

## DAY 1

### **SETUP YOUR INSTANCE:**

# Setup Your Instance

- 1. Assign User Permissions**
- 2. Configure Additional Permissions**
- 3. Provision Your Instance**

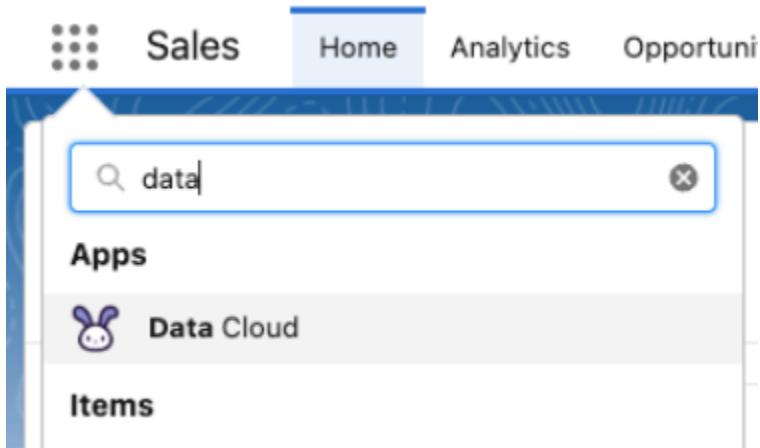
### **Setup Steps**

#### **Step 1 - Assign User Permissions**

To setup and configure your Data Cloud instance it is required to enable **Admin** permission set to the relevant Salesforce CRM user. Login to your Salesforce CRM instance where Data Cloud licenses were provisioned and validate whether or not the necessary permission has been pre-assigned to your user.

Click on the application switcher (top left) and type data in the search field (see example on the right).

If the Data Cloud application is found and you can navigate to it then proceed to the [\*\*Step 2\*\*](#) below.



If you don't see the Data Cloud application in the application switcher then complete the following task:

- Following the instructions in [this help article](#):
  - Assign Data Cloud Admin permission set to your Salesforce CRM admin user

## Step 2 - Configure Additional Permissions

In order to support proper data generation in your learning org enable below specified feature and assign additional permission to your user.

Navigate to **Setup** and in the Quick Find search for the User Interface and click on **User Interface** link at the bottom of the menu items:

The screenshot shows the Salesforce Setup search interface. A search bar at the top contains the text "user interface". Below the search bar, a list of items is displayed under the heading "User Interface". The items are:

- Action Launcher
- Action Link Templates
- Actions & Recommendations
- App Menu
- Custom Labels
- Density Settings
- > Global Actions
  - Lightning App Builder
  - Lightning Extension
  - Loaded Console Tab Limit
  - Path Settings
  - Quick Text Settings
  - Record Page Settings
  - Rename Tabs and Labels
- > Sites and Domains
  - Tabs
  - Themes and Branding
- > Translation Workbench
  - User Interface

Enable Set Audit Fields upon Record Creation... option under **Setup** and save settings.

**Setup**

- Enable Enhanced Page Layout Editor
- Enable Streaming API
- Enable Dynamic Streaming Channel Creation
- Enable "Set Audit Fields upon Record Creation" and "Update Records with Inactive Owners" User Permissions i
- Enable "Delete from Field History" and "Delete from Field History Archive" User Permissions i
- Enable Custom Object Truncate

Navigate to **Users > Permissions Sets** and configure new permission set:

Click “New” -> Allow modification for date field -> save-> manage assignment->select your user’Swati’-> Save & Assign

- Label = Allow modification for case dates
- Click on **System Permissions** link in the **System** section:

**System**

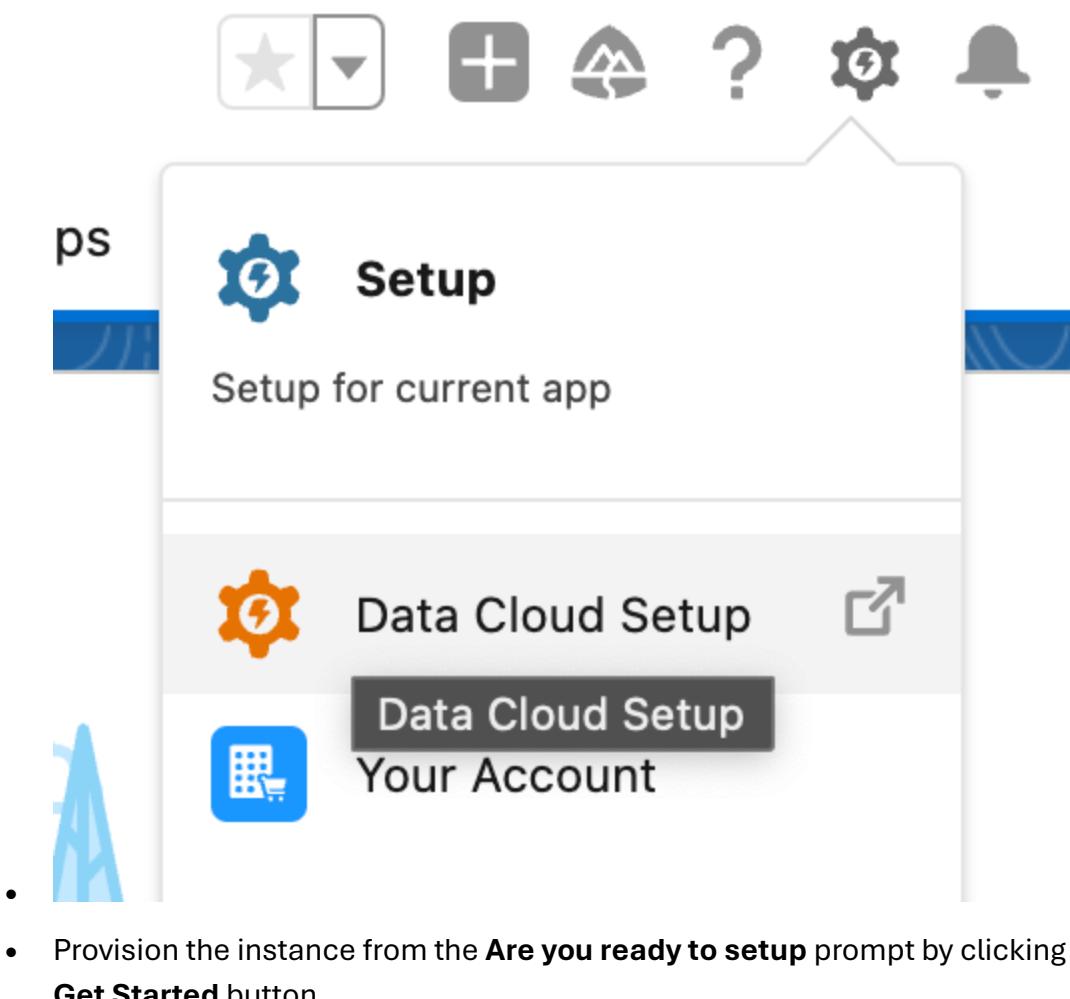
Settings that apply across all apps, such as record and user management <a href="#">Learn More</a>	<b>System Permissions</b> Permissions to perform actions that apply across apps, such as "Modify All Data"
	<b>Service Providers</b> Permissions that let users switch to other websites using single sign-on.

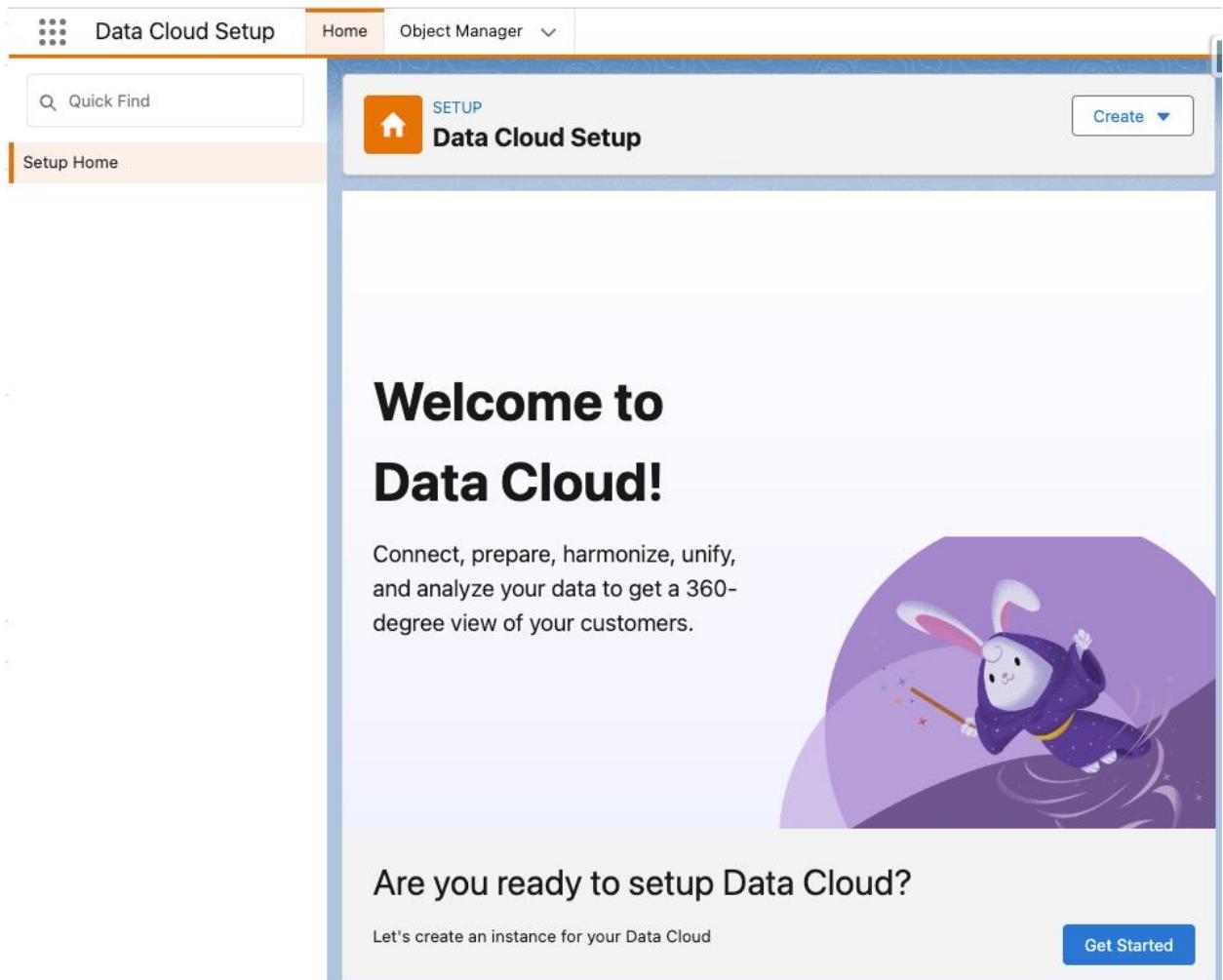
- Click **Edit**, search and enable Set Audit Fields upon Record Creation option:
- |   |  |
|---|--|
| Service Process User<br><input type="checkbox"/> <b>Set Audit Fields upon Record Creation</b><br><input type="checkbox"/> Share Einstein Discovery Models | <input type="checkbox"/> Grants users the access to Service Process Data<br><input checked="" type="checkbox"/> Set audit fields, such as Created By and Last Modified By, when you create a record (API only).<br><input type="checkbox"/> <span style="color: #0070C0;">i</span> Allow user to export Einstein Discovery models to Quin. |
|---|--|
- Click **Save** and confirm in the confirmation dialog.
  - Click **Manage Assignments** and assign that permission set to your user.

### Step 3 - Provision Your Instance

Make sure to complete the Data Cloud provisioning process. You only need to complete it once, after assigning your user **Admin** permission set.

- Click on the **Setup**  in the top right (you might need to reload the page to account for above-assigned permission sets)
- Navigate to **Data Cloud Setup > Setup Home**





- Confirm that all steps completed successfully (as per the screenshot below)

## Are you ready to set up Data Cloud?

- Planning your Data Cloud instance.
- Creating your Data Cloud instance.
- Populating your Data Cloud instance.
- Ensuring your instance is ready.

Your instance is located on: **CDP2-AWS-PROD1-USEAST1** ⓘ

⚠ You may see “Failed to populate your Data Cloud instance, retrying now”. This happens when there’s a high volume of new provisioning requests. In most cases, **this process is resolved by simply waiting until the process completes**. Thanks for your patience as we do everything to meet the extreme demand of this program. ⚠

A screenshot showing a progress bar at the top of a light gray box. Below the progress bar, the text "Are you ready to set up Data Cloud?" is displayed in a large, bold, black font. Underneath this, there is a list of four items, each preceded by a small icon:

- ✓ Planning your Data Cloud instance.
- ✓ Creating your Data Cloud instance.
- ⌚ Failed to populate your Data Cloud instance, retrying now
- ⌚ Ensuring your instance is ready.

## Prepare Your Data



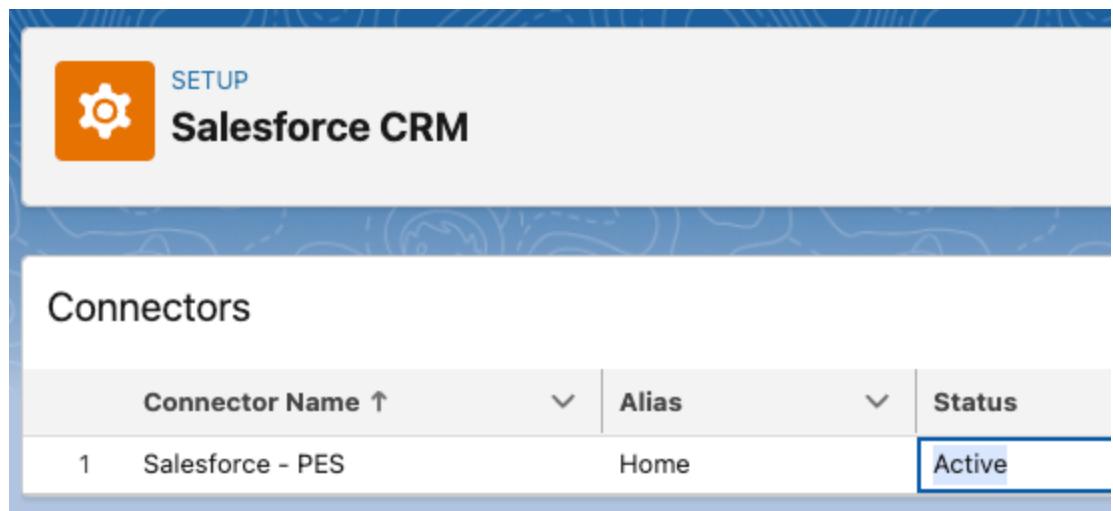
- 1. Update Connection Name as Salesforce – Home Org**
- 2. Service Cloud Bundle Package Installation**
- 3. Salesforce CRM Data – Install Learning Package**
- 4. Initialize data using Flow & Verify Data**
- 5. Create Amazon S3 account**
- 6. Upload files to S3 bucket within folder ecommerce-data**

### Step 4 - Update Connection Name

For our scenario we are loading data from the same Salesforce CRM org where Data Cloud instance is provisioned in. This org is usually referred to as “home org”.

Once the Data Cloud provisioning completes the connection to that CRM org will be established. All that is required here to update the name of that connection.

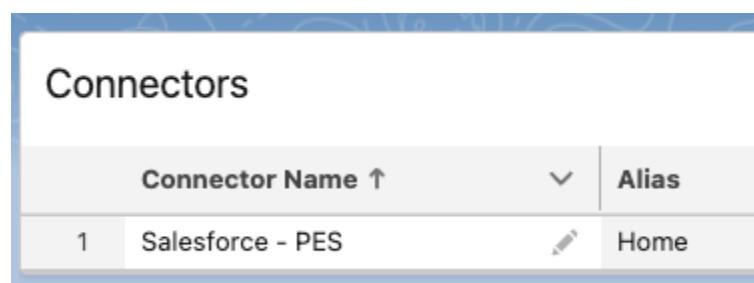
- Navigate to **Data Cloud Setup > Salesforce CRM**
- Confirm that your connection was established and has Active status:



The screenshot shows the 'Salesforce CRM' setup page under 'SETUP'. Below it, the 'Connectors' section lists one entry:

Connector Name ↑	Alias	Status
1 Salesforce - PES	Home	Active

- Next update the name of the connector. This is quite helpful when you connect to more than one org, especially when you start configuring data ingestion.
  - Click on the edit icon in the Name column:



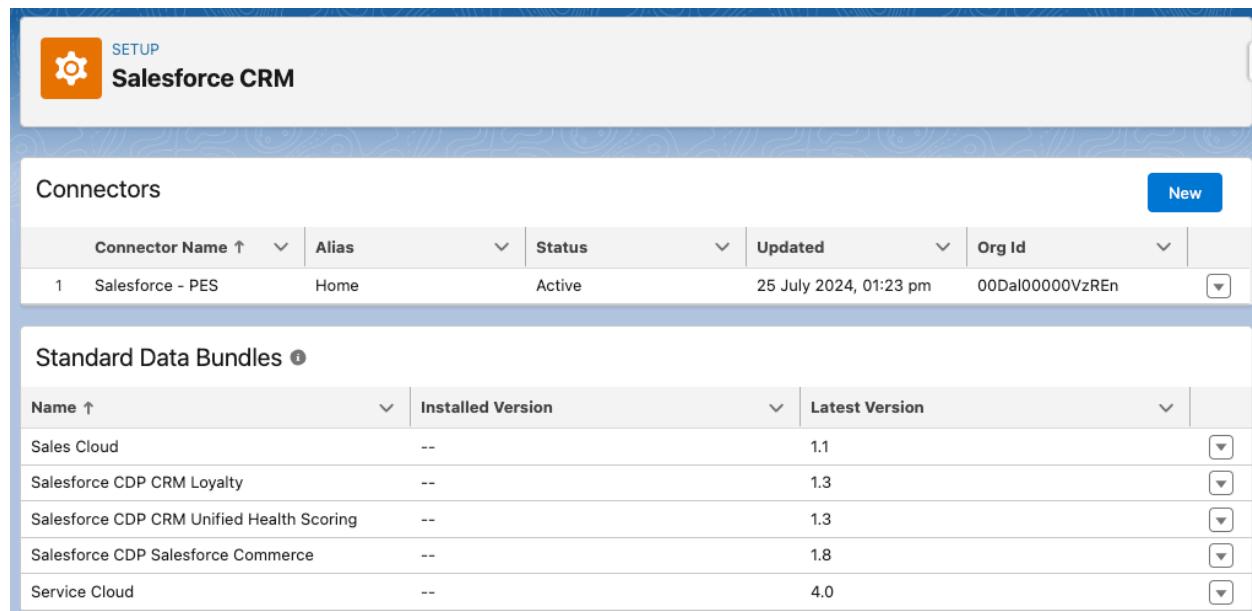
The screenshot shows the 'Salesforce CRM' setup page under 'SETUP'. Below it, the 'Connectors' section lists one entry, with the 'Name' column containing an edit icon:

Connector Name ↑	Alias
1 Salesforce - PES	Home

- Update the name to **Salesforce - Home Org** and save the change:

Connectors		
	Connector Name ↑	Alias
1	Salesforce - Home Org	Home

At this stage the connector for Salesforce CRM will enable data ingestion, but if you need to ingest data using standard data bundles then these will need to be installed separately (as a package), below from the same screen:



The screenshot shows the Salesforce Setup interface with the following sections:

- Connectors:** A table listing one connector:
 

Connector Name ↑	Alias	Status	Updated	Org Id
Salesforce - PES	Home	Active	25 July 2024, 01:23 pm	00Dal00000VzREn
- Standard Data Bundles:** A table listing five bundles:
 

Name ↑	Installed Version	Latest Version
Sales Cloud	--	1.1
Salesforce CDP CRM Loyalty	--	1.3
Salesforce CDP CRM Unified Health Scoring	--	1.3
Salesforce CDP Salesforce Commerce	--	1.8
Service Cloud	--	4.0

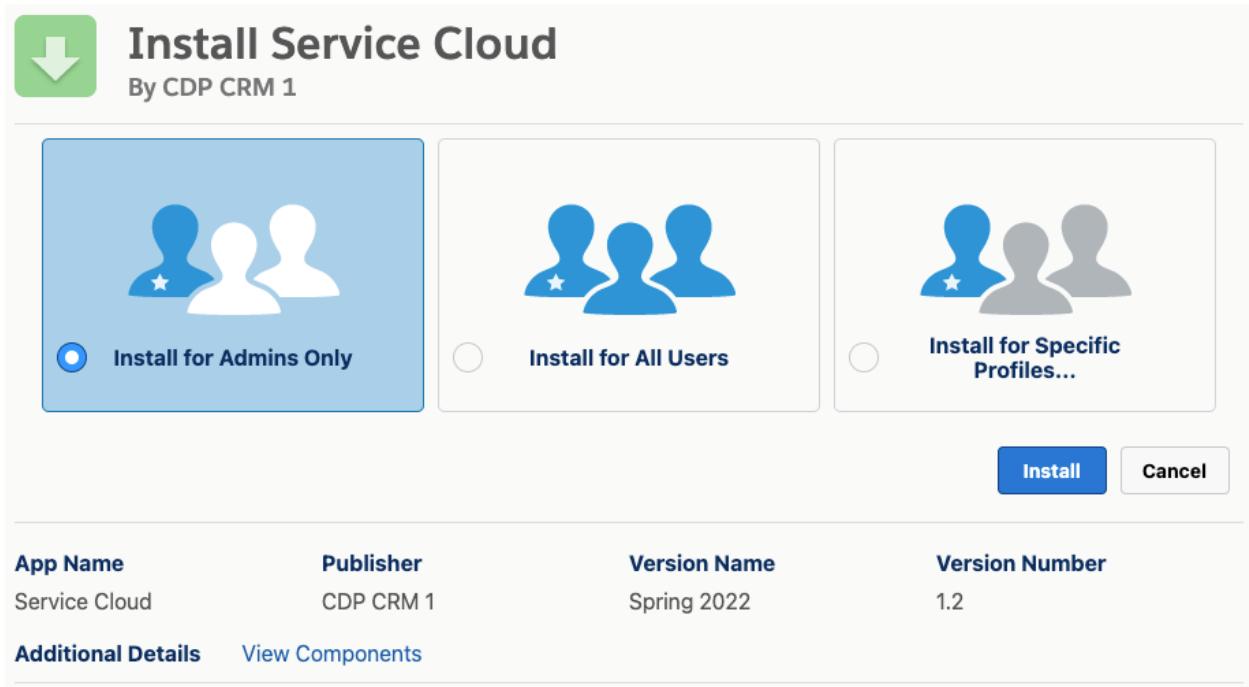
The latest Service Cloud bundle has compatibility issues with the current version of the learning course and will raise conflicts in your trial version of Data Cloud environment. Instead install version 1.2 that was used when this guide was developed and published.

1. In the browser window copy the URL and paste it into the new tab/window.
  - a. In the URL remove anything after the top level domain of your instance, and append the following: /packaging/installPackage.apexp?p0=04tB0000000UpWu
 

i.e.g. something like <https://mydc-dev-ed.lightning.force.com/packaging/installPackage.apexp?p0=04tB0000000UpWu>
  - ii. where mydc-dev-ed.lightning.force.com is the top level domain of my org

- b. Alternatively you can also click on this link and login with your org credentials:  
[https://login.salesforce.com/packaging/installPackage.apexp?p0=04tB000000UpWu&isd\\_tp=p1](https://login.salesforce.com/packaging/installPackage.apexp?p0=04tB000000UpWu&isd_tp=p1)

2. Choose Install For Admin Only option and install the package:



a.

Wait until package install completes or you receive confirmation via email:

 **Install Service Cloud**  
By CDP CRM 1

---

 Installation Complete!

---

**Done**

---

App Name	Publisher	Version Name	Version Number
Service Cloud	CDP CRM 1	Spring 2022	1.2

Package Service Cloud Install Successful   Inbox 

**support@salesforce.com** <support@salesforce.com>

to vsilak@salesforce.com ▾

Your request to install package "Service Cloud Spring 2022" was successful.

Organization: Salesforce - PPS (00Dam000004kuJR)

User: Vladimir Silak (005am000001Lhe5)

Package: Service Cloud (04tB0000000UpWu)

You can close the tab of the browser and return to the **Data Cloud Setup > Salesforce CRM**. Verify that bundle package now has installed version populated:

The screenshot shows the Salesforce Data Cloud interface. In the 'Connectors' section, there is one entry: 'Salesforce - PES' with alias 'Home', status 'Active', and updated on '25 July 2024, 01:23 pm'. In the 'Standard Data Bundles' section, there is a table with columns 'Name', 'Installed Version', and 'Latest Version'. The table contains the following data:

Name	Installed Version	Latest Version
Sales Cloud	--	1.1
Salesforce CDP CRM Loyalty	--	1.3
Salesforce CDP CRM Unified Health Scoring	--	1.3
Salesforce CDP Salesforce Commerce	--	1.8
Service Cloud	1.2	4.0

**Objectives:** In the course of this exercise you will:

- Initialize and verify sample data in your CRM org
- Download sample commerce data files and upload them to Amazon S3 bucket

## 1. Salesforce CRM Data

To configure necessary data model in your CRM org and generate records to be used in this workshop install learning package and run flow included in it.

### 1.1. Install Learning Package

NOTE: The Learning Data Model Package has only been tested with the Data Cloud Trial Org provided as part of the Partner Learning Camp Training Curriculum. It has not been tested with the different SDO, IDO and EDO orgs.

1. Assuming you are logged into your org and have it opened in the browser window copy the URL and paste it into the new tab/window.
  - a. In the URL remove anything after the top level domain of your instance, and append the following: `/packaging/installPackage.apexp?p0=04tDm0000000yOj`

i.e.g. something like <https://mydc-dev-ed.lightning.force.com/packaging/installPackage.apexp?p0=04tDm000000yOj>

ii.where mydc-dev-ed.lightning.force.com is the top level domain of my org

- b. Alternatively you can also click on this link and login with your org credentials:  
<https://login.salesforce.com/packaging/installPackage.apexp?p0=04tDm000000yOj&isdt=p=p1>

2. Choose Install For Admin Only option and install the package:

The screenshot shows the Salesforce package installation dialog for the 'Install Data Cloud - learning data model' package by Salesforce. At the top, there's a green download icon and the package title. Below it, a warning message asks what to do if existing component names conflict with ones in the package, with two options: 'Do not install.' (selected) and 'Rename conflicting components in package.' Underneath, there are three installation options:

- Install for Admins Only** (selected): Shows three blue user icons, one with a star.
- Install for All Users**: Shows three blue user icons, one with a star.
- Install for Specific Profiles...**: Shows three user icons, one blue with a star, one grey, one white.

At the bottom right are 'Install' and 'Cancel' buttons. Below the dialog, detailed package metadata is shown:

App Name	Publisher	Version Name	Version Number
Data Cloud - learning data model	Salesforce	Spring 2023 - Apr 19	1.4

**Description**  
 Follow instructions in this guide to complete data initialization in your org.

**Additional Details** [View Components](#)

a.

