AZURE PROJECT

RAVEENA PRABAKARAN

TIGER ANALYTICS

**TIGER ANALYTICS**

Building an ETL Pipeline using Azure Data Services

# DESCRIPTION

Use the data analytics stack to build a data pipeline using Data Factory, Databricks and Synapse.

# Problem Statement:

As a Data Engineer, you’ve been asked to access the services that can help with ETL of data in the cloud data storage to enable analytics through Synapse. In this POC, we will be collecting the data from SQL Database using ADF and the transformed data will be the source for databricks to run complex transformations and once data is analysed using Databricks, it is synced into synapse analytics data warehouse as historical dataset for enabling various analytics.

**Domain**: Analytics

# Steps for building ETL pipeline :

In this project, perform the following steps:

* Create a Resource Group.

A screenshot of a computer

Description automatically generatedResource Group 🡪 FinalProject

A screenshot of a computer

Description automatically generated

* Create a Storage account.

1. Blob Storage

A screenshot of a computer

Description automatically generated

Blob storage 🡪 rvblobfinalproject

A screenshot of a computer

Description automatically generated

1. Azure Data Lake Storage 🡪 rvadlsfinalproject

A screenshot of a computer

Description automatically generated

* Create an Azure SQL Database.

Server Sql Database

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

Sql Database --> finalprojectdb

A screenshot of a computer

Description automatically generated

* Create a data factory.

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

Data Factory 🡪 rvfinalprojectdf

A screenshot of a computer

Description automatically generated

* Configure Databricks cluster

Creating Azure Databricks

A screenshot of a computer

Description automatically generated

Databricks 🡪 rvfinalprojectdatabricks

A screenshot of a computer

Description automatically generated

Creating Cluster 🡪 rvfinalprojectcluster

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* Create Synapse analytics Data Warehouse.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

SQLPOOL

A screenshot of a computer

Description automatically generated

* Use the different Azure data factory tools to build a pipeline (SQL Database-> Copy-> ADLS Gen 2 -> Transform using Databricks -> Copy to Synapse DW).
* Use Databricks notebook for mounting ADLS Gen 2 storage, transforming the data (clean, join, filter, aggregate, pivot) and persist result to ADLS.
* Schedule and Monitor the pipeline and activity runs.

# Questions that need to be answered/Evaluation steps while building the ETL Pipeline

**Task 1: Create a dataflow with the following requirement:**

1. Create a data stream named CleaningGenreRomance and perform data cleansing on the Genre column using Derived Column and case expression. (While collecting data it was observed that some genres have spelling mistakes like romance, Romence for Romance, comedy, Comdy for

Comedy.)

Dataflow cleaninggenre

# A screenshot of a computer Description automatically generated

Datapipeline pipelinetocleangenre

A screenshot of a computer

Description automatically generated

Case Expression

A screenshot of a computer

Description automatically generated

case( Genre  =='Romence','Romance', Genre  =='Romance','Romance', Genre  =='romance','Romance', Genre  =='Drama','Drama',Genre  =='Animation','Animation', Genre  =='Fantasy','Fantasy', Genre  =='Action','Action', Genre  =='comedy','Comedy',Genre =='Comdy','Comedy',Genre =='Comedy','Comedy')

Output in ADLS:

Graphical user interface, application, table, Excel

Description automatically generated Graphical user interface, application, table, Excel

Description automatically generated

Output in Sql Database :

Graphical user interface, application, table

Description automatically generated

1. Create a data stream named CountMoviesBasedOnGenre that can calculate number of ﬁlms for each genre and store it as a separate dataset in ADLS under folder name “solution/genreCount”

Data Flow : Countfilmdataflow

A screenshot of a computer

Description automatically generated

Datapipeline: countfilmpipeline

A screenshot of a computer

Description automatically generated

Output in ADLS

Graphical user interface, text, application, email

Description automatically generated

Countoutput file

Graphical user interface, text, application, Word

Description automatically generated

1. Create a new stream named JoinMovieCountWithCleanData. Perform join operation on

CountMoviesBasedOnGenre with CleaningGenreRomance stream and store the same in the Azure SQL Database.

Graphical user interface, text, application

Description automatically generated

create table JoinMovieCountWithCleanData (

Genre varchar(200),

Nooffilms int,

Film varchar(200),

LeadStudio varchar(200),

AudienceScore int,

Profitability float ,

RottenTomatoes int,

WorldwideGross varchar(200),

Years int );

DataFlow : joindataflow

Graphical user interface, text, application

Description automatically generated

Pipeline : pipelineforinnerjoin

A screenshot of a computer

Description automatically generated

Outptut in SQL

Database:

Graphical user interface, text, application

Description automatically generated

**Task 2: Create the following activity pipeline**

1. Get the clean data from Azure SQL DB. Create an activity that can copy the data from SQLDB to ADLS Gen2.

Datapipeline :

A screenshot of a computer

Description automatically generated

Output in ADLS :

Graphical user interface, text, application

Description automatically generated

Output File:

Text

Description automatically generated

1. Create an activity that can use Azure Databricks to read the data from the ADLS Gen2 and perform rank operation on the Genre column. Ensure this activity gets activated only after the data is stored in ADLS from SQL DB. The result of Databricks must be stored in the ADLS.

account\_name = "rvadlsfinalproject"

container\_name = "adlstoadls"

input\_relative\_path = "input"

output\_relative\_path = "output"

adls\_path = 'abfss://%s@%s.dfs.core.windows.net/%s' % (container\_name, account\_name,input\_relative\_path)

adls\_output\_path= 'abfss://%s@%s.dfs.core.windows.net/%s' % (container\_name, account\_name,output\_relative\_path)

spark.conf.set("fs.azure.account.auth.type.%s.dfs.core.windows.net" %account\_name, "SharedKey")

spark.conf.set("fs.azure.account.key.%s.dfs.core.windows.net" %account\_name ,"HwZ0JciEFkqRHSNQGYECXjD+94YMR5SJMwjDJ5E/MIrgBebJa+jx7q3PWxqmKNN/H6bGjAvziNrB+AStx1PzSA==")

from pyspark.sql.types import StructType,StructField,IntegerType,StringType,DoubleType

txnSchema = StructType([

StructField("Film",StringType(),True),

StructField("Genre",StringType(),True),

StructField("LeadStudio",StringType(),True),

StructField("AudienceScore",IntegerType(),True),

StructField("Profitability",DoubleType(),True),

StructField("RottenTomatoes",IntegerType(),True),

StructField("WorldwideGross",StringType(),True),

StructField("Years",IntegerType(),True),

])

dfmovies = spark.read.option('header', 'true').option('delimiter', ',').schema(txnSchema).csv(adls\_path)

dfmovies.registerTempTable("movies")

resultmovies = spark.sql("SELECT \*, RANK () OVER (ORDER BY Genre) AS Rank\_no FROM movies ")

resultmovies.write.option('header', 'true').option('delimiter', ',').csv(adls\_output\_path)

Pipeline:

Graphical user interface, text, application

Description automatically generated

Output in ADLS :

Text

Description automatically generated with medium confidence

Graphical user interface, application, table, Excel

Description automatically generated

1. Create a ﬁnal activity that will read the output of previous activity in ADLS and store the same in Synapse.

create table movies(

Genre varchar(200),

FilmCount int,

Film varchar(200),

LeadStudio varchar(200),

AudienceScore int,

Profitability float ,

RottenTomatoes int,

WorldwideGross varchar(200),

Years int,

Rank\_no int)

Pipeline : rankpipeline

Graphical user interface, text, application

Description automatically generated

Output in Synapse:

Graphical user interface

Description automatically generated with medium confidence