

CONFIGURE DATA MODEL:

In this exercise you will map previously configured data streams to the Customer 360 Data Model within your Data Cloud instance.

Objectives: In the course of this exercise you will:

- Map ingested data from Data Streams to Data Model entities
- Familiarise yourself with common Data Model entities
- Create new data model entity, additional fields and update/create new data relationships between entities
- Enable value suggestion for selected attributes to improve segmentation experience

Preamble

The key objective of the data model configuration is to maintain and use as much as possible standard data model. This enables future-proofing of the solution and implementation, scalability, interoperability and integrations that are few of the key value propositions of Data Cloud. While customizations are absolutely permitted they can also decrease the standardization of your engagements and ongoing support of the solutions.

A great example of the less desired approach is having an attribute such as **Loyalty Points Balance** without any other relationship to the broader **Loyalty** entity group. The urge might be simply map this

attribute to a custom field on **Individual** and this is not a good practice right from the start.

In the previous exercise you prepared number of custom formula fields across various data sources. Throughout this exercise observe and reflect on how these are used with standard data model entities.

⚠️ In cases where you can't find appropriate attribute in data stream take a step back and revisit data stream configuration, assessing the need for a formula or an additional field to be included from the original data source. ⚠️

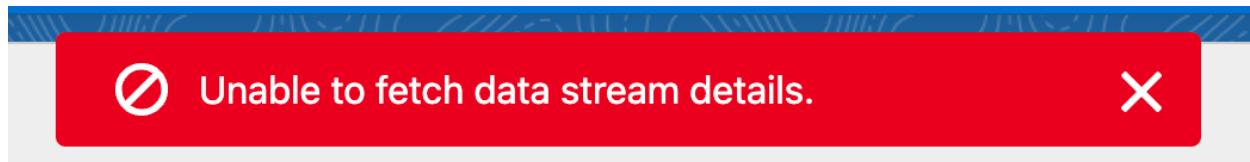
1. Salesforce CRM Data

One of the benefits from using Service Cloud starter bundles is that data streams are automatically mapped into the data model. Its done for customer demographics as well as engagement data sets. While it is certainly helpful always review the mappings to ensure they suit your customer requirements.

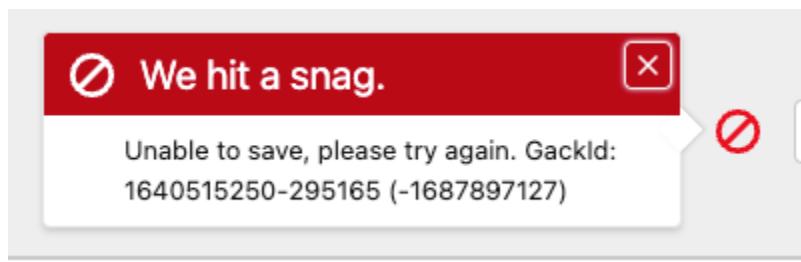
At the same time we will exercise the principal of resource efficiency only mapping data that is required by the customer requirements. This approach enables lean data model, focuses on values that are realized through the defined use cases and prevents the scope creep of validation and testing. The remaining data points can be

added to the model at any time with ease, therefore we are not loosing anything by following with this approach.

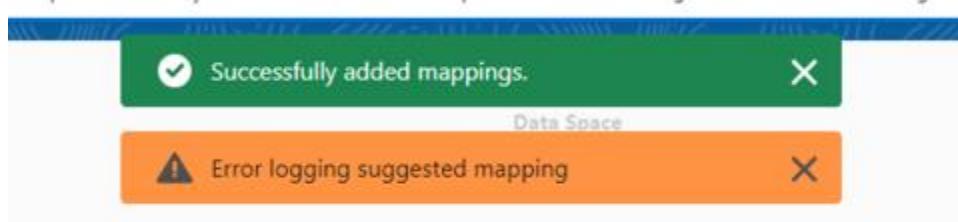
⚠ If you experience the error message “Unable to fetch data stream details” while configuring mappings disregard it and continue with the steps in the guide. ⚠



⚠ Similarly if at any point of saving mapping configuration you experience an error “We hit a snag.” close the mapping dialog, return to the Data Streams/Data Lake Objects tab and retry the mapping steps once again. ⚠



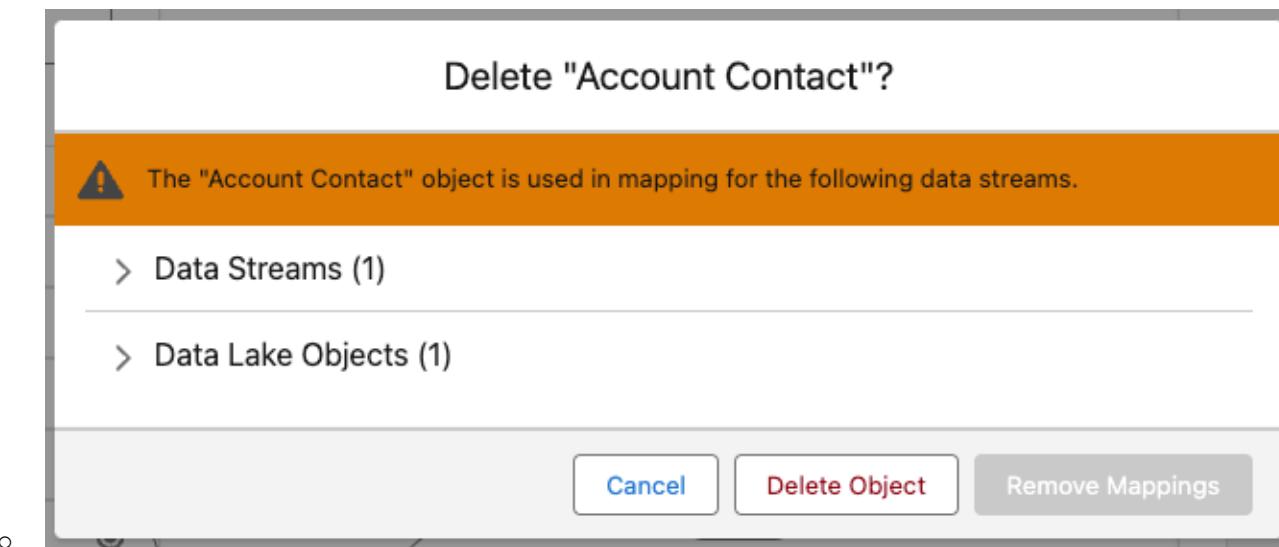
⚠ Even after hitting Save, you may see a yellow warning error but the mapping has been saved. The warning message with *Error logging suggested mapping* won’t impact your subsequent activities. ⚠



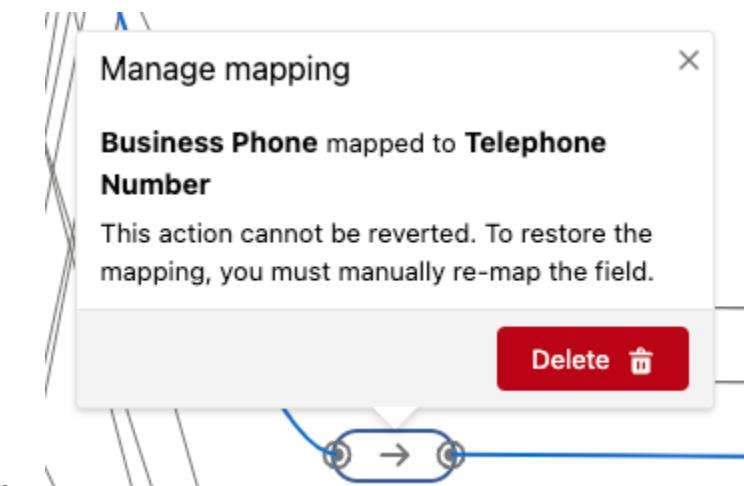
1.1 Contact

While this object is automatically mapped into the data model as part of bundle deploy we need to make few adjustments to the deployed configuration. Please ensure to familiarise yourself with the [Party Data Model](#) and relationships across the objects in the Party Subject Area prior to making any changes in this area.

- In the **Data Streams** locate and open the Contact_<ORG_ALIAS> data stream
- Click Review on the right in the Data Mapping area
- We are not using **Account** or **Account Contact** objects in our solution, therefore remove mappings for **Account Contact** object by clicking on the **Remove Mappings** () icon on the right to its name.
 - Confirm action by clicking on **Delete Object** in the confirmation dialog



- The **Contact Point Phone** object has a warning (⚠) sign indicating that some mandatory mappings are missing. Let's fix that and remap the field that we want to use for this mapping:
 - Delete mapping between Business Phone and Telephone Number by clicking on the “pill” and using the Delete option on the mapping canvas:



- Configure following mapping:

DLO

DLO Field → DMO Field

DMO

Contact	Mobile Phone	→	Telephone Number	Contact Point Phone
	Mobile Phone	→	Formatted E164 Phone Number	

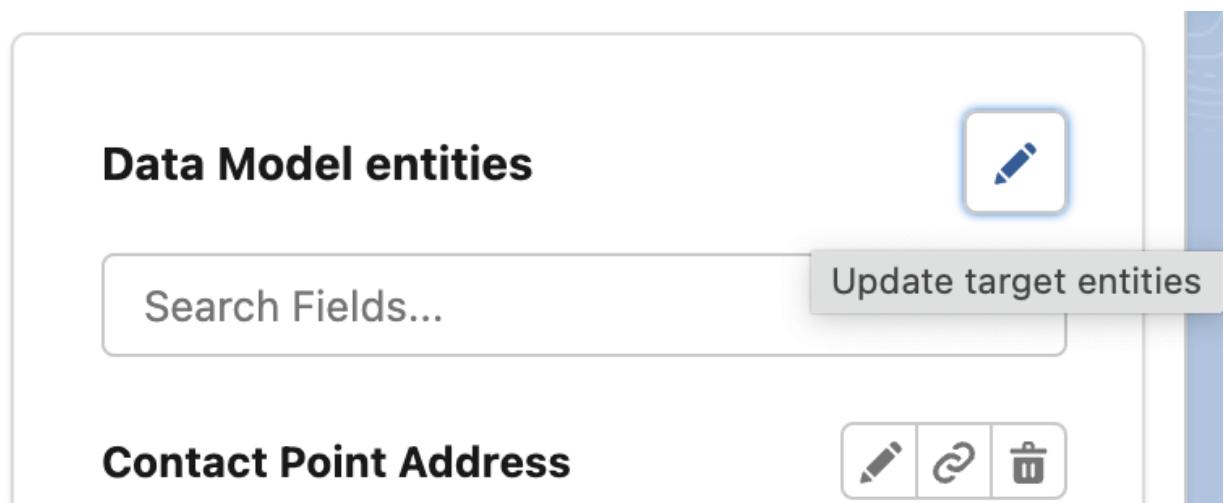
- Save your progress

Without saving at this stage the system will not allow you to remove mappings in the following step as it auto-saves after removal of the field or object mappings

- For the **Individual** object:
 - Delete mapping to Photo URL field
 - Map External ID to External Record Id field
- Save your progress 

There is one key attribute from the Contact object that we need to include into identity resolution process as well as map to the Loyalty subject area DMOs.

- Click on the pen () icon next to the **Data Model entities** to update target entities:



- Add Loyalty Member Currency and Loyalty Program Member
 - If you are not familiar with Loyalty Cloud data model please familiarize yourself with it through the [Loyalty Management Developer Guide](#)

As the system proposes mapping of attributes it matches by name and type you'll notice that for both **Loyalty Member Currency** and **Loyalty Program Member** the system auto-mapped **Created Date** and **Last Modified Date** attributes. We will be removing these.

- Delete mapping of Created Date and Last Modified Date attributes from both Loyalty Member Currency and Loyalty Program Member objects.
- Configure following mappings:

DLO	DLO Field	→	DMO Field	DMO
Contact	Contact ID	→	Loyalty Member Currency Id (primary key)	Loyalty Member Currency
	Contact ID	→	Loyalty Program Member	
	Contact ID	→	Loyalty Program Member Id (primary key)	Loyalty Program Member
	RAVG Loyalty ID	→	Membership Number	

- Save your progress 

 Despite already having a **Party** field on the **Loyalty Program Member** we will add new field to enable customized mapping of the data relations later.

- Click Add New Field under **Loyalty Program Member > Unmapped fields**
 - Field Label = Individual Id
 - Data Type = Text

- Save and map to Contact ID from Contact
 - Save your progress ✓
-  We will be adding one more entity, but have to do it separately from above entities to avoid raising an error due to a conflict with required attribute mapping being removed. If you add that entity along with Loyalty Member Currency and Loyalty Program Member the platform won't allow you to change the Primary Key mapping after you saved the mappings.
- Click on **Update Target Entities** icon () next to the **Data Model entities** title
 - Add Party Identification entity
 - Remove auto-mappings for Created Date and Last Modified Date fields
 - Configure following mappings:

DLO	DLO Field	→	DMO Field	DMO
Contact	Contact ID	→	Party Identification Id (primary key)	Party Identification
	Contact ID	→	Party	
RAVG Loyalty ID		→	Identification Number	

- Save your progress ✓

Adding Formula Fields and Forcing Data Stream Refresh

There is one attribute on the **Contact** object that we did not ingest, but it is required for one of our segmentation use cases. This provides an opportunity to explore how the data ingestion can be updated after initial configuration. It is important to know how the changes to the data stream configuration will be processed by the CRM connector in this case though. As [per documentation](#) adding new formula field **will not** trigger the refresh of the data stream, so we need to ensure that adding new field will. Therefore if we are to just add a formula field the records in DLO will be updated only when corresponding source record is changed in CRM or when full refresh is initiated.

As in our case we need to both add new field and configure a new formula based off it we will have all the data required in place for the further activities in the course.

⚠ If you ingested RAVG Loyalty Points Balance field at the initial ingestion by mistake, choose another field (e.g. Contact Description) to be added as this will ensure that full refresh is initiated. Otherwise in the later exercises you'll be having challenges with segmentation activities. ⚠

- Using instructions on [this help page](#) enable **Read Access** to the RAVG Retail Points Balance field for the **Contacts** object.
- Navigate to the **Data Streams** tab and open **Contact** object data stream
- Click on Add Source Fields action button
- Update the following properties of the field:
 - Field Label = RAVG Loyalty Points Balance
 - Field API Name = RAVG_Loyalty_Points_Balance_c (*keep system suggested value with single _ character*)
- Select that field and save.
- Click on Update Status action button to refresh the metadata of the data stream.



- Click on New Formula Field action button and configure field as follows:
 - Field Label = RAVG Loyalty Points Bucket
 - Formula Return Type = Text
 - Paste the following code into the formula field
 - Using appropriate [functions & operators](#) calculate the value as follows:

- Points Balance \leq 3000 = Low
- Points Balance between 3001 and 5000 = Medium
- Points Balance $>$ 5000 = High
- Points Balance \geq 15000 = Extreme
- Test your formula for following values:
 - 2357 = Low
 - 5000 = Medium
 - 5001 = High
 - 15000 = Extreme
- If you can't make the working formula or getting unexpected results have a look at the [reference below](#).
- Save new field.

* Functional formula for the RAVG Loyalty Points Bucket field:

```
IF(
  sourceField['RAVG_Retail_Points_Balance__c'] <= 3000, "Low",
  IF(
    sourceField['RAVG_Retail_Points_Balance__c'] <= 5000, "Medium",
    IF(
      sourceField['RAVG_Retail_Points_Balance__c'] < 15000, "High",
      "Extreme"
    )))

```

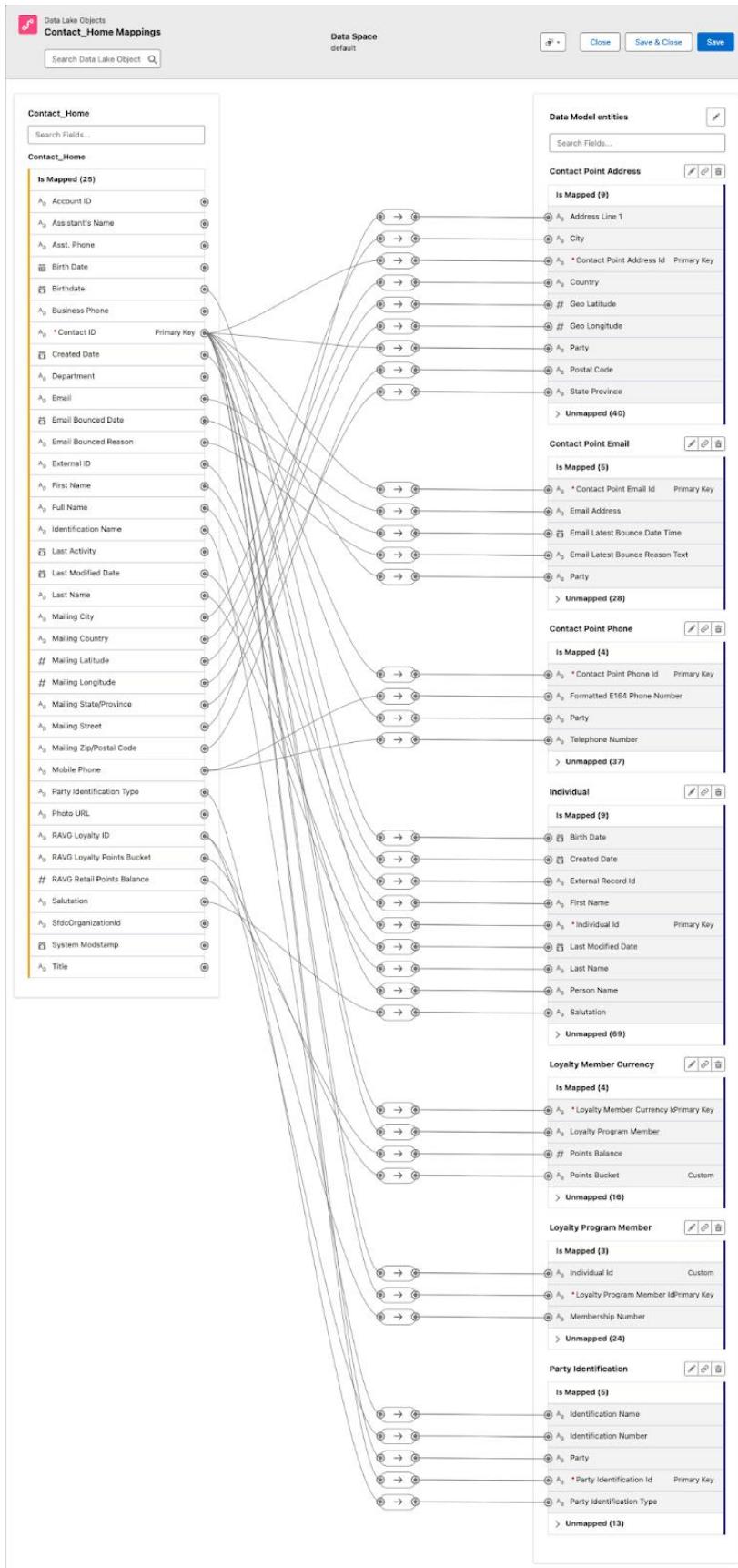
Next map the newly added fields to the data model:

- Click on Review link
- Configure following mappings:

If you've ingested this field during initial stream configuration the name will be RAVG Retail Points Balance. There is no significant difference in the outcome, as data model provides that “translation” layer mapping to the same common destination field, regardless of the source field name.

DLO	DLO Field	→	DMO Field	DMO
Contact	RAVG Loyalty Points Balance	→	Points Balance	Loyalty Member Currency

- Save your progress 
- Click Add New Field under **Loyalty Member Currency**
 - Field Label = Points Bucket
 - Data Type = Text
 - Select Enable Value Suggestion
 - Save and map to RAVG Loyalty Points Bucket from Contact



- Save your progress 

 **What we achieved here?** Let's have a quick reflection on what that model mapping enables:

- **Contact Point Address**, **Contact Point Email** and **Contact Point Phone** represent contact details for an individual across various channels.
- **Individual** entity represents people in the database along with their profile and demographic attributes.
- **Loyalty Program Member** entity will provide details around program membership, such as membership number. It is also an object that joins other entities in the **Loyalty** group to the individual.
- **Loyalty Member Currency** entity contains details around program points. We've included **Points Bucket** custom field here as it logically extends this object.
- **Party Identification** will enable unification of the individuals using specified identifiers.

These entities will enable segmentation on customer and loyalty attributes, such as amount of points available for redemption and unification of customer profiles using name and other attributes.

The mapping to the **Party Identification** object enables configuration of the profile unification for individuals with the matching loyalty profile but different customer identifiers.

With the enabled value suggestion the segmentation criteria that uses that attribute will be quite intuitive and friendly for the end user.

1.1.1 Update Relationships with Loyalty Entity Group

For the workshop scenario we made an assumption that there is only one loyalty program that is maintained by the brand. By default the Customer 360 Data model will configure 1:M relationship from **Individual** entity to support multiple programs and further within the **Loyalty** entity group the relationships would be configured as 1:M.

The consideration for this relationship configuration though is that attributes will be available in segmentation, but will not be exposed for the activation purposes. E.g. if **Membership Number** is required for the personalization within audience segment in the activation target then current configuration won't allow for it to be included with the segment.

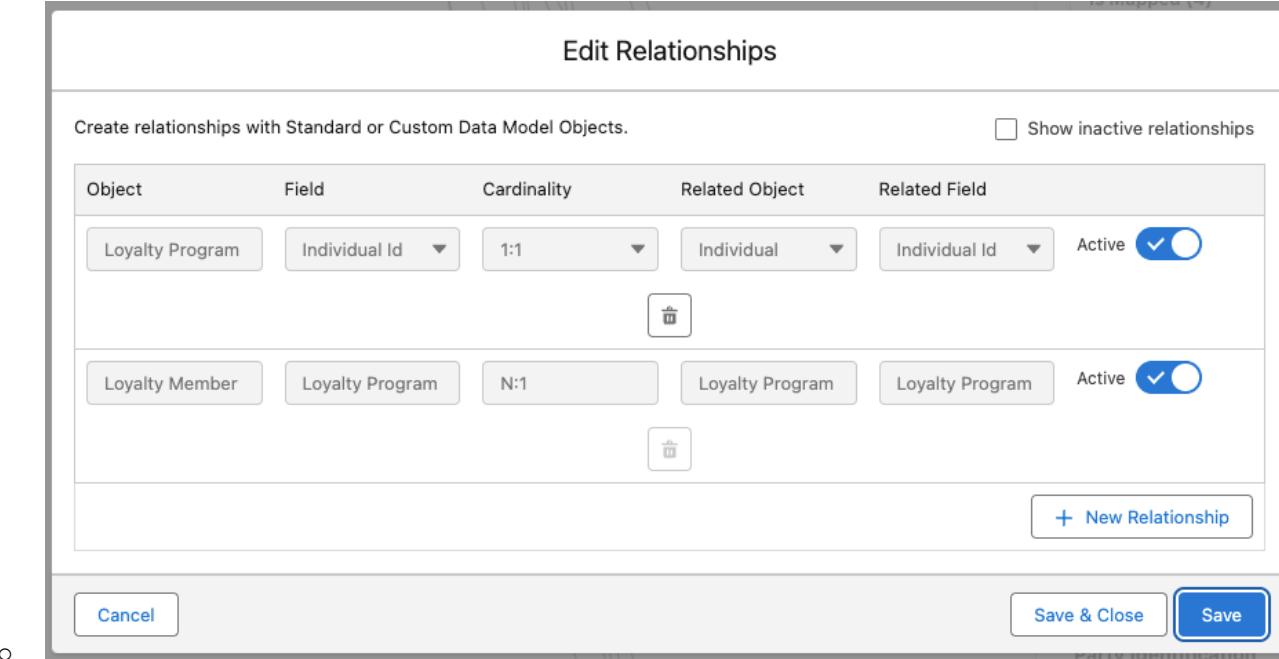
Purely for educational purposes let's "modify" one of the relationships, specifically between the **Individual** and **Loyalty Program Member** from standard 1:M to 1:1. To achieve that you will be configuring new relationship, not really modifying an existing one. If you were to map **Contact ID** to the standard **Party** field on the **Loyalty Program Member** DMO the result would be that standard relationship would've been considered "active" by the platform enforcing 1:M treating of it for the purpose of segmentation and

activation. [This help article](#) provides additional insights around relationships.

⚠ Do take a note, that change like this will require data relationship not being used in segments and activations. Therefore for real implementations if this approach is chosen and will need to be reverted in the future (e.g. new brands with loyalty programs added to the setup) **all segments and activations** that use that relationship will need to be deleted and re-created after modifications. This might be quite a significant undertaking, therefore don't take such a decision lightly. ⚠

Navigate to the **Data Model** tab, locate and open the **Loyalty Program Member** object.

- Click on **Relationships** and then **Edit** button.
- Click + **New Relationship**.
- Add new relationship via Individual Id field
 - Field = Individual Id
 - Cardinality = 1:1
 - Related Object = Individual
 - Related Field = Individual Id
- Click **Save** and observe new relationship configured and it's **Active** status:



Save and close the dialog.

1.2 Case

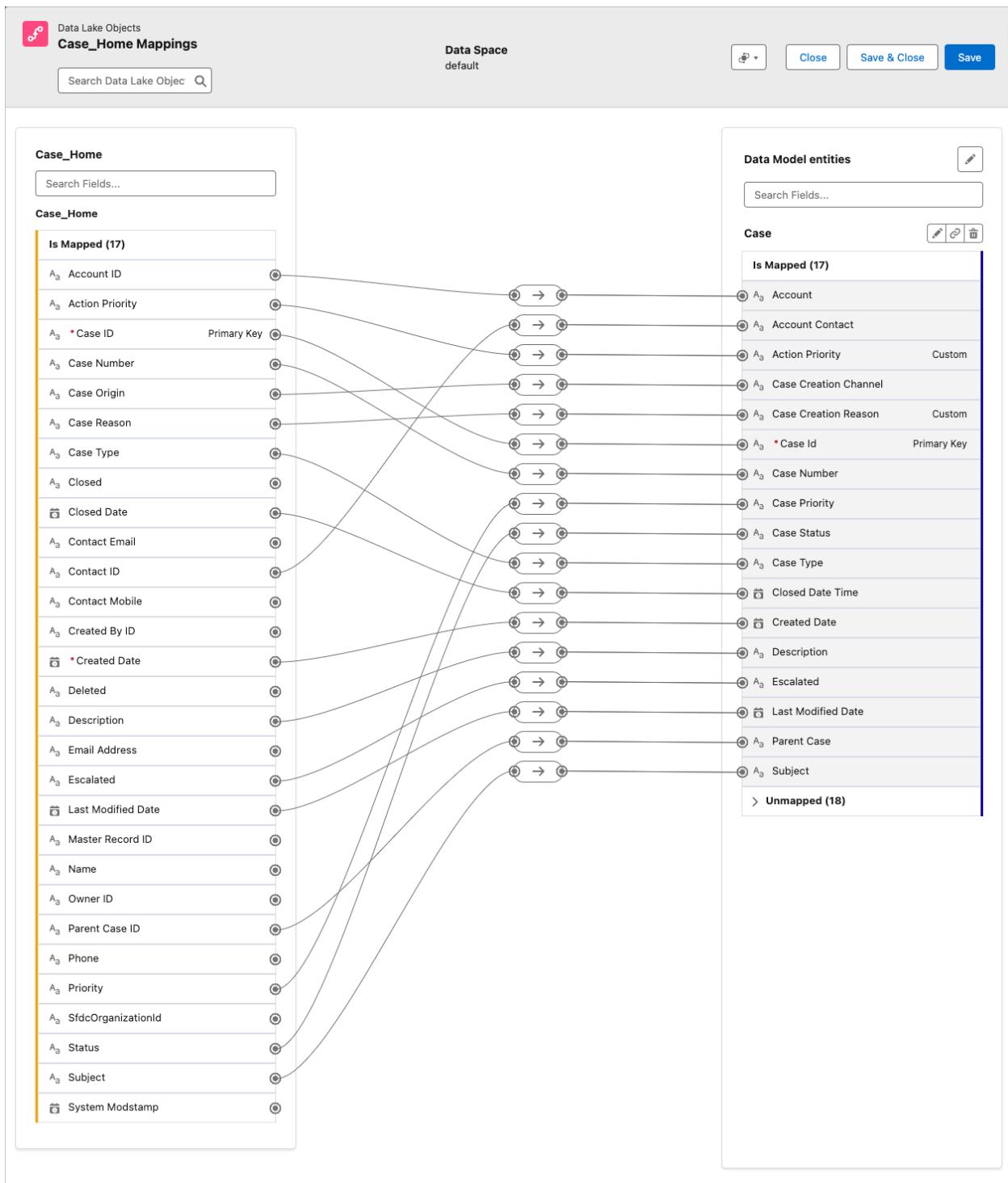
The Case object is automatically mapped to the data model during bundle deployment. Although we have a custom field that will need to be mapped to the data model as custom fields are not automatically configured.

- In the **Data Streams** locate and open the **Case_<ORG_ALIAS>** data stream
- Click Review on the right in the **Data Mapping** section

We'll add few fields before mapping them as every time the field is added the platform will save the model and will raise an error if there are unsaved mappings. This way we'll avoid few saves in between the steps.

- Click Add New Field under **Case** (DMO):
 - Field Label = Case Creation Reason
 - Data Type = Text
 - Select Enable Value Suggestion
 - Save.
- Add another field:
 - Field Label = Action Priority
 - Data Type = Text
 - Select Enable Value Suggestion
 - Save.
- Configure following mappings:

DLO	DLO Field	→ DMO Field	DMO
Case_<ORG_ALIAS>	Action Priority	→ Action Priority	Case
	Case Reason	→ Case Creation Reason	
	Escalated	→ Escalated	



- Save your progress but remain on the same screen

1.2.1 Create Relationship Between Case and Individual

- Click on the **Edit Relationships** icon (🔗) on the right of the **Case**
- Click **Edit** and subsequently **+ New Relationship**
 - Field = Account Contact
 - Cardinality = N:1
 - Related Object = Individual
 - Related Field = Individual Id

Object	Field	Cardinality	Related Object	Related Field	Active
Case	Parent Case	N:1	Case	Case Id	<input checked="" type="checkbox"/>
Case	Account Contact	N:1	Individual	Individual Id	<input checked="" type="checkbox"/>
<input type="button" value="+ New Relationship"/>					

Cancel Save & Close

- Save and close the dialog. ✓

1.2.2 Enable Value Suggestion for Standard Attributes

While we enabled value suggestion on couple of custom attributes, there are few standard attributes for which it will be beneficial to enable value suggestion.:

- Click on the **Edit Properties** icon (📝) on the right of the **Case**
- enable value suggestion for the following attributes:
 - Case Status

- Escalated

Custom Object						
Field Label	Field API Name	Key	Data Type	Enable Value Sugg...	Key Qualifier	
1 Account	ssot__AccountId__c		Text	<input type="checkbox"/>		
2 Account Contact	ssot__AccountContactId...		Text	<input type="checkbox"/>		
3 Action Priority	Action_Priority__c		Text	<input checked="" type="checkbox"/>		
4 Case Creation Chan...	ssot__CaseCreationCha...		Text	<input type="checkbox"/>		
5 Case Creation Reas...	Case_Creation_Reason...		Text	<input checked="" type="checkbox"/>		
6 Case Id	ssot__Id__c	Primary Key	Text	<input type="checkbox"/>		
7 Case Number	ssot__CaseNumber__c		Text	<input type="checkbox"/>		
8 Case Priority	ssot__CasePriorityId__c		Text	<input type="checkbox"/>		
9 Case Status	ssot__CaseStatusId__c		Text	<input checked="" type="checkbox"/>		
10 Case Type	ssot__CaseTypeId__c		Text	<input type="checkbox"/>		
11 Closed Date Time	ssot__ClosedDateTime__c		DateTime			
12 Created Date	ssot__CreatedDate__c		DateTime			
13 Data Source	ssot__DataSourceId__c		Text	<input type="checkbox"/>		
14 Data Source Object	ssot__DataSourceObjec...		Text	<input type="checkbox"/>		
15 Description	ssot__Description__c		Text	<input type="checkbox"/>		
16 Escalated	ssot__IsEscalated__c		Text	<input checked="" type="checkbox"/>		
17 Key Qualifier Accou...	KQ_AccountContactId__c		Text	<input type="checkbox"/>		"Account Contact" Key Qu
18 Key Qualifier Case Id	KQ_Id__c		Text	<input type="checkbox"/>		"Case Id" Key Qu
19 Key Qualifier Parent...	KQ_ParentCaseld__c		Text	<input type="checkbox"/>		"Parent Case" Key Qu
20 Last Modified Date	ssot__LastModifiedDate...		DateTime			
21 Parent Case	ssot__ParentCaseld__c		Text	<input type="checkbox"/>		
22 Subject	ssot__Subject__c		Text	<input type="checkbox"/>		

Back

- Save and close the dialog. 

 **What we achieved here?** Let's have a quick reflection on what we've configured and what that model mapping enables:

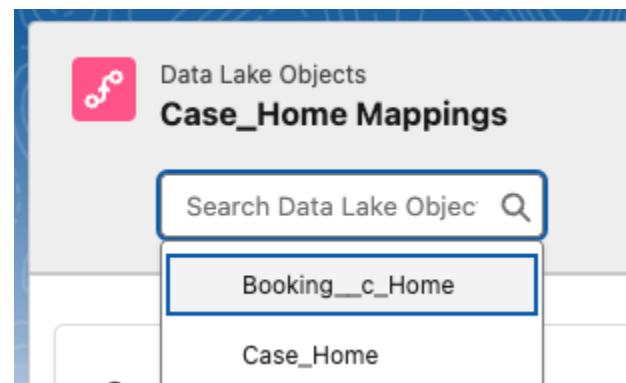
- **Case** entity enables access to the reported challenges experienced by customers with regards to vehicle rentals or booking process. Given the deployment and mapping of the source data was automated we made minimal changes to the configuration, yet still including custom or standard fields that are required for our scenario.

