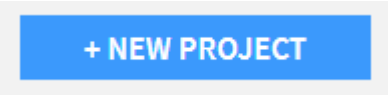
Step 0 | Create Project

- Create a new empty Dataiku project using the "New Project" button and give it a descriptive name.



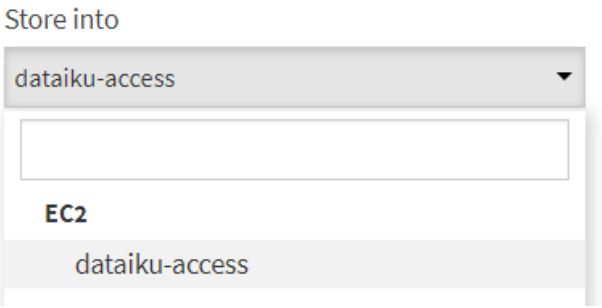
Step 1 | Import necessary tables.

- 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
- Simply go to the Dataset tab in Dataiku – NEW DATASET>filesystem>BROWSE…> "[dataset name]" then enter the name of the 8 data set tables
- 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!



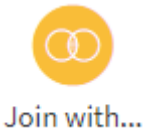
- Either use a code recipe (UNION* ALL in SQL) or the visualize Stack Recipe to combine the 4 **Flight** datasets into a single table.

    - 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"
    - The diagram below is showing the union of two datasets – note how the colours overlap to create a new colour.

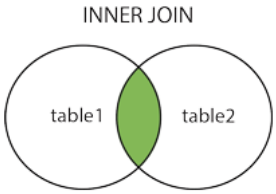
- Double click on each of the tables and use the black dropdown arrow on each column to '*analyze'*
    - Note this isn't a necessary step, however is a useful accessibility feature of Dataiku to gain insight about your table.
    - For example, you can now answer the following questions (& many more) 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* – this is done through the "prepare" visual recipe
        - INNER JOINS select records that match a certain criterion (Join Condition) in both



- 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 | 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) – note these are two separate join commands in the same visual recipe

Step 5 | 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 6 | 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 7 | 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 8 | Export and turn in your final resulting flow

- Export your final table or take a screenshot of your work, return to the Forage portal to upload it, and check your work!
- For help with exporting your finished table, see the video titled: "Exporting Data"