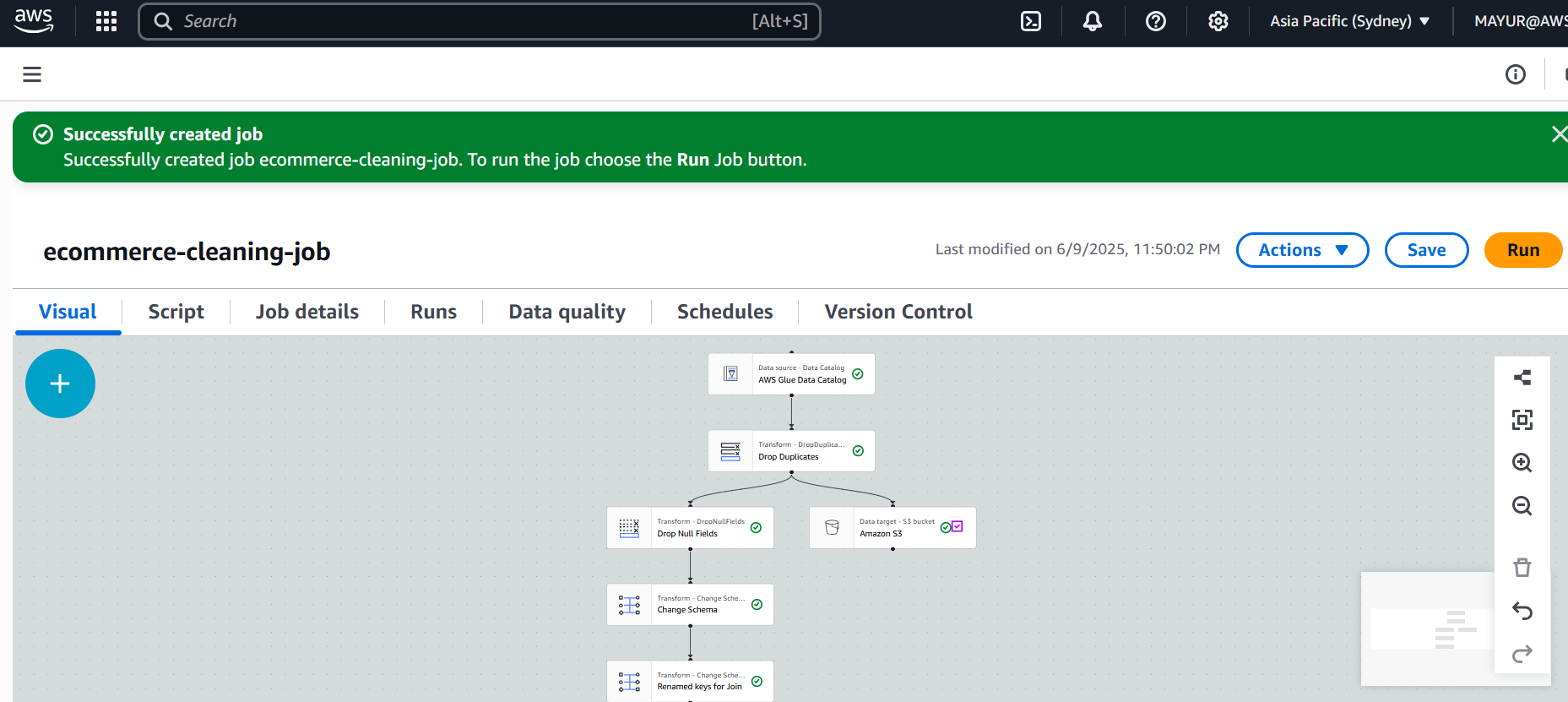
**Visual Structure**



**Script**

import sys

from awsglue.transforms import \*

from awsglue.utils import getResolvedOptions

from pyspark.context import SparkContext

from pyspark.sql import SparkSession

from awsglue.context import GlueContext

from awsglue.job import Job

from pyspark.sql.functions import to\_date

# Initialize Glue context

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

sc = SparkContext()

glueContext = GlueContext(sc)

spark = glueContext.spark\_session

job = Job(glueContext)

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

# Load sales\_data table from Glue Catalog

sales\_data = glueContext.create\_dynamic\_frame.from\_catalog(

database="ecommerce\_db", table\_name="sales\_data"

)

# Load product\_data table from Glue Catalog

product\_data = glueContext.create\_dynamic\_frame.from\_catalog(

database="ecommerce\_db", table\_name="product\_data"

)

# Convert to DataFrame for transformations

sales\_df = sales\_data.toDF()

product\_df = product\_data.toDF()

# ---------------------- ETL CLEANING ---------------------- #

# 1. Remove Duplicates

sales\_df = sales\_df.dropDuplicates()

# 2. Fill Missing/Null Values

# Replace nulls in quantity and price with 0

sales\_df = sales\_df.fillna({'quantity': 0, 'price': 0.0})

# 3. Fix Date Format (Assuming 'order\_date' is in MM/dd/yyyy)

sales\_df = sales\_df.withColumn("order\_date", to\_date("order\_date", "MM/dd/yyyy"))

# 4. Join with product data on 'product\_id'

final\_df = sales\_df.join(product\_df, on="product\_id", how="left")

# ---------------------- SAVE TO S3 ---------------------- #

# Write the cleaned data to S3 in Parquet format

final\_df.write.mode("overwrite").parquet("s3://ecommerce-raw-data/output/")

# Finish the job

job.commit()