# Phase 1

## Inner Join

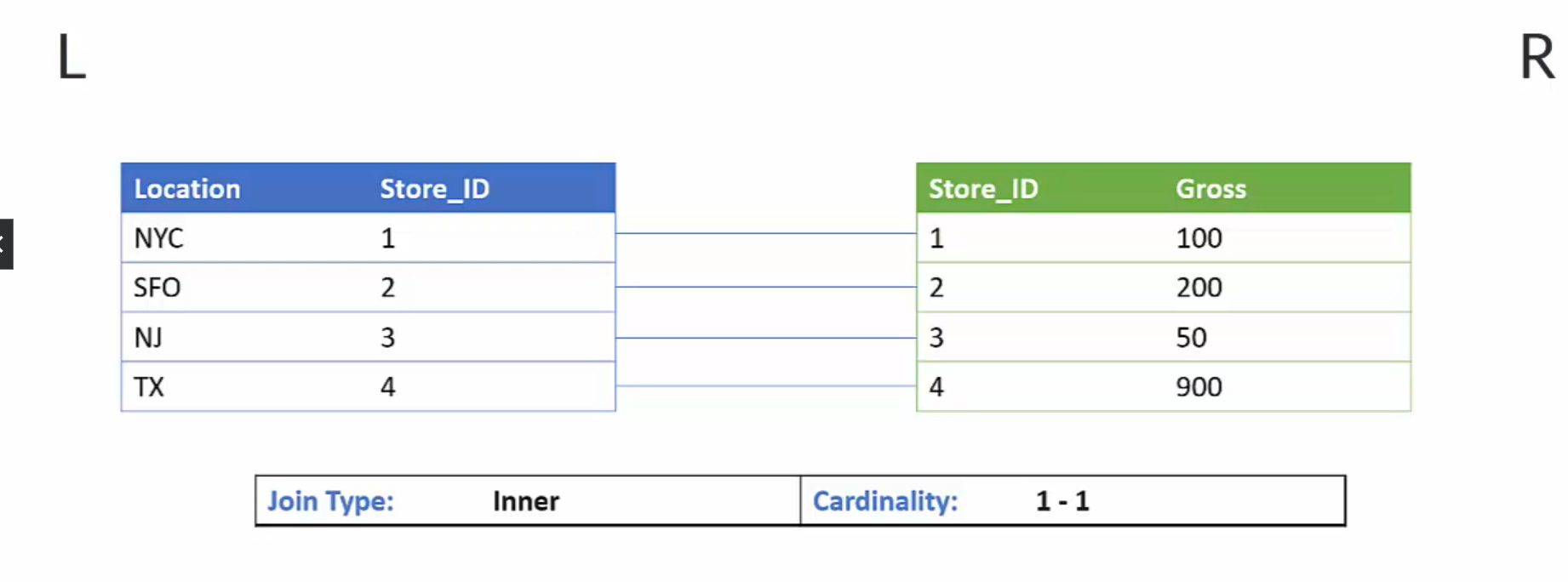


Figure 1 Inner Join Cenario.

A screenshot of a computer

Description automatically generated

Figure 2 Result of Inner Join.

A screenshot of a computer

Description automatically generated

Figure 3 I-N cardinality for Inner Join.

In MD.Address table we have ADDRESSID as key figure

Querry: -

select \* from "FL\_HDI\_DB\_1"."MD.BusinessPartner" as A

Inner Join "FL\_HDI\_DB\_1"."MD.Addresses" as B

on A."ADDRESSES.ADDRESSID" = B."ADDRESSID"

1-N Cardinality

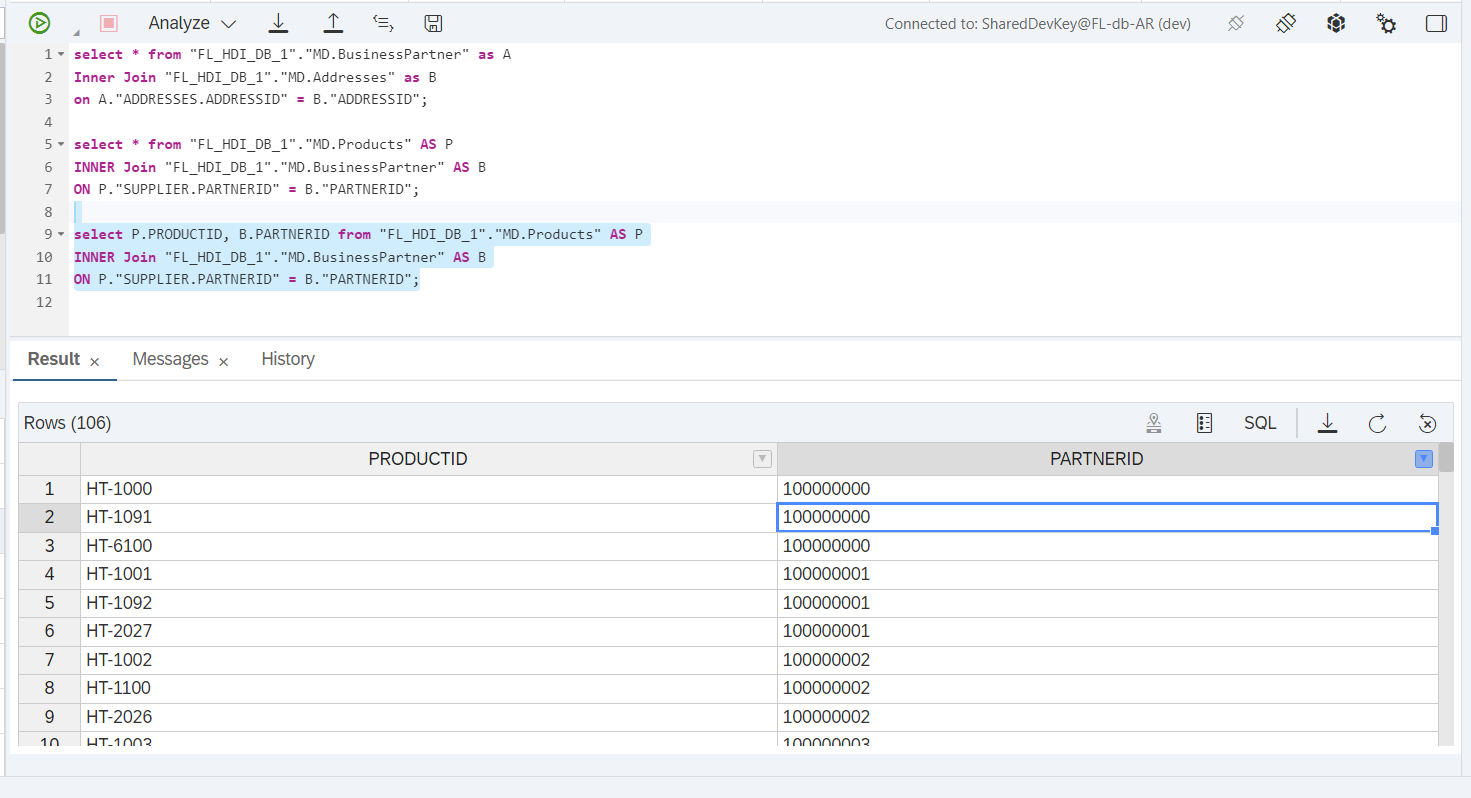
We will join the Products table with Business partner table.

