



Experience AI queries through SQL Data Insights

Guillaume Arnould
Matilda Dika
Andrea Di Maio
Fabrizio Pietrucci

Table of Contents

DOCUMENT ABSTRACT AND PURPOSE	4
1.0 OVERVIEW 5	
1. <i>SQL DATA INSIGHTS OVERVIEW.....</i>	5
2. <i>AI SEMANTIC BUILT-IN FUNCTIONS</i>	5
A. <i>AI_SIMILARITY</i>	6
B. <i>AI_SEMANTIC_CLUSTER.....</i>	6
C. <i>AI_ANALOGY</i>	7
3. <i>SETTING UP SQL DATA INSIGHTS</i>	8
A. <i>STEP 1: Db2 AND z/OS SETUP</i>	8
B. <i>STEP 2: CONFIGURE SQL DATA INSIGHTS UI AND TRAINING SERVICES</i>	8
C. <i>STEP 3: ENABLE AI.....</i>	9
D. <i>STEP 4: ANALYZE DATA.....</i>	9
E. <i>STEP 5: RUN AI QUERIES</i>	9
2.0 LAB EXERCISES PREPARATION	10
SQDL DATA INSIGHT HANDS-ON LAB ENVIRONMENT	11
SQLDI & DVM DOCUMENTATION AND LAB WORKBOOKS.....	12
3.0 ACCESSING SQL DATA INSIGHTS UI.....	13
3. <i>CONNECTING TO YOUR DATABASE</i>	16
4. <i>ENABLE AI</i>	18
5. <i>MONITOR AI OBJECT TRAINING</i>	23
6. <i>REVIEW THE TRAINING RESULTS AND INTERPRETABILITY</i>	24
7. <i>ENABLE AI ON A Db2 FOR z/OS VIEW</i>	28
8. <i>MONITOR AI OBJECT TRAINING</i>	33
4.0 RUN SEMANTIC QUERIES	35
5.0 <i>ENABLE AI ON A DB2 FOR Z/OS USER DEFINED TABLE FUNCTION USING DVM (DATA VIRTUALIZATION MANAGER)</i>	41

Disclaimer

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results like those stated here.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings, or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs, or services available in all countries in which IBM operates or does business.

Workshops, sessions, and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

Document Abstract and Purpose

SQL Data Insights (SQL DI) is an AI powered feature of Db2 for z/OS®. SQL DI uses deep learning which exploits advanced IBM® Z hardware/software technologies to extend the standard SQL language and infuse the Db2 engine with SQL-based semantic queries on tables and views.

The first section of this lab provides an Overview of SQL DI for an understanding of the semantic queries included in SQL DI and a high-level description of how to configure and enable this powerful feature.

The second section will take you through several hands-on exercises to show you how to login to the SQL DI user interface and connect to a SQL DI enabled Db2 subsystem. You will learn how train a table to support semantic queries by enabling AI against the relevant columns. Then, you will invoke the various types of queries covered in section one against your newly trained table, which exhibits how SQL DI delivers powerful and efficient AI-powered queries to Db2.

1.0 Overview

1. SQL Data Insights overview

SQL DI is an optionally installable feature of Db2 13 for z/OS that brings AI capabilities directly to your Db2 data without requiring the costly extract-transform-load (ETL) processes to move the data off the IBM Z mainframe or hiring a team of data scientists to design machine-learning models. You can simply choose a Db2 table or view to enable for the AI query functionality and then use SQL to issue semantic queries against the data.

After you install the SQL DI graphical user interface (GUI) component that runs on z/OS, you can browse and select from a set of Db2 tables and views and enable them for AI query. Enabling AI query on a table or view trains an unsupervised neural network deep learning model that is stored in Db2.

After a Db2 table or view is enabled for an AI query, you can run AI semantic queries against the table or view by using SQL. You can run the SQL through the built-in SQL editor, an application program, or a SQL tool such as SPUFI or QMF. AI semantic queries utilize the semantic similarity, clustering, and analogy functions that are built into the Db2 engine.

2. AI semantic built-in functions

Db2 13 supports three AI semantic built-in functions shown in the table below. The semantic query capability that is provided by SQL DI can be applied to use cases across many industries.

You can use similarity queries to find groups of similar entities to decide on market segmentation or find a group of customers that behaves similarly to other groups, which have applications in the retail, finance, and insurance industries.

You can use dissimilarity queries to find outliers from the norm, which has applications in financial anomaly detection and fraud detection.

You can use semantic clustering queries to form a cluster of entities to test whether an extra entity belongs in the cluster. You can use semantic clustering in many contexts where similarity or dissimilarity queries are used as a broader test of similarity or dissimilarity to multiple entities.

You can use analogy queries to determine whether a relationship between a pair of entities applies to a second pair of entities, which has applications in retail, such as to determine whether a customer has a preference for a product and whether other customers have the same degree of preference for other products.

Db2 built-in AI semantic function	Semantic query type
AI_SIMILARITY	Similarity query
AI_SIMILARITY	Dissimilarity query
AI_SEMANTIC_CLUSTER	Inductive reasoning semantic clustering query
AI_ANALOGY	Inductive reasoning analogy query

a. AI_SIMILARITY

AI_SIMILARITY computes a similarity score using the values returned by expression-1 and expression-2. Results of AI_SIMILARITY is a score representing as a floating-point number between -1.0 and 1.0. A score of 1.0 means that the two entities are similar (or the same), and a score of -1.0 means that the two entities are dissimilar.

AI_SIMILARITY function can be used for similarity and dissimilarity by sorting the AI_SIMILARITY score in descending and ascending order, respectively.

```
AI_SIMILARITY (expression-1 USING MODEL COLUMN column-name,  
                expression-2 USING MODEL COLUMN column-name )
```

Conceptual Examples:

```
AI_SIMILARITY ('BLACKBERRY', 'RASPBERRY' USING MODEL COLUMN FRUIT)
```

```
AI_SIMILARITY ('APPLE', 'RASPBERRY' USING MODEL COLUMN FRUIT)
```

The first function that compares 'BLACKBERRY' and 'RASPBERRY' returns a similarity score of 0.86 because both are berries and close in size.

The second function that compares 'APPLE' and 'RASPBERRY' returns a similarity score of 0.45, which indicates that there is some similarity, but the two are not as similar as in the other example. Even though apples and raspberries are fruits, they are not both berries, and they are not similar in size.

If the expression argument is a standalone column reference and no model column is explicitly specified with the USING MODEL COLUMN clause, the standalone column is the model column that is used to evaluate the function, and the model is the model table associated with the table that the column belongs to.

Conceptual Examples:

```
AI_SIMILARITY (FRUIT, 'BLACKBERRY')
```

b. AI_SEMANTIC_CLUSTER

AI_SEMANTIC_CLUSTER creates a cluster of up to 3 entities and produces a score that indicates how strongly an additional entity belongs to the cluster.

Results of AI_SEMANTIC_CLUSTER is a score representing as a floating-point number between -1.0 and 1.0. A score of 1.0 indicates a strong fit between the member argument and the cluster, and a score of -1.0 indicates a poor fit.

```
AI_SEMANTIC_CLUSTER (member-expression USING MODEL COLUMN column-name,  
                           clustering-expressions)
```

Conceptual Examples:

```
AI_SEMANTIC_CLUSTER ('APPLE' USING MODEL COLUMN FRUIT,  
                           'RASPBERRY', 'BLACKBERRY', 'BLUEBERRY')
```

```
AI_SEMANTIC_CLUSTER ('STRAWBERRY' USING MODEL COLUMN FRUIT,  
                           'RASPBERRY', 'BLACKBERRY', 'BLUEBERRY')
```

The first example creates a cluster of the last three arguments, 'RASPBERRY', 'BLACKBERRY', and 'BLUEBERRY' and returns a score describing the first argument's inclusion into the cluster. The first argument, 'APPLE', is a fruit, but it might not fit well in this cluster because it is not a berry and is sized differently. The score that is computed for this function is 0.23.

The second example creates the same cluster with the last three arguments. When the first argument 'STRAWBERRY' is tested for inclusion into the group it's score is 0.82. 'STRAWBERRY' is much more related to the clustered group than 'APPLE' because it is a berry and closer in size.

If the *member-expr* is a standalone column reference and no model column is explicitly specified with the USING MODEL COLUMN clause, the standalone column is the model column that is used to evaluate the function, and the model is the model table associated with the table that the column belongs to.

Conceptual Examples:

```
AI_SEMANTIC_CLUSTER(FRUIT, 'APPLE')
```

c. AI_ANALOGY

AI_ANALOGY computes the relationships across two pairs of entities and returns an analogy score. The result of AI_ANALOGY is a floating-point number, but not bounded by -1.0 and +1.0. A higher number indicates a stronger analogy relationship.

```
AI_ANALOGY(source-1, target-1, source-2, target-2)
```

Conceptual Examples:

```
AI_ANALOGY('STRAWBERRY' USING MODEL COLUMN FRUIT, 'RED', 'LEMON',  
           'YELLOW')
```

```
AI_ANALOGY('STRAWBERRY' USING MODEL COLUMN FRUIT, 'RED', 'BLUEBERRY',  
           'ORANGE')
```

Consider the analogy that a strawberry is to red as a lemon is to yellow. To determine whether this analogy is a good one, think about the relationship between the first two entities 'STRAWBERRY' and 'RED'. In this example the relationship is based on color. Naturally, one might think that a strawberry is colored red. But does the same relationship apply to the second pair of entities 'LEMON' and 'YELLOW'? The relationship is applied to the second pair of arguments, 'LEMON' and 'YELLOW' to determine whether the analogy is good. In the first example, the analogy is good, so it has a higher analogy score of 0.87.

The second example shown above represents a poor analogy. The first two arguments remained the same as before, thus the relationship is still color-based. When the color-based relationship is examined for the second pair of arguments, 'BLUEBERRY' and 'ORANGE', it does not hold well. Blueberries are not colored orange, so the analogy score is much lower at -0.24.

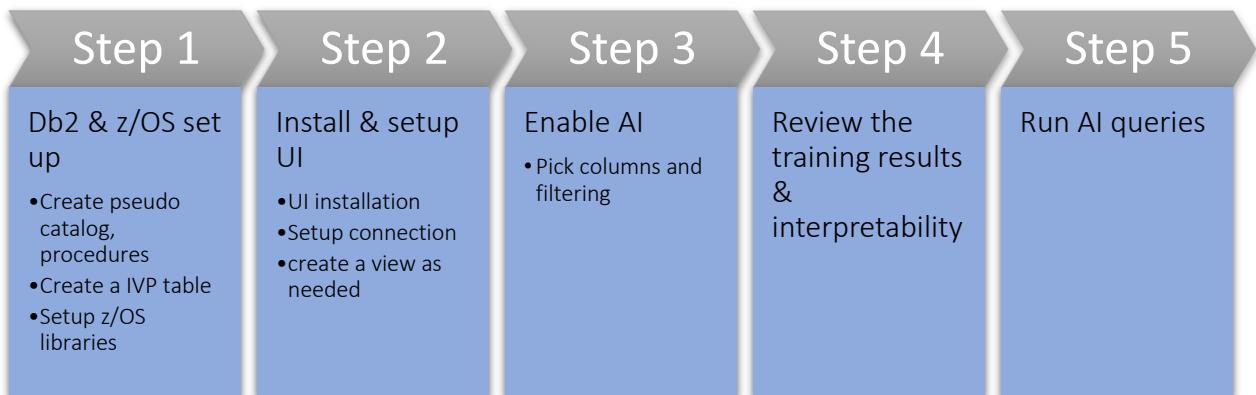
If an expression argument is a standalone column reference and no model column is explicitly specified with the USING MODEL COLUMN clause, the standalone column is the model column that is used to evaluate the function, and the model is the model table associated with the Db2 table that the column belongs to.

Conceptual Examples:

```
AI_ANALOGY(FRUIT, 'STRAWBERRY')
```

3. Setting up SQL Data Insights

In this HOL, the environment is already set up for you. For your reference, the steps to enable AI queries are below. The actual hands on exercises start at step 3, as shown in the image below, and are covered in section 2 of this lab.



a. Step 1: Db2 and z/OS setup

Update Db2 13 Function Level V13R1M500. JDBC application used by SQL DI user interface requires APPLCOMPAT V13R1M500

Set up required Db2 objects (DSNTIJAI) : Create pseudo catalog, stored procedure, model table data base.

Create installation sample table: DSNAIDB.CHURN (DSNTIJAV)

Install necessary z/OS prerequisite for

- IBM Z AI Data Embedding library
- IBM Z AI Optimization library
- IBM Z Deep Neural Network library.

b. Step 2: Configure SQL Data Insights UI and training services

Set up ZFS directory, identify ports.

Follow the installation script `./sqldi.sh create`.

Set up RACF user-id (RACF group SQLDIGRP).

Add connection to Db2 for z/OS.

Update setting of training parameters as needed

- Spark setup
- Parallel threads for preprocessing

- Db2 load utility information (zLOAD).

NOTE: The next 3 steps are actually covered in Section 2 of this lab.

c. Step 3: Enable AI

From SQL Data Insights UI, select the object to enable AI and start “enable AI”. This invokes training under the cover.

d. Step 4: Analyze Data

Once the training is completed, examine the data using “analyze data” from SQL Data Insights UI.

e. Step 5: Run AI queries

Execute queries against the object with AI enabled from any Db2 interface. In the lab, we will use “RUN query” interface in UI.

2.0 Lab Exercises Preparation

In this section you will log into SQL Data Insights to learn how to enable and use AI powered semantic queries against your Db2 tables.

Icons

The following symbols appear in this document at places where additional guidance is available.

Icon	Purpose	Explanation
	Important!	This symbol calls attention to a particular step or command. For example, it might alert you to type a command carefully because it is case sensitive.
	Information	This symbol indicates information that might not be necessary to complete a step but is helpful or good to know.
	Troubleshooting	This symbol indicates that you can fix a specific problem by completing the associated troubleshooting information.

SQL Data Insight Hands-On Lab Environment

SQLDI Software

SQLDI software is installed on ZB01 (Montpellier Client Engineering for Systems) z/OS host. Ask the Administrator of that environment to get your access.

DVM Server Software

DVM Server software is also installed on ZB01 (Montpellier Client Engineering for Systems) z/OS host. Ask the Administrator of that environment to get your access.

DVM Client Software on IBM provided laptops

When available, IBM will provide laptops or VMs for the workshops with preloaded VMWare images with DVM client software.

SQLDI & DVM Documentation and Lab Workbooks

SQL Data Insights Technical documentation:

- <https://www.ibm.com/docs/en/db2-for-zos/13?topic=running-ai-queries-sql-data-insights>

DVM Technical Documentation:

- <https://www.ibm.com/docs/en/dvm/1.1.0>

Lab Workbook documents are available in IBM box folder:

- <https://ibm.ent.box.com/folder/171652054325>

Details on FreddieMac datasets can be found directly here. For this Hands-on lab we used family loans 1Q2021 data from FreddieMac.

- <https://www.freddiemac.com/research/datasets/sf-loanlevel-dataset>

3.0 Accessing SQL Data Insights UI

1. Purpose of the hands-on labs

The purpose of the hands-on labs is to get you familiar with how to unlock Db2 for z/OS data relationship using IBM Db2 for z/OS SQL Data Insight by exercising different scenarios.

2. Content of the hands-on labs

The hands-on labs consist of the following 3 parts:

- 1.1. Discover SQL Data Insight Web User interface to enable Db2 for z/OS tables and Db2 view for AI semantic queries. Run your first semantic query.
- 1.2. Run several Semantic queries on the “enabled for AI” data to explore semantic concepts: similarity, analogy, clustering.
- 1.3. Virtualize Db2 and VSAM data using Data Virtualization Manager for z/OS (DVM). Create consolidate view in DVM and publish it as a Db2 UDTF to be used by SQL Data Insight.

