Role)** |
| A close up of a logo  Description generated with high confidence  Data Engineer (DV/CPD) |

### **Navigate to Data virtualization**

1. Start at the Navigation Menu.

Click Collect 🢧 Data virtualization.

A picture containing sitting, dark, screen, sign

Description automatically generated A picture containing clock

Description automatically generated

### **Create the PostgreSQL database connection**

1. We will need to create a connection to a PostgreSQL table for use in this lab. Select Data Sources – SQL editor and enter the following SQL:

call QPLEXSYS.setRdbcX('PostgreSQL', '03c07569-09f5-42d9-8135-c1d647c59920.btdl8mld0r95fevivv30.databases.appdomain.cloud','32121', 'ibmclouddb', 'CryptoProtocolVersion=TLSv1.2', 'ibm\_cloud\_7e30807a\_5e03\_4bb5\_87f5\_6550a69de7cd','6627c6d6f8746916ec7fa3c0ade3d722297dfcd1e9b91cec7c53515e2efd1843', '1', '0', '', '', 'qpendpoint\_5:6419', ?, ?, ?);

Note: You can also find this SQL to download/copy/paste here: <https://ibm.box.com/v/DV-CreatePostgreSQL>

1. Select Run all to execute the SQL and create the connection in Data Virtualization.

Graphical user interface, application

Description automatically generated

### **Creating virtualized tables**

1. With the data sources successfully created, the next step is to virtualize the tables needed for this exercise.

From the top left go to SQL editor 🢧 Virtualize.

Text, application

Description automatically generated

1. In the search bar, enter the string STOCK and click the search icon (magnifying glass).

A screenshot of a cell phone

Description automatically generated

1. Click the gear, then select Group tables with identical names.  
   A close up of a sign

   Description automatically generated

This uses the DV feature of “Schema Folding” which groups tables with the same names across data sources and shows them as a single “table” selection. (Notice how two tables called CUSTOMER\_TRANSACTIONS from to different databases are treated as one.)

1. Select tables: CUSTOMER\_TRANSACTIONS and STOCK\_SYMBOLS 🢧Click Add to cart 🢧

View cart.

Graphical user interface, application

Description automatically generated Diagram

Description automatically generated

1. In the Review cart and Virtualize tables section, select button My virtualized data 🢧Uncheck box Submit to catalog 🢧 Click Virtualize.

A picture containing clock, meter

Description automatically generated

1. Notice the **two** virtual tables you just created, which are actually comprised of three different tables in three databases in the IBM Cloud. The CUSTOMER\_TRANSACTIONS virtual table is a schema folded table pointing to **two** remote Db2 Warehouse tables and the STOCK\_SYMBOLS virtual table points to **single** table in the PostgreSQL database.

Click View my virtualized data.

A screenshot of a cell phone

Description automatically generated

1. You will now be creating a virtualized view of these virtualized tables using the SQL editor.

Select Menu (My virtualized data) 🢧 SQL editor.

A screenshot of a cell phone

Description automatically generated

1. Copy and paste the SQL below into the SQL editor.

|  |  |
| --- | --- |
| **A close up of a logo  Description generated with high confidence**  Data  Engineer | Note: You can copy and paste this SQL directly from the Unified Desktop by using the File Browser to open this file:  🢧 Home 🢧 Downloads 🢧 DVCaching\_SQL.txt.  Alternately, you can download this file by opening a browser tab and using this link: <https://ibm.box.com/v/Workshop-V9-DVCashing-SQL> |