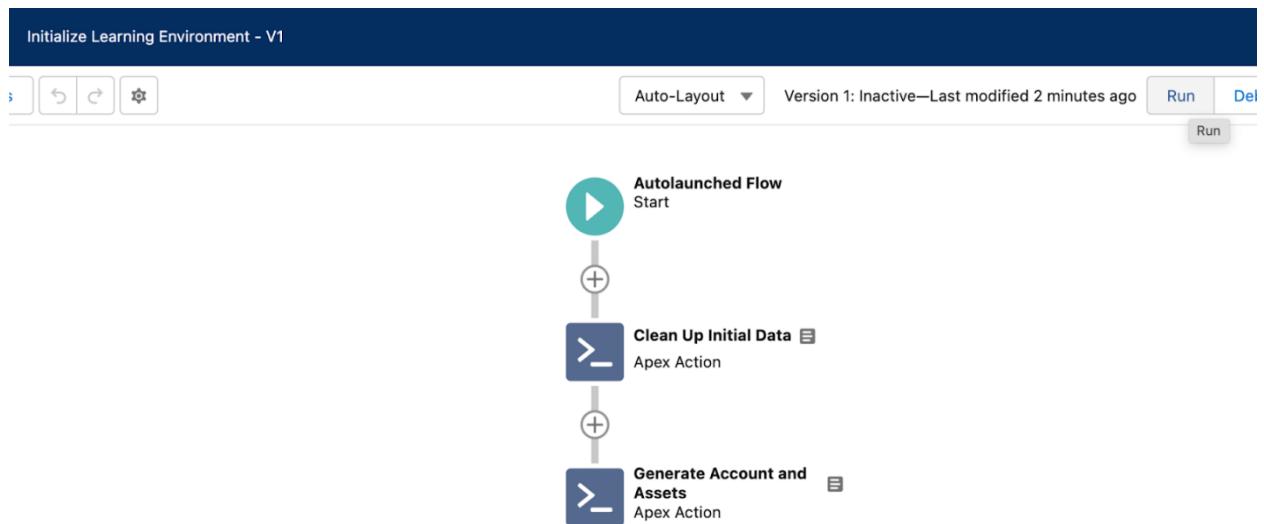
Click **Done** when installation completes.

NOTE: If you get an error stating that ‘There are problems that prevent this package from being installed’ make sure you have followed all of the hands on instructions in the ‘Setup’ hands on activity detailed in [Ex: Data Cloud - Setup Your Instance](#) including the Set Audit Fields upon Record Creation... option

## 1.2. Initialize and Verify Data

In order to prepare the records within your learning org run the flow deployed with the learning package.

- Navigate to **Setup > Process Automation > Flows**
- Locate and open Initialize Learning Environment flow
- Click **Run** button to execute flow



- The system will open new tab/window and upon completion you should see the following message:

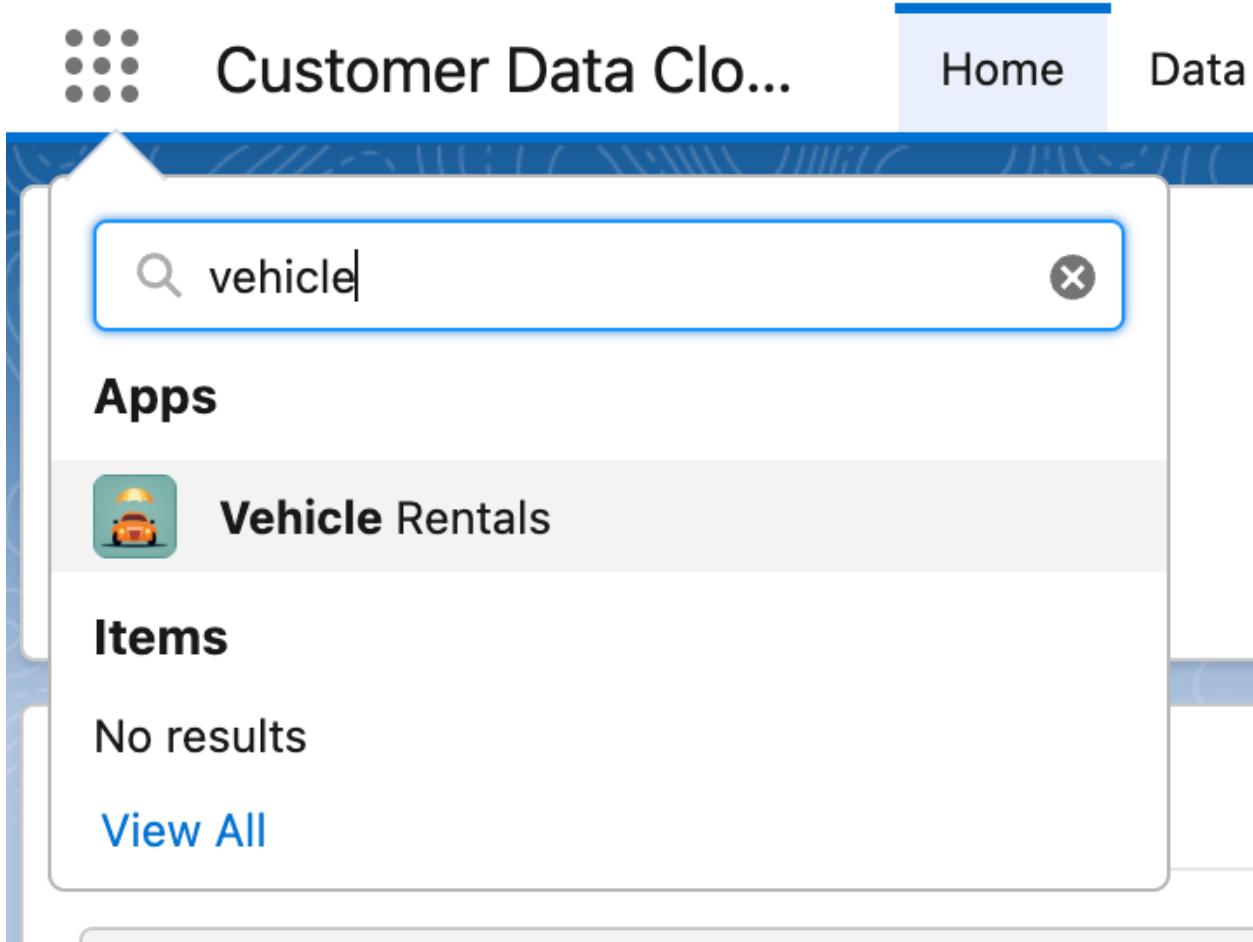
**Initialize Learning Environment**

---

**YOUR FLOW FINISHED**

- Close this browser tab/window.

- Next verify records that were created as a result of this flow execution.
- In the application switcher (top left) locate and open **Vehicle Rentals** application.



- On the **Rentals** tab change list view to All and verify that there are 22 records

**Vehicle Rentals**

Rentals ▾ Bookings ▾

**Rentals**

**Recently Viewed ▾**

22 items

**LIST VIEWS**

All

1 ✓ Recently Viewed (Pinned list)

**Vehicle Rentals**

Rentals ▾ Bookings ▾ Contacts ▾ Rental Preferences ▾ Cases ▾ Accounts ▾

Search...

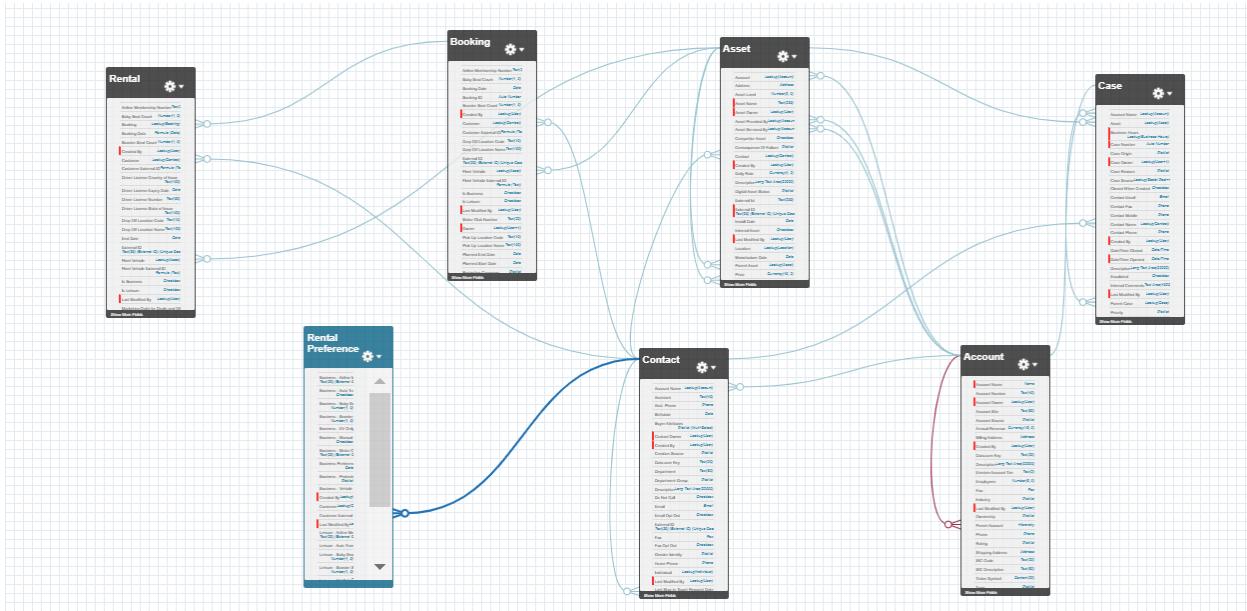
New Import Change Owner Printable View Assign Label

Search this list...

	Rental ID	Actions
1	R-202506-0022	⋮
2	R-202506-0023	⋮
3	R-202506-0024	⋮
4	R-202506-0025	⋮
5	R-202506-0026	⋮
6	R-202506-0027	⋮
7	R-202506-0028	⋮
8	R-202506-0029	⋮
9	R-202506-0030	⋮
10	R-202506-0031	⋮
11	R-202506-0032	⋮

- Similar to that verify other tabs:

- Bookings = 8 records
- Contacts = 100 records
- Rental Preferences = 7 records
- Assets = 25 records
- Cases = 5 records



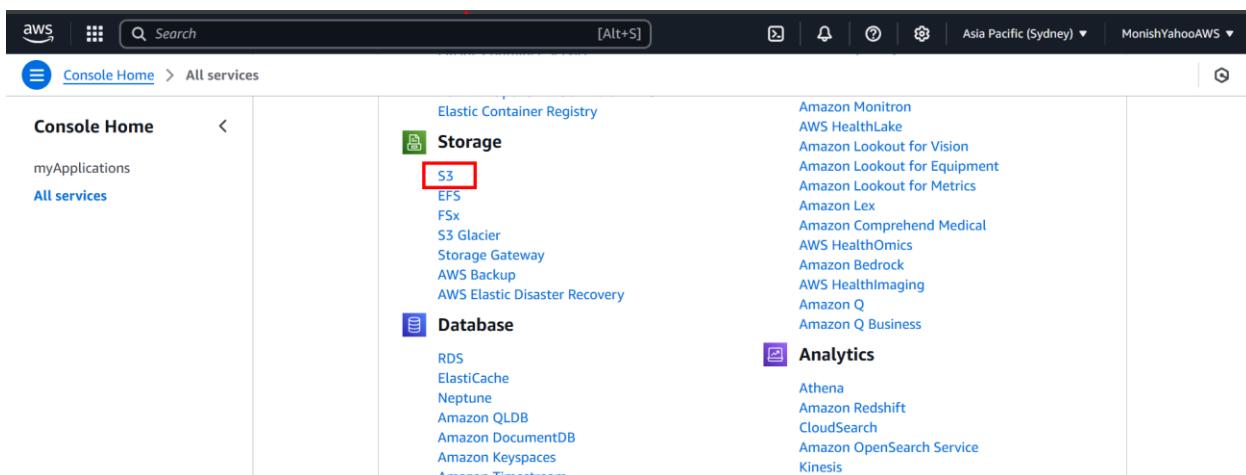
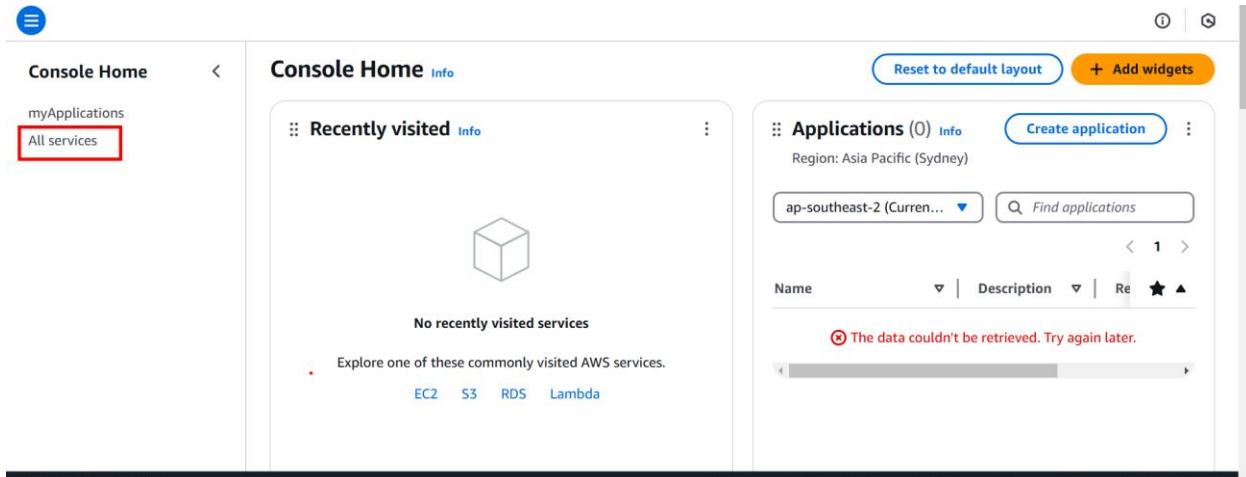
As long as you've got all the counts as expected the data is ready and you can proceed to the next step.

## 2. S3 Data From Commerce Platform

If you already have an S3 instance that you can use no need to sign up for a new one. Take a note of your [programmatic credentials](#) (Access Key ID and Secret Access Key) that can be used to access the account.

If you don't have [Amazon S3](#) account - [register for the free tier](#), and then follow instructions in [Setup Amazon S3 User For Data Cloud Guide](#) to create dedicated user with limited permissions for this workshop.

After signup:



Bucket Name - <monish>s3bucket

Leaving all default options and submit

Click on bucket <monish>s3bucket

Create folder:

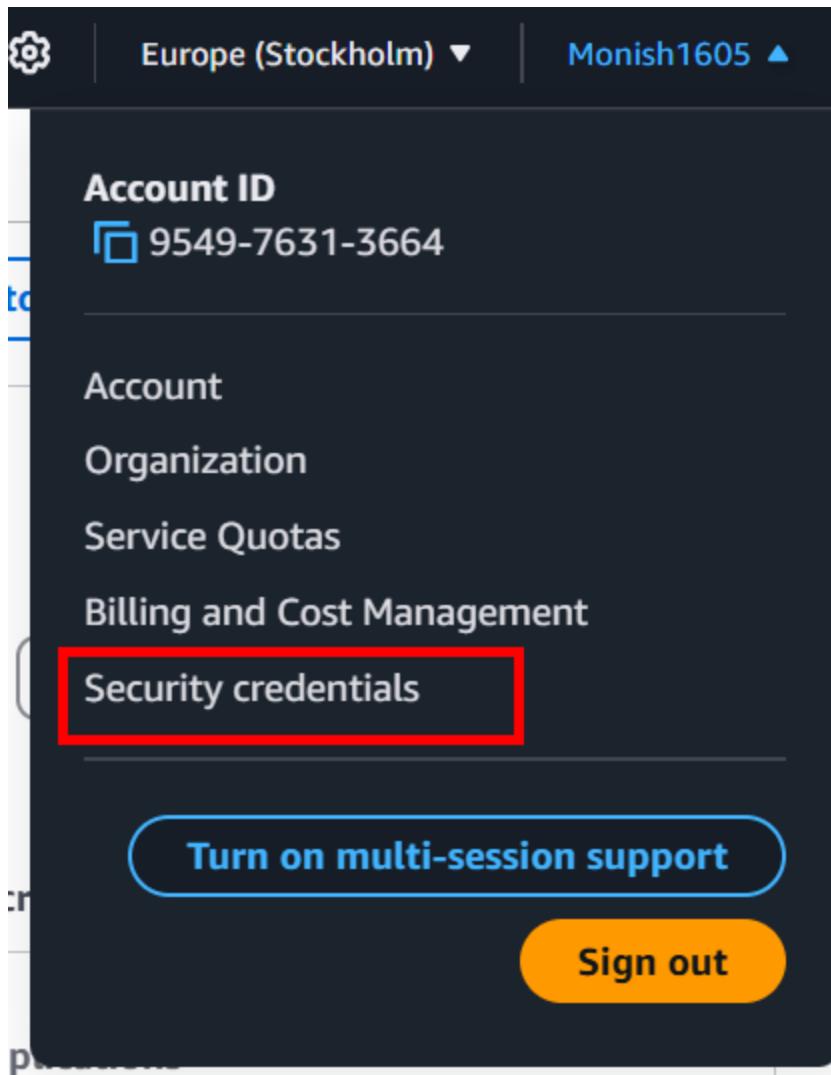
ecommerce-data

Leave all default options and scroll down - Create it

Select folder -> Click upload – Add files

Select 3 files shared in the Teams

Upload



On Top Right profile -> Security Creds -> Access Key - Create access key

The screenshot shows the AWS IAM Access Keys page. At the top, it says "Access keys (0)" and has a "Create access key" button. A note below explains that access keys can be used for AWS CLI, PowerShell, and SDK calls. The main table header includes columns for "Access key ID", "Created on", "Access key last used", "Region last used", "Service last used", and "Status". A message at the bottom encourages using short-term credentials instead of long-term access keys.

Access key ID	Created on	Access key last used	Region last used	Service last used	Status
No access keys					

As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials.[Learn more](#)

## Alternatives to root user access keys Info

 Root user access keys are not recommended

We don't recommend that you create root user access keys. Because you can't specify the root user in a permissions policy, you can't limit its permissions, which is a best practice.

Instead, use alternatives such as an IAM role or a user in IAM Identity Center, which provide temporary rather than long-term credentials. [Learn More](#) 

If your use case requires an access key, create an IAM user with an access key and apply least privilege permissions for that user. [Learn More](#) 

### Continue to create access key?

I understand creating a root access key is not a best practice, but I still want to create one.

[Cancel](#)

[Create access key](#) 

[Copy paste Access Key and Secret access Key](#)

 **Note:** for the workshop we use .csv files that are ingested from S3. The import process supports GZIP compression too, so if the inbound data volume is large it might be practical to compress it before deploying to S3 and subsequently ingesting into Customer Data Platform. ZIP compression while supported might cause long term challenges as this compression is not supported with **Aggregate Mode** setting that results in multiple files being combined prior to import.

### 2.1 Upload files to S3 bucket

Log into Management Console and proceed to [S3 service](#) and create a new bucket (name it as you wish). Note, if you followed the guide to setup permissions you don't need to create another bucket.



[Download following files to your computer:](#)

- [S3 Customer Profile.csv](#) (100 records)

- [S3 Order Headers.csv](#) (252 records)  
*depending on your browser you might need to use Right-Click + Save As to save this (Order Headers) file*
- [S3 Order Line Items.csv](#) (649 records)

To download the S3 Order Headers.csv file, please Right+Click and Save As

In your [S3 bucket](#) create a folder called ecommerce-data and upload previously downloaded files in this directory.

Amazon S3 > Buckets > ravg-summer22 > ecommerce-data/ > Upload

## Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose **Add files**, or **Add folders**.

Files and folders (3 Total, 85.7 KB)					<a href="#">Remove</a>	<a href="#">Add files</a>	<a href="#">Add folder</a>
All files and folders in this table will be uploaded.							
<input type="text"/> <a href="#">Find by name</a> <span style="float: right;">&lt; 1 &gt;</span>							
<input type="checkbox"/>	Name	Folder	Type	Size			
<input type="checkbox"/>	S3 Customer Profile.csv	-	text/csv	9.7 KB			
<input type="checkbox"/>	S3 Order Headers.csv	-	text/csv	40.7 KB			
<input type="checkbox"/>	S3 Order Line Items.csv	-	text/csv	35.3 KB			

## ecommerce-data/

[Objects](#) [Properties](#)

### Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 invent](#) them permissions. [Learn more](#)

[Find objects by prefix](#)

<input type="checkbox"/>	Name	Type
<input type="checkbox"/>	<a href="#">S3 Customer Profile.csv</a>	CSV
<input type="checkbox"/>	<a href="#">S3 Order Headers.csv</a>	CSV
<input type="checkbox"/>	<a href="#">S3 Order Line Items.csv</a>	CSV

## CONFIGURE DATA INGESTION:

In this exercise you will go through the configuration of the required data streams to ingest the data from multiple sources into your Data Cloud instance.

The screenshot shows the Salesforce Setup interface with the 'Permission Sets' page open. The left sidebar shows navigation options like 'Users', 'Apps', and 'Feature Settings'. The main area displays a list of permission sets, with 'Data Cloud Salesforce Connector' selected and highlighted with a red box. The 'Object Settings' section in the detail view is also highlighted with a red box.

Action	Permission Set Name	Description	License
<input type="checkbox"/> Clone	Data Cloud Home Org Integration User	Allows integration user to access entities specific to ...	Cloud Integration User
<input type="checkbox"/> Clone	Data Cloud Marketing Admin	Allows access to Data Cloud Setup if the user is also...	Customer Data Cloud for Marketing
<input type="checkbox"/> Clone	Data Cloud Marketing Manager	Gives view access to most Data Cloud features and ...	Customer Data Cloud for Marketing
<input type="checkbox"/> Clone	Data Cloud Marketing Specialist	Gives view access to most Data Cloud features a...	Customer Data Cloud for Marketing
<input checked="" type="checkbox"/> Clone	Data Cloud Salesforce Connector	Allows object and field access for ingestion to Dat...	Cloud Integration User
<input type="checkbox"/> Clone	Data Cloud User	Gives view access to Data Cloud	Data Cloud
<input type="checkbox"/> Clone	DeliveryEstimationServicePermSet		Cloud Integration User

Go to contact Object in Object setting and give Read access

Same step to case , account object for Read access.

Click Edit and at field level, check the Read only check box and **SAVE**.

**Permission Sets**

**Data Cloud Salesforce Connector**

Object Permissions

Permission Name	Enabled
Read	<input checked="" type="checkbox"/>
Create	<input type="checkbox"/>
Edit	<input type="checkbox"/>
Delete	<input type="checkbox"/>
View All Records	<input checked="" type="checkbox"/>
Modify All Records	<input type="checkbox"/>
View All Fields	<input checked="" type="checkbox"/>

Field Permissions

Field Name	Field API Name	Read Access	Edit Access
Account Name	Accountid	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Asset	Assetid	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Business Hours	BusinessHoursId	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Case Number	CaseNumber	<input type="checkbox"/>	<input type="checkbox"/>
Case Origin	Origin	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Case Owner	OwnerId	<input type="checkbox"/>	<input type="checkbox"/>

**Permission Sets**

**Data Cloud Salesforce Connector**

Object Permissions

Permission Name	Enabled
Read	<input checked="" type="checkbox"/>
Create	<input type="checkbox"/>
Edit	<input type="checkbox"/>
Delete	<input type="checkbox"/>
View All Records	<input checked="" type="checkbox"/>
Modify All Records	<input type="checkbox"/>
View All Fields	<input checked="" type="checkbox"/>

Field Permissions

Field Name	Field API Name	Read Access	Edit Access
Account Name	Accountid	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Asset	Assetid	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Business Hours	BusinessHoursId	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Case Number	CaseNumber	<input type="checkbox"/>	<input type="checkbox"/>
Case Origin	Origin	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Case Owner	OwnerId	<input type="checkbox"/>	<input type="checkbox"/>

**Setup** | Home | Object Manager | **Permission Sets**

**Data Cloud Salesforce Connector**

Object Permissions

Permission Name	Enabled
Read	<input checked="" type="checkbox"/>
Create	<input type="checkbox"/>
Edit	<input type="checkbox"/>
Delete	<input type="checkbox"/>
View All Records	<input checked="" type="checkbox"/>
Modify All Records	<input type="checkbox"/>
View All Fields	<input checked="" type="checkbox"/>

Field Permissions

Field Name	Field API Name	Read Access	Edit Access
Account Name	Name	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Account Number	AccountNumber	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Account Owner	OwnerId	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Account Site	Site	<input checked="" type="checkbox"/>	<input type="checkbox"/>

The screenshot shows the Salesforce Setup interface. On the left, the navigation sidebar includes 'Hyperforce Assistant', 'Users' (with 'Permission Set Groups' and 'Permission Sets' selected), 'Apps' (with 'Mobile Apps', 'Salesforce' (including 'Mobile Builder for the Seller-Focused Experience'), 'Feature Settings', 'Digital Experiences', 'Salesforce CMS', 'Settings', 'Functions', 'Sales' (with 'Accounts' (selected) and 'Person Accounts'), and 'Salesforce Scheduler'. The main content area is titled 'Permission Sets' and shows a permission set named 'Data Cloud Salesforce Connector'. It includes sections for 'Object Permissions' (listing Read, Create, Edit, Delete, View All Records, Modify All Records, and View All Fields with checkboxes for Enabled status) and 'Field Permissions' (listing Account Name, Assistant, Asst. Phone, and Birthdate with checkboxes for Field API Name, Read Access, and Edit Access). The 'Save' button is highlighted with a red box.

## Data cloud in AppLauncher

The screenshot shows the Data Cloud Setup page in the AppLauncher. The sidebar on the left lists 'App Launcher' (selected), 'Data Cloud' (selected), and other items like 'Data Spaces', 'Feature Manager', 'Admin Tools', 'Developer Tools', 'SALEFORCE INTEGRATIONS', 'EXTERNAL INTEGRATIONS', and 'CloudWatch Metrics'. The main content area is titled 'Welcome to Data Cloud' and includes a sub-section 'Your Home Org Details' stating 'Your Data Cloud instance is live and connected to your home org.' At the bottom, there is a table with columns: Home Org ID (000g10000057CBF), Home Org Instance (CDP2-AWS-PROD8-CACENTRAL1), Tenant Endpoint (gm3w0y10gq3wky3dmjstsmdbmq.c360a.salesforce.com), Data Spaces (1), and Connections (1).

Welcome to Data Cloud

Secure Transform Unify Explore Segment Act

Bring together data from all of your systems to generate unified profiles of your customers.

**Get Started**

Learn about Data Cloud with quick videos

Find Data Cloud documentation

Understand usage-based pricing

**Quarterly Performance**

CLOSED \$0 OPEN (>70%) \$0 GOAL --

500K  
400K  
300K  
200K  
100K  
0

May June July Aug Sep

Add the opportunities you're working on, then come back here to view your performance.

**Today's Events**

Looks like you're free and clear the rest of the day.

**Today's Tasks**

Nothing due today. Be a go-getter, and check back soon.

**Recent Records**

**Key Deals - Recent Opportunities**

As of Today 5:16 AM

Resources

- About Salesforce Data Cloud
- Data Cloud Features and Learning Path
- Get Started Using Data Cloud
- Learn About Data, Ethics, and Consent
- Data Cloud Developer Center

Nothing needs your attention right now. Check back later.

Learn About Data Cloud

Welcome to Data Cloud

Secure Transform Unify Explore Segment Act

Bring together data from all of your systems to generate unified profiles of your customers.

**Get Started**

Learn about Data Cloud with quick videos

Find Data Cloud documentation

Understand usage-based pricing

**Quarterly Performance**

CLOSED \$0 OPEN (>70%) \$0 GOAL --

500K  
400K  
300K

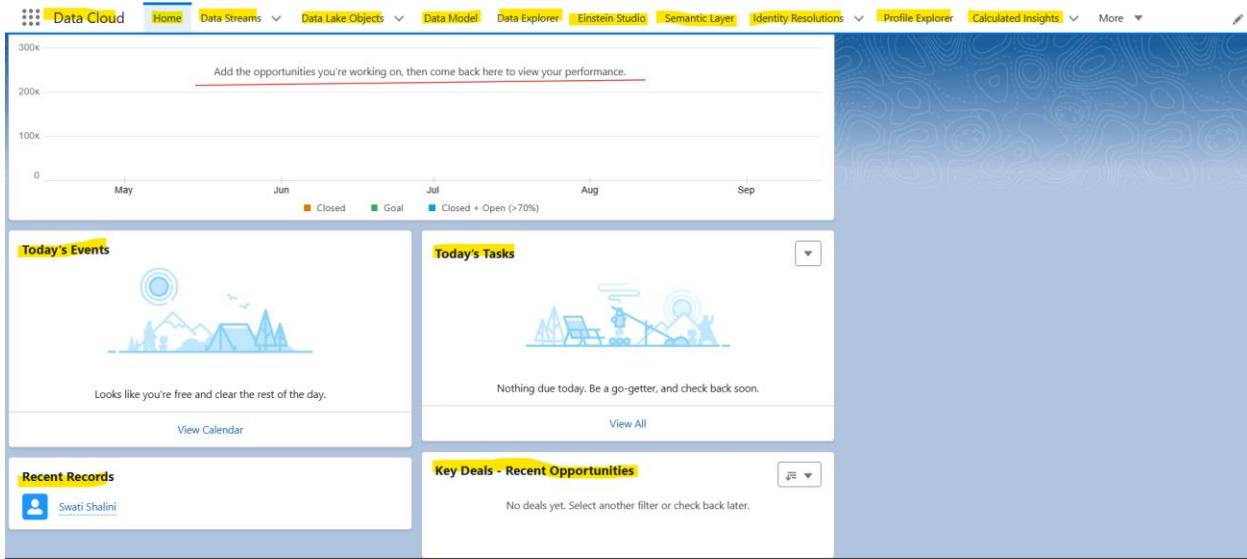
As of Today 5:16 AM

**Assistant**

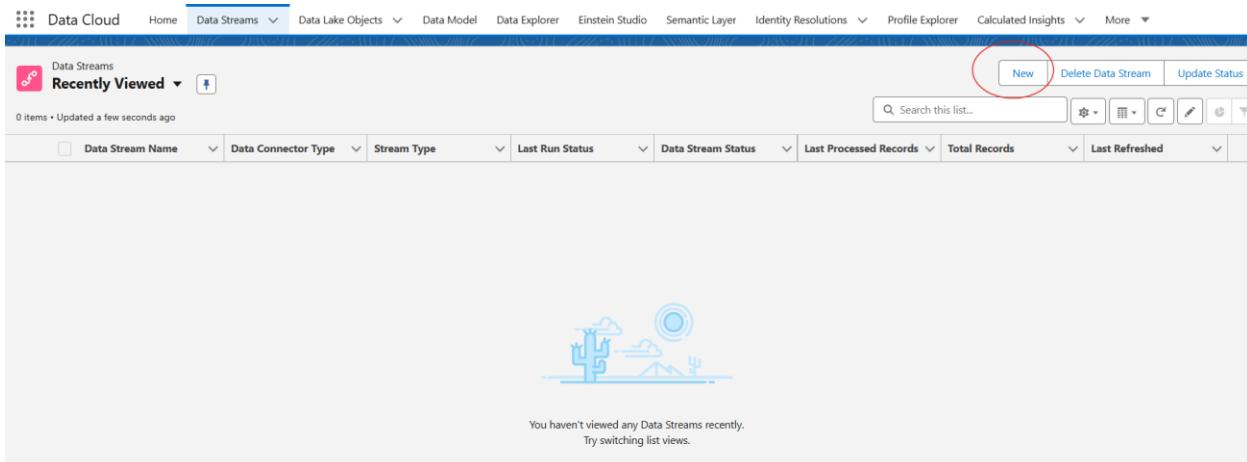
Nothing needs your attention right now. Check back later.

Resources

- About Salesforce Data Cloud
- Data Cloud Features and Learning Path
- Get Started Using Data Cloud
- Learn About Data, Ethics, and Consent
- Data Cloud Developer Center



## Data Stream:



**New Data Stream**

Select the data source from which you can ingest or federate data. Only sources that are already connected to Data Cloud appear on this list. [Learn More](#)

**Connected Sources**

- Salesforce CRM
 

Import objects from Salesforce CRM

**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.

- File Upload
 

Upload file from your local drive
- Installed Data Kits & Packages
 

Import data streams from preconfigured data kits and packages.

**Explore Other Connectors**

Filter Connectors  Generally Available  Beta

[Next](#)

**New Data Stream**

To ensure data is ingested from fields and objects created in the future, we recommend granting View All Fields (Global) system permission on the Data Cloud Salesforce Connector permission set. [Learn More](#)

Select an org to ingest data from, then select an object or data bundle.

**Salesforce Org**

Salesforce - Home Org [View Bundles](#) [View Objects](#)

Object	Description
Digital Engagement	5 Objects • Created by Salesforce
Field Service Related	4 Objects • Created by Salesforce
Sales	3 Objects • Created by Salesforce
Service	3 Objects • Created by Salesforce
Field Service Core	25 Objects • Created by Salesforce
Incident Management	9 Objects • Created by Salesforce
Salesforce Order Management	30 Objects • Created by Salesforce

**Bundle Details**

**Service**

Type: Standard Data Bundle  
Description: Automate service processes and streamline workflows to transform the agent experience.

**Objects included (3)**

Account Case Contact

[Previous](#) [Next](#)

**New Data Stream**

Data Cloud creates a data lake object (DLO) to store data from each Salesforce object in the data stream. Select an object to configure details for its related DLO. After saving the data stream, you can't change these configurations.

**Service Objects**

Account

Case

Contact

**Account Fields (68)**

55 fields selected

Required	Field Name	Field Label	Field API Name
1 <input checked="" type="checkbox"/>	Year Started	Year Started	YearStarted
2 <input checked="" type="checkbox"/>	Website	Website	Website
3 <input checked="" type="checkbox"/>	Account Type	Account Type	Type
4 <input checked="" type="checkbox"/>	Tradestyle	Tradestyle	Tradestyle
5 <input checked="" type="checkbox"/>	Ticker Symbol	Ticker Symbol	TickerSymbol
6 <input checked="" type="checkbox"/>	System Modstamp	System Modstamp	SystemModstamp
7 <input checked="" type="checkbox"/>	Account Site	Account Site	Site
8 <input checked="" type="checkbox"/>	SIC Description	SIC Description	SicDesc
9 <input checked="" type="checkbox"/>	SIC Code	SIC Code	Sic

[Previous](#) [Next](#)

### Go to contact and EDIT custom field, API field n save

New Data Stream

Data Cloud creates a data lake object (DLO) to store data from each Salesforce object in the data stream. Select an object to configure details for its related DLO. After saving the data stream, you can't change these configurations.

**Contact Fields (74)**

Service Objects: Account, Case, **Contact**

Standard Fields (69) Custom Fields (5) Formula Fields (0)

Field Name	Field Label	Field API Name	Data Type
RAVG Retail Points Balance	RAVG Retail Points Balance	RAVG_Retail_Points_Balance_c	Number
RAVG Retail Membership Number	RAVG Retail Membership Number	RAVG_Retail_Membership_Number_c	Text
Level	Level	Level_c	Text
Languages	Languages	Languages_c	Text
External ID	External ID	External_ID_c	Text

Previous Next

New Data Stream

Data Cloud creates a data lake object (DLO) to store data from each Salesforce object in the data stream. Select an object to configure details for its related DLO. After saving the data stream, you can't change these configurations.

**Contact Fields (74)**

Service Objects: Account, Case, **Contact**

Standard Fields (69) Custom Fields (5) Formula Fields (0)

Field Name	Field Label	Field API Name	Data Type
RAVG Retail Points Balance	RAVG Retail Points Balance	RAVG_Retail_Points_Balance_c	Number
RAVG Loyalty ID	RAVG Loyalty ID	RAVG_Loyalty_ID_c	Text
Level	Level	Level_c	Text
Languages	Languages	Languages_c	Text
External ID	External ID	External_ID_c	Text

Previous Next

**No space so below is right one:**

New Data Stream

Data Cloud creates a data lake object (DLO) to store data from each Salesforce object in the data stream. Select an object to configure details for its related DLO. After saving the data stream, you can't change these configurations.

**Contact Fields (74)**

Service Objects

- Account
- Case
- Contact**

Standard Fields (69)   Custom Fields (5)   Formula Fields (0)

Field Name	Field Label	Field API Name	Data Type
RAVG Retail Points Balance	RAVG Retail Points Balance	RAVG_Retail_Points_Balance_c	Number
<b>RAVG Loyalty ID</b>	<b>RAVG Loyalty ID</b>	<b>RAVG_Loyalty_ID_c</b>	Text
Level	Level	Level_c	Text
Languages	Languages	Languages_c	Text
External ID	External ID	External_ID_c	Text

Previous      Next

## Deploy

New Data Stream

Put the finishing touches on your data stream(s).

\* Data Space **default**

This bundle creates 3 data stream(s) and data lake object(s).

**'Service' Data Stream Bundle Configuration Details**

Object Name	Category	Refresh Mode	Data Space Filtering
1 Contact_Home	Profile	Upsert	Set Filters
2 Account_Home	Profile	Upsert	Set Filters
3 Case_Home	Engagement	Upsert	Set Filters

Frequently Asked Questions

**What are data space filters?**  
Data space filters let you determine which records from the data lake object are available in the context of a data space.

**What is a refresh mode?**  
After the initial data ingestion, you can opt to replace only the fields for which new data was received (partial refresh) or to replace the entire record with the data received (incremental refresh). When refresh mode is incremental, existing values can be replaced by blank values.

Deploy

You can create a formula field in contact / case / account Obj and Deploy....

New Data Stream

Data Cloud creates a data lake object (DLO) to store data from each Salesforce object in the data stream. Select an object to configure details for its related DLO. After saving the data stream, you can't change these configurations.

Service Objects		Account Fields (68)																																			
Account		56 fields selected																																			
Case		Search...																																			
Contact		New Formula Field																																			
		Standard Fields (61)    Custom Fields (7)    Formula Fields (0) <table border="1"> <thead> <tr> <th>Field Name</th> <th>Field Label</th> <th>Field API Name</th> <th>Data Type</th> </tr> </thead> <tbody> <tr><td>1 Upsell Opportunity</td><td>Upsell Opportunity</td><td>UpsellOpportunity_c</td><td>Text</td></tr> <tr><td>2 SLA Serial Number</td><td>SLA Serial Number</td><td>SLASerialNumber_c</td><td>Text</td></tr> <tr><td>3 SLA Expiration Date</td><td>SLA Expiration Date</td><td>SLAEExpirationDate_c</td><td>Date (yyyy-MM-dd)</td></tr> <tr style="background-color: yellow;"><td>4 SLA</td><td>SLA 2025</td><td>SLA_2025_c</td><td>Text</td></tr> <tr><td>5 Number of Locations</td><td>Number of Locations</td><td>NumberofLocations_c</td><td>Number</td></tr> <tr><td>6 Customer Priority</td><td>Customer Priority</td><td>CustomerPriority_c</td><td>Text</td></tr> <tr><td>7 Active</td><td>Active</td><td>Active_c</td><td>Text</td></tr> </tbody> </table>				Field Name	Field Label	Field API Name	Data Type	1 Upsell Opportunity	Upsell Opportunity	UpsellOpportunity_c	Text	2 SLA Serial Number	SLA Serial Number	SLASerialNumber_c	Text	3 SLA Expiration Date	SLA Expiration Date	SLAEExpirationDate_c	Date (yyyy-MM-dd)	4 SLA	SLA 2025	SLA_2025_c	Text	5 Number of Locations	Number of Locations	NumberofLocations_c	Number	6 Customer Priority	Customer Priority	CustomerPriority_c	Text	7 Active	Active	Active_c	Text
Field Name	Field Label	Field API Name	Data Type																																		
1 Upsell Opportunity	Upsell Opportunity	UpsellOpportunity_c	Text																																		
2 SLA Serial Number	SLA Serial Number	SLASerialNumber_c	Text																																		
3 SLA Expiration Date	SLA Expiration Date	SLAEExpirationDate_c	Date (yyyy-MM-dd)																																		
4 SLA	SLA 2025	SLA_2025_c	Text																																		
5 Number of Locations	Number of Locations	NumberofLocations_c	Number																																		
6 Customer Priority	Customer Priority	CustomerPriority_c	Text																																		
7 Active	Active	Active_c	Text																																		

Previous      Next

New Data Stream

New Formula Field

Formula is invalid, please verify.

Choose items from the functions and fields tabs to create a formula expression.

* Field Label: SLA 2025	* Field API Name: SLA_2025_c	* Formula Return Type: Text
Search function	Attributes	
Search attribute		Tested Value
SLA		SLA
Active		20
Deleted		Test
		Output
		20

Back      Save

Previous      Next

New Data Stream

\* Data Space: default

This bundle creates 3 data stream(s) and data lake object(s).

'Service' Data Stream Bundle Configuration Details

Object Name	Category	Refresh Mode	Data Space Filtering
1 Account_Home	Profile	Upsert	Set Filters
2 Contact_Home	Profile	Upsert	Set Filters
3 Case_Home	Engagement	Upsert	Set Filters

**What are data space filters?**  
Data space filters let you determine which records from the data lake object are available in the context of a data space.

**What is a refresh mode?**  
After the initial data ingestion, you can opt to replace only the fields for which new data was received (partial refresh) or to replace the entire record with the data received (incremental refresh). When refresh mode is incremental, existing values can be replaced by blank values.

Deploy

## You can Refresh or Run the Obj

The screenshot shows the Salesforce Data Cloud interface with the 'Data Streams' tab selected. A list of three data streams is displayed:

	Data Stream Name	Data Connector Type	Stream Type	Last Run Status	Data Stream Status	Last Processed Rec...	Total Records	Last Refreshed
1	Case_Home	Salesforce CRM	Ingest	None	Active			
2	Account_Home	Salesforce CRM	Ingest	None	Active			
3	Contact_Home	Salesforce CRM	Ingest	None	Active			

A context menu is open over the 'Contact\_Home' row, with the 'Refresh Now' option highlighted.

**Objectives:** In the course of this exercise you will:

- Configure data ingestion from Salesforce CRM and Amazon S3 bucket
- Apply on-import transformation to data
- Define additional auxiliary attributes to assist with data modelling and support data lineage

**Note:** Throughout this exercise you will require to replace following placeholders with respective values:

<ORG\_ALIAS> = your Salesforce CRM instance identifier (15 char org ID), appended to the data stream names by default.

### Preamble

As you complete the tasks in this exercise notice additional formula fields that are created throughout the data streams. These mainly fall into 3 categories:

- referential data points enabling data lineage (e.g. supporting feeds from multiple geographies)
- elements of clean-up and encapsulation of business logic to derive simple values to be used in segmentation
- additional auxiliary data points that enable distributed, normalized data load fit for Customer 360 Data Model

The last category is the trickiest as you need to ensure uniqueness of the values across one or more entities. Due to “flattened” incoming data set where multiple normalized entities in other system(s) were included in single record often without additional required identifiers these are now need to be created before we can map data to the Customer 360 Data Model.

Please note that we provide implementation approach that works for the simplified use case that underpins current workshop. In real life implementations these decisions need to be made way before implementation, during the data discovery, audit and mapping design exercises.

## Lookup Objects

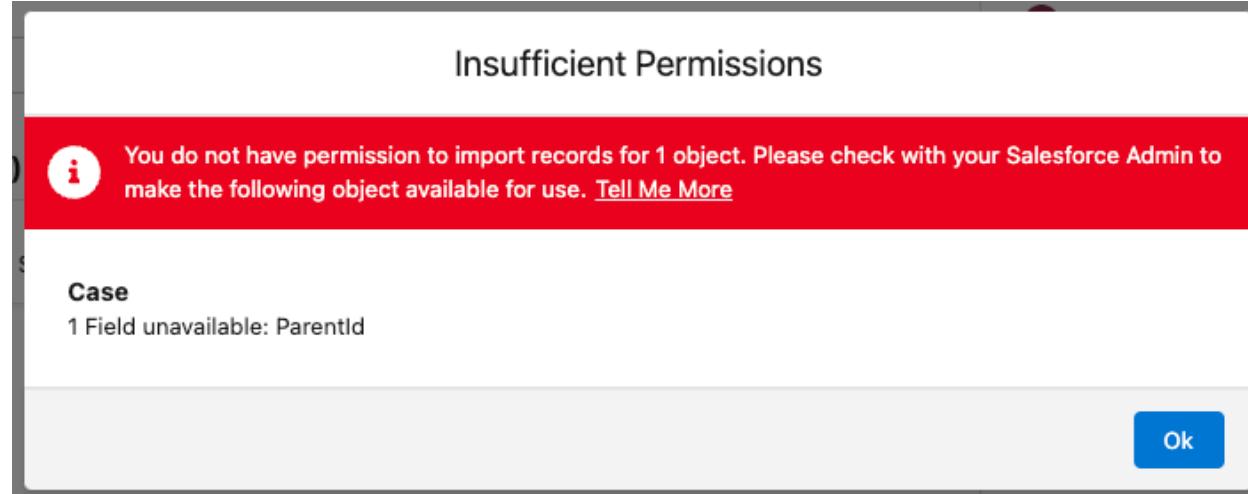
There are number of attributes designed to be used as foreign keys within the Customer 360 Data Model objects. And subset of the above-mentioned auxiliary data points provide values for these objects. We will refer to these as **Lookup entities** within the workshop guide and use additional data streams that make use of the same data sources, although configured with variations to avoid creation of the duplicate values in the Data Model.

### 1. Salesforce CRM Data

#### 1.1. Service Cloud Data Bundle

This data bundle includes **Account**, **Contact** and **Case** objects. For the purpose of our workshop we will only make use of **Contact** and **Case** objects. We will go through the experience of removing unnecessary object (Account) from the deployed bundle configuration.

- Navigate to **Data Cloud** application and open **Data Streams** tab
- Following instructions from [help page](#)
  - Start configuration of the **Service** Standard Data Bundle
  - 
  - Upon selection of the bundle and attempt to proceed further you'll get an error message



- Follow the steps outlined in the help page [linked in the notification](#) to enable **Read Access** field level permission for the Parent Case field of the **Cases** object.

### Field Permissions

Field Name	Read Access	Edit Access
Account Name	<input type="checkbox"/>	<input type="checkbox"/>
Asset	<input type="checkbox"/>	<input type="checkbox"/>
Business Hours	<input type="checkbox"/>	<input type="checkbox"/>
Case Number	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Case Origin	<input type="checkbox"/>	<input type="checkbox"/>
Case Owner	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Case Reason	<input type="checkbox"/>	<input type="checkbox"/>
Case Source	<input type="checkbox"/>	<input type="checkbox"/>
Closed When Created	<input type="checkbox"/>	<input type="checkbox"/>
Contact Email	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Contact Fax	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Contact Mobile	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Contact Name	<input type="checkbox"/>	<input type="checkbox"/>
Contact Phone	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Created By	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Date/Time Closed	<input type="checkbox"/>	<input type="checkbox"/>
Date/Time Opened	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Description	<input type="checkbox"/>	<input type="checkbox"/>
Escalated	<input type="checkbox"/>	<input type="checkbox"/>
Internal Comments	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Last Modified By	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Parent Case	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Priority	<input type="checkbox"/>	<input type="checkbox"/>

Do take a note about the need to enable necessary access permissions when connecting to the Salesforce CRM platform. While current error only highlights the need to update field level in real implementations you'll find the need to enable access to objects as well as fields.

- Following similar steps update field level **Permissions for Contacts object, custom fields External ID and RAVG Retail Membership Number enabling Read Access.**
- Once permissions updated retry selecting the bundle and proceeding to the next step of the configuration wizard

During this configuration step you can rename the Field Label and Field API Name values for the fields that will be ingested into the platform. Use it as an opportunity to simplify or refine the source field names to ease mapping steps later in the process.

⚠️ At the time of writing should you make any changes to the field selections prior to renaming Field Label and/or API Name, or adding new formula field these selections will be reset and all fields will be re-selected again. Hence the steps below are structured in a way that avoids repetitive actions.

- Select the **Contact** object in the list of **Service Objects** on the left side of the modal to begin configuring Contact fields
  - By clicking on the name of the field in the **Field Label** column modify the details of the following field:
    - Update the **RAVG Retail Membership Number** Field Label to RAVG Loyalty ID

Contact Fields (57)

	Required	Header Label	Field Label
1	<input checked="" type="checkbox"/>	Title	Title
2	<input checked="" type="checkbox"/>	System Modstamp	System Modstamp
3	<input checked="" type="checkbox"/>	Salutation	Salutation
4	<input type="checkbox"/>	Reports To ID	Reports To ID
5	<input checked="" type="checkbox"/>	RAVG Retail Membership Number	RAVG Loyalty ID
6	<input checked="" type="checkbox"/>	Photo URL	<input type="checkbox"/> Update 30 selected items
7	<input checked="" type="checkbox"/>	Business Phone	
8	<input type="checkbox"/>	Owner ID	

- Update the **RAVG\_Retail\_Membership\_Number\_c** Field API Name to RAVG\_Loyalty\_ID\_c

Contact Fields (59)

Required	Header Label	Field Label	Field API Name
	Title	Title	Title
	System Modstamp	System Modstamp	SystemModstamp
	Salutation	Salutation	Salutation
	Reports To ID	Reports To ID	ReportsTold
	RAVG Retail Membership Number	RAVG Loyalty ID	RAVG_Loyalty_ID_c
	Photo URL	Photo URL	PhotoURL

In order to later align the field that contains Birth Date from eCommerce source we need to adjust the data type for the **Birthdate** field of the Contact object. If we don't make this

change now, it would be quite challenging to apply that change later. Although you will notice that you are not able to modify system proposed data types. Therefore instead we need to configure new formula field.

- Add a new formula field to the Contact object:
  - Field Label = Birth Date
  - Formula Return Type = Date
  - Transformation Formula = sourceField['Birthdate']
    -  [Good article](#) to understand how date/time types work in Data Cloud

In order to later include data from this object into profile matching rules we need to configure few additional fields that will be mapped to the Party entity later. These attributes don't exist by default, but are required for Party Identification mapping in order for it to be used in the Identity Resolution.

- Add a new formula field:
  - Field Label = Identification Name
  - Formula Return Type = Text
  - Transformation Formula = "RAVG Retail Membership" (notice the double-quotes around the value, these need to be included)
- Add another formula field:
  - Field Label = Party Identification Type
  - Formula Return Type = Text
  - Transformation Formula = "Loyalty Program" (notice the double-quotes around the value)

As an example of data transformation on-load to provide simple calculations and enable encapsulation of the business logic right from the start. The formula attributes can be immediately used for segmentation use cases once the field is mapped into the data model. Take a note that formula syntax is case sensitive. And the way the formula is constructed in the example the values testing is also case sensitive.

- Select the **Case** object in the list of **Service Objects**

- Create new formula field by clicking the **New Formula Field** button in the top right area of the New Data Stream modal:
  - Label = Action Priority
  - Formula Return Type = Text
  - Using appropriate [functions & operators](#) calculate the value as follows:
    - Priority = **High** and Escalated = **true** > P1 - Escalated High
    - Priority = **High** and Escalated = **false** > P2 - High
    - Priority = **Medium** and Escalated = **true** > P3 - Escalated Medium
    - Priority = **Low** and Escalated = **true** > P4 - Escalated Low
    - Priority = **Medium** and Escalated = **false** > P5 - Medium
    - Priority = **Low** and Escalated = **false** > P6 - Low
  - Test your formula for following values (case sensitive):
    - true + High = P1 - Escalated High
    - true + Low = P4 - Escalated Low
    - false + Medium = P5 - Medium
    - false + High = P2 - High
  - If you can't make the working formula or getting unexpected results have a look at the [reference below](#).
  - Click **Save** to save your formula

If you made any mistakes you can edit a formula field by going to the Data Stream and clicking on 'Edit Formula' on the right hand side.

Data Stream  
**Case\_00DDo000001B0t8**

Stream Type	Data Stream Status	Last Run Status	Last Refreshed	Last Processed Records	Total Records
Ingest	Active	Success	7/29/2023, 1:58 AM	6	5

**Fields**    **Details**    **Activity History**

**Data Properties**

Object Category Engagement	Data Lake Object Name Case_00DDo000001B0t8
Object API Name Case_00DDo000001B0t8__dl	

**Fields (40)**

Header	Field Label	Field API	Data Type	Field Use	Formula	Status	Action
1	cdp_sys_Parti...	cdp_sys_Parti...	DateTime	✓			▼
2	Data Source O...	DataSourceO...	Text	✓			▼
3	Data Source	DataSource__c	Text	✓			▼
4	Action Priority	Action_Priority...	Text	✓			▼
5	AccountId	Account ID	AccountId__c	Text			Edit Formula
6	Case_Jurisdic...	Case Jurisdict...	Case_Jurisdic...	Text			Disable Field
7	CaseNumber	Case Number	CaseNumber_...	Text			▼

As you observe objects and fields selected for the setup notice that while you can choose fields that you need and exclude those that you don't, the objects that are included in the bundle are not selectable. You can however configure each object separately without using the bundle. For now we will proceed with deselecting some of the fields on Case and Contact objects

- For the **Case** object let's remove following fields that we won't be using in this workshop:
  - Company
  - Last Viewed Date
  - Last Referenced Date
  - Last Modified By ID
  - Contact Phone
  - Contact Fax
  - Internal Comments.

- Select the **Contact** Service Object and remove all non-required fields using checkbox next to **Required** header and then include following fields:
  - RAVG Retail Membership Number
  - Party Identification Type (earlier created formula field)
  - Mobile Phone
  - Identification Name (earlier created formula field)
  - External ID
  - Birth Date (earlier created formula field)
- Proceed to the next step, review the automatically assigned **Category** values for each of the data streams to be created, **Refresh Mode** and **Refresh Schedule** and deploy the bundle.

Notice that we did not specify **Data Space Filtering** options for any of the object, and they will be associated with **default** Data Space automatically.

\* Functional formula for the Action Priority field:

```
IF(sourceField['IsEscalated'] == "true"
, IF(
  sourceField['Priority'] == "High", "P1 - Escalated High", IF(
    sourceField['Priority'] == "Medium", "P3 - Escalated Medium", "P4 - Escalated Low"
  ))
, IF(
  sourceField['Priority'] == "High", "P2 - High", IF(
    sourceField['Priority'] == "Medium", "P5 - Medium", "P6 - Low"
  ))
)
```

## 1.2 Remove Account Data Stream

One of the implications from using data bundle is that it is not possible to choose which objects should be included or excluded from the deployed configuration. The implied benefit though is that bundle deploy not only configures data streams, but it also takes care of the data modelling step for the standard fields.

For the purpose of our solution the **Account** object is not included in the design, therefore we want to remove it. This will minimize unnecessary costs associated with processing of the records.