3. Hands-on lab environment

When the lab starts, you will need to start and connect to VM Windows machines where all components are already configured for you. Please find the VM Windows Image URL and the corresponding username and password assigned to you by the instructors to login your virtual machine to start your lab. For example, if you are assigned Student 1, your virtual machine can be accessed from:

<https://cloud.skytap.com/v2/vms/61145be6173bde13324cd949382de0f0/desktops> and your password to login is: *78uaVG+a7Aiezh9*

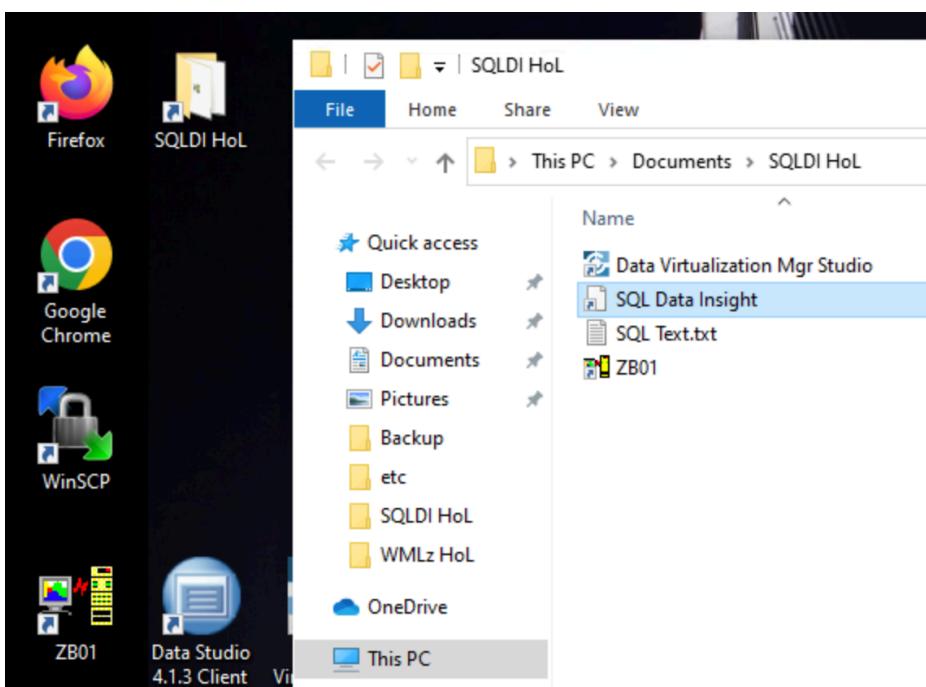
Student	Skytap Image URL	Skytap Password	Windows User/Password	SQLDI User/Password
Student1	https://cloud.skytap.com/v2/vms/61145be6173bde13324cd949382de0f0/desktops	78uaVG+a7Aiezh9	IBM-DATAZ / data4IBMz	zdw01 / pwd01mop
Student2	https://cloud.skytap.com/v2/vms/6eb743f2f48ea8e3f688873b86ef7d9a/desktops	97%!m9gAy2yq_4v	IBM-DATAZ / data4IBMz	zdw02 / pwd02mop
Student3	https://cloud.skytap.com/v2/vms/d5e5685d80c4ccfd_b6f3d4552aae2c2d/desktops	9qvtj28S%5a2XVt	IBM-DATAZ / data4IBMz	zdw03 / pwd03mop
Student4	https://cloud.skytap.com/v2/vms/4856a7ba4dfbc12c760e3e704f9b545a/desktops	3P+UH-QrPol6KHF	IBM-DATAZ / data4IBMz	zdw04 / pwd04mop
Student5	https://cloud.skytap.com/v2/vms/7fcbb9906bce635777e86dc4afe4f60ff/desktops	2y8Ug+Lc_t4tr8N	IBM-DATAZ / data4IBMz	zdw05 / pwd05mop
Student6	https://cloud.skytap.com/v2/vms/88747b796e237a3feec92ad21733f53e/desktops	-jLt4Udj4Q057%	IBM-DATAZ / data4IBMz	zdw06 / pwd06mop
Student7	https://cloud.skytap.com/v2/vms/31fb5866c19f511c27813a2ae319b53/desktops	KR%GRTL2hY_LO_T	IBM-DATAZ / data4IBMz	zdw07 / pwd07mop
Student8	https://cloud.skytap.com/v2/vms/0b05285c837db5ddc80e774c666da3d6/desktops	Yp!O+YqMs2P6Cm2	IBM-DATAZ / data4IBMz	zdw08 / pwd08mop
Student9	https://cloud.skytap.com/v2/vms/c52ec66ca3301dec44102ffd68da11c4/desktops	8N5tl5SYB6WOn3+	IBM-DATAZ / data4IBMz	zdw09 / pwd09mop
Student10	https://cloud.skytap.com/v2/vms/535b78beb53589f895822e778abe32eb/desktops	bEdq4d%jhCkuH0w	IBM-DATAZ / data4IBMz	zdw10 / pwd10mop
Student11	https://cloud.skytap.com/v2/vms/856b13ac90c21d10ece1176359540ce1/desktops	LHRV9i+8LP0v-aB	IBM-DATAZ / data4IBMz	zdw11 / pwd11mop
Student12	https://cloud.skytap.com/v2/vms/fb9fa7d7283d0b6b986bd32e62ad4b98/desktops	I3W0lQ94mXTc4-M	IBM-DATAZ / data4IBMz	zdw12 / pwd12mop
Student13	https://cloud.skytap.com/v2/vms/939c356c14d6bc9c35529741607e0910/desktops	J_H!_9Hdaccbu4V	IBM-DATAZ / data4IBMz	zdw13 / pwd13mop
Student14	https://cloud.skytap.com/v2/vms/b3714f713230518c30f3b732803e577a/desktops	p9J86CXBr+w1Q-	IBM-DATAZ / data4IBMz	zdw14 / pwd14mop

Student15	https://cloud.skytap.com/v2/vms/4a9e712817add61f8cd8b04728175a74/desktops	Bf+WIek9bRd%uCq	IBM-DATAZ / data4IBMz	zdw15 / pwd15mop
Student16	https://cloud.skytap.com/v2/vms/06c04c6900e6e6153a32a4137292e210/desktops	OZ!BrCABxhc7kA%	IBM-DATAZ / data4IBMz	zdw16 / pwd16mop
Student17	https://cloud.skytap.com/v2/vms/c86398098584ee3997619313e8fc0f6d/desktops	8K10EW!%UG%eRD3	IBM-DATAZ / data4IBMz	zdw17 / pwd17mop
Student18	https://cloud.skytap.com/v2/vms/ed13268c3546ef5e5270d914c3328f28/desktops	Z%9gcvuy_K_xKxq	IBM-DATAZ / data4IBMz	zdw18 / pwd18mop
Student19	https://cloud.skytap.com/v2/vms/186595446f9a879a77fbfc4badc6692e/desktops	tL+0h_lwsXtxGI4	IBM-DATAZ / data4IBMz	zdw19 / pwd19mop
Student20	https://cloud.skytap.com/v2/vms/af0ce91e06348dfc07dfeba4c9962f71/desktops	we5ef!vRgKRo-Lg	IBM-DATAZ / data4IBMz	Zdw20 / pwd20mop
Student21	https://cloud.skytap.com/v2/vms/7b6fd17b3f857512faa25216f322400d/desktops	2qZml8Tv_y!7IW1	IBM-DATAZ / data4IBMz	Zdw21 / pwd21mop
Student22	https://cloud.skytap.com/v2/vms/09f6437e059de40a4c0fb06b244e4457/desktops	APGHZG%_DYENOIO	IBM-DATAZ / data4IBMz	Zdw22 / pwd22mop
Student23	https://cloud.skytap.com/v2/vms/48b05dd9d3d7e524ec65966a99053457/desktops	k9+GgiRstoF2bb7	IBM-DATAZ / data4IBMz	Zdw23 / pwd23mop
Student24	https://cloud.skytap.com/v2/vms/7c8635621f78397069a53a44d2977765/desktops	95kwfl5tjt-Vk!k	IBM-DATAZ / data4IBMz	Zdw24 / pwd24mop
Student25	https://cloud.skytap.com/v2/vms/f33e2c0252e58b621e5f892615ea15ad/desktops	Rm!niyr5%zddxl+	IBM-DATAZ / data4IBMz	Zdw25 / pwd25mop
Student26	https://cloud.skytap.com/v2/vms/42931ef332160be549ddc307d300213/desktops	2w-mBKotq65as_!	IBM-DATAZ / data4IBMz	Zdw26 / pwd26mop

Your SQLDI logon URL is shared across all users and is: <https://10.3.58.61:15001/sqldi>

2. Logon to SQL Data Insights User Interface

- On the desktop of your lab machine, you should see a folder called “SQLDI HoL”.
- Open that folder.
- In it, you should see a URL (link) for “SQL Data Insight”





If you do not see the SQDI HoL folder or the “SQL Data Insight” link in the folder, then refer to your lab worksheet. You will find value for <login url>
Open a browser and type that address into your browser

- Opening the URL by either method, you may see the following warning:



Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to bac1.dmz. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

What can you do about it?

The issue is most likely with the website, and there is nothing you can do to resolve it.

If you are on a corporate network or using anti-virus software, you can reach out to the support teams for assistance. You can also notify the website's administrator about the problem.

[Learn more...](#)

[Go Back \(Recommended\)](#)

[Advanced...](#)

- If so, **click** on advanced and then the **Accept the Risk and Continue** button that appears in the subsequent dialog.
- The SQL Data Insights **Sign in** dialog should appear.
- **Enter** the values for <username> and <password> provided on the lab worksheet, and **click** the **Sign in** button

Sign in to SQL Data Insights

Username

Password

 (

Sign in

3. Connecting to your database

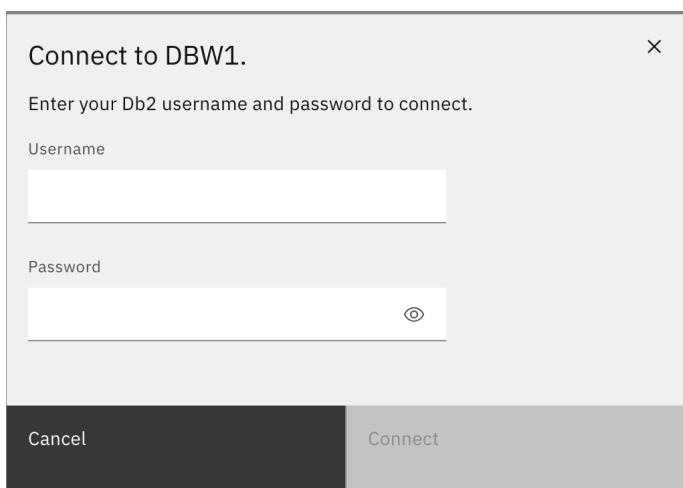
Once logged in, you should be presented with the Connections page, listing the database subsystems that have been connected to SQL DI.

- From this list, find your database connection named DBW1. Click on the '3-dot action menu' on the right side of the row for your connection and select **Connect**.

Connections

Connections				Add connection +
Name	Host/Location	Type	Updated on	
DBW1	10.3.58.61/RDBNDBW0	Db2 v13	Aug 30, 2022 12:51 PM	
				<div>Connect Edit List AI objects Delete</div>

- You should be presented with an authentication dialog to connect to your database.
- Enter the same <username> and <password> values from your worksheet and click **Connect**



- Click on the '3-dot action menu' on the right side of the row for your connection and select **List AI objects**

Connections

Name	Host/Location	Type	Updated on
DBW1 ✓	10.3.58.61/RDBNDBW0	Db2 v13	Aug 30, 2022 12:51 PM

⋮ |
 Disconnect |
 Edit |
 List AI objects |
 Delete

You will be shown a view with the list of AI objects which have been created for the database.

⚠ Please note that during this lab you will likely see objects other than the one that you will create and work with, so make sure to only use your AI object which will be identified by the schema name <schema> from your lab worksheet.

AI objects

DBW1				
<input type="text" value="churn"/> × Add object + Run query ✖				
Name	Schema	Type	Status	Last updated
CHURN	DSNAIDB	Table	✓ Enabled	Aug 30, 2022 2:20 PM
Resources per page: 10 1–1 of 1 items 1 of 1 page ◀ ▶				

4. Enable AI

In this step, you will select a table and create an AI object which involves training a model against the selected table data so that the related table is “enabled” for the semantic queries supported by SQL DI.

- Click on the **Add object** button at the top-right of the AI Objects list to select the table to be enabled for AI.

AI objects

DBW1				
Name	Schema	Type	Status	Last updated
CHURN	DSNAIDB	Table	Enabled	Aug 30, 2022 2:20 PM
Resources per page: 10 1–1 of 1 items				

- You will be presented with the **Add object** view. From it, click on the **schema** drop-down list. In the list, scroll down to your assigned <schema> name from your worksheet. Click the **check-box** for your schema name, and then click outside of the list box to close it.

Add object

DBW1

Choose one or more schemas to list associated Db2 objects. From the list, select the Db2 objects to add as new AI objects or enable for AI query.

Schema: 1 × Select schema ▾

Nam Schema Last activity

<input checked="" type="checkbox"/> ZDW10	ZDW10	
<input type="checkbox"/> ZDW11		
<input type="checkbox"/> ZDW12		
<input type="checkbox"/> ZDW13		
<input type="checkbox"/> ZDW14		

No object available. Select one or more schemas to list available objects.

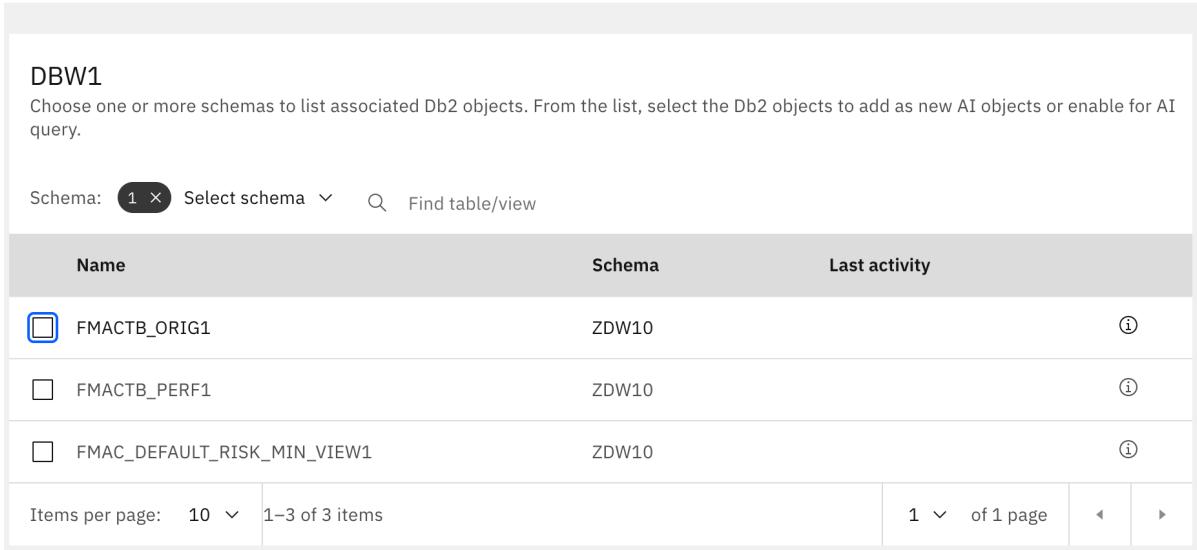
Items per page: 10 0–0 of 0 items

Your user's FreddieMac tables and view should be listed in the displayed page. If the row for your **FMACTB_ORIG1** table seems disabled (the checkbox in the name column is gray and not selectable) contact one of the lab administrators.

- Preview the data to get an idea of the type of data in the table:

- Click on the information icon  located on right side of the row for your table **FMACTB_ORIG1**

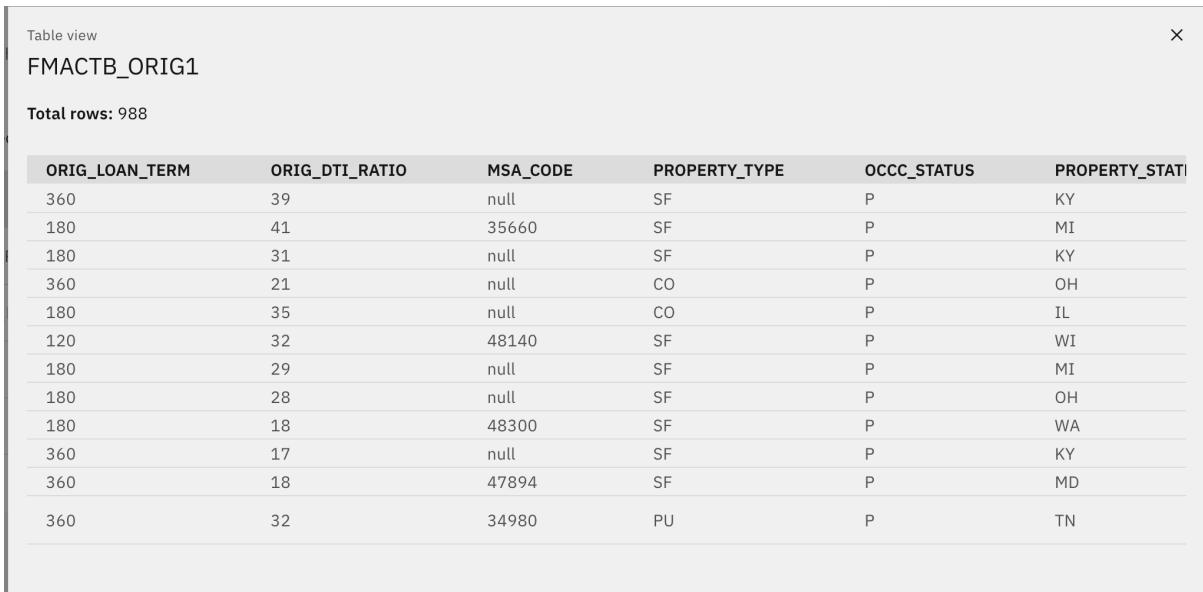
Add object



The screenshot shows the DBW1 interface. At the top, it says "DBW1" and "Choose one or more schemas to list associated Db2 objects. From the list, select the Db2 objects to add as new AI objects or enable for AI query." Below this is a search bar with "Schema: 1 X Select schema" and a "Find table/view" button. The main area is a table with three columns: "Name", "Schema", and "Last activity". It lists three tables: FMACTB_ORIG1 (Schema ZDW10), FMACTB_PERF1 (Schema ZDW10), and FMAC_DEFAULT_RISK_MIN_VIEW1 (Schema ZDW10). Each table has an information icon (i) next to it. At the bottom, there are pagination controls: "Items per page: 10" (with a dropdown arrow), "1-3 of 3 items", "1 of 1 page", and navigation arrows.

Name	Schema	Last activity
<input type="checkbox"/> FMACTB_ORIG1	ZDW10	
<input type="checkbox"/> FMACTB_PERF1	ZDW10	
<input type="checkbox"/> FMAC_DEFAULT_RISK_MIN_VIEW1	ZDW10	

A popup window should appear with an indication of the number of records in the table as well as a sample of rows returned for the table, helping you to understand the type of data in the table. If instead, you get an error trying to access your **FMACTB_ORIG1** table, contact a lab administrator.



The screenshot shows a "Table view" of the FMACTB_ORIG1 table. It displays 988 total rows. The table has six columns: ORIG_LOAN_TERM, ORIG_DTI_RATIO, MSA_CODE, PROPERTY_TYPE, OCCC_STATUS, and PROPERTY_STATI. The data is as follows:

ORIG_LOAN_TERM	ORIG_DTI_RATIO	MSA_CODE	PROPERTY_TYPE	OCCC_STATUS	PROPERTY_STATI
360	39	null	SF	P	KY
180	41	35660	SF	P	MI
180	31	null	SF	P	KY
360	21	null	CO	P	OH
180	35	null	CO	P	IL
120	32	48140	SF	P	WI
180	29	null	SF	P	MI
180	28	null	SF	P	OH
180	18	48300	SF	P	WA
360	17	null	SF	P	KY
360	18	47894	SF	P	MD
360	32	34980	PU	P	TN

- Click the close button on the top right of the pop-up window once you are finished previewing the data.

