import sys

from awsglue.transforms import \*

from awsglue.utils import getResolvedOptions

from pyspark.context import SparkContext

from awsglue.context import GlueContext

from awsglue.job import Job

from awsglue.dynamicframe import DynamicFrame

from pyspark.sql import functions as SqlFuncs

args = getResolvedOptions(sys.argv, ["JOB\_NAME"])

sc = SparkContext()

glueContext = GlueContext(sc)

spark = glueContext.spark\_session

job = Job(glueContext)

job.init(args["JOB\_NAME"], args)

CustomerTrusted\_node1 = glueContext.create\_dynamic\_frame.from\_options(

format\_options={"multiline": False},

connection\_type="s3",

format="json",

connection\_options={

"paths": ["s3://stedi-supraja/customer/trusted/"],

"recurse": True,

},

transformation\_ctx="CustomerTrusted\_node1",

)

AccelerometerTrusted\_node1 = glueContext.create\_dynamic\_frame.from\_options(

format\_options={"multiline": False},

connection\_type="s3",

format="json",

connection\_options={

"paths": ["s3://stedi-supraja/accelerometer/trusted/"],

"recurse": True,

},

transformation\_ctx="AccelerometerTrusted\_node1",

)

JoinCustomer\_node1 = Join.apply(

frame1=CustomerTrusted\_node1,

frame2=AccelerometerTrusted\_node1,

keys1=["email"],

keys2=["user"],

transformation\_ctx="JoinCustomer\_node1",

)

DropFields\_node1 = DropFields.apply(

frame=JoinCustomer\_node1,

paths=["x", "y", "user", "timeStamp", "z"],

transformation\_ctx="DropFields\_node1",

)

DropDuplicates\_node1 = DynamicFrame.fromDF(

DropFields\_node1.toDF().dropDuplicates(),

glueContext,

"DropDuplicates\_node1",

)

CustomerCurated\_node2 = glueContext.write\_dynamic\_frame.from\_options(

frame=DropDuplicates\_node1,

connection\_type="s3",

format="json",

connection\_options={

"path": "s3://stedi-supraja/customer/curated/",

"partitionKeys": [],

},

transformation\_ctx="CustomerCurated\_node2",

)

job.commit()