- Navigate to **Data Streams** tab, click on the action arrow on the right and choose **Delete Data Stream** option:

	Data Stream Name	Data Connector Type	Stream Type	Last Run Status	Data Stream Sta...	Last Processed ...	Total Records	Last Refreshed
1	Account_Home	Salesforce CRM	Ingest	None	Active			
2	Case_Home	Salesforce CRM	Ingest	None	Active			
3	Contact_Home	Salesforce CRM	Ingest	None	Active			

- Confirm the action in the dialog clicking on **Delete** button and you will get the error message:

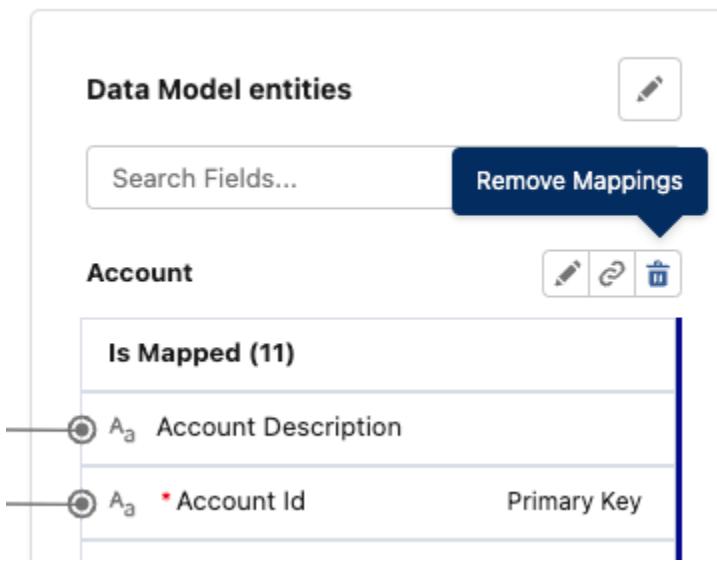
	Data Stream Name	Data Connector Type	Stream Type	Last Run Status	Data Stream Sta...	Last Processed ...	Total Records	Last Refreshed
1	Account_Home	Salesforce CRM	Ingest	None	Active			
2	Case_Home	Salesforce CRM	Ingest	None	Active			
3	Contact_Home	Salesforce CRM	Ingest	None	Active			

As the data stream is mapped into the data model we can't simply delete it without removing all those mappings first. Even though we will be going through the mapping process later in the workshop at this stage we will look into removal of the pre-configured mappings.

- Click on the **Account** data stream name
- Click on the **Review Mappings** link on the right

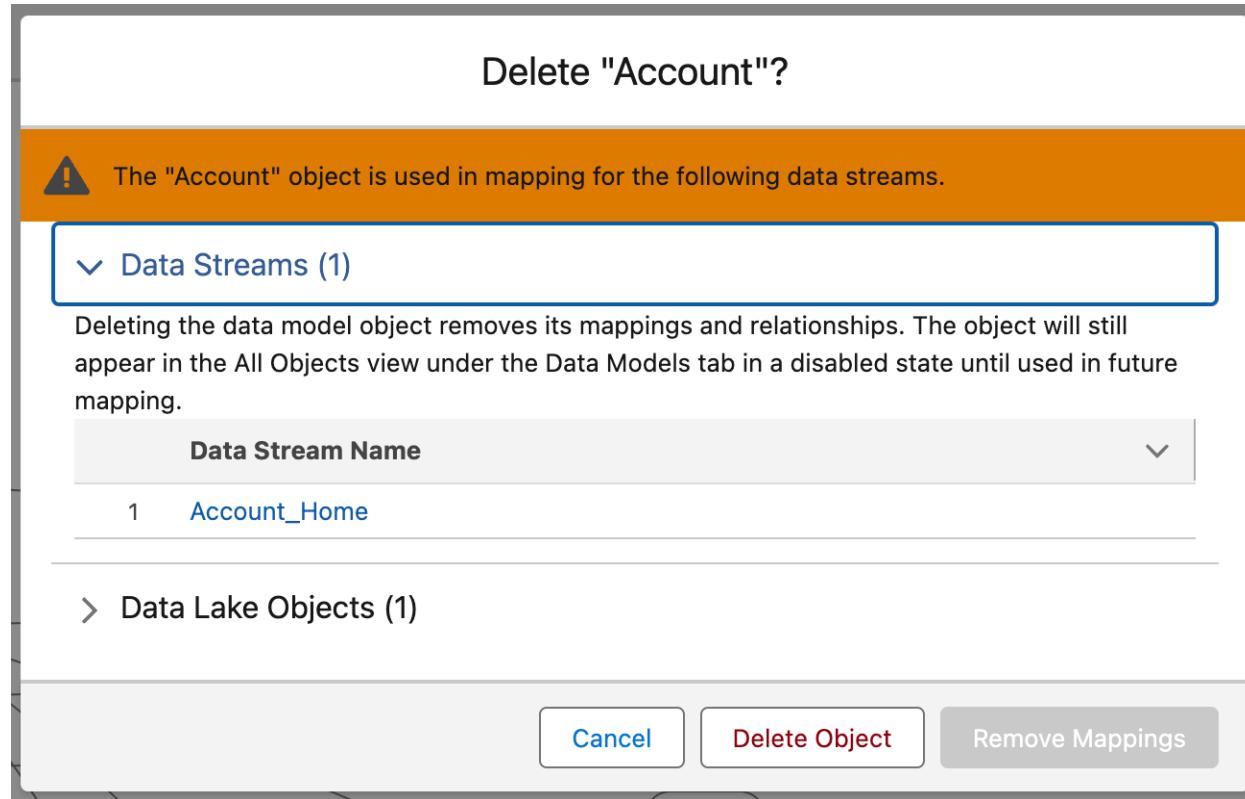
If you are unable to see 'Review Mappings' on a Data Stream, make sure you are accessing it from the 'Data Cloud' app.

- In the mapping canvas you'll notice that **Account** data stream is mapped to 3 data model objects: **Account**, **Contact Point Address** and **Contact Point Phone**. We'll remove the mappings one by one.
- Start with **Account**, click on the **Remove Mappings** (  ) icon on the right to its name:



The screenshot shows the 'Data Model entities' screen. At the top, there is a search bar labeled 'Search Fields...' and a blue button labeled 'Remove Mappings'. Below this, the 'Account' entity is listed with three icons: a pencil, a copy, and a trash can. A sub-section titled 'Is Mapped (11)' lists two fields: 'A\_a Account Description' and 'A\_a \* Account Id Primary Key'. The 'Primary Key' label is positioned to the right of the 'Account Id' field.

- Notice in the confirmation dialog the **Remove Mappings** button is disabled and you can only **Delete Object**:



This is expected as the **Account** data model object is not used in any other configuration, such as identity resolution or segmentation. And there is only single data stream that provides data for this model. Therefore removing of mapping will result in no data to be exposed as this model and instead the platform guides you to delete the object instead.

**i** Do take a note that deletion in this case (due to **Account** being a standard data model object) will not delete object definition itself, rather it will simply remove that object from any place where data model is used until another data stream is mapped to it. For any **custom** data model entities that you might create this action will actually delete the definition as well.

- Click **Delete Object** to proceed with mappings removal.
- Now proceed with the similar steps for the **Contact Point Address**
- Notice that in this case both actions **Delete Object** and **Remove Mappings** are enabled:

New Data Stream

*i* View All permissions are required for any standard or custom objects. [Tell Me More](#)

Select a Salesforce org and a Data Bundle or Object to create a new data stream.

\* Salesforce Org  
Salesforce - PES      [Data Bundles](#)      [All Objects](#)

**Standard Data Bundles (3)**

- Digital Engagement | 3 Objects · Created by Salesforce
- Sales | 3 Objects · Created by Salesforce
- Service | 3 Objects · Created by Salesforce

**Custom Data Bundles (0)**

[Previous](#)      [Next](#)

### Delete "Contact Point Address"?

**⚠** The "Contact Point Address" object is used in mapping for the following data streams.

**▼ Data Streams (2)**

Deleting the data model object removes its mappings and relationships. The object will still appear in the All Objects view under the Data Models tab in a disabled state until used in future mapping.

	Data Stream Name
1	Contact_Home
2	Account_Home

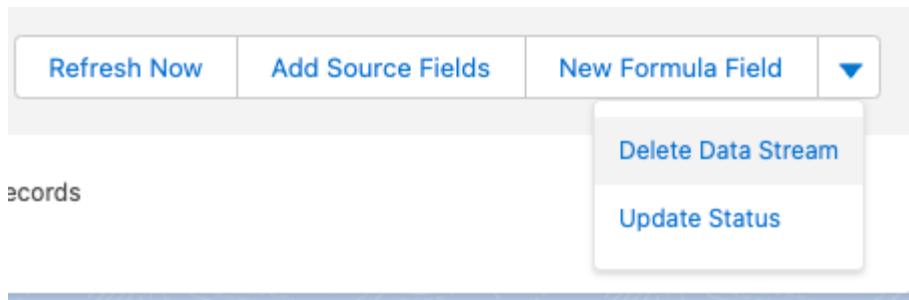
**➤ Data Lake Objects (2)**

Would you like to delete the data model object or remove only the current mapping?

[Cancel](#)      [Delete Object](#)      [Remove Mappings](#)

- As in this case the data model object is used in more than a single mapping the system provides a choice to remove only current (**Account**) data stream mapping or for all streams.

- We want to preserve mapping for **Contact** data stream, therefore choose **Remove Mappings** as an action.
- Lastly repeat the steps for the **Contact Point Phone** object with **Remove Mappings** action and not deleting the object.
- Once all of the mappings removed close the canvas.
- On the Details screen of the **Account** data stream click on the action menu (top right) and choose **Delete Data Stream** action:



- 
- Confirm the deletion in the dialog to proceed with deletion.
- The details screen will close and you'll notice that the status of the data stream changed to **Deleting**
- At this stage since there was no data synchronized the process should complete reasonably quick. Try refreshing the list view after a minute or so to confirm that the stream was deleted.

**!** In your implementations the process of deleting the data stream might take noticeable time. There are few factors involved in the processing speed, one of which is data volume. So keep that in mind and just patiently wait until completion if you need to reconfigure the stream for example.

If you experience any issues deleting the Account stream, leave it as-is and move forward. As long as the object is not mapped into the data model it will not interfere with any further activities.

### 1.3 Booking Custom Object

Records in the Booking object represent vehicle booking information made by prospects and customers. As this is a custom object by default access to it from the Data Cloud will

not be enabled. Comparing to the **Case** object earlier where you've experienced the error when trying to deploy the bundle due to the field not being available in this case you would not even see the object.

- Navigate to **Data Streams** tab and create new data stream
- Choose **Salesforce CRM** source and instead of bundle selection choose **All Objects** option

New Data Stream

---

**i** View All permissions are required for any standard or custom objects.[Tell Me More](#)

Select a Salesforce org and a Data Bundle or Object to create a new data stream.

\* Salesforce Org

Salesforce - Home Org

Data Bundles

All Objects

Salesforce Objects (151)

- Try searching for **Booking** object and confirm it does not appear in the list
- Using instructions on [this help page](#) enable **Read** and **View All** permissions for **Bookings, Rental Preferences** and **Rentals** objects as well as enable **Read Access** to all fields for these objects.

**💡 Usability tip** - when you need to enable **Read Access** for a large number of fields depending on your preference you can use mouse clicks or navigate with the keyboard once you click on the first checkbox using **Tab/Shift+Tab** and **Space** to change the selection.

- Restart configuration of the new data stream from **Salesforce CRM** source and under **All Objects** this time locate **Booking** and proceed to next step.
- Review fields and deselect those that we won't need in our solution. Note that unlike bundle objects you can remove any field from the Booking fields except **Record ID** that is already identified as a Primary Key and **System Modstamp**.
  - Remove Last Modified By ID and Last Referenced Date fields

The significance of inclusion or exclusion of the fields from the object might not appear as high in our scenario. In practice when you implement customer solutions the objects quite

often get very “wide” with large number of fields. Minimizing number of ingested fields simplifies the need to revisit each and every field during ingestion and mapping configurations.

- Add a new formula field to encapsulate some of the business logic from two into a single field:
  - Label = Purpose
  - Formula Return Type = Text
  - Using appropriate [functions & operators](#) calculate the value as follows:
    - If only Is Leisure attribute value equals true use Personal
    - If only Is Business attribute value equals true use Business
    - If both Is Leisure and Is Business attribute values equal true use Mixed
    - If both Is Leisure and Is Business attribute values equal false use Unspecified value
  - Test your formula for following values:
    - Is Leisure = true, Is Business = true = Mixed
    - Is Leisure = false, Is Business = true = Business
    - Is Leisure = false, Is Business = false = Unspecified
  - If you can't make the working formula or getting unexpected results have a look at the [reference below](#).
- As booking represents the event choose **Category** = Engagement

**⚠** Notice that for **Engagement** category you are required to choose the field that represents the Event Time. It might not always be straight forward when it comes to choosing the “right” field, especially if you have more than one date. As you’ve learnt in the lesson prior to this hands on exercise the **Engagement** category requires immutable field, meaning that value of the event time should not change over lifetime of the record.

- When in focus you can see there are quite a few fields available for selection as **Event Time Field**:
  -
- Choose Booking Date field.

- Proceed to the next step.
- Notice that unlike with bundle you can update the name of the stream prior to deploying it. Leave it without changes for consistency, so that all CRM streams will have org ID appended to it and deploy the stream.
- Also leave selection of the **default** Data Space.
- Click “Deploy” to deploy the Data Stream

\* Functional formula for the Purpose field:

```
IF(
    AND(sourceField['Is_Leisure__c'] == "true", sourceField['Is_Business__c'] == "true")
    , "Mixed"
    , IF(
        AND(sourceField['Is_Leisure__c'] == "true", sourceField['Is_Business__c'] == "false")
        , "Personal"
        , IF (
            AND(sourceField['Is_Leisure__c'] == "false", sourceField['Is_Business__c'] == "true")
            , "Business"
            , "Unspecified"
        )
    )
)
```

**⚠ Note:** the data stream for the **Booking** object can fail to refresh on the first run. This will not prevent the data from flowing through later and on the subsequent runs the data will be populated. So if you experience that, please don't get alarmed and simply wait for an hour until the stream refreshes successfully. ⚠

## 1.4 Rental Custom Object

Records in the Rental object represent vehicle rental information made by customers. As you've already enabled access to this object you can proceed right to the configuration of the new data stream.

- Navigate to **Data Streams** tab and create new data stream
- Choose **Salesforce CRM** source and locate **Rental** and proceed to the next step
- Remove Last Modified By ID and Last Referenced Date fields
- Add a new formula field to encapsulate some of the business logic similar to Booking:
  - Field Label = Purpose
  - Formula Return Type = Text
  - Using appropriate [functions & operators](#) calculate the value as follows:
    - If only Is Leisure attribute value equals true use Personal
    - If only Is Business attribute value equals true use Business
    - If both Is Leisure and Is Business attribute values equal true use Mixed
    - If both Is Leisure and Is Business attribute values equal false use Unspecified value
  - Test your formula for following values:
    - Is Leisure = true, Is Business = true = Mixed
    - Is Leisure = false, Is Business = true = Business
    - Is Leisure = false, Is Business = false = Unspecified
  - If you can't make the working formula or getting unexpected results have a look at the [reference above](#).

**i** For the following formula the source fields already have appropriate data type as they have been defined accordingly in the source system. In practice ensure to always review the data type of the fields you use for calculations or use appropriate function to convert original value to the needed target type.

- Add new formula field to categorize family rentals:

- Field Label = Combined Children Seat Count
- Formula Return Type = Number
- Using appropriate [functions & operators](#) calculate the value as the sum of Baby Seat Count and Booster Seat Count attribute values.
- Test and validate your formula with empty and few combinations of values.
- If you can't make the working formula or getting unexpected results have a look at the [reference below](#).
- Set following properties:
  - Category = Engagement
  - Event Time Field = Created Date
- Proceed to the next step and deploy the stream.

\* Functional formula for the Combined Children Seat Count field:

```
NUMBER(sourceField['Baby_Seat_Count__c']) +
NUMBER(sourceField['Booster_Seat_Count__c'])
```

## 1.5 Rental Preference Custom Object

Records in the Rental Preference object represent customer personal preferences for the vehicles they rent. These form an expedited search for each customer profile when they are making new reservation.

- Navigate to **Data Streams** tab and create new data stream
- Choose **Salesforce CRM** source and locate **Rental Preference** and proceed to the next step
- Set Category = Other

At this stage of configuration it is important to review how the system parsed the inbound data file and assigned a **data type** to each of the identified fields. The main reason for that is later in the configuration when you'll need to map these fields into the data model **any type mismatch will prevent you from doing so**, resulting in the need for you to go back to data stream setup and redo it.

- Review and confirm that both **Leisure Preferences Last Updated Date** and **Business Preferences Last Updated Date** have been identified as **Date (yyyy-MM-dd)** data types:

Rental Preference Fields (33)					<input type="button" value="Search..."/>	<input type="button" value="New Form"/>
	<input checked="" type="checkbox"/> Header Label	Field Label	Field API Name ↑	Formul...	Data Type	
1	<input checked="" type="checkbox"/> System Modstamp	System Modstamp	SystemModstamp		DateTime (yyyy-MM-dd'T'	
2	<input checked="" type="checkbox"/> Owner ID	Owner ID	OwnerId		Text	
3	<input checked="" type="checkbox"/> Rental Preference ID	Rental Preference ID	Name		Text	
4	<input checked="" type="checkbox"/> Leisure - Vehicle Type	Leisure - Vehicle Type	Leisure_Vehicle_Type_c		Text	
5	<input checked="" type="checkbox"/> Leisure - Protection Coverage	Leisure - Protection Coverage	Leisure_Protection_Coverage_c		Text	
6	<input checked="" type="checkbox"/> Leisure Preferences Last Updated Date	Leisure Preferences Last Updated Date	Leisure_Preferences_Last_Updated_Dat...		Date (yyyy-MM-dd)	

- For our data model these fields need to be of **Date** type. The CRM connector does not allow modification of the data type, therefore if these need to be corrected, you'll need to create formula fields with required data type and use those in data modelling exercise.
- If your fields have been correctly identified - skip the following steps and don't create custom fields. Move to [this step](#) instead.

- Only if in your environment these fields have been misidentified with **DateTime (yyyy-MM-dd)** perform the following steps:
  - Update field label for the following fields (note these should be under 40 characters):
    - Leisure Preferences Last Updated Date = Leisure Pref Last Updated DateTime
    - Business Preferences Last Updated Date = Business Pref Last Updated DateTime
  - Add new formula field:
    - Field Label = Leisure Preferences Last Updated Date
    - Formula Return Type = Date
    - Formula =
 

```
sourceField['Leisure_Preferences_Last_Updated_Date__c']
```
  - Add new formula field:
    - Field Label = Business Preferences Last Updated Date
    - Formula Return Type = Date

- Formula =
 

```
sourceField['Business_Preferences_Last_Updated_Date__c']
```
- Update field label for the following fields (note these should be under 40 characters):
  - Leisure - Protection Coverage = Leisure - Protection Coverage - Source
  - Business - Protection Coverage = Business - Protection Coverage - Source
- Add new formula field:
  - Field Label = Leisure - Protection Coverage
  - Formula Return Type = Text
  - Using appropriate [functions & operators](#) calculate the value as follows:
    - If Leisure - Protection Coverage attribute supplied - use its value
    - Otherwise use NONE value
  - Test and validate your formula with empty and any values.
  - If you can't make the working formula or getting unexpected results have a look at the [reference below](#).
- Add another formula field called Business - Protection Coverage using the same approach and logic as for the Leisure - Protection Coverage field.
- Proceed to the next step and deploy the stream.

\* Functional formula for the formula fields

#### Leisure - Protection Coverage

```
IF(ISEMPTY(sourceField['Leisure_Protection_Coverage__c']), "NONE",
sourceField['Leisure_Protection_Coverage__c'])
```

#### Business - Protection Coverage

```
IF(ISEMPTY(sourceField['Business_Protection_Coverage__c']), "NONE",
sourceField['Business_Protection_Coverage__c'])
```

## 2. Applying Data Transforms

As you went through the configuration and review of the data streams you might have noticed that **Rental** stream in particular contains mixed data. The majority of attributes are relevant to the engagement (rental transaction), such as vehicle details, insurance coverage, start and end date and so forth. But there are a few attributes that are relevant for the customer profile, such as details of their driver license and various membership program numbers.

When the data stream is mapped to the data model it is not possible to map a given data stream to the mixed categories data model objects. Therefore we will take advantage of [Data Transform functionality](#), extracting various identifiers such as membership and driver license numbers and storing them in a separate data lake object.

 Comparing Data Transforms with formula fields you'll notice that former operates within the stream context, meaning you can access all records within the data stream. While formulas as you now know only operate in the context of a single record / row of data.

 This guide walks you through the configuration of the Streaming Data Transform. If you are interested to explore Batch Data Transform as alternative, while achieving the same outcome you can follow [Ex: Data Cloud - Configure Batch Data Transform](#) guide instead of step 2.2 below.

### 2.1 Create New Data Lake Object

Before configuring the Data Transform it is necessary to create a destination data lake object where resulting records will be stored. Similar to data streams this choice affects billing as well as what objects in the data model can be mapped to.

- Navigate to **Data Lake Objects** tab and create new object choosing **New** option in method selection
- Set following properties:
  - Data Lake Object Name = Party Identification Collection
  - Category = Profile

- Notice 4 default fields that were created for you. These enable maintenance of data lineage as well as other system usage

Field Label	Field API Name	Data Type	Primary Key	Record Modified Field
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"/> 
cdp_sys_SourceVersion	cdp_sys_SourceVersion	Text	<input type="checkbox"/>	<input type="checkbox"/> 

[+ Add Field](#)

- Add following new fields:

Field Label	Data Type	Primary Key
-------------	-----------	-------------

<b>Party Identification ID</b>	Text	1
Party	Text	0
Party Identification Type	Text	0
Identification Name	Text	0
Identification Number	Text	0

### New Data Lake Object

Add or remove fields, then select the primary key for your new Data Lake Object.

Field Label	Field API Name	Data Type	Primary Key	Record Modified Field
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"/>
Party Identification ID	Party_Identification_ID	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Party	Party	Text	<input type="checkbox"/>	<input type="checkbox"/>
Party Identification Type	Party_Identification_Type	Text	<input type="checkbox"/>	<input type="checkbox"/>
Identification Name	Identification_Name	Text	<input type="checkbox"/>	<input type="checkbox"/>
Identification Number	Identification_Number	Text	<input type="checkbox"/>	<input type="checkbox"/>

+ Add Field

○

- Save the object.

#### 2.1.1 Add the newly created Data Lake Object to your default Data Space

During configuration of the data streams you've noticed that all objects have been associated with **default Data Space**. While for the newly created Data Lake Object this is not the case. This will prevent it from being available in subsequent exercises. Therefore let's address it now.

- Go to the **Data Spaces** tab
- Select the **default** Data Space

The screenshot shows a table titled "Data Spaces" with a header row containing columns for "Data Space Name", "Data Space API N...", "Status", "Description", "Last Modified Date", and "Last Modified By". There is one item listed:

Data Space Name	Data Space API N...	Status	Description	Last Modified Date	Last Modified By
1 default	default	ACTIVE	Default data space where all the current DLOs are made members	7/7/2023, 12:06 AM	autproc

- Click on **Add Data**
- Select the Party Identification Collection Data Lake Object and click **Next**

Add Data

Select data below to include it in this data space

Data Lake Object	Object Name	Object API Name
	<input checked="" type="checkbox"/> Party Identification Collection	Party_Identification_Collection_dll

- Select  **Add objects without filter**

Add Data

Review your data selected and set filters, if needed.

Selected Objects (1)	Object Name	Object Type	Filters
	Party Identification Collection	Data Lake Object	<input type="button" value="Set Filters"/>

You are about to add 1 object(s) to the data space "default" with no filters applied.  
Anyone with access to this space will see the entire object. Click the acknowledgement or add filters to proceed.

Add objects without filters

- Save the configuration

## 2.2 Create New Streaming Data Transform

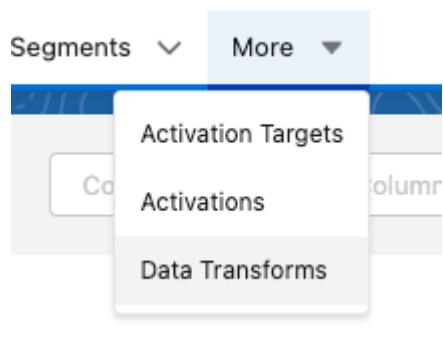
If you want to explore Batch Data Transform as an alternative approach with the same outcome please follow [Ex: Data Cloud - Configure Batch Data Transform](#) guide instead of current section of the guide. If you want to configure both types of data transforms please follow additional instructions at the start of the batch data transform configuration guide.



---

As we now have a destination object to keep records we can configure new data transform.

- Navigate to **Data Transforms** tab and create new data transform.
  - **Note:** You may need to click the **More** in the navigation bar in order to view this tab:

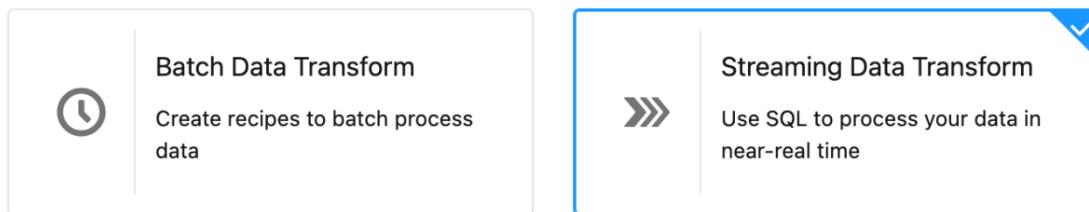


- Select Streaming Data Transform

### New Data Transform

---

Choose type of data transform to start



- Set following properties:
  - Data Transform Label = Extract Party Identifiers from Rentals
  - Target Data Lake Object = Party Identification Collection
  - Description = Extract driver license and loyalty program membership numbers from car rentals
  - Click **Next** to proceed to the next screen

**Details**

* Data Transform Label Extract Party Identifiers from Rentals	* Data Transform API Name Extract_Party_Identifiers_from_Rentals
* Target Data Lake Object Party Identification Collection	
<small> ⓘ Make sure the target data lake object exists. If not, navigate to Data Lake Objects to create one first.</small>	
Description Extract driver license and loyalty program membership numbers from car rentals	

- Exploring examples (click on the **Examples** button above the query editor) and/or [functions reference](#) create a SQL expression that extracts below listed identifiers following these rules:
  - <Customer> field is always selected as Party target field
  - <Customer>\_<Membership Number> is selected as Party\_Identification\_ID\_\_c target field
  - For **Airline Membership Number**
    - "Loyalty Program" text is selected as Party\_Identification\_Type\_\_c target field
    - "Airline Membership" text is selected as Identification\_Name\_\_c target field
    - <Airline Membership Number> field is selected as Identification\_Number\_\_c target field
  - For **Driver License Number**
    - "Person Identification" text is selected as Party\_Identification\_Type\_\_c target field
    - "Driver License" text is selected as Identification\_Name\_\_c target field
    - <Driver License Number> field is selected as Identification\_Number\_\_c target field
  - For **Motor Club Number**
    - "Loyalty Program" text is selected as Party\_Identification\_Type\_\_c target field
    - "Motor Club Membership" text is selected as Identification\_Name\_\_c target field

- <Motor Club Number> field is selected as Identification\_Number\_\_c target field
- For **RAVG Retail Membership Number**
  - "Loyalty Program" text is selected as Party\_Identification\_Type\_\_c target field
  - "RAVG Retail Membership" text is selected as Identification\_Name\_\_c target field
  - <RAVG Retail Membership Number> field is selected as Identification\_Number\_\_c target field
- Always add a criteria to only choose records that contain value (not null or empty) in respective identification number field.
- Ensure to use **Check Syntax** button to validate your expression and save it.
- If you find it difficult writing a query have a look at the **reference below**.