- **Select** the checkbox to the left of the Name column for your table.
- **Click** the **Enable AI query** button on the bottom-right of the screen, which should have become enabled once you selected your **FMACTB_ORIG1** table.

Add object

The screenshot shows a 'DBW1' interface for selecting schemas. A table lists objects with columns 'Name', 'Schema', and 'Last activity'. The 'Name' column contains checkboxes. The row for 'FMACTB_ORIG1' has a checked checkbox in the first column. The 'Schema' column shows 'ZDW10' for all rows. The 'Last activity' column shows an info icon for each row. At the bottom, there are buttons for 'Cancel', 'Add object', and 'Enable AI query'.

Name	Schema	Last activity
<input checked="" type="checkbox"/> FMACTB_ORIG1	ZDW10	(i)
<input type="checkbox"/> FMACTB_PERF1	ZDW10	(i)
<input type="checkbox"/> FMAC_DEFAULT_RISK_MIN_VIEW1	ZDW10	(i)

Items per page: 10 1–3 of 3 items

Cancel Add object Enable AI query

The column configuration dialog should be displayed next.

- Select and configure the columns to be analyzed (shown on the next page)

The next page shows the Column configuration page. This allows you to select the columns from the table to be used in the modeling as well as adjust the data-types that will be used for each column.

For this exercise, we will use all columns in the table.

At this point, you have the option to re-categorize your columns to a different SQL DI data type. However, you have to understand the data to know what SQL DI data types are valid for that given column.

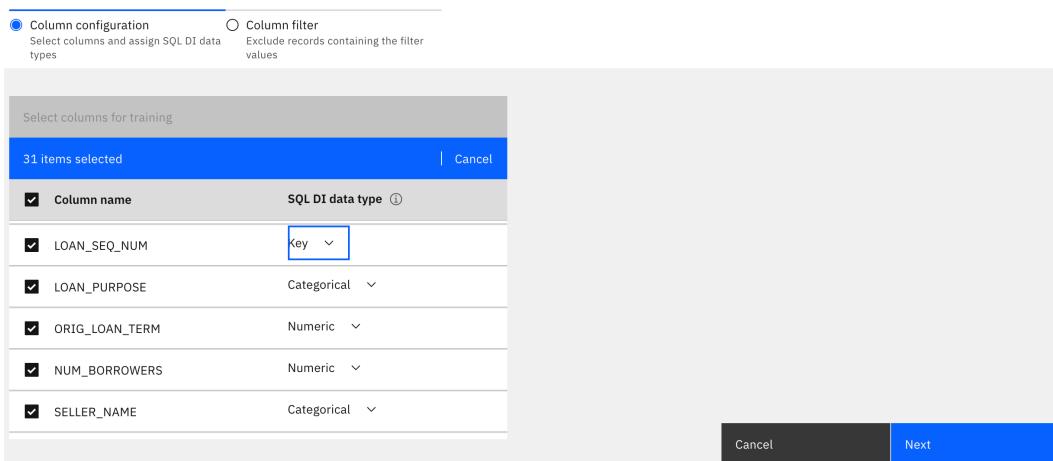
- Data of the SQL DI categorical type are discrete values that are treated separately if they are not exactly equal. A column with a numeric SQL type can be given the SQL DI categorical type if the values are intended to be treated individually. A decimal column that holds an account ID or a social security number is a good example of when you might want to choose an SQL DI categorical type for a column of a numeric SQL type.
 - Data of an SQL numeric type are continuous values where values that are “close” to one another are considered together. Only columns that have numeric SQL types can be chosen to have a numeric SQL DI type. A numeric column that holds the price of a good is an example of a numeric SQL DI type, where close values such as \$9.99 and \$10.00 might be considered together.
 - A third SQL DI data type is key. Assign the SQL DI key type to columns that represent the entire row in an AI query. If your table or view has a single-column unique key, it can be chosen as an SQL DI type key.
- For now, only change the **LOAN_SEQ_NUM** column to data type **Key**. When processing an AI query that includes a column with the `key` type, SQL DI evaluates the affected rows in their entirety and effectively compares all the values in one row to those in another, not just the values in the column of the `key` type.

- **Select the check box** next to “column name” column header shown in the table. All columns of the table should be auto-selected.
- Once **all columns are selected** and **LOAN_SEQ_NUM** has been set to SQL DI data type Key, click the **Next** button

Enable AI query

ZDW10.FMACTB_ORIG1

Enabling an object for AI query requires column configuration and model training. To enable ZDW10.FMACTB_ORIG1 for AI query, select one or more columns and assign [import](#) the column configuration of the AI object from a JSON file. Make sure the columns defined in the file are consistent with those in ZDW10.FMACTB_ORIG1.



Next displayed is the **Column filter** page. This page allows you to exclude data known not to be valid for analysis such as null or missing values. This can be done globally for all columns, or on a per-column basis as suggested by the dialog's options.

The **FMACTB_ORIG1** table has been set up to include only valid data, so you do not have to specify any filter.

- **Click the enable button** to proceed.

Enable AI query

ZDW10.FMACTB_ORIG1

Enabling an object for AI query requires column configuration and model training. To enable ZDW10.FMACTB_ORIG1 for AI query, select one or more columns and assign [import](#) the column configuration of the AI object from a JSON file. Make sure the columns defined in the file are consistent with those in ZDW10.FMACTB_ORIG1.

Column configuration
Select columns and assign SQL DI data types

Column filter
Exclude records containing the filter values

Global filter values ⓘ

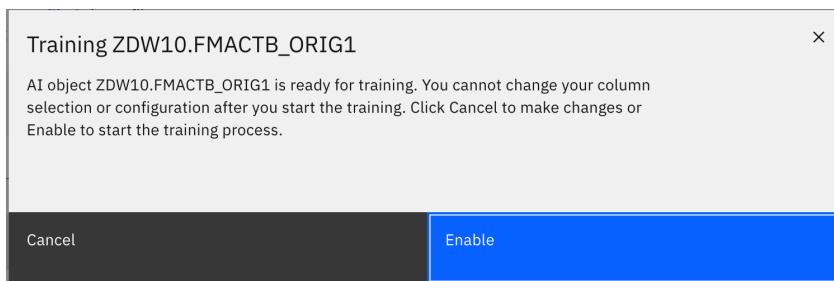
Add +

Column name	Column-specific filter values ⓘ
FIRST_PAYMENT	Enter values separated by semicolon
FIRST_TIME_HOMEBUYER_FLAG	Enter values separated by semicolon
MATURITY_DATE	Enter values separated by semicolon
OCCC_STATUS	Enter values separated by semicolon
CHANNEL	Enter values separated by semicolon

Back Enable

You are presented with one last point of interaction where you can choose to proceed or cancel the operation.

- Click the **enable** button



5. Monitor AI object training

After several seconds on a “spin/wait” cursor being displayed in grayed out view of your window, you should be taken back at the main list of AI Objects for the database that you are connected to.

You should see a green status box toward the top of your browser window indicating that your AI query is being enabled for your table.

 **Enabling AI query.** AI query is being enabled for "FMACTB_ORIG1". Expand the object row to view progress.

- **Click** the X close symbol on the right side of the status box to close it.



Your table should be listed in the table. Its status should be **Enabling** with a spin-wait cursor to the left of the status.

- Every 30 seconds or so, click the “refresh” button on your browser since your object training status does not update on its own.

This training should take at least 1.5 minutes. Expect this to elongate if there are multiple people training their model at the same time.

AI objects

DBW1				
<input type="text" value="zdw10"/> X Add object + Run query				
Name	Schema	Type	Status	Last updated
▼ FMACTB_ORIG1	ZDW10	View	⌚ Enabling	Aug 31, 2022 3:57 PM
Resources per page: 10 1–1 of 1 items				
1 of 1 page ▶				

- Once your table’s AI object changes to Enabled status, you may proceed to the next section. If it does not change after a couple of minutes, or changes to a state other than enabled, contact a lab administrator.

AI objects

DBW1				
<input type="text" value="zdw10"/> X Add object + Run query				
Name	Schema	Type	Status	Last updated
▼ FMACTB_ORIG1	ZDW10	View	✅ Enabled	Aug 31, 2022 3:59 PM
Resources per page: 10 1–1 of 1 items				
1 of 1 page ▶				

6. Review the training results and interpretability

Now that your AI object has been trained, you can review some of the details associated with the model and the table data it has analyzed. This helps you understand the nature of the data that was used for model training and how the model will interact with it.

View the model

- From the list of AI objects, identify your model <schema>.FMACTB_ORIG1. Click on the action menu on the right and select **View model**.

AI objects

The screenshot shows a list of AI objects under the schema DBW1. A search bar at the top contains 'zdw10'. The table has columns: Name, Schema, Type, Status, and Last updated. One row is visible: FMACTB_ORIG1, ZDW10, View, Enabled, Aug 31, 2022 3:59 PM. Below the table are pagination controls (Resources per page: 10, 1-1 of 1 items) and a dropdown menu with options: Disable AI query, Analyze data, View model (which is highlighted with a blue border), Remove, and Export column co.

The **Model details** tab is displayed, showing summary information about the model status, when it was trained, and by whom.

Model details

ZDW10.FMACTB_ORIG1

Training history Cluster center

Name	Status	Start time	End time	Created by
FMACTB_ORIG1	Completed	Aug 31, 2022 3:57 PM	Aug 31, 2022 3:59 PM	ZDW10

- Click on the **Cluster minimums** tab of the **Model details** view.

This shows the centroids for the numeric columns of the table that was included in the training.

A centroid is a vector representing the means for the variables for the observations in the cluster.

Model details

ZDW10.FMACTB_ORIG1

The screenshot shows a table titled "Cluster minimums" with the following data:

Column name	# of clusters	Minimum values
CREDIT_SCORE	5	605 728 769 792 9999
MI_PERT	3	0 6 12
MSA_CODE	3	-7.2e+75 10420 30700
NUM_BORROWERS	3	1 2 3
NUM_UNITS	2	1 2
ORIG_CLTV	4	11 53 68 79
ORIG_DTI_RATIO	5	2 10 23 31 39
ORIG_INTEREST_RATE	5	1.875 2.5 2.75 2.875 3

- When finished looking at the model details, click on the AI/Objects link towards the top of the page.

SQL Data Insights

[Connections](#) / [AI objects](#) /

Model details

You should be taken back to the AI Objects list.

- Analyze the data for the AI object you have trained
- Click on the action menu for your AI object and select the **Analyze data** option.

AI objects

The Analyze data page appears. The first of the 4 tabs on this page displays the details of each column, showing its Db2 data type and SQL DI data type.

Analyze data

ZDW10.FMACTB_ORIG1 Last updated: Nov 23, 2023 4:09 PM ⏲

Object details	Data statistics	Column influence	Column discriminator
Total rows: 988			
Column name	Db2 data type	SQL DI data type	
NUM_UNITS	INTEGER	Numeric	
INTEREST_ONLY_INDICATOR	CHAR	Categorical	
PROPERTY_VALUATION_METHOD	INTEGER	Numeric	
HARP_INDICATOR	CHAR	Categorical	
PROGRAM_INDICATOR	CHAR	Categorical	
PREHARP_LOAN_SEQ_NUM	VARCHAR	Categorical	
SUPER_CONFORMING_FLAG	CHAR	Categorical	
SERVICER_NAME	VARCHAR	Categorical	
SPILED_NAME	VARCHAR	Categorical	

- Click on the **Data statistics** tab above the table. This view provides some common statistics for each of the columns that were trained, providing a better understanding of the nature of the table data.

Analyze data

ZDW10.FMACTB_ORIG1 Last updated: Nov 23, 2023 4:09 PM ⏲

Object details	Data statistics	Column influence	Column discriminator
CREDIT_SCORE	INTEGER	182	792
ORIG_LTV	INTEGER	81	80
ORIG_CLTV	INTEGER	81	80
POSTAL_CODE	INTEGER	275	81200
ORIG_DTI_RATIO	INTEGER	48	37
MSA_CODE	INTEGER	144	16984
ORIG_LOAN_TERM	INTEGER	10	360
ORIG_UPB	INTEGER	349	160000
ORIG_INTEREST_RATE	FLOAT	22	0.07500

- Click on the **Column influence** tab

For each column that was included in the training, the graph shows the Influence score.

Influence

The Influence score represents data presence. The more data that is missing from a column, the closer to 0 a column's score approaches. You can see from the graph below that the FMACTB_ORIG1 table doesn't have any/much missing data except for the column HARP_INDICATOR for instance; additionnaly the influence scores are very similar across the columns, as each column seems to have equal influence.

Analyze data



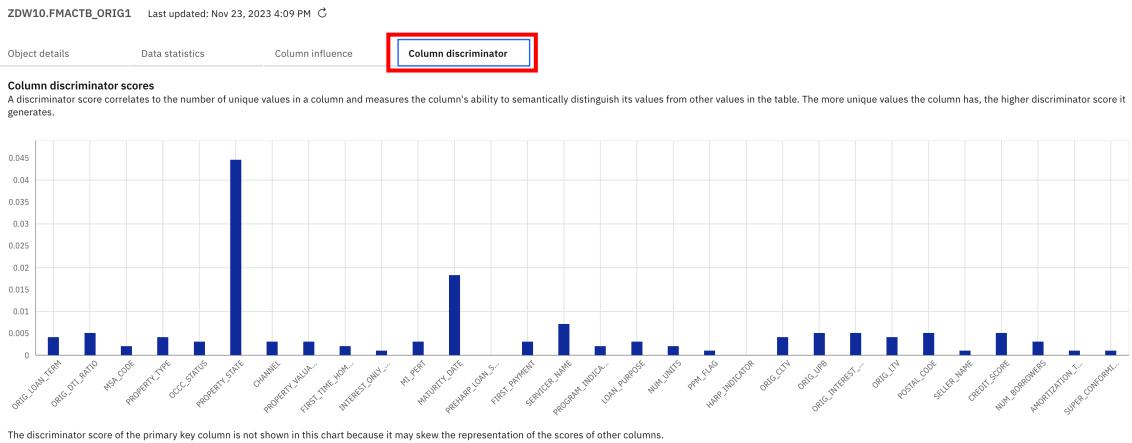
The influence plot can be hidden to allow the graph to show more details for the Discriminator values for each column.

- Click on the **Column Discriminator** tab.

The graph re-scales the y-axis to bring out better detail of the Discriminator scores.

The columns with higher scores indicate they have a greater variation in the values for that column. Columns with greater variation of values are likely to better contribute to similarity query calculations. This helps show which columns may be more important than others for semantic queries.

Analyze data



This concludes the review of the training results.

- Click on the **AI Objects** link towards the top of the page to go back to the AI Objects list view.

Connections / **AI objects** /

Analyze data

ZDW10.FMACTB_ORIG1 Last updated: Aug 31, 2022 5:40 PM ⏺

Object details Data statistics **Column influence**

7. Enable AI on a Db2 for z/OS View

In the previous step, you selected a table and created an AI object which involved a training of a model against the selected table data. Now we are going to use a Db2 for z/OS view which is a join of 2 tables. Then we are going to “enable” for the semantic queries against this view supported by SQL DI.

- In the AI objects list view, Click on the **Add object** button at the top-right of the AI Objects list to select the view to be enabled for AI.

AI objects

DBW1				
<input type="text" value="zdw10"/> x Add object + Run query ⟳				
Name	Schema	Type	Status	Last updated
✓ FMACTB_ORIG1	ZDW10	View	Enabled	Aug 31, 2022 4:39 PM
Resources per page: 10 1–1 of 1 items				

- You will be presented with the **Add object** view. From it, click on the **schema** drop-down list. In the list, scroll down to your assigned <schema> name from your worksheet. Click the **checkbox** for your schema name, and then click outside of the list box to close it.

Add object

DBW1

Choose one or more schemas to list associated Db2 objects. From the list, select the Db2 objects to add as new AI objects or enable for AI query.

Schema: 1 × Select schema ^

Name	Schema	Last activity
ZDW10	ZDW10	
ZDW11		
ZDW12		
ZDW13		
ZDW14		

No object available. Select one or more schemas to list available objects.

Items per page: 10 ▾ 0–0 of 0 items

1 ▾ of 1 page ◀ ▶

Your user's FreddieMac tables and view should be listed in the displayed page. You should notice that the row for your **FMACTB_ORIG1** table seems disabled (the checkbox in the name column is gray and not selectable). As we already enabled AI on this table, this is normal.

For this lab, we have created for you in Db2 for z/OS a view called **FMAC_DEFAULT_RISK_MIN_VIEW1** which joins 2 tables on the LOAN_SEQ_NUM column. The 2 tables are FMACTB_ORIG1 (that you used already) and FMACTB_PERF1 which contains performance data from customer loans.

- Preview the data of the view **FMAC_DEFAULT_RISK_MIN_VIEW1** created for you to get an idea of the type of data in the table:
- Click on the information icon ⓘ located on right side of the row for your table **FMAC_DEFAULT_RISK_MIN_VIEW1**

Add object

DBW1

Choose one or more schemas to list associated Db2 objects. From the list, select the Db2 objects to add as new AI objects or enable for AI query.

Schema: 1 Select schema

Name	Schema	Last activity	
<input type="checkbox"/> FMACTB_ORIG1	ZDW10	Aug 31, 2022 4:39 PM	<input type="button" value="i"/>
<input type="checkbox"/> FMACTB_PERF1	ZDW10		<input type="button" value="i"/>
<input type="checkbox"/> FMAC_DEFAULT_RISK_MIN_VIEW1	ZDW10		<input type="button" value="i"/>

Items per page: 10 1–3 of 3 items | 1 of 1 page |

A popup window should appear with an indication of the number of records in the view as well as a sample of rows returned for the view/table, helping you to understand the type of data in the table. If instead, you get an error trying to access your **FMAC_DEFAULT_RISK_MIN_VIEW1** table, contact a lab administrator.