**DROP VIEW VIEW\_CUST\_TXN\_SYMBOL\_COM;**

**CREATE VIEW VIEW\_CUST\_TXN\_SYMBOL\_COM**

**AS**

**SELECT**

**"SYM"."COMPANY",**

**"SYM"."SYMBOL",**

**"CUSTID",**

**"TOTAL\_QUANTITY",**

**"TXN\_COUNT"**

**FROM**

**"USER1001"."STOCK\_SYMBOLS" "SYM",**

**(SELECT**

**"USER1001"."CUSTOMER\_TRANSACTIONS"."CUSTID" "CUSTID",**

**"USER1001"."CUSTOMER\_TRANSACTIONS"."SYMBOL" "SYMBOL",**

**"USER1001"."CUSTOMER\_TRANSACTIONS"."TRANSACTION\_DATE"**

**"TRANSACTION\_DATE",**

**SUM("USER1001"."CUSTOMER\_TRANSACTIONS"."UNITS\_TRADED") as**

**"TOTAL\_QUANTITY",**

**COUNT(\*) as "TXN\_COUNT"**

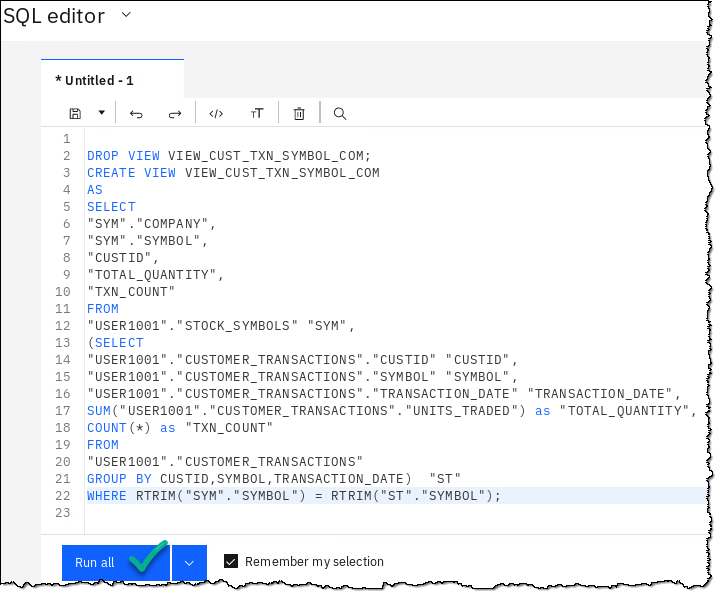
**FROM**

**"USER1001"."CUSTOMER\_TRANSACTIONS"**

**GROUP BY CUSTID,SYMBOL,TRANSACTION\_DATE) "ST"**

**WHERE RTRIM("SYM"."SYMBOL")= RTRIM("ST"."SYMBOL");**

1. Click Run all which should create the virtualized view successfully.



|  |  |
| --- | --- |
| **A close up of a logo  Description generated with high confidence**  Data  Engineer | Note: The first time this script is run, the DROP VIEW statement will fail since the view does not exist. It will execute cleanly on subsequent runs. |

1. Preview the contents of the virtualized view to confirm the data is being correctly retrieved.

Go to Menu (SQL editor) 🢧 My virtualized data.

A screenshot of a cell phone screen with text

Description automatically generated

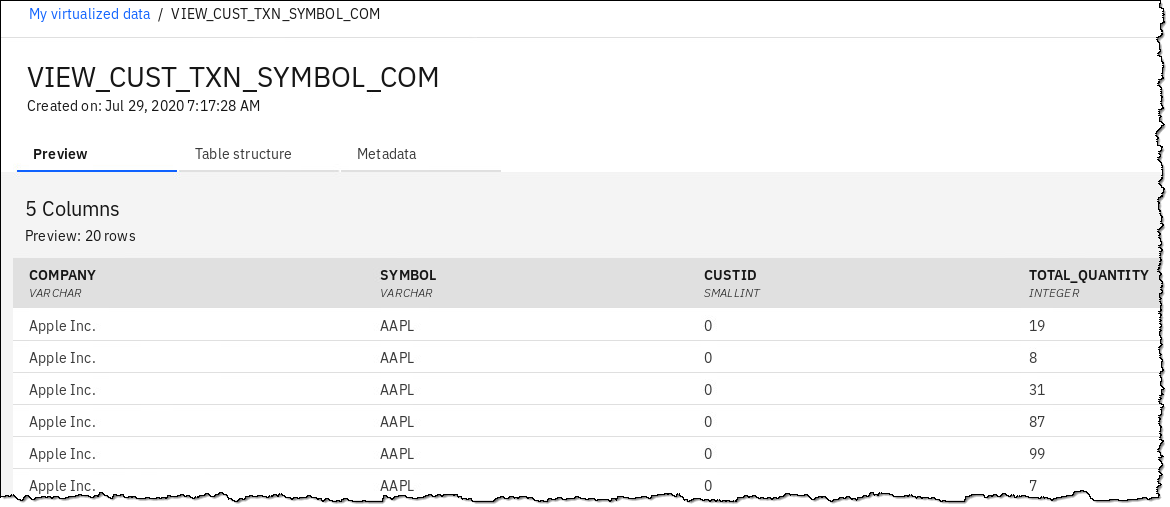
1. On the view you just created, VIEW\_CUST\_TXN\_SYMBOL\_COM, click the ellipses at the end of that line … and then select Preview.