#### \* Query Expression for the Extract Party Identifiers from Rentals

Ensure to replace <ORG\_ALIAS> (including < and > characters) with the value of your org ID from respective DLO object names:

```

SELECT Rental_c_<ORG_ALIAS>_dll.Customer_c__c as Party__c
      , Concat(Rental_c_<ORG_ALIAS>_dll.Customer_c__c, "_",
Rental_c_<ORG_ALIAS>_dll.Airline_Membership_Number_c__c) as
Party_Identification_ID__c
      , "Loyalty Program" as Party_Identification_Type__c
      , "Airline Membership" as Identification_Name__c
      , Rental_c_<ORG_ALIAS>_dll.Airline_Membership_Number_c__c as
Identification_Number__c
FROM Rental_c_<ORG_ALIAS>_dll
WHERE Rental_c_<ORG_ALIAS>_dll.Airline_Membership_Number_c__c <> ""
AND Rental_c_<ORG_ALIAS>_dll.Airline_Membership_Number_c__c <> ""

```

UNION

```
SELECT Rental_c_<ORG_ALIAS>_dll.Customer_c__c as Party__c
, Concat(Rental_c_<ORG_ALIAS>_dll.Customer_c__c, "_",
Rental_c_<ORG_ALIAS>_dll.Driver_License_Number_c__c ) as Party_Identification_ID__c
, "Person Identification" as Party_Identification_Type__c
, "Driver License" as Identification_Name__c
, Rental_c_<ORG_ALIAS>_dll.Driver_License_Number_c__c as Identification_Number__c
FROM Rental_c_<ORG_ALIAS>_dll
WHERE Rental_c_<ORG_ALIAS>_dll.Driver_License_Number_c__c <> ""
AND Rental_c_<ORG_ALIAS>_dll.Driver_License_Number_c__c <> ""
```

UNION

```
SELECT Rental_c_<ORG_ALIAS>_dll.Customer_c__c as Party__c
, Concat(Rental_c_<ORG_ALIAS>_dll.Customer_c__c, "_",
Rental_c_<ORG_ALIAS>_dll.Motor_Club_Number_c__c) as Party_Identification_ID__c
, "Loyalty Program" as Party_Identification_Type__c
, "Motor Club Membership" as Identification_Name__c
, Rental_c_<ORG_ALIAS>_dll.Motor_Club_Number_c__c as Identification_Number__c
FROM Rental_c_<ORG_ALIAS>_dll
WHERE Rental_c_<ORG_ALIAS>_dll.Motor_Club_Number_c__c <> ""
AND Rental_c_<ORG_ALIAS>_dll.Motor_Club_Number_c__c <> ""
```

UNION

```
SELECT Rental_c_<ORG_ALIAS>_dll.Customer_c__c as Party__c
```

```

, Concat(Rental_c_<ORG_ALIAS>_dll.Customer_c__c, "_",
Rental_c_<ORG_ALIAS>_dll.RAVG_Retail_Membership_Number_c__c) as
Party_Identification_ID__c

, "Loyalty Program" as Party_Identification_Type__c

, "RAVG Retail Membership" as Identification_Name__c

, Rental_c_<ORG_ALIAS>_dll.RAVG_Retail_Membership_Number_c__c as
Identification_Number__c

FROM Rental_c_<ORG_ALIAS>_dll

WHERE Rental_c_<ORG_ALIAS>_dll.RAVG_Retail_Membership_Number_c__c <> ""
AND Rental_c_<ORG_ALIAS>_dll.RAVG_Retail_Membership_Number_c__c <> ""

As an example here is the part of the functional query for the reference:

```

**New Data Transform**

---

\* Expression

[Examples](#)

```

SELECT Rental_c_Home__dll.Customer_c__c as Party__c
, Concat(Rental_c_Home__dll.Customer_c__c, "_", Rental_c_Home__dll.Airline_Membership_Number_c__c) as
Party_Identification_ID__c
, "Loyalty Program" as Party_Identification_Type__c
, "Airline Membership" as Identification_Name__c
, Rental_c_Home__dll.Airline_Membership_Number_c__c as Identification_Number__c
FROM Rental_c_Home__dll
WHERE Rental_c_Home__dll.Airline_Membership_Number_c__c <> ""
AND Rental_c_Home__dll.Airline_Membership_Number_c__c <> ""

UNION

SELECT Rental_c_Home__dll.Customer_c__c as Party__c
, Concat(Rental_c_Home__dll.Customer_c__c, "_", Rental_c_Home__dll.Driver_License_Number_c__c) as
Party_Identification_ID__c
, "Person Identification" as Party_Identification_Type__c
, "Driver License" as Identification_Name__c

```

Valid Syntax

[Check Syntax](#)

Once the Data Transform is deployed it will continuously stream data from the source DLO.

**i** Few things to take a note of with regards to Data Transform:

- There is a limited number (3) of Data Transforms that are included with the org, additional transforms can be purchased, with [total limit of 25](#).
- Data Transform can use only one source DLO, e.g. in our case we would not be able to combine same identifiers from the **Booking** DLO. Additional streaming data transform can be created that uses the same target DLO.
- Within a data transform with **UNION** statement there can be only up to 5 **SELECT** queries. For additional data points new data transform need to be created.

Take these into account when planning and designing your solutions for real customers.

### 3. Commerce Platform Data

**Note:** you will need your S3 details - bucket name, access and secret keys to configure data streams listed in this section.

#### 3.1 Configure S3 Connector

In order for us to configure ingestion from the S3 bucket it is necessary to configure it as a data source first.

- Navigate to **Data Cloud Setup > Connectors** and start configuration of a new connector
- Choose **Amazon S3** in the list of sources and proceed to the next step:
  -
- Set following details:
  - Connection Name = RAVG Retail eCommerce S3
  - Authentication Option = Access Key/Secret Based
  - Specify your AWS access key, AWS secret access key and Bucket Name
  - Parent Directory = ecommerce-data/
- Click **Test Connection** and verify that connection was established successfully:

New Amazon S3 Source

* Connection Name RAVG Retail eCommerce S3	* Connection API Name RAVG_Retail_eCommerce_S3
<b>Authentication Details</b>	
* AWS access key .....	* AWS secret access key .....
<b>Connection Details</b>	
* Bucket Name ravg-spring24	* Parent Directory ⓘ ecommerce-data/
<b>Test Connection</b>	✓ Connection was established.
<b>Back</b>	<b>Save</b>

- Save the configuration.

The connection is now ready to be used in new data streams configurations:

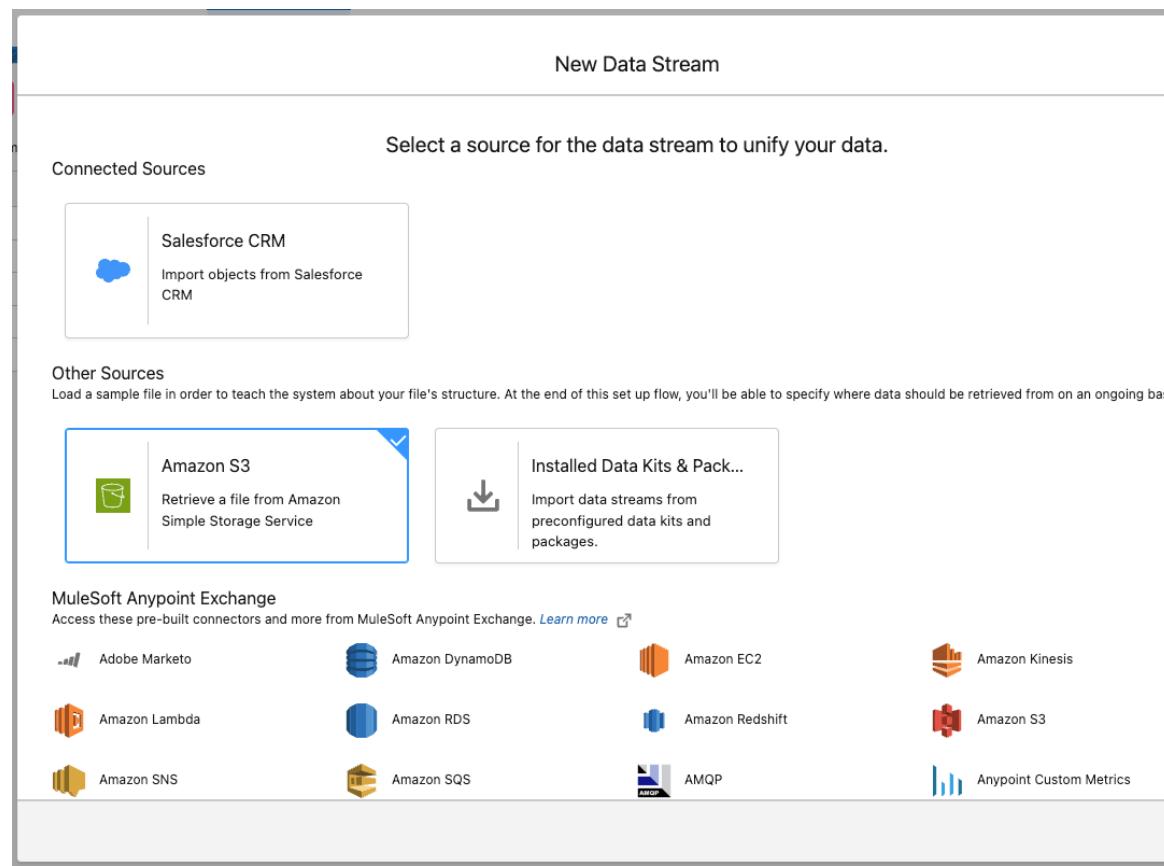
Setup		More Connectors											
		New											
1 Connections													
<table border="1"> <thead> <tr> <th>Connection Name</th> <th>Connection Type</th> <th>Connector Method</th> <th>Status</th> <th>Last Modified</th> </tr> </thead> <tbody> <tr> <td>RAVG Retail eCommerce S3</td> <td>Amazon S3</td> <td>Source</td> <td>Active</td> <td>Mar 28, 2024, 10:33 AM</td> </tr> </tbody> </table>				Connection Name	Connection Type	Connector Method	Status	Last Modified	RAVG Retail eCommerce S3	Amazon S3	Source	Active	Mar 28, 2024, 10:33 AM
Connection Name	Connection Type	Connector Method	Status	Last Modified									
RAVG Retail eCommerce S3	Amazon S3	Source	Active	Mar 28, 2024, 10:33 AM									

**i** As you go through the configuration steps for eCommerce feeds notice that Source value is the same in all configurations. The feeds are sourced from the same system and that notion is carried forward into the platform. And while we not actively using it in this workshop, this creates additional data point that can be referenced when data is used in segmentation or insights.

### 3.2 Customer Profile Feed

Customer profile feed contains profile and demographics attributes of the customers similar to the Contact object in CRM that you configured earlier.

- Following instructions from [help page](#)
  - Configure **Amazon S3** data stream (notice new tile now appearing on the wizard dialog):



- Set following properties:
  - Connection = RAVG Retail eCommerce S3
  - File Type = CSV
  - Import from Directory = leave empty
  - File Name = S3 Customer Profile.csv
  - Source = keep default value

New Data Stream

Select your Amazon S3 connection and choose your object(s).

\* Connection  
RAVG Retail eCommerce S3

Bucket Name  
ravg-spring24

Parent Directory  
ecommerce-data/

**File Details**

\* File Type  
 CSV  
 Parquet

Import from Directory ⓘ

\* File Name ⓘ  
S3 Customer Profile.csv

**Source Details**

\* Source ⓘ  
AwsS3\_RAVG\_Retail\_eCommerce\_S3

Previous            Next

- Proceed to the next step
  - Data Lake Object = + New Data Lake Object
  - Data Lake Object Label = eComm Customer Profile
  - Category = Profile
  - Primary Key = Customer Ref Number

Following formula fields will be mapped into the Party entity to support profile matching rules.

- Add a new formula field:
  - Label = Party Identification Id
  - Formula Return Type = Text
  - Using appropriate [function](#) calculate the value as follows:
    - LoyaltyProgram\_RAVGRetailMembership\_<Customer Ref Number>
    - If you can't make the working formula or getting unexpected results have a look at the [reference below](#).
- Save & Add another formula field:
  - Label = Party Identification Type

- Formula Return Type = Text
- Transformation Formula = "Loyalty Program" (notice the double-quotes around the value)
- Save & Add another formula field:
  - Label = Identification Name
  - Formula Return Type = Text
  - Transformation Formula = "RAVG Retail Membership" (notice the double-quotes around the value)

Following formula field will be used as a unique identifier for the email address mapping.

- Save & Add another formula field:
  - Label = Contact Point Email Id
  - Formula Return Type = Text
  - Using appropriate [function](#) calculate the value as follows:
    - <Customer Ref Number>\_ContactPointEmail\_<Email>
    - If you can't make the working formula or getting unexpected results have a look at the [reference below](#).

At this stage of configuration it is important to review how the system parsed the inbound data file and assigned a **data type** to each of the identified fields. In this file the platform result is actually quite accurate with the exception of one field.

- Update data type for **Email** field to Email:

S3 Customer Profile.csv (9)					<a href="#">New Formula Field</a>
<a href="#">Supported Fields (6)</a>		Unsupported Fields (0)	Lineage Fields (3)	Formula Fields (0)	
Source Header	Field Label	Field API Name	Data Type		
1 First Name	First Name	First_Name	Text	Text	
2 Surname	Surname	Surname	Text	Number	
3 Customer Ref Number	Customer Ref Numb...	Customer_Ref_Num...	Text	DateTime	
4 Rewards Program ID	Rewards Program ID	Rewards_Program_ID	Text	Date	
5 Birth Date	Birth Date	Birth_Date	Date (yyyy-MM-dd)	Email	
6 Email	Email	Email	Text	Phone	

- Proceed to next step
- Keep **default** Data Space selected and Refresh Mode = Upsert
- Keep Schedule with the following settings:
  - Frequency = None
  - Ensure to  Refresh Data Stream immediately option
- Deploy the data stream.

\* Functional formula for the Party Identification Id field:

```
CONCAT("LoyaltyProgram_RAVGRetailMembership_", sourceField['Customer Ref Number'])
```

\* Functional formula for the Contact Point Email Id field:

```
CONCAT(sourceField['Customer Ref Number'], "_ContactPointEmail_", sourceField['Email'])
```

### 3.3 Order Feed

Order data feed contains details of the customer transactions, such as transaction date and total amount.

- Following instructions from [help page](#)

- Configure **Amazon S3** data stream
- Set following properties:
  - Connection = RAVG Retail eCommerce S3
  - File Type = CSV
  - Import from Directory = leave empty
  - File Name = S3 Order Headers.csv
  - Source = keep default value
- Proceed to the next step
  - Data Lake Object = + New Data Lake Object
  - Data Lake Object Label = eComm Order
  - Category = Engagement
  - Event Time Field = Order Date
  - Primary Key = Order Number

Data lineage provides an opportunity to narrow result set during segmentation, especially for the cases when a single Data Cloud instance includes data feeds for more than one geography or sub-brand. It won't be enforced, rather an additional element to control what data should be accounted for when targeting a certain segment.

- In order to associate the [reference of this dataset to a specific ecommerce realm/instance](#) create new formula field that will contain pre-defined text:
  - Field Label = Internal Business Unit
  - Formula Return Type = Text
  - Transformation Formula = "eComm\_RAVG\_Retail" (notice the double-quotes around the value)
- Set Organization Unit Identifier = Internal Business Unit

At this stage of configuration it is important to review how the system parsed the inbound data file and assigned a **data type** to each of the identified fields. In this file the platform result is actually quite accurate, but we still want to make one change.

- Depending on your preference you might horizontally scroll to the right in the fields pane or resize each column to make **Data Type** column visible:

S3 Order Headers.csv (12)					<a href="#">New Formula Field</a>
Supported Fields (8)		Unsupported Fields (0)	Lineage Fields (3)	Formula Fields (1)	
Source Header	Field Label	Field API Name	Data Type		
1 Customer Number	Customer Number	Customer_Number	Text		
2 Order Number	Order Number	Order_Number	Text		
3 Delivery Method	Delivery Method	Delivery_Method	Text		
4 Store Number	Store Number	Store_Number	Number		
5 Order Date	Order Date	Order_Date	DateTime (yyyy-MM-dd ...		
6 Status	Status	Status	Text		
7 Total Amount	Total Amount	Total_Amount	Number		
8 Loyalty Points Accrued	Loyalty Points Accru...	Loyalty_Points_Accr...	Number		

- One of the key fields for our design is **Customer Number**. We are planning to use this as a customer identifier later on therefore ensure it to be a Text data type (which it is).
- Proceed to next step
- Keep **default** Data Space and Refresh Mode = Upsert
- Keep Schedule with the following settings:
  - Frequency = None
  - Ensure to  Refresh Data Stream immediately option
- Deploy the data stream.

### 3.4 Store Lookup Data

Order data feed includes details for the store at which a purchase occurred. One way to extract these would be to use Data Transforms. Another approach that we will follow this time is to re-ingest the same source file but using different settings for the data stream. With CRM this method would not work, as the connector only supports a single data stream to be configured for any given object.

Configure additional data stream to ingest unique set of store details that later will be mapped to the Sales Store entity.

- Following instructions from [help page](#)
  - Configure **Amazon S3** data stream
  - Set following properties:
    - Connection = RAVG Retail eCommerce S3
    - File Type = CSV
    - Import from Directory = leave empty
    - File Name = S3 Order Headers.csv
    - Source = keep default value
  - Proceed to the next step
    - Data Lake Object = + New Data Lake Object
    - Data Lake Object Label = eComm Store Number Lookup
    - Category = Other
    - Primary Key = Store Number

With this data stream we will be mapping a limited number of fields into the data model and data type assigned by the platform will suit our needs. Hence there is no need to review and make any changes for this data stream.

- Proceed to next step
- Keep **default** Data Space and Refresh Mode = Upsert
- Keep Schedule with the following settings:
  - Frequency = None
  - Ensure to  Refresh Data Stream immediately option
- Deploy the data stream.

### 3.5 Line Items Feed

Order Line Items data feed contains details of each item in the customer transactions, such as product name, quantity and cost.

- Following instructions from [help page](#)
  - Configure **Amazon S3** data stream
  - Set following properties:
    - Connection = RAVG Retail eCommerce S3
    - File Type = CSV
    - Import from Directory = leave empty
    - File Name = S3 Order Line Items.csv
    - Source = keep default value
  - Proceed to the next step
    - Data Lake Object = + New Data Lake Object
    - Data Lake Object Label = eComm Line Item
    - Category = Other

At this stage of configuration it is important to review how the system parsed the inbound data file and assigned a **data type** to each of the identified fields. The main reason for that is later in the configuration when you'll need to map these fields into the data model **any type mismatch will prevent you from doing so**, resulting in the need for you to go back to data stream setup and redo it.

- Notice that **SKU** field was identified as Number data type, while we need it to be set as Text:

S3 Order Line Items.csv (11)					<a href="#">New Formula Field</a>
<a href="#">Supported Fields (8)</a>		Unsupported Fields (0)	Lineage Fields (3)	Formula Fields (0)	
Source Header	Field Label	Field API Name	Data Type		
1 Quantity	Quantity	Quantity	Number		Text
2 Subtotal	Subtotal	Subtotal	Number		Number
3 Related Activity	Related Activity	Related_Activity	Text		DateTime
4 Order Number	Order Number	Order_Number	Text		Date
5 Loyalty Points	Loyalty Points	Loyalty_Points	Number		Email
6 Unit Price	Unit Price	Unit_Price	Number		Phone
7 Order Line Number	Order Line Number	Order_Line_Number	Number		URL
8 SKU	SKU	SKU	Number		Percent
					Boolean

- In addition ensure that both Order Number and Order Line Number are marked as Text:

○	4 Order Number	Order Number	Order_Number	Text
○	7 Order Line Number	Order Line Number	Order_Line_Number	Text

If you are to inspect the source file you'd notice that there is no single field that represents a Primary Key on the Order Line Item feed, although there is enough details to make a composite key using combination of few existing fields which can be achieved with a formula.

- Create new formula field that will combine Order Number and Order Line Number fields:
  - Field Label = Order Line Key
  - Formula Return Type = Text
  - Using appropriate [function](#) calculate the value as follows:
    - <Order Number>\_<Order Line Number>

- e.g. testing with values TRX-100 and 1 should result in TRX-100\_1  
*(notice that sequence of fields in **Tested Value** area might not be in order you expect them to be)*
- If you can't make the working formula or getting unexpected results have a look at the [reference below](#).
- **Save** & set Primary Key = Order Line Key and proceed to next step.
- Keep **default** Data Space and Refresh Mode = Upsert
- Keep Schedule with the following settings:
  - Frequency = None
  - Ensure to  Refresh Data Stream immediately option
- Deploy the data stream.

Functional formula for the Order Line Key field:

`CONCAT(sourceField['Order Number']+ '_' + sourceField['Order Line Number'])`

Further inspection of the Order Line Items feed would result in discovery of additional valuable data points that we want to represent and configure as separate objects in the resulting data model mapping to Goods Product and another custom entities. These are product SKUs and related activities associated with respective products in the same order line item record.

As you've already experienced one way to handle this would be by configuring additional data streams re-ingesting the same source file with different configurations of the primary keys. Another approach that we mentioned earlier would be by using Data Transforms. We already configured streaming version to extract party identification records from the Rental records. For the remaining data points we will use batch data transform option.

 As you are configuring batch data transform try to analyze and contrast this method with re-ingestion. Consider when you configure target lake object and as a result have upfront design vs resulting object is constructed for you. Also notice the difference in

schedule to automate the process. While we set **Frequency** = None for the S3 data streams, you can explore the options by editing any of the data streams without saving any changes. Compare these options to the ones available for batch data transforms. Last but not least consider what transformations can be applied when using ingestion comparing to data transform. All those considerations are good to undertake when you are designing your solution for the customer implementations.

### **3.6 Product SKU and Related Activity Lookup Data**

In order for us to configure data transforms for the product SKUs and related activities we need to create target data lake objects.

#### **3.6.1 Create Product SKU DLO**

- Navigate to **Data Lake Objects** tab and create new object choosing **New** option in method selection
- Set following properties:
  - Data Lake Object Name = eComm Product Lookup
  - Category = Other
  - Add following new fields:

<b>Field Label</b>	<b>Data Type</b>	<b>Primary Key</b>
Product Id	Text	1
Related Activity Id	Text	0

<b>Product Id</b>	<b>Text</b>	1
Related Activity Id	Text	0

**New Data Lake Object**

Add or remove fields, then select the primary key for your new Data Lake Object.

* Data Lake Object Name eComm Product Lookup	* Data Lake Object API Name eComm_Product_Lookup	* Category Other
---	---	---------------------

Select the appropriate category for the data in this Data Lake Object. Category selection affects billing and Data Model Object mapping. [Learn More in Help](#)

Field Label	Field API Name	Data Type	Primary Key	Record Modified Field	
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"/>	
cdp_sys_SourceVersion	cdp_sys_SourceVersion	Text	<input type="checkbox"/>	<input type="checkbox"/>	
Product Id	Product_Id	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Related Activity Id	Related_Activity_Id	Text	<input type="checkbox"/>	<input type="checkbox"/>	

[+ Add Field](#)

[Back](#) **Save**

- Save the object.

Inspect the **Fields** section of the configured object and notice that in addition to default and custom fields that we specified one additional field was configured as a **Key Qualifier**:

The screenshot shows the configuration of a Data Lake Object named 'eComm Product Lookup'. At the top, there are tabs for 'Fields' (which is selected) and 'Details'. In the 'Details' section, the object is categorized as 'Other' and has an 'Active' status. The 'Fields' section lists seven fields with their respective labels, API names, data types, and usage qualifiers. The 'Product Id' field is highlighted with a blue border and a checkmark, indicating it is the primary key.

Field Label	Field API Name	Data Type	Field Used As	Key Qualifier
cdp_sys_SourceV...	cdp_sys_SourceVers...	Text		
Data Source	DataSource__c	Text		
Data Source Object	DataSourceObject__c	Text		
Internal Organizati...	InternalOrganization...	Text		
KQ_Product_Id	KQ_Product_Id__c	Text		"Product Id" ...
Product Id	Product_Id__c	Text	Primary Key	
Related Activity Id	Related_Activity_Id__c	Text		

[This section](#) of the documentation provides great insights into what these qualifiers are and why/when they should be configured. In our scenario we don't really have a chance for the key clashes as such, due to limited overlap across CRM and eCommerce data sources.

But as we have an opportunity let's explore configuration of this feature too. Considering potential to add another source with transactional details that might include another set of SKU values. There is some level of likelihood for two systems to have same identifier value used as a SKU. And using this hypothesis let's configure key qualifier for the **Product Id** field in our object.

- Click on the  to the right of the **Product Id** field and choose Add Key Qualifier option:
  - Start creation of the new key qualifier and set following details:
    - Key Qualifier Label = Product Id PK
    - Description = Primary key of the eComm Product Lookup key qualifier

The screenshot shows a modal dialog titled "New Key Qualifier". Inside, there are two input fields with validation stars: "Key Qualifier Label" containing "Product Id PK" and "Key Qualifier API Name" containing "Product\_Id\_PK". Below these is a "Description" field containing "Primary key of the eComm Product Lookup key qualifier". At the bottom right are "Cancel" and "Save" buttons.

- Save the record. Then choose it in the list of qualifiers and click **Add**:

The screenshot shows a list of "Key Qualifiers" with one item selected: "Product\_Id\_PK" (API Name: "Product Id PK", Description: "Primary key of the eComm..."). Below the list is a "New Key Qualifier" button and at the bottom right are "Cancel" and "Add" buttons.

- Verify it was configured by clicking on the  next to **KQ\_Product\_Id** field and choosing View Key Qualifiers option:

Fields (8)						<input type="text"/> Search...
	Field Label	Field API Name	Data Type	Field Used As	Key Qualifier	
1	cdp_sys_SourceV...	cdp_sys_SourceVers...	Text			
2	Data Source	DataSource__c	Text			
3	Data Source Object	DataSourceObject__c	Text			
4	Internal Organizati...	InternalOrganization...	Text			
5	KQ_Product_Id	KQ_Product_Id__c	Text			
6	Product Id	Product_Id__c	Text	Primary Key		

- You will see configured key qualifier in the list:

Key Qualifier Details			
Select a key qualifier to view all the fields that share the same key qualifier.			
Key Qualifiers			
<input type="button" value="Select..."/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Product_Id_PK	Product Id PK	eComm Product Lookup	Product Id
1			<input type="button" value="Close"/>

This key qualifier now when combined with primary key will provide accurate reference to the respective record in scenarios when there are multiple sources of data with same values.

### 3.6.2 Create Related Activity DLO

Next let's configure another object for the related activity lookup records.

- From the **Data Lake Objects** tab create new object choosing **New** option in method selection
- Set following properties:
  - Data Lake Object Name = eComm Related Activity Lookup
  - Category = Other
  - Add following new fields:
    - Field Label = Related Activity, Data Type = Text, Primary Key =

**New Data Lake Object**

Add or remove fields, then select the primary key for your new Data Lake Object.

*Data Lake Object Name	*Data Lake Object API Name	*Category
eComm Related Activity Lookup	eComm_Related_Activity_Lookup	Other

Select the appropriate category for this Data Lake Object. Category selection is required for Data Model Objects and Billing.

**Field Labels**

Field Label	Field API Name	Data Type	Primary Key	Record Modified Field
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"/>
cdp_sys_SourceVersion	cdp_sys_SourceVersion	Text	<input type="checkbox"/>	<input type="checkbox"/>
Related Activity	Related_Activity	Text	<input checked="" type="checkbox"/>	<input type="checkbox"/>

[Back](#)

- Save the object.

We won't configure key qualifier for this object, as it simply represents the collection of category values. Even if they will be coming from various sources the design of the object is quite simple and will handle same values without issues.

### 3.6.3 Add Created DLOs to the Data Space

In order for us to use the created DLOs in the solution configuration by mapping to the data model they need to be associated with the default data space.

- Navigate to **Data Spaces** tab and open **default** data space by clicking on its name
- Click **Add Data Lake Objects** in top right
- Choose both DLOs created in previous steps and proceed to the next step:

Add Objects to default

Press 'enter' key to search

Select data lake objects to include in this data space. You can include data lake objects in more than one data space.

2 data lake objects selected.

Data Lake Object	Data Lake Object Name	Object API Name
	<input checked="" type="checkbox"/> eComm Product Lookup	eComm_Product_Lookup_dll
	<input checked="" type="checkbox"/> eComm Related Activity Lookup	eComm_Related_Activity_Lookup_dll

[Cancel](#) [Next](#)

- Confirm your intent to add them without filters and save configuration:

Add Objects to default

Review your data selected and set filters, if needed.