Table view

FMAC_DEFAULT_RISK_MIN_VIEW1

Total rows: 7725

ORIG_LOAN_TERM	ORIG_DTI_RATIO	MODIFICATION_FLAG	MSA_CODE	PROPERTY_TYPE
360	39		null	SF
360	39		null	SF
360	39		null	SF
360	39		null	SF
360	39		null	SF
360	39		null	SF
360	39		null	SF
360	39		null	SF
180	41		35660	SF
180	41		35660	SF
180	41		35660	SF
180	41		35660	SF

- If you scroll to the right of the table view on your screen you can find out for example **the CURRENT_LOAN_DLQCY_STATUS** column which indicates if the customer already defaulted on their loans payments.
- Click** the close button on the top right of the pop-up window once you are finished previewing the data.
- Select** the checkbox to the left of the Name column for your view.

- Click the **Enable AI query** button on the bottom-right of the screen, which should have become enabled once you selected your **FMAC_DEFAULT_RISK_MIN_VIEW1** table.
- Add object

DBW1

Choose one or more schemas to list associated Db2 objects. From the list, select the Db2 objects to add as new AI objects or enable for AI query.

Schema: 1 × Select schema ▾ Find table/view

Name	Schema	Last activity
<input type="checkbox"/> FMACTB_ORIG1	ZDW10	Aug 31, 2022 4:39 PM ⓘ
<input type="checkbox"/> FMACTB_PERF1	ZDW10	ⓘ
<input checked="" type="checkbox"/> FMAC_DEFAULT_RISK_MIN_VIEW1	ZDW10	ⓘ

Items per page: 10 ▾ 1–3 of 3 items 1 ▾ of 1 page ⏪ ⏩

Cancel Add object Enable AI query

The column configuration dialog should be displayed next.

- Select and configure the columns to be analyzed (shown on the next page)

The next page shows the Column configuration page. This allows you to select the columns from the view to be used in the modeling as well as adjust the data-types that will be used for each column.

For this exercise, we will use all columns in the view.

At this point, you have the option to re-categorize your columns to a different SQL DI data type. However, you have to understand the data to know what SQL DI data types are valid for that given column.

- Like in previous chapter, only change the **LOAN_SEQ_NUM** column to data type **Key**. When processing an AI query that includes a column with the `key` type, SQL DI evaluates the affected rows in their entirety and effectively compares all the values in one row to those in another, not just the values in the column of the `key` type.
- Select the **check box** next to “column name” column header shown in the table. All columns of the table should be auto-selected.
- Once **all columns are selected** and **LOAN_SEQ_NUM** has been set to SQL DI data type **Key**, click the **Next** button

Enable AI query

ZDW10.FMAC_DEFAULT_RISK_MIN_VIEW1

Enabling an object for AI query requires column configuration and model training. To enable ZDW10.FMAC_DEFAULT_RISK_MIN_VIEW1 for AI query, select one [import](#) the column configuration of the AI object from a JSON file. Make sure the columns defined in the file are consistent with those in ZDW10.FMAC_DEFAULT

Column configuration Column filter
Select columns and assign SQL DI data types Exclude records containing the filter values

Select columns for training

44 items selected | Cancel

Column name	SQL DI data type
<input checked="" type="checkbox"/> LOAN_SEQ_NUM	Key
<input checked="" type="checkbox"/> CREDIT_SCORE	Numeric
<input checked="" type="checkbox"/> FIRST_PAYMENT	Categorical
<input checked="" type="checkbox"/> FIRST_TIME_HOMEBUYER_FLAG	Categorical
<input checked="" type="checkbox"/> MATURITY_DATE	Categorical

Cancel | Next

Next displayed is the **Column filter** page. This page allows you to exclude data known not to be valid for analysis such as null or missing values. This can be done globally for all columns, or on a per-column basis as suggested by the dialog's options.

The **FMAC_DEFAULT_RISK_MIN_VIEW1** view has been set up to include only valid data, so you do not have to specify any filter.

- Click the **enable** button to proceed.

Enable AI query

ZDW10.FMAC_DEFAULT_RISK_MIN_VIEW1

Enabling an object for AI query requires column configuration and model training. To enable ZDW10.FMAC_DEFAULT_RISK_MIN_VIEW1 for AI query, select one [import](#) the column configuration of the AI object from a JSON file. Make sure the columns defined in the file are consistent with those in ZDW10.FMAC_DEFAULT

Column configuration Column filter
Select columns and assign SQL DI data types Exclude records containing the filter values

Global filter values [①](#)

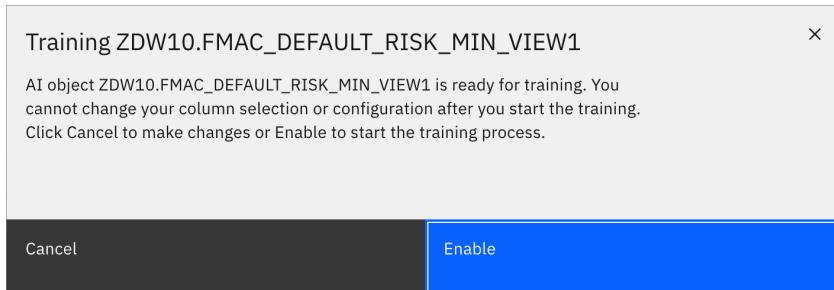
Add +

Column name	Column-specific filter values ①
LOAN_SEQ_NUM	Enter values separated by semic
FIRST_PAYMENT	Enter values separated by semic
FIRST_TIME_HOMEBUYER_FLAG	Enter values separated by semic
MATURITY_DATE	Enter values separated by semic
OCCC_STATUS	Enter values separated by semic

Back | Enable

You are presented with one last point of interaction where you can choose to proceed or cancel the operation.

- **Click the enable button**



8. Monitor AI object training

After several seconds on a “spin/wait” cursor being displayed in grayed out view of your window, you should be taken back at the main list of AI Objects for the database that you are connected to.

You should see a green status box toward the top of your browser window indicating that your AI query is being enabled for your view.

Enabling AI query. AI query is being enabled for "FMAC_DEFAULT_RISK_MIN_VIEW1". Expand the object row to view progress.

- **Click the X close symbol on the right side of the status box to close it.**



Your view should be listed in the table. Its status should be **Enabling** with a spin-wait cursor to the left of the status.

- Every 30 seconds or so, click the “refresh” button on your browser since your object training status does not update on its own.

This training should take at least 1.5 minutes. Expect this to elongate if there are multiple people training their model at the same time.

DBW1				
Search: zdw10				X
Name	Schema	Type	Status	Last updated
✓ FMACTB_ORIG1	ZDW10	View	Enabled	Aug 31, 2022 4:39 PM
✗ FMAC_DEFAULT_RISK_MIN_VIEW1	ZDW10	View	Enabling	Sep 1, 2022 5:43 PM

Once your view's AI object changes to Enabled status, you may proceed to the next section. If it does not change after a couple of minutes, or changes to a state other than enabled, contact a lab administrator.

DBW1					
zdw10				Add object	Run query
Name	Schema	Type	Status	Last updated	
FMACTB_ORIG1	ZDW10	View	Enabled	Aug 31, 2022 4:39 PM	⋮
FMAC_DEFAULT_RISK_MIN_VIEW1	ZDW10	View	Enabled	Sep 1, 2022 5:45 PM	⋮

Resources per page: 10 ▾ 1–2 of 2 items 1 ▾ of 1 page ⏪ ⏩

You can see here that we are going to enable queries not only for tables but also on views created on Db2 for z/OS. This way you can enrich data you are going to enable for AI queries but located on different Db2 for z/OS table.

4.0 Run semantic queries

The following is a description of the different types of semantic queries SQL DI supports.

Understanding how each type of query works and how it can help you take advantage of the relationships in your data is crucial.

For reference as you work through the queries further below, refer back to these descriptions as well as the discussion in section one of this lab.

Semantic similarity:

A similarity query identifies groups of similar records or entities in records. Consider selecting semantic similarity if your query intends to identify the similarities of customer characteristics and behaviors in industries, such as commerce, finance, and insurance.

Semantic dissimilarity:

A dissimilarity query finds the outliers from the norm in records. Consider selecting semantic dissimilarity if your query intends to detect operational anomalies, fraudulent activities, and other patterns of deviation.

Semantic clustering:

A clustering query forms a cluster of entities in records and evaluates whether an additional entity belongs in the cluster. Consider selecting semantic clustering if your query intends to examine similarities or dissimilarities across multiple entities in a broader context.

Semantic analogy:

An analogy query determines if the relationship between two entities applies to that of a second pair of entities. Consider selecting semantic analogy if your query intends to discover your customers' preference for a specific product and the degree of their affinity for other products.

- Bring up the query tool to explore the semantic queries
- From the AI objects list, **click** on the **Run query** button at the top-right of the page.

AI objects

The screenshot shows a user interface for managing AI objects. At the top, there's a search bar containing the text "zdw10". To the right of the search bar are buttons for "Add object" and "Run query". The "Run query" button is highlighted with a red box. Below the search bar is a table with columns: Name, Schema, Type, Status, and Last updated. There is one row visible in the table, showing "FMACTB_ORIG1" under Name, "ZDW10" under Schema, "View" under Type, "Enabled" under Status, and "Aug 31, 2022 4:39 PM" under Last updated. At the bottom of the table, there are pagination controls: "Resources per page: 10", "1–1 of 1 items", "1 of 1 page", and navigation arrows.

Some notes on the Run query page that appears (shown below):

The run query tool allows you to explore different semantic queries. By clicking on the **Add SQL +** tab at the top right of the SQL edit box, it creates another query tab where you can enter another query (up to a max of 10 tabs)

Clicking on the Query type drop-down at the top of the page will populate your query edit box with a default query of the selected type. This is optional as you can type or paste in the query that you want to run.



Note that before you run a query, check to make sure the schema (circled in red below) is set to your <schema> value provided in your worksheet. If the schema is not set properly for your userid, the query will fail, with an error displayed below the **Run** button

```
SELECT * FROM
  (SELECT C.*,
    AI_SIMILARITY(LOAN_SEQ_NUM, 'F21Q11276092') AS SIMILARITY
    FROM <schema>.FMACTB_ORIG1 C
    WHERE LOAN_SEQ_NUM <> 'F21Q11276092')
  WHERE SIMILARITY > 0.3
  ORDER BY SIMILARITY DESC
  FETCH FIRST 20 ROWS ONLY;
```

- Run a semantic similarity query to get accustomed to the query tool
- Click the **SQL-1** tab and then click the **Query type** drop-down list and select **Semantic similarity**.

The SQL edit box should be populated with the statement shown in the image below.

- Make sure to replace the SQL text with the following entry:

```
SELECT * FROM
  (SELECT C.*,
    AI_SIMILARITY(LOAN_SEQ_NUM, 'F21Q11276092') AS SIMILARITY
    FROM <schema>.FMACTB_ORIG1 C
    WHERE LOAN_SEQ_NUM <> 'F21Q11276092')
  WHERE SIMILARITY > 0.3
  ORDER BY SIMILARITY DESC
  FETCH FIRST 20 ROWS ONLY;
```



Be careful with pasting the queries into the SQL tool. Sometimes extended code point characters, such as “smart quotes” are introduced, which will cause SQL errors.

- Make sure to change the **schema** to the <schema> value in your worksheet!

In this example, consider a problem where Jane, a business analyst at a large banking institution, identifies a customer with a current loan with default of payment., and she wants to find out whether other customer's loan share similar characteristics. If she can identify such customer's loans, she might be able to take some preventive actions that forestall the customers from defaulting from paying their loan.

Assume that the loan sequence number who defaulted has LOAN_SEQ_NUM 'F21Q11276092'. The example below shows the SQL that the analyst can issue to find other similar customers.

- Click the Run button.

Run query

Choose a query type to populate the query editor and then edit and run the query.

The screenshot shows a query editor interface. At the top, there's a dropdown menu labeled "Query type (optional)" with "Semantic similarity" selected. Below it, a tab bar has "SQL-1" selected, followed by tabs for "SQL-2", "SQL-3", and "SQL-4". The main area contains the following SQL code:

```

SELECT * FROM
  (SELECT C.*, AI_SIMILARITY(LOAN_SEQ_NUM, 'F21Q11276092') AS SIMILARITY
   FROM ZDW10.FMACTB_ORIG1 C
   WHERE LOAN_SEQ_NUM <> 'F21Q11276092')
  WHERE SIMILARITY > 0.5
 ORDER BY SIMILARITY DESC
  FETCH FIRST 20 ROWS ONLY;

```

At the bottom, there are two buttons: "Clear" and "Run", with "Run" being highlighted in blue.

- Scroll your window down to show the content below the Run button. You should see a results table listing loans similar to the loan identified as 'F21Q11276092'. The values for the categorical columns are mostly the same or similar. If you scroll to the far right of the table, you can see the similarity score generated for the retrieved customer rows. Notice that the most similar result is listed at the top, as described by our query.

If you instead have an SQL error message and the problem is not immediately apparent to you, contact a lab instructor.

- It is worth noting that you can export the result set data to a CSV file by clicking on the up arrow at the top-left of the window, next to the **Result set** label

CREDIT_SCORE	FIRST_PAYMENT	FIRST_TIME_HOMEBUYER_FLAG	MATURITY_DATE	MSA_CODE	MI_PERT	NUM_UNITS	OCCC_STATUS	ORIG_CLTV	ORIG_DTT_RATIO	ORIG_UPB
749	2021-03-01	Y	2051-02-01	null	35	1	P	97	15	67000
680	2021-03-01	N	2051-02-01	null	25	1	P	84	49	90000
722	2021-03-01	N	2046-02-01	null	12	1	P	82	44	235000

Now that you are familiar with running a query in the tool and examining the results, you can explore the queries in the next step

1. Explore other semantic queries

In this section, you can enter the queries provided below. These queries should have also been made available to you (probably as a file on your desktop) so that you can copy and paste them into the query tool.

If you have not been told where the file is, make sure to ask your lab instructor.



Be careful with pasting the queries into the SQL tool. Sometimes extended code point characters, such as “smart quotes” are introduced, which will cause SQL errors.

- SQL-1 AI_SIMILARITY -- Find loans who are the most similar to the loan sequence number with LOAN_SEQ_NUM 'F21Q11276092' that was known to have defaulted their payment and including LOAN_SEQ_NUM 'F21Q11276092'. This particular query is targeting a different table now. Note you are now using the Db2 View joining the 2 tables FMACTB_ORIG1 and FMACTB_PERF1. Using this trained view gives us the possibility to enrich the data we used for creating our AI model and thus produces a different result set.

```
select loan_seq_num, similarity from
  (select loan_seq_num,
          avg(AI_SIMILARITY(LOAN_SEQ_NUM, 'F21Q11276092' using model column
LOAN_SEQ_NUM)) as similarity
   from ZDWxx.FMAC_default_risk_min_view1
  -- where loan_seq_num <> 'F21Q11276092'
 group by loan_seq_num ) as temp
 WHERE SIMILARITY > 0.3
 order by similarity desc
fetch first 100000 rows only ;
```

Here you can see that LOAN_SEQ_NUM 'F21Q11276092' has a smiliraty score not exactly equal to 1. This result is normal according to the way the model was created using NLP (Natural Language Processing) and sampling on the data.



We are now going to use only the Db2 view **ZDWxx.FMAC_DEFAULT_RISK_MIN_VIEW1** for the rest of our queries.

- SQL-2 AI_SIMILARITY – Same as above but now find loans who are the most similar to the loan sequence number with LOAN_SEQ_NUM 'F21Q11276092' that was known to have defaulted their payment and excluding LOAN_SEQ_NUM 'F21Q11276092'

```
select loan_seq_num, similarity from
  (select loan_seq_num,
          avg(AI_SIMILARITY(LOAN_SEQ_NUM, 'F21Q11276092' using model column
LOAN_SEQ_NUM)) as similarity
   from ZDWxx.FMAC_default_risk_min_view1
  where loan_seq_num <> 'F21Q11276092'
 group by loan_seq_num ) as temp
 WHERE SIMILARITY > 0.3
 order by similarity desc
fetch first 100000 rows only ;
```

- SQL-3 AI_DISSIMILARITY -- Find loans who are the least similar to the loan sequence number with LOAN_SEQ_NUM 'F21Q11276092' that was known to have defaulted their payment and exclude LOAN_SEQ_NUM 'F21Q11276092'

```
select loan_seq_num, similarity from
  (select loan_seq_num,
          avg(AI_SIMILARITY(LOAN_SEQ_NUM, 'F21Q11276092' using model column
LOAN_SEQ_NUM)) as similarity
   from ZDWxx.FMAC_default_risk_min_view1
  where loan_seq_num <> 'F21Q11276092'
 group by loan_seq_num ) as temp
 WHERE SIMILARITY < 0.3
order by similarity asc
fetch first 100000 rows only ;
```



Here you can see that similarity scores could be negative values. They may range from -1 (very dissimilar) to 1 (very similar). In this particular example we display the most dissimilar LOAN_SEQ_NUM compared to 'F21Q11276092'.

- SQL-4 AI_SIMILARITY -- Use “defaulted” loans and return information on which US state customers are living. The idea here is to try identify any differences based on where people are living and propensity to default based on your property_state.

```
SELECT DISTINCT AI_SIMILARITY( 1      USING MODEL COLUMN
CURRENT_LOAN_DLQCY_STATUS, PROPERTY_STATE), PROPERTY_STATE
FROM ZDWxx.FMAC_DEFAULT_RISK_MIN_VIEW1
ORDER BY 1 DESC ;
```



Here you can see that most of the US states are quite identical. There is a slight differences between OHIO (OH) and WISCONCIN (WI) but nothing important.

- SQL-5 AI_SIMILARITY -- Used “defaulted” and return information on whether customers are FIRST_TIME_HOMEBUYERS or not.

```
SELECT DISTINCT AI_SIMILARITY( 1      USING MODEL COLUMN
CURRENT_LOAN_DLQCY_STATUS, FIRST_TIME_HOMEBUYER_FLAG ),
FIRST_TIME_HOMEBUYER_FLAG
FROM ZDWxx.FMAC_DEFAULT_RISK_MIN_VIEW1
ORDER BY 1 DESC ;
```

Result set

	FIRST_TIME_HOMEBUYER_FLAG
0.08156681060791016	N
0.01002710685133934	Y



Here you can see that most of the customers in our data who defaulted are not first-time homebuyers. Maybe an interesting finding for Jane our bank analyst to find out the reasons why.....

