Enhancing Denodo with Interzoid APIs

+

Example #1 - Using Interzoid's Company Data Matching API with Denodo to find redundant data within a database, step-by-step

September 2020

All examples performed with Denodo Express v7

Introduction

In this walkthrough example, **Denodo Express** is used to perform a join on a virtual database with Interzoid's Cloud-based Company Matching API using algorithmically generated "similarity keys." These keys will serve as the basis of a match to identify similar company names within a database. This essentially turns Denodo into a powerful data matching tool.

If you do not have Denodo Express, it is available as a free download here: https://www.denodo.com/en/denodo-platform/denodo-express

The premise of matching data is that Interzoid's Company Matching JSON API generates "similarity keys" based on algorithms and data-specific knowledge bases. These keys help identify similar company or organization names within a single database or across multiple databases as part of an SQL Join (or other similar data matching) function.

The use of similarity keys helps to overcome the issue of inconsistent alphanumeric data for representing company names within a dataset. The technique can provide significantly greater impact and value for Data Virtualization efforts and the various data initiatives that make use of a logical data layer. It can also help identify data quality challenges in underlying source data that need to be addressed.

It is important and useful not only to identify cases of redundant data records in a single dataset, but data matching via similarity keys can also significantly enhance the ability to match and join data across multiple datasets where data can be inconsistently represented. This increases the accuracy and value of data residing within the Data Virtualization layer.

For example, the following similarity keys were generated for the company names below.

Company	Similarity Key
B of A	gh89jRfV0SR11wdJwDO0Cvzh4xCg
Bank of America	gh89jRfV0SR11wdJwDO0Cvzh4xCg
PG&E	4rtY6ImDHMQUX6EMlBHD9H17SAoTamH6zCBjFm1
GE	LpOcDqC1XH5uYnt65tBJhmBh7RhDNsxa
Gen. electric	LpOcDqC1XH5uYnt65tBJhmBh7RhDNsxa
Kellogs	Rf6RCPXSmrZOp8FKRSjQuRvfzO3ef
Pacific Gas	4rtY6ImDHMQUX6EMlBHD9H17SAoTamH6zCBjFm1
PGE Inc	4rtY6ImDHMQUX6EMlBHD9H17SAoTamH6zCBjFm1

Using the Interzoid Company Matching API, likely matches will generate the same similarity key. You can see that similar company names generate the same similarity key. This makes them easy to identify as part of a data query.

How to achieve matching results with Denodo Express using data virtualization:

There are **five steps** in this example:

Step #1: Create a Data Source and Base View within Denodo Express for the source data

Step #2: Configure the Interzoid Company Matching API as a Data Source

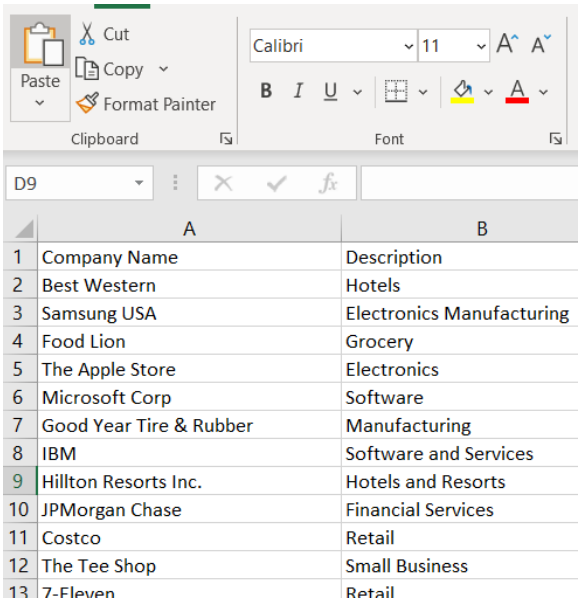
Step #3: Configure the Interzoid API Base View

Step #4: Define the Derived View using a Join function

Step #5: Execute the Join function to display results

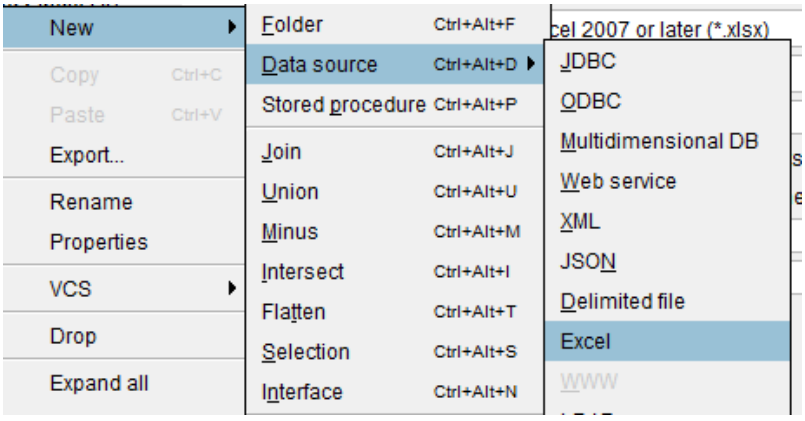
Step #1 – Create a Data Source and Base View within Denodo Express for the source data

In this example, for simplicity we will use an Excel workbook file as the data source (we could have just as easily used JDBC, ODBC, or one of the several other data sources available within Denodo). It is a list of company names with a simple description. *Filename:* "Example 1 – Companies.xlsx"