**Selected Objects (2)**

Data Lake Object Name	Object Type	Filters
eComm Product Lookup	<input checked="" type="checkbox"/> Data Lake Object	<a href="#">Set Filters</a>
eComm Related Activity Lookup	<input checked="" type="checkbox"/> Data Lake Object	<a href="#">Set Filters</a>

**⚠ You are about to add 2 object(s) to the data space "default" with no filters applied.**  
Anyone with access to this space will see the entire object. Click the acknowledgement or add filters to proceed.

Add objects without filters

[Previous](#) [Save](#)

Next we proceed with creation of the data transform that will be used to populate those objects.

### 3.6.4 Create Batch Data Transform populating Lookup DLOs

For both lookup DLOs we need to extract unique set of records. To achieve this we will use **eComm Line Item** records, get unique product SKUs and related activities and then output the resulting data sets into previously configured DLOs.

- Navigate to **Data Transforms** tab and start configuration of a new data transform
- Choose **Batch Data Transform** type and proceed to the next step
- Choose **Data Lake Objects** and proceed to the next step

You'll be taken to the Batch Data Transform canvas where you can author the actual transformations. Start by adding input data to the canvas:

- Choose **eComm Line Item** object, deselect all fields and re-select only following:
  - Related Activity
  - SKU

The screenshot shows the 'Add Input Data' dialog. On the left is a list of fields with checkboxes, and on the right is a 'Selected Fields' table.

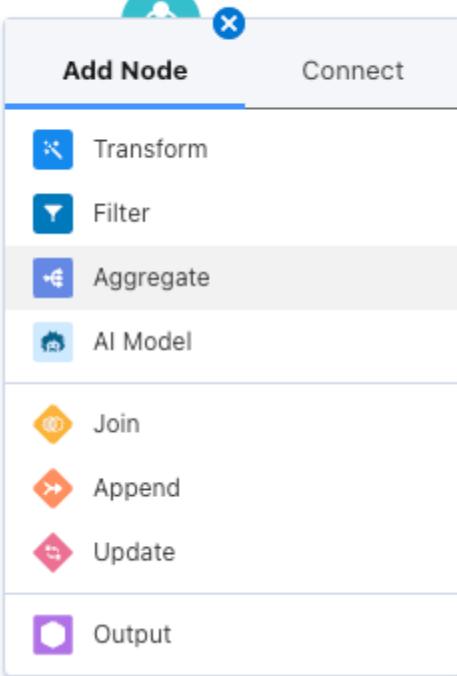
**Available Fields:**

Name	API Name	Category	Type
<input type="checkbox"/> Booking_c_00Da...	Booking_c_00Dam000...	ENGAGEMENT	Local
<input type="checkbox"/> Case_00Dam000...	Case_00Dam000004k...	ENGAGEMENT	Local
<input type="checkbox"/> Contact_00Dam0...	Contact_00Dam00000...	PROFILE	Local
<input type="checkbox"/> Party Identificatio...	Party_Identification_Co...	PROFILE	Local
<input type="checkbox"/> Rental_Preferenc...	Rental_Preference_c_0...	OTHER	Local
<input type="checkbox"/> Rental_c_00Dam0...	Rental_c_00Dam00000...	ENGAGEMENT	Local
<input type="checkbox"/> eComm Customer...	eComm_Customer_Pro...	PROFILE	Local
<input checked="" type="checkbox"/> eComm Line Item	eComm_Line_Item__dl...	OTHER	Local
<input type="checkbox"/> eComm Order	eComm_Order__dl...	ENGAGEMENT	Local
<input type="checkbox"/> eComm Product L...	eComm_Product_Look...	OTHER	Local
<input type="checkbox"/> eComm Related A...	eComm_Related_Activi...	OTHER	Local
<input type="checkbox"/> eComm Store Nu...	eComm_Store_Number...	OTHER	Local

**Selected Fields (eComm\_Line\_Item\_\_dl...):**

Name	API Name
<input type="checkbox"/> # Quantity	Quantity_c
<input type="checkbox"/> A <sub>3</sub> Order Number	Order_Number_c
<input checked="" type="checkbox"/> A <sub>3</sub> SKU	SKU_c
<input type="checkbox"/> # Unit Price	Unit_Price_c
<input type="checkbox"/> A <sub>3</sub> Data Source	DataSource_c
<input checked="" type="checkbox"/> A <sub>3</sub> Related Activity	Related_Activity_c
<input type="checkbox"/> A <sub>3</sub> Data Source Object	DataSourceObject
<input type="checkbox"/> A <sub>3</sub> KQ_Order_Line_Key	KQ_Order_Line_K...
<input type="checkbox"/> A <sub>3</sub> Order Line Key	Order_Line_Key
<input type="checkbox"/> # Subtotal	Subtotal_c
<input type="checkbox"/> A <sub>3</sub> Internal Organization	InternalOrganization
<input type="checkbox"/> A <sub>3</sub> Order Line Number	Order_Line_Numb...
<input type="checkbox"/> # Loyalty Points	Loyalty_Points_c

- Proceed to the next step
- Add an aggregate node to the data source:

- 
- The screenshot shows a data processing interface with a central workspace and a floating node selection dialog box. The dialog has a blue header bar with the text "Add Node" and "Connect". Below this are several categories of nodes:
- Transform
  - Filter
  - Aggregate (highlighted with a light gray background)
  - AI Model
  - Join
  - Append
  - Update
  - Output
- Below the dialog, there is a list of steps or actions:
- Update the name to Get Unique Related Activities
  - Add an aggregate:
    - Unique Related Activity
  - Add a row grouping:
    - Related Activity

The screenshot shows the configuration interface for the 'Get Unique Related Activities' aggregate node. At the top, there's a summary text: 'Summarize large amounts of data with aggregates and groups. Add an aggregate to start. To get final column results, run the batch data transform.' Below this is a toggle switch for 'Hierarchical Aggregation'. Under 'Aggregates', there's a section for 'eComm\_Line\_Item\_dll' with a sub-section for 'Unique Related Activity'. A plus sign button is available to add more aggregates. The 'Group Rows' section contains a row for 'eComm\_Line\_Item\_dll' under 'Related Activity', with a plus sign button to add more rows. The 'Group Column' section has a plus sign button to add columns. On the right side, there's a preview table showing the results:

Preview	Columns
A_a Related Activity	# Unique Related Activity
Training	1
Biking	1
Winter	1
Climbing	1
Camping	1
Hiking	1

- - Click **Apply** to save the configuration of the aggregate node.
- Add another aggregate node to the data source:
  - Update the name to Get Unique Product SKUs
  - Add an aggregate:
    - Unique SKU
  - Add a row grouping:
    - SKU
    - Related Activity

**AGGREGATE**  
Get Unique Product SKUs

Summarize large amounts of data with aggregates and groups. Add an aggregate to start. To get final column results, run the batch data transform.

Hierarchical Aggregation

\* Aggregates

eComm\_Line\_Item\_dll  
**Unique SKU**

+

Group Rows

eComm\_Line\_Item\_dll  
SKU

eComm\_Line\_Item\_dll  
Related Activity

+

Group Column

+

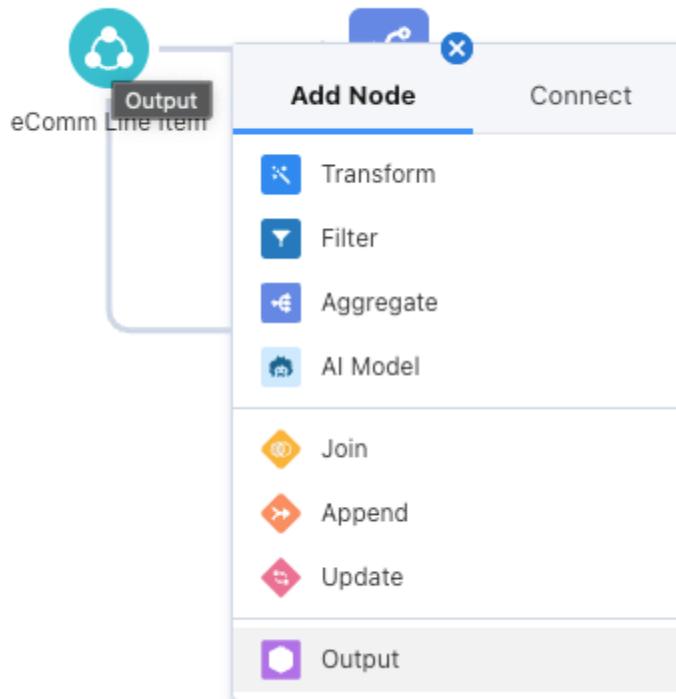
- 

Preview		Columns
A_a SKU	A_a Related Activity	# Unique SKU
101144	Training	1
101206	Training	1
101140	Biking	1
101225	Winter	1
101133	Training	1
101101	Climbing	1
101164	Training	1
101224	Winter	1
101222	Training	1
101139	Training	1
101187	Camping	1
101177	Camping	1
101183	Biking	1
101143	Biking	1
101223	Winter	1
101179	Climbing	1

- Click **Apply** to save the configuration of the aggregate node.
- Your canvas should now look like this:



- Add an output node to the **Get Unique Related Activities** aggregate node:



- - Rename to Output Related Activities
  - Set Data Lake Object Name = eComm Related Activity Lookup
  - Keep suggested mapping of Related Activity to Related\_Activity\_\_c

Preview	Columns
<b>A_a Related Activity</b>	# Unique Related Ac
Training	1
Biking	1
Climbing	1
Winter	1
Camping	1
Hiking	1

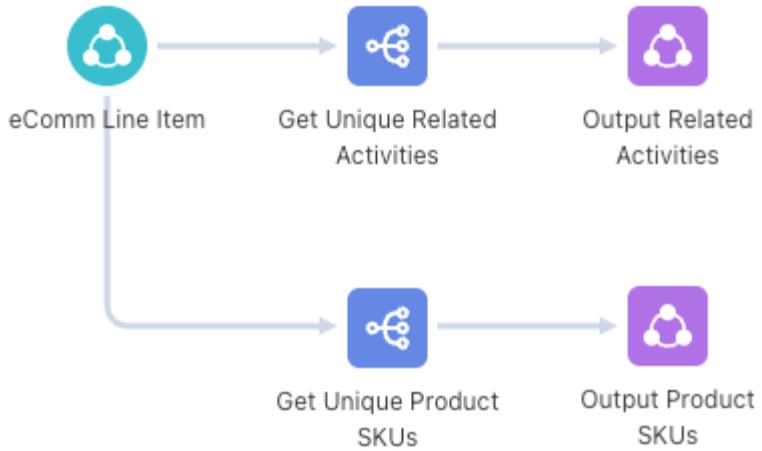
- Click **Apply** to save the configuration of the output node.

- Following similar process add an output node to the **Get Unique Product SKUs** aggregate node
  - Rename to Output Product SKUs
  - Set Data Lake Object Name = eComm Product Lookup
  - Map SKU to Product\_Id\_\_c
  - Keep suggested mapping of Related Activity to Related\_Activity\_Id\_\_c

Preview		
	Columns	
A_a SKU	A_a Related Activity	# Unique
101206	Training	1
101140	Biking	1
101135	Climbing	1
101226	Winter	1
101216	Training	1
101213	Biking	1
101153	Biking	1
101168	Training	1
101108	Camping	1
101156	Biking	1
101139	Training	1

- Click **Apply** to save the configuration of the output node.

Your canvas should now look like this:



Click **Save** button and set name = Extract Unique Product SKUs and Related Activities and save the data transform. Click **← Back to Data Transforms** button to close the editor and return to the Data Transforms tab. In the list click on the  button on the right of the newly created data transform and choose **Schedule** option:

	Data Transform Name	Data Space	Type	Data Trans...	Last Run Status	Last Run Time	
1	<a href="#">Extract Unique Product SKUs and Related Activities</a>		Batch	Active	None		<input checked="" type="checkbox"/>
2	<a href="#">Extract Party Identifiers from Rentals</a>		Streaming	Active	In Progress	3/28/2024, 6:12 A	<input type="checkbox"/>

We won't be scheduling the run of the data transform, but to compare review available options under the **Scheduled** option of **Run Mode**:

Schedule Data Transform

If a batch data transform is running at the same time another one is scheduled to begin, then the following one runs next at its scheduled interval.

**Run Mode**

Manual    Scheduled

**\* Frequency**

Hour(s)

**\* Run Every**

Hours

**\* Start Time**

America/Los\_Angeles

**\* On Days**

Monday    Tuesday    Wednesday    Thursday    Friday    Saturday    Sunday

Cancel the setting and using the same  control open menu and choose **Run Now** option:

Data Transforms		Recently Viewed	New	Update Status
2 items • Updated 4 minutes ago				
				
<input type="checkbox"/>	Data Transform Name	Data Space	Type	Data Trans... Last Run Status Last Run Time
1	<input type="checkbox"/> Extract Unique Product SKUs and Related Activities	Batch	Active	None
2	<input type="checkbox"/> Extract Party Identifiers from Rentals	Streaming	Active	In Progress 3/28/2024, 6:12 A
<input type="button" value="Run Now"/> <input type="button" value="Schedule"/>				

Once the processing completes you will be able to verify the results in the corresponding data lake objects.

 Note that first processing of the data transform might take a while to get started. You don't need to wait until it's finish, rather come back at the completion of the next exercise to verify records count as listed below. 

#### 4. Validation

Once you configured all above-listed data streams and data transforms, and the data for them has been ingested (you might need to use **Update Status** button to refresh it) and processed validate the **Total Records** counts for each respective stream under the **Data Streams** tab:

- Booking\_c\_<ORG\_ALIAS> = 8
- Case\_c\_<ORG\_ALIAS> = 5
- Contact\_c\_<ORG\_ALIAS> = 100
- eComm Customer Profile = 100
- eComm Line Item = 649
- eComm Order = 252
- eComm Store Number Lookup = 35
- Rental\_c\_<ORG\_ALIAS> = 22
- Rental\_Preference\_c\_<ORG\_ALIAS> = 7

As we have two data transforms configured navigate to the **Data Lake Objects** tab and verify **Total Records** count for the following objects (you might need to use **Update Status** button to populate the details):

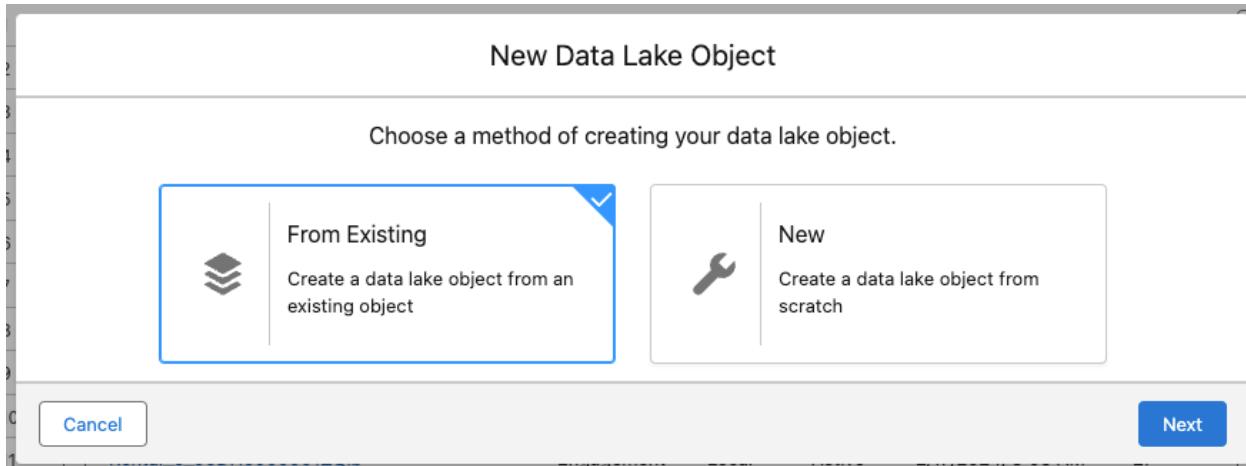
- eComm Product Lookup = 125
- eComm Related Activity Lookup = 6
- Party Identification Collection = 16

### **Configure Batch Data Transform:**

#### **Additional Steps - Evaluate If Required**

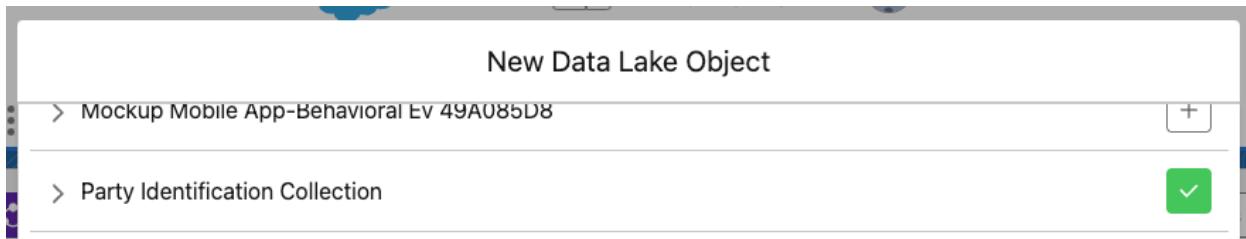
If you are exploring Batch Data Transform as an alternative to Streaming Data Transform, meaning you only configuring batch version then simply follow the document without any alterations.

If you are in fact configuring batch version **in addition** to streaming version then you need to create another data lake object as two data transforms as long as they are not both streaming type can not populate the same destination DLO. This will be quite easy, as you already created **Party Identification Collection** DLO.



Therefore for this guide create new DLO using **From Existing** option in the configuration dialog...

... and then selecting **Party Identification Collection** in the list



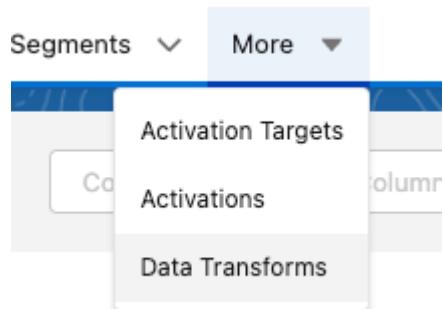
Set the name to Party Identification Collection - Batch to distinguish both easily, choose Profile as category and save. Ensure to add this object to the **default** data space prior to continuing with the main section below.

This should be the only change needed in order to proceed with this guide and configure batch data transform in addition to the streaming version. Do note that you'll need to use **Party Identification Collection - Batch** object in the output node of the last step of configurations below. Don't worry, if you'll forget this the platform will give you a warning with error notification upon attempt to save.

## Main Section

This guide will walk you through the steps of configuration of the Batch Data Transform to extract necessary Party Identification records from various DLOs.

- Navigate to **Data Transforms** tab and create new data transform.
  - **Note:** You may need to click the **More** in the navigation bar in order to view this tab:



- Select Batch Data Transform and click **Next**

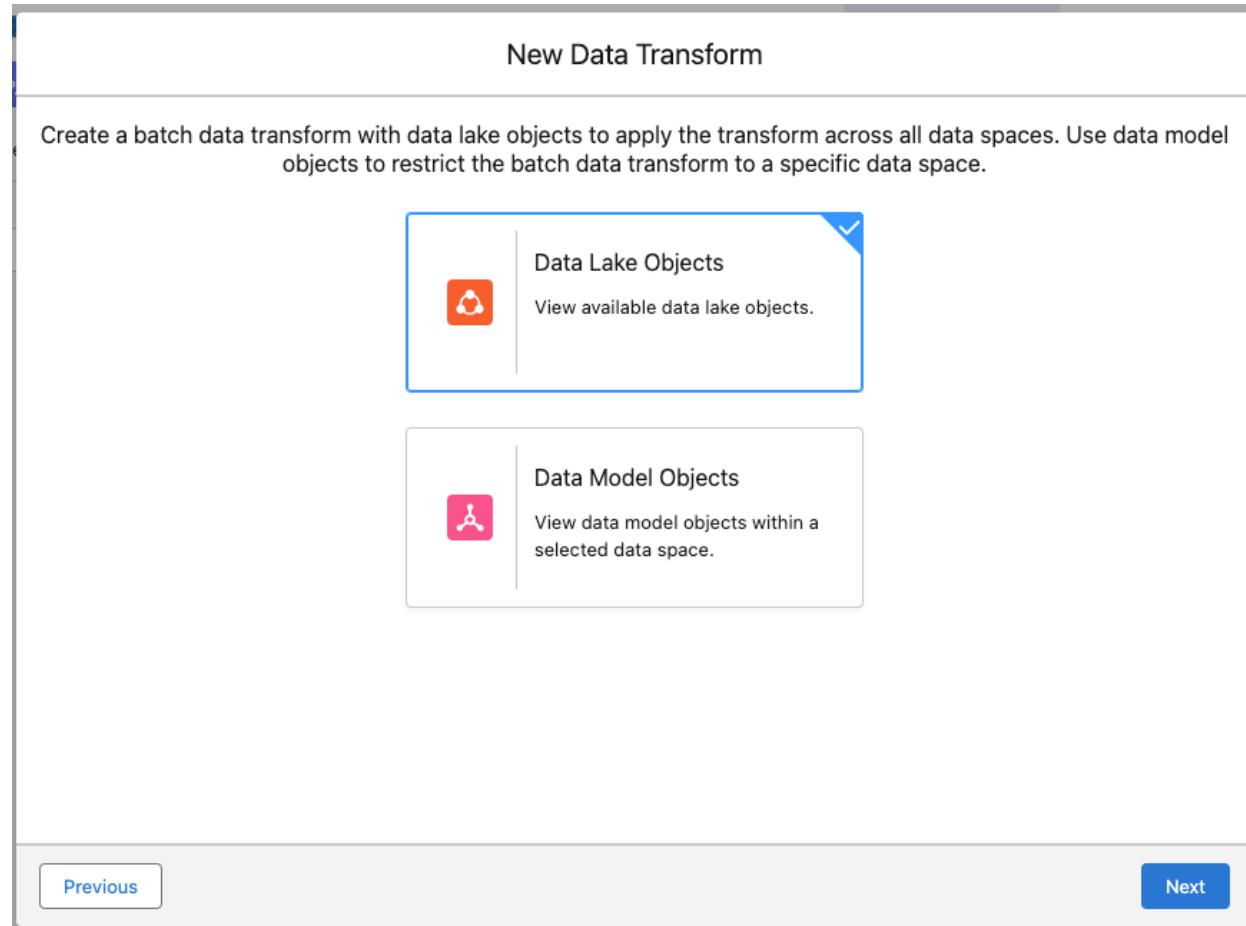
### New Data Transform

Choose a type of data transform

The screenshot shows a 'New Data Transform' wizard. At the top, it says 'Choose a type of data transform'. Below are two options:

- Batch Data Transform**: Represented by a clock icon. The text says: 'Use a visual editor to modify your data as needed or at scheduled intervals.' This option is highlighted with a blue border.
- Streaming Data Transform**: Represented by three right-pointing arrows icon. The text says: 'Use SQL to process your data in near-real time.'

- Select Data Lake Objects and click **Next**



- The action will open the new browser tab/window where you will configure necessary transformation steps.

## 1. Configure Input Data

To begin with the transforms we first need to identify the source of the data that will be manipulated.

- Click on **Add Input Data** button
- Choose **Rental\_c\_<ORG\_ALIAS>** object and remove selection from all fields clicking on the checkbox next to the **Name** title:

The screenshot shows a data selection interface with two main panels. On the left, a list of fields is displayed with columns for Name, API Name, and Category. On the right, a 'Selected Fields' panel lists specific fields with their corresponding API names.

Name	API Name	Category
Account_00DHo000001ZQlb	Account_00DHo000001ZQlb__dll	OTHER
Booking_c_00DHo000001ZQlb	Booking_c_00DHo000001ZQlb__dll	ENGAGEMENT
Case_00DHo000001ZQlb	Case_00DHo000001ZQlb__dll	ENGAGEMENT
Contact_00DHo000001ZQlb	Contact_00DHo000001ZQlb__dll	PROFILE
Drink Kiosk-Behavioral Events 1B88B24	Drink_Kiosk_Behavioral_Events_1B88B24__dll	ENGAGEMENT
Mockup Mobile App-Behavioral Ev 49A085D8	Mockup_Mobile_App_Behavioral_Ev_49A085D8__dll	ENGAGEMENT
Party Identification Collection	Party_Identification_Collection__dll	PROFILE
Party Identification Collection v02	Party_Identification_Collection_v02__dll	PROFILE
Rental_Preference_c_00DHo000001ZQlb	Rental_Preference_c_00DHo000001ZQlb__dll	OTHER
Rental_c_00DHo000001ZQlb	Rental_c_00DHo000001ZQlb__dll	ENGAGEMENT
eComm Customer Profile	eComm_Customer_Profile__dll	PROFILE
eComm Line Item	eComm_Line_Item__dll	OTHER
eComm Order	eComm_Order__dll	ENGAGEMENT
eComm Product Lookup	eComm_Product_Lookup__dll	OTHER
eComm Related Activity Lookup	eComm_Related_Activity_Lookup__dll	OTHER
eComm Store Number Lookup	eComm_Store_Number_Lookup__dll	OTHER

Name	API Name
# Baby Seat Count	Baby_Seat_Count_c__c
A <sub>3</sub> Drop Off Location Code	Drop_Off_Location_Code_c__c
A <sub>3</sub> Last Modified Date	LastModifiedDate__c
A <sub>3</sub> Data Source	DataSource__c
A <sub>3</sub> Data Source Object	DataSourceObject__c
A <sub>3</sub> Driver License Country of Issue	Driver_License_Country_of_Issue_c__c
A <sub>3</sub> External ID	External_Id_c__c
A <sub>3</sub> SfdcOrganizationId	SfdcOrganizationId__c
Booking Date	Booking_Date_c__c
A <sub>3</sub> Rental ID	Name__c
A <sub>3</sub> Is Business	Is_Business_c__c
System Modstamp	SystemModstamp__c
Last Viewed Date	LastViewedDate__c

- Locate and select the following fields:

- Customer
- Driver License Number
- Airline Membership Number
- Motor Club Number
- RAVG Retail Membership Number

Add Input Data

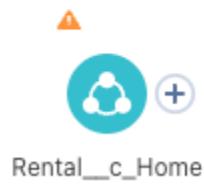
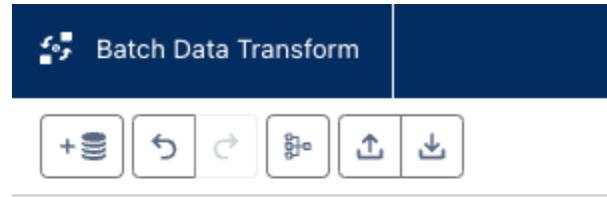
Name	API Name	Category	Type
<input type="checkbox"/> Booking_c_Ho...	Booking_c_Home_dll	ENGAGEMENT	Local
<input type="checkbox"/> Case_Home	Case_Home_dll	ENGAGEMENT	Local
<input type="checkbox"/> Contact_Home	Contact_Home_dll	PROFILE	Local
<input type="checkbox"/> Party Identificat...	Party_Identification_...	PROFILE	Local
<input type="checkbox"/> Party Identificat...	Party_Identification_...	PROFILE	Local
<input type="checkbox"/> Rental_Preferen...	Rental_Preference_c...	OTHER	Local
<input checked="" type="checkbox"/> Rental_c_Home	Rental_c_Home_dll	ENGAGEMENT	Local
<input type="checkbox"/> eComm Custom...	eComm_Customer_P...	PROFILE	Local
<input type="checkbox"/> eComm Line Item	eComm_Line_Item_...	OTHER	Local
<input type="checkbox"/> eComm Order	eComm_Order_dll	ENGAGEMENT	Local
<input type="checkbox"/> eComm Product...	eComm_Product_Lo...	OTHER	Local
<input type="checkbox"/> eComm Related...	eComm_Related_Act...	OTHER	Local
<input type="checkbox"/> eComm Store N...	eComm_Store_Numb...	OTHER	Local

Selected Fields (Rental\_c\_Home\_dll)

Name	API Name
<input checked="" type="checkbox"/> A3 Motor Club Number	Motor_Club_Number_c__c
<input checked="" type="checkbox"/> A3 Driver License Number	Driver_License_Number_c__c
<input checked="" type="checkbox"/> A3 Customer	Customer_c__c
<input checked="" type="checkbox"/> A3 Airline Membership Number	Airline_Membership_Number_c...
<input checked="" type="checkbox"/> A3 RAVG Retail Membership Nu...	RAVG_Retail_Membership_Nu...