- SQL-6 AI_SEMANTIC_CLUSTER -- Still based on LOAN_SEQ_NUM F21Q11276092, find the cluster of similar loans. In this first query we want a cluster centered on only 1 LOAN_SEQ_NUM.

```
SELECT * FROM
  (SELECT C.* , AI_SEMANTIC_CLUSTER(loan_seq_num, 'F21Q11276092') AS
SIMILARITY
    FROM ZDWxx.FMAC_default_risk_min_view1 C
      WHERE loan_seq_num <> 'F21Q11276092')
      WHERE SIMILARITY > 0.3
      ORDER BY SIMILARITY ASC
      FETCH FIRST 20 ROWS ONLY;
```



Here you can scroll to the result produced by this query and find the CURRENT_LOAN_DLQCY_STATUS column. The values returned for other LOAN_SEQ_NUM are likely to be 0 (not yet defaulted). This type of query is particularly useful to identify cluster of loans with same characteristics and with likelihood of defaulting in the future.

- SQL-7 AI_SEMANTIC_CLUSTER -- Based on SQL-1 results, find the similar customers with the group of other LOAN_SEQ_NUM.

```
SELECT AI_SEMANTIC_CLUSTER(LOAN_SEQ_NUM, 'F21Q11276092',
  'F21Q11276366' , 'F21Q11276333') AS SCORE, X.*
FROM ZDWxx.FMAC_DEFAULT_RISK_MIN_VIEW1 X
ORDER BY SCORE DESC
fetch first 10 rows only ;
```



Here you can notice we return several lines for the same LOAN_SEQ_NUM. Our view joining FMACT_ORIG1 (1 line per loan) and FMACTB_PERF1 (several lines per loan) has indeed multiple lines for the same LOAN_SEQ_NUM.
This time our cluster is using 3 different centers.

- SQL-8 AI_ANALOGY: Examines relationships between Sales channel for the loans and servicer names. This type of query can help identify servicer names most likely to use sales channel 'R' the same way 'US Bank N.A.' uses channel 'C'.

```
SELECT DISTINCT SERVICER_NAME,
          AI_ANALOGY('C' USING MODEL COLUMN CHANNEL,
                     'U.S. BANK N.A.' USING MODEL COLUMN SERVICER_NAME,
                     'R', SERVICER_NAME ) AS ANALOGY_SCORE
FROM ZDWxx.FMAC_DEFAULT_RISK_MIN_VIEW1
GROUP BY SERVICER_NAME
ORDER BY ANALOGY_SCORE DESC ;
```

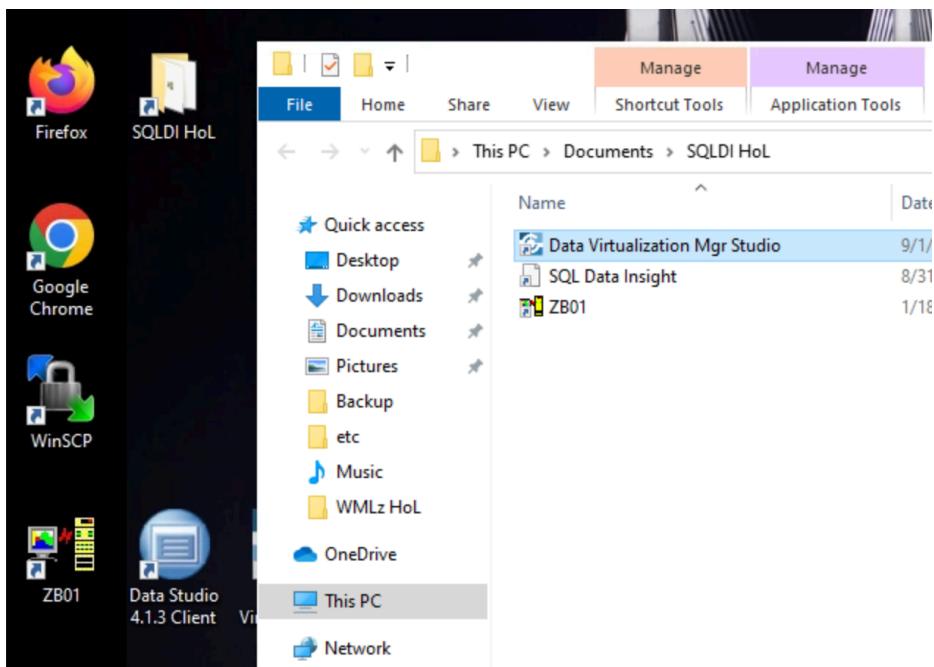
5.0 Enable AI on a Db2 for z/OS User Defined Table Function using DVM (Data Virtualization Manager)

In the previous step, you selected a view and created an AI object which involved a training of a model against the selected view data. Imagine your data is not stored in a Db2 for z/OS table and you want to be able to use this data for AI queries too.

We are going now to use a Data Virtualization Manager (DVM) to create a User Defined Table Function (UDTF) in Db2 for z/OS. This UDTF view will be a join of a Db2 table and a VSAM file. When the UDTF view is created in Db2 for z/OS, we are going to “enable” this UDTF for the semantic queries supported by SQL DI.

1. Accessing Data Virtualization Manager Studio client

- Start the DVM studio client. On the desktop of your lab machine, you should see a folder called “SQLDI HoL”.
- Open that folder. In it you should see a program (link) for “Data Virtualization Mgr Studio”

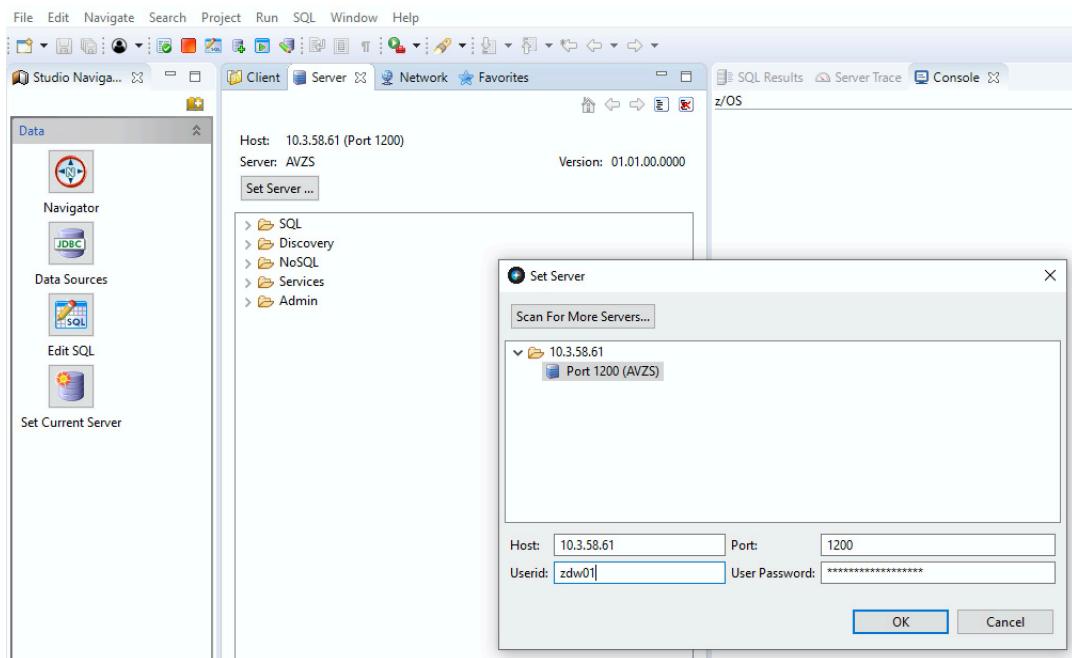


If you do not see the SQDI HoL folder or the SQL Data Insights link in the folder, then refer to your lab instructor for assistance

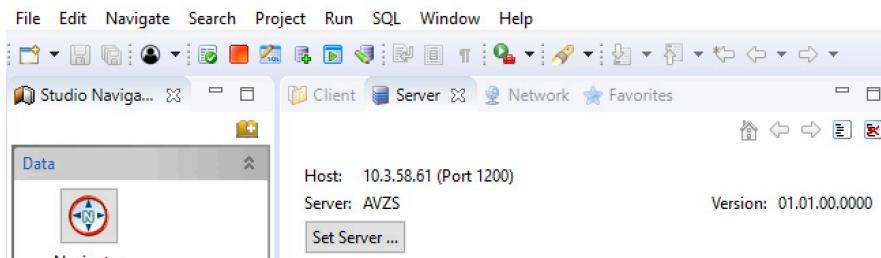


Opening the client will start a program on your environment.

- In the initial screen, Click on the Set Server button and provide your Username and password for ZB01 (10.3.58.61) and port 1200 for the system you are using:



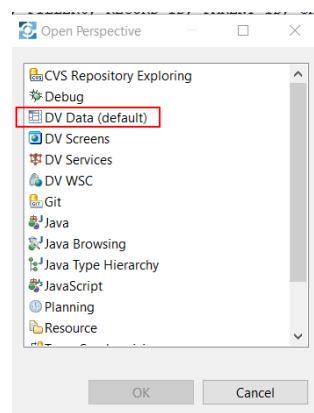
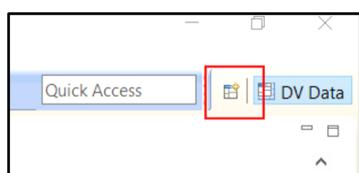
- The window title displays the current perspective:



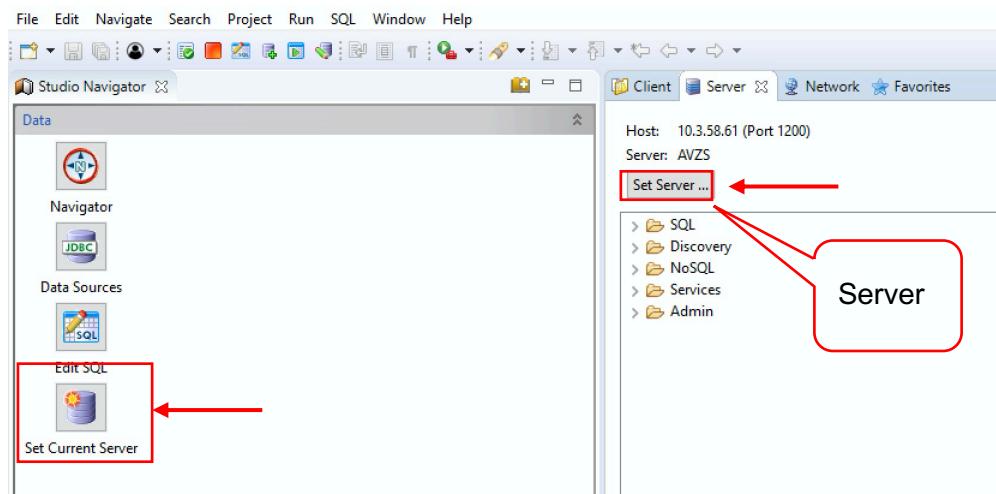
- Confirm that DV Data is active by checking the top right corner:



- If necessary, click to switch to the DV Data perspective as shown below and select **DV data (default)**:



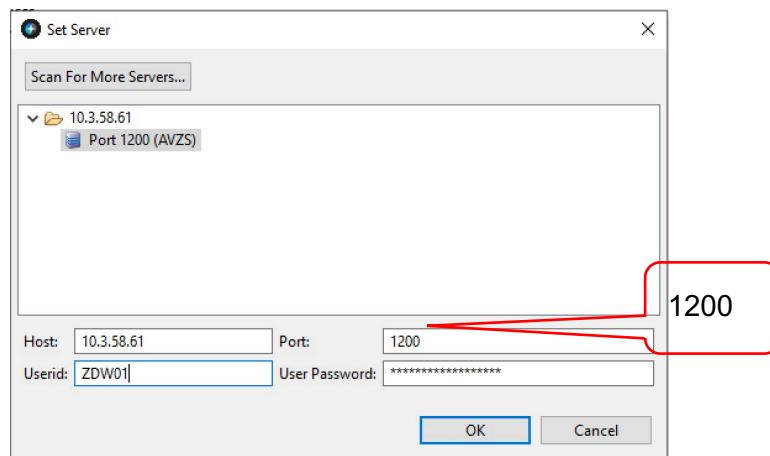
- Click on the **Set Current Server** icon in the leftmost frame, or click on the **Set Server** button on the **Server** tab.



- Specify the host name and port, and your user name and password in the pop-up-dialog box.
 - Connection parameters are:

host name: **zb01**
 or IP address: **10.3.58.61**
 port: **1200**

The User ID and password for ZB01 is supplied to you at the HoL or you can request it from the class administrator.

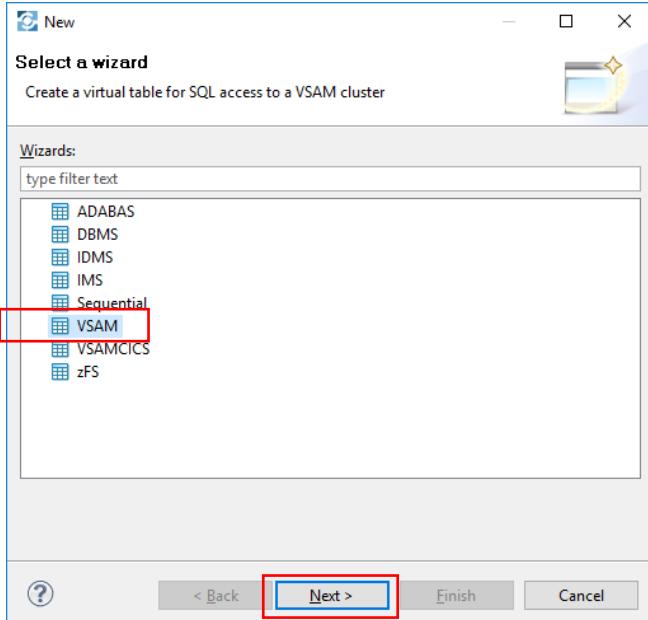


- Populate the fields and hit 'OK'. You should find the connection **10.3.58.61** the next time.

1.1.1. Create a virtual table to access VSAM

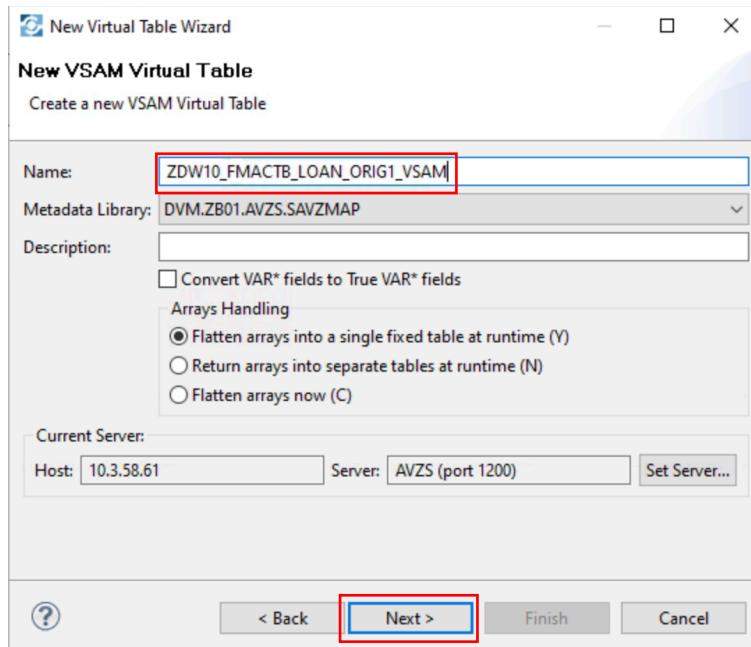
- To see the existing virtual tables on this DVM server:
 - Expand SQL
 - Expand Data

- Expand AVZS (DVM Server)
- Expand Virtual Tables
- To create a new virtual table, Right-click anywhere in the list of displayed tables and choose **Create Virtual Table(s)**.
- Click on **VSAM** in the pop-up-window and then click the **Next** button.



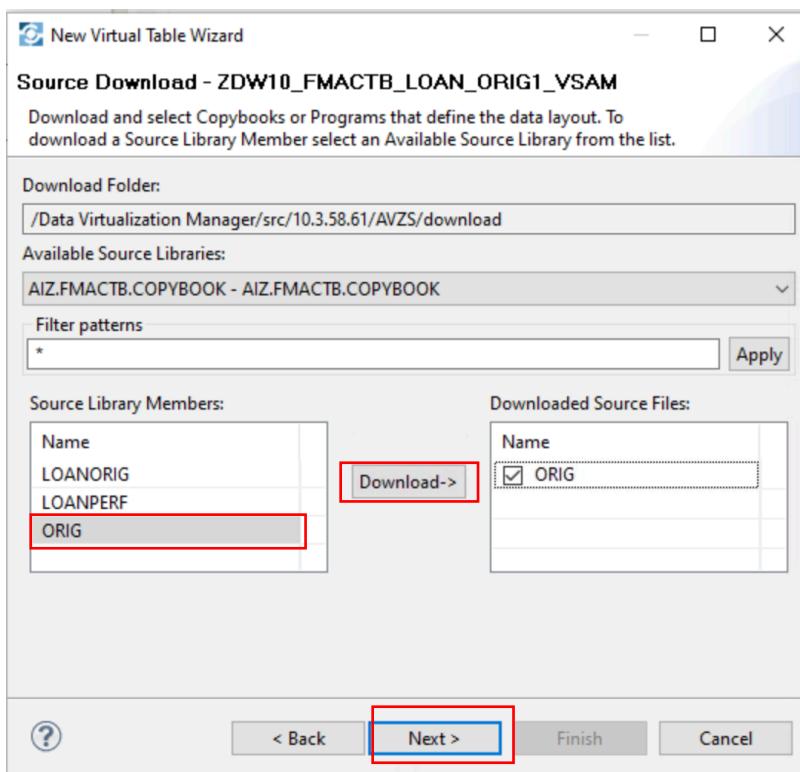
- Enter a name, and a description if desired, for your new virtual table. You should use **your user ID** as the prefix of the virtual table name (i.e. "ZDWxx_FMACTB_LOAN_ORIG1_VSAM"). Then, click **Next** to continue.