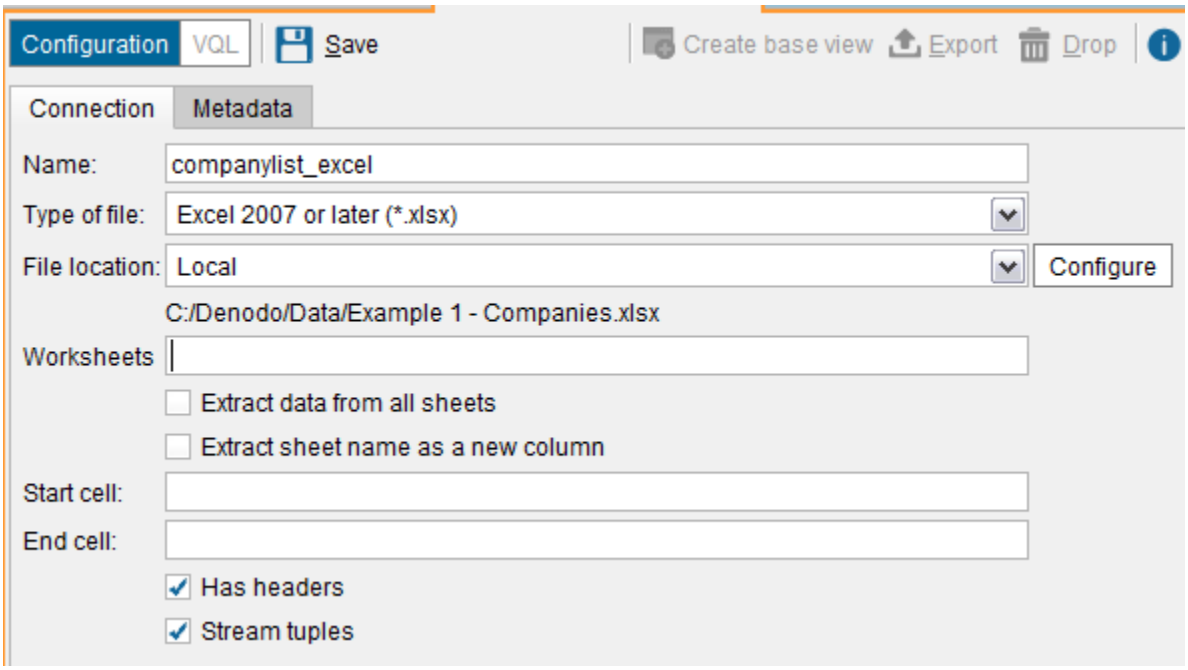


	A	B
1	Company Name	Description
2	Best Western	Hotels
3	Samsung USA	Electronics Manufacturing
4	Food Lion	Grocery
5	The Apple Store	Electronics
6	Microsoft Corp	Software
7	Good Year Tire & Rubber	Manufacturing
8	IBM	Software and Services
9	Hillton Resorts Inc.	Hotels and Resorts
10	JPMorgan Chase	Financial Services
11	Costco	Retail
12	The Tee Shop	Small Business
13	7-Eleven	Retail

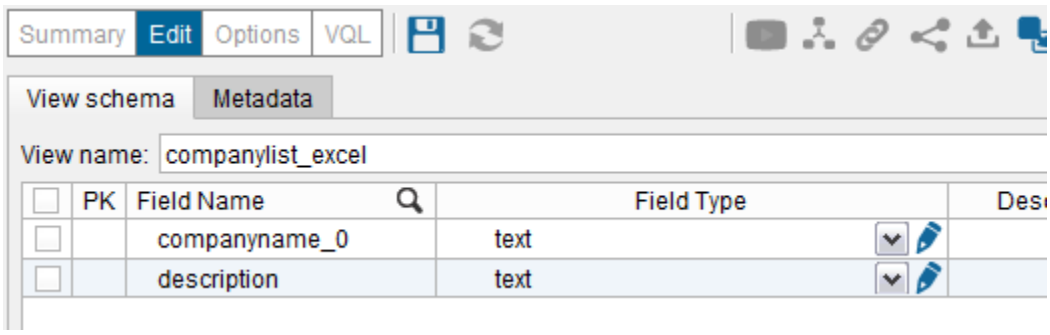
Within the Virtual DataPort Administration Tool (VDP) of Denodo Express, select "Database Administration" and create a new Server for this walkthrough. Call it "Example1." Then, from the "Example1" folder in the Server tree, create a new folder called "Data Sources" and another one called "Base Views." Within the "Data Sources" folder, create a new Excel Data Source .



