

A Tutorial on Data Virtualization within IBM Cloud Private for Data

Sanjit Chakraborty

Table of Contents

INTRODUCTION.....	3
1. ACCESS CREDENTIALS	4
1.1. ACCESS CREDENTIAL FOR DB2 DATABASE	4
1.2. ACCESS CREDENTIAL FOR INFORMIX DATABASE	4
1.3. SETTING UP THE DATABASES AND SAMPLE TABLES	4
1.4. SIGN IN TO ICP FOR DATA WEB CONSOLE AS ADMINISTRATOR.....	5
2. CREATE ANALYTIC PROJECT	6
3. DATA VIRTUALIZATION	7
3.3. ADDING A NEW DATA SOURCE FOR INFORMIX.....	8
3.4. CREATING VIRTUALIZED TABLE	9
3.5. CREATING JOINED VIRTUAL TABLE.....	10
3.6. PUBLISH VIRTUALIZED TABLE	11
3.7. ACCESS INFORMATION FOR VIRTUAL TABLE	12
4. VIRTUAL TABLE FROM IDE	13
4.1. START IDE	13
4.2. LAUNCH RSTUDIO	13
4.3. LOAD R SCRIPT	13
4.4. RUN THE SCRIPT	14

Introduction

In this tutorial, you will learn how to use Data Virtualization (DV) add-on component within the IBM® Cloud Private for Data (ICP for Data) to integrates data sources across multiple types and locations and turns it into one logical data view. Creating connections to your data sources enables you to quickly view across your organization's data. This virtual data platform enables real-time analytics without moving data, duplication, ETLs, and additional storage requirements, so processing times are greatly accelerated. This brings real-time insightful results to decision-making applications or analysts more quickly and dependably than existing methods.

You will use two different data sources on Db2 and Informix, which includes sample records from mortgage industry and creates a virtual table than can query from RStudio.

Before you continue with this tutorial make sure DV is provisioned on your environment. It's an add-on component in ICP for Data.

1. Access Credentials

To work through the demo, you will use Db2 and Informix databases.

1.1. Access credential for Db2 database

JDBC connection credential for Db2:

JDBC Host name	<Same IP address as your web console>
Port number	50000
Database name	MORTGAGE
User ID	db2inst1
Password	password
Db2	Version 11.1
JDBC connection string	jdbc:db2://<same IP as Web Console>:50000/MORTGAGE

1.2. Access credential for Informix database

JDBC connection credential for Informix:

JDBC Host name	<Same IP address as your web console>
Port number	9088
Database name	MORTGAGEDB
User ID	informix
Password	in4mix
Informix	Version 12.10.FC12W1DE
JDBC connection string	<code>jdbc:informix-sqli://<same IP as Web console>:9088/mortgagedb: INFORMIXSERVER=informix;user=informix;password=in4mix</code>

1.3. Setting up the databases and sample tables

- a) Log in to the cluster where ICP for Data is deployed.
- b) From your home directory, clone the tutorial sample files:

```
git clone https://github.com/sanjitc/ICP4XTutorial.git
```
- c) Change to the tutorials directory:

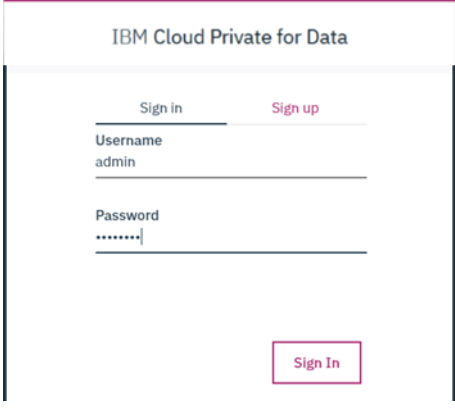
```
cd ICP4XTutorial/tutorials
```
- d) Run the following command to load the sample data into a Db2 database:

```
./load_samples.sh -t mortgage-001
```
- e) Run the following command to load the sample data into an Informix database:

```
./load_samples.sh -t data-virtualization-001
```
- f) After the data loading process completes, instance of Db2 and Informix are hosted on your cluster as a Docker container.