Products we have PRODUCTID as primary key

1 Partner has multiple products

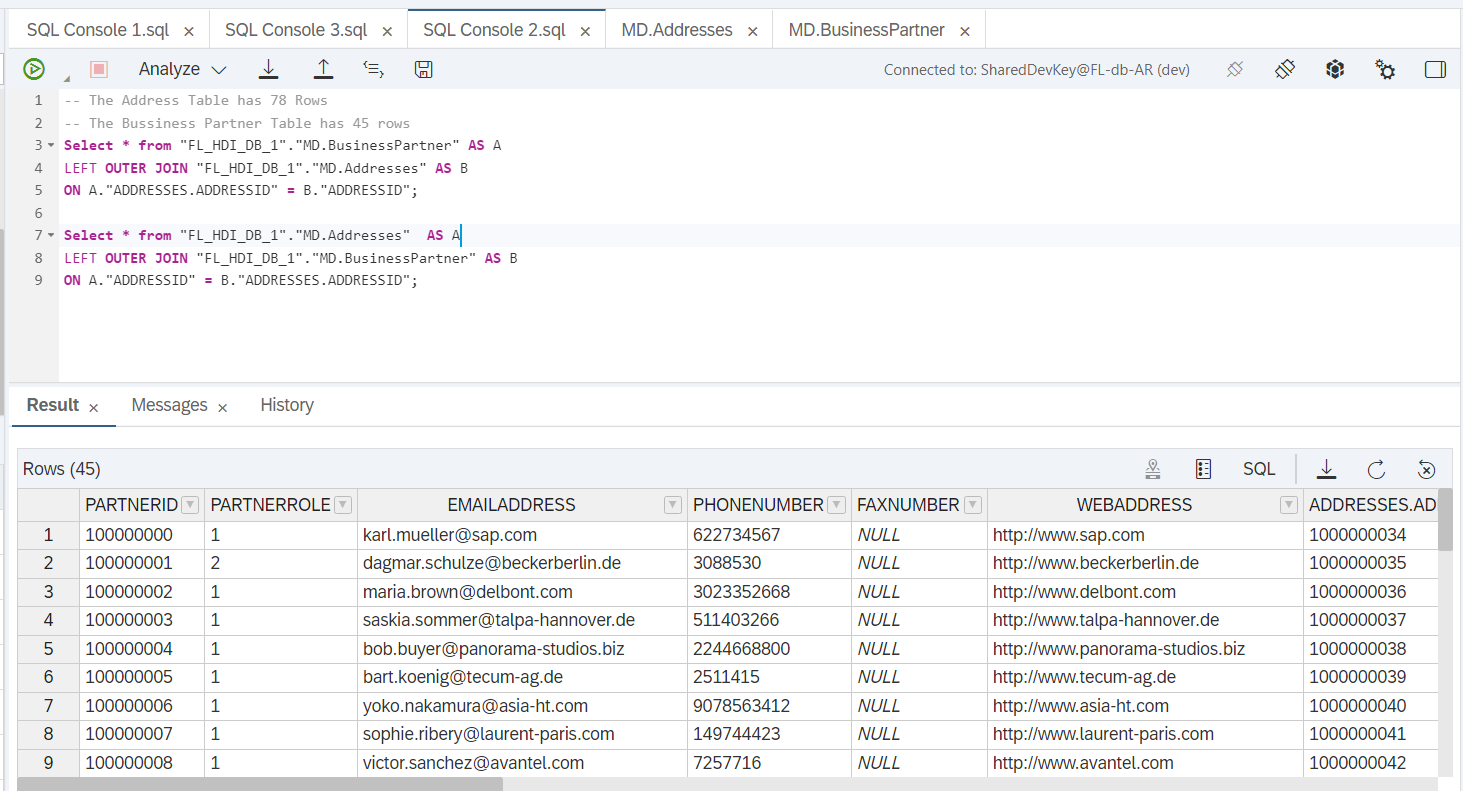
In sql querry we cannot give the cardinality depending on the data it takes the cardanility

In this case for one partner we have multiple product to it



## Left Outer Join

Left outer join means we want all the data from the left table weather or not it have matching records in the right table



Here in this case left table has 45 rows so in the output it is displaying 45 rows and vice versa in 2nd Querry we have the left table as Address table it have 78 rows so in the output it is showing us 78 rows in output.

## Right Outer Join

In this case of right outer join we need to take the whole data from right table whether there is any matching element or not from the left table

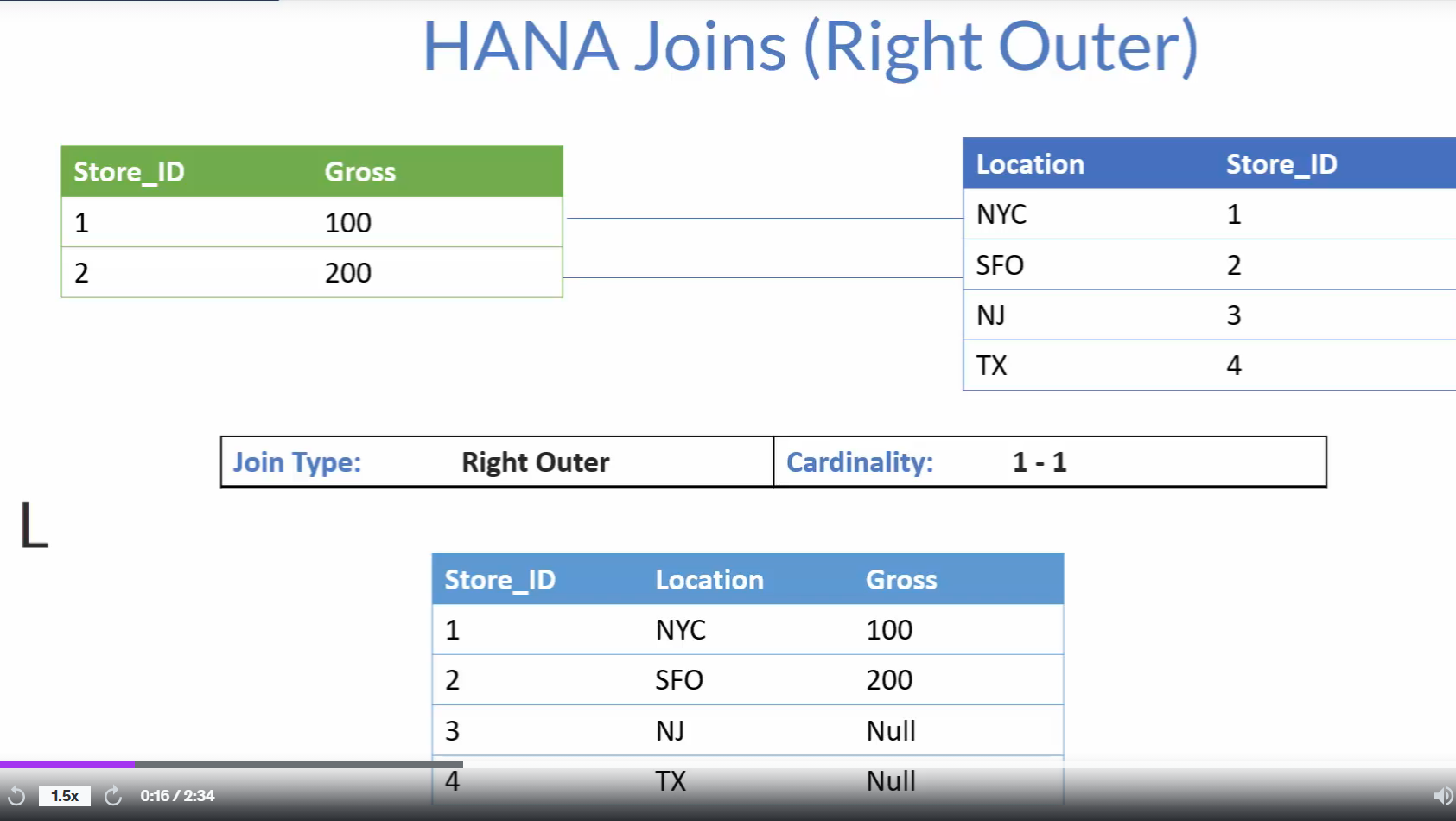


Figure 4 Right Outer Join.

## Full Outer Join

In Full Outer Join the data from both tables gets fetched that means it the left table doesn’t have any common elements from the right table it will also get fetched and vice versa if the right table doesn’t have any common elements from the left table even though it will fetch the entire data common data as well.

