# **Collect: Virtualizing & Caching from z/OS – Deeper Dive**

## Lab overview

Data Virtualization (DV) is an important component of IBM Cloud Pak for Data (CPD) to help integrate data sources across multiple types and locations and turn them into one logical data view. As shown in the previous DV Caching Deep Dive lab, virtualizing tables from databases hosted in the cloud and then leveraging caching helps to drastically improve query performance.

In addition to the cloud, almost all enterprises will have large amounts of data stored in on-premises data sources. Data hosted on z/OS systems can be a major portion of that.

While there exist multiple ways to access data residing in a z/OS system, using IBM Data Virtualization Manager (DVM) is one of the most popular ones. Cloud Pak for Data Virtualization (CPD-DV) has the ability to virtualize and ingest any mainframe data that is available, leveraging the Data Virtualization Manager for z/OS.

## IBM DVM

IBM Data Virtualization Manager (DVM) for z/OS® provides virtual, integrated views of data residing on IBM Z®. It enables users and applications read/write access to IBM Z data in place, without having to move, replicate or transform the data.

IBM DVM enables data structures that were designed independently to be used together. Traditional data movement approaches can negatively impact the opportunity to benefit from data where and when it is needed. By unlocking IBM Z data using popular, industry-standard APIs, DVM for z/OS can save you time and money.

DVM for z/OS provides access to IBM Z data sources such as VSAM (Virtual Sequential Access Method), IBM Db2 for z/OS, Adaptable Database System (ADABAS), Integrated Database Management System (IDMS), IBM Information Management System (IMS), IBM System Management Facilities (SMF) and non-IBM Z data sources, all without the need for mainframe skills.

You can also simplify the development of AI applications directly from IBM Z data with CPD.

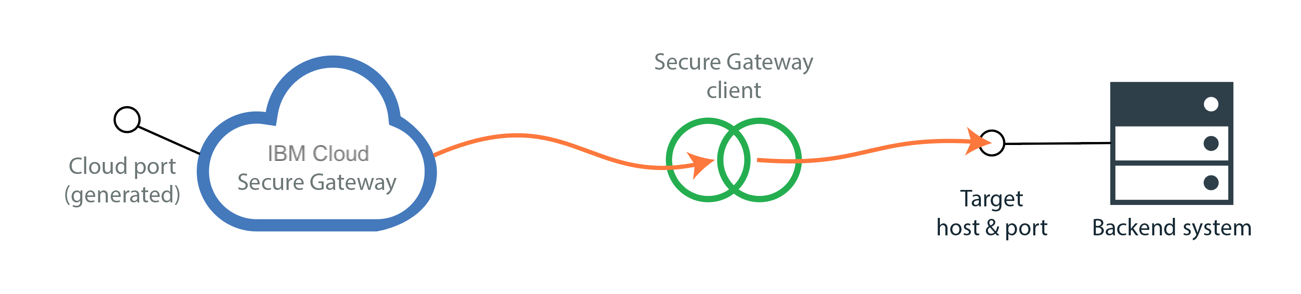
For more information on DVM for z/OS, check out this link: <http://ibm.biz/DV-Manager-ZOS>.

Also check out the IBM Demos page: <http://ibm.biz/IBM-Demos>

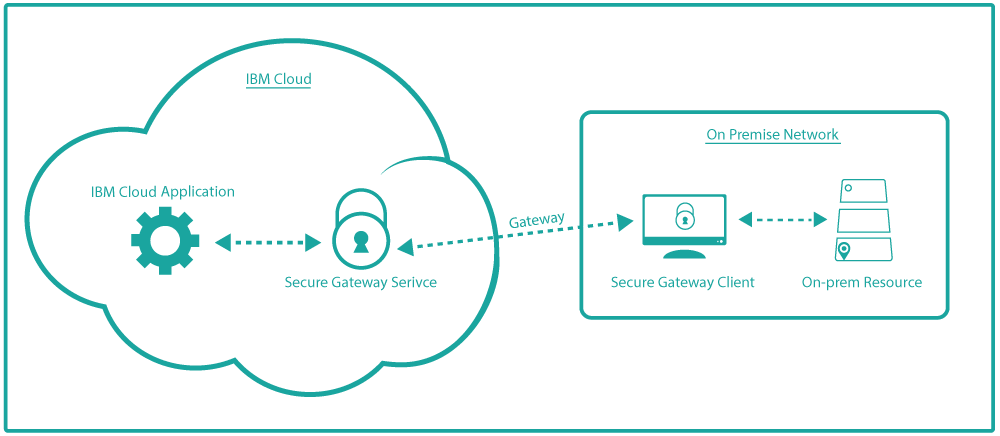
### **Secure Gateway**

Accessing on-prem data from an external system can be challenging and there are different ways to achieve that. A simple and cost-effective way in the IBM Cloud is to use the Secure Gateway service.

