Healthcare Data Engineering Project using Azure Databricks and Azure SQL Database

1. Project Overview: This project aims to build a scalable and secure data engineering pipeline for healthcare data using Microsoft Azure services. The goal is to ingest, transform, and store patient records for analysis and reporting, with optional machine learning enhancements.

2. Tools and Technologies Used:

- **Azure Databricks**: For data processing and transformation using PySpark
- **Azure Data Lake Storage (ADLS)**: To store raw patient data securely
- **Azure Data Factory (ADF)**: For orchestrating data movement (optional)
- **Azure SQL Database**: As the final destination for the cleaned and transformed data
- **PySpark / Python**: For scripting data transformations

3. Dataset Description: The dataset contains synthetic patient information with the following schema:

- Name (string)
- Age (integer)
- Gender (string)
- Blood Type (string)
- Medical Condition (string)
- Date of Admission (date)
- Doctor (string)
- Hospital (string)
- Insurance Provider (string)
- Billing Amount (double)
- Room Number (integer)
- Admission Type (string)
- Discharge Date (date)
- Medication (string)
- Test Results (string)

4. Step-by-Step Process:

Step 1: Data Ingestion

- Mounted Azure Data Lake Storage container using dbutils.fs.mount

- Loaded the healthcare_dataset.csv file into a PySpark DataFrame

```
▶ ✓ Yesterday (16s)                                    1

dbutils.fs.mount(
    source=f"wasbs://bcproject4@divyastudentstorage.blob.core.windows.net",
    mount_point="/mnt/bcproject4/",
    extra_configs={
        f"fs.azure.account.key.divyastudentstorage.blob.core.windows.net": "xaAX//VXMH5T8jSTL5xlTIXCU0c7Z/sC9JbXzkpggH2U57cy9/AFo00d4dnGME5gZuGLf0hYuVwf+ASt8jJHOw=="
    }
)

print("Mount successful!")
```

```
Mount successful!
```

Step 2: Data Exploration

- Validated the schema and data types
- Checked for missing/null values (result: no missing values in any columns)

```
▶ ✓ 12:23 PM (17s)                                    2

# Read CSV file into DataFrame
df = spark.read.csv("/mnt/bcproject4/healthcare_dataset.csv", header=True, inferSchema=True)

# Show a sample of the data
df.show(5)

# Print schema
df.printSchema()
```

▸ (3) Spark Jobs

▸ df: pyspark.sql.dataframe.DataFrame = [Name: string, Age: integer ... 13 more fields]

```
only showing top 5 rows

root
 |-- Name: string (nullable = true)
 |-- Age: integer (nullable = true)
 |-- Gender: string (nullable = true)
 |-- Blood Type: string (nullable = true)
 |-- Medical Condition: string (nullable = true)
 |-- Date of Admission: date (nullable = true)
 |-- Doctor: string (nullable = true)
 |-- Hospital: string (nullable = true)
 |-- Insurance Provider: string (nullable = true)
 |-- Billing Amount: double (nullable = true)
 |-- Room Number: integer (nullable = true)
 |-- Admission Type: string (nullable = true)
 |-- Discharge Date: date (nullable = true)
 |-- Medication: string (nullable = true)
 |-- Test Results: string (nullable = true)
```

```
▶ ✓ 12:23 PM (4s)                                    3

from pyspark.sql.functions import col, sum as _sum

# Count nulls in each column
null_counts = df.select([_sum(col(c).isNull().cast("int")).alias(c) for c in df.columns])
null_counts.show()
```

▸ (2) Spark Jobs

▸ null_counts: pyspark.sql.dataframe.DataFrame = [Name: long, Age: long ... 13 more fields]

Name	Age	Gender	Blood Type	Medical Condition	Date of Admission	Doctor	Hospital	Insurance Provider	Billing Amount	Room Number	Admission Type	Discharge Date	Medication	Test Results
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Step 3: Data Transformation in Databricks

- Added derived fields:

- LengthOfStay: Number of days between admission and discharge
- HighBillingFlag: Flag patients with billing > $10,000
- AgeGroup: Categorized patients as Child (<18), Adult (18-59), or Senior (60+)

```
✓ 12:23 PM (1s)                                                    4

from pyspark.sql.functions import col, datediff, when

# Start with original DataFrame (since there are no nulls)
df_transformed = df

# Add Length of Stay (in days)
df_transformed = df_transformed.withColumn(
    "LengthOfStay",
    datediff(col("Discharge Date"), col("Date of Admission"))
)

# Flag high billing patients (> $10,000)
df_transformed = df_transformed.withColumn(
    "HighBillingFlag",
    (col("Billing Amount") > 10000).cast("int")
)

# Categorize patients based on Age
df_transformed = df_transformed.withColumn(
    "AgeGroup",
    when(col("Age") < 18, "Child")
    .when((col("Age") >= 18) & (col("Age") < 60), "Adult")
    .otherwise("Senior")
)

# Show result
df_transformed.show(5)

▸ (1) Spark Jobs
▸ df_transformed: pyspark.sql.dataframe.DataFrame = [Name: string, Age: integer ... 16 more fields]
```

Step 4: Data Export to Azure SQL Database

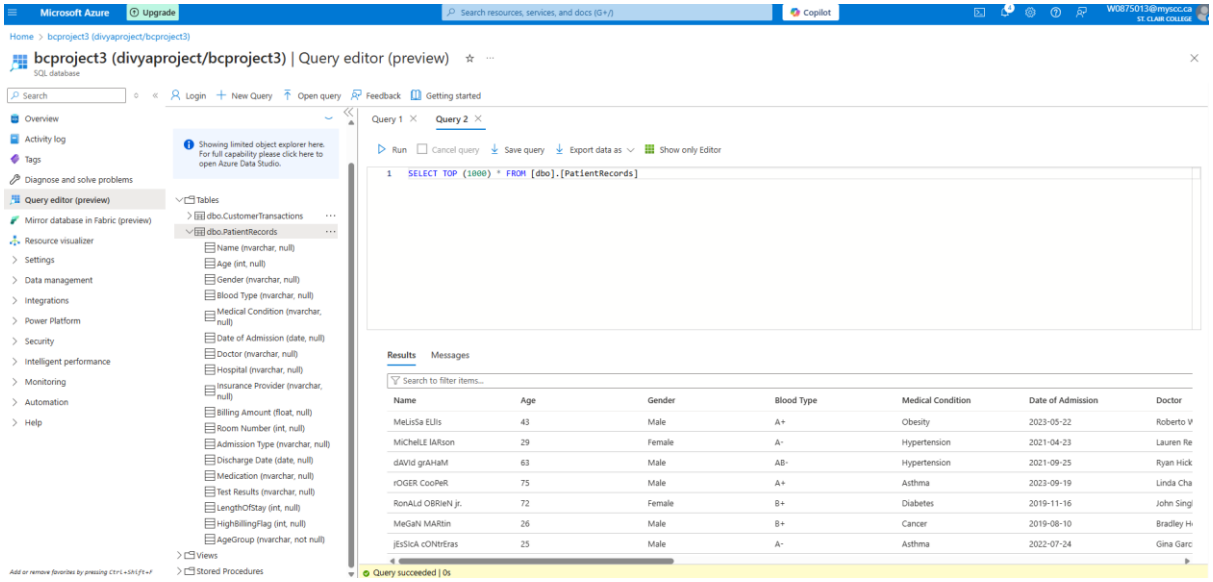
- Established JDBC connection to Azure SQL Database using SQL Server driver
- Wrote the transformed DataFrame into dbo.PatientRecords table using .write.jdbc()

jdbc_url = "jdbc:sqlserver://divyaproject.database.windows.net:1433;database=”DatabaseName"

connection_properties = {
"user": "YourSQLUsername",
"password": "YourSQLPassword",
"driver": "com.microsoft.sqlserver.jdbc.SQLServerDriver"
}

Outcome:

- Successfully created an end-to-end data pipeline for healthcare data
- Cleaned and transformed data was securely stored in Azure SQL Database


Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
Copilot
W0875013@mysc.ca
ST CLAIR COLLEGE
Home > bcproject3 (divyaproject/bcproject3)
bcproject3 (divyaproject/bcproject3) | Query editor (preview)
SQL database
Search
Overview
Activity log
Tags
Diagnose and solve problems
Query editor (preview)
Mirror database in Fabric (preview)
Resource visualizer
Settings
Data management
Integrations
Power Platform
Security
Intelligent performance
Monitoring
Automation
Help
Add or remove favorites by pressing Ctrl+Shift+F
Login
New Query
Open query
Feedback
Getting started
Showing limited object explorer here. For full capability please click here to open Azure Data Studio.
Tables
dbo.CustomerTransactions
dbo.PatientRecords
Name (nvarchar, null)
Age (int, null)
Gender (nvarchar, null)
Blood Type (nvarchar, null)
Medical Condition (nvarchar, null)
Date of Admission (date, null)
Doctor (nvarchar, null)
Hospital (nvarchar, null)
Insurance Provider (nvarchar, null)
Billing Amount (float, null)
Room Number (int, null)
Admission Type (nvarchar, null)
Discharge Date (date, null)
Medication (nvarchar, null)
Test Results (nvarchar, null)
LengthOfStay (int, null)
HighBillingFlag (int, null)
AgeGroup (nvarchar, not null)
Views
Stored Procedures
Query 1
Query 2
Run
Cancel query
Save query
Export data as
Show only Editor
1 SELECT TOP (1000) * FROM [dbo].[PatientRecords]
Results
Messages
Search to filter items...
Name
Age
Gender
Blood Type
Medical Condition
Date of Admission
Doctor
MeLisSa ELIs
43
Male
A+
Obesity
2023-05-22
Roberto V
MiChelLE lARson
29
Female
A-
Hypertension
2021-04-23
Lauren Re
dAVId grAHaM
63
Male
AB-
Hypertension
2021-09-25
Ryan Hick
rOGER CooPeR
75
Male
A+
Asthma
2023-09-19
Linda Cha
RonALd OBRIeN jr.
72
Female
B+
Diabetes
2019-11-16
John Singl
MeGaN MARtin
26
Male
B+
Cancer
2019-08-10
Bradley H
jEsSIcA cONtrEras
25
Male
A-
Asthma
2022-07-24
Gina Garc
Query succeeded | 0s