» Quick Start » Databricks

Databricks

Install flypipe

```
[ ]: %pip install flypipe
```

Create a temproary view representing a table

```
[36]: df = spark.createDataFrame(
          schema=("_fruit",),
          data=[
              ("ORANGE",),
              ("WATERMELON",),
              ("LEMON",),
          ]
      )

      df.createOrReplaceTempView("table")

      display(df)
```

_fruit
ORANGE
WATERMELON
LEMON

Create a graph

[35]:

```
from flypipe import node
from flypipe.datasource.spark import Spark
from flypipe.schema import Schema, Column
from flypipe.schema.types import String

import pyspark.sql.functions as F
@node(
    type="pyspark",
    dependencies=[
        Spark("table").select("_fruit").alias("df")
    ],
    output=Schema(
     Column("fruit", String(), "fruit description"),
    )
)
def clean(df):
    df = df.withColumnRenamed('_fruit', 'fruit')
    df = df.withColumn('fruit', F.lower(F.col('fruit')))
    return df

@node(
    type="pyspark",
    dependencies=[
       clean.select("fruit").alias("df")
    ],
    output=Schema(
        Column("fruit", String(), "fruit description"),
        Column("color", String(), "color of the fruit"),
    )
)
def color(df):

    replacements = {
        "orange": "orange",
        "watermelon": "red",
        "lemon": "yellow",
    }

    df = df.withColumn("color", F.col("fruit"))
    df = df.replace(list(replacements.keys()), list(replacements.values()), "color")
    return df

@node(
    type="pyspark",
    dependencies=[
       clean.select("fruit").alias("df")
    ],
    output=Schema(
        Column("fruit", String(), "fruit description"),
        Column("category", String(), "category of the fruit"),
    )
)
def category(df):

    replacements = {
        "orange": "citric",
        "watermelon": "sweet",
        "lemon": "citric",
    }

    df = df.withColumn("category", F.col("fruit"))
    df = df.replace(list(replacements.keys()), list(replacements.values()), "category")
    return df


@node(
    type="pyspark",
    dependencies=[
       color.select("fruit", "color"),
       category.select("fruit", "category")
    ],
    output=Schema(
        Column("fruit", String(), "fruit description"),
        Column("color", String(), "color of the fruit"),
        Column("category", String(), "category of the fruit"),
    )
)
def fruits(color, category):
    return color.join(category, on="fruit", how="left")
```

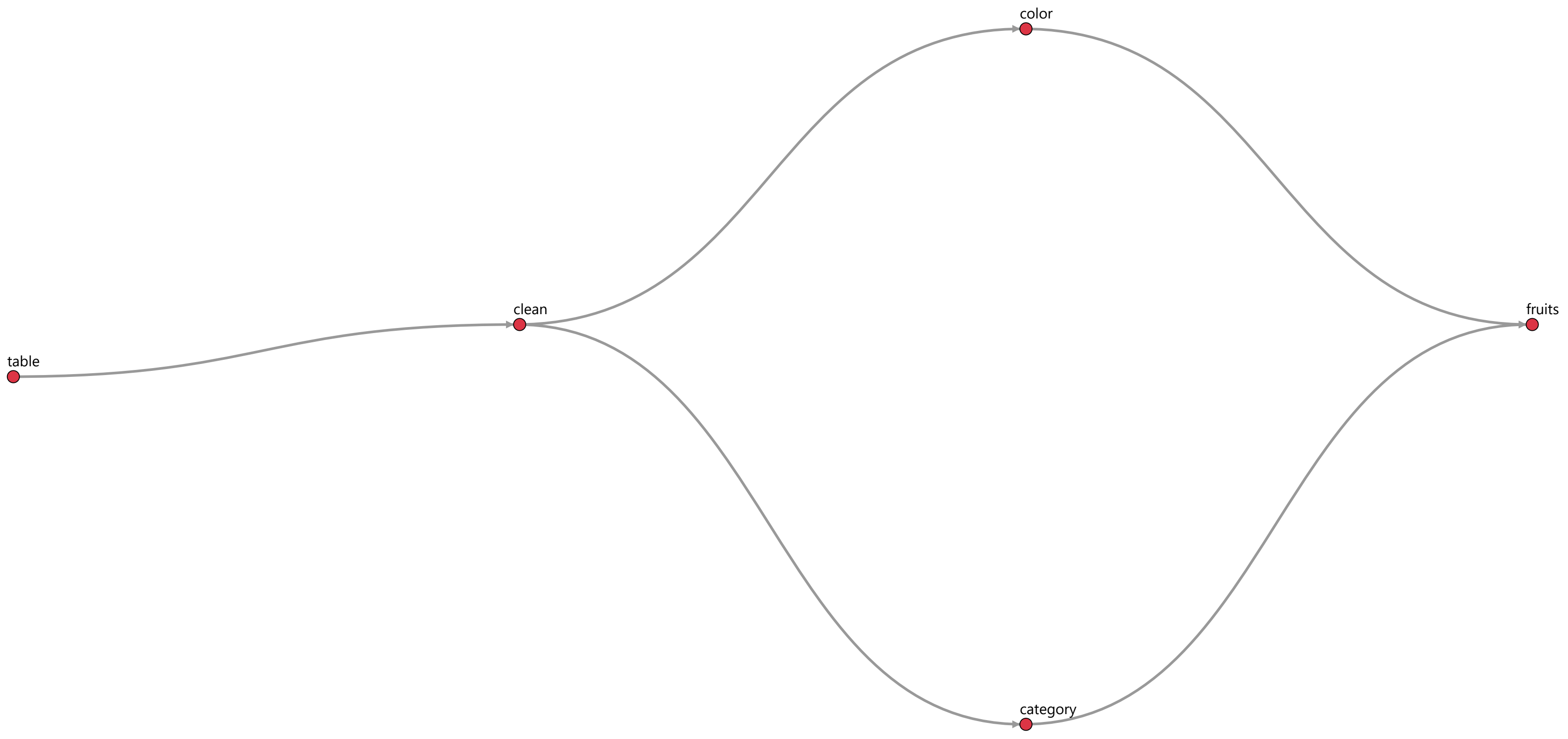
Graph

[33]:

```
displayHTML(fruits.html())
```

[33]:

Flypipe



Run

[37]:

```
df = fruits.run(spark)
display(df)
```

fruit	color	category
orange	orange	citric
lemon	yellow	citric
watermelon	red	sweet