IBM Secure Gateway for IBM Cloud maintains a single persistent encrypted connection between the Secure Gateway client in the on-prem network and the Secure Gateway server in the IBM Cloud. Data can be securely transmitted bidirectionally between on-prem and external resources.



The IBM Secure Gateway (SG) consists of a Gateway service, an on-prem SG Client and an on-prem Destination.



External applications can securely connect to the Gateway service, which in turn connects to the on-prem destination via the configured SG client. The external applications are unaware of the on-prem destination details. Theyonly knows the IBM Cloud Secure Gateway service endpoint and the corresponding credentials to be used. More information about IBM Cloud Secure Gateway service is available in its documentation: <http://ibm.biz/Getting-Started> and <http://ibm.biz/Client-Install>.

In this lab, you will create a DV virtualized table that spans three different data sources: one z/OS on-prem data source via IBM DVM, and two Db2 Warehouse tables hosted in the IBM Cloud. This allows querying across all of them via a single DV query and then uses the caching capability of DV to improve query times.

In our scenario, the Trade Co. Business Analyst (BA) builds on the dashboard created in the previous DV Caching Deeper Dive lab using data from a DV view, which in turn points to two remote IBM Cloud Db2 Warehouse data sources underneath. However, there is now a need for more historical analysis with the addition of a third z/OS data source that hosts transaction data from previous years. The BA works with the Data Engineer and DV Admin, who can create the necessary virtual objects (tables and views) and a data cache for the final virtualized view to help address the data retrieval performance issues.

### **Setup (informational only)**

Information in this section is only for information purposes. All the setup steps have already been performed on your workshop cluster to facilitate this lab completion.

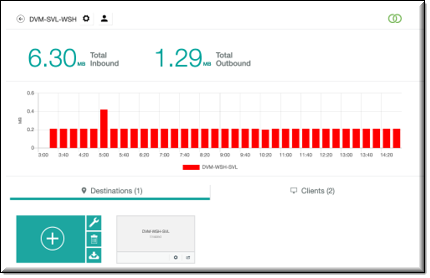
z/OS Data:

In this lab example, a transactional process on z/OS has stored records of stock market trades into a VSAM data set. VSAM data sets arrange data in fixed format records by an index key, by a relative byte address, or by a relative record number. On the mainframe, VSAM data sets are cataloged and used frequently for easy retrieval by z/OS application and processes. DVM for z/OS has the ability to create a customized virtual table or virtual view definition over this VSAM data set.

CPD-DV, using a DVM for z/OS connection, can query the virtual table defined over the VSAM data using ANSI-standard SQL, and virtualize this stock trade data with other DV data sources.

Access:

The VSAM data and the DVM for z/OS application reside on a mainframe server (z/OS) at the Washington Systems Center on IBM premises. This system is in a secured network zone and access from the CPD system is provided through a Secure Gateway client, running inside the IBM network configured to an IBM Cloud Secure Gateway service.

*Connection:*

IBM DVM for z/OS comes with its own set of JDBC drivers and CPD-DV uses these to connect and virtualize tables from the z/OS system. For this lab, these JDBC drivers have already been loaded into the Cloud Pak for Data instance and are ready to create a connection. We will show you how we did it here. ***NOTE: You do not need to complete these steps for this workshop.***

To upload the DVM JDBC driver, from the navigation menu go to Connections.

A screenshot of a cell phone

Description automatically generated

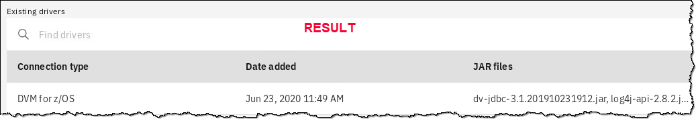
Click on Upload driver.



Provide a name and upload the JDBC jars along with the names of the Driver class name and the JDBC URL prefix. Click on Upload.

A screenshot of a cell phone

Description automatically generated



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

A screen shot of a computer

Description automatically generated

This lab follows the same narrative as that of the DV Caching Deeper Dive lab. 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 the visualizations. 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 DV Admin to create a cache for the View, which helps speed up query times significantly and hence rendering of the dashboard(s).