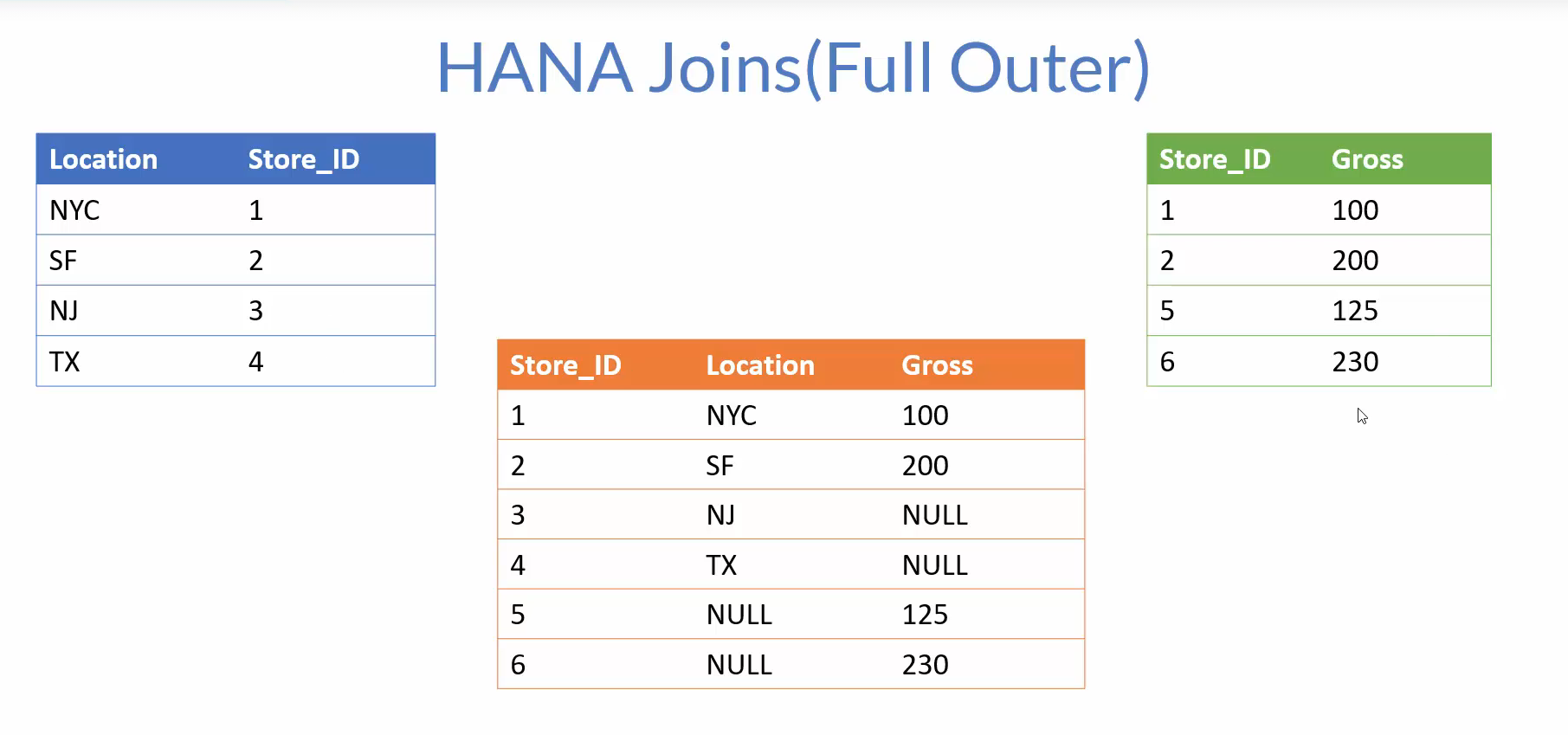
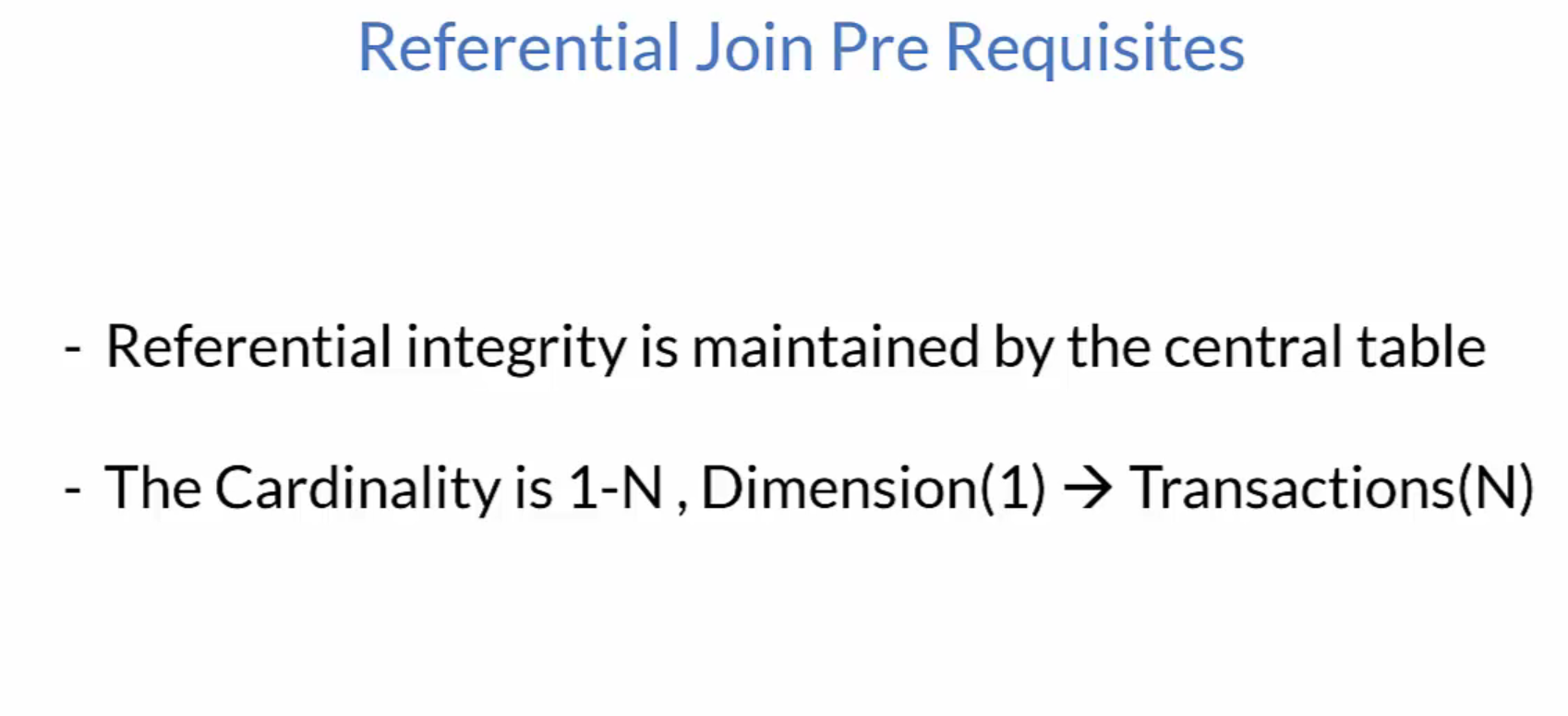


Figure 5 Full Outer Join Example.

Even we swap the position of tables we will still get the same result.

## Referential Join

They were introduced in the hana to improve the performance this are fundamental inner join with some special characteristics.

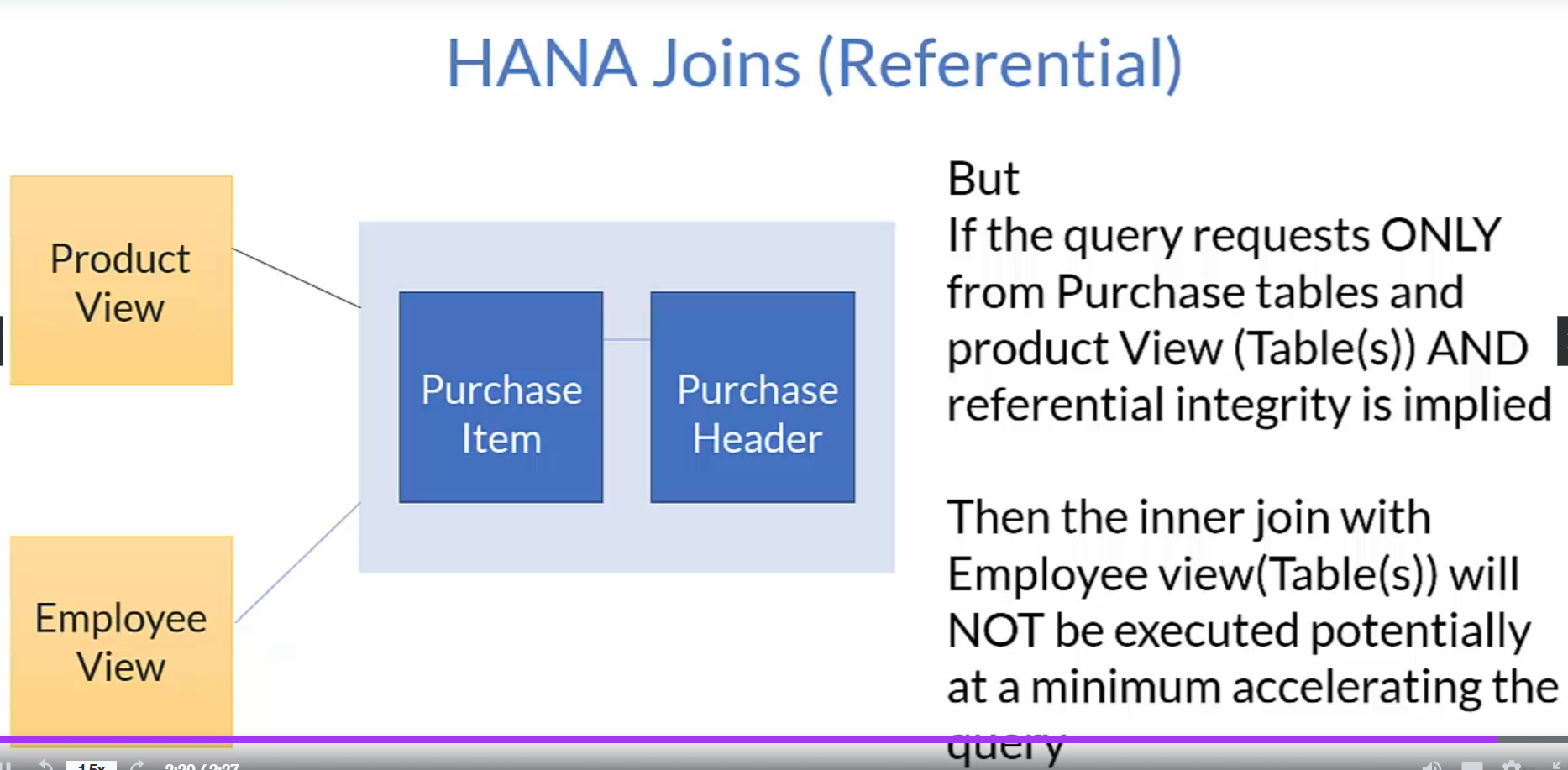


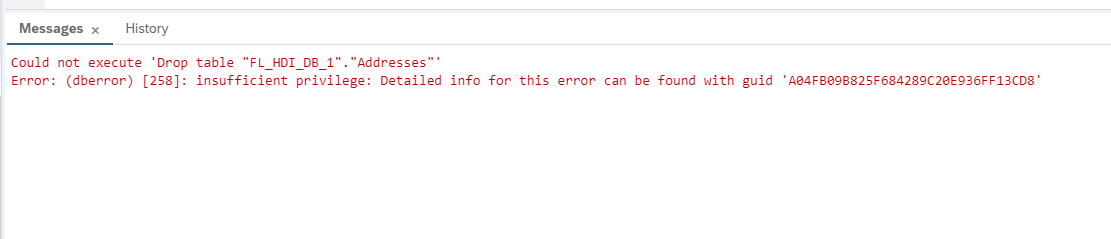
The central table is responsible for Referential Integrity



Figure 6 Scenario of referential Integrity.

