**Nisha More 1008982**

**SPARK PRACTICAL TASK**

1. Create dataframe using CDR.txt file with following fields

**ID, CALLING\_NUM, CALLED NUMBER, START TIME, END TIME, CALL TYPE, CHARGE, CALL RESULT**

Provide appropriate data types.

# import spark session

from pyspark.sql import SparkSession

spark = SparkSession.builder.getOrCreate()

from pyspark.sql.types import \*

from pyspark.sql.functions import \*

callColumns = [

StructField("ID",StringType()),

StructField("CALLING\_NUM",LongType()),

StructField("CALLED\_NUM",LongType()),

StructField("START\_TIME",TimestampType()),

StructField("END\_TIME",TimestampType()),

StructField("CALL\_TYPE",StringType()),

StructField("CHARGE",DoubleType()),

StructField("CALL\_RESULT",StringType()),

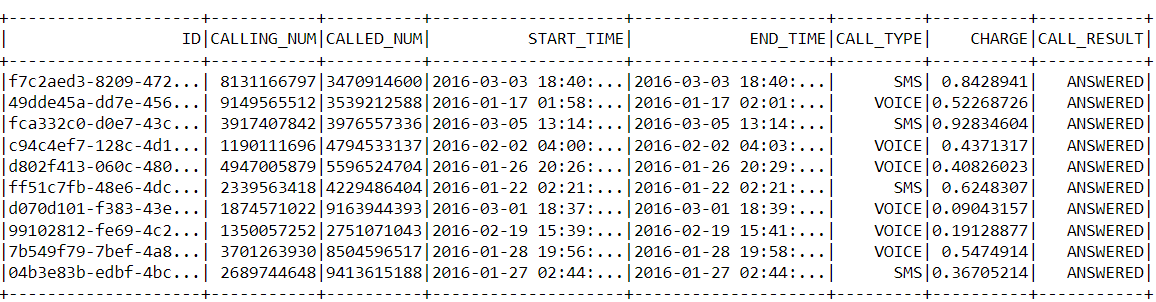
]

callschema = StructType(callColumns)

callDF = spark.read.schema(callschema).option('sep','|').csv("/user/nivassbig/Projt/CDR.txt")

callDF.printSchema()

callDF.show(10)



1. Find out the call duration of each call from above dataframe.

callDF = callDF.withColumn("CALL\_DURATION", (unix\_timestamp(col("END\_TIME")) - unix\_timestamp(col("START\_TIME"))))

callDF1 = callDF.select("CALL\_DURATION","START\_TIME","END\_TIME")

A white paper with numbers and numbers

Description automatically generated with medium confidence

1. Find out total number of calls made monthly wise in year 2016.

callDF\_2016 = callDF.filter(year("START\_TIME") == 2016)

monthly\_2016 = callDF\_2016.groupBy(year("START\_TIME").alias("Year"), month("START\_TIME").alias("Month")).count()

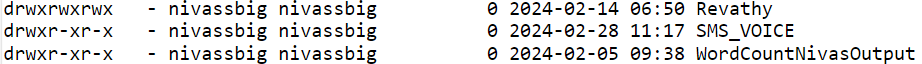
monthly\_2016.show(12)

A grid of numbers and words

Description automatically generated

1. Extract the details based on "**Voice**" and "**SMS**" and store the same in csv format partitioning based on "Voice" and "SMS".

callDF.write.option("header", True).partitionBy("CALL\_TYPE").mode("overwrite").csv("SMS\_VOICE")



1. Create dataframe used **CallDetails\_Data.csv** file according to the header making sure that schema is explicitly defined.
2. Create a new dataframe with following:
3. Remove "**–"** from phone number.
4. Round **Intl Charge** to nearest integer.
5. Customers with churn as "**TRUE**"
6. Cust serv calls greater than **0**

Save the dataframe on hdfs in parquet format.

1. From above dataframe find out phone numbers whose day calls are more than evening calls.
2. Find out phone numbers whose account length is more than **100**.
3. Create dataframe from **Amazonprodreviews.txt** by properly cleansing the data and creating appropriate schema.
4. From above dataframe filter out **product ids** and their **review scores** and save the same on Hive as external table.