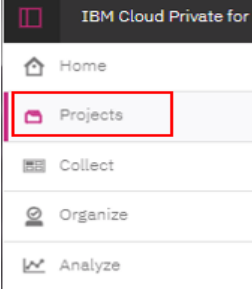

1.4. Sign in to ICP for Data web console as Administrator

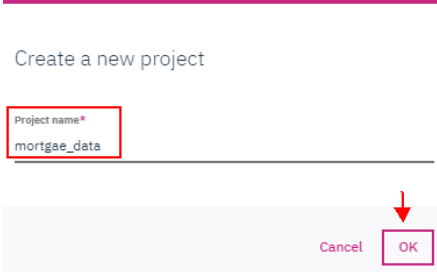
You should use latest version of Firefox or Google Chrome browser to access the ICP for Data web console. Starting from here all instruction needs to execute on ICP for Data web console only. You need to login as admin who has administrator privileges.

	<p>Sign in to the ICPD web console as user 'admin' and password is 'password'.</p>
---	--

2. Create analytic project

Create an analytic project that you going to use for virtualizing data.

	<p>Create a new analytical project by 'Projects' from right pane.</p> <p>Click on the  New project icon</p>
---	--

	<p>Provide a project name as mortgage_data and click OK</p> <p>On the next 'Create project' window, click on Create</p>
---	--

3. Data Virtualization

Data virtualization (DV) integrates data sources across multiple types and locations and turns it into one logical data view. Creating connections to your data sources enables you to quickly view across your organization's data. This virtual data platform enables real-time analytics without moving data, duplication, ETLs, and additional storage requirements, so processing times are greatly accelerated. This brings real-time insightful results to decision-making applications or analysts more quickly and dependably than existing methods.

3.1. Giving users access to data virtualization (optional)

In order for a user to have access to the data virtualization service, you must assign them to appropriate data virtualization roles.

This is for information only. In this demo you will use user 'admin', which has all necessary virtualization roles.

ADD USERS

Grant access to users

Find

<input type="checkbox"/>	Name	Username	Role
<input checked="" type="checkbox"/>	admin	admin	Admin
<input checked="" type="checkbox"/>	deng1	deng1	Engineer
<input checked="" type="checkbox"/>	dst1	dst1	User
<input checked="" type="checkbox"/>	dstw1	dstw1	User

1. Select **Collect > Virtualize data** from left pane
2. Select **Menu > Manage users > Add users** from top
3. Check all users that you created earlier and keep their default role.
4. Click **Add**

3.2. Adding a new data source for Db2

DV supports many relational and non-relational data sources, as well as files that reside on a local disk or network file system, that you can add to your data source ecosystem. After a data source has been added, any user that has virtualize permission has the ability to create virtual tables. DV agents connect to relational data sources using JDBC protocol. In this demo you will add two data sources, one for Db2 and other one for Informix.

Define a data connection to Db2. Use following connection information with 'Add connection'. If you already have an existing database connection, select that for Db2 data source.

Add data source ⓘ

Connect to the data source Step 2 of 2

Data source type
Db2 Family ▼

Host name
[Redacted]

Port
50000

Database Name
MORTGAGE

Username
db2inst1

Password

Options
Enter JDBC connection options

☐ Use SSL
☐ Verify server SSL certificate ⓘ
SSL certificate (optional)

Cancel Back Add

1. Go to **Collect > Virtualized data > Menu > Data sources**
2. Click **Add data source > New data source**
3. Update data source with following information:
Data source type = Db2
Host name = <IP of node 1>
Port = 50000
Database Name = MORTGAGE
Username = db2inst1
Password = password
4. Click **Add**

3.3. Adding a new data source for Informix

Let's add a new data source for the Informix. First define a data connection to informix. Use following connection information with 'Add connection'. If you already have an existing database connection, select that for Informix data source.