If we request data from all three things that are Product view, Employee View, and Cube(Purchase Item, Purchase Header) it performs an inner join between all three things but in case we are requesting data from Product view and Cube it will run inner Join between Product view and Cube only it will increase the performance of our query.





If this error is coming then you need to delete the hdbtable first from the BAS then it will automatically reflect here in the data explorer.

## Creation of Calculation View

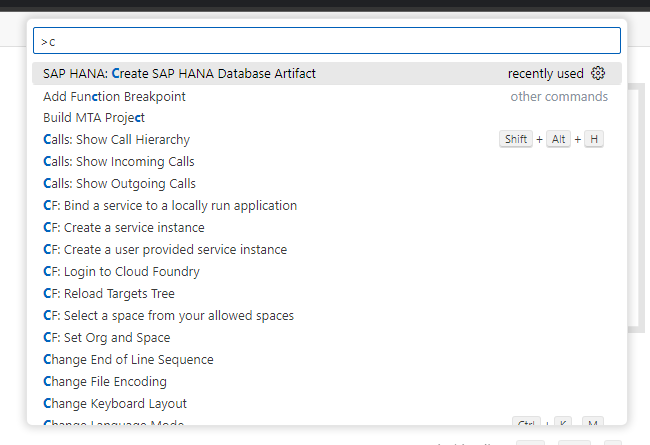


Figure 7 Navigating to Console.

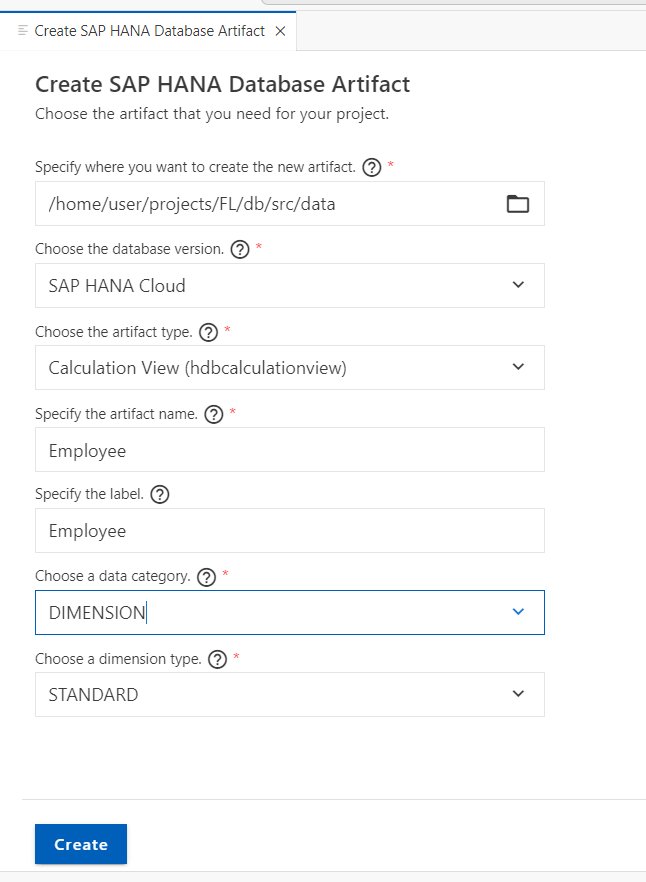


Figure 8 Selecting the Calculation View and Dimension.

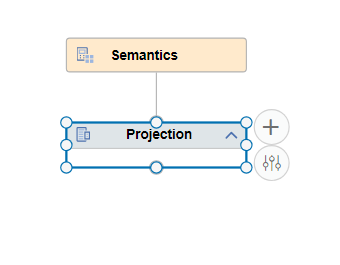


Figure 9 Click on Add-to-add Table to Calculation View.

A screenshot of a computer

Description automatically generated

Figure 10 Adding it to View.

A screenshot of a computer

Description automatically generated

Figure 11 Drag the marked are for mapping.

A screenshot of a computer

Description automatically generated

Figure 12 You can just drag and drop it in output column.

A screenshot of a computer

Description automatically generated

Figure 13 Selecting multiple by using Shift and select and then add to output.

A screenshot of a computer

Description automatically generated

Figure 14 By double clicking you can add to output.

A screenshot of a computer

Description automatically generated

Figure 15 For removing single output Column.

* If you click on Remove mapping it will remove entire from data source also.
* Remove All Mapping Will Remove all the fields from the output column as well as source.
* Removing the output column will only remove the fields from the output column.
* Removing all the output columns will remove all the fields from the output column.
* We can also select the output column by shift + click multiple fields and right click and Removing Output Column

A screenshot of a computer

Description automatically generated

Figure 16 Assigning the Key.

A screenshot of a computer

Description automatically generated

Figure 17 We can double click and change the name of output field.

We can select multiple fields by using shift + click in sequence.

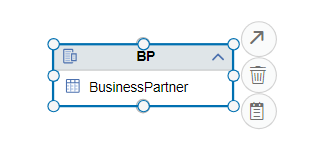


Figure 18 Click on the arrow and drag it to join.

A screenshot of a computer

Description automatically generated

Figure 19 Dragging it to join.

A screenshot of a computer

Description automatically generated

Figure 20 For creation of join drag the arrow and drop it to appropriate node.

A screenshot of a computer

Description automatically generated

Figure 21 If you want to swap the two node you can use them give the cardinality assign the type of join.

### Creation of Calculated Column.

A screenshot of a computer

Description automatically generated

Figure 22 Go to calculated column and click on add.

A screenshot of a computer

Description automatically generated

Figure 23 Giving name to calculated column and using Expression Editor.

A screenshot of a computer

Description automatically generated

Figure 24 After Adding the expression validate the Syntax.

A screenshot of a computer

Description automatically generated

Figure 25 If syntax is right, you will get the prompt.

A screenshot of a computer

Description automatically generated

Figure 26 Syntax to Concatenate.

A screenshot of a diagram

Description automatically generated

Figure 27 Hybrid Calculation View using Product table and Bpartner View.

### Filter

Restricting the view to contain only the records from EMEA Region

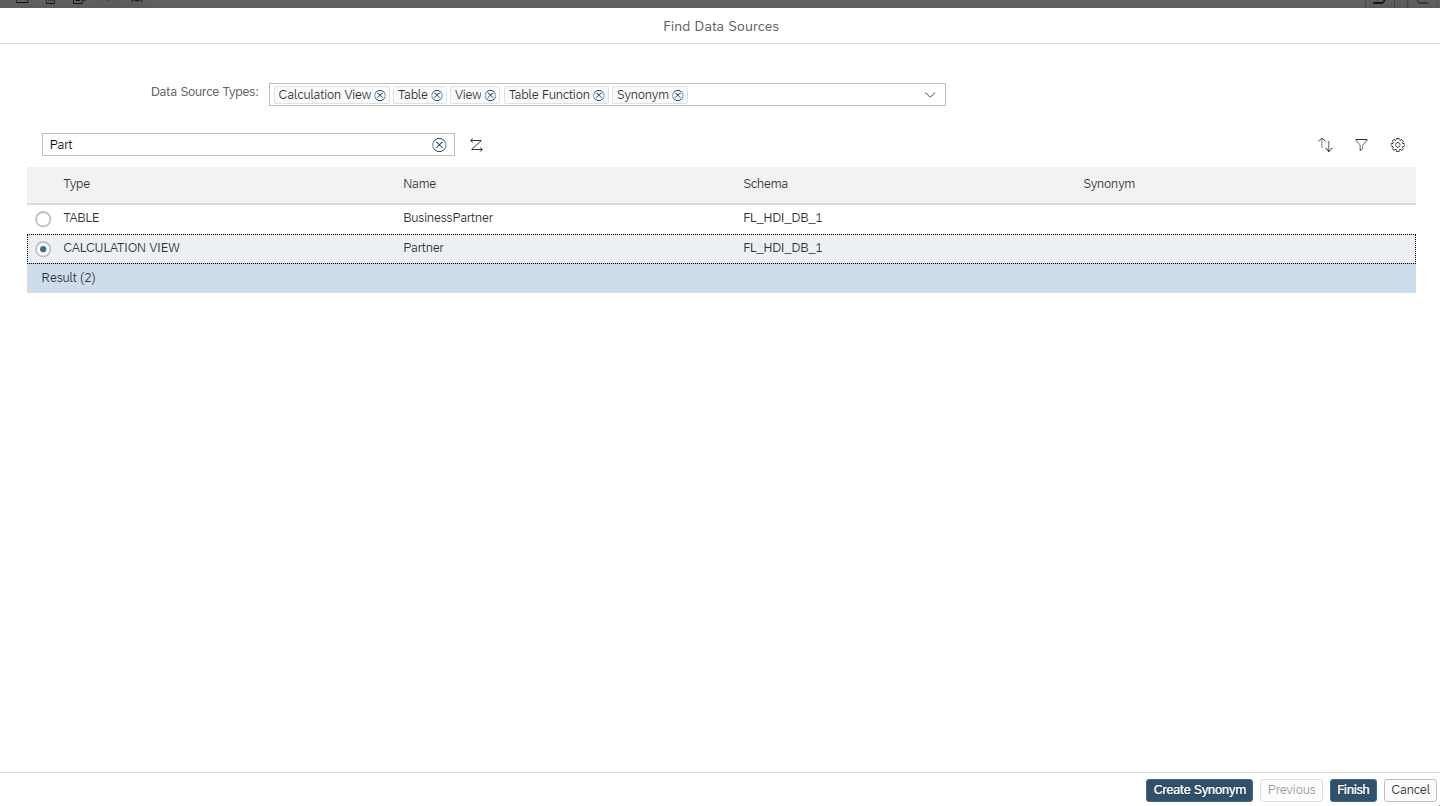


Figure 28 Using Existing View in Projection.