A screenshot of a cell phone

Description automatically generated

|  |  |
| --- | --- |
| **A close up of a logo  Description generated with high confidence**  Data  Engineer | It may take a minute or two for the view to render.  If this step fails, check to make sure that you have copied the SQL code properly into the SQL editor and run it again. Keep in mind that a successful execution of the SQL may not necessarily mean the view was actually created properly. |

1. The Virtualized View preview should render as below.



## Creating a project to work in

The Virtualized View referencing different data sources is now created. The Data Engineer will share the Virtualized View with the Business Analyst (BA) who can then work on creating the dashboard.

|  |
| --- |
| **Persona (Role)** |
| A picture containing clothing  Description automatically generated  Business Analyst |

With the access to the Virtualized View in place, the BA begins to work on the dashboard by creating an Analytics Project.

1. Click Navigation Menu 🢧 Projects 🢧 New project.

A screenshot of a video game

Description automatically generated

1. Choose Analytics project 🢧 Next.

A screenshot of a cell phone

Description automatically generated

1. Click on tile Create an empty project.

A screenshot of a cell phone

Description automatically generated

1. Provide a name of ‘your initials-DV-Project’ and description for the project and click Create.

A screenshot of a cell phone

Description automatically generated

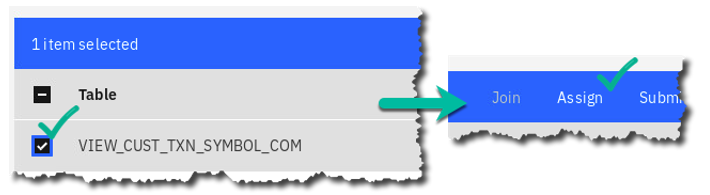
### **Adding the previously created DV view to the new project**

1. Start at the Navigation Menu 🢧 Click Collect 🢧 Data virtualization 🢧 Menu (Data Sources) 🢧 My virtualized data.

A picture containing meter

Description automatically generated

1. Check VIEW\_CUST\_TXN\_SYMBOL\_COM 🢧 click Assign.



1. Choose your\_initials-DV-Project (whatever you have named it) 🢧 Assign.

A screenshot of a cell phone

Description automatically generated

1. The DV View is now successfully assigned to the project.

Click Go to project.

A screenshot of a cell phone

Description automatically generated

1. On the Project page, click the Assets tab.

Review the Data assets section where the newly assigned virtualized view should now show up, along with the Connection it needs in order to render it from the data sources. (Your connection Name will vary from what is shown below.)