- We extended relationships to include mapping to the **Individual** entity, ensuring that cases are related to individuals despite us not using standard relationship via **Account** entity.
- Lastly we've enabled value suggestion for few attributes. As these represented as picklist or checkboxes in the source data it is safe to configure them as such, resulting in improved user experience during segmentation.

1.3 Booking

The Booking is a custom object in CRM and therefore it will need to be manually mapped to the data model. Even more so, we will need to create a new data model object to connect it to. Assuming you remained on the mapping dialog:

- In the **Search Data Lake Objects**  field locate and open the `Booking__c_<ORG_ALIAS>` data stream



- In the **Data Model entities** on the right click **Select Objects**
- Click **Custom Data Model** tab and then **+ New Custom Object**
 - Object Label = Booking

- Deselect following fields:
 - Created By Id
 - Customer External ID
 - Is Business
 - Is Leisure
 - Owner ID
 - System Modstamp
- Rename label for the following fields:
 - Booking ID to Booking Reference Number
 - Deleted to Is Deleted
 - Record ID to Booking Id
 - SfdcOrganizationId to Internal Organization
- Enable value suggestion for the following fields:
 - Drop Off Location Code
 - Drop Off Location Name
 - Pick Up Location Code
 - Pick Up Location Name
 - Protection Coverage
 - Purpose
- Save the new object 

- The system will map all of the attributes for you as the model was constructed based on data stream. Verify the mapping and adjust if needed for the respective fields:



- Save your progress and remain on the same screen.

⚠ Please ensure that **Booking__c_<ORG_ALIAS>** field Customer is mapped to the **Booking** DMO to the Customer field. If it was not automatically done for you please do it yourself. **⚠**

💡 What we achieved here? Let's have a quick reflection on what we've configured and what that model mapping enables:

- **Booking** custom entity is an example use case where there is no standard data model object, therefore we had to create a net new custom DMO. The process was to use the respective data stream as a source for the desired schema, and with few tweaks we were able to create new DMO and mappings were configured automatically as a result.
- Attributes of this object enable identification of customers who rent vehicles without prior reservation, book using various partner loyalty programs to receive discounts and so forth. For few attributes we've enabled value suggestion ensuring that users get improved user experience and don't need to remember set values used in these fields.

1.3.1 Create Relationship Between Booking and Individual

- Click on the **Edit Relationships** icon () on the right of the **Booking**
- Click **New** and subsequently **+ New Relationship**
 - Field = Customer

- Cardinality = N:1
- Related Object = Individual
- Related Field = Individual Id
- Save and close the dialog. 

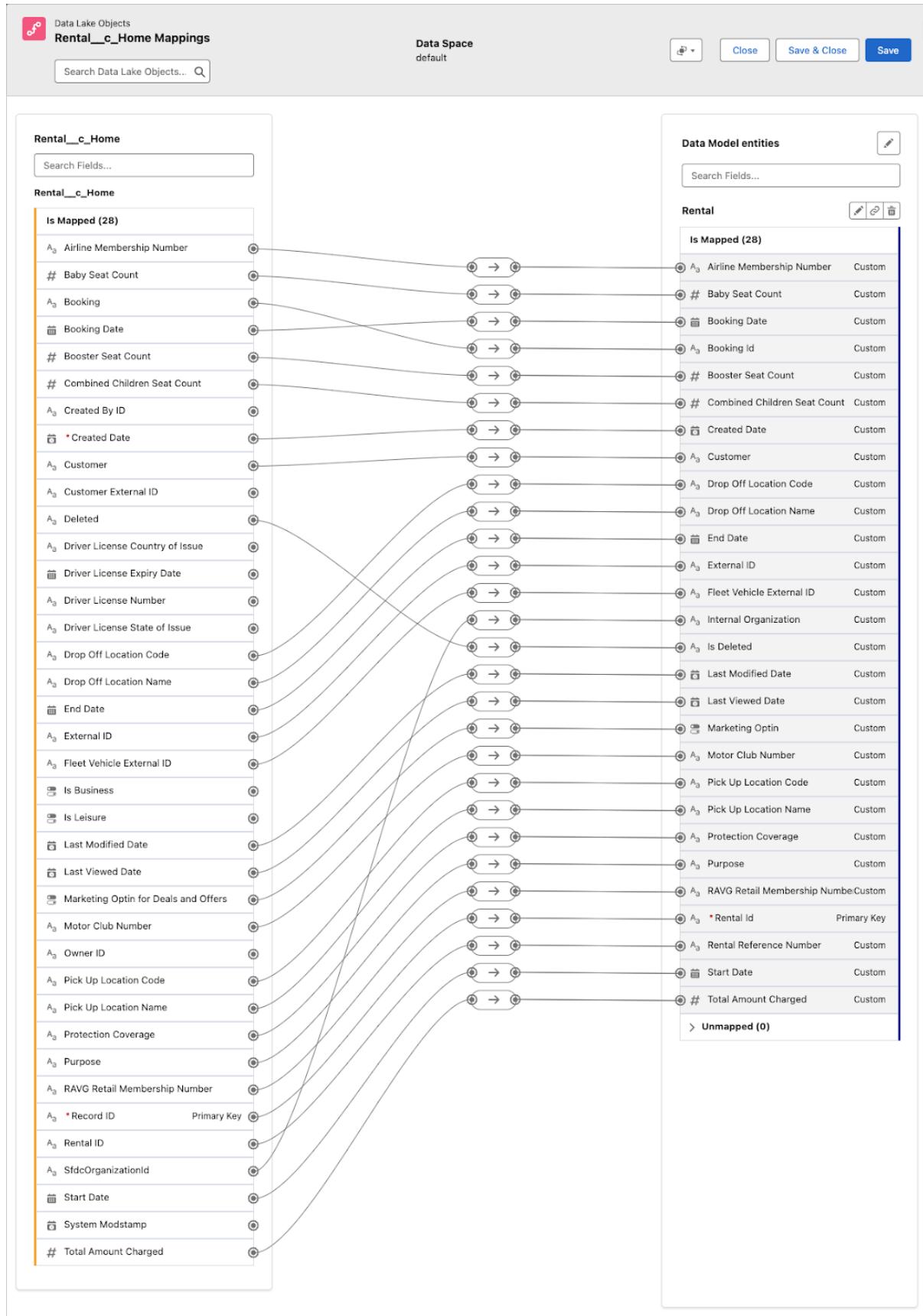
1.4 Rental

Similar to Booking the Rental is another custom object in CRM and therefore we will be doing manual mapping with the creation of the custom DMO.

- In the **Search Data Lake Objects**  field locate and open the Rental_c_<ORG_ALIAS> data stream
- In the **Data Model entities** on the right click **Select Objects**
- Click **Custom Data Model** tab and then **+ New Custom Object**
 - Object Label = Rental
 - Deselect following fields:
 - Created By Id
 - Customer External ID
 - Driver License Country of Issue
 - Driver License Expiry Date
 - Driver License Number
 - Driver License State of Issue

- Is Business
 - Is Leisure
 - Owner ID
 - System Modstamp
- Rename label for the following fields:
 - Booking to Booking Id
 - Deleted to Is Deleted
 - Marketing Optin for Deals and Offers to Marketing Optin
 - Record ID to Rental Id
 - Rental ID to Rental Reference Number
 - SfdcOrganizationId to Internal Organization
- Enable value suggestion for the following fields:
 - Drop Off Location Code
 - Drop Off Location Name
 - Fleet Vehicle External ID
 - Pick Up Location Code
 - Pick Up Location Name
 - Protection Coverage
 - Purpose
- Save the new object 

- The system will map all the attributes for you as the model was constructed based on data stream.



 **What we achieved here?** Let's have a quick reflection on what we've configured and what that model mapping enables:

- **Rental** custom entity is another example use case where there is no standard data model object, therefore we had to create a net new custom DMO. The process was to use the respective data stream as a source for the desired schema, and with few tweaks we were able to create new DMO and mappings were configured automatically as a result.
- Attributes of this object enable identification of customers with vehicle rental history, with or without prior booking, book using various partner loyalty programs to receive discounts and so forth. For few attributes we've enabled value suggestion ensuring that users get improved user experience and don't need to remember set values used in these fields.

Take a note of the fact that while we have various loyalty program as well as drivers license numbers we have not mapped any of those. If you remember we've configured data transform to extract those attributes into a separate DLO. Let's look at it next.

1.4.1 Create Relationships Between Rental and Booking + Individual

- Click on the **Edit Relationships** icon () on the right of the **Rental**
- Click **New** and subsequently **+ New Relationship**
 - Field = Booking Id

- Cardinality = 1:1
- Related Object = Booking
- Related Field = Booking Id
- Click + New Relationship
 - Field = Customer
 - Cardinality = N:1
 - Related Object = Individual
 - Related Field = Individual Id

Edit Relationships

Create relationships with Standard or Custom Data Model Objects.

Object	Field	Cardinality	Related Object	Related Field	Active	Delete
Rental	Booking Id	1:1	Booking	Booking Id	<input checked="" type="checkbox"/>	
Rental	Customer	N:1	Individual	Individual Id	<input checked="" type="checkbox"/>	
+ New Relationship						

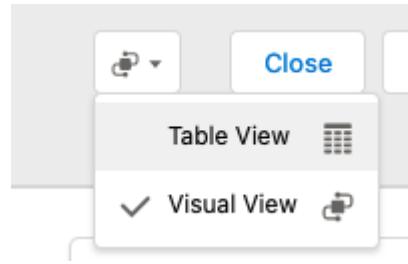
[Cancel](#) [Save & Close](#) [Save](#)

- Save and close the dialog.

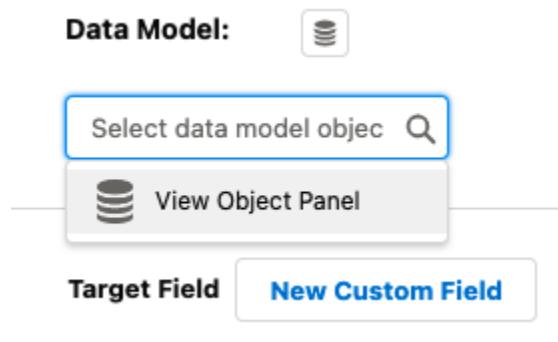
1.5 Party Identification Collection

This DLO contains collection of the various identifiers that we've extracted from the **Rental** data stream using data transform. Now it's time to use it and map to the data model.

- Navigate to the **Data Lake Objects** tab, locate and open the Party Identification Collection data lake object
- Click **Start** button on the right to begin mapping
- This time you will use different interface for mapping the data stream. Switch the view from **Visual** to **Table**:



- Open object selection dialog using **View Object Panel** link in the object selection field:



- Add Party Identification object
- Given that we've designed the schema for the collection DLO the mapping of Party Identification Collection fields to the corresponding Party Identification fields is done for you. Verify the results and adjust if needed:

Data Lake Objects
Party Identification Collection Mappings

Search Data Lake Objects...

Party Identification Collection Data Model: Party Identification

Select data model object

Party Identification Collection (8)

		Target Field	New Custom Field
Party Identification Type	A _a	↔	Search data model object field Party Identification Type X
Identification Name	A _a	↔	Search data model object field Identification Name X
Identification Number	A _a	↔	Search data model object field Identification Number X
Party	A _a	↔	Search data model object field Party X
Party Identification ID *	A _a	↔	Search data model object field Party Identification Id X
		Primary Key	

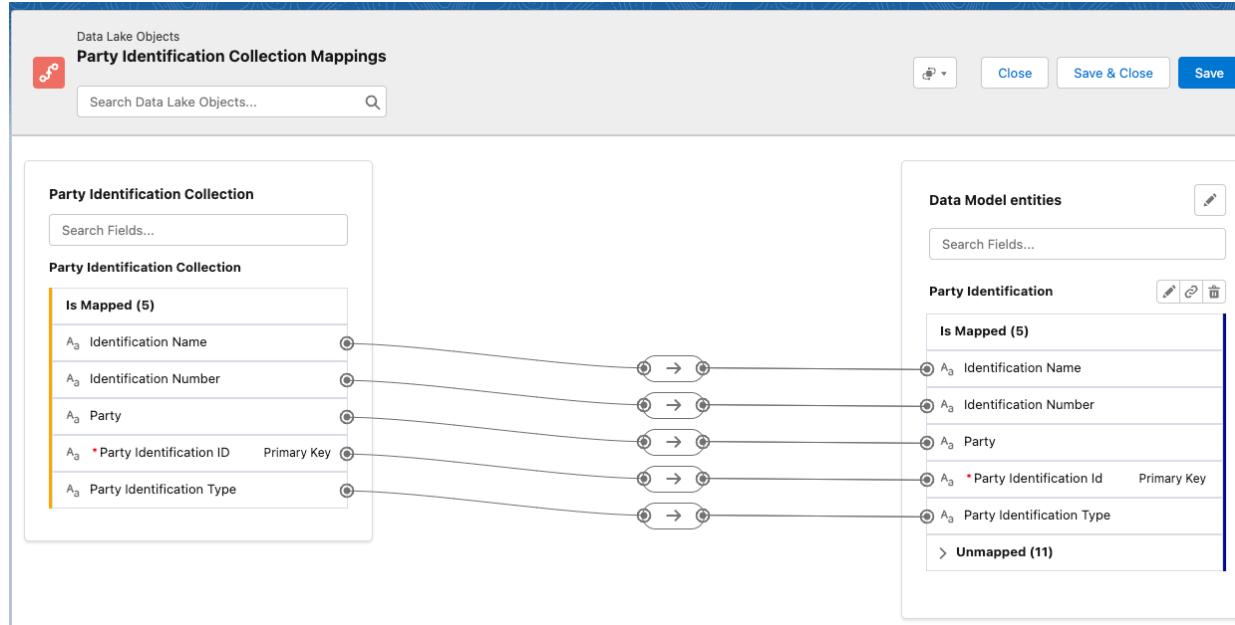
- Save your progress
- You may get an additional warning message when saving the progress here, feel free to ignore it and continue with the following step.

Successfully added mappings.

Data Space

Error logging suggested mapping

- Switch back to **Visual View** and verify the mapping looks as below:



💡 Feel free to choose either of the ways to map data. Keep in mind that when single data stream is mapped to more than one entity you'll need to switch object that you map to in the object selector dialog rather than seeing them all at once.

💡 **What we achieved here?** Let's have a quick reflection on what that model mapping enables:

- **Party Identification** entity is used for the unification process (Identity Resolution) where various identifiers matched against individual profile records.
- Fields from this DMO are not exposed in segmentation and activation canvas. That is the reason why we retained fields representing various memberships on both **Booking** and **Rental** DMOs.

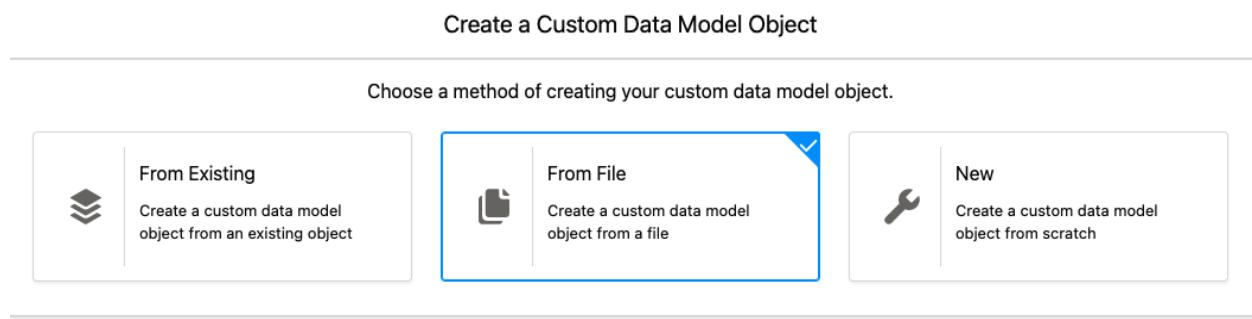
1.6 Rental Preference

For our scenario we will be extending our data model to include rental preferences for the customers. The preferences are stored in a custom object in CRM and have 1:1 relationship with **Contact** object.

While there are various ways to achieve this task as you've experienced in previous steps, this time we will be creating a new custom data model object using externally defined schema. Once completed we will proceed with the mapping for the respective data stream.

Let's proceed with creation of the DMO:

- Navigate to the **Data Model** tab and click the **New** button. Choose **From File** option in the dialog and click **Next**.



- Download [rental_preference_schema_248.csv](#) file, upload it on the next screen and proceed to the next step.
- Set following properties:
 - Object Label = Rental Preference

- Object Category = Other
 - Object Description = Individual preferences for leisure and business rentals
- Enable value suggestion for the following attributes:
 - Leisure - Vehicle Type
 - Leisure - Protection Coverage
 - Business - Vehicle Type
 - Business - Protection Coverage

Create a Custom Data Model Object

* Object Label

* Object API Name

* Object Category

Object Description

⚠️ Category selection affects how data lake objects that are mapped to data model objects are counted for profile billing purposes. [Learn More in Help](#)

Field Label	Field API Name	Data Type	Primary Key	Enable Value Suggestion
Data Source	DataSource	Text	<input type="checkbox"/>	<input type="checkbox"/>
Data Source Object	DataSourceObject	Text	<input type="checkbox"/>	<input type="checkbox"/>
Internal Organization	InternalOrganization	Text	<input type="checkbox"/>	<input type="checkbox"/>
Rental Preference ID	Rental_Preference_Id	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Rental Preference Reference Number	Rental_Preference_Reference_Number	Text	<input type="checkbox"/>	<input type="checkbox"/>
Individual Id	Individual_Id	Text	<input type="checkbox"/>	<input type="checkbox"/>
Customer External ID	Customer_External_Id	Text	<input type="checkbox"/>	<input type="checkbox"/>
Leisure - Vehicle Type	Leisure_Vehicle_Type	Text	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Leisure - Auto Transmission	Leisure_Auto_Transmission	Boolean	<input type="checkbox"/>	<input type="checkbox"/>
Leisure - Manual Transmission	Leisure_Manual_Transmission	Boolean	<input type="checkbox"/>	<input type="checkbox"/>
Leisure - EV Only	Leisure_EV_Only	Boolean	<input type="checkbox"/>	<input type="checkbox"/>
Leisure - Baby Seat Count	Leisure_Baby_Seat_Count	Number	<input type="checkbox"/>	<input type="checkbox"/>
Leisure - Booster Seat Count	Leisure_Booster_Seat_Count	Number	<input type="checkbox"/>	<input type="checkbox"/>
Leisure - Protection Coverage	Leisure_Protection_Coverage	Text	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Leisure - Airline Membership Number	Leisure_Airline_Membership_Number	Text	<input type="checkbox"/>	<input type="checkbox"/>
Leisure - Motor Club Number	Leisure_Motor_Club_Number	Text	<input type="checkbox"/>	<input type="checkbox"/>
Leisure Preferences Last Updated Date	Leisure_Preferences_Last_Updated_Date	Date	<input type="checkbox"/>	<input type="checkbox"/>
Business - Vehicle Type	Business_Vehicle_Type	Text	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Business - Auto Transmission	Business_Auto_Transmission	Boolean	<input type="checkbox"/>	<input type="checkbox"/>
Business - Manual Transmission	Business_Manual_Transmission	Boolean	<input type="checkbox"/>	<input type="checkbox"/>
Business - EV Only	Business_EV_Only	Boolean	<input type="checkbox"/>	<input type="checkbox"/>
Business - Baby Seat Count	Business_Baby_Seat_Count	Number	<input type="checkbox"/>	<input type="checkbox"/>
Business - Booster Seat Count	Business_Booster_Seat_Count	Number	<input type="checkbox"/>	<input type="checkbox"/>
Business - Protection Coverage	Business_Protection_Coverage	Text	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Business - Airline Membership Number	Business_Airline_Membership_Number	Text	<input type="checkbox"/>	<input type="checkbox"/>
Business - Motor Club Number	Business_Motor_Club_Number	Text	<input type="checkbox"/>	<input type="checkbox"/>
Business Preferences Last Updated Date	Business_Preferences_Last_Updated_Date	Date	<input type="checkbox"/>	<input type="checkbox"/>

+ Add Field

-

- Save the object ✓
- Notice that even though the object has been created, it might not be available immediately under the **Data Model** tab.
Change the list view to **All** to locate and verify that the object was successfully created.

Next let's proceed with data mapping.

- In the **Data Streams** locate and open the `Rental.Preference__c_<ORG_ALIAS>` data stream
- Click **Start** link on the right
- Click **Select Objects** and add Rental Preference from the **Custom Data Model** tab
- As the data stream closely matches to the Data Model Object the mapping is done for you. But we need to make small adjustment to it:
 - Delete mapping to Rental Preference ID (Primary Key)
- Configure following mappings:

DLO	DLO	→	DMO	DMO
	Field		Field	

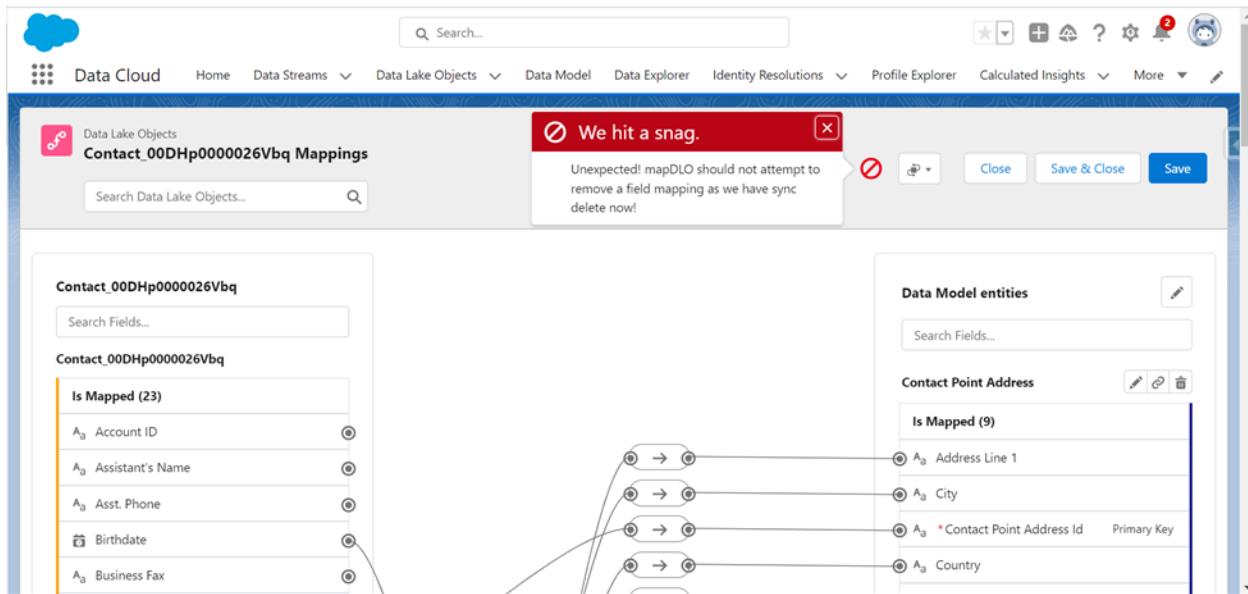
Rental.Preference__c_<ORG_ALIAS>	Customer	→	Individual ID	Rental Preference
	Record ID	→	Rental Preference ID	

		(primary key)	
Rental Preference ID	→	Rental Preference Reference Number	



- Save your progress ✓

⚠ When mapping fields from your DLO to DMO in the data streams tab, you may see an error message:



To resolve try following steps:

- Make sure it's the right mapping
- Click on the 'Save' button opposed to 'Save & Close'
- Close the mapping screen and try again
- Go to the "Data Lake Objects" tab and try mapping from there

💡 **What we achieved here?** Let's have a quick reflection on what we've configured and what that model mapping enables:

- **Rental Preference** custom entity is an example use case where using outcomes of the initial customer data analysis we created the schema file for the custom entity and deployed it ahead of the mapping activity.

- The value suggestion was enabled on reasonably large number of attributes as we expect this object to be one of the primary ones used in segmentation, therefore making it easy for users to work with data.

1.6.1 Create Relationship Between Rental Preference and Individual

- Click on the **Edit Relationships** icon () on the right of the **Rental Preference**
- Click **New** and subsequently **+ New Relationship**
 - Field = Individual Id
 - Cardinality = 1:1
 - Related Object = Individual
 - Related Field = Individual Id
- Save and close the dialog. 

2. eCommerce Data

Cloud Storage data streams are not automatically mapped into the data model therefore you need to complete all mappings manually.

2.1 eComm Customer Profile

With the **eComm Customer Profile** data stream we won't be mapping it to the **Loyalty Program Member** DMO even though we

have key attribute. If you inspect the data in the source file you'll notice that not all records contain value for the **Rewards Program ID** column. Therefore we will run into incohesive result in the data model, unless we utilize data transform to extract only records with values. Plus we don't have any other meaningful attributes that would add any value from this mapping.

This last point is subjective though and in real implementation can be argued upon, as even the fact that an individual is a member of the loyalty program can already be a criteria for marketing activities as an example. Therefore in order to still map the data in a cohesive way we would've configured yet another data stream for the same file, but used **Rewards Program ID** through the formula field to explicitly handle records without that attribute. But for our purposes we will avoid implementing that approach.

Assuming you remained on the mapping dialog:

- In the **Search Data Lake Objects**  field locate and open the eComm Customer Profile data stream
- Click **Select Objects** and add Contact Point Email, Individual and Party Identification
- Notice that some of the fields have been auto-mapped, let's complete remaining fields. Configure following mappings:

DLO

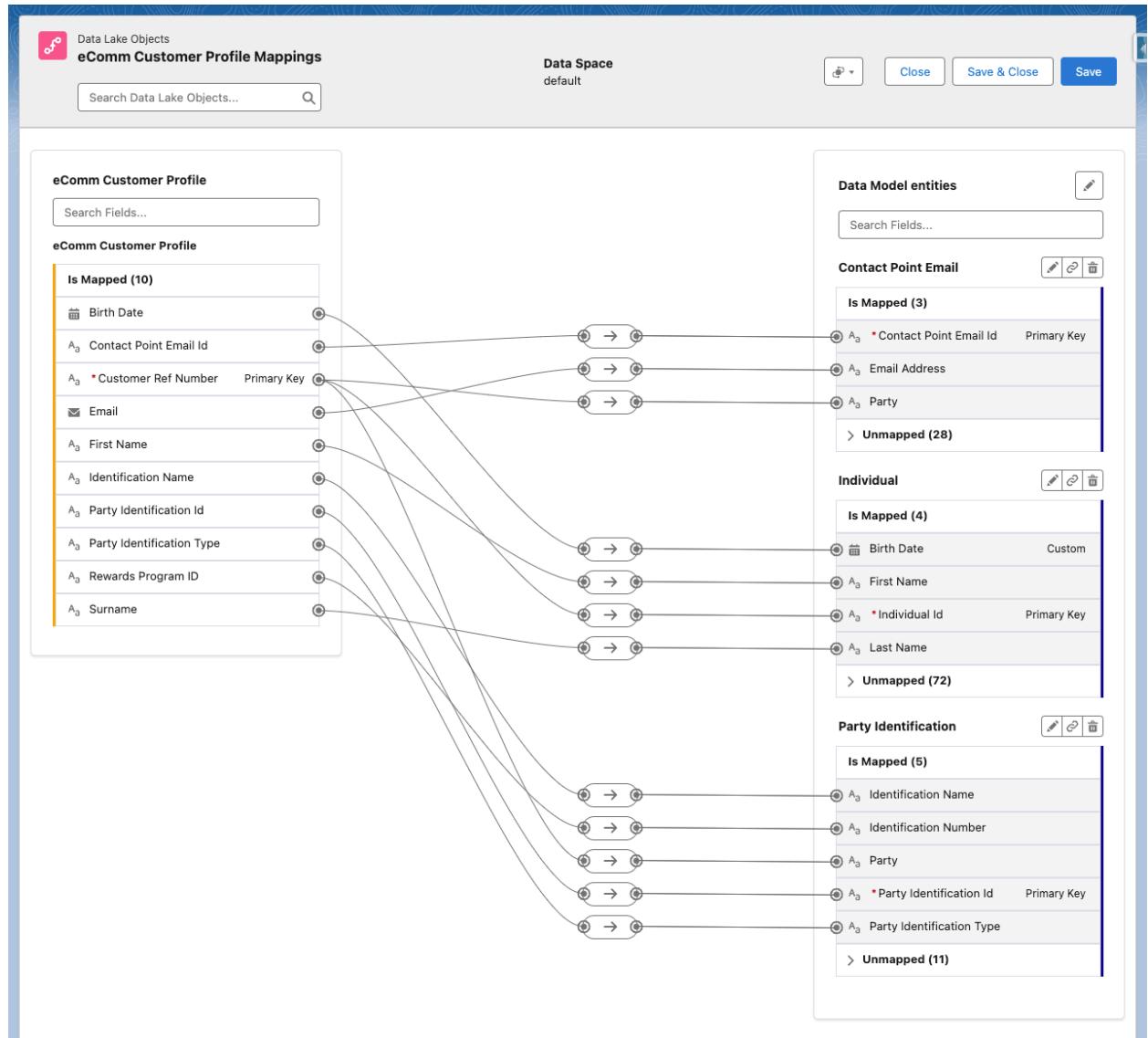
DLO Field

→ **DMO Field**

DMO

eComm Customer Profile	Customer Ref Number	→	Party	Contact Point Email
	Email	→	Email Address	
	Customer Ref Number	→	Individual Id	Individual
	Customer Ref Number	→	Party	Party Identification
	Rewards Program ID	→	Identification Number	

- Save your progress 
- We have Birth Date field left on the **eComm Customer Profile** stream, but if you try to map it to **Individual.Birth Date** you'll notice that system does not let you do that. This is because the standard attribute is of **DateTime** data type, while our data stream attribute is of type **Date**. For our solution let's add a custom field to allow for mapping.
- Click Add New Field under **Individual**:
 - Field Label = Birth Date
 - Data Type = Date
 - Save.
- Map **eComm Customer Profile.Birth Date** to **Individual.Birth Date** (Custom) field.



- Save your progress

What we achieved here? Let's have a quick reflection on what that model mapping enables:

- **Contact Point Email** entity will maintain email addresses for an individual.

- **Individual** entity represents people in the database along with their profile and demographic attributes.
- **Party Identification** will enable unification of the individuals using specified identifiers.

These entities will enable segmentation on customer attributes and unification of customer profiles. Because its a second data feed with customer profile details there is a chance that for same individuals the email address in this stream can be different from the previously ingested and mapped via CRM Contact data stream. And that is something we'll need to address later during configuration of the Identity Resolution reconciliation rules.

The mapping to the **Party Identification** object extends the configuration approach we took in the lead to the profile unification for individuals with the matching loyalty profile but different customer identifiers.

2.1.1 Update CRM Contact Mapping

Due to the fact that we now have 2 different **Birthday Date** fields mapped in the data model this creates a challenge as that field is part of our individual matching rules. At this stage the CRM data stream does not support modifications to the data type, nor does it allow deletion and re-adding of the same field. Therefore there are at least two options to choose from here:

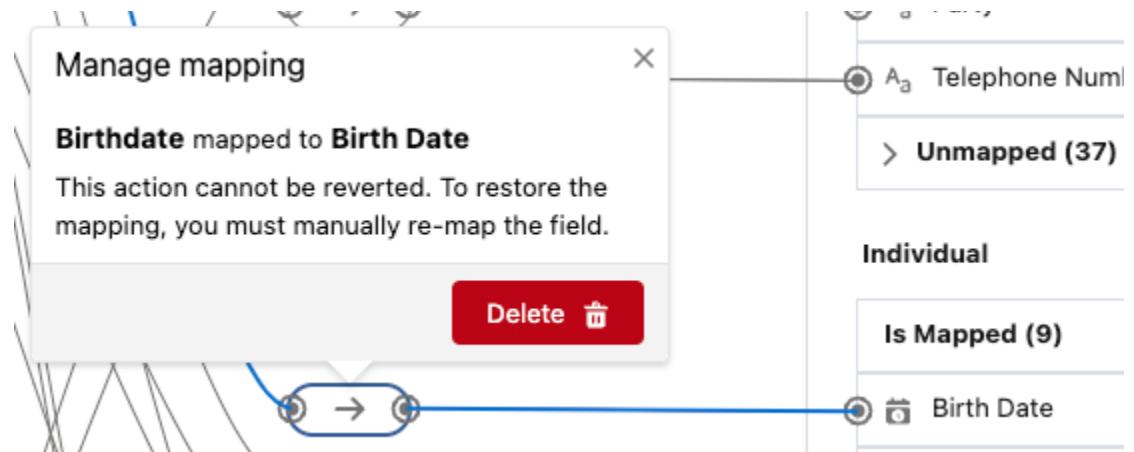
1. Disable the **Birthdate** field in the data stream, modify/create a new custom field in CRM and then add that field as new source field on the **Contact_<ORG_ALIAS>** data stream.
2. Create a formula field that will simply cast the **Birthdate** field to be of **Date** data type and then use that field when mapping to the data model.

Evaluating both options the second one seems easier to apply.