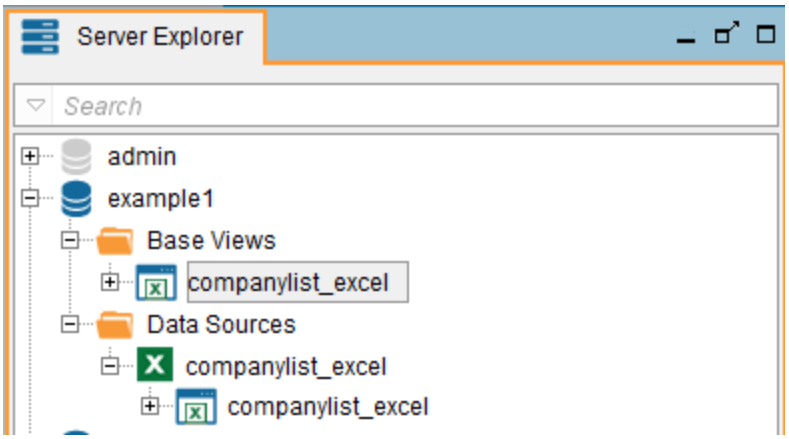
Next, change the File Location setting to “Local” and click the “Configure” button. Provide the path of the included Excel file and click “Ok.” Name the Data Source “companylist_excel”. Check the “has headers” option and click “Save”:



Next, click “Create base view” on the top. Once the Base View has been created and is displayed, click “Save”



You should now have both a Data Source and the corresponding Base View in your Server tree (you might need to drag the Base View to the Base View folder):



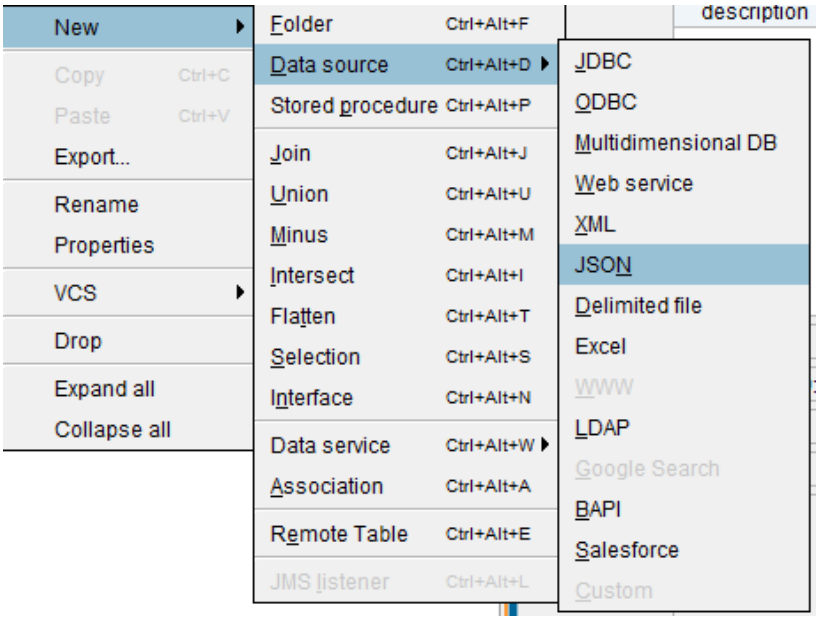
Step #2 – Configure the Interzoid Company Matching API as a Data Source

The API we will use is a REST based API that we will access using query parameters with a URL location. Data is returned from this API in JSON format, so we will set the data source up in Denodo accordingly.

For more technical information about the API:

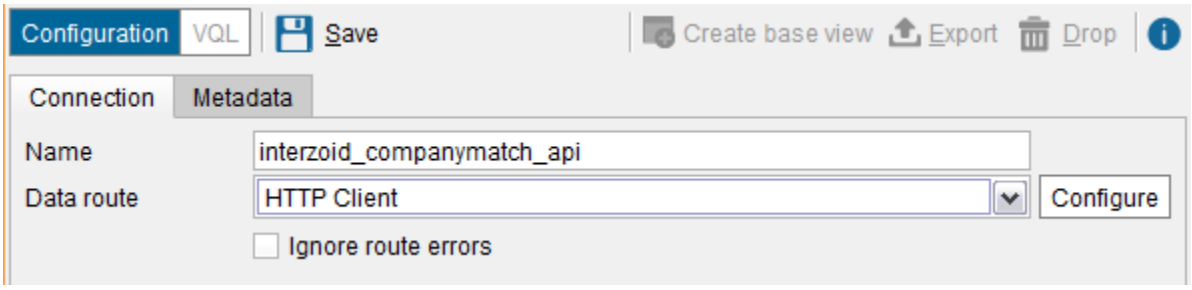
https://interzoid.com/services/getcompanymatchadvanced

Create a new data source of type “JSON”



Name the data source: “interzoid_companymatch_api”

For the “Data Route”, choose “HTTP Client” and then click the “Configure” button:



The end point for this particular API is:
https://api.interzoid.com/getcompanymatchadvanced

Calling the API requires three parameters:

License: A license key issued from Interzoid (a limited key is provided for this example, additional license keys can be obtained from Interzoid)