Cancel Next

- Click **Next** to close the dialog. You will have new node added to the canvas:



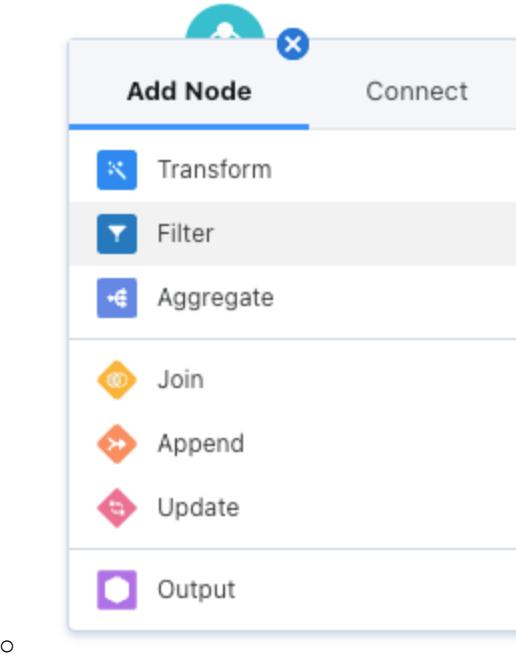
## 2. Filter And Transform Records

As we are operating with 4 various identifiers in the next steps we'll be separating records that have respective identifiers provided with the rental records.

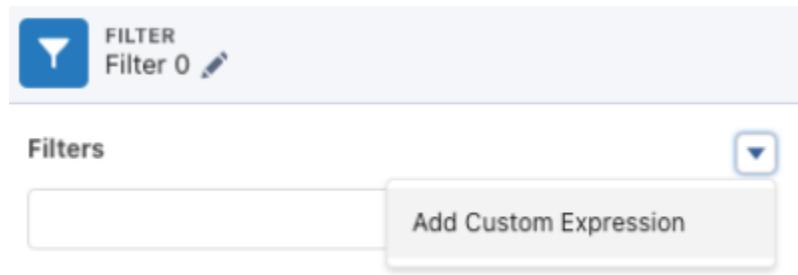
### 2.1 Driver License Number

This part of the process will extract details from records with Driver License Number specified in the Rental record.

- Add a new **Filter** node to the canvas:



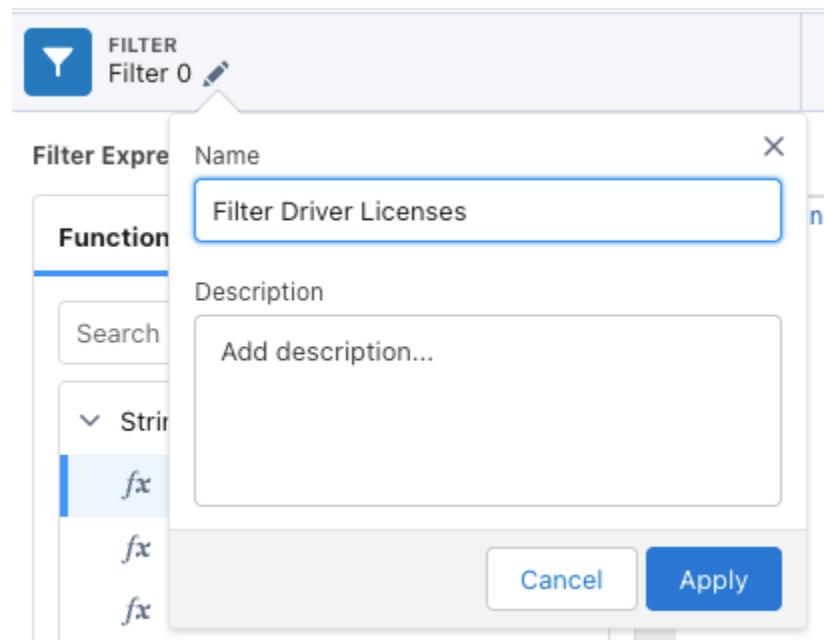
- Click on **▼** (arrow down) icon in the **Filters** pane and choose Add Custom Expression:



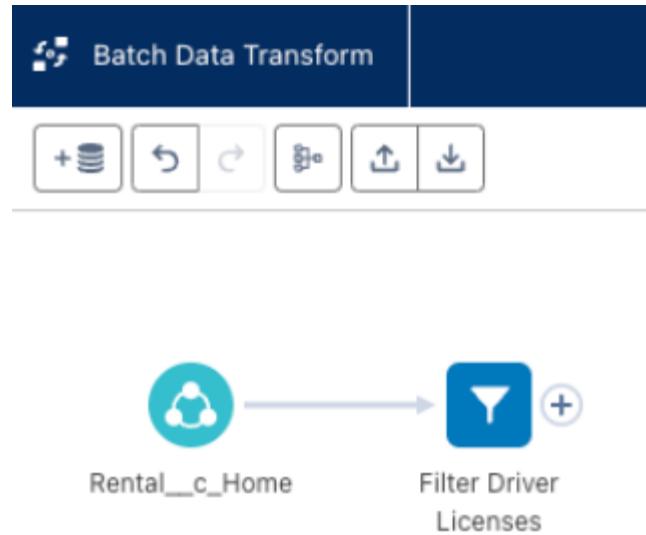
- - Use following expression value = `char_length(Driver_License_Number_c__c) > 0`
  - Verify that you get **22** records in the Preview (you'll have to manually count rows)

Preview	Columns		
Motor Club Number	A_a Airline Membership Number	A_a Driver License Number	A_a Customer
		O90-32062748	003am000002
		I80-66214240	003am000002
		N81-00828434	003am000002
-309008573		R00-86659649	003am000002
		P58-19259175	003am000002
-309008573		P58-19259175	003am000002
		X58-95447518	003am000002
		F25-08359644	003am000002

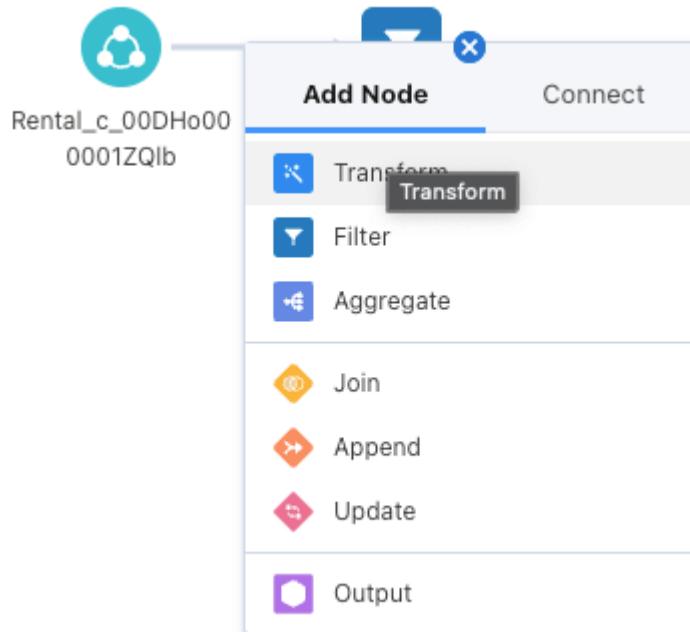
- Update the name of the filter to Filter Driver Licenses:



- Apply changes. Your canvas should look similar to this:



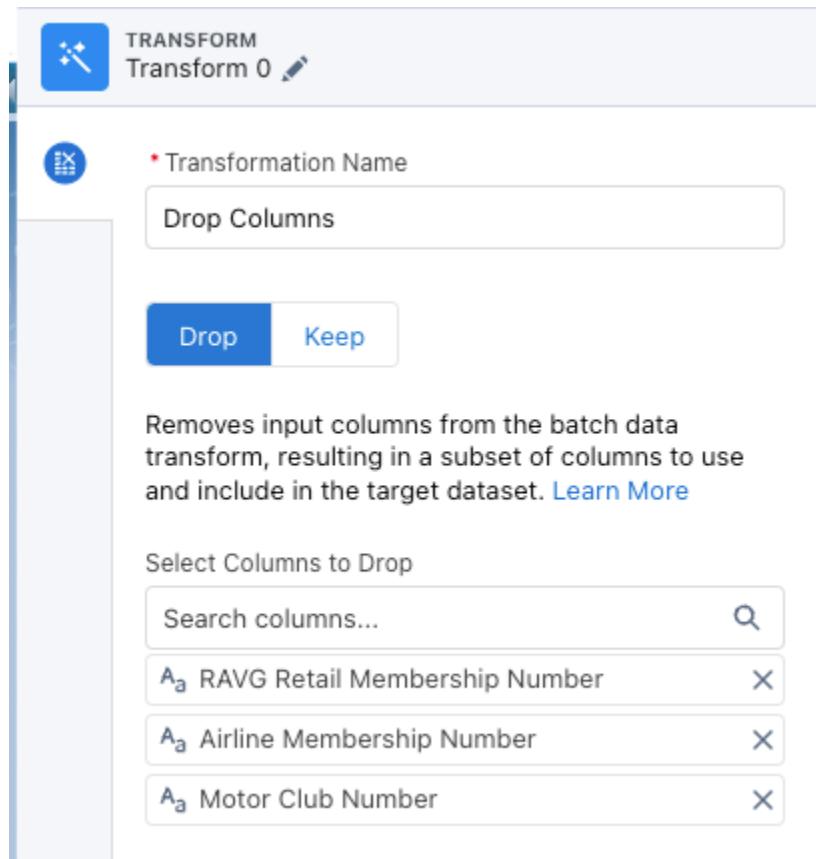
- Add a new **Transform** node after the previously configured filter:



- - As we only focus on Driver Licenses let's remove other identification columns. Click on **Drop Columns**:



- - In the list of columns choose following:
    - RAVG Retail Membership Number (will be preselected)
    - Airline Membership Number
    - Motor Club Number



- - Click **Apply** to save the change
  - With selected **Driver License Number** column click on **Edit Attributes** icon:

A <sub>3</sub> Driver License Number	A <sub>3</sub> Customer
O90-32062748	003am000002IT8wAAE
I80-66214240	003am000002IT8zAAE
N81-00828434	003am000002IT9RAAU

- o and update the details as follows:
  - Set **Label** = Identification Number
  - Set **API Name** = Identification\_Number\_\_c

\* Transformation Name  
Edit Attributes

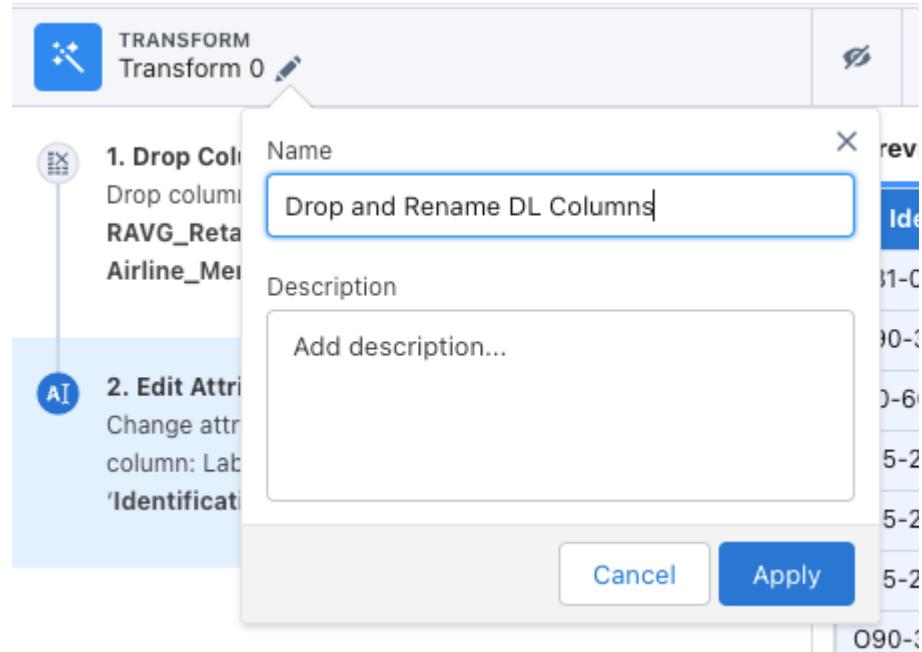
AI Updates the column attributes. [Learn More](#)

Label  
Identification Number

API Name  
Identification\_Number\_\_c

Length  
If left blank, default value is 255.

- Click **Apply**
- o Update the name of the node to Drop and Rename DL Columns:

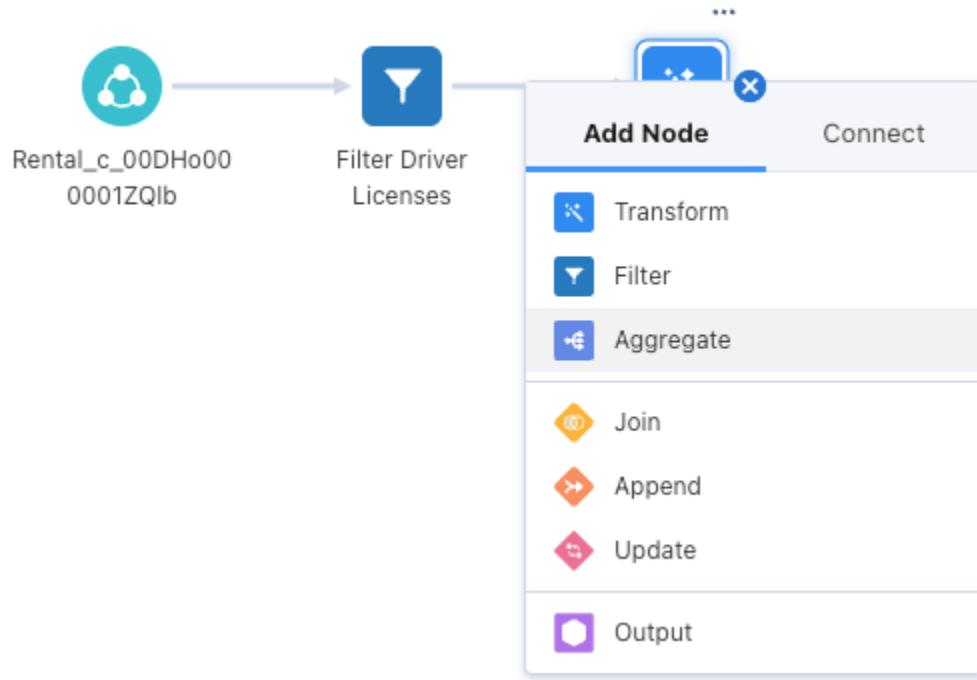


Notice that our resulting dataset contains duplicate values if you are to compare combination of **Identification Number** and **Customer** fields.

Preview		Columns	
A <sub>a</sub>	Identification Number	A <sub>a</sub>	Customer
	O90-32062748		003al000009E7T2AAK
	I80-66214240		003al000009E7T5AAK
	N81-00828434		003al000009E7TXAA0
	X58-95447518		003al000009E7TIAAO
	X58-95447518		003al000009E7TIAAO
	H03-02370198		003al000009E7TEAA0
	H03-02370198		003al000009E7TEAA0

Let's address this.

- Close the details of the Transform node and add a new **Aggregate** node to it:



- Click on + to add new aggregate
  - Select **Unique** and then **Identification Number**

The screenshot shows the "Batch Data Transform" interface. On the left, there is a section titled "AGGREGATE Aggregate 0" with a description: "Summarize large amounts of data with aggregates and groups. Add an aggregate to start. To get final column results, run the batch data transform." Below this, there are four input fields: "Hierarchical Aggregation" (disabled), "Aggregates" (empty), "Group Rows" (empty), and "Group Column" (empty). On the right, a modal window titled "Select Aggregate" is open. It has a search bar "Search column to aggregate..." and a table with columns "Name", "API Name", and "Source". The table lists the following rows:

	Name	API Name	Source
Sum	A_a Customer	Customer_c__c	Rental_c_00Dam00
Average	A_a Identification Number	Identification_Number__c	Rental_c_00Dam00
Count			
Maximum			
Minimum			
Stddev			
Stddev			
Varp			
Var			

The "Unique" row is selected, and the "Identification Number" row is highlighted with a red box. At the bottom of the table, there are two rows of binary data: "E29-003333044" and "F25-083506111".

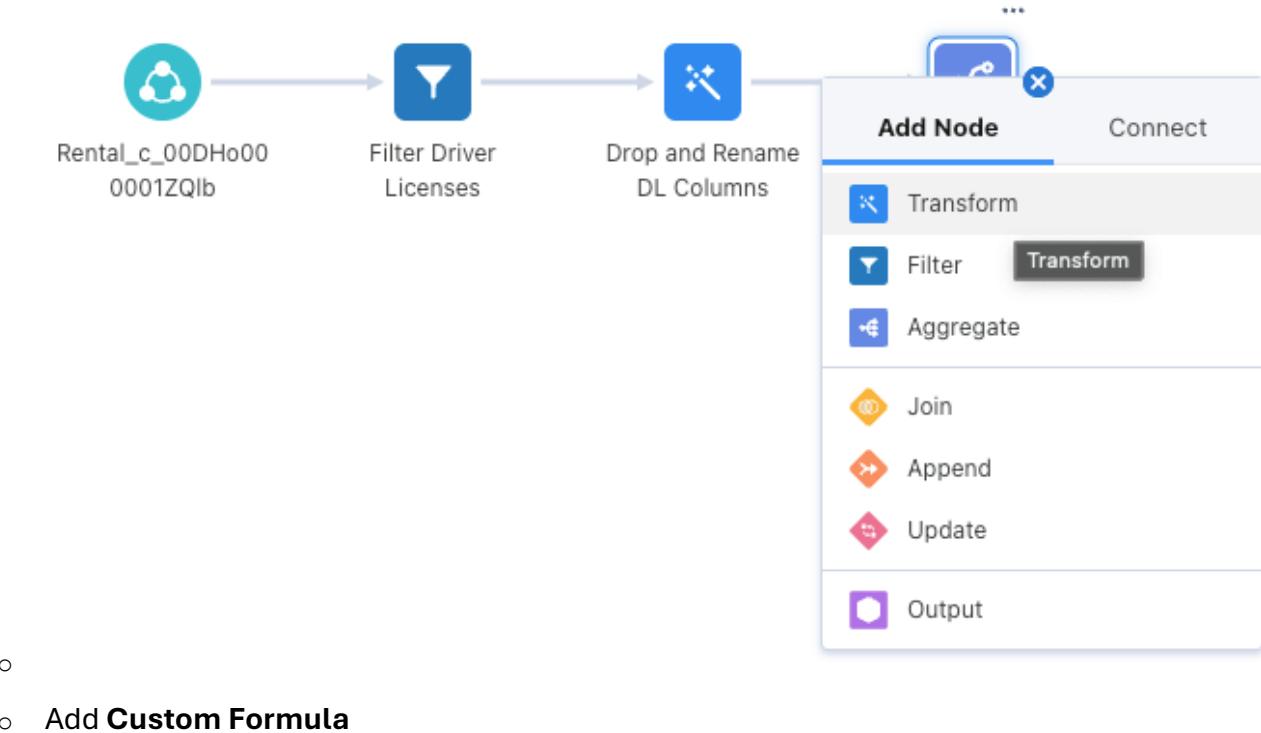
- Add following **Group Rows** options in this exact order:
  - Customer
  - Identification Number

Preview		Columns
A <sub>a</sub> Customer	A <sub>a</sub> Identification Number	# Unique Id
003am000002IT8wAAE	O90-32062748	1
003am000002IT8zAAE	I80-66214240	1
003am000002IT9RAAU	N81-00828434	1
003am000002IT9NAAU	R00-86659649	1
003am000002IT9MAAU	P58-19259175	1
003am000002IT9CAAU	X58-95447518	1
003am000002IT8yAAE	E25-08359644	1
003am000002IT8oAAE	O90-32062748	1
003am000002IT91AAE	N15-25466146	1
003am000002IT98AAE	H03-02370198	1
003am000002IT90AAE	O90-32062748	1

- Click **Apply**
- Notice in the Preview that you are now getting only **11** records
- Rename the node to Select Unique DL Identifiers
- Apply the changes. Your canvas should look similar to this:



- There are few extra fields that we need for the final result. Let's add those. Add a new **Transform** node after the previously configured aggregate:



- Add **Custom Formula**

The screenshot shows a "Date and Time" configuration dialog with a table:

Aa	Text
Aa C	Now
003H	Custom Formula
003Ho000001haKL6IAM	180-00214240
003Ho000001haKL6IAM	N15-2546614

A tooltip is displayed over the "Custom Formula" cell, stating: "Calculates and returns a result based on existing values in fields."

- Set value = concat(Customer\_c\_\_c, '\_', Identification\_Number\_\_c)
- Set **Output Type** = Text
- Keep **Length** with value **40**
- Set **Column Label** = Party Identification ID
- Set **API Name** = Party\_Identification\_ID\_\_c
- Click **Apply**

- Add another **Custom Formula**:

- Value = 'Person Identification' (notice single quote characters, as we are hardcoding text value)
- Output Type = Text
- Keep **Length** with value **40**
- Column Label = Party Identification Type
- API Name = Party\_Identification\_Type\_\_c

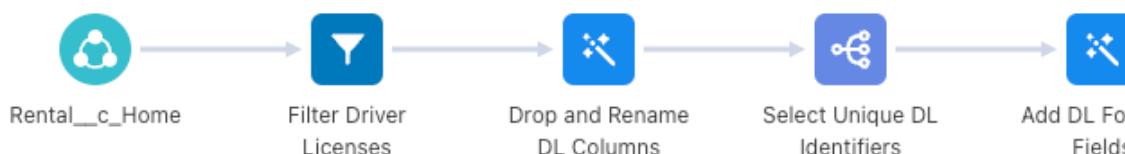
- Add another **Custom Formula**:

- Value = 'Driver License' (including single quote characters, as we are hardcoding text value)
- Output Type = Text
- Keep **Length** with value **40**
- Column Label = Identification Name
- API Name = Identification\_Name\_\_c

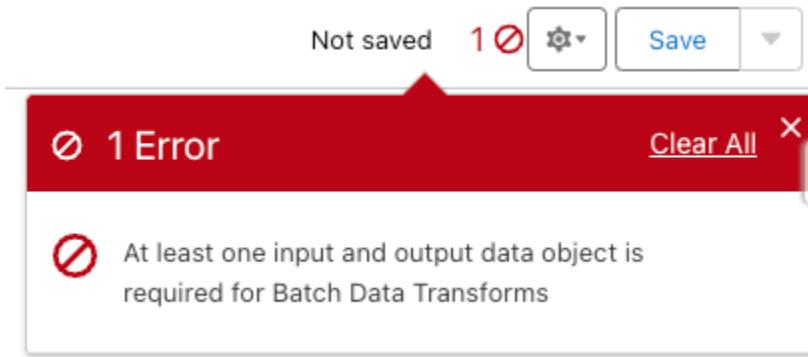
- The configuration of the node should look similar to this:

A3 Customer	A3 Identification Number	# Unique Identification Number	A3 Party Identification ID
003am000002IT8wAAE	090-32062748	1	003am000002IT8wAAE_090-32062748
003am000002IT8zAAE	I80-66214240	1	003am000002IT8zAAE_I80-66214240
003am000002IT9RAAU	N81-00828434	1	003am000002IT9RAAU_N81-00828434
003am000002IT9NAAU	R00-86659649	1	003am000002IT9NAAU_R00-86659649
003am000002IT9MAAU	P58-19259175	1	003am000002IT9MAAU_P58-19259175
003am000002IT9CAAU	X58-95447518	1	003am000002IT9CAAU_X58-95447518
003am000002IT8yAAE	E25-08359644	1	003am000002IT8yAAE_E25-08359644
003am000002IT8oAAE	O90-32062748	1	003am000002IT8oAAE_O90-32062748
003am000002IT91AAE	N15-25466146	1	003am000002IT91AAE_N15-25466146
003am000002IT98AAE	H03-02370198	1	003am000002IT98AAE_H03-02370198
003am000002IT90AAE	O90-32062748	1	003am000002IT90AAE_O90-32062748

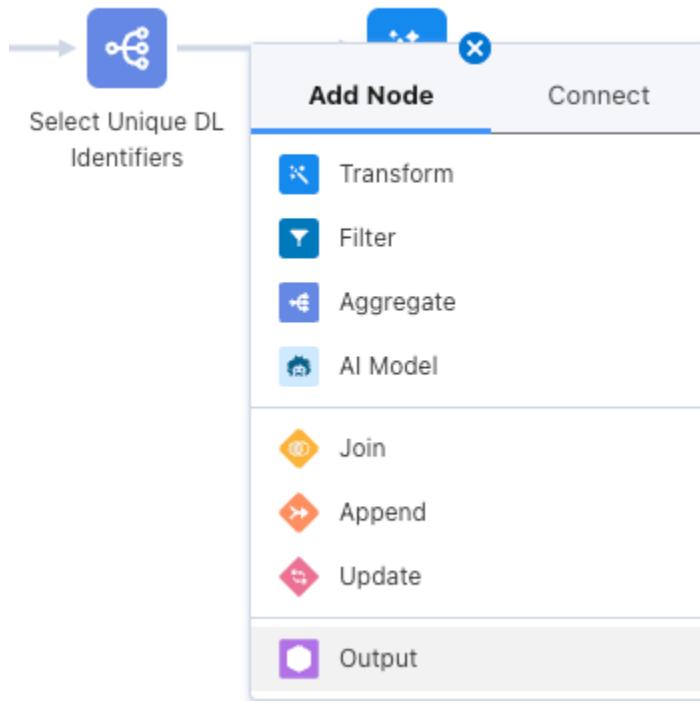
- Rename the node to Add DL Formula Fields
- Apply the changes. Your canvas should look similar to this:



Now we can apply similar configurations for the other identifiers. Although before moving to next steps let's save the progress by clicking the **Save** button in the top right, set the name Extract Party Identifications from Rentals (Batch) and try saving your progress. You'll notice that the configuration will not be saved as there is no output node configured for the transform:



Let's add one to ensure we can save the progress. Close the error notification and add **Output** node after the last transform:



- Choose Party Identification Collection (or Party Identification Collection - Batch if you already have streaming insight configured) object and configure options as follows:
  - Notice that mapping is mostly completed for you as the names of the fields are matched
  - Map **Customer** field to the Party\_\_c
  - Leave **# Unique Identification ...** unmapped (data source fields will be automatically populated in the destination DLO)

The screenshot shows the Power BI Data Flow canvas with a pipeline starting from 'Rental\_c\_Home' and ending at 'Output 0'. The steps in the pipeline are:

- Filter Driver Licenses
- Drop and Rename DL Columns
- Select Unique DL Identifiers
- Add DL Formula Fields
- Output 0

Below the pipeline, there is a section titled 'Write To' with a dropdown set to 'Data Lake Object'. Underneath it, a field labeled 'Data Lake Object Name' contains 'Party Identification Collection'.