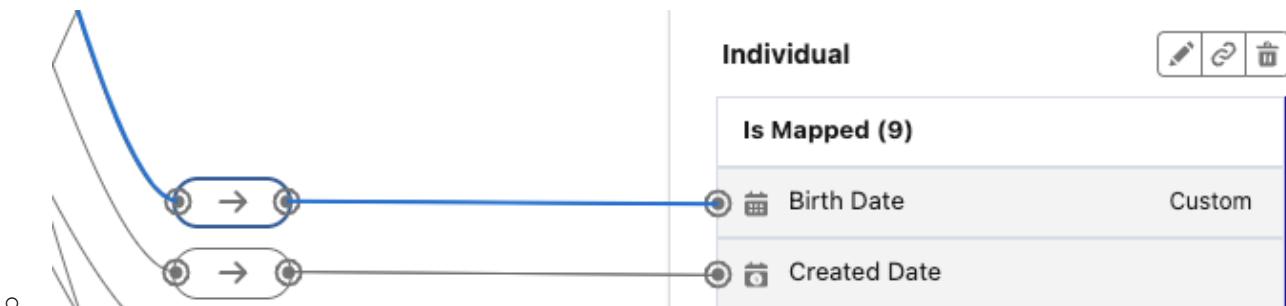
Although as [per documentation](#) adding new formula field does not trigger the full refresh of the data stream and therefore we will have no values populated in the new field until either source record is updated in CRM or full refresh of that data stream occurs (once a fortnight). Knowing that ahead of time we already created a formula field in the [Data Ingestion](#) activity and now simply need to update the mapping.

Therefore let's change the mapping that we've configured for **Contact_<ORG_ALIAS>** data stream and use new custom field instead:

- In the **Search Data Lake Objects**  field locate and open the Contact_<ORG_ALIAS> data stream
- Remove mapping between **Birthday** and **Birth Date** fields:



- Map Contact_<ORG_ALIAS>.Birth Date to Individual.Birth Date (custom) field:



- Save your progress ✓

💡 What we achieved here? Let's have a quick reflection on why did we make this change:

Further in the course when configuring identity resolution process the rules that specify how different individuals should be matched don't allow for cross-attribute comparison. Therefore if customer profile data from eCommerce stream is mapped to one field, while customer data from CRM is mapped to another we won't be able to use both fields in the same rule (compare values across both). Therefore the easiest change at this stage was to introduce new

formula field for the stream and use that instead of standard values both on data stream and data model.

 This exercise is an example of how you can address similar situation in your real implementations. It does not necessarily have to be related to a single data type, rather a case where you need to make a change without doing complete reconfiguration of the data stream.

2.2 eComm Order

The eComm Order data stream will need a few custom fields added to the target DMO - Sales Order. We've already added custom fields to the object from the mapping screen, let's look into an alternative method that will result in less clicks overall.

- Under **Data Model** tab, adjust your view of Data Model objects from **Mapped to All**
- Use the Search tool on the right side of the screen to search for the **Sales Order** DMO and open it
- Click on **Edit** button on the right
- Scroll to the bottom of the field list and click on **+ Add Field**
 - Label = Delivery Method
 - Type = Text
 - Select Enable Value Suggestion
- Add another custom field

- Label = Loyalty Points Accrued
- Type = Number

WebStoreSales Store	WebStoreSalesStoreId	Text	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Delivery Method	Delivery_Method	Text	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Loyalty Points Accrued	Loyalty_Points_Accrued	Number	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

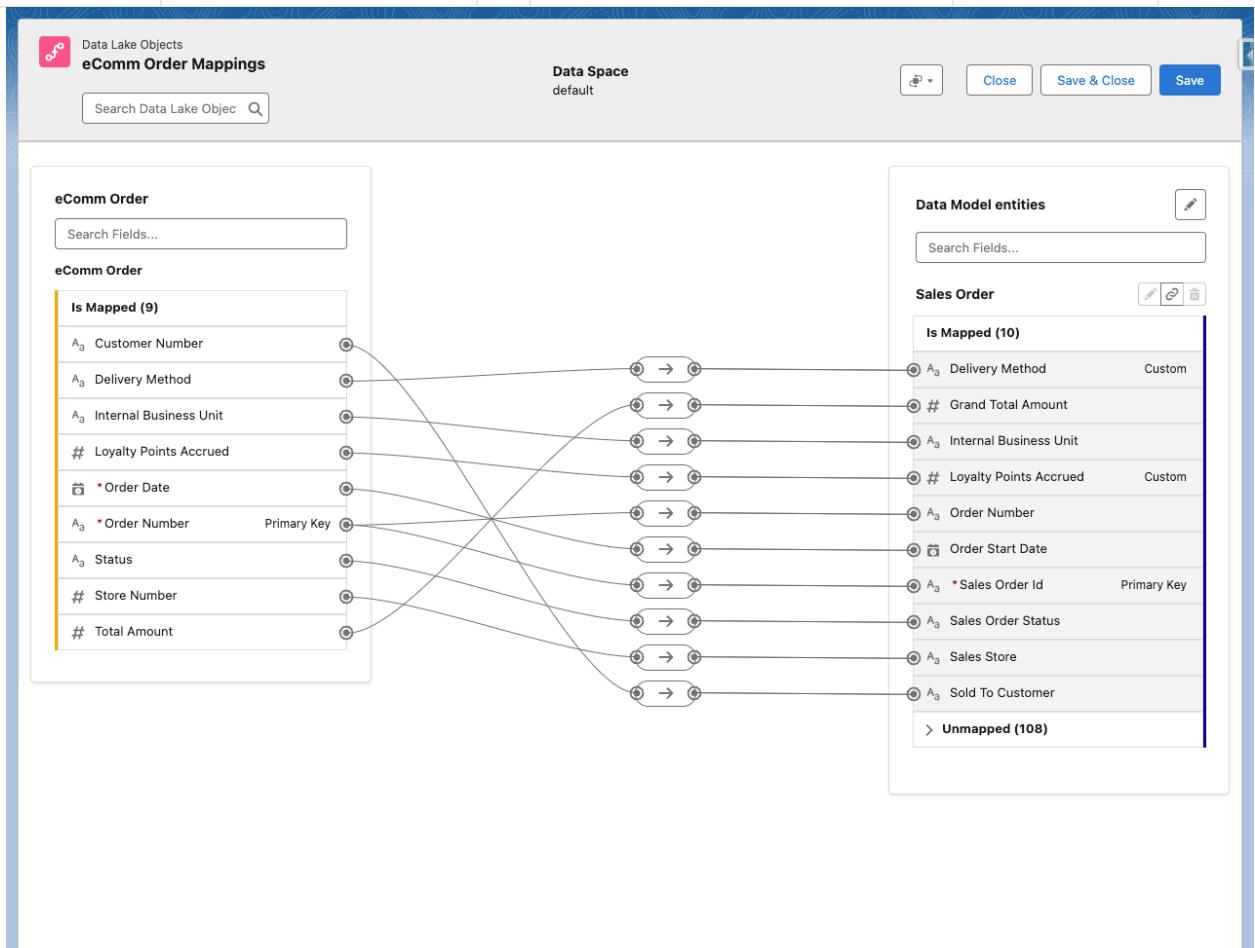
[+ Add Field](#)

[Cancel](#) [Save](#)

- Save the schema.
- In the **Data Streams** locate and open the eComm Order data stream
- Click **Start** link on the right
- Click **Select Objects** and add Sales Order
- Delete mapping to Total Amount field
- Configure following mappings:

DLO	DLO Field	→ DMO Field	DMO
eComm Order	Customer Number	→ Sold To Customer	Sales Order
	Order Date	→ Order Start Date	
	Order Number	→ Sales Order Id (primary key)	
	Status	→ Sales Order Status	

	Store Number	→ Sales Store	
	Total Amount	→ Grand Total Amount	



- Save your progress

What we achieved here? Let's have a quick reflection on what that model mapping enables:

- **Sales Order** entity will maintain summary of transactions (orders) completed by customers over time.

This entity enables segmentation using purchase information, e.g. number of purchases completed by a customer over certain period of time. We also configured store reference value but the actual store details still need to be mapped in order to be available for selection in segmentation criteria.

In addition to that, we mapped Customer Number to the Sold To Customer field, upon inspection you will notice that this field is not by default related to an individual, therefore that is something we take care of next.

2.2.1 Create Relationship Between Sales Order and Individual

To ensure **Sales Order** entity is correctly related to the **Individual** entity we need to configure new relationship. This time we will perform this task right from the mapping screen.

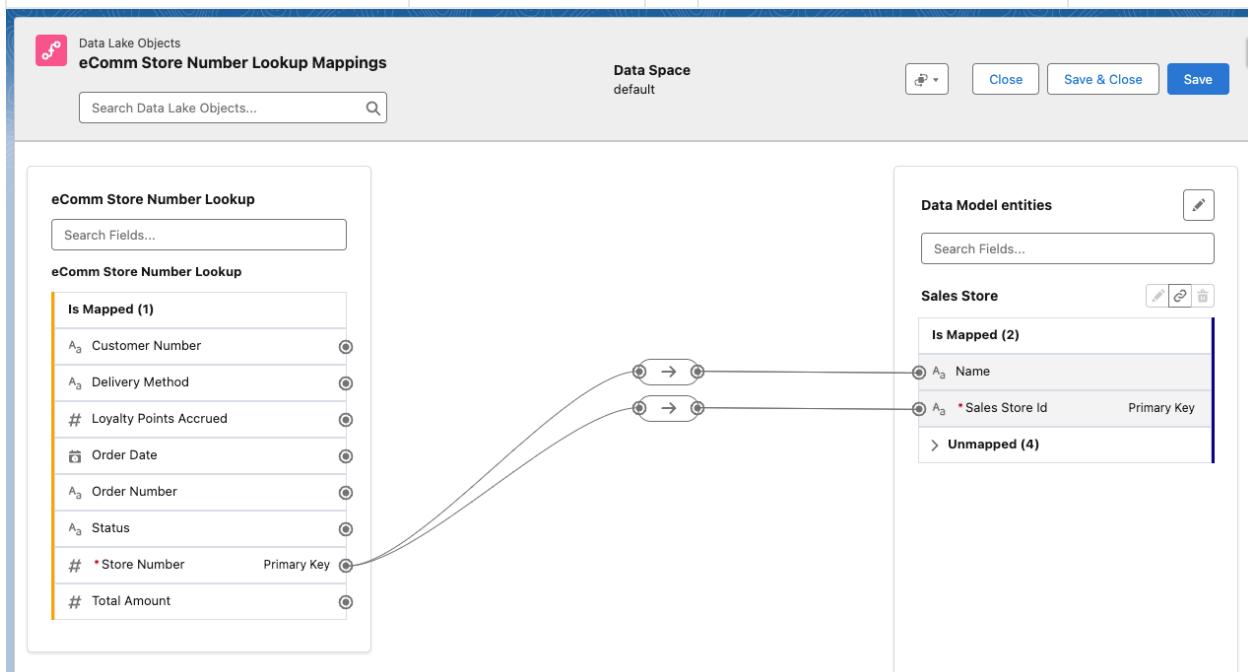
- Click on the **Edit Relationships** icon () on the right of the **Sales Order**
- Click on **New** and subsequently **+ New Relationship**
- Add new relationship via Sold To Customer field
 - Cardinality = N:1
 - Related Object = Individual
 - Related Field = Individual Id

Save and close the dialog  and remain on the mapping screen.

2.3. eComm Store Number Lookup

- In the **Search Data Lake Objects**  field locate and open the eComm Store Number Lookup data stream
- Click **Select Objects** and add Sales Store
- Configure following mappings:

DLO	DLO Field	→	DMO Field	DMO
eComm Store Number Lookup	Store Number	→	Name	Sales Store
	Store Number	→	Sales Store Id (primary key)	



Save your progress  and remain on the mapping dialog.

 **What we achieved here?** Let's have a quick reflection on what that model mapping enables:

- **Sales Store** entity is another example of a lookup table for a store details where we avoided creating custom field on Sales Order entity.

Note that while use of Store Numbers as a Name value is not ideal, we choose that to make the value selectable in segmentation criteria. Practical option would be to source actual names that matched with the store numbers, so that user ends up using familiar store names instead of their reference numbers. For our workshop we simply making functional schema with little data that is available.

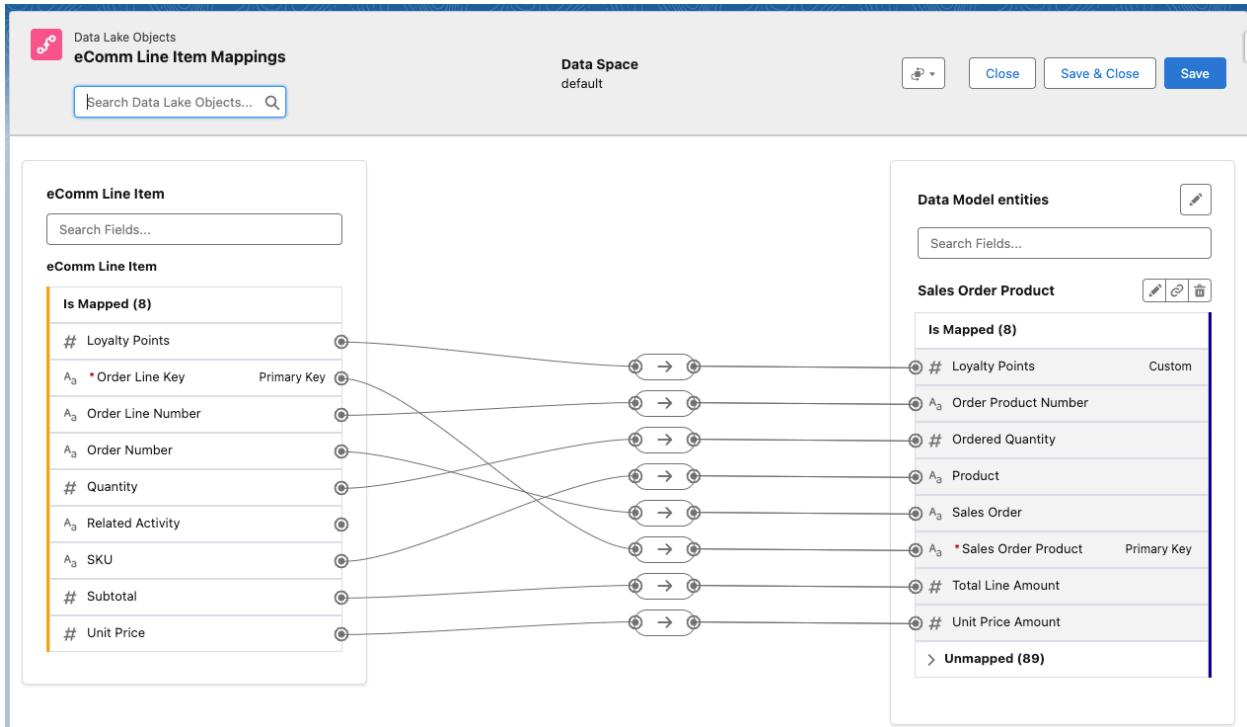
2.4 eComm Line Item

- In the **Search Data Lake Objects**  locate and open the eComm Line Item data stream
- Click **Select Objects** and add Sales Order Product
- Configure following mappings:

DLO	DLO Field	→ DMO Field	DMO
eComm Line Item	Order Line Key	→ Sales Order Product (primary key)	Sales Order Product

	Order Line Number	→ Order Product Number	
	Order Number	→ Sales Order	
	Quantity	→ Ordered Quantity	
	SKU	→ Product	
	Subtotal	→ Total Line Amount	
	Unit Price	→ Unit Price Amount	

- Save your progress 
- Click **Add New Field** under **Sales Order Product**
 - Label = Loyalty Points
 - Type = Number
 - Save field
- Map **eComm Line Item.Loyalty Points** to **Sales Order Product.Loyalty Points**



Save your progress and remain on the mapping dialog.

What we achieved here? Let's have a quick reflection on what that model mapping enables:

- **Sales Order Product** entity will maintain details of each line item within an order completed by customers over time.

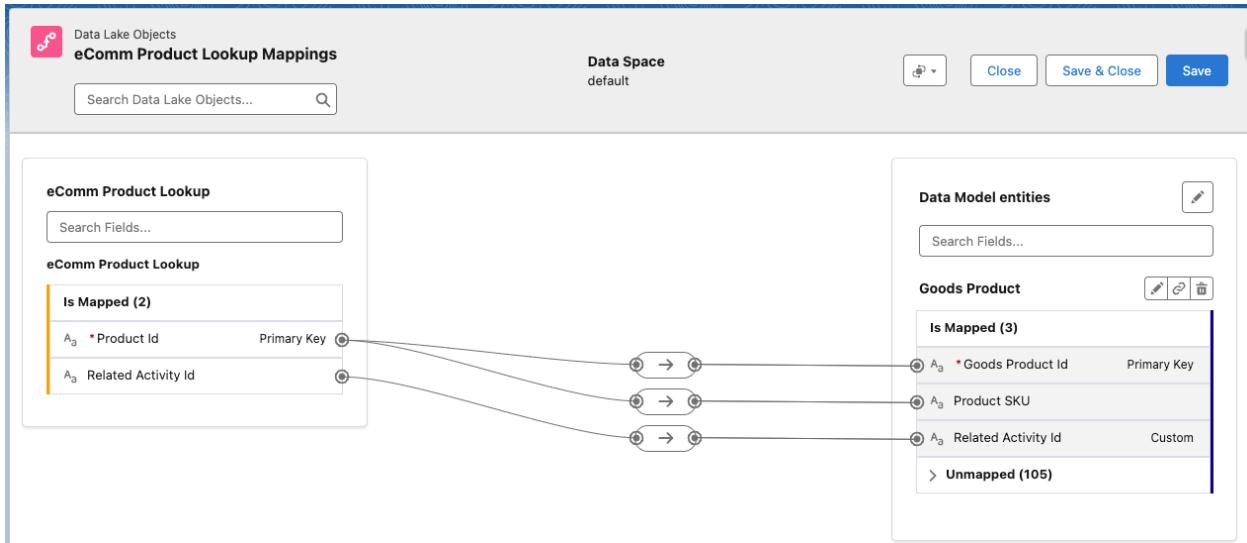
This entity enables use cases such as identifying individuals with more than 2 items purchased in a single transaction or target customers purchasing particular products. The product details data though is still not mapped and we take care of it next.

2.5 eComm Product Lookup

- In the **Search Data Lake Objects**  locate and open the eComm Product Lookup data stream
- Click **Select Objects** and add Goods Product
- Configure following mappings:

DLO	DLO Field	→ DMO Field	DMO
eComm Product Lookup	Product Id	→ Goods Product Id (primary key)	Goods Product
	Product Id	→ Product SKU	

- Save your progress 
- Click **Add New Field** under **Goods Product**
 - Label = Related Activity Id
 - Type = Text
 - Save and map Related Activity Id to Related Activity Id



Save your progress and remain on the mapping dialog.

What we achieved here? Let's have a quick reflection on what that model mapping enables:

- **Goods Product** entity will maintain details of product that is linked to respective line item within a sales order.

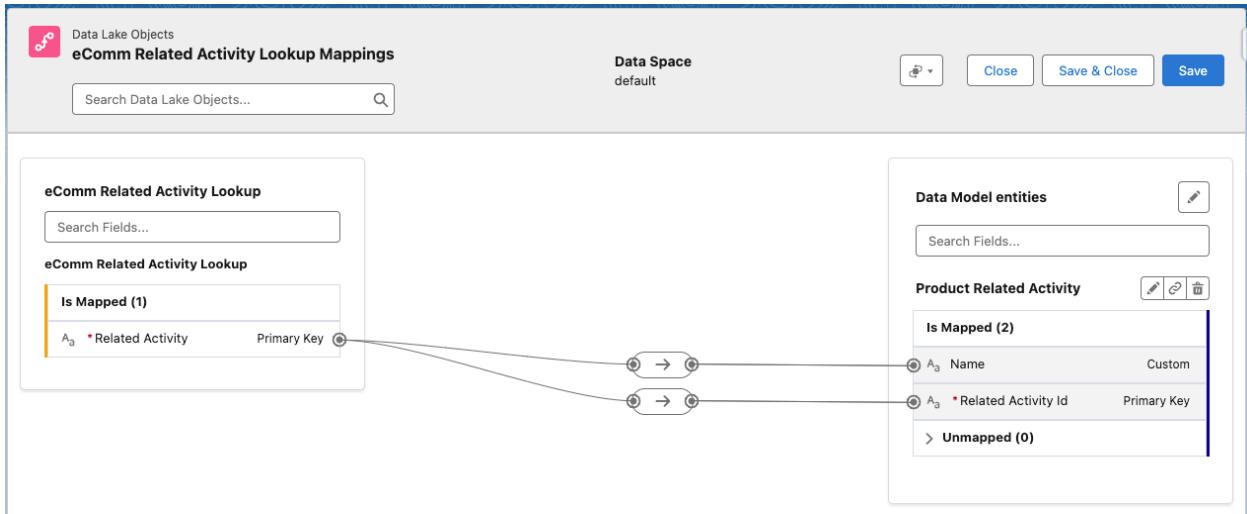
This entity extends segmentation use cases that take advantage of purchase data, specifically around product SKUs.

We also included mapping of the Related Activity attribute, let's create a custom data model object for it and establish the relationship back to Goods Product.

2.6 eComm Related Activity Lookup

Additional attribute that associates product with outdoor activities requires new custom object to be added to the data model.

- In the **Search Data Lake Objects**  locate and open the eComm Related Activity Lookup data stream
- Click **Select Objects**
- Click **Custom Data Model** tab and then **+ New Custom Object**
 - Object Label = Product Related Activity
 - Ensure to only select following fields:
 - Data Source
 - Data Source Object
 - Internal Organization
 - Related Activity
 - Rename label to Related Activity Id
- Save the new object. 
- Click **Add New Field** under **Product Related Activity**
 - Label = Name
 - Type = Text
 - Select Enable Value Suggestion
 - Save and map **eComm Related Activity Lookup.Related Activity to Product Related Activity.Name**



- Save your progress but don't close the mapping dialog just yet.

💡 What we achieved here? Let's have a quick reflection on what that model mapping enables:

- **Product Related Activity** entity will maintain list of outdoor activities (e.g. running, hiking, camping) that can be associated with products.

This entity extends segmentation use cases related to transactional attributes with ability to identify customers who purchased products with related activity different or matched to the outdoor interest captured in their profile.

As this is a custom object it is not related to any other entity. We need it to be related to the Goods Product, so let's take care of it next.

2.6.1 Create Relationship Between Product Related Activity and Goods Product

- Click on the **Edit Relationships** icon () on the right of the **Product Related Activity**
- Click **New** and subsequently **+ New Relationship**
 - Field = Related Activity Id
 - Cardinality = 1:1
 - Related Object = Goods Product
 - Related Field = Related Activity Id
- Save and close the dialog. 

3. Review Data Model Relationships

Before we proceed with any further configurations it is important to ensure that all data relationships were configured correctly and according to our expectations and requirements.

Navigate to **Data Model** tab, open each of the following objects and review the relationships to ensure all configured accordingly. If you find any relationships missing then navigate back to the respective DLO and complete relationship mappings. Search for relevant instructions in sections 1 or 2.

- Individual (x10)

Relationships								Edit
Object	Field	Key Qualifier (Fie...	Cardinality	Related Object	Related Field	Key Qualifier (Rel...		
1 Booking	Customer	KQ_Customer_c	ManyToOne	Individual	Individual Id	KQ_Id		
2 Case	Account Contact	KQ_AccountContactId	ManyToOne	Individual	Individual Id	KQ_Id		
3 Contact Point Address	Party	KQ_PartyId	ManyToOne	Individual	Individual Id	KQ_Id		
4 Contact Point Email	Party	KQ_PartyId	ManyToOne	Individual	Individual Id	KQ_Id		
5 Contact Point Phone	Party	KQ_PartyId	ManyToOne	Individual	Individual Id	KQ_Id		
6 Loyalty Program Member	Individual Id	KQ_Individual_Id	OneToOne	Individual	Individual Id	KQ_Id		
7 Party Identification	Party	KQ_PartyId	ManyToOne	Individual	Individual Id	KQ_Id		
8 Rental	Customer	KQ_Customer_c	ManyToOne	Individual	Individual Id	KQ_Id		
9 Rental Preference	Individual Id	KQ_Individual_Id	OneToOne	Individual	Individual Id	KQ_Id		
10 Sales Order	Sold To Customer	KQ_SoldToCustomerId	ManyToOne	Individual	Individual Id	KQ_Id		

•

- Loyalty Program Member (x2)

Relationships								Edit
Object	Field	Key Qualifier (Fie...	Cardinality	Related Object	Related Field	Key Qualifier (Rel...		
1 Loyalty Member Currency	Loyalty Program Member	KQ_LoyaltyProgramMe...	ManyToOne	Loyalty Program Member	Loyalty Program Member Id	KQ_Id		
2 Loyalty Program Member	Individual Id	KQ_Individual_Id	OneToOne	Individual	Individual Id	KQ_Id		

•

- Sales Order (x3)

Relationships								Edit
Object	Field	Key Qualifier (Fie...	Cardinality	Related Object	Related Field	Key Qualifier (Rel...		
1 Sales Order	Sales Store	KQ_SalesStoreId	ManyToOne	Sales Store	Sales Store Id	KQ_Id		
2 Sales Order	Sold To Customer	KQ_SoldToCustomerId	ManyToOne	Individual	Individual Id	KQ_Id		
3 Sales Order Product	Sales Order	KQ_SalesOrderId	ManyToOne	Sales Order	Sales Order Id	KQ_Id		

•

- Sales Order Product (x2)

Relationships								Edit
Object	Field	Key Qualifier (Fie...	Cardinality	Related Object	Related Field	Key Qualifier (Rel...		
1 Sales Order Product	Product	KQ_ProductId	ManyToOne	Goods Product	Goods Product Id	KQ_Id		
2 Sales Order Product	Sales Order	KQ_SalesOrderId	ManyToOne	Sales Order	Sales Order Id	KQ_Id		

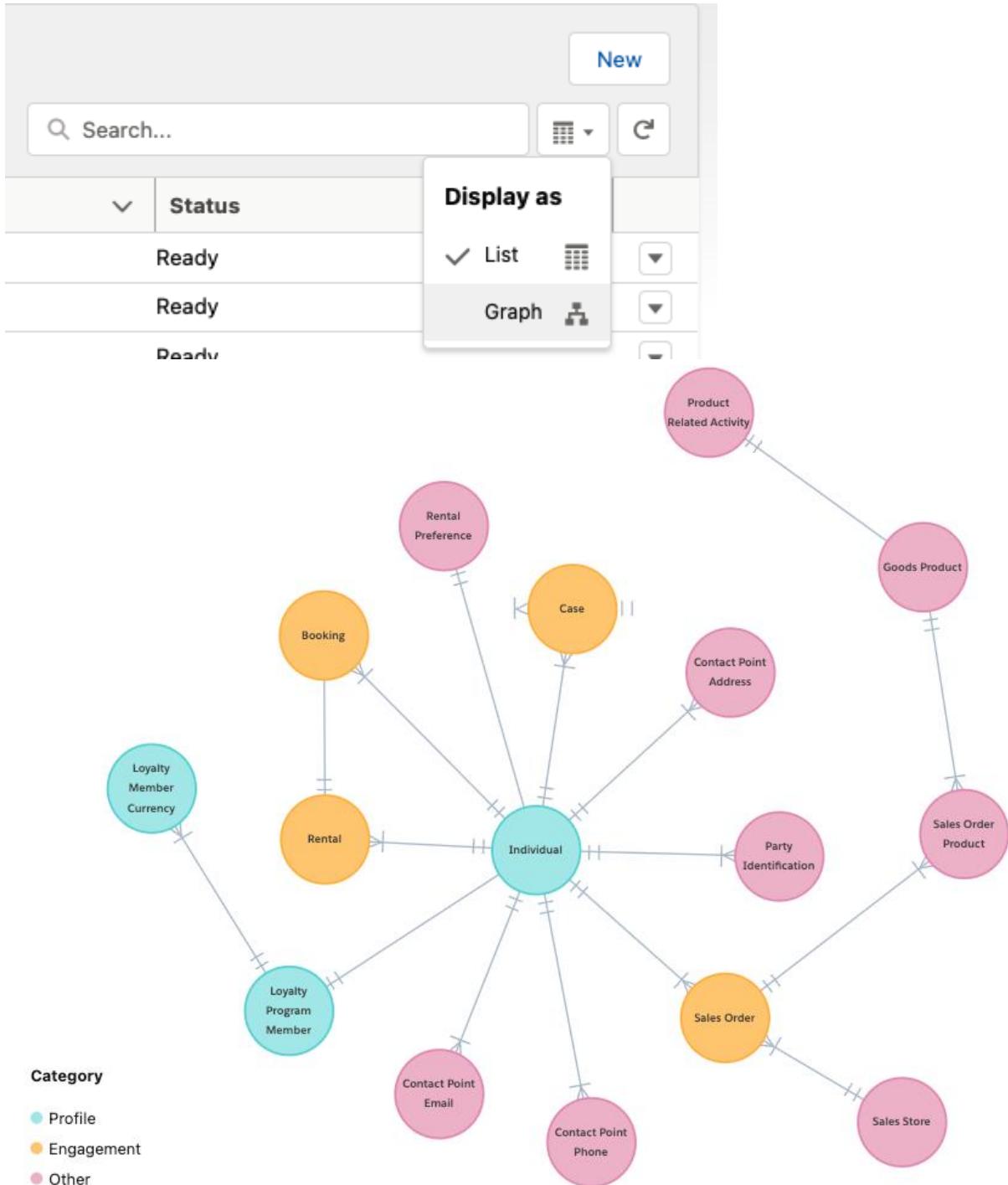
•

- Goods Product (x2)

Relationships								Edit
Object	Field	Key Qualifier (Fie...	Cardinality	Related Object	Related Field	Key Qualifier (Rel...		
1 Product Related Activity	Related Activity Id	KQ_related_activity	OneToOne	Goods Product	Related Activity Id	KQ_Related_Activity_Id		
2 Sales Order Product	Product	KQ_ProductId	ManyToOne	Goods Product	Goods Product Id	KQ_Id		

•

While on the **Data Model** tab switch view from **List** to **Graph** and verify that model loads, there are no orphans and that relationship are similar to the model below.



⚠️ When viewing the Data Model Object relationships, you may see

duplicate instance of the same relationship. This is a UI bug that sometimes show duplicated relationships. You can ignore this and move forward and it should not cause any issues downstream in the hands on activities. !

4. Update Key Qualifier Configuration

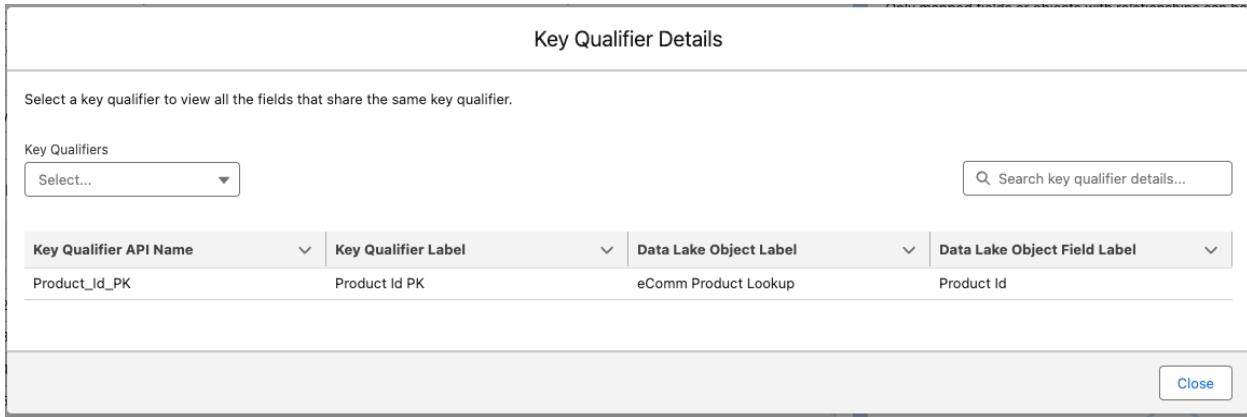
At this stage it is not intuitively noticeable, but our configuration has one challenge. Let's walk through the steps to identify required change and then apply necessary configuration to address it.

4.1 Inspecting Key Qualifier Relationships

If you review relationships of the **Goods Product** DMO more closely you will notice one interesting detail:

Relationships								Edit
Object	Field	Key Qualifier (Fie...	Cardinality	Related Object	Related Field	Key Qualifier (Rel...		
1 Goods Product	Related Activity Id	KQ_Related_Activity_Id	OneToOne	Product Related Activity	Related Activity Id	KQ_Related_Activity		
2 Sales Order Product	Product	KQ_ProductId	ManyToOne	Goods Product	Goods Product Id	KQ_Id		

The **Sales Order Product** DMO has the **Product** field associated with the key qualifier KQ_ProductId. If you revisit the configuration you did for the **eComm Product Lookup** DLO then you'll notice that only 1 field was configured in that key qualifier - Product Id:

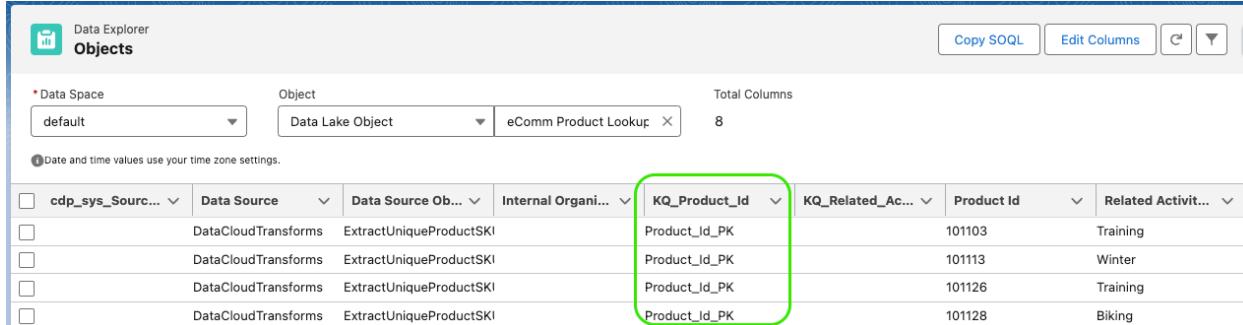


What happens here is that **Product** field on the **Sales Order** **Product** DMO is considered as a foreign key to the object that has primary identifier field configured with key qualifier. And therefore the expectation is that in any other place where that field is referenced the foreign key fields are also included in the same key qualifier configuration. That will enable correct matching when objects are joined for the purpose of calculated insights build or in segmentation relationships. Without it the join will not work and results will be different from expected.

i If you want to test out the results without applying any changes feel free to bookmark this section and return back once you reach the **Configure Segments** exercise and test out VALIDATION - Cross-entities segment creation. As soon as you add second criteria that uses **Related Activity** detail the count of the segment will drop to 0. After applying changes below the segment count will be “fixed” and return to the expected value.

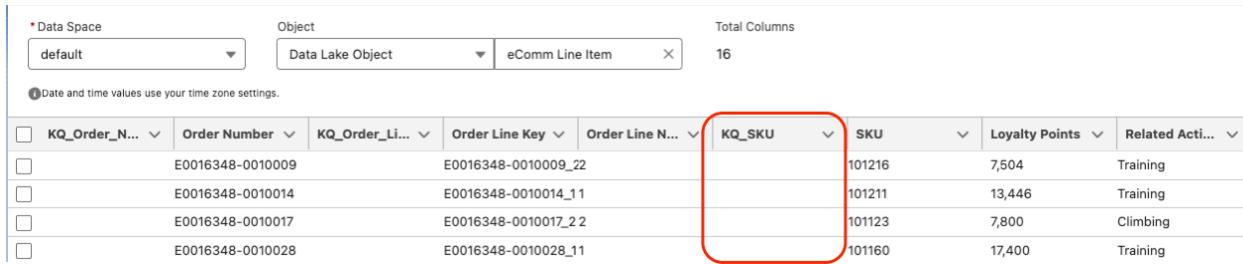
There is one more method we can use to validate the configuration difference. Navigate to **Data Explorer** and open eComm Product

Lookup DLO. In the list of records notice that column **KQ_Product_Id** contains value **Product_Id_PK**:



cdp_sys_Source	Data Source	Data Source Obj	Internal Organi...	KQ_Product_Id	KQ_Related_Ac...	Product Id	Related Activit...
	DataCloudTransforms	ExtractUniqueProductSKI		Product_Id_PK		101103	Training
	DataCloudTransforms	ExtractUniqueProductSKI		Product_Id_PK		101113	Winter
	DataCloudTransforms	ExtractUniqueProductSKI		Product_Id_PK		101126	Training
	DataCloudTransforms	ExtractUniqueProductSKI		Product_Id_PK		101128	Biking

Now open the eComm Line Item DLO (this one is mapped to the **Sales Order Product** DMO), adjust columns to display **KQ_SKU** and **SKU** (this one is mapped to the **Product** field on the **Sales Order Product** DMO) you'll notice that key qualifier field is not populated:



KQ_Order_N...	Order Number	KQ_Order_Li...	Order Line Key	Order Line N...	KQ_SKU	SKU	Loyalty Points	Related Acti...
	E0016348-0010009		E0016348-0010009_22			101216	7,504	Training
	E0016348-0010014		E0016348-0010014_11			101211	13,446	Training
	E0016348-0010017		E0016348-0010017_22			101123	7,800	Climbing
	E0016348-0010028		E0016348-0010028_11			101160	17,400	Training

Now let's apply necessary change to ensure that both objects can be properly connected through the use of the common key qualifier.

4.2 Configure Key Qualifier for eComm Line Item DLO

Navigate to the **Data Lake Objects** tab, locate and open **eComm Line Item** DLO. On the right of the **SKU** field click on the  (arrow down) icon and select **Add Key Qualifier** action:

10	Order Line Number	Order_Line_Number...	Text
11	Order Number	Order_Number__c	Text
12	Quantity	Quantity__c	Number
13	Related Activity	Related_Activity__c	Text
14	SKU	SKU__c	Text
15	Subtotal	Subtotal__c	Number

Add Key Qualifier
View Key Qualifiers
Remove Key Qualifier

Choose an existing record **Product_Id_PK** in the presented table and click **Add**:

Add Key Qualifier

Adding a key qualifier ensures accurate data is used throughout Data Cloud. Select an existing key qualifier or create a new one.

Key Qualifiers	Key Qualifier API Name ↑	Key Qualifier Label	Status	Description
<input checked="" type="radio"/> Product_Id_PK	Product_Id_PK	✓	Primary key of the eComm Product Lo...	

New Key Qualifier

Cancel Add

Once the configuration is saved click on the same arrow down icon on the right of the **SKU** field and choose **View Key Qualifiers** option now. You should see 2 records presented as per screenshot below:

Key Qualifier Details

Select a key qualifier to view all the fields that share the same key qualifier.

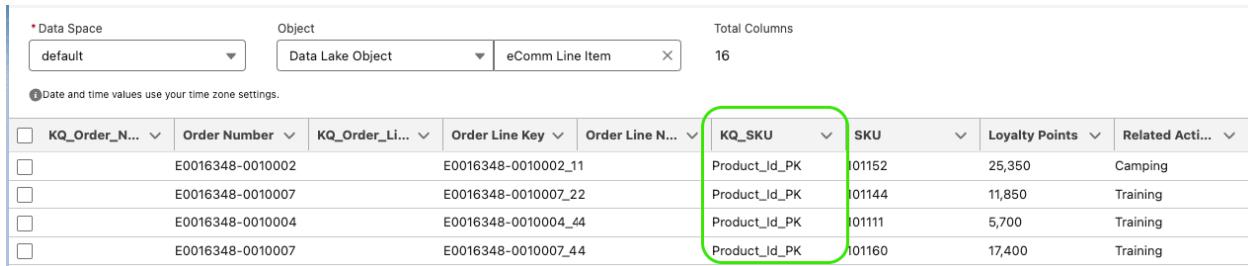
Key Qualifiers	Key Qualifier API Name	Key Qualifier Label	Data Lake Object Label	Data Lake Object Field Label
Select...	Product_Id_PK	Product Id PK	eComm Line Item	SKU
	Product_Id_PK	Product Id PK	eComm Product Lookup	Product Id

Search key qualifier details...

Close

This change will now enable appropriate join between **Goods Product** and **Sales Order Product** DMOs. If you return to Data

Explorer now and check the **eComm Line Item** object you'll notice that **KQ_SKU** field now includes Product_Id_PK value:



Data Space		Object	Total Columns
default	Data Lake Object	eComm Line Item	16
Date and time values use your time zone settings.			
KQ_Order_N...	Order Number	KQ_Order_Li...	Order Line Key
<input type="checkbox"/>	E0016348-0010002		E0016348-0010002_11
<input type="checkbox"/>	E0016348-0010007		E0016348-0010007_22
<input type="checkbox"/>	E0016348-0010004		E0016348-0010004_44
<input type="checkbox"/>	E0016348-0010007		E0016348-0010007_44
KQ_SKU	SKU	Loyalty Points	Related Acti...
Product_Id_PK	01152	25,350	Camping
Product_Id_PK	01144	11,850	Training
Product_Id_PK	01111	5,700	Training
Product_Id_PK	01160	17,400	Training

And if you've reserved making the changes until testing the segment counts you should be able to return to the segmentation activity and get the expected counts without making any changes to the segment criteria.

 **What we achieved here?** Let's have a quick reflection on what impact key qualifier configuration has.

This part of the exercise is a reminder to pay special attention when configuring key qualifiers, as not only single source field might need to be included in the configuration, but also any other field that is then linked to it via data relationship. The challenge this exercise highlights is that key qualifier configuration is done on the DLOs, while data relationships are configured for the DMOs. This makes it hard to instantly and intuitively identify needed setup, and rather requires planning ahead at the design time to review and consider fields that will require key qualifiers, their place in the data model and identification any other relations that will make use of those values in other objects.

Configure Identity Resolution:

Identity Resolution is responsible for producing a [Unified Individual](#) that combines all profile data, aggregates all contact points and is linked to check, engagement and transactional data. This exercise will walk you through the steps needed to configure matching and reconciliation rules.

1. Validate Data Mapping

Before configuring resolution rules review configuration of the key data sources that will be used in the resolution.

Navigate to **Data Model** tab and open the **Contact Point Email** object. Verify that you have Contact_<ORG_ALIAS> and eComm Customer Profile data lake objects mapped:

The screenshot shows the 'Details' tab selected for the 'Contact Point Email' object. The 'Details' section is expanded, displaying the following information:

Object Label	Contact Point Email	Object API Name	ssot__ContactPointEmail__dlm
Category	OTHER	Created By	Automated Process, Thu Jul 25 2024
Mapped data streams	Contact_Home, eComm Customer Profile	Mapped data lake objects	Contact_Home, eComm Customer Profile

Also verify that under **Relationships** tab there is configured relationship to Individual object via Party field:

Relationships							
Object	Field	Key Qualifier (Fie...	Cardinality	Related Object	Related Field	Key Qualifier (Rel...	
1 Contact Point Email	Party	KQ_PartyId	ManyToOne	Individual	Individual Id	KQ_Id	Edit

Similar to that open **Party Identification** object and verify that you have Contact_<ORG_ALIAS>, eComm Customer Profile and Party Identification Collection data lake objects mapped:

Details	
▼ Details	
Object Label	Object API Name
Party Identification	ssot__PartyIdentification__dlm
Category	Created By
OTHER	Automated Process, Thu Jul 25 2024
Mapped data streams	Mapped data lake objects
Contact_Home , eComm Customer Profile	Contact_Home , Party Identification Collection , eComm Customer Profile

And verify that under **Relationships** tab there is configured relationship to Individual object via Party field:

Relationships							
Object	Field	Key Qualifier (Fie...	Cardinality	Related Object	Related Field	Key Qualifier (Rel...	
1 Party Identification	Party	KQ_PartyId	ManyToOne	Individual	Individual Id	KQ_Id	Edit

2. Configure Resolution Rules

Navigate to the **Identity Resolutions** tab and create new ruleset with following attributes:

- Data Space = default
- Primary Data Model Object = Individual

- Ruleset ID = leave blank
- Ruleset Name = Main
- Ruleset Description = Main ruleset with email and party identification matching rules
- Switch Run jobs automatically into disabled state

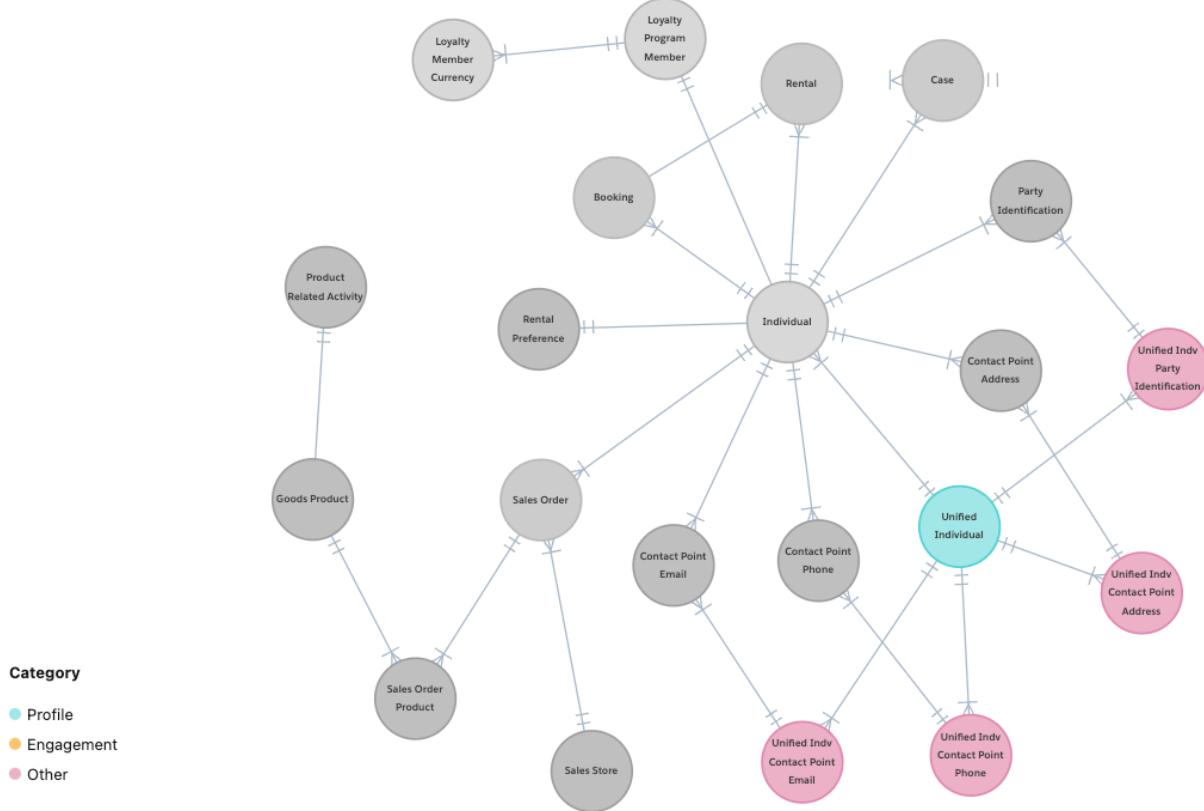
Take a note of the entity objects that are expected to be created once the ruleset is created:

Data Model Object	Data Model Object API Name	Input Data Model Object
1 Unified Individual	UnifiedIndividual	Individual
2 Unified Indv Contact Point Email	UnifiedContactPointEmail	Contact Point Email
3 Unified Indv Contact Point Phone	UnifiedContactPointPhone	Contact Point Phone
4 Unified Indv Contact Point Address	UnifiedContactPointAddress	Contact Point Address
5 Unified Indv Party Identification	UnifiedPartyIdentification	Party Identification

Save ruleset.

At this stage it's worth validating that necessary objects have been configured successfully. Navigate to **Data Model** tab, switch to the **Graph** view and review the schema confirming that new Unified ... objects have been created (represented

in color on the screenshot) and that **Unified Individual** object has relationships to **5** objects:



Note: you might need to refresh the screen if you see the objects but not all relationships have been configured.

As you can see there are just **5** initially listed, and that is expected. As a result of Identity Resolution process the system will produce unified individual profiles, and link those with associated records. By the time we configure Identity Resolution then relevant relationships will be created for these records.

Now return to the **Identity Resolution** tab to configure the Main ruleset with the following Match Rules:

- Rule 1:
 - Match Rule = Fuzzy Name and Normalized Email
 - Add additional criteria:
 - Object = Individual
 - Field = Birth Date
 - Match Method = Exact
 - Check the Match on Blank option
 - Update name = Fuzzy Name and Normalized Email and Exact Birth Date

⚠ This is exactly the point where earlier manipulations with mapping Birth Date field from various data streams could've led us into difficulty. As you can now experience the rule configuration you can understand why we needed to consolidate all variations into a single field.

- Add another match rule using Custom Rule option:
 - Set criteria as follows:
 - Object = Party Identification
 - Field = Identification Number

- Match Method = Exact
 - Party Identification Type = Loyalty Program
 - Party Identification Name = Airline Membership
- Update name = Exact Party ID for Airline Loyalty Program
- Add another match rule using Custom Rule option:
 - Set criteria as follows:
 - Object = Party Identification
 - Field = Identification Number
 - Match Method = Exact
 - Party Identification Type = Loyalty Program
 - Party Identification Name = Motor Club Membership
 - Update name = Exact Party ID for Motor Club Loyalty Program
- Add another match rule using Custom Rule option:
 - Set criteria as follows:
 - Object = Party Identification
 - Field = Identification Number

- Match Method = Exact
 - Party Identification Type = Loyalty Program
 - Party Identification Name = RAVG Retail Membership
- Update name = Exact Party ID for RAVG Retail Loyalty Program
- Add another match rule using Custom Rule option:
 - Set criteria as follows:
 - Object = Party Identification
 - Field = Identification Number
 - Match Method = Exact
 - Party Identification Type = Person Identification
 - Party Identification Name = Driver License
 - Update name = Exact Party ID for Driver License Person Identifier

Ignore the warning about consideration to add more criteria in the Party Identification rules and save the configuration

 . The matching rules summary should look like the example below:

Match Rules

Edit

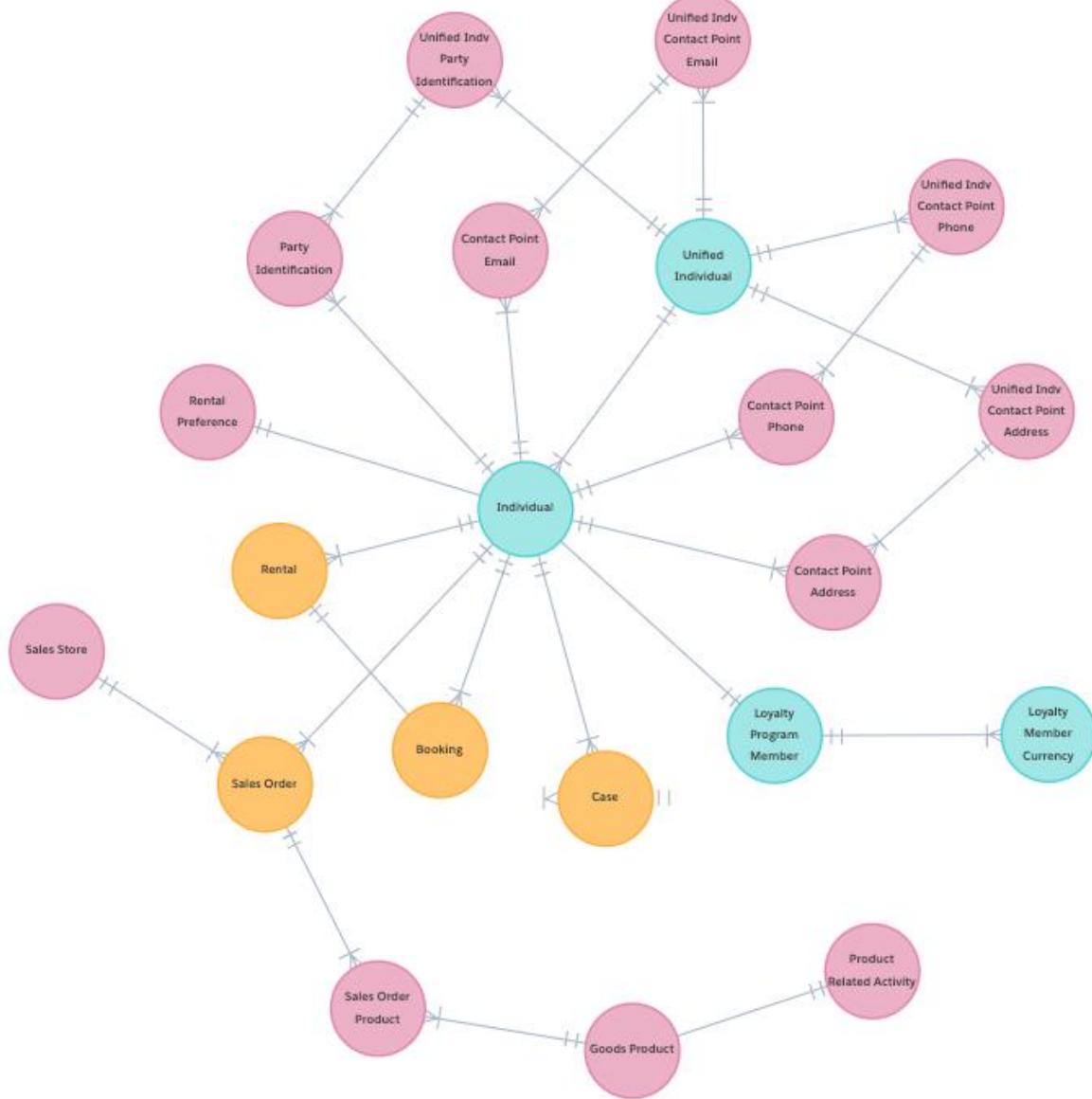
Fuzzy Name and Normalized Email and Exact Birth Date **OR** **⚠️** Exact Party ID for Airline Loyalty Program **OR** **⚠️**
Exact Party ID for Motor Club Loyalty Program **OR** **⚠️** Exact Party ID for RAVG Retail Loyalty Program **OR** **⚠️** Exact
Party ID for Driver License Person Identifier

Take a note that once you save the match ruleset the status will change to Publishing and then subsequently change to Published.

The screenshot shows the 'Identity Resolution Main' interface. At the top, there's a header with a user icon and the text 'Identity Resolution Main'. Below the header, there are four columns of information:

Data Space	Primary Data Model Object	Ruleset ID	Ruleset Status
default	Individual		Publishing

Right now we should confirm that all of the Unified ... objects have been interlinked (no orphans) similar to the example below:



💡 What we achieved here? Let's have a quick reflection on what that identity resolution configuration implies.

First to differentiate **Match Rules** from **Match Criteria**. Think of rules as a collection of criteria that instruct the matching

process on how records from **Individual DMO** should be evaluated.

Any given **match rule** can have one or more **match criteria** that are combined into a logic using AND logical condition. So reviewing first rule that we just configured it instructs that in order for two records to be matched they need to have matching **First Name AND Last Name AND Email Address AND Birth Date**. The methods of matching instruct how precise the comparison should be and whether or not original values need to be “normalized” before matching. As a result when you consider design of the match rules consider if how strict your instructions are based on the number of criteria that are included in a match rule. More criteria results in stricter matching, potentially leading to lower matching rates. And having less criteria leads to more relaxed matching resulting in potentially higher matching rates.

Next we configured additional match rules, and as you no doubt noticed these are combined into a logic using OR logical condition. This means that as the identity resolution processing tries to match records following criteria it goes over the list of rules to try and find match. As a result of

logical OR the more rules you configure the higher is the likelihood to find a match. This is because there are more ways to stack and compare records against each other. And subsequently this leads to potentially higher match rate, of course subject to criteria defined within each rule.

There is no universal reference for what the ideal matching rate should be. Every customer context and use cases are different and besides logical decisions a lot depends as you might guess on their actual data. But the balance is ultimately expresses the level of trust that your customers will be happy with when various combinations of criteria and rules are defined. This is one of the areas where it's worth spending good amount of time to audit source data and design and agree the proposed configuration with your customers to unlock and enable the resulting use cases.

3. Configure Reconciliation Rules

Back to the **Identity Resolution** tab. For our scenario the default configuration of the reconciliation rules using Most Frequent is suitable for all data points as we did not configure any additional data sources for the records that represent individual people and their profile and demographic attributes. In real implementation you'll most

likely have more than a single source of profile data and therefore it is important to configure rules that define how attributes for the unified profile will be constructed from all sources.

Update the rules following the steps below:

- **Contact Point Address** and **Contact Point Phone** - leave without modifications as we only have a single source feeding each of these data model objects
- **Contact Point Email**
 - update the **Default Rule** setting clicking on the  icon next to it, changing from Most Frequent to Source Priority with source **Contact_<ORG_ALIAS>** being in the first place:

Total Unified Profiles: 0

Edit Default Reconciliation Rule for Contact Point Email

This reconciliation rule applies to all fields in this data model object by default. Any standard or custom fields you map to this object inherit this rule. You can change the rule for specific fields by editing them individually.

Default Reconciliation Rule

Source Priority	<input checked="" type="checkbox"/> Ignore Empty Values
-----------------	---

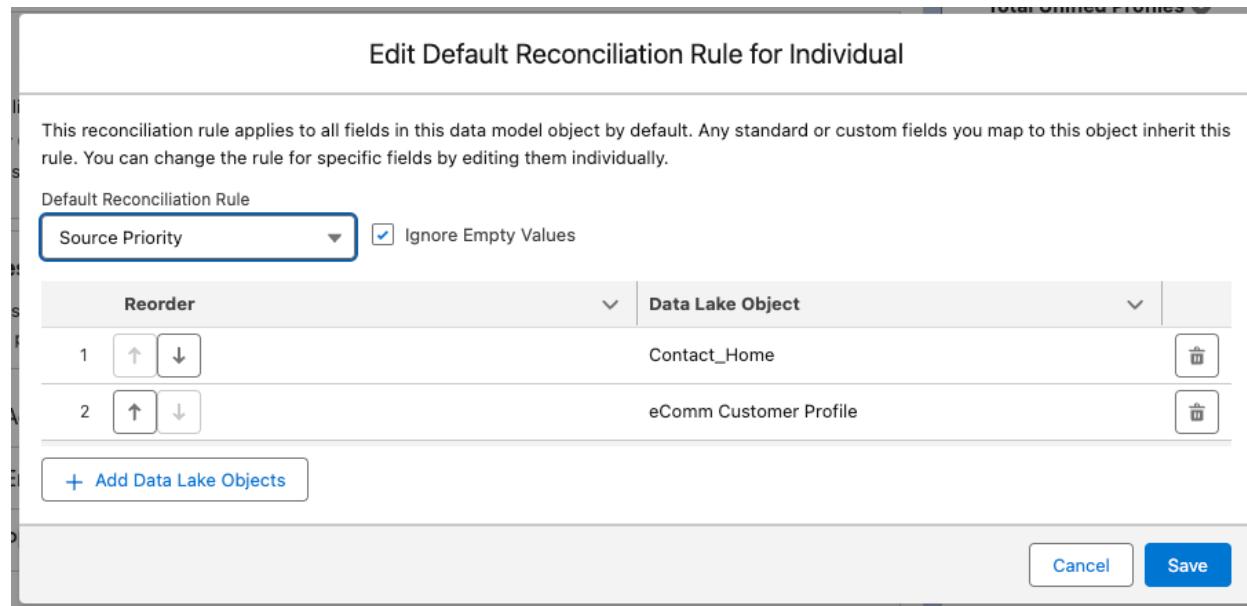
Reorder

	Data Lake Object	
1	Contact_Home	
2	eComm Customer Profile	

+ Add Data Lake Objects

Cancel Save

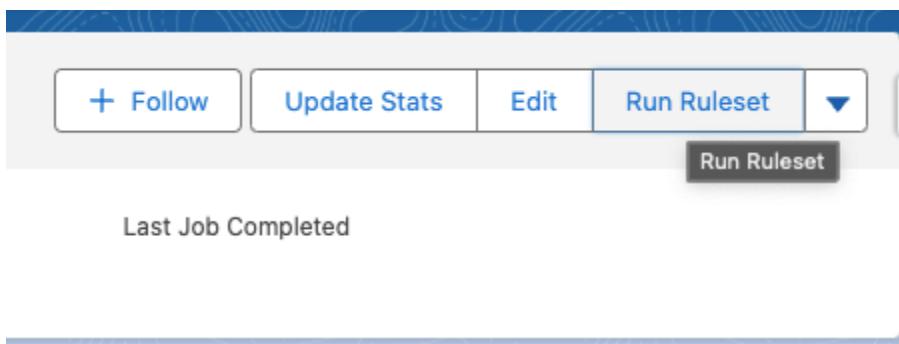
- Save and repeat the above steps for **Individual** attributes:



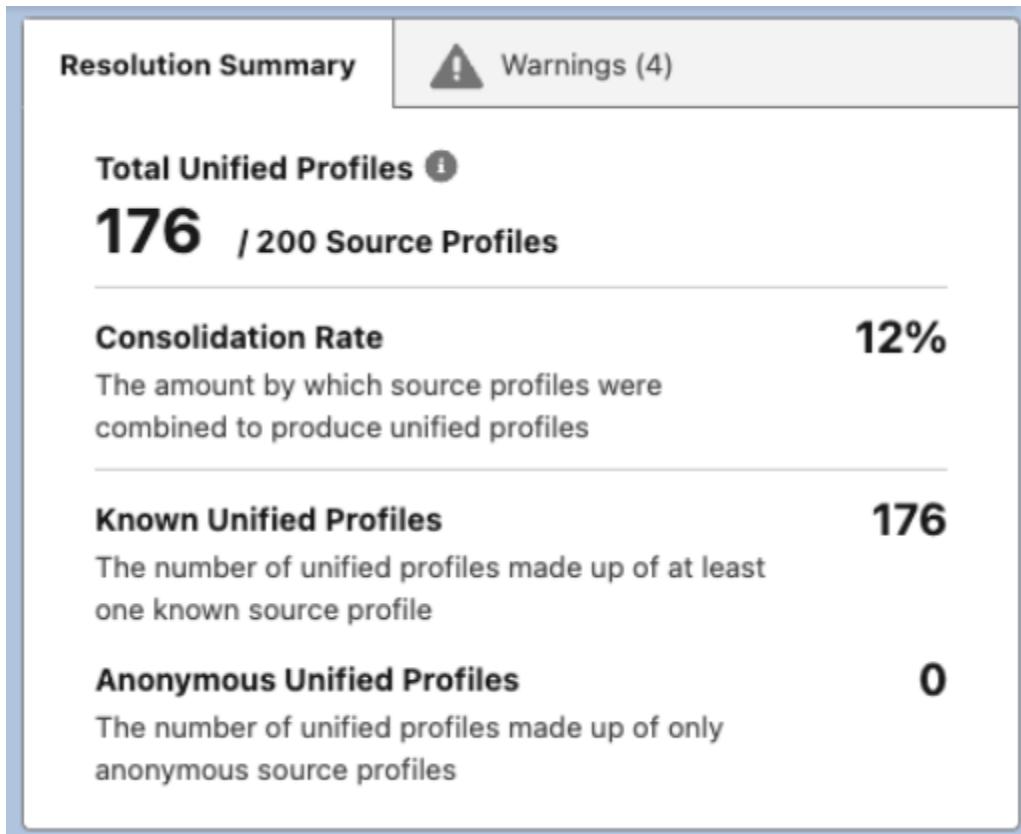
Leave the **Party Identification** setup to follow default configuration.

4. Identity Resolution Summary

Initiate Identity Resolution process by clicking on the **Run Ruleset** action button in the top right area of the screen:



As a result the **Last Job Status** should be updated to In Progress. Once the processing of the rules starts it takes some time (up to 30 mins) and upon completion you will be able to see the summary outlining the counts of sources and matched individuals as well as total number of unified profiles created as per the following example:



You can also see the statistics for the resolution job in the **Processing History** tab:

Ruleset Properties	Details	<u>Processing History</u>																
Daily summaries contain the aggregate results of all runs of this ruleset from a single date.																		
Automatic runs: Disabled Daily Processing Summary																		
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Run Date</th><th>Total Source P...</th><th>Total Unified P...</th><th>Total Known P...</th><th>Consolidation ...</th><th>Total Unknown...</th><th>Processed Re...</th><th>Aggregate Sta...</th></tr> </thead> <tbody> <tr> <td>1 2024-04-29</td><td>200</td><td>176</td><td>176</td><td>12%</td><td>0</td><td>200</td><td>Succeeded</td></tr> </tbody> </table>			Run Date	Total Source P...	Total Unified P...	Total Known P...	Consolidation ...	Total Unknown...	Processed Re...	Aggregate Sta...	1 2024-04-29	200	176	176	12%	0	200	Succeeded
Run Date	Total Source P...	Total Unified P...	Total Known P...	Consolidation ...	Total Unknown...	Processed Re...	Aggregate Sta...											
1 2024-04-29	200	176	176	12%	0	200	Succeeded											

To understand the results of the reconciliation in our scenario consider the fact that we are including and reconciling individual records coming from a two sources, Salesforce CRM Contact object and S3 file. These records are unique at each respective source, but they were crafted to provide some examples that demonstrate how records with various degrees of data match have been handled by the identity resolution process.

The expected count of **Known Unified Profiles** should be = 176. This is a total number of records that comprises from both Salesforce CRM and S3 data sources. Based on that the **Consolidation Rate** should be = 12%. This is how many records were matched between Salesforce CRM profiles and profiles originated from S3 as well as some of the profiles that were matched across Salesforce CRM records only.

4.1 Different Results in Resolution Summary

It is possible that your results may vary from the above detailed resolution summary. One of the common scenarios

is when you get **Known Unified Profiled** = 189 and **Consolidation Rate** = 6%. If you get this result then most likely cause to it has to do with **Birth Date** match criteria, because either data mapping or configuration of the custom field was done incorrectly.

For any other results the cause would be similar - either mistake in data mapping or misconfiguration of custom fields.

In order to troubleshoot any result that is different from expected follow the next section to see if you can identify the cause.

5 Explore Computed Unified Profiles

In the **Data Modelling** and **Identity Resolution** lessons you already get introduced to the various tools that enable exploration of the data in Data Cloud. The following steps guide you through the use of some of the tools in order to explore specific unified (reconciled) profile records. The objective is to see the results of the Identity Resolution configuration we performed in the current exercise.

To begin with review the following table with the extract of certain records from two data streams that we want to inspect in Data Cloud. Notice the variance in field values highlighted for convenience in addition to the emails in S3 feed are all enclosed in <> characters. For the cases where only CRM records are compared the differences are highlighted in different color.

From CRM Contact:

R	Cust			Fir	Las	h	bers
ef	omer			st	t	D	hip
#	Exter			Na	Na	at	Num
	nal ID	Email		me	me	e	ber
C	c0054	WSTEVENSON.10000.005	Wil	Stev	1		
R	cf8-	4@hotmail.marketingclou	l	ens	9		
M	904e-	ddemo.com		on	9		
-1	4475-					1-	
	a20b-					0	

	02fde 98f09 d4				9- 1 5	
C R M -2	c0068 c86- ef05- 4513- bc55- 316e4 56520 06	KSTEVENSON.10000.0068 @gmail.marketingcloudde mo.com	Ca rl	Stev ens on	1 9 7 2- 0 6- 0 3	
C R M -3	c0071 c1c- 8110- 4389- abd4- 2bb3 b444 bdf1	AFRIES.10000.0071@hotmail.com	Au gu st	Frie s	1 9 5 1- 0 9- 0 6	9251 2679 1 1- 0 9- 0 6
C R	c0076 c39- 9e73-	VCHANDLER.10000.0076 @yahoo.marketingclouddemo.com	Vir gili o	Cha ndl er		5021 2683 5

M	4d0c-							
-4	920b-							
	78728							
	a73e3							
	e4							
C	c0072	CMICHAEL.10000.0072@y	Ca	Mic	1			
R	c54-	ahoo.marketingclouddem	roll	hael	9			
M	5ad2-	o.com		Ali		7		
-5	4e5b-			ce		3-		
	acbb-					0		
	b71c9					6-		
	44e6f					0		
	61					1		
C	ctravc	travelcaroll@gmail.market	Ali	Mic				
R	d6-	ingclouddemo.com	ce	hael				
M	3dbf-							
-6	4671-							
	8192-							
	df8e3							
	e6041							
	bf							

C	c0080	SCHUN.10000.0080@hot mail.marketingclouddemo .com	Ste phen	Chu n	1 9 8 2- 0 3- 1 3		
C	cstev	stevech@yahoo.marketing clouddemo.com	Ste ve	Chu n			
C	c0082	VALLRED.10000.0082@ho tmail.marketingclouddem .com	Ver ma	Allr ed	1 9 8 8- 0 5-		

	032f6 6d				0 9	
C	cVALL	VALLRED@gmail.marketingclouddemo.com	Ver	A.		3236
R	c3-		na			5898
M	5d39- - 44b0- 1 aabf- 0 f2f90 cb7d d22					9
C	c0120	kivey.10000.0120@yahoo.marketingclouddemo.com	Kry	Ivey	1	
R	cb3-		sti		9	
M	a749- - 4844- 1 9ab2- 1 a48e1 00a03 2c		na		7	
					6-	
					0	
					1-	
					1	
					9	
C	c0121	ltrammell.10000.0121@yahoo.marketingclouddemo.com	Lel	Tra	1	7207
R	c86-		ah	mm	9	2066
M	0acf- - 497d-			ell	8	1
					1-	

1	bf06-				1	
2	308cc				2-	
	2d75f				2	
	8f				5	

From S3 Customer Profile feed

R	e	Custom	f	er Ref	#	Number	Email	Fir	st	Sur	Na	na	at	me	me	e	ID
S	0032700	<WSTEVENSON.10000.00	3	000ay71	-	zAAA	54@hotmail.marketingclo	Wil	Ste	1							
		uddemo.com>						fre	ven	9							
								do	son	9							

S 3 -	0032700 000xsJU - UAA2 2	<KSTEVENSON.10000.006 8@gmail.marketingclouddemo.com>	Kar l	Ste ven son	1 9 7 2- 0 6- 0 3	
S 3 -	0032700 000elh3 - mAAA 3	<AFRIES.10000.0071@gm ail.marketingclouddemo.c om>	Au gu st	Frie s	1 9 5 1- 0 9- 0 6	925 126 791
S 3 -	0032700 0000xfTJ - AAY 4	<VCHANDLER.10000.007 6@yahoo.marketingcloud demo.com>	Vir gili o	Cha ndl er	1 9 9 8- 0 4-	502 126 835

						1	
						7	
S 3 -	0032700 000TPwF oAAL 5	<CMICHAEL.10000.0072 @yahoo.marketingcloudd emo.com>	Ca roll	Mic hae l	1 9 7 3- 0 6- 0 1		
S 3 -	0032700 000iq02 YAAQ 6	<SCHUN.10000.0080@ho tmail.marketingclouddem o.com>	Ste ph en	Ch un	1 9 8 2- 0 3- 1 3		
S 3	0032700 00037Z3 SAAU	<VALLRED.10000.0082@h otmail.marketingcloudde mo.com>	Ver na	Allr ed	1 9 8	323 658 989	

-						8-	
7						0	
						5-	
						0	
						9	

In addition to that we also need to account for the **CRM Rentals** data stream, as it provides records for the party identification matching based on few membership options or driver license number. Below is the list of the relevant rental records with unique permutation of key identifiers noted in one or more rentals:

R e f #	Externa l ID	Custo mer Externa l ID	CR M Con tact Ref	RAVG Retail Memb ership Numb er	Airline Memb ership Numb er	Motor Club Memb ership Numb er	Driv er Lice nse
R -1	r003f10 e-bf8c- 4c0a- 8c71-	ctravcd 6-3dbf- 4671- 8192-	CR M-6	FGH- 3686- 61759			N15- 2546 6146

	136633 2cd1a9	df8e3e 6041bf					
R - 2	r004f10 - 2-dabe- 4f89- acc3- 274320 b7d2b6	c0072c 54- 5ad2- 4e5b- acbb- b71c94 4e6f61	CR M-5	FGH- 3686- 61759	O90- 3206 2748		
R - 3	r007f3c - e-2fe3- 4410- a447- 781cc6 bfe56e	c0082c 44- 1841- 499f- b6cb- 894200 32f66d	CR M-9	32365 8989		E25- 0835 9644	
R - 4	r010ab - ae- 652a- 45f0- 96aa-	c0080c 00- e036- 466f- 8e34-	CR M-7			O90- 3206 2748	

	35a77f7 18c42	e8c046 8cd17f				
R - 5	r011f66 2-0226- 4d7d- 9b15- c0e132 b978f4	cstevcb b-be16- 41d5- ab9f- 3ad106 e919d1	CR M-8			O90- 3206 2748
R - 6	r019df6 1-c9d0- 44ce- 94bf- 77eb46 43c8a1	c0120c b3- a749- 4844- 9ab2- a48e10 0a032c	CR M- 11		RA- 30900 8573	P58- 1925 9175
R - 7	r021a3c 2-f770- 4488- b321- c13b64 867c2a	c0121c 86- 0acf- 497d- bf06- 308cc2 d75f8f	CR M- 12		RA- 30900 8573	R00- 8665 9649

Taking into account the matching rules that we configured in the Identity Resolution the following outcomes are **expected** to be observed comparing corresponding rows from each source:

- CRM-1 and S3-1 records to be matched on “fuzzy” name (slightly different spelling of both) + normalized email (discards <> from S3 stream) + same last name and Birth Date.
- CRM-2 and S3-2 records to be matched following same rules as above
- CRM-3 and S3-3 records to be matched through the party identification (same loyalty number) despite both having different email address.
- CRM-4 and S3-4 records to be matched through the party identification (same loyalty number) despite one (CRM-4) missing birth date.
- CRM-5 and S3-5 not matched due to the difference in the first name, as it is not just misspelling but rather two word name compared to single word name.
- CRM-6 does not have a matching profile in S3 customers set, but it was designed as a match to the CRM-5 through the party identification. Direct comparison of these records will definitely result in not

matched outcome. Although once we inspect rental records we will find following details:

- Rentals R-1 and R-2 are linked to the different individuals and have different driver license numbers, but they both share the same airline membership number. This match should establish link between respective individuals leading to CRM-5 and CRM-6 being matched.
- CRM-7 and S3-6 records to be matched as all fields used in matching contain the same values
- CRM-8 does not have a matching profile in S3 customers set, and similar to CRM-6 was designed as a match for CRM-7 via party identification. From the rental records:
 - R-4 and R-5 have matching driver license numbers, therefore CRM-8 and CRM-7 to be matched
 - This will also lead to CRM-8 to be matched with S3-6 record
- CRM-9 and S3-7 not matched due to the difference (typo) in the first name and CRM-9 missing the loyalty number.

- CRM-10 and S3-7 records to be matched through the party identification (same loyalty number) despite both having different details in other fields.
- CRM-9 has a rental record R-3 that includes loyalty number matching it as a result to CRM-10 and S3-7
- CRM-11 and CRM-12 have different details, but rental records R-6 and R-7 include same airline membership number. This might be two family members sharing the code, and while having every other detail different ultimately result in matching outcome between these individuals.

So visualizing the above into the simple table with reference to each case:

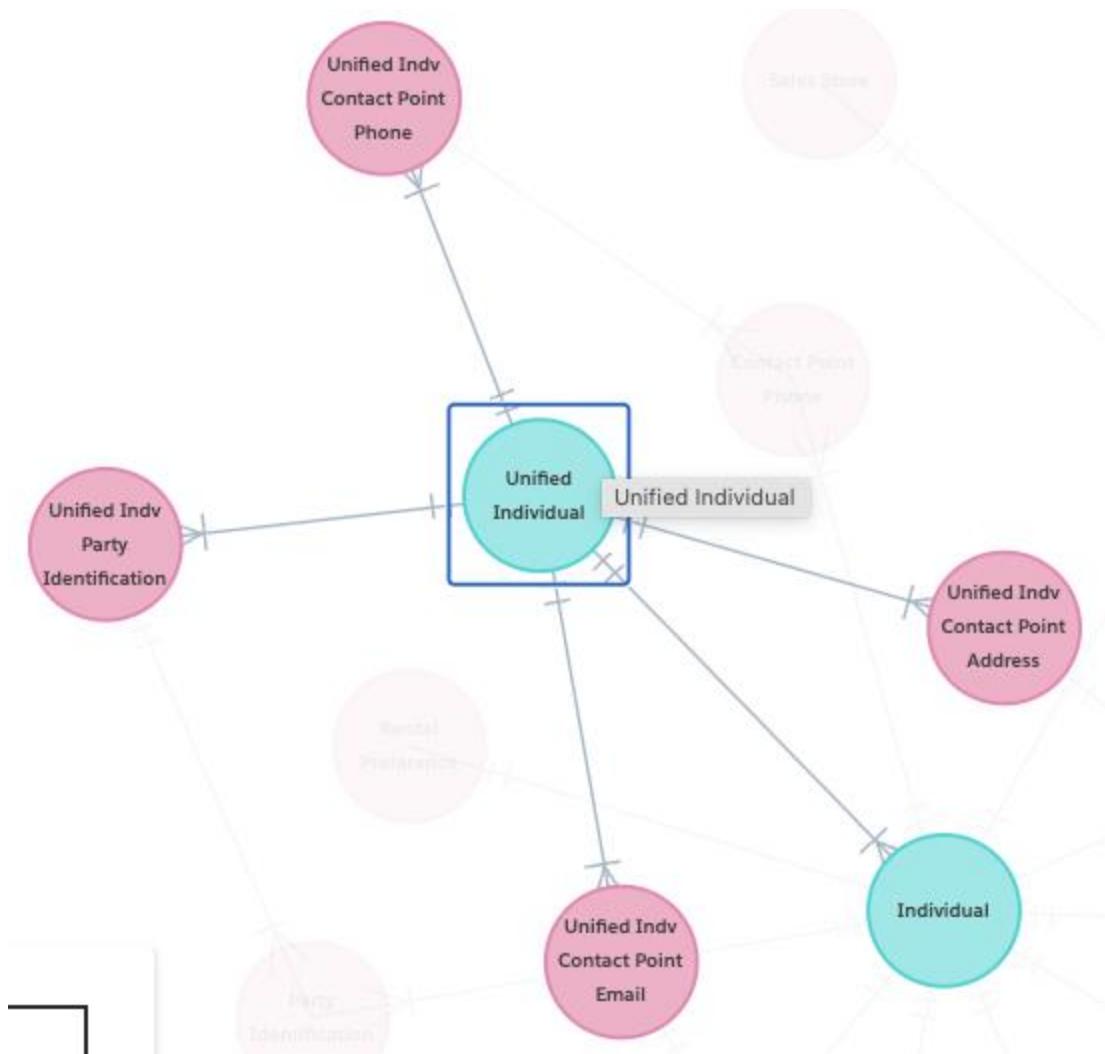
Match Set 1	CRM-1	S3-1	
Match Set 2	CRM-2	S3-2	
Match Set 3	CRM-3	S3-3	
Match Set 4	CRM-4	S3-4	
Non-Match Set 1	CRM-5	S3-5	
Match Set 5	CRM-5	CRM-6	

Match Set 6	CRM-7	S3-6	CRM-8
Match Set 7	CRM-10	S3-7	CRM-9
Match Set 8	CRM-11	CRM-12	

Total expected count of Unified Individual records from the above set = **8**

5.1 Understanding Value Of Bridge Objects In Data Model

In order for you to explore the results, it's worth revising the model to understand what objects store what information. If we inspect the data model in the graph view you'll notice **Individual** object is shown as directly connected to **Unified Individual**:



Under relationships tab for **Individual** object there is no link to the **Unified Individual**, and instead there is relationship configured to the **Unified Link Individual** object:

The screenshot shows a Data Model Object (Individual) page. At the top, there is a header with a person icon, the text "Data Model Object Individual", and buttons for "Edit" and "Delete". Below the header, there is a table with columns: Type, Object Status, Mapped data streams, and Mapped data lake objects. The values are Standard, Ready, 2, and 2 respectively. Under the "Relationships" tab, there is a table titled "Relationships" with columns: Object, Field, Key Qualifier (Field), Cardinality, Related Object, Related Field, and Key Qualifier (Related Field). The table lists 11 relationships, such as Booking to Customer, Case to Account Contact, and Sales Order to Sold To Customer.

Locating the **Unified Link Individual** object and reviewing it's details and relationships you can see that it is of type **Bridge** and that it “links” **Individual** with **Unified Individual** objects:

The screenshot shows a Data Model Object (Unified Link Individual) page. At the top, there is a header with a person icon, the text "Data Model Object Unified Link Individual", and a dropdown arrow. Below the header, there is a table with columns: Type, Object Status, Mapped data streams, and Mapped data lake objects. The values are Bridge, Ready, 0, and 0 respectively. Under the "Relationships" tab, there is a table titled "Relationships" with columns: Object, Field, Key Qualifier (Field), Cardinality, Related Object, Related Field, and Key Qualifier (Related Field). The table lists 2 relationships, such as Individual to Individual Id and Unified Link Individual to Unified Individual Id.

This is how all of these objects come together in relation to the Identity Resolution:

DMO	Type	Purpose / Description
-----	------	-----------------------

Individual	Profile	<p>Collection of profile records for individuals that are sources from one or more Profile category data lake objects mapped to it and intended to be unified as a result of Identity Resolution process.</p> <p>Corresponds to 200 Source Profiles in the Identity Resolution Summary</p>
Unified Individual	Profile	<p>Profile records created as a result of unification of Individual profile records. Have system generated mutable identifier.</p> <p>Corresponds to 176 Total Unified Profiles in the Identity Resolution Summary</p>

Unified Link Individual	Bridge	<p>Bridge object that contains links between Individual and Unified Individual records, populated as a result of Identity Resolution process.</p> <p>Based on number of links between unique Individual and Unified Individual records the value of Consolidation Rate can be established in the Identity Resolution Summary.</p> <p>This object is the target for inspection of the unification results, investigating how individual records are matched and what records are linked to a given unified profile.</p>
--------------------------------	--------	---

Interpreting the descriptions of the above objects in order to identify and validate our expected matched sets (8 records) we would need to follow this approach:

1. Locate records in **Individual** DMO with specific **External Record Id** values
2. Note down **Individual Id** values from the identified result set

3. Locate records in **Unified Link Individual** DMO for the identified id's
4. Compare actual results against planned, establish and explore any discrepancies to understand why the differences have been introduced.

We will perform these explorations using combination of tools to get you familiar with options.

5.2 Performing Search Of Selected Records for Validation Purposes

In order for us to inspect the profiles we need to begin with getting the corresponding **Individual Id** for the records originated from **CRM Contact** object as these will be unique to your org. Navigate to the **Data Explorer** tab and using either Data Lake Object > Contact_<ORG_ALIAS> or Data Model Object > Individual locate records for the corresponding **Customer External ID** values from the [CRM Contact table](#) above:

The screenshot shows the 'Data Explorer Objects' interface. At the top, there is a header with a bar chart icon and the text 'Data Explorer Objects'. Below the header, there are three filter fields: 'Data Space' set to 'default', 'Object' set to 'Data Model Object' with 'Individual' selected, and 'Total Columns' set to '13'. The 'Individual' selection in the Object field is highlighted with a blue border.

Here is the list of all external ID values that you can paste into the filter criteria:

c0054cf8-904e-4475-a20b-02fde98f09d4,c0068c86-ef05-4513-bc55-316e45652006,c0071c1c-8110-4389-abd4-2bb3b444bdf1,c0076c39-9e73-4d0c-920b-78728a73e3e4,c0072c54-5ad2-4e5b-acbb-b71c944e6f61,ctravcd6-3dbf-4671-8192-df8e3e6041bf,c0080c00-e036-466f-8e34-e8c0468cd17f,cstevcbb-be16-41d5-ab9f-3ad106e919d1,c0082c44-1841-499f-b6cb-89420032f66d,cVALLc3-5d39-44b0-aabf-f2f90cb7dd22,c0120cb3-a749-4844-9ab2-a48e100a032c,c0121c86-0acf-497d-bf06-308cc2d75f8f

Internal Organization Last Modified

NULL	4/15/2017
Field: External Record Id Operator: in Value: c0054cf8-904e-4475-a20b-C	
Done	
NULL NULL NULL NULL NULL	

Filter

Show Me All Records

Matching all of these filters

New Filter*

c0054cf8-904e-4475-a20b-C
 02fde98f09d4,c0068c86-ef05-4513-
 bc55-316e45652006,c0071c1c-8110-
 4389-abd4-2bb3b444bdf1,c0076c39-
 9e73-4d0c-920b-
 78728a73e3e4,c0072c54-5ad2-4e5b-
 acbb-b71c944e6f61,ctravcd6-3dbf-
 4671-8192-df8e3e6041bf,c0080c00-
 e036-466f-8e34-
 e8c0468cd17f,cstevccb-be16-41d5-
 ab9f-3ad106e919d1,c0082c44-1841-
 499f-b6cb-89420032f66d,cVALLc3-
 5d39-44b0-aabf-
 f2f90cb7dd22,c0120cb3-a749-4844-
 9ab2-a48e100a032c,c0121c86-0acf-
 497d-bf06-308cc2d75f8f

[Add Filter](#) [Remove All](#)

Adjusting the columns to display only details you need will give you results similar to the one below (order of records may vary in your org):

<input type="checkbox"/> Data Source Object	External Record Id	Individual Id	First Name	Last Name	Birth Date
<input type="checkbox"/> Contact	c0071c1c-8110-4389-abd4-2bb3b444bdf1	003Dp000004EZpMIAW	August	Fries	9/6/1951
<input type="checkbox"/> Contact	c0054cf8-904e-4475-a20b-02fde98f09d4	003Dp000004EZpElAW	Will	Stevenson	9/15/1991
<input type="checkbox"/> Contact	c0120cb3-a749-4844-9ab2-a48e100a032c	003Dp000004EZpvIAG	Krystina	Ivey	1/19/1976
<input type="checkbox"/> Contact	ctravcd6-3dbf-4671-8192-df8e3e6041bf	003Dp000004EZpaIAG	Alice	Michael	
<input type="checkbox"/> Contact	c0072c54-5ad2-4e5b-acbb-b71c944e6f61	003Dp000004EZpNIAW	Caroll Alice	Michael	6/1/1973
<input type="checkbox"/> Contact	c0068c86-ef05-4513-bc55-316e45652006	003Dp000004EZpJIAW	Carl	Stevenson	6/3/1972
<input type="checkbox"/> Contact	c0121c86-0acf-497d-bf06-308cc2d75f8f	003Dp000004EZpwIAG	Lelah	Trammell	12/25/1981
<input type="checkbox"/> Contact	c0080c00-e036-466f-8e34-e8c0468cd17f	003Dp000004EZpVIAW	Stephen	Chun	3/13/1982
<input type="checkbox"/> Contact	c0076c39-9e73-4d0c-920b-78728a73e3e4	003Dp000004EZpRIAW	Virgilio	Chandler	
<input type="checkbox"/> Contact	c0082c44-1841-499f-b6cb-89420032f66d	003Dp000004EZpXIAW	Verma	Allred	5/9/1988
<input type="checkbox"/> Contact	cVALLc3-5d39-44b0-aabf-f2f90cb7dd22	003Dp000004EZpYIAW	Verna	A.	
<input type="checkbox"/> Contact	cstevccb-be16-41d5-ab9f-3ad106e919d1	003Dp000004EZpZIAW	Steve	Chun	

5.2.1 First Verification Point For Different Results

If your identity resolution summary figures are as expected feel free to skip this subsection and [continue with the inspection](#) of the results.

If you have different results in resolution summary one of the reasons could be related to **Birth Date** field. So if your control records in the **Individual** DMO have empty values for the **Birth Date** field then this would be one explanation for the discrepancy in the results.

* Data Space
default

Object
Data Model Object Individual

Total Columns 13

Date and time values use your time zone settings.

	Data Source Object	External Record Id	Individual Id	First Name	Last Name	Birth Date
<input type="checkbox"/>	View Contact	c0054cf8-904e-4475-a20b-...	003al000009E7SIAAK	Will	Stevenson	
<input type="checkbox"/>	View Contact	c0068c86-ef05-4513-bc55-...	003al000009E7SqAAK	Carl	Stevenson	
<input type="checkbox"/>	View Contact	c0071c1c-8110-4389-abd4-2...	003al000009E7StAAK	August	Fries	
<input type="checkbox"/>	View Contact	c0072c54-5ad2-4e5b-acbb-...	003al000009E7SuAAK	Carroll Alice	Michael	
<input type="checkbox"/>	View Contact	c0076c39-9e73-4d0c-920b-...	003al000009E7SyAAK	Virgilio	Chandler	
<input type="checkbox"/>	View Contact	c0080c00-e036-466f-8e34-...	003al000009E7T2AAK	Stephen	Chun	
<input type="checkbox"/>	View Contact	c0082c44-1841-499f-b6cb-8...	003al000009E7T4AAK	Verma	Allred	
<input type="checkbox"/>	View Contact	cVALLc3-5d39-44b0-aabf-f...	003al000009E7T5AAK	Verna	A.	
<input type="checkbox"/>	View Contact	cstevccb-be16-41d5-ab9f-3a...	003al000009E7T6AAK	Steve	Chun	
<input type="checkbox"/>	View Contact	ctravcd6-3dbf-4671-8192-df...	003al000009E7T7AAK	Alice	Michael	
<input type="checkbox"/>	View Contact	c0120cb3-a749-4844-9ab2-...	003al000009E7TSAA0	Krystina	Ivey	
<input type="checkbox"/>	View Contact	c0121c86-0acf-497d-bf06-3...	003al000009E7TTAA0	Lelah	Trammell	

First step from here is to verify that your records were correctly ingested. Instead of Individual DMO choose **Contact_<ORG_ALIAS>** DLO as the object and filter the results using the same set of external ID values on the **External ID** field. Ensure to adjust columns to include both **Birth Date** (custom) and **Birthdate** (standard) fields:

* Data Space
default

Object
Data Lake Object Contact_Home

Total Columns 72

Date and time values use your time zone settings.

	External ID	Contact ID	First Name	Last Name	Birth Date	Birthdate
<input type="checkbox"/>	c0072c54-5ad2-4e5b-acbb-...	003al000009E7SuAAK	Carroll Alice	Michael		01/06/1973, 10:00 am
<input type="checkbox"/>	ctravcd6-3dbf-4671-8192-df...	003al000009E7T7AAK	Alice	Michael		03/06/1972, 10:00 am
<input type="checkbox"/>	c0068c86-ef05-4513-bc55-...	003al000009E7SqAAK	Carl	Stevenson		25/12/1981, 11:00 am
<input type="checkbox"/>	c0076c39-9e73-4d0c-920b-...	003al000009E7SyAAK	Virgilio	Chandler		09/05/1988, 10:00 am
<input type="checkbox"/>	c0121c86-0acf-497d-bf06-3...	003al000009E7TTAA0	Lelah	Trammell		15/09/1991, 10:00 am
<input type="checkbox"/>	cstevccb-be16-41d5-ab9f-3a...	003al000009E7T6AAK	Steve	Chun		13/03/1982, 10:00 am
<input type="checkbox"/>	c0082c44-1841-499f-b6cb-8...	003al000009E7T4AAK	Verma	Allred		06/09/1951, 10:00 am
<input type="checkbox"/>	c0054cf8-904e-4475-a20b-...	003al000009E7SIAAK	Will	Stevenson		19/01/1976, 11:00 am
<input type="checkbox"/>	c0080c00-e036-466f-8e34-...	003al000009E7T2AAK	Stephen	Chun		
<input type="checkbox"/>	c0071c1c-8110-4389-abd4-2...	003al000009E7StAAK	August	Fries		
<input type="checkbox"/>	c0120cb3-a749-4844-9ab2-...	003al000009E7TSAA0	Krystina	Ivey		
<input type="checkbox"/>	cVALLc3-5d39-44b0-aabf-f...	003al000009E7T5AAK	Verna	A.		

As you can see in the above screenshot the **Birth Date** field has no values, and is the field that we mapped to the respective field in the **Individual** DMO. So it appears that

mapping is not the issue here. We can also observe that **Birthdate** field has values for number of records, from which we can conclude that our configuration of the custom formula field is not working. To resolve this issue you will need to revisit configuration of the formula field for the **Contact_<ORG_ALIAS>** data stream and adjust appropriately.

Should the values be populated in this field then next logical step to validate would be revisiting the mapping of the same **Contact_<ORG_ALIAS>** data stream ensuring that the field was indeed mapped.

5.2.2 Continue Inspection Of Selected Records

Take a note of the **Individual Id** values from this result against each **External Record Id** in a way that you can subsequently copy-paste those values in other places for filtering purposes. It might help to have a table created with the similar structure, where you can start capturing your findings:

Validation Scenario	External Record Id	Individual Id	Unified Individual Id	Result As Expected?

Next, change the object to **Unified Link Individual** and using one of the Individual Id values noted in the previous step filter the results to one record for our exploration. Adjust columns limiting view to only needed one, with at minimum including **Data Source**, **Individual Id** and **Unified Individual Id**:

The screenshot shows the Data Explorer interface with the following details:

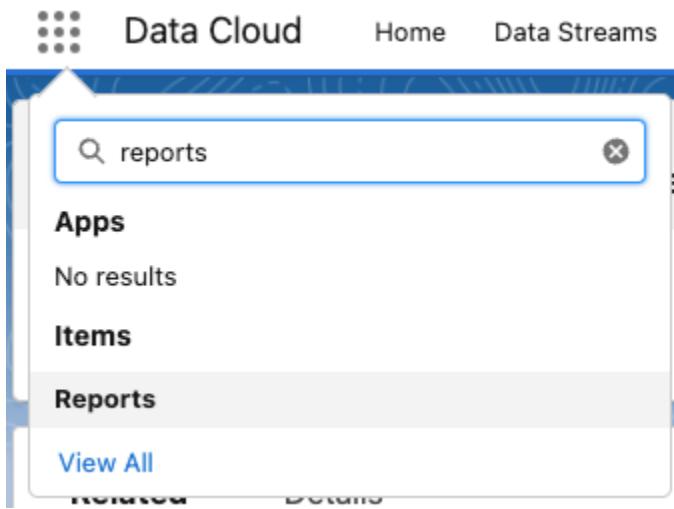
- Total Columns:** 8
- Filter:**
 - Show Me All Records
 - Matching all of these filters:
 - Individual Id in 003am000002IT8fAAE, 003am000002IT8kAAE, 003am000002IT8nAAE, 003am000002IT9NAAU
 - Add Filter
 - Add Filter Logic
- Data Explorer Objects:**
 - Data Space: default
 - Object: Data Model Object
 - Unified Link Individual
 - Total Columns: 8
 - Created Date: 05/09/2024, 09:33 am
 - Data Source: Salesforce_Home
 - Individual Id: 003al000009E7SzAAK
 - Unified Individual Id: eb78a515bf4ea83661fc8969d4160bd

While you can update filter to include all records of interest, you'll notice the pasting of the comma-separated list of ids leads to expansion of the UI beyond the screen view:

Don't let it stop you, as you'd want to take a note of all corresponding Unified Individual Id values and add them to the table where you capture results of the findings. At this stage you should notice that few of the entries have the same **Unified Individual Id** values for different **Individual Id**

values. This is expected as according to our expected results table we should have few records matched into one profile.

An alternative approach for getting these details is through Salesforce Reports.



Navigate to the **Reports** tab or if it's not presented in you org locate **Reports** in the application switcher and create new report:

Choose **Data Cloud** category and review available report types. Choose **Unified Link Individual with Individual** report type and start report configuration:

- Group rows by Unified Individual Id
- Include following columns:
 - Individual Id
 - External Record Id

- Data Source
- First Name
- Last Name
- Birth Date
- For convenience include filter on External Record Id equals and then paste the list of the identifiers we used earlier:
 - c0054cf8-904e-4475-a20b-02fde98f09d4,c0068c86-ef05-4513-bc55-316e45652006,c0071c1c-8110-4389-abd4-2bb3b444bdf1,c0076c39-9e73-4d0c-920b-78728a73e3e4,c0072c54-5ad2-4e5b-acbb-b71c944e6f61,ctravcd6-3dbf-4671-8192-df8e3e6041bf,c0080c00-e036-466f-8e34-e8c0468cd17f,cstevcbb-be16-41d5-ab9f-3ad106e919d1,c0082c44-1841-499f-b6cb-89420032f66d,cVALLc3-5d39-44b0-aabf-f2f90cb7dd22,c0120cb3-a749-4844-9ab2-a48e100a032c,c0121c86-0acf-497d-bf06-308cc2d75f8f
- Save and run the report naming it Unification results for validation. Disable **Subtotals** and **Grand Total** switches at the bottom of the report:



- The results will provide you with details (Unified Individual Id + Individual Id) that we otherwise collected through multiple separate steps:

Report: Unified Link Individual with Individual Unification results for validation

Total Records
12

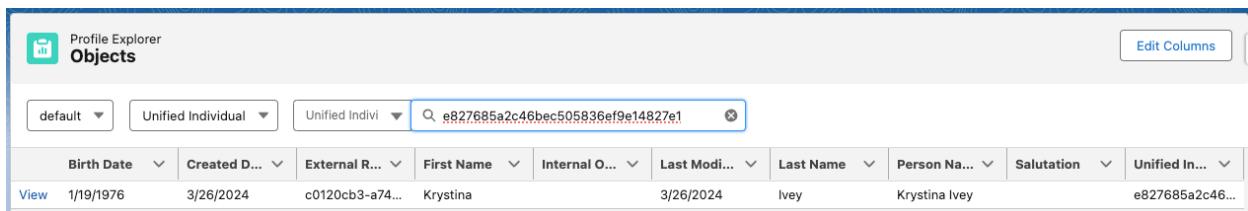
<input type="checkbox"/> Unified Individual Id ↑	Individual Id	External Record Id	Data Source	First Name	Last Name	Birth Date
<input type="checkbox"/> 1da76f415981e4d1b91ef809521e1552 (4)	003al000009E7TAAK	cstravcd6-3dbf-4671-8192-df8e3e6041bf	Salesforce_Home	Alice	Michael	-
	003al000009E7T6AAK	cstevccb-be16-41d5-ab9f-3ad106e919d1	Salesforce_Home	Steve	Chun	-
	003al000009E7SuAAK	c0072c54-5ad2-4e5b-acbb-b7c944e6f61	Salesforce_Home	Caroll Alice	Michael	1/6/1973
	003al000009E7T2AAK	c0080c00-e036-466f-8e34-e8c0468cd17f	Salesforce_Home	Stephen	Chun	13/3/1982
<input type="checkbox"/> 1f019eb5c22c549e325ab5c6c3e87325 (1)	003al000009E7SIAAK	c0054cf8-904e-4475-a20b-02fde98f09d4	Salesforce_Home	Will	Stevenson	15/9/1991
<input type="checkbox"/> 462b7fb8890f85cc371ef55d7b4755de (1)	003al000009E7SyAAK	c0076c39-9e73-4d0c-920b-78728a73e3e4	Salesforce_Home	Virgilio	Chandler	-
<input type="checkbox"/> 49841d01d009f29f50761a9074d39782 (2)	003al000009E7TTAA0	c0121c86-0acf-497d-bf06-308cc2d75f8f	Salesforce_Home	Lelah	Trammell	25/12/1981
	003al000009E7TSAA0	c0120cb3-a749-4844-9ab2-a48e100a032c	Salesforce_Home	Krystina	Ivey	19/1/1976
<input type="checkbox"/> 550ea90377a846c9863fb041c6add20e (2)	003al000009E7T4AAK	c0082c44-1841-499f-b6cb-89420032f66d	Salesforce_Home	Verma	Allred	9/5/1988
	003al000009E7T5AAK	cVALLc3-5d39-44b0-aabf-f2f90cb7dd22	Salesforce_Home	Verna	A.	-
<input type="checkbox"/> 752192262849c4156a4f67d0d0218e58 (1)	003al000009E7SqAAK	c0068c86-ef05-4513-bc55-316e45652006	Salesforce_Home	Carl	Stevenson	3/6/1972
<input type="checkbox"/> a72b63a9bba45c68b9888f3d86c8520c (1)	003al000009E7StAAK	c0071c1c-8110-4389-abd4-2bb3b444bd1	Salesforce_Home	August	Fries	6/9/1951

💡 For practical implementations consider what tool might be the most appropriate for getting necessary details. Keep in mind that data explorer is limiting result set to 200 records max, while report has larger threshold. If you are not filtering your results in the report and have large volume of records to go over then it might have additional cost impact, especially if you'll be running it multiple times during validation stage.

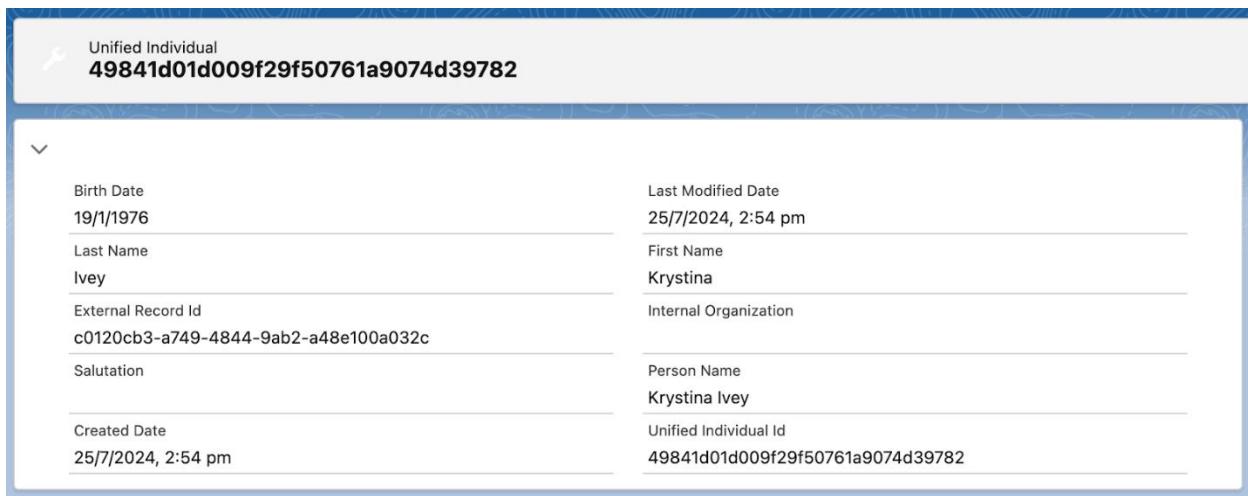
For the next step in our exploration navigate to **Profile Explorer** tab. Choose **default** data space and **Unified**

Individual in the object selector field, choose Unified Individual Id attribute and then paste one the values from the above results (data explorer search or report) into the search field. Preferably one of those having more than 1 Individual Id values linked to it.

 Hint: you'll need to use Enter/Return key when pasting the value in order to initiate the search.



Open the profile record using **View** link and notice that by default the profile page does not include a lot of details that might help our investigation:



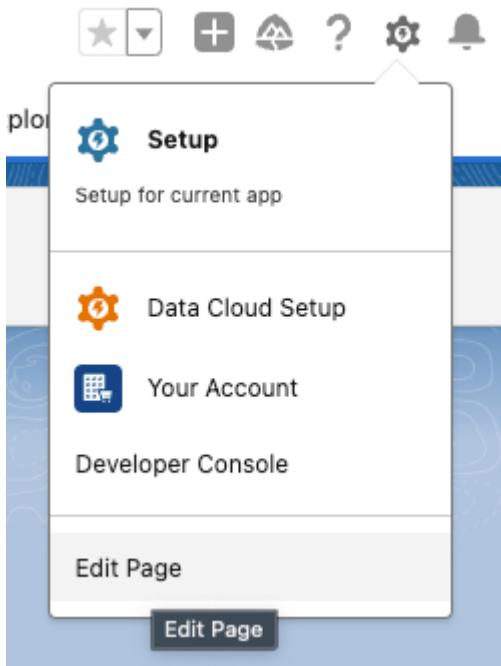
It's possible in some orgs at this point you may see an error message indicating "**This component can't run because a**

required configuration is missing". If so follow the resolution as outlined in this [knowledge article](#)

Let's improve this.

5.3 Configure Unified Individual Record Page Layout

Click on the  icon in the top-right section of the screen and choose **Edit Page** option:



Feel free to update current configuration of Data Cloud Highlights Panel and Data Cloud Details Panel choosing which fields to display there. It's as simple as clicking on

respective panel in the middle section of the screen and then selecting fields in the area on the right.

As an example include **Person Name** field into highlights panel, while removing it from the details panel and instead adding **Birth Date** field to both panels and reordering fields to your liking:

[Page > Data Cloud Highlights Panel](#)

Fields

Unified Individual Id

Person Name

Birth Date

Select...

[Page > Data Cloud Detail Panel](#)

Fields



External Record Id

Salutation

Unified Individual Id

First Name

Birth Date

Last Name

Created Date

Last Modified Date

Internal Organization

Select...

The result should look similar to this:

Unified Individual
e827685a2c46bec505836ef9e14827e1

Person Name	Krystina Ivey	Birth Date	1/19/1976																				
<table border="1"> <tr> <td>External Record Id</td> <td>c0120cb3-a749-4844-9ab2-a48e100a032c</td> <td>Salutation</td> <td></td> </tr> <tr> <td>Unified Individual Id</td> <td>e827685a2c46bec505836ef9e14827e1</td> <td>First Name</td> <td>Krystina</td> </tr> <tr> <td>Birth Date</td> <td>1/19/1976</td> <td>Last Name</td> <td>Ivey</td> </tr> <tr> <td>Created Date</td> <td>3/25/2024, 9:05 PM</td> <td>Last Modified Date</td> <td>3/25/2024, 9:05 PM</td> </tr> <tr> <td>Internal Organization</td> <td colspan="3"></td> </tr> </table>				External Record Id	c0120cb3-a749-4844-9ab2-a48e100a032c	Salutation		Unified Individual Id	e827685a2c46bec505836ef9e14827e1	First Name	Krystina	Birth Date	1/19/1976	Last Name	Ivey	Created Date	3/25/2024, 9:05 PM	Last Modified Date	3/25/2024, 9:05 PM	Internal Organization			
External Record Id	c0120cb3-a749-4844-9ab2-a48e100a032c	Salutation																					
Unified Individual Id	e827685a2c46bec505836ef9e14827e1	First Name	Krystina																				
Birth Date	1/19/1976	Last Name	Ivey																				
Created Date	3/25/2024, 9:05 PM	Last Modified Date	3/25/2024, 9:05 PM																				
Internal Organization																							

Add Component(s) Here

Next add **Tabs** component to the page and move Data Cloud Details Panel component under the **Details** tab:

Unified Individual
e827685a2c46bec505836ef9e14827e1

Person Name	Krystina Ivey	Birth Date	1/19/1976
-------------	---------------	------------	-----------

Related **Details**

External Record Id	c0120cb3-a749-4844-9ab2-a48e100a032c	Salutation	
Unified Individual Id	e827685a2c46bec505836ef9e14827e1	First Name	Krystina
Birth Date	1/19/1976	Last Name	Ivey
Created Date	3/25/2024, 9:05 PM	Last Modified Date	3/25/2024, 9:05 PM
Internal Organization			

Next switch to the **Related** tab and add few components there:

- Add **Data Cloud Profile Related Records** component and configure as follows:
 - Data Space = default

- Unified Individual DMO = Unified Individual
- Unified Individual Link = Unified Link Individual
- Related Data Model Object = Individual
- Select Fields = Individual Id, Person Name, External Record Id, Data Source Object, Birth Date
- Component Icon = Individual
- The screen should now look similar to this:

The screenshot shows a 'Unified Individual' profile page with the ID e827685a2c46bec505836ef9e14827e1. The top navigation bar includes 'Page > Data Cloud Profile Related ...', 'Data Space default', and search fields for 'Unified Individual DMO', 'Unified Individual Link', and 'Related Data Model Object'. The main content area displays a table titled 'Individual' with columns: Individual Id, Person Name, External Record Id, Data Source Object, and Birth Date. Two rows are visible: one for Krystina Ivey (Individual Id 003am000002IT9MAAU) and another for Lelah Trammell (Individual Id 003am000002IT9NAAU). A blue sidebar on the right contains a button 'Add Component(s) Here' and a list of component types: 'Individual', 'Person Name', 'External Record Id', 'Data Source Object', 'Birth Date', and 'Select...'. At the bottom, there's a 'Component Icon' section with 'Individual' selected.

- Save the progress and repeating above steps add related records components for the following objects:
 - Unified Indv Party Identification
 - Unified Indv Contact Point Email
- The tab content should now look close to this:

Related Details

Individual

Individual Id	Person Name	External Record Id	Data Source Object	Birth Date
003am000002IT9MAAU	Krystina Ivey	c0120cb3-a749-4844-9...	Contact	1976-01-19T00:00:00.00...
003am000002IT9NAAU	Lelah Trammell	c0121c86-0acf-497d-bf...	Contact	1981-12-25T00:00:00.00...

Unified Indv Party Identification

Party	Party Identification Type	Identification Name	Identification Number
e827685a2c46bec505836ef9e1...	Person Identification	Driver License	P58-19259175
e827685a2c46bec505836ef9e1...	Loyalty Program	Motor Club Membership	RA-309008573
e827685a2c46bec505836ef9e1...	Loyalty Program	RAVG Retail Membership	720720661
e827685a2c46bec505836ef9e1...	Person Identification	Driver License	R00-86659649
e827685a2c46bec505836ef9e1...	Loyalty Program	RAVG Retail Membership	

Unified Indv Contact Point Email

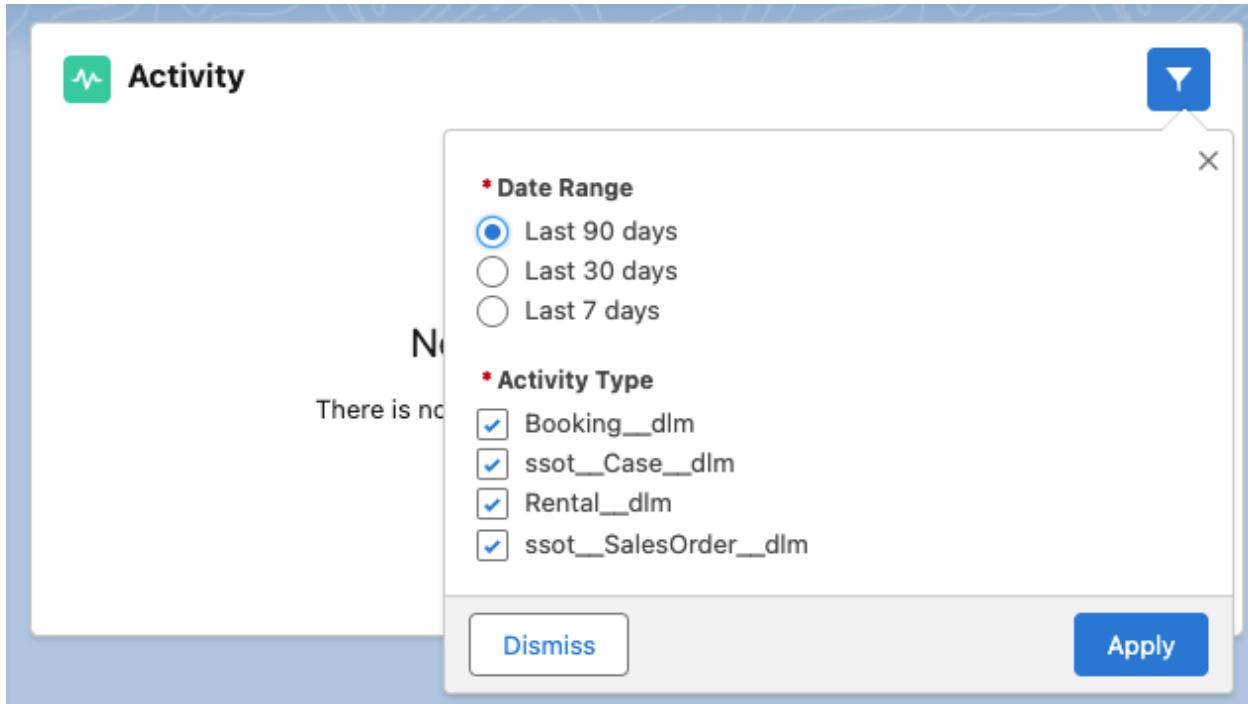
Party	Email Address
e827685a2c46bec505836ef9e14827e1	ltrammell.10000.0121@yahoo.marketingclouddemo.com
e827685a2c46bec505836ef9e14827e1	kivey.10000.0120@yahoo.marketingclouddemo.com

○ Next add **Data Cloud Profile Engagements** component to the column on the right and configure it for default data space with all available **Engagement DMO** objects selected:

The screenshot shows a page titled "Data Cloud Profile Engage..." with a breadcrumb trail "Page > Data Cloud Profile Engage...". On the left, there is a large empty white area. In the center, there is a component titled "Activity" with a green icon. Below the title, it says "No engagement to display" and provides a message: "No records are available in the selected date range. Adjust the date range filter and try again." To the right of the activity component, there are several configuration options: "Data Space" set to "default", "Unified Individual DMO" set to "Unified Individual", "Unified Individual Link" set to "Unified Link Individual", and a "Select Engagement DMOs" section containing buttons for "Booking", "Case", "Rental", and "Sales Order", with a "Select..." button below them.

At this stage there are no configured calculated insights, but in the future you might adjust layout to include **Data Cloud Profile Insights** component to present calculated insight(s).

Save and Activate this page as the Org Default. While **Activity** component might not display any records you can update filter criteria to extend date range and if you have related records these will be displayed in this component.



Here is an example of the final page layout we have configured:

The screenshot displays a detailed customer profile for "Krystina Ivey" (Individual ID: e827685a2c46bec505836ef9e14827e1). The page includes sections for "Related" (Individual, Unified Indv Party Identification, Unified Indv Contact Point Email) and "Details" (Birth Date: 1/19/1976). On the right, there is an "Activity" feed showing three rental events from March 25, 2024.

Party	Identification Name	Identification Number
e827685a2c46bec505836ef9e14827e1	RAVG Retail Membership	720720661
e827685a2c46bec505836ef9e14827e1	Driver License	R00-86659649
e827685a2c46bec505836ef9e14827e1	Motor Club Membership	RA-309008573
e827685a2c46bec505836ef9e14827e1	Driver License	P58-19259175

Exploring the related records you can see details of

individual records that were matched for this particular example. Although if you've selected a different Unified Individual Id value you might see record that was not matched to any other and will have it's own unified individual record.

5.4 Completing Inspection Of Identified Records

Repeat the same steps to explore other records from the identified list. At the end the actual result set should be as follows:

Ref #	Result	If matched, to what?	As expected?	So what happened?
CRM-1	Not matched		No	The difference in first name value was not considered as a match by the "fuzzy" logic, rather these were treated as completely different values.

				Something to keep an eye on when planning to use "fuzzy" matching.
CRM-2	Matched	S3-2	Yes	The name variation did not stipulate enough difference for the records to be considered different.
CRM-3	Matched	S3-3	Yes	Despite the difference in email addresses party identification matching worked to match these records.
CRM-4	Matched	S3-4	Yes	Despite the absence of the

				birth date attribute on CRM contact party identification matching worked in this case.
CRM- 5	Matched	CRM-6, CRM-7, CRM-8, S3-5, S3-6	No	Matching of the CRM-5 and CRM- 6 went as expected via R-1 and R-2 rentals,
CRM- 6	Matched	CRM-5, CRM-7, CRM-8, S3-5, S3-6	No	where they share the same Airline Membership Number.
CRM- 7	Matched		No	Similarly CRM-7 and CRM-8 were matched via R-4 and R-5 rentals,
CRM- 8	Matched		No	this time sharing same Driver License Number.

If we explore these rentals alongside with R-2 then it will become obvious that same Driver License Number is used across all three, therefore matching all 4 individuals.

The unexpected match with S3-5 is a result of a "fuzzy" name matching despite the variance in additional word in first name attribute for CRM-5 record.

				Again, something to keep in mind when inspecting source data and designing matching rules.
CRM-9	Matched	CRM-10, S3-7	Yes	CRM-10 and S3-7 matched via shared loyalty number, that in turn joined to CRM-9 through the rental R-3.
CRM-10	Matched	CRM-9, S3-7	Yes	
CRM-11	Matched	CRM-12	Yes	While two different individual details the rental records for both share the airline membership number resulting in matching of these profiles.
CRM-12	Matched	CRM-11	Yes	

S3-1	Not matched		No	See CRM-1 record above
S3-2	Matched	CRM-2	Yes	See matched record above
S3-3	Matched	CRM-3	Yes	See matched record above
S3-4	Matched	CRM-4	Yes	See matched record above
S3-5	Matched	CRM-5, CRM-6, CRM-7, CRM-8, S3-6	No	See matched records above
S3-6	Matched	CRM-5, CRM-6, CRM-7, CRM-8, S3-5	No	
S3-7	Matched	CRM-9, CRM-10	Yes	See matched records above

As you can see the expected matching outcomes were not observed in full and instead the platform produced slightly different result.

One quite interesting outcome is unification of 6 profiles - CRM-5, CRM-6, CRM-7, CRM-8, S3-5 and S3-6. Specifically due to the same driver license across 4 rental transactions the CRM-7 and CRM-8 were related to CRM-5. This is something that may occur unintentionally while preparing or extracting data, or it could be an inadvertent error made during the process of data entry. But it certainly shows the benefit of identifying and inspecting anomalies (e.g. too large number of unified individual records into a single profile) across data post unification with intent to identify mistakes.

Total actual count of Unified Individual records from the above listed set = **8** matching our expectation. Although the outcome is not quite the same and after correction of data we might end with slightly higher count.

This figure is quite important to understand, as it has direct impact on the consumption of one of the billable account

allocations. Therefore when you are configuring matching rules in the customer implementation it is advisable to prepare a test dataset similar to how we've done it in this exercise and then verify that outcomes of the unification process are matching your customer requirements.

⚠ After completing the Identity Resolution exercise, if you find that the Total Unified Profiles does not match as outlined in the exercises, you may need to review the earlier data ingestion and data modeling exercises to see what was ingested and mapped to troubleshoot. If you want to troubleshoot you can start looking at the above items but if you are not able to figure it out and if the counts vary slightly, (for e.g. 180 Unified Profiles compared to 176 Unified Profiles) feel free to move forward with the rest of the exercises. ⚠

Configure Calculated Insights - Visual Insights Builder:

In this exercise you will configure calculated values using data ingested into your Data Cloud instance. And then in the Segmentation Course you will use this along with other data attributes to configure audience segments.

⚠ Note: Before you can proceed with Calculated Insights (CI) configuration listed in this exercise it is important that your Identity Resolution rules completes publishing. If you try to configure CI ahead of publishing then few objects required for the code to work will not be present in your instance. If you configure CI before publishing finishes you'll find that using calculated insights in segmentation will result in 0 counts as unified profile records were not available during calculations.

Ex: Configure Calculated Insights - Visual Insights Builder

If you don't want to explore the functionality of the Insights Builder feel free to [skip to the SQL code](#) that you can deploy using Create with SQL option when creating new insight.

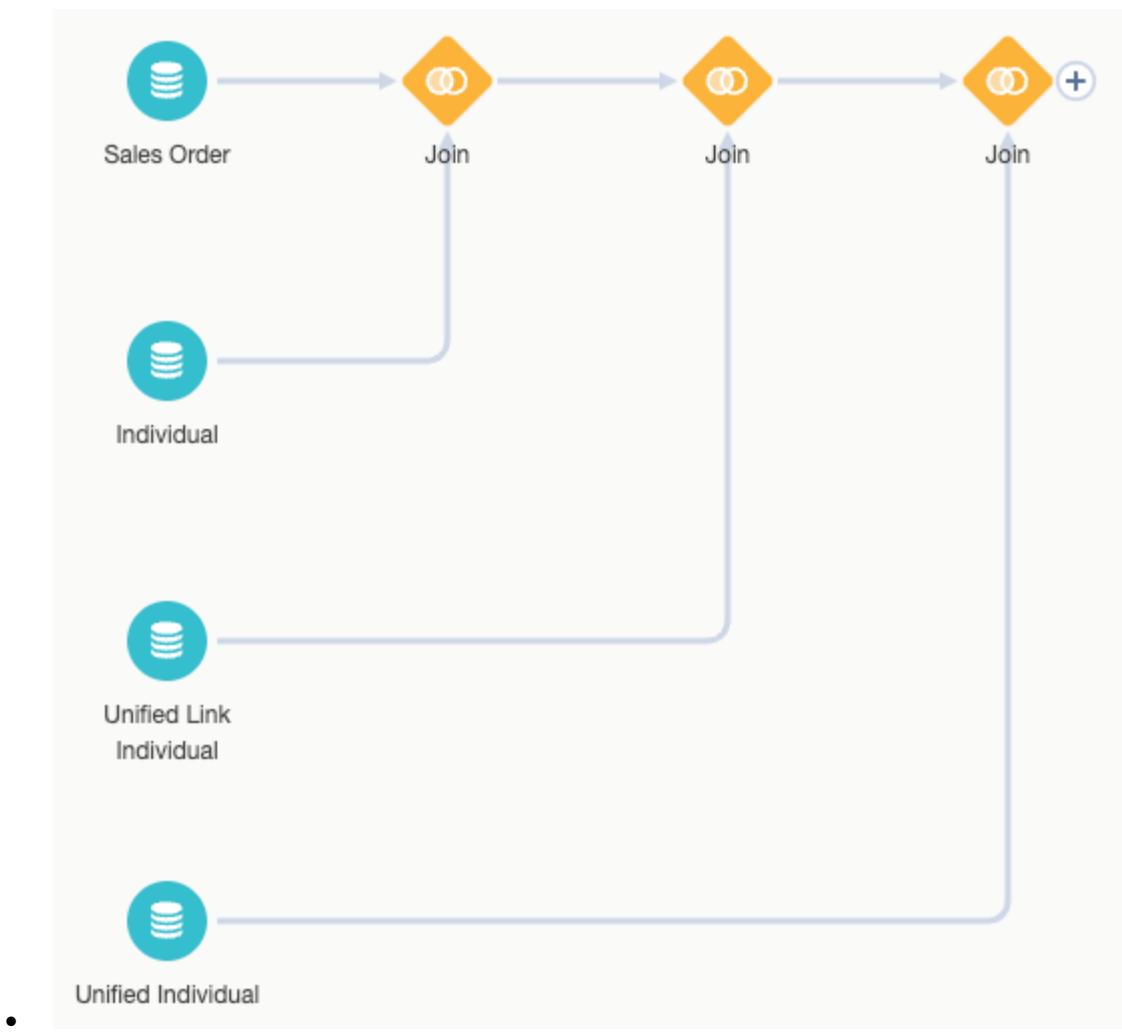
Spend By Customer

For this calculated insight we will aggregate sales data from Sales Order object. The objective is to aggregate following measures at individual and unified individual levels:

- Sum of **Grand Total Amount**
- Average of **Grand Total Amount**
- Get the lowest **Grand Total Amount**
- Get the highest **Grand Total Amount**
- Count all transactions from **Sales Order**

Let's create a calculated insight using the visual builder experience:

- Navigate to **Calculated Insights** and create new with Create with Visual Builder option, confirming Calculated Insight in the dialog
- Begin by clicking **Get Started** button
- Locate and choose Sales Order object as your input data
- Add a **Join** node linking it to the Unified Individual object, the system will configure 3 joins adding **Individual**, **Unified Link Individual** and **Unified Individual** to the canvas:



- The system will open the details of the **Join** to **Unified Individual**
 - Update the name of the **Join** = **Join to Unified Individual**
 - Verify following:
 - Join Type = INNER

- Join keys are on Unified Link Individual.Unified Individual Id = Unified Individual.Unified Individual Id

Fields Name	API Name	Type	Source
Unified Individual Id	UnifiedRecordId_c	Text	Unified Link Individual
Unified Individual Id	UnifiedIndividual.ssot__Id__c	Text	Unified Individual

- Click the **Apply** button to accept the proposed configuration
- Click on the Join to **Individual**:
 - Update the name of the **Join** = Join to Individual
 - Verify following:
 - Join Type = INNER
 - Join keys are on Sales Order.Sold To Customer = Individual.Individual Id and in addition to it Sales Order.Key Qualifier Sold To Customer = Individual.Key Qualifier Individual Id
 - Apply your changes

Fields Name	API Name	Type	Source
Sold To Customer	ssot__SoldToCustomerId_c	Text	Sales Order
Key Qualifier Sold To Customer	KQ_SoldToCustomerId_c	Text	Sales Order
Individual Id	Individual.ssot__Id_c	Text	Individual
Key Qualifier Individual Id	Individual.KQ_Id_c	Text	Individual

- Click on the Join to **Unified Link Individual**:
 - Update the name of the **Join** = Join to Unified Link Individual
 - Verify following:
 - Join Type = INNER
 - Join keys are on Individual.Individual Id = Unified Link Individual.Individual Id
 - Remove join key on key qualifiers as we did not configure it for Individual DMO:

JOIN KEYS

Join to Individual	Unified Link Individual
Individual Id	= Individual Id
Key Qu... KQ	= Key Qu... KQ
+	

Fields Name

Individual Id
Key Qualifier Individual Id
Calculated Insight

Delete Key Qualifier?

Key Qualifiers ensure accurate data is used in your Calculated Insight. [Visit Salesforce Help to Learn More](#)

Cancel **Delete**

- Add another join keys:
 - Individual.Data Source = Unified Link Individual.Data Source

Tip: Pay special attention to ensure the Join key is Individual.Data Source and Unified Link Individual.Data Source
(Not Data Source Object)

JOIN
Join to Unified Link Individual

Select Join Keys

*Join Data Source = *Unified Link Individual Data Source

Cancel **Apply**

⚠ Note: The reason we are also joining to the Data Source to avoid a cross join between the Individual and Unified Link Individual object (in case there are duplicate Individual IDs from different data sources) resulting in overcount of the

results

With the Fully Qualified Keys feature the preferred way to avoid a cross join is to use this capability. Please review the documentation on [Fully Qualified Keys](#) as well as how it's used in [Calculated Insights](#).

Since Key Qualifiers was not configured in earlier steps we are not using it in our exercise.

Fields Name	API Name	Type	Source
Individual Id	ssot__Id__c	Text	Individual
Data Source	ssot__DataSourceId__c	Text	Sales Order
Individual Id	UnifiedLinkIndividual.SourceRecordId__c	Text	Unified Link Individual
Data Source	UnifiedLinkIndividual.ssot__DataSourceId...	Text	Unified Link Individual

- Click **Apply** to save the changes
- Time to build up the aggregations. Add an **Aggregate** node by clicking on + icon next to the last join node.
Define the following:
 - Measures:
 - Sum of **Grand Total Amount** from **Sales Order**, name = **LifetimeValue**

- Average of **Grand Total Amount** from **Sales Order**, name = AvgPurchaseValue
- Minimum of **Grand Total Amount**, name = LowestOrderValue
- Maximum of **Grand Total Amount**, name = HighestOrderValue
- Count of **Sales Order Id**, name = LifetimeTransactionsCount
- Dimensions:
 - Choose **Unified Individual Id** from **Unified Individual**, name = UnifiedID
 - Choose **Individual Id** from **Individual**, name = CustomerID

AGGREGATE Aggregate 

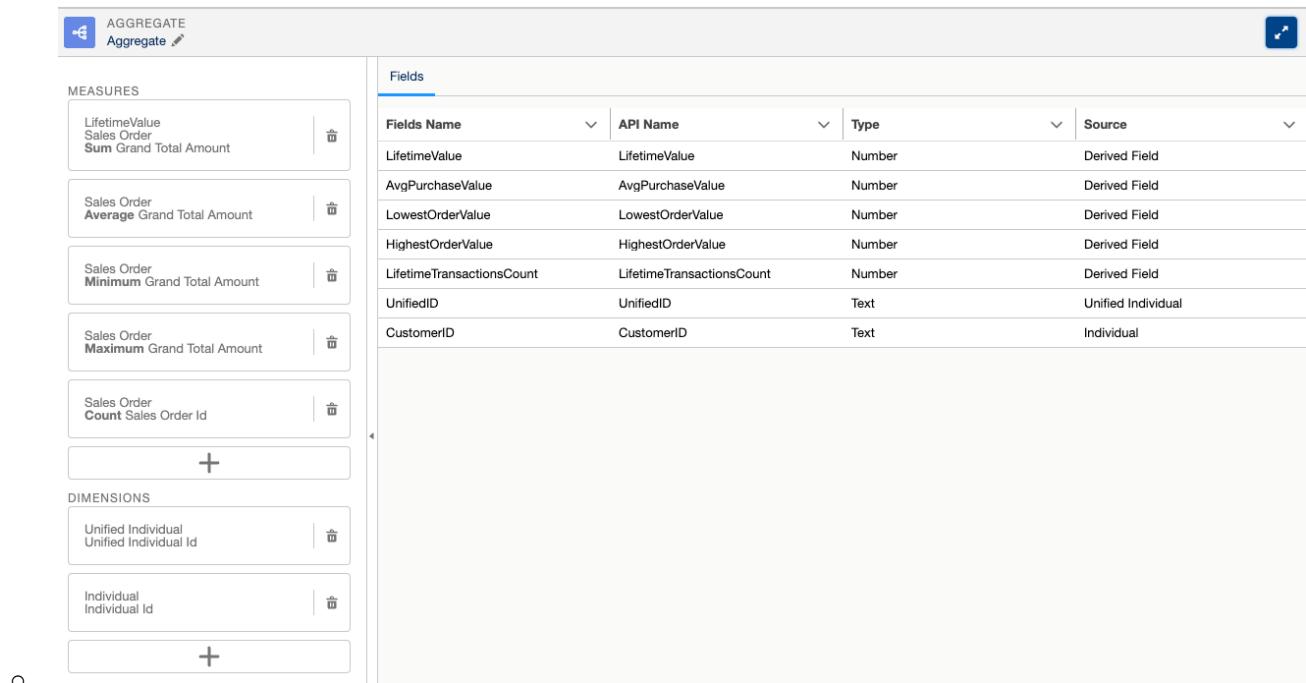
Fields				
Fields Name	API Name	Type	Source	
LifetimeValue	LifetimeValue	Number	Derived Field	
AvgPurchaseValue	AvgPurchaseValue	Number	Derived Field	
LowestOrderValue	LowestOrderValue	Number	Derived Field	
HighestOrderValue	HighestOrderValue	Number	Derived Field	
LifetimeTransactionsCount	LifetimeTransactionsCount	Number	Derived Field	
UnifiedID	UnifiedID	Text	Unified Individual	
CustomerID	CustomerID	Text	Individual	

MEASURES

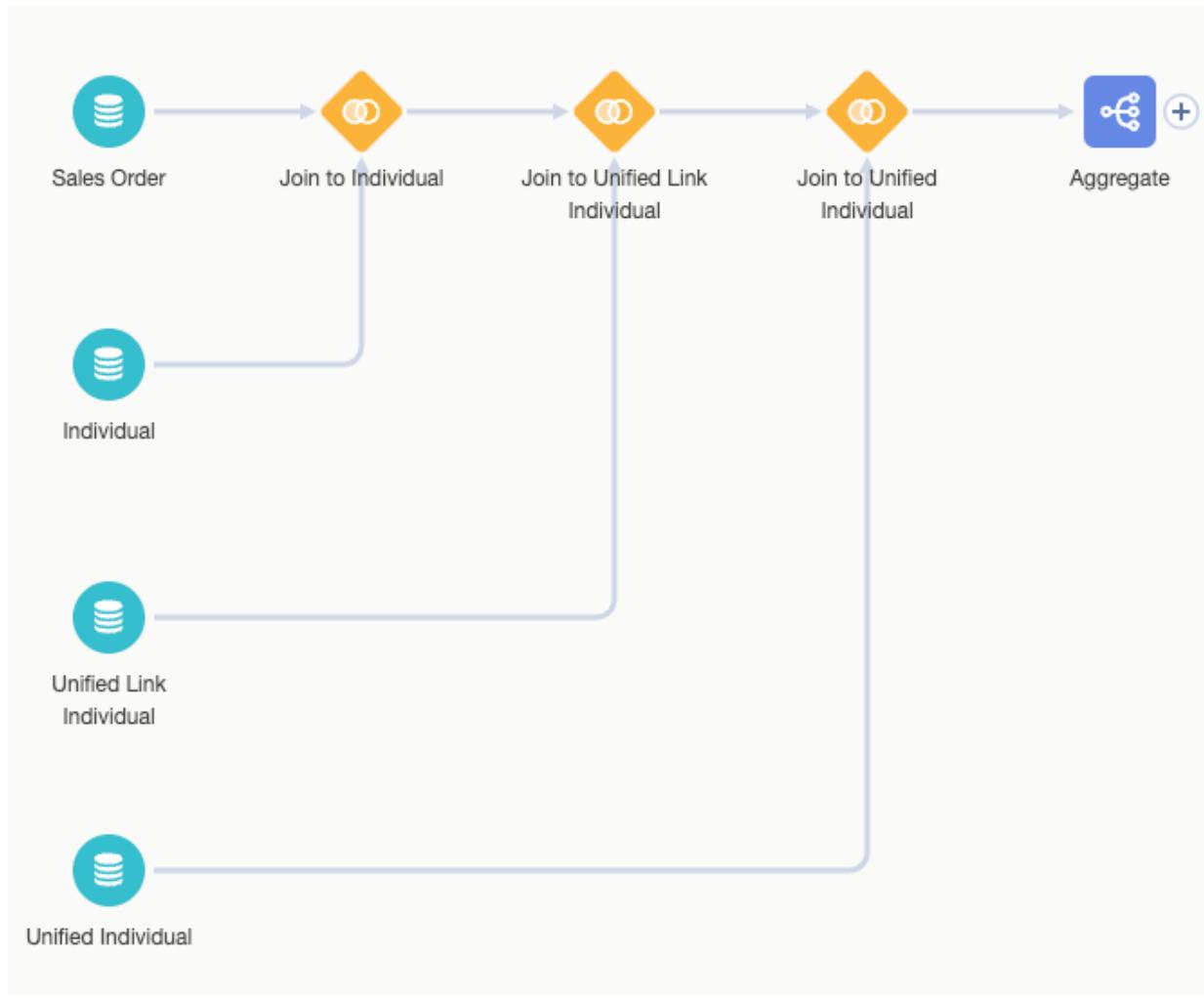
- LifetimeValue
Sales Order Sum Grand Total Amount
- Sales Order Average Grand Total Amount
- Sales Order Minimum Grand Total Amount
- Sales Order Maximum Grand Total Amount
- Sales Order Count Sales Order Id