A screenshot of a cell phone

Description automatically generated

### **Creating a dashboard within the project to use the data asset:**

1. Click + Add to project and choose Dashboard as the asset type.

A screenshot of a cell phone

Description automatically generated

1. Under *Name* type: Company Stock Analysis 🢧 Create.



1. Choose the Freeform (default) layout 🢧 OK.

A screenshot of a cell phone

Description automatically generated

1. On top left of the dashboard canvas in Selected sources and click + (Add a source).

A screenshot of a cell phone

Description automatically generated

1. Click Data Assets 🢧 USER1001.VIEW\_CUST\_TXN\_SYMBOL\_COM 🢧 Select.

A screenshot of a cell phone

Description automatically generated

1. Click the newly added data source to expand the view to see the columns available to be used in the dashboard.

A screenshot of a cell phone

Description automatically generated

1. Drag and drop the COMPANY column on the canvas.

(It may take a minute or two for this column to render as a widget – please be patient.)

A screenshot of a cell phone

Description automatically generated

1. Make sure to select the COMPANY widget on the canvas.

Then at the top right select the tab Fields.

A screenshot of a cell phone

Description automatically generated

1. On the top left (above Tab 1) change visualization type from list to column by selecting:

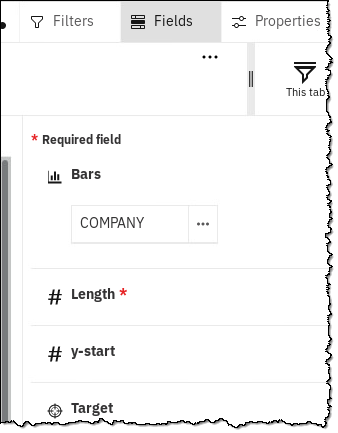
List box 🢧 All visualizations 🢧 Column.

Then close the visualization select box X.

A screenshot of text

Description automatically generated

1. On the right of the screen, you should see this visualization formatting box.



1. Format the visualization by performing these steps, in order:

1. Make sure the COMPANY column is used for Bars (if it is not, drag it there).

2. Drag and drop the TXN**\_**COUNT column or Color.

3. Drag and drop the TOTAL**\_**QUANTITY column for # Length.



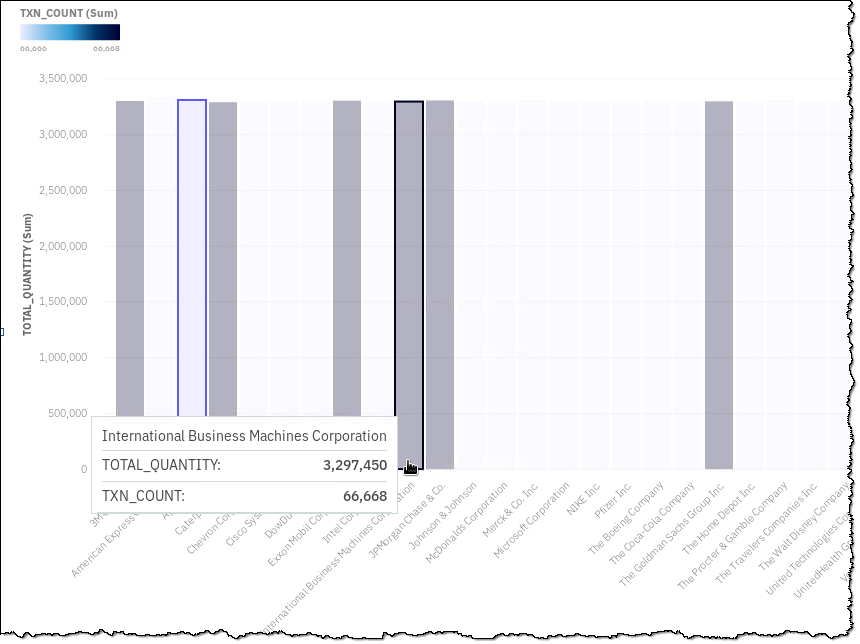
1. Click the Save icon to ensure your work is saved.

A picture containing clock, meter

Description automatically generated

1. Perform some analysis of various companies to see the quantity traded and the transaction counts by hovering on an individual company from within the visualization.