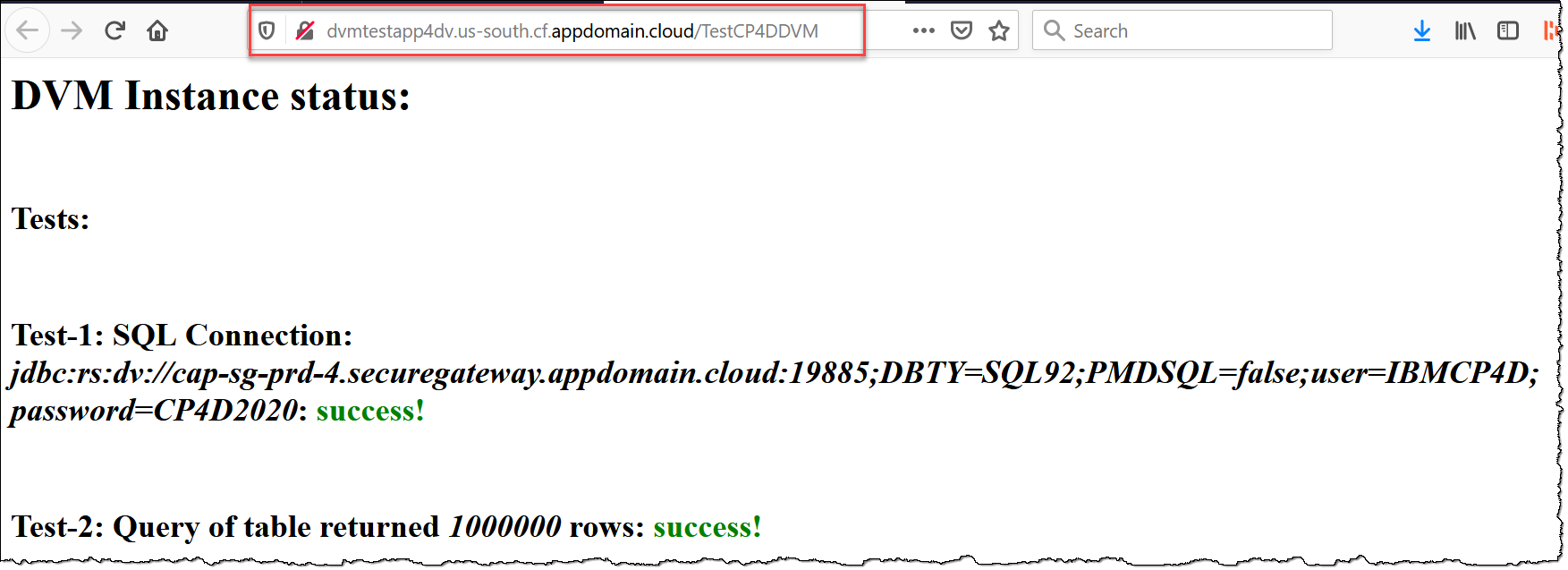
## Adding the DV connection to DVM for z/OS

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

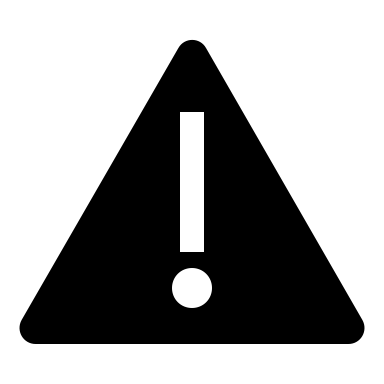
|  |
| --- |
| **Persona (Role)** |
| A close up of a logo  Description generated with high confidence  Data Engineer (DV/CPD) |

1. Before proceeding with the creation of the data connection, ensure that the data source is accessible.

Click on this embedded [link](http://dvmtestapp4dv.us-south.cf.appdomain.cloud/TestCP4DDVM) to confirm, or type: <https://ibm.biz/cpd-workshop-dvm>.



The above link (http://dvmtestapp4dv.us-south.cf.appdomain.cloud/TestCP4DDVM) is a verification test running in the IBM Cloud to confirm if the z/OS and DVM systems are accessible and hence able to be queried.

 If the above link times out or throws an error like this:

A screenshot of a cell phone

Description automatically generated

that means that the z/OS and DVM systems are not accessible for some reason.

Please contact Rajesh Kartha. Email [kartha@us.ibm.com](mailto:kartha@us.ibm.com) Slack: @Rajesh Kartha.

### **Navigating to Data virtualization**

1. Start at the Navigation Menu.

Click Collect a Data virtualization.

A close up of a sign

Description automatically generated A picture containing clock

Description automatically generated

### **Creating a new data connection to DVM for z/OS**

1. The JDBC drivers for connecting to DVM for z/OS have already been loaded and the system is ready to create a connection.

Click Add new data source.



1. Enter the details from the table below and select DVM for z/OS from the drop down.

Note: You can get the Connection information by downloading this file: <http://ibm.biz/DV-Z-OS>.

(If you are using the Unified Desktop, simply open a browser and type in the URL.)

|  |  |
| --- | --- |
| Connection Name | POT-DVM |
| Description | DVM Connection |
| Connection type | DVM for z/OS |
| JDBC URL | jdbc:rs:dv://cap-sg-prd-4.securegateway.appdomain.cloud:19885;DBTY=SQL92;PMDSQL=false |
| Username | IBMCP4D |
| Password | CP4D2020 |

1. Click on Test connection to test it.

A screenshot of a cell phone

Description automatically generated

1. Once the Test connection succeeds, click on Create to create the connection.

A screenshot of a cell phone

Description automatically generated

The connection is successfully created.

A screenshot of a cell phone

Description automatically generated

## Remove existing virtual tables and views (if they exist)

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 a Data virtualization.

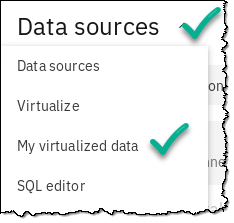
A close up of a sign

Description automatically generated A picture containing clock

Description automatically generated

### **Delete virtual views and tables from the previous DV Caching lab**

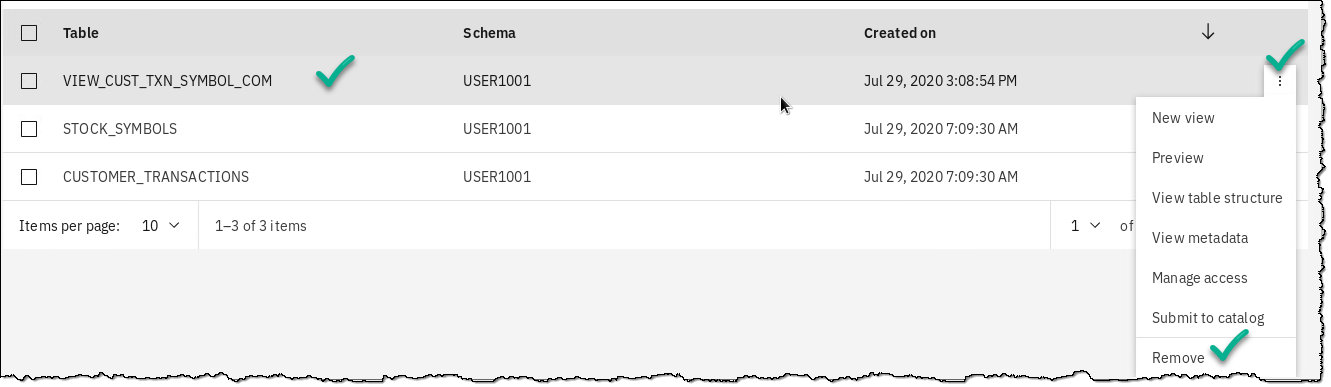
1. From the drop-down menu (Data Sources) select My virtualized data.



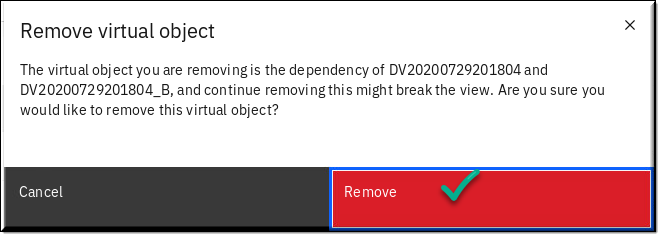
1. Find the virtualized table VIEW\_CUST\_TXN\_SYMBOL\_COM.

(Note: If you did not do the Data Virtualization Deeper Dive lab before this one, then these assets will not exist and you can skip this and go to the next section.)

Click the ellipses on that table (at the end of the line) then Remove.



1. Click Remove to confirm.



1. In the same way, remove virtual tables: STOCK\_SYMBOLS and CUSTOMER\_TRANSACTIONS.

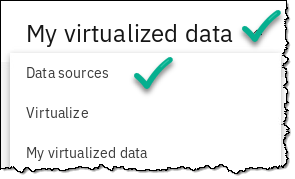


## Virtualizing the remote tables and creating a view

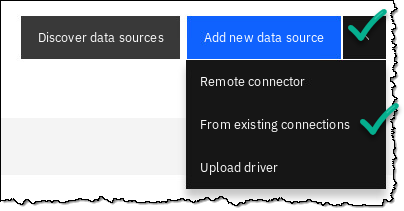
### **Add the DVM data source**

1. With the connection added in the section Adding the DV connection to DVM for z/OS, we will now add that as a data source to our Data Virtualization environment.

From the drop-down Menu (My virtualized data) select Data sources.



1. On the top far right, select Add new data source a From existing connections



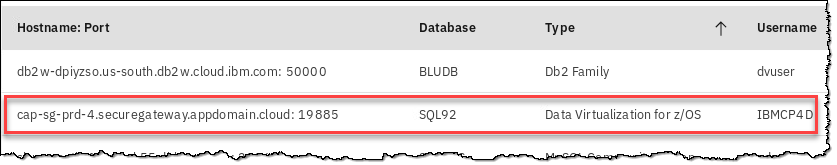
1. Select the button for POT-DVM aNext.

A close up of a logo

Description automatically generated

|  |  |
| --- | --- |
| **A close up of a logo  Description generated with high confidence**  Data  Engineer | If you receive an error message that says the data source is already created, that is OK.  Just move on to the next step. The data source should be available to review. Click the refresh button on your browser to make sure you can see it. |

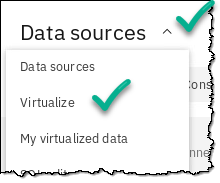
1. You should now see this data source in your DV Data Sources screen.



### **Create virtualized tables**

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

From the drop-down menu (My Data sources) select a Virtualize.



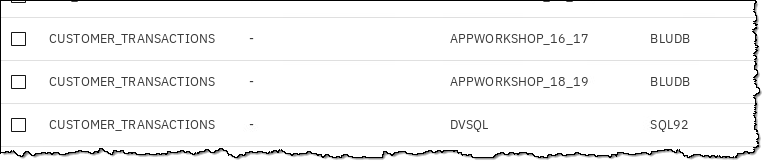
1. In the search bar, enter the string stock.

A screenshot of a cell phone

Description automatically generated

1. Notice there are three tables available called CUSTOMER\_TRANSACTIONS.

The first two tables are from two Db2 Warehouse on Cloud databases called BLUDB, and the third table is from our z/OS database called SQL92 that is available through the DVM connection you created earlier.



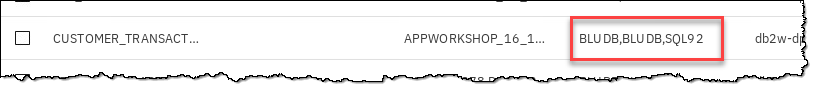
1. Click the gear icon, then select Group tables with identical names.

Graphical user interface, application

Description automatically generated

1. This “Table Grouping” (AKA “Schema Folding”) leverages a powerful DV feature that groups tables with the same names across data sources and presents them as a single entity. For you RDBMS people, think of it as a “union all” join across databases of different types (Db2, SQL Server, Oracle, Db2 z/OS, etc.) located anywhere in the world, all with one click of a button!

Example: Notice how three tables called CUSTOMER\_TRANSACTIONS from three different databases (two called BLUDB, one called SQL92) are treated as one table.



1. Select tables CUSTOMER\_TRANSACTIONS and STOCK\_SYMBOLS.

Click Add to cart.

A screenshot of a cell phone

Description automatically generated

1. Click View cart.

A close up of a logo

Description automatically generated

1. In the section Review cart and Virtualize tables, notice the Grouped tables column which shows the number 3 to indicate the three CUSTOMER\_TRANSACTIONS tables residing in three different databases to be schema folded as a single virtual table.

Select the My virtualized data button.

Uncheck the Submit to catalog box.

Click Virtualize.

A screenshot of a social media post

Description automatically generated

1. Click View my virtualized data to view the tables created.

A screenshot of a cell phone

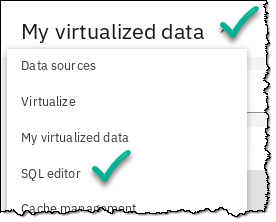
Description automatically generated

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) a SQL editor.



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: <http://ibm.biz/DV-Caching-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"**

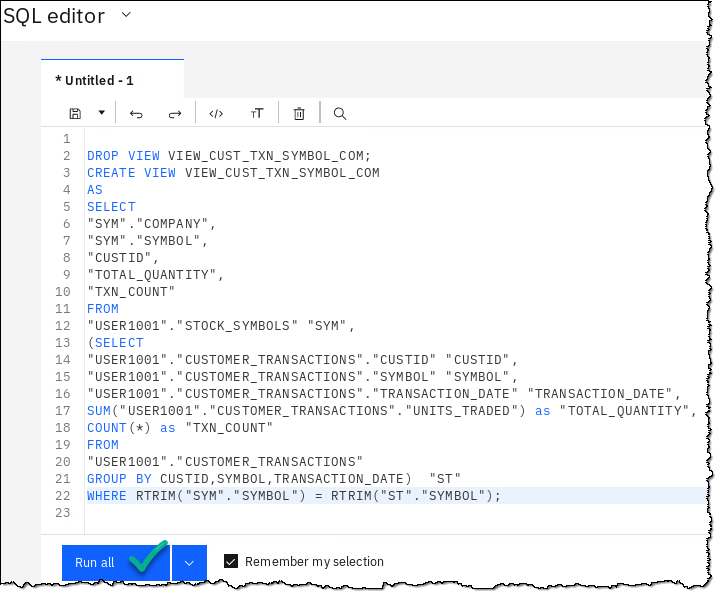
**FROM**

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

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

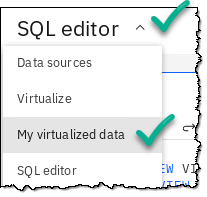
**WHERE RTRIM("SYM"."SYMBOL")= RTRIM("ST"."SYMBOL");**

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. Select menu (SQL editor) a My virtualized data.



1. The virtualized view should now be here: VIEW\_CUST\_TXN\_SYMBOL\_COM.

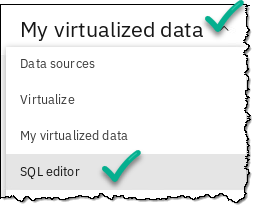
A screenshot of a cell phone

Description automatically generated

### **Check performance**

1. With the VIEW in place, it would be worthwhile to run a simple aggregated SQL query in the SQL editor to see its performance.

Select menu (My virtualized data) a SQL editor.



1. Replace the existing SQL with the following by just typing it in, copy from /Downloads/DV\_CheckPerformance.txt file or download from <http://ibm.biz/DV-Check-Performance>.

**SELECT COMPANY, COUNT(\*) as COUNT**

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

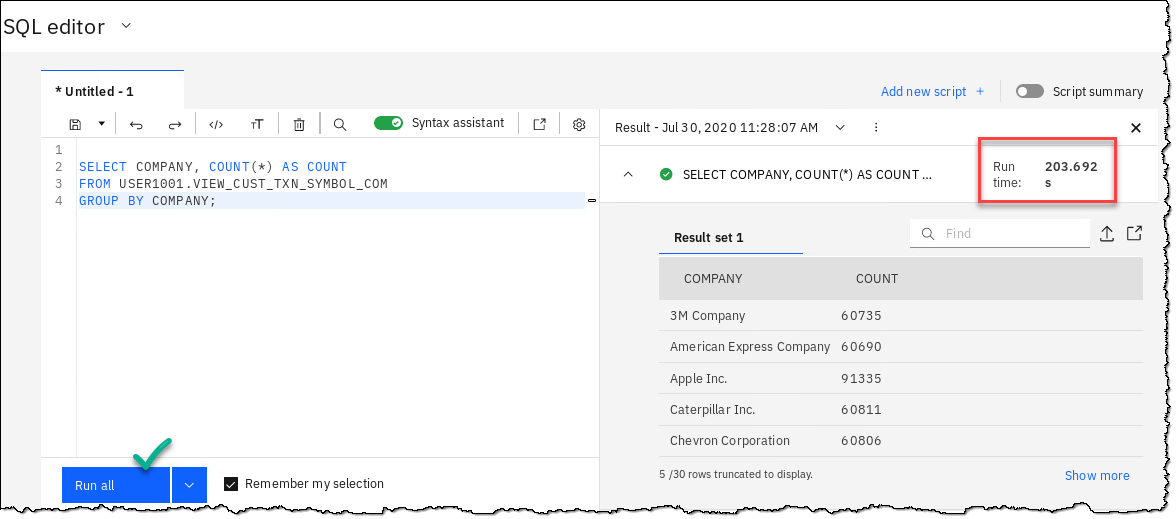
**GROUP BY COMPANY;**

(This query performs a simple COUNT across the tables based on the companies traded.)

1. Click Run all.

The aggregated query is run over the View which in turns gets executed across all three underlying tables in the different data sources via the schema folded virtual table created earlier.

Because we do not have a cache and it is going against all the data in every table, this will take a few minutes to complete – please be patient.



1. Notice the time taken for running the query.

The time taken for the simple query execution is unacceptable. The Data Engineer reaches out to the DV Admin to create a cache

|  |  |
| --- | --- |
| **A close up of a logo  Description generated with high confidence**  Data  Engineer | Based on the time taken to execute the simple aggregated query over the View, makes it a perfect candidate to create a DV data cache from and that will be the next step.  If this step fails, check to make sure that you have copied the SQL code properly in the SQL editor and run it again. Keep in mind that a successful run of the SQL may not mean it was actually created properly. |

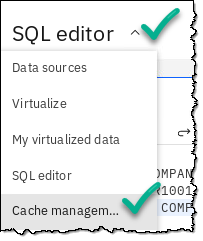
## Creating the DV data cache

Given the performance issues observed during the SQL execution, creating a DV data cache from the View is a prudent approach.

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

The DV Admin can 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. Click Menu (SQL editor) a Cache Management.



1. You should be presented with your Cache Management Dashboard.

A screenshot of a cell phone

Description automatically generated

1. Click on Data caches.

A screenshot of a cell phone

Description automatically generated

1. Towards the bottom right of the screen, find and click Add new cache +.

A picture containing holding, monitor, computer

Description automatically generated

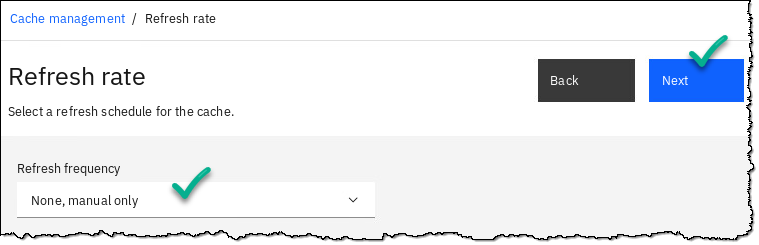
1. This opens a SQL Editor to type in the query from which the cache will be created. Type in the query to select all columns from the virtualized view:

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

Click Next.