Note: In the screen shot, user "ZDW10_" was used. Please use the ZDWxx ID assigned to you.

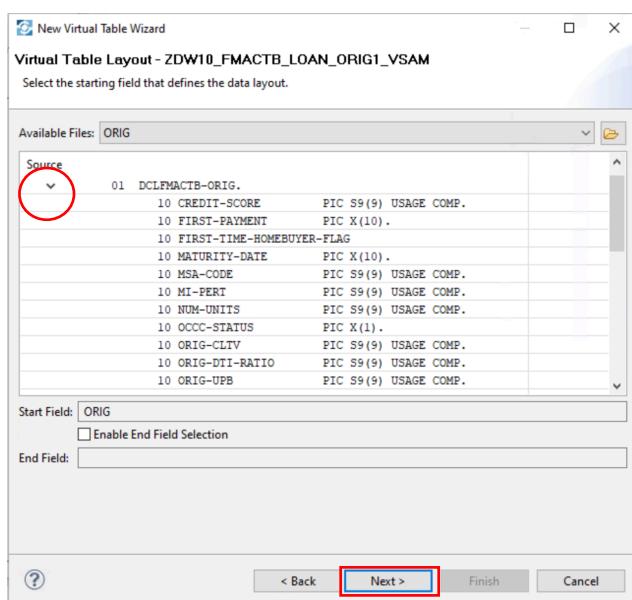


- If there is no value in **Available Source Libraries**, please select the drop-down box at **Available Source Libraries** and select "AIZ.FMACTB.COPYBOOK – AIZ.FMACTB.COPYBOOK".

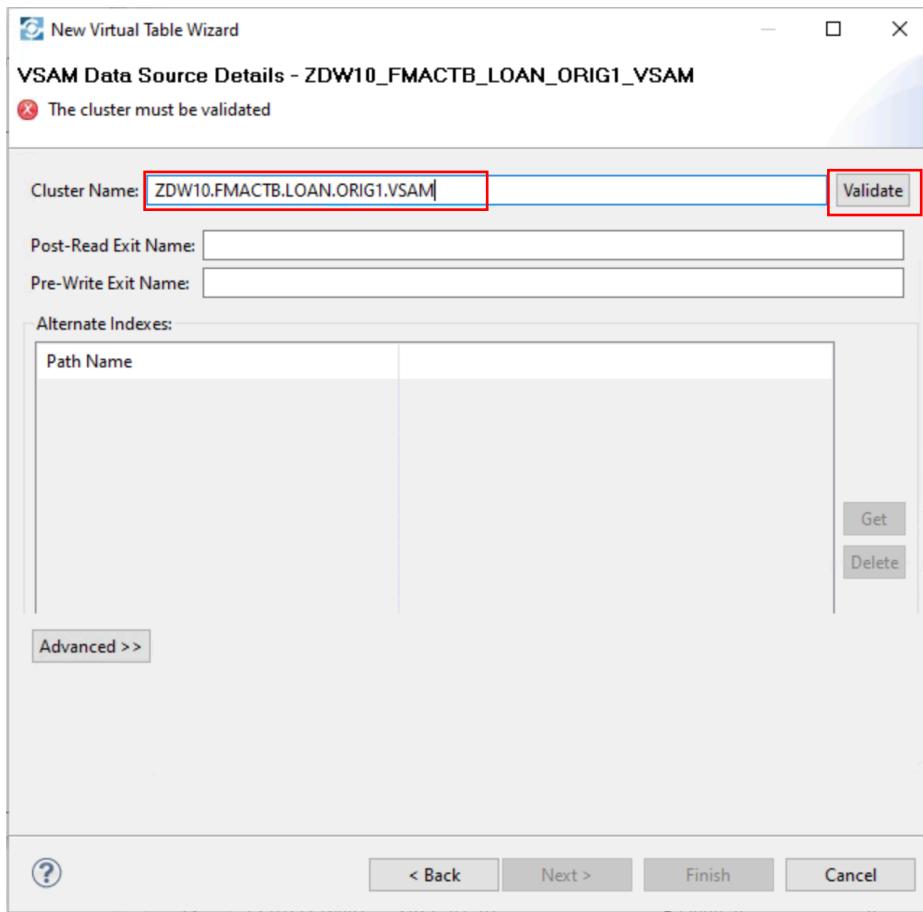
- Specify the copybook member to use. If the member is not yet downloaded, click **ORIG** member on the left, then **Download**.
- If the member is already downloaded then **click on the checkbox** of the **ORIG** member's Name on the right, and then click the **Next** button.



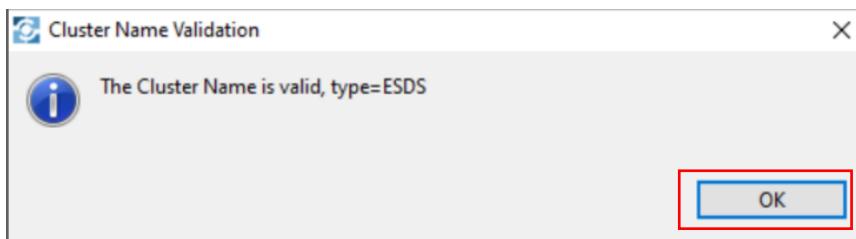
- Click on the ">" to confirm the file layout is correct and then click the **Next** button.



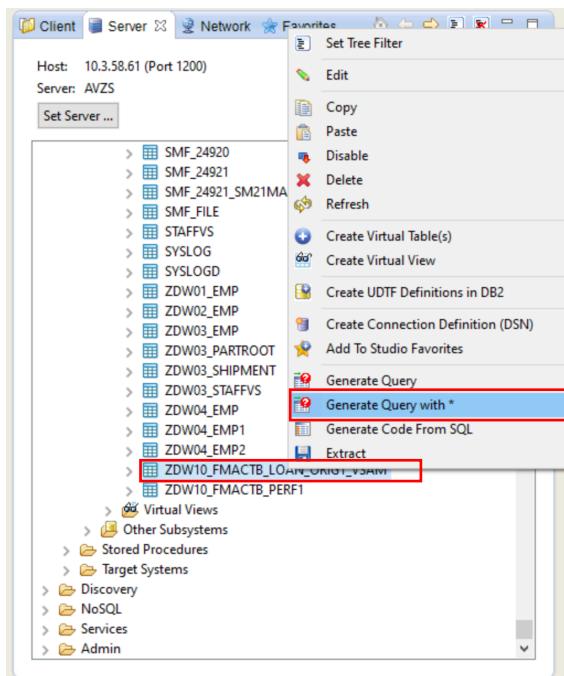
- Specify the data set name for the VSAM cluster. The name to use for this data set is **ZDWxx.FMACTB.LOAN.ORIG1.VSAM**.
- There is a warning indicating that you must validate that the data set exists and is a VSAM file. Click the **Validate** button.



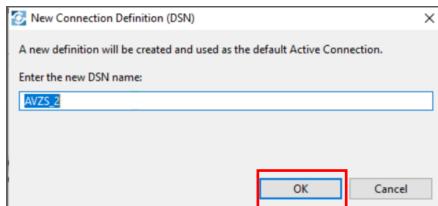
- A pop-up window will inform you that the cluster is valid. Click the **OK** button. Then, click the **Finish** button.



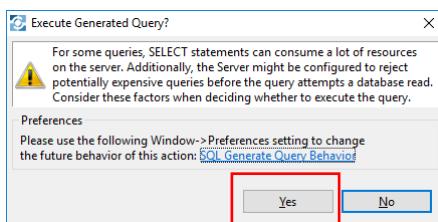
- Click the **Finish** button of the **New Table Wizard** dialogue
- The new virtual table is displayed in the list. Right-click on the table name and choose **Generate Query with ***.



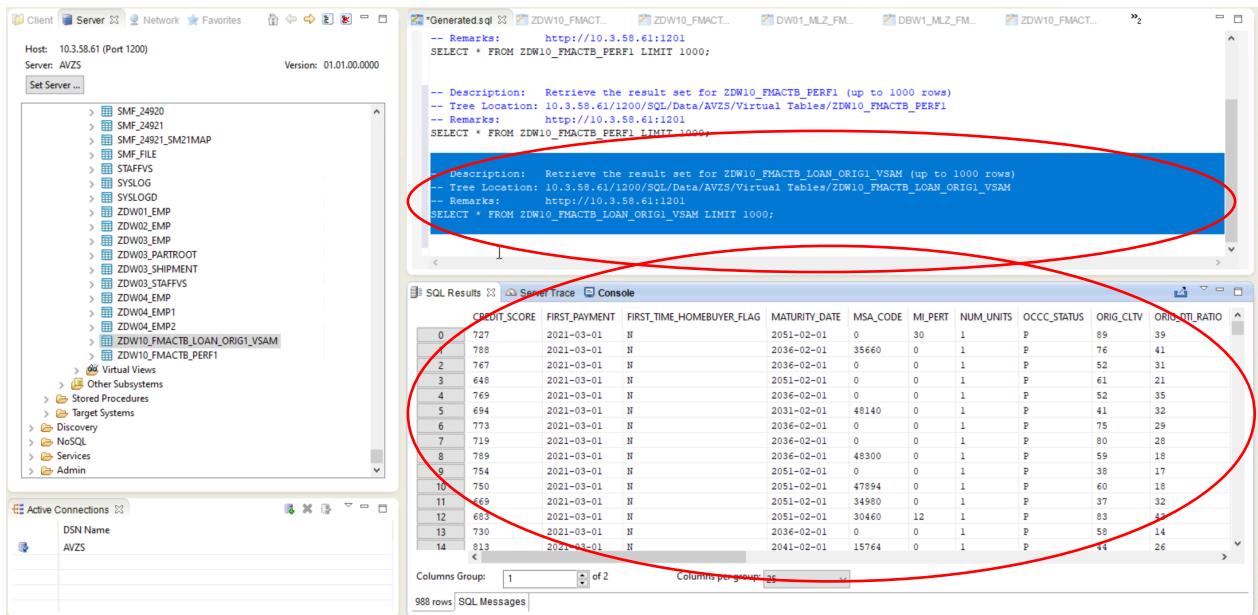
- You may receive the following pop-up message to create a new connection definition. Click **OK** to proceed.



- By default, a warning dialog will open due to the potential consumption of resources on the data source. Click on the **Yes** button to execute the query.

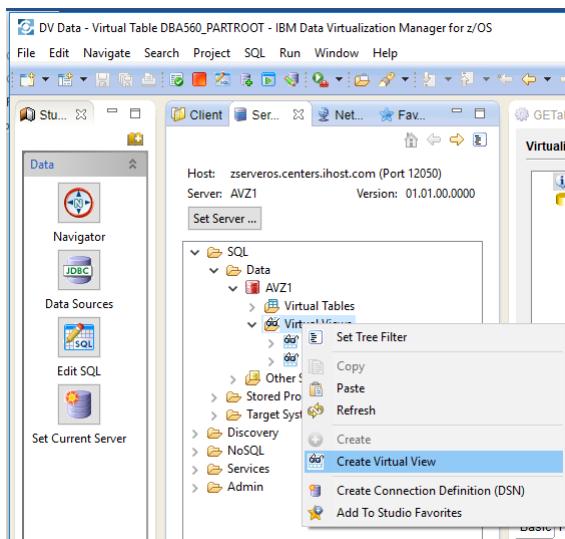


The SQL select statement will be displayed in the editor and the result set is displayed in a window below the editor.



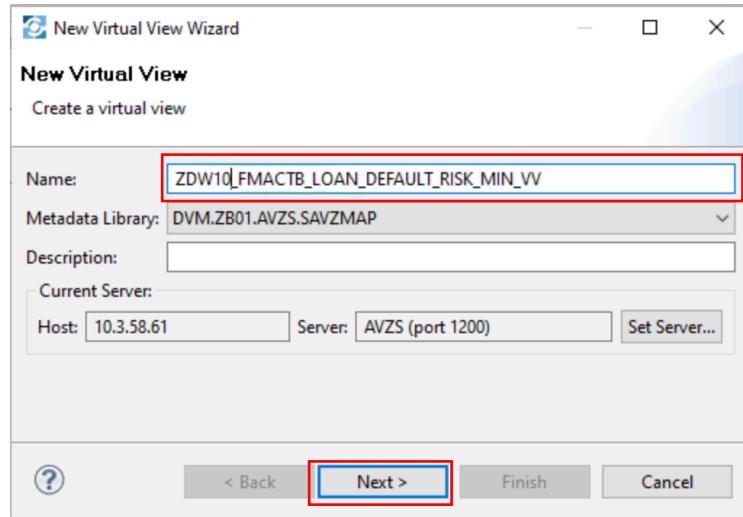
1.1.2. Create a view to join Db2 and VSAM data

- To see the existing definitions of virtual tables on this DVM server:
 - Expand SQL
 - Expand Data
 - Expand AVZS (DVM Server)
 - Expand Virtual Tables
- Look for **ZDWxx_FMACTB_PERF1** virtual table. This virtual table was created for you in advance and connects to Db2 for z/OS.
- To create a new virtual view to join DB2 table and VSAM dataset using the **LOAN_SEQ_NUM** columns, Right-click anywhere in the list of displayed virtual views and choose **Create Virtual View**.

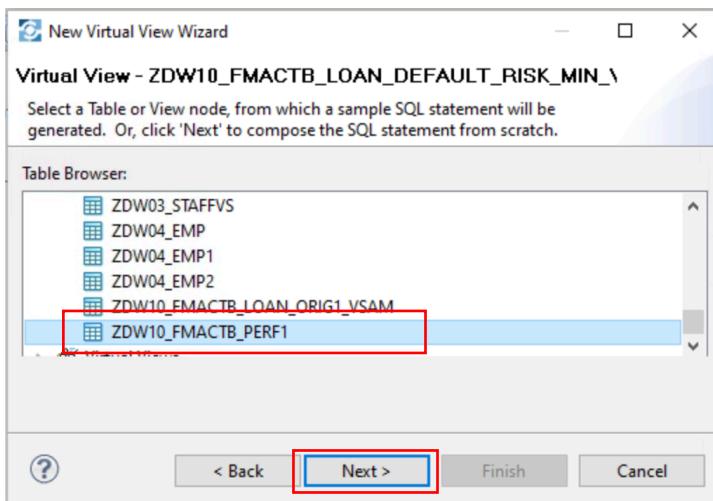


- Enter a name, and a description if desired, for the new virtual view. You should use **your user ID** as the prefix of the virtual view name (i.e. "**ZDWxx_FMACTB_LOAN_DEFAULT_RISK_MIN_VV**".) Then, click **Next** to continue.

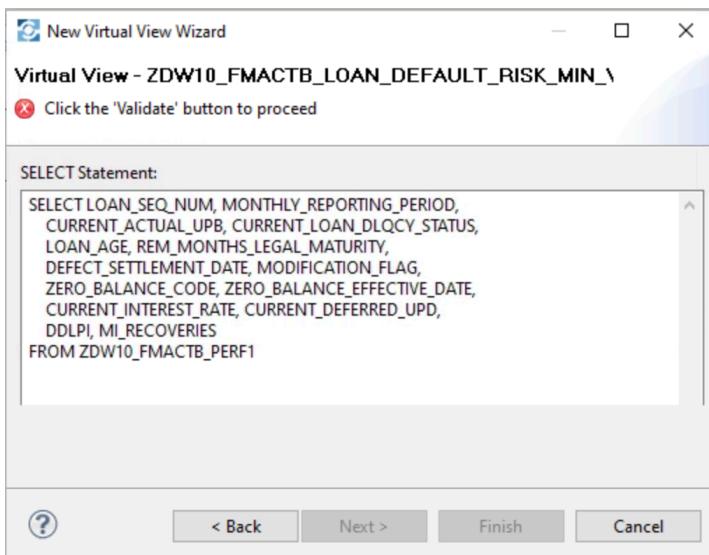
Note: In the screen shot, user "ZDW10_" was used. Please use the ZDWxx ID assigned to you.



- Select the virtual table **ZDWxx_FMACTB_PERF1** to be included in the view. Click the **Next** button.



- A select statement will be generated to retrieve all the data from the virtual table.