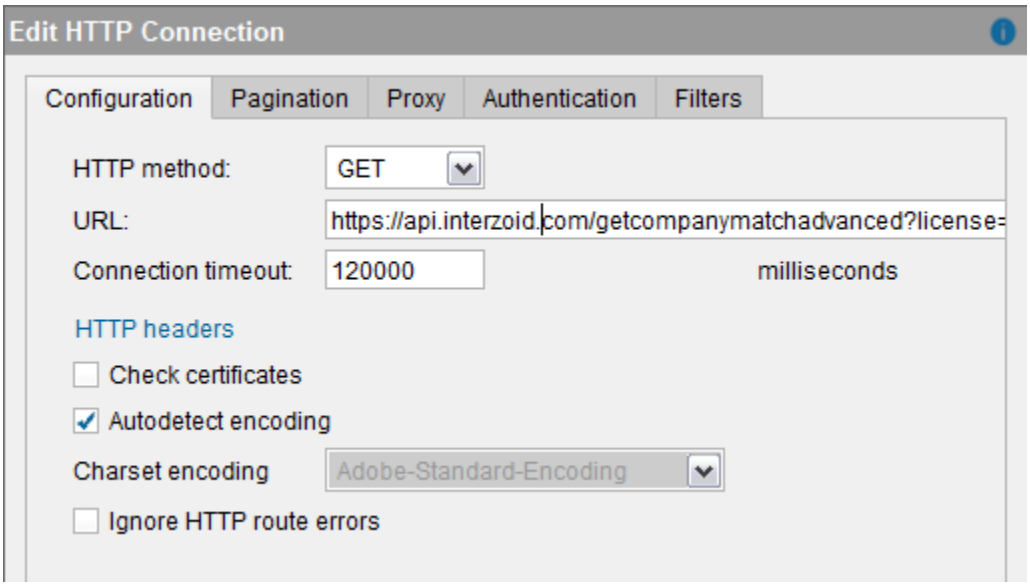
Company: Company name from which similarity key will be generated

Algorithm: Which similarity key generation algorithm will be applied

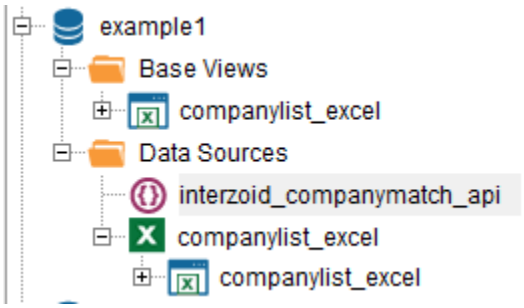
In this example, we will hard-code values for two of the parameters in the call, “license” and “algorithm.” For the third parameter, “company name”, we will use an “interpolated value”, as this tells Denodo to use a variable when calling the API. In our case, the variable will be company name values from the company list Base View that we have already set up from the source data in the Excel file.

To achieve this, in the URL field when configuring the HTTP client, we will provide the endpoint along with the query parameters. You can see the two hard-coded values for “license” and “algorithm”, and then also how the interpolated variable is denoted within the API call URL using an @ symbol and brackets:

https://api.interzoid.com/getcompanymatchadvanced?license=1a1fd54a2f8908405e8b5726c280560a&company=@{company_name}&algorithm=wide

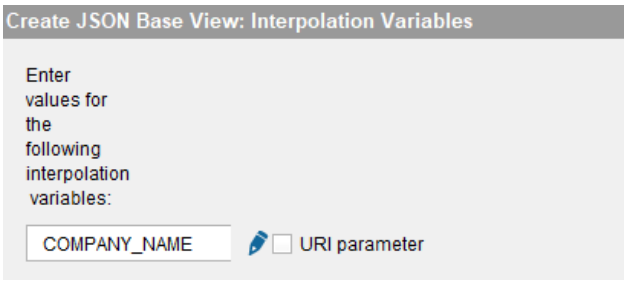


Click “OK”, and then “Save.” The configured Data Source will appear on the Server tree:

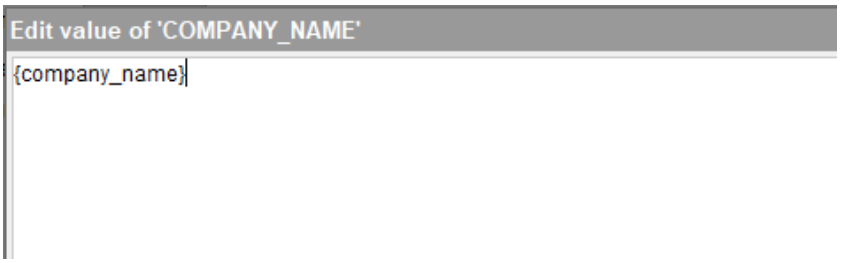


Step #3 – Configure the Interzoid API Base View

On the Configuration Panel for the “Interzoid_companymatch_api” Data Source, click “Create Base View.” You will see the following:

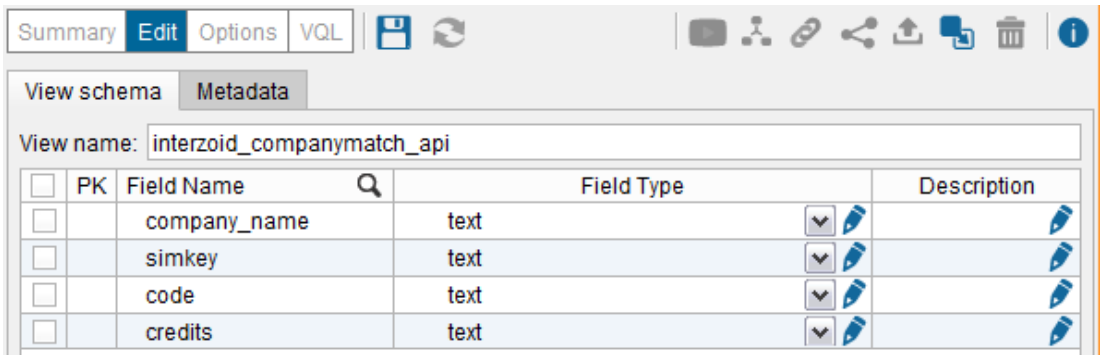


Click the blue edit pen and then enter the interpolation variable notation below including the brackets:



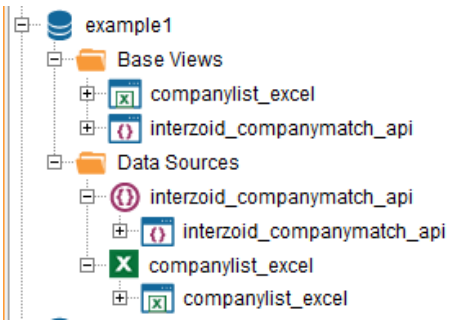
Then click “OK”, then “Next”, and then “OK” again at the “Configure JSON Wrapper” panel without any changes.

