Flight Plan: Matillion and Snowflake Test Flight

Workshop Use Case
Flights Data Analysis: Creating Orchestration and Transformation Jobs

In this session, we will work with a publicly available Flights dataset, loading the data into Snowflake and transforming it into an analytics-ready format.

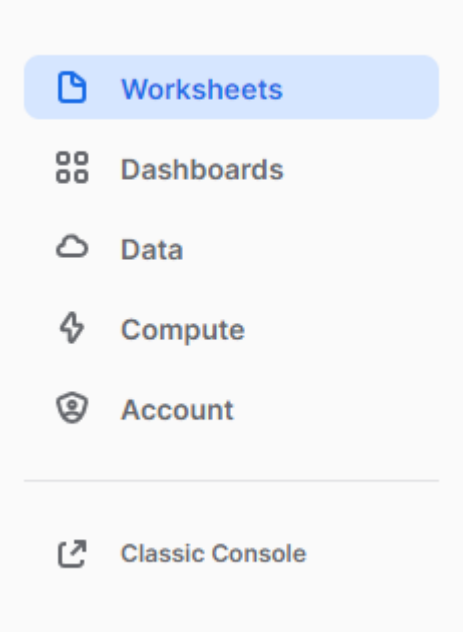
Activities in this exercise will include:

1. Resizing Snowflake data warehouse
2. Loading the Flights dataset from files into Snowflake
3. Loading Airports data from a database into Snowflake
4. Transforming the data to identify the frequency of flights between cities

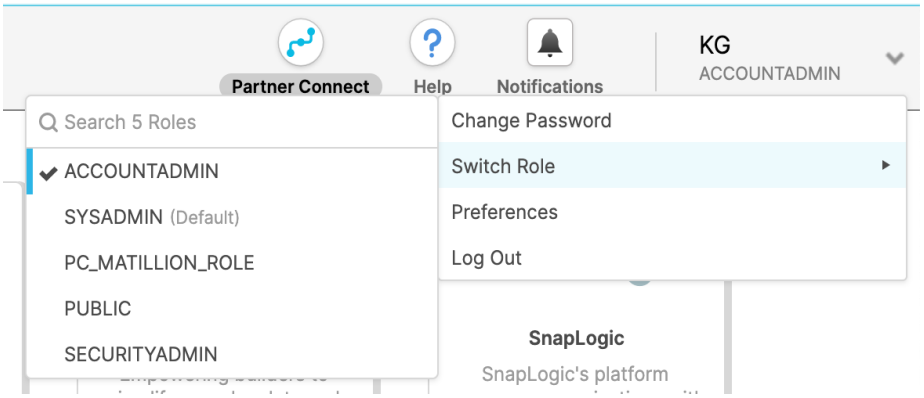
ENVIRONMENT SETUP

Please do steps 1-5 below before the Flight Plan Workshop. Steps 6 and on will be done during the workshop.

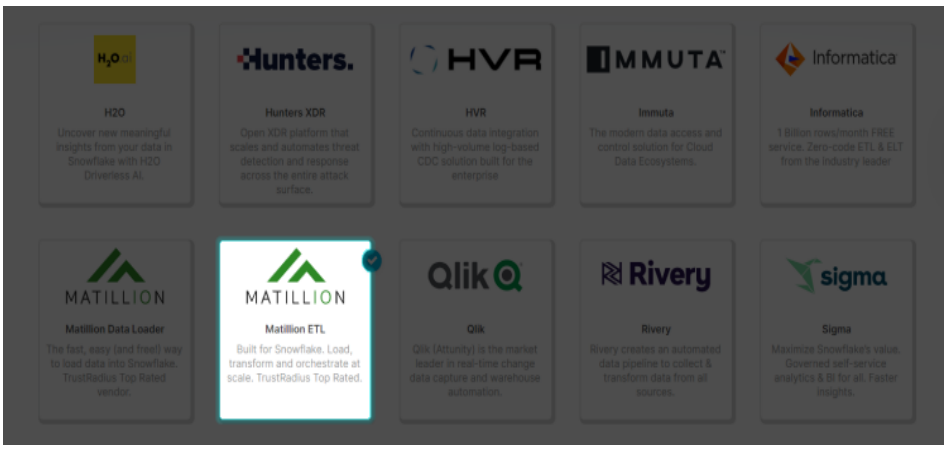
1. Log in to **Snowflake with your Snowflake account**.
   After logging in you might be taken to Snowflake's new Snowsight UI with url app.snowflake.com, in this case, you need to switch back to the Classic UI. Click **"Classic Console"** on the left of the screen to go to the Classic UI



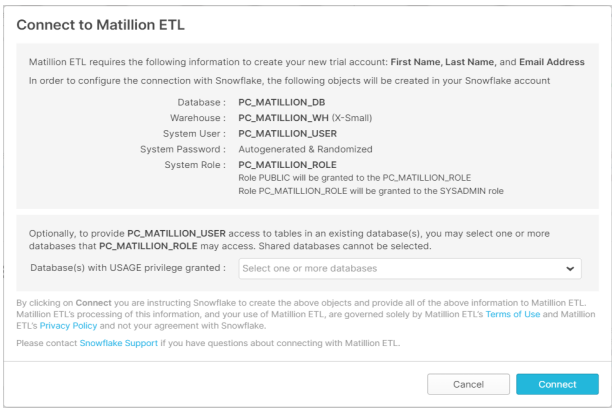
2. Ensure that the account is set to an **ACCOUNTADMIN** role. This can be done by navigating to the top-right of the page within Snowflake where the username is, clicking and specifying **Switch Role > ACCOUNTADMIN**.



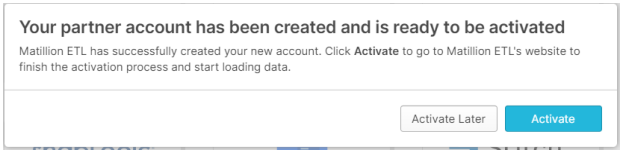
3. Navigate to the top of the page, and click on the **Partner Connect** icon. Within the applications that appear, find Matillion ETL (Do not select Matillion Data Loader) and click it.



4. A screen will appear, detailing the instance of Matillion that will be spun up for the new trial. Click **Connect**

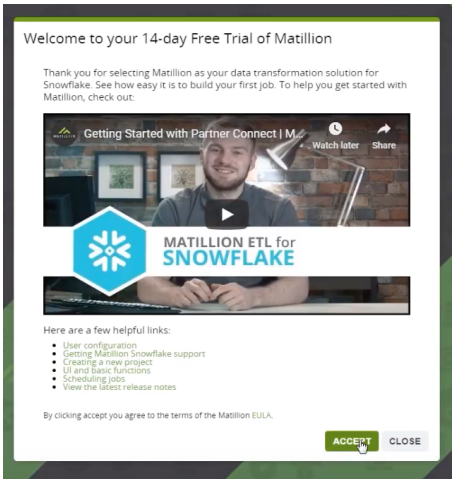


5. Then click the following **Activate** button



6. This will open a new window that will provide details about the new trial, links to useful information and there will be a YouTube video detailing how

to use Matillion with Partner Connect. Click **Accept** to proceed, which will open the Matillion instance. You will also receive an email with your Matillion credentials and a link to Matillion ETL

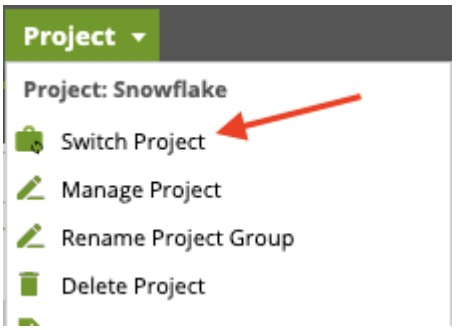


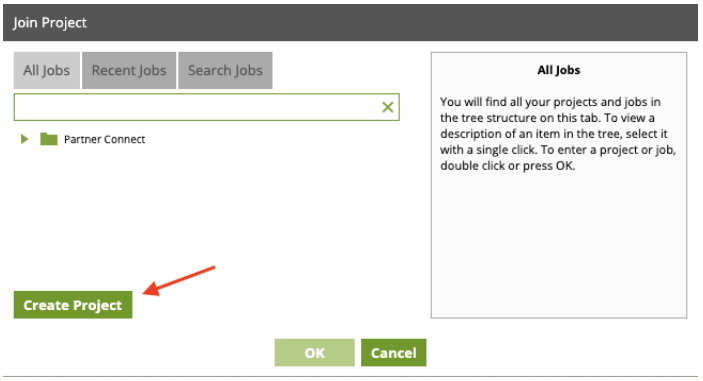
—----------

IMPORTANT: PLEASE STOP HERE UNTIL THE WORKSHOP. We will do the rest during our Flight Plan session.