DIMENSIONS

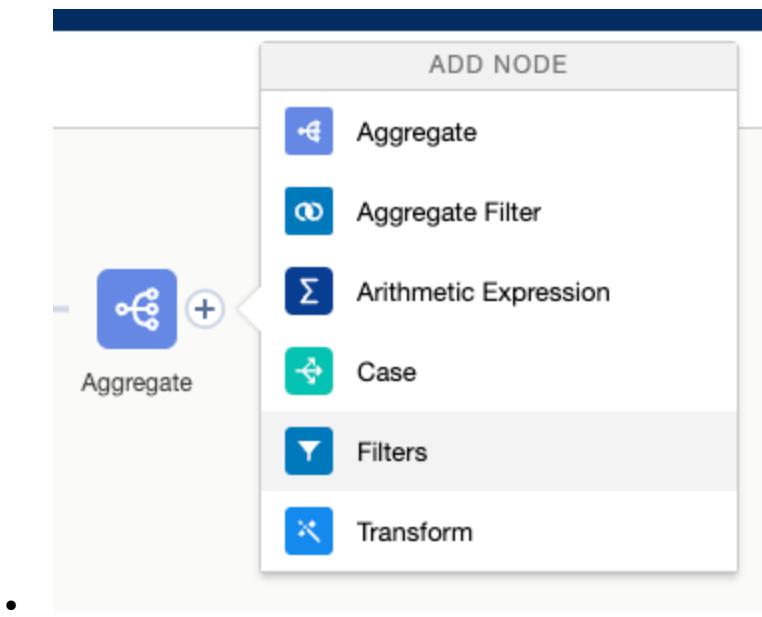
- Unified Individual Unified Individual Id
- Individual Individual Id



- Click **Apply** to save the configurations
- The canvas should look similar to the one below:



- Add **Filters** node to the canvas:



-

- Add a new criteria by clicking on **+** button:
 - Choose **Sales Order Status** field

Define Filter

Search Fields

Fields Name	Source	Type
Order Number	Sales Order	Text
Order Start Date	Sales Order	DateTime
Sales Order Id	Sales Order	Text
Sales Order Status	Sales Order	Text
Sales Store	Sales Order	Text
Sold To Customer	Sales Order	Text
Birth Date	Individual	Date
Created Date	Individual	DateTime

Cancel Next

- Click **Next**
- Set operator = equal
- Set value = Completed

Define Filter

Sales Order > Sales Order Status

* Operator

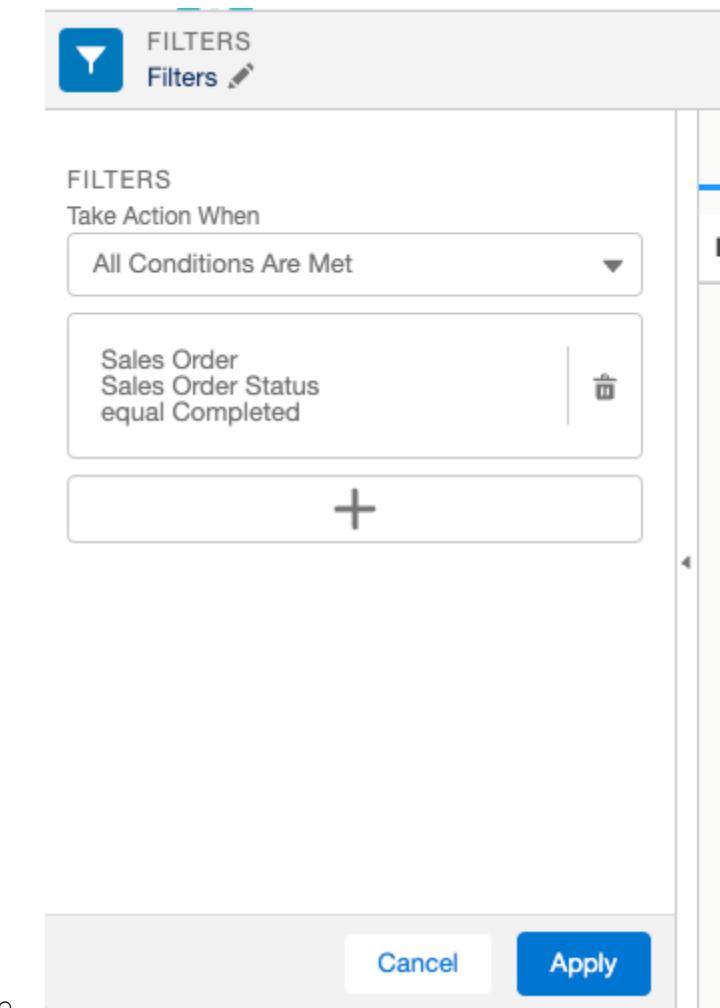
equal

* Value

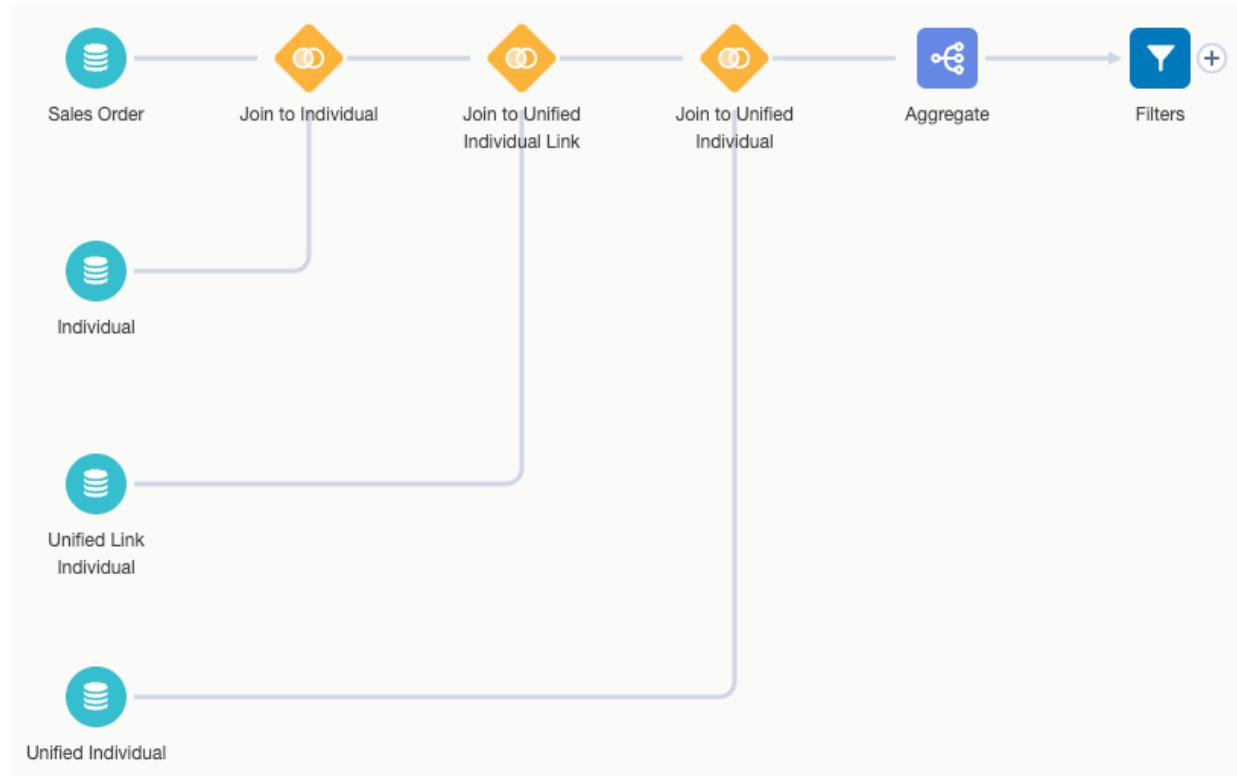
Completed

○

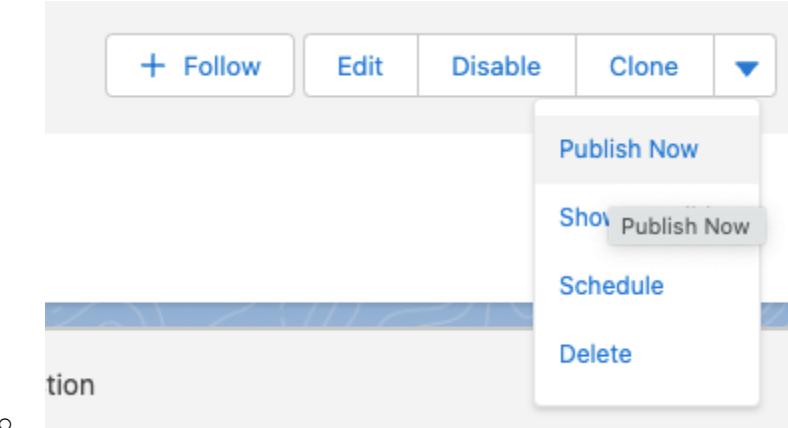
- Click **Apply**



- Click **Apply** one more time to save the configuration. The canvas now should look like this:



- Save and Run the calculated insight:
 - Name = Spend By Customer Visual Builder
 - Description = Aggregated insights on customer spend from Sales Order data
 - Schedule = Not Scheduled
- Once enabled in the drop down on the right choose **Publish Now** option to initiate processing job:



- The **Last Run Status** should change to Pending :

A screenshot of the calculated insight details page. The insight is titled 'Spend By Customer Visual Builder'. It shows the following details:

Data Space	Status	Last Run Time	Last Run Status
default	Active		Pending

The processing of the insight will take some time (30-40 mins) as the platform will pick up new calculated insight and schedule processing job.

Note: The goal of this exercise was to show you the Visual Insight builder capabilities. But if you encounter an error stating 'An Internal Error has occurred' when saving the insight, please use the [SQL](#) shown below to create this insight via SQL.

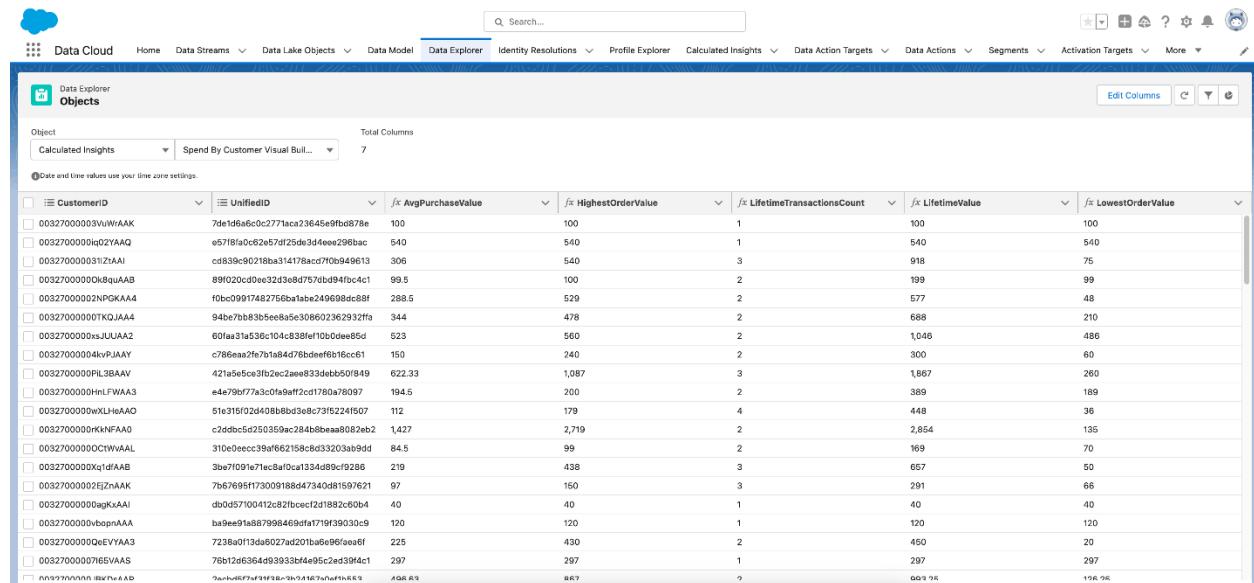
SQL of the generated Visual Insight

Inspect the SQL generated by the Calculated Insight. The SQL should be similar to below format.

View Calculated Insight Results

Once the Calculated Insight has finished processing you can see the result in the Data Explorer tab

Check the output of your CI to verify the results



The screenshot shows the Data Cloud Data Explorer interface. The top navigation bar includes Home, Data Streams, Data Lake Objects, Data Model, Data Explorer (which is selected), Identity Resolutions, Profile Explorer, Calculated Insights (selected), Data Action Targets, Data Actions, Segments, Activation Targets, and More. Below the navigation is a search bar and a toolbar with various icons. The main area is titled 'Objects' and shows a table for 'Calculated Insights'. The table has columns: Object, Total Columns, AvgPurchaseValue, HighestOrderValue, LifetimeTransactionsCount, LifetimeValue, and LowestOrderValue. The table contains numerous rows of data, each with a unique ID and various numerical values.

Object	Total Columns	AvgPurchaseValue	HighestOrderValue	LifetimeTransactionsCount	LifetimeValue	LowestOrderValue
003270000039vWAAK	7ded1d6a6c0c2771aca23645e9fb0d878e	100	100	1	100	100
00327000004h02VAAQ	e57f1810e0c626570f25de3d4ee029b0ac	540	540	1	540	540
003270000031zAAI	cdb39a90218a314178acd7fb949613	306	540	3	918	75
00327000004k8qUAAB	89f020cd0e3232e8d7d73d94fb0c41	99.5	100	2	199	99
00327000002hNPQKAA4	f0bc09917482756ba1be249698dc88f	288.5	529	2	577	48
003270000007KQJAA4	94be76b83b5ee8a5e308602362937fe	344	478	2	688	210
00327000003xJUUAZ2	0f0aa31a536d104e838fe10f000ee85d	523	560	2	1,048	486
003270000044kvPJAAY	c786eaa2f67ba1b4d78bdedf616c61	150	240	2	300	60
0032700000PRL3BAAV	421a5e6c03fb2ec2ew833debb50f849	622.33	1,087	3	1,867	260
00327000004hLFWAA3	e4e790773a30fa1fe2cc1780a78097	194.5	200	2	389	189
00327000009XLHbAAO	51e315f02d408b8b3e8c73f5224f507	112	179	4	448	36
00327000009rKNFAAO	c2adbc5d250359ac2848bb8aa8082eb2	1,427	2,719	2	2,854	135
0032700000OCWIWAAL	310e0ecc399f62158c8d3203a9d9d	84.5	99	2	169	70
0032700000Xkj1dAAB	3bu57091e71e8a0ca1334d890f9286	219	438	3	657	50
003270000021zJAAK	7b67695f173009188a47340d8159762	97	150	3	291	66
00327000009gkXAAI	db0d5f70041282b2f0ec12d1882c60b4	40	40	1	40	40
0032700000vbgnAAA	ba9ee91a887998469dfa1719f39030c9	120	120	1	120	120
0032700000QEVEYAA3	7238ad013da6027ad201ab696faoef	225	430	2	450	20
00327000007h65VAA5	76b1263634d493933d44e95c2e4394d1	297	297	1	297	297
003270000084KvNAA5	7ae4b4077421913b-3h3A15767aef4b15	498.63	867	2	602.95	196.95

This completes the exercise. Please return to the Partner Learning Camp to continue course.

Configure Calculated Insights - Using SQL:

In this exercise you will configure calculated values using data ingested into your Data Cloud instance. And then use

these along with other data attributes to configure audience segments.

 **Note:** Before you can proceed with Calculated Insights (CI) configuration listed in this exercise it is important that your Identity Resolution rules completes publishing. If you try to configure CI ahead of publishing then few objects required for the code to work will not be present in your instance. If you configure CI before publishing finishes you'll find that using calculated insights in segmentation will result in 0 counts as unified profile records were not available during calculations.

1. Unification Results

In the Identity Resolution exercise we were exploring predefined records utilizing Profile Explorer to validate whether or not **Individual** records were in fact linked through the matching process. But how can you identify those records that did in fact matched without being explicitly told? This CI is aimed to aid with validation process as it aggregates number of individual records that are linked to a given **Unified Individual** profile.

- Navigate to **Calculated Insights** and create new with Create with SQL Expression option

- Paste the following SQL into the **SQL Query** field:

```

SELECT IndividualIdentityLink_dlm.UnifiedRecordId_c as
UnifiedRecordId_c
, IndividualIdentityLink_dlm.SourceRecordId_c as
SourceRecordId_c
, SubQ.MatchesCount_c as MatchCount_c
, MIN(SubQ.MatchesCount_c) as Included_c
FROM IndividualIdentityLink_dlm
JOIN (
SELECT IndividualIdentityLink_dlm.UnifiedRecordId_c as
UID_c
, COUNT(1) as MatchesCount_c
FROM IndividualIdentityLink_dlm
GROUP BY IndividualIdentityLink_dlm.UnifiedRecordId_c
) as SubQ ON
(IndividualIdentityLink_dlm.UnifiedRecordId_c =
SubQ.UID_c)
GROUP BY UnifiedRecordId_c, SourceRecordId_c,
MatchCount_c

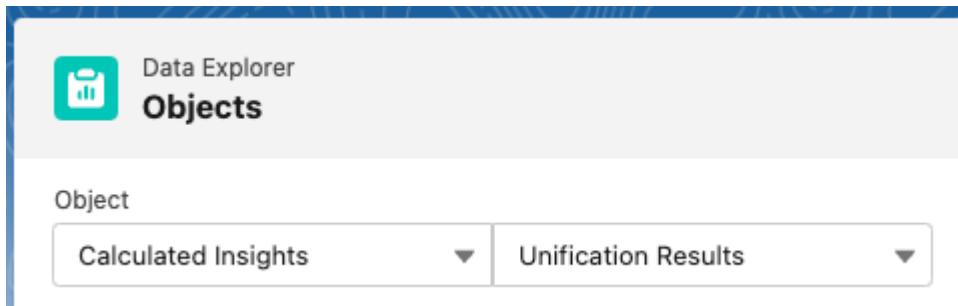
```

- Validate your syntax using Check Syntax button
- Set Name = Unification Results
- Set Description = Identify number of source Individual records matched to a given Unified Individual profile
- Keep Schedule = Not Scheduled

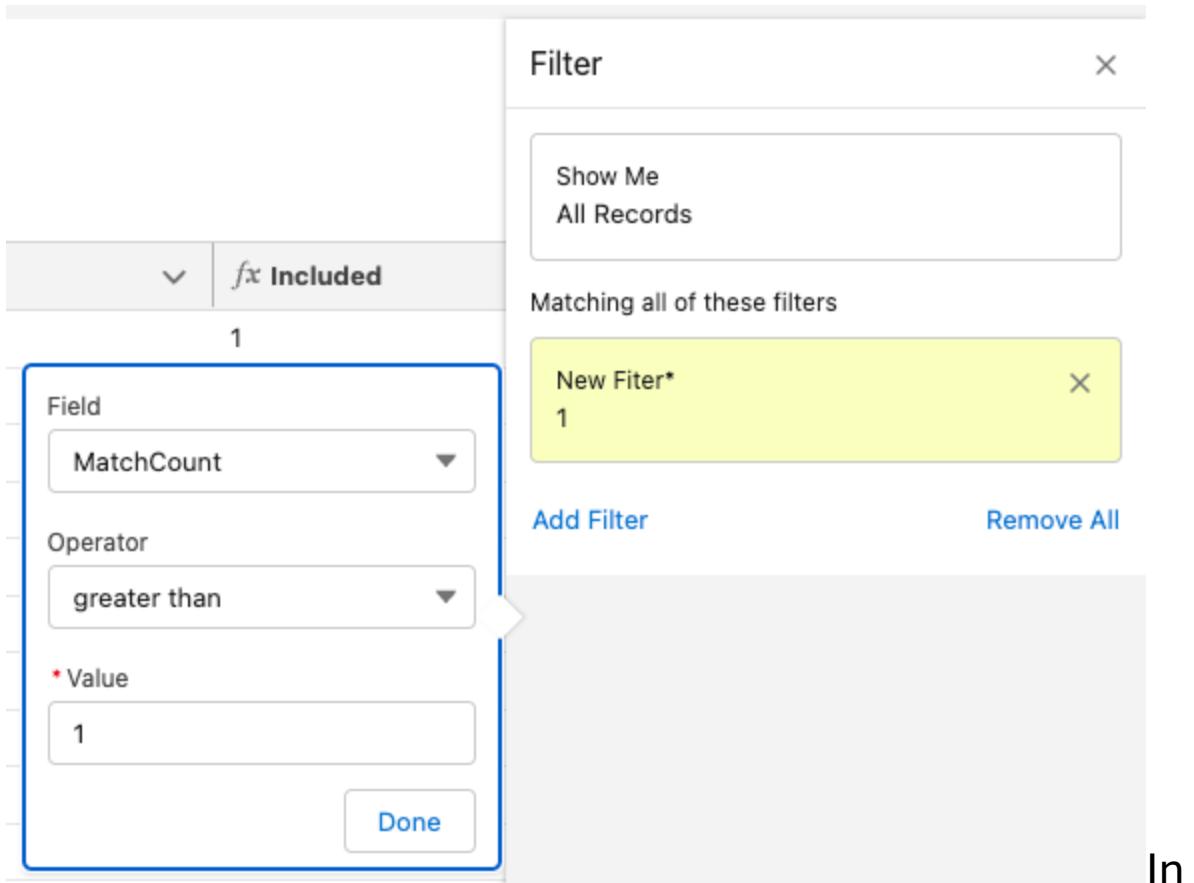
- Enable and Run the insight.

The processing of the insight will take some time (30-40 mins) as the platform will pick up new calculated insight and schedule processing job, but you can proceed with exercises.

Once the processing completes inspect the resulting CI in the Data Explorer.



Use Filter to set the criteria for the records choosing those that have MatchCount > 1 . Feel free to inspect any records beyond that point too to understand if this approach would be helpful in your real customer implementations.



In addition to exploring the results in Data Explorer feel free to configure a report with similar criteria.

 **Tip:** Once you explore the results and no longer require a given CI it is a good idea to disable it, avoiding unnecessary processing utilization, especially in the production deploy. Since we did not configure schedule it does not necessarily needs disabling.

2. Customer Total Purchases By Store

In this CI, we are going to count the number of purchases made by each customer grouping results by store. In

addition we will include the first store they purchased from. The reason we are identifying first store the customer purchased from by using the FIRST function is to showcase how you can possibly activate a dimension (non-numeric value) for these type of use cases.

Normally you cannot include dimensions from a calculated insight in the activation. CI only allows you to activate measures as additional attributes.

An option to solve for this need is to **activate text data is by using sql function "FIRST" or any other text related function in CI**. This step makes the text data column a ***non-aggregatable measure*** which can then be used in an activation. The hands on exercises in the Activation course walk through how to add this attribute in the activation configuration.

⚠ Note when using this method, remember that the CI should be coded in a way so that the value returned as a measure (in this case store) aligns to the logic being used in the Group By or where Clause.

Let's create the actual insight:

- Navigate to **Calculated Insights** and create new with Create with SQL Expression option
- Set name = Customer Total Purchases by Store
- Paste below SQL into the **SQL Query** field:
- SELECT

```

UnifiedIndividual_dlm.ssot_Id_c as UnifiedID_c ,
ssot_Individual_dlm.ssot_Id_c as CustomerID_c ,
ssot_SalesOrder_dlm.ssot_SalesStoreId_c as
StorePurchasedFrom_c ,
COUNT(ssot_SalesOrder_dlm.ssot_Id_c) as
TransactionsCount_c ,
FIRST(ssot_SalesOrder_dlm.ssot_SalesStoreId_c)
as FirstStorePurchasedFrom_c
FROM
ssot_SalesOrder_dlm
JOIN
IndividualIdentityLink_dlm
on ssot_SalesOrder_dlm.ssot_SoldToCustomerId_c
= IndividualIdentityLink_dlm.SourceRecordId_c
JOIN
UnifiedIndividual_dlm
ON IndividualIdentityLink_dlm.UnifiedRecordId_c =
UnifiedIndividual_dlm.ssot_Id_c
JOIN

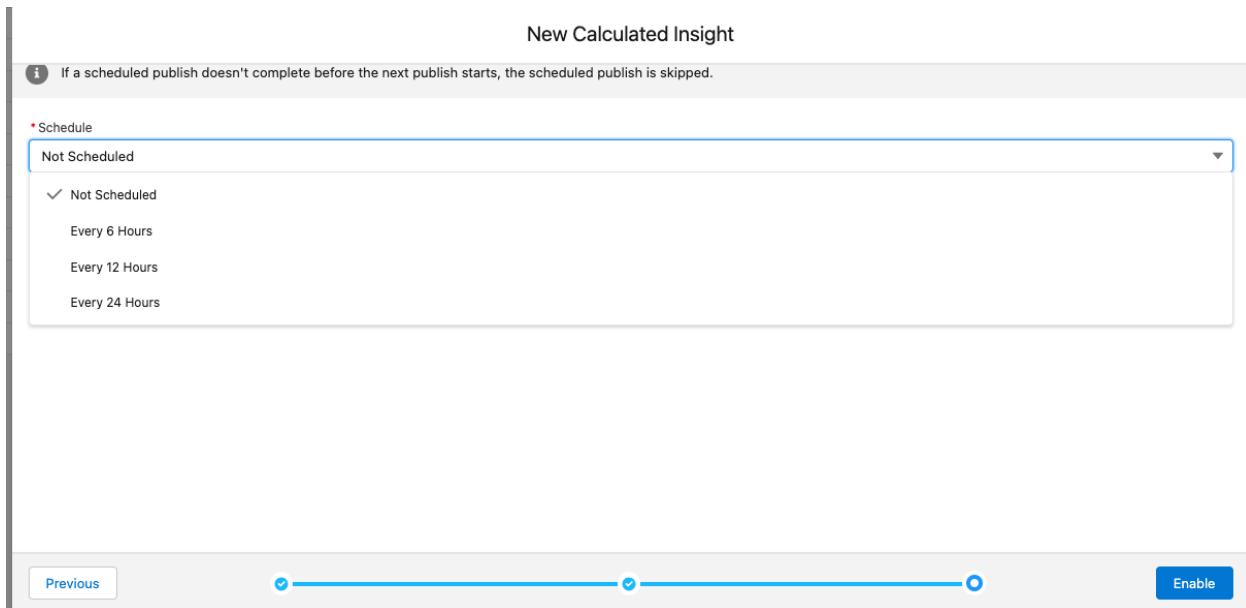
```

```

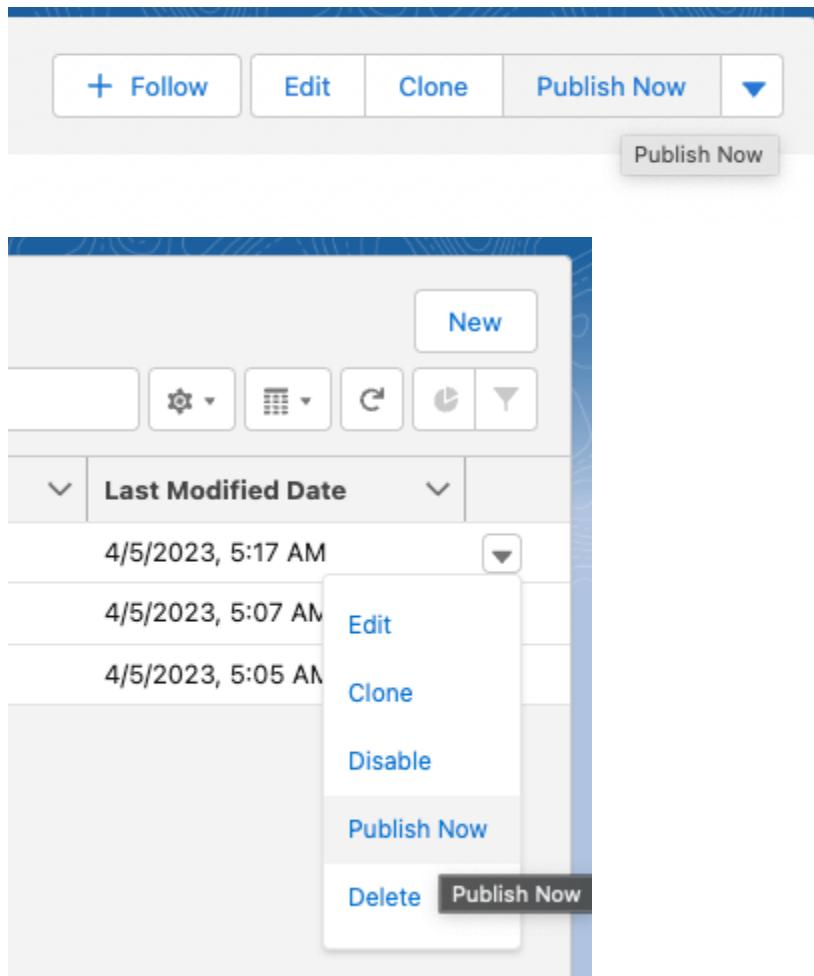
ssot_Individual_dlm
ON IndividualIdentityLink_dlm.SourceRecordId_c =
ssot_Individual_dlm.ssot_Id_c
WHERE
ssot_SalesOrder_dlm.ssot_SalesOrderStatusId_c =
'Completed'
GROUP BY
UnifiedID_c,
CustomerID_c,
ssot_SalesOrder_dlm.ssot_SalesStoreId_c

```

- Validate your syntax using Check Syntax button and activate the insight.
- Keep Schedule as Not Scheduled
 - Take a note that you can specify the schedule for a Calculated Insight to run every 6, 12 or 24 hours.



- Once the Calculated Insight is saved for the processing to be done immediately using **Publish Now** action from either details screen or from the CI list view:



View Calculated Insight Results

Once the Calculated Insight has finished processing you can see the result in the Data Explorer tab

Check the output of your CI to verify the results

The screenshot shows the Data Explorer Objects interface. At the top, there are dropdown menus for 'Data Space' (set to 'default') and 'Object' (set to 'Calculated Insights'), followed by a search bar containing 'Customer Total Purchases by St...'. To the right are buttons for 'Edit Columns', 'C', a magnifying glass, and a refresh icon. Below this, a note says 'Date and time values use your time zone settings.' The main area is a table with the following data:

	CustomerID	StorePurchasedFrom	UnifiedID	FirstStorePurchasedFrom	TransactionsCount
<input type="checkbox"/>	0032700000EeXf5AAF	2685	6670f12d06419e9472a2da946e12c13b	2685	3
<input type="checkbox"/>	00327000000yZ0AAI	4335	0a02dfef0814dedccbc545f623994dd2	4335	2
<input type="checkbox"/>	00327000003pUjdAAE	2465	ef980acb084db815493104aa95643fff	2465	1
<input type="checkbox"/>	00327000000TKQJAA4	4665	94be7bb83b5ee8a5e308602362932ffa	4665	2
<input type="checkbox"/>	0032700000OCTWvAAL	4005	310e0ecc39af662158c8d33203ab9dd	4005	2
<input type="checkbox"/>	0032700000lJ2A5AAI	5435	05d329fa59e7779f89c136af8db37263	5435	2
<input type="checkbox"/>	0032700000kXLZzAAO	2465	490a679e62b320be1b855233ac7deb80	2465	2
<input type="checkbox"/>	00327000005bjfmAAA	1475	da7edbdb2618eba8693f7b0f3e7d4dd	1475	1
<input type="checkbox"/>	0032700000mwvRAAY	1475	7120c5ba3b7162c937e5d3fb4ecd7162	1475	2
<input type="checkbox"/>	0032700000chUS4AM	2575	fe89128ccd589e61e5bed513b3bd708f	2575	2
<input type="checkbox"/>	0032700000crOViaAM	6095	701986425da5333d2167578471c957af	6095	2
<input type="checkbox"/>	0032700000wxLHeAAO	2465	51e315fd2408b8bb3e8c73f5224f507	2465	4
<input type="checkbox"/>	0032700000HMIEsAAP	5435	c6a4f2aaff56e0843eeef5888fc9bd639	5435	1
<input type="checkbox"/>	0032700000bjnSAAU	5545	7fbef4b2689fdbf7fb67c21a144c6cf0	5545	2

Configure Calculated Insights - Deploy from Package:

Ex: Configure Calculated Insights - Deploy from Package

For this calculated insight we will use unmanaged package and deploy it into your learning org. The idea behind packaging the CI provides an opportunity for reuse of best practices and quick deployment in the customer instances. Let's look into configuration of an insight taken from the

[GitHub repository](#). Specifically we'll configure RFM scores.

Although the SQL is supplied, the object and field names will vary when applied to the data model configured in our environments. The steps below will guide you through the configuration of the package and then deploying it as a new calculated insight.

Deploying CI from the Package

Before you can configure new calculated insight from the package the package itself has to be deployed in your CDP org.

1. In the URL remove anything after the top level domain of your instance, and append the following:

/packaging/installPackage.apexp?p0=04t5Y000001OhjL

- a. e.g. something like https://mydc-dev-ed.lightning.force.com/packaging/installPackage.apexp?p0=04t5Y000001OhjL
- b. where mydc-dev-ed.lightning.force.com is the top level domain of my org

2. Alternatively you can also click on this link and login with your org credentials:

<https://login.salesforce.com/packaging/installPackage.apexp?p0=04t5Y000001OhjL&isdtp=p1>

3. Choose **Install for Admins Only** and wait until package is deployed.

Install Data Cloud - RFM Score CI
By Salesforce

What if existing component names conflict with ones in this package?

Do not install.
 Rename conflicting components in package.

Install for Admins Only

Install for All Users

Install for Specific Profiles...

Install **Cancel**

App Name	Publisher	Version Name	Version Number
Data Cloud - RFM Score CI	Salesforce	Spring 2023	1.0

Additional Details [View Components](#)

4.

5. Navigate to the **Calculated Insights** tab and create new insight.

6. Choose **Create from a Package** option and choose installed package from the **Ready To Publish** list.

a. If you are installing the package into the org that does not have Identity Resolution configured the package will appear under **Not Ready to Publish** list as example below:

Installed Packages

Select a Calculated Insight from one of your installed packages.

▼ Ready to Publish (0)

▼ Not Ready to Publish (1)

One or more required objects/fields are missing.

RFM Scores SOURCE

* Required Data Model Objects

b.

7. Use **Check Syntax** on the **Expression** view to validate the query and adjust any table/field names if required to match your data model.
8. Activate the calculated insight keeping Schedule option Not Scheduled.

Note that you are not able to edit neither the name of the insight nor any of the field names.

1. Click **Publish Now** after the Calculated Insight has been created to initiate processing of it.

Field Name	Field API Name	Data Type	Field Type
1 Recency	Recency__c	# Number	Metric(Aggregatable)
2 Monetary	Monetary__c	# Number	Metric(Aggregatable)
3 UnifiedID	UnifiedID__c	% Text	Dimension
4 Frequency	Frequency__c	# Number	Metric(Aggregatable)
5 RFMCombined	RFMCombined__c	# Number	Metric(Aggregatable)

Once the processing completes inspect the newly calculated values in the Data Explorer and Profile Explorer for a randomly selected individual-two.

If you are having issues deploying the CI from a package, review the SQL logic below and create it manually

▼ RFM Scores

```

SELECT sub2.cust_id__c as UnifiedID__c, First(sub2.rfm_recency__c*100 +
sub2.rfm_frequency__c*10 +sub2.rfm_monetary__c) as RFMCombined__c,
First(sub2.rfm_recency__c) as Recency__c, First(sub2.rfm_frequency__c) as Frequency__c,
First(sub2.rfm_monetary__c) as Monetary__c From ( select UnifiedIndividual__dlm.ssot__Id__c
as cust_id__c, ntile(4) over (order by MAX( ssot__SalesOrder__dlm.ssot__OrderStartDate__c ))
as rfm_recency__c, ntile(4) over (order by count( ssot__SalesOrder__dlm.ssot__Id__c )) as
rfm_frequency__c, ntile(4) over (order by avg(
ssot__SalesOrder__dlm.ssot__GrandTotalAmount__c )) as rfm_monetary__c FROM
ssot__SalesOrder__dlm LEFT JOIN IndividualIdentityLink__dlm ON
ssot__SalesOrder__dlm.ssot__SoldToCustomerId__c
=IndividualIdentityLink__dlm.SourceRecordId__c LEFT Join UnifiedIndividual__dlm ON
UnifiedIndividual__dlm.ssot__Id__c = IndividualIdentityLink__dlm.UnifiedRecordId__c GROUP BY
UnifiedIndividual__dlm.ssot__Id__c ) as sub2 GROUP BY sub2.cust_id__c

```

Data Cloud - Configure Streaming Insights:

In this exercise you'll be building the infrastructure for a Streaming Insight and Data Action. This particular example will stream information from a Mobile app, where we will

build a Data Action that looks for customers who have entered a store (by entering a defined geofence). The Target for the Action will be a Salesforce Platform Event, where the customer's entry into the store can trigger, for example, creation of a Salesforce Task, Lead, or Case (this part is out of scope for this exercise, we will only send to the platform bus). We will not stream any actual data in at this time, just set up the pieces of the puzzle.

Note: This exercise is optional

The goal of this exercise is to conceptually showcase how Streaming Insights can be used in Data Cloud. It appears in some orgs the steps outlined below were erroring. So feel free to review the steps outlined and skip the actual hands on steps.

1. Setup Mobile Application

- Navigate to **Customer Data Cloud Setup**
- On the left hand side, click **Website & Mobile Apps**
- Create a new Connector with the following settings
 - Connector Name: Mockup Mobile App
 - Connector Type: Mobile App
- Download the [realtimeschema.json](#) file

- Upload the downloaded file, review the fields and save the schema:

Connector Details

Connector Name	Connector Type	Connector Status
Mockup Mobile App	Mobile App	Needs Data Stream

Schema

Event Name ↑	developerName	Category	Field Count
1 MobileEngagement	MobileEngagement	Engagement	53

⚠ **Note:** Normally we will start uploading data for this exercise we will skip this part

2. Setup Data Stream

- Create a new data stream by going back to the **Data Cloud** application, **Data Streams** tab
- For the data stream source, choose Mobile App and proceed to the next step

New Data Stream

Select a source for the data stream to unify your data.

Connected Sources

Salesforce CRM
Import objects from Salesforce CRM

Mobile App
Import events from iOS and Android applications

Other Sources

Load a sample file in order to teach the system about your file's structure. At the end of this set up flow, you'll be able to specify where data should be retrieved from on an ongoing basis.

Amazon S3
Retrieve a file from Amazon Simple Storage Service

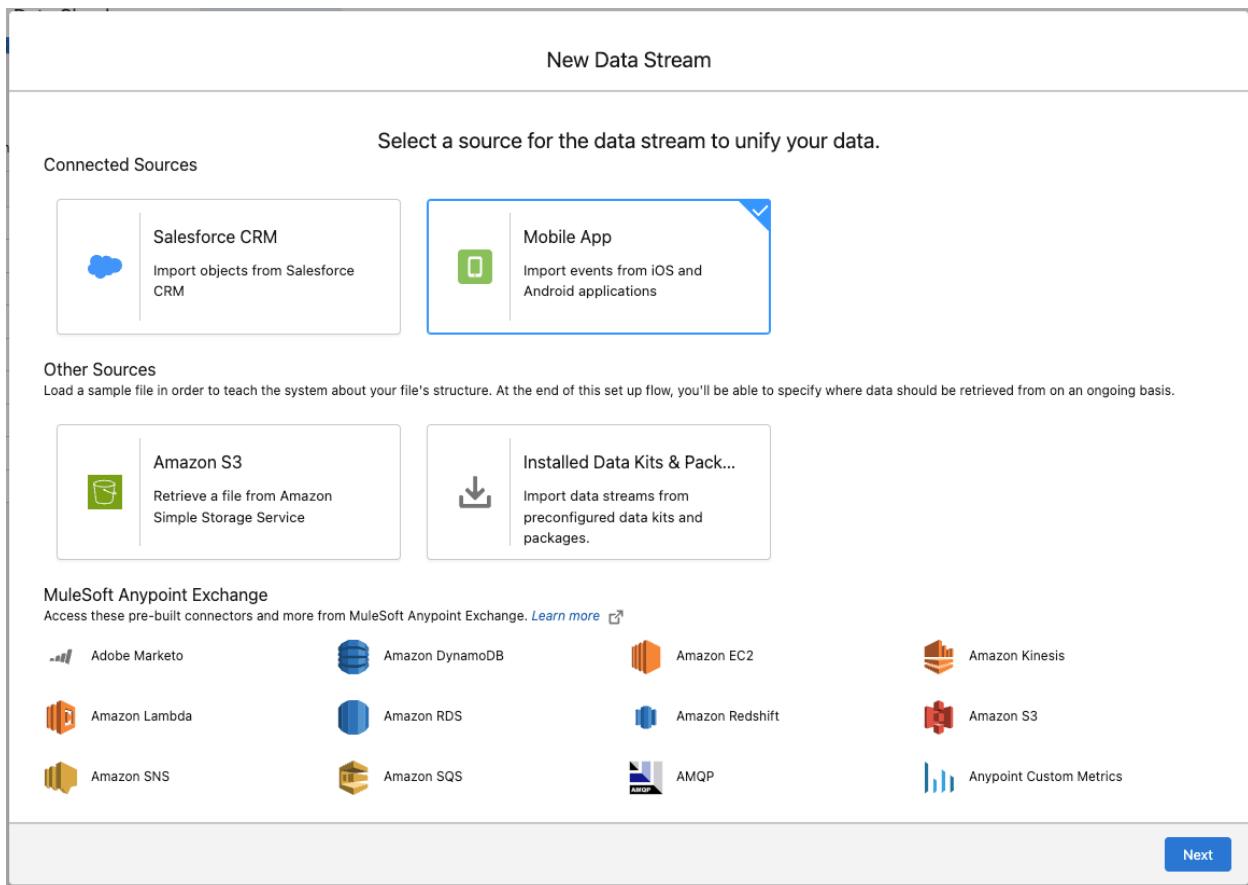
Installed Data Kits & Pack...
Import data streams from preconfigured data kits and packages.

MuleSoft Anypoint Exchange

Access these pre-built connectors and more from MuleSoft Anypoint Exchange. [Learn more](#)

Adobe Marketo	Amazon DynamoDB	Amazon EC2	Amazon Kinesis
Amazon Lambda	Amazon RDS	Amazon Redshift	Amazon S3
Amazon SNS	Amazon SQS	AMQP	Anypoint Custom Metrics

Next



- Under **Mobile App**, choose the mobile app you created earlier
- Use the checkbox to select the **MobileEngagement** event from the **Mockup Mobile App** connector that you configured earlier and proceed to the next step:

New Data Stream

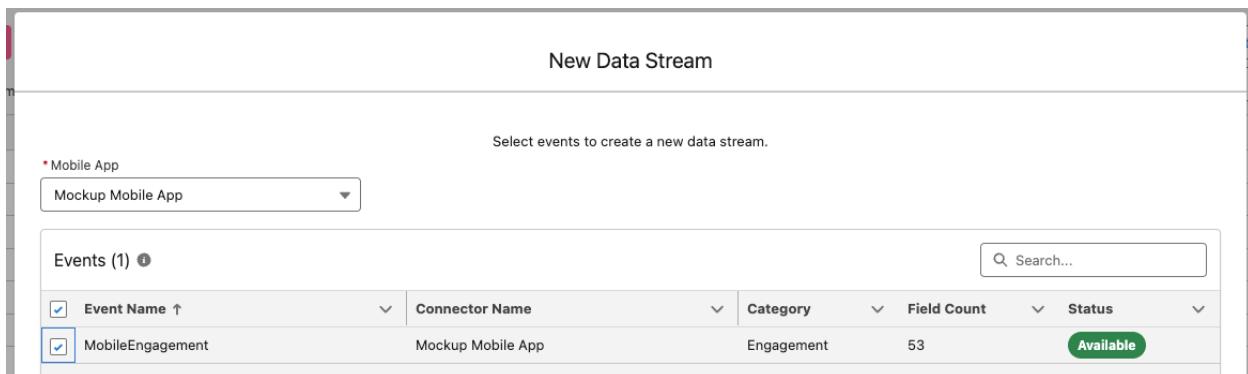
Select events to create a new data stream.

* Mobile App

Mockup Mobile App

Events (1) Search...

Event Name ↑	Connector Name	Category	Field Count	Status
<input checked="" type="checkbox"/> MobileEngagement	Mockup Mobile App	Engagement	53	Available



- Review the fields noting that unlike with other data streams you are not given an option to choose what fields to be configured, update field names or create formula fields:

New Data Stream

Selected Events																																																	
MobileEngagement																																																	
Consumer Details Primary Key Category Record Modified Date eventId Engagement dateTime																																																	
Fields (53) <table border="1"> <thead> <tr> <th>Source Header</th> <th>Field Name</th> <th>Field API Name</th> <th>Data Type</th> </tr> </thead> <tbody> <tr><td>1 FormCompanyName</td><td>FormCompanyName</td><td>MobileEngagement_FormCompanyN...</td><td>Text</td></tr> <tr><td>2 coordinateX</td><td>coordinateX</td><td>MobileEngagement_coordinateX_c</td><td>Number</td></tr> <tr><td>3 MissingDocuments</td><td>MissingDocuments</td><td>MobileEngagement_MissingDocume...</td><td>Text</td></tr> <tr><td>4 OrderQuantity</td><td>OrderQuantity</td><td>MobileEngagement_OrderQuantity_c</td><td>Number</td></tr> <tr><td>5 Direct</td><td>Direct</td><td>MobileEngagement_Direct_c</td><td>Number</td></tr> <tr><td>6 OrganicSearch</td><td>OrganicSearch</td><td>MobileEngagement_OrganicSearch_...</td><td>Number</td></tr> <tr><td>7 geofenceStatus</td><td>geofenceStatus</td><td>MobileEngagement_geofenceStatus...</td><td>Text</td></tr> <tr><td>8 compostWeight</td><td>compostWeight</td><td>MobileEngagement_compostWeight...</td><td>Number</td></tr> <tr><td>9 FormCompanySize</td><td>FormCompanySize</td><td>MobileEngagement_FormCompanyS...</td><td>Text</td></tr> <tr><td>10 eventType</td><td>eventType</td><td>eventType_c</td><td>Text</td></tr> <tr><td>11 VideoWatchStatus</td><td>VideoWatchStatus</td><td>MobileEngagement_VideoWatchStat...</td><td>Text</td></tr> </tbody> </table>		Source Header	Field Name	Field API Name	Data Type	1 FormCompanyName	FormCompanyName	MobileEngagement_FormCompanyN...	Text	2 coordinateX	coordinateX	MobileEngagement_coordinateX_c	Number	3 MissingDocuments	MissingDocuments	MobileEngagement_MissingDocume...	Text	4 OrderQuantity	OrderQuantity	MobileEngagement_OrderQuantity_c	Number	5 Direct	Direct	MobileEngagement_Direct_c	Number	6 OrganicSearch	OrganicSearch	MobileEngagement_OrganicSearch_...	Number	7 geofenceStatus	geofenceStatus	MobileEngagement_geofenceStatus...	Text	8 compostWeight	compostWeight	MobileEngagement_compostWeight...	Number	9 FormCompanySize	FormCompanySize	MobileEngagement_FormCompanyS...	Text	10 eventType	eventType	eventType_c	Text	11 VideoWatchStatus	VideoWatchStatus	MobileEngagement_VideoWatchStat...	Text
Source Header	Field Name	Field API Name	Data Type																																														
1 FormCompanyName	FormCompanyName	MobileEngagement_FormCompanyN...	Text																																														
2 coordinateX	coordinateX	MobileEngagement_coordinateX_c	Number																																														
3 MissingDocuments	MissingDocuments	MobileEngagement_MissingDocume...	Text																																														
4 OrderQuantity	OrderQuantity	MobileEngagement_OrderQuantity_c	Number																																														
5 Direct	Direct	MobileEngagement_Direct_c	Number																																														
6 OrganicSearch	OrganicSearch	MobileEngagement_OrganicSearch_...	Number																																														
7 geofenceStatus	geofenceStatus	MobileEngagement_geofenceStatus...	Text																																														
8 compostWeight	compostWeight	MobileEngagement_compostWeight...	Number																																														
9 FormCompanySize	FormCompanySize	MobileEngagement_FormCompanyS...	Text																																														
10 eventType	eventType	eventType_c	Text																																														
11 VideoWatchStatus	VideoWatchStatus	MobileEngagement_VideoWatchStat...	Text																																														
Previous	Next																																																

- Proceed to the next step and deploy the data stream:

New Data Stream

First, select the data space and define a prefix for this stream, then set the filters if needed

Data Space
default

Engagement Data Streams to be Deployed
All selected engagement events will be added to the following data stream(s) and will require mapping before use:

Data Stream Name: Mockup Mobile App-BE [Set Filters](#)

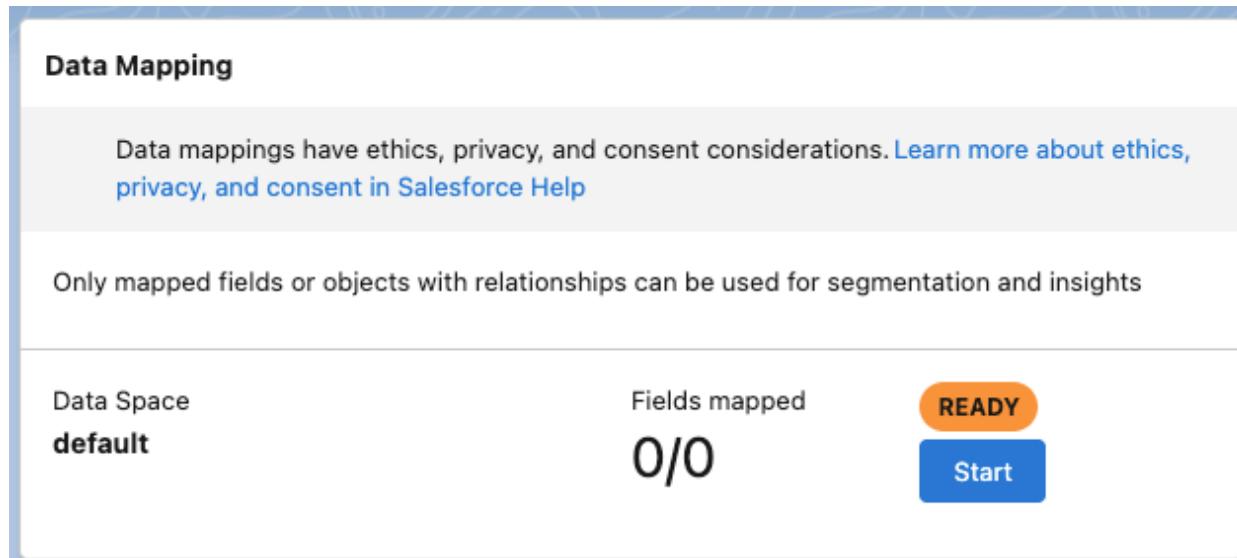
Event to Add	Category	Total Fields	Refresh Mode
MobileEngagement	Engagement	53	Incremental

[Previous](#) [Deploy](#)

Next we need to map this newly created data stream and

create a custom data model object rather than using any of the default ones:

- Click **Start** link on the right in the Data Mapping section



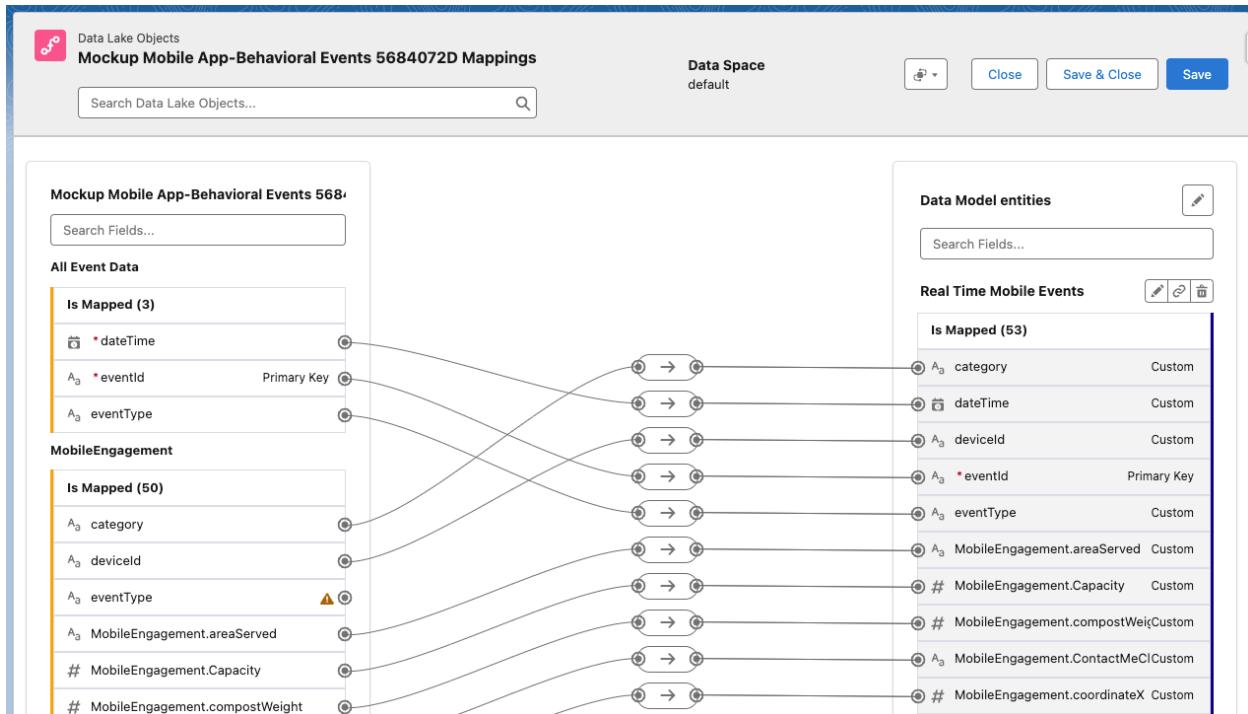
- Click **Select Objects**
- Click **Custom Data Model** tab and then **+ New Custom Object**
 - Object Label = Real Time Mobile Events
 - Verify that the Object API name set to Real_Time_Mobile_Events, as it will be used in the query below
 - For simplicity we'll keep all the fields

New Custom Object

Object Label	Real Time Mobile Events	Object API Name	Real_Time__Mobile_Events	Object Category	Engagement
⚠️ Category selection affects how Data Lake Objects are mapped to Data Model Objects. Learn More in Help					
You can change the labels and data types for the selected fields. Only selected items will be added to the custom object.					
Fields	Field Label	Field API Name	Key	Data Type	Enable Value Suggestion
1	category	category__c		Text	<input type="checkbox"/>
2	DataSource	DataSource__c		Text	<input type="checkbox"/>
3	DataSource Object	DataSourceObject__c		Text	<input type="checkbox"/>
4	dateTime	dateTime__c		Datetime	<input type="checkbox"/>
5	deviceid	deviceid__c		Text	<input type="checkbox"/>
6	eventId	eventId__c	Primary Key	Text	<input type="checkbox"/>
7	eventType	eventType__c		Text	<input type="checkbox"/>
8	MobileEngagement.areaServed	MobileEngagement_areaServed__c		Text	<input type="checkbox"/>
9	MobileEngagement.Capacity	MobileEngagement_Capacity__c		Number	<input type="checkbox"/>
10	MobileEngagement.compostWeight	MobileEngagement_compostWeight__c		Number	<input type="checkbox"/>
11	MobileEngagement.ContactMeClicked	MobileEngagement_ContactMeClicked__c		Text	<input type="checkbox"/>
12	MobileEngagement.coordinateX	MobileEngagement_coordinateX__c		Number	<input type="checkbox"/>
13	MobileEngagement.coordinateY	MobileEngagement_coordinateY__c		Number	<input type="checkbox"/>
14	MobileEngagement.CustomerQuestion	MobileEngagement_CustomerQuestion__c		Text	<input type="checkbox"/>
15	MobileEngagement.DeviceTemperature	MobileEngagement_DeviceTemperature__c		Number	<input type="checkbox"/>
16	MobileEngagement.Direct	MobileEngagement_Direct__c		Number	<input type="checkbox"/>

[Back](#) [Save](#)

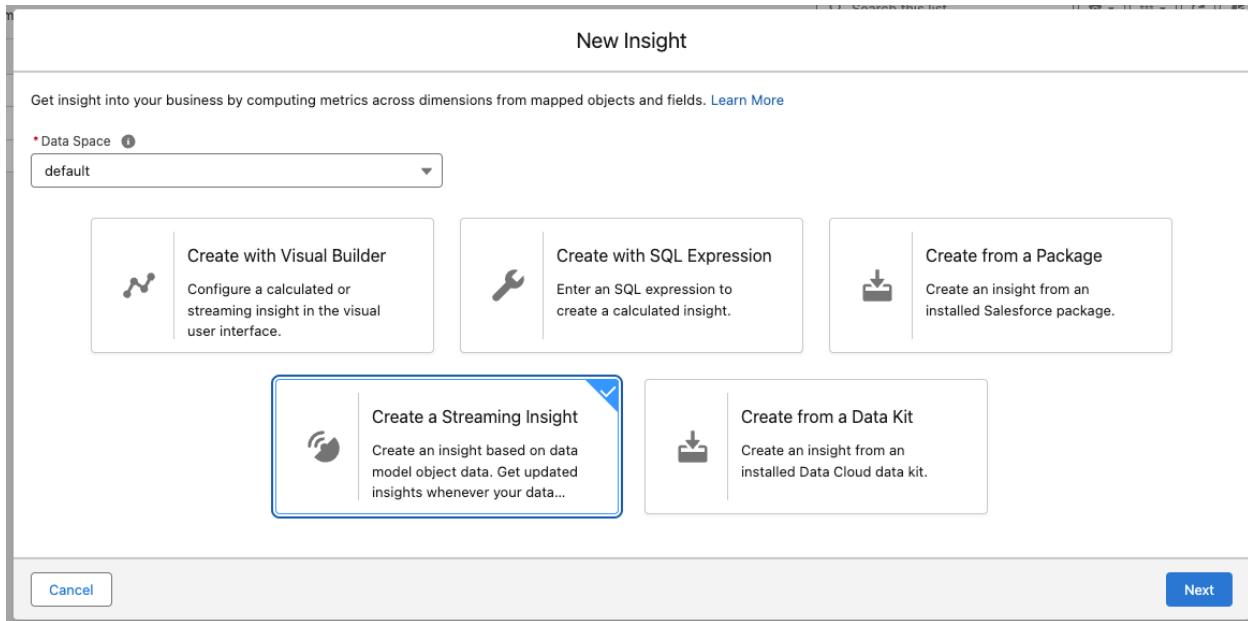
- Save the new object. 
- The mapping will be complete for you and the data stream will be ready.



3. Setup Streaming Insight

- Navigate to the **Calculated Insights** tab

- Click **New** and then **Create Streaming Insights**



⚠ Note: See the model and notice the differences between a Calculated Insight. You will see that the number of objects available under “Fields” is fewer, and “Functions” is severely limited, with just SUM and COUNT available. Also note that the skeleton query under “Expression” has changed to reflect the structure needed for a Streaming Insight

- Create Streaming Insight
 - Name = Geo Fence Stream
 - Description = Streaming Insight to count the number of times an individual entered a geofence in the past 5 minutes
- In the Expression section, add the following query

```

SELECT COUNT (
Real_Time_Mobile_Events__dlm.MobileEngagement_geofenceStatus__c ) as entergeofence__c,
UnifiedIndividual__dlm.ssot__Id__c as customerid__c,
Real_Time_Mobile_Events__dlm.MobileEngagement_geofenceStatus__c as geofenceevent__c,
WINDOW.START as start__c,
WINDOW.END as end__c
FROM Real_Time_Mobile_Events__dlm
JOIN IndividualIdentityLink__dlm ON
Real_Time_Mobile_Events__dlm.deviceId__c =
IndividualIdentityLink__dlm.SourceRecordId__c
JOIN UnifiedIndividual__dlm ON
UnifiedIndividual__dlm.ssot__Id__c =
IndividualIdentityLink__dlm.UnifiedRecordId__c
GROUP BY window(
Real_Time_Mobile_Events__dlm.dateTime__c , '5 MINUTE'),
customerid__c,
geofenceevent__c

```

- Click **Check Syntax** to ensure everything is good and the syntax is correct.
- Click **Save and Run** to activate the streaming insight.

4. Create Data Action Target

- Before creating a Data Action, we first need to define a Data Action Target
- Navigate to the **Data Action Targets** tab and click **New**
 - Action Target Name: Salesforce Platform Event (or any other name you want)
 - Action Target Type: Salesforce Platform Event
 - The Salesforce Org will be pre-selected for you pointing to your home org:

New Data Action Target

* Action Target Name Salesforce Platform Event	* Action Target Api Name Salesforce_Platform_Event
* Action Target Type Salesforce Platform Event	* Select Salesforce Org Salesforce - Home Org
.	
<input type="button" value="Cancel"/> <input type="button" value="Save"/>	

- Save the data action target

5. Create the Data Action

- Navigate to the **Data Actions** tab.
- Create a new Data Action and select the Action Target you just created:

The screenshot shows a table titled "Data Action Target" with one row. The columns are "Data Action Target Name", "Target Type", "Created By", and "Modified Date". The row contains the following data:

Data Action Target Name	Target Type	Created By	Modified Date
Salesforce Platform Event	Salesforce Platform Event	Vladimir Silak	May 07, 2024

- Select the Data Action Configuration as following
 - Data Space: default
 - Object Type: Calculated Insight
 - Primary Object: Geo Fence Stream
 - Related Object: Unified Individual

The screenshot shows the "Data Space and Objects" configuration section with the following settings:

- * Data Space: default
- * Object Type: Calculated Insight
- * Primary Object: Geo Fence Stream
- Related Object (Optional): Unified Individual

- Proceed to the next step
- Add Related Attributes to include with the Data Action:
 - Add the following attributes
 - **First Name**
 - **Last Name**
 - **Unified Individual Id**

New Action

Related Attributes		Summary																							
Select up to 10 attributes to add to the data action. If you don't need to add attributes, return to the previous page and delete the related object.		Data Action Target  Salesforce Platform Event  Salesforce Platform Event																							
Related Object Unified Individual		Data Space  default																							
<table border="1"> <thead> <tr> <th>Attribute Name</th> <th>Attribute Type</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/> Birth Date</td> <td>Date</td> </tr> <tr> <td><input type="checkbox"/> Created Date</td> <td>DateTime</td> </tr> <tr> <td><input type="checkbox"/> External Record Id</td> <td>Text</td> </tr> <tr> <td><input checked="" type="checkbox"/> First Name</td> <td>Text</td> </tr> <tr> <td><input type="checkbox"/> Internal Organization</td> <td>Text</td> </tr> <tr> <td><input type="checkbox"/> Last Modified Date</td> <td>DateTime</td> </tr> <tr> <td><input checked="" type="checkbox"/> Last Name</td> <td>Text</td> </tr> <tr> <td><input type="checkbox"/> Person Name</td> <td>Text</td> </tr> <tr> <td><input type="checkbox"/> Salutation</td> <td>Text</td> </tr> <tr> <td><input checked="" type="checkbox"/> Unified Individual Id</td> <td>Text</td> </tr> </tbody> </table>		Attribute Name	Attribute Type	<input type="checkbox"/> Birth Date	Date	<input type="checkbox"/> Created Date	DateTime	<input type="checkbox"/> External Record Id	Text	<input checked="" type="checkbox"/> First Name	Text	<input type="checkbox"/> Internal Organization	Text	<input type="checkbox"/> Last Modified Date	DateTime	<input checked="" type="checkbox"/> Last Name	Text	<input type="checkbox"/> Person Name	Text	<input type="checkbox"/> Salutation	Text	<input checked="" type="checkbox"/> Unified Individual Id	Text	Primary Object  Geo Fence Stream  Calculated Insight	
Attribute Name	Attribute Type																								
<input type="checkbox"/> Birth Date	Date																								
<input type="checkbox"/> Created Date	DateTime																								
<input type="checkbox"/> External Record Id	Text																								
<input checked="" type="checkbox"/> First Name	Text																								
<input type="checkbox"/> Internal Organization	Text																								
<input type="checkbox"/> Last Modified Date	DateTime																								
<input checked="" type="checkbox"/> Last Name	Text																								
<input type="checkbox"/> Person Name	Text																								
<input type="checkbox"/> Salutation	Text																								
<input checked="" type="checkbox"/> Unified Individual Id	Text																								
		Related Object  Unified Individual Data Model Object																							
		Related Attributes <ol style="list-style-type: none"> 1. First Name 2. Last Name 3. Unified Individual Id 																							

- Proceed to the next step
- Configure the Event and Action Rules as following
 - Event Rules
 - Record Created
 - Record Update
 - Action Rules

- All Conditions are Met (AND)
 - Object: Geo Fence Stream
 - Attribute: geofenceevent
 - Operator: Is Equal To
 - Value: entered

New Action

Event and Action Rules

Event Rules ⓘ
Select record actions that trigger the action.

Record Created
 Record Updated
 Record Deleted

Action Rules

* Publish data when:

All Conditions are Met (AND)

* Object: Geo Fence Stream
* Attribute: geofenceevent
* Operator: Is Equal To
* Value: entered

+ Add Condition

Summary

Data Action Target
Salesforce Platform Event
Salesforce Platform Event

Data Space
default

Primary Object
Geo Fence Stream
Calculated Insight

Event Rules
Record Created OR Record Updated

Action Rules
geofenceevent = entered

Feel free to review the other options available

- Proceed to the next step
- Give your Action a name:
 - Action Name: Customer Entering Store
 - The API name should self populate

New Action

Properties	Summary
Action Name <input type="text" value="Customer Entering Store"/>	Data Action Target <input checked="" type="checkbox"/> Salesforce Platform Event <input type="checkbox"/> Salesforce Platform Event
Action API Name <input type="text" value="Customer_Entering_Store"/>	
Description <input type="text"/>	
	Data Space <input checked="" type="checkbox"/> default
	Primary Object <input checked="" type="checkbox"/> Geo Fence Stream <input type="checkbox"/> Calculated Insight
	Related Object <input checked="" type="checkbox"/> Unified Individual <input type="checkbox"/> Data Model Object
	Event Rules Record Created OR Record Updated
	Action Rules

Back  Save and Publish

- Click **Save and Publish**
- The data action has now been created:

The screenshot shows the configuration details for a Data Action named 'Customer Entering Store'. Key information includes:

- Data Space:** default
- Status:** Processing
- Created By:** CDP Admin
- Created Date:** 8/4/2023, 8:22 AM
- Last Modified Date:** 8/4/2023, 8:29 AM

The configuration tab is selected, showing the following sections:

- Action Targets (1):** A table with one entry: Salesforce Platform Event (Status: ACTIVE, Target Type: Salesforce Platform Event).
- Event Rules:** A section showing the rule: Record Created OR Record Updated.
- Conditions:** A section showing the condition: Geo Fence Stream.geofenceevent = entered.
- Attributes (8):** A table listing attributes with their data types and objects:

Attribute Name	Data Type	Object
end	DateTime	Geo Fence Stream
start	DateTime	Geo Fence Stream
customerid	Text	Geo Fence Stream
entergeofence	Number	Geo Fence Stream
geofenceevent	Text	Geo Fence Stream
First Name	Text	Unified Individual
Unified Individual Id	Text	Unified Individual
Last Name	Text	Unified Individual

i Note - Apart from attributes selected in the previous step, five other attributes will be included in the Streaming Insight. These include the measures and dimension from our Streaming Insight. The attributes include:

- end
- start
- customerid
- entergeofence
- geofenceevent