Edit connection ⓘ

Data source type
Informix ▼

Host name
[Redacted]

Port
9088

Database Name
mortgagedb

Username
informix

Password

Informix server
informix|

Options
Enter JDBC connection options

☐ Use SSL

1. Go to **Collect > Virtualized data > Menu > Data sources**
2. Click **Add data source > New data source**
3. Update data source with following information:
Data source type = Informix
Host name = <IP of node 1>
Port = 9088
Database Name = mortgagedb
Username = informix
Password = in4mix
Informix server = informix
4. Click **Add**

3.4. Creating virtualized table

The most common mechanism for virtualizing data is to create a "view" or virtual table. You can create a virtual table to segment data from one or more tables. Such segmentation can be vertical (either a subset or superset of columns based on a selection of chosen columns) or horizontal (an explicit set of rows or records based on a conditional expression) or both. You can then run queries against the resulting virtual table.

Virtualize

Find

Filters

Databases

Available tables

12 total available tables

Table	Schemas	Databases	Grouped tables
<input checked="" type="checkbox"/> MORTGAGE_CUSTOMER	DB2INST1	MORTGAGE	1
<input type="checkbox"/> MORTGAGE_DEFAULT	DB2INST1	MORTGAGE	1
<input type="checkbox"/> MORTGAGE_JOIN	DB2INST1	MORTGAGE	1
<input checked="" type="checkbox"/> MORTGAGE_PROPERTY	DB2INST1	MORTGAGE	1
<input type="checkbox"/> bo0	"Informix"	soldb	1
<input type="checkbox"/> energyusage	"Informix"	soldb	1
<input checked="" type="checkbox"/> mortgage_default	"Informix"	mortgagedb	1

Items per page: 10 | 1-10 of 12 items

1 of 2 pages

< 1 >

Menu

1. Click **Collect > Virtualized data > Menu > Virtualize**

2. Select three tables **MORTGAGE_CUSTOMER**, **MORTGAGE_PROPERTY** from MORTGAGE database and **mortgage_default** from mortgagedb, then click **Add to cart**

3. Click **View cart**

4. Click **Next**

1. Select **project** to assign virtualized table to your analytics project. Then, choose the **mortgage_data** project.
2. Choose **publish to catalog** for include virtualized table to the data catalog. This operation will create a publishing request, a data steward must approve the request before the asset is added to the enterprise data catalog.
3. Click **Virtualize** to complete the process

Name the view as **CUSTOMER_DEFAULT** and schema as **ICP4D**, then click **Next**. Make sure use uppercase for view and schema name.

Join virtual objects: Review

Name and review your joined virtual table.

Cancel Back Next

View Name

customer_default1

Schema Name

ice4d

Preview

APPLIED_ONLINE	CARD_DEBT	NO_OF_CARDS	RESIDENCE	INCOME	LOAN_AMOUNT
Y	2698	2	P	45537	8885
Y	44	2	O	49789	9340
Y	645	1	O	44272	10095

1. Select **project** to assign virtualized table to your analytics project. Then, choose the **mortgage_data** project.
2. Click **Create view**

Join virtual objects: Assign

Assign and manage your new joined view below.

Cancel Back Create view

Assign to

☐ Data request ☒ Project ☐ None

mortgage_data

View

customer_default1

Schema

ice4d

Publish to catalog

☒ Yes ☐ No

3.6. Publish virtualized table

A data steward needs approve the published request before the asset is added to the enterprise data catalog.