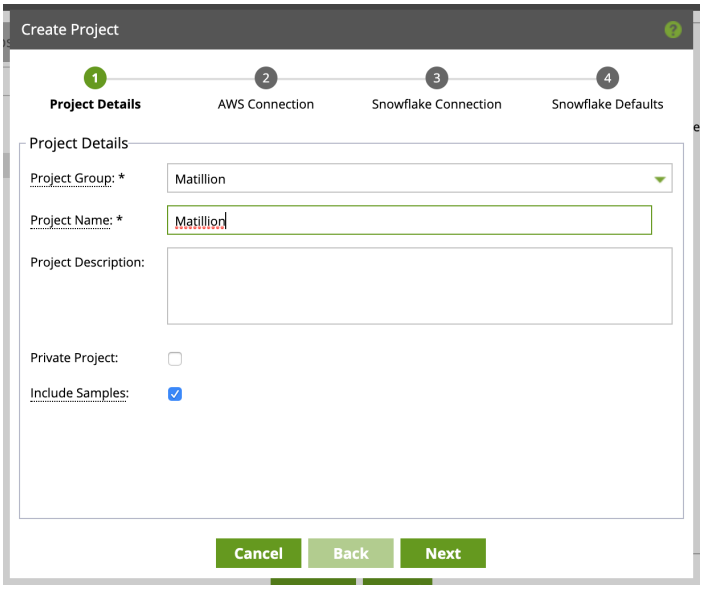
—----------

7. Find the email with the subject “Matillion ETL for Snowflake Partner Connect” in your inbox. This will contain the credentials you’ll need to log in to your Matillion instance.
8. Once logged into your Matillion ETL instance, create a new project by clicking **Switch Project** on the Project menu to the upper left of the screen. Then click the Create Project button.

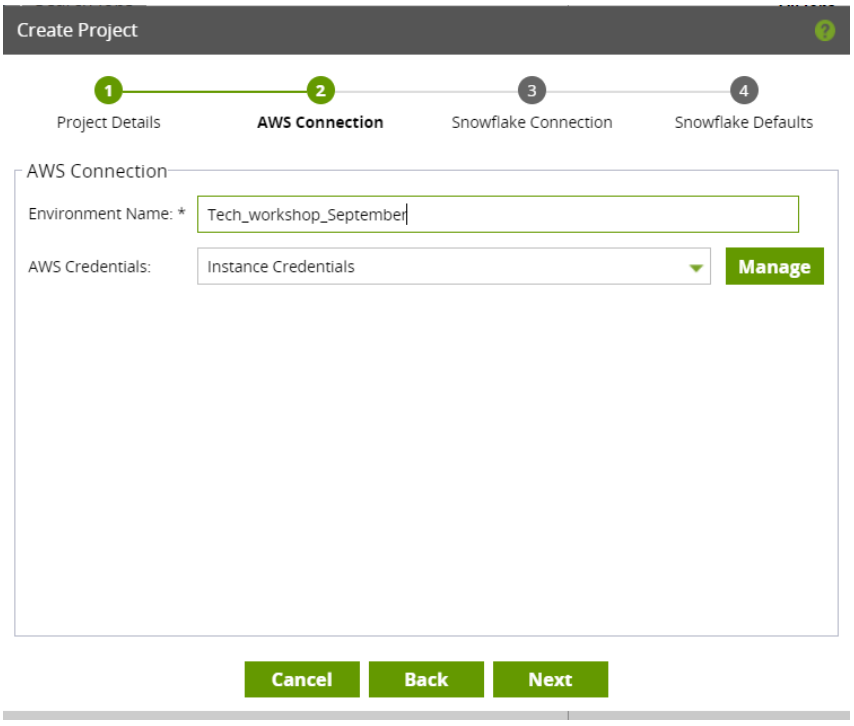




9. Enter **Matillion** as the new name for the Project Group and Project Name. This can be anything that the user chooses. Click **Next** to go to the AWS Connection screen.



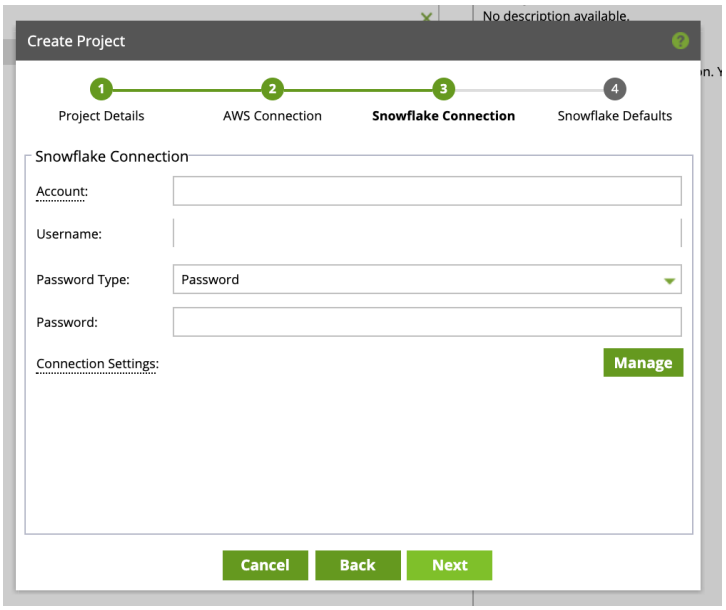
10. Within the AWS Connection screen, click **Next**. Enter **Tech_workshop_{current month}** for Environment Name and click **Next** to go to the Snowflake Connection screen.



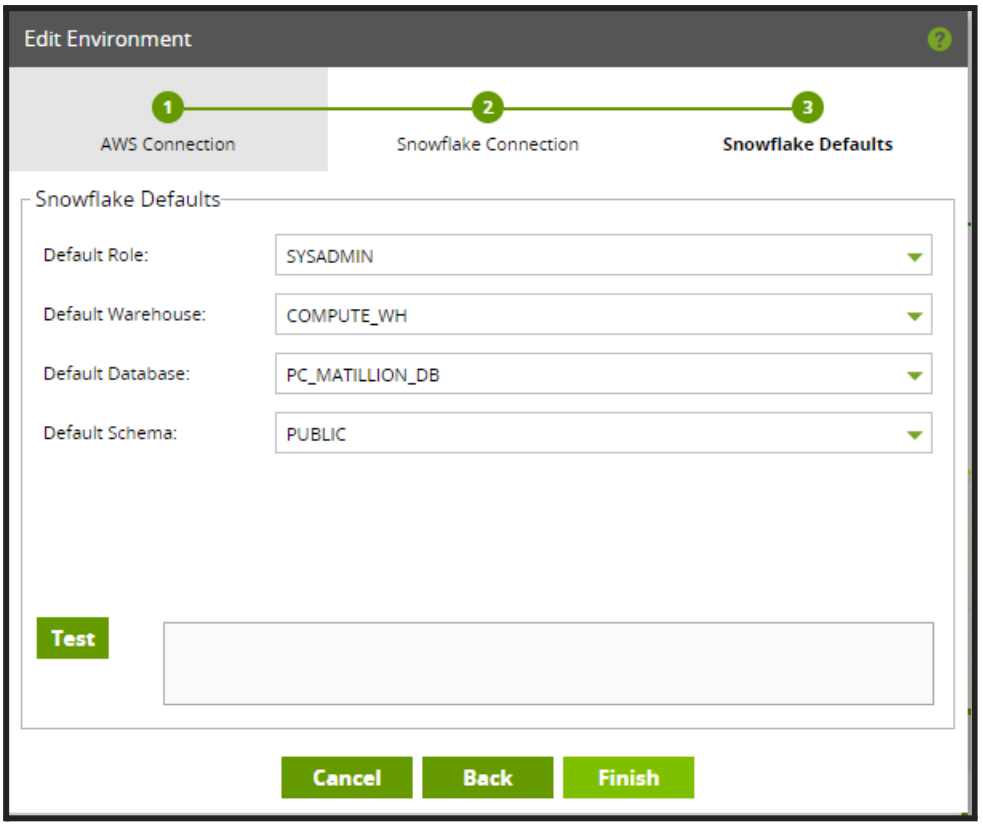
11. Within the Snowflake connection screen, enter the credentials that would be used to log in to the Snowflake instance.
    - Take note of the tooltip that states you need to leave the snowflakecomputing.com portion off the **Account** name. For

example, if your Snowflake account URL is https://DBA08909.snowflakecomputing.com/console/login, just enter DBA08909 in the Account field below

- Ensure that the "Password Type" field is set to **Password** and the appropriate password has been entered in the Password field.
- Click **Next** to go to the Snowflake Defaults screen.

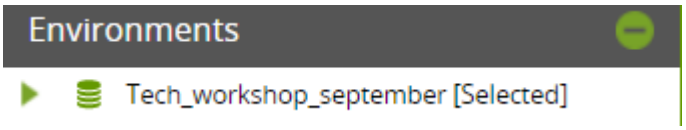


12. Within the Snowflake Defaults screen, click **each drop-down menu and select the items shown below** and see the role(s) drop down.
    - If there are no roles present, there is an issue with the credentials entered in the Snowflake Connection screen.
    - Configure the Snowflake Defaults as shown below



- Click **Test** to validate the connection to the Snowflake instance. Click **Finish** to create the project.

13. Further validate the connection in Snowflake by clicking **Environments** in the lower-left hand corner of the screen which should read as the Environment Name given in the AWS Connection screen.



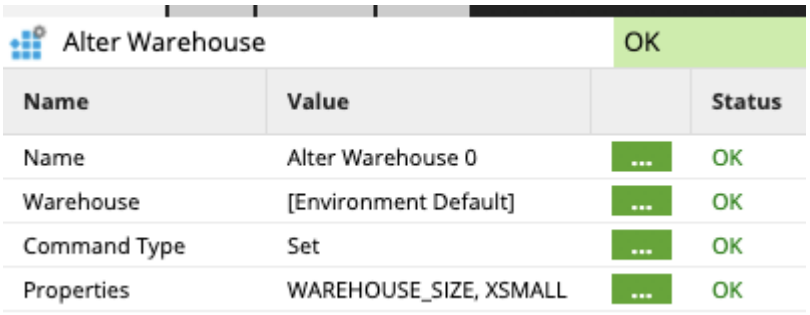
Exercises

Exercise 1

Goal 1: Learn how to use Alter Warehouse component, Data Transfer component and S3 load Generator to load flat files into Snowflake.

1. Login to METL, using the link and credentials provided to you in an email from Matillion
2. Right-click in the Project Browser and create new folder called “FlightPlan”
3. Right-click on the FlightPlan folder and create new Orchestration job called “o-FlightPlan.” Switch to the new job.
4. Using the Components menu in the left pane, search and locate the “Alter Warehouse" component and drag it across the Start component until you see the connection arrow appear between the two components.

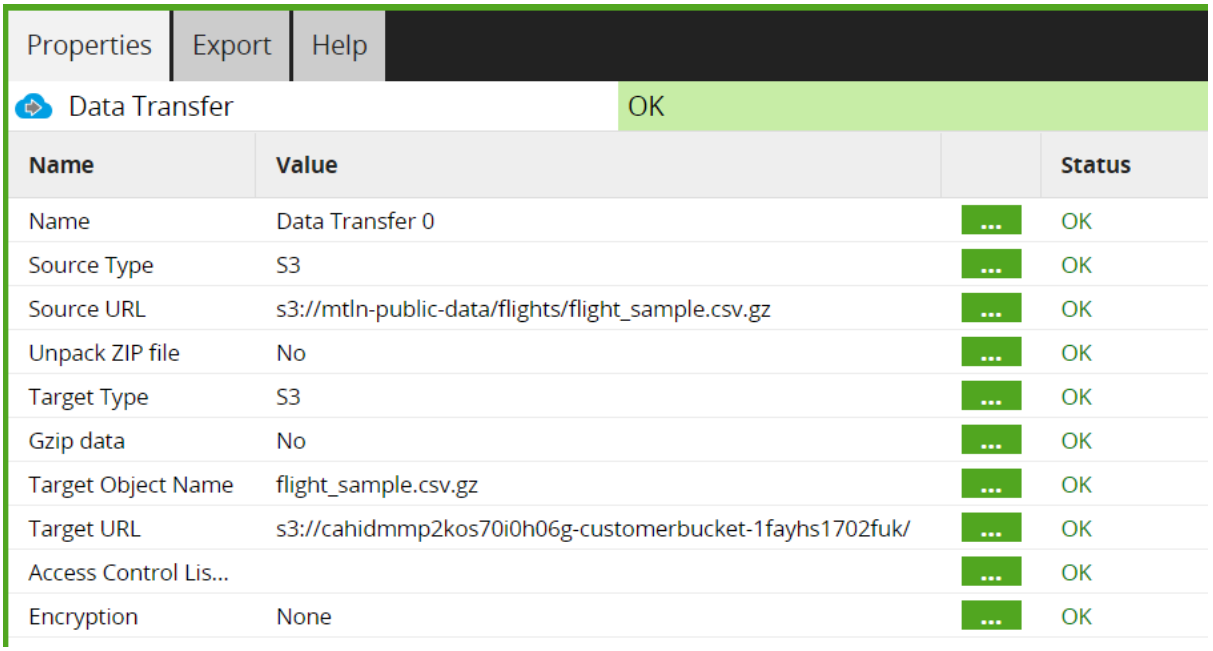
    1. Set warehouse size to XSMALL:
        1. change the command type to “Set”
        2. Edit the Properties setting and select the WAREHOUSE_SIZE property and type “XSMALL” in the value field



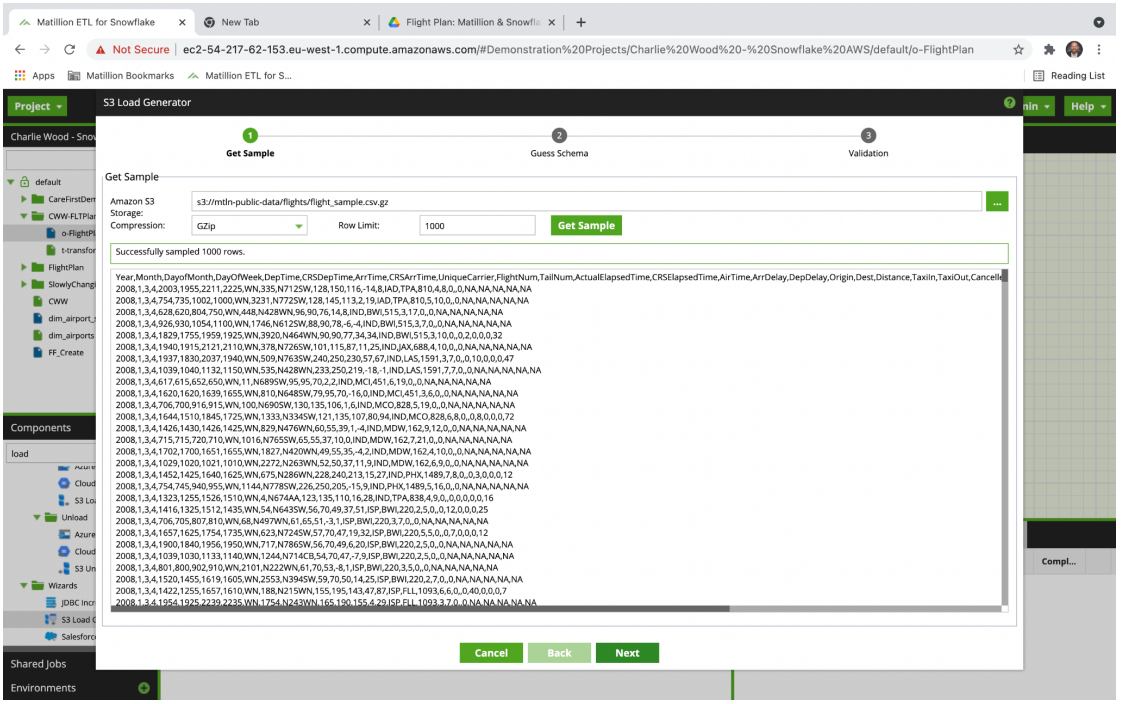
5. Using the component menu, search and locate the **‘Data Transfer’** component and drag it onto the canvas after the Alter Warehouse component

    Data Transfer Component configuration

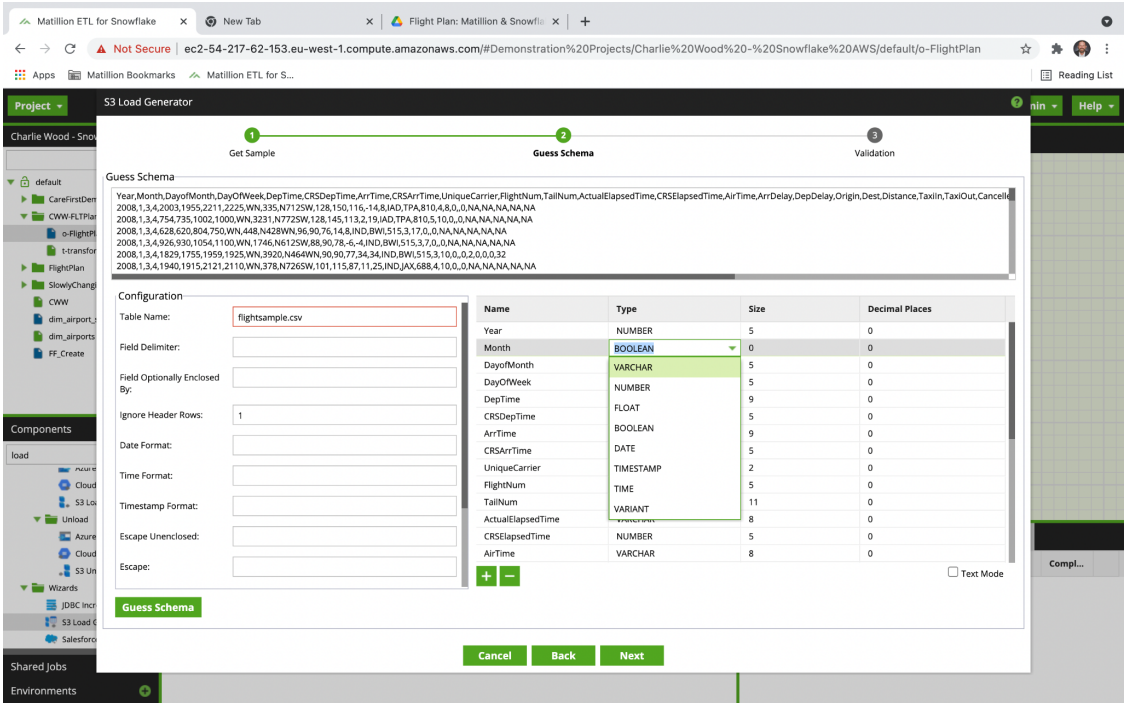
    1. Source Type: **S3**
    2. Source URL: **s3://mtln-public-data/flights/flight_sample.csv.gz**
    3. Target Type: **S3**
    4. Target Object Name: **flight_sample.csv.gz**
    5. Target URL: ***select the default bucket that comes with the Partner Connect instance, it should be visible when you open the selector e.g. s3://*******-customerbucket-*******/***



6. Once the Data Transfer component is in a valid state (green outline) Right click on the Data Transfer component and select “Run Component”
7. Using the component menu, search and locate the ‘S3 Load generator’ and drag on to the canvas
    1. S3 Load Generator Component configuration
        1. S3 bucket location: ***locate the flight_sample.csv.gz file in the default bucket that comes with the Partner Connect instance, it should be visible when you open the selector e.g. s3://********-customerbucket-*******/flight_sample.csv.gz***
        2. Set compression type to GZip
        3. Set Row limit to 1000 - for max read depth and more consistent table builds
        4. Click Get Sample and click next



5. Review the table as created by METL and modify any data types that did not translate.
    - Typically, this occurs on 0 byte columns where the data type is set to Boolean. There are 3 Boolean data types. Change all boolean to VARCHAR and size to 2. Review and confirm the changes have been applied.



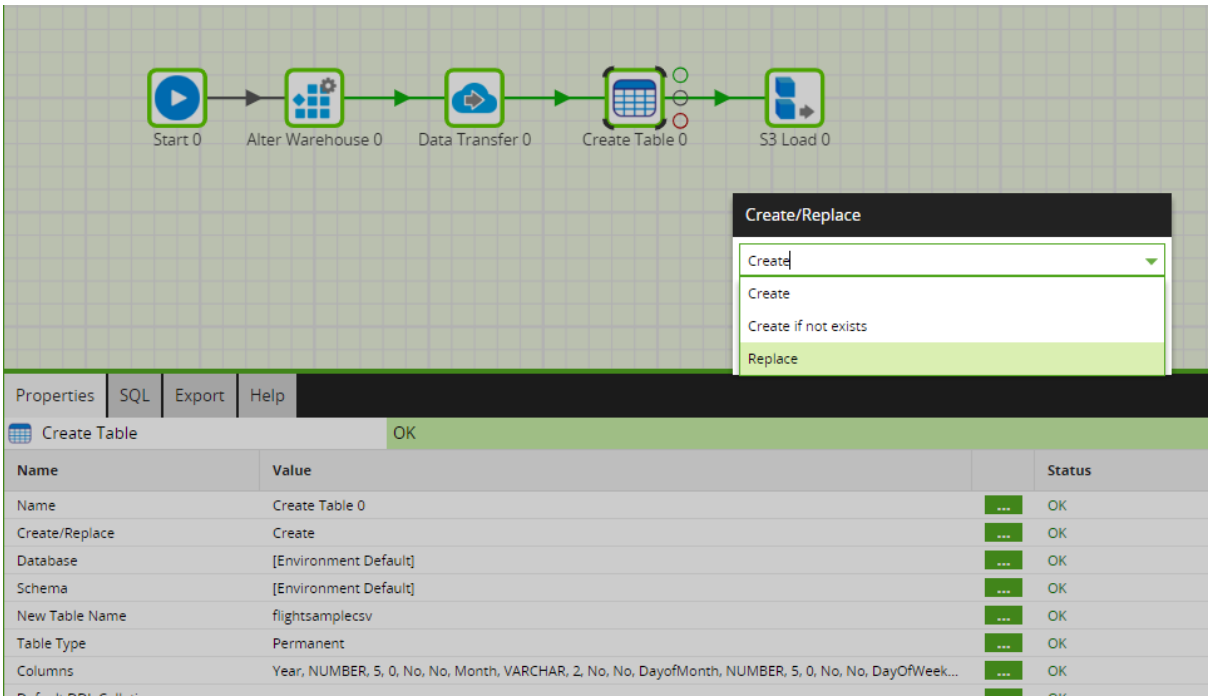
6. Now change the table name to “flight_sample”

7. Change field delimiter to a comma (“,”) and hit NEXT
8. On the last screen you can test your build and then Click on Create and Run

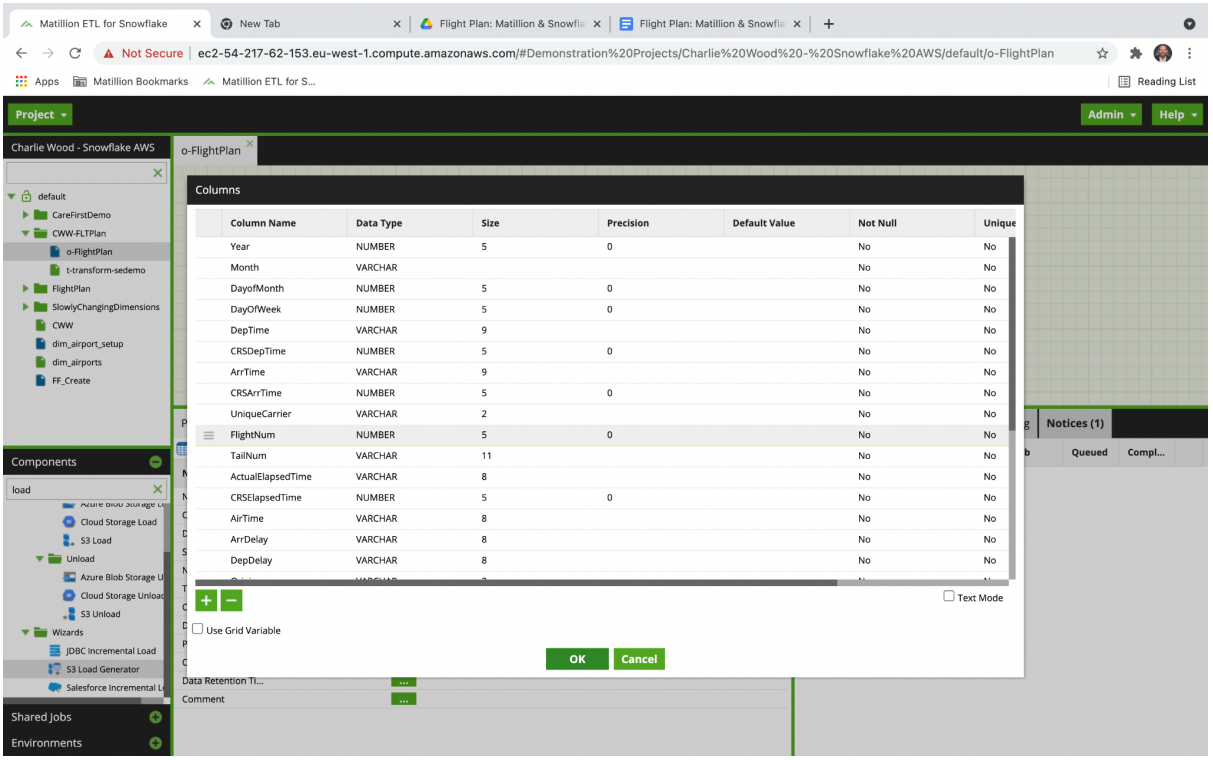
Once complete METL will drop two components onto the canvas for configuration. The first is Create Table and the second is S3 Load. Let’s modify the create table and then the S3 load.

Goal 2: Understand table creation and Snowflake table properties.

1. Note the Create/Replace option and select “Replace”; this will make your job restartable. You can modify your environment at this point if you wish to point to a different CDW or schema but for now let’s leave as a default.



2. Open the Columns property and review column for data types (change any BOOLEAN datatypes to VARCHAR)



Now that you have configured the job go ahead and right click on the canvas and select the Run Job option. This will run all 3 components including Altering the Warehouse, Creating the table in Snowflake and loading the data into this Snowflake table directly from S3.

Exercise 2

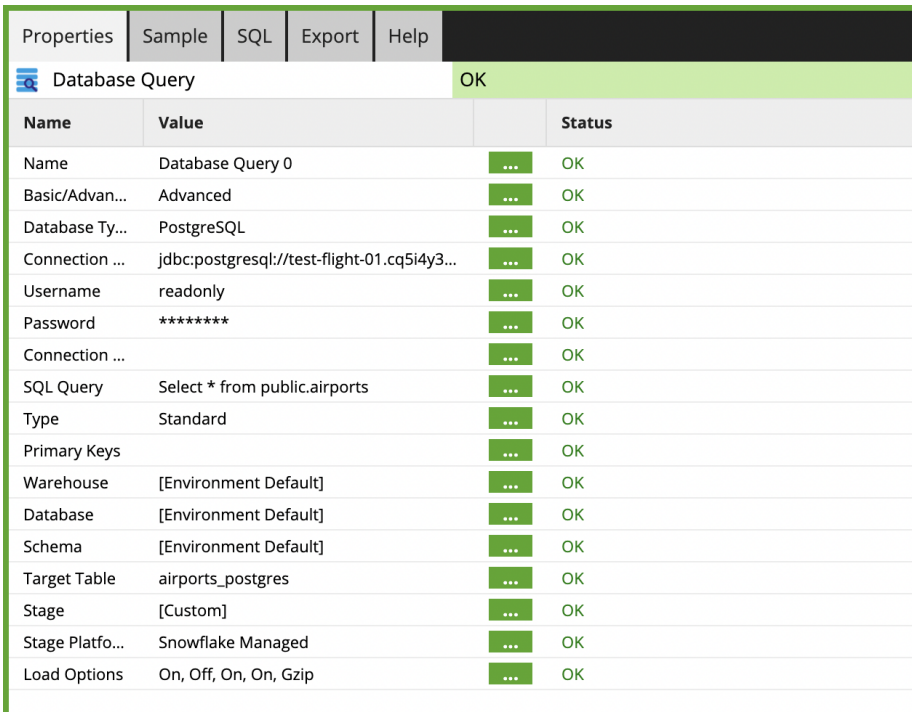
Load data from a database

Goal: Understand how to configure a Database Query component to extract data from a PostgreSQL database and load the results into Snowflake

1. Search for 'Database Query' Component panel and drag it onto the workspace. Click on the S3 Load component, move your mouse to the very top circle, click, drag and connect to the Database Query component
2. Configure
    a. Database Type: PostgreSQL
    b. Connection URL:

jdbc:postgresql://test-flight-library-psql.c4ttnsr9vh61.eu-west-1.rds.amazonaws.com/testflight

c. Username: readonly
d. Password: metl1234 (use the “Store in component” option)
e. SQL Query: select * from public.airports
f. Leave Warehouse, Database, and Schema as [Environment Default]
g. Target Table: airports_postgres
h. Stage Platform: Snowflake Managed



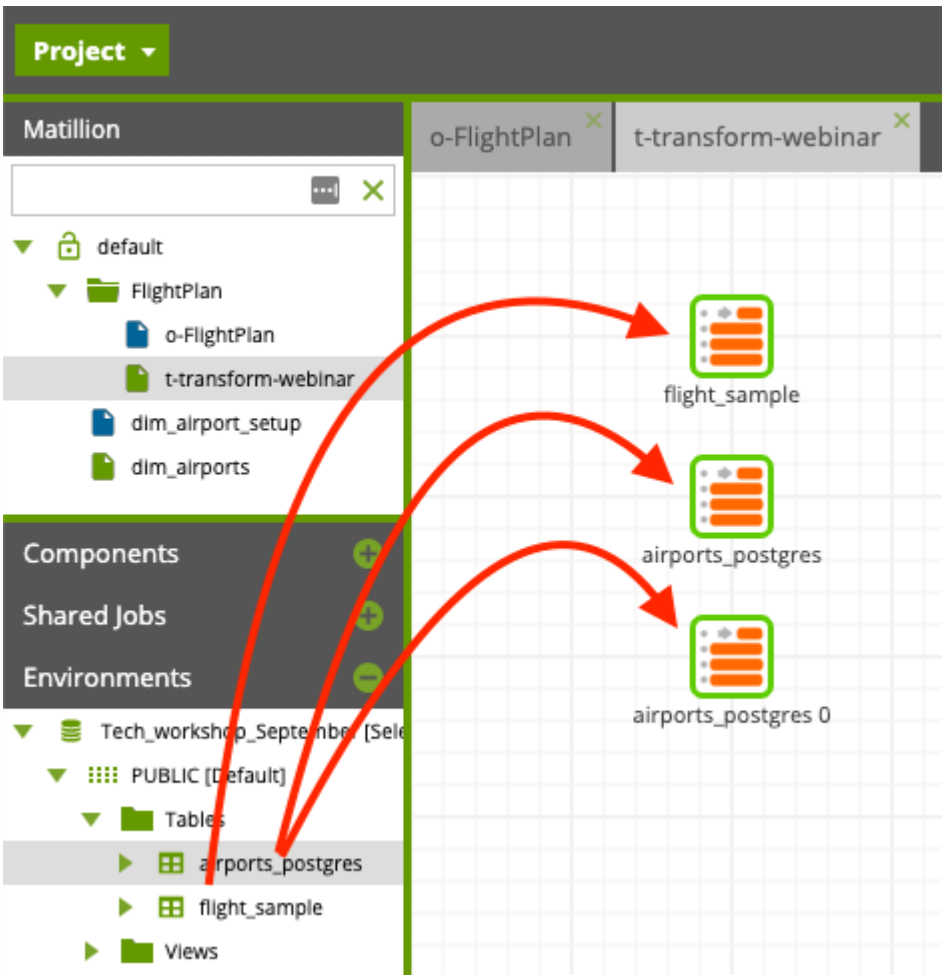
3. Run the Database Query component by right clicking on the component and selecting the first option “Run Component”

Exercise 3

Goal: Build a Transformation Job that joins and aggregates our Flight and Airport data

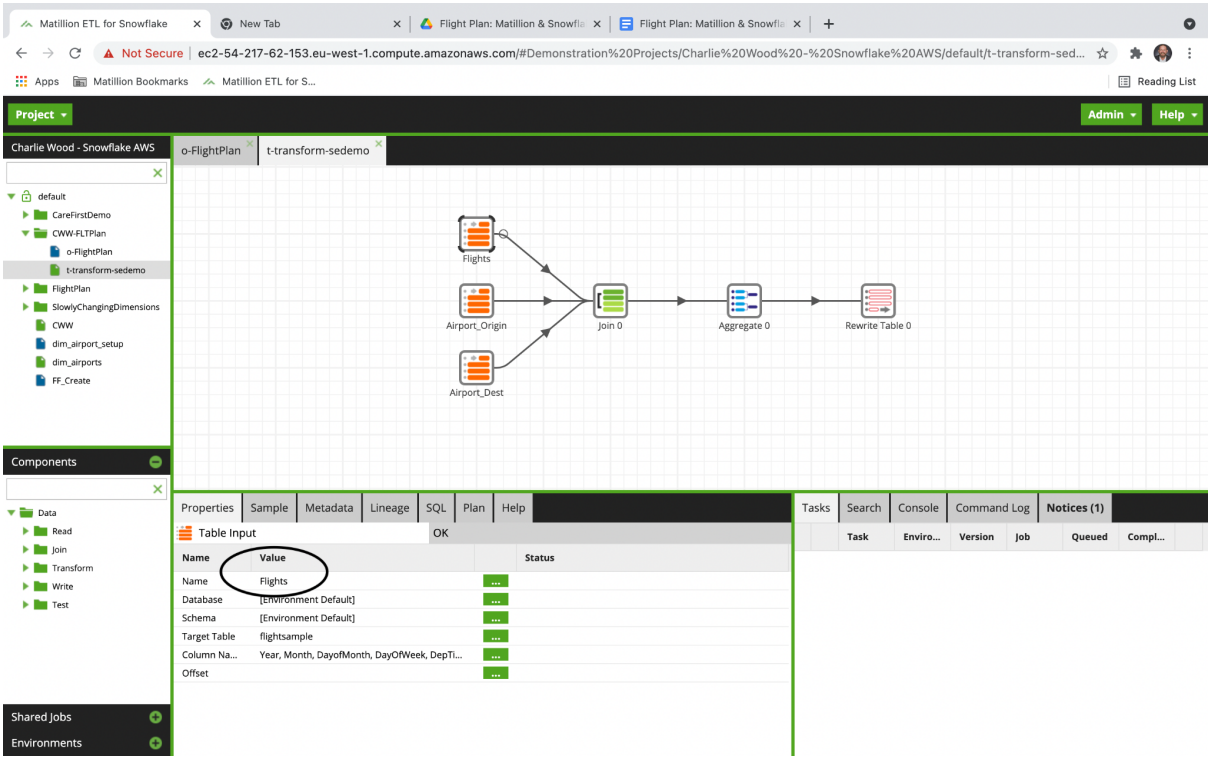
The marketing department wants to understand which routes between which cities are the most popular. This can be done by comparing the number of flights between the origin and destination cities and aggregating the data.

1. Right-Click on the FlightPlan folder in the top-left pane and create a Transformation Job named it “t-transform-sedemo”. Switch to the new job.
2. Using the Environments browser in the bottom left pane, drag and drop “flight_sample” once and “airports_postgres” table twice onto the canvas.

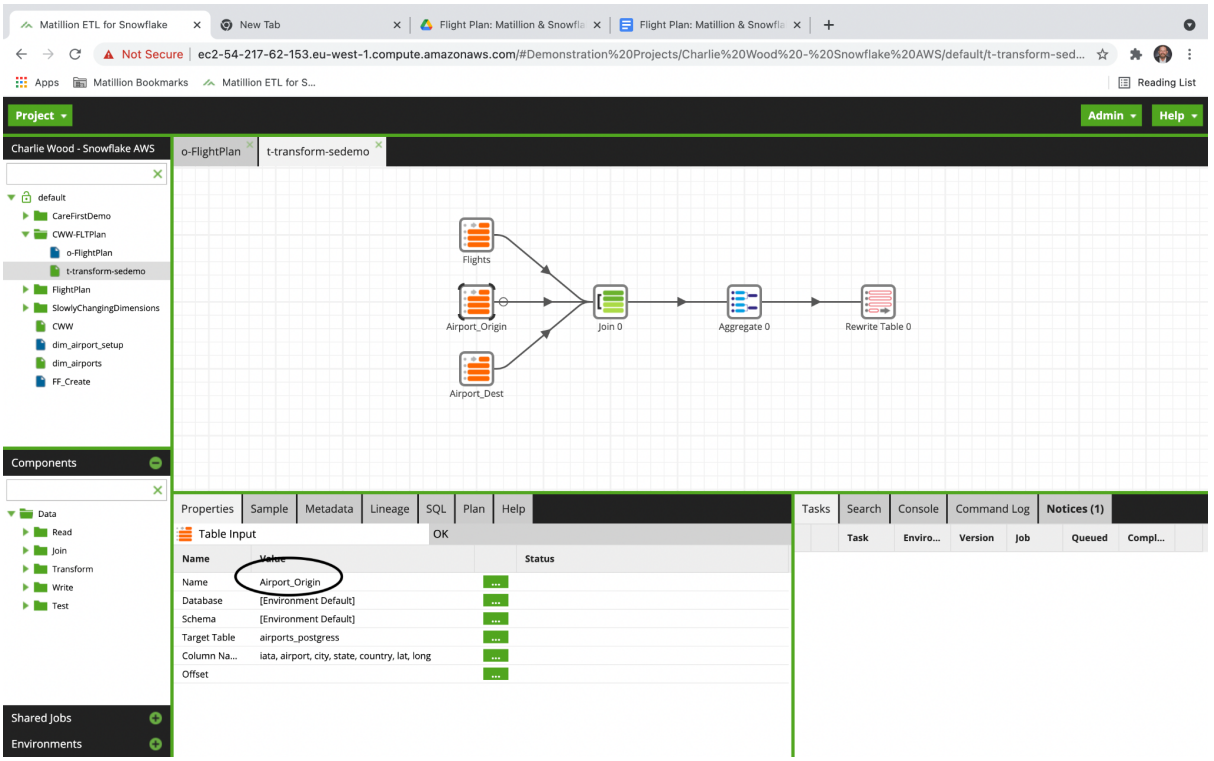


3. Name the components as shown below

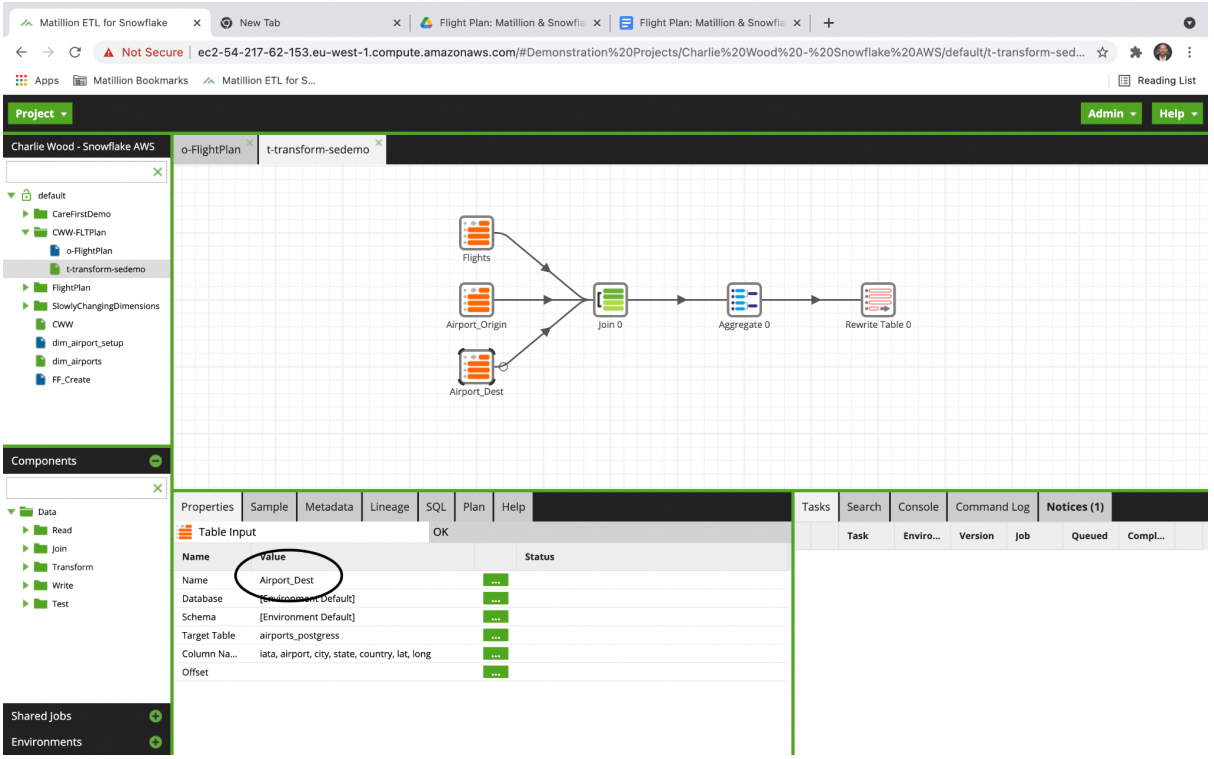
Flights



Airport_Origin

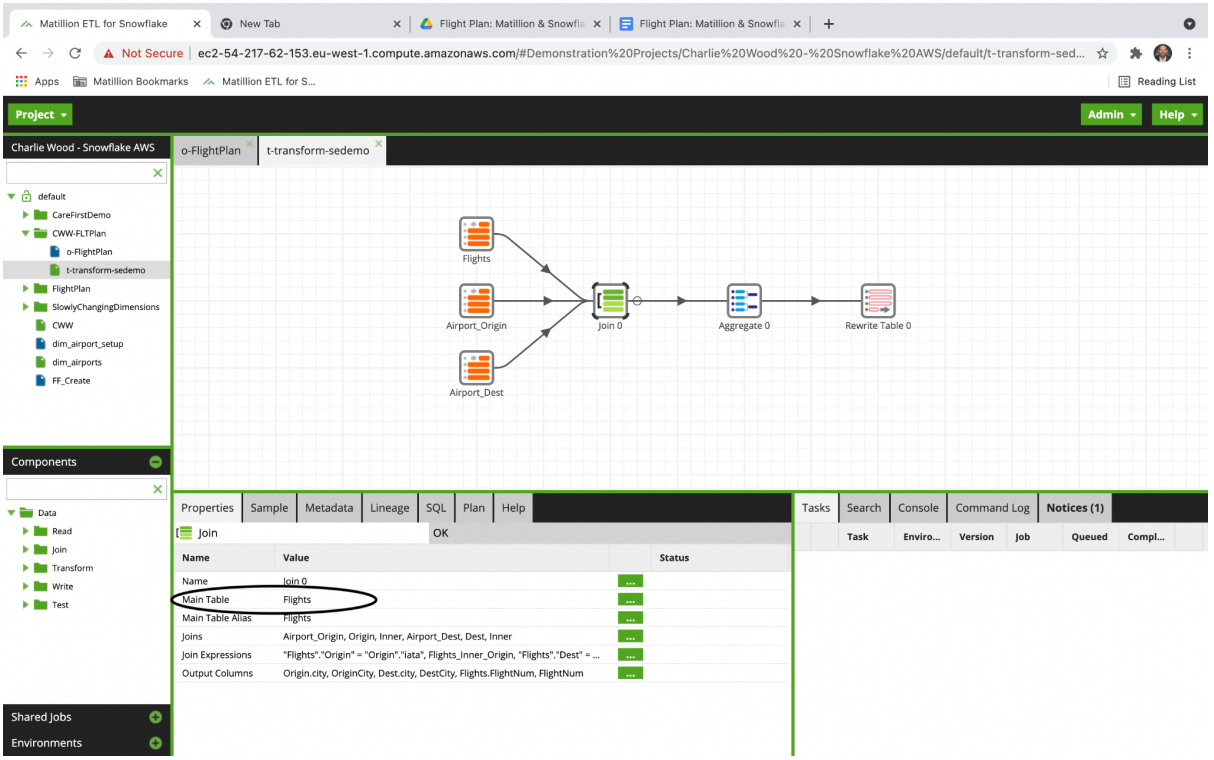


Airport_Dest

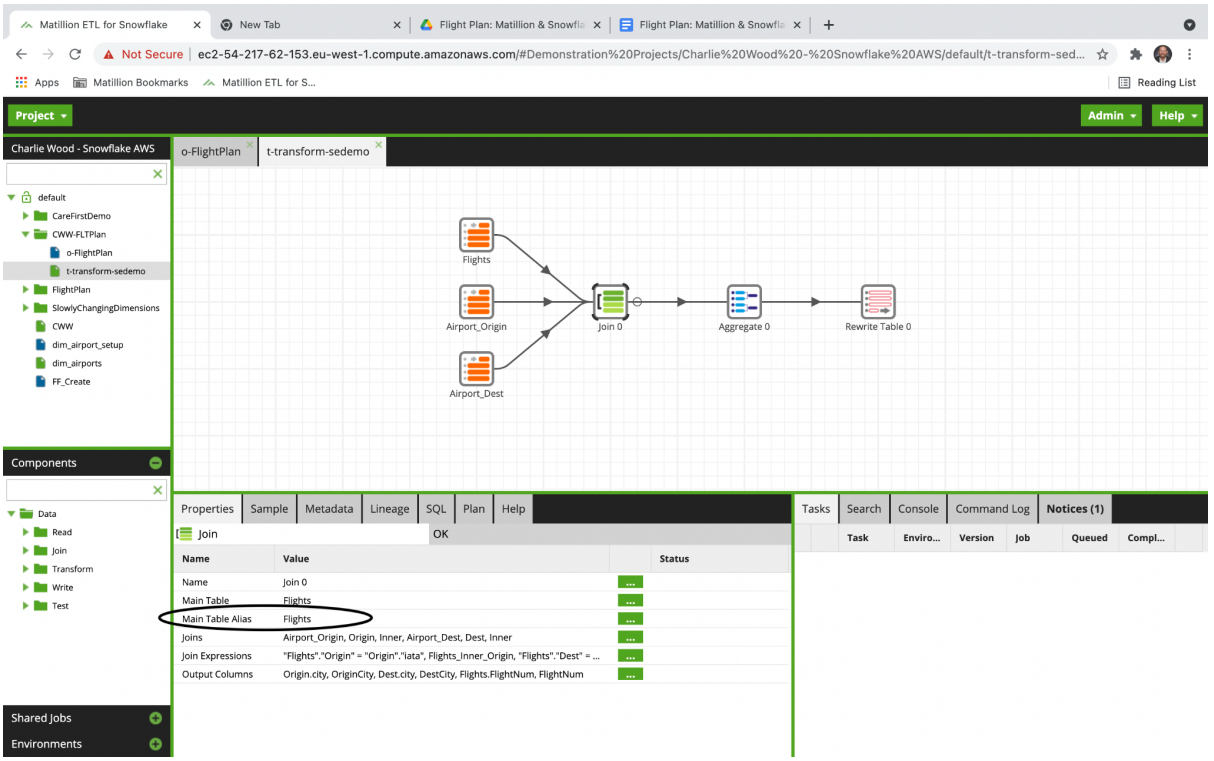


4. Using the Components palette, search for a Join component and drag it onto the canvas.
5. Connect all three Table Input components to the Join component
6. Configure the following properties on the Join component

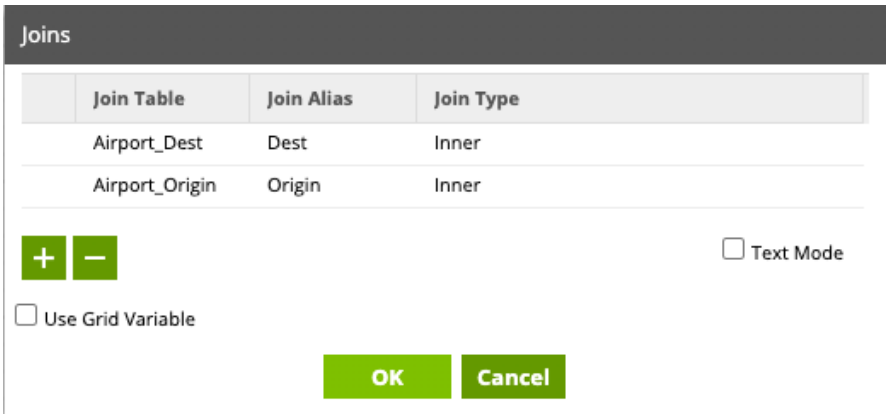
Main Table: Flights



Main Table Alias: Flights



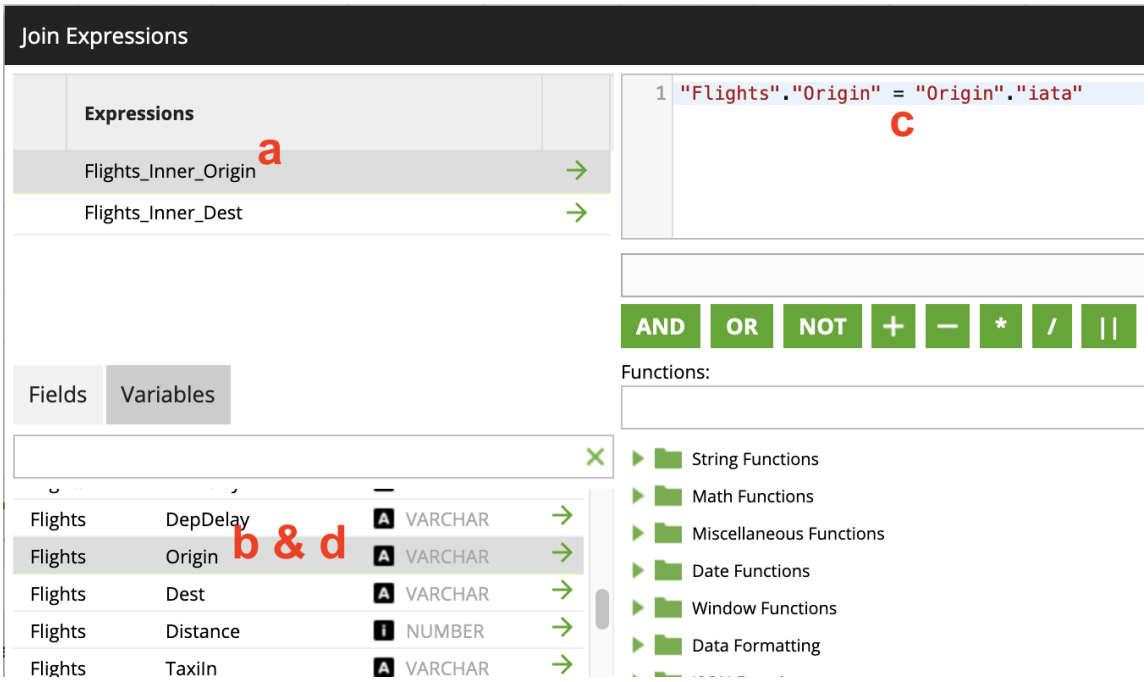
7. Use the “+” in the lower left of the “Joins” dialog to add the joins. Configure shown below



8. Open the Join Expressions dialog and the do the following for each Join:
    a. Select the Expression of interest in the upper left
    b. Double-click the "Flights"."Origin" or "Flights"."Dest" field in the field selector to the lower left
    c. Add an "=" sign to the expression
    d. Double-click the "Origin"."iata" or "Dest"."iata" field in the field selector to the lower left

    Join Expressions should appear similar to those listed below:

Expressions	**Expression Definition**
Flights_Inner_Origin	"Flights"."Origin" = "Origin"."iata"
Flights_Inner_Dest	"Flights"."Dest" = "Dest"."iata"



9. Add the Output Columns as shown below

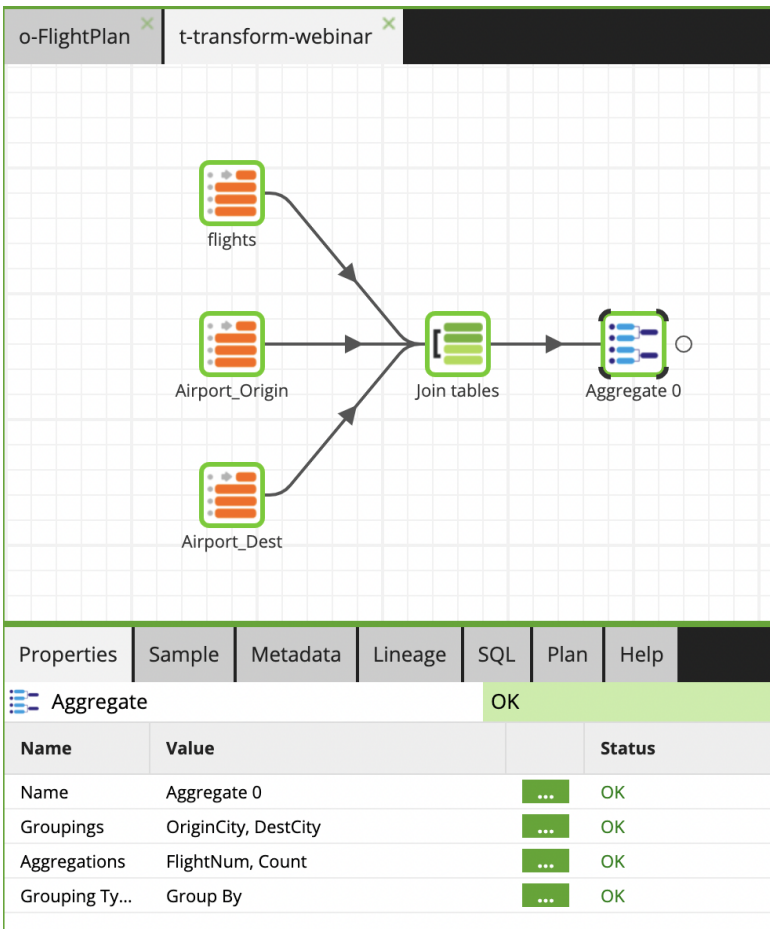
Output Columns

Input Column	Output Column
Origin.city	OriginCity
Dest.city	DestCity
Flights.FlightNum	FlightNum

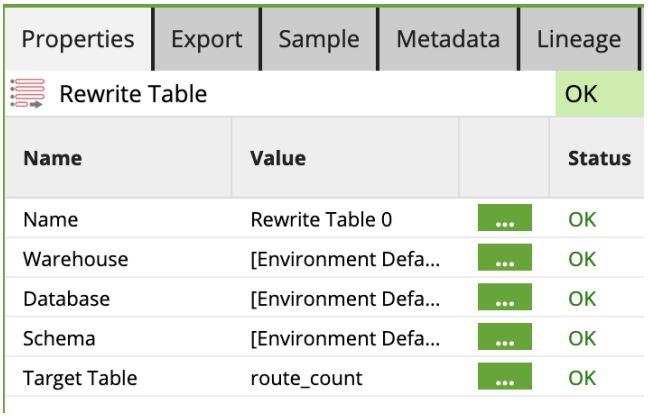
10. Search for an Aggregate component in the Component Palette. Drag one onto the canvas and connect it after the Join component
11. Add “OriginCity” and “DestCity” to the “Groupings” property
12. For the “Aggregations”, set the “Source Column” to “FlightNum” and the “Aggregation Type” to “Count”

Aggregations

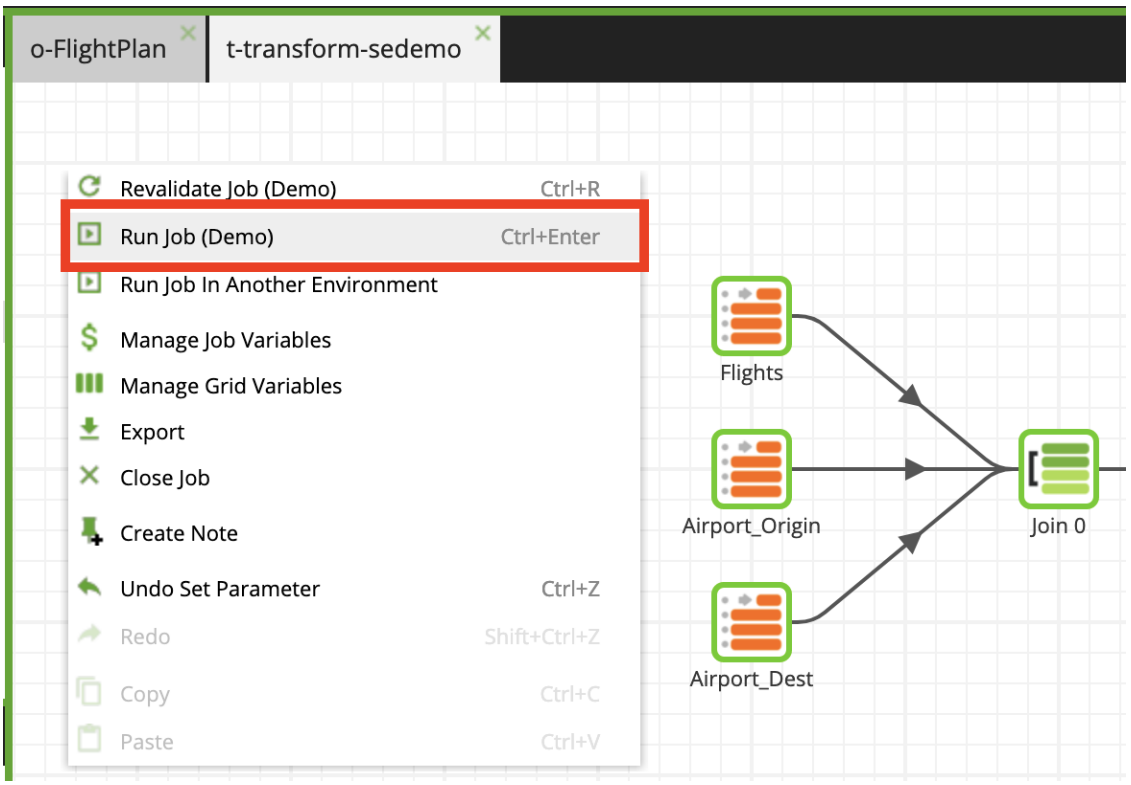
Source Column	Aggregation Type
FlightNum	Count



13. Finish the transformation by dragging a “Rewrite Table” component onto the canvas from the components palette and connect to the Aggregate component. Set the “Target Table” to “route_count”



14. Run the job by right-clicking on the canvas grid and selecting “Run Job”



15. Last, drag the “t-transform-sedemo” Transformation Job onto the end of the “o-FlightPlan” Orchestration job to create an end-to-end ELT job