You should then see the following:



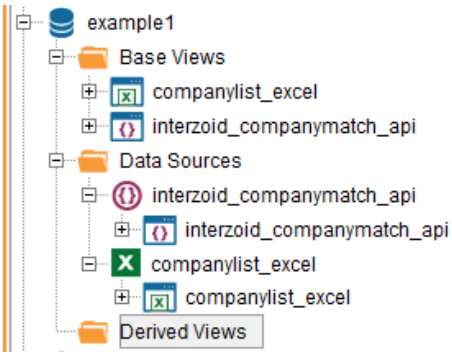
The “company_name” field is the input parameter to the API. The other three are all output parameters that will comprise the output results. All four data fields are listed here.

Click the blue “Save” disk to save the Base View. You should now have the new Base View available on the Server tree as shown:

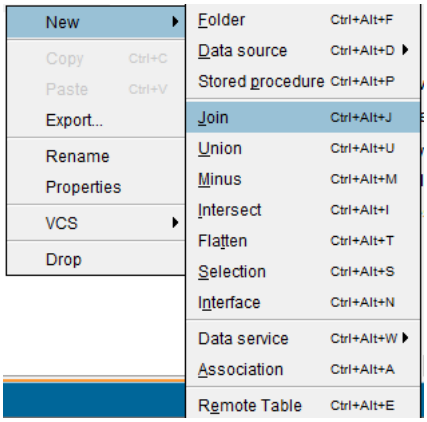


Step #4 – Create the Derived View using a Join function

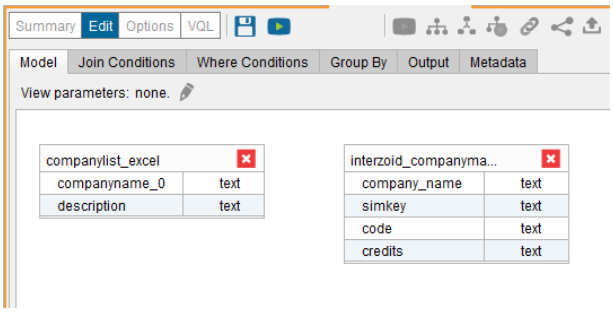
Create a new folder called “Derived Views” to follow Denodo guidelines. This is where the Data Virtualization will occur and from where we can execute our Join function. This will call the API with each value in the company list data source generating similarity keys for each and storing them within a Derived View.



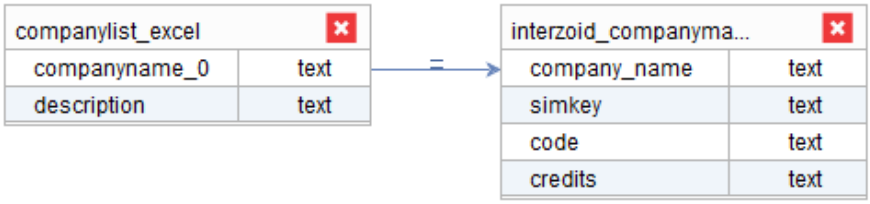
In this folder, create a new Join function:



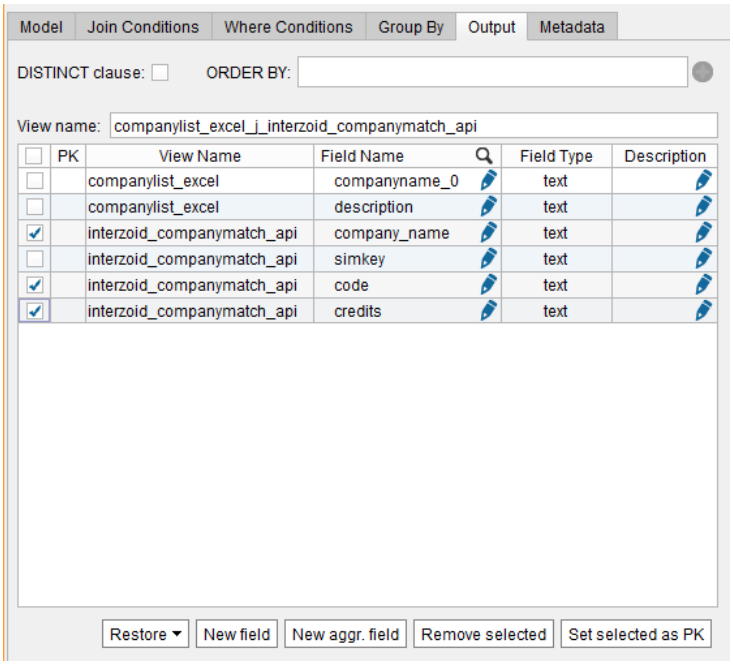
Then, drag the two Base Views from the tree onto the Join configuration panel. The panel should look like this:



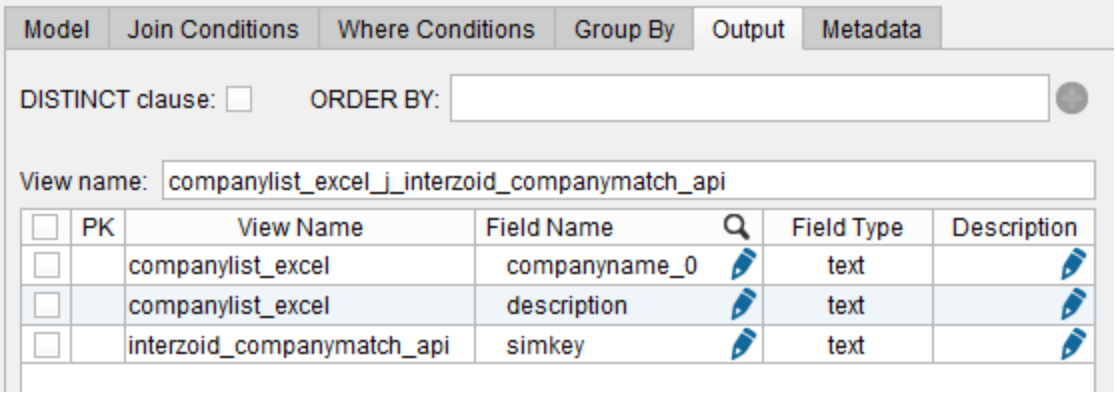
Next, drag the “companyname_0” field of the company list Base View to the “company_name” field of the Interzoid API Base View. You should see a line connecting the two fields (defining the Join) like this:



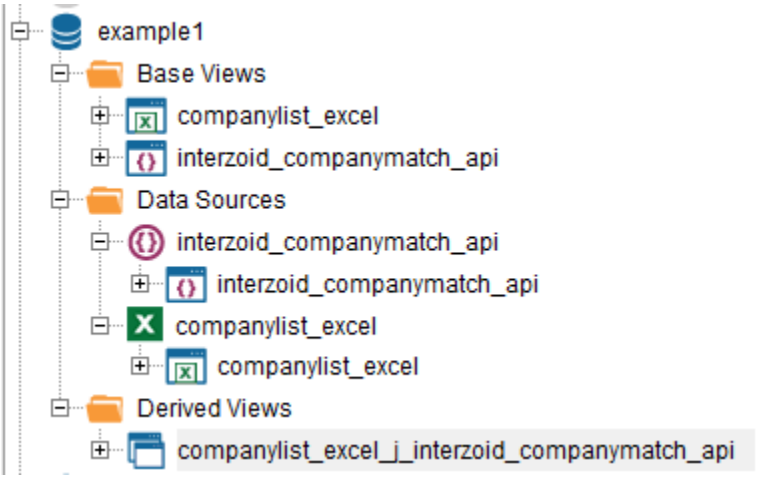
Next, define the output of the Derived View. Click the “output” tab. Then, check the three fields as shown below.



The first field selected is the input parameter to the API. This will be the same as the company name field in the Excel worksheet, so we do not need to see it twice. Also, “code” and “credits” are administrative data coming from the API that are not relevant to the Join, so check those as well. After the fields are checked, click the “Remove selected” button the bottom of the panel. You should now see this:

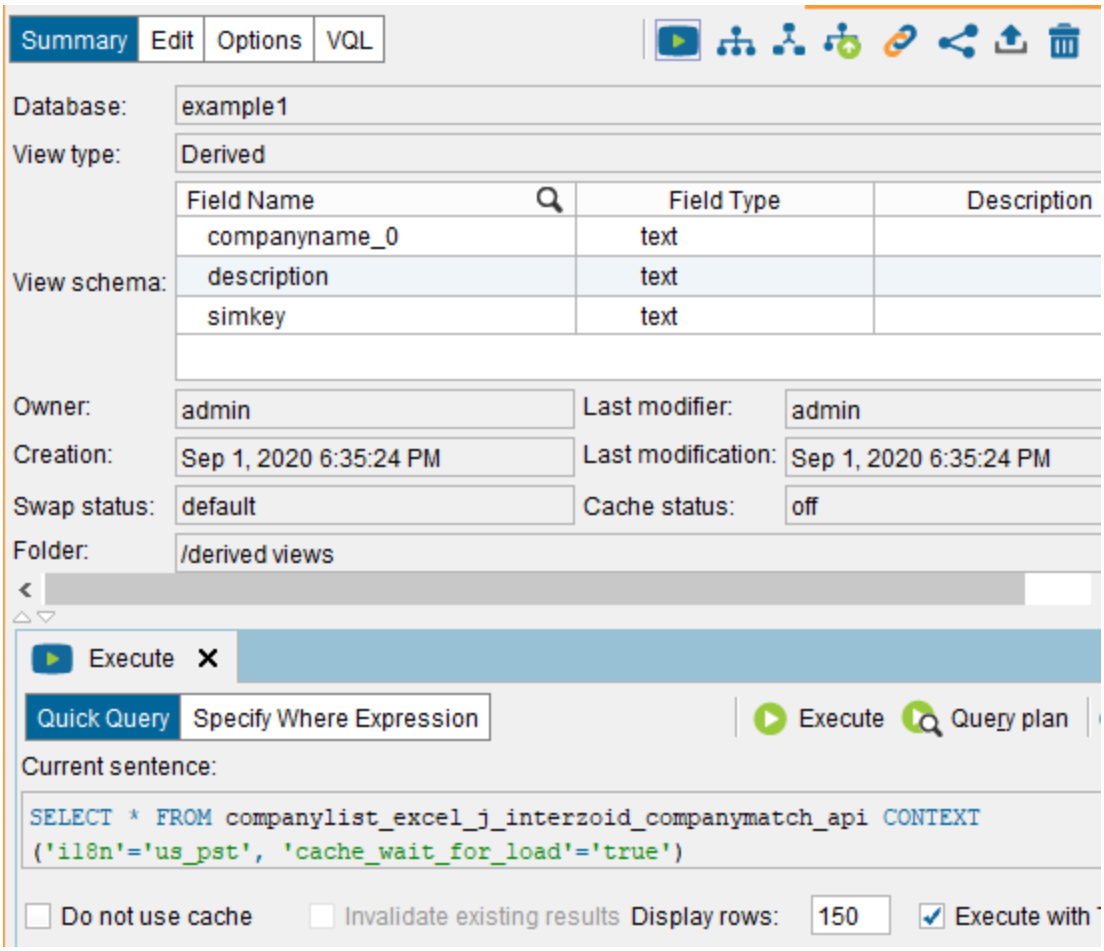


Click the blue disk “Save” button. The Join function’s Derived View will now appear in the Derived Views folder on the Server tree:



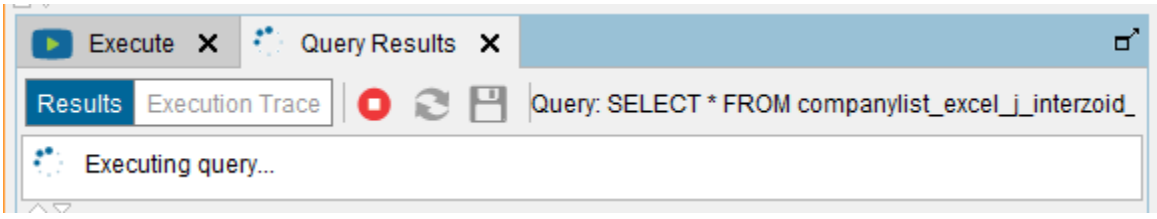
Step #5 – Execute the Join function to display results

Within the configuration panel for the Derived View, click the “Execution Panel” arrow at the top. You should see the following:

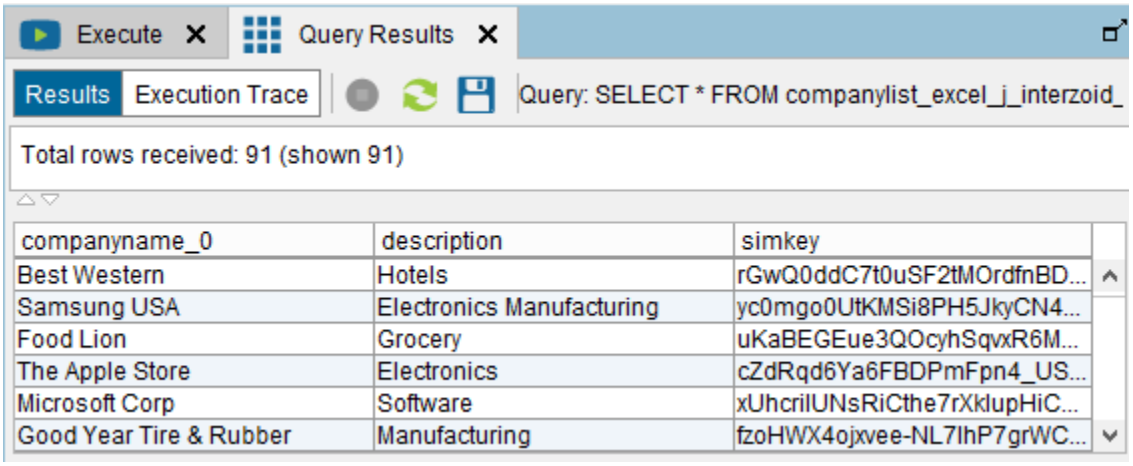


We are ready to execute the Join function that will build the Derived View by calling the API once for each record in the company list Base View. To execute, click the white arrow in the green circle “Execute” button.

You should see the “Executing query…” message. This will take a few seconds as Denodo is going to the Web to execute the Interzoid Company Matching API for the purposes of generating similarity keys.

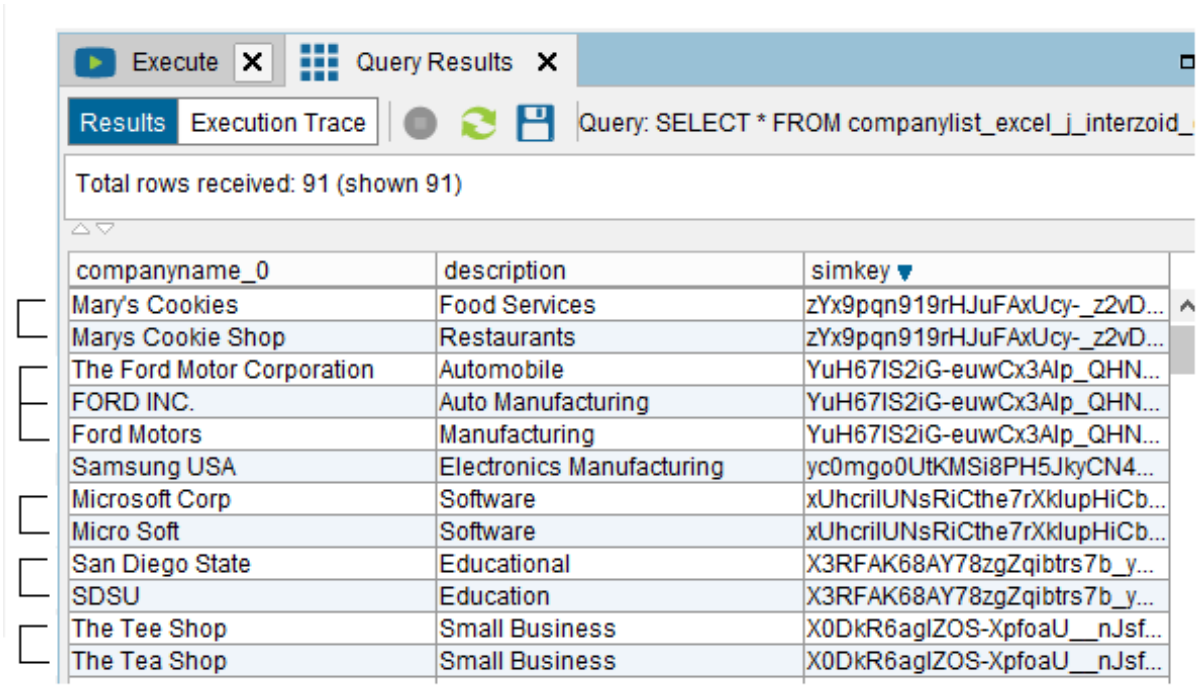


You will then see the results as shown. For each company name, you will see a generated similarity key ("simkey": a string of alphanumeric characters):



companyname_0	description	simkey
Best Western	Hotels	rGwQ0ddC7t0uSF2tMOrdfnBD...
Samsung USA	Electronics Manufacturing	yc0mgo0UtKMSi8PH5JkyCN4...
Food Lion	Grocery	uKaBEGEue3QOcyhSqvxR6M...
The Apple Store	Electronics	cZdRqd6Ya6FBDPmFpn4_US...
Microsoft Corp	Software	xUhcrilUNsRiCthe7rXklupHiC...
Good Year Tire & Rubber	Manufacturing	fzoHWX4ojxvee-NL7IhP7grWC...

The next thing we want to do is to sort the data by "simkey" (this can also be done by adding an "order by" clause to the SQL query prior to execution of the query). To do this within the query results, click the "simkey" column header twice to sort in descending order. Sorting by "simkey" will cause similar company names to line up next to each other. This makes the potential matches easy to find. You should see something like the following:



companyname_0	description	simkey
Mary's Cookies	Food Services	zYx9pqn919rHJuFAxUcy-_z2vD...
Marys Cookie Shop	Restaurants	zYx9pqn919rHJuFAxUcy-_z2vD...
The Ford Motor Corporation	Automobile	YuH67IS2iG-euwCx3Alp_QHN...
FORD INC.	Auto Manufacturing	YuH67IS2iG-euwCx3Alp_QHN...
Ford Motors	Manufacturing	YuH67IS2iG-euwCx3Alp_QHN...
Samsung USA	Electronics Manufacturing	yc0mgo0UtKMSi8PH5JkyCN4...
Microsoft Corp	Software	xUhcrilUNsRiCthe7rXklupHiCb...
Micro Soft	Software	xUhcrilUNsRiCthe7rXklupHiCb...
San Diego State	Educational	X3RFAK68AY78zgZqibtrs7b_y...
SDSU	Education	X3RFAK68AY78zgZqibtrs7b_y...
The Tee Shop	Small Business	X0DkR6aglZOS-XpfoaU__nJsf...
The Tea Shop	Small Business	X0DkR6aglZOS-XpfoaU__nJsf...

Where from here?

Now that matches have been identified, there are several potential courses of action to take, depending on business requirements.

For example, data can be exported for further analysis. Within Denodo, results can be saved/exported to a delimited file for review or additional processing.

If this is a raw marketing list, redundant data could be eliminated to save outreach costs.

If this is a customer or prospect list, logic can be created to collapse redundant records in a single instance while maintaining important data from each of the records to maintain in the surviving record.

Not only can similarity keys be used to identify redundant records in a single Base View within Denodo, but multiple Base Views can have similarity keys appended, and then be used as the basis for matching across these Base Views. This provides much higher match rates when comparing and combining multiple data sources, rather than depending on exact matches as the basis of a Join.

This walkthrough is only one example of integrating Interzoid APIs within Denodo. There are also matching APIs that include specific matching algorithms for other types of data such as individual names (Bob = Robert, Johnson = Jonsen). Also, there are data enrichment APIs available to add additional data elements via the Cloud to existing data that can enhance various data initiatives.

The possibilities are limitless.

For more information, contact support@interzoid.com or visit https://www.interzoid.com