A screenshot of a computer

Description automatically generated

Figure 29 Adding Filter Expression to restrict the data.

A screenshot of a computer

Description automatically generated

Figure 30 Verify the No of entries in Data Preview and in Explorer.

A screenshot of a computer

Description automatically generated

Figure 31 In both the cases the number of entries are same.

### Calculation View Cube with star join

A blue squares with white text

Description automatically generated

Figure 32 Creation of Purchase Cube.

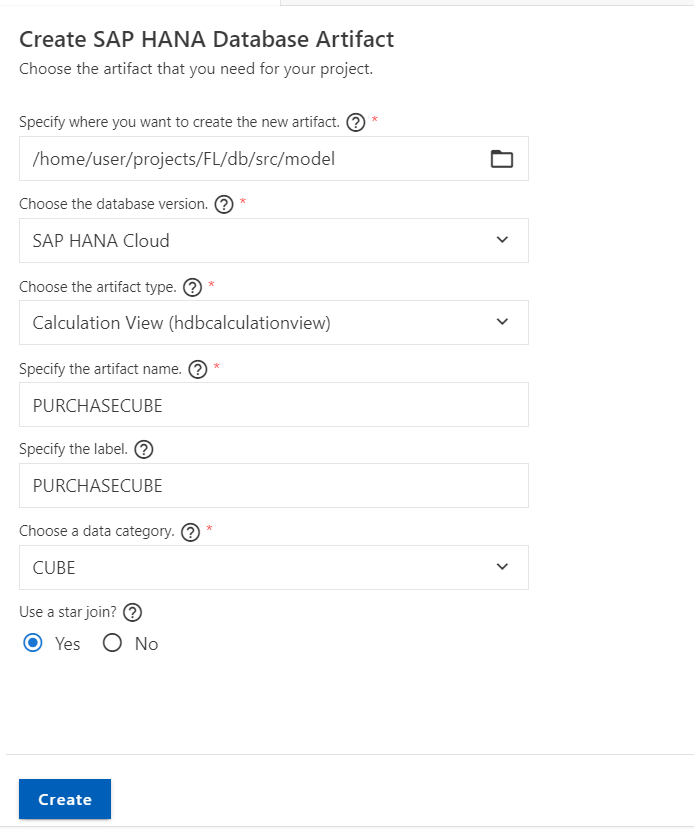


Figure 33 Creating a Calculation View with Star Join.

We will firt connect Po header and po item projections

A screenshot of a computer

Description automatically generated

Figure 34 Header to item level data it should be inner join and 1-n cardinality as one value in header may have multiple entries in item.

Now we are going to add 2 dimension that we have created for Employee and partner

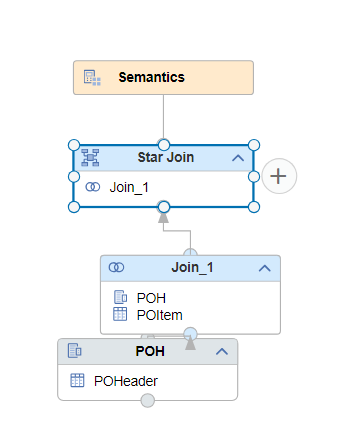


Figure 35 Click on plus symbol.

A screenshot of a computer

Description automatically generated

Figure 36 Adding this to Star Join.

Now we will use referential join it will be n-1 Join 1 will be n and header will be 1 that means n towards the fact table.

As per observation in case of referential join we can give different cardinality between different tables.

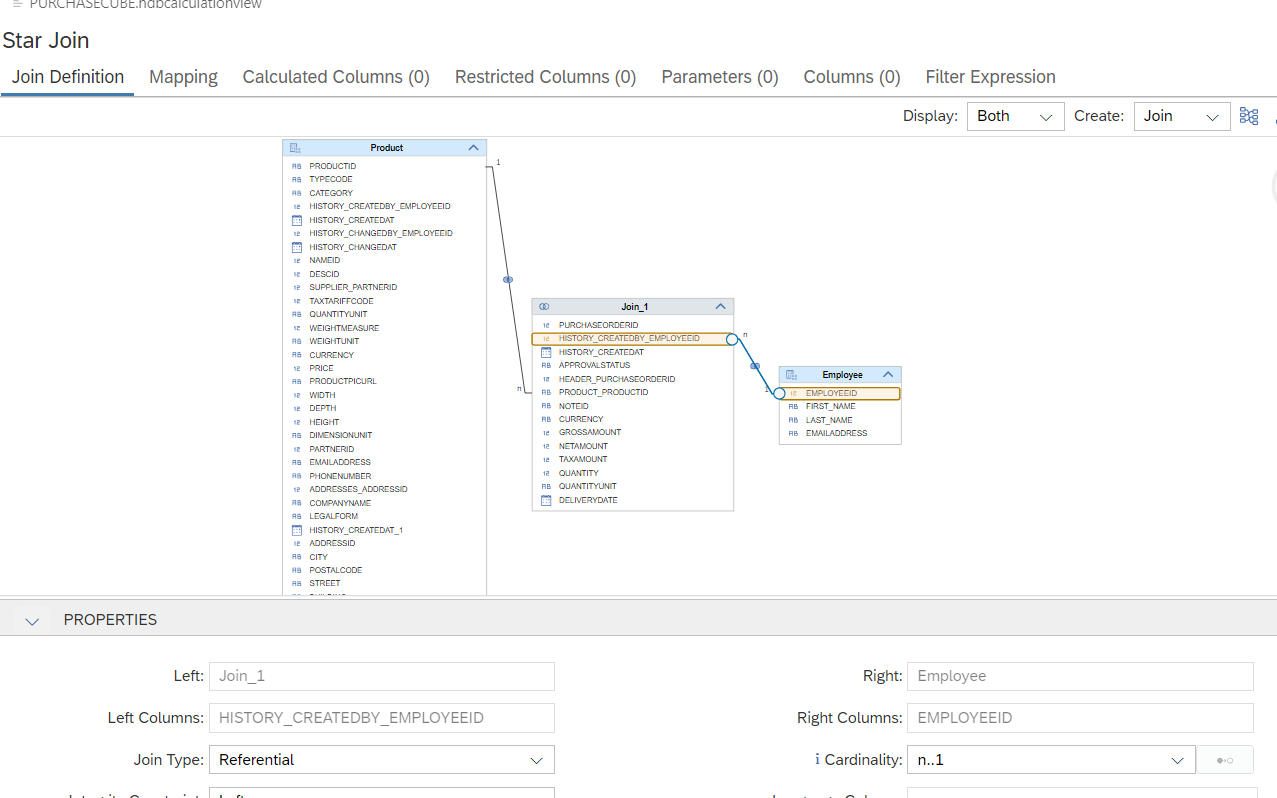


Figure 37 Creating Cardinality for star join between the tables.

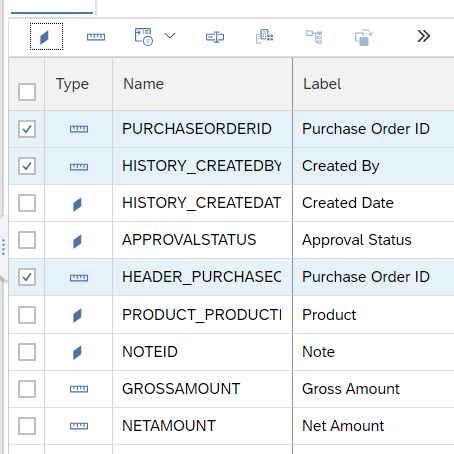


Figure 38 We can assign them as dimension also.