Pending Publish to Catalog Requests

Search

Name	Type	Project	Owner	Date Updated	Status
icp4d.customer_default1	view	-	admin	8 Jan 2019, 6:00 PM	Pending
Asset		Schema			
customer_default1		icp4d			





Approve

Reject

1. Click on  access the **Home** page
2. Click on **Pending Publish to Catalog Requests**
3. Click on  icon on left for virtual table **CUSTOMER_DEFAULT** that you created
4. Click on **Approve**

3.7. Access information for virtual table

To access virtual table from external application, you need the JDBC connection information. Click on **Collect** > **Virtualized data** > **Menu** > **Service details** to find out access information.

Virtualized data	Virtualize	SQL editor	Data sources	Manage users	Connection details	About
Access information						
Username		user999				
Password		myS-u#C-5#TE419e		 		
JDBC connection URL		jdbc:db2://dv-server.zen.svc.cluster...				

4. Virtual Table From IDE

You can access virtual table from inside or outside of ICP for Data. In this example of R script that you used to access virtual table. ICP for Data comes with and add RStudio that provides an Integrated Development Environment (IDE) for working with R.

4.1. Start IDE




Make sure IDE has started before you can use it.


Select your project **mortgage_data** from project menu

RStudio with R 3.4.3

CPU CORES
—

GPUS
—



Choose **Environments** tab. If RStudio not running, click on  icon to start.

4.2. Launch RStudio

From inside of the analytic project choose **Assets > RStudio** and click on **Open RStudio**.

Home > Projects > mortgage_data1

Assets 10

Data Sources 1

Jobs 0

Environments 6

Collaborators 1

Recent

Data sets 5

Notebooks 0

Scripts 0

Models 0

Model groups 0

Cognos dashboards 0

Data Refinery flows 0

▼ RStudio 5

R 4

TXT 1

Modeler flows 0

Search by RStudio file name

RStudio 5

NAME

TYPE

PATH

LAST MODIFIED

mortgage_dv_demo.R R /rstudio/mortgage_demo/mortgage_dv_demo.R 7 Feb 2019, 5:14 PM

spark_mtcars.R R /rstudio/ibm-sparkaaas-demos/spark_mtcars.R 7 Feb 2019, 5:05 PM

spark_kernel_basic.R R /rstudio/ibm-sparkaaas-demos/spark_kernel_basic.R 7 Feb 2019, 5:05 PM

spark_flights.R R /rstudio/ibm-sparkaaas-demos/spark_flights.R 7 Feb 2019, 5:05 PM

Open RStudio

4.3. Load R script

Within the RStudio, go to File > New File > R script to run script. Copy the following script and paste it in the source section. This simple R script will simply run a SELECT statement against the virtual table and retrieve data and create a treemap to visually displays mortgage default by residency type.

Before you run the script, please change the **url**, **databaseUsername** and **databasePassword** according to your system, which you found in step 3.7 (Access information for virtual table).

```
#####
### This is a sample R code that access the virtual table from Data Virtualization environment and create a treemap
### to visually displays mortgage default by residency type.
###
### Before executing the R script update URL, databaseUsername and databasePassword variables according to your environment .
### Variable values can be found in 'Collect -> Virtualized data -> Menu -> About' section in Data Virtualization environment.
#####

library(RJDBC)                # Load the "RJDBC" package
library(treemap)              # Call the TREEMAP package
library(dplyr)                # Load DPLYR package

driverClassName <- "com.ibm.db2.jcc.DB2Driver" # Load JDBC driver using Class.forName method
driverPath <- "/dbdrivers/db2jcc4.jar"         # Db2 JDBC driver path in RStudio docker image

url <- "<JDBC Connection URL>"                # Update JDBC connection URL according to your
                                              # environment information in 'Collect -> Virtualized
                                              # data -> Menu -> About'

databaseUsername <- "<username>"              # Database user
databasePassword <- "<password>"              # Database user password

drv <- JDBC(driverClassName, driverPath)        # Initialize Java VM and load Java JDBC driver
conn <- dbConnect(drv, url, databaseUsername, databasePassword) # Create JDBC connection using dbConnect()

selStr <- "SELECT * FROM ICP4D.MORTGAGE_CUSTOMER_DEFAULT
          WHERE \"mortgage_default\" = 'YES';"   # Query string for retrieve dataset

md <- dbSendQuery(conn, selStr)                 # Let's run a query
mortgageData <- fetch(md, -1)                  # Store query result in a data frame

mortgageData1 <- mortgageData %>%              # Expand residency type to meaningful name
  mutate(RESIDENCE = case_when(RESIDENCE == "O" ~ "Owner Occupier",
                               RESIDENCE == "P" ~ "Private Renting",
                               RESIDENCE == "L" ~ "Living With Parents",
                               RESIDENCE == "S" ~ "Sheltered"))

head(mortgageData1)                           # Return first parts of data frame

treemap(mortgageData1,                        # Create a treemap to visually displays,
        index = c("RESIDENCE"),              # mortgage default by residency type.
        vSize = "YRS_CURRENT_EMP",
        title = "Mortgage Default by Residency Type"
        )

dbDisconnect(conn)                            # Disconnect and close the connection
```