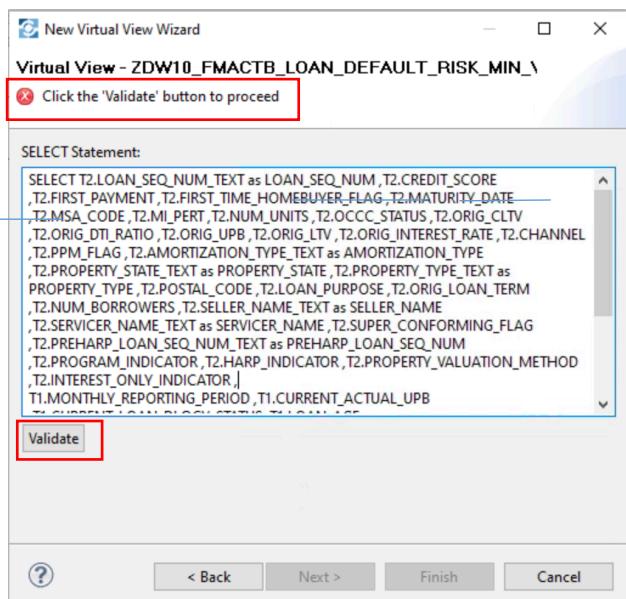
- Modify the statement to include in the join the VSAM virtual table you created with the virtual table in Db2created for you. Use this statement:

```

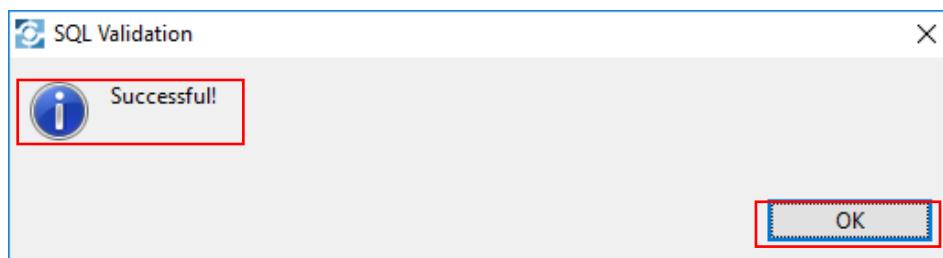
SELECT T2.LOAN_SEQ_NUM_TEXT as LOAN_SEQ_NUM ,T2.CREDIT_SCORE ,T2.FIRST_PAYMENT
,T2.FIRST_TIME_HOMEBUYER_FLAG ,T2.MATURITY_DATE ,T2.MSA_CODE ,T2.MI_PERT
,T2.NUM_UNITS ,T2.OCCC_STATUS ,T2.ORIG_CLTV ,T2.ORIG_DTI_RATIO ,T2.ORIG_UPB
,T2.ORIG_LTV ,T2.ORIG_INTEREST_RATE ,T2.CHANNEL ,T2.PPM_FLAG
,T2.AMORTIZATION_TYPE_TEXT as AMORTIZATION_TYPE ,T2.PROPERTY_STATE_TEXT as
PROPERTY_STATE ,T2.PROPERTY_TYPE_TEXT as PROPERTY_TYPE ,T2.POSTAL_CODE
,T2.LOAN_PURPOSE ,T2.ORIG_LOAN_TERM ,T2.NUM_BORROWERS ,T2.SELLER_NAME_TEXT as
SELLER_NAME ,T2.SERVICER_NAME_TEXT as SERVICER_NAME ,T2.SUPER_CONFORMING_FLAG
,T2.PREHARP_LOAN_SEQ_NUM_TEXT as PREHARP_LOAN_SEQ_NUM ,T2.PROGRAM_INDICATOR
,T2.HARP_INDICATOR ,T2.PROPERTY_VALUATION_METHOD ,T2.INTEREST_ONLY_INDICATOR ,
T1.MONTHLY_REPORTING_PERIOD ,T1.CURRENT_ACTUAL_UPB ,T1.CURRENT_LOAN_DLQCY_STATUS
,T1.LOAN_AGE ,T1.REM_MONTHS_LEGAL_MATURITY ,T1.DEFECT_SETTLEMENT_DATE
,T1.MODIFICATION_FLAG ,T1.ZERO_BALANCE_CODE ,T1.ZERO_BALANCE_EFFECTIVE_DATE
,T1.CURRENT_INTEREST_RATE ,T1.CURRENT_DEFERRED_UPD ,T1.DDLPI ,T1.MI_RECOVERIES
FROM ZDWxx_FMACTB_PERF1 T1 INNER JOIN ZDWxx_FMACTB_LOAN_ORIG1_VSAM T2
ON T1.LOAN_SEQ_NUM = T2.LOAN_SEQ_NUM_TEXT

```

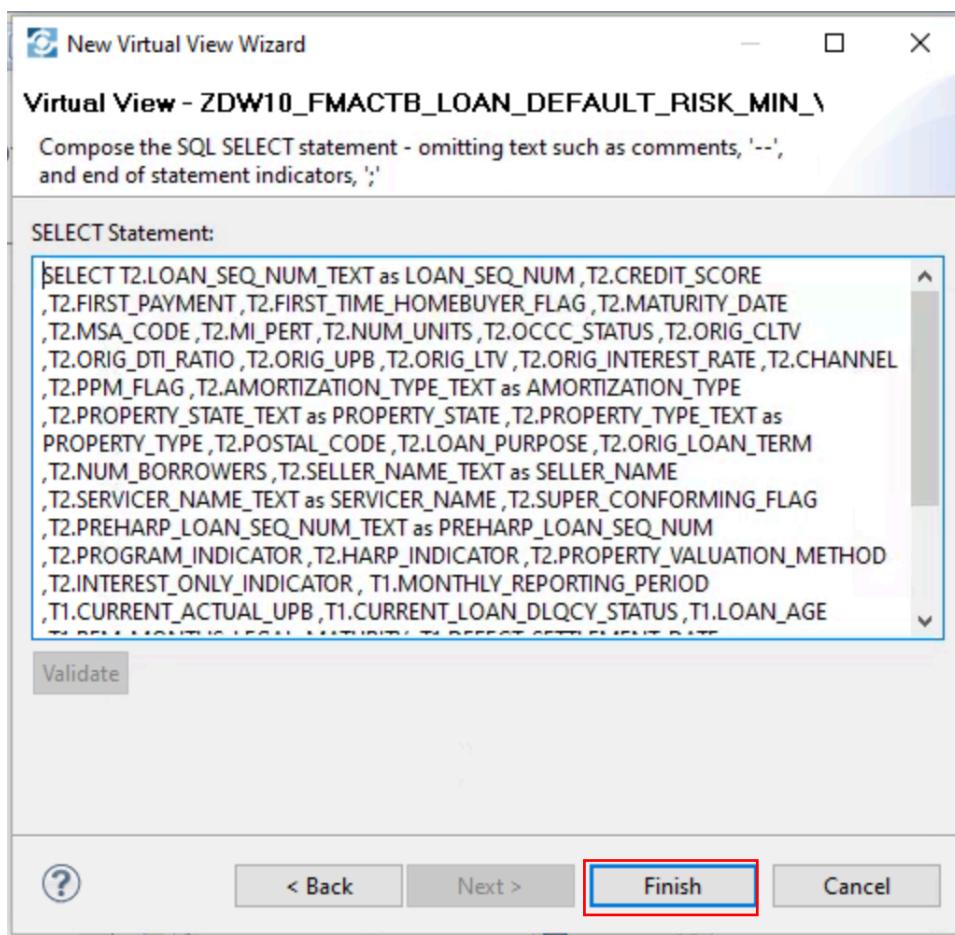
There is a warning indicating that the SQL statements needs to be validated. Click the **Validate** button.



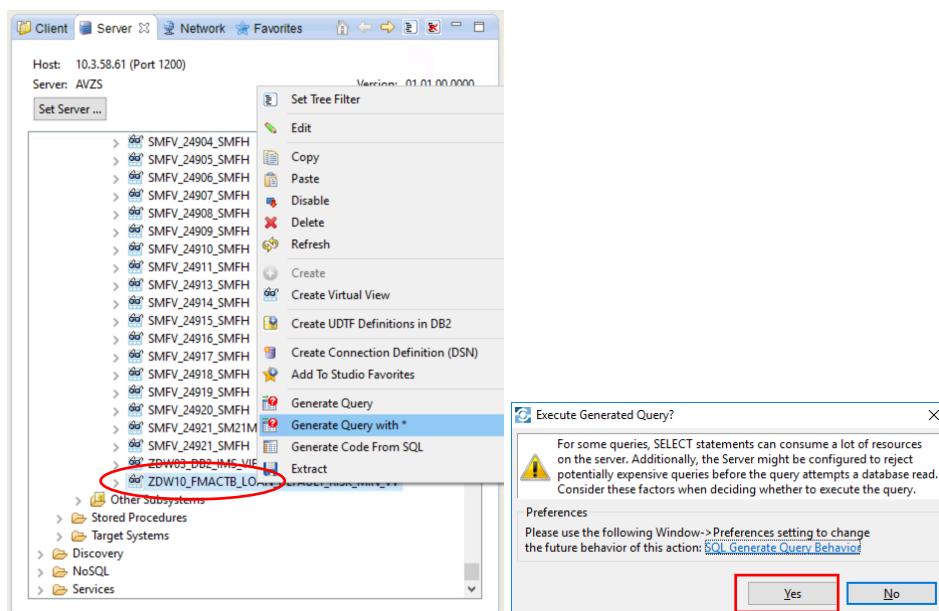
- A pop-up window will indicate if the SQL will run successfully when the virtual view is accessed. Click OK.



- Click the **Finish** button to finish the view creation.



- The new virtual view will be displayed in the view list. Right-click on the new view and choose **Generate Query with ***.
- By default, a warning dialog will open due to the potential consumption of resources on the server. Click on the **Yes** button to execute the query.



The SQL select statement will be displayed in the editor and the result set is displayed in a window below the editor.

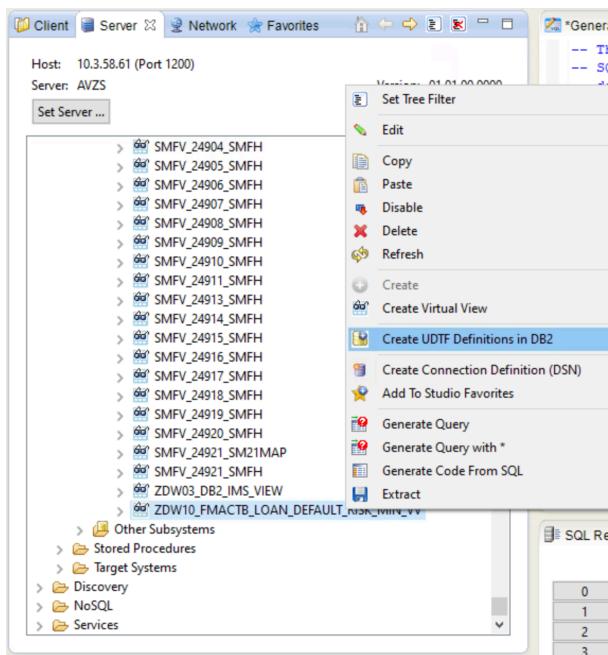
The screenshot shows the IBM Data Virtualization Manager interface. In the top right, there's a code editor window titled "Generated.sql" containing several SQL queries related to ZDW10_FMACTB_PERFI and ZDW10_FMAC_LOAN_DEFAULT_RISK_MIN_VV. A red oval highlights the bottom-most query. Below the editor is a "SQL Results" window displaying a table with 10000 rows. The table has columns: LOAN_SEQ_NUM, MONTHLY_REPORTING_PERIOD, CURRENT_ACTUAL_UPB, CURRENT_LOAN_DLQCY_STATUS, LOAN_AGE, and REM_M. The data shows five rows of loan information, with the last row being highlighted by a red oval.

LOAN_SEQ_NUM	MONTHLY_REPORTING_PERIOD	CURRENT_ACTUAL_UPB	CURRENT_LOAN_DLQCY_STATUS	LOAN_AGE	REM_M
0	F20Q41275812	2021-02-01	296000.0	0	360
1	F20Q41275812	2021-03-01	296000.0	1	359
2	F20Q41275812	2021-04-01	296000.0	2	358
3	F20Q41275812	2021-05-01	295000.0	3	357
4	F20Q41275812	2021-06-01	295000.0	4	356

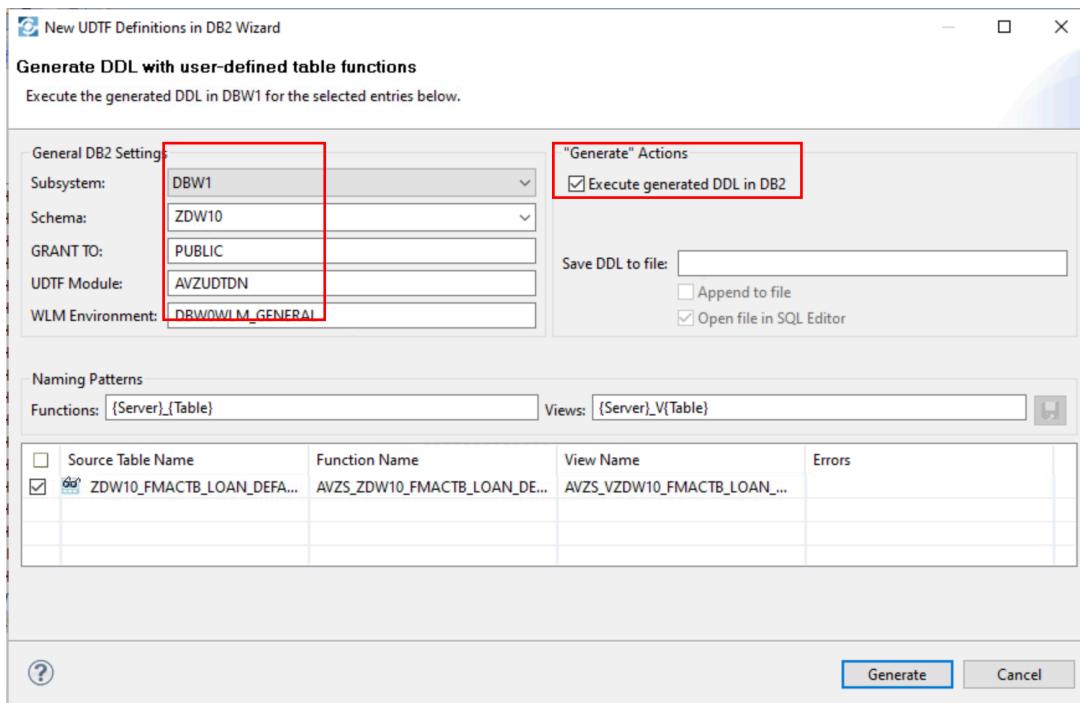
1.1.3. Create a UDTF view in Db2 for z/OS

We are now going to use the just created view to create a User Defined Table Function in Db2 for z/OS. A UDTF is composed of a view and a function defined in Db2 for z/OS running in WLM environment. When calling the view in Db2, the associated function is triggered and connects to DVM to execute the query using the corresponding DVM view.

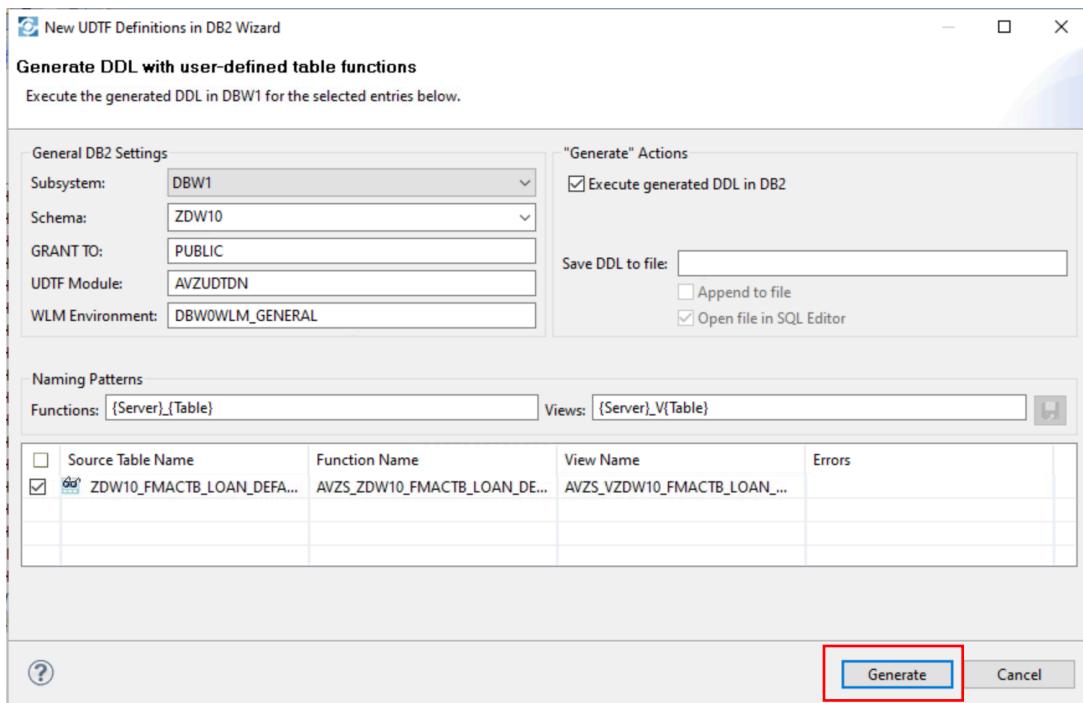
- To see the existing definitions of virtual tables on this DVM server:
 - Expand SQL
 - Expand Data
 - Expand AVZS (DVM Server)
 - Expand Virtual Tables
- Look for **ZDWxx_FMACTB_LOAN_DEFAULT_RISK_MIN_VV** virtual table.
- To create a new UDTF definition in Db2 for z/OS, Right-click on **ZDWxx_FMACTB_LOAN_DEFAULT_RISK_MIN_VV** view and choose **Create UDTF definitions in Db2**.



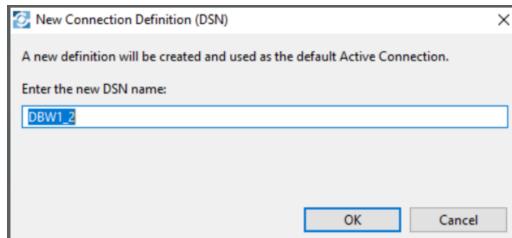
- Fill in the following screen with correct information:
 - Subsystem is: **DBW1**
 - Schema is: **ZDWxx** (replace xx with your userid)
 - GRANT TO is: **PUBLIC**
 - UDTF Module is: **AVZUDTDN**
 - WLM Environment is: **DBW0WLM_GENERAL**
- Make sure to select also “**Execute generated DDL in DB2**”
- Please note the names of the new view and function as we are going to use it later on.



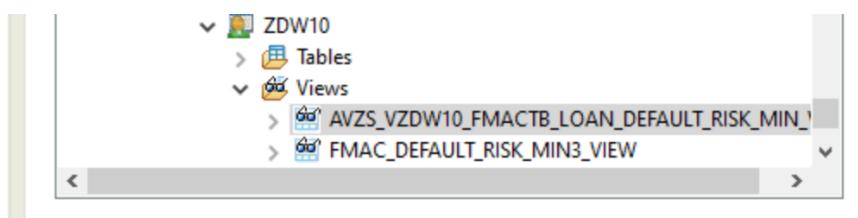
- When all information are filled in, click Generate to execute DDL in Db2 and create the new view and associated function.



- You may receive a pop-up window to create a new database connection. Click Ok to proceed.



You will then see a new view created under your schema ZDWxx in the DVM tree view under DBW1 subsystem.



Congratulation. You created your first UDTF view in Db2 for z/OS. We are now going to enable AI queries on this new view in SQL Data Insight.