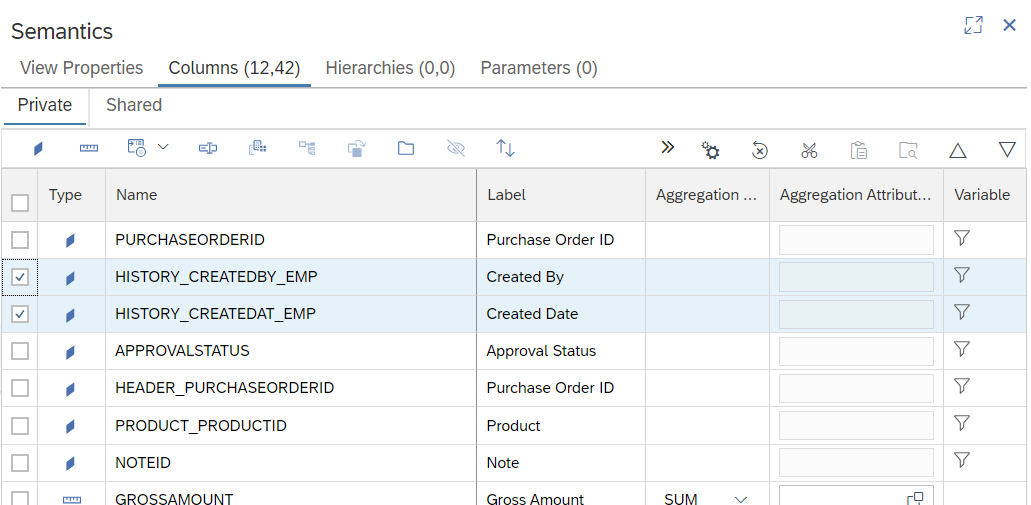


Figure 39 We can move the output column up and down as per our need.

Some of the column especially that are coming from dimensions go in the shared tab why?

A screenshot of a computer

Description automatically generated

Figure 40 Always keep the measure in Value Axis 1 and Dimension in Label Axis 1.

### Show Linkage

It will drill down till the base table from where the element is actually coming from.

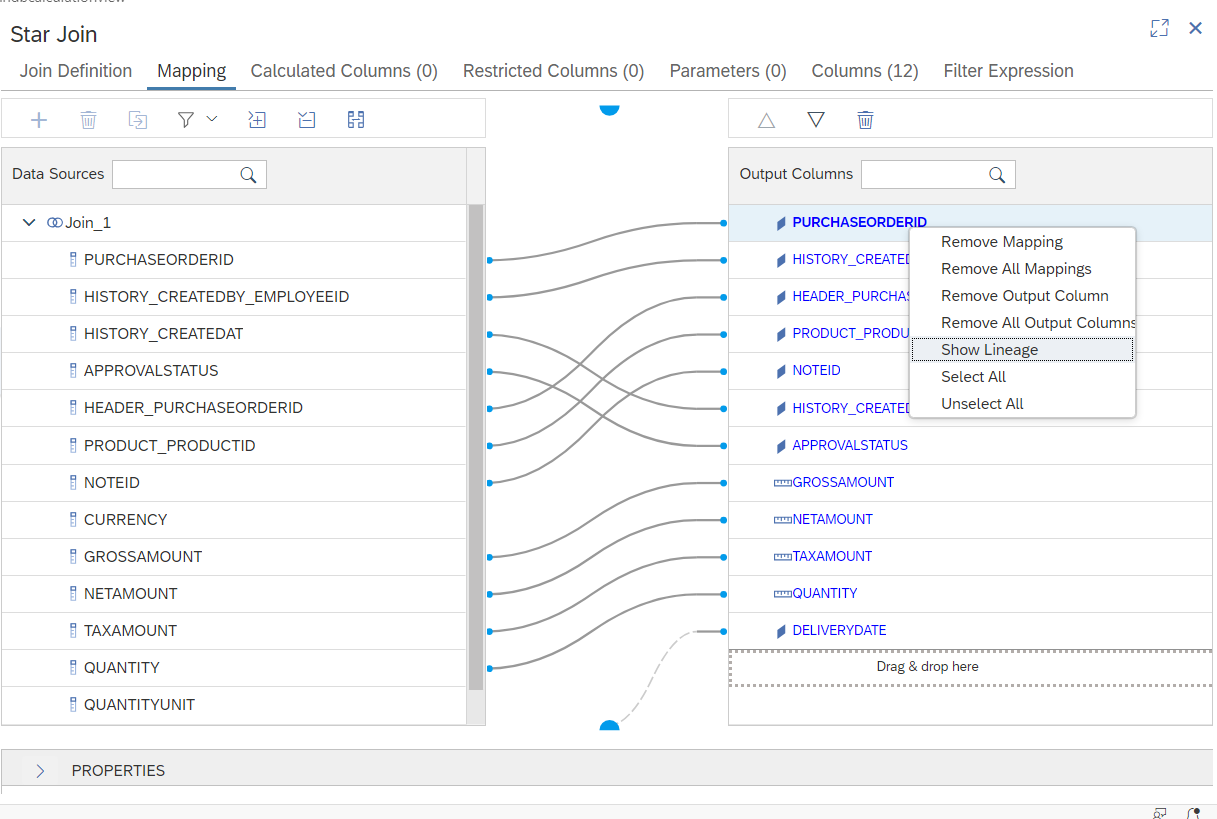


Figure 41 Linkage is always in Mapping it will drill down the base table.

A screenshot of a diagram

Description automatically generated

Figure 42 It highlights in this manner.

### Semantics Layer

Semantics Layer is a very important node because here you get to decide what the reporting layer will actually be seeing from the view so this layer which can be made report friendly.

In other words, this layer can be used to present metadata and data in a fashion which the consumers of the report will want to see in

We have semantics node as a mandatory node whether it is cube type of calculation view or dimension type of calculation view

A screenshot of a computer

Description automatically generated

Figure 43 Section under security.

Default Client: - Which has to do with filtering the data based on the session client

A screenshot of a computer

Description automatically generated

Figure 44 These properties here have to do with the optimization.

Need to find.

1. Static Cache

The static cache allows the query results of calculation views to be cached in memory. This avoids having to re-evaluate the view each time the view is accessed. By using cached results for frequent queries, you may be able to reduce memory and CPU consumption and increase performance.

Because the cache is refreshed only periodically, the cached data might be stale if the tables used by the view were updated since the last refresh. However, the data retention period that determines for how long the cached data is reused before a refresh occurs is configurable, allowing you to determine the maximum acceptable age of the data snapshot.

To keep the cache size to a minimum, you can select a subset of the columns to be cached (by default all columns are cached). In addition, you can further reduce the cache size by adding frequently used filters.

When implementing cached views, you must test them carefully to validate that the queries use the cached data and that the correct results are returned. You should apply caching only on processed and reduced data (topmost view). It should not be applied on raw data.

1. MDS Cubes
2. Snapshots

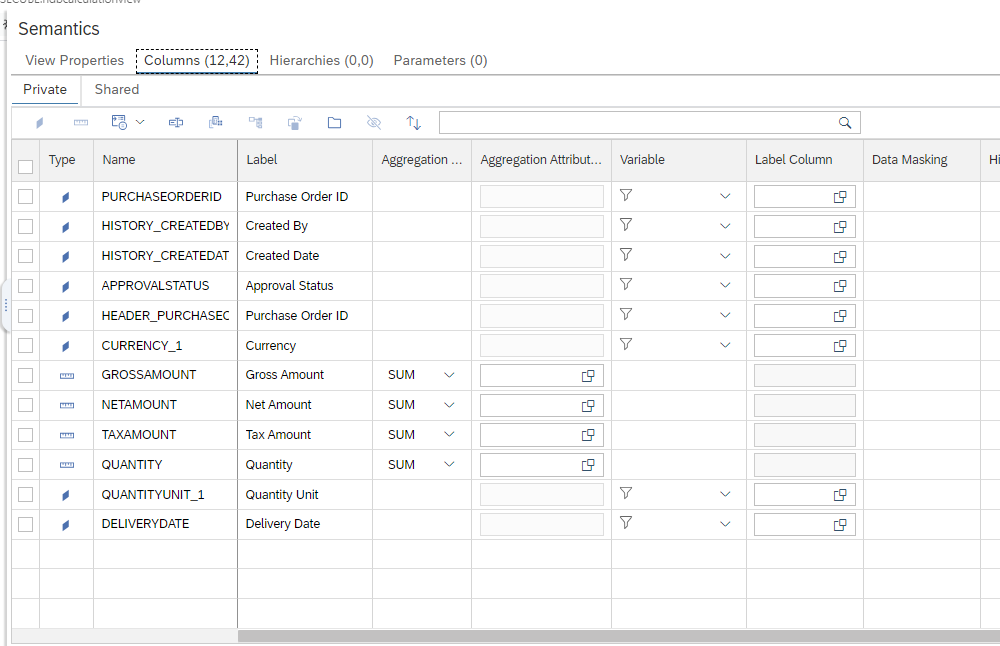


Figure 45 In private we have all the tables that are coming from central table.

A screenshot of a computer

Description automatically generated

Figure 46 While in case of Shred data is coming from the Dimensions.

#### Assigning Semantics

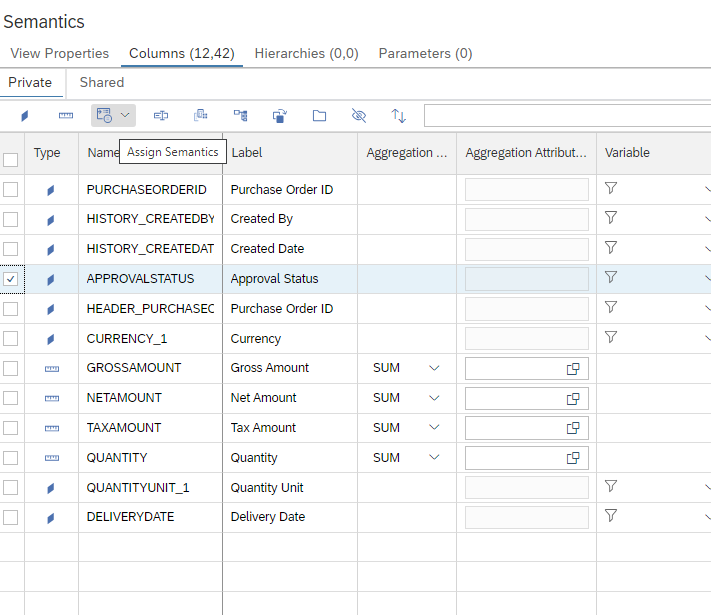


Figure 47 We can assign semantics to attribute.