Below this, there is a section titled 'Map batch data transform columns to Data Lake Object columns.' It shows a list of columns being mapped:

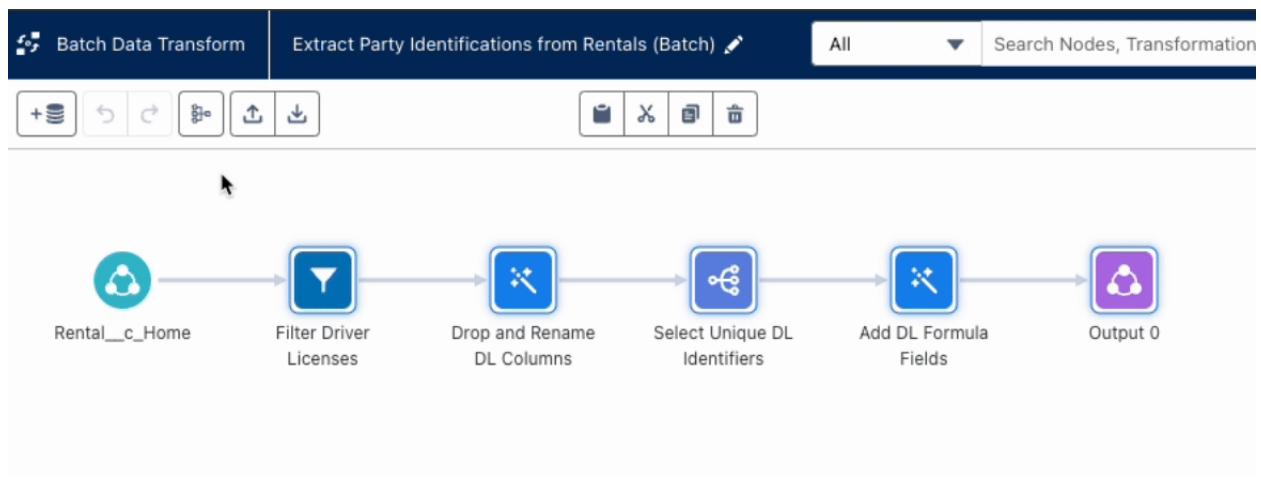
Batch Data Transform ...	Party Identification Collection
A_a Customer	A_a Party__c
A_a Identification Number	A_a Identification_Number__c
# Unique Identification ...	Search Columns...
A_a Party Identification ID	A_a Party_Identification_ID__c
A_a Party Identification Ty...	A_a Party_Identification_Type__c
A_a Identification Name	A_a Identification_Name__c

- Apply the changes. Your canvas should look similar to this:

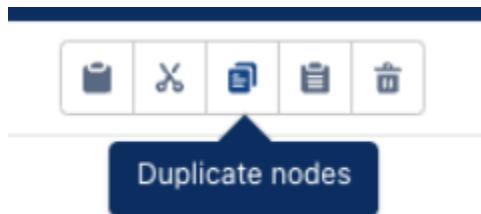
Now let's actually save your work. Click on the **Save** button, set the name Extract Party Identifications from Rentals (Batch) and save your progress. This time you should see confirmation of successful save.

## 2.2 Airline Membership Number

While we can repeat most of the configuration steps above let's try simplifying the process a little. Select all the transformation nodes that you've configured in the previous section (excluding **Output 0**) with click-n-drag overlay over them:



And once selected click on the **Duplicate nodes** icon above:



Before adjusting configuration of the duplicated nodes connect the Filter to the Input Data node:

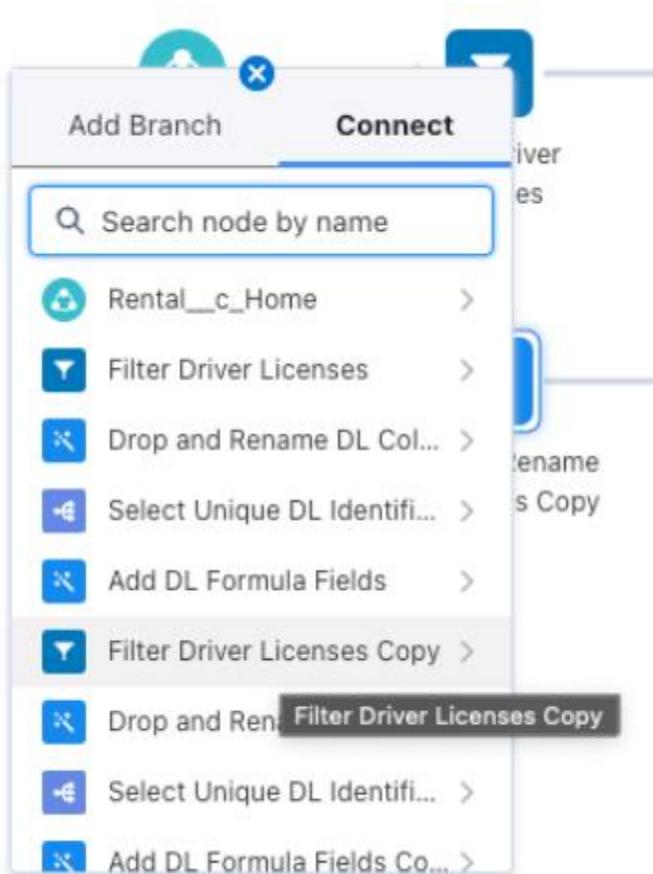
- Click on the split-arrows icon on the **Rental** Input Data node:



Rental\_c\_Home

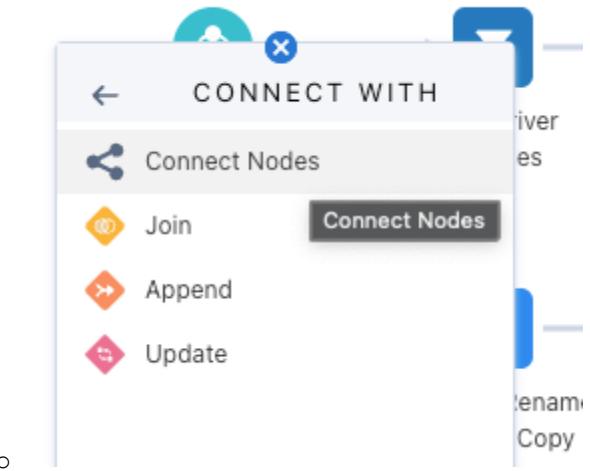
○

- Switch to the **Connect** tab and choose **Filter Driver Licenses Copy** node:



○

- and choose **Connect Nodes**:



You'll be switched to the configuration of the Filter node, so let's reconfigure it as well as others for the next identifier.

- Update the name of the node to Filter Airline Membership Numbers
- Update the filter expression to `char_length(Airline_Membership_Number_c__c) > 0`
  - Verify that you get **5** records in the Preview
- Apply the changes
- Select **Drop and Rename DL Columns Copy** node
  - Update name to Drop and Rename AIR Columns
  - Delete second step in the transform:

The screenshot shows a data transformation step titled "Drop and Rename AIR Columns". It has two main sections: "1. Drop Columns" and "2. Edit Attributes".

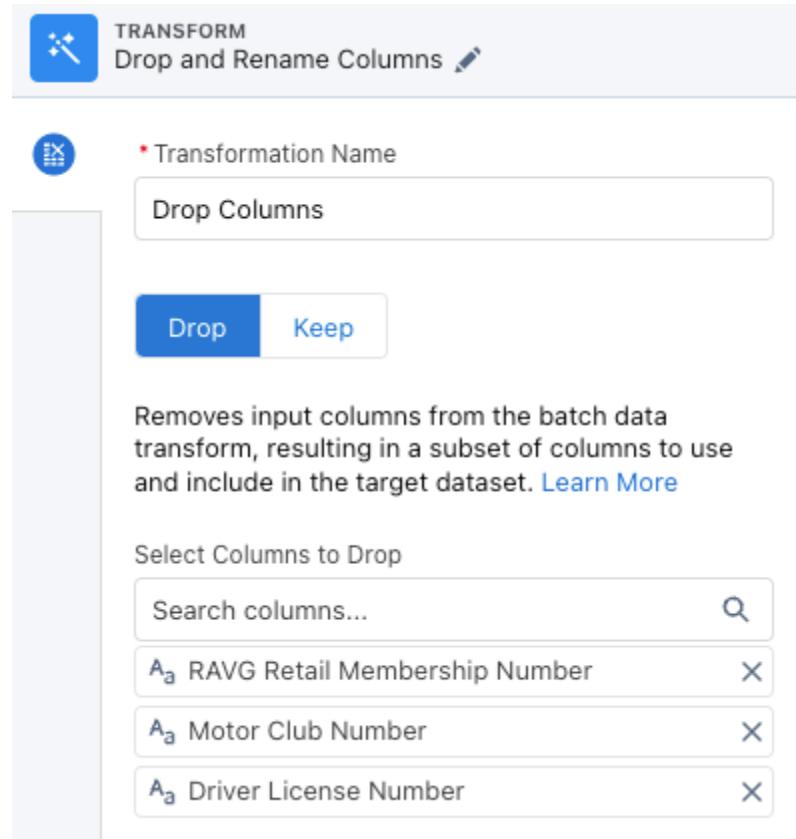
- 1. Drop Columns:** Describes dropping columns: Motor\_Club\_Number\_c\_\_c, RAVG\_Retail\_Membership\_Number\_c\_\_c, and Airline\_Membership\_Number\_c\_\_c.
- 2. Edit Attributes:** Describes changing attributes for the 'Driver\_License\_Number\_c\_\_c' column, setting the label to 'Identification Number' and the API Name to 'Identification\_Number\_\_c'.

A "Delete Transformation" button is visible above the preview area, which contains a list of rows with columns labeled "A\_a Identifica" and "N15-2546614".

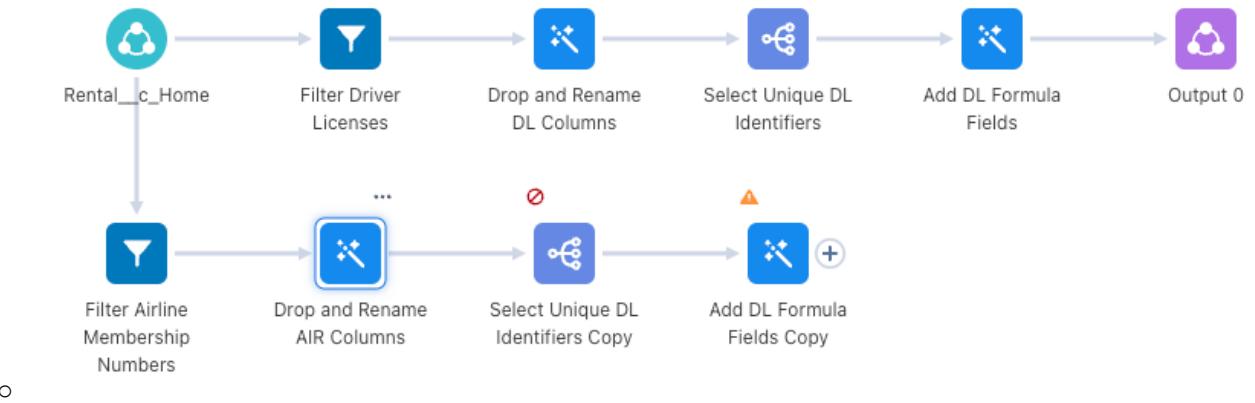
- Click on icon to edit first step:

The screenshot shows the same transformation step after clicking the edit icon. The "Edit Transformation" button is now highlighted. The "Drop Columns" section remains the same, but the "Edit Attributes" section is no longer visible.

- In the list of columns to drop remove **Airline Membership Number** and add **Driver License Number**:



- Apply the change and verify that you now see **Airline Membership Number** column in the Preview.
- With selected **Airline Membership Number** column click on **Edit Attributes** icon and update details as follows:
  - Set **Label** = Identification Number
  - Set **API Name** = Identification\_Number\_\_c
- Apply changes and close the details of the node.
- The following aggregate node will be highlighted with an error state and you will get an error notice popup on the right and that's okay, as we'll be correcting it next:



- Select **Select Unique DL Identifiers Copy** node
  - Update name to **Select Unique AIR Identifiers**
  - In the **Aggregates** click on the current value and change selection to **Unique** and **Identification Number**
  - In the **Group Rows** delete all fields but **Customer** and re-add **Identification Number**:

**AGGREGATE**  
Select Unique AIR Identifiers

Summarize large amounts of data with aggregates and groups. Add an aggregate to start. To get final column results, run the batch data transform.

Hierarchical Aggregation

\* Aggregates

Rental_c_Home__dll	Unique Identification Number
--------------------	------------------------------

+

Group Rows

Rental_c_Home__dll	Customer
--------------------	----------

Rental_c_Home__dll	Identification Number
--------------------	-----------------------

+

**Preview** **Columns**

A_a Customer	A_a Identification Number	# Unique Identification
003al000009E7SuAAK	FGH-3686-61759	1
003al000009E7T7AAK	FGH-3686-61759	1

- Notice that you now only get **2** records in the Preview.
- Apply changes and close the details of the node.
- Select **Add DL Formula Fields Copy** node:
  - Rename to **Add AIR Formula Fields**

- Click on the step 1 and then edit icon:

**TRANSFORM**  
Add AIR Formula Fields 

**Edit Transformation**

**1. Formula**  
Calculate values: `concat(Customer_c__c, '_'; Driver_License_Number_c__c)`

Your formula that creates column  
"Party\_Identification\_ID\_c" uses a field that doesn't exist. Update or remove the fields  
Driver\_License\_Number\_c\_\_c. Cause: Can't find the identifier "Driver\_License\_Number\_c\_\_c" in SQL expression [concat(Customer\_c\_\_c, '\_', Driver\_License\_Number\_c\_\_c)] at [1:28].

**2. Formula**  
Calculate values: 'Person Identification'

**3. Formula**  
Calculate values: 'Driver License'

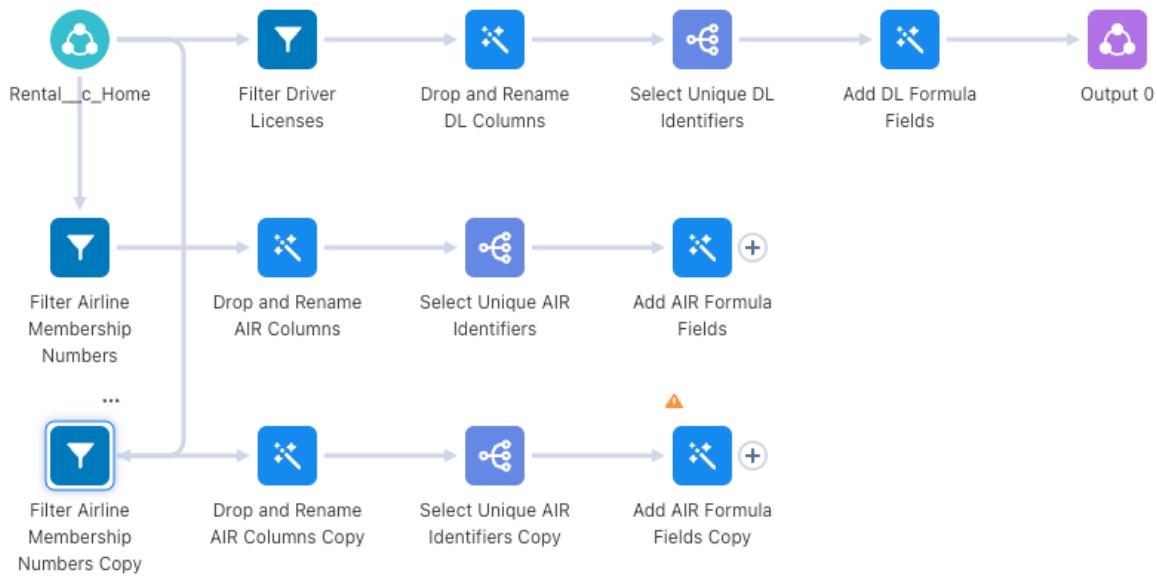
- Update the formula expression to `concat(Customer_c__c, '_', Identification_Number__c)`
- Select and edit step 2:
  - Update the formula expression to 'Loyalty Program'
- Select and edit step 3:
  - Update the formula expression to 'Airline Membership'
- The node configuration should look similar to this:

mber	Unique Identification Number	Party Identification ID	Party Identification Type	Airline M
1		003al000009E7SuAAK_FGH-3686-61759	Loyalty Program	Airline M
1		003al000009E7T7AAK_FGH-3686-61759	Loyalty Program	Airline M

Two out of four identifiers are catered for, save the progress by clicking on the **Save** button and let's move to the next one.

## 2.3 Motor Club Membership

Following a similar approach select and duplicate 4 nodes that you just configured for the Airline Membership Number and connect **Rental** Input Data node with the filter node:

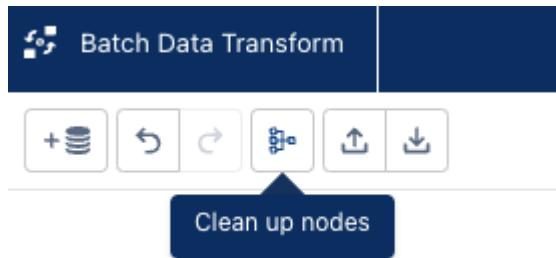


The layout got a little bit mixed up, but we'll fix it shortly.

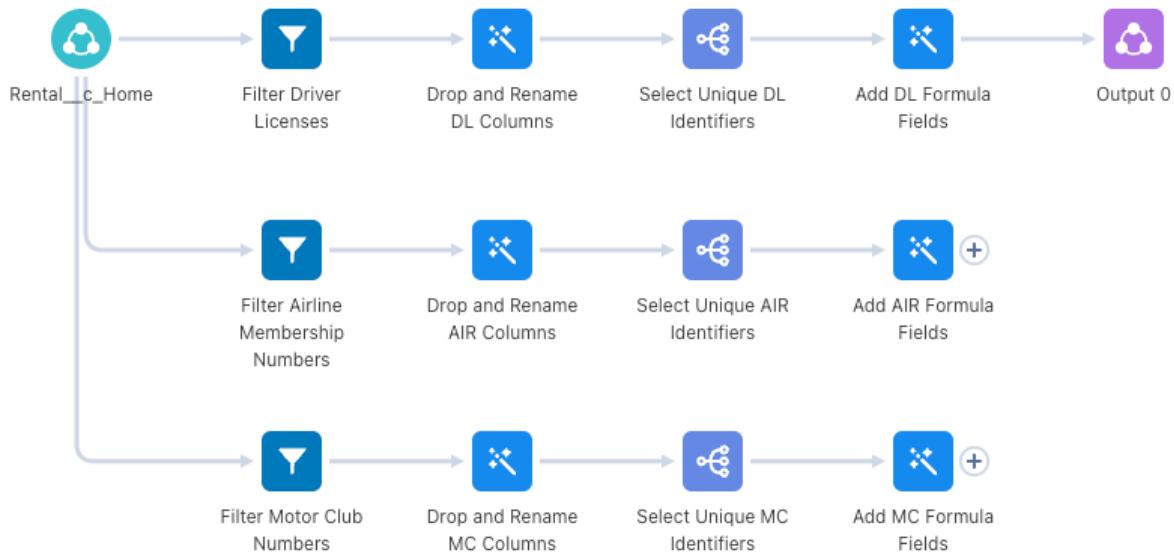
- Configure the opened filter activity as follows:
  - Set name = Filter Motor Club Numbers
  - Set expression = char\_length(Motor\_Club\_Number\_c\_c) > 0
  - Verify that you get **2** records in the Preview
  - Apply the changes

- Configure the following transform node as follows:
  - Set name = Drop and Rename MC Columns
  - Delete second step in the transform
  - Click on  icon to edit first step
  - In the list of columns to drop remove **Motor Club Number** and add **Airline Membership Number**
  - Apply the change and verify that you now see **Motor Club Number** column in the Preview.
  - With selected **Motor Club Number** column click on **Edit Attributes** icon and update details as follows:
    - Set **Label** = Identification Number
    - Set **API Name** = Identification\_Number\_\_c
  - Apply changes and close the details of the node.
- Configure the following aggregate node as follows:
  - Update name to Select Unique MC Identifiers
  - In the **Aggregates** click on the current value and change selection to **Unique** and **Identification Number**
  - In the **Group Rows** delete all fields but **Customer** and re-add **Identification Number**
  - Apply changes and close the details of the node
- Configure the following transform node as follows:
  - Rename to Add MC Formula Fields
  - Select and edit step 1 updating the expression to concat(Customer\_c\_\_c, '\_', Identification\_Number\_\_c)
  - No need to update second step
  - Select and edit step 3 updating the expression to 'Motor Club Membership'

Now before we proceed to the next identifier let's rearrange the nodes on the canvas. Click on the **Clean up nodes** button



and the tool will lay the nodes in a nice way without any confusing intertwining:



Save your progress. Now to the last remaining identifier.

## 2.4 RAVG Retail Membership Number

Continue with the previously chosen approach duplicate 4 nodes that you've just configured for the Motor Club Numbers connect **Rental** Input Data node with the filter node.

- Configure the opened filter activity as follows:
  - Set name = Filter RAVG Memberships
  - Set expression = char\_length(RAVG\_Retail\_Membership\_Number\_c\_\_c) > 0
  - Verify that you get **1** record in the Preview
  - Apply the changes

- Configure the following transform node as follows:
  - Set name = Drop and Rename RAVG Columns
  - Delete second step in the transform
  - Click on  icon to edit first step
  - In the list of columns to drop remove **RAVG Retail Membership Number** and add **Motor Club Number**
  - Apply the change and verify that you now see **RAVG Retail Membership Number** column in the Preview.
  - With selected **RAVG Retail Membership Number** column click on **Edit Attributes** icon and update details as follows:
    - Set **Label** = Identification Number
    - Set **API Name** = Identification\_Number\_\_c
  - Apply changes and close the details of the node.
- Configure the following aggregate node as follows:
  - Update name to Select Unique RAVG Identifiers
  - In the **Aggregates** click on the current value and change selection to **Unique** and **Identification Number**
  - In the **Group Rows** delete all fields but **Customer** and re-add **Identification Number**
  - Apply changes and close the details of the node
- Configure the following transform node as follows:
  - Rename to Add RAVG Formula Fields
  - Select and edit step 1 updating the expression to concat(Customer\_c\_\_c, '\_', Identification\_Number\_\_c)
  - No need to update second step
  - Select and edit step 3 updating the expression to 'RAVG Retail Membership'
- Click on the **Clean up nodes** button to re-arrange the canvas

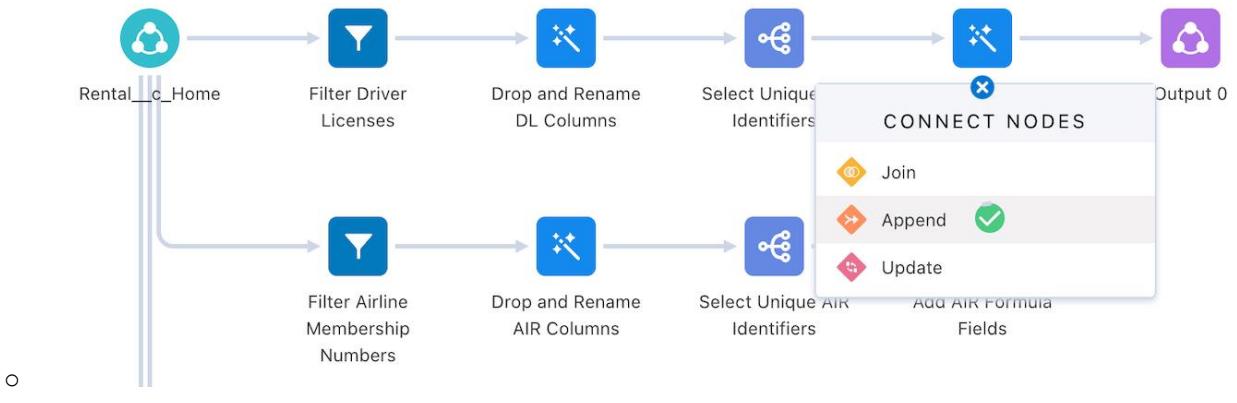
### 3. Combining Results

At this stage we have 4 separate datasets that are representing selection of different records but with the same field structure plus one of the datasets is configured with the output node.

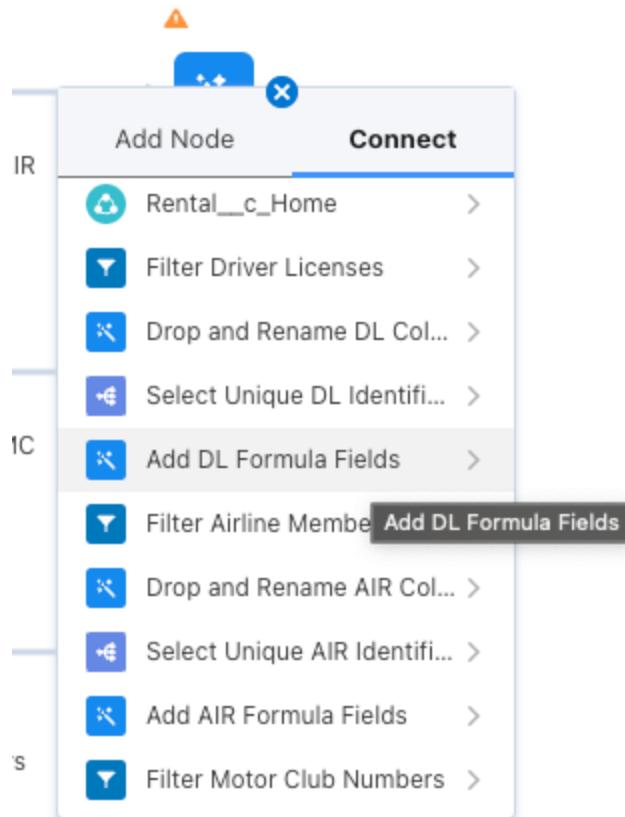


We are now going to combine all of the datasets together into a single dataset that will be later used as an overall output of this data transform.

- Starting at the second row, click on the + icon next to the **Add AIR Formula Fields** node and drag it to the one above it (**Add DL Formula fields**) and release the mouse key when you hover over the **Append** option:



- If you can't make that work you can also click on the + icon next to the **Add AIR Formula Fields** node, in the dialog switch to **Connect** tab and then choose **Add DL Formula Fields** node in the list and choose **Append** option there



- Review the field mapping between those datasets, these should've been mapped for you as we've got the same names for each of the attributes:

The screenshot shows the 'Append 0' node configuration in a data transform workflow. The left pane displays the 'Data to Append' section with two sub-options: 'Add DL Formula Fields' and 'Add AIR Formula Fields'. Below these is a note about mapping node column names to batch data transform column names. A 'Map all columns' toggle switch is shown. The right pane shows a preview of the appended data in a table format. The table has columns: A<sub>3</sub> Customer, A<sub>3</sub> Identification Number, # Unique Identification Number, A<sub>3</sub> Party Identification ID, A<sub>3</sub> Party Identification Type, and A<sub>3</sub> Identification Name. The data rows show various entries such as '003al000009E7TAAK\_O90-32062748' and '003al000009E7TAAK\_I80-66214240'. The preview table also includes a header row with column names.

- Now repeat the approach connecting newly configured **Append 0** node with **Add MC Formula Fields** node
- Then repeat the process for **Append 1** and last **Add RAVG Formula Fields** node
- Feel free to use **Clean up nodes** button to clean up layout. The result of the configurations should look as follows:



It's time to save your progress and return to the Data Transforms list.

In the real implementation next logical step would be to set the schedule for the data transform to run. For the purpose of our exercise let's initiate manual run to validate the

results of the processing. Using  in the list view choose **Run Now** option from the dropdown:

	Data Transform Name ↑	Data Space	Data Transfor...	Type	Last Run Status	Last Run Time	
1	<input type="checkbox"/> Extract Party Identifications from Rentals (Batch)	Active	Batch	None			<div style="border: 1px solid #ccc; padding: 5px;"><span>Run Now</span> <span>View</span> <span>Run Now</span> <span>Schedule</span> <span>Delete</span> <span>Update Status</span></div>

Once the process completes (you'll notice **Last Run Status** changing from **None** to **In Progress** to **Success**) navigate to the **Data Lake Objects** tab and verify that you get **16** records

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