1.1.4. Enable AI on a Db2 for z/OS UDTF View

In the previous step, you created a UDTF view in Db2 for z/OS called **AVZS_VZDWxx_FMACTB_LOAN_DEFAULT_RISK_MIN_VV**. Now we are going to “enable” for the semantic queries against this view supported by SQL DI.

- Log back to the SQL Data Insight web user interface in your browser.
- In the AI objects list view Select the view to be enabled for AI
- Click on the **Add object** button at the top-right of the AI Objects list.

AI objects

DBW1				
Name	Schema	Type	Status	Last updated
FMACTB_ORIG1	ZDW10	View	Enabled	Aug 31, 2022 4:39 PM
FMAC_DEFAULT_RISK_MIN_VIEW1	ZDW10	View	Enabled	Sep 1, 2022 5:45 PM
Resources per page: 10 1–2 of 2 items				

- You will be presented with the **Add object** view. From it, click on the **schema** drop-down list. In the list, scroll down to your assigned <schema> name from your worksheet. Click the check-box for your schema name, and then click outside of the list box to close it.

Add object

DBW1

Choose one or more schemas to list associated Db2 objects. From the list, select the Db2 objects to add as new AI objects or enable for AI query.

Schema: 1 × Select schema ▾

Name	Schema	Last activity
<input checked="" type="checkbox"/> ZDW10	ZDW10	ZDW10
<input type="checkbox"/> ZDW11	ZDW11	
<input type="checkbox"/> ZDW12	ZDW12	
<input type="checkbox"/> ZDW13	ZDW13	
<input type="checkbox"/> ZDW14	ZDW14	

No object available. Select one or more schemas to list available objects.

Items per page: 10 0–0 of 0 items

Your user's FreddieMac tables and view should be listed in the displayed page. You should notice that the row for your **FMACTB_ORIG1** and **FMAC_DEFAULT_RISK_MIN_VIEW1** tables seem disabled (the checkbox in the name column is gray and not selectable). As we already enabled AI on these tables, this is normal.

We are now going to use the new UDT view just created and called

AVZS_VZDWxx_FMACTB_LOAN_DEFAULT_RISK_MIN_VV which joins 1 Db2 table and 1 VSAM file on the LOAN_SEQ_NUM column.

- To Preview the data of the view **AVZS_VZDWxx_FMACTB_LOAN_DEFAULT_RISK_MIN_VV** created for you to get an idea of the type of data in the table, Click on the information icon  located on right side of the row for your table
AVZS_VZDWxx_FMACTB_LOAN_DEFAULT_RISK_MIN_VV

DBW1			
Choose one or more schemas to list associated Db2 objects. From the list, select the Db2 objects to add as new AI objects or enable for AI query.			
Schema:	1  Select schema 	 Find table/view	
<input type="checkbox"/>	FMACTB_ORIG1	ZDW10	 Aug 31, 2022 4:39 PM 
<input type="checkbox"/>	FMAC_DEFAULT_RISK_MIN_VIEW1	ZDW10	 Sep 1, 2022 5:45 PM 
<input type="checkbox"/>	FMACTB_PERF1	ZDW10	 Sep 1, 2022 5:54 PM 
<input type="checkbox"/>	AVZS_VZDW10_FMACTB_LOAN_DEFAULT_RISK_MIN_VV	ZDW10	
Items per page: 10 		1–4 of 4 items	1  of 1 page  

A popup window should appear with an indication of the number of records in the view as well as a sample of rows returned for the view. table, helping you to understand the type of data in the table. If instead, you get an error trying to access your **AVZS_VZDWxx_FMACTB_LOAN_DEFAULT_RISK_MIN_VV** table, contact a lab administrator.

Table view				
AVZS_VZDW10_FMAC_LOAN_DEFAULT_RISK_MIN_VV				
Total rows: 0				
ORIG_LOAN_TERM	ORIG_DTI_RATIO	MODIFICATION_FLAG	MSA_CODE	PROPERTY_TYPE
360	39		0	SF
360	39		0	SF
360	39		0	SF
360	39		0	SF
360	39		0	SF
360	39		0	SF
360	39		0	SF
180	41		35660	SF
180	41		35660	SF
180	41		35660	SF
180	41		35660	SF

- If you scroll to the right of the table view on your screen you can find out for example the **CURRENT_LOAN_DLQCY_STATUS** column which indicates if the customer already defaulted on their loans payments.
- **Click** the close button on the top right of the pop-up window once you are finished previewing the data.
- **Select** the checkbox to the left of the Name column for your view.
- **Click the Enable AI query button** on the bottom-right of the screen, which should have become enabled once you selected your **ZDWxx_FMACTB_LOAN_DEFAULT_RISK_MIN_VV** view.

The screenshot shows a dialog box titled "DBW1" with the instruction: "Choose one or more schemas to list associated Db2 objects. From the list, select the Db2 objects to add as new AI objects or enable for AI query." Below this is a table with columns: Name, Schema, and Last activity. The table contains four rows:

Name	Schema	Last activity
FMACTB_ORIG1	ZDW10	Aug 31, 2022 4:39 PM
FMAC_DEFAULT_RISK_MIN_VIEW1	ZDW10	Sep 1, 2022 5:45 PM
FMACTB_PERF1	ZDW10	Sep 1, 2022 5:54 PM
<input checked="" type="checkbox"/> AVZS_VZDW10_FMAC_LOAN_DEFAULT_RISK_MIN_VV	ZDW10	

At the bottom, there are buttons for "Cancel", "Add object" (which is blue), and "Enable AI query".

The column configuration dialog should be displayed next.

- Select and configure the columns to be analyzed (shown on the next page)
- The next page shows the Column configuration page. This allows you to select the columns from the view to be used in the modeling as well as adjust the data-types that will be used for each column.

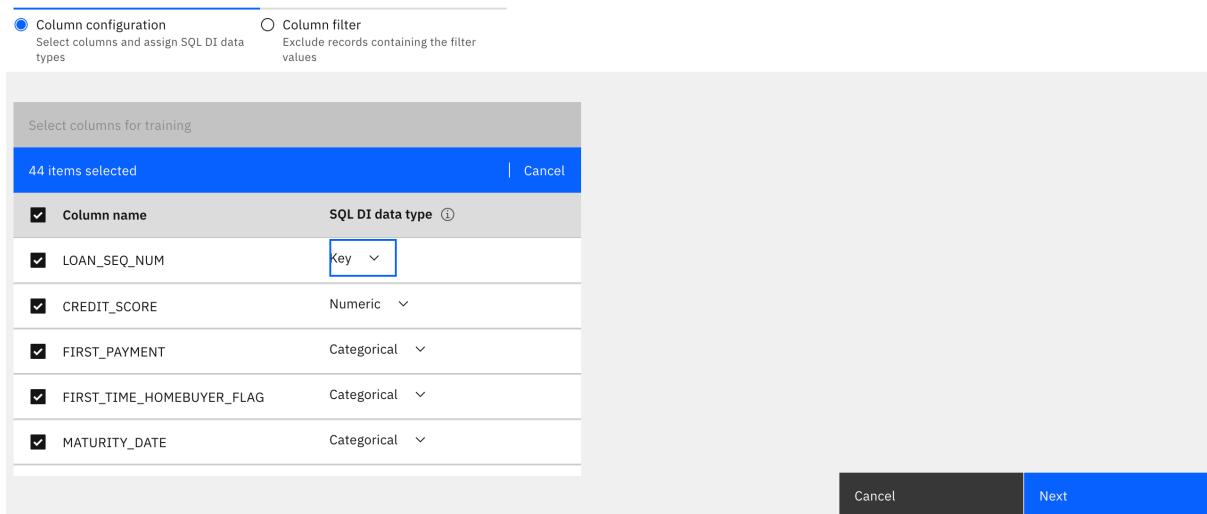
For this exercise, we will use all columns in the view.

- At this point, you have the option to re-categorize your columns to a different SQL DI data type. However, you have to understand the data to know what SQL DI data types are valid for that given column.
- Like in previous chapter, only change the **LOAN_SEQ_NUM** column to data type **Key**. When processing an AI query that includes a column with the `key` type, SQL DI evaluates the affected rows in their entirety and effectively compares all the values in one row to those in another, not just the values in the column of the `key` type.
- **Select the check box** next to “column name” column header shown in the table. All columns of the table should be auto-selected. Once **all columns are selected** and **LOAN_SEQ_NUM** has been set to SQL DI data type Key, click the **Next** button

Enable AI query

ZDW10.AVZS_VZDW10_FMAC_LOAN_DEFAULT_RISK_MIN_VV

Enabling an object for AI query requires column configuration and model training. To enable ZDW10.AVZS_VZDW10_FMAC_LOAN_DEFAULT_RISK_MIN_VV for AI query, select or [import](#) the column configuration of the AI object from a JSON file. Make sure the columns defined in the file are consistent with those in ZDW10.AVZS_VZDW10_FMAC_LOAN_DE



Next displayed is the **Column filter** page. This page allows you to exclude data known not to be valid for analysis such as null or missing values. This can be done globally for all columns, or on a per-column basis as suggested by the dialog's options.

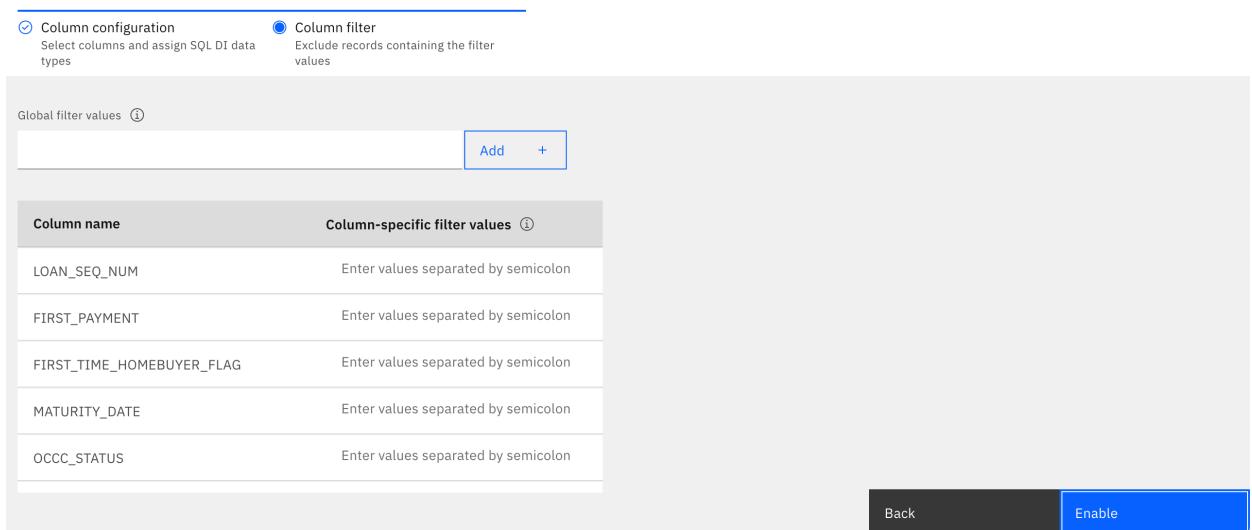
The **AVZS_VZDWxx_FMACTB_LOAN_DEFAULT_RISK_MIN_VV** view has been set up to include only valid data, so you do not have to specify any filter.

- Click the **enable** button to proceed.

Enable AI query

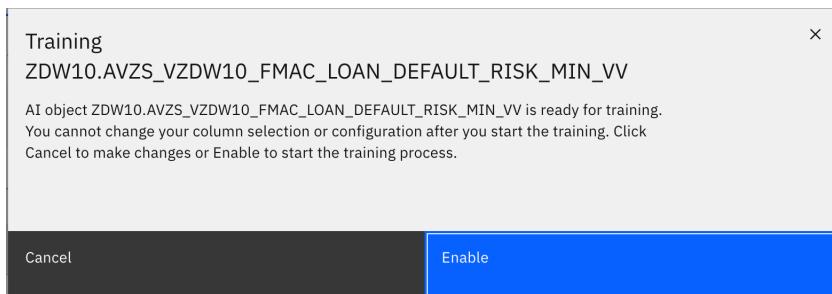
ZDW10.AVZS_VZDW10_FMAC_LOAN_DEFAULT_RISK_MIN_VV

Enabling an object for AI query requires column configuration and model training. To enable ZDW10.AVZS_VZDW10_FMAC_LOAN_DEFAULT_RISK_MIN_VV for AI query, select or [import](#) the column configuration of the AI object from a JSON file. Make sure the columns defined in the file are consistent with those in ZDW10.AVZS_VZDW10_FMAC_LOAN_DE



You are presented with one last point of interaction where you can choose to proceed or cancel the operation.

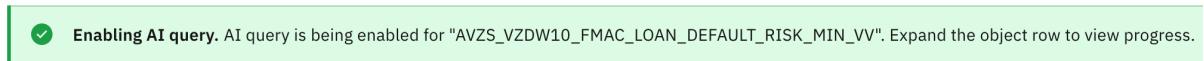
- Click the **enable** button



- Monitor AI object training

After several seconds on a “spin/wait” cursor being displayed in grayed out view of your window, you should be taken back at the main list of AI Objects for the database that you are connected to.

You should see a green status box toward the top of your browser window indicating that your AI query is being enabled for your view.



- Click the X close symbol on the right side of the status box to close it.



Your view should be listed in the table. Its status should be **Enabling** with a spin-wait cursor to the left of the status.

- Every 30 seconds or so, click the “refresh” button on your browser since your object training status does not update on its own.

This training should take at least 1.5 minutes. Expect this to elongate if there are multiple people training their model at the same time.

DBW1					
Name	Schema	Type	Status	Last updated	
FMACTB_ORIG1	ZDW10	View	Enabled	Aug 31, 2022 4:39 PM	⋮
FMAC_DEFAULT_RISK_MIN_VIEW1	ZDW10	View	Enabled	Sep 1, 2022 5:45 PM	⋮
FMACTB_PERF1	ZDW10	Table	Enabled	Sep 1, 2022 5:54 PM	⋮
AVZS_VZDW10_FMAC_LOAN_DEFAULT_RISK_MIN_VV	ZDW10	View	Enabling	Sep 2, 2022 3:55 PM	⋮

Resources per page: 10 1–4 of 4 items

- Once your view’s AI object changes to Enabled status, you may proceed to the next section. If it does not change after a couple of minutes, or changes to a state other than enabled, contact a lab administrator.

DBW1					
zdw10					Add object + Run query
Name	Schema	Type	Status	Last updated	
FMACTB_ORIG1	ZDW10	View	Enabled	Aug 31, 2022 4:39 PM	⋮
FMAC_DEFAULT_RISK_MIN_VIEW1	ZDW10	View	Enabled	Sep 1, 2022 5:45 PM	⋮
FMACTB_PERF1	ZDW10	Table	Enabled	Sep 1, 2022 5:54 PM	⋮
AVZS_VZDW10_FMAC_LOAN_DEFAULT_RISK_MIN_VV	ZDW10	View	Enabled	Sep 2, 2022 4:04 PM	⋮

Resources per page: 10 1–4 of 4 items 1 of 1 page ⌂ ⌃ ⌄

You can see here that we have enabled semantic queries not only for tables but also on views created on Db2 for z/OS and VSAM files. This way you can enrich data you are going to enable for AI queries but located on different Db2 for z/OS table.

2. Explore other semantic queries

In this section, you can now run similar queries than before but now on the VSAM data made available through the Db2 UDTF view. These queries should have also been made available to you (probably as a file on your desktop) so that you can copy and paste them into the query tool.

If you have not been told where the file is, make sure to ask your lab instructor.



Be careful with pasting the queries into the SQL tool. Sometimes extended code point characters, such as “smart quotes” are introduced, which will cause SQL errors.

- Bring up the query tool to run the semantic queries
- From the AI objects list, **click** on the **Run query** button at the top-right of the page.

AI objects

DBW1					
zdw10					Add object + Run query
Name	Schema	Type	Status	Last updated	
FMACTB_ORIG1	ZDW10	View	Enabled	Aug 31, 2022 4:39 PM	⋮

Resources per page: 10 1–1 of 1 items 1 of 1 page ⌂ ⌃ ⌄

Some notes on the Run query page that appears (shown below):

The run query tool allows you to explore different semantic queries. By clicking on the **Add SQL +** tab at the top right of the SQL edit box, it creates another query tab where you can enter another query (up to a max of 10 tabs)

Clicking on the Query type drop-down at the top of the page will populate your query edit box with a default query of the selected type. This is optional as you can type or paste in the query that you want to run.



Note that before you run a query, check to make sure the schema is set to your <schema> value provided in your worksheet. If the schema is not set properly for your userid, the query will fail, with an error displayed below the **Run** button

- SQL-9 AI_SIMILARITY -- Find loans who are the most similar to the loan sequence number with LOAN_SEQ_NUM 'F21Q11276092' that was known to have defaulted their payment and including LOAN_SEQ_NUM 'F21Q11276092'. This particular query is targeting a different table now.

 Note you are now using the Db2 UDTF View joining the Db2 table **FMACTB_PERF1** and the VSAM **FMACTB_LOAN_ORIG1_VSAM**. Using this trained view gives us the possibility to enrich the data we used for creating our AI model using VSAM data and may produce a different result set.

```
select loan_seq_num, similarity from
  (select loan_seq_num,
          avg(AI_SIMILARITY(LOAN_SEQ_NUM, 'F21Q11276092' using model column
LOAN_SEQ_NUM)) as similarity
   from ZDWxx. AVZS_VZDWxx_FMACTB_LOAN_DEFAULT_RISK_MIN_VV
  where loan_seq_num <> 'F21Q11276092',
   group by loan_seq_num ) as temp
 WHERE SIMILARITY > 0.3
 order by similarity desc
 fetch first 100000 rows only ;
```

 Here you should expect the query to take more time to execute. In fact our VSAM data is quite larger than the Db2 table.

Congratulations! You finished this Hands-On Lab. We really value your feedback, so please reach us to provide some feedback on these materials. Thanks in advance.

End of lab.