Example: hover over the column for International Business Machines Corporation.



|  |  |
| --- | --- |
| Business Analyst | I noticed a delay while retrieving my data in this dashboard which in the long run will be unsuitable for me to use repeatedly.  I need to reach out to my CPD Administrator and Data Engineer to find a way to make this perform better. |

## Creating the Data Virtualization cache

### **Creating the DV data cache**

|  |
| --- |
| **Persona (Role)** |
| A close up of a logo  Description automatically generated  Administrator (DV/CPD) |

The DV Admin will first analyze the queries executed against DV and identify those generated by the dashboard. Looking at the query will help understand what type of cache needs to be created to improve performance for a better dashboard experience.

1. Start at the Navigation Menu.
2. Click Collect 🢧 Data virtualization 🢧 Menu (Data Sources) 🢧 Cache Management.

A picture containing clock, meter

Description automatically generated

1. You will be presented with your Cache management dashboard.

A screenshot of a cell phone

Description automatically generated

1. Click tab Queries.

A screenshot of a cell phone

Description automatically generated

1. At the far right of the screen select Filter 🢧 Not using cache 🢧 Response time: 1m-10m 🢧 Last 24 hours🢧 Apply.

A screenshot of a cell phone

Description automatically generated

1. The list of queries executed are shown with their average execution times if run more than once. Note the response time which explains the delay in working within the dashboard. (The example below is greater than 1 minute and 10 seconds)

Select the down arrow by first Query ID to see the actual SQL being run.

NOTE: If the query you chose does not contain ‘SELECT VIEW\_CUST\_TXN\_SYMBOL\_COM at the start of the query, use the back-arrow key to return to the cache, scroll down and view another query.

A screenshot of a social media post

Description automatically generated

Click View details.



1. Record (or screen shot) the Query ID, which will be used later in this lab. (Often, recording only the last 6 digits is all that is necessary.)

A screenshot of a cell phone

Description automatically generated

One could choose to create a cache from this particular query, but since the dashboards will be using all columns from the View created anyway, the DV Admin and the BA decides to create a cache for the entire virtualized view instead.

1. Go back the main Cache Management page and click Data caches.

A close up of a logo

Description automatically generated A screenshot of a cell phone

Description automatically generated

1. At the right of the page, click Add new cache.

A close up of a logo

Description automatically generated

1. This opens a SQL Editor to enter the query from which the cache will be created.

Type in the query to select all columns from the virtualized view:

**SELECT \* FROM "USER1001"."VIEW\_CUST\_TXN\_SYMBOL\_COM"**

Click Next.

A screenshot of a cell phone

Description automatically generated

1. In the Refresh rate choose the default (None, manual only) and click Next.

A screenshot of a cell phone

Description automatically generated

1. Enter your initials at the end of the Cache name (yours may vary) then confirm the details on the final page.

Click Create.

A screenshot of a cell phone

Description automatically generated A close up of a sign

Description automatically generated

The cache creation process may take some time and the main Cache Management page will reflect the work in progress.

A screenshot of a cell phone

Description automatically generated

1. Once the cache creation is complete, the newly created cache shows up under the Activedata caches along with other details and its size.

A screenshot of a cell phone

Description automatically generated

|  |  |
| --- | --- |
| Admin | KNOWN ISSUE – There exists a defect/issue wherein queries executed before the cache is created and activated fails to use the cache. This is because the query plan is already cached in a relational database such as Db2, so the work around is to clear the Db2 package cache and collect table statistics on the virtual tables created above. You will do that next. |

1. Click Menu (Cache management) 🢧 SQL Editor.

A screenshot of a cell phone

Description automatically generated

1. Remove the view SQL and replace it with the following:

**FLUSH PACKAGE CACHE DYNAMIC**

Click Run all.

A screenshot of a cell phone

Description automatically generated

In a production setting, flushing the entire package may not always be a good idea as it could affect query execution for all users of the database. (We did it here to make this lab simpler.)

Instead, one could opt to selectively flush the package corresponding to particular queries. For more details on that see: <http://ibm.biz/FLUSH-PACKAGE>.

1. Next, check if the table statistics have been collected. Replace the previous SQL with:

**SELECT TABNAME, STATS\_TIME, CARD , TYPE**

**FROM SYSCAT.TABLES**

**WHERE TABSCHEMA='USER1001'**

**AND**

**TABNAME**

**IN ('STOCK\_SYMBOLS','CUSTOMER\_TRANSACTIONS');**

|  |  |
| --- | --- |
| **A close up of a logo  Description generated with high confidence**  Data  Engineer | Note: You can copy and paste this SQL directly from the Unified Desktop by using the File Browser to open this file:  A picture containing monitor, computer, sitting, clock  Description automatically generated 🢧 Home 🢧 Downloads 🢧 DV\_SelectTabname.txt.  Alternately, you can download this file by opening a browser tab and using this link: <https://ibm.box.com/v/DV-SelectTabname> |

1. Click Run all.

A screenshot of a cell phone

Description automatically generated

If the cardinality for the virtual tables created before shows -1, this indicates that table statistics have not been collected. These statistics will be collected now.

A screenshot of a cell phone

Description automatically generated

|  |  |
| --- | --- |
| **A close up of a logo  Description generated with high confidence**  Data  Engineer | Note: The SYSPROC.NNSTATS is a procedure to collect statistics for remote tables: <http://ibm.biz/SYSPROC-NNSTATS>. |

1. If your statistics are not current, then do this step. If they are current, you can skip this step.

**CALL SYSPROC.NNSTAT(NULL, 'USER1001',**

**'CUSTOMER\_TRANSACTIONS','', '',2,'/tmp/collstats1.log',?,1);**

**CALL SYSPROC.NNSTAT(NULL, 'USER1001',**

**'STOCK\_SYMBOLS','', '',2,'/tmp/collstats1.log',?,1);**

|  |  |
| --- | --- |
| **A close up of a logo  Description generated with high confidence**  Data  Engineer | Note: You can copy and paste this SQL directly from the Unified Desktop by using the File Browser to open this file:  A picture containing monitor, computer, sitting, clock  Description automatically generated 🢧 Home 🢧 Downloads 🢧 DV\_CallSysProc.txt.  Alternately, you can download this file by opening a browser tab and using this link: <https://ibm.box.com/v/DV-CallSysProc> . |

1. Click Run all.

A screenshot of a cell phone

Description automatically generated

1. To confirm if the table statistics have now been collected, replace the previous SQL with the following:

**SELECT TABNAME, STATS\_TIME, CARD , TYPE**

**FROM SYSCAT.TABLES**

**WHERE TABSCHEMA='USER1001'**

**AND**

**TABNAME**

**IN ('STOCK\_SYMBOLS','CUSTOMER\_TRANSACTIONS');**

1. Click Run all.

A screenshot of a social media post

Description automatically generated

The actual rows counts for the virtual tables are shown as expected instead of -1.

With the package cache flushed and the table level statistics collected, all queries henceforth referencing the virtualized view should start using the cached copy instead of accessing the underlying data sources directly.

The DV Admin informs the BA about the cache creation and asks the BA to re-run the dashboard.

|  |
| --- |
| **Persona (Role)** |
| A picture containing clothing  Description automatically generated  Business Analyst |

1. Return to the dashboard created previously.

Navigation Menu 🢧 Projects🢧 your\_initials-DV-Project 🢧 Assets 🢧 Dashboards 🢧 Company Stock Analysis

1. The dashboard should now render faster with much better response times while analyzing the stock data. The queries executed by the dashboard against the virtualized view now use the DV cache.
2. Return to Data Virtualization, to Menu 🢧 Cache Management 🢧 Queries to get some insights into the queries you just ran.

*Note:* Because there is sometimes a slight delay in getting all the information about the latest queries and cache info via the UI, you can search for the query by Query ID (that you hopefully noted in the previous step.)

Click on the Search (magnifying glass) icon.

A close up of a logo

Description automatically generated

Enter the Query ID to list the query.

A screenshot of a cell phone

Description automatically generated

1. Expand the entry to view the query details.

A screenshot of a cell phone

Description automatically generated

1. Click on View details to show the number of executions and execution times.

A screenshot of a cell phone

Description automatically generated

Notice in the above example (yours may vary somewhat) the query took under 1 second to run with the cache, while without cache it was 2 minutes and 18 seconds.

1. Confirm that this is the same query that was executed in the previous steps, by confirming the Query ID and by clicking on SQL.

A screenshot of a social media post

Description automatically generated

The query execution time has reduced **significantly** with the cache now being utilized instead of fetching the data from the underlying sources!

The cache hits should also start showing up under Cache Management 🢧 Data caches.

A screenshot of a cell phone

Description automatically generated

## Reviewing the Cache Management UI

The Cache Management UI helps the DV Admin to create, monitor and manage caches. It provides information about the active/inactive caches like size, hit counts etc. along with options to work with them.

A screenshot of a cell phone

Description automatically generated

The UI also provides details about the queries executed against DV. This is particularly useful for the DV Admin in identifying slow running queries and potentially creating caches to make them run faster.

Play with some of the features of the Query section using Filters.

A screenshot of a cell phone

Description automatically generated

## Refreshing the Data Virtualization cache

As data changes in the underlying data sources, it is important to refresh the cache and reflect the freshly updated data. The cache creation process provides an option to set the cache refresh rate.

A screenshot of a cell phone

Description automatically generated

One can choose from the following options in the drop down:

1. *None, manual only:* After the initial refresh, all subsequent refresh has to be manually performed by the DV Admin.
2. *Hourly:* The cache gets refreshed hourly on the set minute of the hour.
3. *Daily:* The cache gets refreshed at a specified time of day.
4. *Weekly:* The cache gets refreshed at a specified time on specific day(s) in a week.
5. *Monthly:* The cache gets refreshed at the specified time on specific day(s) and week(s) in a month.

To avoid delays and unexpected results, the best time to refresh the cache should be chosen based on the lowest DV usage/low query traffic.

## Deactivating/Activating/Deleting the Data Virtualization cache

The Cache Management UI provides some options to help manage caches. After a cache is created, it gets refreshed/populated by data and remains in the active state by default, to be considered by the Query Optimizer while generating the query plans.

A screenshot of a cell phone

Description automatically generated

An active cache can be deactivated so the Query Optimizer will not consider it while generating the query plans. This could be useful if there are issues related to the cached data and needs investigation.

A cache can be deleted, which moves it into the inactive state and eventually cleaned up. The list of deleted caches can be listed using the Deleted caches option. It shows the original cache details and if needed, you could re-create them.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

## Caching guidelines

The DV cache is a very useful feature for improving DV query performance. However, there are some important points to remember:

1. The DV cache can only be created by the DV Admins.
2. A cached object has the same permissions as the underlying DV object.
3. Use queries that perform some sort of aggregation from the underlying data to create the cache. Performing aggregations help create summaries, and thus limit data cached.
4. When possible avoid duplicating data that exists in its original form in the underlying data sources.
5. Identify the most commonly used query or parts of queries and consider creating the cache from them. That way multiple users and queries can benefit from those caches.
6. Limit the size of the data to be cached. Note that the cache uses the storage allocated to the DV instance, hence it is common to all DV users. Large caches will limit the availability of space for other users/scenarios.
7. Time taken to refresh a cache will depend on the amount of data, latency and availability of the underlying data sources.
8. A cache will not be used for a query if there is no match for the statement or column(s) and the virtual object being queried.
9. The DV Cache Management UI takes couple of minutes to refresh the queries executed, cache hit counts and other relevant information.

## Lab conclusion

Data Virtualization (DV), as part of the Collect phase, facilitates accessing data from various data sources and running queries across them. Since data movement is limited, all of the access rules and policies for creating copies remain preserved. In situations where query response times are paramount, DV provides the caching facility, which was covered in this lab.

**\*\* End of Lab 11 - Collect: Data Virtualization Caching – Deeper Dive**

Lab by Rajesh Kartha, Edited by Burt Vialpando and Kent Rubin