If any R packages not already installed, you need to install them first before execute the above script. Use **install.packages("<package name>")** command to install a R package.

4.4. Run the script

Highlight the R script in the RSudio and click on **Run** to retrieve data.

Projects > mortgage_data > RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

999

Project: (None)

mortgage-select-1.R

Source on Save

Run

Source

```

3 library(RJDBC)           # Load the "RJDBC" package
4 library(treemap)         # Call the TREEMAP package
5 library(dplyr)           # Load DPLYR package
6
7 driverClassName <- "com.ibm.db2.jcc.DB2Driver" # Load JDBC driver using Class.forName method
8 driverPath <- "/dbdrivers/db2jcc4.jar"         # Db2 JDBC driver path in RStudio docker image
9
10 url <- "jdbc:db2://dv-server.zen.svc.cluster.local:32051/bigsql" # JDBC connection URL for database bigsql on DV
11 databaseUsername <- "user999" # Database user
12 databasePassword <- "myS-uPC-5#TE419e" # Database user password
13
14 drv <- JDBC(driverClassName, driverPath) # Initialize Java VM and load Java JDBC driver
15 conn <- dbConnect(drv, url, databaseUsername, databasePassword) # Create JDBC connection using dbConnect()
16
17 selStr <- "SELECT * FROM ICP40.MORTGAGE_CUSTOMER_DEFAULT" # Query string for retrieve dataset
18 WHERE `mortgage_default` = 'YES';"
19
20 md <- dbSendQuery(conn, selStr) # Let's run a query
21 mortgaseData <- fetch(md, n=1) # Store query result in a data frame
38.1 (Top Level) >

```

Console

Terminal

```

2      20      1 Owner Occupier
3      14      1 Owner Occupier
4      27      2 Private Renting
5       9      1 Private Renting
6      15      2 Sheltered

```

```

> treemap(mortgaseData,
+   index = c("RESIDENCE"), # Create a treemap to visually displays,
+   vsize = "YRS_CURRENT_EMP", # mortgage default by recedency type.
+   title = "Mortgage Default by Residency Type"
+ )
>
> dbDisconnect(conn) # Disconnect and close the connection
[1] TRUE

```

Environment History Connections

Global Environment

mortgaseData

180 obs. of 11 variables

Values

databasePassword	"myS-uPC-5#TE419e"
databaseUsername	"user999"
driverClassName	"com.ibm.db2.jcc.DB2Driver"
driverPath	"/dbdrivers/db2jcc4.jar"
selStr	"SELECT * FROM ICP40.MORTGAGE_CUSTOMER_DEFAULT \n WHERE `mortgage_d_`"
url	"jdbc:db2://dv-server.zen.svc.cluster.local:32051/bigsql"

Files Plots Packages Help Viewer

Zoom Export Publish

Mortgage Default by Residency Type

15