A screenshot of a computer

Description automatically generated

Figure 48 We can assign semantics to attribute.

A screenshot of a computer

Description automatically generated

Figure 49 Assigning Semantics for measure.

#### Show Lineage from Semantics

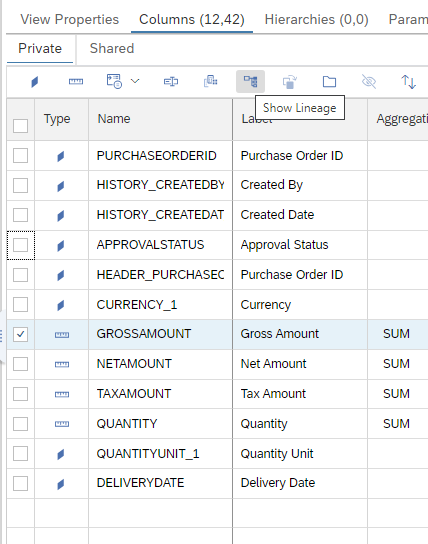


Figure 50 Showing lineage of the field.

A screenshot of a computer

Description automatically generated

Figure 51 Showing the from where it is coming.

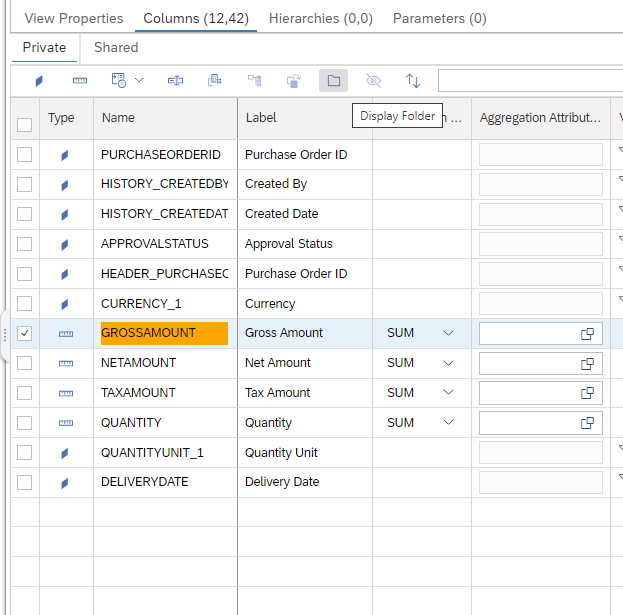


Figure 52 Display folder that will be visible to Reporting layer.

A screenshot of a computer

Description automatically generated

Figure 53 We can differentiate measure dimension as per our preference.

#### Data Masking

This is basically to create a abstraction in your column information for security reason and authorization purposes so you can block a certain part of your column information

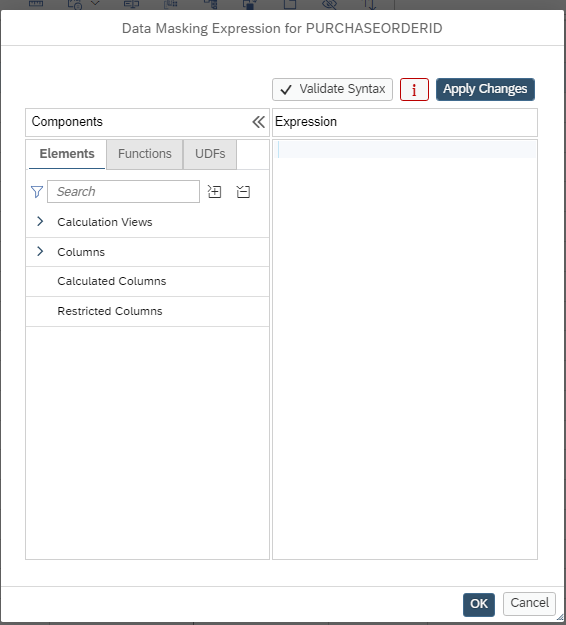


Figure 54 Data Masking.

A screenshot of a computer

Description automatically generated

Figure 55 Customize column display we can customize as per our need.

A screenshot of a computer

Description automatically generated

Figure 56 Different Options.

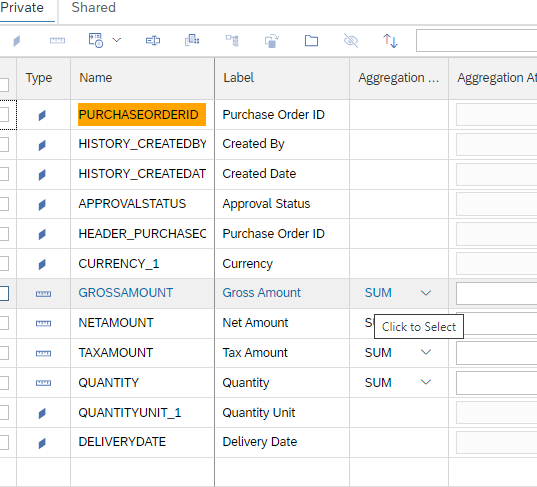


Figure 57 Label is visible to reporting layer.

Label Column: - It is a column to service the reporting layer again where the column will give you description

### Nodes

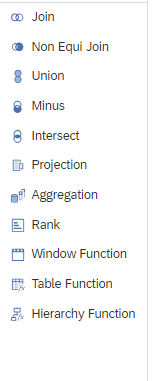


Figure 58 Different Nodes Available.

#### Projection Node.

A screenshot of a computer

Description automatically generated

Figure 59 Projection Node.

Calculated Column: - A Column which can be built on top of original column(s) in the view. It can also be used independently for eg to display staticvalues , set flags and so on



Figure 60 Creating Calculated Column.

We can create calculated columns , parameters and Filter Expression at the projection level.

#### Join Node

We create the join in the join definition which type of join cardinality we also have calculated column parameters filter expression at this stage

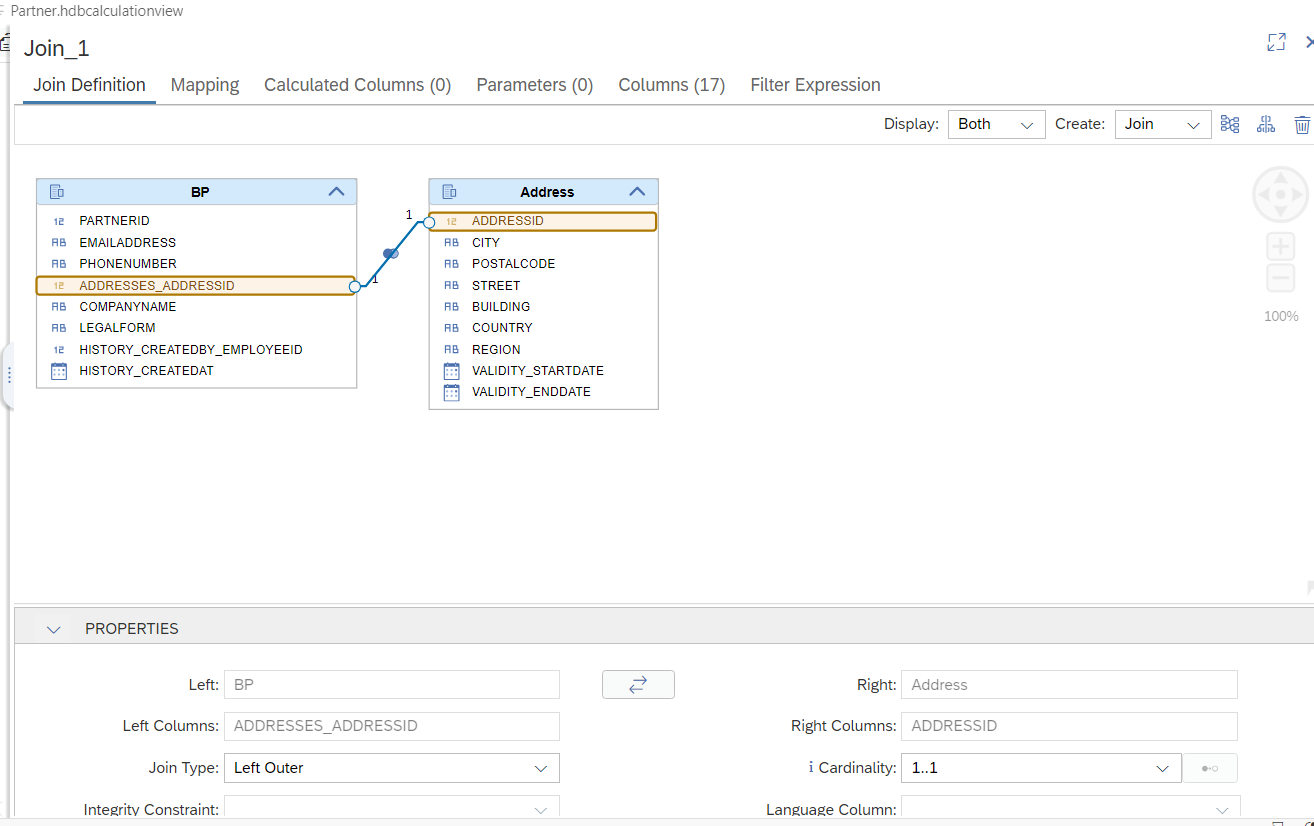


Figure 61 Join Node.

