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 awsglue import DynamicFrame

import gs\_derived

from pyspark.sql import functions as SqlFuncs

def sparkSqlQuery(glueContext, query, mapping, transformation\_ctx) -> DynamicFrame:

for alias, frame in mapping.items():

frame.toDF().createOrReplaceTempView(alias)

result = spark.sql(query)

return DynamicFrame.fromDF(result, glueContext, transformation\_ctx)

def sparkAggregate(

glueContext, parentFrame, groups, aggs, transformation\_ctx

) -> DynamicFrame:

aggsFuncs = []

for column, func in aggs:

aggsFuncs.append(getattr(SqlFuncs, func)(column))

result = (

parentFrame.toDF().groupBy(\*groups).agg(\*aggsFuncs)

if len(groups) > 0

else parentFrame.toDF().agg(\*aggsFuncs)

)

return DynamicFrame.fromDF(result, glueContext, transformation\_ctx)

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

sc = SparkContext()

glueContext = GlueContext(sc)

spark = glueContext.spark\_session

job = Job(glueContext)

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

# Script generated for node insurance input

insuranceinput\_node1683401737069 = glueContext.create\_dynamic\_frame.from\_options(

format\_options={

"quoteChar": '"',

"withHeader": True,

"separator": ",",

"optimizePerformance": False,

},

connection\_type="s3",

format="csv",

connection\_options={

"paths": [

"s3://chaitealatte/Input/caFimaNfipInsuranceRecordLevel2009to2022.csv"

],

"recurse": True,

},

transformation\_ctx="insuranceinput\_node1683401737069",

)

# Script generated for node weather input

weatherinput\_node1683356733176 = glueContext.create\_dynamic\_frame.from\_options(

format\_options={

"quoteChar": '"',

"withHeader": True,

"separator": ",",

"optimizePerformance": False,

},

connection\_type="s3",

format="csv",

connection\_options={

"paths": [

"s3://chaitealatte/Input/caNoaaStationDayLevelWeather2005to2023\_v3.csv"

]

},

transformation\_ctx="weatherinput\_node1683356733176",

)

# Script generated for node Flood input

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

format\_options={

"quoteChar": '"',

"withHeader": True,

"separator": ",",

"optimizePerformance": False,

},

connection\_type="s3",

format="csv",

connection\_options={

"paths": ["s3://chaitealatte/Input/caFloodEventsDayLevel.csv"],

"recurse": True,

},

transformation\_ctx="Floodinput\_node1",

)

# Script generated for node Change Schema

ChangeSchema\_node1683402227034 = ApplyMapping.apply(

frame=insuranceinput\_node1683401737069,

mappings=[

("id", "string", "fima\_record\_id", "string"),

("policyEffectiveDate", "string", "policy\_date", "date"),

("policyYear", "string", "year", "int"),

("policyCost", "string", "policy\_cost", "int"),

("policyCount", "string", "policy\_count", "int"),

("fips\_code", "string", "fips\_code", "int"),

("reportedZipCode", "string", "zip\_code", "int"),

("floodZone", "string", "flood\_zone", "string"),

("latitude", "string", "lat", "float"),

("longitude", "string", "lng", "float"),

("occupancyType", "string", "occupancy\_type", "string"),

("floorsCount", "string", "floors", "int"),

(

"totalBuildingInsuranceCoverage",

"string",

"total\_building\_insurance\_coverage",

"float",

),

(

"totalContentsInsuranceCoverage",

"string",

"total\_contents\_insurance\_coverage",

"float",

),

],

transformation\_ctx="ChangeSchema\_node1683402227034",

)

# Script generated for node Change Schema

ChangeSchema\_node1683356805655 = ApplyMapping.apply(

frame=weatherinput\_node1683356733176,

mappings=[

("\_c0", "string", "stationID", "string"),

("\_c1", "string", "date", "date"),

("PRCP", "string", "PRCP", "float"),

("SNOW", "string", "SNOW", "float"),

("SNWD", "string", "SNWD", "float"),

("TMAX", "string", "TMAX", "float"),

("TMIN", "string", "TMIN", "float"),

("lat", "string", "lat", "float"),

("lng", "string", "lng", "float"),

("county\_name", "string", "county\_name", "string"),

("fips\_code", "string", "fips\_code", "float"),

],

transformation\_ctx="ChangeSchema\_node1683356805655",

)

