**Step 0 | Create Project**

* Create a new empty Dataiku project using the “New Project” button and give it a descriptive name.



**Step 1 | Import necessary tables.**

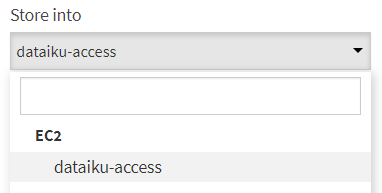
* Import\* the 8 data tables
  + Flight Data Tables
    - **av\_engine\_data\_aic\_psql**
    - **av\_engine\_data\_axm\_psql**
    - **av\_engine\_data\_fron\_psql**
    - **av\_engine\_data\_pgt\_psql**
  + Supporting Data Tables
    - **av\_manufacturing\_supply\_chain\_psql**
    - **av\_bom\_manufacturing\_psql**
    - **av\_esn\_rul\_psql**
    - **av\_lkp\_airport\_codes\_t\_psql**
* Importing the data sets creates a copy from the remote database into your local Dataiku session
* For help with importing a data set, see the video titled: “Importing Data”

**Step 2 | Union all flight tables to consolidate data.**

* Single click on the **av\_engine\_data\_aic** table, then either use the Visual Prepare Recipe or SQL recipe to create a formula to overwrite the column t24 with the new value of (t24 + 459.67).



* + The reason we are doing this to only one table, is because this airline stored their t24 column in Rankine, the other airlines kept their temps in a standard format.
  + For help with creating a column with a formula, see the video titled: “Creating Columns”
  + **Important:** whenever you are creating a new table, be sure to store your resulting data in the EC2 “dataiku-access” connection!



* Either use a code recipe (UNION\* ALL in SQL) or the visualize Stack Recipe to combine the 4 **Flight** datasets into a single table.



* + Union combines the 4 datasets into 1 dataset with all elements from the 4 datasets
  + For help with UNIONs, see the video titled: “Combining Data”

+  = 

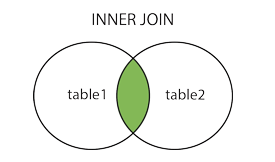
* Double click on each of the tables and use the black dropdown arrow on each column to ‘*analyze’*
  + This is helpful in gaining insight about your table.
  + For example, you can now answer the following questions (& many more) about your data:
    - Is it right-skewed or left-skewed?
    - How many unique values are in the column?
    - What is the min/max?

**Step 3 | Calculate departure & destination latitude/longitudes for each row in the consolidated flight table**

* Either user a code recipe or visual recipe to create an INNER Join on the created table from step 1 with the **av\_lkp\_airport\_codes\_t\_psql** table. We are going to be capturing the *destination* latitude and longitude for each flight.



* + Join on *destination\_icao* and *airport\_icao*
    - *ICAO is a code for an airport*
  + Important: Rename the latitude/longitude columns from the airport lookup table to be ***destination****\_latitude* and ***destination****\_longitude.*
* INNER JOINS select records that match a certain criterion (Join Condition) in bot



* Either user a code recipe or visual recipe to create an INNER Join on the created table from step 1 with the **av\_airport\_code\_lkup** table. This time, we are going to be capturing the *depart* latitude and longitude of each flight.
  + Join on depart*\_icao* and *airport\_icao*
  + Important: Rename the latitude/longitude columns from the airport lookup table to be ***depart****\_latitude* and ***depart****\_longitude*.
* For help with INNER JOINs, see the video titled: “Combining Data”

**Step 4 | Calculate each flight’s total distance and LPT temperature.**

* Use a visual Prepare recipe to calculate distance of each flight in the table modified in step 2. (Calculations provided below)
* Column Name: *distance\_between\_airport\_miles*

Formula:

* + *7917.5/2\*atan2(sqrt(pow(cos(destination\_latitude\*3.14159/180)\*sin(abs(destination\_longitude\*3.14159/180-depart\_longitude\*3.14159/180)),2)+pow(cos(depart\_latitude\*3.14159/180)\*sin(destination\_latitude\*3.14159/180)-sin(depart\_latitude\*3.14159/180)\*cos(destination\_latitude\*3.14159/180)\*cos(abs(destination\_longitude\*3.14159/180-depart\_longitude\*3.14159/180)),2)),sin(depart\_latitude\*3.14159/180)\*sin(destination\_latitude\*3.14159/180)+cos(depart\_latitude\*3.14159/180)\*cos(destination\_latitude\*3.14159/180)\*cos(abs(destination\_longitude\*3.14159/180-depart\_longitude\*3.14159/180)))*
* Column Name: *t50 (total temperature at low pressure turbine – LPT outlet)*

Formula:

* + *if(t50<1410, t50,1410+2\*(t50-1410))*
* For help with creating a column with a formula, see the video titled: “Column Creation”

**Step 5 | Build a supporting KPI table.**

* Use a code recipe or visual recipe to INNER JOIN\* the manufacturing tables to create a table that has the KPIs (Key Performance Indicators) of each part and the engine serial number (or *ESN)* they associate with.
  + Datasets used: **av\_manufacturing\_supply\_chain & av\_bom\_manufacturing**
  + Join on *PN* (Part Number) and *SN* (Serial Number)

**Step 6 | Join the new KPI table to the consolidated flights table.**

* Use an INNER JOIN to join the table that you created in Step 4 with the table you created in Step 3
* Join on *ESN*
  + ***NOTE:*** *This join may take several minutes.*

**Step 7 | Join final table with table from step 5 to get remaining useful life (RUL\*) for each engine.**

* Use a code recipe or visual recipe to INNER JOIN the table you created in step 5 with the **av\_esn\_rul** table.
* Join on *ESN*
* *RUL – shows the number of cycles remaining until an engine needs to be overhauled*
  + ***NOTE:*** *This join may take several minutes.*

**Step 8 | Export and turn in your final resulting flow**

* Export your final table, return to the InsideSherpa portal to upload it, and check your work!
* For help with exporting your finished table, see the video titled: “Exporting Data”