#### Star Join Node.

It also has Join Definition as first node mapping calculated column Restricted Column, Filter Expression and Parameters we have extra feature of restricted columns in Star Join Node.

Restricted Columns: - You can restrict the data based on a tuple value .

##### Restricted Columns

* Used to selectively service different user base.
* Restricting data based on the row level information.
* Can be done on one column or multiple.
* Can you multiple operators.
* Can also use expression editor.

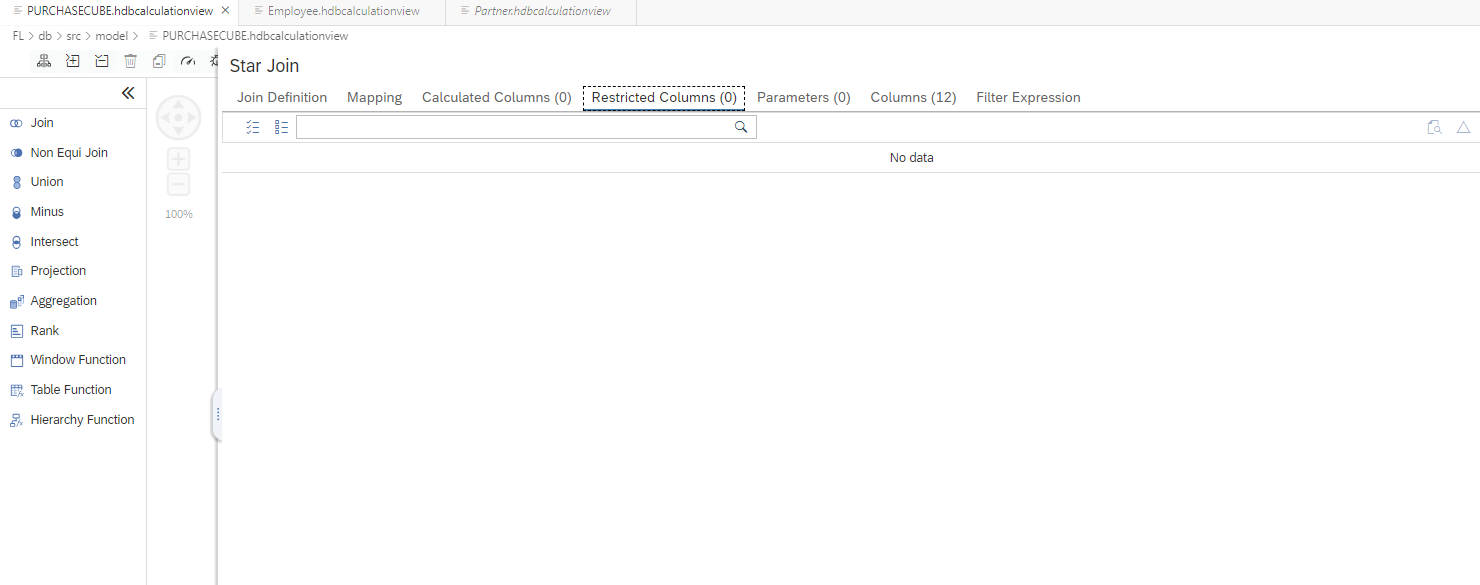


Figure 62 Navigate to restricted column.

A screenshot of a computer

Description automatically generated

Figure 63 Add a new restricted column.

A screenshot of a computer

Description automatically generated

Figure 64 Can also restrict using Fixed value or input parameter.

Include it means that only get to see the data with DE and not include (not marked means we will get except) that data.

We can also add the same column and different value and different column and different value.

A screenshot of a computer

Description automatically generated

Figure 65 Adding Multiple Column.

A screenshot of a computer

Description automatically generated

Figure 66 Can also use expression.

A screenshot of a computer

Description automatically generated

Figure 67 Data Preview for filter expression.

# Phase 2

The idea here is importing SFLIGT data in DBADMIN and will create virtual tables on top of that in HDI.

Import the tables to the host or DBADMIN then create virtual tables into this specific HDI

And will do all your ETL building on top of these virtual tables and all the data targets will store in HDI container.

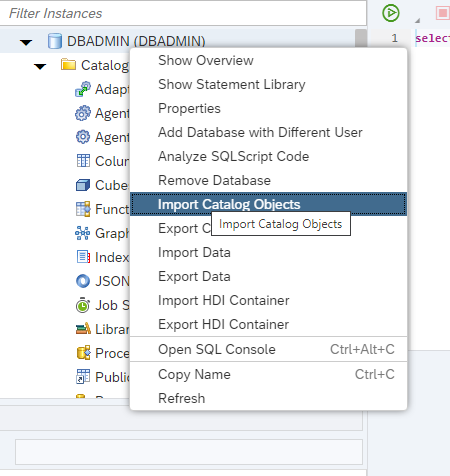


Figure 68 Importing the SFLIGHT data.

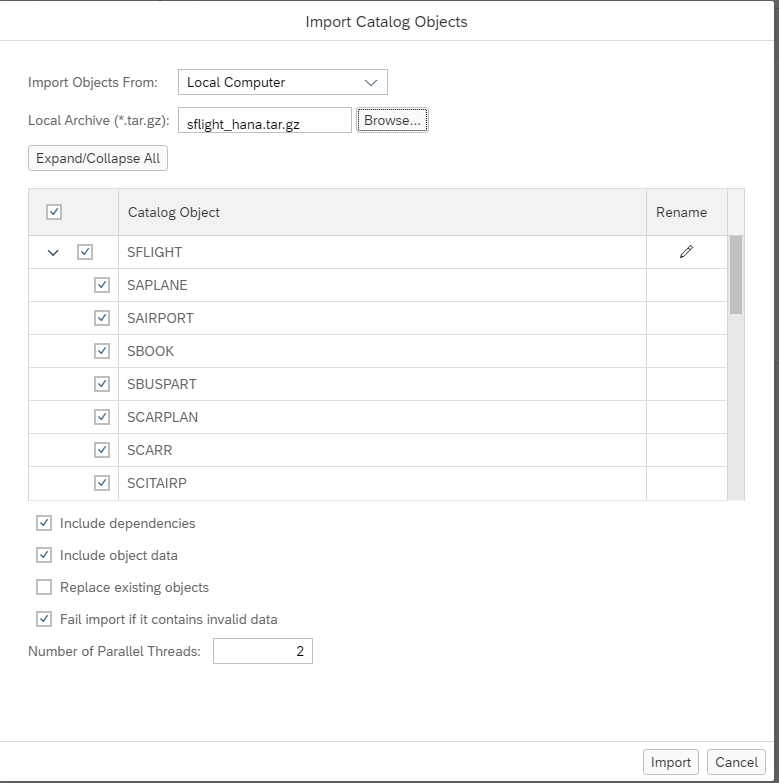


Figure 69 Importing the zip file.

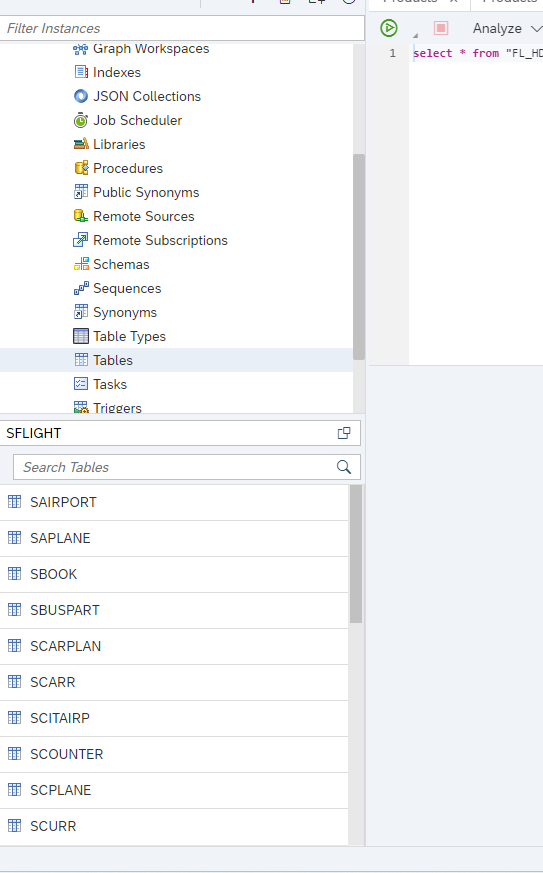


Figure 70 Successfully imported the SFLIGHT data.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated