**Data Transformation**

|  |
| --- |
|  |

1. Azure Data Factory(ADF) Pipeline

A computer screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Look for all tables.
2. Copy Each table into BRONZE container.
3. Data Transformation – BRONZE to SILVER
4. Data Transformation – SILVER to GOLD
5. Azure Databricks(ADB) – Code Notebooks for Data Transformation

|  |
| --- |
| Azure Databricks Instance |
|  |
|  |

|  |
| --- |
| Azure Databricks Compute Cluster |
|  |
|  |

|  |
| --- |
| Python Code To create Mount point for Azure Blob Storage Container – For Bronze Container |
| configs = {    "fs.azure.account.auth.type": "CustomAccessToken",    "fs.azure.account.custom.token.provider.class": spark.conf.get("spark.databricks.passthrough.adls.gen2.tokenProviderClassName")  }  try:    dbutils.fs.mount(      source = "abfss://bronze@storageaccmtechdemo.dfs.core.windows.net/",      mount\_point = "/mnt/bronze",      extra\_configs = configs)    print("Mount Point created successfully")  except:    print("Mount Point already exists") |
|  |

|  |
| --- |
| Python Code To create Mount point for Azure Blob Storage Container – For Silver Container |
| configs = {    "fs.azure.account.auth.type": "CustomAccessToken",    "fs.azure.account.custom.token.provider.class": spark.conf.get("spark.databricks.passthrough.adls.gen2.tokenProviderClassName")  }  try:    dbutils.fs.mount(      source = "abfss://silver@storageaccmtechdemo.dfs.core.windows.net/",      mount\_point = "/mnt/silver",      extra\_configs = configs)    print("Mount Point created successfully")  except:    print("Mount Point already exists") |
|  |

|  |
| --- |
| Python Code To create Mount point for Azure Blob Storage Container – For Gold Container |
| configs = {    "fs.azure.account.auth.type": "CustomAccessToken",    "fs.azure.account.custom.token.provider.class": spark.conf.get("spark.databricks.passthrough.adls.gen2.tokenProviderClassName")  }  try:    dbutils.fs.mount(      source = "abfss://gold@storageaccmtechdemo.dfs.core.windows.net/",      mount\_point = "/mnt/gold",      extra\_configs = configs)    print("Mount Point created successfully")  except:    print("Mount Point already exists") |
|  |

|  |
| --- |
| Python Code for Data Transformation – BRONZE to SILVER |
| from pyspark.sql.functions import from\_utc\_timestamp, date\_format  from pyspark.sql.types import TimestampType  dbutils.fs.ls("mnt/bronze/SalesLT/")  table\_name = []  for i in dbutils.fs.ls("mnt/bronze/SalesLT/"):      table\_name.append(i.name.split('/')[0])  for i in table\_name:      path = "/mnt/bronze/SalesLT/" + i + "/" + i + ".parquet"      df = spark.read.format("parquet").load(path)      column = df.columns      for col in column:          if "Date" in col or "date" in col:              df = df.withColumn(col, date\_format(from\_utc\_timestamp(df[col].cast(TimestampType()), "UTC"), "yyyy-MM-dd"))      output\_path = "/mnt/silver/SalesLT/" + i + "/"      df.write.format("delta").mode("overwrite").save(output\_path)  display(df) |
|  |

|  |
| --- |
| Python Code for Data Transformation – SILVER to GOLD |
| from pyspark.sql import SparkSession  from pyspark.sql.functions import col, regexp\_replace  dbutils.fs.ls("mnt/silver/SalesLT/")  table\_name = []  for i in dbutils.fs.ls("mnt/silver/SalesLT/"):      table\_name.append(i.name.split('/')[0])  for name in table\_name:      path = "/mnt/silver/SalesLT/" + name      print(path)      df = spark.read.format("delta").load(path)      # Get the list of column names      column\_names = df.columns      for old\_col\_name in column\_names:          # Convert column name from ColumnName to Column\_Name format          new\_col\_name = "".join(["\_" + char if char.isupper() and not old\_col\_name[i-1].isupper() else char for i, char in enumerate(old\_col\_name)]).lstrip("\_")          # Change the column name using withColumnRenamed and regexp\_replace          df = df.withColumnRenamed(old\_col\_name, new\_col\_name)      output\_path = "/mnt/gold/SalesLT/" + name + "/"      df.write.format("delta").mode("overwrite").save(output\_path)  display(df) |
|  |

1. Azure Data Factory pipeline RUNS

|  |
| --- |
|  |
|  |
|  |
|  |

1. Delta tables stored in Silver Container

|  |
| --- |
|  |

1. Delta tables stored in Gold Container

|  |
| --- |
|  |