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



1. Confirm the details on the final page and click Create.

(Note: Your cache name will vary from what is shown below.)

A screenshot of a cell phone

Description automatically generated

1. 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 Active data caches along with other details and its size.

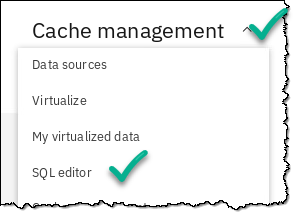
A screenshot of a cell phone

Description automatically generated

### **Flush the package cache and updating the statistics**

|  |  |
| --- | --- |
| Admin | KNOWN ISSUE – There exists a defect/issue wherein a query 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. |

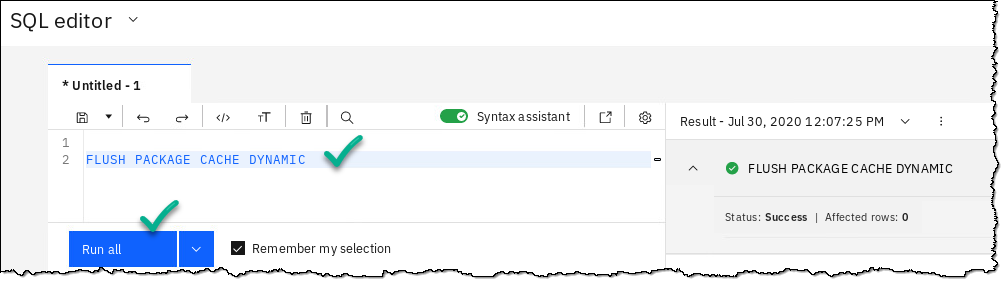
1. Click Menu (Cache management) 🢧 SQL Editor.



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

**FLUSH PACKAGE CACHE DYNAMIC**

1. Click Run all.



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 the following: (You can also download this SQL from <http://ibm.biz/DV-Select-Tab>. ).

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

**FROM SYSCAT.TABLES**

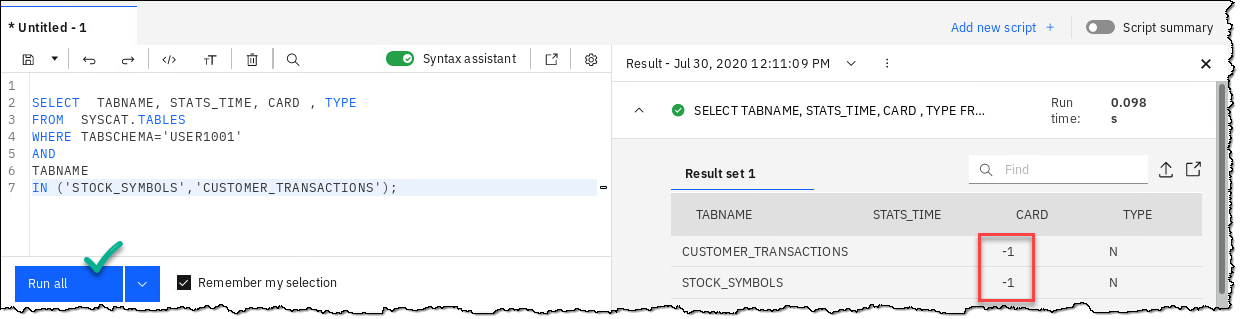
**WHERE TABSCHEMA='USER1001'**

**AND**

**TABNAME**

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

1. Click Run all.



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.

Note: the SYSPROC.NNSTATS is a procedure to collect statistics for remote tables: <http://ibm.biz/SYSPROC-NNSTATS>.

1. If your card shows -1, then perform this step. If not, then you can skip this step.

Replace the previous SQL with the following:

(Or use SQL found here: <http://ibm.biz/DV-Call-SYS>)

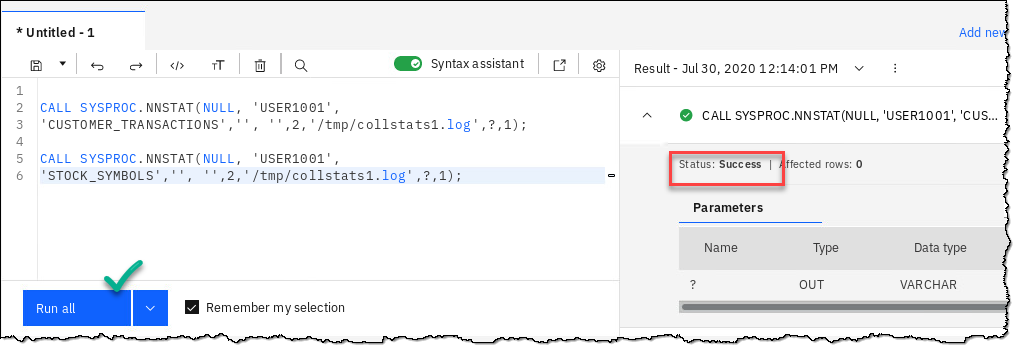
**CALL SYSPROC.NNSTAT(NULL, 'USER1001',**

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

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

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

1. Click Run all.



1. To confirm if the table statistics have now been collected, replace the previous SQL with the following (use the SQL from before):

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

**FROM SYSCAT.TABLES**

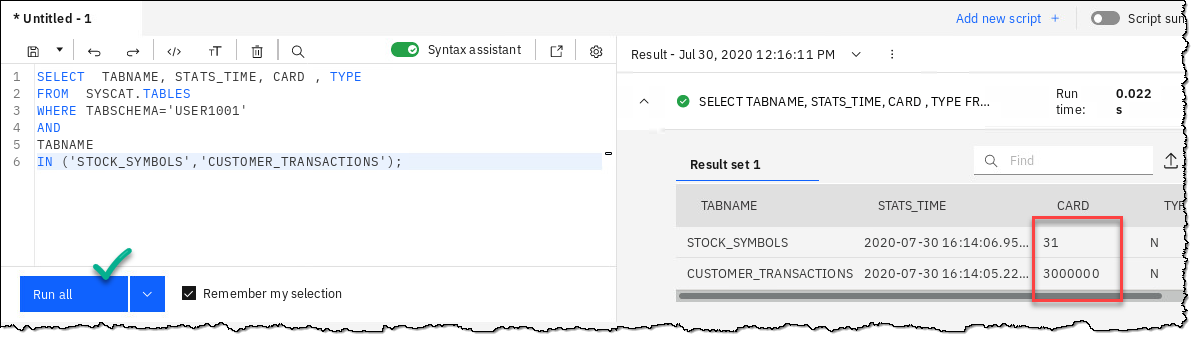
**WHERE TABSCHEMA='USER1001'**

**AND**

**TABNAME**

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

1. Click Run all.



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

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

Re-run the previously execute aggregated query on the View:

Remove the previous SQL and type in this again:

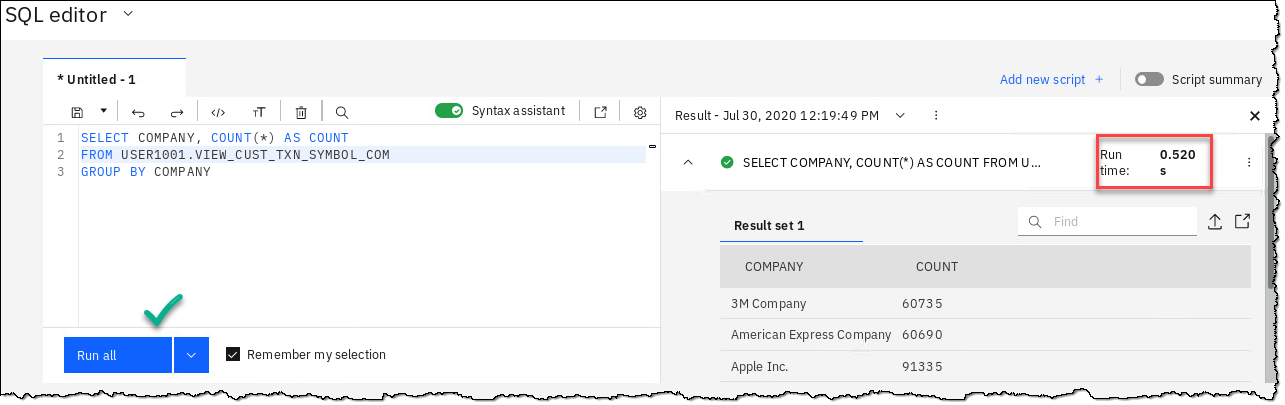
**SELECT COMPANY, COUNT(\*) as COUNT**

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

**GROUP BY COMPANY;**

This query performs a simple COUNT across the tables based on the companies traded.

1. Click Run all.



The Data Engineer the informs the BA of the cache creation and shares the View and other relevant details with the BA to continue the dashboard work.

## Lab conclusion

Data Virtualization (DV), as part of the Collect phase, facilitates accessing data from various data sources such as DVM for z/OS and performs queries across them. Since data movement is limited, all the access rules and the 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 12 –** Collect: Virtualizing & Caching from z/OS – Deeper Dive

Lab by Dave Trotter and Rajesh Kartha, Edited by Burt Vialpando and Kent Rubin - IBM