# Script generated for node ApplyMapping

ApplyMapping\_node2 = ApplyMapping.apply(

frame=Floodinput\_node1,

mappings=[

("event\_id", "string", "event\_id", "int"),

("cz\_name\_str", "string", "county\_name", "string"),

("begin\_location", "string", "location", "float"),

("begin\_date", "string", "date", "timestamp"),

("event\_type", "string", "event\_type", "string"),

("cz\_fips", "string", "fips\_code", "int"),

("flood\_cause", "string", "flood\_cause", "string"),

("begin\_lat", "string", "lat", "float"),

("begin\_lon", "string", "lng", "float"),

("flood\_impact\_in\_days", "string", "flood\_impact\_days", "string"),

],

transformation\_ctx="ApplyMapping\_node2",

)

# Script generated for node Derived Column

DerivedColumn\_node1683402884141 = ChangeSchema\_node1683402227034.gs\_derived(

colName="insurance\_coverage",

expr="total\_building\_insurance\_coverage+total\_contents\_insurance\_coverage",

)

# Script generated for node Derived Column- TEMP

DerivedColumnTEMP\_node1683356973901 = ChangeSchema\_node1683356805655.gs\_derived(

colName="TEMP", expr="(TMAX+TMIN)/2"

)

# Script generated for node drop field

dropfield\_node1683323481989 = DropFields.apply(

frame=ApplyMapping\_node2,

paths=[

"event\_id",

"county\_name",

"location",

"flood\_cause",

"lat",

"flood\_impact\_days",

"event\_type",

],

transformation\_ctx="dropfield\_node1683323481989",

)

# Script generated for node drop field

dropfield\_node1683323859245 = DropFields.apply(

frame=ApplyMapping\_node2,

paths=["county\_name"],

transformation\_ctx="dropfield\_node1683323859245",

)

# Script generated for node SQL Query- groupby

SqlQuery0 = """

SELECT fips\_code, year, SUM(LOG(policy\_cost)) AS log\_cost, SUM(LOG(`insurance\_coverage`)) AS log\_insured, first(policy\_count) AS policy\_count

FROM myDataSource

GROUP BY fips\_code, year;

"""

SQLQuerygroupby\_node1683402963169 = sparkSqlQuery(

glueContext,

query=SqlQuery0,

mapping={"myDataSource": DerivedColumn\_node1683402884141},

transformation\_ctx="SQLQuerygroupby\_node1683402963169",

)

# Script generated for node Drop Fields

DropFields\_node1683420277249 = DropFields.apply(

frame=DerivedColumn\_node1683402884141,

paths=["policy\_count", "insurance\_coverage"],

transformation\_ctx="DropFields\_node1683420277249",

)

# Script generated for node Aggregate

Aggregate\_node1683357104556 = sparkAggregate(

glueContext,

parentFrame=DerivedColumnTEMP\_node1683356973901,

groups=["date", "fips\_code"],

aggs=[["prcp", "avg"], ["snow", "avg"], ["snwd", "avg"], ["temp", "avg"]],

transformation\_ctx="Aggregate\_node1683357104556",

)

# Script generated for node Drop Fields

DropFields\_node1683358189503 = DropFields.apply(

frame=DerivedColumnTEMP\_node1683356973901,

paths=["date", "prcp", "snow", "snwd", "tmax", "tmin", "county\_name", "temp"],

transformation\_ctx="DropFields\_node1683358189503",

)

# Script generated for node Derived Column

DerivedColumn\_node1683418684259 = DerivedColumnTEMP\_node1683356973901.gs\_derived(

colName="YEAR", expr="YEAR(date)"

)

# Script generated for node Aggregate

Aggregate\_node1683323508594 = sparkAggregate(

glueContext,

parentFrame=dropfield\_node1683323481989,

groups=["date", "fips\_code"],

aggs=[["date", "count"]],

transformation\_ctx="Aggregate\_node1683323508594",

)

# Script generated for node Drop Duplicates

DropDuplicates\_node1683358233703 = DynamicFrame.fromDF(

DropFields\_node1683358189503.toDF().dropDuplicates(),

glueContext,

"DropDuplicates\_node1683358233703",

)

# Script generated for node Drop Fields

DropFields\_node1683418762446 = DropFields.apply(

frame=DerivedColumn\_node1683418684259,

paths=["lat", "lng", "county\_name", "fips\_code", "temp"],

transformation\_ctx="DropFields\_node1683418762446",

)

# Script generated for node Drop Fields

DropFields\_node1683418909678 = DropFields.apply(

frame=DerivedColumn\_node1683418684259,

paths=[

"stationid",

"prcp",

"snow",

"snwd",

"tmax",

"tmin",

"lat",

"lng",

"county\_name",

"fips\_code",

"temp",

],

transformation\_ctx="DropFields\_node1683418909678",

)

# Script generated for node Drop Duplicates

DropDuplicates\_node1683418961715 = DynamicFrame.fromDF(

DropFields\_node1683418909678.toDF().dropDuplicates(),

glueContext,

"DropDuplicates\_node1683418961715",

)

# Script generated for node flood\_events

flood\_events\_node1683324035632 = glueContext.write\_dynamic\_frame.from\_options(

frame=dropfield\_node1683323859245,

connection\_type="s3",

format="csv",

connection\_options={

"path": "s3://chaitealatte/Output/Flood/flood\_events/",

"partitionKeys": [],

},

transformation\_ctx="flood\_events\_node1683324035632",

)

# Script generated for node ca\_county\_yearly\_insurance\_stats

ca\_county\_yearly\_insurance\_stats\_node1683420188848 = glueContext.write\_dynamic\_frame.from\_options(

frame=SQLQuerygroupby\_node1683402963169,

connection\_type="s3",

format="csv",

connection\_options={

"path": "s3://chaitealatte/Output/insurance/ca\_county\_yearly\_insurance\_stats/",

"partitionKeys": [],

},

transformation\_ctx="ca\_county\_yearly\_insurance\_stats\_node1683420188848",

)

# Script generated for node insurance

insurance\_node1683420376246 = glueContext.write\_dynamic\_frame.from\_options(

frame=DropFields\_node1683420277249,

connection\_type="s3",

format="csv",

connection\_options={

"path": "s3://chaitealatte/Output/insurance/insurance/",

"partitionKeys": [],

},

transformation\_ctx="insurance\_node1683420376246",

)

# Script generated for node county\_daily\_weather

county\_daily\_weather\_node1683418319772 = glueContext.write\_dynamic\_frame.from\_options(

frame=Aggregate\_node1683357104556,

connection\_type="s3",

format="csv",

connection\_options={

"path": "s3://chaitealatte/Output/weather/county\_daily\_weather/",

"partitionKeys": [],

},

transformation\_ctx="county\_daily\_weather\_node1683418319772",

)

# Script generated for node county\_daily\_flood

county\_daily\_flood\_node1683323998972 = glueContext.write\_dynamic\_frame.from\_options(

frame=Aggregate\_node1683323508594,

connection\_type="s3",

format="csv",

connection\_options={

"path": "s3://chaitealatte/Output/Flood/county\_daily\_flood/",

"partitionKeys": [],

},

transformation\_ctx="county\_daily\_flood\_node1683323998972",

)

# Script generated for node weather\_stations

weather\_stations\_node1683418606872 = glueContext.write\_dynamic\_frame.from\_options(

frame=DropDuplicates\_node1683358233703,

connection\_type="s3",

format="csv",

connection\_options={

"path": "s3://chaitealatte/Output/weather/weather\_stations/",

"partitionKeys": [],

},

transformation\_ctx="weather\_stations\_node1683418606872",

)

# Script generated for node weather

weather\_node1683418832996 = glueContext.write\_dynamic\_frame.from\_options(

frame=DropFields\_node1683418762446,

connection\_type="s3",

format="csv",

connection\_options={

"path": "s3://chaitealatte/Output/weather/weather/",

"partitionKeys": [],

},

transformation\_ctx="weather\_node1683418832996",

)

# Script generated for node date

date\_node1683418982430 = glueContext.write\_dynamic\_frame.from\_options(

frame=DropDuplicates\_node1683418961715,

connection\_type="s3",

format="csv",

connection\_options={

"path": "s3://chaitealatte/Output/weather/date/",

"partitionKeys": [],

},

transformation\_ctx="date\_node1683418982430",

)

job.commit()